2024 American Control Conference (ACC)
July 8-12, 2024. Toronto, Canada

Model Predictive Control Barrier Functions: Guaranteed Safety with Reduced Conservatism and Shortened Horizon

Hossein Abdi[1], Pan Zhao[2], Naira Hovakimyan[3], and Reza Ghabcheloo[1]

***Abstract*— In this study, we address the problem of safe control in systems subject to state and input constraints by integrating the Control Barrier Function (CBF) into the Model Predictive Control (MPC) formulation. While CBF offers a conservative policy and traditional MPC lacks the safety guarantee beyond the finite horizon, the proposed scheme takes advantage of both MPC and CBF approaches to provide a guaranteed safe control policy with reduced conservatism and a shortened horizon. The proposed methodology leverages the sum-of-square (SOS) technique to construct CBFs that make forward invariant safe sets in the state space that are then used as a terminal constraint on the last predicted state. CBF invariant sets cover the state space around system fixed points. These islands of forward invariant CBF sets will be connected to each other using MPC. To do this, we proposed a technique to handle the MPC optimization problem subject to the combination of intersections and union of constraints. Our approach, termed Model Predictive Control Barrier Functions (MPCBF), is validated using numerical examples to demonstrate its efficacy, showing improved performance compared to classical MPC and CBF.**

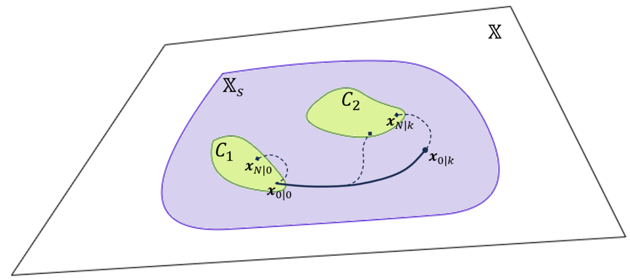

Fig. 1. Model Predictive Control Barrier Functions (MPCBF). The white polygon ($\mathbb{X} \subseteq \mathbb{R}^n$) indicates the feasible state space for the system (1). The purple area ($\mathbb{X}_s \subseteq \mathbb{X}$) is the set of safe states. And, green zones (C_j) show the created forward-invariant sets using the concept of CBF. The solid and dashed curves demonstrate the real and predicted trajectory of the system, respectively. In each time step, the MPCBF forces the predicted state at the last step ($\boldsymbol{x}_{N|k}$) to enter one of the forward-invariant sets while the states at preceding steps can still remain outside any of the invariant sets to reduce the conservativeness.

I. INTRODUCTION

In recent years, Model Predictive Control (MPC) and Control Barrier Function (CBF) have been widely adopted for robotic control subject to safety constraints. MPC is a control policy in which the current control action is obtained by solving a finite horizon constrained optimal control problem at each time-step [1]. Although MPC provides a straightforward way for formulating state/input constraints, it lacks the safety guarantee beyond the finite and short horizon due to inaccurate approximation of terminal cost function and the absence of an invariant set as the terminal constraint. One solution for this problem is to use a longer horizon, which will increase complexity and the computational cost in the optimization [2]. CBFs, on the other hand, are strong mathematical tools that reactively map the constraints defined over the states directly onto control constraints [3]. The idea of safe control using CBFs was first introduced by [4] which was further inspired by the barrier certificates presented by [5], [6]. Many studies have recently emerged addressing the safe control issues by CBFs in miscellaneous applications: Autonomous Cars and Mobile Robots [7], [8], [9], [10]; Multi-robot Systems [11], [12]; Quadrotors [13], [14]; and Legged Robots [15], [16].

Despite the widespread utilization of CBFs on safety-critical systems, the validation through mathematical proof of the CBF that has been put into practical use remains an unresolved matter in numerous research investigations. In recent years, few systematic methods based on SOS optimization have been suggested for the creation of valid CBFs [17], [18]. CBFs restrain the system states to *always stay* in an invariant set that is generally a subset of the safety set, which is quite conservative. Even though one could possibly increase the invariant set with new methods, however, the restriction of the system states to always stay in the invariant set is a fundamental limit and still causes conservatism. Consequently, despite verifying and enforcing the safety properties that CBF can offer, it is still a challenging issue to find a less conservative invariant set.

A. Contribution

To address the deficiencies of the CBF and MPC, this paper puts forth a systematic methodology to provide analytical safety assurance for the safety-critical systems subject to input and state constraints. In particular, our proposed technique leverages the SOS-based Process introduced in our previous work [17] to synthesize CBFs that produce analytically valid yet conservative forward invariant safe sets.

*This work was supported in part by Academy of Finland (SA) 345517, under SA-NSF joint call on artificial intelligence and wireless communication, and in part by NSF through the RI Grant 2133656.

