A Convex Model Predictive Control Barrier Function Approach for Differentially Flat Systems

M. Cody Priess

Abstract—**This brief provides a novel approach for online synthesis of predictive control barrier function (CBF)-based safety filters and controllers for differentially flat, input-constrained systems. A local convex approximation to the safe set is inflated around the current system state and used to formulate a set of discrete-time affine CBF constraints. These constraints are enforced over a finite prediction horizon using a discrete-time convex model predictive control (MPC) formulation that can be solved at each sample instant. This approach is tractable for cases when the original safe set is defined by one or more concave or convex-ellipsoidal barrier functions, and is applicable even when the number or configuration of these barriers can change during runtime. This methodology is extended to enable simultaneous control Lyapunov function (CLF) enforcement within the same MPC framework, and both approaches are demonstrated in simulation.**

Index Terms—**Collision avoidance, motion planning, optimal control, robot control, vehicle dynamics.**

I. Introduction

CONTROL barrier functions (CBFs) have gained recent popularity as a rigorous means of providing safety-critical control for a variety of dynamic systems [1]. CBF-based safety filters allow for provable safety enforcement to be composed with arbitrary nominal commands, such as those resulting from real-time user input or AI-driven behaviors. This is a powerful tool for a host of problems in autonomy and robotics; however, there are limitations with the current formulations of these filters that limit their practical utility.

One of the common challenges with CBF-based safety filters is ensuring feasible solutions exist under actuation constraints or when multiple safety conditions are imposed on a system. Due to their typical implementation as online optimization-based controllers [1], infeasibility of a given optimization may not be discovered until the system reaches a configuration where no input that satisfies both the actuation constraints and safety constraints can be found. As the application space for CBF methods has grown, this so-called *viability problem* has received considerable attention.

The use of “nominal evading maneuvers” to construct a feasible barrier function for a system was demonstrated in [2]; however, unless this maneuver is selected such that the system has a closed-form solution over some interval, numerical integration would be required to approximate the system response. Similar is the backup set approach [3], where a small backup set is assumed invariant under the action of a backup controller; however, a barrier function constructed using this approach may be difficult to compute in real-time [4] and still requires user development or prior knowledge of both a backup set and backup controller.

Predictive safety filters (PSFs) constitute a broad class of potential solutions based around these basic methods. An excellent overview of PSFs (amongst many other safety filter approaches) is provided in [5]. That said, the majority of PSF approaches explored in literature either require a user-determined nominal maneuver or backup safety controller, or they require the solution to a nonlinear model predictive control (MPC) problem at runtime [6], [7]. An interesting alternative is offered in [8], which uses an iterative linearization method to address nonlinear dynamics and constraints in an MPC-based CBF problem. While convex, this approach requires multiple iterations to converge, as well as some initial guess for the state and input trajectories in order to bootstrap the method.

To address these challenges, we propose a convex MPC approach to CBF safety filters based around differential flatness [9]. To enable this, we leverage the idea of convex free-space decomposition [10] to enable real-time computation of a local convex approximation to the safe set. This approach is tractable for cases in which the desired safe set is a composition of multiple safe sets defined by either general concave or convex ellipsoidal barrier functions. By restricting our attention to differentially flat systems, we can exploit the vast array of efficient computational and analytic tools that are available for linear MPC problems. While flatness-based MPC has been explored previously in [11] and [12], to the best of our knowledge, this approach has not been extended to CBF-based safety filters or safety-critical controllers. Interestingly, when formulated in this fashion, the problem of ensuring forward invariance of a highly complex composite safe set bears significant resemblance to flatness-based path planning problems from the mobile robotics or UAS literature [13].

The contributions of this work are as follows. We develop an efficient, online approach for producing a convex approximation to a safe set defined by one or more concave or convex ellipsoidal barrier functions. This approximation permits the utilization of a broader class of barrier functions than can be handled in a convex fashion by traditional discrete-time CBF methods. Because this convex approximation is generated anew at each sample instant, this method also supports cases where the configuration or number of the constituent safe sets change during operation. We leverage this approximation along with differential flatness to develop a convex, flat MPC approach for CBF-based safety filters (FMPC-CBF). This method does not require explicit development of a backup control policy and its formulation within the barrier function to determine an achievable control action that keeps the system within the safe set. We also extend this approach to enable simultaneous satisfaction of a control Lyapunov function (CLF) stability condition within the same MPC framework (FMPC-CLF-CBF). Finally, we demonstrate the application of these methods to an example mobile robotics problem.

II. Background

A. Control Barrier Functions

Consider a Lipschitz-continuous nonlinear system

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

with $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ (we will henceforth drop the explicit dependence on t). Define a safe set $\mathcal{C}$ for the system such that $\mathcal{C} = \{x \in \mathbb{R}^n : h(x) \geq 0\}$, where $h(x) : \mathbb{R}^n \to \mathbb{R}$ is a smooth so-called “barrier” function that defines the boundary of the safe set [1]. In addition, if

$$\exists\, u \text{s.t. } \dot{h}(x,u) \geq -f_\alpha(h(x)) \tag{2}$$

with $f_\alpha(\bullet)$ an extended class-$\mathcal{K}$ function, then $\mathcal{C}$ is invariant, and $h(x)$ is a CBF [1]. Note that, from this point, we will make the common assumption that $f_\alpha(h(x)) := \alpha h(x)$, $\alpha \in \mathbb{R} > 0$.

Received 30 July 2024; revised 14 April 2025; accepted 29 May 2025. This work was supported by the MITRE Independent Research and Development Program. Recommended by Associate Editor G. Orosz.

