An Analytical Framework for Control Synthesis of Cyber-Physical Systems with Safety Guarantee

Luyao Niu[1*], Abdullah Al Maruf[2*], Andrew Clark[1], J. Sukarno Mertoguno[3], and Radha Poovendran[2]

***Abstract*— Cyber-physical systems (CPS) are required to operate safely under fault and malicious attacks. The simplex architecture and the recently proposed cyber resilient architectures, e.g., Byzantine fault tolerant++ (BFT++), provide safety for CPS under faults and malicious cyber attacks, respectively. However, these existing architectures make use of different timing parameters and implementations to provide safety, and are seemingly unrelated. In this paper, we propose an analytical framework to represent the simplex, BFT++ and other practical cyber resilient architectures (CRAs). We construct a hybrid system that models CPS adopting any of these architectures. We derive sufficient conditions via our proposed framework under which a control policy is guaranteed to be safe. We present an algorithm to synthesize the control policy. We validate the proposed framework using a case study on lateral control of a Boeing 747, and demonstrate that our proposed approach ensures safety of the system.**

I. Introduction

Cyber-physical systems (CPS) are subject to random failures and malicious cyber attacks, which have been reported in applications such as transportation [1] and power system [2]. Failures and attacks can potentially cause safety violation of the physical components, which leads to severe harm to the plants and humans.

Fault tolerant control schemes [3]–[5] and architectures such as simplex [6]–[8] have been proposed to address random failures. These approaches are effective in CPS when some components are verified to be fault-free. This requirement, however, may not be viable for all CPS, especially those subject to malicious attacks.

A malicious adversary can exploit the vulnerabilities in the cyber subsystem and intrude into CPS. The adversary can then cause common mode failures across different components, rendering fault-tolerant schemes designed for random failures inadequate. A seminal work recently proposed a cyber resilient architecture, named Byzantine fault tolerant++ (BFT++) [9], for CPS under malicious cyber attacks. BFT++, which is applied to CPS with redundant controllers, uses one of the controllers as backup. The other controllers are engineered to crash upon malicious attack, which triggers automatic and fast controller recovery using the backup. If the controllers are restored in time, then the system can guarantee safety. This architecture exploits the fact that the cyber subsystem operates on a shorter timescale than the physical subsystem, which has an inherent natural resilience from the physical dynamics against limited cyber disruptions.

Following BFT++, several other unpublished yet effective approaches have appeared with different implementations of recovery and backup. In parallel, alternative approaches are proposed in [10], [11] for CPS without redundancy. The controller in these approaches is programmed to restart proactively or periodically to recover the system from malicious attack. The physical subsystem can then maintain safety by utilizing its natural resilience and tuning the controller availability.

The aforementioned architectures [9]–[11], which we collectively refer as cyber resilient architectures (CRAs) have found successful applications in different CPS. While the CRAs can independently provide safety guarantees, the analyses undertaken are distinct and specific to the systems or architectures. Hence these analyses may not be readily extended from one CPS to another. Therefore a common framework which allows a general method of analysis for these seemingly unrelated yet novel architectures is of key interest. Such a framework will also enable comparison among different architectures under a common baseline. Currently, such an analytical framework does not exist.

In this paper, we propose a common framework that models the simplex architecture and the CRAs. We then present a control policy synthesis with safety guarantee using our proposed framework, which applies to any of these architectures. We make the following specific contributions:

- We construct a hybrid system to model CPS implementing the simplex architecture and CRAs. We propose a common framework that captures these architectures.
- We derive the sufficient conditions for a control policy to satisfy safety with respect to any specified budget.
- We propose an algorithm to compute a control policy that satisfies our derived conditions. Our proposed algorithm converges to a feasible solution, given its existence, within finite number of iterations.
- We validate our proposed approach using a case study on lateral control of a Boeing 747. We show that our proposed approach guarantees the safety of Boeing 747 with respect to the given budget constraint.

The reminder of the paper is organized as follows. Section II presents the related work. Section III introduces the CPS model and presents the problem statement. Section IV gives our proposed framework. Section V presents our proposed

*Authors contributed equally to this work.

[1]Luyao Niu and Andrew Clark are with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609 {lniu,aclark}@wpi.edu

[2]Abdullah Al Maruf and Radha Poovendran are with the Network Security Lab, Department of Electrical and Computer Engineering, University of Washington, Seattle, WA 98195-2500 {maruf3e,rp3}@uw.edu

[3]J. Sukarno Mertoguno is with Information and Cyber Sciences Directorate, Georgia Tech Research Institute, Atlanta, GA 30332 {karno}@gatech.edu

solution approach. Section VI contains a case study on a Boeing 747. Section VII concludes the paper.

II. Related Work

Safety verification [12], [13] and safety controller synthesis [14]–[17] for CPS operated in benign environment have been extensively studied.

Fault tolerant controllers [3]–[5] and architectures [6]–[8] have been widely adopted for CPS that may incur faults. One of the well-known fault tolerant designs is the simplex architecture. This architecture consists of a main controller which is vulnerable to random failures and a safety controller which is verifiable and fault-free [6]. Under certain conditions, e.g., the main controller experiences a fault, a decision module instantaneously switches to the safety controller. The decision module switches back to the main controller after the main controller recovers from fault. These fault tolerant approaches assume that there is no common failure for all components which may not hold for malicious attack.

There exist two main trends of approaches to address CPS under malicious cyber attacks. The first category aims at protecting the system from malicious attacks using control- and game-theoretic approaches [18]–[20]. These approaches detect the attack and then filter its impact. The second body of literature focuses on designing attack tolerant systems. The CRAs [9]–[11], [21] belong to this category.

BFT++ and other variants [9] are applied when CPS have redundant controllers. One of the redundant controllers is used as backup and equipped with a buffer storing the time-delayed inputs. The non-backup controllers are deliberately engineered to crash following a malicious exploit, e.g., by implementing software diversity [22] or memory/instruction randomization [23]. Sensing the crash, BFT++ recovers the controllers quickly from the backup whose integrity is ensured by flushing the buffer.

The CRA proposed in [10] and also the restart-based mechanisms [21], [24]–[27] are applicable to CPS that do not have redundancy. These approaches reset the cyber subsystem to a 'clean' state via restart to recover from attack. The authors of [10] tunes controller availability for the safety of physical subsystem, whereas the restart-based mechanisms use reachability analysis [25]–[27] for safety guarantee.

III. System Model and Problem Formulation

We first give some notations before presenting the system model. Then we state the problem investigated in this paper.

A continuous function $\alpha : [-b, a) \to (-\infty, \infty)$ belongs to extended class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$ for some $a, b > 0$. Throughout this paper, we use $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$, and $\mathbb{Z}_{\geq 0}$ to denote the set of real numbers, non-negative real numbers, positive real numbers, and non-negative integers, respectively. Given a vector $x \in \mathbb{R}^n$, we denote its i-th entry as $[x]_i$, where $i = 1, \ldots, n$.

