Systematic Design of Discrete-time Control Barrier Functions using Maximal Output Admissible Sets

Victor Freire, and Marco M. Nicotra, *Member, IEEE*

***Abstract*— This paper introduces a novel discrete-time control barrier function (DCBF) that stems directly from discrete-time set invariance theory. The proposed DCBF provides necessary and sufficient conditions for certifying control invariance and can be used to synthesize a constrained control policy. Moreover, the DCBF can be constructed for arbitrary sets of state and input constraints by taking advantage of maximal output admissible set theory. The resulting DCBF-based controller is proven to be safe and recursively feasible. Numerical examples showcase the effectiveness of the scheme by comparing it to existing constrained control approaches.**

***Index Terms*— Constrained control, Control barrier functions, Optimal control, Predictive control for linear systems, Predictive control for nonlinear systems.**

I. Introduction

CONTROL barrier functions (CBFs) have recently garnered attention from the constrained control community by serving as both a certificate of safety and a tool for synthesizing constrained control laws [1]. The principle behind CBF-based control is to select an input that is as close as possible to a nominal control action while also guaranteeing constraint enforcement. Due to their conceptual simplicity, computational efficiency, and overall performance, CBF-based controllers have been implemented successfully on a wide variety of applications [2]–[5].

The main challenge associated with this approach is that identifying CBFs for arbitrary constraint sets is challenging. As a result, it is common practice to rely on “candidate” CBFs, i.e., scalar functions that ensure constraint satisfaction when positive, but are not guaranteed to remain positive in the future. Unfortunately, this myopic approach can cause the CBF-based optimization problem to become infeasible, especially as the number of state and input constraints increases. As a result, the systematic design of CBFs is still an open research question. In [6], the authors introduce control-sharing barrier functions which are groups of CBFs that retain their validity even when superimposed. Nonetheless, finding functions with the control-sharing property is also difficult. In [7], [8], the authors show how to construct a CBF by introducing a prestabilizing controller (called the backup policy) and integrating the resulting closed-loop dynamics backwards in time to construct a valid CBF. Unfortunately, the constructed CBF may not have a closed-form expression since the ordinary differential equation is often solved numerically. Another common constrained control approach is based on barrier Lyapunov functions [9]. These approaches, while providing sufficient conditions for safety, are rarely necessary as in the case of CBFs and may lead to more conservative performance.

This paper proposes a paradigm shift to the discrete-time domain, where we leverage the well-established maximal output admissible set (MOAS) theory [10] to systematically design suitable CBFs. The main contributions are as follows: a) we define a discrete-time control barrier function (DCBF) that provides necessary and sufficient conditions for control invariance, b) we show how to systematically construct a DCBF for general nonlinear systems subject to arbitrary state and input constraints, c) we formulate a recursively feasible DCBF-based constrained controller, d) we specialize the proposed framework to the linear case, where we present closed-form expressions for all results.

The DCBFs presented in this paper differ from the discrete-time exponential CBFs (DECBFs) proposed in [11], which provide only sufficient conditions for set invariance by replacing the differential conditions of continuous-time CBFs with difference equations. Although DECBFs have been successfully implemented in discrete-time applications [12], [13], they suffer from similar drawbacks as continuous-time CBFs: finding a DECBF for arbitrary constraint sets is challenging, and resorting to candidate DECBFs can lead to infeasibility issues in the DECBF-based program.

The remainder of the paper is organized as follows: Section II provides a brief overview of the theory behind CBFs and DECBFs; Section III introduces the new DCBF formulation and the associated DCBF-based program; Section IV shows how, given arbitrary state and input constraints, DCBFs can be obtained by projecting the MOAS onto the state-space. The approach is then specialized to the linear case, where the MOAS can be obtained in closed form; Section V compares the performance of our approach with existing constrained control strategies, namely: DECBFs [11], model predictive control (MPC) [14], and command governors (CGs) [15].

Manuscript received 15 March 2023; revised 9 May 2023; accepted 31 May 2023. This work was supported by NSF-CMMI award #2046212. Recommended by Senior Editor S. Tarbouriech. (Corresponding author: Victor Freire.)

The authors are with the Department of Electrical, Computer & Energy Engineering, University of Colorado, Boulder, CO 80309 USA (email: vifr9883@colorado.edu; marco.nicotra@colorado.edu).

II. Preliminaries

This section summarizes existing results in CBF literature. Please note we modified some notation with respect to its original reference to ensure cohesiveness throughout the paper.

A. Continuous-time Control Barrier Functions

Consider a continuous-time system $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$, with f locally Lipschitz, $\mathbf{x} \in \mathbb{R}^n$, and $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^m$. Nagumo's Theorem [16] states that a compact set $\mathcal{C} \subseteq \mathbb{R}^n$ is control invariant if and only if

$$\forall \mathbf{x} \in \partial\mathcal{C}, \quad \exists \mathbf{u} \in \mathcal{U}: \quad f(\mathbf{x}, \mathbf{u}) \in \mathcal{T}_{\mathcal{C}}(\mathbf{x}), \tag{1}$$

where $\mathcal{T}_{\mathcal{C}}(\mathbf{x})$ is the tangent cone to $\mathcal{C}$ in $\mathbf{x}$ [16, Def. 3.1]. In essence, (1) states that, whenever $\mathbf{x}$ belongs to the boundary of the set $\mathcal{C}$, there exists an input $\mathbf{u} \in \mathcal{U}$ such that the vector field f does not point towards the exterior of $\mathcal{C}$. Since this condition is defined only on the boundary $\partial\mathcal{C}$ and provides no insight whenever $\mathbf{x} \in \mathrm{Int}(\mathcal{C})$, modern CBF literature developed an equivalent condition that spans the entirety of the set whenever $\mathcal{C}$ is given as the superlevel set of some function $h : \mathbb{R}^n \to \mathbb{R}$:

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) \geq 0\}. \tag{2}$$

Definition 1: [17, Def. 2] A continuously differentiable function $h \in C^1 : \mathbb{R}^n \to \mathbb{R}$ is a control barrier function (CBF) on the open set $\mathcal{D} \subseteq \mathbb{R}^n$ for the set $\mathcal{C} \subset \mathcal{D}$ satisfying (2) if there exists an extended class $\mathcal{K}$ function α such that

$$\sup_{\mathbf{u} \in \mathcal{U}} \left[\dot{h}(\mathbf{x}, \mathbf{u})\right] \geq -\alpha\big(h(\mathbf{x})\big), \quad \forall \mathbf{x} \in \mathcal{D}, \tag{3}$$

where $\dot{h}(\mathbf{x}, \mathbf{u}) = \nabla h(\mathbf{x}) f(\mathbf{x}, \mathbf{u})$.

