Sampling based Computation of Viability Domain to Prevent Safety Violations by Attackers

Kunal Garg Alvaro A. Cardenas Ricardo G. Sanfelice

arXiv:2110.08632v1 [eess.SY] 16 Oct 2021

***Abstract*— This paper studies the security of cyber-physical systems under attacks. Our goal is to design system parameters, such as a set of initial conditions and input bounds so that it is secure by design. To this end, we propose new sufficient conditions to guarantee the safety of a system under adversarial actuator attacks. Using these conditions, we propose a computationally efficient sampling-based method to verify whether a set is a viability domain for a general class of nonlinear systems. In particular, we devise a method of checking a modified barrier function condition on a finite set of points to assess whether a set can be rendered forward invariant. Then, we propose an iterative algorithm to compute the set of initial conditions and input constraint set to limit what an adversary can do if it compromises the vulnerable inputs. Finally, we utilize a Quadratic Program approach for online control synthesis.**

I. Introduction

Security has become one of the most critical problems in the field of Cyber-Physical Systems (CPS), as illustrated by several incidents of attacks that happened in the past few years [1], [2]. There are two types of security mechanisms for protecting CPS [3] i) proactive, which considers design choices deployed in the CPS *before* attacks, and ii) reactive, which take effect after an attack is detected.

While reactive methods are less conservative than proactive mechanisms, they heavily rely on fast and accurate attack detection mechanisms. Although there is a plethora of work on attack detection for CPS [4], [5], it is generally possible to design a stealthy attack such that the system behavior remains close to its expected behavior, thus evading attack-detection solutions [6]. Intrusion detection systems also produce a large number of false positives, which can lead to a large operational overhead of security analysts dealing with irrelevant alerts [7]. On the other hand, a proactive method can be more effective in practice, particularly against stealthy attacks. Attacks on a CPS can disrupt the natural operation of the system. One of the most desirable system properties is *safety*, i.e., the system does not go out of a safe zone. Safety is an essential requirement, violation of which can result in failure of the system, loss of money, or even loss of human life, particularly when a system is under attack [8].

In most practical problems, safety can be realized as guaranteeing forward-invariance of a safe set. Control barrier function (CBF) based approach [9] to guarantee forward invariance of the safe region has become very popular in the last few years since a safe control input can be efficiently computed using a Quadratic Program (QP) with CBF condition as the constraint. Most of the prior work on safety using CBFs, e.g., [9], assumes that the *viability* domain, i.e., the set of initial conditions from which forward invariance of the safe set can be guaranteed, is known. In practice, it is not an easy task to compute the viability domain for a nonlinear control system. Optimization-based methods, such as Sum-of-Squares (SOS) techniques, have been used in the past to compute this domain (see [10]). However, SOS-based approaches are only applicable to systems whose dynamics is given by polynomial functions, thus limiting their applications. Another method popularly used in the literature for computing the viability domain is Hamilton-Jacobi (HJ) based reachability analysis, see, e.g., [11]. However, such an analysis is computationally expensive, particularly for higher dimensional systems. We propose a novel sampling-based method to compute the viability domain for a general class of nonlinear control systems to overcome these limitations.

In this work, we consider a general class of nonlinear systems under actuator attacks and propose a method of computing a set of initial conditions and an input constraint set such that the system remains *secure by design*. In particular, we consider actuator manipulation, where an attacker can assign arbitrary values to the input signals for a subset of the actuators in a given bound. We consider the property of safety with respect to an unsafe set and propose sufficient conditions using sampling of the boundary of a set to verify whether the set is a *viability domain* under attacks. Using these conditions, we propose a computationally tractable algorithm to compute the set of initial conditions and the input constraint set such that the system's safety can be guaranteed under attacks. In effect, our proposed method results in a secure-by-design system that is resilient against actuator attacks. Finally, we leverage these sets in a QP-based approach with provable feasibility for real-time online feedback synthesis. The contributions of the paper are summarized below:

1) We present sampling-based sufficient conditions to assess whether a given set can be rendered forward invariant for a general class of nonlinear system. To the best of the authors' knowledge, this is the first work utilizing sampled-data approach for computing a viability domain;
2) We present a novel iterative algorithm to compute a viability domain and an input constraint set to guarantee

The research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-20-1-0253. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

system safety under attacks. Unlike [10], [11], the proposed method is applicable for general nonlinear control systems and is scalable with the system dimension.

Notation: Throughout the paper, $\mathbb{R}$ denotes the set of real numbers and $\mathbb{R}_+$ denotes the set of non-negative real numbers. We use $|x|$ to denote the Euclidean norm of a vector $x \in \mathbb{R}^n$. We use ∂S to denote the boundary of a closed set $S \subset \mathbb{R}^n$ and $\text{int}(S)$ to denote its interior and $|x|_S = \inf_{y \in S} |x - y|$, to denote the distance of $x \in \mathbb{R}^n$ from the set S. The Lie derivative of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ along a vector field $f : \mathbb{R}^n \to \mathbb{R}^m$ at a point $x \in \mathbb{R}^n$ is denoted as $L_f h(x) := \frac{\partial h}{\partial x}(x) f(x)$.

II. Problem Formulation

We start with defining a model for the system and the attacker considered in this paper.

A. System model

Consider a nonlinear control system $\mathcal{S}$ given as

$$\mathcal{S} : \begin{cases} \dot{x} = F(x, u) + d(t, x), \\ x \in \mathcal{D}, u \in \mathcal{U}, \end{cases} \tag{1}$$

where $F : \mathcal{D} \times \mathcal{U} \to \mathbb{R}^n$ is a known function continuous on $: \mathcal{D} \times \mathcal{U}$, with $\mathcal{D} \subset \mathbb{R}^n$ and $\mathcal{U} \subset \mathbb{R}^m$, $d : \mathbb{R}_+ \times \mathbb{R}^n \to \mathbb{R}^n$ is unknown and represents the unmodeled dynamics, $x \in \mathcal{D}$ is the system state, and $u \in \mathcal{U}$ is the control input.

B. Attacker model

In this paper, we consider attacks on the control input of the system. In particular, we consider an attack where a subset of the components of the control input is compromised. Under such an attack, the system input takes the form:

$$u = (u_v, u_s), \tag{2}$$

where $u_v \in \mathcal{U}_v \subset \mathbb{R}^{m_v}$ represents the *vulnerable* components of the control input that might be compromised or attacked, and $u_s \in \mathcal{U}_s \subset \mathbb{R}^{m_s}$ the *secure* part that cannot be attacked, with $m_v + m_s = m$ and $\mathcal{U} := \mathcal{U}_v \times \mathcal{U}_s$. Under this class of attack, we assume that we know which components of the control input are vulnerable. For example, if the system has four inputs so that $u = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T$, and u_1, u_3 can be attacked, then we assume that this information is known, and u_v is comprised of u_1 and u_3. We discuss how to address the assumption of which components of the control input are vulnerable in Remark 1 in Section IV.

