Iterative Convex Optimization for Model Predictive Control with Discrete-Time High-Order Control Barrier Functions

Shuo Liu*[1], Jun Zeng*[2], Koushil Sreenath[2] and Calin A. Belta[1]

***Abstract*— Safety is one of the fundamental challenges in control theory. Recently, multi-step optimal control problems for discrete-time dynamical systems were formulated to enforce stability, while subject to input constraints as well as safety-critical requirements using discrete-time control barrier functions within a model predictive control (MPC) framework. Existing work usually focus on the feasibility or the safety for the optimization problem, and the majority of the existing work restrict the discussions to relative-degree one control barrier functions. Additionally, the real-time computation is challenging when a large horizon is considered in the MPC problem for relative-degree one or high-order control barrier functions. In this paper, we propose a framework that solves the safety-critical MPC problem in an iterative optimization, which is applicable for any relative-degree control barrier functions. In the proposed formulation, the nonlinear system dynamics as well as the safety constraints modeled as discrete-time high-order control barrier functions (DHOCBF) are linearized at each time step. Our formulation is generally valid for any control barrier function with an arbitrary relative-degree. The advantages of fast computational performance with safety guarantee are analyzed and validated with numerical results.**

I. Introduction

A. Motivation

Safety-critical optimal control is a central problem in robotics. For example, reaching a goal while avoiding obstacles and minimizing energy can be formulated as a constrained optimal control problem by using continuous-time control barrier functions (CBFs) [1], [2]. By dividing the timeline into small intervals, the problem is reduced to a (possibly large) number of quadratic programs, which can be solved at real-time speeds. However, this approach can be too aggressive due to the lack of predicting ahead.

Model predictive control (MPC) with CBFs [3] considers the safety problem in the discrete-time domain, and provides a smooth control policy as it involves future state information along a receding horizon. However, the computational time is relatively large and increases dramatically with a larger horizon, since the optimization itself is usually nonlinear and non-convex. An additional issue of this nonlinear model predictive formulation is the feasibility of the optimization. For CBFs with relative-degree one, relaxation techniques have been introduced in [4]. In this paper, we address the above challenges with a proposed convex MPC with linearized, discrete-time CBFs, under an iterative approach. In contrast with the real-time iteration (RTI) approach introduced in [5], which solves the problem through iterative Newton steps, our approach solves the optimization problem formulated by a convex MPC iteratively for each time step. We show that the proposed approach can significantly reduce the computational time, compared to the state of the art introduced in [4], even for CBFs with high relative-degree, without sacrificing the controller performance. The feasibllity rate of our proposed method also outperforms that of the baseline method in [4] for large horizon lengths.

B. Related work

1) Model Predictive Control (MPC): MPC is widely used in modern control systems, such as controller design in robotic manipulation and locomotion [6], [7] to obtain a control strategy as a solution to an optimization problem. Stability was achieved in [8] by incorporating discrete-time control Lyapunov functions (DCLFs) into a general MPC-based optimization problem to realize real-time control on a robotic system with limited computational resources. More and more recent work like [9] emphasizes safety in robot design and deployment since it is an important criterion for real-world tasks. Some works consider safety criteria through the introduction of additional repelling functions [1], [10] while some works regard obstacle avoidance as one concrete scenario in terms of safety criteria for robots [11]–[13]. Those safety criteria are usually formulated as constraints in optimization problems. This paper can be seen in the context of MPC with safety constraints.

2) Continuous-Time CBFs: It has recently been shown that to stabilize an affine control system while also satisfying safety constraints and control limitations, CBFs can be unified with control Lyapunov functions (CLFs) to form a sequence of single-step optimization programs [1], [2], [14], [15]. If the cost is quadratic, the optimizations are quadratic programs (QP), and the solutions can be deployed in real time [1], [16]. Adaptive, robust and stochastic versions of safety-critical control with CBFs were introduced in [17]–[21]. For safety constraints expressed using functions with high relative degree with respect to the dynamics of the system, exponential CBFs [22] and high-order CBFs (HOCBFs) [23]–[25] were proposed.

3) Discrete-Time CBFs: Discrete-time CBFs (DCBFs) were introduced in [26] as a means to enable safety-critical control for discrete-time systems. They were used in a

* Authors contributed equally.

This work was supported in part by the NSF under grants IIS-2024606 and CMMI-1931853.

[1]S. Liu and C. Belta are with the department of Mechanical Engineering, Boston University, Brookline, MA, 02215, USA {liushuo, cbelta}@bu.edu. [2]J. Zeng and K. Sreenath are with the University of California, Berkeley, CA, 94720, USA {zengjunsjtu, koushils}@berkeley.edu

Implementation code is released on https://github.com/ShockLeo/Iterative-MPC-DHOCBF.

nonlinear MPC (NMPC) framework to create NMPC-DCBF [3], wherein the DCBF constraint was enforced through a predictive horizon. This method was also utilised in a multi-layer control framework in [27], where DCBFs with longer horizons were considered in the MPC problem serving as a mid-level controller to guarantee safety. Generalized discrete-time CBFs (GCBFs) and discrete-time high-order CBFs (DHOCBFs) were proposed in [28] and [29] respectively, where the DCBF constraint only acted on the first time-step, i.e., a single-step constraint. MPC with DCBF has been used in various fields, such as autonomous driving [30] and legged robotics [31]. For the work above, the CBF constraints are either limited to be activated at the first time-step [26], [28], [29] to improve the optimization feasibility at the cost of sacrificing the safety performance, or for multiple or all steps [27], [30], [31] with additional performance optimization from other modules, such as multi-layer control [27], [30] or planning [31], which needs to specified for different platforms. A decay-rate relaxing technique [32] was introduced for NMPC with DCBF [4] for all time-steps to enhance the safety and feasibility at the same time, but the computation itself is overall still nonlinear and non-convex which could be computationally slow for large horizons and nonlinear dynamical systems, and the discussion in [4] is limited to relative-degree one. In this paper, we generalize relaxing technique for DHOCBF and largely optimize the computational time compared to all existing work.

C. Contributions

We propose a novel approach to the NMPC with discrete-time CBFs that is significantly faster than existing approaches. In particular, the contributions are as follows:

- We present a model predictive control strategy for safety-critical tasks, where the safety-critical constraints can be enforced by DHOCBFs. The decay rate in each constraint can be relaxed to enhance the feasibility in optimization and to ensure forward invariance of the intersection of a series of safety sets.
- We propose an optimal control framework for guaranteeing safety, where the DHOCBF constraints as well as the system dynamics are linearized at each iteration, and considered as constraints in a convex optimization solved iteratively.
- We show through numerical examples that the proposed framework is significantly faster than existing methods, without sacrificing safety and feasibility.

II. Preliminaries

In this section, we introduce some definitions and results on CBF and MPC.

A. Discrete-Time High-Order Control Barrier Function (DHOCBF)

In this work, safety is defined as forward invariance of a set $\mathcal{C}$, i.e., a system is said to be *safe* if it stays in $\mathcal{C}$ for all time, given that it is initialized in $\mathcal{C}$. We consider the set $\mathcal{C}$ as the superlevel set of a discrete-time function $h : \mathbb{R}^n \to \mathbb{R}$:

$$\mathcal{C} \coloneqq \{\mathbf{x}_t \in \mathbb{R}^n : h(\mathbf{x}_t) \geq 0\}. \tag{1}$$

We consider a discrete-time control system in the form

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t), \tag{2}$$

where $\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^n$ represents the state of system (2) at time step $t \in \mathbb{N}, \mathbf{u}_t \in \mathcal{U} \subset \mathbb{R}^q$ is the control input, and function f is locally Lipschitz.

