Department: Head
Editor: Name, xxxx@email

# Online Attack Recovery in Cyber-Physical Systems

**Luis Burbano**
University of California, Santa Cruz

**Kunal Garg**
University of California, Santa Cruz

**Santiago J. Leudo**
University of California, Santa Cruz

**Alvaro A. Cardenas**
University of California, Santa Cruz

**Ricardo G. Sanfelice**
University of California, Santa Cruz

***Abstract*—This document reviews strategies to mitigate attacks against Cyber-Physical Systems and presents a taxonomy to classify them. We then identify trends, insights, and open challenges in this emerging area of research.**

**SECURITY** is a process that includes prevention, detection, and response to attacks. Incident response, the last of these steps, takes a significant role when considering Cyber-Physical Systems (CPS) due to their real-time constraints and safety risks.

An incident response strategy usually requires a plan for collecting and keeping logs of sensitive events, analyzing them to identify the causes, and giving them a priority level [1]. Response strategies then focus on containing the attack, closing vulnerabilities, eradicating the threat, and recovering the system (e.g., re-imaging the affected system). These actions usually involve human analysts and can take hours or even days until the system is fully restored.

While all these actions are needed in an offline review of an attack, in Cyber-Physical Systems, we have real-time requirements that cannot wait until an offline review of the event is completed. For example, an autonomous vehicle under attack may crash or run over pedestrians; similarly, an attack on the power grid can create an immediate cascading outage that leaves a large portion of a country without power for several days. Therefore, in addition to an offline incident response plan, we also need a strategy to mitigate attacks promptly; i.e., we need *online* attack recovery algorithms that keep the system safe and allow it to complete its mission. The scope of this

 Published by the IEEE Computer Society  

paper focuses on real-time CPS reconfigurations for mitigating the impact of an attacker that has partially compromised the system. That is, we focus on online actions over the CPS to decrease the effects of attacks.

Creating an online recovery strategy that mitigates CPS attacks is a challenging task. We need to design new control strategies to overcome additional complications, e.g., increased uncertainty (when system sensors are compromised) or limited actuation capability (when system actuators are compromised). Designing correct and effective responses is crucial, as incorrect response actions can exacerbate the problem.

In this paper, we create a taxonomy to classify attack recovery on CPS from the perspective of control and security. Our contributions include

- We unify and identify the threat model, propose a categorization for the response strategies and identify how these strategies are evaluated.
- This unification allowed us to create new didactic figures illustrating each strategy.
- We crystallize and systematize the work in online recovery for CPS to make clear and definite the current state of the field. With this aim, we present a table summarizing the characteristics of research papers covering more than a decade.
- We summarize the results from the table, discussing trends, insights, and challenges for future directions.

The next three sections outline the main considerations of our proposed taxonomy. In the *Threat Model* Section, we identify the adversary model, specifying the attacker's objective and the specific devices that are considered compromised by the attacker. Then, in the *Attack Mitigation* Section, we classify the different types of attack-recovery proposals. We encountered inconsistencies in the terminology for each strategy, which motivated us to propose a unified categorization. To the best of our knowledge, we are the first paper discussing these strategies from the same perspective for comparison. Third, in the *Evaluation* Section, we study the metrics and experiments used to evaluate the effectiveness of these methods. To this end, we identify the property the system should preserve during attacks. We also study the effect of the recovery strategy both during an attack and without attacks to determine whether researchers assess their strategy thoroughly. We finally apply our taxonomy to online attack recovery papers for CPS, summarize the findings we obtained in the *Discussion* Section, and conclude the paper in the *Conclusion* Section.

Overall we hope this primer can give a quick introduction to newcomers to the field as well as experienced researchers who may appreciate seeing the work on online recovery for CPS contextualized among different alternatives in design and evaluation.

## THREAT MODEL

**Figure 1** presents a schematic of the CPS that we consider in the paper. A CPS integrates a physical process (e.g., water flow) with computational devices (e.g., embedded computer-based controllers). It contains four main elements: 1) the physical process under control, 2) the sensors, which measure physical variables of the process, $y$, 3) the controller, which receives the sensor measurements and computes the control action $u$, and 4) the actuator, which receives the control signal $u$ and modifies the physical process by feeding $v$.

