Cyber-resilience and attack tolerance for cyber-physical systems

Mariana Segovia-Ferreira

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2021IPPAS005

Thèse de doctorat

TELECOM SudParis

IP PARIS

Cyber-Resilience and Attack Tolerance for Cyber-Physical Systems

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom SudParis

École doctorale n°626 de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat: Informatique

Thèse présentée et soutenue à Évry, le 20/05/2021, par

MARIANA SEGOVIA-FERREIRA

Composition du Jury :

Nora CUPPENS Professeure, Polytechnique Montréal	Présidente
Yvon KERMARREC Professeur, IMT Atlantique	Rapporteur
Pascal LAFOURCADE Maître de Conférences, Université Clermont Auvergne	Rapporteur
Luca DE CICCO Maître de Conférences, Politecnico di Bari	Examinateur
Urko ZURUTUZA Maître de Conférences, Universidad de Mondragón	Examinateur
Joaquin GARCIA-ALFARO Professeur, Télécom SudParis	Directeur de thèse
Ana Rosa CAVALLI Professeure Emérite, Télécom SudParis	Co-Encadrante
Jose Manuel RUBIO-HERNAN Maître de Conférences, Télécom SudParis	Invité

In memory of my grandmother Maria Luisa.

Acknowledgements

Countless people have contributed to the successful conclusion of this work. As I finally reach the end of this journey, I want to express my gratitude to all of them.

Firstly, I would like to thank my Ph.D. supervisors Prof. Joaquín García-Alfaro, Prof. Ana R. Cavalli, and Prof. José Manuel Rubio-Hernan for their guidance and support, both at the scientific and human level. Their remarks, dedication, and enthusiasm have been essential for achieving this project. Their confidence and trust in me have been vital to carry out this work in a warm and comfortable atmosphere. They are a great team to work with and there are no words to express my gratitude towards them.

I am also deeply grateful to Prof. Javier Baliosian for trusting me and for his encouragement to start this project. This opportunity would not have been possible without his support.

I cannot forget to acknowledge the support from the Cyber CNI Chair of Institut Mines-Télécom. The chair is supported by Airbus Defence and Space, Amossys, EDF, Nokia, BNP Paribas, and the Regional Council of Brittany. In addition, I thank my colleagues and professors from the Chair CNI for their feedback and help.

I also thank Prof. Yvon Kermarrec and Prof. Pascal Lafourcade for reviewing this manuscript, and Prof. Nora Cuppens, Prof. Luca de Cicco, and Prof. Urko Zurutuza for attending my defense.

I want to thank my colleagues from Télécom SudParis for their support, advice, and friendship: Fabien Charmet, Antoine Bernard, Mustafizur Shahid, Keren Saint-Hilaire, Maria Freire-Hermelo, Mohammed El Barbori, Ender Alvarez, Nesrine Kaaniche, Anna Guinet, and Miroslav Setkic.

Acknowledgements

Likewise, I thank Sandra Gchweinder, Véronique Guy, and Marlène Khenoussi for their support and help with the administrative tasks. Also, thanks to my French teachers Prof. Sophie Sousa and Prof. Nicoline Lagel for their help to learn this beautiful language.

I also express my gratitude to Diego Rivera, David Pàmies-Estrems, and Pamela Carvallo for their warm welcome when I arrived in France.

Finally, I am extremely grateful to my family for their support during this adventure, for their help to improve myself every day and their encouragement to do what makes me happy.

Paris, 20 May 2021 T.D.

Abstract

This thesis investigates the resilience of Cyber-Physical Systems (CPS). CPS integrate computation and networking resources to control a physical process often related to critical infrastructures, such as energy distribution, health care, industrial process control, among others. The adoption of new communication capabilities comes at the cost of introducing new security threats that need to be properly handled. An attack may have dangerous consequences in the physical world putting in danger the safety of the people, the environment and the controlled physical processes. For this reason, cyber-resilience is a fundamental property to ensure attack tolerance, *i.e.*, the system must maintain the correct operation of a set of crucial functionalities despite ongoing adversarial misbehavior. For that, threats must be addressed at cyber and physical domains at the same time.

We aboard the system reaction creating a synergy between control-theoretic information and cybersecurity methods to absorb and recover from the threat. We propose two approaches using different paradigms. The first one is based on a detection and reaction strategy to attenuate cyber-physical attacks driven by reflective programmable networking to take control of adversarial actions. The mechanism builds upon the concept of software reflection and programmable networking. The second approach proposes a resilient-by-design strategy. The approach is based on a Moving Target Defense paradigm, driven by a linear switching of state-space matrices, and applied at both the physical and network layers of a CPS. We provide a step-by-step procedure that takes a transfer function, representing the dynamics of the physical process and we show that the final system maintains stability. As a result, we obtain a resilient CPS design structured using a topology of decentralized controllers.

Also, we present metrics to quantify the cyber-resilience level of a system based on the design, structure, stability, and performance under the attack. The metrics provide reference points to evaluate whether the system is better prepared to face adversaries. This way, it is possible to quantify the ability to recover from an adversary using its mathematical model. We evaluated the proposed approaches using numerical simulations and obtained promising results. Finally, we identified several possibilities for future research perspectives to improve existing knowledge in the field.

Résumé

Cette thèse porte sur la résilience des systèmes cyber-physiques qui intègrent des ressources de calcul et de réseau pour contrôler un processus physique lié à des infrastructures critiques. L'utilisation de l'acquisition et le traitement des données sur un système de contrôle en réseau permet d'exécuter des tâches automatiquement et à distance.

L'adoption de nouvelles capacités de communication se fait au prix de l'introduction de nouvelles menaces pour la sécurité qui doivent être traitées correctement. Une attaque peut avoir des conséquences dangereuses dans le monde physique et mettre en danger la sécurité des personnes, de l'environnement et des processus physiques contrôlés. Pour cette raison, la cyber-résistance est une propriété fondamentale pour assurer la tolérance aux attaques. Le système doit maintenir le bon fonctionnement d'un ensemble de fonctionnalités cruciales malgré les comportements malveillants. Pour cela, les menaces doivent être traitées simultanément dans les domaines cyber et physique. Les cyberattaques ont une capacité limité de produire des dommages dans les systèmes cyber-physiques. Pour cette raison, nous considérons de nouveaux adversaires, appelés adversaires cyber-physiques, qui utilisent des stratégies de contrôle théorique pour causer des dommages physiques via le système informatique. Les attaques cyber-physiques peuvent être difficiles à détecter. À ce titre, la résilience est particulièrement pertinente et le développement de systèmes cyber-physiques capables de survivre à une attaque en toute sécurité est un défi actuel.

L'objectif principal de cette thèse est de développer une approche de résilience pour les systèmes cyber-physiques qui permet de poursuivre le fonctionnement du système de manière sûre, même en cas d'attaque. Nous abordons la réaction du système en créant une synergie entre l'information de la théorie du contrôle et les méthodes de cybersécurité pour absorber la menace et remettre le système dans son état correcte. Nous proposons deux approches utilisant des paradigmes différents. La première propose une stratégie de détection et de réaction visant à atténuer les attaques cyber-physiques, qui s'appuie sur des actions des *programmable reflective networks* pour prendre le contrôle des actions adverses. Le mécanisme s'appuie sur le concept de *software reflection* et les réseaux programmables qui résulte satisfait à l'auto-remédiation dans les situations d'adversité et le système continue de fonctionner de manière autonome.

La seconde approche propose une stratégie de résilience par conception. L'approche est basée sur un paradigme de *moving target defense*, piloté par une commutation linéaire des matrices d'état-espace, et appliqué à la fois aux couches physique et réseau d'un système cyber-physique. L'objectif était de concevoir un système qui, sans recourir à un mécanisme de détection, avait la capacité de restaurer les fonctions du système en transformant les connaissances d'attaquant en inutiles. Nous fournissons une procédure étape par étape qui prend une fonction de transfert, représentant la dynamique du processus physique et nous montrons que le système final maintient la stabilité. En conséquence, nous obtenons une conception de système résiliente structurée selon une topologie de contrôleurs décentralisés.

Nous présentons également des mesures pour quantifier le niveau de cyber-résilience d'un système basé sur la conception, la structure, la stabilité et la performance pendant l'attaque. Les mesures fournissent des points de référence pour évaluer si le système est mieux préparé pour faire face aux adversaires. Ainsi, il est possible de quantifier la capacité de récupération d'un adversaire en utilisant son modèle mathématique.

Nous avons évalué les approches proposées avec des simulations numériques et nous avons obtenu des résultats prometteurs. Enfin, nous avons identifié plusieurs possibilités de perspectives de recherche futures pour améliorer les connaissances existantes dans le domaine.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

Symbol	Description
$ARMAX$	Autoregressive Moving Average Exogeneous.
ARX	Autoregressive Exogeneous.
CPS	Cyber-Physical Systems.
DoS	Denial Of Service.
HMI	Human Machine Interfaces.
ICS	Industrial Control Systems.
IT	Information Technology.
LTI	Linear Time-Invariant.
LTV	Linear Time-Variant.
$MIMO$	Multiple Inputs Multiple Outputs.
MTD	Moving Target Defense
MTU	Master Terminal Units.
NCS	Networked Control Systems.
OT	Operational Technology.
PLC	Programmable Logic Controllers.
RTU	Remote Terminal Units.
$SCADA$	Supervisory Control and Data Acquisition technology.
$SISO$	Single Inputs Single Outputs.

Notations

Symbol	Description
Γ and Ω	Ponderation matrices.
$\hat{x}_{t\|t-1}$	Vector of estimated state variables before applying the rectification.
$\hat{x}_t$	Vector of estimated state variables after applying the rectification.
$\mathcal{P}$	Co-variance of the i.i.d. Gaussian signal.
A	State matrix.
B	Input matrix.
C	Output matrix.
J	Quadratic cost.
K_f	Kalman gain.
L	Feedback grain.
$P_{t\|t-1}$	A priori error covariance.
P_t	A posteriori error covariance.
Q	Process noise variance.
R	Output noise variance.
r_t	Residue.
S	Riccati equation solution.
u_t	Control input vector.
u'_t	Control inputs injected by the adversary.
u^*_t	Optimal control input vector.
v_t	Output noise.
w_t	Process noise.
x_t	Vector of state variables.
y_t	Vector of the sensors measurements.
y'_t	Measurements injected by the adversary.

1 Introduction

1.1 Resilience in Cyber-Physical Systems

Traditionally, the design of industrial systems was based on an isolation model, where the control of the Operational Technology (OT) was separated from the Information Technology (IT). Today, both OT and IT are integrated since the physical processes are controlled by Cyber-Physical Systems (CPS). CPS integrate modern computation and networking resources into traditional physical environments. They have emerged mainly on the Industrial Control System (ICS) domain using data acquisition and processing on a Networked Control System (NCS) [1] to execute industrial tasks automatically and remotely [2].

Such integration has several advantages, for example, low maintenance costs, high reliability, and more flexibility, efficiency, and effectiveness to control the physical process [3]. The use of the technology to build a new generation of CPS play an important role in current critical national-wide infrastructures, such as electrical transmission, energy distribution, manufacturing, supply chain, waste recycling, public transportation, health care, industrial process control, water infrastructure, and several others [1, 4].

CPS use sensor measurements to get information about the physical process, then control processing units analyze it and make decisions that are performed by system actuators, e.g., to maintain the stability of the physical processes. Ensuring the control of such data exchanges is a challenging problem that requires a combination of both network and industrial control security. CPS are designed to recover from process faults and failures with a limited impact on the system operations. However, CPS can be disrupted by cyber-physical attacks [5, 6].

Cyber-physical attacks can manifest significant physical effects [7]. For example, they may put at risk human safety, cause harm in natural environments, interrupt industrial process continuity, and violate environmental regulation. Hence, they can lead to large

economic losses, generate legal problems, and damage the reputation of the affected organizations [8]. In addition, cyber-physical attacks may be hard to detect [9, 10]. For this reason, resilience is especially relevant and developing CPS that can survive an attack safely is a current challenge.

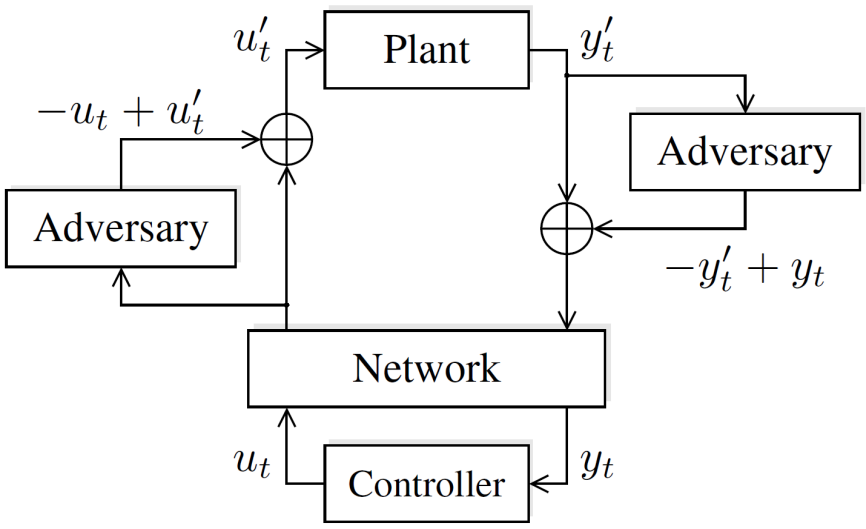


Figure 1.1 – Cyber-physical industrial attack. Variables u_t and y_t are the correct input and output vectors of the system. Variables u'_t and y'_t represent the attack vectors.

A CPS normally is composed of many control loops and the controllers execute a model that is created according to the physical process dynamics. Figure 1.1 shows how a cyber-physical adversary attacks one control loop using a block diagram representation, which is the typical control theory community representation. The $\bigoplus$ symbol represents a *summing junction*, *i.e.*, the sum of input signals.

To take control of the physical process, the adversary sends a malicious command u'_t to the Plant[1] that will be executed by the actuators. After that, to deceive the controller and go unnoticed, the adversary modifies the sensors' readings y'_t to inject a measurement value y_t. This value is created using the controllers' system model. This way, the values correspond to normal operation, *i.e.*, they are correlated with the correct command u_t that was previously modified by the adversary. As a result, from the controller point of view, the physical process execution is correct, since it sends a command u_t and it receives an answer y_t which is correctly verified by the model. Cyber-physical attacks affect the system state and disrupt normal operation conditions creating an attack.

This dissertation focuses on resilience techniques to build CPS tolerant to cyber-physical attacks. We consider that the system is a combination of cyber and physical components working together under discrete and continuous industrial domains [11]. We devote our

[1] Often referred to as *System* in the related literature.

work to protection techniques addressing networked control systems, *i.e.*, a subset of cyber-physical systems dedicated to industrial control processes, usually performing critical functions.

In this document, we use the words *resilience* and *cyber-resilience* indifferently. However, both terms are not exactly the same. Resilience describes a system capable of preparing, absorbing, recovering, and adapting to adverse effects [12]. It is a concept used across many scientific domains aside from computer science. As a consequence, it involves safety but does not necessarily require cyber strategies to achieve it. On the contrary, cyber-resilience focuses on the use of cyber components and strategies to build an attack tolerant system, but it does not imply safety. In this dissertation, we focus on resilience considering both aspects, cyber-resilience for the IT components and also the safety of the physical process and its surrounding environment. Some literature uses the concept of *cyber-physical resilience* to name the combination of both concepts. However, we believe that cyber-physical resilience is the same as resilience. For this reason, we prefer the term resilience and sometimes we emphasize the use of cyber strategies to achieve resilience using the term *cyber-resilience*.

1.2 Motivation

Ensuring safety using only information security tools is not enough in CPS. Cybersecurity approaches do not cover all the possible vulnerabilities in the cyber components. For example, because mechanisms to protect specific vulnerabilities may not exist or be too expensive to implement for low probability events. Even when the approach is implemented, it is not free of false negatives.

As pointed out in [13], large research efforts have focused on intrusion detection for CPS, but there is little less discussion about what to do after the intrusion is detected, *i.e.*, in reaction 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 under the presence of successful attacks against the system. The resulting systems should satisfy high availability requirements to guarantee the execution of the critical tasks. It should be able to guarantee that the whole system remains operational even in the presence of attacks and if that means to work under graceful degradation modes. As a result, cybersecurity approaches should be complemented with secure control theory which provides attack models and a description of the interaction between the physical world and the control system. It provides a better understanding of the attacks' consequences, development of new detection methods, response mechanisms, and architectures, that make the control systems more resilient to possible attacks and failures.

We focus on availability and integrity attacks since they are the main security issue in CPS [14]. Pure cyber attacks have limited damage to the system [15]. For this reason, we consider new adversaries that use control-theoretic strategies to cause physical damage. In particular, cyber-physical integrity attacks can rapidly move the system to unsafe states as showed in Figure 1.1. Also, cyber-physical DoS attacks can be launched using integrity attacks to cause significant damage. In this case, the integrity of the messages is compromised with two objectives. First, to disrupt the communication between the controller and the plant, generating a loss of the system 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 unavailability of the system to the authorized users in order 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.

1.3 Objectives

In this dissertation, we investigate the resilience of Cyber-Physical Systems. We establish the following objectives.

- **Objective 1.1:** Analyze the threats in CPS with a focus on cyber-physical adversaries. Identify the existing limitations in cybersecurity solutions and future research perspectives.
- **Objective 1.2:** Study the existing mitigation and resilience techniques to achieve attack tolerance. Understand how to build resilient systems and whether these techniques are appropriate for the new challenges created by the cyber-physical adversaries reviewed in the previous objective.
- **Objective 1.3:** Create new resilience techniques to preserve the safe operation even under attack. We consider the control-theoretic perspective of the problem. Also, our objective is to investigate software reflection as a promising technique for attack mitigation and how to build resilience-by-design systems that are able to adapt themselves and recover.
- **Objective 1.4:** Evaluate the resilience of a system and measure the improvement generated by a resilience approach.
- **Objective 1.5:** Validate the proposed mechanisms via simulation.

1.4 Contributions

The main objective of this thesis is to develop a resilience approach for Cyber-Physical Systems that allows continuing the system operation in a safe manner even under attack.

For this purpose, we focus on two different approaches. The first one is based on detection and reaction. The second one proposes a resilient-by-design approach based on moving target defense techniques.

Most of the cyber-physical security solutions focus on either cyber adversaries or physical adversaries, but not both at the same time. Our proposed strategies combine cybersecurity and control-theoretic approaches to build a solution that contemplates the cyber and the physical components of a CPS. Control-theory and cybersecurity are research areas that provide significant contributions from different perspectives to solve security issues in CPS. They are complementary and working together can provide more efficient and effective solutions.

We devote our work to resilience techniques addressing cyber-physical adversaries damaging CPS dedicated to industrial control processes. We aim at creating innovative solutions for CPS safety and security by reducing the gap between control-theoretic techniques and cybersecurity approaches. Our solutions achieve attack tolerance and graceful degradation assuming a combination of cyber and physical components working together [11]. We focus specifically on closed-loop networked control systems. This means that the control system handles dynamic feedback to maintain a correlation among the different variables on the industrial system.

We summarize the complete list of contributions in this dissertation as follows.

