


SOS Construction of Compatible Control Lyapunov and Barrier Functions

Conference Paper**Author(s):**

Schneeberger, Michael; [Dörfler, Florian](#) ; Mastellone, Silvia

Publication date:

2023-11-22

Permanent link:

<https://doi.org/10.3929/ethz-b-000660773>

Rights / license:

[Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International](#)

Originally published in:

IFAC-PapersOnLine 56(2), <https://doi.org/10.1016/j.ifacol.2023.10.1058>

SOS Construction of Compatible Control Lyapunov and Barrier Functions

Michael Schneeberger * Florian Dörfler ** Silvia Mastellone *

* *Institute of Electrical Engineering, FHNW, Windisch, Switzerland*
(e-mails: michael.schneeberger@fhnw.ch, silvia.mastellone@fhnw.ch)

** *Department of Information Technology and Electrical Engineering, ETH Zürich, Zürich, Switzerland, (e-mail: dorfler@control.ee.ethz.ch)*

Abstract:

We propose a novel approach to certify closed-loop stability and safety of a constrained polynomial system based on the combination of Control Lyapunov Functions (CLFs) and Control Barrier Functions (CBFs). For polynomial systems that are affine in the control input, both classes of functions can be constructed via Sum Of Squares (SOS) programming. Using two versions of the Positivstellensatz we derive an SOS formulation seeking a rational controller that — if feasible — results in compatible CLF and multiple CBFs.

Copyright © 2023 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

1. INTRODUCTION

When dealing with systems that have state constraints, it is crucial to have a controller that ensures both stability and compliance with the constraints. In most cases, feedback control design focuses mainly on achieving stability, while protection functions only engage when constraints are violated. Unfortunately, this approach results in downtime and the need to investigate faults. By contrast, a controller that is both stable and safe can trade off control performance for the ability to prevent unsafe states.

The focus of this paper is on constructing compatible CLFs and CBFs that can certify both stability and safety in control systems. A CLF establishes conditions for the existence of a stabilizing controller for a given control system. Similarly, a CBF guarantees the existence of a controller that can render the control system safe. According to Wieland and Allgöwer (2007), a system is considered safe if any state trajectory starting from a safe set of states remains within an allowable region defined by the state constraints. For systems that are affine in the input, a controller that meets the CLF and CBF conditions can be implemented by solving an online Quadratic Programming (QP), see Ames et al. (2019). However, for some states, these conditions may conflict with each other. For both to hold jointly for all states, the control-sharing property is additionally required, see Grammatico et al. (2013); Xu (2016). Finding such compatible CLF and CBF is generally difficult. For polynomial systems, however, this can be achieved by formulating SOS constraints and solving them via Semidefinite Programming (SDP).

The contributions of this paper is twofold: first, we derive SOS constraints on compatible CLF and multiple CBFs using two versions of the Positivstellensatz, and second, an algorithm is developed that efficiently finds solutions to these SOS constraints by maximizing a surrogate of the volume of the safe set. The conditions on compatible CBFs are formulated in Isaly et al. (2022); Tan and Dimarogonas (2022) but without giving a method to construct them.

A constructive approach described by Clark (2021) is based on the introduction of additional SOS constraints to enforce compatibility. In this paper, we reveal a correspondence between the SOS constraint derived from the CLF, resp. CBF, condition, and the existence of a rational controller that renders the closed-loop system stable, resp. safe. By restricting such controllers to be identical, we derive a new set of SOS constraints that guarantee compatibility between a CLF and multiple CBFs without the introduction of additional SOS constraints.

These SOS constraints contain bilinear terms, and hence, cannot be directly converted to an SDP. We therefore present an alternating algorithm that searches simultaneously for a CLF and multiple CBFs by repeatedly solving two SDPs. In particular, the algorithm seeks to maximize the volume of the safe set, which is given by the intersection of the invariant sets defined by each CBF. Multiple CBFs offer additional flexibility to increase the volume of the safe set when a single CBF does not suffice. Similar approaches were presented in Anghel et al. (2013); Kundu et al. (2019); Wang et al. (2022) but they either only searched for a single CLF or a single CBF. Korda et al. (2014) proposed another intriguing approach for identifying a safe set using an infinite-dimensional linear programming problem. We demonstrate the utility of our approach with a power converter control example for which safety is of paramount importance.

The paper is structured as follows: In Section 2, we introduce the notation adopted in the paper and review some preliminaries. Then, we recall the definition of a CLF and CBF and derive a rational controller in Section 3. In Section 4, we combine the CLF and CBF by introducing the control-sharing property. Section 5 defines the SOS program encoding the CLF and CBF conditions, and Section 6 presents the algorithm that solves the SOS program. Numerical simulation are given in Section 7. Finally, Section 8 is dedicated to concluding remarks and future work.

2. PRELIMINARIES & NOTATION

2.1 Notation

Across this paper, we adopt the following notation. The shorthand $[t] := \{1, 2, \dots, t\}$ is used to denote a range of numbers. A scalar function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ is positive definite w.r.t. x^* , if $V(x^*) = 0$ and $V(x) > 0$ for $x \neq x^*$. $R[x]$ refers to the set of scalar polynomials in variables $x \in \mathbb{R}^n$, and $\Sigma[x]$ refers to the set of scalar SOS polynomials in x . A polynomial $p(x)$ is an SOS polynomial if it can be written as $p(x) = \sum_{i=1}^k g_i(x)^2$ for $g_i(x) \in R[x]$ and $i \in [k]$. If $p(x) \in \Sigma[x]$, it can be expanded to

$$p(x) = Z(x)^T Q Z(x),$$

where $Z(x)$ is a vector of monomials in x , and Q is a square positive semidefinite matrix. In the following, we consider a polynomial control system described as

$$\dot{x} = \mathcal{F}_c(x, u) = f(x) + G(x)u_c, \quad (1)$$

where $f(x) \in (R[x])^n$ and $G(x) \in (R[x])^{n \times m}$ are polynomial matrices, and $u_c \in \mathbb{R}^m$ is the control input vector. System (1) with polynomial state feedback control policy $u(x) \in (R[x])^m$ results in a closed-loop polynomial system of the form

$$\dot{x} = \mathcal{F}_a(x) = f(x) + G(x)u(x) \quad (2)$$

The state x^* is called an equilibrium of (2), if $\mathcal{F}_a(x^*) = 0$.