The author is with the Department of Robotics and Autonomous Systems, The MITRE Corporation, McLean, VA 22102 USA (e-mail: mpriess@mitre.org).

Digital Object Identifier 10.1109/TCST.2025.3585086

One of the common applications of the condition in (2) is to mechanize a minimally invasive optimization-based safety controller [1]

$$\begin{aligned} u^\star = \arg\min_{u\in\mathbb{R}^m} \quad & \|u-\bar{u}\|_2^2 \\ \text{s.t.}\ & L_f h(x) + L_g h(x)u \geq -\alpha h(x) \end{aligned} \tag{3}$$

where $\bar{u}$ is some nominal control action, and $L_f h(x) + L_g h(x)u$ is the Lie derivative [14] of $h(x)$ along $\dot{x}$. The solution to (3) is generally computed by solving a convex quadratic program (QP) at every sample instant.

B. Differential Flatness

A system (1) is differentially flat if there exist outputs $y \in \mathbb{R}^m$ and smooth, invertible functions Λ, ϕ, and ψ^{-1} such that [12]

$$\begin{aligned} y &= \Lambda\left(x, u, \ldots, u^{(p)}\right) \\ x &= \phi\left(y, \dot{y}, \ldots, y^{(q-1)}\right) \\ u &= \psi^{-1}\left(y, \dot{y}, \ldots, y^{(q)}\right). \end{aligned}$$

We can then define $z := \left[y^T\ \dot{y}^T\ \cdots\ y^{(q-1),T}\right]^T$, $v := y^{(q)}$. We will refer to the vectors $z \in \mathbb{R}^{mq}$ and $v \in \mathbb{R}^m$ as the "flat states" and "flat inputs", respectively.

Proposition 1: For a differentially flat, control affine system of the form in (1), when Λ is a function of x only and has uniform relative degree (the relative degree with respect to all elements of u is the same) the mapping $u = \psi^{-1}(z, v)$ has the form $u = \psi_z^{-1}(z) + \psi_v^{-1}(z)v$.

Proof: The qth derivative of $y = \Lambda(x)$ is given by the Lie derivative [14]

$$\begin{aligned} y^{(q)} &= L_f^q \Lambda(x) + L_g L_f^{q-1} \Lambda(x) u \\ \rightarrow v &= L_f^q \Lambda(\phi(z)) + L_g L_f^{q-1} \Lambda(\phi(z)) u. \end{aligned} \tag{4}$$

Invertibility of ψ^{-1} ensures that (4) can be rearranged to give u as a function of z and v, affine in v. Further, when the system given by (1) and $\Lambda(x)$ has uniform relative degree q, then for $i = 1, 2, \ldots, q-1$, the terms $L_g L_f^{i-1} \Lambda(x) = 0$ [14]. Thus, u does not appear in any derivatives of y lower than q, and therefore does not implicitly appear in z.□

The dynamics in the flat coordinates are then described by the linear chain-of-integrators system

$$\dot{z} = Fz + Gv. \tag{5}$$

Assuming zero-order-hold inputs, the system (5) can be trivially discretized with sample period Δt to

$$z_{t+1} = F_d z_t + G_d v_t. \tag{6}$$

III. Convex Barrier Approximation

With some abuse of notation, for a barrier function $h(x)$ defined in the original system coordinates, define an analogous barrier in the flat coordinates, i.e., define $h(z) = h(\phi(z))$. Consider a composite safe set in the flat coordinates

$$\begin{aligned} \mathcal{C} &= \bigcap_{j=1}^{r} \mathcal{C}_j \\ &= \left\{z : h_j(z) \geq 0,\ \forall j \in \mathcal{J} = \{1, \ldots, r\}\right\}. \end{aligned} \tag{7}$$

Even if the individual barriers $h_j(z)$ above are simple, $\mathcal{C}$ described in (7) may be highly nonlinear and/or nonconvex. To proceed with a convex formulation for an MPC, a locally convex approximation to $\mathcal{C}$ (which we will denote as $\tilde{\mathcal{C}}$) must be found. To permit efficient computation, we restrict our attention to barriers $h_j(z)$ which are either concave or described by convex positive-semidefinite ellipsoids in z, i.e.

$$\mathcal{C}_j = \left\{z : -\left(z - p_j\right)^T W_j \left(z - p_j\right) + 1 \geq 0\right\} \tag{8}$$

with $p_j \in \mathbb{R}^{mq}$, $W_j \succeq 0$. Note that this is a broader class of barriers than can be handled in a convex fashion by traditional discrete-time CBFs [15]. Assume that out of the set of r barriers in (7), r_c barriers are concave, and r_e are convex ellipsoids.

A. Bounding Hyperplanes: Concave Barriers

Inspired by [10], for concave barrier functions $h_j(z)$ bounding hyperplanes for a local convex region may be found by first expanding an initial ellipsoid centered around the current flat state vector $d := z_t \in \mathcal{C}$, and finding the points $z_j^\star$ where this ellipsoid contacts the barriers. For each barrier $h_j(z)$, and for some $Q \succ 0$, this can be formulated as a constrained minimization problem

$$z_j^\star = \arg\min_{z\in\mathbb{R}^{mq}} \quad (z-d)^T Q (z-d) \tag{9a}$$

$$\text{s.t.} \quad h_j(z) = 0. \tag{9b}$$

Remark 1: Equation (9) only defines a convex optimization problem when $h_j(z)$ is affine; however, due to the assumptions that $h_j(z)$ is concave and $d \in \mathcal{C}$, (9) could instead be formulated as a general nonlinear convex optimization problem by replacing (9b) with $h_j(z) \leq 0$. Since $Q \succ 0$, this modified problem is strictly convex and has a globally unique solution $z_j^\star$ that lives on the boundary $h_j(z) = 0$. Unfortunately, the solution to this new problem would require satisfaction of both complementary slackness and dual feasibility conditions, and in general must be solved via a nonlinear programming approach. However, since the solution to this modified problem is unique over $h_j(z) \leq 0$ and is always on the boundary $h_j(z) = 0$ [satisfying the original constraint (9b)], then it must also be unique for the original problem (9).

