Compositional Safety Analysis using Barrier Certificates [*]

Christoffer Sloth
Department of Computer Science
Aalborg University
9220 Aalborg East, Denmark
csloth@cs.aau.dk

George J. Pappas
Department of Electrical and Systems Engineering
University of Pennsylvania
Philadelphia, PA 19104 USA
pappasg@seas.upenn.edu

Rafael Wisniewski
Section for Automation & Control
Aalborg University
9220 Aalborg East, Denmark
raf@es.aau.dk

ABSTRACT

This paper proposes a compositional method for verifying the safety of a dynamical system, given as an interconnection of subsystems. The safety verification is conducted by the use of the barrier certificate method; hence, the contribution of this paper is to show how to obtain compositional conditions for safety verification.

We show how to formulate the verification problem, as a composition of coupled subproblems, each given for one subsystem. Furthermore, we show how to find the compositional barrier certificates via linear and sum of squares programming problems.

The proposed method makes it possible to verify the safety of higher dimensional systems, than the method for centrally computed barrier certificates. This is demonstrated by verifying the safety of an emergency shutdown of a wind turbine.

Categories and Subject Descriptors

I.6.4 [**Simulation and Modeling**]: Model Validation and Analysis

General Terms

Verification,Theory

Keywords

Compositionality, Safety analysis, Dynamical systems, Reachable sets, Sum of squares

1. INTRODUCTION

Safety verification is an important part of developing a control system. Safety verification ensures that a control system does not violate any state constraints. Numerous methods have been developed for verifying the safety of a system; see [5] for a review. These methods range over analytical methods, numerical simulation-based methods, and discrete abstraction methods.

The safety verification determines if the reachable states of a system intersect a set of unsafe states. Computing the reachable states of a dynamical system is in general very difficult, as seen in [9]; hence, it may only be possible for systems of low dimension. Therefore, several methods have been developed to approximate the reachable set of a dynamical system. In [4], the reachable states are approximated based on simulated trajectories, by exploiting that trajectories initialized close to each other stay in the proximity of each other.

Another class of methods, e.g., [1] verifies the safety of a system, by using the vector field to find invariant sets that do not include the unsafe states. Similarly, the papers [13, 12], provide a method for calculating barrier certificates for safety analysis of continuous, stochastic, and hybrid systems. The idea of these works is to find a barrier function that is decreasing along system trajectories, and has a zero level set (a so called barrier), which no solution trajectory crosses. If the set of initial states is a subset of the zero sublevel set of the barrier function, and the set of unsafe states is in its complement, then the system is safe.

Common to the previously mentioned methods is that they verify the safety of a system, by studying a system directly. However, it may be beneficial to study a system as an interconnection of subsystems, and decompose the verification problem into smaller subproblems. This is suggested for compositional stability analysis in [16].

In this paper, we show how the barrier certificates in [13, 12] can be generated for a system, given as an interconnection of subsystems. Compositional conditions are given for finding barrier certificates. Additionally, linear matrix inequalities (LMIs) and sum of squares (SOS) are used to generate the barrier certificates, which are solved numerically, by use of SOSTOOLS for MATLAB [14].

The paper is organized as follows. Section 2 explains the verification problem in terms of barrier certificates, and Section 3 explains how to reformulate the verification problem by a composition of certificates generated individually for each subsystem. Section 4 shows how to compute the barrier certificates, both via LMIs and polynomial inequalities. Section 5 demonstrates the use of the method, by proving safety of a shutdown procedure for a wind turbine, and Section 6 comprises conclusions.

[*]This work was supported by MT-LAB, a VKR Centre of Excellence for the Modeling of Information Technology.

HSCC'12, April 17–19, 2012, Beijing, China.

2. SAFETY VERIFICATION USING BARRIER CERTIFICATES

In this section, we present the barrier certificate method, which can be used to verify the safety of a dynamical system.

We consider a continuous system given as a system of ordinary differential equations

$$\dot{x} = f(x, d), \tag{1}$$

where $x \in \mathbb{R}^n$ is the state and $d \in D \subseteq \mathbb{R}^m$ is the disturbance input.

For some measurable and essentially bounded disturbance function $\bar{d} : \mathbb{R}_{\geq 0} \to D$, i.e., $\bar{d} \in \mathcal{L}_\infty(\mathbb{R}_{\geq 0}, D)$, we denote the solution of the Cauchy problem (1) with $x(0) = x_0$ on an interval $[0, T]$ by $\phi_{x_0}^{\bar{d}}$, i.e.,

$$\frac{d\phi_{x_0}^{\bar{d}}(t)}{dt} = f\left(\phi_{x_0}^{\bar{d}}(t), \bar{d}(t)\right) \tag{2}$$

for almost all $t \in [0, T]$.

We consider a system given by $\Gamma = (f, X, X_0, X_\mathrm{u}, D)$, where $f : \mathbb{R}^{n+m} \to \mathbb{R}^n$ is continuous, $X \subseteq \mathbb{R}^n$, $X_0 \subseteq X$, $X_\mathrm{u} \subseteq X$, and $D \subseteq \mathbb{R}^m$. In the safety verification, we only consider trajectories initialized in X_0 that are contained in the set X. We verify if there exists a trajectory that can reach an unsafe set X_u.

For a map $f : A \to B$ and subset $C \subset A$, we write $f(C) \equiv \{f(x) | \; x \in C\}$. Thus, the safety of a system Γ is defined as follows.

Definition 1 (Safety). *Let $\Gamma = (f, X, X_0, X_\mathrm{u}, D)$ be given. A trajectory $\phi_{X_0}^{\bar{d}} : [0, T] \to \mathbb{R}^n$ is unsafe if there exists a time $t \in [0, T]$ and a disturbance $\bar{d} \in \mathcal{L}_\infty(\mathbb{R}_{\geq 0}, D)$, such that $\phi_{X_0}^{\bar{d}}([0, t]) \cap X_\mathrm{u} \neq \emptyset$ and $\phi_{X_0}^{\bar{d}}([0, t]) \subseteq X$.*

We say that a system Γ is safe if there are no unsafe trajectories.

For a function $f : \mathbb{R}^n \to \mathbb{R}$, $\mathcal{Z}(f)$ denotes the set

$$\mathcal{Z}(f) = \{x \in \mathbb{R}^n | f(x) = 0\}. \tag{3}$$

The safety property can be verified using the following.

Proposition 1 (Strict barrier certificate [13]). *Let $\Gamma = (f, X, X_0, X_\mathrm{u}, D)$ be given. If there exists a differentiable function $B : X \to \mathbb{R}$ satisfying*

$$B(x) \leq 0 \quad \forall x \in X_0, \tag{4a}$$

$$B(x) > 0 \quad \forall x \in X_\mathrm{u}, \text{ and} \tag{4b}$$

$$\frac{\partial B}{\partial x}(x) f(x, d) < 0 \quad \forall (x, d) \in \mathcal{Z}(B) \times D. \tag{4c}$$

Then the system Γ is safe.

Proposition 1 states that a trajectory initialized within the zero sublevel set of a function B, cannot cross the zero level set $\mathcal{Z}(B)$, if B is decreasing (along system trajectories) on the zero level set. This is illustrated in Figure 1.

The set of barrier certificates satisfying Proposition 1 is nonconvex, due to (4c). However, the following more conservative proposition has a convex set of feasible barrier certificates. The convexity property becomes apparent in the computation of the barrier certificates in Section 4.

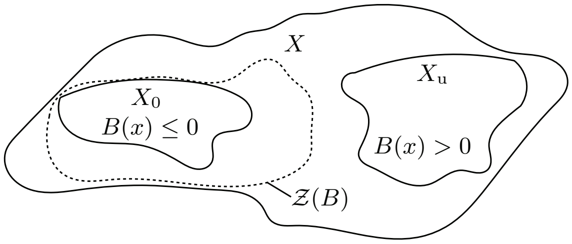

Figure 1: Illustration of a set X, which contains the initial set X_0 and the unsafe set X_u. The dashed line illustrates the zero level set of B.

Corollary 1 (Weak barrier certificate [13, 12]). *Let $\Gamma = (f, X, X_0, X_\mathrm{u}, D)$ be given. If there exists a differentiable function $B : X \to \mathbb{R}$ satisfying*

$$B(x) \leq 0 \quad \forall x \in X_0, \tag{5a}$$

$$B(x) > 0 \quad \forall x \in X_\mathrm{u}, \text{ and} \tag{5b}$$

$$\frac{\partial B}{\partial x}(x) f(x, d) \leq 0 \quad \forall (x, d) \in X \times D. \tag{5c}$$

Then the system Γ is safe.