Definition 1 (Relative degree [33])**.** The output $\mathbf{y}_t = h(\mathbf{x}_t)$ of system (2) is said to have relative degree m if

$$\begin{aligned} &\mathbf{y}_{t+i} = h(\bar{f}_{i-1}(f(\mathbf{x}_t, \mathbf{u}_t))),\ i \in \{1, 2, \dots, m\}, \\ \text{s.t. } &\frac{\partial \mathbf{y}_{t+m}}{\partial \mathbf{u}_t} \neq 0_q, \frac{\partial \mathbf{y}_{t+i}}{\partial \mathbf{u}_t} = 0_q,\ i \in \{1, 2, \dots, m-1\}, \end{aligned} \tag{3}$$

i.e., m is the number of steps (delay) in the output $\mathbf{y}_t$ in order for the control input $\mathbf{u}_t$ to appear.

In the above definition, we use $\bar{f}(\mathbf{x}_t)$ to denote the uncontrolled state dynamics $f(\mathbf{x}_t, 0)$. The subscript i of function $\bar{f}(\cdot)$ denotes the i-times recursive compositions of $\bar{f}(\cdot)$, i.e., $\bar{f}_i(\mathbf{x}_t) = \underbrace{\bar{f}(\bar{f}(\dots, \bar{f}}_{i\text{-times}}(\bar{f}_0(\mathbf{x}_t))))$ with $\bar{f}_0(\mathbf{x}_t) = \mathbf{x}_t$.

We assume that $h(\mathbf{x}_t)$ has relative degree m with respect to system (2) based on Def. 1. Starting with $\psi_0(\mathbf{x}_t) \coloneqq h(\mathbf{x}_t)$, we define a sequence of discrete-time functions $\psi_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \dots, m$ as:

$$\psi_i(\mathbf{x}_t) \coloneqq \triangle\psi_{i-1}(\mathbf{x}_t, \mathbf{u}_t) + \alpha_i(\psi_{i-1}(\mathbf{x}_t)), \tag{4}$$

where $\triangle\psi_{i-1}(\mathbf{x}_t, \mathbf{u}_t) \coloneqq \psi_{i-1}(\mathbf{x}_{t+1}) - \psi_{i-1}(\mathbf{x}_t)$, and $\alpha_i(\cdot)$ denotes the i^{th} class κ function which satisfies $\alpha_i(\psi_{i-1}(\mathbf{x}_t)) \leq \psi_{i-1}(\mathbf{x}_t)$ for $i = 1, \dots, m$. A sequence of sets $\mathcal{C}_i$ is defined based on (4) as

$$\mathcal{C}_i \coloneqq \{\mathbf{x}_t \in \mathbb{R}^n : \psi_i(\mathbf{x}_t) \geq 0\},\ i = \{0, \dots, m-1\}. \tag{5}$$

Definition 2 (DHOCBF [29])**.** Let $\psi_i(\mathbf{x}_t),\ i \in \{1, \dots, m\}$ be defined by (4) and $\mathcal{C}_i,\ i \in \{0, \dots, m-1\}$ be defined by (5). A function $h : \mathbb{R}^n \to \mathbb{R}$ is a Discrete-Time High-Order Control Barrier Function (DHOCBF) with relative degree m for system (2) if there exist $\psi_m(\mathbf{x}_t)$ and $\mathcal{C}_i$ such that

$$\psi_m(\mathbf{x}_t) \geq 0,\ \forall x_t \in \mathcal{C}_0 \cap \dots \cap \mathcal{C}_{m-1}. \tag{6}$$

Theorem 1 (Safety Guarantee [29])**.** *Given a DHOCBF $h(\mathbf{x}_t)$ from Def. 2 with corresponding sets $\mathcal{C}_0, \dots, \mathcal{C}_{m-1}$ defined by* (5)*, if $\mathbf{x}_0 \in \mathcal{C}_0 \cap \dots \cap \mathcal{C}_{m-1}$, then any Lipschitz controller $\mathbf{u}_t$ that satisfies the constraint in* (6)*, $\forall t \geq 0$ renders $\mathcal{C}_0 \cap \dots \cap \mathcal{C}_{m-1}$ forward invariant for system* (2)*, i.e., $\mathbf{x}_t \in \mathcal{C}_0 \cap \dots \cap \mathcal{C}_{m-1}, \forall t \geq 0$.*

Remark 1. *The function $\psi_i(\mathbf{x}_t)$ in* (4) *is called a i^{th} order discrete-time control barrier function (DCBF) in this paper. Since satisfying the i^{th} order DCBF constraint ($\psi_i(\mathbf{x}_t) \geq 0$) is a sufficient condition for rendering $\mathcal{C}_0 \cap \dots \cap \mathcal{C}_{i-1}$ forward invariant for system* (2) *as shown above, it is not necessary to formulate DCBF constraints up to m^{th} order as* (6) *if the control input $\mathbf{u}_t$ could be involved in some optimal*

control problem, which allows us to choose an appropriate order for the DCBF constraint to reduce the computation. In other words, the highest order for DCBF could be m_{cbf} with $m_{cbf} \leq m$. We can simply define a i^{th} order DCBF $\psi_i(\mathbf{x}_t)$ in (4) *as*

$$\psi_i(\mathbf{x}_t) \coloneqq \triangle\psi_{i-1}(\mathbf{x}_t, \mathbf{u}_t) + \gamma_i\psi_{i-1}(\mathbf{x}_t), \tag{7}$$

where $0 < \gamma_i \leq 1, i \in \{1, \ldots, m_{cbf}\}$.

The expression in (7) follows the format of the first order DCBF proposed in [26] and could be used to define a DHOCBF with arbitrary relative degree.

B. Model Predictive Control

Consider the problem of regulating to a target state $\mathbf{x}_r$ for the discrete-time system (2) while making sure that safety is guaranteed by ensuring $\psi_0(\mathbf{x}_t) = h(\mathbf{x}_t) \geq 0$. The following optimal control problem takes future N states into account as prediction at each time step t:

NMPC-DCBF:

$$\min_{\mathbf{U}_t, \Omega_t} \; p(\mathbf{x}_{t,N}) + \sum_{k=0}^{N-1} q(\mathbf{x}_{t,k}, \mathbf{u}_{t,k}, \omega_{t,k}) \tag{8a}$$

$$\text{s.t. } \mathbf{x}_{t,k+1} = f(\mathbf{x}_{t,k}, \mathbf{u}_{t,k}), \; k=\{0, \ldots, N-1\} \tag{8b}$$

$$\mathbf{u}_{t,k} \in \mathcal{U}, \mathbf{x}_{t,k} \in \mathcal{X}, \omega_{t,k} \in \mathbb{R}, \; k=\{0, \ldots, N-1\} \tag{8c}$$

$$h(\mathbf{x}_{t,k+1}) \geq \omega_{t,k}(1-\gamma)h(\mathbf{x}_{t,k}), 0 < \gamma \leq 1, \tag{8d}$$

$$k = \{0, \ldots, N-1\},$$

where $\mathbf{x}_{t,k+1}$ denotes the state at time step $k+1$ predicted at time step t obtained by applying the input vector $\mathbf{u}_{t,k}$ to the state $\mathbf{x}_{t,k}$. In (8a), $q(\cdot)$ and $p(\cdot)$ denote stage and terminal costs, respectively, and $\omega_{t,k}$ is a slack variable. The discrete-time dynamics is represented by (8b) and the constraints of state and control input along the horizon are captured by (8c). The DCBF constraint in (8d) is proposed in [26] and is designed to ensure the forward invariance of the set $\mathcal{C}$ based on (1). The above formulation was first proposed in [3], and then later generalized in [4], where the decay rate $(1-\gamma)$ of the CBF was relaxed by slack variable $\omega_{t,k}$ to enhance safety and feasibility.