Proposition 1: [1, Cor. 2] The set $\mathcal{C} \subset \mathbb{R}^n$ given in (2) is control invariant if and only if $h(\mathbf{x})$ is a CBF.

A CBF certifies the existence of an input $\mathbf{u}(t) \in \mathcal{U}$ that ensures $\mathbf{x}(t) \in \mathcal{C}$. A common approach for finding such an input is to use CBF-based programs [1]. However, given an arbitrary state constraint set $\mathcal{X} \subseteq \mathbb{R}^n$, the literature provides few methods for systematically designing CBFs. In fact, even when $\mathcal{X}$ is not control invariant, it is common practice to take $\mathcal{C} = \mathcal{X}$ and design a function $h(\mathbf{x})$ that satisfies (2), but may not satisfy (3). We call such a function a *candidate* CBF [8]. Since candidate CBFs cannot certify control invariance, they are vulnerable to infeasibility issues when used in CBF-based programs, especially under input constraints.

In this paper, we show that the discrete-time domain offers a new perspective for the systematic design of CBFs given arbitrary sets of state and input constraints. We will do so by proposing a new discrete-time CBF formulation, which differs from the one currently found in the literature and summarized in the next subsection.

B. Discrete-time Exponential Control Barrier Functions

Consider a discrete-time system

$$\mathbf{x}^+ = f(\mathbf{x}, \mathbf{u}), \tag{4}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^m$, and $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a continuous function. The following definition was proposed as an extension to continuous-time CBFs.

Definition 2: [11, Def. 4] A continuous function $h \in C^0 : \mathbb{R}^n \to \mathbb{R}$ is a discrete-time exponential control barrier function (DECBF) for the closed set $\mathcal{C} \subseteq \mathbb{R}^n$ satisfying (2) if there exists a positive scalar $\lambda \in (0, 1]$ such that

$$\sup_{\mathbf{u} \in \mathcal{U}} \left[\Delta h(\mathbf{x}, \mathbf{u})\right] \geq -\lambda h(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{C}, \tag{5}$$

where $\Delta h(\mathbf{x}, \mathbf{u}) \triangleq h\big(f(\mathbf{x}, \mathbf{u})\big) - h(\mathbf{x})$.

Proposition 2: [11] The closed set $\mathcal{C} \subseteq \mathbb{R}^n$ given in (2) is control invariant if $h(\mathbf{x})$ is a DECBF.

These results were obtained by replacing the continuous-time CBF condition (3) with a difference equation. Given an arbitrary state constraint set $\mathcal{X} \subseteq \mathbb{R}^n$, however, they provide no guidance on how to construct a DECBF. In practice, the approach is implemented by proposing a candidate DECBF and tuning $\lambda \in (0, 1]$ to help ensure $\mathbf{x}_k \in \mathcal{X}$, $\forall k \geq 0$. Furthermore, DECBFs provide only sufficient conditions for control invariance as opposed to the stronger necessary and sufficient conditions provided by continuous-time CBFs.

In this paper, we provide a definition for discrete-time control barrier functions (DCBFs) that is both necessary and sufficient for the control invariance of a set $\mathcal{C}$. Then, given arbitrary state constraints $\mathcal{X}$, we show how to construct a control invariant subset $\mathcal{C} \subseteq \mathcal{X}$ and find an associated DCBF.

III. Discrete-time Control Barrier Functions

Consider the discrete-time system (4) and a set $\mathcal{C} \subseteq \mathbb{R}^n$. As detailed in [16], $\mathcal{C}$ is control invariant if and only if

$$\forall \mathbf{x} \in \mathcal{C}, \quad \exists \mathbf{u} \in \mathcal{U}: \quad f(\mathbf{x}, \mathbf{u}) \in \mathcal{C}. \tag{6}$$

Notice that the discrete-time control invariance condition (6) is defined on the entirety of the set $\mathcal{C}$, as opposed to only its boundary $\partial\mathcal{C}$ as in the continuous-time case (1). With this in mind, consider the following definition and its consequence.

Definition 3 (Discrete-time Control Barrier Function): A continuous function $h \in C^0 : \mathbb{R}^n \to \mathbb{R}$ is a DCBF for the closed set $\mathcal{C} \subseteq \mathbb{R}^n$ satisfying (2) if

$$\sup_{\mathbf{u} \in \mathcal{U}} \left[h\big(f(\mathbf{x}, \mathbf{u})\big)\right] \geq 0, \quad \forall \mathbf{x} \in \mathcal{C}. \tag{7}$$

Proposition 3: The closed set $\mathcal{C} \subseteq \mathbb{R}^n$ given in (2) is control invariant if and only if $h(\mathbf{x})$ is a DCBF.

Proof: Let $\mathcal{C}$ be control invariant. It follows from (2) and (6) that for all $\mathbf{x} \in \mathcal{C}$, there exists a control input $\mathbf{u} \in \mathcal{U}$ such that $h\big(f(\mathbf{x}, \mathbf{u})\big) \geq 0$. Therefore, h is a DCBF. Conversely, let h be a DCBF. It follows from (2) and (7) that for all $\mathbf{x} \in \mathcal{C}$, there exists a control $\mathbf{u} \in \mathcal{U}$ such that $f(\mathbf{x}, \mathbf{u}) \in \mathcal{C}$. Therefore, $\mathcal{C}$ is a control invariant set. ■

Note that, unlike the continuous-time case, $h(\mathbf{x})$ does not need to be continuously differentiable. As a result, it is simple to obtain $h(\mathbf{x})$ whenever $\mathcal{C}$ is defined by the the intersection of multiple state constraints.

