# A survey on cyber-resilience approaches for cyber-physical systems

Mariana Segovia-Ferreira, Jose Rubio-Hernan, Ana Cavalli, Joaquin Garcia-alfaro

# A Survey on Cyber-Resilience Approaches for Cyber-Physical Systems

MARIANA SEGOVIA-FERREIRA*, JOSE RUBIO-HERNAN*, ANA ROSA CAVALLI*, and JOAQUIN GARCIA-ALFARO*, SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, France

Concerns for the resilience of Cyber-Physical Systems (CPS) in critical infrastructure are growing. CPS integrate sensing, computation, control, and networking into physical objects and mission-critical services, connecting traditional infrastructure to internet technologies. While this integration increases service efficiency, it has to face the possibility of new threats posed by the new functionalities. This leads to cyber-threats, such as denial-of-service, modification of data, information leakage, spreading of malware, and many others. Cyber-resilience refers to the ability of a CPS to prepare, absorb, recover, and adapt to the adverse effects associated with cyber-threats, e.g., physical degradation of the CPS performance resulting from a cyber-attack. Cyber-resilience aims at ensuring CPS survival, by keeping the core functionalities of the CPS in case of extreme events. The literature on cyber-resilience is rapidly increasing, leading to a broad variety of research works addressing this new topic. In this article, we create a systematization of knowledge about existing scientific efforts of making CPS cyber-resilient. We systematically survey recent literature addressing cyber-resilience with a focus on techniques that may be used on CPS. We first provide preliminaries and background on CPS and threats, and subsequently survey state-of-the-art approaches that have been proposed by recent research work applicable to CPS. In particular, we aim at differentiating research work from traditional risk management approaches, based on the general acceptance that it is unfeasible to prevent and mitigate all possible risks threatening a CPS. We also discuss questions and research challenges, with a focus on the practical aspects of cyber-resilience, such as the use of metrics and evaluation methods, as well as testing and validation environments.

CCS Concepts: • **General and reference** → **Surveys and overviews**; Reliability; • **Computer systems organization** → **Reliability**; *Availability*; *Maintainability and maintenance*; • **Software and its engineering** → **Software fault tolerance**; • **Security and privacy** → **Access control**; **Authorization**; **Intrusion detection**; **Mitigation**; • **Networks** → Network reliability.

Additional Key Words and Phrases: Cyber-Physical System, Critical Infrastructure, Cyber Security, Cyber-Resilience, Dependability, Attack Mitigation, Graceful Degradation.

**ACM Reference Format:**
Mariana Segovia-Ferreira, Jose Rubio-Hernan, Ana Rosa Cavalli, and Joaquin Garcia-Alfaro. 2024. A Survey on Cyber-Resilience Approaches for Cyber-Physical Systems . 1, 1 (May 2024), 36 pages. https://doi.org/10.1145/3652953

## 1 INTRODUCTION

Traditionally, the design of industrial systems was based on an isolation model, where the control of the operational technology was separated from the information technology. Today, both operational and information technology are integrated. Industrial physical processes are controlled by Cyber-Physical Systems (CPS) that integrate modern

Authors' address: Mariana Segovia-Ferreira, segovia@telecom-sudparis.eu; Jose Rubio-Hernan, rubio_he@telecom-sudparis.eu; Ana Rosa Cavalli, ana.cavalli@telecom-sudparis.eu; Joaquin Garcia-Alfaro, joaquin.garcia_alfaro@telecom-sudparis.eu, SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, 19 place Marguerite Perey, Palaiseau, France, 91120.

computation and networking resources into traditional physical environments. They have emerged mainly in the industrial control system domain, using data acquisition and processing over networked control systems [1] to automate the remote execution of industrial tasks [2]. Such integration has several advantages, for example, low maintenance costs, high reliability, flexibility, efficiency, and effectiveness to control the physical process [3]. The use of computation and networking resources to build a new generation of CPS plays an important role in current critical nationwide infrastructures, such as electrical transmissions, energy distribution, manufacturing, supply chain, waste recycling, public transportation, health care, industrial process control, water infrastructure, and several others [1, 4].

CPS are composed of a physical process, sensors, actuators, and controllers. The sensors collect information about the physical process and send it to the controllers. Then, the controllers analyze the received information and calculate how to optimize the behavior of the physical process. As a result, the controllers send commands to the actuators to execute the corrective actions on the physical process. For example, to maintain the stability of the physical processes. However, CPS can be disrupted by cyber-physical attacks [5, 6], i.e., situations resulting from a cyber-attack, but manifesting physical effects, such as performance degradation [7]. These situations may put human safety at risk, cause harm in natural environments, interrupt industrial process continuity, and violate environmental regulations. Hence, cyber-physical attacks can lead to large economic losses, generate legal problems, and damage the reputation of the affected organizations [8]. Many concerns have been raised about the vulnerabilities of control systems. Recent history provides several cases of attacks on industrial infrastructures, which illustrate the threat that they represent. In particular, the security of industrial CPS is drawing great attention after the Stuxnet malware [9, 10] that considerably affected the performance of a uranium enrichment plant. The consequences of this event showed the dangers of successful cyber-threats carried out against CPS. Also, the well-known Ukraine attack [11] targeted power distribution networks causing outages as well as lasting damage. Another example is the Australian water services attacked by a disgruntled employee who infiltrated the system network and altered the control signals [12]. The adversary took control of 150 sewage pumping stations resulting in the evacuation of one million liters of untreated sewage, over three months, into stormwater drains and onto local waterways. More examples of similar events can be found in [13].

Although pure cyber-attacks have shown limited damages to recent CPS [14], full damages are feasible when considering adversaries that perpetrate control-theoretic manipulation, resulting from cyber-attacks, but leveraging physical disruption. This puts the focus on cyber-physical integrity attacks, which can rapidly move the system to unsafe states. Ensuring the control of CPS data exchanges is a challenging problem that requires a combination of both network and industrial control security. In addition, cyber-physical attacks may be hard to detect [15, 16]. For this reason, resilience[1] is especially relevant [17]. Developing CPS that can safely survive an attack is a current challenge [18].

Ensuring safety using only information security tools is not enough in the CPS domain. Cybersecurity approaches do not cover all the possible vulnerabilities in the cyber components. For example, specific vulnerabilities may not have remediation mechanisms or they may be too expensive to implement. Even when the approach is implemented, detection algorithms are not free of false negatives and the remediation techniques may not be triggered. As pointed out in [19], large research efforts have focused on intrusion detection for CPS, but there is little discussion about what to do after the intrusion is detected, i.e., in remediation approaches that mitigate the effects of an attack. Most of the responses are manual or hardwired with a fixed response that cannot be configured. For this reason, attack tolerance should be enforced in critical systems to provide a correct service in the presence of successful attacks against the system [20]. The resulting CPS should satisfy high availability[2] requirements to guarantee the execution of the critical

[1] In this article, we use the words *resilience* and *cyber-resilience* indifferently.
[2] In our work, availability means that legitimate users and processes have access to the system (and the resources of the system) whenever they need.

tasks. It should be able to guarantee that the whole system remains operational even in the presence of attacks, even if that means working under graceful degradation modes. As a result, cybersecurity approaches should be complemented by secure control theory that provides attack models and a description of the interaction between the physical world and the control system. This will provide a better understanding of the attacks' consequences, development of new detection methods, response mechanisms, and architectures. It will also make the control systems more resilient to possible attacks and failures.

In this article, we focus on cyber-resilience techniques to build CPS tolerant to cyber-physical attacks. We consider that the CPS is a combination of cyber and physical components working together under discrete and continuous industrial environments [21]. We devote our work to protection techniques addressing networked control systems, i.e., a subset of CPS dedicated to industrial control processes, usually performing critical functions. We analyze strategies that combine or have the potential to combine cybersecurity and control-theoretic approaches to build a solution that contemplates the cyber and the physical components of a CPS to face the challenges created by cyber-physical adversaries. We differentiate research work from traditional risk management approaches, based on the general acceptance that it is unfeasible to prevent and mitigate all possible risks threatening a CPS. We also discuss questions and research challenges, with a focus on the practical aspects of cyber-resilience, such as the use of metrics and evaluation methods, as well as testing and validation environments.

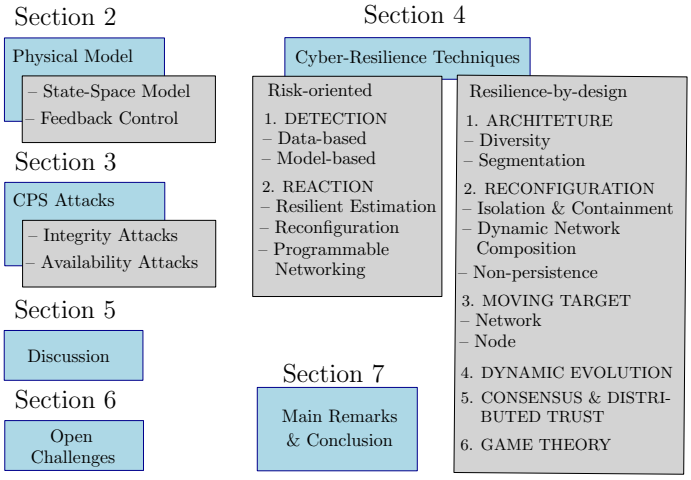


Fig. 1. Organization of this article.

The remainder of this article is outlined as follows. Section 2 explains how to model a CPS and define the feedback control executed in the CPS controller. Section 3 provides a control-theoretic model of the cyber-physical integrity and availability attacks that we address in this article. Section 4 provides our literature survey on cyber-resilience techniques to address the previously defined attacks. The selected literature was analyzed and classified based on risk-oriented techniques and resilience-by-design techniques. These two approaches are closely related. Remediation techniques are sometimes considered part of cyber-resilience. For this reason, we analyze the difference between them and we classify the collected proposals into two categories – (1) *detection and reaction* techniques, and (2) *resilience-by-design* proposals. The approaches in each category are further classified into subcategories, as depicted in Fig. 1. Sections 5 and 6 discuss

research challenges and present open issues in the cyber-resilience area to lead future research. Section 7 concludes the article with our conclusions and main remarks.

## 2 BACKGROUND ON CYBER-PHYSICAL SYSTEMS

Cyber-Physical Systems (CPS), mathematically modeled in our work as networked control systems, are composed of distributed control systems and autonomous agents that need to make decisions in real time. They consist of two main parts. First, a cyber layer, containing the computing and network functionalities. Second, a physical layer, representing dynamic automation processes. Both together manage the distributed resources that monitor the behavior of physical phenomena and take the necessary actions to get control over them [1]. The CPS becomes easier to automate at the cost of increasing the interaction between physical and cyber layers [2]. However, as a consequence, they become more vulnerable to new threats. Malicious actions in these systems are usually conducted by cross-layer adversaries that aim at harming the physical processes through the integration of physical and cyber layer attacks to cause, e.g., physical damages [13].

CPS use a model able to manage and control the physical evolution of the system states. Controlling the states is a challenge since they follow the laws of the involved physical process, e.g., energy, water, or moving systems [22]. For this reason, the physical properties of the system are used to create a model represented for the feedback control. This feedback control has to be able to regulate and manage the behavior of the system, i.e., a model able to confirm that the commands sent to the physical layer are executed correctly and the information coming from the physical states (through the sensors) is consistent with the predicted behavior of the system.

In a CPS, the *plant* (also referred to as *system* by some authors) is the physical process that we want to control. The *actuators* perform physical actions over that process and the *sensors* collect the modifications produced at the physical layer. Using the data collected by the sensors, the feedback *controller* generates a residue between the data received from the sensors and the reference obtained after modeling the system. This residue, named *control error* by some authors, is used by the controller to create the *control input* to rectify, if necessary, the physical states using the actuators. The threat models explained in Section 3 use some of the parameters and equations explained next.

### 2.1 Physical Model

How to obtain the model used in the feedback controller is a well-known problem in the control domain. Different techniques have been developed to provide a reference and generate the control input at each time step [23–26]; and also to create feedback control [27–29]. The model can be obtained using a representation that relates to each possible input signal, the corresponding output signal. The two main mathematical approaches to model this are the *transfer function* and the *state-space model*. Both representations are equivalent since they are based on the differential equations that model the behavior of the physical process being controlled.

Normally, a CPS design process starts with the transfer function since it is the most direct form starting from the differential equations of the physical process. The transfer function $G(s)$ is the ratio of the Laplace transformation using the complex variable $s$ of the output $Y(s)$ to that of the input $U(s)$. It is represented as shown in Equation (1) by the division of two polynomials, the numerator is created by taking the coefficients $b_i$ of the output differential equation and the denominator using the coefficients $a_i$ of the input differential equation.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\sum_{i=0}^{m} b_i s^{m-i}}{\sum_{i=0}^{n} a_i s^{n-i}} \tag{1}$$

A transfer function with multiple inputs and multiple outputs is usually represented in a matrix which indicates the relationship of each input and each output of the system. Using well-known control theory techniques [30], it is possible to transform the transfer function into a state-space model by expressing the differential equations into matrices forms, cf. Equation (2) as follows:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + v_k \end{aligned} \tag{2}$$

where $x_k \in \mathbb{R}^n$ is the vector of the state variables at the $k$-th time step, $u_k \in \mathbb{R}^p$ is the control signal and $w_k \in \mathbb{R}^n$ is the process noise that is assumed to be a zero-mean Gaussian white noise with covariance $Q$, *i.e.* $w_k \sim N(0, Q)$. Controllers are normally implemented in a discrete form.

Moreover, $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times p}$ are respectively the *state* matrix and the *input* matrix. The value of the output vector $y_k \in \mathbb{R}^m$ represents the measurements produced by the sensors that are affected by a noise $v_k$ assumed as a zero-mean Gaussian white noise with covariance $R$, *i.e.* $v_k \sim N(0, R)$ and $C \in \mathbb{R}^{m \times n}$ is the output matrix that maps the state $x_k$ to the system output.

### 2.2 Feedback Control

The previous equations define mathematically the behavior of a physical system. These equations are used by the feedback control to generate a closed-loop system. The output of the feedback control influences the input signal, e.g., to rectify the possible errors generated by the system. To build this type of feedback, two relevant mechanisms are *Proportional-Integral-Derivative* (PID) controllers and *Linear Quadratic Gaussian* (LQG) controllers. LQG controllers provide feedback that holds better results than PID controllers [31]. LQG is a well-known technique for designing optimal dynamic feedback control laws. This optimal solution combines a Linear-Quadratic Estimator (LQE) with a Linear-Quadratic Regulator (LQR). These two components are independent but work together taking into account the measurement noise and process disturbance.

The goal of an LQG controller is to produce a control law $u_k$ such that a quadratic cost $J$, that is a function of both the state $x_k$ and the control input $u_k$, is minimized:

$$J = \lim_{n \to \infty} E\left[\frac{1}{n}\sum_{i=0}^{n-1}(x_i^T \Gamma x_i + u_i^T \Omega u_i)\right] \tag{3}$$

where $\Gamma$ and $\Omega$ represent positive definite cost matrices [32].

It is well-known that a *Kalman filter*-based LQE can be combined with a traditional LQR to solve the aforementioned control problem, as follows:

(1) *Kalman filter*-based LQEs use noisy measurements and produce an optimal state estimation $\hat{x}_k$ of $x$ (state);
(2) the LQR, based on the state estimation $\hat{x}_k$, provides the control law $u_k$ that solves the problem (cf. Equation (3)).

A Kalman filter can estimate the state as follows:

- Predict (*a priori*) system state $\hat{x}_{k|k-1}$ and covariance:

$$\begin{aligned} \hat{x}_{k|k-1} &= A\hat{x}_{k-1} + Bu_{k-1} \\ P_{k|k-1} &= AP_{k-1}A^T + Q \end{aligned}$$

- Update parameters and (*a posteriori*) system state and covariance:

$$\begin{aligned} K_k &= (P_{k|k-1}C^T)(CP_{k|k-1}C^T + R)^{-1} \\ \hat{x}_k &= \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1}) \\ P_k &= (I - K_kC)P_{k|k-1} \end{aligned}$$

where $K_k$ and $P_k$ denote, respectively, the Kalman gain and the *a posteriori* error covariance matrix, and $I$ is the identity matrix of appropriate dimensions.