Consider a CPS consisting of a cyber subsystem and a physical subsystem. The physical subsystem is modeled by a plant that evolves following

$$\dot{x}_t = f(x_t) + g(x_t)u_t, \tag{1}$$

where $x_t \in \mathcal{X} \subset \mathbb{R}^n$ is the system state and $u_t \in \mathcal{U} \subset \mathbb{R}^m$ is the control input. Functions $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are assumed to be Lipschitz continuous. We also assume that $\mathcal{U} = \prod_{i=1}^{m} [u_{i,min}, u_{i,max}]$ with $u_{i,min} < u_{i,max}$. The physical plant is normally recommended to be operated within a certain range $\mathcal{C} = \{x \in \mathcal{X} : h(x) \geq 0\}$, where $h : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function. We assume that set $\mathcal{C}$ is compact. Given the system state x, the actuator signal u is determined by a control policy $\mu : \mathcal{X} \to \mathcal{U}$.

Although the physical plant evolves in continuous time, the cyber subsystem interacts with the physical subsystem following functioning cycles. We assume that the sensors can directly measure the physical state x. At each functioning cycle $k \in \mathbb{Z}_{\geq 0}$, the cyber subsystem measures $x_{k\delta}$ and updates the actuator signal $u_{k\delta} = \mu(x_{k\delta})$. The actuator signal remains constant during each functioning cycle k. In the remainder of this paper, we refer to a functioning cycle as an epoch with length $\delta > 0$.

The system is subject to a malicious attack initiated by an intelligent adversary. The adversary aims at driving the physical plant outside $\mathcal{C}$ to damage it. The adversary can exploit the vulnerabilities in the cyber subsystem and intrude into the system. Once the adversary intrudes successfully, it gains access to the software, actuators, and other peripherals. As a consequence, the actuator signal is corrupted by the adversary and deviates from the desired control policy $\mu(\cdot)$. To recover the system from attack, the CRAs and other mechanisms have been proposed, as reviewed in Section I and II. Let $t_1 \geq 0$ be a time instant when the adversary corrupts the cyber subsystem. The CRAs eliminate the adversary from the system at some time $\tilde{t} > t_1$. We denote the time instant when the adversary successfully corrupts the cyber subsystem again for the first time after $\tilde{t}$ as $t_2 > \tilde{t}$. We define the interval $[t_1, t_2]$ as an *attack cycle*. Note that the length $A = t_2 - t_1$ of each attack cycle varies, and is dependent on the adversary. Later in Section V, we will compute a lower bound for A to guarantee system safety.

Due to the malicious attack, the physical plant may have to be temporarily operated outside $\mathcal{C}$. To avoid causing irreversible damage to the plant, we need to minimize the amount of time that the CPS is operated outside $\mathcal{C}$ or minimize how far the physical state x deviates from $\mathcal{C}$. We capture the instantaneous damage incurred by the plant when operated outside $\mathcal{C}$ as a cost $L : \mathbb{R} \to \mathbb{R}_{\geq 0}$, defined as

$$L(h(x)) = \begin{cases} L_1(-h(x)), & \text{if } h(x) < 0 \\ 0, & \text{if } h(x) \geq 0 \end{cases} \tag{2}$$

where $L_1 : \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$ is a monotone non-decreasing function. When $L_1(-h(x)) = 1$, then $\int_t L(x)\,\mathrm{d}t$ is equal to the amount of time such that $x_t \notin \mathcal{C}$. When $L_1(-h(x)) = -h(x)$ for all $x \notin \mathcal{C}$, Eqn. (2) models the deviation of the physical plant from the boundary of $\mathcal{C}$. We define the physical safety with respect to budget B as follows.

Definition 1 (Physical Safety with Respect to Budget B)**.** *The physical plant is safe with respect to a budget B if the*

following relation holds for any attack cycle $[t_1, t_2]$*:*

$$J = \int_{t=t_1}^{t_2} L(h(x_t))\, dt \leq B. \tag{3}$$

Eqn. (3) enforces an upper bound on the cost incurred by the system during any attack cycle. When $B = 0$, Definition 1 recovers the strict safety constraint $x_t \in \mathcal{C}$ for all $t \geq 0$ as a special case. Given Definition 1, the problem of synthesizing a control policy with safety guarantee is stated as follows:

Problem 1. *Synthesize a control policy* $\mu : \mathcal{X} \to \mathcal{U}$ *for the CPS such that Definition 1 is satisfied for a given budget* B*.*

IV. Our Proposed Cyber Resilient Framework

In this section, we first detail the timing behaviors of the CRAs. Then we construct a hybrid system that models CPS adopting any of these architectures. The simplex architecture reviewed in Section II is also incorporated in our framework for completeness. We finally reformulate Problem 1 using the constructed hybrid system.

A. Timing Behaviors of the CRAs

In this subsection, we present the timing behaviors of the CRAs [9]–[11], which track the status of the cyber subsystem. Note that the status of the cyber subsystem are discrete. When the adversary intrudes into the system at epoch j, the cyber subsystem changes from the normal to the corrupted status, indicating the adversary can arbitrarily manipulate the controllers. If the CPS have redundant controllers as discussed in [9], the non-backup controllers will crash by epoch $j + N_1$ in the worst-case, which triggers controller restoration, leading the cyber subsystem to transit from the corrupted status to the restoration status. In practical implementations of BFT++, we observe that $N_1 = 2$ and the buffer length is chosen to be greater than N_1. Denote the worst-case number of epochs needed for controller restoration as N_2. Then the cyber subsystem returns to the normal status using the backup controller by epoch $j+N_1+N_2$. To ensure safety of the physical subsystem, crash delay N_1 needs to be small enough, and the restoration time N_2 needs to be tolerated by the physical subsystem's resilience $\Delta(x)$ which is determined by the physical state and system dynamics.

When CPS have no redundant controllers, cyber subsystem recovery can be triggered by either the attack or the timer [10], [11]. If the controller crashes due to attack, which takes at most N_3 epochs, then the system reboots and re-initializes the controller to recover it. The worst-case time needed for such controller recovery, denoted as N_4, is in general larger than BFT++, i.e., $N_4 \geq N_2$. We remark that the controller restart is triggered by the timer when the attack does not crash the controller. If the recovery is triggered by the timer and there is no attack, then the system restart is executed every $N_4 + N_5$ epochs, where N_5 is the number of epochs elapsed in the normal status during one restart period. Safety of the physical subsystem is then ensured by tuning the time between two consecutive restarts and the controller availability.