Corollary 1 states that a trajectory of a system initialized in a state within the zero sublevel set of a nonincreasing function (along system trajectories), cannot reach the complement of the zero sublevel set.

The difference between Proposition 1 and Corollary 1 is that (5c), in contrast to (4c), must hold for all states and all disturbances. Additionally, the inequality constraint (4c) is strict weathers it is weak in (5c).

3. COMPOSITIONAL BARRIER CERTIFICATES

In this section, we assume that a dynamical system is given as an interconnection of subsystems. This allows the safety verification to be split up into smaller subproblems in addition to some coupling constraints.

To provide an overview of the proposed compositional setup, we initially consider an example from [16], consisting of three interconnected subsystems. The interconnection of the three subsystems is shown in Figure 2. Properties of the interconnected system are to be analyzed by studying its components as isolated systems, in conjunction with their coupling.

Let each subsystem be described by a system of continuous ordinary differential equations and an output map

$$\Sigma_1 : \begin{cases} \dot{x}_1 = f_1(x_1, d_1, u_1) \\ y_1 = h_1(x_1) \end{cases} \tag{6a}$$

$$\Sigma_2 : \begin{cases} \dot{x}_2 = f_2(x_2, d_2, u_2) \\ y_2 = h_2(x_2) \end{cases} \tag{6b}$$

$$\Sigma_3 : \begin{cases} \dot{x}_3 = f_3(x_3, d_3, u_3) \\ y_3 = h_3(x_3), \end{cases} \tag{6c}$$

where $x_i \in X_i \subseteq \mathbb{R}^{n_i}$ is the state, $d_i \in D_i \subseteq \mathbb{R}^{m_i}$ is the disturbance, and $u_i \in \mathbb{R}^{q_i}$ is an interconnection input, given by $u_i = g_i(x_1, \ldots, \hat{x}_i, \ldots, x_k)$. Here, $\hat{x}_i$ indicates that x_i is removed. Additionally, $y_i \in \mathbb{R}^{r_i}$ is an interconnection

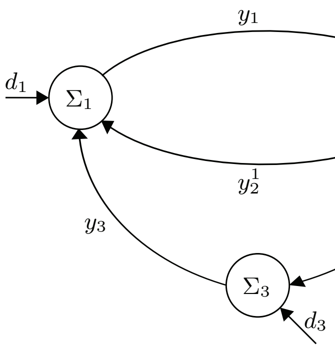

Figure 2: Interconnection of three subsystems $\Sigma_1, \Sigma_2, \Sigma_3$.

output, given by the map $h_i : \mathbb{R}^{n_i} \to \mathbb{R}^{r_i}$. Note that the interconnection of the subsystems gives a relation between u_i and y_i. In Figure 2, $y_2 = (y_2^1, y_2^2)$, $u_1 = (y_2^1, y_3)$, $u_2 = y_1$, and $u_3 = y_2^2$.

The output y_i belongs to the set

$$Y_i \equiv h_i(X_i) \subseteq \mathbb{R}^{r_i}. \tag{7a}$$

Similarly, u_i belongs to the set

$$U_i \equiv g_i(X_1, \ldots, \hat{X}_i, \ldots, X_k) \subseteq \mathbb{R}^{q_i}. \tag{7b}$$

In the remainder of the paper, we present a method for generating barrier certificates for some general topology of the interconnection of subsystems.

Let $k \in \mathbb{N}$ be the number of subsystems. For $i = 1, \ldots, k$ we consider a system $\Gamma = (\{f_i\}, \{X_i\}, \{X_{0,i}\}, \{X_{\mathrm{u},i}\}, \{D_i\})$, where $\{f_i\}$ is a collection of continuous vector fields with $f_i : \mathbb{R}^{n_i+m_i+q_i} \to \mathbb{R}^{n_i}$, $X_i \subseteq \mathbb{R}^{n_i}$, $X_{0,i}, X_{\mathrm{u},i} \subseteq X_i$, and $D_i \subseteq \mathbb{R}^{m_i}$. Let

$$X = X_1 \times \cdots \times X_k \subseteq \mathbb{R}^n, \tag{8a}$$
$$X_0 = X_{0,1} \times \cdots \times X_{0,k} \subseteq X, \tag{8b}$$
$$X_\mathrm{u} = X_{\mathrm{u},1} \times \cdots \times X_{\mathrm{u},k} \subseteq X, \tag{8c}$$
$$D = D_1 \times \cdots \times D_k \subseteq \mathbb{R}^m, \tag{8d}$$
$$U = U_1 \times \cdots \times U_k \subseteq \mathbb{R}^q, \text{and} \tag{8e}$$
$$Y = Y_1 \times \cdots \times Y_k \subseteq \mathbb{R}^r. \tag{8f}$$

Remark 1. *The assumption that the sets X and D are given as cartesian products of X_i and D_i in (8), limits the sets that can be directly expressed; however, by using multiple sets, the original set can, in principle, be approximated. Therefore, the previous restriction does not theoretically restrict the method, but it may complicate the computations involved in the safety verification.*

In the following, we present two lemmas that show how to compose the inequality constraints on the barrier function and its derivative in Proposition 1 into separate constraints for the subsystems and coupling constraints. We omit the proofs of both lemmas, as they are straightforward.

In Lemma 1, we let the vector field be given as an interconnection of subsystems, and show that (4c) can be composed into an inequality constraint for each subsystem, and a coupling constraint.

Lemma 1. *Let $k \in \mathbb{N}$. Let $x = (x_1, \ldots, x_k) \in X$, $d = (d_1, \ldots, d_k) \in D$, $u = (u_1, \ldots, u_k) \in U$, $y = (y_1, \ldots, y_k) \in Y$, where X, D, U, Y are given as shown in (8). For $i = 1, \ldots, k$ let*

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_k \end{bmatrix} = \begin{bmatrix} f_1(x_1, d_1, u_1) \\ \vdots \\ f_k(x_k, d_k, u_k) \end{bmatrix} = f(x, d), \tag{9a}$$
$$u_i = g_i(x_1, \ldots, \hat{x}_i, \ldots, x_k), \tag{9b}$$
$$y_i = h_i(x_i). \tag{9c}$$

Suppose that there is a bijective map $\Upsilon : U \to Y$.

Then there exists a continuous function $\varphi : \mathbb{R}^n \to \mathbb{R}$ such that

$$\varphi(x) f(x, d) < 0 \quad \forall (x, d) \in X \times D \tag{10}$$

if for $i = 1, \ldots, k$ there exist continuous functions $\varphi_i : \mathbb{R}^{n_i} \to \mathbb{R}$ and $\gamma_i : \mathbb{R}^{q_i+r_i} \to \mathbb{R}$ such that for all $(x_i, d_i, u_i) \in X_i \times D_i \times U_i$

$$\varphi_i(x_i) f_i(x_i, d_i, u_i) < \gamma_i(u_i, h(x_i)) \text{ and} \tag{11a}$$
$$\sum_i \gamma_i(u_i, h(x_i)) \leq 0. \tag{11b}$$

Lemma 1 can be used to decompose (4c) into an inequality constraint for each subsystem in addition to a coupling constraint.

Lemma 2. *Let $k \in \mathbb{N}$. For $i = 1, \ldots, k$ let $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$ be a continuous function, and $X_i \subseteq \mathbb{R}^{n_i}$ be compact. There exists a constant $c_i \in \mathbb{R}$ for all i such that*

$$f_i(x_i) - c_i \leq 0 \quad \forall x_i \in X_i \text{ and} \tag{12a}$$
$$\sum_i c_i \leq 0 \quad \forall x_i \in X_i \tag{12b}$$

if and only if

$$\sum_i f_i(x_i) \leq 0 \quad \forall x_i \in X_i. \tag{13}$$

Using Lemma 1 and Lemma 2, we rewrite Proposition 1 as follows.