[1] Hossein Abdi and Reza Ghabcheloo are with the Faculty of Engineering and Natural Sciences, Tampere University, Finland. {hossein.abdi, reza.ghabcheloo}@tuni.fi

[2]Pan Zhao is with the Department of Aerospace Engineering and Mechanics, University of Alabama, Tuscaloosa, AL 35487 USA. pan.zhao@ua.edu

[3] Naira Hovakimyan is with the Department of Mechanical Science and Engineering, University of, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA. nhovakim@illinois.edu

These safe sets are considered as a terminal constraint in Model Predictive Control formulation. We generate the safe sets offline based on the fixed points of the system. Since the CBF provides a forward invariant safe set for the states beyond the horizon, the safety of the system will be assured even in a short-horizon MPC. To reduce conservatism and connect these islands of forward invariant sets, a Model Predictive Controller is employed to provide safe trajectories out of invariant sets. Consequently, the introduced control methodology (MPCBF) generates a non-conservative as well as safety-guaranteed trajectory in the state space.

B. Related Works

In recent years, CBF has been incorporated into MPC in several studies. In [19] the barrier functions are incorporated into the cost function of the model predictive controller to convert the constrained MPC framework to an unconstrained one. In their proposed method, the drawbacks of MPC still exist if there is a finite horizon with an inaccurate approximation of terminal cost and lack of an invariant set as the terminal constraint.

There exist efforts to implement the idea of incorporating CBF into MPC for different nonlinear systems in [2], [20], [21], and [22] where CBF is appended as a safety constraint for all predicted states in MPC. In the methodology they have introduced, however, one can discern the inherent disadvantages of MPC, as MPC inherently ensures safety within a finite horizon, while the pivotal aspect of ensuring safety is for trajectories beyond the horizon.

In [23] the MPC and CBF are utilized separately in two different levels of control architecture. While MPC is utilized as a planner to enforce high-level safety and tracking, the CBF is employed to enforce low-level safety and tracking.

A novel approach named predictive safety filter is proposed in [24] where the concept of MPC-CBF is employed to filter the nominal control input based on safety concerns. In this work, the predicted states at the last step are forced to enter the invariant set defined by a single CBF. The idea of the predictive safety filter was first proposed in [25] in which MPC exclusively has been utilized for safety guarantee. This idea was implemented on a real-world car-like robot in [26].

C. Organization of the Paper

The paper is organized as follows: In section II, we provide a review of Model Predictive Control (MPC), Control Barrier Function (CBF), and SOS-based technique for synthesizing CBFs and invariant sets. Then, in section III, we discussed our control methodology MPCBF. The proposed safe control scheme has been validated, and the numerical results are discussed in section IV. Finally, section V concludes the paper.

D. Definition and Notation

Definition 1: A control system $\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k)$ is considered safe if $\boldsymbol{x}_k \in \mathbb{X}_s \subseteq \mathbb{X}$, and $\boldsymbol{u}_k \in \mathbb{U}$, $\forall k \in \mathbb{Z}_{\geq 0}$. Where $\mathbb{X}$ and $\mathbb{X}_s$ denote the sets of all feasible states and safe states, respectively. Furthermore, the $\mathbb{U}$ represents the allowable set of control inputs.

Definition 2: An arbitrary set $A \subseteq \mathbb{R}^n$ is called a forward or positively invariant set for a continuous dynamic system $\dot{\boldsymbol{x}} = f(\boldsymbol{x}(t))$ if for any $\boldsymbol{x}(0) \in A$, it satisfies $\boldsymbol{x}(t) \in A$, $\forall t \geq 0$. Furthermore, the set A is called a controlled invariant set for the system $\dot{\boldsymbol{x}} = f(\boldsymbol{x}(t), \boldsymbol{u}(t))$ if there exists a continuous feedback control law $\boldsymbol{u}(t)$ that assures the existence and uniqueness of the solution on $t \in \mathbb{R}^+$ such that A is positively invariant for the closed-loop system [27].

Definition 3: A continuous function $\alpha(\cdot) : [0, a) \rightarrow [0, \infty)$, $a > 0$ is a class κ function if it is strictly increasing and $\alpha(0) = 0$. The function $\alpha(\cdot)$ is also a class κ_∞ function if it belongs to class κ as well as $a = \infty$ and $\lim_{r\to\infty} \alpha(r) = \infty$. Moreover, a continuous function $\beta(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ is said to belong to extended class κ_∞ if it is strictly increasing and $\beta(0) = 0$.

Definition 4: A function $f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^m$ is locally Lipschitz continuous in D if there exists a constant $M > 0$ such that for all $x \in D$, the Lipschitz condition holds: $\|f(x) - f(x')\| \leq M\|x - x'\|$ holds.

Definition 5: The common notation $L_\xi \eta(x)$ is utilized for the Lie derivative of $\eta(x)$ along the vector field $\xi(x)$ i.e. $L_\xi \eta(x) = \frac{\partial \eta(x)}{\partial x} \xi(x)$.

