Convex Synthesis of Control Barrier Functions Under Input Constraints

Pan Zhao, *Member, IEEE*, Reza Ghabcheloo, Yikun Cheng, *Graduate Student Member, IEEE*, Hossein Abdi, and Naira Hovakimyan, *Fellow, IEEE*

***Abstract*—This letter presents a systematic method based on the sum of square (SOS) optimization to synthesize control barrier functions (CBFs) for nonlinear polynomial systems subject to input constraints. The approach consists of two design steps. In the first step, using a linear-like representation of the nonlinear dynamics, an SOS optimization problem is formulated to search for an initial CBF and controller jointly. In the second step, an iterative optimization procedure involving the solution of a series of SOS problems is proposed to alternatively update the CBF and the controller to increase the invariant set defined by the CBF. The efficacy of the proposed approach is validated using numerical examples.**

***Index Terms*—Constrained control, optimization, nonlinear systems, safety.**

I. Introduction

Control barrier functions (CBFs) have emerged as a practical tool for controlling nonlinear systems with safety constraints [1], [2]. CBFs can be used to synthesize control signals to enforce state constraints by ensuring set invariance, while not resorting to a specific control law [1]. On the other hand, input constraints exist in almost any real-world system. Consideration of input constraints in control design is therefore needed for practical implementation.

In most of the existing work on CBFs [1], [2], [3], [4], [5], a function is heuristically or intuitively selected and *assumed* to be a CBF. However, this heuristic or intuitive design procedure works only for naive scenarios, e.g., simple box constraints on position or velocity [1], [2], [3], [4], [5]. Given an intuitively designed candidate function, one still needs to verify that this function is indeed a CBF, which is particularly important in the presence of input constraints. Otherwise, the feasibility of the QP problem formulated to construct the control signal and safety of the system cannot be guaranteed.

Contributions: This letter presents a systematic method based on convex optimization (in particular, sum of squares (SOS) optimization [6]) for the synthesis of CBFs in the presence of input constraints for nonlinear polynomial systems. Our method consists of two design steps, each yielding a valid CBF. The initial design is to jointly search an initial CBF and a controller, using a state-dependent linear-like form of the nonlinear dynamics and SOS techniques, in which the volume of the CBF-defined invariant set is maximized. The second step is to alternatively update the CBF obtained from the first step and the controller while fixing the other using the original nonlinear dynamics. Numerical examples are included to validate the effectiveness of the proposed method.

Related Work: A closely related problem is Lyapunov-based nonlinear control synthesis using SOS optimization. Leveraging a linear-like representation, [7] proposed to search for a state-dependent Lyapunov function (LF) and a controller jointly using SOS programming. In [7], nonlinear terms involving the product of LF coefficients and controller gain are avoided by constraining the LF to depend on states whose derivatives are not directly affected by any control input. The result in [7] motivates our proposed approach to CBF synthesis (the first step). Finally, [8] proposed to iteratively search for control Lyapunov functions (CLFs) via SOS optimization without searching for a controller.

Convex optimization-based nonlinear control synthesis under input constraints has also been studied. The author of [9] introduced a polytope model to characterize the saturation and formulated the design problem as an SOS problem by extending the linear matrix inequality (LMI) approach for linear systems with input saturation. In [10], a system with saturating input was converted to a new system with unsaturated input and a vanishing disturbance input, to which a standard design method inspired by [7] can be applied. Also inspired by [7], the authors of [11] formulated an SOS optimization problem to compute the optimal feedback gain for predictive control of a nonlinear system subject to input constraints. This letter motivates the first step of our SOS-based method for CBF synthesis.

CBF synthesis has been explored recently based on SOS optimization [12], [13], [14], linearization [15], machine learning [16], [17] and Hamilton-Jacobi reachability analysis [18], [19]. In [12], the authors proposed to sequentially

Manuscript received 17 March 2023; revised 24 May 2023; accepted 16 June 2023. Date of publication 7 July 2023; date of current version 28 July 2023. This work was supported in part by the Air Force Office of Scientific Research (AFOSR) under Grant FA9550-21-1-0411; in part by the National Aeronautics and Space Administration (NASA) through ULI under Grant 80NSSC22M0070; and in part by NSF through RI under Grant 2133656. Recommended by Senior Editor S. Tarbouriech. *(Corresponding author: Pan Zhao.)*

Pan Zhao, Yikun Cheng, and Naira Hovakimyan are with the Department of Mechanical Science and Engineering, University of Illinois at Urbana–Champaign, Urbana, IL 61801 USA (e-mail: panzhao2@illinois.edu; yikun2@illinois.edu; nhovakim@illinois.edu).

Reza Ghabcheloo and Hossein Abdi are with the Faculty of Engineering and Natural Sciences, Tampere University, 33720 Tampere, Finland (e-mail: reza.ghabcheloo; hossein.abdi@tuni.fi).

Digital Object Identifier 10.1109/LCSYS.2023.3293765

update a CBF and a controller while fixing the other using SOS optimization. However, the design procedure needs an LF to calculate an initial CBF, while how to obtain such an LF was not given. In [13], the author presented an SOS optimization-based method for CBF synthesis, which, unfortunately, is computationally heavy and does not guarantee the return of a valid CBF. Very few works considered CBF synthesis under input constraints. In [15], the authors proposed to synthesize CBFs for discrete-time nonlinear systems with input constraints, by first searching for a quadratic CBF for a linearized system and then refining it for the nonlinear systems. However, the limit to quadratic functions makes the design overly conservative. Additionally, the design procedure requires a global solution of nonlinear programming problems, which, by itself, is a challenging problem. Very recently, [14] proposed an SOS optimization-based method for CLF/CBF synthesis that both accounts for input constraints and explicitly avoids the construction of a nominal controller, potentially improving performance and reducing the computational cost. However, an initial valid LF is needed for the synthesis procedure.