Proposition 2. *Let $k \in \mathbb{N}$ and let the dynamical system $\Gamma = (\{f_i\}, \{X_i\}, \{X_{0,i}\}, \{X_{\mathrm{u},i}\}, \{D_i\})$ be given. If there exist differentiable functions $B_i : X_i \to \mathbb{R}$, constants $\alpha_i, \beta_i \in \mathbb{R}$, and functions $\gamma_i : \mathbb{R}^{q_i+r_i} \to \mathbb{R}$ for $i = 1, \ldots, k$ such that*

$$B_i(x_i) + \alpha_i \leq 0 \quad \forall x_i \in X_{0,i}, \tag{14a}$$
$$B_i(x_i) - \beta_i > 0 \quad \forall x_i \in X_{\mathrm{u},i}, \tag{14b}$$
$$\frac{\partial B_i}{\partial x_i}(x_i) f_i(x_i, d_i, u_i) < \gamma_i(u_i, h_i(x_i)) \quad \text{for all } u_i \in U_i,\ x_i \in \mathcal{Z}(B_i),\ d_i \in D_i, \tag{14c}$$

and for all $u_i \in U_i$, $x_i \in \mathcal{Z}(B_i)$

$$\sum_i \alpha_i \geq 0, \sum_i \beta_i \geq 0, \sum_i \gamma_i(u_i, h_i(x_i)) \leq 0. \tag{14d}$$

Then the system Γ is safe.

Proof. We show that Proposition 2 ensures that the conditions in Proposition 1 are satisfied. Let $x \equiv (x_1, \ldots, x_k)^\mathrm{T}$ and $B : \mathbb{R}^{n_1+\cdots+n_k} \to \mathbb{R}$ be defined as $B(x) = \sum_i B_i(x_i)$.

By Lemma 2, (14a) and (14b) are by the satisfaction of (14d) equivalent to

$$B(x) \leq 0 \quad \forall x \in X_0, \tag{15a}$$

$$B(x) > 0 \quad \forall x \in X_{\mathrm{u}}. \tag{15b}$$

Finally, by (14c) and (14d)

$$\sum_i \frac{\partial B_i}{\partial x_i}(x_i) f(x_i, d_i, u_i) < \sum_i \gamma_i(u_i, h_i(x_i)) \leq 0 \\ \textit{for all } u_i \in U_i,\ x_i \in \mathcal{Z}(B_i),\ d_i \in D_i. \tag{15c}$$

This is by Lemma 1 equivalent to (4c). Hereby, the system Γ is safe. □

The inequality constraints (14a)-(14c) must be satisfied for each subsystem, and (14d) couples the subproblems. Notice that the function B is decreasing along the solution, but each function B_i is not necessarily decreasing along the solution.

In the following, we rewrite Corollary 1 using the same technique.

Corollary 2. *Let $k \in \mathbb{N}$ and let the dynamical system $\Gamma = (\{f_i\}, \{X_i\}, \{X_{0,i}\}, \{X_{\mathrm{u},i}\}, \{D_i\})$ be given. If there exist differentiable functions $B_i : X_i \to \mathbb{R}$, constants $\alpha_i, \beta_i \in \mathbb{R}$, and functions $\gamma_i : \mathbb{R}^{q_i + r_i} \to \mathbb{R}$ for $i = 1, \ldots, k$ such that*

$$B_i(x_i) + \alpha_i \leq 0 \quad \forall x_i \in X_{0,i}, \tag{16a}$$

$$B_i(x_i) - \beta_i > 0 \quad \forall x_i \in X_{\mathrm{u},i}, \tag{16b}$$

$$\frac{\partial B_i}{\partial x_i}(x_i) f_i(x_i, d_i, u_i) \leq \gamma_i(u_i, h_i(x_i)) \\ \textit{for all } u_i \in U_i,\ x_i \in X_i,\ d_i \in D_i, \tag{16c}$$

and for all $u_i \in U_i$, $x_i \in X_i$

$$\sum_i \alpha_i \geq 0, \sum_i \beta_i \geq 0, \sum_i \gamma_i(u_i, h_i(x_i)) \leq 0. \tag{16d}$$

Then the system Γ is safe.

Proposition 2 and Corollary 2 provide compositional conditions for the safety verification. In the next section, we show how to compute the barrier certificates.

4. COMPUTATION OF BARRIER CERTIFICATES

In this section, we show how to compute barrier certificates from the conditions set up in Section 2 and Section 3.

Remark that any desired computational method may be applied to find the barrier certificates, and that different methods can be applied on different subproblems for the compositional conditions in Section 3. This is beneficial if some subsystems are linear and others are polynomial.

To demonstrate the computation of barrier certificates, we show how to compute the barrier certificates using sum of squares programming and linear programming. The primary focus is on sum of squares programming, as it is a generalization of linear programming. Therefore, we only explicitly formulate LMI conditions for the solution of Corollary 2.

To do the computations in a tool such as MATLAB, we restrict the vector fields to be linear (for linear programs) and polynomial (for sum of squares programs). Furthermore, we parameterize the barrier certificates as polynomials, respectively quadratic forms, and require the invariant, initial, unsafe, and disturbance sets to be given by linear and polynomial equality or inequality constraints.

First, we set up some notation about polynomials.

Definition 2 (Polynomial [10]). *A polynomial p in n variables $x_1, \ldots, x_n$ is a finite linear combination of monomials*

$$p(x) = \sum_\alpha c_\alpha x^\alpha = \sum_\alpha c_\alpha x_1^{\alpha_1} \cdot \ldots \cdot x_n^{\alpha_n}, \tag{17}$$

where $c_\alpha \in \mathbb{R}$ and the sum is over a finite number of n-tuples $\alpha = [\alpha_1, \ldots, \alpha_n]$ with $\alpha_i \geq 0$.

The total degree of a monomial x^α is $\alpha_1 + \cdots + \alpha_n$. Additionally, the total degree of a polynomial is equal to the highest degree of its component monomials. The degree of a polynomial p is denoted by $\deg(p)$.

We only consider polynomials with real valued variables, and denote the set of polynomials in n variables by $\mathcal{P}_n$. Recall that a map $f : \mathbb{R}^n \to \mathbb{R}^m$ is said to be polynomial if its coordinate functions are polynomials, i.e., $f_i \in \mathcal{P}_n$ for $i = 1, \ldots, m$; hence, $f \in \mathcal{P}_n^m$.

Sum of squares polynomials are used in the generation of safety certificates and are explained in the following, based on [10].

Definition 3. *A polynomial $p \in \mathcal{P}_n$ is called sum of squares (SOS) if*

$$p = \sum_{i=1}^k p_i^2 \tag{18}$$

for some polynomials $p_i \in \mathcal{P}_n$ with $i = 1, \ldots, k$.

We denote the set of sum of squares polynomials in n variables by Σ_n.

The set of sum of squares polynomials is a subset of nonnegative polynomials [10], which can be treated using semidefinite programming, as described below.

The existence of a sum of squares decomposition of a polynomial $p \in \mathcal{P}_n$, with $d = \deg(p)$, can be expressed as a semidefinite programming feasibility problem. Therefore, the formulation of a problem as sum of squares makes the problem computationally tractable; however, the number of decision variables in the program is

$$N = \binom{n + 2d}{2d} = \frac{(n + 2d)!}{2d!n!}. \tag{19}$$

In the search for sum of squares polynomials, it is exploited that the existence of a SOS decomposition of a polynomial p is equivalent to the existence of a positive semidefinite matrix $Q = Q^{\mathrm{T}} \geq 0$ such that

$$p = Z^{\mathrm{T}} Q Z, \tag{20}$$

where Z is a vector of monomials of degree less than or equal to half the degree of p.

Let $k, l \in \mathbb{N}$, let $\alpha_{i,j} \in \mathcal{P}_n$ for $(i, j) \in \{1, \ldots, l\} \times \{1, \ldots, k\}$, and $w_j \in \mathbb{R}$. An SOS programming problem is

$$\underset{(c_1, \ldots, c_k) \in \mathbb{R}^k}{\text{minimize}} \sum_{j=1}^k w_j c_j \text{ subject to} \tag{21a}$$

$$\alpha_{i,0} + \sum_{j=1}^k \alpha_{i,j} c_j \text{ is SOS } \forall i = 1, \ldots, l. \tag{21b}$$

It is seen that an SOS programming problem is a minimization of a linear cost, subject to SOS feasibility constraints.

4.1 Computation of Barrier Certificates

To compute barrier certificates using sum of squares programming, we restrict the vector fields to be polynomial. Furthermore, the invariant, initial, unsafe, and disturbance sets must be semialgebraic sets, i.e., be given by polynomial inequalities, as follows.