- Analysis of the literature related to CPS and cyber-physical attacks (Objective 1.1).
- A systematic bibliographic review on the resilience definition and security approaches focusing on detection-reaction strategies and cyber-resilience to shed some light on challenges, advances, and open research questions in this area. In particular, the review tackles the topic from a control-oriented perspective and cybersecurity point of view (Objective 1.2).
- A detection-reaction approach based on software reflection and programmable networks that provide cyber-physical attack attenuation (Objective 1.3).
- A resilient moving-target mechanism to tolerate different cyber adversary models. The approach is based on switched control and network reconfiguration (Objective 1.3).
- Evaluation metrics to compare and analyze the system resilience (Objective 1.4).
- Construction of numeric simulation to validate the new mechanisms (Objective 1.5).

1.5 Publications

Part of the work covered in this dissertation has already been published in different international peer-reviewed journals or conferences. We list the scientific publications that are directly related to the work in this thesis.

Journal papers

- M. Segovia, J. Rubio-Hernan, A.R. Cavalli, J. Garcia-Alfaro, *Cyber-Resilience - A Systematic Survey of Resilience Techniques for Cyber-Physical Systems*, *[Under Evaluation]*.
- M. Segovia, J. Rubio-Hernan, A.R. Cavalli, J. Garcia-Alfaro, *Switched-Based Resilient Control of Cyber-Physical Systems*, in IEEE Access, vol. 8, pp. 212194-212208, 2020, doi: 10.1109/ACCESS.2020.3039879.

Conference papers

- M. Segovia, J. Rubio-Hernan, A.R. Cavalli, J. Garcia-Alfaro, *Switched-based Control Testbed to Assure Cyber-Physical Resilience by Design*, *[Under Evaluation]*.
- M. Segovia, J. Rubio-Hernan, A.R. Cavalli, J. Garcia-Alfaro, *Cyber-Resilience Evaluation of Cyber-Physical Systems*, 19th IEEE International Symposium on Network Computing and Applications (NCA 2020), pp. 1-8, Boston, USA, November 2020, doi: 10.1109/NCA51143.2020.9306741.
- M. Segovia, A.R. Cavalli, N. Cuppens, J. Rubio-Hernan, J. Garcia-Alfaro, *Reflective Mitigation of Cyber-Physical Attacks*, 5th Workshop on the Security of Industrial Control Systems & of Cyber-Physical Systems (CyberICPS 2019), 24th European Symposium on Research in Computer Security (ESORICS 2019), pp.19-34, Springer, Luxembourg, September 2019, doi: 10.1007/978-3-030-42048-2_2.
- M. Segovia, A.R. Cavalli, N. Cuppens, J. Garcia-Alfaro, *A Study on Mitigation Techniques for SCADA-driven Cyber-Physical Systems*, Foundations and Practice of Security (FPS 2018), pp. 257-264, Springer, LNCS 11358, Montreal, Canada, November 2018, doi: 10.1007/978-3-030-18419-3_17.

1.6 Organization

This dissertation is divided into six chapters. To facilitate the reading, we included in every chapter the corresponding experimental part. The rest of the document is structured as follows.

- **Chapter 2, State of The Art.** This chapter provides the background of the dissertation, including a systematic review of related work. It contributes to Objectives 1.1 and 1.2.
- **Chapter 3, Detection-Reaction Paradigm.** This chapter develops our analysis of the reaction solutions to mitigate cyber-physical attacks. It contributes to Objectives 1.3 and 1.5.
- **Chapter 4, Moving-Target Paradigm.** This chapter contributes with a resilience-by-design approach that designs a Cyber-Physical System as a Switched Control System using a Moving Target Defense approach. It contributes to Objectives 1.3 and 1.5.
- **Chapter 5, Evaluation.** This chapter presents resilience metrics to evaluate the system resilience. It contributes to Objectives 1.4 and 1.5.
- **Chapter 6, Conclusion and Future Research.** This chapter concludes the dissertation and provides some future research lines.

2 State of The Art

2.1 Cyber-Physical Systems

Cyber-Physical Systems (CPS), also called Networked Control Systems (NCS), are 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 components of the cyber layer control the behavior of the physical layer and the feedback of the physical layer affects the decisions of the cyber layer. The CPS becomes easier to automate at the cost of increasing the interaction between physical and cyber layers [2]. However, as a consequence, they get more vulnerable to attacks. Malicious actions in these systems[1] 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 [16].

The cyber layer uses security mechanisms similar to the mechanisms for traditional information systems. The physical layer has different requirements and can be controlled in different ways [17]. For example, considering the model of the involved physical process. For that reason, it is important to see the system as a whole, also thinking about the information flows to and from the cyber layer and the interconnected networks to determine how to protect them. The integration between layers is an important point to evaluate and determine how to protect the information flow [7].

Different surveys have addressed the special requirements and design considerations of CPS. For example, Ge *et al.* [1] review methodologies to design distributed networked control systems. The paper presents an overview of the possible system configurations,

[1] Often referred as *plant* in the related literature.

challenging issues regarding communication, computation, and control aspects; and methodologies to design distributed networked control systems. For example, based on undirected and directed graphs, fixed and time-varying topologies, as well as time-triggered and event-triggered mechanisms.

Do *et al.* [7] analyze the architecture, vulnerabilities, attack points, and famous attacks in CPS with a focus on SCADA[2] technologies. In [18], Molina *et al.* review the state of the art of CPS applications, network requirements and the application of SDN approaches for mission-critical CPS. Also, in [2], Zhang *et al.* provide an overview on the theoretical development of Networked Control Systems using sampled-data control, networked control, and event-triggered control.

Lun *et al.* [4] survey the latest research trends about cyber-physical systems security analyzing the most used designs, architectures, testbeds, and attack types. Also, Giraldo *et al.* [19] survey security and privacy in CPS. They provide a taxonomy based on CPS application domain, indicating whether the proposal contains cyber or physical security measures, and if it is about prevention, detection, or response to attacks.

A summary of the surveys that have been conducted about CPS, defense-reaction cybersecurity mechanism, and cyber-resilience can be found in Table 2.1. To systematize the literature review presented in this chapter, we pursued a strategy to search and accumulate the relevant publications. For that, we used keyword search to make the first selection of potentially relevant scientific publications. We considered databases such as Google Scholar, IEEE Xplore, DBLP, and Science Direct to collect the publications. Articles were filtered with the keywords *resilience*, *detection*, *attenuation*, *mitigation*, *resilience metrics*, *resilience measure*, *resilience evaluation*, and *cyber-physical system*. The most relevant literature was filtered according to their titles and abstracts. Finally, we included other publications using cross-references from the first dataset.

The collected data was later processed based on inclusion and exclusion criteria. The inclusion criteria for this study were based on the following conditions. The proposal has to refer to computer science and cyber-security field. It should focus on technical mechanisms to make the system recover itself with little or no human interaction. We are not interested in organizational frameworks or manual procedures. Also, the proposal should be useful for CPS.

We exclude the literature written in another language than English or with full content access denied. We did not include literature that does not give any guarantees about the feasibility of a potential implementation. Also, the quality of the papers was assessed considering whether the fundamental concepts and their related properties are adequately described.

[2]Supervisory Control and Data Acquisition (SCADA) is a technology to monitor industrial and critical infrastructures based on CPS. It specifies, for example, the control system architecture, the networked data communications and high-level supervision of the physical process.

Survey	Year	General CPS	CPS Attacks	Resilience Definition	Detection-Reaction	Resilience Approaches	Resilience Evaluation
Cholda *et al.* [20]	2009					✓	
Teixeira *et al.* [5, 6]	2012		✓				
Linkov *et al.* [21]	2013						✓
Cheminod *et al.* [22]	2013	✓			✓		
Bodeau *et al.* [23]	2013					✓	
Arghandeh *et al.* [24]	2016			✓			
Zhang *et al.* [2]	2016	✓					
Hosseini *et al.* [25]	2016			✓			✓
Do *et al.* [7]	2017		✓		✓		
Molina and Jacob [18]	2017	✓					
Ge *et al.* [1]	2017	✓					
Giraldo *et al.* [19]	2017	✓			✓		
Humayed *et al.* [26]	2017	✓	✓				
Alguliyev *et al.* [8]	2018	✓	✓				
Gholami *et al.* [27]	2018			✓			✓
Ding *et al.* [28]	2018				✓	✓	
Jain *et al.* [29]	2018						✓
Mahmoud *et al.* [30]	2019		✓		✓		
Sanchez *et al.* [16]	2019		✓				
Lun *et al.* [4]	2019	✓	✓				
Linkov and Trump [31]	2019	✓					✓
Bhusal *et al.* [32]	2020			✓			✓
Sepúlveda Estay *et al.* [33]	2020		✓		✓		
Weerakkody *et al.* [17]	2020	✓	✓		✓	✓	
Yaacoub *et al.* [34]	2020	✓	✓		✓		
Mohebbi *et al.* [35]	2020			✓			✓
Mishra *et al.* [36]	2020			✓		✓	
Clédel *et al.* [37]	2020			✓			✓

Table 2.1 – Surveys related to cyber-resilience in CPS.

2.1.1 Architecture

A typical CPS architecture is composed of a multitude of physically and functionally heterogeneous components that work in a distributed networked-based manner [1]. An industrial CPS architecture is depicted in Figure 2.1. This architecture may have one or more networks for the different physical processes situated normally in remote locations that are monitored and controlled from a central control room located in the control LAN [38].

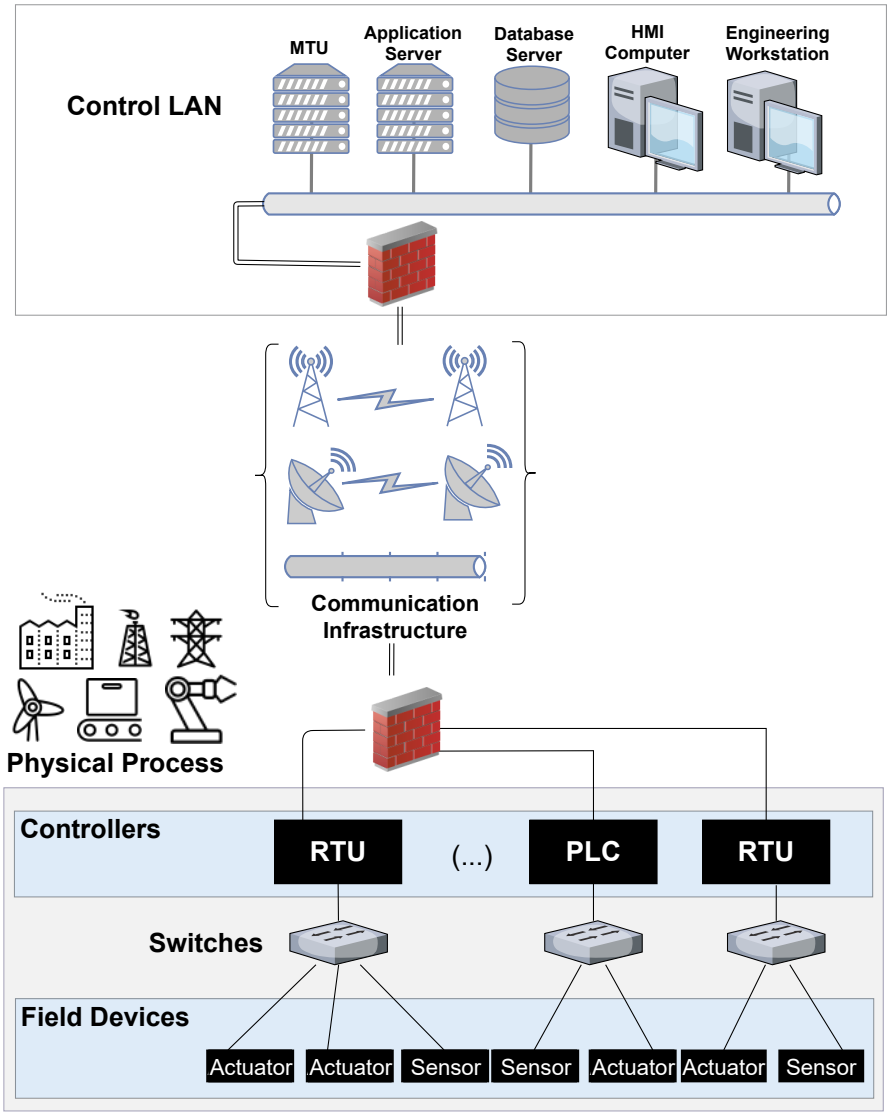


Figure 2.1 – CPS Architecture.

The remote locations are sites equipped with sensors and actuators regulated by controllers, such as Remote Terminal Units (RTU) or Programmable Logic Controllers (PLC) [7]. The sensors are monitoring devices responsible for retrieving measurements

related to a specific physical phenomenon, such as temperature, pressure, flow, or speed, and feed them to the controller. The actuators are control devices, such as valves, motors, compressors, or pumps, in charge of performing actions that are needed to correct the dynamics of the system [39]. The collected information from the field is transferred to the control LAN, where a Master Terminal Unit (MTU) processes and stores the information from the controllers and a Human Machine Interface (HMI) displays the information for human monitoring functions. The whole physical process operation is based on control commands from the control LAN and the sensor measurements from field devices.

RTUs are standalone data acquisition and control units that monitor and control the industrial equipment at the field location. Their tasks are to control and acquire data from process equipment, and to communicate the collected data to the MTU located in the control LAN. Modern RTUs may also communicate between them. PLCs are small industrial microprocessor-based computers. The most significant differences concerning an RTU are in size and capability. An RTU has more inputs and outputs than a PLC, and much more local processing power. For example, to post-process the collected data before generating alerts toward the MTU via the HMI. In contrast, PLCs are often represented by pervasive sensors with communication capabilities. PLCs have two main advantages over traditional RTUs. First, they are general-purpose devices enforcing a large variety of functions, and second, they are physically compact.

The components in a CPS are connected through a communication network. The use of communication networks adds more flexibility to the system and reduces the implementation cost of new installations. The communication protocols used in traditional control systems are required to comply with the constraints imposed by industrial standards (e.g., to cover regulation such as delays and faults). Some of the commonly used protocols are Modbus, Profinet, DNP3, IEC-60870-5-104, and EtherNet/IP [40]. Some of the protocols (e.g., Modbus, DNP3, and Profinet), are not designed to provide security from a traditional information perspective. Current systems use these protocols over TCP/IP or UDP/IP communications (e.g., Modbus over TCP, DNP3 over TCP or UDP, and Profinet over TCP). These combinations can provide some security mechanism at the transport or network layers. This is not enough to ensure control-data protection.

2.1.2 Control Theory Model

System Dynamics

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 [41]. 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. The system models to control a physical process are not unique. On the contrary, an interesting property is that a system has many equivalent representations that can be obtained, for example, using matrix factorization techniques.

Figure 2.2 shows the networked feedback control of a CPS. The *plant* 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* in the diagram, is used by the controller to create the *control input* in order to rectify, if necessary, the physical states using the actuators.

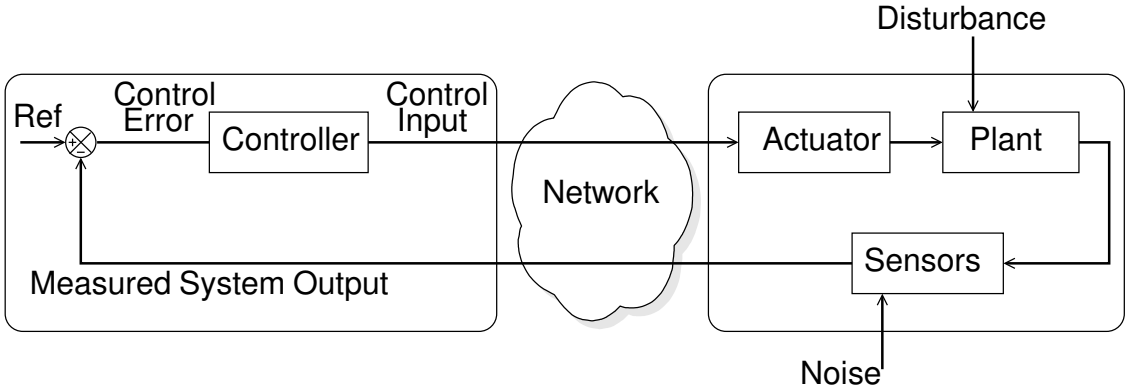


Figure 2.2 – Diagram of a networked feedback control system.

Physical Model

How to obtain the model used in the feedback controller is a very 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 [42–45]; and also to create feedback control [46–48]. 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. For this reason, it is possible to transform one representation into the other and vice-versa.

Normally, a CPS design process starts with the transfer function since it is the most direct form starting from the differential equations of the 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 showed in Equation (2.1) 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.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\sum_{i=0}^{m} b_i s^{m-i}}{\sum_{i=0}^{n} a_i s^{n-i}} \tag{2.1}$$

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 [49], it is possible to transform the transfer function into a state-space model by expressing the differential equations into matrices forms, cf. Equation (2.2) as follows:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + v_k \end{aligned} \tag{2.2}$$

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.* $w_k \sim N(0, Q)$. In this dissertation, we use the discrete-time model since the controllers are normally implemented in 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.

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 [50]. 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_t such that a quadratic cost J, that is a function of both the state x_t and the control input u_t, is minimized:

$$J = \lim_{n\to\infty} E\left[\frac{1}{n}\sum_{i=0}^{n-1}(x_i^T \Gamma x_i + u_i^T \Omega u_i)\right] \tag{2.3}$$

where Γ and Ω represent positive definite cost matrices [51].

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. the *Kalman filter*-based LQE, using the noisy measurements, produces an optimal state estimation $\hat{x}_t$ of the state x;
2. the LQR, based on the state estimation $\hat{x}_t$, provides the control law u_t that solves the problem (cf. Equation (2.3)).

A Kalman filter can estimate the state as follows:

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

$$\hat{x}_{t|t-1} = A\hat{x}_{t-1} + Bu_{t-1}$$

$$P_{t|t-1} = AP_{t-1}A^T + Q$$

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

$$K_t = (P_{t|t-1}C^T)(CP_{t|t-1}C^T + R)^{-1}$$

$$\hat{x}_t = \hat{x}_{t|t-1} + K_t(y_t - C\hat{x}_{t|t-1})$$

$$P_t = (I - K_tC)P_{t|t-1}$$

where K_t and P_t 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_t provided by the LQR is a linear controller:

$$u_t = L\hat{x}_t \tag{2.4}$$

where L denotes the feedback gain of the LQR that minimizes the control cost (cf. Equation (2.3)), which is defined as follows [52, 53]:

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

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

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

Hence, we assume the modeling of cyber-physical systems as *Linear Time-Invariant* (LTI) discrete systems, whose feedback control mechanisms are regulated by LQG controllers.

Linear and Non-Linear Models

The previously presented physical model is a linear system. However, most of the models in a real physical process present some nonlinearities. Nonlinear systems may show complex effects and there is no general method for designing this type of controller [54] although some methods and techniques applicable to particular classes of nonlinear control problems are presented in the literature [55].