Remark 1: Given the set $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n \mid g(\mathbf{x}) \geq 0\}$, where $g \in C^0 : \mathbb{R}^n \to \mathbb{R}^{n_c}$, condition (2) is satisfied by the continuous function $h(\mathbf{x}) = \min\big(g(\mathbf{x})\big)$.

Remark 2: Given the polyhedral set $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n \mid H\mathbf{x} \leq \mathbf{c}\}$, with $H \in \mathbb{R}^{n_c \times n}$ and $\mathbf{c} \in \mathbb{R}^{n_c}$, condition (2) is satisfied by the continuous function $h(\mathbf{x}) = \min(\mathbf{c} - H\mathbf{x})$.

A. Safe Control Invariant Sets

Given an arbitrary state constraint set $\mathcal{X} \subseteq \mathbb{R}^n$, Proposition 3 has two consequences: (i) If the set $\mathcal{X}$ is control invariant, then any DCBF $h(\mathbf{x})$ is also a DECBF that satisfies (5) with $\lambda = 1$; (ii) if the set $\mathcal{X}$ is not control invariant, then it is impossible to find a DCBF for it. The second consequence can be addressed by finding a control invariant subset $\mathcal{C} \subset \mathcal{X}$ and finding a DCBF for $\mathcal{C}$.

Definition 4 (Safe Control Invariant Set): Given the constraint set $\mathcal{X} \subseteq \mathbb{R}^n$, a control invariant set $\mathcal{C} \subseteq \mathbb{R}^n$ is safe if $\mathcal{C} \subseteq \mathcal{X}$.

Based on Definitions 3 and 4, the design of a suitable DCBF for arbitrary state constraints $\mathcal{X}$ can be achieved by identifying a closed safe control invariant set $\mathcal{C} \subseteq \mathcal{X}$. We give guidance for finding such a set $\mathcal{C}$ in Section IV. The following subsection explains how the resulting DCBF can be used to develop a control law that guarantees constraint satisfaction.

B. Discrete-time Control Barrier Function-based Programs

Similar to CBFs [1], the proposed DCBFs can be used as an add-on unit that bestows safety properties to a nominal controller $\kappa(\mathbf{x})$ by solving the DCBF-based program

$$\begin{aligned} \min_{\mathbf{u} \in \mathcal{U}} \quad & \|\mathbf{u} - \kappa(\mathbf{x})\|^2 \\ \text{s.t.} \quad & h\big(f(\mathbf{x}, \mathbf{u})\big) \geq 0. \end{aligned} \tag{8}$$

The closed-loop system filtered by the DCBF-based program has the following properties.

Theorem 1 (Recursive Feasibility and Safety): Given the state constraint set $\mathcal{X} \subseteq \mathbb{R}^n$, let $\mathcal{C} \subseteq \mathcal{X}$ be a safe control invariant set and let $h \in C^0 : \mathbb{R}^n \to \mathbb{R}$ be an associated DCBF. Moreover, given a nominal controller $\kappa : \mathbb{R}^n \to \mathbb{R}^m$ and the update equations $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)$, let $u(\mathbf{x})$ be the solution to the DCBF-based program (8) at $\mathbf{x}$. Then, given the initial condition $\mathbf{x}_0 \in \mathcal{C}$, the control law $\mathbf{u}_k = u(\mathbf{x}_k)$ is such that:

1) The DCBF-based program (8) is feasible for all $k \geq 0$;
2) The closed-loop response satisfies $\mathbf{x}_k \in \mathcal{X},\ \forall k \geq 0$.

Proof: Given $\mathbf{x}_k \in \mathcal{C}$, it follows by Definition 3 that the DCBF-based program (8) is feasible at $\mathbf{x}_k$. Since the solution exists, the one-step update $\mathbf{x}_{k+1} = f\big(\mathbf{x}_k, u(\mathbf{x}_k)\big)$ is guaranteed to satisfy $h(\mathbf{x}_{k+1}) \geq 0$, and (2) implies $\mathbf{x}_{k+1} \in \mathcal{C}$. Recursive feasibility then follows directly from the requirement $\mathbf{x}_0 \in \mathcal{C}$. As for safety, it is sufficient to note that, due to Definition 4, the set $\mathcal{C}$ is such that $\mathbf{x}_k \in \mathcal{C} \Rightarrow \mathbf{x}_k \in \mathcal{X}$. ■

Although (8) is generally a nonlinear program, given a control-affine system $f(\mathbf{x}, \mathbf{u}) = f_x(\mathbf{x}) + f_u(\mathbf{x})\mathbf{u}$ and polyhedral constraints $\mathcal{U} = \{M\mathbf{u} \leq \mathbf{b}\}$, $\mathcal{C} = \{H\mathbf{x} \leq \mathbf{c}\}$, the DCBF-based program (8) reduces to a quadratic program (QP)

$$\begin{aligned} \min_{\mathbf{u}} \quad & \|\mathbf{u} - \kappa(\mathbf{x})\|^2 \\ \text{s.t.} \quad & H f_u(\mathbf{x})\mathbf{u} \leq \mathbf{c} - H f_x(\mathbf{x}), \\ & M\mathbf{u} \leq \mathbf{b}. \end{aligned} \tag{9}$$

IV. Constructing Safe Invariant Sets

In this section, we leverage the rich literature on maximal output admissible sets (MOASs) to construct a safe control invariant set for arbitrary safety sets under input constraints. We first introduce the approach for general nonlinear systems and then specialize it to linear systems.

