Control Barrier Function-Based Attack-Recovery with Provable Guarantees

Kunal Garg, Ricardo G. Sanfelice, and Alvaro A. Cardenas

***Abstract*— This paper studies provable security guarantees for cyber-physical systems (CPS) under actuator attacks. Specifically, we consider safety for CPS and propose a new attack-detection mechanism based on a zeroing control barrier function (ZCBF) condition. To reduce the conservatism in its implementation, we design an adaptive recovery mechanism based on how close the state is to violating safety. We show that the attack-detection mechanism is sound, i.e., there are no false negatives for adversarial attacks. Finally, we use a Quadratic Programming (QP) approach for online recovery (and nominal) control synthesis. We demonstrate the effectiveness of the proposed method in a case study involving a quadrotor with an attack on its motors.**

I. Introduction

Cyber-physical systems (CPS) such as autonomous and semi-autonomous air, ground, and space vehicles must maintain their safe operation and achieve mission objectives under various adversarial environments, including cyber-attacks. Security measures can be classified into two types of mechanisms [1]: i) proactive, which considers design choices deployed in the CPS *before* attacks, and ii) reactive, which takes effect after an attack is detected. A proactive method, which considers design choices deployed in the CPS *before* attacks, can result in a conservative design. On the other hand, reactive methods, which take effect after an attack is detected, heavily rely on fast and accurate attack-detection mechanisms. An optimal approach to achieving resilience against cyber attacks must utilize the benefits of the two approaches while minimizing their limitations.

There is a plethora of work on attack detection for CPS, see, e.g., [2]–[5]. However, as discussed in [6], a knowledgeable attacker can design stealthy attacks that can disrupt the nominal system behavior slowly in order to evade these detection mechanisms. Such methods can lead to system failure by pushing the system out of its safe operating limits. Thus, a new attack-detection mechanism must be devised based on the closeness of the system to violating safety. *Safety*, i.e., the state of the system does not leave a safe zone, is an essential requirement, violation of which can result in significant (financial and performance) losses, particularly when a system is under attack [7]. In most practical problems involving CPS, safety can be realized as guaranteeing forward invariance of a safe set. One of the most common approaches to guaranteeing that system trajectories stay in a safe set is based on employing a control barrier function (CBF), as it allows a real-time implementable quadratic programming (QP)-based control synthesis framework [8].

In our prior work [9], we use a proactive scheme consisting of only designing a safe feedback law using a CBF. One disadvantage of that approach is that the resulting controller is conservative due to the system possibly being not under attack and still assumed to be under attack. In contrast, this paper proposes a reactive security mechanism that activates a potentially conservative controller only after an attack is detected. In particular, we consider actuator attacks, where an attacker can assign *arbitrary* values to the input signals for a subset of the actuators. Furthermore, we allow multiple attacks on the system and provide conditions for guaranteed safety under repeated attacks on system actuators.

In this paper, we consider the safety property with respect to an unsafe set and propose an attack-detection mechanism based on CBF conditions for safety. We use an adaptive parameter based on how close the system is to violating the safety requirement and use this adaptive parameter in the attack detection to reduce conservatism. Using this detection mechanism, we propose a switching-based strategy for recovery from a *nominal* feedback law (to be used when there is no attack) to a *safe* feedback law, when the system is under an adversarial attack. While there is work on CBF-based safety of CPS under faults and attacks [9]–[11], to the best of the authors' knowledge, this is the first work utilizing CBF conditions for attack detection.

Notation: Throughout the paper, $\mathbb{R}$ denotes the set of real numbers, and $\mathbb{R}_{\geq 0}$ 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. 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 $L_f h(x) \coloneqq \frac{\partial h}{\partial x}(x) f(x)$.

II. Problem formulation

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 that is continuous on $\mathcal{D} \times \mathcal{U}$, with $\mathcal{D} \subset \mathbb{R}^n$ and $\mathcal{U} \subset \mathbb{R}^m$, $d : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \to \mathbb{R}^n$ is unknown and represents the unmodeled dynamics, $x \in \mathcal{D}$ is the system state, and $u \in \mathcal{U}$ is the control

Research partially supported by NSF Grants no. ECS-1710621, CNS-2039054, and CNS-2111688, by AFOSR Grants no. FA9550-19-1-0053, FA9550-19-1-0169, and FA9550-20-1-0238, and by ARO Grant no. W911NF-20-1-0253.