Notations: Let $\mathbb{R}$, $\mathbb{R}^n$, $\mathbb{R}^{m\times n}$ and $\mathcal{S}^n$ denote the sets of real numbers, n-dimensional real vectors, m by n real matrices, and $n \times n$ real symmetric matrices, respectively. $\mathbb{Z}_1^n$ denotes the integer set $\{1, 2, \dots, n\}$. $\mathbb{R}[x]$ denotes the set of polynomials with real coefficients, while $\mathcal{S}^n[x]$ and $\mathbb{R}^{m\times n}[x]$ denote the sets of $n\times n$ real symmetric polynomial matrices and of $m \times n$ real matrices, respectively, whose entries are polynomials of x with real coefficients. Finally, we let $\Sigma[x]$ denote the set of SOS polynomials of x.

II. Preliminaries and Problem Setting

Consider a nonlinear control-affine system

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

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $f : \mathbb{R}^n \to \mathbb{R}^n$ and $B : \mathbb{R}^n \to \mathbb{R}^m$ are polynomial and locally Lipschitz continuous functions. The system (1) needs to satisfy state constraints given by

$$x(t) \in \mathcal{X} \triangleq \{x \in \mathbb{R}^n : c_i(x) \le 0,\ \ i = 1, \dots, p\}, \tag{2}$$

for all $t \ge 0$, where $c_i : \mathbb{R}^n \to \mathbb{R}$ is a polynomial function, and $\mathcal{X}$ is a compact set. Additionally, (1) is subject to input constraints of the form

$$u(t) \in \mathcal{U} \triangleq \{u \in \mathbb{R}^m : D_j u \le 1,\ \ i = 1, \dots, q\}, \tag{3}$$

for all $t \ge 0$, where $D_j \in \mathbb{R}^{1\times m}$. Without loss of generality, we assume the interior of $\mathcal{X}$ contains the origin. If this assumption does not hold, one can possibly shift the system so that $\mathcal{X}$ defined using the shifted states contains the origin.

Definition 1 (CBF [1]): A continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ is a CBF for the system (1), if there exists an extended class $\mathcal{K}$ function $\alpha(\cdot)$ such that for all $x \in \mathcal{X}$, there exists a $u \in \mathbb{R}^m$ satisfying

$$\dot{h}(x) = L_f h(x) + L_B h(x)u \ge -\alpha(h(x)), \tag{4}$$

where $L_f h(x) \triangleq \frac{\partial h(x)}{\partial x} f(x)$, $L_B h(x) \triangleq \frac{\partial h(x)}{\partial x} B(x)$.

As noted in [1], $h(x)$ being a CBF ensures the set

$$\mathcal{X}_h \triangleq \{x \in \mathbb{R}^n : h(x) \ge 0\} \tag{5}$$

is forward invariant: if $x(0) \in \mathcal{X}_h$, then there exists a control law $u(t) \in \mathcal{U}$ such that for all $t \ge 0$, $x(t) \in \mathcal{X}_h$. Definition 1 allows us to consider all control signals for each $x \in \mathcal{X}$ and $t \ge 0$ that render $\mathcal{X}_h$ forward invariant:

$$K_{\mathrm{cbf}}(x) \triangleq \{u \in \mathcal{U} : L_f h(x) + L_B h(x)u \ge -\alpha(h(x))\}. \tag{6}$$

A polynomial $l(x)$ is an SOS if there exist polynomials $l_1(x), \dots, l_m(x)$ such that $l(x) = \sum_{i=1}^m l_i^2(x)$. A polynomial $l(x)$ of degree $2d$ is an SOS iff there exists a positive semidefinite (PSD) matrix Q such that $l(x) = y^T(x)Qy(x)$, where $y(x)$ is a column vector whose entries are all monomials in x with degree up to d [6]. An SOS decomposition for a given $l(x)$ can be computed using semidefinite programming (SDP) (by searching for a PSD matrix Q). When a polynomial $l(x)$ is not exactly determined, but its coefficients are affinely parameterized in terms of some unknowns, the search for these unknowns which render $l(x)$ an SOS can still be performed via SDP [7].

Proposition 1 [7, Proposition 2]: Let $F(x)$ be an $N \times N$ symmetric polynomial matrix of degree $2d$ in $x \in \mathbb{R}^n$. Then, $F(x) \ge 0\ \forall x \in \mathbb{R}^n$, if $v^T F(x) v \in \Sigma[x, v]$, where $v \in \mathbb{R}^N$.

Proposition 2 [7, Proposition 10]: Let $F(x)$ be an $N \times N$ symmetric polynomial matrix of degree $2d$ in $x \in \mathbb{R}^n$, and let $\mathcal{X}$ be a set defined as $\mathcal{X} = \{x \in \mathbb{R}^n : g_l(x) \ge 0,\ \ l = 1, \dots, s\}$, $v \in \mathbb{R}^N$. Suppose there exist SOS polynomials $l_l(x, v)$, $l = 1, \dots, s$ whose degree in v is equal to two, such that $v^T F(x)v - \sum_{l=1}^s l_l(x, v)g_l(x) \in \Sigma[x, v]$. Then, $F(x) \ge 0$ for all $x \in \mathcal{X}$.

III. SOS Optimization-Based Synthesis of CBFs Under Input Constraints

In this section, we first present an approach to synthesizing CBFs under input constraints using state-dependent linear-like forms and SOS optimization. We then introduce an iterative algorithm involving SOS optimization to further improve the CBFs from the initial design.

A. Initial CBF Synthesis Using Linear-Like Forms

Inspired by [7], [11], we make the following assumptions.

Assumption 1: The nonlinear system (1) has a a state-dependent linear-like representation [7]:

$$\dot{x} = A(x)z(x) + B(x)u, \tag{7}$$

where $A(x) \in \mathbb{R}^{n\times N}$ is a polynomial matrix, and $z(x) \in \mathbb{R}^N$ is a vector of monomials such that $z(x) = 0$ iff $x = 0$.

Assumption 2: There exists a compact algebraic set $\bar{\mathcal{X}} = \{x \in \mathbb{R}^n | h_0(x) \ge 0\}$, where

$$h_0(x) \triangleq 1 - z^T(x)P_0^{-1}z(x) \tag{8}$$

for some positive definite matrix P_0, such that $\mathcal{X} \subseteq \bar{\mathcal{X}}$.