A. Maximal Output Admissible Sets

Let the nonlinear system in (4) be stabilizable, and $\bar{x} : \mathbb{R}^l \to \mathbb{R}^n$ and $\bar{u} : \mathbb{R}^l \to \mathbb{R}^m$ be two C^0 functions such that

$$\bar{x}(\mathbf{r}) = f\big(\bar{x}(\mathbf{r}), \bar{u}(\mathbf{r})\big), \qquad \forall \mathbf{r} \in \mathbb{R}^l. \tag{10}$$

By construction, $\mathbf{r} \in \mathbb{R}^l$ is a parametrization of all the possible equilibrium points of the system. It can also be treated as the reference for a prestabilizing control law $\pi : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^m$ such that the sequence of maps

$$\Pi_{k+1}(\mathbf{x}, \mathbf{r}) = f_\pi\big(\Pi_k(\mathbf{x}, \mathbf{r}), \mathbf{r}\big), \tag{11}$$

with $\Pi_0(\mathbf{x}, \mathbf{r}) = \mathbf{x}$, and $f_\pi(\mathbf{x}, \mathbf{r}) \triangleq f\big(\mathbf{x}, \pi(\mathbf{x}, \mathbf{r})\big)$, satisfies

$$\lim_{k \to \infty} \Pi_k(\mathbf{x}, \mathbf{r}) = \bar{x}(\mathbf{r}). \tag{12}$$

Notably, the map $\Pi_k : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^n$ takes any state $\mathbf{x}$ to the k-th step of its closed-loop trajectory under the feedback policy π, subject to a constant reference $\mathbf{r}$. Given a C^0 output function $c : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$, define a set of output constraints $\mathcal{Y} \subseteq \mathbb{R}^p$ that captures all the state and input constraints, i.e.

$$c(\mathbf{x}, \mathbf{u}) \in \mathcal{Y} \iff (\mathbf{x}, \mathbf{u}) \in \mathcal{X} \times \mathcal{U}, \tag{13}$$

let $c_\pi(\mathbf{x}, \mathbf{r}) \triangleq c\big(\mathbf{x}, \pi(\mathbf{x}, \mathbf{r})\big)$, be the output constraint function of the prestabilized system.

Definition 5 (Maximal Output Admissible Set): Given the nonlinear system (4) subject to output constraints $c(\mathbf{x}, \mathbf{u}) \in \mathcal{Y}$, let $\pi(\mathbf{x}, \mathbf{r})$ be a prestabilizing control law such that (11)-(12) hold. Then, the MOAS of the prestabilized system is

$$\mathcal{O}_\infty \triangleq \{(\mathbf{x}, \mathbf{r}) \in \mathbb{R}^n \times \mathbb{R}^l \mid c_\pi\big(\Pi_k(\mathbf{x}, \mathbf{r}), \mathbf{r}\big) \in \mathcal{Y},\ \forall k \in \mathbb{N}\}.$$

Since the size and shape of the MOAS depends on the prestabilizing control law, the choice of $\pi(\mathbf{x}, \mathbf{r})$ influences the degree to which $\mathcal{O}_\infty$ is an inner approximation of $\mathcal{C}$. Note that, if the nominal control law $\kappa(\mathbf{x})$ is stabilizing, it is always possible to set $\pi(\mathbf{x}, \mathbf{r}) = \kappa(\mathbf{x})$.

B. MOAS-based Safe Control Invariant Sets

Let $\text{Proj}_x \mathcal{O}_\infty \subseteq \mathbb{R}^n$ be the projection of $\mathcal{O}_\infty \subseteq \mathbb{R}^n \times \mathbb{R}^l$ onto the state-space $\mathbb{R}^n$. That is,

$$\text{Proj}_x \mathcal{O}_\infty = \{\mathbf{x} \in \mathbb{R}^n \mid \exists \mathbf{r} \in \mathbb{R}^l, (\mathbf{x}, \mathbf{r}) \in \mathcal{O}_\infty\}. \tag{14}$$

Then, we have the following result.

Proposition 4: $\text{Proj}_x \mathcal{O}_\infty$ is a safe control invariant set.

Proof: Let $\mathbf{x} \in \text{Proj}_x \mathcal{O}_\infty$. Then, there exists a $\mathbf{r} \in \mathbb{R}^l$ such that $(\mathbf{x}, \mathbf{r}) \in \mathcal{O}_\infty$. By definition of $\mathcal{O}_\infty$, $c_\pi\big(\Pi_0(\mathbf{x}, \mathbf{r}), \mathbf{r}\big) = c_\pi(\mathbf{x}, \mathbf{r}) = c\big(\mathbf{x}, \pi(\mathbf{x}, \mathbf{r})\big) \in \mathcal{Y}$. By (13), it follows that $\mathbf{x} \in \mathcal{X}$ and, thus, $\text{Proj}_x \mathcal{O}_\infty \subseteq \mathcal{X}$. Let $\mathbf{u} = \pi(\mathbf{x}, \mathbf{r})$ and note that $c(\mathbf{x}, \mathbf{u}) \in \mathcal{Y}$ implies $\mathbf{u} \in \mathcal{U}$. Further, $f(\mathbf{x}, \mathbf{u}) = f_\pi(\mathbf{x}, \mathbf{r}) = f_\pi\big(\Pi_0(\mathbf{x}, \mathbf{r}), \mathbf{r}\big) = \Pi_1(\mathbf{x}, \mathbf{r})$. It follows by definition of $\mathcal{O}_\infty$ that $\big(f(\mathbf{x}, \mathbf{u}), \mathbf{r}\big) \in \mathcal{O}_\infty$. Thus,

$f(\mathbf{x},\mathbf{u}) \in \mathrm{Proj}_x\mathcal{O}_\infty$ and $\mathrm{Proj}_x\mathcal{O}_\infty$ is control invariant by (6). ■

It follows from Proposition 3 that any continuous function $h \in C^0 : \mathbb{R}^n \to \mathbb{R}$ such that $\mathrm{Proj}_x\mathcal{O}_\infty = \{\mathbf{x} \mid h(\mathbf{x}) \geq 0\}$ is a DCBF for $\mathrm{Proj}_x\mathcal{O}_\infty$. This observation serves as a bridge between the rich literature on MOASs and the emerging CBF literature. While computing the MOAS for general nonlinear systems is difficult, methods for estimating it for certain classes of nonlinear systems can be found in [18], [19]. The following section specializes these results to linear systems, for which the MOAS can be computed in closed form.