Such attack models have been used in prior work, see e.g., [12], and can be implemented in practice by designing the dynamic range of the actuator to preserve its bounds. It can also be implemented in software with the help of a reference monitor [13] between the controller and the actuator that can check if the desired control inputs satisfy the security policy [12]. As discussed in [14], various prototypical attacks, such as *stealth attacks*, *replay attacks*, and *false-data injection attacks* can be captured by the attack model in (2). In addition to representing a real-world scenario where system actuators have physical limits, constraining the vulnerable control input u_v in the set $\mathcal{U}_v$ has several advantages:

1) It restricts how much an attacker can change the nominal operation of the system [15], and can be implemented physically, so an attacker cannot bypass it.
2) It can be utilized to design a detection mechanism, e.g., if $u_v \notin \mathcal{U}_v$, a flag can be raised signifying that the system is under an attack. Schemes that raise a threshold-based flag are commonly used as detection mechanism [5].
3) The constraint set $\mathcal{U}_v$ can be designed appropriately such that the system remains *secured* under attacks, as discussed in Section IV (see also [12], [15]).

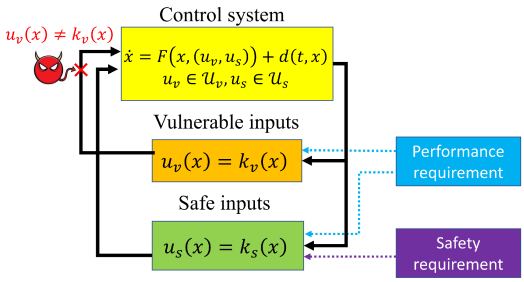

Fig. 1. Overview of our approach.

Now, we present the control design objectives. Consider a non-empty, compact set $S \subset \mathbb{R}^n$, referred to as safe set, to be rendered forward invariant. We make the following assumption on the unmodeled dynamics d in (1):

Assumption 1. *There exists $\delta > 0$ such that $|d(t, x)| \leq \delta$ for all $t \geq 0$ and $x \in \mathcal{D}$.*

We consider two properties when designing the control law, an *essential property (safety)*, imposed while designing the secure input u_s, and a *desirable property (performance)*, imposed while designing both u_s and u_v (see Figure 1). The problem we study in this paper is as follows.

Problem 1. *Given the system in (1) with unmodeled dynamics d that satisfies Assumption 1, a set S and the attack model in (2), design a feedback law $k_s : \mathbb{R}^n \to \mathcal{U}_s$, and find a set of initial conditions $X_0 \subset S$ and the input constraint set $\tilde{\mathcal{U}}_v \subset \mathcal{U}_v$, such that for all $x(0) \in X_0$ and $u_v : \mathbb{R}_+ \to \tilde{\mathcal{U}}_v$, the closed-loop trajectories $x : \mathbb{R}_+ \to \mathbb{R}^n$ of (1) resulting from using $u_s = k_s(x)$ satisfy $x(t) \in S$ for all $t \geq 0$.*

In plain words, we consider the problem of designing a feedback law u_s and compute a set of initial conditions X_0 and input constraint set $\tilde{\mathcal{U}}_v$, such that even under an attack as per the attack model (2), the system trajectories do not leave the safe set S, i.e., the system is secure by design. In this work, we assume that the safe set is given as $S := \{x \mid B(x) \leq 0\}$ where $B : \mathbb{R}^n \to \mathbb{R}$ is a sufficiently smooth user-defined function.

C. Outline of approach

Given a control system (1), and an attack model (2), we first identify a safe set $S \subset \mathbb{R}^n$ and the vulnerable input

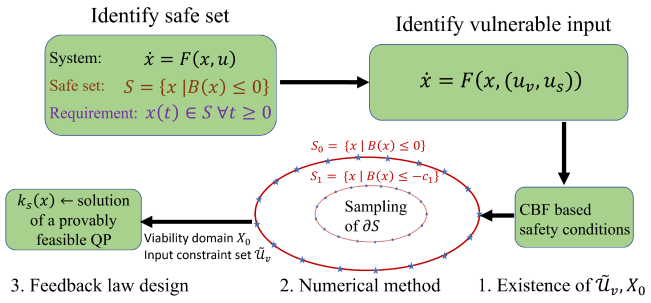

Fig. 2. Approach for safe feedback design under attacks.

u_v. Then, our approach to solving Problem 1 involves the following steps (see Figure 2):

1) *Establish the existence of X_0 and $\mathcal{U}_v$ (Section III)*: leverage CBFs to find sufficient conditions to check whether there exist a set of initial conditions X_0, input constraint set $\tilde{\mathcal{U}}_v \subset \mathcal{U}_v$, and a control input u_s for all $x \in X_0$ that can solve Problem 1;
2) *Numerical method for computation of X_0 and $\mathcal{U}_v$ (Section IV)*: use conditions in step 1) to formulate a numerical method for computing sets X_0 and $\tilde{\mathcal{U}}_v$;
3) *Feedback law synthesis (Section V)*: use the sets X_0 and $\tilde{\mathcal{U}}_v$ from step 2) to design a feedback control law $u_s = k_s(x)$ that solves Problem 1.

D. Preliminaries

Next, we present preliminaries on forward invariance.

Definition 1. *A set $S \subset \mathbb{R}^n$ is termed as* forward invariant *for system (1) if every solution $x : \mathbb{R}_+ \to \mathbb{R}^n$ of (1) satisfies $x(t) \in S$ for all $t \ge 0$ and for all initial conditions $x(0) \in S$.*

Next, we review a sufficient condition for guaranteeing forward invariance of a set in the absence of an attack. For the sake of simplicity, in what follows, we assume that every solution of (1) exists and is unique in forward time for all $t \ge 0$ whether or not there is an attack on the system.[1]

Lemma 1 ([17])**.** *Given a continuously differentiable function $B : \mathbb{R}^n \to \mathbb{R}$, the set $S = \{x \mid B(x) \le 0\}$ is forward invariant for (1) with $d \equiv 0$ if the following condition holds:*

$$\inf_{u \in \mathcal{U}} L_F B(x,u) \le 0 \quad \forall x \in \partial S. \tag{3}$$

Satisfaction of (3) means the set S is termed as a *viability* domain for system (1) with $d \equiv 0$. Using Lemma 1, and following the notion of robust CBF in [18], we can state the following result guaranteeing forward invariance in the presence of disturbance d.