A set \mathcal{X} is called forward invariant (Khalil, 2002) with respect to (2) if for every $x_0 \in \mathcal{X}$, $x(t) \in \mathcal{X}$ for all $t \in \mathbb{R}_+$. We say that a system (2) is safe (Wieland and Allgöwer, 2007) w.r.t. an allowable set of states $\mathcal{X}_a \subseteq \mathbb{R}^n$ and the safe states $\mathcal{X}_s \subseteq \mathbb{R}^n$, if \mathcal{X}_s is forward invariant and $\mathcal{X}_s \subseteq \mathcal{X}_a$.

2.2 SOS Programming

An SOS program minimizing a quadratic cost function subject to SOS constraints is defined as follows:

$$\begin{aligned} & \underset{v}{\text{minimize}} && v^T P v + c^T v \\ & \text{subject to} && p_i(x, v) \in \Sigma[x] \quad i = 1, \dots, N. \end{aligned}$$

Here $P \in \mathbb{R}^{l \times l}$ is a positive semi-definite matrix, $c \in \mathbb{R}^l$ is a vector, and $p_i(x, v) \in \Sigma[x]$ is an SOS polynomial in $x \in \mathbb{R}^n$ parameterized by $v \in \mathbb{R}^l$, i.e. it can be expanded as

$$p_i(x, v) = Z_i(x)^T Q_i(v) Z_i(x), \quad (3)$$

where $Q_i(v) = Q_{0,i} + v_1 Q_{1,i} + \dots + v_l Q_{l,i}$ is a square positive semidefinite matrix linearly parameterized by v encoding the coefficients of the polynomial.

When deriving the SOS constraints in the following subsections, the parameter v is omitted for simplicity, and the polynomial is simply written as $p_i(x)$ instead.

2.3 Positivstellensätze

In the following, we present a version of the Positivstellensatz that can be regarded as a specialization of the weak Positivstellensatz. This version will be relevant for our analysis later on.

Theorem 1. Given polynomials $f_1(x), f_2(x), g_1(x), h_1(x), \dots, h_{m+1}(x) \in R[x]$ such that

$$\left\{ x \in \mathbb{R}^n \mid f_1(x) \geq 0, f_2(x) \geq 0, g_1(x) \neq 0, \right. \\ \left. h_1(x) = 0, \dots, h_{m+1}(x) = 0 \right\} = \emptyset, \quad (4)$$

then there exist polynomials

- $f_{\text{cone}}(x) = s_1(x)f_1(x) + s_2(x)f_2(x) + s_3(x)f_1(x)f_2(x)$
- $h_{\text{ideal}}(x) = p_1(x)h_1(x) + \dots + p_{m+1}(x)h_{m+1}(x)$

such that

$$-f_{\text{cone}}(x) - h_{\text{ideal}}(x) - g_1(x)^{2k} \in \Sigma[x] \quad (5)$$

for all $x \in \mathbb{R}^n$, where $p_1(x), \dots, p_{m+1}(x) \in R[x]$ are polynomials, $s_1(x), s_2(x), s_3(x) \in \Sigma[x]$ are SOS polynomials, and $k \in \mathbb{N}_+$.

Proof. Using (Bochnak et al., 2013, Theorem 4.4.2), we note that the cone P generated by $f_1(x)$ and $f_2(x)$ is contained in $\left\{ s_0(x) + s_1(x)f_1(x) + s_2(x)f_2(x) + s_3(x)f_1(x)f_2(x) \mid s_0, s_1, s_2, s_3 \in \Sigma[x] \right\}$. ■

If the set (4) is empty, the Positivstellensatz ensures that the polynomials $s_1(x), s_2(x), \dots, p_{m+1}(x)$ exist without specifying their degree. To find these polynomials computationally will require to iteratively increase the degree of the polynomials until a solution can be found for (5). An increase in the degree of the polynomials will, however, deteriorate computation time. For many practical examples of interest, however, low-degree polynomials suffice.

The following theorem states a version of the Positivstellensatz that can be seen as specializations of Putinar's Positivstellensatz.

Theorem 2. Given polynomials $f_1(x), f_2(x), h_1(x), \dots, h_{m+1}(x) \in R[x]$ such that the set $\left\{ x \in \mathbb{R}^n \mid f_2(x) \geq 0 \right\}$ is compact and

$$\left\{ x \in \mathbb{R}^n \mid f_1(x) \geq 0, f_2(x) \geq 0, \right. \\ \left. h_1(x) = 0, \dots, h_{m+1}(x) = 0 \right\} = \emptyset, \quad (6)$$

then there exist polynomials

- $f_{\text{cone}}(x) = f_1(x) + s_2(x)f_2(x)$
- $h_{\text{ideal}}(x) = p_1(x)h_1(x) + \dots + p_{m+1}(x)h_{m+1}(x)$

such that

$$-f_{\text{cone}}(x) - h_{\text{ideal}}(x) \in \Sigma[x] \quad (7)$$

for all $x \in \mathbb{R}^n$, where $p_1(x), \dots, p_{m+1}(x) \in R[x]$ are polynomials and $s_2(x) \in \Sigma[x]$ is an SOS polynomial.