The optimal control law $u_k$ provided by the LQR is a linear controller: $u_k = L\hat{x}_k$, where $L$ denotes the feedback gain of the LQR that minimizes the control cost (cf. Equation (3)), which is defined as follows [33]:

$$L = -(B^T S B + \Omega)^{-1} B^T S A$$

with $S$ being the matrix that solves the following discrete-time algebraic Riccati equation:

$$S = A^T S A + \Gamma - A^T S B [B^T S B + \Omega]^{-1} B^T S A$$

## 3 BACKGROUND ON CYBER-PHYSICAL THREATS

Control systems use safety mechanisms to handle failures and avoid accidents. Nevertheless, these control mechanisms cannot detect intentional malicious actions, such as cyber-physical attacks. Next, we present some existing cyber-physical adversary models and attack families.

### 3.1 Adversary Models

The consequences of a successful cyber-physical attack can be more damaging than aggression on other networks because control systems are at the core of many critical infrastructures. We differentiate three main adversary models [34]:

- **Physical Adversary –** The adversary has physical access to the CPS and can damage it by performing physical actions. For example, the adversary may cut the brakes of a connected autonomous car, destroy the valves that release the pressure in an industrial system, or perturb temperature sensor measurements by modifying their local surroundings [6, 35].
- **Cyber Adversary –** The adversary can perform cybersecurity attacks (e.g., man-in-the-middle, buffer overflow, shell exploits, or others). The adversary has only knowledge about computation, storage, and network resources. Because of that, the attack can be easily detected by control-theoretic fault detection techniques [36]. Authors have systematized existing CPS security research analyzing the taxonomy of threats, vulnerabilities, and attacks from the CPS components perspective, with a special focus on cyber components [37] and cyber adversaries [8]. They also present the main difficulties and solutions in the estimation of the consequences of cyber-attacks, in terms of modeling, detection, and the development of security architectures.
- **Cyber-Physical Adversary –** The adversary perpetrates cyber-attacks to cause tangible damage to physical components, for instance, by adding disturbances to a physical process via the exploitation of vulnerabilities in some computing and networking resources of the system. The cyber-physical adversary is a combination of the two previous adversaries [38]. First, the adversary uses a cyber-attack to gain position in the system from a remote location. Then, the adversary learns about the physical model to generate an attack with physical consequences but without being physically placed in the CPS physical location. It can be hard to detect and locate a cyber-physical adversary, whose attacks may often be confused with faults in the system.

### 3.2 Attack Families

Different cyber-physical attack families have been reported in the literature. Authors in [14] provide control-theoretic models for integrity and denial-of-service (DoS) attacks. Similar techniques have been reported in [39], naming them deception and disruption cyber-physical attacks, respectively. The work in [14] shows that a traditional DoS attack

does not have a significant effect when the system is in a steady state. However, the violation of integrity properties in such attacks can rapidly move the system to unsafe states.

A convenient attack classification in the existing literature is the one proposed in [6], which introduced the attack space as a three-dimensional graphical characterization of the attacks. It considers the following three dimensions: the adversary's a priori knowledge of the system's model, the disruption of resources, and the disclosure of resources. The knowledge of the system's model allows the adversary to develop sophisticated attacks, which have more severe consequences and are harder to detect with traditional approaches. The disclosure of resources lets the adversary to obtain sensitive information, which may be used to generate knowledge about the system, but cannot be used to disrupt the system's operation. Finally, the disruption of resources can be used to affect the system's operation (e.g., maintaining the stability of the system).

Fig. 2 depicts block diagrams representation of cyber-physical adversaries attacking a control loop. The $\bigoplus$ symbol represents a *summing junction*, i.e., the sum of input signals. To take control of the physical process, the adversary may send a malicious command $u_{attack}$ to the *System* that will be executed by the actuators. After that, to deceive the controller and go unnoticed, the adversary may modify the sensors' readings $y_{attack}$ to inject a measurement value $y$. The adversary may use a combination of different commands $u$ and measurements $y$ to deceive the controller and damage the system.

Next, we outline some cyber-physical attacks following the taxonomy presented in [6]. Cyber-physical adversaries use integrity attacks to exploit vulnerabilities in the control mechanism and take control of the physical process. For this reason, all the attacks are assumed to inject malicious traffic. However, they are classified into different categories because they exploit different vulnerabilities in the control loop. As a consequence, these attacks produce different effects on the physical process and they may require different approaches to be solved.

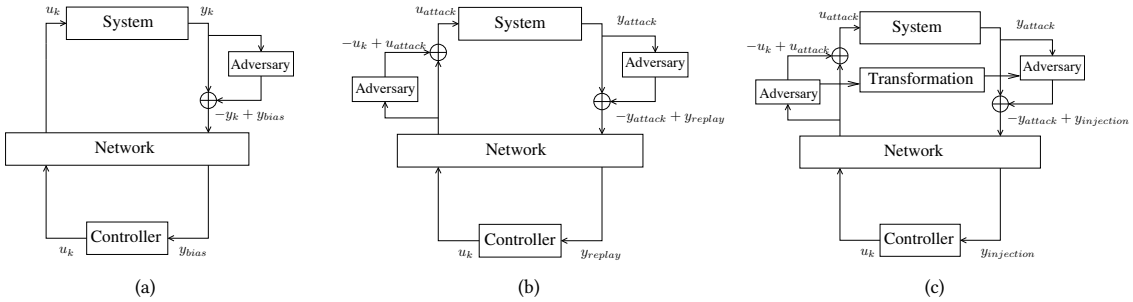


Fig. 2. (a) Stealth attack. (b) Replay attack. (c) Covert attack.

*3.2.1 False-Data Injection Attack or Stealth Attack.* In this attack family (cf. Fig. 2(a)), the adversary modifies some sensor readings by applying physical interference, at the sensor device, or by perturbing the communication channel to disrupt the system [5, 13, 40]. To carry out attacks from this family, the adversary needs knowledge about the behavior of the system, such as the system dynamics, the command signals, and the control detection threshold. The adversary drives slowly the control decisions out of the correct behavior and produces wrong control decisions to cause a malfunction in the system. From a control-theoretic perspective, the injected false data should not affect the system residues (cf. Section 2). This means that the injected data should not alter the sensor measurement variations. Otherwise, the attack would be easily detected.

*3.2.2 Replay Attack.* Fig. 2(b) shows adversaries conducting a cyber-physical replay attack by modifying some sensor readings (e.g., by replicating previous measurements, corresponding to normal operating conditions). Then, the adversaries modify the control input to affect the system state. These adversaries are not required to know the system process model, but access to all the sensors is required to carry out a successful attack. This type of adversary is undetectable with a monitor detector which only verifies sensors' measurements. To detect the attack, it is required to add some protection to the input control signal $u_k$ [41], defined in Section 2.

*3.2.3 Covert Attack.* Adversaries, depicted in Fig. 2(c), read and add to both, the control data and the sensor measurements. The difference with the replay attack is that the adversary needs *a priori* knowledge about the system process to create a transformation that is correlated with the control model, i.e., the attack requires knowing the behavior of the physical system as well as the behavior of the feedback control. This type of adversary is considered undetectable if measurements are compatible with the physical process. In other words, the attack cannot be distinguished from the regular system operations [36].

*3.2.4 DoS Attack.* A denial-of-service (DoS[3]) aims at disrupting the communication between the remote elements (e.g., elements related to supervisory control and data acquisition protocols) and local elements closely related to the system (e.g., terminal units and programmable logic controllers connected to the sensors and actuators of the system), hence disrupting the availability of feedback control [42]. By disconnecting the controller from the physical device, it is possible to avoid the process monitoring and make the system vulnerable to other malicious actions [43].

It is worth noting that cyber-physical DoS attacks are launched using integrity attacks to cause significant damage. In this case, the attack compromises the integrity of the messages, as shown in Fig. 2(b), with two objectives. First, to disrupt the communication between the controller and the system, generating a loss of system's supervision that may be not easy to detect. Second, to inject malicious messages to move the system from the stability point. This way, the adversary generates an unavailability of the system to the authorized users to make it available just for the malicious actions. As a result, this adversary affects the integrity of the system to generate also an availability problem.

*3.2.5 Command Injection Attack.* This attack uses the protocols and device vulnerabilities to inject false commands into the control systems to disrupt control actions or system settings. It is similar to the attack shown in Fig. 2(a), but the adversary injects the malicious traffic in the control command, i.e., in $u_k$. For example, by overwriting the remote registers associated with some supervisory control or exploiting the data acquisition protocols [44].

*3.2.6 Zero Dynamics Attack.* This attack family assumes vulnerabilities present in the dynamics of the system concerning properties used to monitor and control the behavior. This attack is similar to the command injection attack, but it makes an unobservable state unstable and disrupts this unobservable part of the system without being detected by the controller [6, 45]. A solution to avoid this kind of attack is to update the architecture of the system to make all the states observable, e.g., by deploying more sensors to avoid unobservable situations in the system.

As we have seen in this section, the existence of availability and integrity vulnerabilities is the main security issue in CPS. Although pure cyber-attacks may have a limited impact on the system, combined with control-theoretic strategies may cause important physical damages [14]. Indeed, cyber-physical integrity attacks can rapidly move the system to unsafe states. Also, cyber-physical DoS attacks can benefit from integrity issues, to cause significant damages. In this case, the integrity of the messages is compromised with two objectives. First, to disrupt the communication between

[3]We can also consider distributed denial-of-service (DDoS), where multiple nodes attack one or many other components.

the controller and the system, hence leading to supervision loss (which is hard to detect). Second, to inject malicious messages to move the system from its stability point. This way, the adversary generates unavailability of the system to authorized users, e.g., to make it available just for the malicious actions. In the next section, we present existing resilience techniques to face cyber-physical adversaries and reduce the impact they may have on the system safety.

## 4 SYSTEMATIC SURVEY ON CYBER-RESILIENCE LITERATURE

Cyber-resilience is the ability of a system to *prepare, absorb, recover*, and *adapt* to adverse effects [46]. The *preparation* phase is characterized by identifying the critical functions or services and stakeholders. It is important to understand the critical functionalities to guide the planning actions. The *absorption* phase involves the capacity of the system to contain the attack under degraded performance. It is the ability of a system to tolerate the stress. Thresholds are important to determine whether a system can absorb a shock or not. During the *recovery* phase, the system starts the process to restore its normal behavior as quickly and efficiently as possible. Finally, the *adaptation* phase involves a postmortem evaluation to improve the response and learn from past experiences.

Although the previously mentioned definition provides a clear view of the resilience stages, it may also be too broad for the CPS domain. A given CPS with unlimited resources (e.g., unlimited time) will eventually recover from all failures and attacks. Hence, resilience should be established considering a minimum group of conditions, e.g., in terms of temporal and computational resources. Under this assumption, and with the CPS context in mind, a more appropriate definition of resilience points out the necessity of providing [47]: (1) full correctness maintenance of the core set of crucial functionalities despite ongoing adversarial misbehavior (i.e., it is acceptable for non-crucial functionalities to be affected temporarily, such as partially degraded or complete failure); and (2) guaranteed recovery of the normal operation of the affected functionalities within a predefined cost limit. In addition, attack tolerance and graceful degradation are two properties that we may want to satisfy in a resilient system. Attack tolerance assumes that attacks can happen and be successful. The overall system must remain operational and provide a correct service. Graceful degradation is the ability of a system to continue functioning even in a lower performance after parts of the system have been damaged, compromised, or destroyed. The efficiency of the system working in graceful degradation usually is lower than the normal performance. It may decrease as the number of failing components grows. The purpose is to prevent a catastrophic failure of the system.

### 4.1 Risk Management vs. Cyber-Resilience

Risk management and resilience are different but related concepts [171]. Although they are both grounded in a similar mindset (e.g., reviewing systems for weaknesses and identifying policies or actions that could mitigate or resolve such weaknesses), substantial differences exist [172].

On the one hand, a risk is assessed by the likelihood of an undesirable event and the consequence of that event using probability distribution functions. On the other hand, resilience is about the remediation of unexpected rare extreme failures, whose likelihood cannot be estimated from historical data. Risk management is concerned with analyzing threat-by-threat to derive a precise quantitative understanding of how a given threat generates harmful consequences. Such exercise works well when the threats are categorized and understood, yet develops limitations when working with complex interconnected systems. Building from this limitation, resilience complements traditional risk-management approaches by reviewing how systems perform and function in a variety of scenarios, agnostic of any specific threat.

In addition, resilience requires thinking in terms of how to manage systemic, cascading effects on other directly and indirectly connected nodes. While risk management centers around the probability of hitting the weak points of

a system, resilience is grounded in ensuring system survival. It finds strategies to keep the functionality of the core system in the face of extreme events. Hence, resilience is based on a general acceptance that it is virtually impossible to prevent or remediate all categories of risk simultaneously, and before they occur [173].

New adversary models, as those presented in Section 3, create new challenges to achieve resilient systems. Indeed, achieving security in a CPS requires solutions that extend beyond what is offered by state-of-the-art cybersecurity products. As a result, a new research area must focus on strategies to face cyber-physical adversaries. In the control-theoretic community, this new area is known as *Resilient Control* [35]. It is worth noting that although these approaches are called *resilient* by control theorists, from a cybersecurity standpoint, resilient control is still dependent on a *detection*

Table 1. Proposed resilience approaches for CPS. (Top) Resilient Control Techniques. (Down) Cyber-Resilience Techniques.

| **Resilient Control Techniques** Section 4.2 | **Layer** | | | | **Proposals** |
|---|---|---|---|---|---|
| | **Physical** | **Network** | **Control** | **Cyber** | |
| **Detection** | | | | | |
| Data-based Approach | | ✓ | | ✓ | [48], [49], [50], [51], [52], [53], [54], [55], [56] |
| Model-based Approach | | | ✓ | | [33], [65], [34], [66], [67], [68], [69],[70], [57], [58], [59], [60], [61], [62], [63], [64], [35] |
| **Reaction** | | | | | |
| Resilient State Estimation | | | ✓ | | [35], [74], [75], [76], [77], [78], [79], [80], [81], [82], [71], [72], [73] |
| Reconfiguration | ✓ | ✓ | ✓ | ✓ | [85], [86], [42], [87], [88], [89], [90], [39], [91], [92] [83], [84] |
| Programmable Networking | | ✓ | | | [93], [94], [95], [96], [97], [98], [99], [100], [101] |

| **Cyber-Resilience Techniques** Section 4.3 | **Phase** | | | **Layer** | | | | **Proposals** |
|---|---|---|---|---|---|---|---|---|
| | **Absorb** | **Survive** | **Recover** | **Physical** | **Network** | **Control** | **Cyber** | |
| **Architecture Design** | | | | | | | | |
| Diversity | | ✓ | | ✓ | ✓ | | ✓ | [109], [110], [111], [112], [113], [114], [115], [102], [103], [104], [105], [106], [107], [108] |
| Segmentation | ✓ | | | | ✓ | | | [116], [117] |
| **Reconfiguration** | | | | | | | | |
| Isolation and Containment | ✓ | | | ✓ | ✓ | | ✓ | [118], [119], [120], [121], [122], [123] |
| Dynamic Network Composition | ✓ | ✓ | | | | | | [101], [124], [125], [126], [127] |
| Non-Persistence | ✓ | ✓ | | | | | | [128], [129] |
| **Moving Target Defense (MTD)** | | | | | | | | |
| Network MTD | ✓ | ✓ | ✓ | | ✓ | | | [137], [138], [139], [140], [141], [142], [143], [144], [130], [131], [132], [133], [134], [135], [136] |
| Node MTD | ✓ | ✓ | ✓ | | | ✓ | ✓ | [137], [149],[150], [141], [132], [133], [146], [150], [145], [146] [147], [148] |
| **Dynamic Software Evolution** | ✓ | ✓ | | | | | ✓ | [151], [87], [152], [153] |
| **Consensus & Distributed Trust** | ✓ | ✓ | | | ✓ | ✓ | | [160], [161], [162], [163], [164], [154], [155], [156], [157], [158], [159] |
| **Game Theory** | ✓ | ✓ | | | | | ✓ | [165], [166], [167], [168], [169], [170] |

and *reaction* paradigm. In other words, although resilient control incorporates in the traditional fault-tolerant control new strategies to face cybersecurity breaches, it still aims at determining how a controller can detect, correctly estimate the system state, and recalculate the required commands despite malicious data. It also aims at responding to the attacks with appropriate countermeasures, to achieve stability and graceful degradation while the system is under attack. This objective can be achieved through a system theoretical analysis of the CPS.

