

Safe Policy Optimization via Control Barrier Function-based Safety Filters

Yiting Chen Pol Mestres Emiliano Dall’Anese Jorge Cortés

Abstract—Control barrier function (CBF)-based safety filters provide a systematic way to enforce state constraints, but they can significantly alter the closed-loop dynamics induced by a nominal, stabilizing controller. In particular, the resulting safety-filtered system may exhibit undesirable behaviors including limit cycles, unbounded trajectories, and undesired equilibria. This paper develops a policy optimization framework to maximally enhance the stability properties of safety-filtered controllers. Focusing on linear systems with linear nominal controllers, we jointly parameterize the nominal feedback gain and safety-filter components, and optimize them using trajectory-based objectives computed from closed-loop rollouts. To ensure that the nominal controller remains stabilizing throughout training, we encode Lyapunov-based stability conditions as smooth scalar constraints and enforce them using robust safe gradient flow. This guarantees feasibility of the stability constraints along the optimization iterates and therefore avoids instability during training. Numerical experiments on obstacle-avoidance problems show that the proposed approach can remove asymptotically stable undesired equilibria and improve convergence behavior while maintaining forward invariance of the safe set.

I. INTRODUCTION

Ensuring safety while achieving high-performance control is a fundamental requirement in modern autonomous and cyber-physical systems, including robotics, transportation, and energy systems. In many applications, safety is formalized as the forward invariance of a prescribed set of “safe” states [1]. Control barrier functions (CBFs) provide a systematic framework for enforcing such safety constraints in control-affine systems [2], [3], commonly through *safety filters* that, given a nominal controller, minimally modify to ensure safety. However, the interaction between nominal controllers and CBF-based safety filters can significantly alter the properties of the resulting closed-loop dynamics. In particular, the safety filter may degrade the stability properties of the nominal controller. For instance, even when the nominal controller ensures global asymptotic stability of the origin, the safety-filtered system may exhibit undesirable behaviors such as asymptotically stable undesired equilibria, limit cycles, or unbounded trajectories [4]–[6]. These observations highlight the need for systematic methods to design controllers with guarantees for both safety and stability properties.

Y. Chen and E. Dall’Anese are with the Department of Electrical and Computer Engineering at Boston University; P. Mestres is with the Department of Mechanical Engineering at the California Institute of Technology; J. Cortés is with the Department of Mechanical and Aerospace Engineering at the University of California San Diego.

This work was supported by the AFOSR Award FA9550-23-1-0740.

Literature Review: Recent works have used Lyapunov-based ideas to stabilize learned controllers, including joint controller–Lyapunov design [7], [8], Lyapunov learning from demonstrations [9], and Lyapunov-regularized policy optimization [10]. However, these works do not address the interaction between nominal controllers and CBF-based safety filters. In parallel, recent results have shown that safety filters can significantly alter the qualitative behavior of the closed-loop system: even when the nominal controller is asymptotically stabilizing, the filtered dynamics may exhibit undesired equilibria, limit cycles, or unbounded trajectories [4]–[6]. Moreover, while some of these obstructions are structural, the choice of nominal controller and safety-filter design can still strongly affect their impact on the closed-loop dynamics [4], [5]. This motivates the development of systematic methods to optimize nominal controllers and safety parameters in order to improve the behavior of safety-filtered systems.

Statement of Contributions: This paper develops a systematic framework to improve the dynamical behavior of safety-filtered controllers through optimization of the nominal controller and safety-filter parameters. The main contributions are as follows:

(c1) We formulate a trajectory-based optimization problem for safety-filtered controllers in which the nominal controller, the CBF-related class- $\mathcal{K}_\infty$ function, and the safety-filter weighting matrix are jointly parameterized. This formulation is designed to shape the closed-loop dynamics induced by safety filtering, with the goal of reducing the effect of undesired equilibria and improving convergence to the desired equilibrium.

(c2) Focusing on linear systems with linear nominal controllers, we encode nominal closed-loop stability through Lyapunov-based matrix inequalities and reformulate these conditions as smooth scalar constraints using leading principal minors. This yields an optimization problem amenable to gradient-based methods while preserving an explicit characterization of stabilizing nominal controllers.

(c3) We develop a training procedure based on robust safe gradient flow and rollout-based gradient estimates. Proposition 1 shows that, when initialized from a stabilizing nominal controller, all iterates remain feasible and therefore keep the nominal controller stabilizing throughout training. The result also quantifies how gradient approximation errors affect convergence to a neighborhood of a KKT point.

(c4) We illustrate the proposed framework on representative obstacle-avoidance examples, including environments with multiple constraints and complex safe-set geometry. The numerical results show that the optimized controllers

improve convergence behavior and can eliminate asymptotically stable undesired equilibria while preserving forward invariance of the safe set.

II. PRELIMINARIES

A. Notation

We denote by $\mathbb{R}$ and $\mathbb{N}_{>0}$ the set of real and positive integers, respectively. Vectors are written in boldface, while scalar quantities are not. Given $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ and $p \in \mathbb{Z}_{>0}$, $\|\mathbf{x}\|_p$ denotes its ℓ_p norm. For a square matrix $G \in \mathbb{R}^{m \times m}$, the i -th leading principal minor is defined as the determinant of $G_{(1:i, 1:i)}$, where $G_{(1:i, 1:i)}$ denotes the submatrix formed by the first i rows and columns of G . If G is positive definite, we define $\|\mathbf{u}\|_G := \sqrt{\mathbf{u}^\top G \mathbf{u}}$. Given a continuously differentiable function $h : \mathbb{R}^n \rightarrow \mathbb{R}$, $\nabla h(\mathbf{x})$ denotes its gradient at $\mathbf{x}$. A function $\alpha : \mathbb{R} \rightarrow \mathbb{R}$ is of extended class $\mathcal{K}_\infty$ if it is strictly increasing, $\alpha(0) = 0$, and $\lim_{s \rightarrow \pm\infty} \alpha(s) = \pm\infty$. A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is L -smooth for some constant $L > 0$ if $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$.

B. Safety filters for control-affine systems

Consider the control-affine system

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}, \quad (1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $g : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ are locally Lipschitz, $\mathbf{x} \in \mathbb{R}^n$ is the state, and $\mathbf{u} \in \mathbb{R}^m$ is the control input.