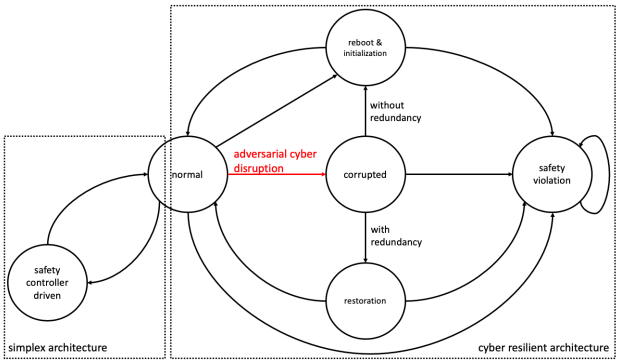

Fig. 1: Hybrid system $H = (\mathcal{X}, \mathcal{U}, \mathcal{L}, \mathcal{Y}, \mathcal{Y}_0, Inv, \mathcal{F}, \Sigma, \mathcal{E}, \Phi)$ captures the simplex architecture (left part) and the CRAs (right part). Each state $y = (x, (l_1, j)) \in \mathcal{Y}$ of the hybrid system captures the continuous physical state $x \in \mathcal{X}$ and the discrete location $l = (l_1, j) \in \mathcal{L}$ including the system status l_1 and the epoch index j. Each circle in the figure represents a discrete location in $\mathcal{L}$, with the epoch indices being omitted. The arrows in the figure represent the transitions Σ of hybrid system H. Each transition is labeled using $e \in \mathcal{E}$, and is enabled when the corresponding clock constraint $\phi \in \Phi$ is satisfied. The detailed labels and clock constraints are given in Table I. The transition in red color is triggered by external event, i.e., the adversarial cyber disruption.

B. Our Proposed Framework

In this subsection, we first construct a hybrid system to model the CPS implementing the simplex architecture or CRAs. We then restate Problem 1 in the context of the hybrid system. We construct a hybrid system $H = (\mathcal{X}, \mathcal{U}, \mathcal{L}, \mathcal{Y}, \mathcal{Y}_0, Inv, \mathcal{F}, \Sigma, \mathcal{E}, \Phi)$, as shown in Fig. 1, where

- $\mathcal{X} \subseteq \mathbb{R}^n$ is the continuous state space modeling the states of the physical subsystem. $\mathcal{U} \subseteq \mathbb{R}^m$ is the set of admissible control inputs of the physical subsystem.
- $\mathcal{L} = \{normal, R\&I, SC, restoration, corrupted, safety\ violation\} \times \mathbb{Z}_{\geq 0}$ is a set of discrete locations[1], with each location $l \in \mathcal{L}$ modeling the status of the system at each epoch index.
- $\mathcal{Y} = \mathcal{X} \times \mathcal{L}$ is the state space of hybrid system H, and $\mathcal{Y}_0 \subseteq \mathcal{Y}$ is the set of initial states.
- $Inv : \mathcal{L} \to 2^{\mathcal{X}}$ is the invariant that maps from the set of locations to the power set of $\mathcal{X}$. That is, $Inv(l) \subseteq \mathcal{X}$ specifies the set of possible continuous states when the system is at location l.
- $\mathcal{F}$ is the set of vector fields. For each $F \in \mathcal{F}$, the continuous system state evolves as $\dot{x} = F(x, u, l)$, where F is jointly determined by the system dynamics and the status of the cyber subsystem, and $\dot{x}$ is the time derivative of continuous state x.
- $\Sigma \subseteq \mathcal{Y} \times \mathcal{Y}$ is the set of transitions between the states of the hybrid system. A transition $\sigma = ((x, l), (x', l'))$ models the state transition from (x, l) to (x', l').

[1]Throughout this paper, we denote $l = reboot\&initialization$ as $l = R\&I$ and denote $l = safety\ controller\ driven$ as $l = SC$ for simplicity.

- $\mathcal{E} = \Gamma \cup \mathbb{Z}_{\geq 0}$ is a set of labels, where Γ is the finite alphabet set. Each $\gamma \in \Gamma$ is labeled on some transition $\sigma \in \Sigma$ indicating the events that triggers the transition.
- Φ is a set of clock constraints, with each $\phi \in \Phi$ being defined as $\phi : \Sigma \times \mathbb{Z}_{\geq 0} \to \{0, 1\}$. Function ϕ maps the time elapsed in each discrete location labeled on each transition to the binary set $\{0, 1\}$, indicating if the transition is enabled or not.

In Fig. 1, we only label the first element of each location $l = (l_1, j)$, where $l_1 \in \{normal, R\&I, SC, restoration, corrupted, safety\ violation\}$, and the epoch index $j \in \mathbb{Z}_{\geq 0}$ is omitted. Variable l_1 represents the status of the system, as explained in Section IV-A. Particularly, we use $R\&I$ to represent controller reboot and initialization for CPS without redundancy, and use SC to represent the status where the system is driven by the safety controller as suggested in the simplex architecture. In the remainder of this paper, we refer to l_1 as the location of H omitting the epoch index when the context is clear. The set of vector fields $\mathcal{F}$ captures the dynamics at each discrete location. For instance, when $l = (R\&I, j)$, we have that $\dot{x} = F(x, u, l) = f(x)$ for all $j \in \mathbb{Z}_{\geq 0}$ since $u_t = 0$.

We label each transition $\sigma = ((x, l), (x', l')) \in \Sigma$ using $e = (\gamma, e_2) \in \mathcal{E}$. The detailed label associated with each transition can be found in Table I. Here we use γ to represent the event that triggers the transition. For instance, the transition from $normal$ to $corrupted$ is triggered by the adversarial cyber disruption, whereas the transition from $corrupted$ to $restoration$ is triggered by controller crash. The element $e_2 \in \mathbb{Z}_{\geq 0}$ denotes the number of epochs elapsed in status l_1 before the occurrence of transition $\sigma = ((x, (l_1, j)), (x', (l_1', j')))$. For example, at most N_1 epochs elapse at location $corrupted$ before the transition from $(x, (corrupted, j))$ to $(x', (restoration, j'))$ occurs. When $e_2 = 0$, it indicates that the transition occurs instantaneously.

A transition in hybrid system H is enabled if and only if a clock constraint ϕ associated with the transition is satisfied. Consider a transition $\sigma = ((x, (l_1, j)), (x', (l_1', j')))$ labeled with $e = (\gamma, e_2)$. A clock constraint ϕ verifies if j, j', and e_2 satisfies $j + e_2 = j'$ with e_2 being determined by the architecture. The clock constraint enables hybrid system H to always track the correct epoch index.

We remark that for each status l_1, we do not depict the transitions $(x, (l_1, j))$ to $(x, (l_1, j + 1))$ for compactness of the figure. These transitions do not cause any system status jump, and only track the evolution of epoch indices. We also define a guard set $\mathcal{G}(l, l')$ as $\mathcal{G}(l, l') = \{x \in \mathcal{X} : ((x, l), (x, l')) \in \Sigma\}$ which represents the set of physical states starting from which the system can transit from location l to l' on the hybrid system H.

The transitions that end at $l_1 = safety\ violation$ are triggered by event $\gamma = Maximum\ tolerance$, indicating that the physical subsystem has not received the correct input in time and has utilized all resilience against the disruption. In this case, safety violation $J > B$ becomes inevitable, and the system cannot recover after safety violation (captured via the self-loop in Fig. 1). We capture the maximum tolerance provided by the physical subsystem using e_2 labeled on the transitions. Note that the tolerance depends on the physical system state x and is denoted as $\Delta(x)$.

We are now ready to translate problem 1 using the context of hybrid system H as follows:

Restatement of Problem 1. Given hybrid system H, synthesize a control policy $\mu : \mathcal{X} \to \mathcal{U}$ such that hybrid system H never reaches status $l_1 = safety\ violation$.