Lemma 2 ([18])**.** *Given a continuously differentiable function $B : \mathbb{R}^n \to \mathbb{R}$, the set $S = \{x \mid B(x) \le 0\}$ is forward invariant for (1) under d satisfying Assumption 1 if the following condition holds:*

$$\inf_{u \in \mathcal{U}} L_F B(x,u) \le -l_B \delta \quad \forall x \in \partial S, \tag{4}$$

where l_B is the Lipschitz constant of the function B.

III. Sufficient Conditions for Safety

In this section, we present sufficient conditions that guarantee the security of the system model (1) against attacks on the input. We say that the system (1) is *secure with respect to the safety property* for a set S if for all initial conditions $x(0) \in S$, $x(t) \in S$ for all $t \ge 0$, $u_v \in \mathcal{U}_v$ and d satisfying Assumption 1. Given B and F, define $H : \mathbb{R}^n \times \mathbb{R}^{m_v} \to \mathbb{R}$:

$$H(x,u_v) := \inf_{u_s \in \mathcal{U}_s} L_F B(x,(u_v,u_s)). \tag{5}$$

It is not necessary that the zero sublevel set S of the function B is a viability domain for system (1). Any non-empty sublevel set $S_c := \{x \mid B(x) \le -c\}$, where $c \ge 0$, being a viability domain is sufficient for safety of the system. Note that the set S_c is non-empty for $0 \le c \le -\min_{x \in S} B(x)$. Define $c_M \in \mathbb{R}$ as

$$c_M := -\min_{x \in S} B(x), \tag{6}$$

so that the set of feasible values for c is given as $[0, c_M]$.[2] The following result provides sufficient conditions for a system to be secured with respect to the safety property.

Proposition 1. *Suppose there exist $c \in [0, c_M]$ and nonempty $\tilde{\mathcal{U}}_v \subset \mathcal{U}_v$ such that*

$$\sup_{u_v \in \tilde{\mathcal{U}}_v} H(x,u_v) \le -l_B \delta \quad \forall x \in \partial S_c, \tag{7}$$

and the system solutions are uniquely defined in forward time for all $x(0) \in S_c$. Then, for each d satisfying Assumption 1, system (1) is secured with respect to the safety property for the set S_c.

Proof. Note that

$$\inf_{\substack{u_s \in \mathcal{U}_s \\ u_v \in \tilde{\mathcal{U}}_v}} L_F(x,(u_v,u_s)) = \inf_{u_v \in \tilde{\mathcal{U}}_v} H(x,u_v) \le \sup_{u_v \in \tilde{\mathcal{U}}_v} H(x,u_v).$$

Thus, from (7), it follows that (4) holds. Thus, per Lemma 2, the set S_c is forward invariant and it holds that the system (1) is secured with respect to the safety property for set S_c. ■

Note that satisfaction of the conditions in Proposition 1 implies that for all $x \in \partial S_c$ and $u_v \in \bar{\mathcal{U}}_v$, there exists an input $u_s \in \mathcal{U}_s$ such that the inequality $L_F B(x,(u_v,u_s)) \le -l_B \delta$ holds. This, in turn, implies that the set S_c is a viability domain for system (1). Condition (7) requires checking the inequality $\sup_{u_v \in \tilde{\mathcal{U}}_v} H(x,u_v) \le -l_B \delta$ for all points on the boundary of the set S_c. Such conditions are commonly used in the literature for control synthesis, assuming that the viability domain is known. However, it is not an easy task to compute a viability domain in practice for a general

[1] In this work, we assume that even under attack, the solution of the system is unique in forward time. It is possible to study the case when this assumption does not hold using the notion of strong invariance (see [16]).

[2] Note that compactness of the set S guarantees existence of c_M.

class of nonlinear systems *a priori*. In the next section, we present a computationally tractable method where we show that checking a modification of the inequality in (7) on a set of sampling points on the boundary is sufficient.

IV. Viability domain under bounded inputs

In this section, we present numerical algorithms to assess whether given system (1) and the function B, there exist c and an input constraint set $\tilde{\mathcal{U}}_v$ such that condition (7) holds. First, we present a sampling-based method for evaluating whether the condition (7) holds by checking a modified inequality at a finite set of sampling points. Then, we propose an iterative method to compute c and the set $\tilde{\mathcal{U}}_v$.

A. Viability domain using sampling data

We start by making the following assumption on the regularity of the function H defined in (5).

Assumption 2. *The function* $\sup_{u_v\in\tilde{\mathcal{U}}_v} H(\cdot,u_v)$ *is Lipschitz continuous on* S *with constant* $l_H>0$.

First, to illustrate the method, we consider the 3-D case, i.e., when $x\in\mathbb{R}^3$. If the compact set $S\subset\mathbb{R}^3$ is diffeomorphic to a unit sphere in $\mathbb{R}^3$, then the sampling points on the boundary of the unit sphere can be used to obtain the points on the boundary of S_c. Thus, without loss of generality, we can study the case when $S\subset\mathbb{R}^3$ is a unit sphere with center $x_o\in\mathbb{R}^3$. Let $\{x_i\}_{\mathcal{I}}$, with each $x_i\in\partial S_c$, denote the set of N_p sampling data points on the boundary of the sublevel set S_c for a given $c\in[0,c_M]$ with c_M defined in (6) and $\mathcal{I}:=\{1,2,\ldots,N_p\}$. The sampling points $\{x_i\}_{\mathcal{I}}$ are such that they constitute a polyhedron $P_{\mathcal{I}}$ with $N_f>0$ triangular faces, $T_1,T_2,\ldots,T_{N_f}$, such that $P_{\mathcal{I}}$ *triangulates* the boundary ∂S_c, i.e., the intersection of any two distinct triangles is either empty, a single vertex, or a single edge. Figure 3 shows an example of triangulation of a unit sphere in $\mathbb{R}^3$. Interested readers on algorithms and details on triangulation are referred to [19], and the references therein.

Note that a tetrahedron is the *minimal* triangulation (i.e., a triangulation with minimum number of triangular faces) for a unit sphere. Using geometric arguments, it is easy to show that the minimum possible value of the maximum of the inter-vertex distances for a tetrahedron inscribed in a unit sphere is $\sqrt{3}$. The corresponding arc-length along the boundary of the unit sphere (denote as d_a) is $2\sin^{-1}\sqrt{\frac{3}{4}}$. It follows that if $d_a\le 2\sin^{-1}\sqrt{\frac{3}{4}}$, then there must be at least $N_p=4$ points in the polyhedron. Finally, with $0\le r_c\le 1$ being the radius of the sphere S_c,[3] the corresponding arc-length for S_c is

$$d_M := 2r_c\sin^{-1}\sqrt{\frac{3}{4}}. \tag{8}$$

Now, to ensure that they are enough sampling points, the following conditions can be imposed on $\{x_i\}_{\mathcal{I}}$ for a given $c\in[0,c_M]$ and $d_a\in\left[0,d_M\right]$

[3]If the set S_c is defined as $S_c=\{x\mid |x|^2-1\le -c\}$, then $r_c=\sqrt{1-c}$.