Let $h : \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuously differentiable function, and define the set

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

The function h is a *control barrier function* (CBF) for the set $\mathcal{C}$ if there exists an extended class $\mathcal{K}_\infty$ function α such that, for all $\mathbf{x} \in \mathcal{C}$, there exists $\mathbf{u} \in \mathbb{R}^m$ satisfying

$$\nabla h(\mathbf{x})^\top (f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}) + \alpha(h(\mathbf{x})) \geq 0. \quad (2)$$

The main feature of CBFs is that they enable the design of safe controllers. Indeed, any locally Lipschitz controller satisfying (2) for all $\mathbf{x} \in \mathcal{C}$ renders $\mathcal{C}$ forward invariant [2, Corollary 2]. A common method to design such controller is through *safety filters*. Given a nominal controller $k : \mathbb{R}^n \rightarrow \mathbb{R}^m$ that stabilizes a desired equilibrium point (taken to be the origin henceforth without loss of generality), safety filters typically search for a controller that minimally modifies k while satisfying the CBF condition (2). This controller can be obtained at every state $\mathbf{x} \in \mathcal{C}$ by solving the following quadratic program (QP):

$$\begin{aligned} v(\mathbf{x}) = \arg \min_{\mathbf{u} \in \mathbb{R}^m} & \frac{1}{2} \|\mathbf{u} - k(\mathbf{x})\|_{G(\mathbf{x})}^2 \\ \text{s.t. } & \nabla h(\mathbf{x})^\top (f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}) + \alpha(h(\mathbf{x})) \geq 0, \end{aligned} \quad (3)$$

where $G : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times m}$ is continuously differentiable and positive definite for all $\mathbf{x} \in \mathbb{R}^n$ (this function prescribes the cost function weights at every state). Since the objective function of (3) is strictly convex, $v(\mathbf{x})$ is a singleton for each $\mathbf{x} \in \mathcal{C}$. Furthermore, if Slater's condition holds for (3) for each $\mathbf{x} \in \mathcal{C}$, then v is locally Lipschitz [11], and renders

$\mathcal{C}$ forward invariant. We also note that v can be expressed in closed-form [5]. Indeed, let $\eta(\mathbf{x}) = \nabla h(\mathbf{x})^\top (f(\mathbf{x}) + g(\mathbf{x})k(\mathbf{x})) + \alpha(h(\mathbf{x}))$. Then,

$$v(\mathbf{x}) = \begin{cases} k(\mathbf{x}), & \text{if } \eta(\mathbf{x}) \geq 0, \\ k(\mathbf{x}) + \bar{u}(\mathbf{x}), & \text{if } \eta(\mathbf{x}) < 0, \end{cases} \quad (4)$$

where

$$\bar{u}(\mathbf{x}) := -\frac{\eta(\mathbf{x}) G(\mathbf{x})^{-1} g(\mathbf{x})^\top \nabla h(\mathbf{x})}{\|g(\mathbf{x})^\top \nabla h(\mathbf{x})\|_{G(\mathbf{x})^{-1}}^2}.$$

Although the safety filter guarantees forward invariance of $\mathcal{C}$ and admits an efficient implementation, it does not, in general, preserve the stabilization properties of the nominal controller. In particular, even if the nominal controller $\mathbf{u} = k(\mathbf{x})$ is designed so that the system $\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})k(\mathbf{x})$ renders the origin globally asymptotically stable, the closed-loop system obtained by applying the safety filter may fail to retain this property. The results in [4]–[6], [12] show that the resulting closed-loop system $\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})v(\mathbf{x})$ can exhibit undesirable dynamical behaviors, such as the presence of additional equilibria (whose number, location, and stability depend on the nominal controller), unbounded trajectories, or limit cycles. Furthermore, as observed in [4], [5], different nominal controllers and different choices of the CBF can lead to different behaviors when combined with a safety filter. In favorable cases, the closed-loop system obtained from the safety filter is also globally asymptotically stable, or has only one unstable undesired equilibrium (this is the best dynamical behavior one can expect in cases where the safe set is not simply connected, due to topological obstructions [13]). In less favorable cases, the safety filter may introduce multiple undesired equilibria, some of which may even be asymptotically stable. However, as shown in [4], the choice of nominal controller or CBF can be used to reduce the region of attraction of the asymptotically stable undesired equilibria.

III. PROBLEM STATEMENT

Motivated by the considerations of Section II-B regarding the impact of safety filters on dynamical behavior, we seek to determine a nominal controller that leads to a closed-loop system with favorable dynamical properties. By this, we mean ensuring the number of undesired equilibria is minimized, preventing the emergence of asymptotically stable undesired equilibria, and maximizing the region of attraction of the desired equilibrium. We formalize this search by considering a parametric family of nominal controllers, candidate CBFs (i.e., continuously differentiable functions whose 0-superlevel set is $\mathcal{C}$), cost function weighting functions, and class $\mathcal{K}$ functions. Specifically, we let $\theta \in \mathbb{R}^d$ be a parameter vector, and define

$$\begin{aligned} k_\theta : \mathbb{R}^n &\rightarrow \mathbb{R}^m, & G_\theta : \mathbb{R}^n &\rightarrow \mathbb{R}^{m \times m}, \\ \alpha_\theta : \mathbb{R} &\rightarrow \mathbb{R}, & h_\theta : \mathbb{R}^n &\rightarrow \mathbb{R}, \end{aligned}$$

where k_θ is locally Lipschitz, G_θ is continuously differentiable and positive definite for all $\mathbf{x} \in \mathcal{C}$, α_θ is an extended

class $\mathcal{K}_\infty$ function, and h_θ is continuously differentiable and its zero-superlevel set is $\mathcal{C}$.

Given θ , the corresponding safety-filtered controller is

$$\begin{aligned} \mathbf{u}_\theta^*(\mathbf{x}) &= \arg \min_{\mathbf{u} \in \mathbb{R}^m} \frac{1}{2} \|\mathbf{u} - k_\theta(\mathbf{x})\|_{G_\theta(\mathbf{x})}^2 \\ \text{s.t. } \quad &\nabla h_\theta(\mathbf{x})^\top (f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}) + \alpha_\theta(h_\theta(\mathbf{x})) \geq 0. \end{aligned} \quad (5)$$

which induces the closed-loop system

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}_\theta^*(\mathbf{x}). \quad (6)$$

To capture the desirable dynamical objectives, we consider a trajectory-based cost function defined over a finite time horizon. Given a finite horizon $T > 0$ and an initial condition $\mathbf{x}_0$, let $\mathbf{x}_\theta(t; \mathbf{x}_0)$ denote the solution of (6) at time t . We define the cost function

$$\mathcal{L}(\theta, \mathbf{x}_0) = \phi(\mathbf{x}_\theta(T; \mathbf{x}_0)) + \lambda \int_0^T \psi(\mathbf{x}_\theta(t; \mathbf{x}_0)) dt, \quad (7)$$

for some $\lambda > 0$. The terminal cost $\phi(\mathbf{x}_\theta(T; \mathbf{x}_0))$ penalizes trajectories that fail to approach the desired equilibrium over the time horizon. In particular, it encourages the state to be close to the origin at time T , thereby promoting convergence to the desired equilibrium. The running cost $\int_0^T \psi(\mathbf{x}_\theta(t; \mathbf{x}_0)) dt$ penalizes deviations from the desired equilibrium along the trajectory. As a result, it discourages trajectories that remain far from the origin for extended periods of time, and promotes faster convergence. The hyper-parameter λ balances the relative importance of the terminal and running costs.