Linear controllers are more predictable, simpler and in most cases, they provide adequate control performance. For this reason, linear approximations are used instead of nonlinear controllers. Many linearization techniques transform the original system model into an equivalent model of a simpler form that allows using linear control techniques to analyze the nonlinear problem [56]. Some linearization techniques are, for example, carleman linearization [57, 58], lie series [59], feedback linearization [59], linearization via changes of variables [59, 60], among many others.

In this dissertation, we assume linear models. To apply our proposed techniques, we assume that the system model has been linearized previously. This is not a limitation of our approach, because of the lack of a general method to implement nonlinear models, they are normally linearized in order to be able to design the controllers.

Distributed Architecture

Most industry control systems are Multiple-Input-Multiple-Output (MIMO) systems [61], *i.e.*, the process consists of several measurement and control signals. There are often dependencies, called *couplings*, between these variables [62]. When designing the controllers for MIMO systems, it is necessary to partition the given problem into manageable subproblems. As a result, the overall plant is no longer controlled by a single MIMO controller but by several independent controllers which altogether represent a decentralized controller [63]. A decentralized control consists of a set of independent

controllers, typically Single-Input-Single-Output (SISO) control loops, *i.e.*, controllers that receive one input and return one control signal as output.

Centralized controllers have better performance. However, decentralized SISO control is often preferred because it has several advantages compared to MIMO design. For example, as pointed out in [64, 65], the operating personnel can restructure the control system by bringing subsystems in and out of service individually. It is easier to implement and the simplified design facilitates understanding and changing it. Also, the tuning is simplified and it is more robust with respect to model errors which leads to a better failure tolerance. For instance, if an actuator or sensor fails, only the individual subsystem involved is affected and only this subsystem needs to be taken out of service with no changes to other parts.

For the above-mentioned reasons, a decentralized controller design is preferred in practical multi-variable process control. The design of such a control system introduces the *pairing problem* which is concerned with defining the system structure, *i.e.*, which of the available plant inputs is to be used to control each of the plant outputs [64]. For a fully non-interacting plant, the choice is obvious. However, in any practical problem, there are interactions in the plant. This means that even if the control system is decentralized, subsystems of the closed-loop design are not independent of each other.

The decoupling problem analyzes how to design systems where dependent variables are implemented independently. When the process interactions are significant, the choice of a control system structure is far from trivial and has been the subject of much research [61, 62, 65–70]. For an $n \times n$ plant, there are n factorial, *i.e.*, $n!$ possible SISO pairings to chose and each controller has only available a part of the overall a priori and a posteriori information. Most of the decoupling control synthesis strategies firstly compute a decoupling precompensator (called decoupler) to turn the resultant system into a more nearly diagonal transfer matrix and then compute the multi-loop controllers.

These system properties allow having many possible implementations for the same system. We use it to implement an approach to improve the system resilience in Chapter 4.

Switched Systems

Switching control techniques are based on changing between different controllers in an adaptive context while achieving stability. Switched systems with all the subsystems described by linear differential equations are called switched linear systems.

Many systems encountered in practice exhibit switching between several subsystems that are dependent on various environmental factors. For example, in a car, the first, second, and third gears experiment different dynamics that can be modeled using different controller models. In addition, switched systems have been also used to control

systems under packet dropout effects and delays [71–73]. By introducing a switching function related to the variation in network-induced delays, the closed-loop is modeled as a time-delay switched system with two switching modes that have different controller gains. In the case of dropout, the idea is similar, both networks from the sensor to the controller and from the controller to the actuator are modeled as two switches indicating that a data packet is dropped out or not [2].

In particular, switched systems have gained major attention in the control theory community in the last years since there exist unstable processes that are not possible to control with just one model, but it is possible to design switched controllers for stabilizing it with piece-wise signals [74, 75].

All systems managed by control-theoretic models require a stability analysis to determine how the system will behave. Indeed, there are many criteria to analyze this issue, such as Lyapunov, root-locus, Routh-Hurwitz, Bode, or Nyquist methods [49]. Switched systems create a new challenge. Even when all the individual models are stable the resulting piece-wise system may be not stable. For this reason, the control theory research community has studied how to ensure the stability of switched control under different systems' characteristics [76, 77]. For example, using techniques for arbitrary switching, such as Common Quadratic Lyapunov Functions and Switched Quadratic Lyapunov Functions [77–80]; or restricted switching, such as slow switching or dwell-time switching, Multiple Lyapunov Functions and Piece-wise Quadratic Lyapunov Functions [74, 77, 80–84]. In an arbitrary switching, there are no restrictions to chose when to change controllers. However, a restricted switching may arise from natural physical constraints of the system. For example, coming back to the car scenario, there is a particular switching sequence to be followed; from the first gear to the second gear, then the third gear, etc. Another arbitrary switching condition may be, for instance, a certain bound on the time interval between two successive switches.

2.2 Cyber-Physical Attacks

Control systems use safety mechanisms to handle failures and avoid accidents. Nevertheless, these control mechanisms cannot detect intentional malicious actions. In this section, we present the definition of cyber-physical adversaries and some known attack examples. Also, we present a classification for cyber-physical adversaries.

2.2.1 Adversary Model

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. In particular, the security of industrial CPS is drawing great attention after

the Stuxnet malware [85, 86] that damaged a nuclear plant and uncovered to the general audience the danger of successful security attacks carried out against such systems. There are several examples of breaches that cause physical damages. For example, the well-known Ukraine attack [87] 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 [88]. 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 on to local waterways.

All these attacks were caused by cyber-physical adversaries, which are different from physical adversaries and cyber adversaries. In the sequel, we describe each of them considering the definitions in [40].

Physical Adversary. This adversary has physical access to the CPS infrastructure and can damage it by performing manual or 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 plant or affect sensor measurements by modifying their local surroundings [6, 17].

Cyber Adversary. The next level adversary is the cyber adversary which performs traditional cybersecurity attacks such as man-in-the-middle, buffer overflow, shell exploits, or others. This adversary has only knowledge about the software and network resources. Because of that, the attack can be easily detected by control-theoretic fault detection. Humayed *et al.* [26] systematize existing CPS security research analyzing the taxonomy of threats, vulnerabilities and attacks from the CPS components perspective, with a special focus on cyber components. In addition, Alguliyev *et al.* [8] analyze and classify existing research on the security of CPS focusing on cyber adversaries. They also present the main difficulties and solutions in the estimation of the consequences of cyber-attacks, attacks modeling, detection, and the development of security architecture.

Cyber-Physical Adversary. A cyber-physical adversary is the most advanced of the adversaries. A cyber-physical attack can cause tangible damage to physical components, for instance, adding disturbances to a physical process via exploitation of vulnerabilities in computing and networking resources of the systems. The cyber-physical adversary is a combination of the two previous adversaries [14]. First, the adversary uses a cyber attack to gain position into the system from a remote location and then, learns about the physical model to generate an attack with physical consequences but without being physically placed in the CPS physical location. For this reason, this adversary represents a danger and a research challenge. Moreover, it can be hard to detect them. It should be also noted that these attacks may often be confused with faults.

2.2.2 Taxonomy of Attacks

Different classifications have been proposed for cyber-physical adversaries. For example, the ones provided by Huang *et al.* [15], Li *et al.* [89], and Teixeira *et al.* [6]. In addition, Sanchez *et al.* [16] provide a detailed bibliographic review of cyber-physical attacks with illustrative examples. The classification in [15] provides control theory models for integrity and denial-of-service (DoS) attacks. A similar classification has been used in the proposal in [90] using the names deception and disruption respectively. Authors in [15] also showed that a DoS attack does not have a significant effect when the system is in a steady state. However, integrity 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 resources, and the disclosure resources. The knowledge of the system's model allows the adversary to develop sophisticated attacks, which have more severe consequences and are hard to detect with traditional approaches. The disclosure resources let the adversary obtain sensitive information, which may be used to generate knowledge about the system, but cannot be used to disrupt the system operation. Finally, the disruption resources can be used to affect the system operation.

In the sequel, we present an overview of cyber-physical attacks following the taxonomy presented in [6] for different adversary models.

False-Data Injection Attack or Stealth Attack. In this attack family, the adversary modifies some sensors readings by physical interference, at individual sensors or using the communication channel to disrupt the behavior of the system [5, 16, 91–95]. To carry out attacks from this family, the adversary needs knowledge about the behavior of the system, such as the system dynamic, the command signal, and the control detection threshold. This way, the adversary drives slowly the control decisions out of the correct behavior and produces wrong control decisions to cause a malfunction in the system. It is worth noting that, from a control-theoretic perspective, the injected false data should not affect the residue (cf. Section 2.1.2). This means that the injected data should not alter the sensor measurement variations. Otherwise, the attack would be detected easily.

Replay Attack. This attack family assumes an adversary that modifies some sensor readings by replicating previous measurements corresponding to normal operation. Then, the adversaries also replicate the control input to affect the system state. These adversaries are not required to know the system process model, but they require access to all the sensors to be successful [5, 53, 91, 96].

Covert Attack. This attack family assumes an adversary that reads and modifies both the controller input and output, i.e., the control data and the sensors measurements. The adversary requires knowledge about the physical system and the behavior of the feedback control to impersonate the feedback controller and evade fault detection [97–99].

Denial of Service (DoS) Attack. DOS attacks aim at disrupting the communication between the MTU and the RTUs/PLCs or between the RTUs/PLCs and sensors or actuators. DoS attacks break the communication between different parts of the system to disrupt the feedback control [100]. By disconnecting the controller from the physical device, it is possible to avoid the process monitoring and let the system vulnerable to other malicious actions [101] .

Zero Dynamics Attack. This attack family assumes vulnerabilities present in the dynamics of the system concerning properties used to monitor and control the behavior. It makes an unobservable state unstable and disrupts this unobservable part of the system without being detected by the controller [6, 102]. A solution to avoid this kind of attack is to update the architecture of the system in order to make all the states observable, e.g., deploying more sensors to avoid unobservable situations into the system.

Command Injection Attacks. This attack uses the protocols and devices vulnerabilities to inject false commands into the control systems to disrupt control actions or system settings. For example, overwriting remote terminal programs or registers [101, 103, 104].

2.3 Cyber-Resilience Definition

According to the literature, cyber-resilience is the ability of a system to prepare, absorb, recover, and adapt to adverse effects [12]. The preparation stage 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 absorb 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 adapt stage involves a postmortem evaluation to improve the response and learn from past experiences.

As pointed out by Arghandeh *et al.* [24], the number of resilience definitions has increased significantly over the last decade making it difficult to find a universal understanding of the term. Although the mentioned definition provides a clear view of the resilience stages, it may be too wide for CPS due to with an unlimited budget, time and effort, eventually any system will recover the operations. For this reason, we argue that resilience should be established considering a group of minimum conditions including, for example, performance and time dimensions.

Another definition provided by Tierney and Bruneau *et al.* [105] says that resilience refers to both inherent strength and the ability to be flexible and adaptable after environmental shocks and disruptive events. Arghandeh *et al.* [24] define cyber-physical resilience in power systems and compare the term with respect to other concepts such as robustness, reliability, risk management, among others. Similarly, Gholami *et al.* [27] describe and classify different high-impact events that may affect resilience and discuss differences between resilience and other well-established concepts, such as security and reliability.

Wei and Ji in [106] present a definition that describes at a high level the properties that a resilient Industrial Control System should meet. Surveys [24, 32, 36] analyze the definition of cyber-resilience for electric grids and power systems. Other resilience definitions are provided in [25, 35, 107] that review resilience definitions that apply to a wide variety of systems, such as economic, social, psychologic, ecologic, educational, engineering systems, among others.

Jackson *et al.* [108] provide a state machine that models how a resilient system works showing its states and the transitions between these states.

In this dissertation, we will consider the definition by Clark and Zonouz in [109] which expresses that resilience aims at:

1. Full correctness maintenance of the core set of crucial functionalities despite ongoing adversarial misbehavior. Hence, it is acceptable for non-crucial functionalities to be affected temporarily (partially degraded or complete failure)

2. Guaranteed recovery of the normal operation of the affected functionalities within a predefined cost limit.

Hence, attack tolerance and graceful degradation are two properties that we want 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 and it may decrease as the number of failing components grows. The purpose is to prevent a catastrophic failure of the system.

2.3.1 Risk Management vs. Resilience

Risk assessment and resilience are different but related concepts. The difference between both concepts is not clear and some literature uses risk assessment methodology for system resilience which may not be the best approach for proving resilience [24]. Risk assessment and resilience are grounded in a similar mindset of reviewing systems for weaknesses and identifying policies or actions that could mitigate or resolve such weaknesses. As pointed out in [24] and [31], there are also substantial differences.

Risk is assessed by the likelihood of an undesirable event and the consequence of that event using probability distribution functions. Resilience is about recovering from unexpected rare extreme failures, whose likelihood can not be estimated from historical data. In addition, risk assessment 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 approaches by reviewing how systems perform and function in a variety of scenarios, agnostic of any specific threat.

Finally, resilience requires thinking in terms of how to manage systemic, cascading effects to other directly and indirectly connected nodes. Resilience is grounded upon ensuring system survival and it finds strategies to keep the functionality of the core system in the face of extreme events. It is based on a general acceptance that it is virtually impossible to prevent or mitigate all categories of risk simultaneously, and before they occur. However, risk assessment centers around the probability of hitting the weak points of a system.

2.3.2 Security Requirements

CPS have special requirements that need to be considered when designing a security mechanism. In addition, traditional IT security solutions may not be appropriate to ensure the correct operation of CPS due to these particular characteristics. For example, CPS are usually large and distributed systems that may also require geographic distribution of its components.

Also, CPS are usually time-critical applications with requirements including high speed, regularity, and synchronization. In general, the physical processes do not need high throughput but demand continuous availability with guaranteed low delay and low jitter. The security methods to be implemented on CPS can not have an adverse impact on these needs as well as on the runtime or the schedule of the tasks.

Normally, a central location aggregates data from multiple locations to support control decisions based on the current state of the system. Often a hierarchical control is used to provide the operators with a comprehensive view of the entire system. The failure of a control function may have substantially different impacts across domains. CPS often require the ability to continue the operations through redundant controls, mitigation techniques, or the ability to operate in a degraded state. The system must be able to detect unsafe conditions and trigger actions to reduce the unsafe conditions to safe ones.

Moreover, the components are usually resource-constrained, e.g. in terms of computation resources, memory, or processing power; and CPS have a long life cycle due to the high upgrade costs which makes the interaction with legacy systems a requirement. Even security patches take much longer to be deployed due to the need for exhaustive testing.

Furthermore, in IT systems, the usual priority order for the three traditional security goals is confidentiality, integrity, and availability. In the CPS realm, the security goals can be prioritized in a different way. The most important factors are the ability of the system to work in high availability, then the integrity of the information and finally the confidentiality. In addition, the availability must be a real-time availability due to CPS work in an operation environment that requires making autonomous decisions in real-time.

In this dissertation, we consider attacks that affect the availability and integrity of CPS. In particular, considering the new challenges created by control-theoretic perspectives, *i.e.*, cyber Denial of Service (DoS) attacks have already been studied and many proposals aboard how to mitigate them [110–113]. In addition, as showed by Huang *et al.* [15] cyber DoS attacks are not a real problem in a CPS after the physical process reached stability. Since when the system is stable, it will tend to continue in that state without extra effort. However, integrity attacks can rapidly move the system to unsafe states. For

this reason, integrity attacks should be a priority. In addition, in the CPS realm, DoS attacks can be launched using integrity attacks to cause significant damage.

In contrast to the traditional cybersecurity domain, cyber-physical adversaries show more tactical interest in degrading the system integrity (i.e., from a short-term perspective) and strategic interest in perturbing the system availability (i.e., from a long-term perspective). For this reason, CPS are susceptible to a new type of cyber-physical DoS attacks. In this case, as showed in Figure 1.1, the integrity of the messages is compromised with two objectives. First, to disrupt the communication between the controller and the plant, generating a loss of the system 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 unavailability of the system to the authorized users in order 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.

To build a CPS capable of ensuring availability and integrity, the system has to follow some properties to be managed and controlled in a safe manner. These properties are the following:

- Stability: a system is stable if the output signal response to a bounded input signal is also bounded. Otherwise, the system is unstable. Time delays and packet dropout have to be considered to handle the stability [114, 115].

- Controllability: this is the ability of a system to bring the process into a desired state [14].

- Observability: refers to the ability to measure the process state and maintain situational awareness [14]. Hence, it is possible to create a map of their states from the output of the system without knowing the initial state.

2.4 Security Approaches

New adversary models, as presented in Section 2.2, have created new challenges to achieve reliable systems. In this section, we show that achieving security in CPS requires tools that extend beyond what is offered in the state of the art software and cybersecurity. In particular, combining tools from both cybersecurity and control theory to defend against malicious behavior. First, we present approaches that work with the traditional detection-reaction strategy that triggers a defensive response when malicious activity is detected. Then, we present strategies to build resilient designs that provide system recovery without triggering any additional behavior.

2.4.1 Detection-Reaction Paradigm

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. As a result, it was born a new research area focused on detection-reaction strategies to face cyber-physical adversaries, which in the control-theoretic community is known as *Resilient Control*.

Resilient control incorporates to the traditional fault-tolerant control new strategies to face cybersecurity breaches. It aims at recovering and resiliently respond to attacks on the control system to achieve stability and graceful degradation of the performance under attack. This objective can be achieved through a system theoretical analysis of the CPS.

Weerakkody *et al.* [17] provides a comprehensive survey about resilient control in CPS, *i.e.*, how a controller can detect, correctly estimate the system state and recalculate the required command despite malicious data. The survey also covers how to mathematically model a CPS and different adversaries. In addition, Chabukswar *et al.* [96] analyze the effect of integrity attacks on control systems and provide a countermeasure to expose and detect such attacks. Also, Pasqualetti *et al.* [116] propose a mathematical framework to analyze attacks in CPS. They also provide centralized and distributed detection and identification mechanisms.

In the sequel, we review the main techniques to detect and react to cyber-physical adversaries.

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 the cyber and physical layers.

The *data-based approach* does not require system and attack models for the detection. It is based on traditional machine learning and pattern recognition techniques [117–119] for analyzing hidden patterns in the observed training dataset, for example, command signals and sensor measurements. This traditional detection technique is not able to detect all kinds of cyber-physical adversaries.

Mitchell *et al.* [120], Cheminod *et al.* [22], and Han *et al.* [121] provide surveys of intrusion detection techniques focusing only on data-based approaches using traditional intrusion detection systems. Ahmed *et al.* [122] provide a survey of trust-based detection and isolation approaches for malicious nodes in sensor networks. In addition, Ding *et al.* [28] survey the development of attack detection for industrial CPS and discusses the control and state estimation in case of an attack. Also, Beaver *et al.* [123] provide an

evaluation of machine learning methods to detect malicious communications in SCADA technologies.

The *model-based 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 are 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 four main strategies for control-theoretic model-based attack detection [124]: *watermark-based detector*, *signal-based detector*, *state relation-based detector*, and *cross layer-based resilient detector*.