Next, we present our survey of solutions in both categories. First, we survey in Section 4.2 *resilient control* techniques under the traditional *detection* and *reaction* paradigm. Then, we survey in Section 4.3 *cyber-resilience* techniques that provide system recovery without triggering any additional behavior. Our literature survey is summarized in Table 1.

### 4.2 Resilient Control

Detection and mitigation for cyber-physical attacks are not trivial. It requires incorporating control-theoretic strategies into traditional cybersecurity approaches to contemplate the new vulnerabilities. In this section, we present resilient control strategies based on detection and reaction mechanisms for CPS.

*4.2.1* ***Detection Approaches.*** There are two main strategies for attack detection in CPS: *data-based* and *model-based* approaches [7]. Data-based and model-based approaches are complementary solutions, together they consider the interaction between both cyber and physical layers.

*4.2.1.1 Data-based Approach.* This approach does not require system and attack models for detection. It is based on traditional machine learning and pattern recognition techniques [48–50] for analyzing hidden patterns in the observed training dataset, for example, control signals and sensor measurements. Mitchell *et al.* [51], Cheminod *et al.* [52], and Han *et al.* [53] provide surveys of intrusion detection techniques focusing only on data-based approaches using traditional intrusion detection systems. Ahmed *et al.* [54] provide a survey of trust-based detection and isolation approaches for malicious nodes in sensor networks. In addition, Ding *et al.* [55] survey the development of attack detection for industrial CPS and discuss control and state estimation in the case of an attack. Also, Beaver *et al.* [56] provide an evaluation of machine learning methods to detect malicious communications in supervisory control and data acquisition protocols.

*Advantages and limitations:* This detection technique considers cyber and network patterns to identify attacks. For this reason, it can detect cyber-attacks. However, it is not able to detect all kinds of cyber-physical attacks, since it does not consider the control model. It has a partial view that does not include the physical components.

*4.2.1.2 Model-based Approach.* This approach uses the model of the systems to detect attacks. The decision is based on the comparison between system observations and model outputs. The system is under attack if the observed data is no longer consistent with the estimated outputs of the normal mode. This comparison may not be obvious because of the presence of model uncertainties, nuisance parameters, and random noise.

There are five main strategies for control-theoretic model-based attack detection [174]: *watermark-based detectors*, *signal-based detectors*, *state relation-based detectors*, *cross layer-based resilient detectors*, and *auxiliary systems detectors*. Next, we summarize the main ideas underlying each strategy.

- In the case of *watermark-based detectors*, a low amplitude noise, called watermark, is added to the control measurements to verify, by using a detection mechanism, that the sensor measurements and commands are not modified, i.e., the control measurements with the watermark have to be correlated with the sensor measurements. For example, Mo *et al.* [33] propose the use of Kalman filters to detect cyber-physical replay attacks by adapting

traditional failure detection mechanisms via watermarking. Miao *et al.* [65] improve the performance of the aforementioned detection mechanism using a stochastic game approach. The work has also been improved by Rubio-Hernan *et al.* [34] to incorporate more advanced adversaries capable of learning the physical model. In the same way, Do *et al.* [66] propose a detection approach based on the knowledge of the system's behavior and its stochastic variations to detect data manipulation.

- *Signal-based detectors* use the signal statistical properties and the system behavior to detect attacks. For example, Arvani *et al.* [67] describe a model to detect and identify random signal data-injections attacks. It is based on discrete wavelet transform analysis to exploit the statistical properties of the signal and the dynamic model of the system. It also uses a chi-square detector to identify anomalies. Lokhov *et al.* [68] present a protocol for detection and localization of disturbance based on a special correlation matrix. The matrix allows: (1) detecting anomalies using spectral methods; (2) localizing a subset of anomalous nodes within the system; and (3) identifying the functional role of the inferred anomaly based on the sensor labels.
- *State relation-based detectors* use the correlation of system states and the system behavior, to identify anomalies. For example, Wang *et al.* [69] propose a relation-graph-based detector scheme to detect false data injection attacks, even when the injected data may seemly fall within a valid and normal range. A correlation model extracts the relation among the different variables of the system to create a graph model with the possible valid system states. The correlation model uses a forward correlation that is not affected by time and a feedback correlation that depends on time. Chen *et al.* [70] present a distributed anomaly detection algorithm using graph theory and spatiotemporal correlations to analyze the physical process in real-time. Amin *et al.* [57] develop a model-based scheme for detection and isolation. The scheme is based on a group of unknown input observers designed for a linear delay-differential system obtained as an analytically approximate model. The generated conditions are delay-dependent, and can also incorporate communication network-induced time-delays in the sensor-control data. To detect and isolate the attacks, they use a residual generation procedure. Also, Dehghani *et al.* [58] present a static state estimation algorithm able to detect integrity attacks against smart grids.
- *Cross-layer based resilient detectors* combine control and cyber techniques in a single intrusion detection system. For example, Zhu *et al.* [59] propose a game-theoretic framework that integrates the discrete-time Markov model for modeling the evolution of cyber states with continuous-time dynamics describing the controlled physical process. The cross-layer design is created between physical and cyber detection layers to maximize the chances of identifying security events. Bobba *et al.* [60] show that protecting only a set of basic measurements is enough to detect attacks against physical and network malicious actions. In addition, Pasqualetti *et al.* [61] use geometric control theory to optimize cross-layer resilient control systems. They conclude that by using a geometric model of the system, it is possible to detect and estimate the system state in the presence of unknown inputs.
- *Auxiliary system detectors* use state observer techniques (e.g., Luenberger observers [62]) to build a digital copy of the system and be able to control its behavior. For example, Shoukry and Tabuada [63] describe an algorithm for state reconstruction from sensor measurements that are corrupted using a Luenberger observer. Also, Schellenberger *et al.* [64] extend an original plant with an auxiliary system that does not add additional delay into the system. The auxiliary system is designed as a linear discrete-time digital copy with similar dynamics to the original system, but capable of conducting attack detection. For this detection strategy, a model of the overall system dynamics and the switching signal of the auxiliary system are needed. The residuals of the Luenberger observer are then monitored for deviations from zero, which indicates the existence of attacks.

*Advantages and limitations:* This detection technique considers the physical model to identify attacks. It is suitable to identify cyber-physical attacks using feedback control. However, the information on traffic patterns and cyber-attacks identification may be limited. For this reason, it is complementary to the previous approach which is based on cyber and network data. Both techniques working together, have a more complete and integral view of the system.

*4.2.2* ***Reaction Approaches.*** As pointed out in [19], large research efforts have focused on intrusion detection. There is little less discussion about what to do after the intrusion is detected, i.e., in remediation approaches that mitigate the effects of an attack. Most of the responses in CPS are manual or hardwired with a fixed response that cannot be configured. In the sequel, we survey some representative proposals under the reaction (after detection) paradigm.

*4.2.2.1 Resilient State Estimation.* When an adversary modifies data, system recovery requires knowing the real state of the system. For this reason, resilient state estimation is a technique that can help in terms of system reaction. It allows a remote defender to maintain an understanding of the system state under attack, even when a subset of inputs and outputs are under the control of an adversary [35]. As a result, the defender can still have reliable state information to apply an appropriate feedback control law, to better understand the portions of the system that have been compromised, and to design attack-specific countermeasures.

Approaches for resilient state estimation can be found in the following literature. Fawzi *et al.* [74] propose an efficient state reconstructor inspired by techniques used in compressed sensing and error correction over real numbers. They also characterize the maximum number of attacks that can be detected and corrected as a function of the system state matrices. Pajic *et al.* [75] present a method for state estimation in the presence of attacks, for systems with noise and modeling errors such as jitter, latency, and synchronization problems that are mapped into parameters of the state estimation procedure. Pajic *et al.* [76] also propose a state estimation approach in the presence of bounded-size noise for sensor attacks where any signal can be injected via compromised sensors.

In addition, Mo and Sinopoli [77] propose a state estimator based on $m$ measurements that can be potentially manipulated by an adversary. The adversary is assumed to have full knowledge about the true value of the state to be estimated and about the value of all the measurements. If the adversary can manipulate up to $l$ of the $m$ measurements, then the estimator works properly when the adversary compromises less than half of the measurements, i.e., $(l < m/2)$. The solution is formulated as an optimization problem where one seeks to construct an optimal estimator that minimizes the worst-case expected cost against all possible manipulations by the adversary. Keller *et al.* [78] propose a state estimation of stochastic discrete-time linear systems in the case of malicious disturbance that switches between unknown input and constant bias. This means that when corrupted control signals are received by the controller, detectors based on Kalman Filters are used to estimate the state of the system and the exogenous unknown input of the system (i.e., the malicious inputs). In addition, the malicious control signal is blocked at the occurrence of data losses, and the unknown input is transformed to a constant bias at the input of the system. Weimer *et al.* [79] introduce a resilient estimator for stochastic systems using a mean squared error for the state that remains finitely bounded and is independent of attacks in measurements.

Shoukry *et al.* [80] and Mishra *et al.* [81] propose secure state estimation algorithms for linear dynamical systems under sensor attacks and in the presence of noise. The approaches are based on satisfiability modulo theory, which is a technique used to express problems that should satisfy constraints, i.e., decision problems using logical formulas expressed in first-order logic [71, 82]. Another technique used to improve the state estimation accuracy is to consider multiple sensor systems instead of one single sensor system [72, 73]. In this case, data fusion is a process in which the received data is integrated from different sensors observing the same system.

*Advantages and limitations:* This approach is useful when sensors, actuators, or network traffic have been compromised. It provides a reliable state of the system even when an adversary injects malicious traffic into the system. As a result, this technique helps the system recovery because it allows maintaining an understanding of the state under attack, even when a subset of inputs and outputs are malicious. The limitation is that it can only repair a maximum number of compromised values. In addition, it is hard to ensure that the control commands are executed correctly by simply using state estimation techniques. For that, complementary actions need to be included in the response plan and to ensure that the estimated data properly reaches its destination.

*4.2.2.2 Reconfiguration.* Once the system is compromised, it is required to ensure that the control commands arrive correctly to the actuators. One possibility to do this is to alter dynamically the configuration of the system to minimize the effects of the attack. For example, changing the network topology, configuration of the devices, firewall rules, or quarantining (rerouting) traffic. In other words, the system structure is modified to face the attacks. For instance, one option would be to increase the number of sensors such that attacks are identified faster or add extra layers of security to those elements that are more vulnerable to cyber-attacks [85]. Components may also be isolated. Li *et al.* [86] propose a decision-making approach for intrusion response aiming to determine the optimal security strategy against the attacks. The strategy tries to secure attack paths with higher priority, in addition to responding to functional failures. Authors assess both cyber and physical domains with an in-depth analysis of attack propagation. Yuan *et al.* [42] propose a resilient controller design for CPS under DoS attacks. The proposal uses a framework that incorporates an IDS and robust control. The robust control in the physical layer is based on an algorithm with value iteration methods and linear matrix inequalities, e.g., for computing the optimal security policy and control laws. The cyber state is modeled as a continuous Markov process to defend against malicious behavior.

Other techniques incorporate dynamically new on-demand capabilities to face the attacks. For example, using pre-configured virtual machines to help affected components, adding new cloud-based services to help with denial-of-service attacks, or distributing tasks in a different organization. Ismail *et al.* [88] propose an optimization of the defense countermeasures deployment. To design the approach, the available information is presented in an attack graph, representing the evolution of the state of the attacker in the system. Then, they find the optimal security policy to maximize the system protection using Markov decision processes. This way, countermeasures are prioritized to respond efficiently to the intrusion. Also, game-theoretic approaches can be used to improve the system response. Kiennert *et al.* [89] survey strategies capable of analyzing the interactions between attackers and defenders, then responding to attacks, via game theory and Markov decision processes.

Based on how frequently the attacks occur, *event-triggered control* schemes instead of time-triggered schemes emerged as appropriate tools to increase the resilience of control systems [39, 90]. The application of event-triggered control to the resilience of CPS has been studied in [83, 84, 91] where the triggering function to generate a new control input is based on the errors of the state variables. Sun *et al.* [92] propose an adaptive event-triggered resilient control to resist asynchronous data attack injection in industrial CPS network communication. Their proposal uses a threshold that dynamically changes and adjusts the control strategy, according to the attack.

*Advantages and limitations:* This approach provides a flexible and dynamic response mechanism that can work only when the system is under attack to provide graceful degradation. It may be designed to protect sensors, actuators, controllers of the network traffic. However, this approach may be hard to test and to ensure the stability of the control feedback when combining malicious and defensive actions over the physical process. It may have hidden undesirable actions or cascade effects that may be harmful to the system. In addition, it increases the complexity, making it complicated to test all possible combinations of malicious actions and dynamic defensive configurations.

*4.2.2.3 Programmable Networks.* Some other proposals are based on programmable networking that enables efficient network configuration that can be used for neutralizing attacks. New networking functionality can be programmed using a minimal set of APIs (Application Programming Interfaces) to compose high-level services. This idea was proposed as a way to facilitate network evolution. Some solutions such as Open Signaling [93], Active Networking [94], and Netconf [95], among others, are early programmable networking efforts and precursors to current technologies such as Software Defined Networking (SDN) [96]. In particular, SDN is a programmable networking paradigm in which the forwarding hardware is decoupled from control decisions. SDN proposes three different functionality planes: (1) data plane, (2) control plane, and (3) management plane. The data plane corresponds to the networking devices, which are responsible for forwarding the data. The control plane represents the protocols used to manage the data plane, such as, to populate the forwarding tables of the network devices. The management plane includes the high-level services and tools, used to remotely monitor and configure the control functionality. Security aspects may have an impact on different plans. For example, a network policy is defined in the management plane, then the control plane enforces the policy and the data plane executes it by forwarding data accordingly.

The idea of using programmable networks for improving security includes the management of denial-of-service (DoS) attacks [97] and segmentation of malicious traffic [98, 99]. Programmable networks provide higher global visibility of the system, which is favorable for attack detection. In addition, a centralized control plane may allow further possibilities to achieve dynamic reconfiguration of network properties, e.g., application of countermeasures. Molina *et al.* [100] survey approaches for SDN controllers that are able to establish different paths between sensors and actuators. Piedrahita *et al.* [101] use SDN and network function virtualization to facilitate automatic incident response to a variety of attacks against industrial networks. The resources are assigned after an attack is detected. SDN and cloud-enabled virtual infrastructure help to respond automatically to sensor attacks and controller attacks by rerouting malicious traffic to a honeypot and transfer the services from the compromised device to a new virtualized device.

*Advantages and limitations:* The programmable networks also provide a dynamic reconfiguration to respond at runtime to malicious actions in the network traffic. These approaches are flexible, however, it may be hard to analyze how the new network configuration affects the network delay and jitter, which is vital in real-time applications. Also, the reconfiguration increases the network complexity and the restoration work may induce hidden undesirable behaviors within the system.

In this section, we have presented detection and reaction mechanisms for cyber-physical adversaries. However, despite the implemented mechanisms, it is still possible to have a system breach. For this reason, it is desired to implement cyber-resilience by design approaches to absorb, survive or recover from threats. Another cyber-resilience taxonomy can be found in [46]. Cyber-resilience demands a system design that provides flexibility, adaptability, and agility to react in real-time to disturbances. In the next section, we survey techniques to build cyber-resilient systems.

### 4.3 Cyber-Resilience Approaches

A growing number of technologies and architectural practices can be used to improve cyber-resilience. In the rest of this section, we cover techniques that may be used to build resilient systems. We provide a taxonomy of cyber-resilience techniques and a literature survey of different proposals that apply them. We analyze the techniques according to the cyber-resilience phase they react and the CPS layer they protect. A resilience solution may work in the absorb, survival, or recovery phase. The absorb phase limits the damage of the attack or extends the surface that the adversary has to attack to be successful. For example, by isolating resources, limiting adversary access, and changing or removing

resources. The survival phase objective is to maintain or maximize the duration of the correct function of the essential system mission. The recovery phase aims at transforming or reconstituting the resources to recover the functionalities after the attack. We also analyze at which level of the system design the resilience approach works. For example, it may be at the physical level considering the hardware of the components, at the control level to face adversaries that exploit the control theory mechanism that is running in the controllers, at the network or cyber level considering the communications or the software of the system. Table 1 sums up the different cyber-resilience strategies and scientific proposals that use them.