Given a distribution ξ over initial conditions, we consider the expected cost

$$\mathcal{L}(\theta) = \int_{\mathbb{R}^n} \mathcal{L}(\theta, \mathbf{x}_0) \xi(\mathbf{x}_0) d\mathbf{x}_0, \quad (8)$$

and the optimization problem

$$\min_{\theta \in \mathbb{R}^d} \mathcal{L}(\theta). \quad (9)$$

Qualitatively, problem (9) seeks parameters θ that lead to closed-loop trajectories that approach the desired equilibrium for initial conditions from a given distribution. In particular, it promotes closed-loop systems with favorable dynamical properties, such as a large region of attraction of the desired equilibrium and the absence of asymptotically stable undesired equilibria.

Our goal is to design a safe controller with favorable dynamical properties by solving (9). In particular, we aim to solve the following problem.

Problem 1: Given system (1), a safe set $\mathcal{C}$, and the parameterized safety filter (5), design an algorithm that solves (9). Additionally, in order to ensure that the origin of the closed-loop system remains asymptotically stable (even if the algorithm is terminated after a finite number of iterations), ensure that all the algorithm iterates yield a stabilizing nominal controller. $\square$

IV. ENFORCING STABILITY CONSTRAINTS

The formulation (9) in Section II-B prioritizes safety by construction, as the CBF-based filter guarantees forward invariance of the set $\mathcal{C}$. However, it does not guarantee that the nominal controller is stabilizing. As mentioned in Problem 1, to address this our goal is to select parameters θ for which the nominal controller renders the origin asymptotically stable.

In this section we explain how to enforce such stability constraints on the nominal controller. We focus on linear systems of the form

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \quad (10)$$

($f(x) = Ax$ and $g(x) = B$) and consider linear nominal controllers $k_\theta(\mathbf{x}) = -K\mathbf{x}$, where $K \in \mathbb{R}^{m \times n}$ is part of the parameter vector θ , i.e., $\theta = (K, \bar{\theta})$, for some other parameters $\bar{\theta}$ parameterizing the rest of the functions $G_{\bar{\theta}}$, $\alpha_{\bar{\theta}}$, $h_{\bar{\theta}}$.

We seek to enforce that the matrix $A - BK$ is Hurwitz. A standard approach is to introduce a Lyapunov function $V(\mathbf{x}) = \mathbf{x}^\top P\mathbf{x}$, with $P \in \mathbb{R}^{n \times n}$ positive definite, and impose the condition

$$(A - BK)^\top P + P(A - BK) \prec 0. \quad (11)$$

The condition (11) is bilinear in (K, P) and therefore non-convex. Using the standard change of variables $Y = KP$ and $Q = P^{-1}$, we obtain the equivalent linear condition

$$AQ + QA^\top - BY - Y^\top B^\top \prec 0, \quad Q \succ 0. \quad (12)$$

In the remainder of the paper, we set $h_\theta \equiv h$, and only optimize over the remaining parameters. For tractability, we adopt simple parametric forms for the functions α_θ and G_θ . Specifically, we parameterize the extended class $\mathcal{K}_\infty$ function as $\alpha_\theta(s) = \alpha s$, where $\alpha > 0$ is a scalar decision variable. This linear parameterization satisfies the $\mathcal{K}_\infty$ properties. Moreover, we note that G_θ affects $\mathcal{L}(\theta)$ only through $\mathbf{u}_\theta^*$ (as in (5)), and $\mathbf{u}_\theta^*$ depends on G_θ only via its inverse (cf. (4)). Therefore, we parameterize G_θ as $G_\theta = R^{-1}$, where $R \succ 0$. Accordingly, we consider a modified version of (9) that incorporates the above constraints:

$$\begin{aligned} &\min_{\theta=(Y,Q,R,\alpha) \in \mathbb{R}^d} \mathcal{L}(\theta) \\ \text{s.t. } &AQ + QA^\top - BY - Y^\top B^\top \prec 0, \\ &Q \succ 0, \\ &R \succ 0, \\ &\alpha > 0, \end{aligned} \quad (13)$$

where recall that the cost function is computed using the closed-loop dynamics (6). In fact, the optimization variables $\theta = (Y, Q, R, \alpha)$ affect $\mathcal{L}(\theta)$ through the following chain of dependencies: the matrices Y and Q determine the nominal controller k_θ through the relation $K = YQ^{-1}$, while the scalar α and the matrix R determine the parameterized functions $\alpha_\theta(s) = \alpha s$ and $G_\theta = R^{-1}$, respectively. These quantities jointly determine the filtered control input $\mathbf{u}_\theta^*(\mathbf{x})$ through (5). The control input $\mathbf{u}_\theta^*(\mathbf{x})$ in turn defines

the closed-loop dynamics (6), which determine the single-trajectory cost in (7). Finally, the expected loss $\mathcal{L}(\theta)$ is obtained by averaging this cost over the trajectories induced by the resulting closed-loop system.

The first three constraints in (13) are linear matrix inequalities (LMI). However, directly handling this constraint within a gradient-based learning framework is challenging, as it requires projecting onto the cone of positive definite matrices or solving a semidefinite program at each iteration. To obtain a representation that is more amenable to gradient-based optimization, we seek to replace the matrix inequality constraints with equivalent scalar algebraic conditions. To this end, we exploit a classical equivalent characterization of positive definiteness based on principal minors.

Lemma 1 ([14, Theorem 7.2.5]): Let $M \in \mathbb{R}^{n \times n}$ be a symmetric matrix. Then M is positive definite if and only if all leading principal minors of M are positive, i.e., $\det(M_{(1:i, 1:i)}) > 0$, for all $i \in [n]$. $\square$

By Lemma 1, we can equivalently replace each LMI in (13) with n inequalities based on leading principal minors. Specifically, we write

$$\Delta(Q, Y) := -AQ - QA^\top + BY + Y^\top B^\top,$$

and then (13) can be equivalently converted to

$$\begin{aligned} & \min_{\theta=(Y, Q, R, \alpha) \in \mathbb{R}^d} \mathcal{L}(\theta) \\ \text{s.t. } & \det(\Delta_{(1:i, 1:i)}(Q, Y)) > 0, \quad i \in [n], \\ & \det(Q_{(1:i, 1:i)}) > 0, \quad i \in [n], \\ & \det(R_{(1:i, 1:i)}) > 0, \quad i \in [m], \\ & \alpha > 0, \end{aligned} \quad (14)$$

where we recall that $\Delta_{(1:i, 1:i)}(Q, Y)$ is the submatrix formed by the first i rows and columns of $\Delta(Q, Y)$. Similarly, $Q_{(1:i, 1:i)}$ and $R_{(1:i, 1:i)}$ denote the leading principal submatrices of Q and R , respectively. In this form, both the objective function and the constraints are non-convex, but the constraints are expressed in a standard form that can be handled by gradient-based methods.