K. Garg and R.G. Sanfelice are with the Department of Electrical and Computer Engineering, and A.A. Cardenas is with the Department of Computer Science and Engineering, University of California, Santa Cruz, CA, 95064, USA e-mail(s): kgarg@umich.edu, {ricardo, alacarde}@ucsc.edu.

input.[1] Similar to [9], in this paper, 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 $\mathcal{U} \coloneqq \mathcal{U}_v \times \mathcal{U}_s$. Under this class of attack, we assume that we know which components of the control input are vulnerable. Under this attack model, the input to the system takes the form

$$u(t,x) = \begin{cases} (\lambda_v(x), \lambda_s(x)) & \text{if } t \notin \mathcal{T}_a; \\ (u_a(t), k_s(x)) & \text{if } t \in \mathcal{T}_a; \end{cases}, \tag{3}$$

where $u_a : \mathbb{R}_{\geq 0} \to \mathcal{U}_v$ is the attack signal on the input u_v, $k_s : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \to \mathbb{R}^{m_s}$ is a *safe* feedback law for the input u_s, to be designed and used when the system is under attack, and the pair $\lambda_v : \mathbb{R}^n \to \mathcal{U}_v, \lambda_s : \mathbb{R}^n \to \mathcal{U}_s$ define the *nominal* feedback law $\lambda = (\lambda_v, \lambda_s)$, to be designed and used when there is no attack. The set $\mathcal{T}_a \subset \mathbb{R}_{\geq 0}$ is the set of time intervals over which an attack is launched on the system input. In particular, for each $i \geq 1$, let $[t_1^i, t_2^i)$ with $t_2^i \geq t_1^i$ denote the interval of time when the attack is launched for the $i-$th time where $t_1^1 \geq 0$, so that $\mathcal{T}_a \coloneqq \bigcup_{i \geq 0} [t_1^i, t_2^i)$. Define

$$\overline{T} \coloneqq \max_{i \geq 1} \{t_2^i - t_1^i\}, \quad T_{na} \coloneqq \min_{i \geq 2} \{t_1^i - t_2^{(i-1)}\}, \tag{4}$$

as the maximum length of the attack and the minimum length of the interval without an attack on the system input, respectively. In this work, we assume that the set $\mathcal{T}_a$ is unknown, and only the maximum period of attack, $\overline{T}$, and minimum period without an attack, T_{na}, are known.

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

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

Problem 1. *Consider the system in (1) with unmodeled dynamics d that satisfies Assumption 1, a set S and the attack model in (2). Design an attack-detection mechanism and a safe input assignment policy such that, for a set of initial conditions $X_0 \subset S$ and attack signals $u_a : \mathbb{R}_{\geq 0} \to \mathcal{U}_v$, the closed-loop trajectories $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ of (1) resulting from applying the designed input policy satisfy $x(t) \in S$ for all $t \geq 0$ and for all $x(0) \in X_0$.*

Note that for the safety requirement as imposed in Problem 1, an attack is adversarial only if it can push the system trajectories out of the set S, as defined below.

[1]It is assumed that each solution x is an absolutely continuous function in time.

Definition 1. *An attack signal $u_a : \mathbb{R}_{\geq 0} \to \mathcal{U}_v$ is* adversarial *if there exist $x(0) \in S$ and a finite $t \geq 0$ such that for any $\kappa : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \to \mathcal{U}_s$, each system trajectory $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ of (1) resulting from applying $u = (u_a, \kappa)$ satisfies $x(t) \notin S$.*

Per the above definition, it is possible that there is an attack on the system but the system does not violate the safety requirement. We use this observation to focus our detection mechanism only on the attacks that can potentially push the system out of the safe set.

We first review the notion of forward invariance of the set S and the corresponding barrier function conditions. In the remainder of the paper, to keep the presentation simple, we assume that the maximal solutions of (1) exists and are unique.

Definition 2. *A set $S \subset \mathbb{R}^n$ is termed as* forward invariant *for system (1) if every solution $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ of (1) satisfies $x(t) \in S$ for all $t \geq 0$ and for all initial conditions $x(0) \in S$.*

Next, we review a sufficient condition for guaranteeing forward invariance of a set without an attack. Following the notion of robust CBF in [12], we can state the following result guaranteeing forward invariance of the set S for the system (1).

Lemma 1 ([12])**.** *Given a continuously differentiable function $B : \mathbb{R}^n \to \mathbb{R}$, the set $S = \{x \mid B(x) \leq 0\}$ is forward invariant for (1) under d satisfying Assumption 1 for some $\delta > 0$, if the following condition holds:*

$$\inf_{u \in \mathcal{U}} L_F B(x,u) \leq -l_B \delta \quad \forall x \in \partial S, \tag{5}$$

where l_B is the Lipschitz constant of the function B.

III. Attack Detection

In this section, we present a method for detecting whether the system (1) is under attack using the barrier function condition (5). In particular, we check whether inequality (5) holds on the boundary of the safe set to raise a flag for an attack. Instead of using the value of the barrier function B, we use the value of its time derivative due to the following reason. The time derivative of the function B on the boundary of the safe set indicates whether the system will violate the safety constraint. Moreover, the time derivative of the function B includes the system dynamics. Hence, it is a better indicator of whether the given system will violate the given safety constraint than the function B itself, which does not capture the system information.