Remark 1: Assumption 1 is not very restrictive as it can always be satisfied when $f(x)$ is a polynomial and does not contain any constant terms, i.e., monomials of zero degrees. If the original state constraint set ($\mathcal{X}$) is compact, we can always find a set $\bar{\mathcal{X}}$ (which will be an ellipsoid when $z(x) = x$) that is sufficiently large such that Assumption 2 holds.

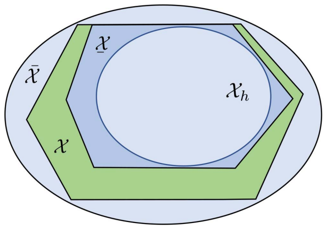

Fig. 1. Illustration of algebraic sets defined in Section III.

With $z(x)$ in (7), we can introduce an inner approximation of $\mathcal{X}$ as

$$\underline{\mathcal{X}} \triangleq \{x \in \mathbb{R}^n : |C_i(x)z(x)| \leq 1,\ i \in \mathbb{Z}_1^{\tilde{p}}\}, \tag{9}$$

where $C_i : \mathbb{R}^n \to \mathbb{R}^{1\times N}$ is a vector-valued polynomial function. An illustration of different sets is given in Fig. 1. Similarly, we introduce an inner approximation of $\mathcal{U}$ as $\underline{\mathcal{U}} \triangleq \{u \in \mathbb{R}^m : |\tilde{D}_j u| \leq 1,\ j \in \mathbb{Z}_1^{\tilde{q}}\}$, where $\tilde{D}_j \in \mathbb{R}^{1\times m}$.

Remark 2: Definition (9) captures a wide range of commonly seen geometric sets. For instance, the constraint $|x_i| \leq a_i$ $(a_i > 0)$ for defining a hypercube can be represented as $|\underbrace{1/a_i \mathbf{1}_i}_{C_i(x)} \underbrace{x}_{z(x)}| \leq 1$ with $\mathbf{1}_i$ being a vector of appropriate dimension with all zero elements except the ith one equal to 1. An ellipsoid constraint $x^T P x \leq 1$ can be represented as $\left|C_i(x)z(x)\right|$ with $C_i(x) = x^T P$ and $z(x) = x$.

In addition, we define $M(x)$ to be an $N \times n$ polynomial matrix whose (i, j)-th entry is given by $M_{ij} = \frac{\partial z_i}{\partial x_j}(x)$. Let $A_j(x)$ denote the j-th row of $A(x)$, $J = \{j_1, j_2, \ldots, j_m\}$ denote the row indices of $B(x)$ whose corresponding row is equal to zero, and define

$$\tilde{x} = (x_{j_1}, x_{j_2}, \ldots, x_{j_m}),$$

which includes all the states whose derivatives are not directly affected by control inputs.

The result is summarized in the following theorem.

Theorem 1: Consider the system (7) subject to the input constraint (3) and state constraint (2). Suppose Assumption 2 hold, and the following optimization problem is feasible:

$$\max_{\substack{X \in \mathcal{S}^N[\tilde{x}],\ Y \in \mathbb{R}^{m\times N}[x],\ l_0, l_1, l_2 \in \Sigma[x, v],\\ l_3^i \in \Sigma[x, v, w]\ \forall i \in \mathbb{Z}_1^{\tilde{p}},\ l_4^j \in \Sigma[x, v, w]\ \forall j \in \mathbb{Z}_1^{\tilde{q}},\ X_0 > 0}} \log\det(X_0) \tag{10}$$

s.t.

$$v^T F_1(x) v - l_0 h_0(x) \in \Sigma[x, v], \tag{11}$$

$$v^T\big(X(\tilde{x}) - X_0\big)v - l_1 h_0(x) \in \Sigma[x, v], \tag{12}$$

$$v^T\big(P_0 - X(\tilde{x})\big)v - l_2 h_0(x) \in \Sigma[x, v], \tag{13}$$

$$\begin{bmatrix} v \\ w \end{bmatrix}^T \begin{bmatrix} 1 & C_i X(\tilde{x}) \\ * & X(\tilde{x}) \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} - l_3^i h_0(x) \in \Sigma[x, v, w],\ \forall i \in \mathbb{Z}_1^{\tilde{p}}, \tag{14}$$

$$\begin{bmatrix} v \\ w \end{bmatrix}^T \begin{bmatrix} 1 & \tilde{D}_j Y(x) \\ * & X(\tilde{x}) \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} - l_4^j h_0(x) \in \Sigma[x, v, w],\ \forall j \in \mathbb{Z}_1^{\tilde{q}}, \tag{15}$$

where $v \in \mathbb{R}^N$ and $w \in \mathbb{R}$ are introduced to convert a matrix-valued polynomial to a scalar polynomial so that an SOS constraint can be formulated,

$$F_1(x) = \sum_{j\in J} \frac{\partial X(\tilde{x})}{\partial x_j}\big(A_j z\big) - \Big\langle M\big(AX(\tilde{x}) + BY\big)\Big\rangle, \tag{16}$$

and we have omitted the dependence of most variables on x for brevity. Then, the following statements hold:

(a) The function

$$h(x) = 1 - z^T(x) X^{-1}(\tilde{x}) z(x) \tag{17}$$

is a CBF for (7) according to Definition 1. Moreover, (4) is satisfied for all $x \in \mathcal{X}$ with a control signal from

$$u(x) = Y(x) X^{-1}(\tilde{x}) z(x) \tag{18}$$

satisfying (4) for all $x \in \mathcal{X}$. Moreover, $\mathcal{X}_h \subseteq \mathcal{X}$, where $\mathcal{X}$ and $\mathcal{X}_h$ are defined in (2), and (5), respectively.

(b) If $x(0) \in \mathcal{X}_h$, under the control law (18), we have $u(t) \in \mathcal{U}$ and $x(t) \in \mathcal{X}_h$ for all $t \geq 0$.