### Attacked Device

The attacker can compromise the information exchanged between sensors, actuators, and controllers of a CPS. There are multiple ways to deploy these attacks: the attackers can spoof devices (if they find authentication weaknesses), they can compromise an end-point (if they find exploitable vulnerabilities in the device), or they can even launch analog attacks by injecting adversarial signals to affect sensor output or actuator actions (e.g., sound waves at specific frequencies can cause a gyroscope to output incorrect sensor measurements). Figure 1 shows a schematic for such attacks.

**Attack on sensors:** The adversary is able to modify sensor measurements $y$. The controller receives a modified version of the measurements $\tilde{y}$. For instance, an attacker may compromise the GPS sensor of a drone to make it crash or land over an adversarial zone.

**Attack on actuators:** The actuator feeds a

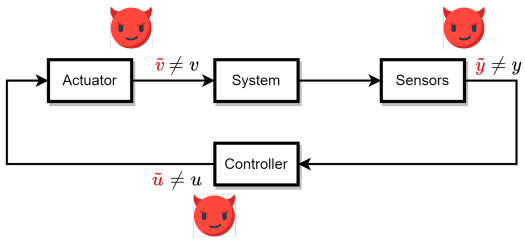


Figure 1: CPS block diagram and model of the adversaries considered in the paper. Variables highlighted in black are the uncompromised variable, while the red variables are the values the attacker might compromise.

modified version of the action $\tilde{v}$ to the system instead of the intended value $v$. An attacker can, for example, compromise the brake system of an autonomous vehicle and produce a crash with other vehicles or pedestrians.

**Attack on the controller:** The controller sends a different control action (denoted as $\tilde{u}$) than the one intended by the original design (denoted as $u$). In this attack, an adversary can send commands to a power grid to disconnect electrical networks, causing blackouts.

### Attacks

We now discuss the ways the attacker can impact the system.

**Availability:** Denial of Service (DoS) attacks focus on compromising data availability. Under such scenarios, the attacker can target 1) the sensors, blocking the controller from receiving sensor measurements; 2) the controller, preventing the actuator from receiving the control commands; or 3) the actuators, preventing the system from performing the desired actions.

**Integrity:** Integrity attacks focus on modifying the data being sent from one component of the CPS to another. Under these attacks, the adversary can modify the controller output, the actuators' action, or the sensors' measurements. Some works consider a particular case of integrity attacks called stealthy attacks [2], [3], [4], [5]; under such attacks, an intelligent attacker that knows how attack-detection works, tries to avoid raising an alert while maximizing the damage.

## ATTACK MITIGATION

We now present the categories we have identified for the works on attack mitigation. We consider whether the attack mitigation is proactive or reactive and provide a taxonomy of the diverse existing recovery methods.

### Adaptation

We identify two types of adaptations to mitigate attacks: proactive and reactive. In **proactive** defenses, the defender anticipates the attacks and considers a mitigation strategy that is always active, even when the system is not under attack. In contrast, **reactive** defenses are only activated after an intrusion detection system raises an alert. Alerts can be generated by real attacks, or they can be false positives (alerts but without malicious activity). Notice that a false positive can activate a real-time reconfiguration of the system, and we need to study the effect of this unnecessary reconfiguration.

We note that several mechanisms can be both proactive and reactive, depending on how they are implemented and utilized. For example, one recovery strategy is to reboot the controller. Depending on the design choices, the reboots can be proactive (e.g., on a periodic schedule) or reactive (after receiving an alert). In Table 1, we point out the mechanisms that researchers use in their recovery mechanisms.

### Recovery method

Next, we illustrate the main patterns we have identified in the literature for responding to attacks.

**Virtual sensors** replace compromised sensors with values of a digital simulation of the physical world [2], [6], [7]. To use virtual sensors, we need an accurate mathematical model of the physics of the system, so that the digital model can follow closely what the physical system would do under a given control command. Highly accurate models are sometimes referred to as *digital twins*. As **Figure 2** illustrates, once the controller receives an alert about an attack in a specific sensor, the recovery system switches from the actual sensors to the digital twin for providing state information to the controller. The authors working on this strategy recommend using the virtual sensors for a time window until an operator is able to respond

to the attack [6], [7]. This operator can determine whether the attack has stopped, and the system can use the physical sensors again.