Given B and F, define

$$H(x,u) \coloneqq L_F B(x,u) + l_B \delta, \tag{6}$$

where δ is the bound on the disturbance d per Assumption 1 and l_B is the Lipschitz constant of the function B. Note that condition (5) can be written as $\inf_{u \in \mathcal{U}} H(x,u) \leq 0$ for all $x \in \partial S$. If an attack signal u_a is adversarial, then it holds that there exists a finite time $t \geq 0$ such that $x(t) \in \partial S$ and $H(x(t), (u_a(t), v_s)) > 0$ for any $v_s \in U_s$ where $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is the solution of (1) resulting from the input (3). Using

this property, a detection mechanism can be devised to flag that the system input is under attack. In particular, if the input u to the system is known at time t when $x(t) \in \partial S$, an attack detection mechanism can be designed by checking the value of $H(x(t), u(t, x(t)))$. However, in the presence of an attack with a delay t_d in detection, it is not possible to know the actual input u to the system. Thus, it is not possible to use the evaluation of H to flag an attack.

To resolve this issue, we note that the function B only depends on the state x. Thus, we use an approximation method to estimate the value of H at any given time t using the consecutive measurements of the function B at time $(t-\tau)$ and t, for some $\tau > 0$. For each $t \geq 0$, define $e_B : \mathbb{R}_{\geq 0} \to \mathbb{R}$

$$e_B(t) \coloneqq \left| \dot{B}(x(t)) - \frac{B(x(t)) - B(x(t-\tau))}{\tau} \right|,$$

as the error between the actual time derivative of the function B and its first-order approximation where $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is the solution of (1). Assume that the function B is twice continuously differentiable, and the function F in (1) is continuously differentiable. Under these conditions, using Taylor's theorem for the first-order approximation of the function B with a second-order error term, there exists $0 \leq \bar{t} \leq \tau$ such that

$$B(x(t-\tau)) = B(x(t)) - \dot{B}(x(t))\tau + \ddot{B}(x(\bar{t}))\frac{\tau^2}{2},$$

where $\ddot{B}$ is the second time derivative of the function B. Assume that $|\ddot{B}(x)| \leq \eta$ for some $\eta > 0$ for all $x \in S_c$. For the sake of brevity, denote $\hat{\dot{B}}(x(t), \tau) = \frac{B(x(t)) - B(x(t-\tau))}{\tau}$ so that we have

$$e_B(t) = |\dot{B}(x(t)) - \hat{\dot{B}}(x(t), \tau)| = \frac{\tau}{2}|\ddot{B}(x(\bar{t}))| \leq \frac{\eta\tau}{2}.$$

Using the bound on e_B, we obtain that for each $t \geq 0$ and $\tau \geq 0$, the following holds:

$$\hat{\dot{B}}(x(t), \tau) - \frac{\eta\tau}{2} \leq H(x(t), u(t)) \leq \hat{\dot{B}}(x(t), \tau) + \frac{\eta\tau}{2}, \quad (7)$$

where $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is the solution of (1) resulting from applying the input $u : \mathbb{R}_{\geq 0} \to \mathcal{U}$. Using (7), it holds that for each $t \geq 0$ and $\tau \geq 0$,

$$\hat{\dot{B}}(x(t), \tau) + \frac{\eta\tau}{2} \leq 0 \implies H(x(t), u(t)) \leq 0.$$

With the above construction, we define the time when a flag for an attack is raised as

$$\hat{t}_d = \inf \left\{ t \;\middle|\; \hat{\dot{B}}(x(t), \tau) > -\frac{\eta\tau}{2}, x(t) \in \partial S \right\}, \quad (8)$$

where η is the bound on the second time-derivative $\ddot{B}$ and $\tau > 0$. We have the following result stating that the attack detection mechanism in (8) detects the attack before the system trajectories leave the safe set.

Lemma 2. *Given a twice continuously differentiable function B, system (1) with d satisfying Assumption 1, a continuously differentiable function F, and an adversarial attack starting at $t = t_1^i$, let $T \geq t_1^i$ be defined as*

$$T = \inf\{t \geq t_1^i \mid H(x(t), u(t, x(t))) > 0, x(t) \in \partial S\}, \quad (9)$$

where $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is the solution of (1) resulting from applying the input $u : \mathbb{R}_{\geq 0} \to \mathcal{U}$ and η is the bound on the second time-derivative $\ddot{B}$. Then, for each $\tau \geq 0$, it holds that $\hat{t}_d \leq T$, where $\hat{t}_d$ is given in (8).