*4.3.1* ***Architecture Design.*** These strategies involve modifying the system architecture to improve the resilience of the system to absorb or survive the attack impact [175].

*4.3.1.1 Diversity.* It uses a heterogeneous set of technologies to minimize the impact of the attack. Different technologies will have different and independent vulnerabilities, which will make the adversary task harder to achieve. In addition, this technique increases the adversary uncertainty and the resources required for a successful attack.

This technique can be applied, for example, using different hardware, software, firmware, or protocols [176]. It is worth noting, that this technique requires adding new components. These components should be different from the previous ones because just adding redundancy makes the system still exploitable by the same adversaries using the same vulnerabilities as in the primary components.

When designing software diversification techniques, it is required to decide what to diversify and when to diversify it [109]. To decide what to diversify, possible techniques are: (1) randomization which works as a compiler optimization and can be applied, for example, at the instruction level by substituting equivalent instruction or sequence of instructions; (2) randomizing the register allocation or reordering instruction. Another option is to apply this technique also at block, loops, functions, data, or even program levels. For example, at the functions level, it is possible to randomize the order of function parameters or the layout in the stack to prevent buffer overflow attacks. At the program level, similar strategies can be applied to randomize the order of the functions within executables and libraries. Different options to decide when to apply the diversification are at implementation time (i.e., when coding) [111], at compiling and linking the source code [102–105, 113–115], or at installation, loading, or execution time [106–108, 177].

Other diversity solutions may work also in a detection-reaction manner. For example, Ouffoué *et al.* [110, 178] use diversification to create attack-tolerant web services. They modeled the services to extract different implementations using variations in style, encoding, and language. The multiple services' implementations allow monitoring for attacks and react by changing the active implementation.

In the case of hardware diversification, it is required to design if all the different components will be active at the same time or if they will act as a cold backup that is activated after the primary system is attacked. For example, authors in [112] use diversity to improve cyber-resilience for industrial control systems. The strategy is implemented using primary and redundant PLCs from different vendors to enhance cyber-resilience.

*Advantages and limitations:* When diversity is used with different implementations, it helps to create a system with independent vulnerabilities. This way, when a component is attacked, it can be disabled to continue working with the diversified copy. This may be applied to sensors, actuators, and controllers attacks. The advantage is that the system keeps all the functionalities working and it is possible to ensure the correct behavior of the control. However, this approach may be expensive due to the required extra hardware and it only addresses attacks at the endpoints. Also, it requires extra management and maintenance effort, for example, to apply the software updates to a wider and more diverse group of components.

*4.3.1.2 Segmentation.* The design of a CPS must consider how to prevent attacks and be more tolerant to intrusions from the beginning. A network segmentation strategy separates logically or physically the components to reduce the attack surface. It also contains and limits the damages of a successful attack. The components may be separated based on their level of criticality, trustworthiness, or functionality [116, 117].

According to the results achieved in [116], this technique also contributes to building more intrusion-tolerant CPS. Network segmentation may be designed considering the Process-Aware Control approach presented in [117]. It establishes that attacks on some components generate a greater risk than attacks on other components in the same system. For this reason, it is important to classify the different network components and the control loops according to the impact they may have on the operation of the CPS. This approach would allow protecting the essential components in a better way. Following this idea, it also allows having the notion of *more insecure* nodes (for example, a node that uses wireless communication technologies), e.g., to place them in a network segment separated from the other nodes that are considered as a trust zone.

A segmented architecture can help to absorb the impact of a compromise and prevent cascading failures [179]. A network susceptible to large cascade failures is likely to have severe damage to disturbances, which limits the absorption and recovery required to build a resilient system. For this reason, the dependencies and links between nodes should be designed to minimize the likelihood that a failure propagates easily from one node to another.

*Advantages and limitations:* This approach is easy to implement and effective to contain the consequences of a compromised component. It also limits cascade effects and the propagation of the attack within the network. The main limitation is that it does not help to recover the system to its normal behavior.

*4.3.2* ***Reconfiguration***. There are different possible reconfiguration options. This technique requires a situational awareness to select pre-considered options, ensuring the intended consequences. For example, in a denial-of-service (DoS) attack, we might dynamically over-provision additional processing capabilities. If an attack comes from the outside, we may reconfigure boundary protections and security policies. During a failure, we may shut down non-essential functions or initialize alternative capabilities to execute critical processing. We classify possible reconfiguration in the following categories.

*4.3.2.1 Isolation and Containment.* These strategies aim at limiting the spread of the adversary by separating compromised from non-compromised components. For example, if an adversary controls a part of the system, it may be necessary to temporarily shut down it to close the adversary's channel while critical mission functions are completed in another portion of the system.

Kwasinski in [122] analyzes this problem for power grid and he shows how service buffers, such as energy storage or a data connectivity reestablishment ensured time, help limit the impact of intra-dependencies on resilience. They explain that without service buffers, failures in an infrastructure component may immediately cascade within the system or onto other infrastructures. For this reason, resource buffers play a critical role in understanding cyber-physical interactions, limiting the negative effect of intra-dependencies, and improving resilience.

Xu *et al.* [123] show that isolation and reconfiguration are effective approaches for service restoration and resilience enhancement. They propose a multi-stage switch strategy based on dynamic programming, considering both isolating and fault reconfiguration. First, they propose the construction of numerous expected fault scenarios. Then, they select some of them and develop their information entropy. Finally, for each typical scenario, a multi-stage switch strategy considers both isolation and fault reconfiguration, through dynamic programming.

Bellini *et al.* [118] analyze Internet of Things (IoT) resilience considering a network-based epidemic spreading approach. The mathematical model assesses infection and communication interactions to reduce a malware outbreak while maintaining the network functionalities at an acceptable level. Disconnecting a network region compromises connectivity. The mobility of resources to an affected area is of critical value for the immediate local control of outbreaks and to prevent the spread.

Chen *et al.* [119] analyze how attacks in communication networks may cause cascading failure in a physical power grid. They find that clusters in physical power grid and communication network are mutually interdependent to survive in cascading failure, operating in the form of isolated subsystems the failures remain interdependent to stay alive when cascading attacks occur. Hence, they consider survival clusters to adjust intra- and inter-links and study the robustness of the system in various attack scenes.

Haque *et al.* [121] analyze resilience for energy delivery systems considering cyber components and service criticality. They estimate the criticality using graph Laplacian matrix and network performance after removing links (i.e., disabling control functions or services) and also analyze the cyber resilience by determining the critical devices using TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) and AHP (Analytical Hierarchy Process) methods. They consider paths as a sequence of services or control functions and assume the removal of links as disabling the service or deactivating the control function rendered by the particular device.

*Advantages and limitations:* As in the previous approach, this technique is effective in containing the consequences of an attack but it does not help to recover the system to its normal behavior. Isolating or disconnecting a component or part of the system in case of compromise prevents the spread and cascade failures. However, this might be also detrimental to the overall resilience of the system if the isolated component is needed to support other components that execute damage-absorbing actions. The recovery actions should be planned with this in mind.

*4.3.2.2 Dynamic Network Composition.* This technique designs the system with dynamic capabilities to face the attacks. For example, distributing tasks in different organizations. Januario *et al.* [125] propose a hierarchical multi-agent framework that is implemented over a distributed middleware with distributed physical devices. The architecture uses software-defined networks and cloud-based virtual infrastructures. Physical and cyber vulnerabilities are taken into account, and state and context awareness of the whole system are targeted. Each multi-agent executes a specific task and adapts its behavior depending on its location and environmental changes. In addition, Chen *et al.* [127] propose an approach to improve resilience using the synchronization of multi-agent systems that address faults and uncertainties in communication links. For that, they transform the resilient control problem into distributed state observers.

Marshall *et al.* [126] present a context-driven decision engine for adaptive resilient control. It integrates diagnostic and prognostic heuristics to establish situational awareness and drive actions. The proposal assesses the system's state of health based on operational availability and drives control decisions based on scenario-specific constraints and priorities. Similarly, Ratasich *et al.* [180] presented a self-healing framework that uses structural adaptation, by adding and removing components, or by changing their interaction, at runtime. Segovia *et al.* [124] proposed an attenuation strategy that uses software-defined networks and software reflection. In case of attack, the approach creates dynamically in the network domain a component on the fly to help or assume the functions of the victim node.

*Advantages and limitations:* This approach changes the configuration of the system periodically and increases the attack effort. In addition, the new configuration may force the adversary to re-implement the attack with each system change. This technique may be effective for attacks that compromise controllers of the network traffic. The limitation of this approach is that it may be hard to ensure the stability of the control feedback when combining malicious and

defensive actions over the physical process. Also, it increases the complexity of the system and it is harder to test, manage and debug.

*4.3.2.3 Non-Persistence.* This technique reduces the adversaries' opportunity to identify and exploit vulnerabilities or maintain access over resources whose access is not continuous in time. It can be applied, for example, to data, applications, or connectivity, making them only accessible during a particular time. In addition, with this technique, a system can periodically refresh to a known previous image to ensure that the current image complies with a secure configuration.

Another option is to implement reversibility. This way, components are designed in a manner that allows them to revert to a safe mode when failed or compromised. This means that the component in the failed mode should not cause any further harm to other components in the system; and second, it should be possible to reverse the state of the component in the process of recovering the system. The system can periodically refresh to a previously known image to ensure that the current system image is correct.

For example, Griffioen *et al.* [128] present a decentralized control system and a procedure to determine when agents should communicate with one another after having been disconnected from the network for a period of time. When agents communicate with one another, they guarantee system resilience against malicious adversaries by utilizing software rejuvenation, a prevention mechanism against unanticipated and undetectable attacks on cyber-physical systems. Without implementing any detection algorithm, the system is periodically refreshed with a secure and trusted copy of the control software to eliminate any malicious modifications to the run-time code and data that may have corrupted the controller.

Pradhan *et al.* [129] present a runtime infrastructure that provides autonomous resilience via self-reconfiguration. The approach relies on the implicit encoding of all possible states a system can reach (the configuration space) and it consists of relevant information about different system goals, functionalities, services, resources, and constraints. At any given time, there is exactly one configuration point that represents the current state of a platform. At runtime, when a configuration point is deemed faulty, the self-reconfiguration infrastructure computes a valid new configuration point that belongs to the same configuration space, and then transition, migrate, or reconfigure to the newly computed configuration point such that failures or anomalies are mitigated.

*Advantages and limitations:* This approach returns the system to a previously safe and known state, which ensures the correct behavior. It is effective for attacks that compromise specific devices such as sensors, actuators, controllers, routers, or switches. However, this solution does not last long, due to the vulnerabilities exploited by the adversary are still present in the previous image, and they can be exploited again.

*4.3.3* ***Moving Target Defenses.*** A static structure allows adversaries to collect information and perform long-term analysis. In addition, the uniformity of components allows adversaries to expand the damage scope after they find one vulnerability. For this reason, Moving Target Defense (MTD) approaches provide strategies that change the system over time to increase its complexity, attack cost, or limit the exposure of vulnerabilities [133]. The mechanisms are usually applied at the network or the node level [132]. Next, we summarize proposals for both levels as well as approaches specially designed for CPS.

*4.3.3.1 Network MTD Approaches.* The *endpoint information* (such as MAC address, IP address, port, protocol, or encryption algorithm) and the *forwarding path* (links and routing nodes) are two key elements in network transmission

and it can be used to identify the source and destination nodes. Hence, it is important to protect this information as part of the attack surface.

Some approaches that protect the endpoint information are as follows. Antonatos *et al.* [139] propose the use of Network Address Space Randomization (NASR) to handle worm attacks. The method analyzes and discriminates the potentially infected endpoints and the nodes are forced to frequently change their IP address by using DHCP protocol. Al-Shaer *et al.* [138] proposed Random Host Mutation that assigns virtual IP addresses that change randomly and synchronously in a distributed way over time. To prevent disruption of active connections, the IP address mutation is managed by network appliances and transparent to the end host.

MacFarland *et al.* [142] hide the endpoint MAC, IP, and port numbers by setting up a DNS hopping controller and synthetic addressing information in place of the real one with the help of NAT rules. This can be considered to be chosen at random within certain validity constraints.

Other approaches protect the forwarding path information, i.e., it randomly selects routing nodes to change the forwarding paths while ensuring reachability. For example, Dolev *et al.* [143] use a secret sharing technique to encrypt data and create $n$ shares, and only fewer than $k$ parts can be allowed to transmit in the same path. In addition, to reconstruct the data, the destination needs to have at least $k$ shares out of the $n$ shares that were sent. The approach objective is to provide private and secure interconnection between the data centers. Aseeri *et al.* [144] propose an approach to improve the diversity of forwarding paths to deal with eavesdropping attacks in the SDN data plane. It uses bidirectional multiple routing paths to reduce the severity of data leakage. The SDN controller applies the multipath mechanism both ways, from the sender side and the receiver side. By negotiating migrating paths between source and destination, the forwarding path is changed randomly during transmission.

Duan *et al.* [130] propose a Random Route Mutation technique that enables changing randomly the route of the multiple flows in a network simultaneously to defend against reconnaissance, eavesdropping, and DoS attacks while preserving end-to-end QoS properties. Ma *et al.* [131] propose an approach for self-adaptive end-point hopping, which is based on adversary strategy awareness and implemented using SDN. This method periodically changes the network configuration in use by communicating endpoints. Potteiger *et al.* [134] propose to implement MTD techniques such as Address Space Randomization (ASR), and Data Space Randomization (DSR) in a mixed time and event-triggered architecture to maintain the safety and availability during the attack. Mixing both architectures allows the system to support predictable operation during normal circumstances while maintaining rapid detection and reconfiguration during an attack. Xu *et al.* [135] propose an MTD technique with a routing randomization method based on deep reinforcement learning. This proposal improves the security against eavesdropping attacks, improving the random routing granularity, real-time and accurate network state awareness, and powerful decision-making. Azab *et al.* [136] propose a novel MTD approach using multi-controller management of SDN. The objective of this multi-controller approach is to detour the runtime workload among multiple controllers and control misbehavior detection without impact on CPS performance.

*Advantages and limitations:* This approach is similar to *Programmable Networks*, the difference is that the re-configurations are periodicals and not triggered by any detection. Due to the pre-configurated system change, it may be possible to predict better the response of the system in each change and also in case of an attack. The approach is effective for attacks that compromise network traffic. One limitation is that the re-configurations increase the network complexity impacting negatively the debugging and managing effort. It may also impact the network performance in each reconfiguration period. For instance, creating loops until all the paths are updated or affecting the latency between nodes.

*4.3.3.2 Node MTD Approaches.* Platform environment and software applications can be diversified to protect from adversaries. Diversity proposes to have many forms of the same object because this design can reduce the probability of intrusion [181]. Address space, instructions, or data randomization are three typical ways to achieve platform environment diversification [182]. Another technique is software application isomerization. In software engineering, isomerization is a mechanism that changes codes dynamically to enhance the heterogeneity of software applications under the premise of ensuring functional equivalence. Depending on the application software life cycle, it can be divided into transformation mechanisms adopted during software compilation and link or transforming mechanisms implemented during software load and execution [132]. In addition, programmable reflection is a meta-programming technique that has the potential to allow a programmable system to manipulate itself at runtime [87].

The previous techniques are software techniques that can be applied to a wide variety of systems. Some CPS-specific MTD approaches have been proposed to control adversaries situated in the end devices, i.e., actuators and sensors. For example, in [150], Giraldo *et al.* propose an MTD strategy that randomly changes the availability of the sensor data, so that it is harder for adversaries to achieve stealthy attacks. This approach uses switched control systems that allow detecting sensor compromise and minimizing the impact of false-data injection attacks. In [147], Giraldo *et al.* present a novel approach for MTD using IoT-enable Data Replication (MTD-IDR). They utilize liner-matrix inequities for the optimization problem, in order to select and optimize the number of replicas of each communicated signal in the system. This approach prevents stealthy attacks and reduces the accuracy of attack to learn the system's model, nevertheless the energy consumption increases and the bandwidth is reduced. Griffioen *et al.* [149] propose an MTD approach for recognizing and isolating CPS integrity attacks on a set of sensors and actuators by introducing stochastic time-varying parameters in the control system. The underlying random dynamics of the system limit the adversary's knowledge of the model. Liu *et al.* [148] propose a strategy by proactively perturbing the primary control gains of the power converter device in DC microgrids (DCmGs) to defend against deception attacks. They highlight the importance of providing explicit conditions for the magnitude and the frequency of the perturbation in order to ensure the voltage stability of the system.