Let $g_X : \mathbb{R}^n \to \mathbb{R}^{k_X}$, $g_{X_0} : \mathbb{R}^n \to \mathbb{R}^{k_{X_0}}$, $g_{X_\mathrm{u}} : \mathbb{R}^n \to \mathbb{R}^{k_{X_\mathrm{u}}}$, and $g_D : \mathbb{R}^m \to \mathbb{R}^{k_D}$ for some $k_X, k_{X_0}, k_{X_\mathrm{u}}, k_D \in \mathbb{N}$ be given as vectors of polynomials $g_i \in \mathcal{P}_n$, i.e., for example $g_X \in \mathcal{P}_n^{k_X}$ and $g_X = [g_1, \dots, g_{k_X}]^\mathrm{T}$. Then

$$X \equiv \{x \in \mathbb{R}^n | g_X(x) \geq 0\}, \tag{22a}$$

$$X_0 \equiv \{x \in \mathbb{R}^n | g_{X_0}(x) \geq 0\}, \tag{22b}$$

$$X_\mathrm{u} \equiv \{x \in \mathbb{R}^n | g_{X_\mathrm{u}}(x) \geq 0\}, \tag{22c}$$

$$D \equiv \{d \in \mathbb{R}^m | g_D(d) \geq 0\}, \tag{22d}$$

where the inequalities in (22) are satisfied *entry-wise*.

Example 1. *We show how* (22) *can be used to form a cylindrical set. Let* $x \in \mathbb{R}^3$, $x_{1,\min}, x_{1,\max}, x_{2,\mathrm{c}}, x_{3,\mathrm{c}}, r \in \mathbb{R}$ *and* g_X *be*

$$g_X(x) = \begin{bmatrix} (x_1 - x_{1,\min})(x_{1,\max} - x_1) \\ r^2 - (x_2 - x_{2,\mathrm{c}})^2 - (x_3 - x_{3,\mathrm{c}})^2 \end{bmatrix}. \tag{23}$$

It is seen that $g_X(x) \geq 0$, *when* $x_1 \in [x_{1,\min}, x_{1,\max}]$ *and* (x_2, x_3) *is in the disk centered at* $(x_{2,\mathrm{c}}, x_{3,\mathrm{c}})$ *with radius* r. *This implies that the set* X *given by* (22a) *and* (23) *is a cylinder.*

In the computation of barrier certificates, we use a generalization of the $\mathcal{S}$-procedure [2], which is shown in Lemma 3.

Lemma 3. *Let* V *be a subset of* $X \subseteq \mathbb{R}^n$. *Let* $f \in \mathcal{P}_n$ *and* $g \in \mathcal{P}_n^k$. *Suppose* $g(x) \geq 0$ *(element-wise) for any* $x \in V$. *If*

1. $\lambda \in \Sigma_n^k$ *and*
2. $f - \lambda^\mathrm{T} g \in \Sigma_n$.

Then $f(x) \geq 0$ *for all* $x \in V$.

Now we can compute barrier certificates that satisfy Proposition 1 using sum of squares.

Proposition 3. *Let the system* $\Gamma = (f, X, X_0, X_\mathrm{u}, D)$ *and polynomials* g_* *shown in* (22) *be given, and let* $\epsilon_1, \epsilon_2 > 0$. *If there exist* $B \in \mathcal{P}_n$, $\lambda_{X_0} \in \Sigma_n^{k_{X_0}}$, $\lambda_{X_\mathrm{u}} \in \Sigma_n^{k_{X_\mathrm{u}}}$, $\lambda_B \in \mathcal{P}_{n+m}$, *and* $\lambda_D \in \Sigma_{n+m}^{k_D}$ *such that*

$$-B - \lambda_{X_0}^\mathrm{T} g_{X_0}, \tag{24a}$$

$$B - \epsilon_1 - \lambda_{X_\mathrm{u}}^\mathrm{T} g_{X_\mathrm{u}}, \text{ and} \tag{24b}$$

$$-\frac{\partial B}{\partial x} f - \epsilon_2 - \lambda_D^\mathrm{T} g_D - \lambda_B^\mathrm{T} B \tag{24c}$$

are sum of squares. Then the system Γ *is safe.*

As Proposition 3 follows directly from Proposition 20 in [13], no proof is provided. However, all conditions follow directly from Lemma 3. Consider (24a), where $B \in \mathcal{P}_n$, $g_{X_0} \in \mathcal{P}_n^{k_{X_0}}$, $\lambda_{X_0} \in \Sigma_n^{k_{X_0}}$, (24a)$\in \Sigma_n$, and $g_{X_0}(x) \geq 0$ for any $x \in X_0$. Then $B(x) \leq 0$ for all $x \in X_0$.

Note that (24c) contains a scalar product between λ_B and B, which are both unknown. This is the reason why the conditions in Proposition 3 cannot be found directly by an SOS programming problem, neither by a linear program for quadratic B. Therefore, we generate an iterative algorithm for solving the problem. This algorithm is similar to iterative algorithms used for solving bilinear matrix inequalities via LMIs, see [6].

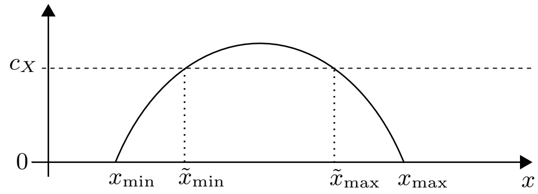

Figure 3: Illustration of g_X and the sets $X = [x_{\min}, x_{\max}]$ and $\tilde{X} = [\tilde{x}_{\min}, \tilde{x}_{\max}]$, given by g_X and $\tilde{g}_X$.

In the following iterative algorithm, it is necessary to get a feasible solution in each step. Therefore, the barrier certificate is initially found for only a subset of the disturbances $\tilde{D} \subseteq D$, initial conditions $\tilde{X}_0 \subseteq X_0$, etc., to ease the feasibility. Let $c_X \in \mathbb{R}_{\geq 0}^{k_X}$ be a vector of nonnegative numbers. Let $\tilde{g}_X = g_X - c_X$ and define

$$\tilde{X} \equiv \{x \in \mathbb{R}^n | \tilde{g}_X(x) \geq 0\} \subseteq X. \tag{25}$$

By decreasing each entry of c_X, the set $\tilde{X}$ is enlarged, and if $c_X = 0$ then $\tilde{X} = X$. This is illustrated in Figure 3 for a set given by $g_X(x) = (x - x_{\min})(x_{\max} - x)$.

It is seen that the map g_X generates the set $X = [x_{\min}, x_{\max}]$ and $\tilde{X} = [\tilde{x}_{\min}, \tilde{x}_{\max}]$. If c_X is greater than the maximum value of g_X, then $\tilde{X} = \emptyset$.

Algorithm 1. *Let the system* $\Gamma = (f, X, X_0, X_\mathrm{u}, D)$ *and polynomials* g_* *shown in* (22) *be given.*

0. ***Initialization**: Choose vectors* $c_{X_0} \in \mathbb{R}_{\geq 0}^{k_{X_0}}, c_{X_\mathrm{u}} \in \mathbb{R}_{\geq 0}^{k_{X_\mathrm{u}}}, c_D \in \mathbb{R}_{\geq 0}^{k_D}$ *such that each entry* $c_{i,*}$ *is sufficiently large and define polynomials* $\tilde{g}_* \equiv g_* - c_*$. *Choose* $\epsilon_1, \epsilon_2 > 0$ *and specify a polynomial* $\lambda_B \in \mathcal{P}_{n+m}$, *e.g., by choosing* $\lambda_B = 0$ *or* 1. *Find* $B \in \mathcal{P}_n$, $\lambda_{X_0} \in \Sigma_n^{k_{X_0}}$, $\lambda_{X_\mathrm{u}} \in \Sigma_n^{k_{X_\mathrm{u}}}$, *and* $\lambda_D \in \Sigma_{n+m}^{k_D}$ *such that*

   $$-B - \lambda_{X_0}^\mathrm{T} \tilde{g}_{X_0}, \tag{26a}$$

   $$B - \epsilon_1 - \lambda_{X_\mathrm{u}}^\mathrm{T} \tilde{g}_{X_\mathrm{u}}, \text{ and} \tag{26b}$$

   $$-\frac{\partial B}{\partial x} f - \epsilon_2 - \lambda_D^\mathrm{T} \tilde{g}_D - \lambda_B^\mathrm{T} B \tag{26c}$$

   are sum of squares.