The optimal solution to (8) at time t is $\mathbf{U}_t^* = [\mathbf{u}_{t,0}^*, \ldots, \mathbf{u}_{t,N-1}^*]$ and $\Omega_t^* = [\omega_{t,0}^*, \ldots, \omega_{t,N-1}^*]$. The first element of $\mathbf{U}_t^*$ is applied to (2) as

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_{t,0}^*) \tag{9}$$

to get the new state $\mathbf{x}_{t+1}$. The constrained finite-time optimal control problem (8) is solved at time step $t+1$, and all future time steps based on the new state $\mathbf{x}_{t+1}$, yielding a safety-critical receding horizon control strategy.

III. Iterative Convex MPC with DHOCBF

In this section, we present an iterative convex MPC for DCBF, which works for general DHOCBFs defined in Sec. II-A.

Algorithm 1 iMPC-DHOCBF

Input: System dynamics (2), candidate CBF constraint, obstacle configurations, initial state $\mathbf{x}(0)$.
Output: Safety-critical optimal control for obstacle avoidance.

```
Set initial guess Ū_0^0 = 0 at t = 0.
Propagate with system dynamics to get initial guess of states X̄_0^0 from initial state x(0).
for t ≤ t_sim − 1 do
    Initialize j = 0.
    while Iteration j (not converged OR j < j_max) do
        Linearize system dynamics / constraints with X̄_t^j, Ū_t^j.
        Solve a convex finite-time constrained optimal control problem (CFTOC) with linearized dynamics / constraints and get optimal values of states and inputs X_t^{*,j}, U_t^{*,j}.
        X̄_t^{j+1} = X_t^{*,j}, Ū_t^{j+1} = U_t^{*,j}, j = j + 1
    end while
    Extract optimized states and inputs X_t^* = X_t^{*,j}, U_t^* = U_t^{*,j} from last iteration and extract u_{t,0}^* from U_t^*.
    Apply u_{t,0}^* with respect to system dynamics (2) to get x_{t+1} = f(x_t, u_{t,0}^*), and record x(t+1) = x_{t+1}.
    Update Ū_{t+1}^0 with U_t^* and propagate to calculate X̄_{t+1}^0.
    t = t + 1.
end for
return closed-loop trajectory [x(0), ..., x(t_sim)]
```

A. Iterative Convex Optimization

The algorithm described in Alg. 1 contains an iterative optimization at each time step t, which is denoted as *iterative* MPC-DHOCBF (iMPC-DHOCBF). Our iterative optimization problem contains three parts for each iteration j: (1) solve a convex finite-time optimal control (CFTOC) problem with linearized dynamics and DHOCBF, (2) check convergence criteria, (3) update state and input vectors for next iteration. Notice that the open-loop trajectory with updated states $\bar{\mathbf{X}}_t^j = [\bar{\mathbf{x}}_{t,0}^j, \ldots, \bar{\mathbf{x}}_{t,N-1}^j]$ and inputs $\bar{\mathbf{U}}_t^j = [\bar{\mathbf{u}}_{t,0}^j, \ldots, \bar{\mathbf{u}}_{t,N-1}^j]$ is passed between iterations, which allows iterative linearization for both system dynamics and DHOCBF locally. As discussed before, "high-order" implies that the relative degree should be larger or equal to one.

The iteration is finished when the convergence error function $e(\mathbf{X}_t^{*,j}, \mathbf{U}_t^{*,j}, \bar{\mathbf{X}}_t^j, \bar{\mathbf{U}}_t^j)$ is within a user-defined normalized convergence criteria, where $\mathbf{X}_t^{*,j} = [\mathbf{x}_{t,0}^{*,j}, \ldots, \mathbf{x}_{t,N}^{*,j}]$, $\mathbf{U}_t^{*,j} = [\mathbf{u}_{t,0}^{*,j}, \ldots, \mathbf{u}_{t,N-1}^{*,j}]$ represent optimized states and inputs at iteration j. To restrict the number of iterations, we limit $j < j_{\max}$, where $j_{\max}$ denotes the maximum numbers of iterations. Therefore, the iterative optimization stops when the cost function reaches a local optimal minimum, whose iteration number is denoted as $j_{t,\text{conv}}$. The optimized states $\mathbf{X}_t^*$ and inputs $\mathbf{U}_t^*$ are passed to the iMPC-DHOCBF formulation for the next time instant. At each time, we record the updated state propagated by the system dynamics with a given discretization time, which allows to extract the output closed-loop trajectory with our proposed iMPC-DHOCBF.

B. Linearization of Dynamics

At iteration j, an improved vector $\mathbf{u}_{t,k}^j$ is considered by linearizing the system around $\bar{\mathbf{x}}_{t,k}^j, \bar{\mathbf{u}}_{t,k}^j$:

$$\mathbf{x}_{t,k+1}^j - \bar{\mathbf{x}}_{t,k+1}^j = A^j(\mathbf{x}_{t,k}^j - \bar{\mathbf{x}}_{t,k}^j) + B^j(\mathbf{u}_{t,k}^j - \bar{\mathbf{u}}_{t,k}^j), \tag{10}$$

where $0 \leq j < j_{\max}$; k and j represent open-loop time step and iteration indices, respectively. We also have

$$A^j = D_{\mathbf{x}} f(\bar{\mathbf{x}}^j_{t,k}, \bar{\mathbf{u}}^j_{t,k}),\ B^j = D_{\mathbf{u}} f(\bar{\mathbf{x}}^j_{t,k}, \bar{\mathbf{u}}^j_{t,k}), \tag{11}$$

where $D_{\mathbf{x}}$ and $D_{\mathbf{u}}$ denote the Jacobian of the system dynamics $f(\mathbf{x}, \mathbf{u})$ with respect to the state $\mathbf{x}$ and the input $\mathbf{u}$. This approach allows to linearize the system at $(\bar{\mathbf{x}}^j_{t,k}, \bar{\mathbf{u}}^j_{t,k})$ locally between iterations. The convex system dynamics constraints are provided in (10) since all nominal vectors $(\bar{\mathbf{x}}^j_{t,k}, \bar{\mathbf{u}}^j_{t,k})$ in current iteration are constant and constructed from previous iteration $j-1$.

C. Linearization of DCBF & DHOCBF

In this section, we show how to linearize the DCBF up to the highest order. At iteration j, in order to linearize $h(\mathbf{x}^j_{t,k})$, an explicit line is projected in the state space to the nearest point $\tilde{\mathbf{x}}^j_{t,k}$ on the boundary of the obstacle from each state $\bar{\mathbf{x}}^j_{t,k}$. Note that $\bar{\mathbf{x}}^j_{t,k}$ is the nominal state vector from iteration $j-1$ for the linearization at iteration j, which means $\bar{\mathbf{x}}^j_{t,k} = \mathbf{x}^{j-1}_{t,k}$. The tangent line passing through the nearest point $\tilde{\mathbf{x}}^j_{t,k}$ is denoted as $h_{\|}(\mathbf{x}^j_{t,k}|\tilde{\mathbf{x}}^j_{t,k})$. This allows us to define a linearized safe set by $h_{\|}(\mathbf{x}^j_{t,k}|\tilde{\mathbf{x}}^j_{t,k}) \geq 0,\ \forall t \in \mathbb{N}$ as shown in Fig. 1 by the green region.