Lemma 2 implies that the attack-detection mechanism in (8) raises an alert on or before the system trajectories reach the boundary of the set ∂S under an attack. In other words, while the detection mechanism (8) can have false positives (i.e., raise an alert when there is no attack), it will never have a false negative (i.e., it will not miss any attack).

For a given c, define

$$S_c \coloneqq \{x \mid B(x) \leq -c\}. \quad (10)$$

One way to make the detection method robust is to check the inequality at the boundary of the set S_c. Define $c_M \in \mathbb{R}$ as

$$c_M \coloneqq -\min_{x \in S} B(x), \quad (11)$$

so that the set S_c is nonempty for all $c \in [0, c_M)$. Now, since it is possible to allow the function H to take positive values in the interior of the safe set S, we use the inequality $H(x, u) \leq \gamma$ for some $\gamma > 0$ instead of $H(x, u) \leq 0$, to detect attacks. Note that a constant $\gamma > 0$ might lead to false positives if γ is too small, or false negatives if γ is too large. To this end, we make the following assumption when the system is not under attack.

Assumption 2. *There exist $\bar{c} \in (0, c_M)$, $\bar{\delta} \in \mathbb{R}$ and a continuous input $\bar{u} : \mathbb{R}^n \to \mathcal{U}$ such that the following inequality holds: $H(x, \bar{u}(x)) \leq -\bar{\delta}B(x)$, for all $x \in S \setminus \text{int}(S_{\bar{c}})$.*

Similar assumptions have been made in the literature on safety using ZCBFs (see e.g. [8]). Note that under Assumption 2, using the comparison lemma, it can be shown that $H(x(t), \bar{u}(x(t))) \coloneqq \dot{B}(x(t)) \leq -\bar{\delta}B(x(t))$ which implies that $B(x(t)) \leq B(x(\bar{t}))e^{-\bar{\delta}(t-\bar{t})}$ for all $t \geq \bar{t}$, where $\bar{t} = \inf\{t \mid x(t) \in \partial S_{\bar{c}}\}$ and $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is the solution of (1) resulting from applying the continuous input $\bar{u}$. Using Assumption 2, we design an adaptive scheme for the parameter γ.

Let $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be an adaptive parameter whose adaptation law is given as $\gamma(t) = \bar{\delta}\bar{c}e^{-\bar{\delta}(t-\bar{t})}$ for $t \geq \bar{t}$, where $\bar{\delta}, \bar{c}$ are as defined in Assumption 2. Note that under Assumption 2, there exists a feedback law $\bar{u} : \mathbb{R}^n \to \mathcal{U}$ such that $H(x(t), \bar{u}(x(t))) \leq \gamma(t)$ for all $t \geq \bar{t}$. Using this observation, we propose a new attack-detection mechanism that raises a flag for the $i-$th time at $t = \hat{t}_d^i$, where

$$\hat{t}_d^i = \inf \left\{ t \geq \max\left\{\bar{t}, \hat{t}_d^{(i-1)}\right\} \;\middle|\; \hat{\dot{B}}(x(t), \tau) > \gamma(t) - \frac{\eta\tau}{2}, \right.$$
$$\left. x(t) \in S \setminus \text{int}(S_{\bar{c}}) \right\}, \quad (12)$$

where η is the bound on the second time-derivative $\ddot{B}$, $\hat{t}_d^0 = -\overline{T}$ and $\tau > 0$.

Remark 1. *Compared to* (8)*, the detection mechanism in* (12) *raises a flag when* $\hat{\dot{B}}(x(t),\tau) > \gamma(t) - \frac{\eta\tau}{2}$ *anywhere in the set* $S \setminus \text{int}(S_{\bar{c}})$*, which is a strip of non-zero measure between the boundaries* $\partial S_{\bar{c}}$ *and* ∂S*. This, along with the adaptive parameter* γ*, provides inherent robustness against small perturbations and measurement uncertainties. As a result, under an attack, the proposed detection mechanism allows the system to get close to the boundary of the safe set as long as the rate at which the system approaches the boundary (dictated by* $\dot{B}$*) is bounded according to Assumption 2.*

Remark 2. *It is worth noting that the proposed detected mechanism focuses on detecting only* **adversarial** *attacks, and not every attack. That is, if there is an attack signal* u_a *such that* $\inf_{u_s \in \mathcal{U}_s} H(x,(u_a,u_s)) \leq 0$ *for* $x \in \partial S$*, i.e., the system is still safe under the attack, the proposed detection mechanism will not detect it.*

IV. Recovery Controller Design

In this section, we present a switching-based control assignment to recover from an adversarial attack based on the detection mechanism from the previous section. To this end, we make the following assumption.