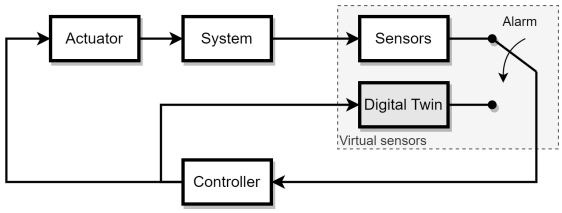


Figure 2: Virtual sensors.

A **Simplex architecture** consists of two *different* controllers [4], [8], [9]: One is a nominal controller (e.g., optimized for performance but without formal guarantees under attack) that computes the system's input when there is no an alert and the second one, termed as *recovery controller*, that takes over after an alert (e.g., a controller that guarantees safety based on formal mathematical analysis). The simplex architecture is different than redundancy or diversity (which we will introduce later on) as the two controllers in the simplex architecture have two very different objectives and designs, whereas, in redundant or diverse implementations, the goal of all the replicas is the same. The simplex architecture is illustrated in **Figure 3**.

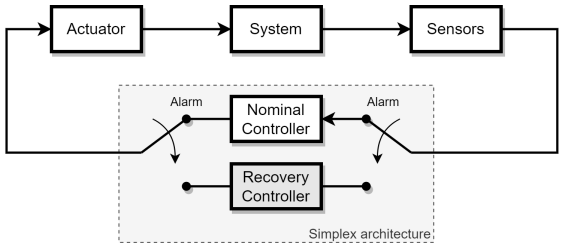


Figure 3: Simplex architecture.

A **Reference monitor** enforces a physical property of the system and checks whether the control commands about to be sent to a physical process violate the property it wants to protect [10]. If there is a predicted violation, the reference monitor blocks or modifies this command, as shown in **Figure 4**. The design of the policy is crucial for this strategy since a legitimate event might potentially trigger a violation and cause a denial of service.

A **Reboot** is a strategy to remove malicious code by restarting the system. While reboots may

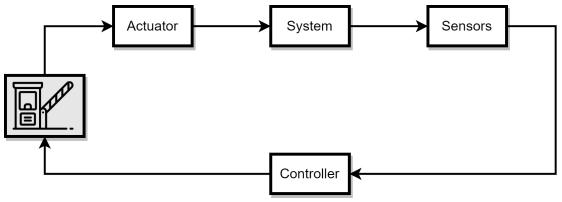


Figure 4: Reference monitor.

not work for certain attacks, they are particularly useful for removing malware residing in memory. Such strategies assume that rebooting the device into a trusted state is possible. Consequently, the attacker loses control of the device, at least for a period after each reboot. While reboots might work on various devices, this strategy has only been studied for protecting the controller, as illustrated in **Figure 5**.

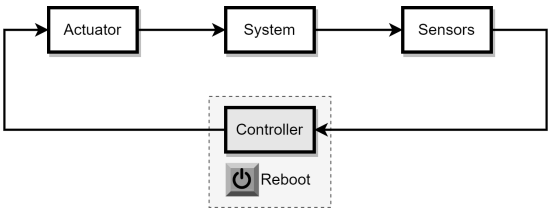


Figure 5: Controller reboot.

Reboot strategies are further divided into **rejuvenation** [11], [12] and **revival** [13], [14]. The rejuvenation reboots are proactive since they frequently restart the system without needing a detection mechanism or an alert. On the other hand, revival reboots are reactive as they only trigger the restart after the attack detector flags an alert.

An **Actuation constraint** restricts what control signals from attackers can do, preventing attackers from driving the system to unsafe places [15]. Instead of blocking the signal like a reference monitor, the defense lets the control command pass to the actuator, but the actuator itself limits what the control signal can do. One popular constraint is to saturate (put bounds) on what the actuator can do, as illustrated in **Figure 6**.