1. ***Fix the barrier certificate**: Fix* B *obtained from the previous step. Choose vectors* $\Delta c_* \geq 0$ *and update* c_*, *such that* $c_* := c_* - \Delta c_*$ *and redefine the polynomials* $\tilde{g}_* \equiv g_* - c_*$. *Find* $\lambda_{X_0} \in \Sigma_n^{k_{X_0}}$, $\lambda_{X_\mathrm{u}} \in \Sigma_n^{k_{X_\mathrm{u}}}$, $\lambda_B \in \mathcal{P}_{n+m}$, *and* $\lambda_D \in \Sigma_{n+m}^{k_D}$ *such that* (26) *are sum of squares.*

2. ***Fix multiplier**: Fix* λ_B *obtained in the previous step. Choose vectors* $\Delta c_* \geq 0$ *and update* c_*, *such that* $c_* := c_* - \Delta c_*$ *and redefine the polynomials* $\tilde{g}_* \equiv g_* - c_*$. *Find* $B \in \mathcal{P}_n$, $\lambda_{X_0} \in \Sigma_n^{k_{X_0}}$, $\lambda_{X_\mathrm{u}} \in \Sigma_n^{k_{X_\mathrm{u}}}$, *and* $\lambda_D \in \Sigma_{n+m}^{k_D}$ *such that* (26) *are sum of squares.*

   If all entries of the vector c_* *are zero, then terminate the algorithm; otherwise, go to step 1.*

If Algorithm 1 terminates, then Γ is safe.

Algorithm 1 alternates between freezing the coefficients of λ_B and B, to remove the product between the two unknown polynomials in (26c). Furthermore, the set of disturbances and the sets for which B should be positive or negative are initially smaller than D, X_0 and X_u and are gradually enlarged until they are equal to X_0 and X_u. In the enlargement of the sets, it is important that a feasible solution is found in each step of the algorithm. Notice that Algorithm 1 is not guaranteed to terminate or converge to the global optimum; however, this is a general problem with non-convex optimization problems, see e.g. [7].

Corollary 1 can be solved directly, via the following SOS programming problem.

Corollary 3. *Let the system $\Gamma = (f, X, X_0, X_\mathrm{u}, D)$ and polynomials g_* shown in (22) be given, and let $\epsilon_1 > 0$. If there exist $B \in \mathcal{P}_n$, $\lambda_{X_0} \in \Sigma_n^{k_{X_0}}$, $\lambda_{X_\mathrm{u}} \in \Sigma_n^{k_{X_\mathrm{u}}}$, $\lambda_X \in \Sigma_{n+m}^{k_X}$, and $\lambda_D \in \Sigma_{n+m}^{k_D}$ such that*

$$-B - \lambda_{X_0}^\mathrm{T} g_{X_0}, \tag{27a}$$

$$B - \epsilon_1 - \lambda_{X_\mathrm{u}}^\mathrm{T} g_{X_\mathrm{u}}, \text{ and} \tag{27b}$$

$$-\frac{\partial B}{\partial x} f - \lambda_X^\mathrm{T} g_X - \lambda_D^\mathrm{T} g_D \tag{27c}$$

are sum of squares. Then the system Γ is safe.

4.2 Computation of Compositional Barrier Certificates

In this subsection, we show how barrier certificates can be expressed in a compositional manner, using SOS optimization for Proposition 2 and Corollary 2, and using LMIs for Corollary 2. The interconnected system can be formulated as one system, but this would increase the number of decision variables involved in the safety verification, compared to the proposed compositional approach. This is an important issue when working with SOS optimization, and is apparent from (19).

Let $k \in \mathbb{N}$ be the number of subsystems, and define $g_* \in \mathcal{P}^{k_*}$. In the decomposition, the considered sets are restricted, as shown in (8), where

$$X_i \equiv \{x_i \in \mathbb{R}^{n_i} | g_{X_i}(x_i) \geq 0\}, \tag{28a}$$

$$X_{0,i} \equiv \{x_i \in \mathbb{R}^{n_i} | g_{X_{0,i}}(x_i) \geq 0\}, \tag{28b}$$

$$X_{\mathrm{u},i} \equiv \{x_i \in \mathbb{R}^{n_i} | g_{X_{\mathrm{u},i}}(x_i) \geq 0\}, \tag{28c}$$

$$D_i \equiv \{d_i \in \mathbb{R}^{m_i} | g_{D_i}(d_i) \geq 0\}, \tag{28d}$$

$$U_i \equiv \{u_i \in \mathbb{R}^{q_i} | g_{U_i}(u_i) \geq 0\}. \tag{28e}$$

Proposition 2 is written in terms of SOS in the following.

Proposition 4. *Let $k \in \mathbb{N}$, the polynomials g_* shown in (28) and the system $\Gamma = (\{f_i\}, \{X_i\}, \{X_{0,i}\}, \{X_{\mathrm{u},i}\}, \{D_i\})$ be given, and let $\epsilon_1, \epsilon_2 > 0$. If there exist $B_i \in \mathcal{P}_{n_i}$, $\alpha_i \in \mathbb{R}$, $\beta_i \in \mathbb{R}$, $\gamma_i \in \mathcal{P}_{q_i+r_i}$, $\lambda_{X_{0,i}} \in \Sigma_{n_i}^{k_{X_{0,i}}}$, $\lambda_{X_{\mathrm{u},i}} \in \Sigma_{n_i}^{k_{X_{\mathrm{u},i}}}$, $\lambda_{B_i} \in \mathcal{P}_{n_i+m_i+q_i}$, $\lambda_{D_i} \in \Sigma_{n_i+m_i+q_i}^{k_{D_i}}$, and $\lambda_{U_i} \in \Sigma_{n_i+m_i+q_i}^{k_{U_i}}$ such that*

$$-B_i - \lambda_{X_{0,i}}^\mathrm{T} g_{X_{0,i}} - \alpha_i, \tag{29a}$$

$$B_i - \epsilon_1 - \lambda_{X_{\mathrm{u},i}}^\mathrm{T} g_{X_{\mathrm{u},i}} - \beta_i, \text{ and} \tag{29b}$$

$$-\frac{\partial B_i}{\partial x_i} f_i - \epsilon_2 + \gamma_i - \lambda_{D_i}^\mathrm{T} g_{D_i} - \lambda_{B_i}^\mathrm{T} B_i - \lambda_{U_i}^\mathrm{T} g_{U_i} \tag{29c}$$

are sum of squares and

$$\sum_i \alpha_i, \sum_i \beta_i, \text{ and } -\sum_i \gamma_i \tag{29d}$$

are sum of squares. Then the system Γ is safe.

Proposition 4 has a product between $\lambda_{B_i}^\mathrm{T}$ and B_i, which implies that Algorithm 1 must be used to solve it. Additionally, dual decomposition should be used to decompose the conditions; however, this is only demonstrated for the following SOS program for solving Corollary 2.

Corollary 4. *Let $k \in \mathbb{N}$, the polynomials g_* shown in (28) and the system $\Gamma = (\{f_i\}, \{X_i\}, \{X_{0,i}\}, \{X_{\mathrm{u},i}\}, \{D_i\})$ be given, and let $\epsilon_1 > 0$. If there exist $B \in \mathcal{P}_{n_i}$, $\alpha_i \in \mathbb{R}$, $\beta_i \in \mathbb{R}$, $\gamma_i \in \mathcal{P}_{q_i+r_i}$, $\lambda_{X_{0,i}} \in \Sigma_{n_i}^{k_{X_{0,i}}}$, $\lambda_{X_{\mathrm{u},i}} \in \Sigma_{n_i}^{k_{X_{\mathrm{u},i}}}$, $\lambda_{X_i} \in \Sigma_{n_i+m_i+q_i}^{k_{X_i}}$, $\lambda_{D_i} \in \Sigma_{n_i+m_i+q_i}^{k_D}$, and $\lambda_{U_i} \in \Sigma_{n_i+m_i+q_i}^{k_{U_i}}$ such that*

$$-B_i - \lambda_{X_{0,i}}^\mathrm{T} g_{X_{0,i}} - \alpha_i, \tag{30a}$$

$$B_i - \epsilon_1 - \lambda_{X_{\mathrm{u},i}}^\mathrm{T} g_{X_{\mathrm{u},i}} - \beta_i, \text{ and} \tag{30b}$$

$$-\frac{\partial B_i}{\partial x_i} f_i + \gamma_i - \lambda_{X_i}^\mathrm{T} g_{X_i} - \lambda_{D_i}^\mathrm{T} g_{D_i} - \lambda_{U_i}^\mathrm{T} g_{U_i} \tag{30c}$$

are sum of squares and

$$\sum_i \alpha_i, \sum_i \beta_i, \text{ and } -\sum_i \gamma_i. \tag{30d}$$

are sum of squares. Then the system Γ is safe.