The solution to (9) may be found efficiently using Lagrange multipliers. Define a Lagrangian $\mathcal{L}_j(z, \lambda) = (z-d)^T Q (z-d) + \lambda h_j(z)$ with $\lambda \in \mathbb{R}$ and an associated gradient vector $\nabla \mathcal{L}_j(z, \lambda)$. Newton's method may be used to efficiently find a root of $\nabla \mathcal{L}_j$ using d as an initial guess.

Once the points $z_j^\star$ are found for each $h_j(z)$, the tangent planes $a_j^T z - b_j = 0$ to the ellipsoid at these points are found as $a_j = 2Q(z_j^\star - d)$, $b_j = a_j^T z_j^\star$ [10], with signs defined such that $a_j^T d \leq b_j$. The components a_j and b_j are concatenated into linear constraint matrices for the concave barriers, i.e., $A_c = \left[a_1\ a_2\ \cdots\ a_{r_c}\right]^T$, $b_c = \left[b_1\ b_2\ \cdots\ b_{r_c}\right]^T$.

While [10] uses a sphere for the initial ellipsoid inflation in (9), we use an ellipsoid that is stretched along the current state derivative vector. This results in a safe set approximation $\tilde{\mathcal{C}}$ that is biased to have more safe space along the current trajectory of the system. Given (5), the autonomous component of the flat state derivative at the current instant is $\dot{z}_t = Fz_t$. For $\dot{z}_t \neq \mathbf{0}$, let $\hat{\dot{z}}_t$ be the unit vector in the direction of $\dot{z}_t$. Define an orthogonal matrix $R = \left[\hat{\dot{z}}_t\ \mathrm{Null}(\hat{\dot{z}}_t^T)\right]$ and a diagonal matrix $M = \mathrm{diag}(\beta, 1, \ldots, 1)$, where $\mathrm{Null}(\hat{\dot{z}}_t^T)$ is an orthonormal basis for the null space of $\hat{\dot{z}}_t^T$, and $\beta > 0$ is a desired ellipsoid stretch factor along the direction $\hat{\dot{z}}_t$. Note that if $\dot{z}_t = \mathbf{0}$, then setting $\beta = 1$ provides uniform inflation around the current state vector. Then, Q in (9) can be found as $Q = R^T M M R$.

An example of how changing β can help generate more useful constraints is shown in Fig. 1, using an arbitrary set of concave (circular) barriers. For larger values of β, the resulting polytopic constraint set (A_c, b_c) is biased to provide more volume along the direction of the current state derivative, at the cost of narrowing it elsewhere.

B. Bounding Hyperplanes: Ellipsoidal Barriers

While (9) describes a convex problem when $h_j(z)$ is concave, when $h_j(z)$ is convex, a different method must be used to ensure that safe bounding hyperplanes can be found. By restricting attention to cases where $h_j(z)$ describes convex ellipsoids, an efficient procedure for finding an approximately maximal-volume inscribed polytope can be developed.

Consider a barrier $h_j(z)$ of the form in (8), with $c = \mathrm{rank}(W_j)$. We begin by approximately evenly distributing a set of points $\gamma_i \in \mathbb{R}^c$, $i = 1, \ldots, l$ on the surface of the unit c-sphere. This may be done efficiently by gridding points within a unit c-cube and projecting them onto the unit c-sphere.

Once placed, the convex hull of the points γ_i is computed along with the corresponding plane normals and centroids for each of

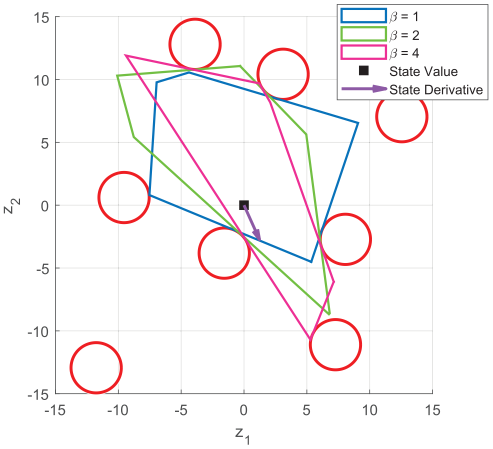
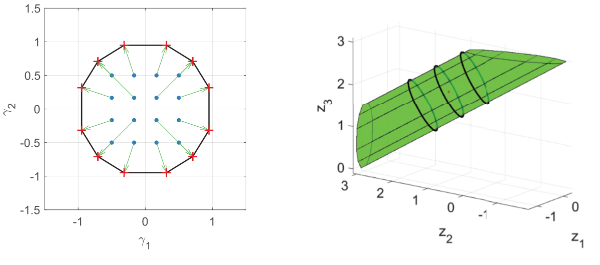


Fig. 1. Example effect of varying ellipsoid skew values β on the resulting constraint set (A_c, b_c) for a system with given initial flat state value and derivative. The red circles are concave barriers $h_j(z)$ defining a composite safe set $\mathcal{C}$. The hollow polygonal regions are the safe set approximations $\tilde{\mathcal{C}}$ for various values of β.