V. Analysis of the Proposed Framework

This section presents the proposed solution approach to Problem 1. We first develop sufficient conditions for the control policy that guarantees safety of the physical subsystem under a cyber attack. Then we formulate the derived conditions as sum-of-squares (SOS) constraints and propose an algorithm to compute a control policy and the corresponding parameters. We finally give the convergence and complexity of our algorithm.

In the following, we derive the sufficient conditions under which a control policy $\mu : \mathcal{X} \to \mathcal{U}$ ensures the system to satisfy Definition 1 with respect to a given budget B. The idea is that if control policy μ drives the physical subsystem to $\mathcal{C}_1 = \{x : h(x) \geq c\} \subseteq \mathcal{C}$ when $l_1 =$

Designs	Starting state $(x, (l_1, j))$	Target state $(x', (l_1', j'))$	Label $e = (\gamma, e_2)$	Clock constraint ϕ
Simplex architecture	$(x, (normal, j))$	$(x, (SC, j'))$	$(Safety\ controller\ activated, e_2)$	$(j + e_2 = j') \wedge (e_2 = 0)$
	$(x, (SC, j))$	$(x, (normal, j'))$	$(Safety\ controller\ inactivated, e_2)$	$(j + e_2 = j') \wedge (e_2 = 0)$
the CRAs	$(x, (normal, j))$	$(x', (corrupted, j'))$	$(Adversarial\ cyber\ disruption, e_2)$	$(j + e_2 = j') \wedge (e_2 \geq 0)$
	$(x, (corrupted, j))$	$(x', (restoration, j'))$	$(Controller\ crash, e_2)$	$(j + e_2 = j') \wedge (e_2 \leq N_1)$
	$(x, (restoration, j))$	$(x', (normal, j'))$	$(Controller\ restored, e_2)$	$(j + e_2 = j') \wedge (e_2 \leq N_2)$
	$(x, (normal, j))$	$(x', (R\&I, j'))$	$(Timer, e_2)$	$(j + e_2 = j') \wedge (e_2 = N_5)$
	$(x, (corrupted, j))$	$(x', (R\&I, j'))$	$(Controller\ crash, e_2)$	$(j + e_2 = j') \wedge (e_2 \leq N_3)$
	$(x, (corrupted, j))$	$(x', (R\&I, j'))$	$(Timer, e_2)$	$(j + e_2 = j') \wedge (e_2 \leq N_5)$
	$(x, (R\&I, j))$	$(x', (normal, j'))$	$(Initialization\ done, e_2)$	$(j + e_2 = j') \wedge (e_2 \leq N_4)$
	$(x, (\cdot, j))$	$(x', (safety\ violation, j'))$	$(Maximum\ tolerance, e_2)$	$(j + e_2 = j') \wedge (e_2 \geq \Delta(x))$

TABLE I: This table shows the transitions with their corresponding labels and clock constraints. The second and third columns give the starting and end states of a transition, respectively. The fourth column presents the label $e = (\gamma, e_2)$ associated with the transition $\sigma = ((x, (l_1, j)), (x', (l_1', j')))$. The trigger event of the transition is denoted as γ, and the time elapsed in l_1 is denoted using $e_2 \in \mathbb{Z}_{\geq 0}$. The fifth column gives the clock constraint ϕ that needs to be satisfied by the transition σ and parameter e_2. Parameters N_1, N_2, N_3, N_4, and N_5 are determined by the architecture design. Parameter $\Delta(x)$ captures the maximum tolerance provided by the physical subsystem.

$normal$ and we can constrain the system trajectory to remain in a set $\mathcal{D} = \{x : h(x) \geq -d\} \supseteq \mathcal{C}$ for any $l_1 \in \{corrupted, R\&I, restoration\}$, then we can limit the worst-case cost incurred during one attack cycle to be bounded by B by tuning choices $c, d \geq 0$. We denote the worst-case number of epochs when the system is at some status $l_1 \in \{corrupted, R\&I, restoration\}$ as N. We then have the following conditions:

Theorem 1. *Consider hybrid system H and let set $\mathcal{C}$ be defined as in Section III. Let $h_c(x) = h(x) - c$ and $h_d(x) = h(x) + d$. We define $\mathcal{C}_1 = \{x : h_c(x) \geq 0\}$ and $\mathcal{D} = \{x : h_d(x) \geq 0\}$. Consider an arbitrary attack cycle denoted as $[t_1, t_2]$ and suppose $x_{t_1} \in \mathcal{C}_1$. If there exist constants $c, d \geq 0$, $\tau > 0$, and a control policy $\mu : \mathcal{X} \to \mathcal{U}$ such that*

$$\frac{\partial h_d}{\partial x}(x)(f(x) + g(x)u) \geq -\frac{c+d}{N\delta}, \ \forall (x,u) \in \mathcal{D} \times \mathcal{U} \quad (4a)$$

$$\frac{\partial h_c}{\partial x}(x)(f(x) + g(x)\mu(x)) \geq \frac{c+d}{\tau}, \ \forall x \in \mathcal{D} \setminus \mathcal{C}_1 \quad (4b)$$

$$\frac{\partial h_c}{\partial x}(x)(f(x) + g(x)\mu(x)) \geq -\alpha(h_c(x)), \ \forall x \in \mathcal{C}_1 \quad (4c)$$

$$\frac{N\delta}{c+d}\int_{s=0}^{d} L_1(s)\,ds + \frac{\tau}{c+d}\int_{s=0}^{d} L_1(s)\,ds \leq B \quad (4d)$$

then system (1) *is safe with respect to budget B by taking policy μ at $l_1 = normal$, provided that $A = t_2 - t_1 \geq \tau + N\delta$. Furthermore, $x_t \in \mathcal{D}$ for $t \in [t_1, t_2]$ and $x_t \in \mathcal{C}_1$ for $t \in [t_1 + \tau + N\delta, t_2]$.*

Proof. The proof consists of two steps. We first find the guard set for each state of hybrid system H when the conditions in Eqn. (4) hold. We then prove that the system is safe with respect to budget B according to Definition 1.

In the first step, we show that if $t_2 \geq t_1 + \tau + N\delta$, $x_{t_1} \in \mathcal{C}_1$ and the conditions in (4) hold, then $x_t \in \mathcal{D}$ for $t \in [t_1, t_2]$. Additionally we show that $x_t \in \mathcal{C}_1$ for $t \in [t_1 + \tau + N\delta, t_2]$ using control policy μ at $l_1 = normal$. As hybrid system H takes N epochs to transition to $l_1 = normal$ after being corrupted, therefore $t_1 + N\delta$ is the time instant when transition $\sigma = (\cdot, normal)$ takes place after t_1. Then for any $t' \in [t_1, t_1 + N\delta]$ and for any $u \in \mathcal{U}$, we have that

$$h(x_{t'}) = h(x_{t_1}) + \int_{t=t_1}^{t'} \dot{h}\,\mathrm{d}t \geq c - \frac{c+d}{N\delta}(t' - t_1) \geq -d, \quad (5)$$

where the inequality holds by Eqn. (4a) and the assumptions that $x_{t_1} \in \mathcal{C}_1$ and $t' \in [t_1, t_1 + N\delta]$. Therefore, we have that $x_{t'} \in \mathcal{D}$ for all $t' \in [t_1, t_1 + N\delta]$, indicating that $\mathcal{G}(R\&I, normal), \mathcal{G}(restoration, normal) \subseteq \mathcal{D}$.