II. Preliminaries and Background

Considering an affine nonlinear control system and its discretized version as below:

$$\dot{\boldsymbol{x}} = f_c(\boldsymbol{x}(t)) + g_c(\boldsymbol{x}(t))\boldsymbol{u}(t) \tag{1a}$$

$$\boldsymbol{x}_{k+1} = f_d(\boldsymbol{x}_k) + g_d(\boldsymbol{x}_k)\boldsymbol{u}_k \tag{1b}$$

where $\boldsymbol{x}(t) \in \mathbb{X} \subseteq \mathbb{R}^n$ (equivalently for discrete-time system: $\boldsymbol{x}_k \in \mathbb{X} \subseteq \mathbb{R}^n$) is state vector, $\boldsymbol{u}(t) \in \mathbb{U} \subseteq \mathbb{R}^m$ ($\boldsymbol{u}_k \in \mathbb{U} \subseteq \mathbb{R}^m$) denotes the control input at the $t \in \mathbb{R}^+$ ($k \in \mathbb{Z}_{\geq 0}$). Also, f_c, g_c, f_d, and g_d are locally Lipschitz continuous functions.

A. Model Predictive Control (MPC)

Consider a discrete-time control system (1), where we omit the subscript d for simplicity in the rest of the paper. Providing that the system is full-state observable, the finite-time optimal control problem can be formulated as (2). In this formulation, $\boldsymbol{x}_{i|k}$ and $\boldsymbol{u}_{i|k}$ are the i^{th} predicted state and control input given the system information at time-step k, respectively. The optimization problem solution at each horizon provides us with the optimal control sequence $\boldsymbol{u}^*_{0:N-1|k}$. Based on the principle concept of the receding horizon scheme, the first element of the computed $\boldsymbol{u}^*_{0:N-1|k}$ is applied to the system as the control command.

$$\boldsymbol{u}^*_{0:N-1|k} = \underset{\boldsymbol{u}_{0:N-1}}{\operatorname{argmin}} \sum_{i=0}^{N-1} J(\boldsymbol{x}_{i|k}, \boldsymbol{u}_{i|k}) + J_N(\boldsymbol{x}_{N|k}) \tag{2a}$$

$$\text{s.t.} \quad \forall\ i \in \{0, \dots, N-1\}:$$

$$\boldsymbol{x}_{i+1|k} = f(\boldsymbol{x}_{i|k}) + g(\boldsymbol{x}_{i|k})\boldsymbol{u}_{i|k}, \tag{2b}$$

$$\boldsymbol{x}_{0|k} = \boldsymbol{x}_k, \tag{2c}$$

$$\boldsymbol{x}_{i|k} \in \mathbb{X}_s, \tag{2d}$$

$$\boldsymbol{u}_{i|k} \in \mathbb{U}. \tag{2e}$$

where $J(\boldsymbol{x}_{i|k}, \boldsymbol{u}_{i|k})$ denotes the stage cost function and $J_N(\boldsymbol{x}_{N|k})$ represents the terminal cost function as an approximation of cost for the rest of the prediction beyond the finite horizon. The dynamic model of the system is inserted as the constraint (2b). The state condition at $i = 0$ in each time-step k is the current measured/observed states (2c). All the predicted states and control inputs are constrained to be in the defined allowable domain of the system (2d),(2e).

B. Control Barrier Function (CBF)

Let $C \subseteq \mathbb{X}_s$ be zero-superlevel set of a smooth and continuously differentiable function $h(\boldsymbol{x}) : \mathbb{X} \rightarrow \mathbb{R}$ that satisfies the following conditions:

$$C = \{\boldsymbol{x} \in \mathbb{X} | h(\boldsymbol{x}) \geq 0\} \tag{3a}$$

$$\partial C = \{\boldsymbol{x} \in \mathbb{X} | h(\boldsymbol{x}) = 0\} \tag{3b}$$

$$Int(C) = \{\boldsymbol{x} \in \mathbb{X} | h(\boldsymbol{x}) > 0\} \tag{3c}$$

where ∂C is the set boundary, and $Int(C)$ is interior of set C. To guarantee that a continuous control system will remain safe for all time, we should make the set C a forward invariant set.

It can be mathematically proven that the function $h(\boldsymbol{x})$ is a valid CBF if there exists an extended class κ_∞ function $\alpha(\cdot)$ such that for the control system [3]:

$$\sup_{\boldsymbol{u} \in \mathbb{U}} [L_{f_c} h(\boldsymbol{x}) + L_{g_c} h(\boldsymbol{x})\boldsymbol{u}] \geq -\alpha(h(\boldsymbol{x})) \tag{4}$$

where $\dot{h}(\boldsymbol{x}, \boldsymbol{u}) = L_{f_c} h(\boldsymbol{x}) + L_{g_c} h(\boldsymbol{x})\boldsymbol{u}$ and L_{f_c}, L_{g_c} denote Lie derivatives. Therefore, satisfying this condition will make the set C a forward invariant set; thus, for all $\boldsymbol{x}_0 \in C$, trajectories will remain inside C.