(a) (b)

Fig. 2. Example bounding hyperplane construction for a convex ellipsoidal barrier. (a) Points gridded in the unit c−cube (blue dots) projected onto the unit c−sphere (red crosses defining γ_i). The convex hull of the projected points (black polytope) defines $\tilde{A}$ and $\tilde{b}$. (b) Polytope from Fig. 2(a) is mapped back to the original ellipsoid in $\mathbb{R}^3$, where it defines a semi-infinite prismatic region (green).

the g facets in the hull. This information is used to describe a set of linear inequalities $(\tilde{A}, \tilde{b})$, with $\tilde{A} \in \mathbb{R}^{g\times c}$, $\tilde{b} \in \mathbb{R}^g$ such that $\tilde{A}\gamma \leq \tilde{b} \rightarrow \|\gamma\| \leq \ 1$.

For an ellipsoid matrix $W_j \geq 0$ given in (8), we can compute a singular value decomposition $W_j = V\Sigma V^T$ (due to symmetry of W_j). Let $\Sigma_c \in \mathbb{R}^{c\times c}$ be the portion of Σ corresponding to the nonzero singular values of W_j, and let V_c be V retaining only the columns corresponding to Σ_c. Then, the linear inequalities $(\tilde{A}, \tilde{b})$ defined on the unit c-sphere can be mapped to the ellipsoid (W_j, p_j) in the original z-coordinates by the affine transformation

$$A_{e,j} = \tilde{A}B_j^{-1}, \quad b_{e,j} = \tilde{b} + \tilde{A}B_j^{-1}p_j \tag{10}$$

with $B_j^{-1} = \Sigma_c^{1/2}V_c^T$. Mapping the linear inequalities instead of the points γ_i onto the ellipsoid ensures that highly skewed or degenerate ellipsoids do not cause numerical problems with the convex hull operation.

The overall process described above is illustrated in Fig. 2 for an example rank-2 ellipsoidal barrier defined in $\mathbb{R}^3$. Note that the green prismatic region [Fig. 2(b)] defined by (A_e, b_e) is strictly contained inside the original ellipsoid, illustrated by a set of slices taken along its unbounded axis (black ellipses).

C. Approximate Safe Set

Once hyperplanes have been defined for each barrier $h_j(z)$, we are left with a set of (possibly redundant) linear inequalities defined by $A = \begin{bmatrix} A_c^T & A_{e,1}^T & \cdots & A_{e,r_e}^T \end{bmatrix}^T$, $b = \begin{bmatrix} b_c^T & b_{e,1}^T & \cdots & b_{e,r_e}^T \end{bmatrix}^T$. The rows of A and elements of b define the hyperplanes (a_i^T, b_i), $i \in \mathcal{I} = \{1, \ldots, w\}$. For each of these hyperplanes, we can compute an associated barrier

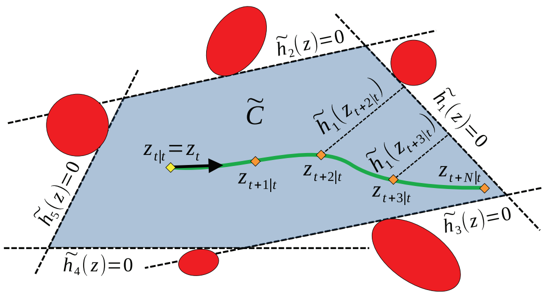


Fig. 3. Illustration of safe set approximation $\tilde{\mathcal{C}}$ and example MPC-derived trajectory.

approximation $\tilde{h}_i(z) = -a_i^T z + b_i$, and the approximate safe set will be $\tilde{\mathcal{C}} = \{z : \tilde{h}_i(z) \geq 0, \ \forall i \in \mathcal{I}\}$. As described, $\tilde{\mathcal{C}}$ will constitute a polytopic region within $\mathcal{C}$, local to and containing z_t (see Fig. 3).

When applied to a discrete-time system [e.g., (6)], the condition that must be satisfied for each $\tilde{h}_i(z)$ at each time instant t to ensure safety is $\Delta\tilde{h}_i(z_t) \geq -\alpha\tilde{h}_i(z_t)$ [7], with $\Delta\tilde{h}_i(z_t) := \tilde{h}_i(z_{t+1}) - \tilde{h}_i(z_t)$ and $0 < \alpha \leq 1$. Note that in cases where the barriers $\tilde{h}_i(z)$ have relative degree $\rho > 1$, invariance of the approximate safe set $\tilde{\mathcal{C}}$ is still ensured so long as a downstream MPC implements the above constraints over a horizon length $N \geq \rho$ [7].