A critical challenge in solving (14) is that, violating the constraints may lead to ill-defined or unbounded values of the objective function $\mathcal{L}(\theta)$, as the underlying cost becomes arbitrarily large or ill-conditioned when the closed-loop system loses stability. As a result, temporary infeasibility during the optimization process can severely disrupt the algorithm and make it difficult to recover. Therefore, it is critical to employ an optimization scheme that maintains the feasibility of the constraints at every iteration, specifically one that does not leave the set of nominal stabilizing controllers. This also ensures that even if the algorithm is terminated after a finite number of iterations, the resulting parameter is feasible. This requirement motivates the algorithmic approach adopted in the next section.

Remark 1: (Alternative approach to enforcing stability): An alternative approach to enforcing stability is to require that the spectral abscissa of $A - BK$ is negative. However, this results in a non-smooth constraint, which makes it

difficult to handle within gradient-based optimization frameworks. $\square$

V. ALGORITHMIC IMPLEMENTATION

In this section, we propose an algorithm to solve (14). We start by rewriting (14) in the standard constrained optimization form

$$\min_{\theta \in \mathbb{R}^d} \mathcal{L}(\theta) \quad (15a)$$

$$\text{s.t. } C_i(\theta) \leq 0, \quad i \in [q], \quad (15b)$$

where $q = 2n + m + 1$ and $C_i(\theta) = -\det(\Delta_{(1:i, 1:i)}(Q, Y)) + \epsilon_0$ for $i \in [n]$, $C_i(\theta) = -\det(Q_{(1:i, 1:i)}) + \epsilon_0$ for $i \in \{n + 1, \dots, 2n\}$, $C_i(\theta) = -\det(R_{(1:i, 1:i)}) + \epsilon_0$ for $i \in \{2n + 1, \dots, 2n + m\}$ and $C_{2n+m+1}(\theta) = -\alpha_\theta + \epsilon_0$. Here, $\epsilon_0 > 0$ is a small tolerance.

We note that (15) is in general a non-convex problem. We propose to solve it through the *robust safe gradient flow* (RSGF) method, which we adopt from [15]–[17]. Specifically, given $\delta, \gamma > 0$, RSGF generates iterates $\{\theta_k\}_{k \in \mathbb{N}}$ as follows:

$$\begin{aligned} \theta_{k+1} &= p(\theta_k) \\ p(\theta) &:= \arg \min_{y \in \mathbb{R}^d} \nabla \mathcal{L}(\theta)^\top (y - \theta) + \frac{1}{2\delta} \|y - \theta\|^2 \\ \text{s.t. } & \gamma \delta C_i(\theta) + \nabla C_i(\theta)^\top (y - \theta) + \frac{1}{2\delta} \|y - \theta\|^2 \leq 0, \quad i \in [q]. \end{aligned} \quad (16)$$

As shown in [15, Lemma 1], under a suitable choice of the parameters δ, γ , the iterates defined by (16) converge to a KKT point of (15). An additional benefit of Algorithm 16 compared to other constrained optimization algorithms such as interior-point [18] or primal-dual methods [19] is that the iterates of (16) are guaranteed to satisfy the constraints of (15) (cf. [15, Lemma 1]). This is particularly relevant in the context of problem (14), where this implies that the algorithm iterates generate stabilizing nominal controllers and positive slopes of the class- $\mathcal{K}_\infty$ function.

A. Gradient approximation and implementation

Unfortunately, in the setting considered here, the function $\mathcal{L}(\theta)$ and its gradient $\nabla \mathcal{L}(\theta)$ are difficult to obtain explicitly, since they involve an integral over all possible initial conditions. Instead, in practice we need to approximate $\nabla \mathcal{L}(\theta)$ by sampling a few initial conditions, which produces an estimate $\widehat{\nabla} \mathcal{L}(\theta)$ of the gradient. Specifically, let $\mathbf{x}_0^{(i)}$, $i \in \{1, \dots, N\}$, be sampled from the distribution ξ . For each initial condition, we simulate the closed-loop system (6) under the parameterized safety filter (5) over a finite horizon T using a time discretization with step size Δt . Let $\mathbf{x}_t^{(i)}$ denote the state at time $t\Delta t$ and θ_k be the current parameter, we compute the control $u_t^{(i)}$ using the parameterized safety filter (5) and propagate the dynamics through a forward Euler discretization: $\mathbf{x}_{t+1}^{(i)} = \mathbf{x}_t^{(i)} + \Delta t(f(\mathbf{x}_t^{(i)}) + g(\mathbf{x}_t^{(i)})u_t^{(i)})$. The cost $\mathcal{L}(\theta)$ is then approximated as

$$\widehat{\mathcal{L}}(\theta) := \frac{1}{N} \sum_{i=1}^N \left(\phi(\mathbf{x}_K^{(i)}) + \lambda \sum_{t=0}^{K-1} \psi(\mathbf{x}_t^{(i)}) \Delta t \right), \quad (17)$$

where $K = T/\Delta t$. Now, the mapping $\theta \mapsto \widehat{\mathcal{L}}(\theta)$ is defined through these closed-loop rollouts and an estimate of the gradient $\nabla \mathcal{L}(\theta)$ can be obtained by directly differentiating the mapping $\widehat{\mathcal{L}}(\theta)$.

Next, replacing $\nabla \mathcal{L}(\theta)$ with $\widehat{\nabla \mathcal{L}}(\theta)$ in (16) yields the following iterative algorithm

$$\begin{aligned} \theta_{k+1} &= \hat{p}(\theta_k) \\ \hat{p}(\theta) &:= \arg \min_{y \in \mathbb{R}^d} \widehat{\nabla \mathcal{L}}(\theta)^\top (y - \theta) + \frac{1}{2\delta} \|y - \theta\|^2 \\ \text{s.t. } \gamma \delta C_i(\theta) + \nabla C_i(\theta)^\top (y - \theta) + \frac{1}{2\delta} \|y - \theta\|^2 &\leq 0, \quad i \in [q]. \end{aligned} \quad (18)$$

We note that (18) is feasible for values of θ in the feasible set, since in that case $y = \theta$ is feasible for (18). The following result characterizes the convergence and constraint satisfaction properties of (18).