Now consider that control policy μ is applied when the system is at $l_1 = normal$. Let $\hat{t}$ be any time when the system is at $l_1 = normal$ for which the trajectory remains in $\mathcal{D} \setminus \mathcal{C}_1$ (i.e. $-d \leq h(x_{\hat{t}}) < -c$). Then we can write

$$h(x_{\hat{t}}) = h(x_{t_1+N\delta}) + \int_{t=t_1+N\delta}^{\hat{t}} \dot{h}\,\mathrm{d}t \geq -d + \frac{c+d}{\tau}(\hat{t} - t_1 - N\delta), \quad (6)$$

where the inequality holds by Eqn. (4b) and the fact that $\mathcal{G}(R\&I, normal), \mathcal{G}(restoration, normal) \subseteq \mathcal{D}$. If $\hat{t} \geq t_1 + N\delta + \tau$, then Eqn. (6) renders $h(x_{\hat{t}}) \geq c$ and thus $x_{\hat{t}} \in \mathcal{C}_1$. Further note that by [14, Thm. 2], $\mathcal{C}_1$ is forward invariant using control policy μ. Thus, $x_t \in \mathcal{C}_1$, $\forall t \in [t_1 + N\delta + \tau, t_2]$. Since $\mathcal{C}_1 \subseteq \mathcal{D}$, we have $x_t \in \mathcal{D}$ for all $t \in [t_1, t_2]$.

In the second step, we quantify the worst-case cost incurred by the system. We first compute the cost incurred during time $[t_1, t_1 + N\delta]$. Suppose the system reaches the boundary of $\mathcal{C}$ at time instants $z_1 \leq z_2 \leq \ldots \leq z_K \leq t_1 + N\delta$, where $z_1 \geq t_1$ and $z_K \leq t_1 + N\delta$. We have that

$$J_1 \doteq \int_{t=t_1}^{t_1+N\delta} L(h(x_t))\,\mathrm{d}t = \int_{t=t_1}^{z_1} L(h(x_t))\,\mathrm{d}t + \sum_{j=1}^{K-1}\int_{t=z_j}^{z_{j+1}} L(h(x_t))\,\mathrm{d}t + \int_{t=z_K}^{t_1+N\delta} L(h(x_t))\,\mathrm{d}t$$

Since $h(x_t) \in \mathcal{C}$ for $t \in [t_1, z_1]$, therefore $L(h(x_t)) = 0$. Additionally, since L_1 is monotone non-decreasing, by Eqn. (5) we have that $L(h(x_t)) = L_1(-h(x_t)) \leq L_1(-c + \frac{c+d}{N\delta}(t - t_1))$ for any $t \in [z_j, z_{j+1}]$ if $h(x_t) \notin \mathcal{C}$. Using these arguments, we have that

$$\begin{aligned} J_1 &\leq \sum_{j=1}^{K-1}\int_{t=z_j}^{z_{j+1}} L_1(-c + \frac{c+d}{N\delta}t)\,\mathrm{d}t + \int_{t=z_K}^{t_1+N\delta} L_1(-c + \frac{c+d}{N\delta}t)\,\mathrm{d}t \\ &= \int_{t=z_1}^{t_1+N\delta} L_1(-c + \frac{c+d}{N\delta}t)\,\mathrm{d}t. \end{aligned}$$

Using Eqn. (5), it follows that $z_1 \geq t_1 + c/(\frac{c+d}{N\delta}) = t_1 + \frac{cN\delta}{c+d}$. Therefore, we have that

$$\begin{aligned} J_1 &\leq \int_{t=t_1+\frac{cN\delta}{c+d}}^{t_1+N\delta} L_1(-c + \frac{c+d}{N\delta}(t - t_1))\,\mathrm{d}t \\ &= \int_{t=0}^{\frac{dN\delta}{c+d}} L_1(\frac{c+d}{N\delta}t)\,\mathrm{d}t = \frac{N\delta}{c+d}\int_{s=0}^{d} L_1(s)\,\mathrm{d}s, \end{aligned}$$

where the above holds by variable substitution and the fact that L_1 is non-negative.

We now quantify the worst-case cost incurred during time $[t_1 + N\delta, t_2]$. By Eqn. (6), $h(x_t) \geq -d + \frac{c+d}{\tau}(t - t_1 - N\delta)$ for $x_t \in \mathcal{D} \setminus \mathcal{C}_1$. Note that $h(x_t) \geq 0$ for all $t \in [t_1 + N\delta + \frac{d\tau}{c+d}, t_2]$ using control policy $\mu(x)$. Therefore we have that

$$\begin{aligned} J_2 &\doteq \int_{t=t_1+N\delta}^{t_2} L(h(x_t))\,\mathrm{d}t \\ &= \int_{t=t_1+N\delta}^{t_1+N\delta+\frac{d\tau}{c+d}} L(h(x_t))\,\mathrm{d}t + \int_{t=t_1+N\delta+\frac{d\tau}{c+d}}^{t_2} L(h(x_t))\,\mathrm{d}t \\ &\leq \int_{t=N\delta}^{N\delta+\frac{d\tau}{c+d}} L_1(d - \frac{c+d}{\tau}(t - N\delta))\,\mathrm{d}t \\ &= \int_{t=0}^{\frac{d\tau}{c+d}} L_1(d - \frac{c+d}{\tau}t)\,\mathrm{d}t = \frac{\tau}{c+d}\int_{s=0}^{d} L_1(s)\,\mathrm{d}s \end{aligned}$$

where the above holds by (6), L_1 is monotone non-decreasing and non-negative, and $L(h(x_t)) = 0$ for $x_t \in \mathcal{C}$.

Therefore we have that the total cost is upper bounded by $J_1 + J_2$, which yields condition (4d). □

The above result also encompasses the case when the physical subsystem is subject to a strict safety constraint, i.e., $x_t \in \mathcal{C}$ for all $t \geq 0$. This case can be captured by letting $B = 0$ in Definition 1. The sufficient conditions for a control policy with strict safety guarantee are given as follows.

Corollary 1. *Consider hybrid system H and a safety set $\mathcal{C}$. Let $h_c(x) = h(x) - c$ and $\mathcal{C}_1 = \{x : h_c(x) \geq 0\}$. Consider an arbitrary attack cycle denoted as $[t_1, t_2]$ and suppose $x_0 \in \mathcal{C}_1$. If there exist constants $c \geq 0$, $\tau > 0$, and a control policy $\mu : \mathcal{X} \to \mathcal{U}$ such that*

$$\frac{\partial h}{\partial x}(x)(f(x) + g(x)u) \geq -\frac{c}{N\delta},\ \forall (x,u) \in \mathcal{C} \times \mathcal{U} \quad (7a)$$

$$\frac{\partial h_c}{\partial x}(x)(f(x) + g(x)\mu(x)) \geq \frac{c}{\tau},\ \forall x \in \mathcal{C} \setminus \mathcal{C}_1 \quad (7b)$$

$$\frac{\partial h_c}{\partial x}(x)(f(x) + g(x)\mu(x)) \geq -\alpha(h_c(x)),\ \forall x \in \mathcal{C}_1 \quad (7c)$$

$x_t \in \mathcal{C}$ for all $t \geq 0$ provided that $A = t_2 - t_1 \geq \tau + N\delta$.