IV. Safety-Critical Predictive Control

A. FMPC-CBF Safety Filter

We will adopt the MPC indexing notation from [6]. For example, $z_{t+k|t}$ would represent the prediction made at physical timestep t of the value of z at k timesteps in the future, and $z_{t:t+N|t}$ would denote the entire sequence of these predictions from $k = 0$ to $k = N$. Let $(z_t, \bar{v}_t)$ be the flat representation of the physical system states x_t and some nominal commanded input $\bar{u}_t$ measured at the current sample instant. Over a prediction horizon $k \in \mathcal{K} = \{0, \ldots, N-1\}$, a model-PSF can be realized by solving the following convex QP at each sample instant t:

$$\begin{aligned} \underset{\substack{\xi, v_{t:t+N-1|t},\\ z_{t:t+N|t}}}{\text{minimize}} \ & \frac{1}{2}\eta\| \left(v_{t|t} - \bar{v}_t\right) \|_2^2 \\ & + \frac{1}{2}\sum_{k=1}^{N-1} \|v_{t+k|t}\|_2^2 - \epsilon\xi \\ \text{s.t.} \ & z_{t|t} \ = z_t, \\ & 0 \leq \xi \leq \bar{\xi} \\ & \Delta\tilde{h}_i\left(z_{t+k|t}\right) \geq -\alpha\tilde{h}_i\left(z_{t+k|t}\right) + \xi \ \forall i \in \mathcal{I}, k \in \mathcal{K} \\ & z_{t+k+1|t} = F_d z_{t+k|t} + G_d v_{t+k|t} \ \forall k \in \mathcal{K} \\ & v_{t+k|t} \in \mathcal{V} \ \forall k \in \mathcal{K} \end{aligned} \tag{11}$$

with nominal command weight $\eta > 0$, slack variable $\xi \in \mathbb{R}$, desired slack value $\bar{\xi} > 0$, slack penalty $\epsilon > 0$, CBF "gain" $0 < \alpha \leq 1$, and $\mathcal{V}$ any other convex polyhedral set defining feasible values of v. An illustration of this MPC and how the predicted states relate to the barriers $\tilde{h}_i(z)$ is shown in Fig. 3.

First proposed for standard CBF safety filters in [16], the nontraditional slack variable ξ is of particular importance. In practice, a system using a CBF-based safety filter may still experience excursions outside the safe set or infeasibility of the QP due to, for example, restrictive input constraints, model mismatch, or sample-to-sample changes in the geometry of the safe set $\tilde{\mathcal{C}}$. These issues are not necessarily correctable simply by increasing conservatism of the CBF constraint through modifications of α, as this still results in a hard constraint on the allowable change in $\tilde{h}_i(z)$. In contrast, as formulated in (11), the variable ξ adds a slightly more conservative buffer that is tolerant of small violations while ensuring that the safety filter still strictly enforces the original CBF constraints.

In contrast to the nonlinear MPC-CBF formulation described in [6], the proposed safety filter (11) is a convex QP that may be solved efficiently using standard techniques. As in [6], recursive feasibility of (11) is difficult to demonstrate, in this case because the polytope associated with $\tilde{\mathcal{C}}$ is state-dependent and changes as the system evolves.

B. *FMPC-CLF-CBF*

Due to linearity of (6), unification of the CBF problem above with a CLF-based control law is straightforward and can be easily mechanized within a standard MPC framework. For a system (6) and set of symmetric cost matrices $Q_c \succ 0$ and $R_c \succ 0$, a terminal cost matrix $P_c \succ 0$ can be found as the unique solution to the associated discrete algebraic Riccati equation (DARE). Then, under mild conditions [17], the following convex QP locally ensures asymptotic stability of the system, subject to satisfaction of the constraints:

$$\begin{aligned}
\underset{\substack{\xi, v_{t:t+N-1|t},\\ z_{t:t+N|t}}}{\text{minimize}} \quad & \frac{1}{2}\sum_{k=0}^{N-1}\left(z_{t+k|t}^T Q_c z_{t+k|t} + v_{t+k|t}^T R_c v_{t+k|t}\right)\\
& + z_{t+N}^T P_c z_{t+N} - \epsilon\xi\\
\text{s.t.} \quad & z_{t|t} = z_t\\
& 0 \leq \xi \leq \bar{\xi}\\
& \Delta\tilde{h}_i\left(z_{t+k|t}\right) \geq -\alpha\tilde{h}_i\left(z_{t+k|t}\right) + \xi \ \forall i \in \mathcal{I}, k \in \mathcal{K}\\
& z_{t+k+1|t} = F_d z_{t+k|t} + G_d v_{t+k|t} \ \forall k \in \mathcal{K}\\
& v_{t+k|t} \in \mathcal{V} \ \forall k \in \mathcal{K}.
\end{aligned} \tag{12}$$

Assuming the constraints are satisfied, this formulation satisfies a CLF comprised of the sum of stage and terminal costs [17].

V. Example

A. *Setup*

Consider an Ackermann-type robotic ground vehicle with first-order longitudinal dynamics

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\\ \dot{s}\end{bmatrix} = \begin{bmatrix} s\cos\theta\\ s\sin\theta\\ 0\\ a_s s\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & s\\ b_s & 0\end{bmatrix}\begin{bmatrix}s_c\\ \kappa_c\end{bmatrix}. \tag{13}$$

In this model, (x, y) describe the position of the vehicle reference point (center of rear axle), θ is the heading angle of the vehicle, s is the forward speed of the vehicle, (a_s, b_s) define the first-order response of the vehicle speed, $s_c \in [s_{\min}, s_{\max}] = [3, 12]$ is the commanded speed for the vehicle, and $\kappa_c \in [\kappa_{\min}, \kappa_{\max}] = [-0.1, 0.1]$ is the commanded curvature. These box constraints on (s_c, κ_c) define the feasible input set $\mathcal{U}$.