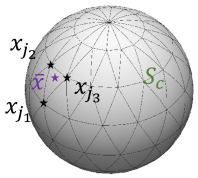

Fig. 3. 3-D case: Triangulating sampling of the boundary ∂S_c.

- For each $x\in\partial S_c$, there exists a triangular face T_j with vertices $x_{j_1},x_{j_2},x_{j_3}\in\{x_i\}_{\mathcal{I}}$, of the polyhedron $P_{\mathcal{I}}$ generated by $\{x_i\}_{\mathcal{I}}$, such that $x_o+\theta(x-x_o)\in T_j$ for some $0\le\theta\le 1$; and
- The following holds:

$$\max_{\substack{l\neq m\\ l,m=1,2,3}} d_{S_c}(x_{j_l},x_{j_m})\le d_a, \tag{9}$$

where $d_{S_c}(x,y)$ denotes the shortest arc-length between the points $x,y\in\partial S_c$.

In plain words, the above conditions require for each point $x\in\partial S_c$, the line joining the center x_o and x intersects a triangular face of the polyhedron such that the distance along the boundary ∂S_c between the vertices of this face is bounded by d_a. Note that smaller d_a requires larger number of sampling points N_p. Now, we show that if the following holds

$$\sup_{u_v\in\tilde{\mathcal{U}}_v} H(x_i,u_v)\le -l_H d_a - l_B\delta \quad \forall i\in\mathcal{I}, \tag{10}$$

where l_B is the Lipschitz constant for B and δ, l_H are as defined in Assumptions 1 and 2, respectively, then, (7) holds. With $c\in[0,c_M]$, the set S_c is non-empty, and with $d_a\in\left[0,d_M\right]$, there exist sufficient points N_p to have a polyhedron that can triangulate the boundary ∂S_c. Under the conditions imposed on $\{x_i\}_{\mathcal{I}}$, for every $\bar{x}\in\partial S_c$, there exists a triangular face T_j with coordinates $x_{j_1},x_{j_2},x_{j_3}\in\{x_i\}_{\mathcal{I}}$ satisfying (9) and $0\le\theta\le 1$ such that $x_o+\theta(\bar{x}-x_o)\in T_j$. Using Lipschitz continuity of $\sup_{u_v} H(\cdot,u_v)$ per Assumption 2, it holds that $\sup_{u_v\in\tilde{\mathcal{U}}_v} H(\bar{x},u_v)\le\sup_{u_v\in\tilde{\mathcal{U}}_v} H(x,u_v)+l_H|\bar{x}-x|$, for all $x,\bar{x}\in\partial S_c$. For $x=x_i$, $i\in\mathcal{I}$, using (10) and the fact that $|x-y|\le d_{S_c}(x,y)$ for all $x,y\in\partial S_c$, we obtain that

$$\sup_{u_v\in\tilde{\mathcal{U}}_v} H(\bar{x},u_v)\le -l_H d_a - l_B\delta + l_H d_{S_c}(\bar{x},x_i),\ \forall i\in\mathcal{I}.$$

Since the projection of $\bar{x}$ lies in the triangular face T_j, it holds that $d_{S_c}(\bar{x},x_{j_k})\le d_{S_c}(x_{j_l},x_{j_k})$ for $l\neq k$, $l,k\in\{1,2,3\}$. Using this and (9), we obtain

$$\sup_{u_v\in\tilde{\mathcal{U}}_v} H(\bar{x},u_v)\le -l_H d_a - l_B\delta + l_H d_a = -l_B\delta,\ \forall\bar{x}\in\partial S_c.$$

Thus, checking the inequality (10) at a finite number of points is a computationally tractable method for assessing whether (7) holds for a given c and $\tilde{\mathcal{U}}_v$. Note that for a given $F, B, \tilde{\mathcal{U}}_v$ and δ, a smaller value of d_a implies that the right-hand side of (10) is less negative, thus, making it easier to satisfy the inequality. At the same time, due to (9), a smaller value of d_a requires more sampling points N_p, and hence,

checking the inequality at more points. Thus, there is a trade-off between the ease of satisfaction of (10) and the number of points at which the inequality should be checked.

The above arguments can be generalized to the $n-$dimensional case. Using the sampling approach in [20] for a unit sphere in $n-$dimension, combined with Delaunay Triangulation of the sampling points (see e.g., [21]), an $(n-1)-$dimensional *simplex* can be obtained. If the compact set $S \subset \mathbb{R}^n$ is diffeomorphic to a unit $(n-1)-$sphere, then sampling points on the boundary of S can be obtained using the sampling points for the $(n-1)-$unit sphere. Thus, we study the case when the set S is an $(n-1)-$unit sphere.

Let $\{x_i\}_{\mathcal{I}}$, with each $x_i \in \partial S_c$, denote the set of N_p sampling data points on the boundary of the sublevel set S_c for a given $c \in [0, c_M]$ with c_M defined in (6) and $\mathcal{I} := \{1, 2, \ldots, N_p\}$. The sampling poins $\{x_i\}_{\mathcal{I}}$ constitute a simplex $\mathcal{S}_{\mathcal{I}}$ with $N_f > 0$ faces, $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_{N_f}$. For a unit sphere in $\mathbb{R}^n$, the minimum number of points in the simplex is $(n+1)$, and the minimum possible value of the maximum of the lengths of its edges is $\sqrt{\frac{2(n+1)}{n}}$. The length, denoted as d_a, of the corresponding arc-length on the boundary ∂S_c is $2r_c \sin^{-1}\sqrt{\frac{(n+1)}{2n}}$, where $0 \leq r_c \leq 1$ is the radius of the sphere S_c. Thus, with $d_a \leq 2r_c \sin^{-1}\sqrt{\frac{(n+1)}{2n}}$, there must be at least $(n+1)$ points in the simplex. For the sake of brevity, define $d_{M,n} := 2r_c \sin^{-1}\sqrt{\frac{(n+1)}{2n}}$. We make the following assumption on the sampling points $\{x_i\}_{\mathcal{I}}$.