Proposition 1: (Constraint satisfaction and convergence): Suppose that for each $\theta \in \mathbb{R}^d$, $\|\widehat{\nabla \mathcal{L}}(\theta) - \nabla \mathcal{L}(\theta)\| \leq \epsilon$, for some $\epsilon > 0$. Further suppose that $\mathcal{L}$ is L -smooth and lower bounded by a constant $\mathcal{L}_{\min}$, and C_i is L_i -smooth for each $i \in [q]$. Assume also that

$$0 < \delta < \min\left\{\frac{1}{\gamma}, \frac{1}{L}, \frac{1}{L_1}, \dots, \frac{1}{L_q}\right\}. \quad (19)$$

Then, if the algorithm is initialized with $\theta_0 \in \mathcal{F} := \{\theta \in \mathbb{R}^d : C_i(\theta) \leq 0, \forall i \in [q]\}$, the sequence of iterates generated by (18) satisfies:

- 1) $\theta_k \in \mathcal{F}$ for all $k \in \mathbb{N}$;
- 2) Let $K^* \in \mathbb{N}$ be such that $K^* \geq \frac{L(\mathcal{L}(\theta_0) - \mathcal{L}_{\min})}{4\epsilon^2}$. Then, there exists $k \in [K^*]$ such that $\|\hat{p}(\theta_k) - \theta_k\| \leq \frac{4\epsilon}{L}$.

Alternatively, if $c > 0$ is such that (18) is feasible in the set $\mathcal{F}_c := \{\theta \in \mathbb{R}^d : C_i(\theta) \leq c, \forall i \in [q]\}$ and $\theta_0 \in \mathcal{F}_c$, then the sequence of iterates converges to $\mathcal{F}$.

Proof: Let us first prove item 1, which follows an argument analogous to that of [15, Lemma 1 (i)]. Let $i \in [q]$ and $k \in \mathbb{N}$. Since C_i is L_i -smooth, by [20, Lemma 1.2.3] we have

$$C_i(\theta_{k+1}) \leq C_i(\theta_k) + \nabla C_i(\theta_k)^\top (\theta_{k+1} - \theta_k) + \frac{L_i}{2} \|\theta_{k+1} - \theta_k\|^2. \quad (20)$$

Now, by definition, θ_{k+1} satisfies

$$\gamma \delta C_i(\theta_k) + \nabla C_i(\theta_k)^\top (\theta_{k+1} - \theta_k) + \frac{1}{2\delta} \|\theta_{k+1} - \theta_k\|^2 \leq 0. \quad (21)$$

By using (21) in (20) we get

$$C_i(\theta_{k+1}) \leq (1 - \gamma\delta)C_i(\theta_k) - \frac{1}{2} \left(\frac{1}{\delta} - L_i \right) \|\theta_{k+1} - \theta_k\|^2. \quad (22)$$

Since $\delta < \frac{1}{\gamma}$ and $\delta < \frac{1}{L_i}$, (22) implies that if $C_i(\theta_k) \leq 0$, then $C_i(\theta_{k+1}) \leq 0$. By induction, and since this argument is valid for all $i \in [q]$, this implies that item 1 holds.

Let us now show item 2. Since $\mathcal{L}$ is L -smooth, we have

$$\mathcal{L}(\theta_{k+1}) \leq \mathcal{L}(\theta_k) + \nabla \mathcal{L}(\theta_k)^\top (\theta_{k+1} - \theta_k) + \frac{L}{2} \|\theta_{k+1} - \theta_k\|^2.$$

By the optimality conditions of (18) (cf. [18, Thm. 12.3]), we have that any feasible $y \in \mathbb{R}^d$ satisfies

$$(y - \hat{p}(\theta_k))^\top (\widehat{\nabla \mathcal{L}}(\theta_k) + \frac{1}{\delta} (\hat{p}(\theta_k) - \theta_k)) \geq 0.$$

In particular, $y = \theta_k$ is feasible (thanks to item 1, because $\theta_k \in \mathcal{F}$, which means that $C_i(\theta_k) \leq 0$ for all $i \in [q]$ and $k \in \mathbb{N}$) and yields

$$\begin{aligned} \mathcal{L}(\theta_{k+1}) - \mathcal{L}(\theta_k) &\leq (\nabla \mathcal{L}(\theta_k) - \widehat{\nabla \mathcal{L}}(\theta_k))^\top (\theta_{k+1} - \theta_k) \\ &\quad - \left(\frac{1}{\delta} - \frac{L}{2} \right) \|\theta_{k+1} - \theta_k\|^2. \end{aligned}$$

Now, after using the Cauchy-Schwartz inequality, the bound $\|\widehat{\nabla \mathcal{L}}(\theta) - \nabla \mathcal{L}(\theta)\| \leq \epsilon$, and $\delta < \frac{1}{L}$, we get that whenever $\|\hat{p}(\theta_k) - \theta_k\| > \frac{4\epsilon}{L}$,

$$\mathcal{L}(\theta_{k+1}) - \mathcal{L}(\theta_k) < -\frac{4\epsilon^2}{L} < 0. \quad (23)$$

Now, since $\mathcal{L} \geq \mathcal{L}_{\min}$, if $K^* \in \mathbb{N}$ was such that $K^* \geq \frac{(\mathcal{L}(\theta_0) - \mathcal{L}_{\min})L}{4\epsilon^2}$ and $\|\hat{p}(\theta_k) - \theta_k\| > \frac{4\epsilon}{L}$ for all $k \in [K^*]$, from (23), we would get that $\mathcal{L}(\theta_{K^*+1}) < \mathcal{L}_{\min}$, which is a contradiction. Therefore, there exists $k \in [K^*]$ such that $\|\hat{p}(\theta_k) - \theta_k\| \leq \frac{4\epsilon}{L}$.

Finally, let us show that if that (18) is feasible in $\mathcal{F}_c$ and $\theta_0 \in \mathcal{F}_c$, then the sequence of iterates converges to the feasible set of (15). This follows from (22), by noting that for any $i \in [q]$, if $C_i(\theta_k) > 0$, then $C_i(\theta_{k+1}) - C_i(\theta_k) < 0$, which implies that the sequence converges to the feasible set of (15) (either asymptotically or in finite time). ■

The smoothness assumptions in Proposition 1 are standard in optimization, and hold if the parameter set is restricted to a compact set. From (22), we can see that the choice of δ and γ affects the maximum rate with which the constraint values can approach the boundary of the safe set. Indeed, smaller values of δ decrease the second term in (22), but increase the first term. Smaller values of $\{L_i\}_{i=1}^q$ allow a wider range of values of δ satisfying (19), which can help adjust such rate. Finally, smaller values of L lead to faster convergence as per Proposition 1 item 2.