Proof. Let's consider the closed semialgebraic set

$$K := \left\{ x \in \mathbb{R}^n \mid f_2(x) \geq 0, \right. \\ \left. h_k(x) \geq 0 \quad k = 1, \dots, m+1, \right. \\ \left. -h_k(x) \geq 0 \quad k = 1, \dots, m+1 \right\},$$

and the quadratic module

$$M(f_2(x), h_1(x), -h_1(x), \dots, -h_{m+1}(x)),$$

then M is Archimedean according to (Laurent, 2009, Theorem 3.17) using the fact that $\{x \in \mathbb{R}^n \mid f_2(x) \geq 0\}$ is compact. The empty set condition (6) is equivalent to the condition that $-f_1(x) > 0$ on K , and, therefore, K is equivalent to

$$-f_1(x) = s_0(x) + s_2(x)f_2(x) + \sum_{k=1}^{m+1} \left(s_{h,k}^p(x) - s_{h,k}^n(x) \right) h_k(x), \quad (8)$$

where $s_0(x), s_2(x), s_{h,1}^p(x), \dots, s_{h,m+1}^n(x) \in \Sigma[x]$ (Laurent, 2009, Theorem 3.20). We note that $s_{h,k}^p(x) - s_{h,k}^n(x)$ can be replaced by a polynomial $p_k(x)$. ■

3. CLOSED-LOOP STABILITY AND SAFETY

In this section, we first review the concept of CLFs and CBFs. Given a polynomial system (1), we then propose an approach to construct a rational controller resulting from the SOS formulation of CLF and CBF conditions. We prove global asymptotic stability and safety of the resulting closed-loop system (2) given such a controller.

3.1 Control Lyapunov Function

For a control system (1) to be stabilizable around the equilibrium point x^* , we require the existence of a control input $u_c \in \mathbb{R}^m$ at every state $x \in \mathbb{R}^n$ that renders the sublevel sets of a scalar polynomial $V(x) \in R[x]$ forward invariant. This motivates the following definition of a CLF:

Definition 1. (Isidori (1995)) Consider a differentiable function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $V(x^*) = 0$ and $V(x) > 0$ for all $x \neq x^*$. Such a scalar function is called a Control Lyapunov Function (CLF) for the control system (1) if

$$\nabla V(x)^T f(x) < 0 \quad (9)$$

for all $x \in \{x \in \mathbb{R}^n \mid \nabla V(x)^T G(x) = 0, x \neq x^*\}$.

The inequality (9) can also be formulated as the empty set condition

$$\left\{ x \in \mathbb{R}^n \mid \nabla V(x)^T f(x) \geq 0, \nabla V(x)^T G(x) = 0, x \neq x^* \right\} = \emptyset. \quad (10)$$

Condition (10) when restricted to *polynomial* CLFs can be solved via SOS programming. Hence, we restrict our attention to polynomial scalar functions $V \in R[x]$ for the rest of the paper.

Next, we replace the inequality constraints $x = (x_1, \dots, x_n) \neq x^*$ in (10) by a single inequality constraint $l(x) \neq 0$ (c.f. Tan and Packard (2004)), where $l(x^*) = 0$ and $l(x) \neq 0$ elsewhere. A single inequality constraint has the advantage that it translates into a simpler SOS constraint. The resulting empty set condition equivalent to (10) is then given by:

$$\left\{ x \in \mathbb{R}^n \mid \nabla V(x)^T f(x) \geq 0, \nabla V(x)^T G(x) = 0, l(x) \neq 0 \right\} = \emptyset. \quad (11)$$

According to Theorem 1, by choosing $k = 1$, the empty set condition (11) becomes

$$-s_1(x)\nabla V(x)^T f(x) - \nabla V(x)^T G(x)p(x) - l(x)^2 \in \Sigma[x], \quad (12)$$

where $s_1(x) \in \Sigma[x]$ is an SOS polynomial, and $p(x) = [p_1(x) \dots p_m(x)]^T \in (R[x])^m$ is a vector of polynomials.

If $s_1(x)$ is strictly positive w.r.t. x^* , the following controller — in the form of a rational function — naturally results from the SOS constraint (12):

$$u_{CLF}(x) := p(x)/s_1(x). \quad (13)$$

Remark 1. Fixing the controller in (1) to $u_c = u_{CLF}(x)$ results in the closed-loop system (2). Hence, stability is asserted by the existence of a Lyapunov function $V(x)$. For the sake of readability, however, we keep denoting $V(x)$ a CLF.

Lemma 1. Given a CLF $V(x)$ for the control system (1) with $s_1(x)$ in (12) strictly positive, the closed-loop system (2) using controller $u(x) = u_{CLF}(x)$ defined in (13) is globally asymptotically stable (GAS).

Proof. The polynomial CLF $V(x)$ is strictly positive definite w.r.t. x^* . Hence, $V(x)$ is radially unbounded. From (12), we derive $s_1(x)\nabla V(x)^T f(x) + \nabla V(x)^T G(x)p(x) < 0$ for all $x \neq x^*$. Since $s_1(x)$ is strictly positive w.r.t. x^* , the division by $s_1(x)$ results in $\nabla V(x)^T (f(x) + G(x)u_{CLF}(x)) < 0$ for all $x \neq x^*$. GAS follows from Theorem 4.2 in Khalil (2002). ■

The strict positivity condition on $s_1(x)$ can be enforced by the SOS constraint

$$s_1(x) - \epsilon_{s1} \in \Sigma[x], \quad (14)$$

for some $\epsilon_{s1} > 0$.