In the following, we show how to prove safety using LMIs based on Corollary 2. The vector field f_i is given by

$$\dot{x} = A_i x_i + B_{1,i} d_i + B_{2,i} u_i \tag{31a}$$

$$y_i = C_i x_i, \tag{31b}$$

where A_i is an $n_i \times n_i$ matrix, $B_{1,i}$ is an $n_i \times m_i$ matrix, $B_{2,i}$ is an $n_i \times m_i$ matrix, and C_i is an $q_i \times n_i$ matrix. We say that $B_i(x_i) = x_i^\mathrm{T} P_i x_i$, $g_*(x_i) = x_i^\mathrm{T} G_* x_i$ where P_i and G_* are symmetric matrices, and $\alpha_i, \beta_i \in \mathbb{R}$. Furthermore, we define γ_i as

$$\gamma_i = \begin{bmatrix} u_i \\ x_i \end{bmatrix}^\mathrm{T} \begin{bmatrix} \Gamma_{u,i} & 0 \\ 0 & \Gamma_{x,i} \end{bmatrix} \begin{bmatrix} u_i \\ x_i \end{bmatrix}, \tag{32}$$

where $\Gamma_{u,i}$ and $\Gamma_{x,i}$ are diagonal matrices.

Corollary 5. *Let $k \in \mathbb{N}$, the polynomials g_* shown in (28) and the system $\Gamma = (\{f_i\}, \{X_i\}, \{X_{0,i}\}, \{X_{\mathrm{u},i}\}, \{D_i\})$ be given, where $\{f_i\}$ is a collection of linear vector fields, and G_* is symmetric. If there exist $P_i = P_i^\mathrm{T}$, $\alpha_i, \beta_i \in \mathbb{R}$, and matrices $\Gamma_{u,i}, \Gamma_{x,i}$ given in (32), $\lambda_{X_{0,i}} \in \mathbb{R}_{\geq 0}$, $\lambda_{X_i} \in \mathbb{R}_{\geq 0}$, $\lambda_{D_i} \in \mathbb{R}_{\geq 0}$, and $\lambda_{U_i} \in \mathbb{R}_{\geq 0}$ such that*

$$-P_i - \lambda_{X_{0,i}} G_{X_{0,i}} - \alpha_i I \geq 0 \tag{33a}$$

$$P_i - \lambda_{X_i} G_{X_i} - \beta_i I > 0, \text{ and} \tag{33b}$$

$$\begin{bmatrix} A_i^\mathrm{T} P_i + P_i A_i + \lambda_{X_i} G_{X_i} & P_i B_{1,i} & P_i B_{2,i} & C^\mathrm{T} \\ B_{1,i}^\mathrm{T} P_i & \lambda_{D_i} G_{D_i} & 0 & 0 \\ B_{2,i}^\mathrm{T} P_i & 0 & \lambda_{U_i} G_{U_i} - \Gamma_{u,i} & 0 \\ C & 0 & 0 & -\Gamma_{x,i} \end{bmatrix} < 0, \tag{33c}$$

and

$$\sum_i \alpha_i \geq 0,\ \sum_i \beta_i \geq 0,\ \sum_i \gamma_i \leq 0. \tag{33d}$$

Then the system Γ *is safe.*

To practically solve the safety problem in Corollary 4, we set up an optimization problem by use of dual decomposition [3]. Dual decomposition can be used to solve different types of optimization problems. We consider only the following type of optimization problem.

$$\begin{aligned}&\text{minimize } f(x) = f_1(x_1, y) + f_2(x_2, y) \text{ subject to}\\ &x_1 \in C_1,\ x_2 \in C_2,\ h_1(x_1, y) + h_2(x_2, y) \leq 0.\end{aligned} \tag{34}$$

We decompose (34) into two separate optimization problems, which are coupled through some additional decision variables as follows

$$\begin{aligned}&\text{minimize } f(x) = f_1(x_1, y_1) + f_2(x_2, y_2) \text{ subject to}\\ &x_1 \in C_1,\ x_2 \in C_2,\ y_1 = y_2,\ h_1(x_1, y_1) + h_2(x_2, y_2) \leq 0.\end{aligned}$$

The dual problem can be set up, as f_1 and f_2 have no shared variables. The Lagrangian for the problem is

$$\begin{aligned}L(x_1, y_1, x_2, y_2, \lambda_1, \lambda_2) &= f_1(x_1, y_1) + f_2(x_2, y_2)\\ &+\lambda_1^{\mathrm{T}}(y_1 - y_2) + \lambda_2\left(h_1(x_1, y_1) + h_2(x_2, y_2)\right).\end{aligned} \tag{35}$$

Let $\lambda = (\lambda_1, \lambda_2)$. The dual function becomes

$$\varphi(\lambda_1, \lambda_2) = \varphi_1(\lambda_1, \lambda_2) + \varphi_2(\lambda_1, \lambda_2), \tag{36}$$

where

$$\varphi_1(\lambda) = \inf_{x_1, y_1}\left(f_1(x_1, y_1) + \lambda_1^{\mathrm{T}} y_1 + \lambda_2 h_1(x_1, y_1)\right), \tag{37a}$$

$$\varphi_2(\lambda) = \inf_{x_2, y_2}\left(f_2(x_2, y_2) - \lambda_1^{\mathrm{T}} y_2 + \lambda_2 h_2(x_2, y_2)\right). \tag{37b}$$

The optimization problems for φ_1 and φ_2 can be solved independently, given values for λ_1 and λ_2. Finally, the master problem is

$$\text{maximize } \varphi_1(\lambda_1, \lambda_2) + \varphi_2(\lambda_1, \lambda_2), \tag{38}$$

with variables λ_1 and λ_2.

To solve the master problem, we utilize the subgradient algorithm given in [15]. Note that all functions in this paper are polynomial, thus differentiable; hence, other gradient methods can be used instead of the subgradient method.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function, and let $x, y \in \mathbb{R}^n$. Then any vector $g \in \mathbb{R}^n$ that satisfies

$$f(y) \geq f(x) + g^{\mathrm{T}}(y - x) \tag{39}$$

is called a subgradient at x.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function. Then the subgradient algorithm gives a sequence of points $\{x^{(k)}\}_{k=0}^{\infty}$ according to

$$x^{(k+1)} = x^{(k)} - \Delta_k g^{(k)}, \tag{40}$$

where $x^{(k)}$ is the k^{th} iterate, $x^{(0)}$ is the initial point, $g^{(k)}$ is a subgradient of f at $x^{(k)}$, and Δ_k is the step size. When the function f to be minimized is differentiable, then $g^{(k)}$ is the unique gradient of f at point $x^{(k)}$.

For diminishing step size, the algorithm is guaranteed to converge to the optimal value, see [11]. Therefore, we use the following diminishing step size

$$\Delta_k = \frac{a}{b + k}, \tag{41}$$

where $a > 0$ and $b \geq 0$.

The following algorithm is used to solve the dual decomposition for the problem shown in (34). Note that we denote by $\bar{x}_1^{(k)}$ and $\bar{y}_1^{(k)}$ the optimal values of x_1 and y_1 for problem (37a) at iteration k, given some λ_1, λ_2.

Algorithm 2. *Given an optimization problem, as shown in* (34).

```
0. Initialization: Let $k = 0$, define the step size $\Delta_k$, and
   choose some $\lambda_1^{(0)}$, $\lambda_2^{(0)}$, $\epsilon > 0$.

1. Solve subproblems:
   Solve (37a) to find $\bar{x}_1^{(k)}$ and $\bar{y}_1^{(k)}$,
   solve (37b) to find $\bar{x}_2^{(k)}$ and $\bar{y}_2^{(k)}$.

2. Update dual variables:
   $\lambda_1^{(k+1)} := \lambda_1^{(k)} - \Delta_k(\bar{y}_2^{(k)} - \bar{y}_1^{(k)})$,
   $\lambda_2^{(k+1)} := \lambda_2^{(k)} + \Delta_k(h_1(\bar{x}_1^{(k)}, \bar{y}_1^{(k)}) + h_2(\bar{x}_2^{(k)}, \bar{y}_2^{(k)}))$,
   $k := k + 1$.
   If $|\lambda_1^{(k+1)} - \lambda_1^{(k)}| > \epsilon$, then go to step 1. Otherwise,
   terminate the algorithm.
```

Note that step 2 in Algorithm 2 tries to maximize (38).