In the case of *watermark-based detectors*, a low amplitude noise, called watermark, is added to the control measurements to verify 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.* [52, 53] propose the use of a watermark to detect replay attacks by adapting traditional failure detection mechanisms. After that, Miao *et al.* [125] improve the performance of this detection mechanism with another algorithm using a stochastic game approach. Then this work was improved by Rubio-Hernan *et al.* [40] in order to incorporate more advanced adversaries capable of learning the physical model. In the same way, Do *et al.* [126] 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.* [127] 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.* [128] present a protocol for detection and localization of disturbance based on a special correlation matrix. The matrix allows to detect anomalies using spectral methods; localize a subset of anomalous nodes within the system; and identify 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.* [129] 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.* [130] present a distributed anomaly detection algorithm using graph theory and

spatiotemporal correlations to analyze the physical process in real-time. Amin *et al.* [131, 132] developed 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 failure or attack, they use a residual generation procedure. Also, Dehghani *et al.* [133] present a static state estimation algorithm able to detect the anomalies in integrity attacks in smart grids.

Finally, *cross-layer based resilient detector* combines control and cyber techniques in a single intrusion detection system. For example, Zhu *et al.* [134] propose a game-theoretic framework that integrates the discrete-time Markov model for modeling the evolution of cyber states with continuous-time dynamics for 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.* [94] 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.* [135] use geometric control theory to optimize cross-layer resilient control systems. They conclude that by using a geometric model of the system is possible to detect faults or estimate the system state in the presence of unknown inputs.

Reaction Approaches

As pointed out in [13], 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 reaction 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.

Resilient state estimation is a technique that can help in the 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 compromised [17]. As a result, a 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.* [136] propose an efficient state reconstructor inspired by techniques used in compressed sensing and error correction over the real numbers. They also characterize the maximum number of attacks that can be detected and corrected as a function of matrices A and C of the system. Pajic *et al.* [137] present a method for state estimation in 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.* [138] also proposed 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 [139] 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 compromised less than half 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.* [140] 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 plant, unknown inputs Kalman filters are used to estimate the state of the system and the malicious unknown input. In addition, when the malicious control signal is blocked at the occurrence of data losses, the unknown input is transformed to a constant bias at the input of the plant. Weimer *et al.* [141] 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.* [142] and Mishra *et al.* [143] 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 [144, 145].

Another technique used to improve the state estimation accuracy is to consider multiple sensor systems instead of one single sensor system [146–148]. In this case, data fusion is a process in which the received data is integrated from different sensors observing the same system.

The previous techniques allow estimating the state of the system even when the controller receives data that has been compromised. It is hard, however, to ensure that the control command is executed correctly by the actuator even when an adversary has compromised the network or some components is not explained by state estimation techniques. For that, other proposals need to be included. Proposed attack mitigation aims at dynamically altering the configuration of the system to minimize the effects of the attack. For example, changing the network topology, devices configurations, firewall rules, or rerouting traffic to honeypots. As a consequence, 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 adding extra layers of security to those elements that are more vulnerable to cyber attacks [19] or components may be intelligently isolated.

Li *et al.* [149] propose a decision-making approach for intrusion response aiming to determine the optimal security strategy against attacks. The strategy tries to secure the

most dangerous attack paths and respond to functional failures. Authors assess both cyber and physical domains with in-depth analysis of attack propagation. Yuan *et al.* [100] propose a resilient controller design for CPS under DoS attacks. The proposal uses a framework that incorporates an IDS and a robust control. The robust control in the physical layer is based on an algorithm with value iteration methods and linear matrix inequalities 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 capabilities on demand to face the attack. 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.

For example, Cavalli *et al.* [150] present a methodology using software reflection to prevent, detect, and mitigate internal attacks to a running Internet Web server. In the software design, some parts are marked as secured, and any modification of these parts will be an unexpected behavior that needs to be analyzed. If these changes turn out to be attacks, then some remediation techniques are activated.

Some other proposals are based on *programmable networking* that enables efficient network configuration that can be used for neutralizing attacks. This way, 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 [151], Active Networking [152], and Netconf [153], among others, are early programmable networking efforts and precursors to current technologies such as Software Defined Networking (SDN) [154]. 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 is not new. Some examples include its use for conducting DoS (Denial of Service) attack mitigation [155] and segmentation of malicious traffic [156, 157]. 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.* [18]

survey approaches for SDN controllers that are able to establish different paths between sensors and actuators. Piedrahita *et al.* [158] 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. In this way, 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.

Based on how frequent the attacks occur, *event-triggered control* schemes instead of time-triggered schemes emerged as appropriate tools to increase the resilience of control systems [90, 159]. The application of event-triggered control to the resilience of CPS has been studied in [160–162] where the triggering function to generate a new control input is based on the errors of state variables.

Ismail *et al.* [163] 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.* [164] survey strategies to do this using both game theory and Markov decision processes to analyze the interactions between the attacker and the defender.

2.4.2 Cyber-Resilience Paradigm

Despite all the implemented cybersecurity prevention mechanisms, it is possible to have a system breach. For this reason, cyber-resilience identifies techniques to absorb, survive or recover from threats. Cyber-resilience demands flexibility, adaptability, and agility with real-time reactions to disturbances. A growing number of technologies and architectural practices can be used to improve the cyber-resilience. In this section, we analyze cyber-resilience techniques.

Most of the existing surveys covering techniques to achieve cyber-resilience in CPS are focused on resilient control, *i.e.*, secure state estimation and calculation of new commands to repair the caused damage. These strategies work mainly as detection-reaction strategies since they require to identify the presence of the adversary previously. For this reason, they are not purely cyber-resilience as defined in the computer science domain. Such terms may be confusing since in the control theory community these strategies are called *resilient control*.

Other existing techniques to improve cyber-resilience are traditional network resilience approaches. For example, Psaier *et al.* [165] survey self-healing techniques based on the principles of autonomic computing and self-adapting system research. Cholda *et*

al. [20] analyze the quality of service, availability, and maintainability. Authors define a concept called quality of resilience which is a unified performance metric that evaluates the frequency and length of service interruption. Other resilience surveys specifics for CPS can be found, for example, in Mishra *et al.* [36] that survey methods to improve the resilience but the scope of the paper is focused on power systems, including a mechanism that may not be applicable for others kind of systems. Also, Bodeau *et al.* [23] present an interesting cyber-resilience framework that identifies goals, objectives, and technique domains that may be used to improve resilience.

Due to the existing resilience surveys cover mainly resilient control techniques, in the rest of this section, we cover other techniques that may be used to build resilient systems. We provide a taxonomy of cyber-resilience techniques and a literature review with different proposals that may be applied in each of these techniques.

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, change or remove 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 does the resilience approach work. 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 2.2 sums up the different cyber-resilience strategies and scientific proposals that use them.

Architecture Design

These strategies involve modifying the system architecture to improve the resilience of the system to absorb or survive the attack impact.

- **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. It is worth noting, that this technique requires adding

Technique	Phase			Layer				Proposals
	Absorb	Survive	Recover	Physical	Network	Control	Cyber	
Architecture design								
Diversity		✓		✓	✓		✓	[179], [180], [181], [182], [175], [176], [177], [178], [171], [172], [173], [174], [166], [167], [168], [169], [170]
Segmentation	✓				✓			[183], [184]
Reconfiguration								
Isolation and Containment	✓			✓	✓		✓	[187], [188], [189], [190], [185], [186]
Dynamic Network Composition	✓	✓						[158], [191], [192], [193]
Non-Persistence	✓	✓						[194], [195]
Moving Target Defense (MTD)								
Network MTD	✓	✓	✓		✓			[205], [206], [207], [208], [201], [202], [203], [204], [197], [198], [199], [200], [196]
Node MTD	✓	✓	✓			✓	✓	[205], [211], [202], [200], [196], [209], [206], [210]
Dynamic Software Evolution	✓	✓					✓	[212], [150], [213], [214]
Consensus and Distributed Trust	✓	✓			✓	✓		[223], [224], [225], [219], [220], [221], [222], [215], [216], [217], [218]
Game Theory	✓	✓					✓	[228], [229], [230], [231], [226], [227]

Table 2.2 – Proposed cyber-resilience approaches for CPS.

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 technique, it is required to decide what to diversify and when to diversify it [179]. 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, 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) [181], at compiling and linking the source code [171–173, 175–178] or at installation, loading or execution time [166–169, 174].

Other diversity solutions may work also in a detection-reaction manner. For example, Ouffoué *et al.* [180, 232] use diversification to create attack tolerant web services. They modeled the services to extract different implementations using variation 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 [182] 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.

- **Segmentation.** The design of a CPS must consider how to prevent attacks and be more tolerant to intrusions from the beginning. Network segmentation strategy separates logically or physically the components to reduce the attack surface, contain and limit the damage of a successful attack. The components may be separated base on their criticality, trustworthy or functionality [183, 184].

  According to the results achieved in [183], this technique also contributes to build more intrusions tolerant CPS. Network segmentation may be designed considering the Process-Aware Control approach presented in [184]. 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) and therefore place them in a network segment separate 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. 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.

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 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.

- **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 [185] 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 to understand cyber-physical interactions, limit the negative effect of intra-dependencies and improve resilience.

  Xu *et al.* [186] 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. They construct numerous expected fault scenarios, then they select some of them and develop their information entropy. Second, for each typical scenario, a multi-stage switch strategy considering both isolating and fault reconfiguration through dynamic programming.

  Bellini *et al.* [187] analyze 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.* [188] analyze how attacks in communication networks may cause cascading failure in 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, provide guidance to adjust intra- and inter-links, and study the robustness of the system in various attack scenes.

Haque *et al.* [190] analyze resilience for energy delivery systems considering cyber components and services 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.

- **Dynamic Network Composition.** This technique designs the system with dynamic capabilities to face the attack. For example, distributing tasks in different organizations. Januario *et al.* [191] 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.* [193] propose an approach to improve resilience using the synchronization of multi-agent systems that address faults and uncertainties on communication links. For that, they transform the resilient control problem into distributed state observers.

  Marshall *et al.* [192] present a context-driven decision engine for adaptive resilient control. The solution integrates diagnostic and prognostic heuristics to establish situational awareness and drives actions. The proposal assesses the system state of health based on operational availability and drive control decisions based on scenario-specific constraints and priorities. Similarly, Ratasich *et al.* [233] presented a self-healing framework that uses structural adaptation, by adding and removing components, or by changing their interaction, at runtime.

- **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 previous known image to ensure that the current system image is correct.

For example, Griffioen *et al.* [194] 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 using 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.* [195] 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.

Moving Target Defense

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, MTD approaches provide strategies that change the system over time to increase its complexity, attack cost, or limit the exposure of vulnerabilities [196]. The mechanisms are usually applied at the network or the node level [200]. Next, we summarize proposals for both levels as well as approaches specially designed for CPS.

- **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 follow. Antonatos *et al.* [208] 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.* [207] 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.* [203] hide the endpoint MAC, IP and port numbers by setting up 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.* [204] use a secret sharing technique to encrypt its 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.* [197] 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.* [198] propose a Random Route Mutation technique that enables changing randomly the route of the multiple flows in a network simultaneously to defend against reconnaissance, eavesdrop and DoS attacks while preserving end-to-end QoS properties. Ma *et al.* [199] 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.

- **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 [234]. Address space, instructions or data randomization are three typical ways to achieve platform environment diversification [235]. Another technique is software application isomerization that 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 mechanism implemented during software load and execution [200]. In addition, programmable reflection is a meta-programming technique that has the potential to allow a programmable system to manipulate itself at runtime [150].

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 [206], Giraldo *et al.* propose a 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 to minimize the impact of false-data injection attacks. Griffioen *et al.* [211] propose a 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. Weerakkody *et al.* [210] proposes a 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.* [205] 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.

Dynamic Software Evolution

Dynamic software evolution uses code generation or modification at runtime to adapt the system behavior and face adversaries.

- **Runtime Code Generation.** Code Generation techniques 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.

- **Software Reflection or Self-Modifying Code** is another technique that allows a system to adapt itself through the ability to examine and modifying 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 code 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 loading it into a object or use reflection to execute it.

He *et al.* [213] propose an approach to modify the software runtime architecture through meta-operators based on reflection. Similarly, Kon *et al.* [214] propose a reflective middleware to deal with highly dynamic environments, supporting the development of flexible and adaptive systems and applications. Mavrogiannopoulos *et al.* [212] present a taxonomy of self-modifying code with the purpose of obfuscation.

Consensus, Secret Sharing and Distributed Trust

Both consensus, secret sharing 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 [236–238], 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 [219, 225] or other distributed observers strategies to manage sensor compromise [224, 239]. Other strategies are distributed function calculation in the presence of malicious agents [223], distributed multi-agent consensus [220–222, 240, 241], resilient vector consensus [215, 217] and resilient leader-followers consensus approaches [216, 218].

Techniques such as secret sharing schemes [242–244] and distributed trust [245, 246] 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. The main approaches to build secret sharing schemes are the Shamir's threshold approach [242] 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 [244] 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 scheme are based on monotone formulas, for example, the proposal in Ito *et al.* [247], the monotone formulae construction [248] and the monotone span programs construction [249, 250]. A monotone function is a function entirely non-increasing or non-decreasing, *i.e.*, its first derivative does not change sign. Every monotone formula computes a monotone function and every monotone function can be implemented using just AND and OR operators. Benaloh and Leichter [248] 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 the reputation that chooses between benevolent and malicious behavior. Then using relationships and inferring rules, different levels of trust are derived for other entities [246]. 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, and can also prescribe actions and outcomes if certain conditions are met [251]. 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 [252].

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 [253]. It studies the decision 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, each player can not optimize its objective independent 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 adversary 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 and Zhu [229] 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.* [228] 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 in order 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.* [254] model the interactions between an attacker and a defender and derived the minimum defense resources required and the optimal strategy of the defender that minimizes the risk. The solution is analyzed in power system. Also, Rao *et al.* [227] 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 [226] 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 [230] 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 [231] considers 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.

2.5 Cyber-Resilience Evaluation

In this section, we address how scientific proposals have been evaluated in the literature. To achieve that, we analyze validation platforms that have been used, including simulation tools, CPS application scenarios and experimental testbeds. We also review proposed evaluation metrics to analyze the cyber-resilience of a system.

2.5.1 Validation Methods

The research community has used three main validation methods to test CPS approaches. This includes formal mathematical proofs, case study simulations [4] and experimentation in testbeds.

Mathematical proofs show through formal approaches the numerical improvement and present illustrative numerical examples. This kind of testing approach is not the most suitable for the work in this dissertation, since it may be hard to quantify formally physical and cyber aspects of our solutions. For this reason, we focus on simulations and testbeds.

Simulations allow having a complete plant model and more complex scenarios to do the tests. Testbeds have the advantage of incorporating physical devices such as sensors and actuators creating more realistic scenarios. However, they are more expensive and normally the implemented system is simpler than in simulation scenarios. In the sequel, we present the main simulation and testbed that have been used in the literature to validate CPS proposals.

Simulations

A frequently used tool for simulations is Matlab/Simulink [4]. It allows programming mathematical algorithms and also provides a graphical programming environment for modeling, simulating and analyzing dynamical systems. Other simulation tools and libraries are built over Matlab. For example, MatPower [255, 256] is a free and open-source tool for electric power system simulation and optimization. It is built as a package for solving different steady-state power system simulation problems. In addition, Zhang *et al.* in [11] propose a CPS visualization framework, using the QEMU system emulator [257] as a visualization machine, and Matlab/Simulink to emulate physical components. This framework allows reproducing both cyber and physical layers. Another possible

integration to incorporate the cyber components to the physical simulation in Matlab is to integrate a network simulator as NS-3 [258] or Omnet++ [259] as showed in [260]. Creating the CPS components and communicate them with specific CPS protocols may be hard work due to these two network simulation tools are for general networks and not CPS focused. For this reason, Queiroz *et al.* [261] proposed a library that integrates predefined CPS components and protocols to use in Omnet++.

With respect to simulation scenarios, the Tennesse Eastman problem [46] is a frequently used process control system model for validation purposes [4]. It presents a multi-loop proportional-integral control law with multiple sensors and actuators that takes a chemical substance to produce a final product. Chabukswar *et al.* [260] presents a reduced version of the Tennessee Eastman challenge problem presenting the transfer function of the physical process and integrating the system models with network simulations with Omnet++ to reproduce the industrial plant.

Another industrial testbed is presented in Downs and Vogel [262], which is the manufacturing process of vinyl acetate monomer. The provided models can be used to create a simulation of the industrial process that takes chemical components to create the vinyl acetate [263–265]. Krotofil and Larsen in [14] also use the Tennesse Eastman and Vinyl Acetate Monomer plant to conduct security tests. Simulations conclude that a successful attack has to manage cyber and physical knowledge.

Myat-Aung in [266] presents a Secure Water Treatment (SWaT) simulation, using Labview and Simulink. The testbed and simulation are based on a security standard (ISA-99) proposed by Industrial Automation and Control Systems. SWaT consists of a 6-stage water treatment process, each stage is autonomously controlled by a local PLC. The local fieldbus communications between sensors, actuators, and PLCs is realized through alternative wired and wireless channels [267]. Havarneanu *et al.* [268] also present a dataset extracted from SWaT to support experimental work. It has information from a 11-day non-stop run that started in empty-state to a fully operational state. For the first seven days the system operated normally without any attacks or faults. During the remaining days, certain cyber and physical attacks were launched on SWaT to collect their data.

Finally, Yu and Jiang [269] present a model to represent an aircraft. They analyze physical faults in the hydraulically-driven control surfaces and they also propose a hybrid fault-tolerant control system.

Experimental Testbeds

The quadruple-tank process by Johansson [270] is a frequently used experimental testbed [4]. It is a multivariable laboratory process consisting of four interconnected water tanks that move the water from one tank to another using pumps and level sensors.

Another commonly used testbed is the Landshark robot [271] which is a fully electric unmanned ground vehicle. It has an onboard computer with a Linux system running. The computer performs all tasks such as PID control, LIDAR, GPS, IMU, and encoders. Also based on unmanned vehicles, Rubio-Heran *et al.* [272] propose a testbed based on Lego Mindstorms EV3 bricks and Raspberry Pi boards as PLCs to control some representative sensors (e.g., distance sensors) and actuators (e.g., speed actuators) using Modbus and DNP3 communication protocols.

For power grids systems, Yardley in [273] proposes a cyber-physical testbed based on commercial tools that combine emulation, simulation, and real hardware to experiment with smart grid technologies. Similarly, Koutsandria *et al.* [274] implement a real-time testbed for cyber-physical systems security on power grids, where the data are cross-checked using cyber and physical elements.

2.5.2 Evaluation Metrics

To build resilient systems, it is important to develop appropriate metrics to assess and demonstrate the utility of the proposed approaches. In this line, different research works have identified attributes to measure the resilience of a system. For example, Linkov *et al.* [21] provide a matrix framework with resilience metrics in cyber systems. These metrics link policy goals to specific system measures, such that resource allocation decisions can be translated into actionable interventions and investments. The metrics have been identified and assessed using quantitative and qualitative measures. However, it does not capture the runtime performance of a system or the temporal component of resilience, which is an important factor to consider.