3.2 Control Barrier Function

Similar to the stability argument, safety can be asserted with the existence of a scalar function $B_1 : \mathbb{R}^n \rightarrow \mathbb{R}$. Specifically, the control system (1) is safe w.r.t. the set of safe states

$$\mathcal{X}_{s,1} = \left\{ x \in \mathbb{R}^n \mid B_1(x) \leq 0 \right\}, \quad (15)$$

if there exists a control input $u_c \in \mathbb{R}^m$ for every state $x \in \mathbb{R}^n$ such that $\mathcal{X}_{s,1}$ is forward invariant. This motivates the following definition of a CBF (cf. Wang et al. (2022)):

Definition 2. Consider a differentiable function $B_1 : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $\mathcal{X}_{s,1}$ is non-empty. Such a scalar function is called a Control Barrier Function (CBF) for the control system (1) if

$$\nabla B_1(x)^T f(x) < 0 \quad (16)$$

for all $x \in \{x \in \mathbb{R}^n \mid \nabla B_1(x)^T G(x) = 0, B_1(x) = 0\}$.

An alternative definition of a CBF (cf. Ames et al. (2019)) involves a supremum and a class K function. This definition, however, is not well suited for polynomial optimization since the resulting functions cannot be directly translated to polynomial inequalities. Similarly to CLF, we restrict our focus on *polynomial* CBFs $B_1(x) \in R[x]$ for the rest of the paper.

Inequality (16) can also be formulated as the empty set condition

$$\left\{ x \in \mathbb{R}^n \mid \nabla B_1(x)^T f(x) \geq 0, \nabla B_1(x)^T G(x) = 0, \right. \\ \left. B_1(x) = 0 \right\} = \emptyset. \quad (17)$$

Assumption 1. The set $\{x \in \mathbb{R} \mid B_1(x) = 0\}$ is compact.

According to Theorem 2 and Assumption 1, the empty set condition (17) is equivalent to the SOS constraint

$$-s_1(x)\nabla B_1(x)^T f(x) - \nabla B_1(x)^T G(x)p(x) \\ - p_{m+1}(x)B_1(x) \in \Sigma[x], \quad (18)$$

where $s_1(x) \in \Sigma[x]$ is an SOS polynomial, $p(x) = [p_1(x) \dots p_m(x)]^T \in (R[x])^m$ is a vector of polynomials, and $p_{m+1}(x) \in R[x]$ is a scalar polynomial.

If $s_1(x)$ in (18) is strictly positive w.r.t. x^* , there exists a controller defined by the rational function

$$u_{\text{CBF},1}(x) := p(x)/s_1(x). \quad (19)$$

Remark 2. As in Remark 1, we denote $B(x)$ as CBF for both the control and closed-loop system.

Lemma 2. Given a CBF $B_1(x)$ for the control system (1) with $s_1(x)$ in (18) strictly positive, the set $\mathcal{X}_{s,1}$ in (15) — if compact (cf. Assumption 1) — is forward invariant w.r.t. system (2) using controller $u(x) = u_{\text{CBF},1}(x)$ defined in (19).

Proof. A compact zero sublevel set of a scalar polynomial $B_1(x)$ is forward invariant w.r.t. closed-loop system (2) with control $u(x)$, if $\nabla B_1(x)^T (f(x) + G(x)u(x)) < 0$ for all $x \in \{x \in \mathbb{R}^n \mid B_1(x) = 0\}$ (Blanchini, 1999, Nagumo's Theorem 3.1). From (18), we derive $s_1(x)\nabla B_1(x)^T f(x) + \nabla B_1(x)^T G(x)p(x) < 0$ for all $x \in \{x \in \mathbb{R}^n \mid B_1(x) = 0\}$. Since $s_1(x)$ is strictly positive w.r.t. x^* , the division by $s_1(x)$ results in $s_1(x)\nabla B_1(x)^T (f(x) + G(x)u_{\text{CBF},1}(x)) < 0$ for all $x \in \{x \in \mathbb{R}^n \mid B_1(x) = 0\}$. ■

4. SOS CONSTRUCTION OF COMPATIBLE CLF AND MULTIPLE CBFs

In this section, we are interested in finding a CLF and multiple CBFs that are compatible with each other. This is achieved by deriving a new set of SOS constraints that result from restricting the rational controllers (13) and (19) to be identical.

Multiple CBFs provide an additional flexibility to increase the volume of the safe set \mathcal{X}_s , which is defined by the intersection of the zero sublevel sets of all the CBFs. Intuitively, this makes sense since the solution of a single CBF can be recovered by equating all CBFs.

4.1 Multiple Control Barrier Functions

Consider a set of CBFs $\{B_i(x)\}_{i \in [t]}$ for some $t \in \mathbb{N}_+$, each defining a forward invariant set

$$\mathcal{X}_{s,i} = \left\{ x \in \mathbb{R}^n \mid B_i(x) \leq 0 \right\}. \quad (20)$$

Multiple CBFs are compatible with each other if they have the control-sharing property (cf. Grammatico et al. (2013)).

Definition 3. Consider the closed-loop system (2) with feedback controller $u(x)$. A set of CBFs $\{B_i(x)\}_{i \in [t]}$ with associated controllers $u_{\text{CBF},i}(x)$ rendering (20) invariant has the control-sharing property, if $u(x) = u_{\text{CBF},i}(x)$ for all $i \in [t]$.

The sets of compatible CBFs $\{B_i(x)\}_{i \in [t]}$ form a new forward invariant set \mathcal{X}_s , called safe set, defined as the intersection of the invariant sets from each $B_i(x)$:

$$\mathcal{X}_s := \bigcap_{i \in [t]} \mathcal{X}_{s,i} = \left\{ x \in \mathbb{R}^n \mid B_i(x) \leq 0 \forall i \in [t] \right\}. \quad (21)$$

Consider a set of allowable states of the form

$$\mathcal{X}_a = \left\{ x \in \mathbb{R}^n \mid w_i(x) \leq 0 \forall i \in [t] \right\}, \quad (22)$$

where $w_1(x), \dots, w_t(x) \in R[x]$ are polynomials. Note that for each polynomial $w_i(x)$, that defines the allowable set \mathcal{X}_a , a corresponding CBF can be assigned.