Before proceeding to the proof, we first explain the purpose of constraints imposed in (10). Condition (11) ensures $\dot{h}(x) \geq 0$ for any $x \in \bar{\mathcal{X}}$ with the control law (18), while (12), and (13) ensure $X(\tilde{x}) \geq X_0$ and $P_0 \geq X(\tilde{x})$, respectively, for any $x \in \bar{\mathcal{X}}$. Condition (14) ensures $\left|C_i(x)z(x)\right| \leq 1$, i.e., the ith condition in defining $\bar{\mathcal{X}}$ in (9) is satisfied, for any $x \in \bar{\mathcal{X}}$, while (15) ensures $\left|\tilde{D}_j u\right| \leq 1$, i.e., the ith input constraint condition is satisfied, for any $x \in \bar{\mathcal{X}}$ with the control law (18).

Remark 3: Similar to [7], the matrix X depends only on $\tilde{x}$. If X depends on all the states, when computing $\frac{dX^{-1}(x)}{dt}$ to calculate $\dot{h}$ as in (22), we have $\frac{dX^{-1}(x)}{dt} = \sum_{j=1}^n \frac{\partial X^{-1}(x)}{\partial x_j}(A_j z + B_j u(x))$ instead of $\frac{dX^{-1}(\tilde{x})}{dt} = \sum_{j\in J} \frac{\partial X^{-1}(\tilde{x})}{\partial x_j}(A_j z)$ currently used in (22). As a result, we could not obtain a convex SOS condition similar to (11) anymore.

Remark 4: Recall that the volume of a generalized ellipsoid $\mathcal{E} = \{v \mid \|Qv\| \leq 1\}$ is proportional to $\det(Q^{-1})$ [20, Sec. 8.4]. We can think of the set $\mathcal{X}_h$ with $h(x)$ defined in (17) as an ellipsoid (although it is not unless $X(\tilde{x})$ is constant and $z(x) = x$). Therefore, maximizing $\log\det X_0$ in (10) is a proxy mechanism for maximizing the volume of $\mathcal{X}_h$.

Proof [Proof of (a)]: According to Proposition 2, (12) implies $X(\tilde{x}) \geq X_0 > 0$, for all $x \in \bar{\mathcal{X}}$. Also, (13) implies $P_0 \geq X(\tilde{x})$ for all $x \in \bar{\mathcal{X}}$, which further implies

$$1 - z^T(x) P_0^{-1} z(x) \geq 1 - z^T(x) X^{-1}(\tilde{x}) z(x),\ \ \forall x \in \bar{\mathcal{X}}. \tag{19}$$

Therefore, we have $\mathcal{X}_h \subseteq \bar{\mathcal{X}}$ with the definitions of $\bar{\mathcal{X}}$ and $\mathcal{X}_h$ in (8) and (5), respectively. On the other hand, for any $x \in \mathcal{X}_h$ with $h(x)$ given by (17), we have

$$z^T(x) X^{-1}(\tilde{x}) z(x) \leq 1,\ \ \forall x \in \mathcal{X}_h. \tag{20}$$

According to Proposition 2, (11) implies

$$F_1(x) \geq 0,\ \ \forall x \in \bar{\mathcal{X}}. \tag{21}$$

Under the dynamics (7) and the control law (18), the definition in (17) implies

$$\dot{h}(x) = -z^T(x)\left[\sum_{j\in J} \frac{\partial X^{-1}(\tilde{x})}{\partial x_j}\big(A_j z\big) + \Big\langle X^{-1}(\tilde{x}) M\Big(A + BYX^{-1}(\tilde{x})\Big)\Big\rangle\right] z(x) \tag{22}$$

$$= z^T(x)X^{-1}(\tilde{x})F_1(x)X^{-1}(\tilde{x})z(x), \quad (23)$$
$$\geq 0, \quad (24)$$

for all $x \in \bar{\mathcal{X}}$, where we have leveraged the fact that $\dot{z}(x) = M(A + BYX^{-1}(\tilde{x}))z(x)$ and $\frac{\partial X^{-1}(\tilde{x})}{\partial x_j} = -X^{-1}(\tilde{x})\frac{\partial X(\tilde{x})}{\partial x_j}X^{-1}(\tilde{x})$ [7, Lemma 5] in deriving (23), and (24) is due to (21). As a result, condition (4) can always be satisfied for any $x \in \bar{\mathcal{X}}$ under the control law $u(x)$ defined by (18). Therefore, $h(x)$ is a valid CBF.

According to Proposition 2, (14) implies $\begin{bmatrix} 1 & C_iX(\tilde{x}) \\ * & X(\tilde{x}) \end{bmatrix} \geq 0$ for all $x \in \bar{\mathcal{X}}$, which, via Schur complement, further indicates $X(\tilde{x})C_i^TC_iX(\tilde{x}) \leq X(\tilde{x})$ for all $x \in \bar{\mathcal{X}}$. Multiplying the preceding inequality by $z^T(x)X^{-1}(\tilde{x})$ and its transpose from the left and right, respectively, leads to $|C_iz(x)|^2 \leq z^T(x)X^{-1}(\tilde{x})z(x)$, $\forall x \in \bar{\mathcal{X}}$, $\forall i \in \mathbb{Z}_1^{\tilde{p}}$, which, together with (20) and the fact that $\mathcal{X}_h \subseteq \bar{\mathcal{X}}$, indicates $|C_iz(x)| \leq 1$ for all $x \in \mathcal{X}_h$ and for all $i \in \mathbb{Z}_1^{\tilde{p}}$. As a result, $\mathcal{X}_h \subseteq \underline{\mathcal{X}} \subseteq \mathcal{X}$.