**Redundancy** based approaches implement redundant sensors and/or actuators to provide a backup to the attacked device [5], [13]. When the attacker affects one device, e.g., a sensor, a backup sensor remains available (that hopefully

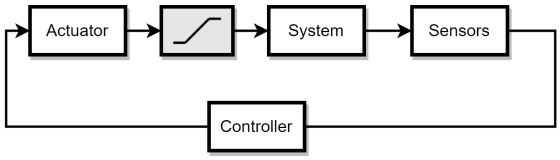

Figure 6: Actuation constraint.

is uncompromised) to provide an unaffected reading.

A **Diversity** based approach complements redundancy approaches. If redundancy approaches are used without diversity, an attacker may be able to compromise all backup devices through the same attack vector. To prevent such scenarios, diversity strategies implement multiple replicas of the controller, each of them with different instruction architectures, so that the same attack cannot be used to gain a foothold in more than one replica [13]. In **Figure 7**, a controller can have, for example, an `x86`-based processor, and a replica may have an `ARM` processor.

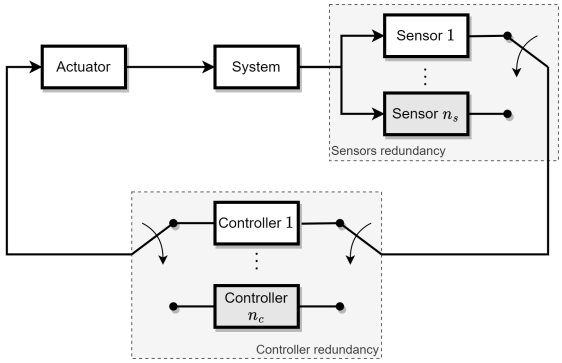

Figure 7: Redundancy and diversity.

## EVALUATION

In this section, we evaluate the methodologies the researchers employed to validate the recovery strategies. We analyze the desirable property they aim to preserve while the system is under attack, how they validate that their strategy successfully preserves such a property, and its effect on the system when there is no attack.

### Security

To measure if a recovery strategy is successful against attacks, we need to define the property the system should preserve even during and after an attack. We identify that the literature focuses on two physical properties: Liveness and Safety.

**Liveness:** The goal of liveness property is for the CPS to fulfill its mission, even under attack. In this case, the researchers allow performance degradation (e.g., a longer time to accomplish the task), but the main goal is to fulfill the original intended mission.

**Safety:** The goal of safety property is to prevent accidents and physical damages. Notice that an autonomous car can achieve its original mission by following a desired path to reach a final destination; however, it can do so while running over a pedestrian. If the primary goal after receiving an alert is to achieve safety, then the controller updates its priority, potentially even de-prioritizing the original mission.

### Adverse byproduct

Most recovery strategies will have negative side effects on the operations of CPS without attacks. Next, we study the negative side effects of the recovery strategy, particularly the impact on the performance when the system is operating without attack. We track whether researchers consider such adverse byproducts in their proposals. In particular, we examine if the authors that proposed *proactive* strategies discussed the **performance degradation** their strategy might produce in the physical system (e.g., actuation saturation might result in a less-responsive CPS). Likewise, we look at whether the authors that propose *reactive* strategies discuss the effect of **false alarms** on the system (e.g. when we switch to a virtual sensor when there is no attack).

### Validation

We also look at the methodology that the authors use to convince readers that their strategies achieve the intended security objectives. Papers mainly use **simulations**, implementation in a **testbed**, or deriving **formal** guarantees through some rigorous mathematical process.

## DISCUSSION

**Table 1** presents the results of applying our taxonomy to several papers. Next, we discuss general insights from these results.