Remark 2. *Note that $\tilde{\mathbf{x}}^j_{t,k}$ represents the optimized value of the minimum distance problem with distance function $h(\cdot)$ between $\bar{\mathbf{x}}^j_{t,k}$ and safe set $\mathcal{C}$. For common smooth and differentiable CBFs, the expression of $\tilde{\mathbf{x}}^j_{t,k}$ as a function of $\bar{\mathbf{x}}^j_{t,k}$ is explicit [34], [35]. For example, when $h(\cdot)$ describes a l_2-norm function with the obstacle being a circular shape, $\tilde{\mathbf{x}}^j_{t,k}$ is exactly the intersection point between $\bar{\mathbf{x}}^j_{t,k}$ and the center of the obstacle. Notice that $\tilde{\mathbf{x}}^j_{t,k}$ could be implicit for general elliptic calculations [36], but it could still be numerically approximated as the values of $\bar{\mathbf{x}}^j_{t,k}$ known at iteration j before the linearization.*

The relative degree of $h_{\|}(\mathbf{x}^j_{t,k}|\tilde{\mathbf{x}}^j_{t,k})$ with respect to system (2) is still m when the relative degree of $h(\mathbf{x}^j_{t,k})$ is m. Thus, in order to guarantee safety with forward invariance based on Thm. 1 and Rem. 1, two sufficient conditions need to be satisfied: (1) the sequence of linearized DHOCBF $\tilde{\psi}_0(\cdot), \ldots, \tilde{\psi}_{m_{\text{cbf}}-1}(\cdot)$ is larger or equal to zero at the initial condition $\mathbf{x}_t$, and (2) the highest-order DCBF constraint $\tilde{\psi}_{m_{\text{cbf}}}(\mathbf{x}) \geq 0$ is always satisfied, where $\tilde{\psi}_i(\cdot)$ is defined as:

$$\begin{aligned} \tilde{\psi}_0(\mathbf{x}^j_{t,k}) &:= h_{\|}(\mathbf{x}^j_{t,k}|\tilde{\mathbf{x}}^j_{t,k}) \\ \tilde{\psi}_i(\mathbf{x}^j_{t,k}) &:= \tilde{\psi}_{i-1}(\mathbf{x}^j_{t,k+1}) - \tilde{\psi}_{i-1}(\mathbf{x}^j_{t,k}) + \gamma_i \tilde{\psi}_{i-1}(\mathbf{x}^j_{t,k}). \end{aligned} \tag{12}$$

Here, we have $0 < \gamma_i \leq 1$, $i \in \{1, \ldots, m_{\text{cbf}}\}$, and $m_{\text{cbf}} \leq m$ (as in (7)).

Remark 3. *From Rem. 1, it follows that that m_{cbf} is not necessarily equal to m. A detailed discussion on this can be found in [4], [28].*

An important issue is *feasibility*. It is possible that $\psi_i(\mathbf{x}^0_{t,k}) \geq 0,\ 1 \leq i \leq m_{\text{cbf}} - 1$, with $k \in \{0, \ldots, N\}$ is not satisifed since the linearized DHOCBF functions

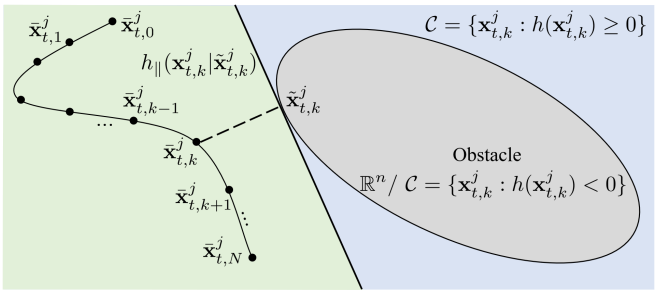


Fig. 1: Linearization of DHOCBF: $h_{\|}(\mathbf{x}^j_{t,k}|\tilde{\mathbf{x}}^j_{t,k}) \geq 0$ represents the linearized safe set locally and is colored in green. Note that $h_{\|}(\mathbf{x}^j_{t,k}|\tilde{\mathbf{x}}^j_{t,k}) \geq 0$ guarantees $h(\mathbf{x}^j_{t,k}) \geq 0$ (colored in blue plus green), which ensures collision avoidance (outside the grey region).

$\tilde{\psi}_0(\cdot), \ldots, \tilde{\psi}_{m_{\text{cbf}}-1}(\cdot)$ are more conservative than the original forms $\psi_0(\cdot), \ldots, \psi_{m_{\text{cbf}}-1}(\cdot)$. This problem can occur when the horizon is too large, or the linearization is too conservative. In order to handle this issue, we introduce a slack variable $\omega^j_{t,k,i}$ with a corresponding decay rate $(1-\gamma_i)$:

$$\tilde{\psi}_{i-1}(\mathbf{x}^j_{t,k+1}) \geq \omega^j_{t,k,i}(1-\gamma_i)\tilde{\psi}_{i-1}(\mathbf{x}^j_{t,k}),\ \omega^j_{t,k,i} \in \mathbb{R}, \tag{13}$$

where $i \in \{1, \ldots, m_{\text{cbf}}\}$. The slack variable $\omega^j_{t,k,i}$ is selected by minimizing a cost function term to satisfy DCBF constraints at initial condition at any time step [4].

Another challenge induced by the DCBF linearization is that the constraints in (13) could be non-convex, since now $\omega^j_{t,k,i}$ and $\mathbf{x}^j_{t,k}$ are both optimization variables. Note that $\tilde{\psi}_0(\mathbf{x}^j_{t,0})$ are constant, thus we can only place $\omega^j_{t,k,i}$ in front of $\tilde{\psi}_0(\mathbf{x}^j_{t,0})$ and move the other optimization variables to the other side of the inequalities. This motivates us to provide the following form for reformulating (13) as convex constraints:

$$\begin{aligned} &\tilde{\psi}_{i-1}(\mathbf{x}^j_{t,k}) + \sum_{\nu=1}^{i} Z_{\nu,i}(1-\gamma_i)^k \tilde{\psi}_0(\mathbf{x}^j_{t,\nu}) \geq \\ &\quad \omega^j_{t,k,i} Z_{0,i}(1-\gamma_i)^k \tilde{\psi}_0(\mathbf{x}^j_{t,0}), \\ &j \leq j_{\max} \in \mathbb{N}^+,\ i \in \{1, \ldots, m_{\text{cbf}}\},\ \omega^j_{t,k,i} \in \mathbb{R}. \end{aligned} \tag{14}$$

In the above, $Z_{\nu,i}$ is a constant that can be obtained recursively by reformulating $\tilde{\psi}_{i-1}(\cdot)$ back to $\tilde{\psi}_0(\cdot)$ given $\nu \in \{0, .., i\}$. We define $Z_{\nu,i}$ as follows. When $2 \leq i, \nu \leq i-2$, we have

$$\begin{aligned} Z_{\nu,i} = \sum_{l=1}^{l_{\max}} &[(\gamma_{\zeta_1} - 1)(\gamma_{\zeta_2} - 1) \cdots (\gamma_{\zeta_{i-\nu-1}} - 1)]_l, \\ &\zeta_1 < \zeta_2 < \cdots < \zeta_{i-\nu-1}, \zeta_s \in \{1, 2, \ldots, i-1\}, \end{aligned} \tag{15}$$

where $[\cdot]_l$ denotes the l^{th} combination of the product of the elements in parenthesis, therefore we have $l_{\max} = \binom{i-1}{i-\nu-1}$. ζ_s denote all ζ in (15). For the case $\nu = i-1$, if $2 \leq i$, we define $Z_{\nu,i} = -1$; if $i = 1$, we define $Z_{\nu,i} = 1$. Beside that, we define $Z_{\nu,i} = 0$ for the case $\nu = i$.