Assumption 3. *Given $c \in [0, c_M]$, the sampling points $\{x_i\}_{\mathcal{I}}$ and $d_a \in [0, d_M]$, for each $x \in \partial S_c$, there exists a face $\mathcal{X}_j$ with vertices $\{x_{j_1}, x_{j_2}, \ldots, x_{j_n}\} \in \{x_i\}_{\mathcal{I}}$, where $j \in \{1, 2, \ldots, N_f\}$, of the simplex $\mathcal{S}_{\mathcal{I}}$ generated by $\{x_i\}_{\mathcal{I}}$, such that $x_o + \theta(x - x_o) \in \mathcal{X}_j$ for some $0 \leq \theta \leq 1$, and the following holds:*

$$\max_{\substack{l \neq m \\ l,m=1,2,\ldots,n}} d_{S_c}(x_{j_l}, x_{j_m}) \leq d_a, \tag{11}$$

where $d_{S_c}(x, y)$ denotes the shortest arc-length between the points $x, y \in \partial S_c$.

We have the following result when S is $(n-1)-$unit sphere.

Theorem 1. *Suppose that the function H defined in (5) satisfies Assumption 2. Given $c \in [0, c_M]$, $d_a \in [0, d_{M,n}]$, and the sampling points $\{x_i\}_{\mathcal{I}}$, if Assumption 3 and (10) hold, then, (7) holds.*

Proof. With $c \in [0, c_M]$, the set S_c is non-empty, and with $d_a \in [0, d_M]$, there exist sufficient points N_p to have a simplex. Now, consider any point $\bar{x} \in \partial S_c$. Under Assumption 3, for every $\bar{x} \in \partial S_c$, there exists a face $\mathcal{X}_j$ of the simplex $\mathcal{S}_{\mathcal{I}}$, such that the line joining the center of the sphere S_c and the point $\bar{x}$ lies on this face. Using Lipschitz continuity of $\sup_{u_v} H(\cdot, u_v)$ under Assumption 2 and (10), it holds that

$$\begin{aligned}\sup_{u_v \in \tilde{\mathcal{U}}_v} H(\bar{x}, u_v) &\leq \sup_{u_v \in \mathcal{U}_v} H(x, u_v) + l_H|\bar{x} - x| \\ &\leq -l_H d_a - l_B\delta + l_H|\bar{x} - x|,\end{aligned}$$

for all $x, \bar{x} \in \partial S_c$. Using the inequality for $x = x_{j_i}$, $i \in \{1, \ldots, n\}$ and the fact that $|\bar{x} - x_{j_i}| \leq d_{S_c}(\bar{x}, x_{j_i}) \leq d_{S_c}(x_{j_k}, x_{j_i})$ for any $k \neq i$, $k \in \{1, \ldots, n\}$ and (11), we obtain that

$$\sup_{u_v \in \tilde{\mathcal{U}}_v} H(\bar{x}, u_v) \leq -l_H d_a - l_B\delta + l_H d_a = -l_B\delta$$

for all $\bar{x} \in \partial S_c$, which completes the proof. ■

Note that there are three set of parameters that can facilitate satisfaction of (10) in the following manner:

- Set $\tilde{\mathcal{U}}_v$: smaller $\tilde{\mathcal{U}}_v$ makes it easier to satisfy (10);
- Parameter c: larger value of c results in smaller values of d_M, thus, reducing the right-hand side of (10), and making it easier to satisfy it; and
- Number of sampling points N_p: larger N_p results in smaller value of $d_{M,n}$.

Based on these observations, an iterative algorithm can be formulated to check whether there exists a feasible c and a non-empty set $\tilde{\mathcal{U}}_v$, such that (10) holds.

B. Iterative algorithm

We formulate our algorithm with the following steps:

1) For a given value of $0 \leq c \leq c_M$, $\tilde{\mathcal{U}}_v$ and number of sampling points N_p, sample $\{x_i\}_{\mathcal{I}}$ from the set ∂S_c and check if (10) holds for all the sampling points;
2) Shrink $\tilde{\mathcal{U}}_v$, increase c and repeat steps 1)-2) until the condition (10) is satisfied for all the sampling points, or there does not exist a c and a non-empty set $\tilde{\mathcal{U}}_v$;
3) Increase N_p and repeat steps 1)-3) until (10) holds or the maximum value (N_{max}) of N_p is reached.

Using these steps, we propose Algorithm 1 which returns a feasible c and a set $\tilde{\mathcal{U}}_v$ such that safety is guaranteed for all $x \in S_c$ and $u_v \in \tilde{\mathcal{U}}_v$. In other words, this algorithm can compute the set of initial conditions S_c, and the set of *tolerable* attacked inputs via $\tilde{\mathcal{U}}_v$ such that the system can satisfy the safety property under attacks. The order in which the parameters $c, \tilde{\mathcal{U}}_v$, and N_p are tuned can be changed, which can potentially change the output of the algorithm.

Remark 1. *If it is unknown which components of the input are vulnerable, then all possible combinations of u_v and u_s can be considered, and Algorithm 1 can be used to compute c for each such combination. Then, the maximum of all such values can be used to define the set S_c, guaranteeing the system's security against attack on any control inputs.*

Remark 2. *The computational complexity of Algorithm 1 is only a function of the number of sampling points N_p (which, in principle, is a user-defined parameter) and is independent of the non-linearity of the function F, and linear in the dimension n. Thus, unlike reachability based tools in [22], [11] where the computational complexity grows*

Algorithm 1: Iterative method for computing $\tilde{\mathcal{U}}_v, c$

Data: $f, g_v, g_s, \mathcal{U}_v, \mathcal{U}_s, B, d_a, \varepsilon_1, \varepsilon_2, \delta, N_{max}, N_{c0}$

Initialize: $\tilde{\mathcal{U}}_v = \mathcal{U}_v, c = 0, N_p = N_{c0}$;

while $N_p < N_{max}$ **do**

  while $c \leq c_M$ **do**

    Sample $\{x_i\}_{\mathcal{I}}$ from $\{B(x) \leq -c\}$;

    while $\tilde{\mathcal{U}}_v \neq \emptyset$ **do**

      if $\{i \in \mathcal{I} \mid H(x_i, u_v) > -l_H d_a + l_B \delta\} \neq \emptyset$ **then**

        $\tilde{\mathcal{U}}_v = \tilde{\mathcal{U}}_v \ominus \varepsilon_1$;

    if $\tilde{\mathcal{U}}_v = \emptyset$ **then**

      $c = c + \varepsilon_2$;

      $\tilde{\mathcal{U}}_v = \mathcal{U}_v$;

  $N_p = 2\, N_p$;

  $c = 0$;

Return: $\tilde{\mathcal{U}}_v, c$;

exponentially with the system dimension n, or SOS based tools [10] that are only applicable to a specific class of systems with linear or polynomial dynamics, Algorithm 1 can be used for general nonlinear system with high dimension.