### General Observations

Our first observation is that the field of attack recovery for CPS is a growing area of

Table 1: Taxonomy application on works on online attack recovery.

| | | [6] Cardenas *et al.* | [10] McLaughlin | [11] Abdi *et al.* | [2] Paridari *et al.* | [13] Mertogumo *et al.* | [3] Alhalali *et al.* | [12] Romagnoli *et al.* | [7] Choi *et al.* | [15] Giraldo *et al.* | [8] Zhang *et al.* | [4] Dash *et al.* | [5] Burbano *et al.* | [14] Niu *et al.* | [9] Garg *et al.* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Year** | 2011 - 2018 | | | | 2019 | | | 2020 | | | 2021 | | 2022 | |
| **Threat Model** | **Attacked device** | | | | | | | | | | | | | | |
| | Sensor | ● | - | - | ● | - | ● | - | ● | ● | ● | ● | ● | - | - |
| | Actuator | - | - | - | - | - | - | - | - | ● | - | - | - | - | ● |
| | Controller | - | ● | ● | - | ● | - | ● | - | ● | - | - | - | ● | - |
| | **Attack Objective** | | | | | | | | | | | | | | |
| | Availability attack | - | - | - | - | - | - | - | - | - | ● | - | - | - | - |
| | Integrity attack | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| | *Stealthy attack** | ● | - | - | ● | - | ● | - | - | - | - | ● | ● | - | - |
| **Attack Mitigation** | **Adaptation** | | | | | | | | | | | | | | |
| | Proactive | - | - | ● | - | - | - | ● | - | ● | - | - | - | - | ● |
| | Reactive | ● | ● | - | ● | ● | ● | - | ● | - | ● | ● | ● | ● | ● |
| | **Recovery method** | | | | | | | | | | | | | | |
| | Virtual sensors | ● | - | - | ● | - | - | - | ● | - | ● | - | - | - | - |
| | Simplex architecture | - | - | ● | - | - | - | ● | - | - | ● | ● | - | - | ● |
| | Reboot | - | - | ● | - | ● | - | ● | - | - | - | - | - | ● | - |
| | Reference monitor | - | ● | - | - | - | - | - | - | - | - | - | - | - | - |
| | Actuation constraint | - | - | - | - | - | - | - | - | ● | - | - | - | - | - |
| | Redundancy | - | - | - | - | ● | ● | - | - | - | - | - | ● | - | - |
| | Diversity | - | - | - | - | ● | - | - | - | - | - | - | - | - | - |
| **Evaluation** | **Security** | | | | | | | | | | | | | | |
| | Safety | - | ● | ● | - | ◐ | ● | ● | - | ● | ● | - | - | ● | ● |
| | Liveness | ● | - | - | ● | - | - | - | ● | - | ● | ● | ● | - | - |
| | **Adverse Byproduct** | | | | | | | | | | | | | | |
| | Performance degradation (proactive) | ○ | ○ | - | ○ | ○ | ○ | - | ○ | ● | ○ | ○ | ○ | ○ | - |
| | False alarms (reactive) | ● | ◐ | ○ | - | - | - | ○ | ● | ○ | - | ● | - | - | - |
| | **Validation** | | | | | | | | | | | | | | |
| | Formal | - | - | ● | ● | - | - | ● | - | ● | ● | - | - | ● | ● |
| | Simulation | ● | ● | - | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | - |
| | Testbed | - | - | ● | - | - | - | - | ● | - | - | ● | ● | - | - |

Legend: ●: feature considered by authors. ◐: feature presents ambiguity. -: feature not considered by authors. ○: the feature does not apply. *Stealthy attacks are a particular case of integrity attacks.

research. While we only found one paper on recovery published by 2011 and two by 2013, beginning in 2017 we started to find at least two publications on attack recovery every year. This shows the growing potential of the field, and a natural progression of interest from the research community is to concentrate on response, as the fields focusing on prevention and detection have become more mature.

**Recovery trends:** The early work on recovery between 2011 and 2018 showed special interest in virtual sensors and reboots. However, we noticed that beginning in 2020, the simplex architecture has received special attention. We believe that virtual sensors were used mainly at the initial stage of development of the field because researchers were interested in an intuitive strategy that was general enough to be applicable to most CPS and were not focused on providing formal mathematical guarantees of security or performance. As the field of attack recovery matured, researchers started to use mathematical tools from the control theory to obtain rigorous guarantees, and the simplex architecture provided that opportunity. Similarly, reboots also received attention at the beginning of the field, but the use of this strategy has reduced; while we can use mathematical tools to analyze reboots, the

interest has decreased, in part because of the limited threat reduction applications.