Weerakkody *et al.* [145] propose an MTD approach to minimize identification in CPS, i.e., to limit the adversary's knowledge of the system model to identify sensor attacks by changing the dynamics of the system as a function of time. Kanellopoulos *et al.* [137] propose an approach to mitigate sensor and actuator attacks by formulating a control algorithm based on MTD that provides a proactive and reactive defense mechanism. It uses a stochastic switching structure to alter the parameters of the system and make it more difficult for the adversary to perform a system reconnaissance. Segovia *et al.* [146] propose an MTD approach that changes the CPS physical model that executes in each node periodically. The system is modeled as a switched control system to improve resilience.

*Advantages and limitations:* Similarly to the previous approach, the re-configurations are periodicals and not triggered by any detection. This is an advantage because modeling the system as a Switched Control System makes it possible to better predict the stability of the system in each change to ensure its correct behavior. In this case, the control theory provides strong mathematical models to understand, limit the damage, and predict the system behavior under attack. This approach may be efficient for sensor, actuator, controller, or network attacks depending on how it is designed. One limitation is that the re-configurations increase the system complexity making the debugging and testing effort bigger.

*4.3.4* ***Dynamic Software Evolution.*** Dynamic software evolution uses code generation or modification at runtime to adapt the system behavior and face adversaries. We can differentiate two main approaches: *Runtime Code Generation* and *Software Reflection.*

The former, Runtime Code Generation, is a particular case of code generation techniques used to create source code at runtime. Some languages support this feature, for example, .NET which provides a mechanism that produces source code in multiple programming languages at runtime, based on a single model that represents the code to render in a language-independent object model. This way, programs can be dynamically created, compiled, and executed at runtime. Code generation involves creating code that never has to be modified once it is generated. If a problem arises, the problem should be fixed in the code generator, and not in the generated source files. This technique may be used to generate diversity in the created software.

The latter, Software Reflection or Self-Modifying Code, is another technique that allows a system to adapt itself through the ability to examine and modify its execution behavior at runtime. As a mitigation technique, software reflection has the potential to allow a system to react and defend itself against availability threats. When malicious activity is detected, the system shall dynamically change the implementation to activate remediation techniques to guarantee that the system will continue to work. Software reflection provides the ability to analyze, inspect, and modify the structure and behavior of an application at runtime. This allows the code to inspect other codes within the same system or even itself. Reflection allows inspecting classes, examining fields, changing accessibility flags, dynamic class loading, method invocation, and attribute usage at runtime even if that information is unavailable at compile time. Also, it is possible to use data marshaling and pull data from an outside source and load it into a Java object or use reflection to execute it.

He *et al.* [152] propose an approach to modify the software runtime architecture through meta-operators based on reflection. Similarly, Kon *et al.* [153] propose a reflective middleware to deal with highly dynamic environments, supporting the development of flexible and adaptive systems and applications. Mavrogiannopoulos *et al.* [151] present a taxonomy of self-modifying code with the purpose of obfuscation.

*Advantages and limitations:* This approach is flexible and dynamic and it works for attacks on sensors, actuators, and controllers. However, it is harder to have control over what is being executed in each node and its effects on the system stability. Also, due to the difficulty of understanding what is being executed, it may be harder to test, debug, and protect the system.

*4.3.5* ***Consensus, Secret Sharing, and Distributed Trust.*** Both consensus and distributed trust approaches have been largely investigated for general computer science problems where some of the subsystems are untrustworthy.

Consensus protocols provide resilience to the byzantine problem, i.e., in the presence of malicious nodes that send incorrect messages to deceive the system. These consensus approaches may be applied at the network level which has been largely studied by the distributed computing research community [183–185], or it may also be applied at the control level which is an active research area in the control theory community. In this case, at each update, the controller ignores suspicious values and computes the control input with the non-suspicious values. For example, using Distributed Kalman Filter for resilient state estimation [162, 163] or other distributed observers strategies to manage sensor compromise [161, 186]. Other strategies are distributed function calculation in the presence of malicious agents [160], distributed multi-agent consensus [154, 155, 164, 187, 188], resilient vector consensus [156, 158], and resilient leader-followers consensus approaches [157, 159].

Techniques such as secret sharing schemes [189–191] and distributed trust [192, 193] may be used to implement, for example, mechanisms that divide the control into shares, such that the system needs to reach a given threshold before granting control, i.e., a data $D$ is divided into $n$ pieces in such a way that $D$ is easily reconstructable from any k pieces, but even complete knowledge of $k-1$ pieces reveals no information about $D$. Secret-sharing schemes are important

tools in cryptography used in many security problems such as multiparty computation, Byzantine agreement, threshold cryptography, access control, attribute-based encryption, distributed certificate authorities, distributed information storage, key management in ad-hoc networks, electronic voting, and many others. A classical approach to building secret-sharing schemes is Shamir's threshold approach [189], which divides the data $D$ using a polynomial of grade $n$. The correctness and privacy of this scheme follow from the Lagrange's interpolation theorem. The undirected s-t-connectivity approach [191] builds the scheme using an undirected graph structure whose share parties between entities are mapped to edges, nodes, and paths to connect those nodes. Other existing schemes are based on monotone formulas, for example, the proposal in Ito *et al.* [194], the monotone formulas construction [195], and the monotone span programs construction [196, 197]. A monotone function is a function entirely non-increasing or non-decreasing, i.e., its first derivative does not change the sign. Every monotone formula computes a monotone function and every monotone function can be implemented using just AND and OR operators. Benaloh and Leichter [195] proved that if an access structure can be described by a monotone formula then it has an efficient perfect secret-sharing scheme.

The distributed trust aims at interacting with the most secure, honest, and trustworthy entities because this minimizes the exposure to risky transactions. One strategy for distributed trust is a human-like mechanism based on reputation that chooses between benevolent and malicious behavior. Then using relationships and inferring rules, different levels of trust are derived for other entities [193]. This way, reputation is an assessment based on the history of interactions with or observations of an entity, either directly with the evaluator (personal experience) or as reported by others (recommendations or third-party verification). A second mechanism to determine trust is using policies that describe the conditions necessary to obtain trust. It can also prescribe actions and outcomes if certain conditions are met [198]. Policies frequently involve the exchange or verification of credentials, which are information issued (and sometimes endorsed using a digital signature) by one entity, and may describe qualities or features of another entity. Also, Distributed Ledger Technologies, like Blockchain, are characterized by transparency, traceability, and security by design. These features make the adoption of Blockchain attractive to enhance information security, privacy, and trustworthiness in very different contexts including distributed trust [199].

*Advantages and limitations:* This approach is useful to address compromised components and it is effective for a determined number of compromised devices. As a result, the information used for the feedback control is more accurate and it is harder to execute commands based on fake information. However, when the majority is wrong, the stability and the correct behavior of the system are also compromised. Another limitation is the required time to synchronize the information between all the nodes. For this reason, the decision process can take a long time which is not suitable for real-time applications.

*4.3.6* ***Game Theory.*** Approaches based on game-theoretic strategies use mathematical models to analyze the situation where players choose a different action in an attempt to maximize their returns [200]. It studies decisions made in an environment in which multiple players interact with each other in a strategic setup. This means that game-theoretic approaches provide resilience trying to maximize the cost of attacking the system or minimize the damage that an adversary can apply to the system. For that, each player tries to optimize an objective function. This objective function depends on the choices of the other players in the game. Thus, players cannot optimize their objective independently of the choices of other players. This technique has been proposed to respond to attacks where the defender chooses the optimal response according to the adversarial actions. Game theory provides tools to model advanced adversaries who know the defense strategies and can adjust the attack strategies accordingly. In addition, it is possible to define games in both physical and cyber layers.

In the last years, there have been many proposals on game-theoretic approaches for CPS. For example, Huang et Zhu [166] propose a dynamic game for long-term interaction between a stealthy adversary and a proactive defender. The stealthy and deceptive behaviors are captured by the multi-stage game of incomplete information, where each player has his private information unknown to the other. Both players act strategically according to their beliefs which are formed by multi-stage observation and learning. In addition, Hasan *et al.* [165] design an adversary-defender game-theoretic model for power systems. The adversary can identify the chronological order in which the critical substations and their protection assemblies can be attacked to maximize the overall system damage. The defender can intelligently identify the critical substations to protect such that the system damage can be minimized. Ismail *et al.* [201] model the interactions between an attacker and a defender and derive the minimum defense resources required and the optimal strategy of the defender that minimizes the risk. The solution is analyzed in power systems. Also, Rao *et al.* [170] propose a resilience approach using a game approach to face adversaries. Their functions consist of an infrastructure survival probability and a cost expressed in terms of the number of components attacked and reinforced. Zhu and Basar [169] propose a game-theoretic approach to manipulate the attack surface of the network and create a moving target defense. The notion of attack surface is defined as the set of vulnerabilities of the system that can potentially be exploited by the adversary. The essential goal is to find an optimal configuration policy for the defender to shift the attack surface that minimizes its risk and damage.

Game-theoretic approaches have also been proposed to learn adversary models and estimate their knowledge about the system dynamics. For example, Sanjab and Saad [167] propose a game-theoretic approach to analyze the interactions between one defender and one adversary over a CPS. In this game, the adversary launches cyber-attacks on several cyber components of the CPS to maximize the potential harm to the physical system while the system chooses to defend a set of cyber nodes to thwart the attacks and minimize potential damage to the physical side. Similarly, Kanellopoulos and Vamvoudakis [168] consider the problem of identifying the cognitive capabilities of adversaries. To categorize them, they use an iterative method of optimal responses that determine the policy of an agent with a determined level of intelligence. Then, they formulate a learning algorithm to train the different intelligence levels without any knowledge about the physics of the system.

*Advantages and limitations:* This approach provides a quantitative mechanism for deciding the optimal strategy to face an attack. It may be effective for attacks on sensors, actuators, controllers, or network traffic depending on how the approach is designed. However, most of the existing proposals focus on cyber or network aspects, without considering the physical model of the process. In addition, as the decisions are calculated at runtime it may be hard to analyze and predict the stability of the physical process.

## 5 DISCUSSION

Control theory and cybersecurity are research areas that provide significant contributions to solve security issues in CPS from different perspectives. Similarly to the IoT domain [17], resilience in the CPS domain is a dual problem with a part in the cyber world and the other part in the physical one. As pointed out in [13, 202], both such domains are complementary disciplines that working together have the potential to provide more efficient and effective solutions.

Control theory provides models that precisely describe the underlying physical process, which enables the prediction of future behavior and unforeseen deviations from it. It models the system to analyze attacks and their corresponding detection, mitigation, and recovery schemes. The cybersecurity research community also offers different approaches for numerous security problems in CPS. Such approaches typically focus on the cyber aspects, such as communication networks, protocols, software, and data.

According to [202], CPS security can be divided into two main categories: information security which focuses on cyber and data security, provides methods that are effective on software layers without using any physical model; and secure control theory, which studies how cyber-attacks affect the control system's physical dynamics. Ensuring safety using only information security tools is not sufficient for CPS. Therefore, they should be complemented with secure control theory that provides an attack model and a description of the interaction between the physical world and the control system. It provides a better understanding of the attacks' consequences, and the development of new detection methods, algorithms, and architectures, that make the control systems more resilient to possible attacks and failures.

Certain attacks are undetectable by traditional control-theoretic approaches, for example in situations when the adversary modifies inputs and outputs to be correlated with the estimated model or when the values are chosen by the adversary to fulfill certain properties as described in [5, 6]. The incorporation of cybersecurity strategies to control theory approaches, provided new tools to build approaches to solve this issue as explained in Section 4. Moreover, cybersecurity approaches do not cover all the possible vulnerabilities in the cyber components. Mechanisms to protect specific vulnerabilities may not exist or be too expensive to implement, and even when they are implemented they are also not free of false negatives.

Furthermore, due to the strong coupling between cyber and physical domains, the tools and methodologies developed to ensure cybersecurity are insufficient to secure CPS. For instance, they can fail against purely physical attacks. As an example [35], the confidentiality of encrypted sensor measurements can be violated by placing unencrypted malicious sensors in close proximity to encrypted sensors. The integrity of sensor measurements can be modified by changing a sensor's local environment while control inputs can be changed by directly manipulating system actuators. In such a scenario, message authentication codes or digital signatures fail to recognize an attack. Availability can be compromised by physically shielding sensors and actuators. In this case, anti-jamming and denial-of-service techniques will fail.

The large scale of a CPS may turn physical protection impractical, leaving the system vulnerable to the previous examples. However, in addition to the exposed vulnerabilities created by basic physical attacks, it is possible to create more advanced cyber-physical attacks that generate the same physical effects but using a remote connection and injecting malicious traffic. As shown in Section 3, malicious traffic can be confused with legitimate traffic and be undetectable. This way, by using control theory models, it is possible to implement new advanced and coordinated attacks to exploit CPS. These attacks are capable of bypassing cyber detection as discussed in the literature: the false data injection attack [203, 204], the replay attack [41], the zero-dynamics attack [205], and the covert attack [206]. Last but not least, insider adversaries and human error that generate security breaches have to be also considered to ensure safety.

## 6 OPEN CHALLENGES

The limitations highlighted in the previous section open several guidelines for future research work on the subjects surveyed in this article that would be beneficial for wider adoption of resilience methods and techniques for CPS. Some representative guidelines are briefly presented next, in this section.

### 6.1 System Modeling

In terms of modeling, a *higher interaction between system components*, e.g., cyber and physical components, would make the results more consistent and convincing. Indeed, a proper combination of the cyber-network and control-physical layers could be expanded towards next-generation cyber-physical systems able to properly correlate and repair cross-layer security incidents. Most of the existing resilience techniques and measures focus on protecting the network,

software, or physical components in an independent manner. In a CPS these elements work together and coordinated actions to attack vulnerabilities in the different components that may have dangerous consequences. More integration between the different layers creates systems with better capabilities to react and defend from adversaries. For that reason, resilience techniques should integrate these concepts and have a global view of the components and their interaction because approaching the problem with partial and independent views is not enough to solve the existing security issues.

Concerning *resilient control and attack models*, the control theory domain shows to be more mature than the computer science and cybersecurity fields. However, the integration of both domains creates new challenges that need to be addressed. For example, how to create attack-tolerant control, i.e., how to design robust control that considers possible attacks. Proactive algorithms and system architectures that are robust to attacks, ensure stability, and the performance thresholds are still required. In addition, the state of the cyber and network components should also be taken into account to consider factors such as the nodes' states and quality of service. To achieve that, it is also needed to improve the existing attack models, i.e., create attack models that characterize better the capabilities of the adversaries. One adversary model was developed in [6] which is based on the available resources to an adversary. However, better models are still required including information such as their computational power, the type of access they may have, the data they collect, their collaborative capabilities, and signals an adversary has access to. This information helps to understand the logic behind the associated defense mechanisms, e.g., to improve or compare them with other security mechanisms.

A promising research opportunity related to the topics of this article can be explored around the use of *digital twins*. In Section 2, we present how to design the control loops using Kalman filters as estimators used for stochastic cases. Such filters explicitly use a noise model for both state and output processes considering the stochastic nature of the dynamical system. Thus, it is more appropriate for CPS and, in general, performs better for stochastic systems. Conversely, an observer, such as the Luenberger observer [62], is typically restricted to the deterministic cases, i.e., when there is no randomness in the states. Observers are used to estimate unmeasured states of a system and have been proposed to detect attacks in CPS. The principle of estimators and observers are similar. An observer is a continuous-time dynamical system that takes as input the measured input and measured output of the system, and produces an estimate of the state of the system as output.

### 6.2 Metrics and Evaluation Methods

Some more efforts are needed on specifying *complexity management to anticipate impacts on resilience*, e.g., to evaluate in a resilience approach how to manage the complexity of the proposal and how to anticipate the impact it may have on the system resilience. The resilience of a system is influenced by several factors that can be managed or exploited to enhance resilience [46, 207]. All resilience-enhancing measures can also cause a negative effect leading to an overall reduction in resilience. For example, to improve resilience, it may be required to use more complexity, such as using new connections, new components, more diversity, etc. As the number and heterogeneity of components grow, they offer more opportunities to regenerate the system. Agents may be able to use additional links to different elements or find replacement resources to ultimately restore their functions. However, high complexity may lead to interactions that are hard to understand, analyze, and protect, causing unforeseen side effects. As a result, greater complexity may also reduce the resiliency of the system. Another example is fail-safe designs that disconnect a component or part of the system in case of compromise. This action prevents the spread and cascade failures. However, this might be

detrimental to the overall resilience of the system if the component is needed to support other components that execute damage-absorbing actions.