So far, we presented sufficient conditions to establish the safety of the system (1) under attacks (Proposition 1), a sampling-based method to verify these conditions using a finite number of sampling points (Theorem 1), and iterative methods to compute the set of initial conditions and the input constraint set to satisfy these conditions (Algorithm 1). Thus, in brief, using the results in this section, we can compute the viability domain S_c and control input constraint set $\tilde{\mathcal{U}}_v \subset \mathcal{U}_v$, such that for all $x \in S_c$ and $u_v \in \tilde{\mathcal{U}}_v$, there exists a control input $u_s \in \mathcal{U}_s$ that can keep the system trajectories in the set S_c at all times. In the next section, we present a method of computing such a control input using a QP formulation.

V. QP BASED FEEDBACK DESIGN

In this section, we use the sufficient conditions from the previous section to design a feedback law for the system (1) that guarantees security with respect to the safety property under Assumption 1. We assume that the control input constraint set is given as $\tilde{\mathcal{U}} := \tilde{\mathcal{U}}_v \times \mathcal{U}_s = \{v \in \mathbb{R}^m \mid u_{j,min} \leq v_j \leq u_{j,max}\}$, i.e., as a box-constraint set where $u_{j,min} < u_{j,max}$ are the lower and upper bounds on the individual control inputs v_j for $j = 1, 2, \ldots, m$, respectively. We can write $\mathcal{U}$ in a compact form as $\tilde{\mathcal{U}} = \{v \mid A_u v \leq b_u\}$ where $A_u \in \mathbb{R}^{2m \times m}, b_u \in \mathbb{R}^{2m}$. Furthermore, we assume that the system model (1) is control affine, and is of the form:

$$\dot{x} = f(x) + g_v(s)u_v + g_s(x)u_s + d(t, x), \tag{12}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$, $g_v : \mathbb{R}^n \to \mathbb{R}^{n \times m_v}$ and $g_s : \mathbb{R}^n \to \mathbb{R}^{n \times (m - m_v)}$ are continuous functions. In this case, the function $H : \mathbb{R}^n \times \mathbb{R}^{m_v} \to \mathbb{R}$ reads

$$H(x, u_v) = \inf_{u_s \in \mathcal{U}_s} L_f B(x) + L_{g_s} B(x) u_s + L_{g_v} B(x) u_v. \tag{13}$$

In addition to the safety requirement in Problem 1, we impose the requirement of convergence of the system trajectories of (12) to the origin. To this end, given a twice continuously differentiable, positive definite function $V : \mathbb{R}^n \to \mathbb{R}_+$ as a candidate Lyapunov function, we use the condition

$$L_f V(x) + L_{g_s} V(x) u_s + L_{g_v} V(x) u_v \leq -\zeta V(x) - l_V \delta, \tag{14}$$

where $\zeta > 0$, to guarantee convergence of the system trajectories to the origin under d satisfying Assumption 1. We assume that the set S is an $(n-1)$-unit sphere, so that we can use the results from the previous section to compute a viability domain for it, and that $0 \in \text{int}(S)$, so that the convergence requirement is feasible. The linear constraints on the control input, and the system model being control affine, helps us formulate a convex optimization problem that can be efficiently solved for real-time control synthesis [9]. We propose the following Quadratic Program (QP) to solve Problem 1. Define $z = (v_s, v_v, \eta, \zeta) \in \mathbb{R}^{m+2}$ and for a given $x \in \mathbb{R}^n$, consider the following QP:

$$\min_{z} \quad \frac{1}{2}|z|^2 + q\zeta \tag{15a}$$

$$\text{s.t.} \quad A_u v_{na} \leq b_u, \tag{15b}$$

$$L_f B(x) + L_{g_s} B(x) v_s \leq -\eta\ (B(x) + c) - \sup_{u_v \in \tilde{\mathcal{U}}_v} L_{g_v} B(x) u_v - l_B \delta, \tag{15c}$$

$$L_f V(x) + L_{g_s} V(x) v_s + L_{g_v} V(x) v_v \leq -\zeta\ V(x) - l_V \delta, \tag{15d}$$

where $q > 0$ is a constant, l_B, l_v are the Lipschitz constants of the functions B and V, respectively, and c and $\tilde{\mathcal{U}}_v$ are the output of Algorithm 1. Here, η and ζ are slack variables used for guaranteeing feasibility of the QP (see [23, Lemma 6]). The first constraint (15b) is the input constraints, the second constraint is the CBF condition from Lemma 2 for forward invariance of the set S_c and the third constraint (15d) is CLF constraint for convergence of the system trajectories to the origin. Note that the secure input v_s is used in both (15c) and (15d), while the vulnerable input v_v is only used in (15d).

Let the optimal solution of (15) at a given point $x \in \mathbb{R}^n$ be denoted as $z^*(x) = (v_s^*(x), v_v^*(x), \eta^*(x), \zeta^*(x))$. In order to guarantee continuity of the solution z^* with respect to x, we need to impose the strict complementary slackness condition on (15) (see [23]). In brief, if the $i-$the constraint of (15), with $i \in \{1, 2, 3\}$, is written as $G_i(x, z) \leq 0$, and the corresponding Lagrange multiplier is $\lambda_i \in \mathbb{R}_+$, then strict complementary slackness requires that $\lambda_i^* G(x, z^*) < 0$, where z^*, λ_i^* denote the optimal solution and the corresponding optimal Lagrange multiplier, respectively. We are now ready to state the following result.

Theorem 2. *Given the functions F, d, B, V and the attack model (2), suppose Assumptions 1-3 hold. Let c and $\tilde{\mathcal{U}}_v$ be the output of the Algorithm 1. Assume that the strict complementary slackness holds for the QP (15) for all $x \in S_c$. Then, the QP (15) is feasible for all $x \in S_c$, and the control law defined as $k_s(x) = v_s^*(x)$ is continuous on $\text{int}(S_c)$, and solves Problem 1 for all $x(0) \in X_0 := \text{int}(S_c)$.*

Proof. Per Theorem 1, the set S_c is a viability domain for the system (12) under Assumption 2. Thus, feasibility

of the QP (15) follows from [23, Lemma 6]. Note also that with V being twice continuously differentiable and under Assumption 2, the Lie derivatives of the functions V and B along f, g_s, and g_v are continuous. Thus, per [23, Theorem 1], the solution z^* of the QP (15) is continuous on $\text{int}(S_c)$. Finally, since the set S_c is compact, it follows from [23, Lemma 7] that the closed-loop trajectories are uniquely defined for all $t \geq 0$. Uniqueness of the closed-loop trajectories, Assumption 1 and feasibility of the QP (15) for all $x \in S_c$ implies that all the conditions of Lemma 2 are satisfied and it follows that the set S_c is forward invariant for the system (12). ■

Remark 3. *In this work, only the control input u_s is used to achieve safety since it is unknown when the vulnerable input u_v comes under an attack. This conservative assumption can be relaxed by utilizing an attack-detection mechanism, which can trigger a switching mechanism from a* nominal *control design, assuming no attacks, to the proposed method under an attack. We leave this detection-based switching mechanism as part of our future work.*