**Most of the attackers affect the integrity of sensors:** We identified that most works deal with attacks on sensors, and all of them consider integrity attacks. Even the strategies that consider DoS attacks model them as integrity attacks since the adversary modifies the information being sent.

**Stealthy attacks:** The works that deal with stealthy attacks implement reactive strategies. By definition, the anomaly detector cannot identify them, and the mitigation strategy cannot modify the system. To evaluate a recovery system that does not identify attacks, the goal is to show that if the attacker wants to remain stealthy, then they cannot damage the system.

**Proactive vs. reactive adaptation:** Reactive strategies require a mechanism to alert about an attack and then modify the system behavior using that information. This makes the reactive defenses' performance dependent on the detector. If the detector cannot identify an attack, the reconfiguration strategy will not modify the system behavior to mitigate the attack. Similarly, a delay in detection might not provide enough time for the defense to reconfigure the system before damage occurs. In contrast, proactive strategies do not require an alert about an attack but may result in conservative approaches that affect the system's performance even without an attack.

**Recovery should guarantee safety and liveness:** Most works attempt to ensure either safety or liveness separately, but the satisfaction of one of these properties does not guarantee the satisfaction of the other. We argue that the recovery strategies should ensure both properties. Zhang *et al.* [8] propose a backup controller that solves an optimization problem that includes the unsafe and target set. The solution steers the system to the target set while avoiding unsafe regions. If the target set is the original destination of the system, we can count this approach as satisfying both liveness and safety.

**Validation Trade-offs** The methods that authors use to validate their strategy by mathematical analysis, using simulations or testbeds, have various advantages and disadvantages. Simulations allow the testing of several scenarios without the risk of damaging a real CPS or its environment of operation. However, models and simulators cannot emulate the real physical dynamics. Instead, the testbed-based validation uses a real CPS and provides a better understanding of the system performance, but it is costly and involves high risk. Nevertheless, these two validations are empirical approaches. Works that use these validation methods cannot characterize the conditions that allow the strategy to work in all cases, as it is impossible to run such experiments over all possible scenarios.

**Assumptions:** Works that use mathematical analysis can characterize the conditions under which the strategy ensures liveness or safety. To obtain those guarantees and make the problem mathematically tractable, the authors need to make assumptions about the CPS characteristics, which may not apply to every CPS. We identified two main assumptions in such works. First, most methods assume a CPS with linear dynamics as the system model because several tools exist to analyze them [2], [8], [11], [12], [15]. Responses require more realistic models that can deal with nonlinear dynamics, but only a few works deal with general nonlinear models [9] or a special class of nonlinear systems [14]. Second, these works make assumptions about the noise on the sensors' measurements or process. Some of them consider bounded noises [8], [9] or a noiseless system altogether [12].

**No common evaluation metric:** We found no standard metric for measuring the system's performance when attacked. Papers considering safety mainly show that the system states are always inside the safe set [9], [11]. The works prioritizing liveness require a different metric to measure that the system completes its designed task. The works we reviewed, however, use different metrics to define success.

**Recovery strategies affect operations without attacks:** We expect that attacks on CPS will be rare, so a CPS will usually operate without attacks. Therefore, defense strategies should minimize any negative side effects their implementation may cause during operations without attacks. Proactive strategies usually result in conservative designs that impact performance despite not being under attack. While the authors of proactive strategies usually mention that their proposal impacts performance, most of them do not quantify this performance degradation.

**Effects of false alarms:** With reactive strategies, the CPS can use devices and controllers that maximize performance during operation without attacks. However, the detection mechanism may send a false alert activating the reactive defenses, impacting the system's performance. We identified that only a few works discuss the response to a false alarm [4], [6], [7], and the metric to evaluate the impact of false alarms is not unified.

As illustrated in the last two paragraphs, most research efforts focus on proving that their proposal is secure; however, they tend not to measure the negative performance degradation of the system when it is not under attack. We argue that we need more research to show that the defenses do not affect the operation of a system under normal conditions.