Proof. The corollary can be proved as a special case of Theorem 1 with $d = B = 0$, which yields that $\mathcal{D} = \mathcal{C}$ and thereby $x_t \in \mathcal{C},\ \forall t \in [t_1, t_2]$. Note that $[t_1, t_2]$ is an arbitrary attack cycle and $x_0 \in \mathcal{C}_1$, rendering $x_t \in \mathcal{C},\ \forall t \geq 0$. □

The above analysis can also be used for the safety controller design of the simplex architecture. The safety controller, which is invoked when the system approaches the boundary of $\mathcal{C}$, can be obtained using Theorem 1 by letting $c = d = B = 0$ to guarantee strict safety with respect to $\mathcal{C}$. Control policy μ only needs to satisfy Eqn. (4c) in this case.

Now we focus on the computation of control policy μ as well as parameters $c, d \geq 0$ and $\tau > 0$ so that safety is satisfied according to Definition 1. Our idea is to translate the conditions in Theorem 1 to a set of sum-of-squares (SOS) constraints. We first make the following assumption.

Assumption 1. *We assume that functions $f(x)$, $g(x)$, and $h(x)$ are polynomial in x. Additionally, we assume that function L_1 is polynomial in $-h(x)$.*

When Assumption 1 holds, $L_1(-h(x))$ can be written as $L_1(-h(x)) = \sum_{i=0}^{k}(-h(x))^i a_i$, where a_i is the coefficient of $(-h(x))^i$ for each $i = 0, \ldots, k$. Next we formulate Eqn. (4) as a set of SOS constraints.

Proposition 1. *Suppose there exist parameters $c, d \geq 0$ and $\theta > 0$ such that the following expressions are SOS:*

$$\frac{\partial h_d}{\partial x}(x)[f(x) + g(x)u] + \frac{c+d}{N\delta} - q(x,u)h_d(x) \quad (8a)$$
$$- \sum_{i=1}^{m}(w_i(x,u)([u]_i - [u]_{i,min}) + v_i(x,u)([u]_{i,max} - [u]_i)),$$

$$\frac{\partial h_c}{\partial x}(x)[f(x) + g(x)\lambda(x)] - (c+d)\theta$$
$$- l(x)h_d(x) + p(x)h_c(x), \quad (8b)$$

$$\frac{\partial h_c}{\partial x}(x)[f(x) + g(x)\lambda(x)] + \alpha(h_c(x)) - r(x)h_c(x), \quad (8c)$$

$$\lambda_i(x) - [u]_{i,min},\quad [u]_{i,max} - \lambda_i(x),\ \forall i = 1, \ldots, m, \quad (8d)$$

and the following inequality holds:

$$B(c+d) - (N\delta + \frac{1}{\theta})\sum_{i=1}^{k}\frac{a_i d^{i+1}}{i+1} \geq 0 \quad (9)$$

where $l(x), p(x), q(x,u), r(x)$ are SOS, $\lambda_i(x)$ is a polynomial in x for each $i = 1, \ldots, m$, and $w_i(x,u)$ and $v_i(x,u)$ are SOS for each $i = 1, \ldots, m$. Then $\mu(x) = \lambda(x) = [\lambda_1(x), \ldots, \lambda_m(x)]^\top$, c, d, and $\tau = \frac{1}{\theta}$ satisfy the conditions in Eqn. (4)*.*

Proof. Consider $x \in \mathcal{D}$ and $[u]_{i,min} \leq [u]_i \leq [u]_{i,max}$ for all $i = 1, \ldots, m$. Then we have that $h_d(x) \geq 0$, $[u]_i - [u]_{i,min} \geq 0$, and $[u]_{i,max} - [u]_i \geq 0$. Since expression (8a), $q(x,u)$, $w_i(x,u)$, and $v_i(x,u)$ are SOS for all $i = 1, \ldots, m$, therefore for all $(x,u) \in \mathcal{D} \times \mathcal{U}$ we can write

$$\frac{\partial h_d}{\partial x}(x)[f(x) + g(x)u] + \frac{c+d}{N\delta} \geq q(x,u)h_d(x)$$
$$+\sum_{i=1}^{m}(w_i(x,u)([u]_i - [u]_{i,min}) + v_i(x,u)([u]_{i,max} - [u]_i)) \geq 0.$$

Thus condition (4a) holds.

Expressions (8b) to (8d) can be proved similarly. Eqn. (9) follows by computing the integrals in Eqn. (4d). Details are omitted due to space constraint. □

Algorithm 1 Heuristic algorithm for computing c, d, τ and control policy $\mu(x)$

1: **Input**: $f(x)$, $g(x)$, B, τ_{max}, c_{max}, $\epsilon_1 > 0$, $\epsilon_2 > 0$
2: **Output:** c, d, τ, $\lambda(x)$
3: **Initialization:** $c = 0$.
4: **while** $c \leq c_{max}$ **do**
5: $\quad d = 0$
6: $\quad$ **while** $d \leq d_{max}$ **do**
7: $\quad\quad$ Maximize θ subject to (8) with c and d fixed.
8: $\quad\quad$ **if** Eqn. (8) is feasible, (9) is satisfied and $\frac{1}{\theta} < \tau_{max}$ **then**
9: $\quad\quad\quad$ **return** d, c, $\tau = \frac{1}{\theta}$, and $\lambda(x)$
10: $\quad\quad$ **else**
11: $\quad\quad\quad$ $d = d + \epsilon_1$
12: $\quad\quad$ **end if**
13: $\quad$ **end while**
14: $\quad$ $c = c + \epsilon_2$
15: **end while**

Simultaneously searching for $\lambda(x), c, d$ and θ that satisfy Proposition 1 leads to bilinearity in (8). To this end, we propose an algorithm to compute $\lambda(x), c, d$ and θ that satisfy Proposition 1, as shown in Algorithm 1. Algorithm 1 first initializes parameters $c = d = 0$. At each iteration, the algorithm maximizes θ using the given c and d. If some θ^* can be found in line 7 which satisfies the conditions in line 8 ($\tau_{max} = \infty$ if not specified), then the algorithm returns c, d, and set $\tau = \frac{1}{\theta^*}$ and $\mu(x) = \lambda(x)$. Otherwise,

the algorithm increases the values of parameters c and/or d and repeat the search process for parameter θ. Algorithm 1 terminates at $c = c_{max}$ and $d = d_{max}$ if no feasible solution to (8) and (9) is found, where $c_{max} = \sup_{x\in\mathcal{C}} h(x)$ and d_{max} is the maximum value of d that satisfies $\sum_{i=0}^{k} N\delta\frac{a_i d^{i+1}}{i+1} \leq (c+d)B$.