The increase in complexity may lead to lower resilience by increasing the number of ways in which one failed component may cause the failure of another. Therefore, in most cases, greater complexity should be avoided when possible unless it directly supports resilience functions. As a consequence, it is not enough only an analysis of the performance impact of an approach. Resilience proposals may also have hidden impacts on the system's behavior and complexity that should be evaluated to consider the reduction in the overall resilience. The quantification and evaluation of this aspect is not trivial. An approach should never be implemented in production systems without an appropriate evaluation of these factors. How to appropriately analyze and measure the resilience enhancement to reveal potential negative impacts and systemic effects is another future research work.

Another line in terms of metrics and evaluation methods relies on *safety ensuring and testing automation*. CPS normally provide critical functionalities. It is essential to ensure stability and correct behavior even under an attack when the inputs are specially modified for malicious purposes. In addition, triggering defensive actions increases the complexity of the system. Hence, with all these aspects happening at the same time may be hard to ensure that safety-critical functions will continue to work properly in any context or situation. Testing and validating the security proposals to ensure physical safety is still an open issue.

### 6.3 Testing and Validation Environments

There is a need to develop global approaches in terms of *scalability validation*. Indeed, real CPS may scale into networks with hundreds or thousands of devices. Conti *et al.* [208] survey validation testbeds and datasets for CPS. As a result, we can observe that scalability makes it difficult to test the system in an integrated manner considering physical, network, and cyber components. To test scalability normally simulation tools are used, but they abstract or forget the physical process part which is the essential part of the CPS. The ideal validation option is experimental testbeds, which may be expensive and also exists limited stable testbed scenarios. Thus, testing scalability while combining physical process, network, and software components is still a challenge. We highlight the need for better CPS testing and validation environments. Numeric simulation tools, such as Matlab® and Simulink®, do not integrate the network and cyber aspects. Network simulation tools are conceived for traditional IT systems and do not integrate the physical process. Hence, performance validation in simulation platforms only gives a partial overview of the whole problem.

In particular, for testing network aspects, it will not be enough to test with reduced quantities of the devices. This presents two new issues. First, creating such a testbed is not easy due to the required investment. Second, the existing testbed scenarios consider only a limited quantity of devices. The lack of realistic scenarios is mainly due to the complexity of creating system models describing the different aspects of a physical process, such as the existing physical process reactions, the physical model involved in those reactions, the physical equipment or components required, the safety and operating constraints, the operating cost function, the sensor signal noise, the process randomness, among others [38]. Designing such a system is a huge effort and insights into real industrial systems are not possible due to justified confidentiality issues.

## 7 CONCLUSION

In Cyber-Physical Systems (CPS), adversaries may disrupt physical processes by injecting malicious traffic, e.g., cyber-physical attacks may use coordinated cross-layer techniques, to get control over the cyber or network layers and disrupt the physical devices. For this reason, attacks over critical processes may end up affecting people, physical environments,

and companies. To develop comprehensive protection for CPS, it is required to layer the three following protection mechanisms: prevention to postpone the attack as much as possible, detection-reaction to identify the attacks and attenuate them, and cyber-resilience to contain the impact of the attack while providing essential services and restoring normal operations as soon as possible.

Cyber-resilience is essential for critical systems that monitor industrial and complex infrastructures based on networked control systems [209]. If the defense strategy relies only on detection and reaction approaches, the system is not protected in case of false negatives, i.e., undetectable attacks or extremely rare events that are not considered in risk management. Attacks might also come from inside, for example, from highly skilled employees acting as malicious insiders. The knowledge that such insiders possess about the system gives them unrestricted access to steal or modify data or even deactivate critical functionalities. It is important to have a CPS capable of maintaining the stability of the system during such situations. The system should be protected at all times including the time required for detecting and responding to attacks. Otherwise, the system could experience disruption, leading to damages.

In this article, we presented a systematization of knowledge about existing scientific efforts of making CPS cyber-resilient. We systematically surveyed recent literature addressing the topic, with a specific focus on techniques that may be used on CPS. We started by surveying control theory formalities for CPS and cyber-physical attacks. Then, we analyzed detection and mitigation techniques to protect CPS. We surveyed some current trends in terms of detection based on control-theoretic model-based approaches that incorporate the physical model to detect cyber-physical adversaries. We also surveyed mitigation techniques aiming to optimize the recovery response of a system under attack. The proposals to build cyber-resilient systems turn around techniques such as diversity, segmentation, resilient control, system reconfiguration, dynamic software evolution, moving target defense, consensus, and game theory paradigms. These techniques provide the ability to absorb, survive, or recover from an attack.

We discussed how the techniques have evolved and we brought clarity to this complex field by treating the major axes of resilience techniques. We identified that the difference between the detection-reaction paradigm and resilience is not clearly defined in the literature, and often, the two concepts are confused. This problem arises for different causes. Firstly, because resilient designs are not easy to conceive. Our natural way of reasoning about security instructions is to detect the problem and then react. Another reason is probably that control theory and computer science have different definitions for the resilience concept. Control theory calls resilient a controller that can keep an understanding of the system state and calculate correct control signals despite malicious information injected at any point of the control loop. To achieve this, the control theory community normally uses approaches that in computer science are considered detection-reaction approaches. On the other hand, from a computer science perspective, a resilient system is capable to prepare, absorb, recover, and adapt to adverse effects. Or as we prefer to define it, a resilient system is capable of maintaining the core set of critical functionalities despite ongoing adversarial misbehavior and guaranteeing the recovery of the normal operation within a predefined cost limit.

As a result of the literature analysis, we identified plenty of research efforts in terms of detection techniques and state estimation to maintain an awareness of the system state despite an attack. However, much less efforts exist in terms of remediation approaches to attenuate the attacks. We identified a lack of adapted resilience techniques for the CPS particular needs. The research in resilience for CPS can be extended and we pointed out several promising directions for future work, with a focus on the practical aspects of cyber-resilience, such as the use of metrics and evaluation methods, as well as testing and validation environments.

**Acknowledgements** — The authors thank the anonymous referees for their valuable comments and helpful suggestions. The authors also acknowledge support from the Cyber CNI chair of the Institut Mines-Télécom, as well as support from the European Commission, under grant agreement 830892 (H2020 SPARTA project).

## REFERENCES

[1] X. Ge, F. Yang, and Q. Han. Distributed networked control systems: A brief overview. *Information Sciences*, 380:117–131, February 2017.
[2] X. M. Zhang, Q. L. Han, and X. Yu. Survey on Recent Advances in Networked Control Systems. *IEEE Transactions on Industrial Informatics*, 12(5):1740–1752, October 2016.
[3] L. Zhang, H. Gao, and O. Kaynak. Network-induced constraints in networked control systems — a survey. *IEEE Transactions on Industrial Informatics*, 9(1):403–416, 2013.
[4] Y. Z. Lun, A. D'Innocenzo, I. Malavolta, and M. D. Di Benedetto. Cyber-Physical Systems Security: a Systematic Mapping Study. *Journal of Systems and Software*, 149:174–216, March 2019. arXiv: 1605.09641.
[5] A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson. Attack Models and Scenarios for Networked Control Systems. In *Proceedings of the 1st International Conference on High Confidence Networked Systems*, HiCoNS '12, pages 55–64, New York, NY, USA, 2012. ACM.
[6] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson. A secure control framework for resource-limited adversaries. *Automatica*, 51:135–148, 2015.
[7] L. Fillatre, I. Nikiforov, P. Willett, et al. Security of scada systems against cyber–physical attacks. *IEEE Aerospace and Electronic Systems Magazine*, 32(5):28–45, 2017.
[8] R. Alguliyev, Y. Imamverdiyev, and L. Sukhostat. Cyber-physical systems and their security issues. *Computers in Industry*, 100:212–223, 2018.
[9] N. Falliere, L. O. Murchu, and E. Chien. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, 5:6, 2011.
[10] D. Corman, V. Pillitteri, S. Tousley, M. Tehranipoor, and U. Lindqvist. NITRD Cyber-Physical Security Panel. 35th IEEE Symposium on Security and Privacy, IEEE S&P 2014, San Jose, CA, USA, May 18-21.
[11] D. U. Case. Analysis of the cyber attack on the ukrainian power grid. *Electricity Information Sharing and Analysis Center (E-ISAC)*, 2016.
[12] J. Slay and M. Miller. Lessons learned from the maroochy water breach. In *Critical Infrastructure Protection*, pages 73–82, Boston, MA, 2008. Springer US.
[13] H. S. Sánchez, D. Rotondo, T. Escobet, V. Puig, and J. Quevedo. Bibliographical review on cyber attacks from a control oriented perspective. *Annual Reviews in Control*, 48:103–128, 2019.
[14] Y. L. Huang, A. A. Cárdenas, S. Amin, Z. S. Lin, H. Y. Tsai, and S. Sastry. Understanding the physical and economic consequences of attacks on control systems. *International Journal of Critical Infrastructure Protection*, 2(3):73 – 83, 2009.
[15] J. Rubio-Hernan, L. De Cicco, and J. Garcia-Alfaro. Event-triggered watermarking control to handle cyber-physical integrity attacks. In *21st Nordic Conference on Secure IT Systems (NordSec 2016)*, pages 3–19. Springer, November 2016.
[16] J. Rubio-Hernan, L. De Cicco, and J. Garcia-Alfaro. Adaptive control-theoretic detection of integrity attacks against cyber-physical industrial systems. *Transactions on Emerging Telecommunications Technologies*, 32(09), 2017.
[17] C. Berger, P. Eichhammer, H. P. Reiser, J. Domaschka, F. J. Hauck, and G. Habiger. A survey on resilience in the IoT: Taxonomy, classification, and discussion of resilience mechanisms. *ACM Comput. Surv.*, 54(7), sep 2021.
[18] S. Jackson. *Architecting Resilient Systems: Accident Avoidance and Survival and Recovery from Disruptions.* 2010.
[19] A. F. M. Piedrahita, V. Gaur, J. Giraldo, A. A. Cardenas, and S. J. Rueda. Leveraging software-defined networking for incident response in industrial control systems. *IEEE Software*, 35(1):44–50, January 2018.
[20] B. Rathnayaka, C. Siriwardana, D. Robert, D. Amaratunga, and S. Setunge. Improving the resilience of critical infrastructures: Evidence-based insights from a systematic literature review. *International Journal of Disaster Risk Reduction*, 78:103123, 2022.
[21] Y. Zhang, F. Xie, Y. Dong, G. Yang, and X. Zhou. High Fidelity Virtualization of Cyber-Physical Systems. *International Journal of Modeling, Simulation, and Scientific Computing*, 4(2):1–26, June 2013.
[22] D. I. Urbina, J. Giraldo, A. A. Cardenas, J. Valente, M. Faisal, N. O. Tippenhauer, J. Ruths, R. Candell, and H. Sandberg. Survey and New Directions for Physics-Based Attack Detection in Control Systems. In *Grant/Contract Reports (NISTGCR)*, pages 1–37. National Institute of Standards and Technology (NIST), Nov 2016.
[23] L. Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1–12, 2010.
[24] G. C. Goodwin, M. Gevers, and B. Ninness. Quantifying the error in estimated transfer functions with application to model order selection. *IEEE Transactions on Automatic Control*, 37(7):913–928, Jul 1992.
[25] L. Ljung. *System identification: Theory for the User.* Prentice-Hall, Inc., 1987.
[26] H. Natke. System identification: Torsten Söderström and Petre Stoica. *Automatica*, 28(5):1069–1071, 1992.
[27] N. L. Ricker. Model predictive control of a continuous, nonlinear, two-phase reactor. *Journal of Process Control*, 3(2):109–123, 1993.
[28] M. Barenthin Syberg. *Complexity Issues, Validation and Input Design for Control in System Identification.* PhD thesis, KTH School of Electrical Engineering, Stockholm, Sweden, 2008.

[29] T. H. Lee, W. S. Ra, S. H. Jin, T. S. Yoon, and J. B. Park. Robust Extended Kalman Filtering via Krein Space Estimation. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E87-A(1):243–250, 2004.

[30] K. Ogata. *Modern Control Engineering*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 4th edition, 2001.

[31] A. Barrientos, I. Aguirre, J. Del Cerro, and P. Portero. LQG vs PID in attitude control of a unmanned aerial vehicle in hover. In *10th International Conference on Advanced Robotics (ICAR) 2001*, pages 599–604, Aug 2001.

[32] G. F. Franklin, J. D. Powell, and M. L. Workman. *Digital control of dynamic systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edition, March 1998.

[33] Y. Mo, S. Weerakkody, and B. Sinopoli. Physical Authentication of Control Systems: Designing Watermarked Control Inputs to Detect Counterfeit Sensor Outputs. *IEEE Control Systems*, 35(1):93–109, February 2015.

[34] J. Rubio-Hernan, L. De Cicco, and J. Garcia-Alfaro. On the use of Watermark-based Schemes to Detect Cyber-Physical Attacks. *EURASIP Journal on Information Security*, 2017:1–25, June 2017.

[35] S. Weerakkody, O. Ozel, Y. Mo, and B. Sinopoli. Resilient Control in Cyber-Physical Systems: Countering Uncertainty, Constraints, and Adversarial Behavior. *Foundations and Trends® in Systems and Control*, 7(1-2):1–252, 2020.

[36] R. S. Smith. Covert Misappropriation of Networked Control Systems: Presenting a Feedback Structure. *IEEE Control Systems*, 35(1):82–92, Feb 2015.

[37] A. Humayed, J. Lin, F. Li, and B. Luo. Cyber-Physical Systems Security – A Survey. *arXiv:1701.04525 [cs]*, January 2017. arXiv: 1701.04525.

[38] M. Krotofil and J. Larsen. Rocking the pocket book: Hacking chemical plants for competition and extortion. *DEF CON*, 23, 2015.

[39] S. M. Dibaji, M. Pirani, D. B. Flamholz, A. M. Annaswamy, K. H. Johansson, and A. Chakrabortty. A systems and control perspective of cps security. *Annual Reviews in Control*, 47:394 – 411, 2019.

[40] 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*, ASIACCS '11, pages 355–366, New York, NY, USA, 2011. ACM.

[41] Y. Mo, R. Chabukswar, and B. Sinopoli. Detecting integrity attacks on SCADA systems. *IEEE Transactions on Control Systems Technology*, 22(4):1396–1407, July 2014.

[42] Y. Yuan, Q. Zhu, F. Sun, Q. Wang, and T. Başar. Resilient control of cyber-physical systems against Denial-of-Service attacks. In *2013 6th International Symposium on Resilient Control Systems (ISRCS)*, pages 54–59, Aug 2013.

[43] W. Gao, T. Morris, B. Reaves, and D. Richey. On SCADA control system command and response injection and intrusion detection. In *2010 eCrime Researchers Summit*, pages 1–9, Oct 2010.

[44] W. Gao and T. H. Morris. On cyber attacks and signature based intrusion detection for modbus based industrial control systems. *The Journal of Digital Forensics, Security and Law: JDFSL*, 9(1):37, 2014.

[45] Y. Chen, S. Kar, and J. M. F. Moura. Dynamic Attack Detection in Cyber-Physical Systems with Side Initial State Information. *IEEE Transactions on Automatic Control*, PP(99):1–1, 2016.

[46] A. Kott and I. Linkov. *Cyber Resilience of Systems and Networks, Springer, 475 pages*. 2019.

[47] A. Clark and S. Zonouz. Cyber-Physical Resilience: Definition and Assessment Metric. *IEEE Transactions on Smart Grid*, 10(2):1671–1684, 2019.

[48] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[49] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[50] T. Hofmann, B. Schölkopf, and A. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36, 01 2007.

[51] R. Mitchell and I.-R. Chen. A survey of intrusion detection techniques for cyber-physical systems. *ACM Comput. Surv.*, 46(4), March 2014.

[52] M. Cheminod, L. Durante, and A. Valenzano. Review of Security Issues in Industrial Networks. *IEEE Transactions on Industrial Informatics*, 9(1):277–293, February 2013. Conference Name: IEEE Transactions on Industrial Informatics.