VI. Numerical Experiments

To showcase the effectiveness of the proposed method, we present an academic example with the system given as

$$\dot{x} = f(x) + Ax + Bu + d(t, x), \tag{16}$$

where $A \in \mathbb{R}^{3\times3}$ and $B \in \mathbb{R}^{3\times2}$. The input constraint sets are $\mathcal{U}_1 = \{u_1 \in \mathbb{R} \mid |u_1| \leq u_{M1}\}$ and $\mathcal{U}_2 = \{u_2 \in \mathbb{R} \mid |u_2| \leq u_{M2}\}$ for some $u_{M1}, u_{M2} > 0$. The safe set is $S = \{x \in \mathbb{R}^3 \mid |x|^2 - 1 \leq 0\}$ corresponding to the function $h(x) = |x|^2 - 1$, i.e., the safe set is the unit sphere. We use randomly generated matrices A and B such that the pairs (A, B_1) and (A, B_2) are controllable, where B_1 and B_2 are the first and the second columns of the matrix B, respectively. The matrices (A, B) and the function f are

$$A = \begin{bmatrix} 0.61 & 0.37 & 2.69 \\ -0.06 & -1.02 & -0.88 \\ 1.33 & -2.71 & 0.91 \end{bmatrix}, B = \begin{bmatrix} -0.24 & 0.04 \\ 0.32 & -0.01 \\ -1.12 & -0.07 \end{bmatrix}, f(x) = 0.01 \begin{bmatrix} x_1^3 + x_2^2 x_3 \\ x_2^3 + x_3^2 x_1 \\ x_3^3 + x_1^2 x_2 \end{bmatrix}.$$

We use MATLAB code from [24] to generate a uniform sampling on the boundary of the unit sphere. Figure 4 shows the maximum value of $\sup_{u_v \in \tilde{\mathcal{U}}_v} H(\bar{x}, u_v) + l_H d_a + l_B \delta$ over

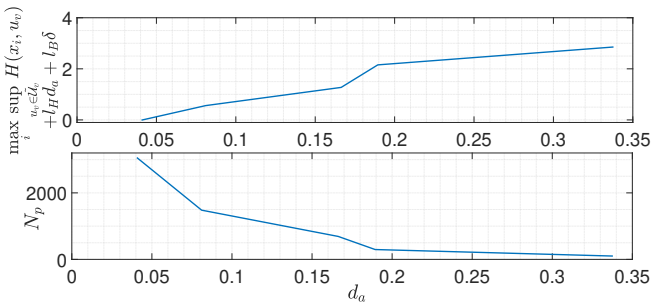

Fig. 4. The value of $\max_i \sup_{u_v \in \tilde{\mathcal{U}}_v} H(\bar{x}, u_v) + l_H d_a + l_B \delta + l_H d_a$ and the number of sampling points for different values of d_a.

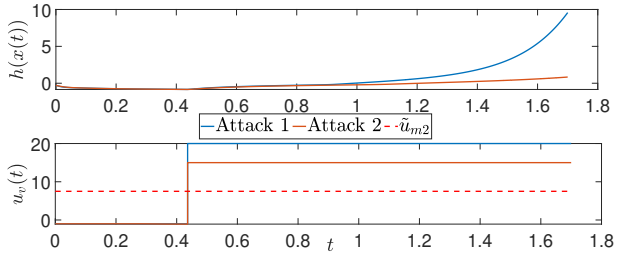

Fig. 5. The vulnerable input u_v and the function h under attacks 1 and 2.

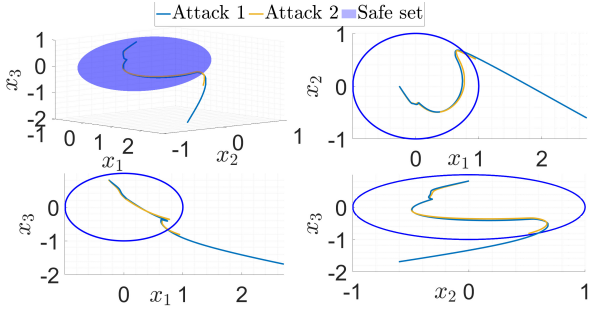

Fig. 6. The closed-loop paths traced by the system under attacks 1 and 2.

the sampling points for different values of d_a. It is observed that condition (10) is satisfied when $d_a = 0.0406$, and the corresponding number of sampling points is $N_p = 3062$. Without loss of generality, we assume that u_2 is vulnerable. We use Algorithm 1 to compute the set $\tilde{\mathcal{U}}_i$ and a value of c such that (10) holds for all the sampling points. With $u_{M1} = 20$ and $u_{M2} = 20$ (defining the sets $\mathcal{U}_s, \mathcal{U}_v$), Algorithm 1 gives $c = 0$ for the viability domain $\{x \mid h(x) \leq c\}$ and $\tilde{u}_{M2} = 7.5$ (defining the set $\tilde{\mathcal{U}}_v$) as the feasible bound on the attack signal u_2. The attack happens at a randomly chosen $\tau = 0.436$ and $\delta = 0.1$ in Assumption 1.

First, we illustrate that the system violates safety when the attack signal u_2 does not satisfy the bounds computed by Algorithm 1. Figure 5 shows the vulnerable input u_v for the initial two attack scenarios (Attack 1 and 2) where $\bar{u}_{M2} = 20$ and $\bar{u}_{M2} = 15$, i.e., the set $\tilde{\mathcal{U}}_v$ is larger than the one computed using the proposed algorithm. Figure 5 also plots the evolution of the barrier function h with time for the two cases. It can be observed that the function h corresponding to this attack takes positive values, and thus, the safety property for the system is violated. Figure 6 plots the corresponding closed-loops paths for the two scenarios, and it can be seen that the system leaves the safe set, thus violating safety.

In the rest of the attack scenarios (Attack 3-6), the bound $|u_v| \leq 7.5$ is imposed as computed by the proposed algorithm. Figure 7 plots the different types of attack signals used in these scenarios, namely, saturated signals with $u_v = 7.5$ and $u_v = -7.5$, square wave and sinusoidal signal, both with amplitude 7.5. The corresponding evolution of the barrier function h illustrates that the system maintains safety in all four scenarios. Figure 8 plots the closed-loops paths for these attack scenarios, and it can be seen that the system trajectories evolve in the safe set at all times, thus maintaining safety. Through this case study, we illustrate that if the system parameters are not chosen according to our proposed method, then there might exist attacks that can lead

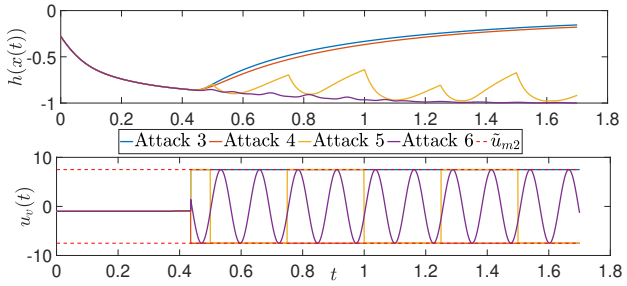

Fig. 7. The vulnerable input u_v and the function h under attacks 3-6.