Proof of (b): According to Proposition 2, (15) implies $\begin{bmatrix} 1 & \tilde{D}_jY(x) \\ * & X(\tilde{x}) \end{bmatrix} \geq 0$ for all $x \in \bar{\mathcal{X}}$ for all $j \in \mathbb{Z}_1^{\tilde{q}}$, which, via Schur complement, further indicates $X^{-1}(\tilde{x})Y^T\tilde{D}_j^T\tilde{D}_jYX^{-1}(\tilde{x}) \leq X^{-1}(\tilde{x})$ for all $x \in \bar{\mathcal{X}}$ for all $j \in \mathbb{Z}_1^{\tilde{q}}$. Multiplying the preceding inequality by $z(x)$ and its transpose from the right and left, respectively, and considering the control law (18), we have $|\tilde{D}_ju|^2 \leq z^T(x)X^{-1}(\tilde{x})z(x)$, which, together with (20), implies $|\tilde{D}_ju| \leq 1$ for all $x \in \mathcal{X}_h$ for all $j \in \mathbb{Z}_1^{\tilde{q}}$. As a result, the control law (18) satisfies $u(x) \in \mathcal{U}$, for all $x \in \mathcal{X}_h$. If $x(0) \in \mathcal{X}_h$, by contradiction, it is easy to show that $u(x(t)) \in \mathcal{U}$, $\dot{h}(x(t)) \geq 0$, and $x(t) \in \mathcal{X}_h$ for all $t \geq 0$. The proof is complete. ■

Problem (10) is a convex optimization problem, as the cost function to be maximized is concave, and all the constraints are SOS constraints with affine dependence on decision variables.

Remark 5: To reduce the conservatism of problem (10), a number of workarounds may be considered. First, one can replace $h_0(x)$ in any of (11)–(15) with $-c_i(x)$ ($i \in \mathbb{Z}_1^p$) to ensure that corresponding conditions hold in $\mathcal{X}$ instead of $\bar{\mathcal{X}}$ to reduce the conservatism. This procedure will lead to increased computational cost since a multiplier needs to be introduced for each $-c_i(x)$ with $i \in \mathbb{Z}_1^p$. As an example, one can replace $l_0h_0(x)$ in (11) with $\sum_{i\in\mathbb{Z}_1^p} l_0^i \cdot (-c_i(x))$ where $l_0^i \in \Sigma[x, v]$. Second, one could try to increase the complexity of polynomial decision variables such as $X(\tilde{x})$ and $Y(x)$ by increasing the degrees and/or states that the variables depend on.

B. Refining CBFs via Iterative SOS Optimization

The use of the (potentially non-unique) linear-like form (7) in Section III-A, the constraint $\dot{h}(x) \geq 0$ through (11) (which is more restrictive than needed by Definition 1), and/or the inner-approximation of $\mathcal{X}$ and $\mathcal{U}$ adopted in Theorem 1 can lead to conservative results. To reduce the conservatism, motivated by [12], we now present an iterative procedure, summarized in Algorithm 1, to alternatively update $h(x)$ and $u(x)$, as well as associated multipliers.

Remark 6: Since $Y(\tilde{x})$ is a polynomial matrix and $z(x)$ is a polynomial vector, $h(x)$ and $u(x)$ from solving problem (10) are guaranteed to have a polynomial form if we restrain $X(\tilde{x})$ to be a constant matrix.

Algorithm 1 Iterative SOS Optimization for Refining a CBF

Input: Initial CBF $h(x)$ and controller $u(x)$ with a polynomial form from solving problem (10), an extended class $\mathcal{K}$ function $\alpha(\cdot)$, MaxIter and Tol

Step 0: With $h(x)$ fixed, solve

$$\max_{\substack{l_h \in \Sigma[x],\, u \in \mathbb{R}^m[x],\, \varepsilon \geq 0 \\ l_x^{i0} \in \Sigma[x]\ \forall i \in \mathbb{Z}_1^p,\, l_u^j \in \Sigma[x]\ \forall j \in \mathbb{Z}_1^q}} \varepsilon \quad (25)$$
$$\text{s.t.} \quad L_fh(x) + L_Bh(x)u(x) + \alpha(h(x)) - l_hh(x) - \varepsilon \in \Sigma[x] \quad (26)$$
$$-h(x) - l_x^{i0}c_i(x) \in \Sigma[x], \ \ \forall i \in \mathbb{Z}_1^p, \quad (27)$$
$$1 - D_ju(x) - l_u^jh(x) \in \Sigma[x], \ \ \forall j \in \mathbb{Z}_1^q, \quad (28)$$

where the dependence of h on x is omitted. If (25) is feasible, set $k = 1$, select a N-dimensional column vector, $y(x)$, whose entries are all monomials in x such that $h(x) = y^T(x)Q_0y(x)$ for some symmetric matrix Q_0, and go to **Iterative Update**; otherwise, go to **Output**.

Iterative Update:

• **Step 1** ($h(x)$ update): Fixing $u(x)$, $l_h(x)$ and $l_u^j(x)$ ($j \in \mathbb{Z}_1^q$), solve

$$\max_{Q \in \mathcal{S}^N,\, \mu_0 \in \mathbb{R},\, l_x^i \in \Sigma[x]\ \forall j \in \mathbb{Z}_1^q} \mu_0 \quad (29)$$
$$\text{s.t.} \quad L_fh(x) + L_Bh(x)u(x) + \alpha(h(x)) - l_hh(x) \in \Sigma[x], \quad (30)$$
$$-h(x) - l_x^ic_i(x) \in \Sigma[x], \ \ \forall i \in \mathbb{Z}_1^p, \quad (31)$$
$$1 - D_ju(x) - l_u^j(x)\ h(x) \in \Sigma[x], \ \ \forall j \in \mathbb{Z}_1^q, \quad (32)$$
$$Q \geq \mu_0 I_N, \ \ Q(1,1) = 1, \quad (33)$$

where $h(x) = y^T(x)Qy(x)$. Set $\mu(k) = \mu_0$.

• **Evaluation**: if $k == 1$ or ($k <$ MaxIter and $\mu(k) - \mu(k-1) >$ Tol), set $k \leftarrow k + 1$ and go to **Step 2**; otherwise, go to **Output**.