Remark 3: The idea of constructing a CBF by prestabilizing the system and finding the associated invariant set has been previously explored in the so-called "backup" CBF literature [7], [8]. This approach differs from ours in two significant points: First, backup CBFs have only been proposed in continuous-time, which not only makes it more challenging to compute $h(\mathbf{x})$, since it requires solving a differential equation, but also because it requires the ability to compute its derivative $\dot{h}(\mathbf{x},\mathbf{u})$; second, backup CBFs prestabilize the system around the target reference (e.g., the origin), as opposed to letting $\mathbf{r}$ be a free variable that parametrizes all the possible references of the prestabilizing control law. Therefore, as illustrated in Fig. 1, the control invariant set $\mathrm{Proj}_x\mathcal{O}_\infty$ is larger than the invariant sets featured in [7], [8].

C. Linear Systems

Consider the discrete-time, linear system

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k, \tag{15}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, and the pair (A, B) is stabilizable. Let the state $\mathcal{X} \subseteq \mathbb{R}^n$ and input $\mathcal{U} = \{M\mathbf{u} \leq \mathbf{b}\}$ constraint sets be polyhedral. Then, there exist matrices $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$ such that we can define outputs $\mathbf{y}_k = C\mathbf{x}_k + D\mathbf{u}_k$, and a set of output constraints $\mathcal{Y} = \{\mathbf{y} \in \mathbb{R}^p : L\mathbf{y} \leq \mathbf{a}\}$, with appropriately sized matrix L and $\mathbf{a}$, such that $C\mathbf{x} + D\mathbf{u} \in \mathcal{Y} \iff (\mathbf{x},\mathbf{u}) \in \mathcal{X} \times \mathcal{U}$. Let $G_x \in \mathbb{R}^{n\times m}$ and $G_u \in \mathbb{R}^{m\times m}$ be such that $G = [G_x^\top \; G_u^\top]^\top$ is a basis for $\mathrm{null}(\begin{bmatrix} A - I_n & B \end{bmatrix})$. Consider the prestabilizing policy $\pi(\mathbf{x},\mathbf{r}) = G_u\mathbf{r} - K(\mathbf{x} - G_x\mathbf{r})$, with gain matrix $K \in \mathbb{R}^{m\times n}$. We can define the closed-loop matrices $A_\pi = A - BK$, $B_\pi = B(G_u + KG_x)$, $C_\pi = C - DK$ and $D_\pi = D(G_u + KG_x)$. With this, the map $\Pi_k : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ has a closed-form expression:

$$\Pi_k(\mathbf{x},\mathbf{r}) = A_\pi^k\mathbf{x} + \left(\sum_{i=0}^{k-1} A_\pi^i\right) B_\pi\mathbf{r}, \quad k \in \{0, 1, 2, \ldots\}.$$

Note that $\Pi_0(\mathbf{x},\mathbf{r}) = \mathbf{x}$. As detailed in [10], the MOAS of the prestabilized linear system is

$$\mathcal{O}_\infty = \{(\mathbf{x},\mathbf{r}) : C_\pi\Pi_k(\mathbf{x},\mathbf{r}) + D_\pi\mathbf{r} \in \mathcal{Y},\ \forall k \in \mathbb{N}\}. \tag{16}$$

For the considered polyhedral constraint sets $\mathcal{X}$ and $\mathcal{U}$, $\mathcal{O}_\infty$ is a polyhedron [10]. Furthermore, if A_π is Schur, (A_π, C_π) is observable, and $\mathcal{Y}$ is compact, then $\mathcal{O}_\infty$ is compact and the following inner approximation is finitely determined [20]:

$$\mathcal{O}_\infty^\epsilon \triangleq \mathcal{O}_\infty \cap (\mathbb{R}^n \times \mathcal{R}^\epsilon), \tag{17}$$

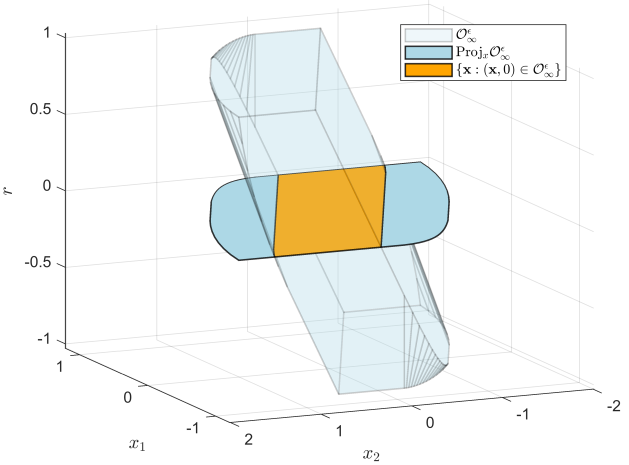

Fig. 1. Example of the strictly output admissible set $\boldsymbol{\mathcal{O}_\infty^\epsilon} \subset \mathbb{R}^{\mathbf{2}} \times \mathbb{R}$ and its projection $\mathrm{Proj}_{\boldsymbol{x}}\boldsymbol{\mathcal{O}_\infty^\epsilon} \subset \mathbb{R}^{\mathbf{2}}$. Also plotted is the control invariant set associated to the prestabilized origin (see Remark 3). This particular example refers to the constrained double integrator system detailed in Example 1.

where $\mathcal{R}^\epsilon \triangleq \{\mathbf{r} \in \mathbb{R}^m : L(CG_x + DG_u)\mathbf{r} \leq (1-\epsilon)\mathbf{a}\}$ is the strictly steady-state admissible reference set for some small $\epsilon \in (0, 1)$. The set $\mathcal{O}_\infty^\epsilon$ is called the strictly output admissible set and can be computed using [10, Algorithm 3.2]. As before, we consider the projection onto the state-space

$$\mathrm{Proj}_x\mathcal{O}_\infty^\epsilon = \{\mathbf{x} \in \mathbb{R}^n : \exists \mathbf{r} \in \mathcal{R}^\epsilon, (\mathbf{x},\mathbf{r}) \in \mathcal{O}_\infty^\epsilon\}. \tag{18}$$

Following the same process as in Proposition 4, it can be shown that $\mathrm{Proj}_x\mathcal{O}_\infty^\epsilon$ is also a safe control invariant set. In addition, when $\epsilon \to 0$, we recover $\mathrm{Proj}_x\mathcal{O}_\infty$. See Fig. 1 for an example of $\mathcal{O}_\infty^\epsilon$ and its projection.