The first observation in the considered problem is that γ_i has to be a diagonal matrix; otherwise, the cost of the optimization problem is not linear. For convenience, we let $\bar{\gamma}_i$ be a vector containing the diagonal elements of γ_i. Let $\lambda \equiv (\lambda_1,\ \lambda_2,\ \lambda_3)$ the dual function is

$$\varphi(\lambda) = \sum_i \varphi_i(\lambda), \tag{42}$$

where

$$\varphi_i(\lambda) \equiv \inf_{\alpha_i, \beta_i, \bar{\gamma}_i} -\lambda_1\alpha_i - \lambda_2\beta_i + \lambda_3^{\mathrm{T}}\bar{\gamma}_i \tag{43}$$

subject to

$$-B_i - \lambda_{X_{0,i}}^{\mathrm{T}} g_{X_{0,i}} - \alpha_i, \tag{44a}$$

$$B_i - \epsilon_1 - \lambda_{X_{\mathrm{u},i}}^{\mathrm{T}} g_{X_{\mathrm{u},i}} - \beta_i, \text{ and} \tag{44b}$$

$$-\frac{\partial B_i}{\partial x_i} f_i + \gamma_i - \lambda_{D_i}^{\mathrm{T}} g_{D_i} \tag{44c}$$

are sum of squares.

Remark that λ_3 is a vector. The dual problem becomes

$$\sup_{\lambda \geq 0} \sum_i \varphi_i(\lambda). \tag{45}$$

In the following, we explain how the subgradient algorithm can be used to solve the previous optimization problem. Let $\alpha_i^*(\lambda)$ be the optimal value of α_i for a given λ. Then the gradients of $\varphi_1(\lambda), \ldots, \varphi_k(\lambda)$ are

$$g_i(\lambda) = \begin{bmatrix}\alpha_i^*(\lambda) & \beta_i^*(\lambda) & \gamma_i^*(\lambda)\end{bmatrix}. \tag{46}$$

From (39) and (46), we get for all $\mu \equiv (\mu_1, \mu_2, \mu_3)$ and $i = 1, \ldots, k$

$$\varphi_i(\mu) \geq \varphi_i(\lambda) + g_i(\mu - \lambda). \tag{47}$$

The function to be maximized is $\varphi(\lambda) = \sum_i \varphi_i(\lambda)$, which has a gradient $g(\lambda^{(k)}) = \sum_i g_i(\lambda^{(k)})$. The vector of multipliers is updated according to (40), and is

$$\lambda^{(k+1)} = \lambda^{(k)} - \Delta_k g^{\mathrm{T}}\left(\lambda^{(k)}\right). \tag{48}$$

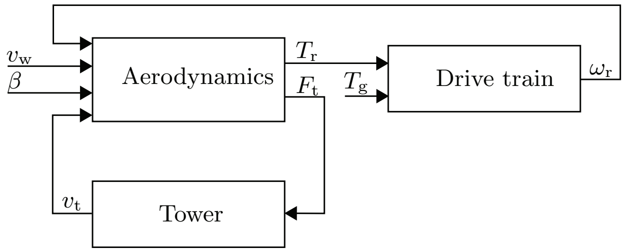

Figure 4: Wind turbine modeled as an interconnection of three subsystems.

It is seen that if $\sum_i \alpha_i \geq 0$ is violated, then $\lambda_1^{(k+1)} > \lambda_1^{(k)}$, as the first element of $g(\lambda^{(k)})$ is negative. This puts a larger penalty on the violation of the constraint through the dual variable λ_1.

5. EXAMPLE

In this section, we demonstrate the applicability of the compositional safety analysis, by analyzing the safety of an emergency shutdown of a wind turbine. The emergency shutdown procedure is simplified for presentation, and the wind turbine model is a slight modification of the CART3 wind turbine model [8].

The wind turbine is modeled as an interconnection of three subsystems: aerodynamics (subsystem 1), tower (subsystem 2), and drive train (subsystem 3). A block diagram of the wind turbine is shown in Figure 4.

The wind turbine is driven by an exogenous input - the wind v_w. Via the aerodynamics, the wind exerts a torque T_r on the rotor shaft, and a force F_t on the top of the tower. This bends the tower and makes the rotor shaft rotate. The rotor shaft is connected to a generator through a gear and a generator shaft. A converter applies a torque T_g to the generator shaft.

The magnitude of the torque T_r and the force F_t depends on the pitch angle β, the rotor speed ω_r, and the wind speed at the rotor $v_\mathrm{w} - v_\mathrm{t}$, given by the speed of the wind v_w and the velocity of the tower v_t. These relations are usually described by lookup tables (C_p and C_t tables); however, we approximate them by polynomials.

In case of severe faults, a wind turbine is shut down by pitching the blades to an angle of $\beta = 90^\circ$, while applying a constant generator torque $T_\mathrm{g} = 3,580$ Nm, until the rotor speed is below a threshold of 0.77 rad/s, from which it is not possible to apply a torque from the generator; hence, the wind turbine is left uncontrolled. At a pitch angle of 90°, the aerodynamic thrust is acting in the opposite direction of the nominal rotation; hence, it decelerates. Additionally, by applying a relatively high generator torque, the rotor shaft is decelerated even faster. This may cause the tower to sway too much or twist the rotor shaft beyond the limit accepted by the turbine structure. Therefore, we verify that this does not happen. The subsystems of the wind turbine are modeled as shown in (49), and will be left without further explanation.

$$\begin{aligned}
\begin{bmatrix}\dot{v}_\mathrm{r}\\ \dot{\omega}_\mathrm{r,f}\end{bmatrix} &= \begin{bmatrix}-c_{v_r}v_\mathrm{r} + (v_\mathrm{w} - v_\mathrm{t})\\ -c_{\omega_\mathrm{r,f}}\omega_\mathrm{r,f} + \omega_\mathrm{r}\end{bmatrix}\\
h_1 &= \begin{bmatrix}p_1\\ p_2\end{bmatrix}, \qquad (49a)\\
\begin{bmatrix}\dot{v}_\mathrm{t}\\ \dot{x}_\mathrm{t}\end{bmatrix} &= \begin{bmatrix}\frac{1}{M_\mathrm{t}}(F_\mathrm{t} - B_\mathrm{t}v_\mathrm{t} - k_\mathrm{t}x_\mathrm{t})\\ v_\mathrm{t}\end{bmatrix} \qquad (49b)\\
h_2 &= v_\mathrm{t}\\
\begin{bmatrix}\dot{\omega}_\mathrm{r}\\ \dot{\theta}_\Delta\\ \dot{\omega}_\mathrm{g}\end{bmatrix} &= \begin{bmatrix}\frac{1}{J_\mathrm{r}}(T_\mathrm{r} - k_\mathrm{r}\theta_\Delta - B_\mathrm{r}(\omega_\mathrm{r} - \frac{1}{N_\mathrm{g}}\omega_\mathrm{g}))\\ \omega_\mathrm{r} - \frac{1}{N_\mathrm{g}}\omega_\mathrm{g}\\ \frac{1}{J_\mathrm{g}}\left(\frac{1}{N_\mathrm{g}}(k_\mathrm{r}\theta_\Delta + B_\mathrm{r}(\omega_\mathrm{r} - \frac{1}{N_\mathrm{g}}\omega_\mathrm{g})) - T_\mathrm{g}\right)\end{bmatrix} \qquad (49c)\\
h_3 &= \omega_\mathrm{r}
\end{aligned}$$

where

$$\begin{aligned}
p_1 &= \big(c_{11} + c_{12}\omega_\mathrm{r,f} + c_{13}v_\mathrm{r} + c_{14}\omega_\mathrm{r,f}^2 + c_{15}v_\mathrm{r}^2\\
&\quad +c_{16}\omega_\mathrm{r,f}v_\mathrm{r}\big)\, v_\mathrm{r}^3\\
p_2 &= (c_{21} + c_{22}\omega_\mathrm{r,f} + c_{23}v_\mathrm{r} + c_{24}v_\mathrm{r}^2 + c_{25}\omega_\mathrm{r,f}v_\mathrm{r})v_\mathrm{r}^2\\
&\quad + c_{26} + c_{27}\omega_\mathrm{r,f}^2
\end{aligned}$$

The parameters of the wind turbine are the following: $M_\mathrm{t} = 7.76\cdot 10^3$ kg, $B_\mathrm{t} = 18.6$ kN/(m/s), $k_\mathrm{t} = 2.7$ MN/m, $N_\mathrm{g} = 43$, $J_\mathrm{r} = 611.1 \cdot 10^3$ kgm^2, $B_\mathrm{r} = 24$ kNm/(rad/s), $k_\mathrm{r} = 24.7 \cdot 10^6$ Nm/rad, $c_{v_r} = 11.65$, $c_{\omega_\mathrm{r,f}} = 21$, $c_{11} = -32.42 \cdot 10^6$, $c_{12} = -746.0\cdot 10^6$, $c_{13} = 53.03\cdot 10^6$, $c_{14} = -1.128\cdot 10^9$, $c_{15} = -18.63\cdot 10^6$, $c_{16} = 384.6\cdot 10^6$, $c_{21} = 8492.6$, $c_{22} = 300.88\cdot 10^3$, $c_{23} = -11.85 \cdot 10^3$, $c_{24} = 3584.0$, $c_{25} = -90.32 \cdot 10^3$, $c_{26} = 318.3$, and $c_{27} = 1.692 \cdot 10^6$. We have omitted the units on the constants c_*, as they have no physical interpretation.