C. CBF Synthesis Using SOS-based Technique

Based on [17], a continuous control system can be written in a state-dependent linear-like representation as below:

$$\dot{\boldsymbol{x}} = A(\boldsymbol{x})z(\boldsymbol{x}) + B(\boldsymbol{x})\boldsymbol{u} \tag{5}$$

where $A(\boldsymbol{x}) \in \mathbb{R}^{n \times N_z}$ is a polynomial matrix, and $z(\boldsymbol{x}) \in \mathbb{R}^{N_z}$ is a vector of monomials such that $z(\boldsymbol{x}) = 0$ iff $\boldsymbol{x} = 0$. Assume that the control system 5 is subject to state and input constraints approximated by:

$$\boldsymbol{x}(t) \in \mathbb{X}_s \triangleq \{\boldsymbol{x} \in \mathbb{R}^n : |C_i(\boldsymbol{x})z(\boldsymbol{x})| \leq 1, i = 1, ..., p\} \tag{6a}$$

$$\boldsymbol{u}(t) \in \mathbb{U} \triangleq \{\boldsymbol{u} \in \mathbb{R}^m : |D_j \boldsymbol{u}| \leq 1, j = 1, ..., q\} \tag{6b}$$

It can be mathematically demonstrated that under Assumption 2 in [17], the following optimization problem provides a valid CBF, corresponding positive-invariant set, and a control law that keeps the state inside the invariant set for all time [17].

$$\max_{\substack{X \in \mathcal{S}^N[\tilde{\boldsymbol{x}}],\, Y \in \mathbb{R}^{m \times N}[\boldsymbol{x}],\, X_0 > 0 \\ l_0, l_1, l_2 \in \Sigma[\boldsymbol{x}, \boldsymbol{v}], \\ l_3^i \in \Sigma[\boldsymbol{x}, \boldsymbol{v}, w]\ \forall i \in [p] = \{1, ..., p\}, \\ l_4^j \in \Sigma[\boldsymbol{x}, \boldsymbol{v}, w]\ \forall j \in [q] = \{1, ..., q\}}} \log\det(X_0) \tag{7a}$$

$$\text{s.t.}:$$

$$\boldsymbol{v}^T F_1(\boldsymbol{x})\boldsymbol{v} - l_0 h_0(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}, \boldsymbol{v}], \tag{7b}$$

$$\boldsymbol{v}^T \left(X(\tilde{\boldsymbol{x}}) - X_0\right) \boldsymbol{v} - l_1 h_0(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}, \boldsymbol{v}], \tag{7c}$$

$$\boldsymbol{v}^T \left(P_0 - X(\tilde{\boldsymbol{x}})\right) \boldsymbol{v} - l_2 h_0(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}, \boldsymbol{v}], \tag{7d}$$

$$\begin{bmatrix} \boldsymbol{v} \\ w \end{bmatrix}^T \begin{bmatrix} 1 & C_i X(\tilde{\boldsymbol{x}}) \\ * & X(\tilde{\boldsymbol{x}}) \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ w \end{bmatrix} - l_3^i h_0(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}, \boldsymbol{v}, w], \forall i \in [p], \tag{7e}$$

$$\begin{bmatrix} \boldsymbol{v} \\ w \end{bmatrix}^T \begin{bmatrix} 1 & D_j Y(\boldsymbol{x}) \\ * & X(\tilde{\boldsymbol{x}}) \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ w \end{bmatrix} - l_4^j h_0(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}, \boldsymbol{v}, w], \forall j \in [q]. \tag{7f}$$

where $\tilde{\boldsymbol{x}} = (x_{j_1}, x_{j_2}, ..., x_{j_m})$ in which j_i denote the row indices of $B(\boldsymbol{x})$ whose corresponding row is equal to zero. $\mathcal{S}^N[\tilde{\boldsymbol{x}}]$ and $\mathbb{R}^{m \times N}[\boldsymbol{x}]$ denote the sets of $N \times N$ real symmetric polynomial matrices and of $m \times N$ real matrices whose entries are polynomials of $\tilde{\boldsymbol{x}}$ and $\boldsymbol{x}$, respectively. Furthermore, l_k represents a polynomial, and $\boldsymbol{v} \in \mathbb{R}^N$ as well as $w \in \mathbb{R}$ are for SOS constraint formulation. Finally, $h_0(\boldsymbol{x})$ is a compact algebraic set defined as: $h_0(\boldsymbol{x}) \triangleq 1 - z^T(\boldsymbol{x}) P_0^{-1} z(\boldsymbol{x})$.

The CBF can be synthesized as below:

$$h(\boldsymbol{x}) = 1 - z^T(\boldsymbol{x}) X^{-1}(\tilde{\boldsymbol{x}}) z(\boldsymbol{x}) \tag{8}$$

The corresponding positive invariant set can be represented by $h(\boldsymbol{x}) \geq 0$.

III. Control Methodology

In this section, we aim to formulate a control policy based on MPC and CBF, which mathematically guarantees the safety of a control system despite the limitations of MPC and CBF. We call the system (1) safe if: (i) states of the dynamic system never reach the unsafe region; (ii) control inputs never violate the physical constraints of the system.

Given a model of the system (1), we are able to the establishment of a standard MPC scheme. Assume that this implementation encompasses a finite and relatively short prediction horizon, alongside an imperfect approximation of the terminal cost function. In addition, assume that the available CBF is too conservative rendering it suitable primarily as a backup controller.