### Pros and Cons of Each Strategy

Lastly, we discuss the advantages and disadvantages of each strategy. Virtual sensors are one of the simplest reactive attack mitigation strategies, which are effective only against sensor attacks. The digital twin model requires only a mathematical model of the physical evolution of the system under different control commands. However, this strategy can only work for a short time if there is not enough redundancy or uncompromised sensors; otherwise, the system eventually becomes an open-loop control system, as it stops using the sensor measurements. Open-loop controllers do not have stability guarantees, and the system states might reach unsafe zones that can harm the system.

With the simplex architecture, designers might synthesize the controllers and the switching to guarantee that the system has some properties (e.g., safety). Depending on the design choices, this strategy might mitigate the attack effects when the adversary compromises either the actuator or the controller. However, designing the controllers might be challenging, and as we discussed above, some formulations are infeasible.

A reference monitor enforces an access control policy. A safety policy should prevent the execution of control commands that will damage the system. However, CPS has continuous and discrete variables, which makes it difficult to predict if a given control action might lead the system to an unsafe state. Defining a safety policy is also time-consuming and error-prone.

Most of the works that deal with attacks against the controller use reboots. However, during a reboot, the system is without control which clearly impacts the performance if there are no attacks. Additionally, rebooting cannot ensure removing the adversary, who may take control of the system again with persistent rootkits. For instance, the adversary might modify the controller ROM and persist even after the reboot if there is no integrity verification of the firmware (e.g., using digital signatures). Even if an attacker cannot modify the ROM content, the firmware might have vulnerabilities that the attacker can exploit. Moreover, attackers may persist if they add hardware to the controller to deploy a physical attack.

Redundancy requires more hardware devices, which is a disadvantage as it increases the cost. Additionally, redundancy strategies would not be able to recover from attacks against all the hardware copies. Diversity makes it more difficult to implement the attacks against all the devices, but it also requires additional efforts to design the CPS. As noted before, the nominal controller of the replica is the same as the original controller, so under an attack, there are no guarantees that the system will provide safety. On the other hand, in the simplex architecture, the recovery controller has a different objective from the primary controller, which is to take the system to a safe place.

**Strategy mixing:** We notice that combining attack mitigation strategies can give us advantages from each of them. Such a combination is not straightforward, and we found that there is almost no mixing of strategies to mitigate attacks. For instance, Garg et al. [9] present a combination of strategies by proposing a risk-based adaptation of the system; rather than responding to fixed alerts, the anomaly detector gives a risk score depending on how close it is to the unsafe regions. The control adaptation then becomes more and more aggressive if the anomaly detector deems the system to be closer and closer to the unsafe regions. We encourage more researchers to study response strategies that soundly combine the best practices of each method.

## CONCLUSIONS

As the level of maturity of research in CPS security continues to grow, we find that online attack recovery is becoming a more relevant area of study. In this article, we introduced a new taxonomy and created new diagrams to illustrate in a unified lens the variety of proposals in this area. We hope that our work will help researchers in the field who are new to the topic as well as experienced researchers in CPS security.

We argue that future proposals can consider combining the best practices of different methods, as some of these methods complement each other well, like using virtual sensors with simplex architecture or new risk-based adaptations.

## ACKNOWLEDGMENT

Research was partially sponsored by NSF CNS-1929410, 1931573, and 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.

Research by R. G. Sanfelice partially supported by NSF Grants no. CNS-2039054 and CNS-2111688, by AFOSR Grants no. FA9550-19-1-0169, FA9550-20-1-0238, and FA9550-23-1-0145, by AFRL Grant nos. FA8651-22-1-0017 and FA8651-23-1-0004.