Assumption 3. *There exists* $\bar{c} \in (0, c_M)$ *such that the following hold:*

$$\inf_{u_s \in \mathcal{U}_s} \sup_{u_a \in \mathcal{U}_a} H(x,(u_a,u_s)) \leq 0 \quad \forall\, x \in S \setminus \text{int}(S_{\bar{c}}). \tag{13}$$

The above assumption implies that the set S_c can be rendered forward invariant under any attack $u_a \in \mathcal{U}_a$. Now, consider a time-interval $[t_2^{(i-1)}, t_1^i)$ over which the system input *is not* under an attack and the interval $t \in [t_1^i, t_2^i)$ over which it *is* under an attack. Define $\mathcal{T}_d := \bigcup [\hat{t}_d^j, \hat{t}_d^j + \overline{T})$ as the set of time intervals when an attack is flagged, where $\hat{t}_d^j$ is the time when the attack is detected for the $j-$th time according to (12), $j \geq 0$ with $\hat{t}_d^0 = -\overline{T}$. Due to $\mathcal{T}_a$ being unknown, the system input is defined as

$$u(t,x) = (u_v(t,x), u_s(t,x)), \tag{14a}$$

$$u_v(t,x) = \begin{cases} \lambda_v(x) & \text{if} \quad t \notin \mathcal{T}_a, \\ u_a(t) & \text{if} \quad t \in \mathcal{T}_a, \end{cases} \tag{14b}$$

$$u_s(t,x) = \begin{cases} \lambda_s(x) & \text{if} \quad t \notin \mathcal{T}_d, \\ k_s(x) & \text{if} \quad t \in \mathcal{T}_d, \end{cases} \tag{14c}$$

where (λ_v, λ_s) constitute the nominal input when there is no attack detected, u_a is the attack signal and u_s is the safe input under attack (see (3)).

We have the following result showing the existence of nominal and safe feedback laws for (14) that can recover the system from an attack.

Theorem 1. *Given system* (1) *with* $F \in \mathcal{C}^1$*,* $B \in \mathcal{C}^2$*, and the attack model* (2)*, suppose that Assumption 1 holds, and that Assumptions 2-3 hold for some* $\bar{c} \in (0, c_M)$*. Then, there exist feedback laws* $\lambda = (\lambda_v, \lambda_s) : \mathbb{R}^n \to \mathcal{U}$ *and* $k_s : \mathbb{R}^n \to \mathcal{U}_s$ *such that under the effect of the input* u *in* (14) *with* $\hat{t}_d^j$ *is defined in* (12)*, the system trajectories of* (1) *satisfy* $x(t) \in S$ *for all* $t \geq 0$ *and for all* $x(0) \in X_0 = \text{int}(S)$*.*

In essence, Theorem 1 provides sufficient conditions for the existence of a control algorithm such that Problem 1 can be solved. While Assumptions 2 and 3 serve different purposes, it is easy to see that the satisfaction of Assumption 3 for some $\bar{c} \in (0, c_M)$ implies that Assumption 2 holds for the same $\bar{c}$. Thus, it is sufficient to verify that Assumption 3 holds. One practical method of finding a subset of the safe set S, where Assumption 3 holds, is the computationally efficient sampling-based method proposed in [9].

Next, we present a control syntheses method to design both the nominal feedback λ and the safe recovery feedback-law k_s in (14). To formulate a tractable optimization problem for control synthesis, we assume that the system (1) is control-affine and is of the form

$$\dot{x} = f(x) + g(x)u + d(t,x), \tag{15}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are continuous functions. Assume that the input constraint set $\mathcal{U}$ is given as $\mathcal{U} = \{u \mid Au \leq b\}$, for some $A : \mathbb{R}^{p \times m}$ and $b \in \mathbb{R}^p$. To synthesize the nominal feedback law λ, we formulate the following QP for each $x \in S$,

$$\min_{(v,\eta)} \quad \frac{1}{2}|v|^2 + \frac{1}{2}\eta^2 \tag{16a}$$

$$\text{s.t.} \quad Av \leq b, \tag{16b}$$

$$L_f B(x) + L_g B(x) v \leq -\eta B(x) - l_B \delta, \tag{16c}$$

where $q > 0$ is a constant and l_B is the Lipschitz constants of the function B. To compute the safe feedback-law k_s, let $g = [g_s \; g_v]$ with $g_s : \mathbb{R}^n \to \mathbb{R}^{n \times m_s}$, $g_v : \mathbb{R}^n \to \mathbb{R}^{n \times m_v}$, and assume that the input constraint set for u_s is given as $\mathcal{U}_s = \{u_s \mid A_s u_s \leq b_s\}$ for some A_s and b_s. The QP for the synthesis of k_s is as follows for each $x \in S \setminus \text{int}(S_{\bar{c}})$,

$$\min_{(v_s,\zeta)} \quad \frac{1}{2}|v_s|^2 + \frac{1}{2}\zeta^2 \tag{17a}$$

$$\text{s.t.} \quad A_s v_s \leq b_s, \tag{17b}$$

$$L_f B(x) + L_{g_s} B(x) v_s \leq -\zeta B(x) - l_B \delta - \sup_{u_v \in \mathcal{U}_v} L_{g_v} B(x) u_v, \tag{17c}$$

Let the solution of the QP (16) be denoted as (v^*, η^*) and that of (17) as (v_s^*, ζ^*). We are now ready to state the following result, based on results in [13].