Importantly, Proposition 1 in the context of problem (14) ensures that if the iterates are initialized with a stabilizing nominal controller, then all iterates remain stabilizing, and if it is initialized with a controller that is *close* to being stabilizing, the iterates will converge to a stabilizing one (provided that the problem remains feasible). In particular, since obtaining a small approximation error ϵ requires sampling a high number of trajectories, and the property that unstable controllers converge to stable ones is independent of ϵ , if the initial nominal controller is not stabilizing, one does not need to sample a high number of trajectories and can instead wait for the parameter iterates to become stabilizing to start sampling a high number of trajectories and eventually lead to a better approximation.

We note that as the error bound ϵ between $\widehat{\nabla \mathcal{L}}(\theta)$ and $\nabla \mathcal{L}(\theta)$ vanishes, the values of $\hat{p}(\theta)$ and $p(\theta)$ coincide (under standard constraint qualification conditions on (15) cf. [21, Thm 5.3]). Since the KKT points of (15) can be characterized as the values of θ for which $\theta = p(\theta)$ (cf. [15, Lemma 1]),

Proposition 1 item 2 ensures that for small ϵ , there exists $k \in [K]$ such that θ_k lies in a neighborhood of a KKT point of (15).

We summarize the overall training procedure in Algorithm 1.

Algorithm 1 Training via Robust Safe Gradient Flow with Trajectory Sampling

```

1: Initialize parameters  $\theta_0$ 
2: for  $k = 0, 1, \dots$  do
3:   Sample  $N$  initial conditions  $\{\mathbf{x}_0^{(i)}\}_{i=1}^N \sim \xi$ 
4:   for each  $i = 1, \dots, N$  do
5:     Set  $\mathbf{x}_0^{(i)} = \mathbf{x}_0^{(i)}$ 
6:     for  $t = 0, \dots, K - 1$  do
7:       Compute control  $u_t^{(i)}$  via (5) with current state
          $\mathbf{x}_t^{(i)}$  and parameter  $\theta_k$ 
8:       Propagate dynamics:
         
$$\mathbf{x}_{t+1}^{(i)} = \mathbf{x}_t^{(i)} + \Delta t (f(\mathbf{x}_t^{(i)}) + g(\mathbf{x}_t^{(i)})u_t^{(i)})$$

9:     end for
10:  end for
11:  Compute approximated loss  $\hat{\mathcal{L}}(\theta_k)$  and gradient
     $\nabla \hat{\mathcal{L}}(\theta_k)$ 
12:  Update parameters:  $\theta_{k+1} = \hat{p}(\theta_k)$  (cf. (18))
13: end for
```

B. Multiple obstacles

The proposed framework naturally extends to the case of multiple obstacles. Suppose the safe set is defined as the intersection

$$\mathcal{C} = \bigcap_{j=1}^M \mathcal{C}_j, \quad \mathcal{C}_j = \{\mathbf{x} \in \mathbb{R}^n \mid h_j(\mathbf{x}) \geq 0\}.$$

A standard approach is to enforce the CBF condition for each barrier function h_j in the safety filter (5), which leads to a quadratic program with multiple constraints. While this formulation is conceptually straightforward, it becomes computationally demanding in the trajectory-based training setting considered here. Indeed, each rollout requires solving a quadratic program at every time step, and over a time horizon T with discretization step Δt , a single rollout involves $T/\Delta t$ solves. When N trajectories are sampled to approximate $\mathcal{L}(\theta)$, this results in solving a large number of quadratic programs. Moreover, in the presence of multiple constraints, the control input is defined implicitly as the optimizer of a constrained quadratic program. As a result, computing the gradient of $\mathcal{L}(\theta)$ with respect to θ requires differentiating through the solution map of this optimization problem, which further increases computational cost and complexity.

To address these issues, we adopt the same log-sum-exp relaxation as in [22] to combine all the CBFs. Specifically, we define $\tilde{h}_\beta(\mathbf{x}) = -\frac{1}{\beta} \log \left(\sum_{j=1}^M \exp(-\beta h_j(\mathbf{x})) \right)$, where $\beta > 0$ is a parameter. The following result characterizes this relaxation.

Lemma 2 ([23, p. 72]): For any $a_1, \dots, a_M \in \mathbb{R}$ and $\beta > 0$, define $\phi_\beta(a_1, \dots, a_M) := -\frac{1}{\beta} \log \left(\sum_{j=1}^M \exp(-\beta a_j) \right)$. It holds that

$$\min_{j \in [M]} a_j - \frac{\log M}{\beta} \leq \phi_\beta(a_1, \dots, a_M) \leq \min_{j \in [M]} a_j. \quad \square$$

By Lemma 2, the set

$$\hat{\mathcal{C}} := \{\mathbf{x} \in \mathbb{R}^n \mid \tilde{h}_\beta(\mathbf{x}) \geq 0\}$$

satisfies $\hat{\mathcal{C}} \subseteq \mathcal{C}$. Furthermore, as $\beta \rightarrow \infty$, $\tilde{h}_\beta(\mathbf{x})$ converges to $\min_j h_j(\mathbf{x})$, and hence $\hat{\mathcal{C}}$ approaches $\mathcal{C}$.

Using $\tilde{h}_\beta$ in place of the individual barrier functions results in a safety filter with a single constraint. This yields several important benefits. First, the number of constraints in the safety filter no longer scales with the number of obstacles, which significantly reduces the computational burden during trajectory rollouts. In particular, the control input can be computed without solving a multi-constraint quadratic program at each time step. Second, from (4), the resulting control input admits a closed-form expression. This eliminates the need to solve an optimization problem online and substantially reduces the overall computational cost. Finally, the closed-form expression makes the dependence of the control input on θ explicit. This avoids differentiating through the solution map of a quadratic program and simplifies the computation of gradients of $\hat{\mathcal{L}}(\theta)$ with respect to θ .

VI. NUMERICAL EXPERIMENTS

In this section, we illustrate the performance of the proposed approach on several planar examples. We consider a single integrator system

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} = \mathbf{u},$$

with $A = 0$ and $B = I_2$, and use linear nominal controllers of the form $\mathbf{u} = -K\mathbf{x}$. The safety filter is constructed using the parameterized CBF formulation as in (5).

In all experiments, the trajectories are generated using a forward Euler discretization with step size $\Delta t = 0.02$. The cost function is chosen as

$$\mathcal{L}(\theta, \mathbf{x}_0) = \|\mathbf{x}(T)\|_2^2 + \lambda \int_0^T \|\mathbf{x}(t)\|_2^2 dt,$$

with $\lambda = 0.1$. The expectation in (8) is approximated using Monte Carlo sampling with batch size 512, where initial conditions are sampled uniformly from the safe set.