The considered region of the state space is

$$X_1 = [2,\, 28] \times [0.77,\, 4], \qquad (50a)$$
$$X_2 = [-0.01,\, 0.07] \times [-0.05,\, 0.05], \qquad (50b)$$
$$X_3 = [0.77,\, 4] \times [-25,\, 25] \cdot 10^{-3} \times [33.2,\, 172.7]. \qquad (50c)$$

Furthermore, the inputs to the subsystems take values in the following sets

$$D_1 = [5,\, 25], \qquad (51a)$$
$$U_1 = [0.77,\, 4] \times [-0.5,\, 0.5], \qquad (51b)$$
$$U_2 = [1.3554,\, 94.413] \cdot 10^3, \qquad (51c)$$
$$U_3 = [-141.86,\, -0.126] \cdot 10^6. \qquad (51d)$$

It is chosen to initialize the system in the so called full load region, corresponding to a wind speed between 11.7 m/s and 25 m/s, where the wind turbine is operated at a constant rotor speed of 3.88 rad/s; hence, the set of initial states is

$$X_{0,1} = [11.7,\, 25] \times [3.8,\, 3.95], \qquad (52a)$$
$$X_{0,2} = [-0.005,\, 0.005] \times [0.01,\, 0.02], \qquad (52b)$$
$$X_{0,3} = [3.8,\, 3.95] \times [6.25,\, 6.27] \cdot 10^{-3} \times [164,\, 171]. \qquad (52c)$$

We should verify that the following unsafe sets cannot be

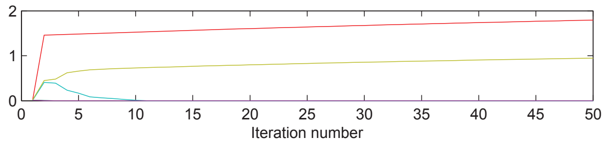

Figure 5: Values of the multipliers λ as a function of the number of iterations.

reached

$$\begin{aligned} X_{\mathrm{u},1} &= [2,\,3] \cup [27,\,28] \times [0.77,\,4], && \text{(53a)}\\ X_{\mathrm{u},2} &= [-0.01,\,0] \cup [0.06,\,0.07] \\ &\quad \times [-0.04,\,-0.03] \cup [0.03,\,0.04], && \text{(53b)}\\ X_{\mathrm{u},3} &= [0.77,\,4] \times [-25,\,-10]\cdot 10^{-3} \cup [10,\,25]\cdot 10^{-3} \\ &\quad \times [33.2,\,172.7]. && \text{(53c)} \end{aligned}$$

Now the verification problem has been set up, and we do the verification using Corollary 4. To allow the verification, we need to characterize X_i, $X_{0,i}$, $X_{\mathrm{u},i}$, and U_i by polynomials. This is accomplished as by specifying a maximum value $x_{\max}$ and minimum value $x_{\min}$ of some variable x, and then defining

$$g \equiv -(x - x_{\min})(x - x_{\max}). \tag{54}$$

The polynomial g is nonnegative for $x \in [x_{\min},\, x_{\max}]$ and otherwise negative.

To give an impression of the convergence of the algorithm, the values of the multipliers $\lambda_1, \ldots, \lambda_6$ are shown in Figure 5. The safety of the system is verified by Corollary 4, and the barrier function is

$$\begin{aligned} B(x) = {}& 0.0388\omega_{\mathrm{r}}^2\theta_\Delta^2 + 0.0350\omega_{\mathrm{r}}^2\theta_\Delta + 0.748\omega_{\mathrm{r}}^2 \\ & - 0.00869\omega_{\mathrm{r}}\omega_{\mathrm{g}} + 0.569\omega_{\mathrm{g}}^2 - 0.00332\omega_{\mathrm{g}}\theta_\Delta \\ & + 1.223\theta_\Delta^2 - 97.0\cdot 10^{-6} v_{\mathrm{t}}^2 x_{\mathrm{t}} - 0.256 v_{\mathrm{t}}^2 \\ & + 0.00173 v_{\mathrm{t}} x_{\mathrm{t}}^2 - 2.15 v_{\mathrm{t}} x_{\mathrm{t}} + 0.0658 v_{\mathrm{t}} - 0.755 x_{\mathrm{t}}^2 \\ & + 0.0785 x_{\mathrm{t}} + 0.00387 v_{\mathrm{r}}^2 - 0.00943 v_{\mathrm{r}}\omega_{\mathrm{r,f}} \\ & - 0.107 v_{\mathrm{r}} + 0.0207\omega_{\mathrm{r,f}}^2 + 0.103\omega_{\mathrm{r,f}} + 1.609. \end{aligned} \tag{55}$$

6. CONCLUSION

We have presented a method for verifying the safety of an interconnection of subsystems. The method is based on the identification of barrier certificates, where the certificates are found for each subsystem, but are coupled through some additional constraints.

The presented method allows the safety verification of higher dimensional systems, as the verification is decomposed into smaller coupled subproblems, and allows subsystems to be analyzed with different computational methods.

The method has been used to verify the safety of an emergency shutdown procedure for a wind turbine.

7. REFERENCES

[1] A. Abate, A. Tiwari, and S. Sastry. Box invariance in biologically-inspired dynamical systems. *Automatica*, 45(7):1601–1610, 2009.

[2] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *SIAM studies in applied mathematics*. SIAM, 1994.

[3] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley. Notes on decomposition methods, 2008.

[4] A. Girard and G. Pappas. Verification using simulation. In J. Hespanha and A. Tiwari, editors, *Hybrid Systems: Computation and Control*, volume 3927 of *Lecture Notes in Computer Science*, pages 272–286. Springer Berlin / Heidelberg, 2006.

[5] H. Guéguen, M.-A. Lefebvre, J. Zaytoon, and O. Nasri. Safety verification and reachability analysis for hybrid systems. *Annual Reviews in Control*, 33(1):25–36, 2009.

[6] J. Helton and O. Merino. Coordinate optimization for bi-convex matrix inequalities. In *Proceedings of the 36th IEEE Conference on Decision and Control*, pages 3609–3613, December 1997.

[7] T. Iwasaki. The dual iteration for fixed-order control. *IEEE Transactions on Automatic Control*, 44(4):783–788, April 1999.

[8] J. Laks, L. Pao, and A. Wright. Control of wind turbines: Past, present, and future. In *American Control Conference*, pages 2096–2103, June 2009.

[9] I. Mitchell, A. Bayen, and C. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005.

[10] P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96(2):293–320, 2003.

[11] B. T. Polyak. Subgradient methods: A survey of Soviet research. In *Proceedings of a IIASA Workshop*, volume 3 of *Nonsmooth Optimization*, pages 5–29, 1977.

[12] S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *Hybrid Systems: Computation and Control*, volume 2993 of *Lecture Notes in Computer Science*, pages 271–274. Springer Berlin / Heidelberg, 2004.

[13] S. Prajna, A. Jadbabaie, and G. J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8):1415–1428, August 2007.

[14] S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo. SOSTOOLS and its control applications. In *Positive Polynomials in Control*, volume 312 of *Lecture Notes in Control and Information Sciences*, pages 273–292. Springer Berlin / Heidelberg, 2005.

[15] N. Z. Shor, K. C. Kiwiel, and A. Ruszcayǹski. *Minimization methods for non-differentiable functions.* Springer-Verlag New York, Inc., New York, NY, USA, 1985.

[16] U. Topcu, A. Packard, and R. Murray. Compositional stability analysis based on dual decomposition. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 1175–1180, December 2009.