Theorem 2. *Given the functions* F, d, B *and the attack model* (2)*, suppose Assumptions 1-3 hold with* $\bar{\delta} > 0$ *and* $\bar{c} \in (0, c_M)$*. Assume that the strict complementary slackness*[2] *holds for the QPs* (16) *and* (17) *for all* $x \in S$ *and* $x \in S \setminus \text{int}(S_c)$*, respectively. Then, the QPs* (16) *and* (17) *are feasible for all* $x \in S$ *and all* $x \in S \setminus \text{int}(S_c)$*, respectively;*

[2]In brief, if the $i-$the constraint of (16), with $i \in \{1,2\}$, is written as $G_i(x,z) \leq 0$ with $z = (v,\eta)$, 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.

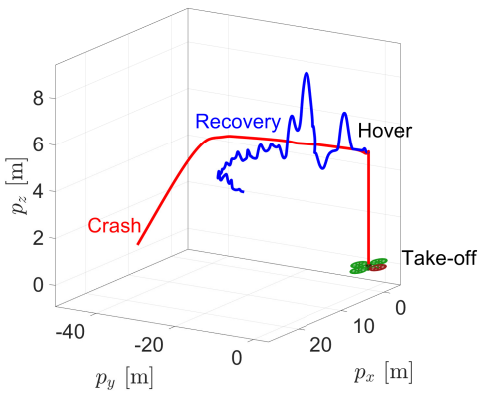

Fig. 1. The closed-loop path traced by the quadrotor with the proposed detection mechanism (in blue) and without the detection mechanism (in red). The vulnerable motor is shown in red.

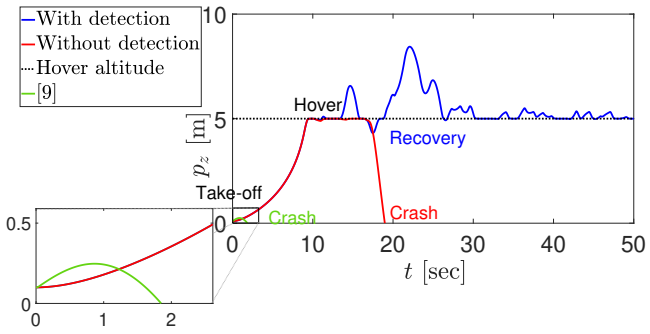

Fig. 2. In the absence of the detection mechanism, the quadrotor crashes (i.e., $z = 0$ m). In the presence of the detection mechanism, the altitude remains close to the desired altitude $z = 5m$ (shown by a black line). The conservative approach in [9], resulting in a crash even without an attack, is shown in green (see the inset plot).

v^* and v_s^* are continuous on $\text{int}(S)$ *and* $x \in \text{int}(S \setminus \text{int}(S_c))$*; and the control input defined in* (14) *with* $\lambda(x) = v^*(x)$*,* $k_s(x) = v_s^*(x)$*, and* $t_d = \hat{t}_d$*, where* $\hat{t}_d$ *is defined in* (12)*, solves Problem 1 for all* $x(0) \in \text{int}(S)$*.*

Thus, the QPs (16) and (17) can be used to synthesize a safe input for a system under attack.

V. Quadrotor Case Study

We consider a simulation case study involving a quadrotor with an attack on one of its motors.[3] The quadrotor dynamics is given as (see [14]):

$$\ddot{p}_x = \frac{1}{m}\Big(\big(c(\phi)c(\psi)s(\theta) + s(\phi)s(\psi)\big)u_f - k_t\dot{p}_x\Big) \tag{18a}$$

$$\ddot{p}_y = \frac{1}{m}\Big(\big(c(\phi)s(\psi)s(\theta) - s(\phi)c(\psi)\big)u_f - k_t\dot{p}_y\Big) \tag{18b}$$

$$\ddot{p}_z = \frac{1}{m}\Big(c(\theta)c(\phi)u_f - mg - k_t\dot{p}_z\Big) \tag{18c}$$

$$\dot{\phi} = p + qs(\phi)t(\theta) + rc(\phi)t(\theta) \tag{18d}$$

$$\dot{\theta} = qc(\phi) - rs(\phi) \tag{18e}$$

$$\dot{\psi} = \frac{1}{c(\theta)}\big(qs(\phi) + rc(\phi)\big) \tag{18f}$$

$$\dot{p} = \frac{1}{I_{xx}}\Big(-k_r p - qr(I_{zz} - I_{yy}) + \tau_p\Big) \tag{18g}$$