Jain *et al.* [29] survey metrics in risk and resilience assessment and management of chemical process systems considering three phases: avoidance, survival, and recovery. They include twenty-four resilience metrics covering both technical and social factors. In addition, Fang *et al.* [275] propose a metric to evaluate the criticality of a component in a network system from the perspective of their contribution to resilience. Specifically, the two proposed metrics quantify the priority with which a failed component should be repaired and the potential loss in the optimal system resilience due to a time delay in the recovery of a failed component. This approach does not analyze the resilience of the system as a whole. Hence, it is not possible to quantify whether a proposed approach improves resilience or not. Also, it does not allow comparing different system designs to determine which one is the best from a resilience point of view.

Francis and Bekera [276] propose a resilience analysis framework and a metric for measuring it. The framework is focused on the achievement of three resilience capacities: adaptability, absorbability, and recoverability. These properties are the basis for the resilience metric. The approach presents a general metric designed to apply to a wide

variety of systems, such as physical, economic, social, ecological, among other types of systems. Due to its generality, the mechanism is not the most suitable for evaluating the reaction of a CPS when facing an attack considering the stability and safety it will provide, since it is not capable of considering all the specific characteristics of this kind of system.

Mohebbi *et al.* [35] review resilience quantification techniques for water, transportation, and cyber infrastructures conceptualizing three types of interdependencies including cyber, physical, and social. Linkov and Trump [31] analyze different resilience definitions and metrics to quantify and assess resilience. In addition, they provide different resilience case studies based on epidemiological and natural disaster events. Also, Bhusal *et al.* [32] provide a review of resilience metrics, evaluation methods, and enhancement strategies for power systems.

Rieger in [277] presents a metric framework that integrates the cognitive, cyber, and physical aspects considering time and data integrity characteristics. Resilience is considered for control stability and the author uses control response and stability as a performance measure. Similarly, Eshghi *et al.* [278] use traditional performance metrics to provide a visualization methodology for operators and indications of issues that show the impact of the disturbances. These metrics are related to state awareness of the real-time operation, but they do not allow evaluating in advance the reaction of the system. In addition, they consider physical threats and cyber threats in a separate manner. Hence, it is not clear if the approach will be able to handle cyber-physical adversaries capable of making the system lose the state monitoring of the system.

Finally, Clark and Zonouz [109] propose a resilience metric for CPS modeled as linear systems with and without actuator saturation. It considers both the physical and cyber aspects of the systems. They quantify the ability of the system to recover from an attack under the assumption that the attack is discovered within a fixed time interval and evaluating its domains of attraction. The proposed physical evaluation is based on the stability evolution of the system. However, it is a mathematical abstract definition that may be hard to apply to practical evaluation.

2.6 Discussion

In this chapter, we have surveyed control theory and cybersecurity strategies applied to CPS. In this section, we discuss why both domains should work together and the new possibilities that this synergy can create to improve the design and resilience in CPS.

Control theory and cybersecurity are research areas that provide significant contributions to solve security issues in CPS from different perspectives. Security in CPS is a dual problem with a part in the cyber world and the other part in the physical one. Hence,

as pointed out in [16, 279], both research 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 [279], CPS security can be divided into two main categories: information security which focusing 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 which 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 2.4.1. 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 [17], the confidentiality of encrypted sensor measurements can be violated by placing un-encrypted 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 showed in Section 2.2, the malicious traffic can be confused with legit 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 [280, 281], the replay attack [96], the zero-dynamics attack [282] and the covert attack [99]. Last but not the least, insider adversaries and human error that generate security breaches have to be also considered to ensure safety.

Although, control theory and cybersecurity are complementary, for both research communities it is still hard to integrate their knowledge or at least diminish the gap between the two domains. In Section 2.4.1, we have provided an overview of the research efforts to integrate both disciplines to improve cybersecurity in CPS.

2.7 Summary

This chapter has provided the state of the art analysis, including an introduction to the background and related work of the dissertation. It has introduced related concepts, the system model, the architecture and surveyed threats.

A comprehensive literature review about security solutions for CPS has been also analyzed. This review is presented such that the proposals are classified according to detection-reaction and resilience techniques. The aim has been to emphasize the context, the main existing challenges and to anticipate the required concepts for the contributions that will be presented in the following chapters. In particular, we have identified that plenty of research effort has been done in detection techniques and resilient state estimation to maintain an awareness of the system state despite an attack. However, much less attention has been paid to adapt reaction and resilience techniques to the needs of CPS. In particular, considering the control-theoretic characteristics of the system.

We have also identified that the difference between detection-reaction and resilience is not clearly defined in the literature and often, these two concepts are mixed. 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 is able to 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 to maintain the core set of critical functionalities despite ongoing adversarial misbehavior and guarantee the recovery of the normal operation within a predefined cost limit.

We have also surveyed some efforts in the literature in terms of cyber-physical evaluation methods as well as metrics to validate the resilience improvement.

Finally, we have discussed and emphasized the need of considering control theoretic approaches to improve cyber-physical systems security. These two complementary domains provide different advantages that can be used to create more secure systems by reducing the existing gap between them.

3 Detection-Reaction Paradigm

3.1 Introduction

Cyber-physical systems (CPS) are modern control systems used to manage and control critical infrastructures [283]. As showed in Chapter 2, the physical properties of such infrastructures are modeled via control-theoretic tools, e.g., *control-loops* and *feedback controllers* [49]. Feedback controllers have to be able to manage the behavior of the CPS, by confirming that the commands are executed correctly and the information coming from the physical states is consistent with the predicted behavior [284]. Feedback controllers are also used to compute corrective actions, e.g., by minimizing the deviation between a reference signal and the system output measurements (cf. Chapter 2, Section 2.1.2).

A CPS is composed of three main layers: (1) the *physical layer*, which involves the physical process monitored and controlled by physical sensors and physical actuators; (2) the *control layer*, which is in charge of regulating the operation of the physical process via control commands; and (3) the *cyber layer*, which is responsible for monitoring operation and supervision tasks. These three layers are interconnected using a communication network. The interconnection between information and operational systems leads to new security threats [9, 285]. Traditional *cyber attacks* are well known and countermeasures have been studied. However, launching a *cyber-physical attack* requires a different knowledge from the one used in traditional cybersecurity and different protection techniques are also required (cf. Chapter 2, Section 2.2).

A physical process has automatic safety measures and operational constraints, e.g., to disable a physical process when certain dangerous conditions are met. For instance, to properly respond when a physical component fails. For this reason, an adversary who aims at damaging the physical process needs to understand how the dynamics of the physical plant work. This means that compromising and disrupting a device or communication channel used to sense or control a physical system is a necessary requirement to perform cyber-physical attacks. The damage can be limited if the

adversary succeeds at affecting the cyber layer, but remains unable to manipulating the control system (*i.e.*, fails at perturbing the physical process). To achieve the desired impact and achieve a cyber-physical attack, the adversary needs to assess how the attack will perform at the control level. Therefore, to achieve a cyber-physical attack, the first step is to get control over the cyber layer, to obtain remote access within the target system. Then, the second step is to learn about the physical process and how the control layer works in order to manipulate the physical layer and cause damage to physical components. Adversaries need to know how the physical process is controlled, failure conditions of the equipment, process behavior and signal processing [9, 285].

In this chapter, we propose a technique to attenuate cyber-physical attacks that uses programmable reflection and programmable networks to sanitize the malicious actions introduced by some cyber-physical injection attacks such as false data injection, bias injection, replay attack, command injection and cover attack [6]. The adversary uses the network to manipulate the process through the modification of specific payloads. Then, the proposed technique uses the network to neutralize the attack effects. We validate the approach using experimental work.

3.2 Contributions

As showed in Chapter 2, Section 2.4.1, most of the proposed approaches are based on a control-theoretic detection strategy combined with a resilient state estimation that allows a remote defender to maintain knowledge of the system state under attack, even when a subset of inputs and outputs are compromised. There are much fewer proposals about how to react after the detection and the system state evaluation, *i.e.*, it still exists the problem to ensure that the correct estimated control commands arrive and are executed correctly by the actuators. For this reason, in this chapter, we propose a mitigation approach to attenuate cyber-physical attacks and sanitize malicious traffic injected into the network.

Our main contributions are as follows: (1) an approach to attenuate cyber-physical attacks and (2) an experimental work that validates the approach via simulation. The approach relies on the use of programmable reflection, which is a meta programming technique that has the potential to allow a programmable system manipulate itself at runtime and the use of programmable networks to sanitize the traffic. The approach builds upon the concept of *programmable reflection* and *programmable networking*. This way, we propose a technique to handle cyber-physical injection attacks and we revisit the use of programmable networking in [157], to achieve a reflective attenuation of CPS attacks. Parts of the contributions explained in this chapter were published in [286, 287].

The outline of this chapter is summarized as follows. Section 3.3 presents the system model, the adversarial model and the problem formulation. Section 3.4 presents our

reflective mitigation approach and Section 3.5 presents the experimental work to validate the proposal. Finally, Section 3.6 discussed the obtained results and Section 3.7 summarizes this chapter.

3.3 Problem Formulation

In this section, we present some initial preliminaries about our assumptions in terms of the system and adversarial modeling and the problem that we want to solve.

System Model — We assume a system modeled as described in Chapter 2, Section 2.1.2 (cf. Equation (2.2)) that is governed by closed-loop feedback controllers. As showed in Figure 3.1(a), the feedback controller collects the *sensor* measurements y_k to determine the state of the system process. Then, the *feedback controller* determines a control input using the received data and the reference obtained from the model. Finally, it sends a control input u_k to the *plant* so that the *actuators* perform the required actions in the physical process. After this, the sensor obtains new measurements y_k and the process is repeated. The values y_k and u_k are exchanged between the feedback controller and the plant through a network. It means that the data will be forwarded through a set of network forwarding devices to reach the appropriate destination. We assume that this network is highly distributed, with real-time traffic and a dynamic system interconnected using a programmable network that is controlled by a *network controller* (e.g., an SDN controller [154]). In addition, we assume a k-resilient or k-vertex-connected network [288]. A network is k-resilient if and only if any two vertices of the network graph are connected by at least k vertex disjoint paths [289]. A communication network is fault-tolerant if it has an alternative path between vertices because the vertex connectivity indicates the minimum number of nodes an adversary has to remove to make the graph no longer connected. As a result, the more disjoint paths, the better. A k-resilient system with N components can tolerate up to k component failures and still function correctly [290, 291].

Also, we consider that the system is protected from a cybersecurity point of view. This means that the system has been created considering all the required security mechanisms according to risk analysis of the system. However, past experience has showed that despite all the prevention actions, attacks are still possible. The proposed approach aims at protecting the system in a contingency mode after the other security mechanisms failed. This way, major failures in the physical process may be prevented.

Adversary Model — Our proposal addresses the cyber-physical injection attacks mentioned in Chapter 2, Section 2.2.2 and we assume an insider adversary. To do this, the adversary firstly exploits a cyber vulnerability to gain access to the network channel and be able to insert or modify packets at will. After that, the physical attack to control the physical process starts. To achieve this, the adversary injects a bias in the payload

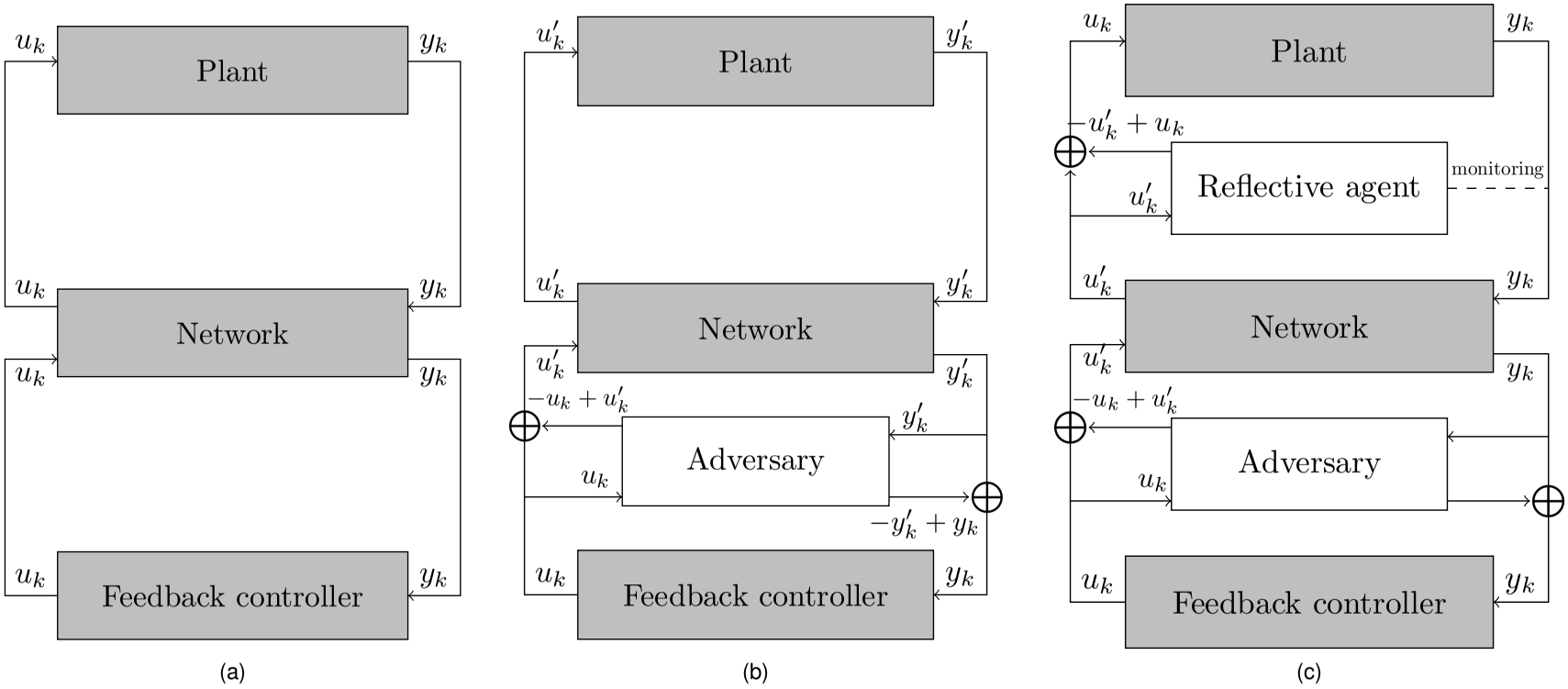


Figure 3.1 – Feedback control view. (a) Normal operation mode. (b) System under attack. (c) Attack attenuation.

of the packets containing the commands or the measures to manipulate the process. The choice of the introduced modifications depends on the specific impact the adversary wants to produce on the process.

Figure 3.1(b) depicts an attack against the closed-control loop. We use the traditional representation of a networked-control system. It shows the way how an adversary conducting a cyber-physical attack is represented by the control system community through block diagrams. The $\bigoplus$ symbol in the figure represents a *summing junction*, *i.e.*, a linear element that outputs the sum of a number of input signals. As showed in Figure 3.1(b), the adversary modifies the control input u_k to inject a modified u'_k value and affects the system state to disrupt normal operation conditions. Then, the adversary modifies the plant measurements y'_k to send a value y_k to the controller. This way, the controller receives a value y_k that is correlated with the command u_k that it previously sent to the plant. This can be achieved by recording and replicating previous measurements corresponding to normal operation conditions or by injecting some values calculated from the adversary estimated control model of the system.

We also assume that the adversary performs its malicious actions in the cyber-physical system, *i.e.*, at the data layer of the network domain. This means that the adversary is not attacking the programmable network itself, e.g, the control layer. We focus on adversaries that use the network to damage the system. Adversaries that may compromise the physical nodes themselves, to damage the system, are out of the scope due to this kind of systems usually have good physical protection mechanism implemented.

Attack Model — Perpetrated by the adversary, the cyber-physical attacks are represented as follows [6]:

$$x_{k+1} = Ax_k + B(u_k + u_k^a) + w_k \tag{3.1}$$

$$y_k = Cx_k + v_k + s_k^a \tag{3.2}$$

where

$$x_{k+1}^a = Ax_k^a + Bu_k^a \tag{3.3}$$

$$s_k^a = -Cx_k^a \tag{3.4}$$

The variable u_k^a represents the contribution of the adversary to the input. Equation (3.3) is the state transformation due to the adversary. In Equations (3.2) and (3.4), the term s_k^a represents the manipulation done by the adversary of the sensor measurements such that the attack is not visible to the operator. It erases the effect of its input on the output.

Detection Technique — Detection techniques for cyber-physical adversaries have been presented in Chapter 2, Section 2.4.1. In particular, to implement this approach, we have used the detection approach proposed in [40, 96]. This mechanism work as

a challenge-response using a watermark to detect cyber-physical attacks. We include in this subsection a short summary of this cyber-physical attack detection approach. It comes directly from references [9, 10, 40, 96, 285] and citations thereof.

The technique adapts an error detector towards an anomaly detector. The resulting scheme provides a cyber-physical attack detector using *linear time-invariant* models of the plant. Built upon *Kalman Filters* and *Linear-Quadratic Regulators* (LQR), the scheme uses authentication watermarks to protect the integrity of physical measurements communicated over the cyber and physical control domains of a networked control system.

Based on the mathematical modeling of the plant defined in Chapter 2, Section 2.1.2, a widely used control technique is the *Linear Quadratic Gaussian* (LQG) approach where the overall 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.

This way, it is possible to design an anomaly detector of malicious stationary signals, to protect a linear-time invariant plant, controlled by a LQG controller under the presence of an adversary applying the attack model previously defined.

We denote by u_k^* the output of the LQR controller given by Equation (2.4) and with u_k the control input that is sent to the plant (cf. Equation (2.2)). The idea is to superpose to the optimal control law u_k^* a watermark signal $\Delta u_k \in \mathbb{R}^p$ that serves as an authentication signal. Thus, the control input u_k is given by:

$$u_k = u_k^* + \Delta u_k \tag{3.5}$$

The watermark signal is a Gaussian random signal that is independent both from the state noise (*i.e.*, w_k) and the measurement noise (v_k). The authentication watermark is used by the detector to identify the malicious signals originated by the adversary defined above. Since the optimal control law u_k^* is equipped with the authentication signal Δu_k, the detector (physically co-located within the controller) triggers an alarm whenever a malicious signal is observed, *i.e.*, whenever the challenge sent by the controller over the plant is not observed within the measurements returned by the plant. Towards this end, [52, 53] propose to employ a χ^2 detector, *i.e.*, a well-known category of real-time anomaly detectors classically used for fault detection in control systems [292], to signal the anomalies identified in the behavior of the plant.

By using the authentication signal, we can now define the alarm signal g_k (cf. Equation (3.6)) using the residues $r_k = y_k - C\hat{x}_{k|k-1}$ generated by the aforementioned estimator. The values of g_k are compared with a threshold γ to decide whether the plant is in a nominal state or under attack. The threshold is tuned to minimize false alarms

[52, 53]. The alarm signal g_k is computed as follows:

$$g_k = \sum_{i=k-w+1}^{k} (y_i - C\hat{x}_{i|i-1})^T \mathcal{P}^{-1} (y_i - C\hat{x}_{i|i-1}) \tag{3.6}$$

where w is the size of the detection window and $\mathcal{P} = (CPC^T + R)$ is the co-variance of an independent and identically distributed Gaussian input signal from the sensors. The plant is considered not under attack if $g_k < \gamma$; otherwise, if $g_k \geq \gamma$, then the plant is considered to be under attack and the detector generates an alarm.