For each experiment, we first choose an initial nominal gain K_0 and then construct (Q_0, Y_0) as follows. If the matrix $A - BK_0$ is Hurwitz, we let P_0 be the positive definite solution of the Lyapunov equation

$$(A - BK_0)^\top P_0 + P_0(A - BK_0) = -I,$$

and define

$$Q_0 = P_0^{-1}, \quad Y_0 = K_0 Q_0.$$

If the matrix $A - BK_0$ is unstable, we simply take $Q_0 = I_2$ and $Y_0 = K_0$. In all experiments, the parameter α is initialized as $\alpha_0 = 5$.

A. Invariance in a Bounded Safe Set

We first consider a bounded safe set defined as a disk,

$$\mathcal{C} = \{\mathbf{x} : r^2 - \|\mathbf{x} - \mathbf{x}_c\|^2 \geq 0\},$$

and the initial conditions are chosen uniformly on a circle inside the disk. By [5], in this setting, the best scenario that one can expect is the absence of undesired equilibria, and that $\mathcal{C}$ coincides with the region of attraction of the origin [5, Proposition 4.10].

In Fig. 1 (left), we show the trajectories obtained with the safety filter applied to the initial controller. There are two undesired equilibria located on the boundary of $\mathcal{C}$, one of which is asymptotically stable and has a region of attraction with positive measure. In Fig. 1 (right), we show the trajectories obtained after training. No undesired equilibrium is observed in this case. All trajectories remain in $\mathcal{C}$ and converge to the origin.

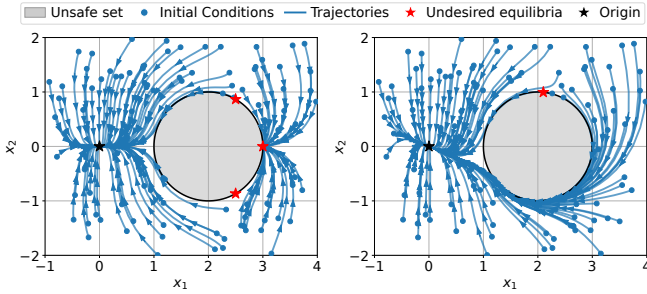


Fig. 1. State trajectories generated by the initial (left) and trained (right) controllers. Gray regions indicate unsafe sets. Under the initial controller, two undesired equilibria appear on the boundary of the safe set, one of which is asymptotically stable and its region of attraction has a positive measure. After training, no undesired equilibrium is observed, and all trajectories remain within the safe set and converge to the origin.

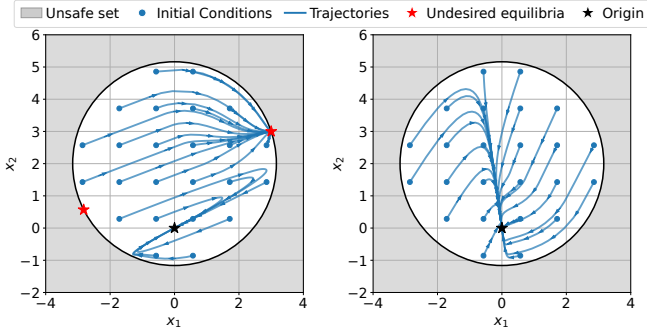


Fig. 2. State trajectories generated by the initial (left) and trained (right) controllers. Gray regions indicate unsafe sets. The initial controller induces an asymptotically stable undesirable equilibrium on the boundary of the obstacle, causing some trajectories to converge to the unsafe set. After training, the asymptotically stable undesirable equilibrium is eliminated, and the resulting controller keeps all trajectories within the safe set while yielding improved convergence behavior.

B. Avoidance of a Single Obstacle

We next consider a single circular obstacle centered at $\mathbf{x}_c$ with radius r , and define the safe set as

$$\mathcal{C} = \{\mathbf{x} : r^2 - \|\mathbf{x} - \mathbf{x}_c\|^2 \leq 0\}.$$

The initial conditions are chosen uniformly on a circle enclosing the obstacle. By [4] and [5, Proposition 4.9], for linear nominal controllers with safety filter, we can not prevent the emergence of undesired equilibria, and the best scenario that one can expect is the presence of a single unstable undesired equilibrium.

In Fig. 2 (left), we show the trajectories obtained with the safety filter applied to the initial controller. There is one undesired equilibrium located on the boundary of the obstacle, which is asymptotically stable and has a region of attraction with positive measure. The trajectories initialized in this region converge to this equilibrium, while all remaining trajectories converge to the origin. After training, as shown in Fig. 2 (right), all trajectories remain in $\mathcal{C}$ and converge to the origin. This behavior indicates the absence of asymptotically stable undesired equilibria. Moreover, the trajectory pattern suggests the presence of a single undesired equilibrium on the boundary, which is unstable, in agreement with [4].

C. Obstacle Avoidance with Multiple Constraints

Finally, we consider a scenario with multiple obstacles. The safe set is defined as the intersection of several constraints, including multiple rotated elliptical obstacles and box constraints defining the workspace. Specifically, the safe set is given by

$$\mathcal{C} = \bigcap_{j=1}^5 \{\mathbf{x} : (\mathbf{x} - \mathbf{c}_j)^\top R_j^\top D_j R_j (\mathbf{x} - \mathbf{c}_j) - 1 \geq 0\} \cap \{\mathbf{x} : x_{\min} \leq x_1 \leq x_{\max}, y_{\min} \leq x_2 \leq y_{\max}\},$$

where $\mathbf{c}_j \in \mathbb{R}^2$ is the center of the ellipse, $R_j \in \mathbb{R}^{2 \times 2}$ is a rotation matrix, and $D_j = \text{diag}(1/a_j^2, 1/b_j^2)$ defines the shape of the ellipse with semi-axes a_j and b_j . To reduce the computational complexity of the safety filter, we use the log-sum-exp approximation of Section V-B to construct a single barrier function, with smoothing parameter $\beta = 5$. The initial conditions are sampled uniformly from the safe region.

Fig. 3 (left) shows the trajectories obtained with the safety filter applied to the initial controller. Three asymptotically stable undesired equilibria and one unstable undesired equilibrium are observed. The asymptotically stable undesired equilibria are located near the top-right corner, as well as on the boundaries of the bottom-left and middle-right obstacles. After training, as shown in Fig. 3 (right), undesired equilibria still exist, but all of them are unstable. All trajectories corresponding to the sampled initial conditions remain within $\mathcal{C}$ and converge to the origin. This result demonstrates that the proposed approach effectively eliminates stable undesired equilibria in complex multi-obstacle environments, while preserving safety and ensuring convergence to the desired equilibrium.