$$\dot{q} = \frac{1}{I_{yy}}\big(-k_r q - pr(I_{xx} - I_{zz}) + \tau_q\big) \tag{18h}$$

$$\dot{r} = \frac{1}{I_{zz}}\big(-k_r r - pq(I_{yy} - I_{zz}) + \tau_r\big), \tag{18i}$$

where $m, I_{xx}, I_{yy}, I_{zz}, k_r, k_t > 0$ are system parameters, $g = 9.8$ is the gravitational acceleration, $c(\cdot), s(\cdot), t(\cdot)$ denote $\cos(\cdot), \sin(\cdot), \tan(\cdot)$, respectively, (p_x, p_y, p_z) denote the position of the quadrotor, (ϕ, θ, ψ) its Euler angles and $u = (u_f, \tau_p, \tau_q, \tau_r)$ the input vector consisting of thrust u_f and moments τ_p, τ_q, τ_r. The relation between the vector u and the individual motor thrusts is given as

$$\begin{bmatrix} u_f \\ \tau_p \\ \tau_q \\ \tau_r \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -l & 0 & l \\ -l & 0 & l & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}, \tag{19}$$

where f_i is the thrust generated by the $i-$th motor for $i \in \{1, 2, 3, 4\}$, $d, l > 0$ are system parameters. We choose the system parameters for simulations as: $I_{xx} = I_{yy} = 0.177$ kg-m^2, $I_{zz} = 0.344$ kg-m^2, $m = 4.493$ kg, $l = 0.1$ m, $d = 0.0024$ m, $k_t = 1$ and $k_r = 1.5$ (see [15]). Furthermore, we consider the bound on each motor given as $|f_i| \leq 27.7$ N for $i \in \{1, 2, 3, 4\}$. We use $\tau = 10^{-3}$. Without loss of generality, we assume that motor #4 is vulnerable. Under an attack on motor #4, it is not possible to keep all the inputs $(u_f, \tau_p, \tau_q, \tau_r)$ close to its desired value simultaneously under an attack on motor #4. Thus, we focus on designing a control law to maintain the desired altitude of the quadrotor (through u_f) and minimize its oscillations (through (τ_p, τ_q)). It implies that τ_r will not be matched with its desired value to control the yaw angle ψ, resulting in an uncontrolled yaw angle increase.

We choose the control objective to make the quadrotor hover at location $(0, 0, 5)$, starting from $(0, 0, 0.2)$. Based on the above observation and the fact that ψ does not contribute to changing the altitude of the quadrotor, the safety constraints are to keep the angles (ϕ, θ) in a given bounded

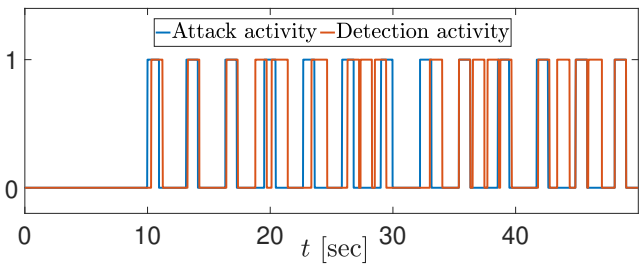

Fig. 3. The attack (respectively, the detection) activity where 1 denotes that the attack is active (respectively, flagged) and 0, that the attack is non-active (respectively, not flagged).

[3]The Matlab code is available at: https://github.com/HybridSystemsLab/QuadrotorCBFAttackRecovery.git
A video of the simulation is available at https://tinyurl.com/ye28ksx3. The authors would like to thank Dr. Adeel Akhtar for providing the MATLAB files for the 3D visualization of the quadrotor in the video.

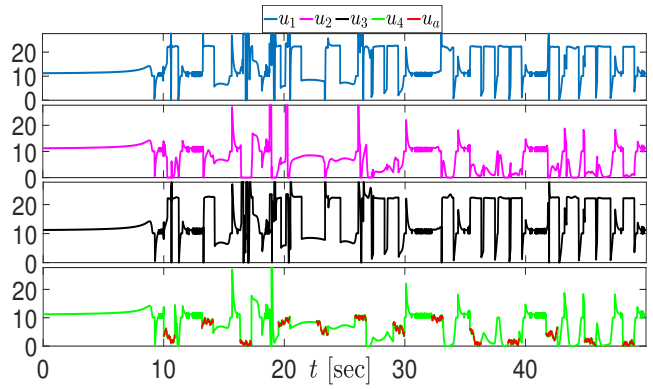

Fig. 4. Thrust f_i of each motor. The thrust of motor 4 under attack is shown in red. The switch in the rest of the motors is seen when an attack is flagged.