Now we briefly characterize the convergence and complexity of Algorithm 1. Our intuition is that if there exists a feasible solution satisfying Eqn. (8) and (9) strictly, then this solution must lie within the interior of the feasible solution set. Thus by choosing ϵ_1 and ϵ_2 appropriately small, the convergence of Algorithm 1 can be guaranteed. We formalize this convergence result in the following proposition.

Proposition 2. *Suppose the set $\mathcal{C}$ is compact and L_1 is polynomial in $-h(x)$ with non-zero degree. Further assume that the feasible solution (c, d) for (8) and (9) satisfy inequality constraints in (8) and (9) strictly with $0 \leq c \leq c_{max}$, $0 \leq d \leq d_{max}$ and $0 < \tau \leq \tau_{max}$. Then Algorithm 1 finds a feasible solution with $0 < \tau \leq \tau_{max}$ in finite number of iterations if ϵ_1 and ϵ_2 are chosen appropriately small.*

Proof. Since $\mathcal{C}$ is compact, we have $c_{max} = \sup_{x\in\mathcal{C}} h(x) < \infty$. Also, since the order of d in $\sum_{i=0}^{k} N\delta\frac{a_i d^{i+1}}{i+1}$ is greater than that of $(c+d)B$ and L_1 is non-negative, therefore $d_{max} < \infty$. Using $c_{max}, d_{max} < \infty$, we have that Algorithm 1 terminates in finite number of iterations.

Let (c, d) satisfy (8) and (9) strictly with $0 \leq c \leq c_{max}$, $0 \leq d \leq d_{max}$, and $0 < \tau \leq \tau_{max}$. Therefore there exists an interval $\mathcal{I} \subseteq \mathbb{R}^2$ for (c, d) with non-zero measure for which (8) and (9) are feasible and $c \in [0, c_{max}]$, $d \in [0, d_{max}]$, $\tau \in (0, \tau_{max}]$. Denote the length of $\mathcal{I}$ in c and d as $\tilde{c} > 0$ and $\tilde{d} > 0$, respectively. Let $\epsilon_1 \in (0, \tilde{c})$ and $\epsilon_2 \in (0, \tilde{d})$. Then Algorithm 1 will always terminate with a feasible solution. Otherwise interval $\mathcal{I}$ contains some infeasible solutions to Eqn. (8) and (9), which contradicts its definition. □

By the proof of Proposition 2, the computational complexity of Algorithm 1 is $\lfloor\frac{c_{max}}{\epsilon_1}\rfloor\lfloor\frac{d_{max}}{\epsilon_2}\rfloor M$, where M is the computational complexity of line 7 in Algorithm 1.

VI. Case Studies

This section presents a case study on lateral control of a Boeing 747. The lateral dynamics for a Boeing 747 (at Mach 0.8 and 40000ft) [28] are given as $\dot{x} = f(x) + g(x)u$, where

$$f(x) = \begin{bmatrix} -0.0558 & -0.9968 & 0.0802 & 0.0415 \\ 0.598 & -0.115 & -0.0318 & 0 \\ -3.05 & 0.388 & -0.465 & 0 \\ 0 & 0.0805 & 1 & 0 \end{bmatrix},$$

$g(x) = [0.00729, -0.475, 0.153, 0]^\top$, $x \in \mathbb{R}^4$ with x_1, x_2, x_3, and x_4 representing the side-slip angle, yaw rate, roll rate, and roll angle, respectively. The cyber subsystem updates the control signal with frequency $20Hz$. The aircraft aims at maintaining the yaw rate x_2 within $\mathcal{C} = \{x : h(x) \geq 0\}$ for passengers' comfort and minimizing potential damage to baggage, where $h(x) = 0.025^2 - x_2^2$. Set $\mathcal{C}$ is represented as the white region in Fig. 2. We set $x_0 = [0.01, 0.025, 0, 0]^\top$, $B = 0.02$, and study the following three scenarios.

Scenario I: The aircraft has redundant controllers. We choose parameter $N = N_1 + N_2$ epochs during which the aircraft is either corrupted or under restoration, where $N_1 = N_2 = 2$ [9]. We consider the system is equipped with a buffer of length 3. Using Algorithm 1, we obtain that $c = 0$, $d = 0.4$, and $\tau = 0.16s$, indicating that we need 4 epochs for the system to be at location $l_1 = normal$ after restoration. We let the length of each attack cycle be 8 epochs. The control policy given by Algorithm 1 is $u = Kx$ with $K = [-9.231 \times 10^{-3}, 0.503, -1.805 \times 10^{-3}, 2.373 \times 10^{-5}]$. We depict the trajectory of the yaw rate over 150 epochs using the black solid line in Fig. 2. The non-smoothness in the trajectory is due to switching between location $normal$ (the controller is available) and locations $corrupted$ and $restoration$ (the controller is unavailable). The adversary enforces the yaw rate to exceed 0.025 from the second to fifth epoch with cost $0.0038 < B$.

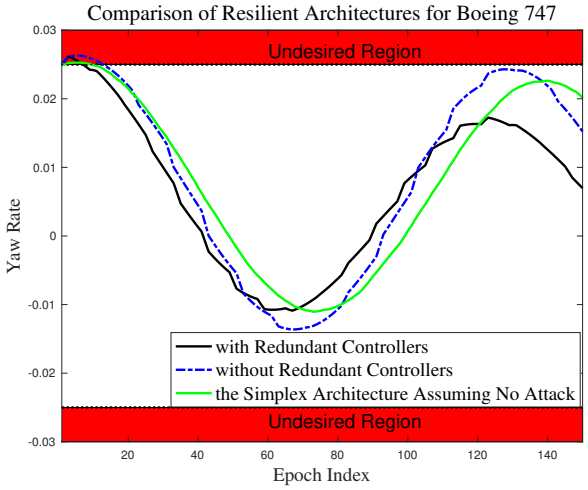

Fig. 2: The yaw rate of a Boeing 747 adopting different architectures over 150 epochs. Set $\mathcal{C} = \{x_2 : 0.025^2 - x^2 \geq 0\}$ represented by the white region. The black solid line depicts the yaw rate of an aircraft with redundancy. The blue dash-dotted line is the yaw rate of an aircraft without redundancy. The green solid line describes the yaw rate of an aircraft using the simplex architecture without attack. The safety controller is invoked when $0.025^2 - x_2^2 < 0$.

Scenario II: The aircraft has no redundant controller. In this case, the aircraft restarts the controller to recover the system. We let $N = N_3 + N_4$ with $N_3 = 2$ and $N_4 = 4$. In this case, we let the adversary attack every 10 epochs. Algorithm 1 gives that $c = 0$, $d = 0.4$, $\tau = 0.18s$ (i.e., 4 epochs), and $u = Kx$ with $K = [0.03017, 0.05395, -4.753 \times 10^{-3}, 7.513 \times 10^{-5}]^\top$. The evolution of yaw rate over 150 epochs is plotted using blue dash-dotted line in Fig. 2. The adversary enforces the yaw rate to exceed 0.025 from the second to eleventh epoch with cost $0.0104 < B$.