• **Step 2** ($u(x)$ update): Fixing $h(x)$, solve

$$\max_{l_h \in \Sigma[x],\, l_u^j \in \Sigma[x]\ \forall j \in \mathbb{Z}_1^q,\, u \in \mathbb{R}^m[x],\, \varepsilon \geq 0} \varepsilon \quad (34)$$
$$\text{s.t. (26) and (28).} \quad (35)$$

Go to **Step 1**.

Output: a CBF $h(x)$ and a controller $u(x)$.

The following theorem gives an analysis of Algorithm 1.

Theorem 2: Consider Algorithm 1. Suppose the optimization problem (25) in **Step 0** is feasible with an appropriate selection of the monomials for $u(x)$, $l_h(x)$, $l_x^i(x)$ ($i \in \mathbb{Z}_1^p$), and $l_u^j(x, v)$ ($j \in \mathbb{Z}_1^q$). Then, the following statements hold:

(a) Problems (34), and (29) are always feasible for $k \geq 1$.

(b) The value of the objective function from solving (29) is monotonically increasing, i.e., $\mu(k) \geq \mu(k-1)$ for all $k \geq 2$.

(c) The function $h(x)$ output by Algorithm 1 is a valid CBF according to Definition 1. Besides, $\mathcal{X}_h \subseteq \mathcal{X}$ with $\mathcal{X}_h$ defined in (5). Moreover, if $x(0) \in \mathcal{X}_h$, under the controller $u(x)$, we have $u(t) \in \mathcal{U}$ and $x(t) \in \mathcal{X}$ for all $t \geq 0$.

Proof [Proof of (a)]: Let us first consider the optimization problem (29) in **Step 1** for $k = 1$. Select the monomials for l_x^i so that l_x^{i0} from **Step 0** can be fully represented by those monomials. Then, by comparing the constraints of the optimization problems (25) and (29) and considering the fact that $l(x) \in \Sigma[x]$ if $l(x) - \beta \in \Sigma[x]$ for any $\beta \geq 0$, one can see that the constraints of (29) can be satisfied with $l_x^i(x) = l_x^{i0}(x)$, $Q = Q_0$, and μ_0 set to be the smallest eigenvalue of Q. Similarly, the optimized variables $u(x)$, $l_h(x)$ and $l_u^j(x)$ ($j \in \mathbb{Z}_1^q$) from **Step 0** and $h(x)$ from **Step 1**, together with $\varepsilon = 0$, satisfy all the constraints of (34).

Next, consider the case of $k \geq 2$. For **Step 1**, by comparing the constraints of problems (34), and (29), one can see that all the constraints of (29) can be satisfied with μ_0, Q and $l_x^i \in \Sigma[x]$ ($i \in \mathbb{Z}_1^p$) from **Step 1** at iteration $k-1$ and $u(x)$, $l_h(x)$ and $l_u^j(x)$ ($j \in \mathbb{Z}_1^q$) from **Step 2** at iteration $k-1$. For **Step 2**, the constraints of (34) can be satisfied with $u(x)$, $l_h(x)$ and $l_u^j(x)$ ($j \in \mathbb{Z}_1^q$) from **Step 2** at iteration $k-1$ and $h(x)$ from **Step 1** at iteration k, together with $\varepsilon = 0$.

Proof of (b): As shown in the proof of (a), for $k \geq 2$, $\mu(k-1)$ (obtained at iteration $k-1$) is a candidate for μ_0 to ensure the feasibility of (29) at iteration k. Therefore, $\mu(k) \geq \mu(k-1)$ for $k \geq 2$.

Proof of (c): If (25) in **Step 0** is infeasible, the $h(x)$ output by Algorithm 1 is exactly the one obtained from solving (10). Under such a case, (c) is the same as statement (b) of Theorem 1 and thus holds according to Theorem 1. Otherwise, the functions $h(x)$ and $u(x)$, produced as outputs by Algorithm 1, should satisfy the constraints (30)–(32) in problem (29). Notice that (30) implies $\dot{h}(x) = L_f h(x) + L_B h(x) u(x) \geq -\alpha(h(x))$ for all $x \in \bar{\mathcal{X}}$. Therefore, $h(x)$ is a valid CBF according to Definition 1. Moreover, (31) implies that for any $i \in \mathbb{Z}_1^p$, for all x satisfying $c_i(x) \geq 0$, we have $-h(x) \geq 0$, which further suggests that $\mathcal{X}_h \subseteq \mathcal{X}$, where $\mathcal{X}$ is defined in (2). Furthermore, (32) indicates that for all $x \in \mathcal{X}_h$, $D_j u(x) \leq 1$, $\forall j \in \mathbb{Z}_1^q$. Thus, for any $x \in \mathcal{X}_h$, following the control law $u(x)$ will ensure that the input constraints (3) and the CBF condition (4) are satisfied simultaneously. It is straightforward to show that if $x(0) \in \mathcal{X}_h$, then $u(x(t)) \in \mathcal{U}$, $\dot{h}(x(t)) \geq -\alpha(h(x(t)))$, and $x(t) \in \mathcal{X}_h$ for all $t \geq 0$. Therefore, (c) is proved. ■

Remark 7: Maximizing μ_0, i.e., the smallest eigenvalue of Q, in (29), is a proxy mechanism for maximizing the volume of $\mathcal{X}_h = \{x | h(x) = y^T(x) Q y(x) \geq 0\}$, denoted by $\mathrm{vol}(\mathcal{X}_h)$.

Remark 8: By removing (15), one can solve problem (10) to synthesize CBFs without considering input constraints. Similarly, by removing (28) and (32), one can use Algorithm 1 to refine CBFs without input constraints.

Remark 9: If we only consider that $x(0)$ is in the CBF-defined set, we can replace $h_0(x)$ in (26), and (30) with $h(x)$ so that CBF condition (4) only needs to hold in $\mathcal{X}_h$ instead of $\mathcal{X}$. This can possibly further improve the volume of $\mathcal{X}_h$.