range, i.e., $|\phi| \leq \phi_M, |\theta| \leq \theta_M$, for some $\phi_M, \theta_M > 0$, and to keep the quadrotor above the ground, i.e., $z > 0$. Thus, the safe set is defined as $S = \Big\{(\phi, \theta, z) \mid |\phi| \leq \phi_M, |\theta| \leq \theta_M, z \leq -\epsilon\Big\}$. We choose $\phi_M = \theta_M = 0.3$ and $\epsilon = 0.02$. The maximum length of the attack is randomly chosen as $\overline{T} = 0.934$ seconds and the period of no attack is chosen as $T_{na} = 2.238$ seconds.

The barrier functions used for enforcing safety are $B_1(z) = -z + 0.02$, $B_2(\phi) = |\phi|^2 - \phi_M^2$ and $B_3(\theta) = |\theta|^2 - \theta_M^2$. The parameters $\bar{\delta}, \bar{c}$ for detection are $\bar{\delta} = 0.1, \bar{c} = \frac{1}{4}(0.3)^2$. Figure 1 shows the closed-loop path traced by the quadrotor. Figure 2 plots the position coordinates (x, y, z). The safety constraint $z \leq 0$ is satisfied at all times, and the quadrotor can hover at an altitude $z = 5$ m. Figure 3 shows the attack and the detection signal. It can be seen that detection has a non-zero delay during some attacks, and zero delay during some attacks. It can also be seen that some of the attacks are not detected, as they do not fall into the category of *adversarial* attack per Definition 1. The attack is flagged according to (12) and remains flagged for the duration $\overline{T}$. The bound $|f_i| \leq 27.7$ N is satisfied for each motor at all times. The vulnerable motor is highlighted in green. Finally, Figure 4 plots the thrust for each motor under nominal conditions as well as under attack.

Thus, the proposed scheme can successfully detect an attack on a quadrotor motor before the quadrotor crashes. Furthermore, the designed safe input can keep the quadrotor in the safe zone even under attack, thus demonstrating a successful recovery after detection. The conservative approach in [9], which assumes that the rotor #4 is constantly under attack, fails to keep the quadrotor from crashing even when there is no attack (see Figure 2). In contrast, the proposed approach is non-conservative and reacts to an adversarial attack, thereby not interfering with the system's nominal functionality.

VI. Conclusions

We presented a novel attack-detection scheme based on the control Barrier function. Our formulation is adaptive, in the sense that the further away the system is from violating safety our recovery controller focuses on performance rather than safety; however, if the system keeps approaching the safety limit, our adaptive mechanism switches to a recovery controller to counteract the potential attack. We demonstrated the efficacy of the proposed method on a simulation example involving an attack on a quadrotor motor.

Future work involves studying more general attacks on CPS, such as attacks on system sensors and simultaneous attacks on system sensors and actuators. As noted in Remark 1, our future investigation also includes studying methods of estimating the time when the attack has stopped.

References

[1] 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

[2] Y. Chen, C. M. Poskitt, and J. Sun, "Learning from mutants: Using code mutation to learn and monitor invariants of a cyber-physical system," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 648–660.

[3] 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*, ser. CCS '18. New York, NY, USA: ACM, 2018, pp. 801–816.

[4] C. Feng, V. R. Palleti, A. Mathur, and D. Chana, "A systematic framework to generate invariants for anomaly detection in industrial control systems," in *2019 Network and Distributed System Security Symposium (NDSS)*.

[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] 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.

[8] 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.

[9] K. Garg, R. G. Sanfelice, and A. A. Cardenas, "Sampling based computation of viability domain to prevent safety violations by attackers," in *IEEE Conference on Control Technology and Applications*, 2022.

[10] A. Clark, Z. Li, and H. Zhang, "Control barrier functions for safe cps under sensor faults and attacks," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 796–803.

[11] B. Ramasubramanian, L. Niu, A. Clark, L. Bushnell, and R. Poovendran, "Linear temporal logic satisfaction in adversarial environments using secure control barrier certificates," in *International Conference on Decision and Game Theory for Security*. Springer, 2019, pp. 385–403.

[12] 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.

[13] K. Garg, E. Arabi, and D. Panagou, "Fixed-time control under spatiotemporal and input constraints: A quadratic programming based approach," *Automatica*, vol. 141, p. 110314, 2022.

[14] A. Lanzon, A. Freddi, and S. Longhi, "Flight control of a quadrotor vehicle subsequent to a rotor failure," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 2, pp. 580–591, 2014.

[15] A. Akhtar, S. L. Waslander, and C. Nielsen, "Fault tolerant path following for a quadrotor," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 847–852.