Scenario III: The aircraft adopts simplex architecture. We assume that the main controller is in a faulty condition

and produces random control input $u_t \in \mathcal{U}$ for each epoch. The safety controller is invoked once $h(x) < 0$. Once the yaw rate exceeds 0.025 (from the first to seventh epoch in Fig. 2), the safety controller drives the yaw rate to $\mathcal{C}$. Note that the simplex architecture assumes that there exists no adversary, which is different with Scenario I and II.

Therefore, the control policy computed using our proposed algorithm ensures safety of the system with respect to specified budget for any of the CRAs or the simplex architecture.

VII. Conclusion

In this paper, we studied the problem of developing a common framework that allows safety analysis and control synthesis of CPS adopting the simplex architecture or the set of cyber resilient architectures including BFT++. We presented the models for cyber and physical subsystems, and formulated the safety property using a budget constraint. Our formulation captures strict safety constraint as a special case. We constructed a hybrid system that models CPS implementing any of these architectures. We derived a set of sufficient conditions for the control policy to satisfy the budget constraint. We translated the conditions into a set of sum-of-squares constraints, and proposed an algorithm to compute the control policy. We analyzed the convergence and complexity of the algorithm. A case study on the lateral control of a Boeing 747 was presented to demonstrate viability of our proposed framework.

References

[1] A. Greenberg, “Hackers remotely kill a Jeep on the highway–with me in it,” 2015. [Online]. Available: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[2] M. R. Lee, J. M. Assante, and T. Conway, “Analysis of the cyber attack on the Ukrainian power grid,” 2016. [Online]. Available: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2016/05/20081514/E-ISAC_SANS_Ukraine_DUC_5.pdf

[3] Y. Zhang and J. Jiang, “Bibliographical review on reconfigurable fault-tolerant control systems,” *Annual Reviews in Control*, vol. 32, no. 2, pp. 229–252, 2008.

[4] F. Sharifi, M. Mirzaei, B. W. Gordon, and Y. Zhang, “Fault tolerant control of a quadrotor UAV using sliding mode control,” in *Conference on Control and Fault-Tolerant Systems (SysTol)*. IEEE, 2010, pp. 239–244.

[5] D. Xu, F. Zhu, Z. Zhou, and X. Yan, “Distributed fault detection and estimation in cyber–physical systems subject to actuator faults,” *ISA Transactions*, vol. 104, pp. 162–174, 2020.

[6] L. Sha, “Using simplicity to control complexity,” *IEEE Software*, vol. 18, no. 4, pp. 20–28, 2001.

[7] S. Bak, D. K. Chivukula, O. Adekunle, M. Sun, M. Caccamo, and L. Sha, “The system-level simplex architecture for improved real-time embedded system safety,” in *15th IEEE Real-Time and Embedded Technology and Applications Symposium*. IEEE, 2009, pp. 99–107.

[8] S. Mohan, S. Bak, E. Betti, H. Yun, L. Sha, and M. Caccamo, “S3A: Secure system simplex architecture for enhanced security and robustness of cyber-physical systems,” in *the 2nd ACM International Conference on High Confidence Networked Systems*, 2013, pp. 65–74.

[9] J. S. Mertoguno, R. M. Craven, M. S. Mickelson, and D. P. Koller, “A physics-based strategy for cyber resilience of CPS,” in *Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2019*, vol. 11009. International Society for Optics and Photonics, 2019, p. 110090E.

[10] M. A. Arroyo, M. T. I. Ziad, H. Kobayashi, J. Yang, and S. Sethumadhavan, “YOLO: frequently resetting cyber-physical systems for security,” in *Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2019*, vol. 11009. International Society for Optics and Photonics, 2019, p. 110090P.

[11] M. Arroyo, H. Kobayashi, S. Sethumadhavan, and J. Yang, “Fired: frequent inertial resets with diversification for emerging commodity cyber-physical systems,” *arXiv preprint arXiv:1702.06595*, 2017.

[12] 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*, vol. 52, no. 8, pp. 1415–1428, 2007.

[13] M. Pajic, Z. Jiang, I. Lee, O. Sokolsky, and R. Mangharam, “Safety-critical medical device development using the UPP2SF model translation tool,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 13, no. 4s, pp. 1–26, 2014.

[14] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in *18th European Control Conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[15] M. H. Cohen and C. Belta, “Approximate optimal control for safety-critical systems with control barrier functions,” in *59th Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 2062–2067.

[16] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, “Learning safe multi-agent control with decentralized neural barrier certificates,” *arXiv preprint arXiv:2101.05436*, 2021.

[17] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, “Fastrack: A modular framework for fast and guaranteed safe motion planning,” in *56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 1517–1522.

[18] M. Pajic, J. Weimer, N. Bezzo, P. Tabuada, O. Sokolsky, I. Lee, and G. J. Pappas, “Robustness of attack-resilient state estimators,” in *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2014, pp. 163–174.

[19] H. Fawzi, P. Tabuada, and S. Diggavi, “Secure estimation and control for cyber-physical systems under adversarial attacks,” *IEEE Transactions on Automatic control*, vol. 59, no. 6, pp. 1454–1467, 2014.

[20] A. A. Cárdenas, S. Amin, and S. Sastry, “Research challenges for the security of control systems.” in *Proceedings of the 3rd Conference on Hot Topics in Security*, vol. 5. USENIX Association, 2008, p. 15.

[21] L. Niu, D. Sahabandu, A. Clark, and P. Radha, “Verifying safety for resilient cyber-physical systems via reactive software restart,” in *To Appear in ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*. ACM/IEEE, 2022.

[22] P. Larsen, A. Homescu, S. Brunthaler, and M. Franz, “SoK: Automated software diversity,” in *2014 IEEE Symposium on Security and Privacy*. IEEE, 2014, pp. 276–291.

[23] G. S. Kc, A. D. Keromytis, and V. Prevelakis, “Countering code-injection attacks with instruction-set randomization,” in *Proceedings of the 10th ACM conference on Computer and communications security*, 2003, pp. 272–280.

[24] F. Abdi, R. Tabish, M. Rungger, M. Zamani, and M. Caccamo, “Application and system-level software fault tolerance through full system restarts,” in *ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2017, pp. 197–206.

[25] F. Abdi, C.-Y. Chen, M. Hasan, S. Liu, S. Mohan, and M. Caccamo, “Guaranteed physical security with restart-based design for cyber-physical systems,” in *ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2018, pp. 10–21.

[26] R. Romagnoli, P. Griffioen, B. H. Krogh, and B. Sinopoli, “Software rejuvenation under persistent attacks in constrained environments,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 4088–4094, 2020.

[27] T. Arauz, J. Maestre, R. Romagnoli, B. Sinopoli, and E. Camacho, “A linear programming approach to computing safe sets for software rejuvenation,” *IEEE Control Systems Letters*, vol. 6, pp. 1214–1219, 2021.

[28] G. F. Franklin, J. D. Powell, A. Emami-Naeini, and J. D. Powell, *Feedback Control of Dynamic Systems*. Prentice hall Upper Saddle River, 2002, vol. 4.