The system (13) is differentially flat, with flat outputs $y := \begin{bmatrix}y_1 & y_2\end{bmatrix}^T = \begin{bmatrix}x & y\end{bmatrix}^T$, flat states $z := \begin{bmatrix}y_1 & y_2 & \dot{y}_1 & \dot{y}_2\end{bmatrix}^T$, and flat inputs $v := \begin{bmatrix}\ddot{y}_1 & \ddot{y}_2\end{bmatrix}^T$. The mappings ϕ, ψ, and their inverses from Section II-B are

$$\phi^{-1}(x) = \begin{bmatrix} x\\ y\\ s\cos\theta\\ s\sin\theta\end{bmatrix}, \ \phi(z) = \begin{bmatrix} y_1\\ y_2\\ \operatorname{atan2}(\dot{y}_2, \dot{y}_1)\\ \sqrt{\dot{y}_1^2 + \dot{y}_2^2}\end{bmatrix} \tag{14}$$

and

$$\begin{aligned}
\psi(x, u) &= \begin{bmatrix}\cos\theta(a_s s + b_s s_c) - \kappa_c s^2 \sin\theta\\ \sin\theta(a_s s + b_s s_c) + \kappa_c s^2\cos\theta\end{bmatrix}\\
\psi^{-1}(z, v) &= \begin{bmatrix}\frac{\ddot{y}_1\dot{y}_1 + \ddot{y}_2\dot{y}_2 - a_v\left(\dot{y}_1^2 + \dot{y}_2^2\right)}{b_v\sqrt{\dot{y}_1^2 + \dot{y}_2^2}}\\ \frac{\ddot{y}_2\dot{y}_1 - \ddot{y}_1\dot{y}_2}{\left(\dot{y}_1^2 + \dot{y}_2^2\right)^{3/2}}\end{bmatrix}.
\end{aligned} \tag{15}$$

In this case, the simple constraints defining $\mathcal{U}$ result in highly nonlinear constraints in the flat coordinates. This is a known challenge with flatness-based MPC formulations [11]; however, the constraints on κ_c and s_c are already affine in v (See Proposition 1), and at every sample instant can be approximated over the horizon length N using the current measured flat state values z_t. Applying box constraints to ensure $\psi^{-1}(z, v) \in \mathcal{U}$, rearranging terms, and fixing $z_{t:t+N|t} \equiv z_t$ yields the following polytopic constraint set defining $\mathcal{V}$:

$$\begin{aligned}
\left(s_{\min} b_v + a_v\|\breve{z}_t\|\right)\|\breve{z}_t\| &\leq \breve{z}_t^T v_{t+k|t} \ \forall k \in \mathcal{K}\\
\breve{z}_t^T v_{t+k|t} &\leq \left(s_{\max} b_v + a_v\|\breve{z}_t\|\right)\|\breve{z}_t\| \ \forall k \in \mathcal{K}\\
\kappa_{\min}\|\breve{z}_t\|^3 &\leq \breve{z}_t^T\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix} v_{t+k|t} \ \forall k \in \mathcal{K}\\
\breve{z}_t^T\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix} v_{t+k|t} &\leq \kappa_{\max}\|\breve{z}_t\|^3 \ \forall k \in \mathcal{K}
\end{aligned} \tag{16}$$

with $\breve{z}_t := \begin{bmatrix}\dot{y}_1 & \dot{y}_2\end{bmatrix}_t^T$. While this formulation only approximates the original box constraints over the MPC horizon, the approximation is always exact at the current sample instant—which is the only element of the optimized input sequence that is actually applied to the system.

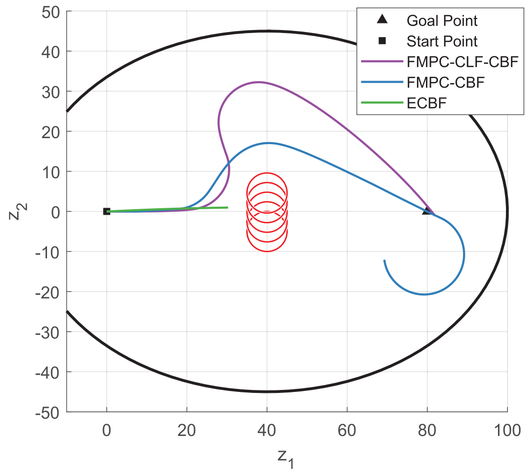


Fig. 4. Trajectories taken by the car-like vehicle described in (13) using the listed control approaches. The red circles are the sets $h_{o,i}(z) = 0$, while the large black ellipse describes the set $h_p(z) = 0$.

The control objective for the vehicle was to reach a static goal position at $\bar{z}_t \equiv \begin{bmatrix}80 & 0 & 0 & 0\end{bmatrix}^T$. A nominal controller $K \in \mathbb{R}^{2\times 4}$ was designed for the flat system such that $\bar{v}_t = -K(z_t - \bar{z}_t)$, with K designed using standard discrete LQR synthesis using diagonal cost matrices $Q_d = \operatorname{diag}(2, 2, 1, 1)$, $R_d = 3 \times I_2$. At every sample instant, the current states of the system were mapped into the flat coordinates using $\phi^{-1}(x)$ (14) and $\bar{v}_t$ was computed as shown above.