Under the previously stated assumptions, $\mathcal{O}_\infty^\epsilon$ is a finitely determined polyhedron. Thus, we can find a matrix $H \in \mathbb{R}^{n_c\times n}$ and vector $\mathbf{c} \in \mathbb{R}^{n_c}$ for some finite integer $n_c > 0$ such that $\mathrm{Proj}_x\mathcal{O}_\infty^\epsilon = \{\mathbf{x} \in \mathbb{R}^n : H\mathbf{x} \leq \mathbf{c}\}$. Because this set is control invariant, it follows from Proposition 3 that $h(\mathbf{x}) = \min(\mathbf{c} - H\mathbf{x})$ is a DCBF. Let $\kappa : \mathbb{R}^n \to \mathbb{R}^m$ be a nominal controller with desired performance properties. Then, we can formulate the following DCBF-based QP

$$\begin{aligned} \min_{\mathbf{u}} \quad & \|\mathbf{u} - \kappa(\mathbf{x})\|^2 && \text{(19a)} \\ \text{s.t.} \quad & HB\mathbf{u} \leq \mathbf{c} - HA\mathbf{x}, && \text{(19b)} \\ & M\mathbf{u} \leq \mathbf{b}, && \text{(19c)} \end{aligned}$$

where (19b) is equivalent to $h(A\mathbf{x} + B\mathbf{u}) \geq 0$ and (19c) ensures $\mathbf{u} \in \mathcal{U}$.

V. Examples

We present two examples to demonstrate the usefulness of DCBFs synthesized from the MOAS. The first example is a double integrator system and the second is a pitch pointing control problem for a fixed-wing F-16 aircraft. For each example, we provide a comparison with MPC [14], CG [15], and DECBF-based control [11]. In all examples, we let $\epsilon = 0.05$ and solve the optimization problems in MATLAB using YALMIP [21] with MOSEK [22]. Projections are computed with MPT3 [23]. All computations are performed in a laptop

PC running Windows 10 with an Intel i5 @ 1.60 GHz CPU and 16 GB RAM.

TABLE I
OPTIMIZATION PROBLEM SOLVE TIMES

	Double integrator		F-16 pitch pointing	
	Avg [ms]	Max [ms]	Avg [ms]	Max [ms]
CG	0.98	1.24	1.41	1.68
DECBF	0.93	1.08	1.25	1.77
MPC	2.71	6.33	3.22	4.45
DCBF	1.41	2.15	1.49	2.01

Example 1: Consider a double integrator system $\mathbf{x} = [x\ \dot{x}]^\top$, $u = \ddot{x}$, with sampling time 0.1 seconds and system matrices

$$A = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}.$$

The state constraint set is $\mathcal{X} = \{\mathbf{x} : |x| \leq 1\}$ and the input constraint set is $\mathcal{U} = \{u : |u| \leq 1.5\}$. The initial condition is $\mathbf{x}_0 = [0\ 0]^\top$ and the nominal controller is $\kappa(\mathbf{x}) = -26.8(x - 1.1) - 12.6\dot{x}$, which stabilizes the system to the unsafe point $[1.1\ 0]^\top$. We choose prestabilizing controller $\pi(\mathbf{x}, r) = -2(x - r) - 2.2\dot{x}$. The candidate DECBFs are $b_1(\mathbf{x}) = 0.25 - 0.25x - 0.1\dot{x}$ and $b_2(\mathbf{x}) = 0.25 + 0.25x + 0.1\dot{x}$. We tuned the parameters λ_i in the DECBF-condition (5) to be close to 1 (high performance) while retaining feasibility of the DECBF-based program, this led to $\lambda_1 = \lambda_2 = 0.42$.

Fig. 2 compares the closed-loop behavior of the different control strategies. As expected, CG exhibits the slowest response, whereas MPC achieves the fastest response. Our approach outperforms DECBFs and achieves comparable results with MPC while being less computationally intensive (see Table I). It is also worth noting that, given a different initial condition $\mathbf{x}_0$, the DECBF-based program may become infeasible, whereas the DCBF-based program is guaranteed to remain recursively feasible whenever $\mathbf{x}_0 \in \text{Proj}_x \mathcal{O}_\infty^\epsilon \subset \mathcal{X}$.

Example 2: Consider the pitch dynamics model of an F-16 aircraft given in [24] and let us discretize it with a sampling time of 0.1 seconds. The state vector is $\mathbf{x} = [\theta\ q\ \alpha\ \delta_e\ \delta_f]^\top$ which collects the pitch, pitch rate, angle of attack, elevator deflection and flaperon deflection, respectively. The input vector is $\mathbf{u} = [\delta_{ec}\ \delta_{fc}]^\top$ which collects the elevator and flaperon deflection commands, respectively. The state and input matrices are

$$A = \begin{bmatrix} 1 & 0.1 & 0.2 & -0.1 & 0 \\ 0 & 1.1 & 4.2 & -0.8 & -0.1 \\ 0 & 0.1 & 1.1 & -0.1 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ -1 & -0.1 \\ -0.1 & 0 \\ 0.9 & 0 \\ 0 & 0.9 \end{bmatrix}.$$