Assumption 2. The sets $\{x \in \mathbb{R} \mid w_i(x) \leq 0\}_{i \in [t]}$ are compact.

To conclude safety, the resulting invariant set \mathcal{X}_s needs to be contained in the allowable set, i.e. $\mathcal{X}_s \subseteq \mathcal{X}_a$. This can be achieved by restricting the invariant set of each CBF to the zero sublevel set of the corresponding $w_i(x)$ as stated by the following lemma.

Lemma 3. Consider a set of CBFs $\{B_i(x)\}_{i \in [t]}$ that have the control-sharing property with input $u(x)$. If for every $i \in [t]$

$$\mathcal{X}_{s,i} \subseteq \left\{ x \in \mathbb{R}^n \mid w_i(x) \leq 0 \right\}, \quad (23)$$

then the closed-loop system (2) using controller $u(x)$ is safe w.r.t. the safe set \mathcal{X}_s (21) and the allowable set \mathcal{X}_a (22).

Proof. For a closed-loop system to be safe, we need to show that \mathcal{X}_s is (i) forward invariant and (ii) $\mathcal{X}_s \subseteq \mathcal{X}_a$. Note that the sets $\mathcal{X}_{s,i}$, $i \in [t]$ are compact due to (23). By Lemma 2, every $\mathcal{X}_{s,i}$, $i \in [t]$, is forward invariant w.r.t. the closed-loop system, and the intersection of forward invariant sets is also forward invariant. This proves the condition (i). The condition (ii) can be seen from

$$\mathcal{X}_s = \bigcap_{i \in [t]} \mathcal{X}_{s,i} \subseteq \bigcap_{i \in [t]} \left\{ x \in \mathbb{R}^n \mid w_i(x) \leq 0 \right\} = \mathcal{X}_a. \quad \blacksquare$$

Finally, we formalize (23) as the empty set condition

$$\left\{ x \in \mathbb{R}^n \mid w_i(x) \geq 0, -B_i(x) \geq 0 \right\} = \emptyset \quad (24)$$

for all $i \in [t]$. Under Theorem 2 and Assumption 2, condition (24) is equivalent to the following SOS condition:

$$B_i(x) - s_{4,i}(x)w_i(x) \in \Sigma[x] \quad (25)$$

where $s_{4,i} \in \Sigma[x]$ is an SOS polynomial for all $i \in [t]$.

4.2 Compatible CLF and CBFs

The notion of stability and safety can be unified by the existence of both a CLF and multiple CBFs that are compatible with each other via the following control-sharing property.

Definition 4. Consider a closed-loop system (2) with feedback controller $u(x)$. A CLF with associated controller

$u_{\text{CLF}}(x)$ stabilizing the system to x^* and set of CBFs $\{B_i(x)\}_{i \in [t]}$ with associated controllers $u_{\text{CBF},i}(x)$ rendering (20) invariant have the control-sharing property, if $u(x) = u_{\text{CLF}}(x)$ and $u(x) = u_{\text{CBF},i}(x)$ for all $i \in [t]$.

With Definition 4 in place, we can state our main proposition:

Proposition 4. Given a CLF $V(x)$ and a set of CBFs $\{B_i(x)\}_{i \in [t]}$ that have the control-sharing property with input $u(x)$. If condition (23) holds for every $i \in [t]$, then the closed-loop system (2) with input $u(x)$ is asymptotically stable and safe w.r.t. the allowable set \mathcal{X}_a (22).

Proof. Lemma 1 and Lemma 3 prove stability and safety for the closed-loop system using input $u(x)$. ■

Based on conditions (12), (14), (18) and (25), the stability and safety condition in Lemma 4 can be restated as the SOS constraints:

$$\begin{aligned} & -\nabla V(x)^T (s_1(x)f(x) + G(x)p(x)) - l(x)^2 \\ & \quad \in \Sigma[x], \\ & -\nabla B_i(x)^T (s_1(x)f(x) + G(x)p(x)) \\ & \quad - p_{m+1,i}(x)B_i(x) \in \Sigma[x], \quad \forall i \in [t] \\ & B_i(x) - s_{4,i}(x)w_i(x) \in \Sigma[x], \quad \forall i \in [t] \\ & s_1(x) - \epsilon_{s1} \in \Sigma[x], \end{aligned} \tag{26}$$

where $s_1(x), s_{4,i}(x) \in \Sigma[x]$ are SOS polynomials, $p(x) = [p_1(x) \dots p_m(x)]^T \in (R[x])^m$ is a vector of polynomials, and $p_{m+1,i} \in R[x]$ is a scalar polynomial.

The resulting control input $u(x) = p(x)/s_1(x)$ renders the system asymptotically stable on \mathcal{X}_a and safe w.r.t. the allowable set \mathcal{X}_a and safe set \mathcal{X}_s .

5. SOS PROGRAM

In this section, we state the SOS program that finds a CLF and multiple CBFs for the control system (1). The SOS program maximizes a surrogate of the volume of \mathcal{X}_s subject to the SOS condition (26).

5.1 Cost function

Ideally, the SOS program would optimize the CLF and CBFs over the volume of the safe set \mathcal{X}_s :

$$\begin{aligned} & \underset{v}{\text{minimize}} \quad -\text{vol} \left(\left\{ x \in \mathbb{R}^n \mid B_i(x, v) \leq 0 \text{ for } i \in [t] \right\} \right) \\ & \text{subject to} \quad \text{SOS constraints (26)} \end{aligned} \tag{27}$$

where $B_i(x, v) = Z_{B,i}(x)^T Q_{B,i}(v) Z_{B,i}(x)$ (cf. (3)). To make (27) computationally tractable, the volume of \mathcal{X}_s is surrogated by the traces of the quadratic matrices encoding the $B_i(x)$. The cost then becomes:

$$c_1(v) = \sum_{i=1}^t \text{tr}(Q_{B,i}(v)). \tag{28}$$

Furthermore, we have empirically observed that the outcome is improved by restricting the CBF to a predefined center point. The justification for this lies in the fact that the CBF condition is only valid when $B(x) = 0$

and, therefore, it does not restrict the CBF from taking arbitrary large values when $B(x) \neq 0$.