Remark 4. *The decay rate in* (14) *used by the iMPC-DHOCBF is partially relaxed compared to the one in* (13) *due to the requirement of the linearization. This can affect the feasibility of the optimization.*

D. CFTOC Problem

In Secs. III-B and III-C, we have illustrated the linearization of system dynamics as well as the safety constraints with DHOCBF. This allows us to consider them as constraints into a convex MPC formulation at each iteration, which we call convex finite-time constrained optimization control (CFTOC). This is solved at iteration j with optimization variables $\mathbf{U}_t^j = [\mathbf{u}_{t,0}^j, \dots, \mathbf{u}_{t,N-1}^j]$ and $\Omega_{t,i}^j = [\omega_{t,0,i}^j, \dots, \omega_{t,N,i}^j]$, where $i \in \{1, \dots, m_{\text{cbf}}\}$.

CFTOC of iMPC-DHOCBF at iteration j:

$$\min_{\mathbf{U}_t^j, \Omega_{t,1}^j, \dots, \Omega_{t,m_{\text{cbf}}}^j} \; p(\mathbf{x}_{t,N}^j) + \sum_{k=0}^{N-1} q(\mathbf{x}_{t,k}^j, \mathbf{u}_{t,k}^j, \omega_{t,k,i}^j) \tag{16a}$$

$$\text{s.t. } \mathbf{x}_{t,k+1}^j - \bar{\mathbf{x}}_{t,k+1}^j = A^j(\mathbf{x}_{t,k}^j - \bar{\mathbf{x}}_{t,k}^j) + B^j(\mathbf{u}_{t,k}^j - \bar{\mathbf{u}}_{t,k}^j), \tag{16b}$$

$$\mathbf{u}_{t,k}^j \in \mathcal{U},\ \mathbf{x}_{t,k}^j \in \mathcal{X},\ \omega_{t,k,i}^j \in \mathbb{R}, \tag{16c}$$

$$\tilde{\psi}_{i-1}(\mathbf{x}_{t,k}^j) + \sum_{\nu=1}^{i} Z_{\nu,i}(1-\gamma_i)^k \tilde{\psi}_0(\mathbf{x}_{t,\nu}^j) \geq \omega_{t,k,i}^j Z_{0,i}(1-\gamma_i)^k \tilde{\psi}_0(\mathbf{x}_{t,0}^j), \tag{16d}$$

In the CFTOC, the linearized dynamics constraints in (10) and the linearized DHOCBF constraints in (14) are enforced with constraints (16b) and (16d) at each open loop time step $k \in \{0, \dots, N-1\}$. The state and input constraints are considered in (16c). The slack variables are unconstrained as the goal of the optimization itself is to minimize the deviation from the nominal DHOCBF constraints with cost term $q(\cdot, \cdot, \omega_{t,k,i}^j)$, while ensuring feasibility of the optimization, as discussed in [32]. Note that, for ensuring the safety guarantee established by the DHOCBF, the constraints (16d) are enforced with $i \in \{0, \dots, m_{\text{cbf}}\}$, where $Z_{\nu,i} \in \mathbb{R}$ is as defined in (14) with $\nu \in \{0, .., i\}$. The optimal decision variables of (16) at iteration j is a list of control input vectors as $\mathbf{U}_t^{*,j} = [\mathbf{u}_{t,0}^{*,j}, \dots, \mathbf{u}_{t,N-1}^{*,j}]$ and a list of slack variable vectors as $\Omega_{t,i}^{*,j} = [\omega_{t,0,i}^{*,j}, \dots, \omega_{t,N-1,i}^{*,j}]$. The CFTOC is solved iteratively in our proposed iMPC-DHOCBF and the solution can be extracted once the convergence criteria or the maximum iteration number $j_{\max}$ is reached, as shown in Alg. 1.

IV. Case Study

In this section, we present numerical results to validate our proposed approach using a unicycle model. We provide a performance comparison with the baseline NMPC-DHOCBF approach. The NMPC-DHOCBF is simply extended by using relaxed DHOCBF based on (8d) in NMPC-DCBF (8), as discussed in [4, Rem. 4].

A. Numerical Setup

1) System Dynamics: Consider a discrete-time unicycle model in the form

$$\begin{bmatrix} x_{t+1}-x_t \\ y_{t+1}-y_t \\ \theta_{t+1}-\theta_t \\ v_{t+1}-v_t \end{bmatrix} = \begin{bmatrix} v_t \cos(\theta_t)\Delta t \\ v_t \sin(\theta_t)\Delta t \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} u_{1,t} \\ u_{2,t} \end{bmatrix}, \tag{17}$$

where $\mathbf{x}_t = [x_t, y_t, \theta_t, v_t]^T$ captures the 2-D location, heading angle, and linear speed; $\mathbf{u}_t = [u_{1,t}, u_{2,t}]^T$ represents angular velocity (u_1) and linear acceleration (u_2), respectively. The system is discretized with $\Delta t = 0.1$. System (17) is subject to the following state and input constraints:

$$\begin{aligned} \mathcal{X} &= \{\mathbf{x}_t \in \mathbb{R}^4 : -10 \cdot \mathcal{I}_{4\times 1} \leq \mathbf{x}_t \leq 10 \cdot \mathcal{I}_{4\times 1}\}, \\ \mathcal{U} &= \{\mathbf{u}_t \in \mathbb{R}^2 : [-7, -5]^T \leq \mathbf{u}_t \leq [7, 5]^T\}. \end{aligned} \tag{18}$$

2) System Configuration: The initial state is $[-3, 0, 0, 0]^T$ and the target state is $\mathbf{x}_r = [3, 0.01, 0, 0]^T$, which are marked as blue and red diamonds in Fig. 2. The circular obstacle is centered at $(0, 0)$ with $r = 1$,, which is displayed in orange. The other reference vectors are $\mathbf{u}_r = [0, 0]^T$ and $\omega_r = [1, 1]^T$. We use the offset $y = 0.01m$ in $\mathbf{x}_r$ to prevent singularity of the optimization problem.

3) DHOCBF: As a candidate DHOCBF function $\psi_0(\mathbf{x}_t)$, we choose a quadratic distance function for circular obstacle avoidance $h(\mathbf{x}_t) = (x_t - x_0)^2 + (y_t - y_0)^2 - r^2$, where (x_0, y_0) and r denote the obstacle center location and radius, respectively. The linearized DHOCBF $\tilde{\psi}_0(\mathbf{x}_{t,k}^j)$ in (12) is defined as $\tilde{\psi}_0(\mathbf{x}_{t,k}^j) \coloneqq h_{\parallel}(\mathbf{x}_{t,k}^j | \tilde{\mathbf{x}}_{t,k}^j)$, with

$$\begin{aligned} h_{\parallel}(\mathbf{x}_{t,k}^j | \tilde{\mathbf{x}}_{t,k}^j) = (\tilde{x}_{t,k}^j - x_0)x_{t,k}^j + (\tilde{y}_{t,k}^j - y_0)y_{t,k}^j \\ -(r^2 - x_0^2 - y_0^2 + \tilde{x}_{t,k}^j x_0 + \tilde{y}_{t,k}^j y_0), \end{aligned} \tag{19}$$

where $h_{\parallel}(\mathbf{x}_{t,k}^j | \tilde{\mathbf{x}}_{t,k}^j)$ is the linearized boundary, whose relative degree is 2; $(\tilde{x}_{t,k}^j, \tilde{y}_{t,k}^j)$ denotes the tangent point of the circular boundary $h(\mathbf{x}_t)$. From (15), we have $Z_{0,2} = \gamma_1 - 1,\ Z_{1,2} = -1,\ Z_{0,1} = 1,\ Z_{2,2} = Z_{1,1} = 0$.