In this example, six barriers of two different types were defined for the system (see Fig. 4). The barrier $h_p(z)$ forced the vehicle to remain inside an ellipse defined in (z_1, z_2), centered at (p_1, p_2), with semi-axis lengths (l_1, l_2). The remaining five barriers $h_{o,i}(z)$ represent obstacles with radius l_o, positioned at locations $(\bar{x}_i, \bar{y}_i)$. Formally,

$$\begin{aligned}
h_p(z) &= 1 - l_2^{-2}(p_2 - z_2)^2 - l_1^{-2}(p_1 - z_1)^2\\
h_{o,i}(z) &= (\bar{x}_i - z_1)^2 + (\bar{y}_i - z_2)^2 - l_o^2.
\end{aligned} \tag{17}$$

The barriers defined by $h_{o,i}(z)$ were placed in a closely spaced, slightly asymmetric line in front of the vehicle. This type of situation is common in, for example, mobile robotics problems where obstacles are represented using occupancy grids computed online using vehicle perception sensors. Defining a circular safety barrier around each "occupied" grid element results in closely spaced barriers as seen in Fig. 4.

The system (13) was simulated for 25 s, using a 4th-order Runge-Kutta solver with a sampling time of $\Delta t = 0.05$ and parameters $a_s = -1.2$, $b_s = 1.2$. An FMPC-CBF of the form in (11) was applied after the nominal controller to enforce the barriers in (17), with parameters $\epsilon = 2000$, $\eta = 10^{-4}$, $\alpha = 0.8$, $\bar{\xi} = 5$, $(p_1, p_2) = (40, 0)$, $(l_1, l_2) =$

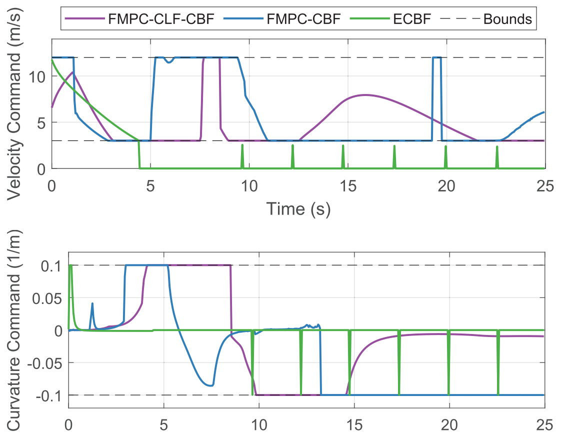


Fig. 5. Vehicle input commands and limits from the example in Fig. 4.

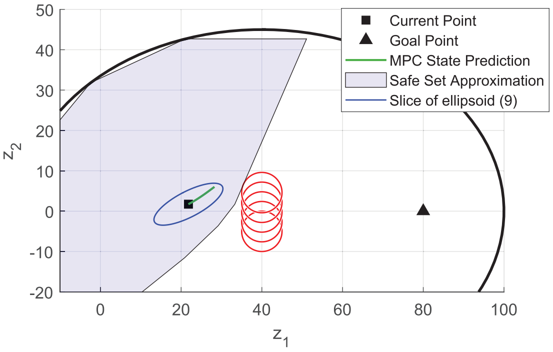


Fig. 6. Detail of the vehicle safe set approximation and predicted MPC state sequence at $t = 4$ when using the FMPC-CBF safety filter.

$(60, 45)$, $l_o = 5$, $\Delta t = 0.05$, and $N = 50$. For prototyping purposes, this MPC was constructed at every timestep using the Yalmip parser [18].

At each sample instant, $\bar{v}_t$ was computed using the nominal controller as shown above. A barrier approximation was constructed using the process outlined in Section III with $\beta = 3$, the MPC in (11) was solved as a convex QP using the Embedded COnic Solver (ECOS) [19], and the first element $v_{t|t}$ of the resulting optimized flat inputs was mapped back into the physical coordinates using $u = \psi^{-1}(z_t, v_{t|t})$ (14) before being provided to the nonlinear system (13).

For comparison, the same scenario was also executed using the FMPC-CLF-CBF controller described in (12), applying a change of variables $\tilde{z}_{t+k|t} = z_{t+k|t} - \bar{z}_t$ to permit regulation to $\bar{z}_t$. This controller used $Q_c = 10^{-5} \times \operatorname{diag}(2, 2, 1, 1)$, $R_c = 0.003 \times I_2$ and with P_c the solution to the associated DARE. All other parameter values were identical to those used in the FMPC-CBF case.

Finally, the scenario was executed with a traditional continuous-time exponential CBF (ECBF) safety filter [1] using the same flatness-based nominal controller K shown above with its outputs mapped back to u using $\bar{u} = \psi^{-1}(z_t, \bar{v}_t)$. Note that a continuous-time ECBF was chosen since the nonlinear dynamics (13) and concave barriers in (17) do not permit a convex formulation for a traditional single-step discrete CBF. The continuous-time ECBF was implemented as a convex QP enforcing nonnegativity of the barriers in (17) via the traditional Lie derivative constraints [1] with additional constraints enforcing the prescribed box bounds on s_c and κ_c and then solved at every timestep in the simulation. Since the barriers in (17) were all of relative degree 2, the gain matrix K_α [1] was chosen using pole placement as $K_\alpha = \begin{bmatrix} 0.72 & 1.70 \end{bmatrix}$.

B. Results

Results are shown in Figs. 4–6. In MATLAB R2023b, on a computer with an Intel Core i7-12800H processor, on average each iteration of the barrier approximation process took 2.7 ms, and solving the FMPC-CBF or FMPC-CLF-CBF using ECOS took 10.6 or 13.3 ms, respectively (ignoring Yalmip overhead). For both the FMPC-CBF and FMPC-CLF-CBF approaches, the vehicle was able to successfully navigate to the desired goal point while maintaining safety with respect to the defined barriers and without violating prescribed input limits (Fig. 5). Note that, due to the nonzero minimum speed constraint, the vehicle moves through and then past the goal point but begins circling back while respecting input limits.