Let us define a center point $(x_{c,i}, B_{c,i}) \in (\mathbb{R}^n, \mathbb{R})$ for each CBF $B_i(x)$, $i \in [t]$. Then, we define a cost function by the deviation from that point:

$$c_2(v) = \sum_{i=1}^t (B_i(x_{c,i}) - B_{c,i})^2 \tag{29}$$

The cost function of the SOS program is set to

$$c(v) = c_1(v) + c_2(v). \tag{30}$$

5.2 SOS Program

Given the cost (30) and the SOS constraint (26), we summarize the SOS program by:

$$\begin{aligned} & \text{find} \quad V, B_i, s_1, s_{4,i}, p_{m+1,i} \in R[x], \\ & \quad p \in (R[x])^m \\ & \text{minimize} \quad \sum_{i=1}^t \text{tr}(Q_{B,i}) + (B_i(x_{c,i}) - B_{c,i})^2 \\ & \text{subject to} \quad -\nabla V^T (s_1 f + Gp) - l^2 \in \Sigma[x] \\ & \quad -\nabla B_i^T (s_1 f + Gp) - p_{m+1,i} B_i \\ & \quad \quad \quad \in \Sigma[x] \quad \forall i \in [t] \\ & \quad B_i - s_{4,i} w_i \in \Sigma[x] \quad \forall i \in [t] \\ & \quad s_1 - \epsilon_{s1} \in \Sigma[x] \\ & \quad s_{4,i} \in \Sigma[x] \quad \forall i \in [t] \\ & \quad V \in \Sigma[x], V(x^*) = 0 \end{aligned}$$

where $(x_{c,i}, B_{c,i}) \in (\mathbb{R}^n, \mathbb{R})$ are the predefined center points for each CBF.

6. ALGORITHM

The SOS problem in Section 5.2 cannot directly be converted to an SDP problem due to bilinear terms in its decision variables. However, an alternating algorithm can find adequate solutions to this non-convex problem.

6.1 Alternating Algorithm

Consider an abstract optimization problem with n_c SOS inequality constraints defined by $f_1(v_1, v_2) \in (R[x])^{n_c}$ that are bilinear in the decision variables $v_1 \in \mathbb{R}^{n_1}$ and $v_2 \in \mathbb{R}^{n_2}$, and a cost function $f_0(v_1) \in \mathbb{R}$ that only depends on v_1 :

$$\begin{aligned} & \underset{v_1, v_2}{\text{minimize}} \quad f_0(v_1) \\ & \text{subject to} \quad f_1(v_1, v_2) \in (\Sigma[x])^{n_c} \end{aligned} \tag{31}$$

The optimization problem (31) cannot be solved directly with an SOS program. Instead, we propose a method that alternates between searching over one variable while holding fixed the other.

Starting from an iteration $v_1^{(0)}$ and $v_2^{(0)}$, the algorithm at iteration k is defined by:

- Step 1: Substitute $v_2 = v_2^{(k-1)}$ and solve for v_1

$$\begin{aligned} & \underset{v_1}{\text{minimize}} \quad f_0(v_1) \\ & \text{subject to} \quad f_1(v_1, v_2^{(k-1)}) \in (\Sigma[x])^{n_c} \end{aligned} \tag{32}$$

- Step 2: Substitute $v_1 = v_1^{(k)}$ and solve for ϵ and v_2

$$\begin{aligned} & \underset{\epsilon, v_2}{\text{minimize}} \quad \epsilon \\ & \text{subject to} \quad f_1(v_1^{(k)}, v_2) + s(v_1^{(k)})\epsilon \in (\Sigma[x])^{n_c} \end{aligned} \tag{33}$$

The algorithm terminates when $|f_0(v_1^{(k)}) - f_0(v_1^{(k-1)})|$ is lower than a given threshold.

Lemma 5. If the initial pair $(v_1^{(0)}, v_2^{(0)})$ is feasible for (31), then feasibility is maintained along the alternating algorithms and the cost $f_0(v_1^{(k)})$ in (32) is non-increasing with iteration k .

Proof. First, we show that if Step 1 in iteration k is feasible for (32) then also Step 2 in iteration k must be feasible (33). For feasibility, we only need to find a point (v_2, ϵ) , for which the constraint in Step 2 holds. From Step 1, we know that $f_1(v_1^{(k)}, v_2^{(k-1)})$ is an SOS polynomial. But then, $(v_2, \epsilon) = (v_2^{(k-1)}, 0)$ is a feasible point of (33). As a consequence, ϵ must be equal or smaller than zero, i.e. $\epsilon \leq 0$.

Next, we show that the solution of iteration k is a feasible point of Step 1 in iteration $k+1$. First, note that $-s(v_1^{(k)})\epsilon$ is an SOS polynomial vector since $\epsilon \leq 0$. But then also $f_1(v_1^{(k)}, v_2^{(k)})$ is an SOS polynomial. Therefore, $(v_1^{(k)}, v_2^{(k)})$ is a feasible point of Step 1 in iteration k . Taken together, we showed that $(v_1^{(k)}, v_2^{(k)})$ remain feasible for (31). Therefore, $f_0(v_1^{(k+1)})$ is either equal or smaller than $f_0(v_1^{(k)})$. ■

6.2 Main algorithm

The main algorithm improves feasibility by introducing an operating region \mathcal{X}_{op} . Under Assumption 2, there exists a scalar polynomial $r(x)$ whose zero sublevel set is compact and strictly contains the allowable set \mathcal{X}_a , i.e.

$$\mathcal{X}_s \subseteq \mathcal{X}_a \subseteq \mathcal{X}_{op} := \{x \in \mathbb{R}^n \mid r(x) \leq 0\}. \quad (34)$$

Note that all state trajectories of interest are contained in the operating region \mathcal{X}_{op} . We will use this fact to slightly alter the empty set conditions and the corresponding SOS constraints developed in the previous chapters.