4) MPC Design: The cost function of the MPC problem consists of stage cost $q(\mathbf{x}_{t,k}^j, \mathbf{u}_{t,k}^j, \omega_{t,k}^j) = \sum_{k=0}^{N-1}(||\mathbf{x}_{t,k}^j - \mathbf{x}_r||_Q^2 + ||\mathbf{u}_{t,k}^j - \mathbf{u}_r||_R^2 + ||\omega_{t,k}^j - \omega_r||_S^2)$ and terminal cost $p(\mathbf{x}_{t,N}^j) = ||\mathbf{x}_{t,N}^j - \mathbf{x}_r||_P^2$, where $Q = P = 10 \cdot \mathcal{I}_4, R = \mathcal{I}_2$ and $S = 1000 \cdot \mathcal{I}_2$.

5) Convergence Criteria: We use the following absolute and relative convergence functions as convergence criteria mentioned in Alg. 1:

$$\begin{aligned} e_{\text{abs}}(\mathbf{X}_t^{*,j}, \mathbf{U}_t^{*,j}) &= ||\mathbf{X}^{*,j} - \bar{\mathbf{X}}^{*,j}|| \\ e_{\text{rel}}(\mathbf{X}_t^{*,j}, \mathbf{U}_t^{*,j}, \bar{\mathbf{X}}_t^j, \bar{\mathbf{U}}_t^j) &= ||\mathbf{X}^{*,j} - \bar{\mathbf{X}}^{*,j}||/||\bar{\mathbf{X}}^{*,j}||. \end{aligned} \tag{20}$$

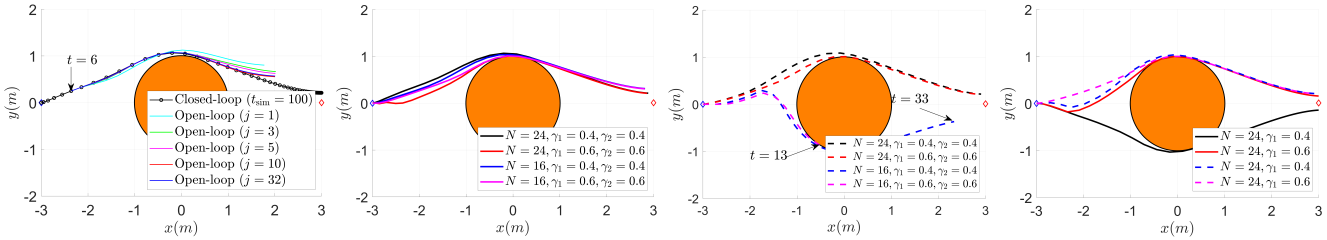
The iterative optimization stops when $e_{\text{abs}} < \varepsilon_{\text{abs}}$ or $e_{\text{rel}} < \varepsilon_{\text{rel}}$, where $\varepsilon_{\text{abs}} = 10^{-4}$, $\varepsilon_{\text{rel}} = 10^{-2}$ and the maximum iteration number is set as $j_{\max} = 1000$.

To make a fair comparison with NMPC-DHOCBF, the hyperparameters P, Q, R, S remain unchanged for all setups.

6) Solver Configurations and CPU Specs.: For iMPC-DHOCBF, we used OSQP [37] to solve the convex optimizations at all iterations. The baseline approach NMPC-DHOCBF is open-source, and was solved using IPOPT [38] with the modeling language Yalmip [39]. We used a Windows desktop with Intel Core i7-8700 (CPU 3.2 GHz) running Matlab for all computations.

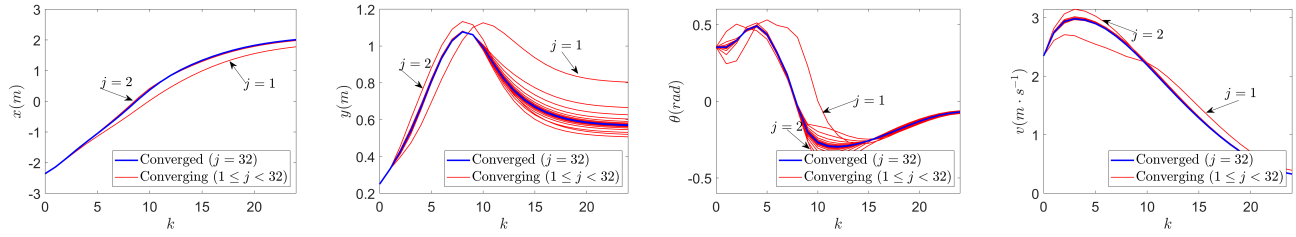
B. Performance

1) Iterative Convergence: The iterative convergence is shown in Figs. 2a, 4 and 3. Fig. 2a shows the closed-loop



(a) iMPC-DHOCBF when $N = 24$, $\gamma_1 = \gamma_2 = 0.4$. (b) iMPC-DHOCBF with $m_{\text{cbf}} = 2$. (c) NMPC-DHOCBF with $m_{\text{cbf}} = 2$. (d) iMPC-DHOCBF and NMPC-DHOCBF with $m_{\text{cbf}} = 1$.

Fig. 2: Open-loop and closed-loop trajectories with controllers iMPC-DHOCBF (solid lines) and NMPC-DHOCBF (dashed lines): (a) several open-loop trajectories at different iterations predicted at $t = 6$ and one closed-loop trajectory with controller iMPC-DHOCBF; (b) closed-loop trajectories with controller iMPC-DHOCBF with different choices of N and γ; (c) closed-loop trajectories with controller NMPC-DHOCBF with different choices of N and γ. Note that two trajectories stop at $t = 13$ and $t = 33$ because of infeasibility; (d) closed-loop trajectories with controllers iMPC-DHOCBF and NMPC-DHOCBF with $m_{\text{cbf}} = 1$. Both methods work well for safety-critical navigation.



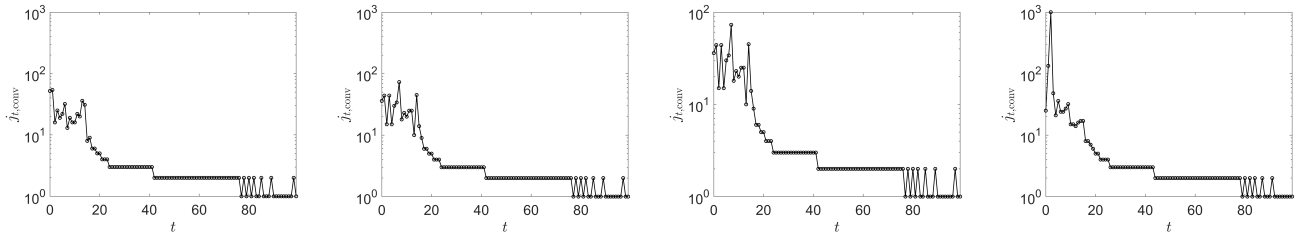
(a) Location x (b) Location y (c) Orientation θ (d) Speed v

Fig. 3: Iterative convergence of all states at converged iteration $j_{6,\text{conv}} = 32$ with $N = 24, m_{\text{cbf}} = 2$, $\gamma_1 = \gamma_2 = 0.4$. iMPC does help to optimize the cost function to reach local optimal minimum.

trajectory (the black line) generated by solving the iMPC-DHOCBF until the converged iteration $j_{t,\text{conv}}$ from $t = 0$ to $t = t_{\text{sim}} = 100$ and open-loop iteratively converging trajectories (colored lines) at different iterations at $t = 6$. Fig. 3 presents more details on the iterative convergence of states at different iterations at $t = 6$ with number of iterations $j_{t,\text{conv}} = 32$. We note that, after around 10 iterations, the converging lines for the states (red lines) nearly overlap with the converged line (blue line) in Fig. 3. This verifies the relations of the converging trajectory (red line) and the converged trajectory (blue line) in Fig. 2a. The optimization is shown to converge at iteration $j_{t,\text{conv}}$ at time step t for different hyperparameters γ under specific convergence criteria (20), shown in Fig. 4. We can see that for the first 15 time steps the iMPC-DHOCBF triggers more iterations to drive the system to avoid the obstacle than time steps after 20 where the system already passes the obstacle. The maximum converged iteration $j_{t,\text{conv}}$ is 1000 at time step $t = 2$ in Fig. 4d with $\gamma_1 = \gamma_2 = 0.6$, which reveals that the peak of the converged iteration over time increases if we choose larger γ. For the majority of the time-steps, the iterative optimization converges within 100 iterations ($j_{t,\text{conv}} < 100$).