Pitch pointing control requires mild angle of attack variations and the control surface deflections are usually limited [24]. The set of safe states is $\mathcal{X} = \{\mathbf{x} : |\alpha| \leq 4\pi/180,\ |\delta_e| \leq 25\pi/180,\ |\delta_f| \leq 20\pi/180\}$ and there are no input constraints as they are virtual commands (i.e., $\mathcal{U} = \mathbb{R}^2$). In this example, we consider the same nominal and prestabilizing controllers (i.e., $\pi = \kappa$) and design them via LQR with state and input weight matrices $Q = \text{diag}([10\ 0\ 10\ 0\ 0])$ and $R = 10^{-3} I_2$, respectively. Given the candidate DECBFs $b_1(\mathbf{x}) = \pi/45 - \alpha$, $b_2(\mathbf{x}) = 5\pi/36 - \delta_e$, $b_3(\mathbf{x}) = \pi/9 - \delta_f$, $b_4(\mathbf{x}) = \pi/45 + \alpha$, $b_5(\mathbf{x}) = 5\pi/36 + \delta_e$ and $b_6(\mathbf{x}) = \pi/9 + \delta_f$, we were unable to identify suitable parameters $\lambda_i \in (0, 1]$, $i = 1, \ldots, 6$, that retain feasibility of the DECBF-based program when using the input weight matrix $R = 10^{-3} I_2$. Thus, we designed a milder nominal controller for the DECBF approach by increasing the input weight matrix to $R_{\text{DECBF}} = 0.09 I_2$, which then enabled us to select $\lambda_i = 0.9$, $i = 1, \ldots, 6$. Although this ad-hoc solution works for the given initial conditions, there is no guarantee that the DECBF will be recursively feasible for other $\mathbf{x}_0$. In contrast, the DCBF proposed in this paper requires no such tuning and is guaranteed to work for any $\mathbf{x}_0 \in \text{Proj}_x \mathcal{O}_\infty^\epsilon$.

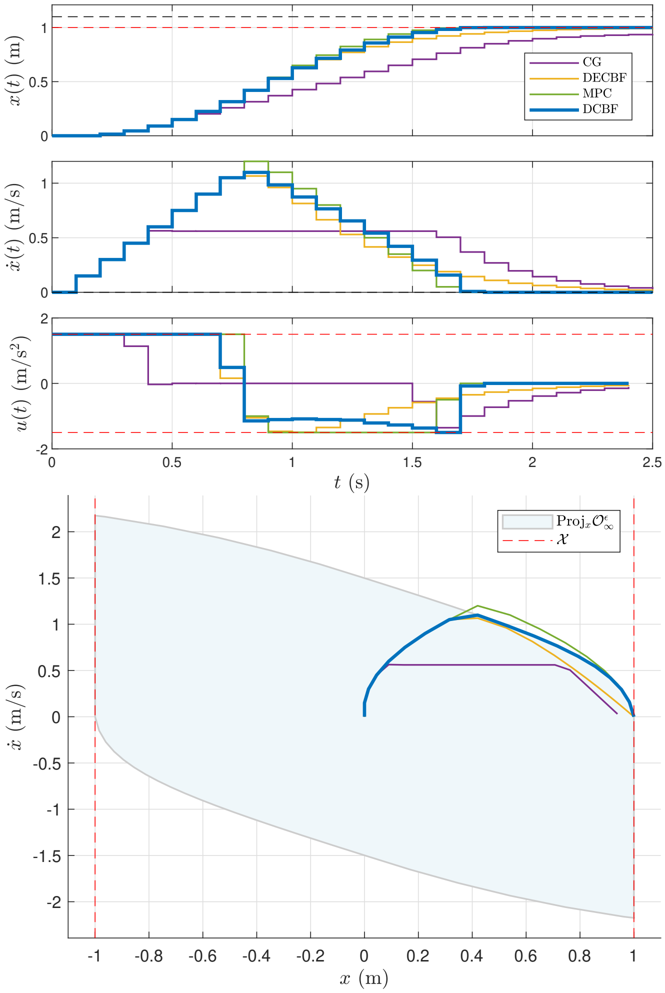

Fig. 2. Simulation of the double integrator model. It can be seen that all approaches successfully enforce the safety constraints on the position $\boldsymbol{x}$ and input $\boldsymbol{u}$ and the system converges to a safe point close to the desired reference. The bottom plot shows the state-space trajectory each approach follows and it can be seen that the system under the DCBF travels along the boundary of Proj$_{\boldsymbol{x}} \boldsymbol{\mathcal{O}}_{\boldsymbol{\infty}}^{\boldsymbol{\epsilon}}$.

Fig. 3 compares the closed-loop response of each method when tasked with reaching a reference pitch $\theta_r = \pi/20$ and flight path angle $\gamma_r = 13\pi/360$, with $\gamma = \theta - \alpha$. Similar to the previous example, our approach achieves comparable results with MPC with faster solve times (Table I). In this case, the CG outperforms the DECBF since we had to detune the control law to retain feasibility of the DECBF-based program. It is also worth noting that, since we took $\pi = \kappa$, the MOAS used for the CG and the DCBF is the same, which explains their similar solvetimes (see Table I).