## References

1. M. Powell, M. Pease, K. Stouffer, C. Tang, T. Zimmerman, J. Hoyt, S. Saravia, A. Sherule, L. Wilcox, and K. Zheng. (2022) Responding to and recovering from a cyber-attack: Cybersecurity for the manufacturing sector. https://www.nccoe.nist.gov/sites/default/files/2022-11/mfg-recovery-project-description-final.pdf.
2. K. Paridari, N. O'Mahony, A. E.-D. Mady, R. Chabukswar, M. Boubekeur, and H. Sandberg, "A framework for attack-resilient industrial control systems: Attack detection and controller reconfiguration," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 113–128, 2017.
3. S. Alhalali, C. Nielsen, and R. El-Shatshat, "Mitigation of cyber-physical attacks in multi-area automatic generation control," *International Journal of Electrical Power & Energy Systems*, vol. 112, pp. 362–369, 2019.
4. P. Dash, G. Li, Z. Chen, M. Karimibiuki, and K. Pattabiraman, "Pid-piper: Recovering robotic vehicles from physical attacks," in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2021, pp. 26–38.
5. L. Burbano, L. F. Cómbita, N. Quijano, and S. Rueda, "Dynamic data integration for resilience to sensor attacks in multi-agent systems," *IEEE Access*, vol. 9, pp. 31 236–31 245, 2021.
6. A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proceedings of the 6th ACM symposium on information, computer and communications security*, 2011, pp. 355–366.
7. H. Choi, S. Kate, Y. Aafer, X. Zhang, and D. Xu, "Software-based realtime recovery from sensor attacks on robotic vehicles," in *Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, 2020, pp. 349–364.
8. L. Zhang, X. Chen, F. Kong, and A. A. Cardenas, "Real-time attack-recovery for cyber-physical systems using linear approximations," in *Proceedings of the 2020 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 2020, pp. 205–217.
9. K. Garg, R. G. Sanfelice, and A. A. Cardenas, "Control barrier function-based attack-recovery with provable guarantees," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 4808–4813.
10. S. McLaughlin, "Cps: Stateful policy enforcement for control system device usage," in *Proceedings of the 29th Annual Computer Security Applications Conference*, 2013, pp. 109–118.
11. 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 *Proceedings of the 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2018, pp. 10–21.
12. R. Romagnoli, B. H. Krogh, and B. Sinopoli, "Design of software rejuvenation for cps security using invariant sets," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 3740–3745.
13. J. S. Mertoguno, R. M. Craven, M. S. Mickelson, and D. P. Koller, "A physics-based strategy for cyber resilience of cps," in *Proceedings of the Autonomous*

*Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2019*, vol. 11009. International Society for Optics and Photonics, 2019, p. 110090E.

14. L. Niu, D. Sahabandu, A. Clark, and P. Radha, "Verifying safety for resilient cyber-physical systems via reactive software restart," in *Proceedings of the 2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*, 2022.
15. 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.

**Luis Burbano** is a Ph.D. student in the Department of Computer Science and Engineering at the University of California, Santa Cruz. He is interested in the security of cyber-physical systems. Contact him at lburbano@ucsc.edu.

**Kunal Garg** is a postdoctoral associate in the REALM lab in the Department of Aeronautics and Astronautics, MIT. This work is done when he was a postdoctoral fellow at the University of California at Santa Cruz. His research interests include Control-theoretic methods for the security of CPS, optimization-based control design, Multi-agent distributed control, Finite-time stability theory for safe distributed control design, Hybrid/Switched-Systems: theory and applications, and Accelerated optimization methods. Contact him at kgarg@mit.edu

**Santiago J. Leudo** is a Ph.D. candidate in Electrical and Computer Engineering, Robotics, and Control, working with Ricardo Sanfelice at the Hybrid Systems Lab at the University of California, Santa Cruz. His research interests include the connections between control and game theory for scenarios in which the dynamics of the systems might exhibit both continuous and discrete behaviors. Contact him at sjimen28@ucsc.edu

**Alvaro A. Cardenas** is an Associate Professor of Computer Science and Engineering at the University of California, Santa Cruz. His research interests focus on cyber-physical systems and IoT security and privacy, including autonomous vehicles, drones, smart home devices, and SCADA systems controlling the power grid and other critical infrastructures. Contact him at alacarde@ucsc.edu

**Ricardo G. Sanfelice** is a Professor at the Department of Electrical and Computer Engineering, University of California at Santa Cruz. His research interests are in modeling, stability, robust control, observer design, and simulation of nonlinear and hybrid systems with applications to power systems, aerospace, and biology. Contact him at ricardo@ucsc.edu