VII. CONCLUSIONS

This paper studied the design of safety-filtered controllers with improved dynamical properties for control-affine systems. We proposed a policy optimization framework that jointly parameterizes the nominal controller and the safety

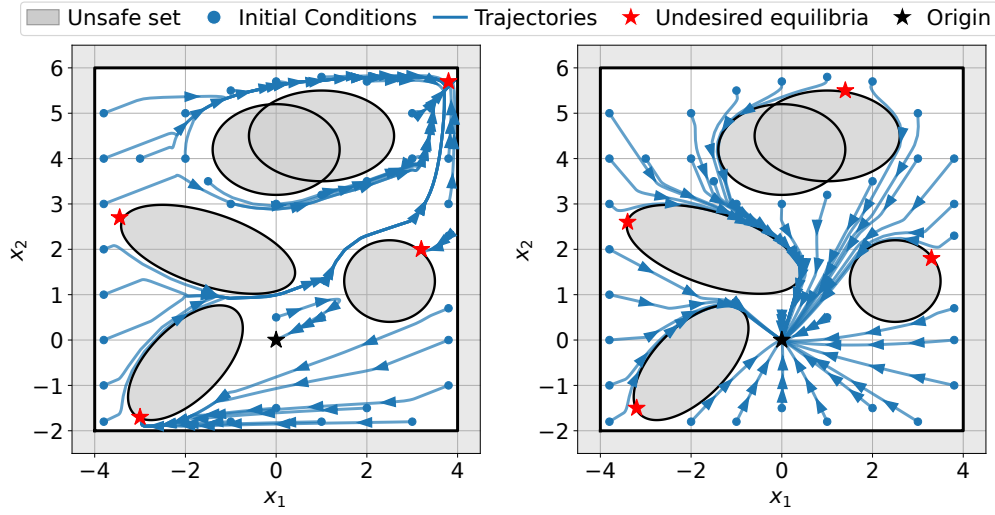


Fig. 3. State trajectories generated by the initial (left) and trained (right) controllers. Gray regions indicate unsafe sets. Given a same set of initial conditions, the trajectories under the trained controller converge to the origin while ensuring obstacle avoidance, whereas many trajectories under the initial controller converge to undesirable equilibria located near the top-right corner and on the boundaries of the ellipsoid obstacles.

filter, and optimizes them using trajectory-based objectives while enforcing Lyapunov-based stability conditions. To ensure that the stability conditions are satisfied throughout the learning process, we incorporate robust safe gradient flow with estimated policy gradient. The resulting method enables systematic exploration of controller parameters while maintaining both safety and stability guarantees. The numerical results demonstrate that the proposed approach effectively reduces undesirable equilibria and improves convergence properties, while preserving forward invariance of the safe set. Future work will focus on extending the proposed framework to more general nonlinear systems and on the design of parameterized nonlinear nominal controllers.

REFERENCES

- [1] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: theory and applications,” in *European Control Conference*, Naples, Italy, 2019, pp. 3420–3431.
- [2] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.
- [3] L. Wang, A. Ames, and M. Egerstedt, “Safety barrier certificates for collisions-free multirobot systems,” *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.
- [4] Y. Chen, P. Mestres, E. Dall’Anese, and J. Cortés, “Characterization of the dynamical properties of safety filters for linear planar systems,” in *2024 IEEE 63rd Conference on Decision and Control (CDC)*. IEEE, 2024, pp. 2397–2402.
- [5] P. Mestres, Y. Chen, E. Dall’Anese, and J. Cortés, “Control barrier function-based safety filters: characterization of undesired equilibria, unbounded trajectories, and limit cycles,” *Journal of Nonlinear Science*, 2026, to appear.
- [6] M. F. Reis, A. P. Aguilar, and P. Tabuada, “Control barrier function-based quadratic programs introduce undesirable asymptotically stable equilibria,” *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 731–736, 2021.
- [7] K. Long, J. Cortes, and N. Atanasov, “Certifying stability of reinforcement learning policies using generalized lyapunov functions,” in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
- [8] J. Wu, A. Clark, Y. Kantaros, and Y. Vorobeychik, “Neural lyapunov control for discrete-time systems,” *Advances in neural information processing systems*, vol. 36, pp. 2939–2955, 2023.
- [9] M. Mittal, M. Gallieri, A. Quaglino, S. S. M. Salehian, and J. Koutnik, “Neural lyapunov model predictive control: Learning safe global controllers from sub-optimal examples,” *arXiv preprint arXiv:2002.10451*, 2020.
- [10] B. Hejase and U. Ozguner, “Lyapunov stability regulation of deep reinforcement learning control with application to automated driving,” in *2023 American Control Conference (ACC)*. IEEE, 2023, pp. 4437–4442.
- [11] M. Alyaseen, N. Atanasov, and J. Cortés, “Continuity and boundedness of minimum-norm CBF-safe controllers,” *IEEE Transactions on Automatic Control*, vol. 70, no. 6, pp. 4148–4154, 2025.
- [12] X. Tan and D. V. Dimarogonas, “On the undesired equilibria induced by control barrier function based quadratic programs,” *Automatica*, vol. 159, p. 111359, 2024.
- [13] D. E. Koditschek, “Exact robot navigation by means of potential functions: some topological considerations,” in *IEEE Int. Conf. on Robotics and Automation*, Raleigh, NC, USA, 1987, pp. 1–6.
- [14] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.
- [15] P. Mestres, A. Marzabal, and J. Cortés, “Anytime safe reinforcement learning,” in *Learning for Dynamics and Control Conference*, N. Ozay, L. Balzano, D. Panagou, and A. Abate, Eds. Proceedings of Machine Learning Research, 2025, vol. 283, pp. 221–232.
- [16] A. Auslender, R. Shefi, and M. Teboulle, “A moving balls approximation method for a class of smooth constrained minimization problems,” *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 3232–3259, 2010.
- [17] A. Allibhoy and J. Cortés, “Control barrier function-based design of gradient flows for constrained nonlinear programming,” *IEEE Transactions on Automatic Control*, vol. 69, no. 6, pp. 3499–3514, 2024.
- [18] J. Nocedal and S. Wright, *Numerical optimization*. Berlin, Heidelberg: Springer, 2006.
- [19] D. Feijer and F. Paganini, “Stability of primal-dual gradient dynamics and applications to network optimization,” *Automatica*, vol. 46, pp. 1974–1981, 2010.
- [20] Y. Nesterov, *Lectures on Convex Optimization*, 2nd ed., ser. Springer Optimization and Its Applications. Springer International Publishing, 2018, vol. 137.
- [21] A. V. Fiacco and J. Kyparisis, “Sensitivity analysis in nonlinear programming under second order assumptions,” *Systems and Optimization*, vol. 66, pp. 74–97, 1985.
- [22] T. G. Molnar and A. D. Ames, “Composing control barrier functions for complex safety specifications,” *IEEE Control Systems Letters*, vol. 7, pp. 3615–3620, 2023.
- [23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2009.