Subject to the stipulated assumptions, safety assurance of the system (1), over a substantial subset of the state space through the classical MPC or CBF schemes theoretically is not attainable. In this case, let's insert the terminal constraint $\boldsymbol{x}_{N|k} \in \mathbb{X}_f$ into the MPC standard formula, where $\boldsymbol{x}_{N|k}$ indicates the predicted state at the end of the horizon (terminal state), and $\mathbb{X}_f$ is a forward-invariant set for the system (1) as the terminal set.

Theorem 1: The standard MPC (2) with the terminal constraint ($\boldsymbol{x}_{N|k} \in \mathbb{X}_f$) has a feasible solution forever if it is feasible at the initial time.

Proof: Since the terminal set $\mathbb{X}_f$ is an invariant set, therefore, there should exist a local control law $\mathcal{K}(\boldsymbol{x}_k)$, which guarantees the remaining of all trajectories starting inside $\mathbb{X}_f$ in that set forever, while simultaneously satisfying the state and input constraints (i.e., $\boldsymbol{x}_k \in \mathbb{X}_s$ and $\mathcal{K}(\boldsymbol{x}_k) \in \mathbb{U}$). This technique derived from MPC literature [24] guarantees the recursive feasibility of the MPC. Assume that (2) is feasible at the initial state $\boldsymbol{x}_k$ with the associated optimal sequence of control inputs $\boldsymbol{u}^*_{0:N-1|k} = \{\boldsymbol{u}^*_{0|k}, \boldsymbol{u}^*_{1|k}, ..., \boldsymbol{u}^*_{N-1|k}\}$. At the next time step $(k+1)$, the feasibility of the input sequence $\boldsymbol{u}^*_{0:N-1|k+1} = \{\boldsymbol{u}^*_{1|k}, \boldsymbol{u}^*_{2|k}, ..., \boldsymbol{u}^*_{N-1|k}, \mathcal{K}(\boldsymbol{x}^*_N)\}$ is also attainable based on the assumption that $\mathbb{X}_f$ is a forward-invariant set.
Thus, the MPC has recursive feasibility (i.e., if the trajectory starts at an initial state $\boldsymbol{x}_k$ with a feasible solution $\boldsymbol{u}^*_{0:N-1|k}$, then the MPC problem will have feasible solution forever). ∎

Despite the theoretical feasibility guarantee that the aforementioned technique can provide, it is still a challenging issue to synthesize the terminal forward invariant set $\mathbb{X}_f$. In what follows, we will provide a technique that connects the small synthesized islands of invariant sets using an MPC framework to enable the system to escape from conservative invariant sets.

A. Forward Invariant Set Synthesis

Employing the SOS-based technique introduced in II-C, using the concept of CBF, a control-invariant safe set denoted as C, centered at the origin, will be synthesized as a zero-superlevel set of the function $h(\boldsymbol{x})$. Additionally, the corresponding control policy denoted as $\mathcal{K}(\boldsymbol{x})$ will ensure the remaining of all trajectories inside C when initiated therein.

Statement 1: The aforementioned technique can be extended to any other fixed points of the system 5 by simply a coordinate shift corresponding to the pair of $\{\tilde{\boldsymbol{x}}^j, \tilde{\boldsymbol{u}}^j\}$, where $\tilde{\boldsymbol{x}}^j$ and $\tilde{\boldsymbol{u}}^j$ denote the j^{th} state and input fixed point. In this case, the generated control-invariant safe sets, and their corresponding CBFs and control laws can be denoted as C_j, $h_j(\boldsymbol{x})$ and $\mathcal{K}_j(\boldsymbol{x})$, respectively.

Theorem 2: The union of the generated C_j sets ($C = C_1 \cup C_2 \cup ... \cup C_{N_{cbf}}$) is a control-invariant set.

Proof: Assume that a trajectory starts at the initial state $\boldsymbol{x}(0) \in C_j$. Applying the control policy $\mathcal{K}_j(\boldsymbol{x}(t))$, we can guarantee that $\boldsymbol{x}(t) \in C_j$, $\forall t \geq 0$. Since $C_j \subseteq C$, simply we can conclude that $\boldsymbol{x}(t) \in C$, $\forall t \geq 0$. ∎

B. Model Predictive Control Barrier Functions (MPCBF)

Having the invariant set C, we can formulate our MPCBF control logic by inserting the terminal constraint $\boldsymbol{x}_{N|k} \in C$ into standard MPC constrained, and finite-horizon optimization problem (2).

Remark 1: The utilization of the synthesized invariant set within a continuous system remains applicable in the context of a discrete-time framework, wherein we implement an MPC methodology. To do this, we start with a continuous model to create the invariant sets and then discretize the model for the MPC framework.

Referring to the definition of the CBF in (3c), the fulfillment of $h_j(\boldsymbol{x}_{N|k}) > 0$ will guarantee the $\boldsymbol{x}_{N|k} \in C_j$. Thus, the constraint of the $\boldsymbol{x}_{N|k} \in C$ (or equivalently $\boldsymbol{x}_{N|k} \in C_1 \cup C_2 \cup ... \cup C_{N_{cbf}}$) can be replaced by $h_1(\boldsymbol{x}_{N|k}) > 0 \vee h_2(\boldsymbol{x}_{N|k}) > 0 \vee ... \vee h_{N_{cbf}}(\boldsymbol{x}_{N|k}) > 0$.
To formulate the aforementioned condition in a standard, constrained optimization problem, we define the following lemma.