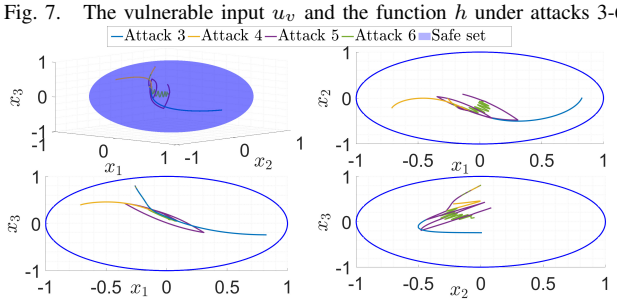

Fig. 8. The closed-loop paths traced by the system under attacks 3-6.

to violation of safety. On the other hand, when the system parameters are designed according to the proposed algorithm, no attack can violate safety, confirming that the system is secure by design.

VII. Conclusion and Future Work

In this paper, we study the problem of computing a viability domain and input constraint set so that the safety of a system can be guaranteed under attacks on the system inputs. In contrast to prior work on the computation of viability domain whose applicability is limited to linear or polynomial dynamics or whose computational complexity grows exponentially with system dimension, our method is computationally efficient and applies to a general class of nonlinear systems. We showed that when the system parameters are chosen using our sampling-based iterative algorithm, the resulting system is resilient to arbitrary attacks, and thus, is secure by design.

Our approach can be used to design bounds (that can either be implemented physically or in a tamper-proof reference monitor) that will prevent attackers from driving control systems to unsafe states. It is efficient (sampling-based viability computation) and general (applicable to non-linear systems). By limiting the range of actuation and the initial set, we are limiting the responsiveness of control action, and in general, systems with our defense might converge slower to the desired set point or trajectory. One way to mitigate this is to use attack-detection mechanisms and switching strategy so that more efficient controllers can be used when the system is not under an attack.

References

[1] R. M. Lee, M. J. Assante, and T. Conway, "German steel mill cyber attack," *Industrial Control Systems*, vol. 30, p. 62, 2014.

[2] N. E. Oueslati, H. Mrabet, A. Jemai, and A. Alhomoud, "Comparative study of the common cyber-physical attacks in industry 4.0," in *2019 International Conference on Internet of Things, Embedded Systems and Communications*. IEEE, 2019, pp. 1–7.

[3] A. Cardenas, "Cyber-physical systems security knowledge area issue." The Cyber Security Body Of Knowledge. [Online]. Available: https://www.cybok.org/media/downloads/Cyber-Physical_Systems_Security_issue_1.0.pdf

[4] H. Choi, W.-C. Lee, Y. Aafer, F. Fei, Z. Tu, X. Zhang, D. Xu, and X. Deng, "Detecting attacks against robotic vehicles: A control invariant approach," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 801–816.

[5] V. Renganathan, N. Hashemi, J. Ruths, and T. H. Summers, "Distributionally robust tuning of anomaly detectors in cyber-physical systems with stealthy attacks," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 1247–1252.

[6] D. I. Urbina, J. A. Giraldo, A. A. Cardenas, N. O. Tippenhauer, J. Valente, M. Faisal, J. Ruths, R. Candell, and H. Sandberg, "Limiting the impact of stealthy attacks on industrial control systems," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 1092–1105.

[7] A. A. Cárdenas, J. S. Baras, and K. Seamon, "A framework for the evaluation of intrusion detection systems," in *2006 IEEE Symposium on Security and Privacy (S&P'06)*. IEEE, 2006, pp. 15–pp.

[8] M. N. Al-Mhiqani, R. Ahmad, W. Yassin, A. Hassan, Z. Z. Abidin, N. S. Ali, and K. H. Abdulkareem, "Cyber-security incidents: a review cases in cyber-physical systems," *Int. J. Adv. Comput. Sci. Appl*, no. 1, pp. 499–508, 2018.

[9] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[10] L. Wang, D. Han, and M. Egerstedt, "Permissive barrier certificates for safe stabilization using sum-of-squares," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 585–590.

[11] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier-value functions for safety-critical control," *arXiv preprint arXiv:2104.02808*, 2021.

[12] J. Giraldo, S. H. Kafash, J. Ruths, and A. A. Cardenas, "Daria: Designing actuators to resist arbitrary attacks against cyber-physical systems," in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2020, pp. 339–353.

[13] U. Erlingsson, *The inlined reference monitor approach to security policy enforcement*. Cornell University, 2004.

[14] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE transactions on automatic control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[15] S. H. Kafash, J. Giraldo, C. Murguia, A. A. Cardenas, and J. Ruths, "Constraining attacker capabilities through actuator saturation," in *2018 Annual American Control Conference*. IEEE, 2018, pp. 986–991.

[16] F. H. Clarke, Y. S. Ledyaev, R. J. Stern, and P. R. Wolenski, *Nonsmooth analysis and control theory*. Springer Science & Business Media, 2008, vol. 178.

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

[18] K. Garg and D. Panagou, "Robust control barrier and control lyapunov functions with fixed-time convergence guarantees," in *2021 American Control Conference (ACC)*, 2021, pp. 2292–2297.

[19] S. Oudot and J.-D. Boissonnat, "Provably good surface sampling and approximation." in *Symposium on Geometry Processing*, 2003, pp. 9–18.

[20] P. Leopardi, "Diameter bounds for equal area partitions of the unit sphere," *Electron. Trans. Numer. Anal*, vol. 35, pp. 1–16, 2009.

[21] M. De Berg, M. Van Kreveld, M. Overmars, and O. Schwarzkopf, "Computational geometry," in *Computational geometry*. Springer, 1997, pp. 1–17.

[22] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances," in *IEEE 56th Conference on Decision and Control*. IEEE, 2017, pp. 2242–2253.

[23] K. Garg, E. Arabi, and D. Panagou, "Fixed-time control under spatiotemporal and input constraints: A QP based approach," *arXiv preprint arXiv:1906.10091*, 2019.

[24] A. Semechko, "Suite of functions to perform uniform sampling of a sphere," gitHub. Retrieved August 24, 2021. [Online]. Available: https://github.com/AntonSemechko/S2-Sampling-Toolbox