[53] S. Han, M. Xie, H. Chen, and Y. Ling. Intrusion detection in cyber-physical systems: Techniques and challenges. *IEEE Systems Journal*, 8(4):1052–1062, 2014.

[54] A. Ahmed, K. Abu Bakar, M. I. Channa, K. Haseeb, and A. W. Khan. A survey on trust based detection and isolation of malicious nodes in ad-hoc and sensor networks. *Frontiers of Computer Science*, 9(2):280–296, April 2015.

[55] D. Ding, Q. L. Han, Y. Xiang, X. Ge, and X. M. Zhang. A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomputing*, 275:1674–1683, January 2018.

[56] J. M. Beaver, R. C. Borges-Hink, and M. A. Buckner. An evaluation of machine learning methods to detect malicious scada communications. In *2013 12th International Conference on Machine Learning and Applications*, volume 2, pages 54–59, 2013.

[57] S. Amin, X. Litrico, S. S. Sastry, and A. M. Bayen. Cyber security of water scada systems—part ii: Attack detection using enhanced hydrodynamic models. *IEEE Transactions on Control Systems Technology*, 21(5):1679–1693, 2013.

[58] M. Dehghani, Z. Khalafi, A. Khalili, and A. Sami. Integrity attack detection in PMU networks using static state estimation algorithm. In *2015 IEEE Eindhoven PowerTech*, pages 1–6, June 2015.

[59] Q. Zhu and T. Başar. Game-theoretic methods for robustness, security, and resilience of cyber-physical control systems: Games-in-games principle for optimal cross-layer resilient control systems. *IEEE Control Systems*, 35(1):46–65, 2015.

[60] R. B. Bobba, K. M. R. Q. Wang, H. Khurana, K. Nahtstedt, and T. J. Overbye. Detecting False Data Injection Attacks on DC State Estimation. In *Proceeding of the 1st Workshop on Secure Control Systems (CPSWEEK)*, pages 1–9. Citeseer, April 2010.

[61] F. Pasqualetti, F. Dorfler, and F. Bullo. Control-Theoretic Methods for Cyberphysical Security: Geometric Principles for Optimal Cross-Layer Resilient Control Systems. *IEEE Control Systems*, 35(1):110–127, Feb 2015.

[62] D. Luenberger. An introduction to observers. *IEEE Transactions on Automatic Control*, 16(6):596–602, December 1971. Conference Name: IEEE Transactions on Automatic Control.

[63] Y. Shoukry and P. Tabuada. Event-Triggered State Observers for Sparse Sensor Noise/Attacks. *IEEE Transactions on Automatic Control*, 61(8):2079–2091, August 2016.

[64] C. Schellenberger and P. Zhang. Detection of covert attacks on cyber-physical systems by extending the system dynamics with an auxiliary system. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1374–1379, December 2017.

[65] F. Miao, M. Pajic, and G. J. Pappas. Stochastic game approach for replay attack detection. In *52nd IEEE Conference on Decision and Control*, pages 1854–1859, Dec 2013.

[66] V. L. Do, L. Fillatre, and I. Nikiforov. A statistical method for detecting cyber/physical attacks on SCADA systems. In *2014 IEEE Conference on Control Applications (CCA)*, pages 364–369, Oct 2014.

[67] A. Arvani and V. S. Rao. Detection and protection against intrusions on smart grid systems. *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, 3(1):38–48, 2014.

[68] A. Y. Lokhov, N. Lemons, T. C. McAndrew, A. Hagberg, and S. Backhaus. Detection of Cyber-Physical Faults and Intrusions from Physical Correlations. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 303–310, Dec 2016.

[69] Y. Wang, Z. Xu, J. Zhang, L. Xu, H. Wang, and G. Gu. SRID: State Relation Based Intrusion Detection for False Data Injection Attacks in SCADA. In *Computer Security - ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part II*, pages 401–418. Springer International Publishing, 2014.

[70] P.-Y. Chen, S. Yang, and J. A. McCann. Distributed Real-Time Anomaly Detection in Networked Industrial Sensing Systems. *IEEE Transactions on Industrial Electronics*, 62(6):3832–3842, June 2015.

[71] Q. Phan and P. Malacaria. All-solution satisfiability modulo theories: Applications, algorithms and benchmarks. In *2015 10th International Conference on Availability, Reliability and Security*, pages 100–109, 2015.

[72] H. Beikzadeh and H. J. Marquez. Multirate observers for nonlinear sampled-data systems using input-to-state stability and discrete-time approximation. *IEEE Transactions on Automatic Control*, 59(9):2469–2474, 2014.

[73] H. Tan, B. Shen, Y. Liu, A. Alsaedi, and B. Ahmad. Event-triggered multi-rate fusion estimation for uncertain system with stochastic nonlinearities and colored measurement noises. *Information Fusion*, 36:313 – 320, 2017.

[74] H. Fawzi, P. Tabuada, and S. Diggavi. Secure Estimation and Control for Cyber-Physical Systems Under Adversarial Attacks. *IEEE Transactions on Automatic Control*, 59(6):1454–1467, June 2014.

[75] M. Pajic, J. Weimer, N. Bezzo, P. Tabuada, O. Sokolsky, I. Lee, and G. J. Pappas. Robustness of attack-resilient state estimators. In *2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pages 163–174, Berlin, Germany, April 2014. IEEE.

[76] M. Pajic, I. Lee, and G. J. Pappas. Attack-Resilient State Estimation for Noisy Dynamical Systems. *IEEE Transactions on Control of Network Systems*, 4(1):82–92, March 2017.

[77] Y. Mo and B. Sinopoli. Secure estimation in the presence of integrity attacks. *IEEE Transactions on Automatic Control*, 60(4):1145–1151, 2015.

[78] J. Keller, K. Chabir, and D. Sauter. Input reconstruction for networked control systems subject to deception attacks and data losses on control signals. *International Journal of Systems Science*, 47(4):814–820, 2016.

[79] J. Weimer, N. Bezzo, M. Pajic, O. Sokolsky, and I. Lee. Attack-resilient minimum mean-squared error estimation. In *2014 American Control Conference*, pages 1114–1119, Portland, OR, USA, June 2014. IEEE.

[80] Y. Shoukry, P. Nuzzo, A. Puggelli, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and P. Tabuada. Secure State Estimation for Cyber-Physical Systems Under Sensor Attacks: A Satisfiability Modulo Theory Approach. *IEEE Transactions on Automatic Control*, 62(10):4917–4932, October 2017.

[81] S. Mishra, Y. Shoukry, N. Karamchandani, S. N. Diggavi, and P. Tabuada. Secure State Estimation Against Sensor Attacks in the Presence of Noise. *IEEE Transactions on Control of Network Systems*, 4(1):49–59, March 2017.

[82] L. De Moura and N. Bjørner. Satisfiability modulo theories: Introduction and applications. *Commun. ACM*, 54(9):69–77, September 2011.

[83] W. Yang, L. Lei, and C. Yang. Event-based distributed state estimation under deception attack. *Neurocomputing*, 270:145 – 151, 2017. Distributed Control and Optimization with Resource-Constrained Networked Systems.

[84] L. Lei, W. Yang, and C. Yang. Event-based distributed state estimation over a wsn with false data injection attack. *IFAC-PapersOnLine*, 49(22):286 – 290, 2016. 6th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2016.

[85] J. Giraldo, E. Sarkar, A. A. Cardenas, M. Maniatakos, and M. Kantarcioglu. Security and Privacy in Cyber-Physical Systems: A Survey of Surveys. *IEEE Design Test*, 34(4):7–17, August 2017. Conference Name: IEEE Design Test.

[86] X. Li, C. Zhou, Y. Tian, and Y. Qin. A dynamic decision-making approach for intrusion response in industrial control systems. *IEEE Transactions on Industrial Informatics*, 15(5):2544–2554, 2019.

[87] A. R. Cavalli, A. M. Ortiz, G. Ouffoué, C. A. Sanchez, and F. Zaïdi. Design of a secure shield for internet and web-based services using software reflection. In *Web Services – ICWS 2018*, pages 472–486, Cham, 2018. Springer International Publishing.

[88] Z. Ismail, J. Leneutre, and A. Fourati. Optimal deployment of security policies: Application to industrial control systems. In *2018 14th European Dependable Computing Conference (EDCC)*, pages 120–127, 2018.

[89] C. Kiennert, Z. Ismail, H. Debar, and J. Leneutre. A survey on game-theoretic approaches for intrusion detection and response optimization. *ACM Comput. Surv.*, 51(5), August 2018.

[90] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 3270–3285, 2012.

[91] A. Cetinkaya, H. Ishii, and T. Hayakawa. Networked control under random and malicious packet losses. *IEEE Transactions on Automatic Control*, PP, 06 2016.

[92] Z. Sun, W. Xue, J. Liu, F. Chen, and X. Lu. Adaptive event-triggered resilient control of industrial cyber physical systems under asynchronous data injection attack. *Journal of the Franklin Institute*, 359(7):3000–3023, 2022.

[93] A. T. Campbell, I. Katzela, K. Miki, and J. Vicente. Open signaling for atm, internet and mobile networks (opensig'98). *SIGCOMM Comput. Commun. Rev.*, 29(1):97–108, January 1999.

[94] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J. Minden. A survey of active network research. *Comm. Mag.*, 35(1):80–86, January 1997.

[95] R. Enns and M. Bjorklund and J. Schoenwaelder and A. Bierman. Network configuration protocol (NETCONF) - Internet Engineering Task Force, RFC 6241. , June 2011.

[96] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, 103(1):14–76, Jan 2015.

[97] R. Sahay, G. Blanc, Z. Zhang, and H. Debar. Towards autonomic DDoS mitigation using Software Defined Networking. In *SENT 2015 : NDSS Workshop on Security of Emerging Networking Technologies*, page ., San Diego, Ca, United States, February 2015. Internet society.

[98] N. Hachem, H. Debar, and J. Garcia-Alfaro. HADEGA: A novel MPLS-based mitigation solution to handle network attacks. In *31st IEEE International Performance Computing and Communications Conference, IPCCC 2012, Austin, TX, USA, December 1-3, 2012*, pages 171–180, 2012.

[99] J. Rubio-Hernan, R. Sahay, L. De Cicco, and J. Garcia-Alfaro. Cyber-physical architecture assisted by programmable networking. *Internet Technology Letters*, page e44, 2018.

[100] E. Molina and E. Yang. Software-defined networking in cyber-physical systems: A survey | Elsevier Enhanced Reader, 2017.

[101] A. F. M. Piedrahita, V. Gaur, J. Giraldo, A. A. Cardenas, and S. J. Rueda. Virtual incident response functions in control systems. *Computer Networks*, 135:147–159, 2018.

[102] T. Jackson, B. Salamat, A. Homescu, K. Manivannan, G. Wagner, A. Gal, S. Brunthaler, C. Wimmer, and M. Franz. *Compiler-Generated Software Diversity*, pages 77–98. Springer New York, New York, NY, 2011.

[103] B. De Sutter, B. Anckaert, J. Geiregat, D. Chanet, and K. De Bosschere. Instruction set limitation in support of software diversity. In *Information Security and Cryptology – ICISC 2008*, pages 152–165, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[104] T. Jackson, A. Homescu, S. Crane, P. Larsen, S. Brunthaler, and M. Franz. Diversifying the software stack using randomized nop insertion. In *Moving Target Defense II*, pages 151–173, New York, NY, 2013. Springer New York.

[105] S. Bhatkar and R. Sekar. Data space randomization. In *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 1–22, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[106] 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*, CCS '03, page 272–280, New York, NY, USA, 2003. Association for Computing Machinery.

[107] A. Homescu, S. Neisius, P. Larsen, S. Brunthaler, and M. Franz. Profile-guided automated software diversity. In *Proceedings of the 2013 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*, pages 1–11, 2013.

[108] L. V. Davi, A. Dmitrienko, S. Nürnberger, and A.-R. Sadeghi. Gadge me if you can: Secure and efficient ad-hoc instruction-level randomization for x86 and arm. In *8th ACM SIGSAC symposium on Information, computer and communications security (ACM ASIACCS 2013)*, pages 299–310, 2013.

[109] P. Larsen, A. Homescu, S. Brunthaler, and M. Franz. SoK: Automated Software Diversity. In *2014 IEEE Symposium on Security and Privacy*, pages 276–291, May 2014. ISSN: 2375-1207.

[110] G. Ouffoué, F. Zaïdi, A. R. Cavalli, and M. Lallali. How web services can be tolerant to intruders through diversification. In *2017 IEEE International Conference on Web Services (ICWS)*, pages 436–443, 2017.

[111] L. Chen and A. Avizienis. N-version programminc: A fault-tolerance approach to rellablllty of software operatlon. In *Twenty-Fifth International Symposium on Fault-Tolerant Computing, 1995, ' Highlights from Twenty-Five Years'.*, page 113, 1995.

[112] A. Chaves, M. Rice, S. Dunlap, and J. Pecarina. Improving the cyber resilience of industrial control systems. *International Journal of Critical Infrastructure Protection*, 17:30–48, June 2017.

[113] F. B. Cohen. Operating system protection through program evolution. *Computers & Security*, 12(6):565 – 584, 1993.

[114] S. Forrest, A. Somayaji, and D. H. Ackley. Building diverse computer systems, 1997.

[115] A. Homescu, S. Brunthaler, P. Larsen, and M. Franz. Librando: Transparent code randomization for just-in-time compilers. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*, CCS '13, page 993–1004, New York, NY, USA, 2013.

[116] B. Genge and C. Siaterlis. An experimental study on the impact of network segmentation to the resilience of physical processes. In *NETWORKING 2012*, pages 121–134, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[117] M. Krotofil and A. A. Cárdenas. Resilience of process control systems to cyber-physical attacks. In *Secure IT Systems*, pages 166–182, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[118] E. Bellini, F. Bagnoli, A. A. Ganin, and I. Linkov. Cyber Resilience in IoT Network: Methodology and Example of Assessment through Epidemic Spreading Approach. In *2019 IEEE World Congress on Services (SERVICES)*, volume 2642-939X, pages 72–77, July 2019. ISSN: 2642-939X.

[119] L. Chen, D. Yue, C. Dou, Z. Cheng, and J. Chen. Robustness of cyber-physical power systems in cascading failure: Survival of interdependent clusters. *International Journal of Electrical Power & Energy Systems*, 114:105374, January 2020.

[120] A. Avizienis, R. Avizienis, and A. V. Avizienis. The Concept of a Software-Free Resilience Infrastructure for Cyber-Physical Systems. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W)*, pages 230–233, June 2016.

[121] M. A. Haque, S. Shetty, and B. Krishnappa. Modeling Cyber Resilience for Energy Delivery Systems Using Critical System Functionality. In *2019 Resilience Week (RWS)*, volume 1, pages 33–41, November 2019.

[122] A. Kwasinski. Modeling of Cyber-Physical Intra-Dependencies in Electric Power Grids and Their Effect on Resilience. In *2020 8th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems*, pages 1–6, April 2020.

[123] J. Xu, T. Zhang, Y. Du, W. Zhang, T. Yang, and J. Qiu. Islanding and dynamic reconfiguration for resilience enhancement of active distribution systems. *Electric Power Systems Research*, 189:106749, December 2020.

[124] M. Segovia, A. R. Cavalli, N. Cuppens, J. Rubio-Hernan, and J. Garcia-Alfaro. Reflective Attenuation of Cyber-Physical Attacks. In *Computer Security*, pages 19–34, Cham, 2020. Springer International Publishing.

[125] F. Januário, A. Cardoso, and P. Gil. A Distributed Multi-Agent Framework for Resilience Enhancement in Cyber-Physical Systems. *IEEE Access*, 7:31342–31357, 2019. Conference Name: IEEE Access.

[126] C. J. Marshall, B. Roberts, and M. W. Grenn. Context-Driven Autonomy for Enhanced System Resilience in Emergent Operating Environments. *IEEE Systems Journal*, 13(3):2130–2141, September 2019. Conference Name: IEEE Systems Journal.

[127] C. Chen, K. Xie, F. L. Lewis, S. Xie, and R. Fierro. Adaptive synchronization of multi-agent systems with resilience to communication link faults. *Automatica*, 111:108636, January 2020.

[128] P. Griffioen, R. Romagnoli, B. H. Krogh, and B. Sinopoli. Secure networked control for decentralized systems via software rejuvenation. In *2020 American Control Conference (ACC)*, pages 1266–1273, 2020.