Lemma 1: The condition $h_1(\boldsymbol{x}_{N|k}) > 0 \vee h_2(\boldsymbol{x}_{N|k}) > 0 \vee ... \vee h_{N_{cbf}}(\boldsymbol{x}_{N|k}) > 0$ will be met if $\lambda_1 h_1(\boldsymbol{x}_{N|k}) + \lambda_2 h_2(\boldsymbol{x}_{N|k}) + ... + \lambda_{N_{cbf}} h_{N_{cbf}}(\boldsymbol{x}_{N|k}) > 0$ for $\lambda_1, \lambda_2, \dots, \lambda_{N_{cbf}}$ with $0 \leq \lambda_j \leq 1$, and $\lambda_1 + \lambda_2 + ... + \lambda_{N_{cbf}} = 1$.

Proof: Let's assume that $h_j(\boldsymbol{x}_{N|k}) < 0$, $\forall\ j \in \{0, \dots, N_{cbf}\}$. Given that $0 \leq \lambda_j \leq 1$, it is impossible for the inequality of $\lambda_1 h_1(\boldsymbol{x}_{N|k}) + \lambda_2 h_2(\boldsymbol{x}_{N|k}) + ... + \lambda_{N_{cbf}} h_{N_{cbf}}(\boldsymbol{x}_{N|k}) > 0$ to hold true. Consequently, there must exist at least one $h_j(\boldsymbol{x}_{N|k})$ that is positive. Equivalently, to hold $h_1(\boldsymbol{x}_{N|k}) > 0 \vee h_2(\boldsymbol{x}_{N|k}) > 0 \vee ... \vee h_{N_{cbf}}(\boldsymbol{x}_{N|k}) > 0$, the condition $\lambda_1 h_1(\boldsymbol{x}_{N|k}) + \lambda_2 h_2(\boldsymbol{x}_{N|k}) + ... + \lambda_{N_{cbf}} h_{N_{cbf}}(\boldsymbol{x}_{N|k}) > 0$ must be satisfied. The extra constraint of $\lambda_1 + \lambda_2 + ... + \lambda_{N_{cbf}} = 1$ will cause to avoid the obvious solution of $\lambda_j \approx 0$. ∎

In accordance with the visual representation presented in Figure 1, the MPCBF formulation is presented in the following manner:

$$\boldsymbol{u}^*_{0:N-1|k} = \underset{\boldsymbol{u}_{0:N-1}, \lambda_j}{\operatorname{argmin}} \sum_{i=0}^{N-1} J(\boldsymbol{x}_{i|k}, \boldsymbol{u}_{i|k}) + J_N(\boldsymbol{x}_{N|k}) \quad (9a)$$

$$\text{s.t.} \quad \forall\ i \in \{0, \dots, N-1\}:$$

$$\boldsymbol{x}_{i+1|k} = f(\boldsymbol{x}_{i|k}) + g(\boldsymbol{x}_{i|k})\boldsymbol{u}_{i|k}, \quad (9b)$$

$$\boldsymbol{x}_{0|k} = \boldsymbol{x}_k, \quad (9c)$$

$$\boldsymbol{x}_{i|k} \in \mathbb{X}_s, \quad (9d)$$

$$\boldsymbol{u}_{i|k} \in \mathbb{U}, \quad (9e)$$

$$\lambda_1 h_1(\boldsymbol{x}_{N|k}) + \lambda_2 h_2(\boldsymbol{x}_{N|k}) + ... + \lambda_{N_{cbf}} h_{N_{cbf}}(\boldsymbol{x}_{N|k}) > 0, \quad (9f)$$

$$0 \leq \lambda_j \leq 1 \quad \forall\ j \in \{1, \dots, N_{cbf}\}, \quad (9g)$$

$$\lambda_1 + \lambda_2 + ... + \lambda_{N_{cbf}} = 1. \quad (9h)$$

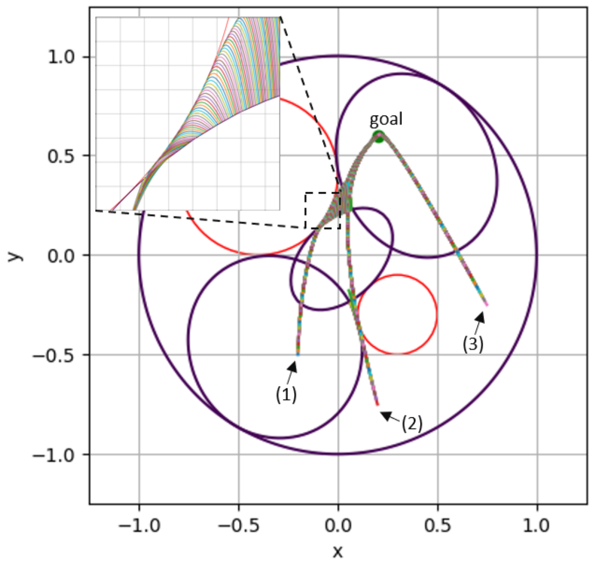

Fig. 2. Predicted trajectories by MPCBF scheme in each time step for three different initial states. The zoom box illustrates that the predicted state at the terminal time step $\boldsymbol{x}_{N|k}$ converges to one of the positively invariant sets, whereas the states at prior time steps may persistently reside outside these sets.