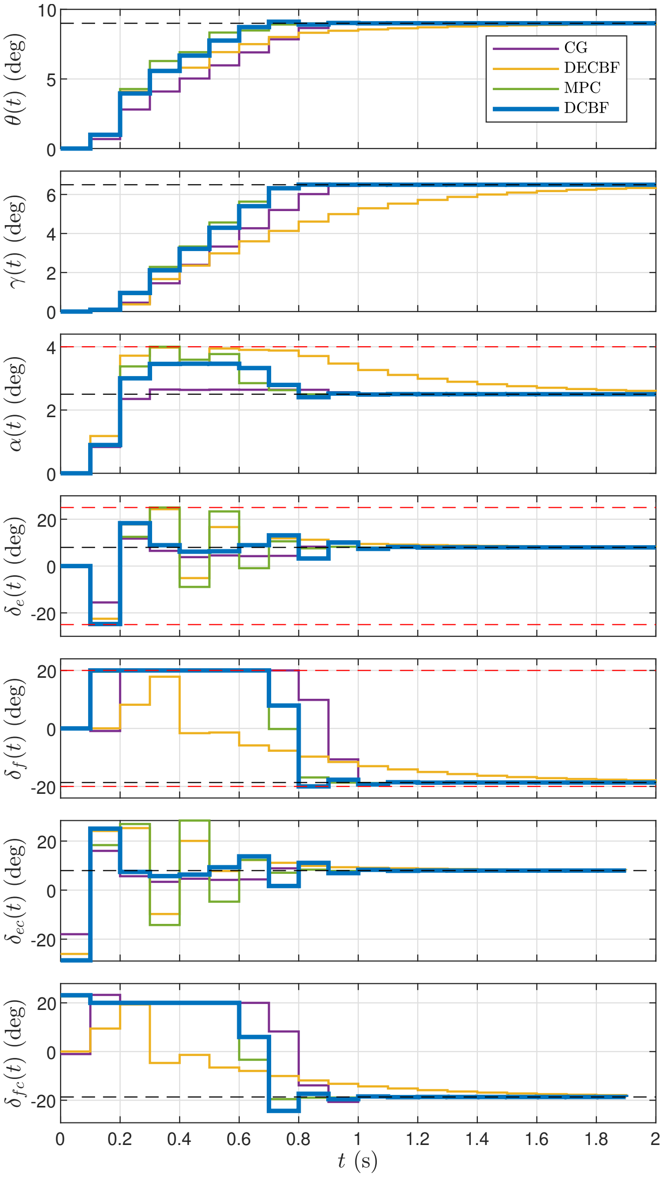

Fig. 3. Simulation of the F-16 aircraft model. It can be seen that all approaches successfully enforce the safety constraints on the angle of attack $\boldsymbol{\alpha}$ and control surfaces deflection $\boldsymbol{\delta_e}$, $\boldsymbol{\delta_f}$. Furthermore, the pitch $\boldsymbol{\theta}$ and flight path angle γ converge to the desired references.

VI. Conclusion

In this paper, we introduced a new definition of discrete-time control barrier functions and provided necessary and sufficient conditions for control invariance. We then showed that the DCBF can be obtained for arbitrary state and input constraints by finding a prestabilizing controller and projecting the maximal output admissible set onto the state space. Numerical simulations showed that, in addition to being safe and recursively feasible, the proposed DCBF-based controller can achieve performances comparable to MPC while incurring lower computational costs. In future work, we will explore the relationship between the prestabilizing control policy and the associated MOAS.

References

[1] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[2] S.-C. Hsu, X. Xu, and A. D. Ames, “Control barrier function based quadratic programs with application to bipedal robotic walking,” *Proc. IEEE Amer. Control Conf. (ACC)*, pp. 4542–4548, 2015.

[3] W. S. Cortez, D. Oetomo, C. Manzie, and P. Choong, “Control barrier functions for mechanical systems: Theory and application to robotic grasping,” *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 2, pp. 530–545, 2019.

[4] B. Capelli, C. Secchi, and L. Sabattini, “Passivity and control barrier functions: optimizing the use of energy,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1356–1363, 2022.

[5] V. Freire and X. Xu, “Flatness-based quadcopter trajectory planning and tracking with continuous-time safety guarantees,” *IEEE Trans. Control Syst. Technol.*, pp. 1–16, early access. 2023.

[6] X. Xu, “Constrained control of input–output linearizable systems using control sharing barrier functions,” *Automatica*, vol. 87, pp. 195–201, 2018.

[7] T. Gurriet, M. Mote, A. Singletary, P. Nilsson, E. Feron, and A. D. Ames, “A scalable safety critical control framework for nonlinear systems,” *IEEE Access*, vol. 8, pp. 187 249–187 275, 2020.

[8] Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames, “Backup control barrier functions: Formulation and comparative study,” *Proc. IEEE Conf. Decis. Control (CDC)*, pp. 6835–6841, 2021.

[9] L. Liu, T. Gao, Y.-J. Liu, S. Tong, C. P. Chen, and L. Ma, “Time-varying IBLFs-based adaptive control of uncertain nonlinear systems with full state constraints,” *Automatica*, vol. 129, p. 109595, 2021.

[10] E. G. Gilbert and K. T. Tan, “Linear systems with state and control constraints: The theory and application of maximal output admissible sets,” *IEEE Trans. Autom. Control*, vol. 36, no. 9, pp. 1008–1020, 1991.

[11] A. Agrawal and K. Sreenath, “Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation,” *Proc. Robot.: Sci. Syst. (RSS)*, vol. 13, 2017.

[12] 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,” *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 01, pp. 3387–3395, 2019.

[13] J. Zeng, B. Zhang, and K. Sreenath, “Safety-critical model predictive control with discrete-time control barrier function,” *Proc. IEEE Amer. Control Conf. (ACC)*, pp. 3882–3889, 2021.

[14] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing, 2017.

[15] A. Bemporad, A. Casavola, and E. Mosca, “Nonlinear control of constrained linear systems via predictive reference management,” *IEEE Trans. Autom. Control*, vol. 42, no. 3, pp. 340–349, 1997.

[16] F. Blanchini, “Set invariance in control,” *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[17] S. Kolathaya and A. D. Ames, “Input-to-state safety with control barrier functions,” *IEEE Control Syst. Lett.*, vol. 3, no. 1, pp. 108–113, 2018.

[18] K. Hirata and Y. Ohta, “Exact determinations of the maximal output admissible set for a class of nonlinear systems,” *Automatica*, vol. 44, no. 2, pp. 526–533, 2008.

[19] A. Cotorruelo, M. Hosseinzadeh, D. R. Ramirez, D. Limon, and E. Garone, “Reference dependent invariant sets: Sum of squares based computation and applications in constrained control,” *Automatica*, vol. 129, p. 109614, 2021.

[20] E. Garone, S. Di Cairano, and I. Kolmanovsky, “Reference and command governors for systems with constraints: A survey on theory and applications,” *Automatica*, vol. 75, pp. 306–328, 2017.

[21] J. Löfberg, “YALMIP: A toolbox for modeling and optimization in MATLAB,” *Proc. Comput. Aided Control Syst. Des. (CACSD)*, pp. 284–289, 2004.

[22] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 10.0.*, 2023.

[23] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, “Multi-parametric toolbox 3.0,” *Proc. Eur. Control Conf. (ECC)*, pp. 502–510, 2013.

[24] K. M. Sobel and E. Y. Shapiro, “A design methodology for pitch pointing flight control systems,” *AIAA J. Guid. Control, Dyn.*, vol. 8, no. 2, pp. 181–187, 1985.