3.4 Reflective Mitigation of Attacks

Our proposed approach triggers an attack attenuation process for cyber-physical attacks (*i.e.*, disruptive attacks leading to system failures) to remain operational and provide system functionality. We assume a resilient system, capable of reacting and defending itself against known threats. Remediation starts right after attacks are detected. The system dynamically and autonomously changes its behavior to activate an attenuation plan that guarantees work continuation. This is carried out through the cooperation of the feedback controller and the network controller. Although these two controllers have different individual objectives and functionalities, they can work in a coordinated way in order to reach a common goal. Both controllers get connected and coordinate the resilience strategies, e.g., to maintain the resilient properties of the system under failure and attacks.

The proposed resilience strategies try to revert the adversary activity. This is done due to a system capable of modifying its configuration to introduce a new virtual component on-the-fly and dynamically reverting the adversary actions. The solution combines a feedback control technique to detect the attack, programmable reflection for creating the new virtual component that will help the affected feedback controller to bypass the attack and a programmable network in order to neutralize the adversary and sanitize the traffic. The complete process is composed of three main phases: (1) detection, (2) reflection and (3) traffic sanitization.

- **Phase 1 – Detection**. A feedback control detection mechanism is executed in the feedback controller. When an attack is detected, it alerts the network controller to start the coordination of the different components in the system.

  The mathematical model that allows controlling the physical process and detect deviations from the normal behavior, can also provide mechanisms to provide attack detection. For example, using physical watermarking that allows authenticating the correct operation of a control system using a challenge-response detector. In our solution, we used the approach explained in [9, 285] (cf. Section 3.3). To

authenticate the exchanged information, this solution injects a known noise in the physical system signals. It is expected that the effect of that noise is also present in the measured output due to the dynamics of the system. The added noise increases the difficulty associated with the learning process of the adversary. It becomes harder for the adversary to identify the system parameters, hence decreasing the chances of the adversary to correlate the proper input and output values.

- **Phase 2 – Reflection**. The feedback controller creates a reflective agent, which is executed within the domain of the network controller. The reflective agent has the control capabilities associated with the victims of the attack. It uses programmable reflection to create, at runtime, a component that executes the same program and equivalent interfaces as the feedback controller. By programmable networking reflection, we refer to the capability of the system to modify its networking behavior, *i.e.*, changing accordingly to what is required. For this reason, an on-demand process for loading and unloading components as services could be performed.
- **Phase 3 – Traffic Sanitization**. The forwarding elements using network programming capabilities allow performing a dynamic network traffic sanitization by modifying the packet containing malicious payloads. The packet affected by the adversary gets sanitized by the reflective agent, which determines what is the correct payload the packet should have. All the network actions required to sanitize the traffic are coordinated by the network controller.

The solution dynamically applies an attenuation technique using the forwarding devices to modify the traffic in order to revert the adversary actions. In a cyber-physical bias injection attack, the physical damage in the system occurs due to modified control commands injected into the plant actions. For this reason, after the traffic is modified by the adversary, the forwarding devices under the command of the network controller intercept those packets and modify them using the reflective agent that knows the physical model of the system and has the ability to determine whether those values in the packets are correct or not. In order to perform the calculations, the reflective agent uses as input the sensor measurements that the plant communicated to the feedback controller previously, since this component executes the same transmission function as the feedback controller, it can determine the correct command values without any model of normal behavior or historic data of the system. In addition, this node can monitor the measured values, since it is in the network control level and has the potential power to see all that is happening in the network. Moreover, since the network is k-resilient this node can be placed in the most convenient path between the plant and the feedback controller.

Figure 3.1(c) shows how our attenuation process works in order to handle the attack perpetrated by the adversary. The adversary modifies the command u_k sent from the

feedback controller to the plant in order to insert a fake command u'_k. After this, the traffic is modified again to sanitize it with the help of the reflective agent calculations that take as input the monitored sensor values y_k captured from the network. This way, the plant receives the correct u_k command and the physical process is not affected by the adversary actions. When the attack is finished, the normal operation of the system can be restored. The original controller can take over again.

To achieve this solution, the feedback controller is made of two sub-components: (a) the control component that is in charge of enforcing the dynamical control objectives (fast dynamics are involved); (b) the supervisory component that communicates in a bi-directional way with the network controller. At the data layer of the network domain, we have network probes and effectors, conducting data monitoring —if instructed by the control domain. Network probes monitor the traffic in the data domain and provide the information to the network controller.

The network controller, based on measurements provided by network probes and feedback provided by the feedback controller, is able to detect a possible threat acting on the control path. In response to such a threat, the reflective agent provides a corrective measure to attenuate the impact. The network controller can be seen as a computing entity that is located at an external location (e.g., a kind of Network Operating System [154]). For instance, it provides resources and abstractions to manage the system using a centralized or decentralized model [9].

Together, both controllers manage the data domain. The feedback controller manages the physical system through physical sensors and actuators deployed at the physical layer. The network controller estimates and manages the data domain through probes and effectors –deployed at the management and control domain.

The network controller analyzes the information and forwards control actions to the effectors. Network rules at the control domain are responsible for enforcing such actions. For instance, when a network probe finds tampered traffic in a network path, it provides the tampered information to the control domain. Then, the network controller, located at the control domain, checks for the available resources and helps in order to enforce the action.

3.5 Experimental Results

Experimental setup — We present in this section some experimental results to validate our approach. We use a physical SCADA testbed, for the generation of Modbus-driven CPS data [293]. The testbed consists of *Lego Mindstorms* EV3 bricks [294] and Raspberry Pi [295] boards that control some representative sensors (e.g., distance sensors) and actuators (e.g., dynamic speed accelerators).

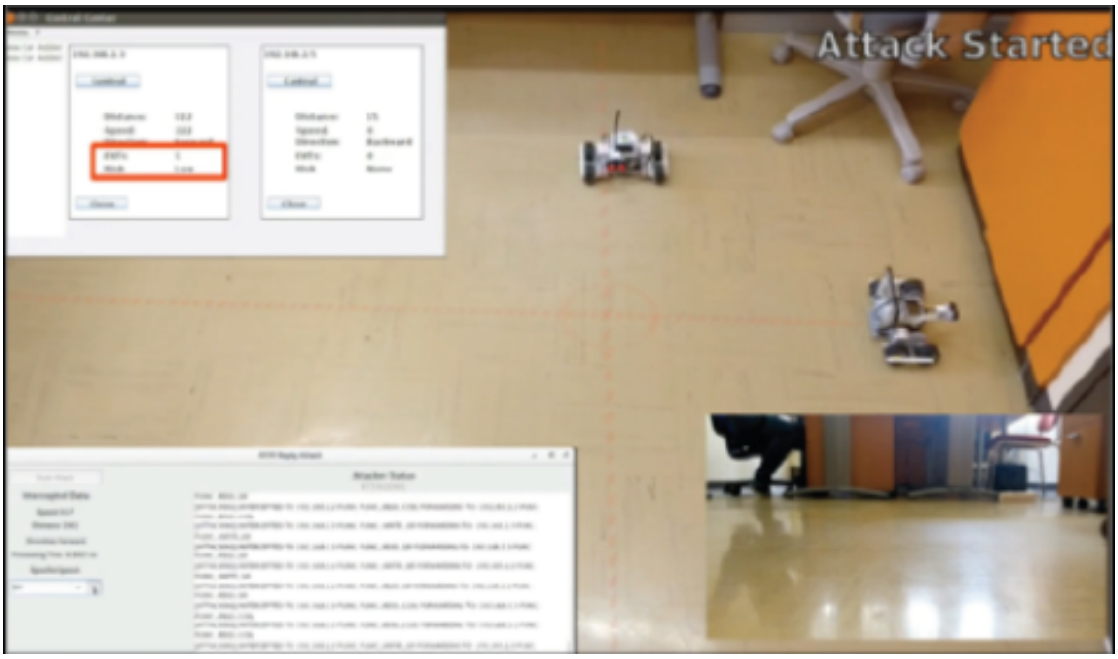
A sample picture of the testbed is showed in Figure 3.2(a). The Modbus SCADA protocol used in the testbed is based on standard Modbus protocol specifications [293]. The testbed implements a kinetic dynamics use case, in which two motion devices perform a deterministic path based on linear motion (backward and forward motion over a bounded square area). We refer the reader to [272] for additional information about this testbed.

Figure 3.2(b) depicts a numeric co-simulation complementing the same scenario, using the collected SCADA data to train a CPS programmable simulator. The implementation uses OMNeT++ (Objective Modular Network Testbed in C++) [259, 296] and leverages a series of shared APIs (Application Programming Interfaces) over the INET [297] and SCADASim [261] libraries, to enforce the use of the Modbus protocol over TCP and UDP traffic. All the components (both in the Lego SCADA testbed and the OMNeT++ co-simulation) are synchronized by feedback controllers. Every motion device has a distance sensor in the frontal part, to measure its relative distance to the boundaries of a unit square area. The distance is transmitted to the feedback controllers via Modbus SCADA messages. The feedback controller computes the relative velocity of each motion device, and the Euclidean distance between the two motion devices, in order to guarantee spatial collision-free operations.

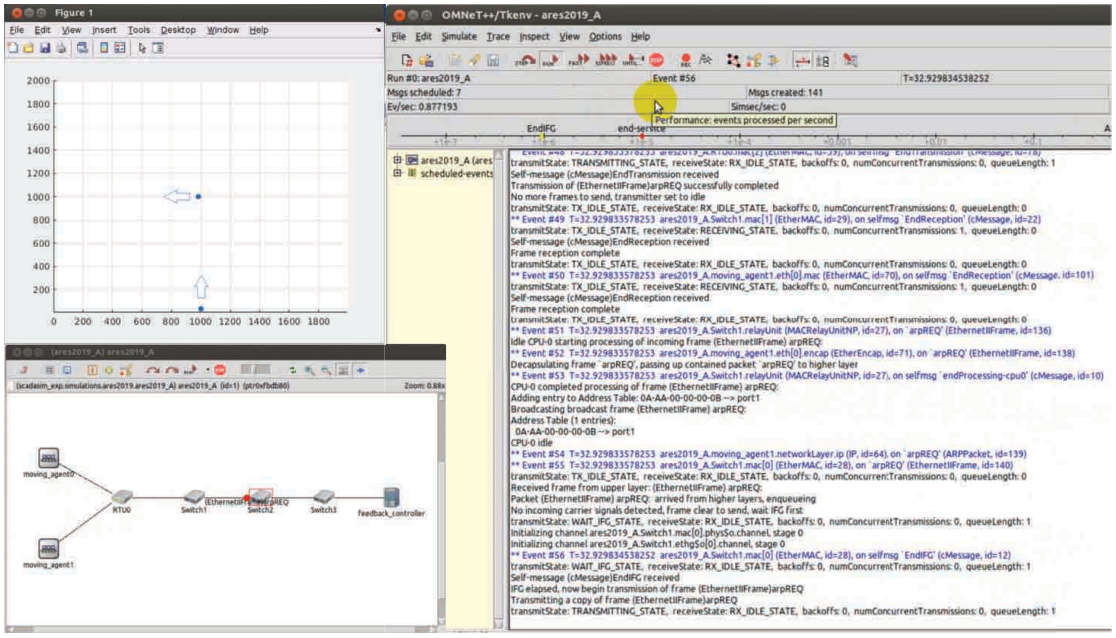
The goal of the adversary is to launch an attack at the control level to move the physical process to an undesirable state resulting in the physical collision of the two motion devices. Figure 3.3 shows the kinetic dynamics of the system during the nominal case (*i.e.*, absence of attacks, left-side); and during the attack (*i.e.*, the moment at which the adversary takes control over the system, right-side). Time is normalized between 0.0 and 1.0, representing the temporal percentage of multiple experimental runs. We can appreciate how the system moves to unstable states, disrupted by the adversary. Figure 3.4 shows the same scenario using a winding graph which is built with a polar representation. The winding graph uses the Fourier transform to turn the time-function signals in Figure 3.3 into a frequency representation. This way, we can verify the periodicity of the signals in the normal case and the disruption in case of an attack.

During the OMNeT++ co-simulation, we analyze the system behavior in the normal operation mode, under attack and using the proposed attenuation approach. In the testbed, the two motion devices follow a trajectory of two meters. The feedback controller coordinates the movement of the motion devices, by sending the relative velocity to the motion device, and receiving back the distance of the motion device to the spatial boundaries. The feedback controller sends a series of Modbus messages to the physical environment of the plant, through a network of traffic programmable forwarders (e.g., SDN switches). The plant contains the physical process itself, the distance sensors and the actuators that perform the commands (accelerators that increase or decrease the relative velocity of the two motion devices).

The adversary starts the cyber-physical attack by either tampering with the controller with fake sensor readings or modifying the control commands sent from the controller. With the OMNeT++ co-simulation, we evaluate the attenuation of the bias injection



(a) Lego Mindstorms Experimental Testbed.



(b) OMNeT++ CPS Co-Simulation.

Figure 3.2 – Evaluation platform. (a) Lego testbed for the generation of SCADA-driven CPS data. (b) CPS co-simulation implemented over OMNeT++

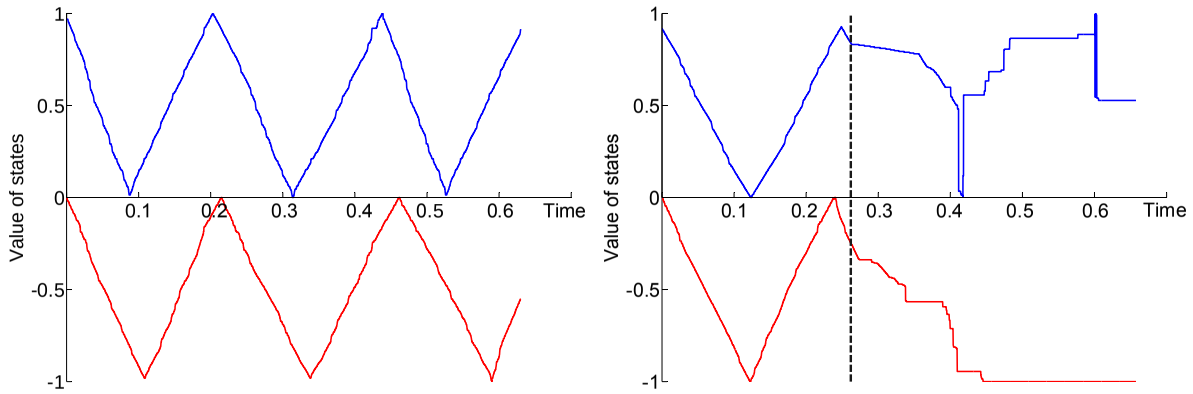


Figure 3.3 – Lego testbed results. Temporal representation of CPS kinetic dynamics associated to the two motion devices (left-side, nominal mode dynamics; right-side, dynamics during the attack). The dotted line represents the moment when an attack starts.

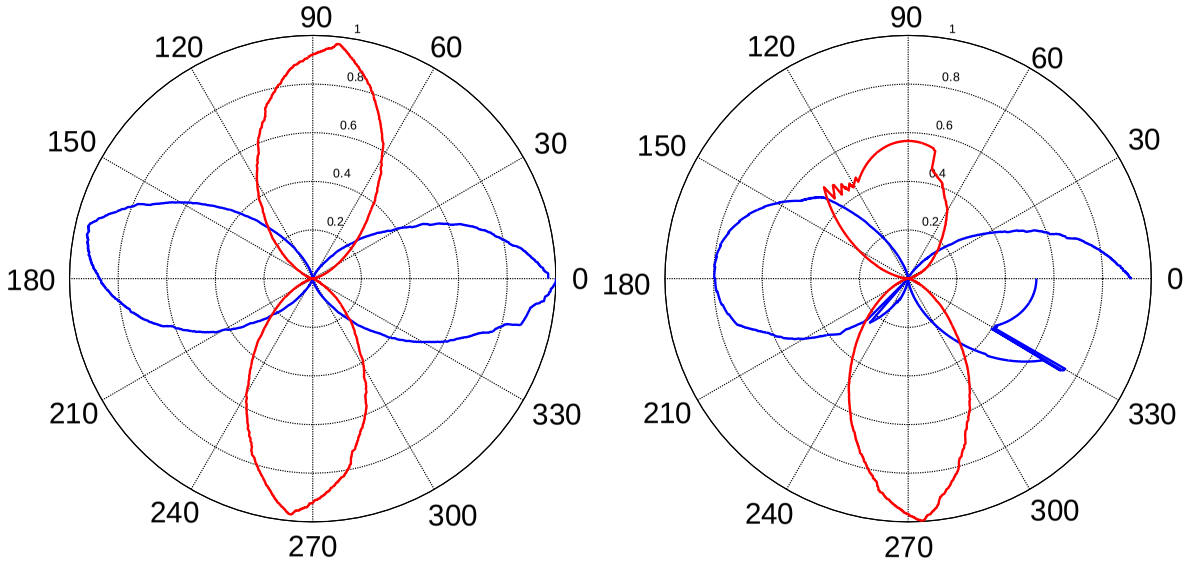


Figure 3.4 – Lego testbed results. Winding graph representation (left-side, nominal case; right-side, attack case.).

attack, *i.e.*, by forging tampered control commands from the controller to the plant. For simplicity reasons, we focus only on the physical part of the cyber-physical attack using the network to damage the system. In other words, we assume an adversary that already found a way to hack the cyber layer and gain remote access to the system.

Each co-simulation evaluates fifty Monte Carlo different runs. In addition, according to the sensor specification, the simulation considers a possible error of up to 1 cm w.r.t. the measured distance value. We also model the network delays using the probability distribution in [298]. Figure 3.5(a) shows the results obtained for the nominal case (*i.e.*, absence of attack), considering the aforementioned possible variation. The plots depict the average Euclidean distance, with 95% confidence intervals, between the motion devices in function of time. The horizontal axis of the plots in Figures 3.5(a–d) provides a normalized time between 0.0 and 1.0, representing the temporal percentage prior to concluding the simulation runs. The vertical axis of the plots in Figures 3.5(a–d) provides the Euclidean distance between the two motion devices, from 0 to 1400 cm. Some further evaluation details are discussed below.

Results — During the perpetration of the attacks, the adversary performs a bias injection of cyber-physical data. The adversary uses the network to modify the exchanged packets between the feedback controller and the plant. We assume an adversary recording and learning the system dynamics from commands and sensor outputs. The adversary performs an initial learning phase, in order to eavesdrop on the data and infer the system dynamics, *i.e.*, the same one used by the feedback controller to guarantee the stability of the system, showed as the nominal case in Figure 3.5(a).

Let u_k be a feedback controller command sent to the actuator of a motion device at time k. Let u_k^{act} be the command received by the actuator at time k, where $0 \leq k \leq T_s$ and T_s be the full duration of each simulation run. The attack interval T_a is limited to the simulation time T_s, as summarized next:

$$u_k^{act} = \begin{cases} u_k & \text{if } k \notin T_a \\ u'_k & \text{if } k \in T_a \end{cases}$$

For our evaluation, we compare two types of adversaries according to the bias injected into the payload of the packets, i.e, according to the difference between the value u'_k injected by the adversary and the real value u_k sent by the controller. This way, we define two adversary models: an *aggressive adversary* and a *non-aggressive adversary*. The aggressive adversary injects in u'_k a bigger difference with respect to the correct command u_k sent by the feedback controller compared to the non-aggressive adversary. In consequence, an aggressive adversary will make the system move faster from its nominal state. Figure 3.5(b) shows the results obtained for the two attack scenarios.