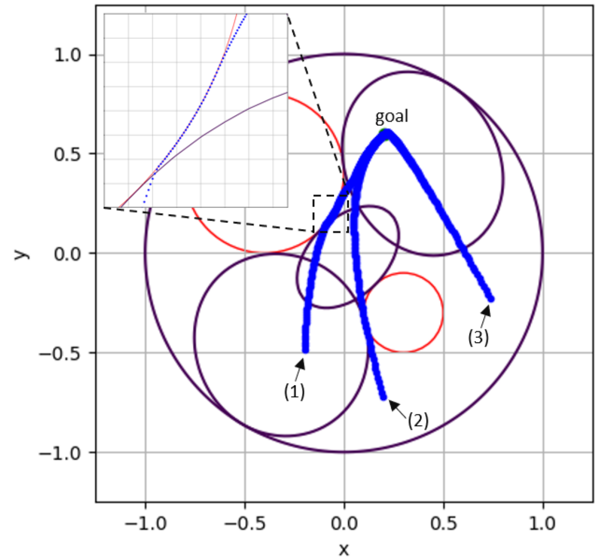

Fig. 3. Real trajectories of the system 11 for three different initial states, running by MPCBF. The zoom box demonstrates that the real trajectory can remain outside any of the invariant sets to reduce conservativeness.

C. MPCBF Safety Filter

The MPCBF also can be employed as the concept of a safety filter. In this case, a performance controller like learning-based policies or user command serves as a stabilizer to create reference control signal $\boldsymbol{u}_{ref}$ and the MPCBF ensures the safety requirements by modifying the $\boldsymbol{u}_{ref}$.

$$\boldsymbol{u}^*_{0:N-1|k} = \underset{\boldsymbol{u}_{0:N-1}}{\text{argmin}} \; ||\boldsymbol{u}_{0|k} - \boldsymbol{u}_{ref}||^2 \tag{10a}$$

$$\text{s.t.} \quad \forall\, i \in \{0, \dots, N-1\}:$$

$$\boldsymbol{x}_{i+1|k} = f(\boldsymbol{x}_{i|k}) + g(\boldsymbol{x}_{i|k})\boldsymbol{u}_{i|k}, \tag{10b}$$

$$\boldsymbol{x}_{0|k} = \boldsymbol{x}_k, \tag{10c}$$

$$\boldsymbol{x}_{i|k} \in \mathbb{X}_s, \tag{10d}$$

$$\boldsymbol{u}_{i|k} \in \mathbb{U}, \tag{10e}$$

$$\lambda_1 h_1(\boldsymbol{x}_{N|k}) + \lambda_2 h_2(\boldsymbol{x}_{N|k}) + ... + \lambda_{N_{cbf}} h_{N_{cbf}}(\boldsymbol{x}_{N|k}) > 0, \tag{10f}$$

$$0 \le \lambda_j \le 1 \quad \forall\, j \in \{1, \dots, N_{cbf}\}, \tag{10g}$$

$$\lambda_1 + \lambda_2 + ... + \lambda_{N_{cbf}} = 1. \tag{10h}$$

IV. Validation and Discussion

In this section, we illustrate the proposed model predictive control barrier functions algorithm using a small-scale linear system as below:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} -1.5 & 0.5 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k \tag{11}$$

As shown in Figure 2 and 3, the system 11 starts from three different initial states $\{(-0.2, -0.5), (0.2, -0.75), (0.75, -0.25)\}$ to reach the goal state $(0.2, 0.6)$. The safe set of states of the system 11, $\mathbb{X}_s$, is defined as states that are inside the unit circle and outside of two red circles with radii of 0.4 and 0.2 located in $(-0.4, 0.4)$ and $(0.3, -0.3)$, respectively. Furthermore, the system is subject to the input constraint of $\boldsymbol{u} \in \mathbb{U} \triangleq \{\boldsymbol{u} \in \mathbb{R} : |\boldsymbol{u}| \le 1\}$.

It can be easily demonstrated that a single MPC with a short horizon, roughly less than $N = 100$ and the time-step $\delta t = 0.01s$, cannot predict the trajectories that reach the goal state. In this case, the traditional MPC is unable to assure safety beyond the horizon.

To implement MPCBF, we synthesized the controlled invariant sets offline on three fixed points of the system $\{(-0.2, -0.6), (0, 0), (0.2, 0.6)\}$ using SOS-based technique described in II-C. For the implementation of our proposed scheme, we used a relatively short number of horizon $N = 10$ with the time-step $\delta t = 0.01s$. The stage cost, in this case study, is defined as the weighted summation of control efforts, changes of control inputs, and errors to goal. Besides, the terminal cost function is approximated as the error between the last predicted state and the goal point. Moreover, we utilized the CasADi [28] framework for automatic differentiation together with the nonlinear optimization solver IPOPT.