2) Convergence with Different Hyperparameters: Fig. 2b, 2c and 2d show the closed-loop trajectories generated by solving iMPC-DHOCBF (solid lines) and NMPC-DHOCBF (dashed lines) at converged iteration $j_{t,\text{conv}}$ from $t = 0$ to $t = t_{\text{sim}} = 45$ with different hyperparameters. Both controllers show good performance on obstacle avoidance. Based on black, red, blue and magenta lines with the highest order of CBF constraint $m_{\text{cbf}} = 2$ in Fig. 2b and 2c, as γ_1, γ_2 become smaller, the system tends to turn further away from the obstacle when it is getting closer to obstacle, which indicates a safer control strategy. From the lines in Fig. 2d where $m_{\text{cbf}} = 1$, we can see that the system can still safely navigate around the obstacle, although it turns away from the obstacle later than when having one more CBF constraint in Fig. 2b and 2c, indicating that having CBF constraints up to the relative degree enhances safety. The blue and magenta dashed lines in Fig. 2c stop at $t = 33$ and $t = 13$ with $N = 16$ as infeasibility happens, which shows that a large horizon is needed to generate complete closed-loop trajectories for some hyperparameters by NMPC-DHOCBF, while iMPC-DHOCBF shows less reliance on selection of horizon since it can generate complete closed-loop trajectories with both $N = 16$ and $N = 24$, as shown in Fig. 2b.

3) Computation Time: In order to compare computational times between our proposed iMPC-DHOCBF and the baseline NMPC-DHOCBF, 1000 independent randomized safe states are generated in state constraint $\mathcal{X}$ in (18). To make a fair comparison, both approaches use the same N and m_{cbf} and the computational time and feasibility are evaluated at those randomized sample states. The distributions of the computation times and infeasibility rates in Tab. I and Tab. II correspond to generating one time-step trajectories. The mean and standard deviation of computation times increase if the horizon N or m_{cbf} become larger for NMPC-DHOCBF



(a) $\gamma_1 = 0.4, \gamma_2 = 0.4$ (b) $\gamma_1 = 0.4, \gamma_2 = 0.6$ (c) $\gamma_1 = 0.6, \gamma_2 = 0.4$ (d) $\gamma_1 = 0.6, \gamma_2 = 0.6$

Fig. 4: Number of iterations $j_{t,\text{conv}}$ at each time-step using controller iMPC-DHOCBF with different values of hyperparameters γ_1, γ_2 with $N = 24, m_{\text{cbf}} = 2$. We can observe that, for almost all time-steps, the iterative optimization converges within 100 iteraitons ($j_{t,\text{conv}} < 10^2$), which is affected very little with respect to hyperparameters.

Approaches		$N = 4$	$N = 8$	$N = 12$	$N = 16$	$N = 20$	$N = 24$
NMPC-DHOCBF	mean / std (s)	3.687 ± 6.360	23.882 ± 17.988	27.329 ± 20.115	28.953 ± 22.058	30.970 ± 23.564	29.929 ± 22.105
($m_{\text{cbf}} = 2$)	infeas. rate	5.8%	27.5%	21.1%	16.4%	14.5%	14.4%
NMPC-DHOCBF	mean / std (s)	2.933 ± 4.678	19.077 ± 14.024	20.418 ± 15.401	22.749 ± 17.039	24.053 ± 17.811	25.365 ± 18.211
($m_{\text{cbf}} = 1$)	infeas. rate	6.3%	13.9%	13.0%	14.6%	13.8%	15.4%
iMPC-DHOCBF	mean / std (s)	0.135 ± 0.294	0.104 ± 0.242	0.102 ± 0.217	0.131 ± 0.301	0.165 ± 0.400	0.135 ± 0.274
($m_{\text{cbf}} = 2$)	infeas. rate	6.3%	8.0%	10.4%	10.9%	10.9%	10.2%
iMPC-DHOCBF	mean / std (s)	0.131 ± 0.286	0.114 ± 0.260	0.109 ± 0.237	0.137 ± 0.316	0.173 ± 0.414	0.152 ± 0.317
($m_{\text{cbf}} = 1$)	infeas. rate	6.3%	8.0%	10.4%	10.9%	10.9%	11.1%

TABLE I: Statistical benchmark for computation time and feasibility between NMPC-DHOCBF and iMPC-DHOCBF with randomized states. The target position is shared among four approaches and the hyperparameters are fixed as $\gamma_1 = \gamma_2 = 0.4$ for all random scenarios.

Approaches		$N = 4$	$N = 8$	$N = 12$	$N = 16$	$N = 20$	$N = 24$
NMPC-DHOCBF	mean / std (s)	3.744 ± 6.445	28.779 ± 20.755	31.319 ± 21.921	33.678 ± 25.328	36.430 ± 26.959	39.543 ± 29.941
($m_{\text{cbf}} = 2$)	infeas. rate	5.6%	28.0%	20.9%	16.8%	17.0%	14.6%
NMPC-DHOCBF	mean / std (s)	3.032 ± 4.536	21.414 ± 16.518	23.121 ± 17.544	24.011 ± 17.711	26.599 ± 19.480	29.671 ± 20.026
($m_{\text{cbf}} = 1$)	infeas. rate	6.4%	17.0%	15.2%	15.5%	16.7%	13.2%
iMPC-DHOCBF	mean / std (s)	0.158 ± 0.326	0.134 ± 0.279	0.163 ± 0.353	0.163 ± 0.373	0.184 ± 0.398	0.164 ± 0.344
($m_{\text{cbf}} = 2$)	infeas. rate	6.1%	8.0%	10.2%	10.7%	10.8%	10.8%
iMPC-DHOCBF	mean / std (s)	0.167 ± 0.340	0.139 ± 0.291	0.170 ± 0.362	0.170 ± 0.379	0.201 ± 0.435	0.176 ± 0.378
($m_{\text{cbf}} = 1$)	infeas. rate	6.1%	8.0%	10.2%	10.7%	10.8%	10.8%

TABLE II: Statistical benchmark between NMPC-DHOCBF and iMPC-DHOCBF with the same randomized states as in Tab. I. The target position is shared among four approaches and the hyperparameters are fixed as $\gamma_1 = \gamma_2 = 0.6$ for all scenarios. Based on Tab. I and Tab. II we conclude that iMPC-DHOCBF outperforms NMPC-DHOCBF in computing time and infeasibility rate.

in Tab. I and Tab. II. Different from NMPC-DHOCBF, the computing time is not heavily influenced by N and m_{cbf} for iMPC-DHOCBF. Based on the data from the two tables, we also notice that larger hyperparameter values for γ will slightly reduce the computation speed for both methods, which is discussed in Sec. IV-B.1 and can be attributed to the rise of converged iteration $j_{t,\text{conv}}$. Compared to NMPC-DHOCBF, the computing speed of our proposed method is much faster with the improvement in computation time directly proportional to the horizon, *e.g.*, $100 \sim 300$ times faster than the baseline given the chosen hyperparameters.