The feedback controller loses its control over the system, while the adversary forces the spatial collision of the two motion devices.

During the attenuation process, the system reacts using reflective programmable networking. The reflective agent takes control of the situation, after a hangover of the feedback controller functionality (which moves to the programmable controller domain). This reflective agent takes control over the adversary communications and neutralizes the attack. For each of the defined adversaries, we simulate two scenarios using different values for the time the solution starts working. This is a parameter of the simulation that depends mainly on the time required for the detection mechanism to detect the attack plus the time required to set up and coordinate all the components working in the approach. Figures 3.5(c)–(d) show how the approach guarantees the controllability property. The first vertical dotted line shows the moment when the attack starts and the second vertical dotted line shows the moment when the technique starts. It is possible to appreciate that the adversary introduces a perturbation in the system. As a consequence, the Euclidean distance between the two motion devices starts to oscillate out of the expected behavior (w.r.t. Figure 3.5(a)).

When the attack is detected, the technique starts working and the reflective agent starts sanitizing the control commands to the moving agents to restore the nominal behavior of the system. Figures 3.6(a–b) show the winding graph of the motion devices under the approach. The attacked device corresponds to the vertically oriented ellipses. It is possible to observe some perturbations, due to the modifications introduced by the reflective agent when thwarting the adversary actions and recover the stability of the process. As a result, the spatial collision between the two devices is avoided and the system keeps working. Notice that the technique takes control of the physical environment in order to conduct the physical environment from an unstable behavior generated by the attack to a stable and safe behavior, converging to the normal behavior of the physical environment. Figures 3.5(c–d) show that the approach neutralizes the effects of the attack right after a short period of instability. The approach does not eliminate the adversary. However, it contains the effects and reorients the system to the nominal case.

We argue that the solution is reflective since it creates a dynamic component at runtime to help with the function of the attacked control loops. In addition, the component is reflected in the network domain. It gets a greater control of the network than the victim component which has only the possibility to communicate through the network data plane. This is an advantage of the approach compared with other techniques such as redundancy, which implies having a copy of the same component as a backup. In that case, the adversary may move the attack to the redundant component. For this same reason, routing-based mitigation techniques are not sufficient since the system may find an alternative route but the adversary may move to the new paths. Other solutions that implement mitigation at the node level, such as diversity, are not enough either since

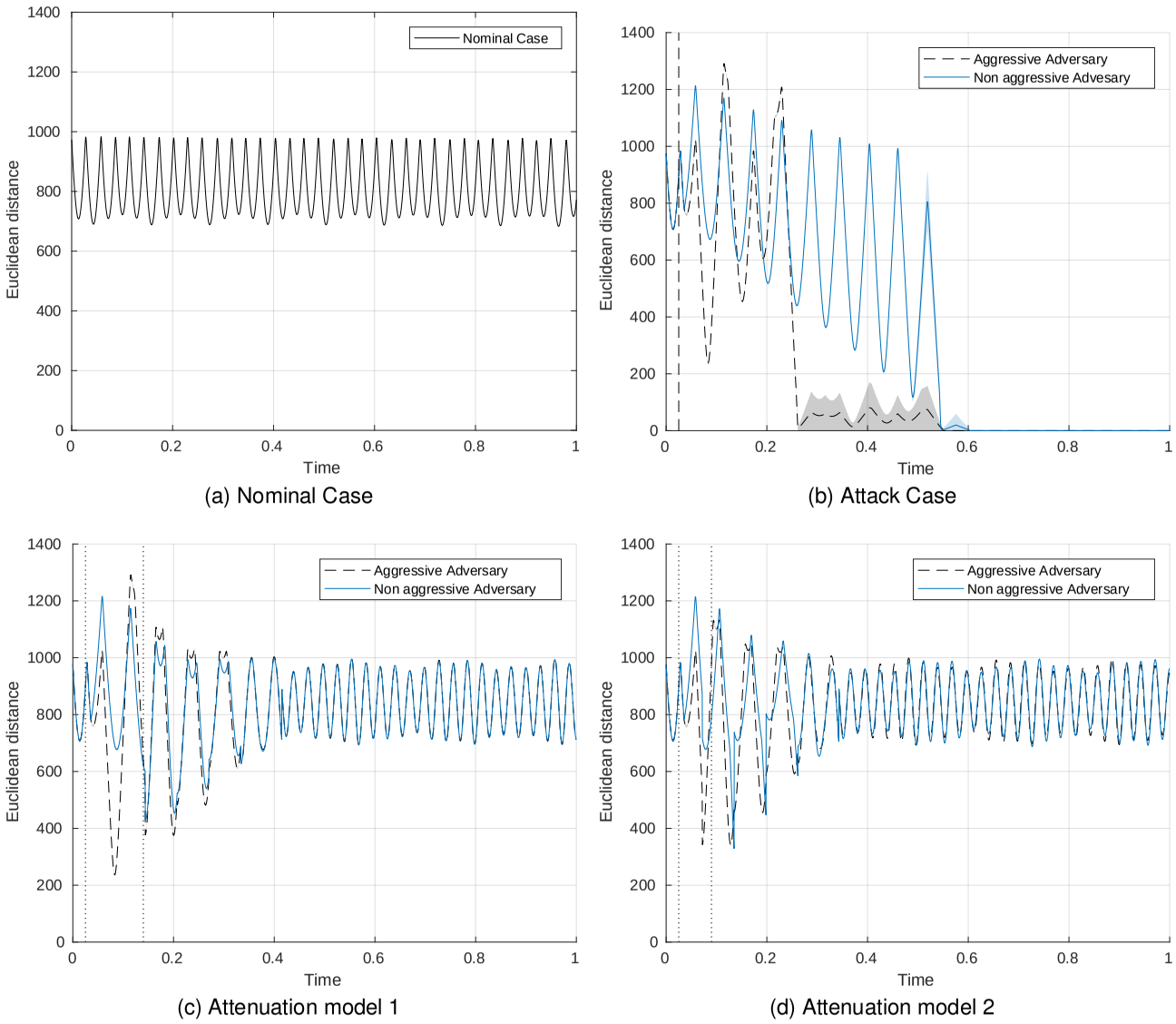
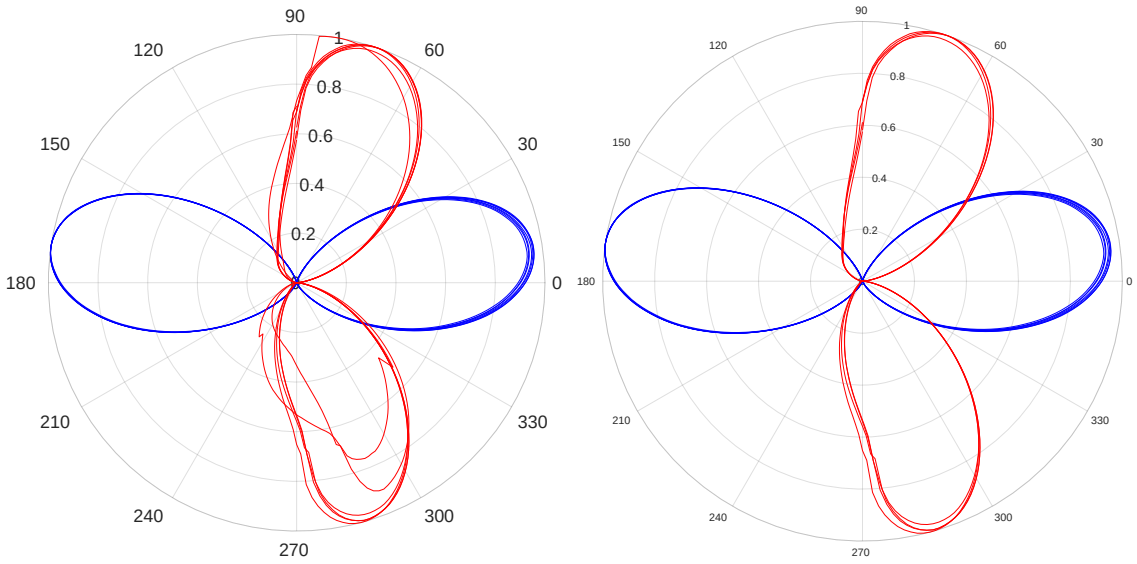


Figure 3.5 – OMNeT++ results. (a–b) Euclidean distance (with 95% confidence intervals), nominal and attack simulations. (c–d) Euclidean distance, attenuation of two different remediation starting time models.



(a) Winding graph of dynamics cycles (attenuation 1)

(b) Winding graph of dynamics cycles (attenuation 2)

Figure 3.6 – OMNeT++ results. Winding graphs for the same attenuation models.

the adversary uses the network to perform the malicious activity. Cyber mechanisms to detect packet injection, such as a Message Authentication Code (MAC), cannot mitigate this kind of attack. Although they allow dropping modified messages, fail at satisfying real-time constraints. In our approach, the network itself is containing the adversary to revert its actions.

3.6 Discussion

The proposed approach showed to be efficient to recover the system from an attack. However, it had some limitations. First, it considers the case when the adversary is in the control LAN and the MTU server sends commands to the plant in the remote location. Hence, the approach is not applicable in the cases where the adversary is the sensors, actuators or the controller itself. In addition, the local RTUs and PCLs may also send command. For this reason, we wanted to improve the approach to consider these cases too and propose a more comprehensive solution.

In addition, the proposed approach works in a detection-reaction manner, but our original objective in this dissertation was to build an approach focused on resilience. We wanted an approach capable of recovering the system without triggering any additional behavior. Our objective was to design a system that without using a detection mechanism has the ability to restore the functions of the system by turning into useless the knowledge that the adversary may have gathered from the system.

To overcome all these limitations, in the next chapter, we propose a Moving Target Defense approach that generates a time-periodic variation in the network and in the control model matrices used to control the physical process. This way, the knowledge that the adversary got from the system will be changed after this period time. For this reason, it is the system design that heals itself without triggering any additional behavior. Also, it considers adversaries situated in the control LAN as well as adversaries in remote locations.

3.7 Summary

In this chapter, we have focused on designing a reaction approach to attenuate the effect of a cyber-physical adversary in CPS.

We consider an adversary that injects malicious traffic in the network and is able to acquire knowledge about the system dynamics prior to starting the attack and successfully get control over the commands and measurements of the system.

To build the approach, we have used programmable networks and software reflection to sanitize the malicious false data injected into the network. New controllers are created on-the-fly in the network domain and the forwarding devices repair the traffic with the help of the new CPS controllers.

We have validated the approach by simulating a cyber-physical system with sensors and actuators. We also discussed the limitations of the approach in terms of adversary and network assumptions as well as the need to use a resilience strategy. We have also analyzed strategies to overcome the limitation. For this reason, in the next chapter, we propose a new approach to apply our new strategies and improve the obtained results with a resilient approach.

4 Resilient Moving-Target Paradigm

4.1 Introduction

CPS attacks are difficult to trace, classify or identify the original threat, which may move or spread, and target multiple components of the system. For this reason, research on cyber-physical attacks and secure control has found increasing interest [16].

The implementation of resilience methods aims at ensuring essential operations, reducing potential damages, maintaining critical functions level and rapid recovery. Resilient CPS are expected to keep an acceptable performance, even in the case of faults, disruptions or attacks. This refers to the ability to ensure that system outputs are correct, within acceptable operating thresholds and the normal operation can be restored despite local faults or attacks.

Resilience-by-design approaches assume the incorporation of resilience against such attacks since the initial conception of the system. Assuring that a system is resilient to cyber-physical attacks is a non-trivial task, since the most natural conception of protection is the detection-reaction base design. In this case, we detect anomalous behavior and trigger additional functionalities. This approach has some limitations. Firstly, the detection approaches are susceptible to false positives and false negatives. Second, it is required that the reaction approach does not interfere with the safety shut down the system of a CPS. Finally, new triggered behaviors should ensure that the system keeps working safely and does not produce any adversary effect in all the possible scenarios, such as considering any adversary action or in case of false negatives.

Traditionally, CPS remain unchanged during long periods. For this reason, they become vulnerable to adversaries who can gather data and use their precise knowledge of the system dynamics, communications and control to damage the system. Moving Target Defense (MTD) mechanisms have emerged as a strategy to add uncertainty about the

state and execution of a system to prevent in a proactive and reactive mode the insider adversaries [200].

For this reason, in this chapter, we focus on building a resilient design. The approach is based on a MTD mechanism using physical model mutations and network reconfiguration. The model mutations use a switched control technique that allows the system to change its design periodically.

4.2 Contributions

The existing network MTD approaches are mainly focused on common Internet applications and may not be suitable for CPS real-time applications. Node MTD approaches are useful to face adversaries that target the platform or software running in a host. The main issue in CPS are adversaries that modify the network traffic. Also, existing CPS-based MTD approaches aim at detecting or mitigating attacks, but they do not offer a resilience solution that allows a system to self-heal from adversaries.

As showed in recent surveys [17, 33, 36], most of the existing approaches require adding extra hardware [205], which may be expensive. Other solutions use detection approaches with recovery strategies [40, 206, 210, 299] that usually use state estimation to maintain an understanding of the system state under attack, even when a subset of inputs and outputs are compromised. These techniques work as traditional detection and mitigation approaches but do not provide resilience-by-design or prevent the execution of malicious commands. Having a reliable estimate allows a defender to better understand the portions of a system that have been compromised and design attack-specific solutions to counter the adversary actions. In addition, these approaches require to include also a mechanism to ensure the correct feedback control after detecting the attack.

In critical CPS, there is a control system that takes action over the physical process and a safety system that reacts to shut down in a safe way when the control system is not working properly. For safety reasons, it is not advisable to create reaction or mitigation mechanisms that may interfere with this safety shutdown.

In this context, we propose an approach that provides resilience-by-design that does not require any detection or mitigation mechanism to work since the system itself is capable of repairing the adversary damage caused by introducing malicious traffic. The proposed system design applies a distributed network and node MTD approach for CPS based on modifying the physical model of each node, *i.e.*, modifying the transfer function that they execute in a coordinated manner that allows facing network adversaries while the globally distributed transfer function of the system remains unchanged.

In this chapter, we focus on resilience via a MTD approach [200], using physical model mutations and network reconfiguration. The proposed approach builds a resilient-by-

design system using a switched control. This technique allows the system to change its design periodically. In this way, the proposed approach allows the system to self-heal by design without any additional detection or reaction mechanism that identifies or mitigates threats rather than the traditional safety system.

Our main contributions are as follows: (1) an approach to build resilient cyber-physical systems capable of ensuring close-loop stability in presence of cyber-physical adversaries and (2) an experimental work that validates the approach via simulation. The approach is innovative since it does not require a detection and reaction mechanism as in the existing literature. The system has the capability of self-healing due to its design, using a collaborative control system with mutating control laws. The network and physical process controllers collaborate to improve the resilience of the system. Parts of the contributions explained in this chapter were published in [209].

The outline of this chapter is summarized as follows. Section 4.3 presents the problem formulation. Section 4.4 presents our moving-target approach and explains the steps to design a resilient design. Section 4.5 presents the experimental work to validate the proposal. Finally, Section 4.6 discussed the obtained results and Section 4.7 summarizes this chapter.

4.3 Problem Formulation

We consider a discrete linear time-invariant (LTI) modeling of the physical processes as described in Chapter 2, Section 2.1.2. Notice that the physical process does not need to be necessarily linear. Non-linear physical processes are usually linearized using well-known techniques as explained in Chapter 2, Section 2.1.2. Likewise, previous work has already showed that, from a security standpoint, adversaries can attack and hide better when systems are modeled as LTI. For instance, since the degree of the polynomial description associated with the physical process is usually higher when systems are modeled as LTI, the number of points available to an adversary to attack and hide is also higher [40]. Hence, it is assumed that security solutions that are valid under the LTI assumption, are also valid under non-LTI assumptions since the non-LTI case is less favorable to the adversary. In other words, by addressing the LTI case, our work tackles the less favorable case for security, rather than the easiest case for the defender.

We assume realizable networked systems, whose physical processes are proper, causal and stable. We assume infrastructure environments that are connected using programmable networks via, e.g., Software Defined Network (SDN) technologies [157]. We also assume that there is secure management of SDN controllers and switches, to synchronize operational and security parameters [286].

The objective of the adversary is to cause a malfunction in the system by modifying the network traffic and affecting the control system. To achieve this, the adversary corrupts the system inputs and outputs that are sent using the data network. In particular, the most powerful adversary is the cyber-physical adversary mentioned in Chapter 2, Section 2.2 because of the ability to estimate the system parameters, *i.e.*, the adversary learns the system dynamics. For example, using techniques such as machine learning, ARX (*autoregressive with exogenous input*) or ARMAX (*autoregressive-moving average with exogenous input*) models. The system model working under the effect of this adversary can be modeled mathematically as:

$$x'_{k+1} = Ax_k + Bu'_k \tag{4.1}$$

where u'_k represents an attack to the control input, *i.e.*, in the commands sent from the controller to the actuators. In addition, this adversary can inject specific malicious measurements designed to deceive the control system:

$$y'_k = C'x_k \tag{4.2}$$

where C' represents an adversary that is able to create a sensor output y'_k that is correlated with the real u_k control input sent by the controller. This means, that the adversary is capable of sending a sensor output according to the system state x_k that the controller is expecting to receive. This attack is designed to mislead the system or destabilize its physical processes. The adversary aims at evading detection, by hiding the actions as faults or errors, whose random nature is much easier to be identified and corrected. The closer the matrices A, B and C that the adversary learned are to the real matrices in the controller, the more difficult is to detect the adversary.

Also, the adversary is assumed to be placed in a remote location but gained access to the internal network by exploiting cyber vulnerabilities. The adversary uses the network traffic to perform the attacks, as an insider. In addition, the adversary is able to change positions in the network. The adversary performs malicious actions in the data layer of the network domain. This means that the adversary is not attacking the SDN plane itself, e.g., the SDN control layer; but the data traffic that is flowing through the SDN network.

4.4 Switched-based Resilient Control

In this approach, we propose to take as an input a CPS modeled by a transfer function and build a resilient equivalent system capable of controlling the same physical process using a Switched Linear Control System as described in Chapter 2, Section 2.1.2.