As illustrated in Figure 2, the predicted states at the terminal time step $\boldsymbol{x}_{N|k}$ enter one of the positively invariant sets to guarantee the safety beyond the horizon, whereas the states at preceding may reside outside these sets. Figure 3 displays the actual trajectories of the system 11 running by MPCBF. It is demonstrated that the real trajectories are able to stay outside of the invariant sets to reduce conservativeness.

V. Conclusion

This paper has addressed the problems of traditional MPC and CBF. In this study, we have taken steps to mitigate the drawbacks inherent in typical MPC and CBF approaches. This was achieved through the development of a novel framework MPCBF, wherein we combined these methods into a standard constrained optimization problem structure. This formulation offers a straightforward and practical tool for implementation in real-world applications. The introduced methodology mathematically guarantees the state and input safety requirements while a standard MPC with a finite horizon lacks such a benefit. Moreover, providing a long-horizon MPC is not computationally efficient, and obtaining an accurate approximation of the terminal cost is generally a challenging task. Furthermore, CBF restricts the trajectories to remain inside a forward-invariant set in a conservative fashion, while our MPCBF algorithm permits the system to explore beyond this invariant set, yet ensuring the safety criteria.

References

[1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[2] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *2021 American Control Conference (ACC)*, pp. 3882–3889, IEEE, 2021.

[3] 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)*, pp. 3420–3431, IEEE, 2019.

[4] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[5] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *International Workshop on Hybrid Systems: Computation and Control*, pp. 477–492, Springer, 2004.

[6] S. Prajna, "Barrier certificates for nonlinear model validation," *Automatica*, vol. 42, no. 1, pp. 117–126, 2006.

[7] H. Abdi, G. Raja, and R. Ghabcheloo, "Safe control using vision-based control barrier function (v-cbf)," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 782–788, IEEE, 2023.

[8] W. Xiao, T.-H. Wang, M. Chahine, A. Amini, R. Hasani, and D. Rus, "Differentiable control barrier functions for vision-based end-to-end autonomous driving," *arXiv preprint arXiv:2203.02401*, 2022.

[9] Y. Meng, Z. Qin, and C. Fan, "Reactive and safe road user simulations using neural barrier certificates," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6299–6306, IEEE, 2021.

[10] N. E. Toulkani, H. Abdi, O. Koskelainen, and R. Ghabcheloo, "Reactive safe path following for differential drive mobile robots using control barrier functions," in *2022 10th International Conference on Control, Mechatronics and Automation (ICCMA)*, pp. 60–65, IEEE, 2022.

[11] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE control systems letters*, vol. 1, no. 2, pp. 310–315, 2017.

[12] Y. Chen, A. Singletary, and A. D. Ames, "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2020.

[13] B. Xu and K. Sreenath, "Safe teleoperation of dynamic uavs through control barrier functions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7848–7855, IEEE, 2018.

[14] L. Wang, E. A. Theodorou, and M. Egerstedt, "Safe learning of quadrotor dynamics using barrier certificates," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2460–2465, IEEE, 2018.

[15] 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)*, pp. 827–834, IEEE, 2016.

[16] 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)*, pp. 8352–8358, IEEE, 2021.

[17] P. Zhao, R. Ghabcheloo, Y. Cheng, H. Abdi, and N. Hovakimyan, "Convex synthesis of control barrier functions under input constraints," *IEEE Control Systems Letters*, 2023.

[18] H. Dai and F. Permenter, "Convex synthesis and verification of control-lyapunov and barrier functions with input constraints," in *2023 American Control Conference (ACC)*, pp. 4116–4123, IEEE, 2023.

[19] Z. Marvi and B. Kiumarsi, "Safety planning using control barrier function: A model predictive control scheme," in *2019 IEEE 2nd Connected and Automated Vehicles Symposium (CAVS)*, pp. 1–5, IEEE, 2019.

[20] 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)*, pp. 7623–7628, IEEE, 2019.

[21] Z. Wu, F. Albalawi, Z. Zhang, J. Zhang, H. Durand, and P. D. Christofides, "Control lyapunov-barrier function-based model predictive control of nonlinear systems," *Automatica*, vol. 109, p. 108508, 2019.

[22] 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)*, pp. 551–557, IEEE, 2021.

[23] U. Rosolia and A. D. Ames, "Multi-rate control design leveraging control barrier functions and model predictive control policies," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 1007–1012, 2020.

[24] K. P. Wabersich and M. N. Zeilinger, "Predictive control barrier functions: Enhanced safety mechanisms for learning-based control," *IEEE Transactions on Automatic Control*, 2022.

[25] K. P. Wabersich and M. N. Zeilinger, "Linear model predictive safety certification for learning-based control," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 7130–7135, IEEE, 2018.

[26] B. Tearle, K. P. Wabersich, A. Carron, and M. N. Zeilinger, "A predictive safety filter for learning-based racing control," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7635–7642, 2021.

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

[28] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.