From the proof of Theorem 1, we notice that $h(x)$ and $u(x)$ from solving (10) satisfy

$$\dot{h}(x) = L_f h(x) + L_B h(x) u(x) \geq 0, \quad \forall x \in \bar{\mathcal{X}}, \tag{36a}$$

$$x \in \underline{\mathcal{X}}, \; u(x) \in \underline{\mathcal{U}}, \quad \forall x \in \mathcal{X}_h, \tag{36b}$$

with $\mathcal{X}_h \subseteq \mathcal{X}$. On the other hand, the constraints (26)–(28) of problem (25) in **Step 0** aim to ensure

$$\dot{h}(x) + \alpha(h(x)) - \varepsilon \geq 0, \quad \forall x \in \bar{\mathcal{X}}, \tag{37a}$$

$$x \in \mathcal{X}, \; u(x) \in \mathcal{U}, \quad \forall x \in \mathcal{X}_h, \tag{37b}$$

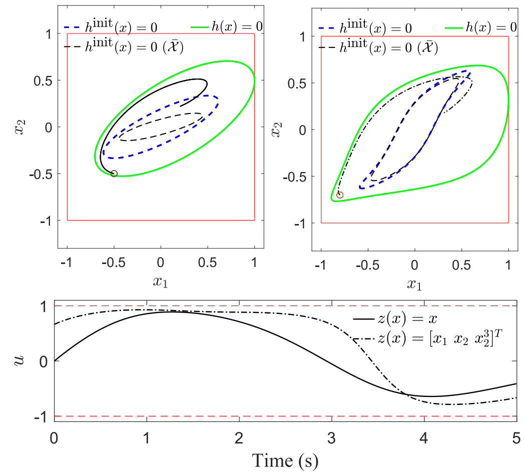

Fig. 2. CBFs in the presence of control limits under $z(x) = x$ (top left) and $z(x) = [x_1\ x_2\ x_2^3]^T$ (top right) as well as control input trajectories (bottom) under the controllers from Algorithm 1 given an initial state (illustrated by circles).

for some constant $\varepsilon \geq 0$. By comparing the constraints (36) and (37) we conjecture that $h(x)$ and $u(x)$ from solving (10) have a high chance to satisfy (37).

Instead of using the controller $u(x)$ output by Algorithm 1, we can alternatively compute the min-norm control signal at each time instant by solving a QP problem [1]:

$$\min_u \|u\| \text{ s.t. } L_f h + L_B h u \geq -\alpha(h(x)) \text{ and } u \in \mathcal{U}. \tag{38}$$

From the analysis in Theorem 2, we know that the optimization problem is always feasible as long as $x(0) \in \mathcal{X}_h$.

IV. Simulation Results

We now test the proposed approach on two numerical examples. We used MATLAB with YALMIP [21] and Mosek [22] to solve all the optimization problems. All codes are available at https://github.com/boranzhao/cbf-sos.

Example 1: Consider a system with an unstable equilibrium at the origin: $\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix} x_1 \\ -x_1 + 0.5x_2 + x_2^3\end{bmatrix} + \begin{bmatrix}1\\0\end{bmatrix} u$, where $x = [x_1, x_2]^T \in \mathbb{R}^2$ and $u \in \mathbb{R}$ are the state and control vectors of the system. The state constraints are given by $x \in \mathcal{X} = \{x | x_1^2 \leq 1,\ x_2^2 \leq 1\}$, illustrated by the red boxes in Fig. 2. The control input is constrained by $|u| \leq 1$.

For applying Theorem 1 for initial CBF design, we tested both $z(x) = x$ and $z(x) = [x_1\ \ x_2\ \ x_2^3]^T$ for obtaining the linear-like dynamics (7), and the degrees of X and Y to be 0 and 2, respectively, for both cases. Additionally, we selected P_0 (used in defining $h_0(x)$ via (8)) to be 2 and 3, respectively, for $z(x) = x$ and $z(x) = [x_1\ x_2\ x_2^3]^T$, to get the smallest $\bar{\mathcal{X}}$ such that $\mathcal{X} \subseteq \bar{\mathcal{X}}$. To reduce the conservatism, we replaced the term dependent on $h_0(x)$ in each of (11)–(15) with a term dependent on $c_i(x)$ so that the corresponding condition holds in $\mathcal{X}$ instead of $\bar{\mathcal{X}}$, as explained in Remark 5. For comparisons, we also included the CBFs obtained using $\bar{\mathcal{X}}$. For refining the CBF via Algorithm 1, we chose the degrees of $y(x)$ and $u(x)$ to be 1 and 3, respectively, for $z(x) = x$, while they were selected to be 3 and 5, respectively, for $z(x) = [x_1\ x_2\ x_2^3]^T$. The initial and refined CBFs obtained are depicted in Fig. 2. One can see that different linear-like representations led to different results.

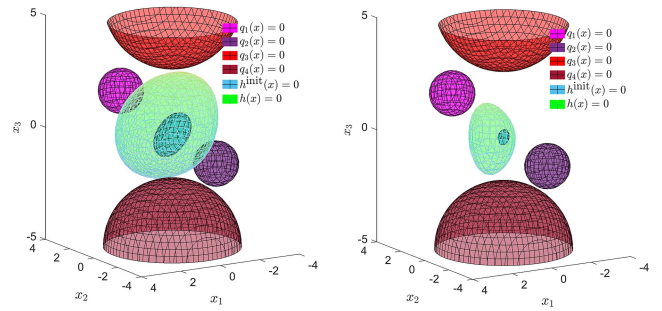

Fig. 3. CBFs in the absence (left) and presence (right) of control limits.

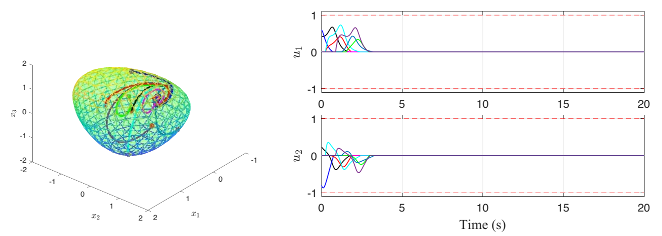

Fig. 4. State trajectories (left) and control inputs (right) under different initial states. The red star denotes the origin, while the red dashed lines denote control limits.