Given an allowable set \mathcal{X}_a defined by $w_i(x)$ for $i \in [t]$ (22), positive definite SOS polynomials $l(x) \in \Sigma[x]$, a positive scalar $\epsilon_{s1} > 0$, center points $(x_{c,i}, B_{c,i}) \in (\mathbb{R}^n, \mathbb{R})$ for each $i \in [t]$, a region of operation defined by $r(x)$ (34) and initial values $s_1(x)$, $p(x)$, and $p_{m+1,i}(x)$ for $i \in [t]$, the main SOS algorithm is summarized below.

- Step 1: Given a controller $u = p/s_1$ find a CLF and multiple CBFs.

$$\begin{aligned} &\text{find} && V, B_i, s_2, s_{3,i}, s_{4,i} \in R[x] \\ &\text{hold fixed} && s_1, p_{m+1,i} \in R[x], p \in (\mathbb{R}[x])^m \\ &\text{minimize} && \sum_{i=1}^t \text{tr}(Q_{B,i}) + (B_i(x_{c,i}) - B_{c,i})^2 \\ &\text{subject to} && -\nabla V^T (s_1 f + Gp) - l^2 + s_2 r \in \Sigma[x] \\ &&& -\nabla B_i^T (s_1 f + Gp) - p_{m+1,i} B_i \\ &&& \quad + s_{3,i} r \in \Sigma[x] \quad \forall i \in [t] \\ &&& B_i - s_{4,i} w_i \in \Sigma[x] \quad \forall i \in [t] \\ &&& s_2 \in \Sigma[x] \\ &&& s_{3,i} \in \Sigma[x] \quad \forall i \in [t] \\ &&& s_{4,i} \in \Sigma[x] \quad \forall i \in [t] \\ &&& V \in \Sigma[x], V(x^*) = 0 \end{aligned}$$

- Step 2: Given a CLF and multiple CBFs find a controller $u = p/s_1$.

$$\begin{aligned} &\text{find} && \epsilon_i \in \mathbb{R}, s_1, s_2, s_{3,i}, p_{m+1,i} \in R[x], \\ &&& p \in (\mathbb{R}[x])^m \\ &\text{hold fixed} && V, B_i \in R[x] \\ &\text{minimize} && \sum_{i=1}^t \epsilon_i \\ &\text{subject to} && -\nabla V^T (s_1 f + Gp) - l^2 + s_2 r \in \Sigma[x] \\ &&& -\nabla B_i^T (s_1 f + Gp) - p_{m+1,i} B_i \\ &&& \quad + s_{3,i} r + \epsilon_i \in \Sigma[x] \quad \forall i \in [t] \\ &&& s_1 - \epsilon_{s1} \in \Sigma[x] \\ &&& s_2 \in \Sigma[x] \\ &&& s_{3,i} \in \Sigma[x] \quad \forall i \in [t] \end{aligned}$$

7. SIMULATION

The alternating algorithm from Section 6.2 is implemented for a three-dimensional vector field realization of a dc/ac power converter model. A feedback controller that avoids unsafe states is of paramount importance for this application since all electrical variables (e.g. voltages and currents) need to be constrained at all times. With the dc voltage and ac currents as states, the control system (1) is defined by

$$f(x) = \begin{bmatrix} -0.05x_1 - 57.9x_2 + 0.00919x_3 \\ 1710x_1 + 314x_3 \\ -0.271x_1 - 314x_2 \end{bmatrix} \quad (35)$$

and

$$G(x) = \begin{bmatrix} 0.05 - 57.9x_2 & -57.9x_3 \\ 1710 + 1710x_1 & 0 \\ 0 & 1710 + 1710x_1 \end{bmatrix}. \quad (36)$$

The allowable set \mathcal{X}_a is defined by two polynomials $w_1(x)$ and $w_2(x)$ encoding the state constraints for current and voltage respectively:

$$\begin{aligned} w_1(x) &= (x_1 + 0.3)^2 + (x_2/20)^2 + (x_3/20)^2 - 0.5^2 \\ w_2(x) &= (x_1/20)^2 + x_2^2 + x_3^2 - 1.2^2. \end{aligned} \quad (37)$$

An operating region \mathcal{X}_{op} in (34) can be parameterized by:

$$r(x) = (x_1/0.8)^2 + (x_2/1.2)^2 + (x_3/1.2)^2 - 1.8. \quad (38)$$

A $r(x)$ that tighter fits the allowable set results in a faster convergence of the alternating algorithm. For illustration purposes, however, we selected a suboptimal operation region. The center points of the CBFs are given by: $x_{c,i} = [-0.3 \ 0 \ 0]^T$ and $B_{c,i} = -10$ for $i \in [2]$. The degrees of the polynomials involved in the algorithm are summarized as follows:

$V(x)$	$B_i(x)$	$s_1(x)$	$s_2(x)$	$s_{3,i}(x)$	$s_{4,i}(x)$	$p(x)$	$p_{m+1,i}(x)$
4	4	2	6	6	3	4	2

For $V(x)$ and $B_i(x)$, the odd degree coefficients are set to zero:

- $\text{deg}(V(x)) = (2, 4)$
- $\text{deg}(B_1(x - x_{c,1})) = (0, 2, 4)$
- $\text{deg}(B_2(x - x_{c,2})) = (0, 2, 4)$.

Figure 7 illustrates the sequence of safe sets calculated at each iteration of the algorithm. The computed CLF and CBFs not only prove that there exists a stabilizing feedback controller but also guarantee that such controllers

renders the system safe w.r.t. a safe set \mathcal{X}_s . By using two CBFs, the algorithm terminates with a safe set that tightly fits the allowable set \mathcal{X}_a .