By contrast, the traditional ECBF safety filter reached a condition where it was unable to determine a feasible control action. This is a common challenge with CBF constraints based on Lie derivatives or simple finite differences; (3) is fundamentally a greedy optimization and is highly susceptible to local minima. In this case, the QP became infeasible due to the command response staying trapped in a local minimum until the required speed command would have to drop below $s_{\min}$ in order to satisfy the safety constraint. Note that when this occurred, a fallback command of $u = \begin{bmatrix} 0, 0 \end{bmatrix}^T$ was implemented in order to handle the indeterminate outputs of the infeasible QP (Fig. 5). While not shown here, this local minimum "trap" persisted even when forcing earlier intervention by using more conservative gains for K_α or when penalizing deviations of s_c more heavily than those of κ_c.

VI. Conclusion

A model-predictive CBF approach for differentially flat systems has been demonstrated and shown to be effective at enforcing safety for an example ground vehicle system. This method permits an arbitrary number of concave or convex-quadratic barriers to be approximated by a conservative inscribed polytope, and updated in real-time as the system states evolve. The CBF enforcement problem can then be formulated as a convex MPC, which permits efficient and reliable solution in real-time. This MPC does not require a preformulated backup controller or nominal maneuver but instead directly computes an optimal trajectory within the intersection of the approximated safe set and the reachable set for the system. This approach has also been extended to enable simultaneous satisfaction of a CLF stability condition over the MPC horizon. These methods are able to produce feasible, safe commands even in cases that are intractable for a traditional continuous-time ECBF safety filter and are suitable for real-time implementation even when the precise configuration of the safe set can evolve during operation.

As discussed above, one of the major challenges with flatness-based control methods is that even trivial constraints in the original coordinates may become highly nonlinear in the flat coordinates. While our approach of linearizing constraints around the current flat state vector z_t produced viable commands, other methods may better approximate the constraints over the entire prediction horizon. In particular, the approach of [8] is able to properly enforce the original input constraints, at the cost of requiring the MPC problem to be solved multiple times at each sample-instant. However, since the problem is convex, solving it multiple times is not an insurmountable computational burden. This suggests that hybridization of a flatness-based approach to convex MPC together with an iterative method for constraint linearization as in [8] may provide an efficient solution to the input constraint problem not only for flat MPC-based CBF methods, but also for more general flatness-based trajectory planning problems.

In future work, we intend to extend the convex safe set approximation method to more general classes of barrier functions, explore alternative methods for addressing flatness-induced constraint nonlinearity, and demonstrate applications of this method to experimental systems.

Acknowledgment

Approved for Public Release; Distribution Unlimited. Public Release Case Number 24-2104.

REFERENCES

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

[2] E. Squires, P. Pierpaoli, and M. Egerstedt, "Constructive barrier certificates with applications to fixed-wing aircraft collision avoidance," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2018, pp. 1656–1661.

[3] T. Gurriet, M. Mote, A. D. Ames, and E. Feron, "An online approach to active set invariance," in *Proc. IEEE Conf. Decis. Control (CDC)*, Miami Beach, FL, USA, Dec. 2018, pp. 3592–3599.

[4] A. Singletary, A. Swann, Y. Chen, and A. D. Ames, "Onboard safety guarantees for racing drones: High-speed geofencing with control barrier functions," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2897–2904, Apr. 2022.

[5] K. P. Wabersich et al., "Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems," *IEEE Control Syst.*, vol. 43, no. 5, pp. 137–177, Oct. 2023.

[6] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *Proc. Amer. Control Conf.*, 2021, pp. 3882–3889.

[7] J. Zeng, Z. Li, and K. Sreenath, "Enhancing feasibility and safety of nonlinear model predictive control with discrete-time control barrier functions," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, Aug. 2021, pp. 6137–6144.

[8] S. Liu, J. Zeng, K. Sreenath, and C. A. Belta, "Iterative convex optimization for model predictive control with discrete-time high-order control barrier functions," in *Proc. Am. Control Conf. (ACC)*, 2023, pp. 3368–3375.

[9] R. M. Murray, M. Rathinam, and W. Sluis, "Differential flatness of mechanical control systems: A catalog of prototype systems," in *Proc. ASME Int. Mech. Eng. Congr. Expo.*, Oct. 1995, pp. 349–357.

[10] R. Deits and R. Tedrake, "Computing large convex regions of obstacle-free space through semidefinite programming," in *Proc. 11th Int. Workshop Algorithmic Found. Robot*. Cham, Switzerland: Springer, Jan. 2015, pp. 109–124.

[11] C. Kandler, S. X. Ding, T. Koenings, N. Weinhold, and M. Schultalbers, "A differential flatness based model predictive control approach," in *Proc. IEEE Int. Conf. Cont. Appl.*, Jun. 2012, pp. 1411–1416.

[12] M. Greeff and A. P. Schoellig, "Flatness-based model predictive control for quadrotor trajectory tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Jul. 2018, pp. 6740–6745.

[13] S. Liu et al., "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1688–1695, Jul. 2017.

[14] H. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[15] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation," in *Proc. Robot., Sci. Syst.*, Cambridge, MA, USA, vol. 13, 2017, pp. 1–10.

[16] C. Bass, "Soft control barrier functions," MITRE Corp., McLean, VA, USA, Tech. Rep. 23-0626, Jun. 2023.

[17] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. San Francisco, CA, USA: Nob Hill, 2009.

[18] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, Sep. 2004, pp. 284–289.

[19] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2013, pp. 3071–3076.