Moreover, $z(x)$ of a higher dimension tends to yield CBFs with larger invariant sets at the price of increased computational cost. Figure 2 also depicts state and input trajectories given an initial state. As expected, the states remained in $\mathcal{X}_h$ and the input constraints were satisfied.

Example 2: Consider a system borrowed from [12], where $x = [x_1, x_2, x_3]^T \in \mathbb{R}^3$ and $u = [u_1, u_2]^T \in \mathbb{R}^2$ are the state and control vectors of the system. The unsafe regions are characterized by four spheres defined by $q_i(x) < 0$ $(i = 1, 2, 3, 4)$, where the expression of the polynomial function $q_i(x)$ is given in [12]. The safe region is therefore characterized by $\mathcal{X} = \{x \in \mathbb{R}^3 | c_i(x) \leq 0,\ i = 1, 2, 3, 4\}$, where $c_i(x) = -q_i(x)$, $i = 1, 2, 3, 4$. The control inputs are constrained by $|u_i| \leq 1$ for $i = 1, 2$. For applying Theorem 1 for an initial CBF, we selected $z(x) = x$, and the degrees of X and Y to be 0 and 2, respectively. Additionally, we selected P_0 (used in defining $h_0(x)$ via (8)) to be 10. For the redesign via Algorithm 1, we chose the degrees of $y(x)$ of $u(x)$ to be 2 and 4, respectively. The initial and refined CBFs in the absence and presence of input constraints are depicted in Fig. 3.

To verify if the function $h(x)$ from Algorithm 1 is indeed an CBF, we simulated the system under a few initial states in $\mathcal{X}_h$, and the min-norm control law in (38). As shown in Fig. 4, for all the tested initial states, the system states stayed in $\mathcal{X}_h$, and the control limits were respected.

V. Conclusion

This letter presents an SOS optimization-based method to synthesize CBFs for nonlinear polynomial systems with input constraints. The method consists of an initial design step that jointly searches for an initial CBF and a control law through SOS optimization based on a linear-like representation of the nonlinear dynamics, and an iterative redesign step that refines the initial CBF by alternatively updating it and the control law through SOS optimization. Numerical examples validate the effectiveness of the proposed approach.

Our future work includes extending the proposed method to synthesize CBFs under model uncertainties.

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, Aug. 2017.

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

[3] W. Xiao and C. Belta, "High-order control barrier functions," *IEEE Trans. Autom. Control*, vol. 67, no. 7, pp. 3655–3662, Jul. 2022.

[4] M. Rauscher, M. Kimmel, and S. Hirche, "Constrained robot control using control barrier functions," in *Proc. IROS*, 2016, pp. 279–285.

[5] Y. Cheng, P. Zhao, and N. Hovakimyan, "Safe and efficient reinforcement learning using disturbance-observer-based control barrier functions," in *Proc. L4DC*, 2023, pp. 104–115.

[6] P. A. Parrilo, "Structured Semidefinite programs and Semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, California Inst. Technol., Pasadena, CA, USA, 2000.

[7] S. Prajna, A. Papachristodoulou, and F. Wu, "Nonlinear control synthesis by sum of squares optimization: A Lyapunov-based approach," in *Proc. 5th Asian Control Conf.*, vol. 1, 2004, pp. 157–165.

[8] W. Tan and A. Packard, "Searching for control Lyapunov functions using sums of squares programming," in *Proc. Allerton Conf. Commun. Control Comput.*, 2004, pp. 210–219.

[9] H. Ichihara, "State feedback synthesis for polynomial systems with input saturation using convex optimization," in *Proc. ACC*, 2007, pp. 2334–2339.

[10] N. Vafamand, M.-M. Mardani, A. Khayatian, and M. Shasadeghi, "Non-iterative SOS-based approach for guaranteed cost control design of polynomial systems with input saturation," *IET Control Theory Appl.*, vol. 11, no. 16, pp. 2724–2730, 2017.

[11] C. Maier, C. Böhm, F. Deroo, and F. Allgöwer, "Predictive control for polynomial systems subject to constraints using sum of squares," in *Proc. CDC*, 2010, pp. 3433–3438.

[12] L. Wang, D. Han, and M. Egerstedt, "Permissive barrier certificates for safe stabilization using sum-of-squares," in *Proc. Amer. Control Conf.*, 2018, pp. 585–590.

[13] A. Clark, "Verification and synthesis of control barrier functions," in *Proc. CDC*, 2021, pp. 6105–6112.

[14] H. Dai and F. Permenter, "Convex synthesis and verification of control-Lyapunov and barrier functions with input constraints," 2022, in *Proc. Amer. Control Conf.*, 2023, pp. 4116–4123. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10156043

[15] K. P. Wabersich and M. N. Zeilinger, "Predictive control barrier functions: Enhanced safety mechanisms for learning-based control," *IEEE Trans. Autom. Control*, vol. 68, no. 5, pp. 2638–2651, May 2023.

[16] A. Robey et al., "Learning control barrier functions from expert demonstrations," in *Proc. CDC*, 2020, pp. 3717–3724.

[17] M. Srinivasan, A. Dabholkar, S. Coogan, and P. A. Vela, "Synthesis of control barrier functions using a supervised machine learning approach," in *Proc. IROS*, 2020, pp. 7139–7145.

[18] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier–value functions for safety-critical control," in *Proc. CDC*, 2021, pp. 6814–6821.

[19] S. Tonkens and S. Herbert, "Refining control barrier functions through hamilton-Jacobi reachability," in *Proc. IROS*, 2022, pp. 13355–13362.

[20] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[21] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. IEEE Int. Symp. Comput.-Aided Control Syst. Design*, 2004, pp. 284–289.

[22] E. D. Andersen and K. D. Andersen, "The MOSEK interior point optimizer for linear programming: An implementation of the homogeneous algorithm," in *High Performance Optimization*. New York, NY, USA: Springer, 2000, pp. 197–232.