The number of variables and constraints involved in the resulting SDP are summarized as follows:

	Step 1	Step 2
Variables	337	372
Constraints	5188	4892

The algorithm was completed in 90 seconds. For each iteration, the SDP algorithm took on average 33 inner iterations to solve step 1 and on average 21 inner iterations to solve step 2.

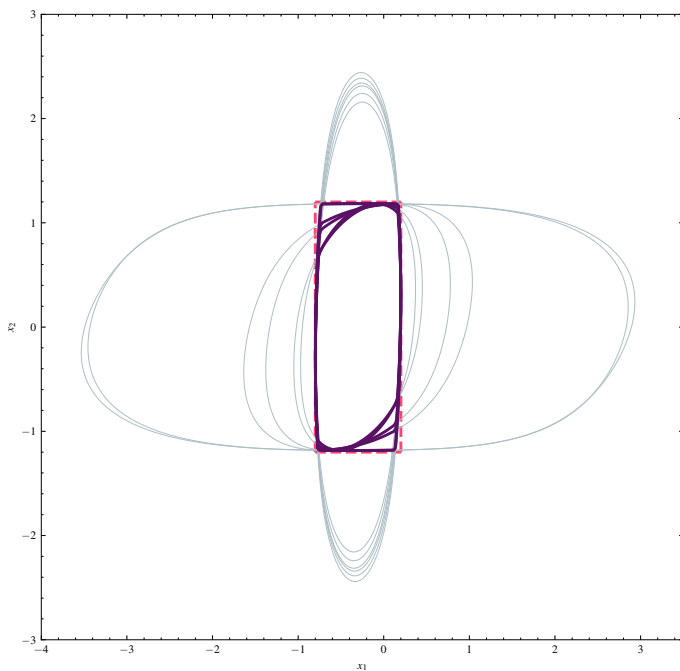


Fig. 1. shows the evolution of the sequence of safe sets \mathcal{X}_s projected to the x_1 and x_2 coordinates over 6 iterations. The boundary of the allowable set \mathcal{X}_a is shown in red. The sequence of zero-level sets of the CBF $B_1(x)$ and $B_2(x)$ are indicated in gray.

8. CONCLUSIONS AND FUTURE WORK

This paper presented a framework to combine stability and safety conditions using CLF and multiple CBFs. By employing two versions of the Positivstellensatz, we synthesized a controller used to prove compatibility between CLF and CBFs. We then formalized SOS constraints that encode compatible CLF and CBF conditions. Finally, we proposed an algorithm that solves the resulting SOS program by iteratively solving two SDPs.

For future work, we plan to study the computational complexity of our algorithm and also to explore a unified framework, that proves stability and safety with weaker CLF and CBF conditions. Moreover, we intend to incorporate noise-robustness into our SOS formulation, as demonstrated by Kang et al. (2023).

REFERENCES

- Ames, A.D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P. (2019). Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, 3420–3431. IEEE.
- Anghel, M., Milano, F., and Papachristodoulou, A. (2013). Algorithmic construction of Lyapunov functions for power system stability analysis. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(9), 2533–2546.
- Blanchini, F. (1999). Set invariance in control. *Automatica*, 35(11), 1747–1767.
- Bochnak, J., Coste, M., and Roy, M.F. (2013). *Real algebraic geometry*, volume 36. Springer Science & Business Media.
- Clark, A. (2021). Verification and synthesis of control barrier functions. In *2021 60th IEEE Conference on Decision and Control (CDC)*, 6105–6112. IEEE.
- Grammatico, S., Blanchini, F., and Caiti, A. (2013). Control-sharing and merging control Lyapunov functions. *IEEE Transactions on Automatic Control*, 59(1), 107–119.
- Isaly, A., Ghanbarpour, M., Sanfelice, R.G., and Dixon, W.E. (2022). On the feasibility and continuity of feedback controllers defined by multiple control barrier functions for constrained differential inclusions. In *2022 American Control Conference (ACC)*, 5160–5165. IEEE.
- Isidori, A. (1995). *Nonlinear Control Systems*. Springer.
- Kang, S., Chen, Y., Yang, H., and Pavone, M. (2023). Verification and synthesis of robust control barrier functions: Multilevel polynomial optimization and semidefinite relaxation. *arXiv preprint arXiv:2303.10081*.
- Khalil, H. (2002). *Nonlinear Systems*. Pearson Education. Prentice Hall.
- Korda, M., Henrion, D., and Jones, C.N. (2014). Convex computation of the maximum controlled invariant set for polynomial control systems. *SIAM Journal on Control and Optimization*, 52(5), 2944–2969.
- Kundu, S., Geng, S., Nandanoori, S., Hiskens, I.A., and Kalsi, K. (2019). Distributed barrier certificates for safe operation of inverter-based microgrids. 1042–1047.
- Laurent, M. (2009). Sums of squares, moment matrices and optimization over polynomials. *Emerging applications of algebraic geometry*, 157–270.
- Tan, W. and Packard, A. (2004). Searching for control Lyapunov functions using sums of squares programming. *sibi*, 1(1).
- Tan, X. and Dimarogonas, D.V. (2022). Compatibility checking of multiple control barrier functions for input constrained systems. *arXiv preprint arXiv:2209.02284*.
- Wang, H., Margellos, K., and Papachristodoulou, A. (2022). Safety verification and controller synthesis for systems with input constraints. *arXiv preprint arXiv:2204.09386*.
- Wieland, P. and Allgöwer, F. (2007). Constructive safety using control barrier functions. *IFAC Proceedings Volumes*, 40(12), 462–467.
- Xu, X. (2016). Control sharing barrier functions with application to constrained control. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, 4880–4885. IEEE.