[129] S. Pradhan, A. Dubey, T. Levendovszky, P. S. Kumar, W. A. Emfinger, D. Balasubramanian, W. Otte, and G. Karsai. Achieving resilience in distributed software systems via self-reconfiguration. *Journal of Systems and Software*, 122:344 – 363, 2016.

[130] Q. Duan, E. Al-Shaer, and H. Jafarian. Efficient random route mutation considering flow and network constraints. In *2013 IEEE Conference on Communications and Network Security (CNS)*, pages 260–268, Oct 2013.

[131] D. Ma, C. Lei, L. Wang, H. Zhang, Z. Xu, and M. Li. A self-adaptive hopping approach of moving target defense to thwart scanning attacks. In *Information and Communications Security*, pages 39–53, Cham, 2016. Springer International Publishing.

[132] C. Lei, H. Q. Zhang, J. L. Tan, Y. C. Zhang, and X. H. Liu. Moving Target Defense Techniques: A Survey. *Security and Communication Networks*, 2018:1–25, July 2018.

[133] J. Zheng and A. S. Namin. A Survey on the Moving Target Defense Strategies: An Architectural Perspective. *Journal of Computer Science and Technology*, 34(1):207–233, January 2019.

[134] B. Potteiger, A. Dubey, F. Cai, X. Koutsoukos, and Z. Zhang. Moving target defense for the security and resilience of mixed time and event triggered cyber–physical systems. *Journal of Systems Architecture*, 125:102420, 2022.

[135] X. Xu, H. Hu, Y. Liu, J. Tan, H. Zhang, and H. Song. Moving target defense of routing randomization with deep reinforcement learning against eavesdropping attack. *Digital Communications and Networks*, 8(3):373–387, 2022.

[136] M. Azab, M. Samir, and E. Samir. "MystifY": A proactive Moving-Target Defense for a resilient SDN controller in Software Defined CPS. *Computer Communications*, 189:205–220, 2022.

[137] A. Kanellopoulos and K. Vamvoudakis. A Moving Target Defense Control Framework for Cyber-Physical Systems. *IEEE Transactions on Automatic Control*, pages 1–1, 2019.

[138] E. Al-Shaer, Q. Duan, and J. Jafarian. Random host mutation for moving target defense. In *Security and Privacy in Communication Networks*, pages 310–327, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[139] S. Antonatos, P. Akritidis, E. Markatos, and K. Anagnostakis. Defending against hitlist worms using network address space randomization. *Computer Networks*, 51(12):3471 – 3490, 2007.

[140] V. Heydari. Moving target defense for securing scada communications. *IEEE Access*, 6:33329–33343, 2018.

[141] R. Zhuang, S. A. DeLoach, and X. Ou. Towards a theory of moving target defense. In *Proceedings of the First ACM Workshop on Moving Target Defense*, MTD '14, page 31–40, New York, NY, USA, 2014. Association for Computing Machinery.

[142] D. C. MacFarland and C. A. Shue. The sdn shuffle: Creating a moving-target defense using host-based software-defined networking. In *MTD '15*, page 37–41, New York, NY, USA, 2015. Association for Computing Machinery.

[143] S. Dolev and S. T. David. SDN-Based Private Interconnection. In *2014 IEEE 13th International Symposium on Network Computing and Applications*, pages 129–136, Aug 2014.

[144] A. Aseeri, N. Netjinda, and R. Hewett. Alleviating eavesdropping attacks in software-defined networking data plane. In *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, CISRC '17, pages 1:1–1:8, New York, NY, USA, 2017. ACM.

[145] S. Weerakkody and B. Sinopoli. A moving target approach for identifying malicious sensors in control systems. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1149–1156, Monticello, IL, USA, September 2016. IEEE.

[146] M. Segovia-Ferreira, J. Rubio-Hernan, R. Cavalli, and J. Garcia-Alfaro. Switched-based resilient control of cyber-physical systems. *IEEE Access*, 8:212194–212208, 2020.

[147] J. Giraldo, M. El Hariri, and M. Parvania. Moving Target Defense for Cyber-Physical Systems Using IoT-Enabled Data Replication. *IEEE Internet of Things Journal*, pages 1–1, 2022.

[148] M. Liu, C. Zhao, Z. Zhang, R. Deng, P. Cheng, and J. Chen. Converter-based Moving Target Defense Against Deception Attacks in DC Microgrids. *IEEE Transactions on Smart Grid*, pages 1–1, 2021.

[149] P. Griffioen, S. Weerakkody, and B. Sinopoli. A moving target defense for securing cyber-physical systems. *IEEE Transactions on Automatic Control*, pages 1–1, 2020.

[150] J. Giraldo, A. Cardenas, and R. G. Sanfelice. A Moving Target Defense to Detect Stealthy Attacks in Cyber-Physical Systems. In *2019 American Control Conference (ACC)*, pages 391–396, 2019.

[151] N. Mavrogiannopoulos, N. Kisserli, and B. Preneel. A taxonomy of self-modifying code for obfuscation. *Comput. Secur.*, 30(8):679–691, 2011.

[152] Z. He, K. Ben, and Z. Zhang. Software Architectural Reflection Mechanism for Runtime Adaptation. In *2008 The 9th International Conference for Young Computer Scientists*, pages 1101–1105, Hunan, China, November 2008. IEEE.

[153] F. Kon, F. Costa, G. Blair, and R. H. Campbell. The case for reflective middleware. *Communications of the ACM*, 45(6), June 2002.

[154] D. Saldaña, A. Prorok, S. Sundaram, M. F. M. Campos, and V. Kumar. Resilient consensus for time-varying networks of dynamic agents. In *2017 American Control Conference (ACC)*, pages 252–258, May 2017. ISSN: 2378-5861.

[155] D. Meng and K. L. Moore. Studies on Resilient Control Through Multiagent Consensus Networks Subject to Disturbances. *IEEE Transactions on Cybernetics*, 44(11):2050–2064, November 2014. Conference Name: IEEE Transactions on Cybernetics.

[156] J. Yan, Y. Mo, X. Li, L. Xing, and C. Wen. Resilient Vector Consensus: An Event-based Approach. In *2020 IEEE 16th International Conference on Control Automation (ICCA)*, pages 889–894, October 2020. ISSN: 1948-3457.

[157] J. Usevitch and D. Panagou. Resilient Leader-Follower Consensus with Time-Varying Leaders in Discrete-Time Systems. pages 5432–5437, December 2019. ISSN: 2576-2370.

[158] M. Shabbir, J. Li, W. Abbas, and X. Koutsoukos. Resilient Vector Consensus in Multi-Agent Networks Using Centerpoints. In *2020 American Control Conference (ACC)*, pages 4387–4392, July 2020. ISSN: 2378-5861.

[159] F. M. Zegers, P. Deptula, J. M. Shea, and W. E. Dixon. Event-Triggered Approximate Leader-Follower Consensus with Resilience to Byzantine Adversaries. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 6412–6417, December 2019. ISSN: 2576-2370.

[160] S. Sundaram and C. N. Hadjicostis. Distributed function calculation via linear iterative strategies in the presence of malicious agents. *IEEE Transactions on Automatic Control*, 56(7):1495–1508, 2011.

[161] T. A. Severson, B. Croteau, E. J. Rodríguez-Seda, K. Kiriakidis, R. Robucci, and C. Patel. A resilient framework for sensor-based attacks on cyber–physical systems using trust-based consensus and self-triggered control. *Control Engineering Practice*, 101:104509, August 2020.

[162] F. Wen and Z. Wang. Distributed Kalman filtering for robust state estimation over wireless sensor networks under malicious cyber attacks. *Digital Signal Processing*, 78:92–97, July 2018.

[163] M. S. Mahmoud and H. M. Khalid. Distributed Kalman filtering: a bibliographic review. *IET Control Theory & Applications*, 7(4):483–501, 2013.

[164] A. Amini, Z. Zeinaly, A. Mohammadi, and A. Asif. Performance Constrained Distributed Event-triggered Consensus in Multi-agent Systems. In *2019 American Control Conference (ACC)*, pages 1830–1835, July 2019. ISSN: 2378-5861.

[165] S. Hasan, A. Dubey, G. Karsai, and X. Koutsoukos. A game-theoretic approach for power systems defense against dynamic cyber-attacks. *International Journal of Electrical Power & Energy Systems*, 115:105432, February 2020.

[166] L. Huang and Q. Zhu. A dynamic games approach to proactive defense strategies against Advanced Persistent Threats in cyber-physical systems. *Computers & Security*, 89:101660, February 2020.

[167] A. Sanjab and W. Saad. On bounded rationality in cyber-physical systems security: Game-theoretic analysis with application to smart grid protection. In *2016 Joint Workshop on Cyber- Physical Security and Resilience in Smart Grids (CPSR-SG)*, pages 1–6, April 2016.

[168] A. Kanellopoulos and K. G. Vamvoudakis. Non-equilibrium dynamic games and cyber–physical security: A cognitive hierarchy approach. *Systems & Control Letters*, 125:59–66, March 2019.

[169] Q. Zhu and T. Başar. Game-Theoretic Approach to Feedback-Driven Multi-stage Moving Target Defense. In *Decision and Game Theory for Security*, volume 8252, pages 246–263. Springer International Publishing, Cham, 2013.

[170] N. S. V. Rao, C. Y. T. Ma, U. Shah, J. Zhuang, F. He, and D. K. Y. Yau. On resilience of cyber-physical infrastructures using discrete product-form games. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 1451–1458, July 2015.

[171] R. Arghandeh, A. von Meier, L. Mehrmanesh, and L. Mili. On the definition of cyber-physical resilience in power systems. *Renewable and Sustainable Energy Reviews*, 58:1060–1069, May 2016.

[172] I. Linkov and B. D. Trump. *The Science and Practice of Resilience*. Risk, Systems and Decisions. Springer International Publishing, Cham, 2019.

[173] F. Flammini. *Resilience of Cyber-Physical Systems: From Risk Modelling to Threat Counteraction*. 01 2019.

[174] J. M. Rubio Hernan. *Detection of attacks against cyber-physical industrial systems*. Theses, Institut National des Télécommunications, July 2017.

[175] J. D. Taft. *Differentiating Resilience*, pages 147–167. 2022.

[176] N. Catano. Program synthesis for cyber-resilience. *IEEE Transactions on Software Engineering*, pages 1–1, 2022.

[177] D. Williams, W. Hu, J. W. Davidson, J. D. Hiser, J. C. Knight, and A. Nguyen-Tuong. Security through diversity: Leveraging virtual machine technology. *IEEE Security Privacy*, 7(1):26–33, 2009.

[178] G. Ouffoué, F. Zaïdi, A. R. Cavalli, and H. Nghia Nguyen. A Framework for the Attack Tolerance of Cloud Applications Based on Web Services. *Electronics*, 10(1):6, 2020.

[179] M. Abdelmalak and M. Benidris. Enhancing power system operational resilience against wildfires. *IEEE Transactions on Industry Applications*, 58(2):1611–1621, 2022.

[180] D. Ratasich, O. Hoftberger, H. Isakovic, M. Shafique, and R. Grosu. A Self-Healing Framework for Building Resilient Cyber-Physical Systems. In *2017 IEEE 20th International Symposium on Real-Time Distributed Computing (ISORC)*, pages 133–140, Toronto, ON, Canada, May 2017. IEEE.

[181] P. Veríssimo, N. Neves, and M. Correia. Intrusion-Tolerant Architectures: Concepts and Design. In *Architecting Dependable Systems*, pages 3–36, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[182] S. Forrest, A. Somayaji, and D. Ackley. Building diverse computer systems. In *Proceedings of the 6th Workshop on Hot Topics in Operating Systems (HotOS-VI)*, HOTOS '97, pages 67–72, Washington, DC, USA, 1997. IEEE Computer Society.

[183] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, July 1982.

[184] A. D. Fekete. Asymptotically optimal algorithms for approximate agreement. In *Proceedings of the Fifth Annual ACM Symposium on Principles of Distributed Computing*, PODC '86, pages 73–87, New York, NY, USA, 1986. ACM.

[185] H. LeBlanc, H. Zhang, and X. Koutsoukos. Resilient Asymptotic Consensus in Robust Networks. *IEEE Journal on Selected Areas in Communications*, 31(4):766–781, 2013.

[186] A. Mitra and S. Sundaram. Distributed observers for lti systems. *IEEE Transactions on Automatic Control*, 63(11):3689–3704, 2018.

[187] H. J. LeBlanc and X. Koutsoukos. Resilient first-order consensus and weakly stable, higher order synchronization of continuous-time networked multiagent systems. *IEEE Transactions on Control of Network Systems*, 5(3):1219–1231, 2018.

[188] S. Dibaji and H. Ishii. Resilient consensus of second-order agent networks: Asynchronous update rules with delays. *Automatica*, 81:123–132, 2017.

[189] A. Shamir. How to share a secret. *Commun. ACM*, *22*(11):612–613, November 1979.

[190] E. F. Brickell. Some ideal secret sharing schemes. In *Advances in Cryptology — EUROCRYPT '89*, pages 468–475, Berlin, Heidelberg, 1990.

[191] A. Beimel. Secret-sharing schemes: a survey. In *International Conference on Coding and Cryptology*, pages 11–46. Springer, 2011.

[192] A. Abdul-Rahman and S. Hailes. A distributed trust model. In *Proceedings of the 1997 Workshop on New Security Paradigms*, NSPW '97, pages 48–60, New York, NY, USA, 1997. ACM.

[193] A. Jøsang. The right type of trust for distributed systems. In *Proceedings of the 1996 Workshop on New Security Paradigms*, NSPW '96, pages 119–131, New York, NY, USA, 1996. ACM.

[194] M. Ito, A. Saito, and T. Nishizeki. Secret sharing scheme realizing general access structure. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 72(9):56–64, 1989.

[195] J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In *Advances in Cryptology - CRYPTO' 88*, pages 27–35, New York, NY, 1990. Springer New York.

[196] E. F. Brickell. Some ideal secret sharing schemes. In *Advances in Cryptology — EUROCRYPT '89*, pages 468–475, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg.

[197] M. Karchmer and A. Wigderson. On span programs. In *[1993] Proceedings of the Eigth Annual Structure in Complexity Theory Conference*, pages 102–111, 1993.

[198] D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Journal of Web Semantics*, 5(2):58–71, 2007. Software Engineering and the Semantic Web.

[199] E. Bellini, Y. Iraqi, and E. Damiani. Blockchain-based distributed trust and reputation management systems: A survey. *IEEE Access*, 8:21127–21151, 2020.

[200] A. Ilavendhan and K. Saruladha. Comparative study of game theoretic approaches to mitigate network layer attacks in vanets. *ICT Express*, 4(1):46 – 50, 2018.

[201] Z. Ismail, J. Leneutre, D. Bateman, and L. Chen. A methodology to apply a game theoretic model of security risks interdependencies between ict and electric infrastructures. In *Decision and Game Theory for Security*, pages 159–171, Cham, 2016. Springer International Publishing.

[202] C. Kwon, W. Liu, and I. Hwang. Security analysis for cyber-physical systems against stealthy deception attacks. In *2013 American Control Conference*, pages 3344–3349, 2013.

[203] Y. Mo and B. Sinopoli. False data injection attacks in control systems. 2010.

[204] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli. False data injection attacks against state estimation in wireless sensor networks. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5967–5972, Dec 2010.

[205] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson. Revealing stealthy attacks in control systems. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 1806–1813. IEEE, Oct 2012.

[206] R. Smith. A decoupled feedback structure for covertly appropriating networked control systems. *IFAC Proceedings Volumes*, 44(1):90 – 95, 2011. 18th IFAC World Congress.

[207] I. Linkov and A. Kott. Fundamental Concepts of Cyber Resilience: Introduction and Overview. In *Cyber Resilience of Systems and Networks*, Risk, Systems and Decisions, pages 1–25. Springer International Publishing, Cham, 2019.

[208] M. Conti, D. Donadel, and F. Turrin. A Survey on Industrial Control System Testbeds and Datasets for Security Research. *IEEE Communications Surveys & Tutorials*, 23(4):2248–2294, 2021.

[209] G. Gonzalez-Granadillo, J. Rubio-Hernán, and J. Garcia-Alfaro. Towards a security event data taxonomy. In *Risks and Security of Internet and Systems*, pages 29–45, Cham, 2018. Springer International Publishing.