4) Optimization Feasibility: The rate of infeasibility increases if the horizon N increases or m_{cbf} is lower for iMPC-DHOCBF. However, these two hyperparameters are shown not to affect the infeasibility rate of the NMPC-DHOCBF method proportionally. As the horizon increases, the infeasibility rate of iMPC-DHOCBF outperforms that of NMPC-DHOCBF. The main reasons for this come from the difference in the convergence criteria and relaxation techniques for CBF constraints, discussed in Rem. 4. The NMPC-DHOCBF under IPOPT should have more strict convergence criteria compared to iMPC-DHOCBF, which obviously limits its feasibility if the number of horizon is large. Besides, NMPC-DHOCBF is equipped with relaxed nonlinear CBF constraints (13), while iMPC-DHOCBF has relaxed linear CBF constraints (14). The linearization of the CBF constraints reduces the feasibility region in the state space, as illustrated in Fig. 1. This meets the expectation of slight decreased feasibility rate of iMPC-DHOCBF when number of horizon is small. However, we can see that the decline in feasibility rate due to relaxed technique is noticeably outperformed by flexible convergence criteria with larger number of horizon N in Tab. I and II, which validates our linearization technique in the iterative optimization.

V. Conclusion & Future Work

We proposed an iterative convex optimization procedure for safety-critical model predictive control (iMPC) design. Central to our approach are relaxations for the system dynamics and for discrete time high-order control barrier functions (DHOCBF) in the form of linearized constraints. We validated the proposed iMPC-DHOCBF approach by applying it to a model of unicycle navigating in an environment with obstacles. We noticed that the computation times for the iMPC-DHOCBF method significantly outperform the

ones corresponding to the baseline, usually with even higher feasibility rate. There are still some limitations of iMPC-DHOCBF that could be ameliorated. One limitation of the proposed method is its linearly relaxed technique will slightly increase infeasibility rate with small size of the horizon. Another limitation is that the feasibility of the optimization and system safety are not always guaranteed at the same time in the whole state space. We will address these limitations in future work with better linearization, different relaxed techniques as well as adaptive warm-up and convergence criterion for the optimization problem.

REFERENCES

[1] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 6271–6278.

[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, 2016.

[3] J. Zeng, B. Zhang, Z. Li, and K. Sreenath, "Safety-critical control using optimal-decay control barrier function with guaranteed point-wise feasibility," in *2021 American Control Conference (ACC)*, 2021, pp. 3856–3863.

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

[5] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM Journal on control and optimization*, vol. 43, no. 5, pp. 1714–1736, 2005.

[6] G. Paolo, I. Ferrara, and L. Magni, "Mpc for robot manipulators with integral sliding mode generation," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 3, pp. 1299–1307, 2017.

[7] N. Scianca, D. De Simone, L. Lanari, and G. Oriolo, "Mpc for humanoid gait generation: Stability and feasibility," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1171–1188, 2020.

[8] R. Grandia, A. J. Taylor, A. Singletary, M. Hutter, and A. D. Ames, "Nonlinear model predictive control of robotic systems with control lyapunov functions," in *Proceedings of Robotics: Science and Systems*, 2022.

[9] T. D. Son and Q. Nguyen, "Safety-critical control for non-affine nonlinear systems with application on autonomous vehicle," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 7623–7628.

[10] J. M. Eklund, J. Sprinkle, and S. S. Sastry, "Switched and symmetric pursuit/evasion games using online model predictive control with application to autonomous aircraft," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 604–620, 2012.

[11] J. V. Frasch, A. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl, "An auto-generated nonlinear mpc algorithm for real-time obstacle avoidance of ground vehicles," in *2013 European Control Conference (ECC)*, 2013, pp. 4136–4141.

[12] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1: 43 scale rc cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[13] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 972–983, 2020.

[14] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle, "Torque saturation in bipedal robotic walking through control lyapunov function-based quadratic programs," *IEEE Access*, vol. 3, pp. 323–332, 2015.

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

[16] Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath, "3d dynamic walking on stepping stones with control barrier functions," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 827–834.

[17] W. Xiao, C. Belta, and C. G. Cassandras, "Adaptive control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2267–2281, 2021.

[18] Q. Nguyen and K. Sreenath, "Optimal robust control for bipedal robots through control lyapunov function based quadratic programs," in *Robotics: Science and Systems (RSS)*, Rome, Italy, July 2015.

[19] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, 2018.

[20] A. Clark, "Control barrier functions for stochastic systems," *Automatica*, vol. 130, p. 109688, 2021.

[21] C. Dawson, Z. Qin, S. Gao, and C. Fan, "Safe nonlinear control using robust neural lyapunov-barrier functions," in *Conference on Robot Learning*. PMLR, 2022, pp. 1724–1735.

[22] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*, 2016, pp. 322–328.

[23] W. Xiao and C. Belta, "High-order control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2022.

[24] X. Tan, W. S. Cortez, and D. V. Dimarogonas, "High-order barrier functions: Robustness, safety, and performance-critical control," *IEEE Transactions on Automatic Control*, vol. 67, no. 6, pp. 3021–3028, 2021.

[25] J. Usevitch and D. Panagou, "Adversarial resilience for sampled-data systems under high-relative-degree safety constraints," *IEEE Transactions on Automatic Control*, pp. 1–1, 2022.

[26] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation." in *Robotics: Science and Systems*, vol. 13. Cambridge, MA, USA, 2017.

[27] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, "Multi-layered safety for legged robots via control barrier functions and model predictive control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 8352–8358.

[28] H. Ma, X. Zhang, S. E. Li, Z. Lin, Y. Lyu, and S. Zheng, "Feasibility enhancement of constrained receding horizon control using generalized control barrier function," in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, 2021, pp. 551–557.

[29] Y. Xiong, D.-H. Zhai, M. Tavakoli, and Y. Xia, "Discrete-time control barrier function: High-order case and adaptive case," *IEEE Transactions on Cybernetics*, pp. 1–9, 2022.

[30] S. He, J. Zeng, and K. Sreenath, "Autonomous racing with multiple vehicles using a parallelized optimization with safety guarantee using control barrier functions," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 3444–3451.

[31] Z. Li, J. Zeng, A. Thirugnanam, and K. Sreenath, "Bridging model-based safety and model-free reinforcement learning through system identification of low dimensional linear models," in *Proceedings of Robotics: Science and Systems*, 2022.

[32] J. Zeng, B. Zhang, Z. Li, and K. Sreenath, "Safety-critical control using optimal-decay control barrier function with guaranteed point-wise feasibility," in *2021 American Control Conference (ACC)*, 2021, pp. 3856–3863.

[33] M. Sun and D. Wang, "Initial shift issues on discrete-time iterative learning control with system relative degree," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 144–148, 2003.

[34] A. Thirugnanam, J. Zeng, and K. Sreenath, "Duality-based convex optimization for real-time obstacle avoidance between polytopes with control barrier functions," in *2022 American Control Conference (ACC)*, 2022, pp. 2239–2246.

[35] ——, "Safety-critical control and planning for obstacle avoidance between polytopes with control barrier functions," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 286–292.

[36] I. V. Skrypnik, *Methods for analysis of nonlinear elliptic boundary value problems*. American Mathematical Soc., 1994, vol. 139.

[37] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "Osqp: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[38] L. T. Biegler and V. M. Zavala, "Large-scale nonlinear programming using ipopt: An integrating framework for enterprise-wide dynamic optimization," *Computers & Chemical Engineering*, vol. 33, no. 3, pp. 575–582, 2009.

[39] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*, 2004, pp. 284–289.