A switched system consists of a finite number of subsystems and a logical rule that orchestrates the switching between the subsystems. It may be modeled as follows:

$$x_{k+1} = f_{\sigma(k)}(x_k, u_k) \tag{4.3}$$

where $k \in \mathbb{Z}^+$ is the time interval, $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^p$ is the control input and σ is the logical rule that orchestrates the switching between the subsystems. It means that σ is a function $\sigma : \mathbb{Z}^+ \rightarrow \mathcal{I}$, where $\mathcal{I} = \{1, ..., N\}$ contains the indexes of the subsystems. A subsystem is determined by a pair $(\mathcal{M}_i,\ \mathcal{G}_i)$ where $\mathcal{M}_i = \{A_i, B_i, C_i : i \in \mathcal{I}\}$ is the set of physical system models and $\mathcal{G}_i = \{V_i, E_i : i \in \mathcal{I}\}$ is the set of graphs that represent the network connections in the CPS. Hence, σ define a piece-wise switching signal that is a time-varying definition of the process model and the network graph that is activated at time k. The physical model activated at time k is then defined by Equation (4.4) as follows:

$$\begin{aligned} x_{k+1} &= A_{\sigma(k)}x_k + B_{\sigma(k)}u_k \\ y_k &= C_{\sigma(k)}x_k \end{aligned} \tag{4.4}$$

whose system communicates through a network determined by the connectivity graph $\mathcal{G}_{\sigma(k)} = [V_{\sigma(k)}, E_{\sigma(k)}]$. The approach aims at protecting the system from network adversaries working at the node level by modifying the controller model and at the network layers modifying the endpoint information. In the sequel, we provide a procedure to build a resilient system.

Step 1 (Models Design): In this section, we analyze how to design the physical models in the subsystems, *i.e.*, how to create the subset of matrices $\mathcal{M}_i = \{A_i, B_i, C_i : i \in \mathcal{I}\}$ that will be activated at each time period.

There are two mechanisms to design equivalent control systems capable of controlling the same physical process. One possibility is to have redundant sensors and actuators, as proposed in [205]. This mechanism requires adding extra hardware to the system. So, the controller can choose at each time period which one to activate.

The approach we propose is to design distributed controllers that modify in time the physical model they use for the feedback. The overall process is controlled by several independent controllers and altogether represent a decentralized controller, *i.e.*, if at time k, it is activated the control model with matrices A_i, B_i, C_i then there will be j controllers with $j \in 1...o$ and each controller will use a set of matrices A_{ij}, B_{ij}, C_{ij} where $A_i = \bigcup_{j=1}^{o} A_{ij}$, $B_i = \bigcup_{j=1}^{o} B_{ij}$ and $C_i = \bigcup_{j=1}^{o} C_{ij}$. Hence, the controllers have available only parts of the overall information.

In the sequel, we analyze how to derive the equivalent models starting from the initial transfer function as represented in Equation (2.1). The objective is to obtain different models expressed in the A_{ij}, B_{ij}, C_{ij} matrices which can be combined to represent the system dynamics as in Equation (5.1) and it allows deriving different sets of controllers capable of controlling the physical process.

Step 1.1: To obtain the equivalent representation we will factorize the matrices applying techniques similar to the ones used by the different approaches for decentralized control design [61, 66]. It consists of combining a diagonal controller $Q(s)$ with a block compensator $D(s)$ in such a way that the controller perceives the process dynamics $G(s)$ as a set of independent processes as showed in Equation (4.5):

$$G(s) \cdot D(s) = Q(s) \tag{4.5}$$

where $D(s)$ and $Q(s)$ are both $n \times n$ matrices of transfer functions, $Q(s)$ is diagonal and $D(s)$ invertible. Hence, the structure of the distributed controllers will be formed for n controllers executing the Q_{ii} transfer functions and each of these controllers is connected with n controllers executing the D_{ij} transfer function. In Figure 4.1(a), we show the structure for a 2×2 example.

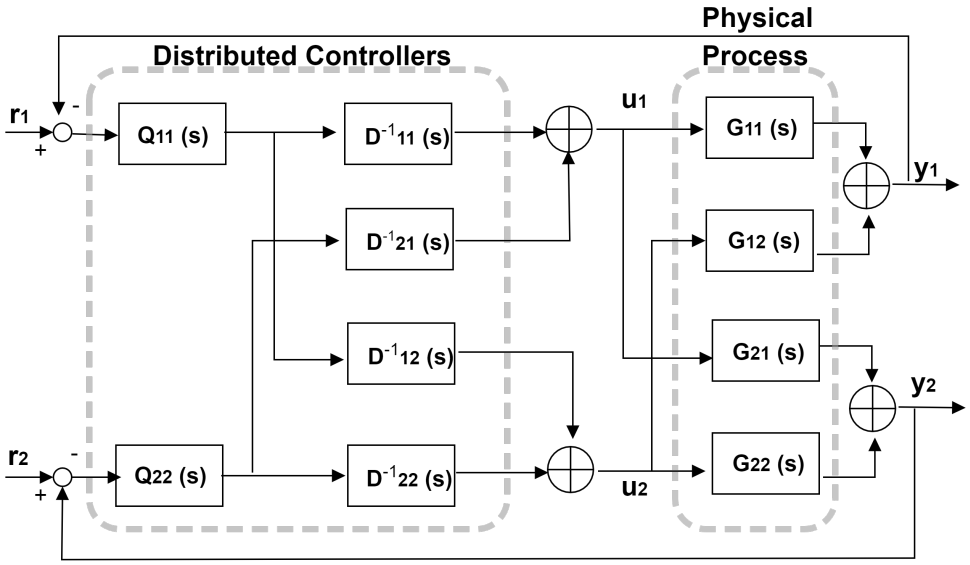
To create this distributed design, the first step is to calculate $adjG(s)$ the adjudged matrix of G which is the transposition of the co-factor matrix of G.

Step 1.2: We build matrix $D(s)$ as follows. For each column $\hat{J} = \{1, .., N\}$, we select a row $\hat{I}$ to set that element $d_{\hat{I}\hat{J}}$ in the matrix $D(s)$ to unity. It is necessary to choose one for each column but not necessarily the diagonal ones.

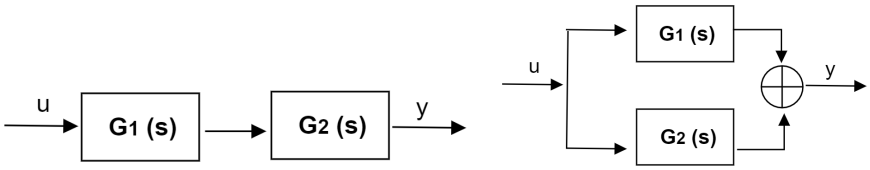
After choosing the elements $(\hat{I}, \hat{J})$ to be set to one, the matrix $D(s)$ can be completed as follows:

$$d_{i\hat{J}} = \frac{adjG_{i\hat{J}}}{adjG_{\hat{I}\hat{J}}}$$

where $adjG_{i\hat{J}}$ is the $(i, \hat{J})$ element of $adjG(s)$ the adjugate matrix of G.



(a) Decentralized models



(b) Serial decomposition (c) Parallel decomposition

Figure 4.1 – Decentralized Resilient Design Architecture.

This means that for each column in the matrix, $\hat{J}$ is fixed and it corresponds to the column where the value was set to one previously. In addition, i varies from $1, .., N$ with $i \neq \hat{I}$. Hence, each element $d_{i\hat{j}}$ is obtained from dividing the element $(i, \hat{J})$ in the $adjG(s)$ matrix between the value in the position $(\hat{I}, \hat{J})$ of the matrix $adjG(s)$.

We have to repeat this process for each column by fixing a new $\hat{J}$ to obtain the complete matrix $D(s)$ corresponding to one single model.

After we obtained the complete matrix $D(s)$, we repeat the whole process by selecting a different row $\hat{I}$ to obtain another model different from the previous one.

Hence, for an $n \times n$ process, there are n^n possible choices of $\hat{I}$ and $\hat{J}$. So, there are n^n possible $D(s)$ since it depends on the possible positions to place the $1s$ values when building matrix $D(s)$.

However, some of those choices can result in non-realizable systems. For example, if the adjudged matrix has a zero value in that entry. Thus, the configuration can be selected depending on the realizability.

Step 1.3: $Q(s)$ is a diagonal matrix built using Equation (4.5) and multiplying $G(s) \cdot D(s)$. Each matrix $D(s)$ gives, as a result, a different matrix $Q(s)$.

In Figure 4.1(a), we define the representation of the controllers' architecture based on the defined matrix $Q(s)$ and $D^{-1}(s)$. Since we want to control the physical process defined by $G(s)$, the controllers will execute $Q(s)$ and $D^{-1}(s)$ due to Equation (4.5). Each entry of these matrices is the transfer function of one controller represented in the figure. Since Matrix Q(s) is a diagonal matrix, we have two controllers Q_{11} and Q_{22} that execute the transfer function in positions (1,1) and (2,2) of matrix $Q(s)$. Then the output of these controllers Q_{ii} goes to controllers D_{ij}^{-1}. It corresponds with the product of matrices $Q(s).D^{-1}(s)$ since each element Q_{ii} multiplies row i in $D^{-1}(s)$ as follows.

$$\begin{bmatrix} q_{11} & 0 \\ 0 & g_{22} \end{bmatrix} \cdot \begin{bmatrix} d_{11}^{-1} & d_{12}^{-1} \\ d_{21}^{-1} & d_{22}^{-1} \end{bmatrix} = \begin{bmatrix} q_{11}d_{11}^{-1} & q_{11}d_{12}^{-1} \\ q_{22}d_{21}^{-1} & q_{22}d_{22}^{-1} \end{bmatrix}$$

In addition, considering Equation (2.1), we have that $G(s).u = Q(s).D^{-1}(s).u = y$. Hence, we have the following equalities:

$$\begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} q_{11}d_{11}^{-1} & q_{11}d_{12}^{-1} \\ q_{22}d_{21}^{-1} & q_{22}d_{22}^{-1} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

The products of transfer functions are controllers in series which corresponds to a representation as in Figure 4.1(b). In the previous equality q_{11} and d_{11}^{-1} are multiplied. Hence, in Figure 4.1, there are controllers in series.

The sums of the transfer function are parallel controllers which correspond to a representation as in Figure 4.1(c). For example, according to the previous equalities, we have the following result.

$$\begin{bmatrix} q_{11}d_{11}^{-1}u_1 + q_{11}d_{12}^{-1}u_2 \\ q_{22}d_{21}^{-1}u_1 + q_{22}d_{22}^{-1}u_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

For that reason, y_1 is expressed as the sum of two components that came from serial controllers.

As a result, Figure 4.1 provides the architecture of the designed system which is correlated with the physical models design (its transfer functions) and the network

design, *i.e.*, Q_{11} will communicate with D_{11}^{-1} and D_{12}^{-1}. But, it will not communicate for example with Q_{22}.

Step 1.4: Due to realizability restrictions, it is possible to have matrices D with many elements equal to 0, which reduces the number of possible generated models. In this case, it is possible to generate other equivalent models using transfer function decomposition techniques.

Step 1.4.1 (Serial Decomposition): A transfer function $G(s)$ may be decomposed in transfer functions that multiply together as showed in Figure 4.1(b). Hence, $G(s) = G_1(s).G_2(s)$. This decomposition is commutative and it is possible to generate combinations of the different factors to create the distributed transfer functions. This can be applied at the level of transfer functions as well as factoring the original transfer function in its poles and zeros representation as follows:

$$G(s) = k \prod_{i=1}^{N} \frac{s - z_i}{s - p_i} \tag{4.6}$$

where the denominator coefficients p_i are the poles, the numerator z_i are the zeros of the transfer function and k is the gain term. This mechanism allows generating different partitions of matrices $Q(s)$ and $D(s)$.

Step 1.4.2 (Parallel Decomposition): In this case, the transfer function $G(s)$ is decomposed into a sum of terms as showed in Figure 4.1(c). Hence, $G(s) = G_1(s) + G_2(s)$. This can be done with a technique called partial fraction decomposition that finds the residues and poles. The terms are as follows:

$$G(s) = k + \sum_{i=1}^{N} \frac{r_i}{s - p_i} \tag{4.7}$$

where the denominator coefficients p_i are called the poles of the transfer function, the numerator r_i is the residue of pole p_i and k is a constant. Hence, after applying this technique to a d_{ij} transfer function, we will obtain a family of d_{ij}^t functions that can be added to obtain the original d_{ij} function. The super index ^{t} indicates de $1..N$ transfer functions that decompose d_{ij}. The corresponding architecture is showed in Figure 4.1 (c). This mechanism allows generating a different distribution of compensator matrices $D(s)$.

To provide more misleading information to the adversary, one may add deceiving controllers that include more variability and mimic a real controller but they execute a transfer function that is compensated by the action of another controller.

Step 1.5: After calculating the sets of matrices $D(s)$ and $Q(s)$, it is possible to take each d_{ij} and q_{ij} entry to calculate its corresponding matrices A, B and C using the procedure to transform a transfer function into a state-space model.

The obtained matrices for each d_{ij}, will be called $A_{D_{ij}}$, $B_{D_{ij}}$ and $C_{D_{ij}}$. In a similar way, it is possible to take the q_{ii} values in $Q(s)$ and calculate its corresponding matrices A, B and C to obtain the matrices $A_{Q_{ij}}$, $B_{Q_{ij}}$ and $C_{Q_{ij}}$.

Step 2 (Network Design): In this section, we analyze how to design the network connectivity graph $\mathcal{G} = [V, E]$ for each of the physical models created in Step 1.

Step 2.1: The transfer functions in $Q(s)$ are controllers that take one input and send one output. Each of them will be executed in one node. For notation, if a node v_q executes the controller q_{ii} then we will call it $v_{q_{ii}}$.

The d_{ij}^{-1} and d_{ij}^{-1t} elements take the output of the q_{jj} element to make their calculations and produce an output control signal. Each d_{ij}^{-1} will be executed in one node v_d and the notation will be $v_{d_{ij}}$ to express that the node v_d executes the transfer function d_{ij}^{-1}.

The network contains also a set of sensor nodes v_s and a set of actuator nodes v_a. If the sensor measures the variables of G_{ij}, then the notation will be v_{s_i}. In a similar way, v_{a_j} represents the actuator that applies the control input j.

Hence, the set of nodes V in graph $\mathcal{G}$ contains the nodes v_q, v_d, v_s and v_a. In the system, there are also network devices, such as routers and switches. However, we are not explicitly including them in the design as we assume a traditional use of them.

Step 2.2: The set of edges E will be defined from the matrices $D(s)$ and $Q(s)$ according to the following four main rules: (1) $(v_{q_{ii}}, v_{d_{ij}}) \in E$; (2) $(v_{d_{ij}}, v_{a_i}) \in E$; (3) $(v_{d_{ij}^t}, v_{a_i}) \in E$; (4) $(v_{s_i}, v_{q_{ii}}) \in E$. An example can be observed in Figure 4.1(a) where according to rule (1) the component q_{11} is connected to d_{11}^{-1} and d_{12}^{-1}. In addition, the output of q_{22} should be sent to d_{21}^{-1} and d_{22}^{-1}. Due to rule (2), the output of components d_{11}^{-1} and d_{21}^{-1} are combined to create the command u_1 that should be received by actuator a_1. In a similar manner, it is created the command for actuator a_2. Rule (3) is equivalent to rule (2) when parallel decomposition is applied. In this particular case, it does not apply. Finally, rule (4) indicates that the sensor s_1 and s_2 measure the data that should be sent to components q_{11} and q_{22} respectively.

Step 2.3: To coordinate the system, there will be an orchestrator, physically located in the SDN controller. The responsibilities of the orchestrator are described as follows:

1. **Choosing a key for the model selection**. There are $\mathcal{I} = \{1, ..., N\}$ possible subsystems to activate and the orchestrator chooses randomly a key K_1 which

will be used to select the next model to activate using a hash function as follows $hash(K_1, j)\ mod\ N$ where j is the switching interval. The common sharing of K_1, j and N allows each device to compute the next active model in a distributed manner. The key is renewed periodically using one of the existing approaches for key generation and distribution such as [300].

2. **Coordinating the network configuration transformation.** Each component will change its network configuration in each switching period of the physical model. To do this, each device gets a real IP address (RIPA) and a virtual IP address (VIPA). The RIPA is used for management purposes making the network configuration transformation transparent to administrators. The VIPA is used to communicate the data packets of the CPS, *i.e.*, the hosts communicate with another host using their VIPAs. In addition, VIPAs change periodically and synchronously in a distributed fashion over time. In every transformation interval, the hosts will be associated with a unique VIPA.

   The VIPA transformation is managed by the SDN devices by selecting an address from the unused address space. Each host will be allocated an IP address ranges to choose the VIPAs and they are selected using a hash function from the designated ranges. Since the VIPAs are chosen from the assigned network sub-nets, there is no need to do a routing update advertisement for internal routers. In addition, SDN devices will forward packets from old connections until the session is terminated or expired.

   Each SDN device is responsible for the management of the hosts in one or more sub-nets. The VIPAs selection is done in a similar way to the physical model selection. It uses a hash function and a secret random key to guarantee unpredictability. If there are p available VIPAs for a host, then the SDN device can compute the index of the VIPA for the switching interval j as $hash(K_2, j)\ mod\ p$. The SDN controller is responsible for the management of the SDN devices and the key K_2 distribution.

3. **Coordinating the transformation time**. The orchestrator has to choose and coordinate the switching in a master-slave mode. It requires a distributed timing synchronization that ensures the achievement and maintenance of a common time for all the nodes of the network. Many proposals have already work on solving this type of issue [301–304].

Step 3 (Switching Function Design): Next, it is required to design the switching function σ which indicates when to change the activated subsystem. In this step, we demonstrate that from the physical point of view, it is possible to use an unrestricted switching signal, this means that there is no minimum switching time required since the proposed subsystem share a Common Quadratic Lyapunov function by design. Hence, this ensures the stability of the proposed switched linear system.

The stability of switched systems depends not only on the dynamics of each subsystem but also on the properties of the switching signals.

For example, even when all the subsystems are stable, the switched system may have divergent trajectories for certain switching signals. In addition, it may be possible to switch between unstable subsystems to make the resulting switched system stable [74]. If one stays at stable subsystems long enough and switches less frequently, one may trade off the energy increase caused by the switching itself or the unstable modes, and maintain the stability of the system.

The switching function may depend on different parameters, such as the time instant k, the current state x_k, the output y_k or the previous active mode $\sigma(\tau)$ for $\tau < k$. However, during an attack, the state or the system output that a controller gets, may not be accurate with respect to the real state in the physical process. For this reason, it is desired that the switching function depends only on the time instant k.

There are many approaches to analyze the stability of a system. In particular, Lyapunov stability theory [56] is based on the idea that at a stable equilibrium, the energy of the system has a local minimum, whereas at an unstable equilibrium, it is at a maximum. It analyzes the behavior of the system in the following form $x_{k+1} = \mathcal{A}x_k$, where $\mathcal{A}$ corresponds to the matrix of the system in an open-loop form executing the defined close-loop inside.

In addition, it is defined a scalar function $V(x)$ which is a Lyapunov function using a quadratic form $V(x) = x^T Px$, where P is a symmetric matrix, positive defined, *i.e.*, all the eigenvalues of P are positive.

The Lyapunov Theorem states that a linear time-invariant discrete-time system $x_{k+1} = \mathcal{A}x_k$ is asymptotically stable if and only if for any positive definite matrix Q, such that $Q = Q^T > 0$ there exists a unique positive definite solution P to the discrete Lyapunov equation:

$$\mathcal{A}^T P\mathcal{A} - P = -Q < 0 \tag{4.8}$$

If this condition meets, the matrix $\mathcal{A}$ is asymptotically stable, *i.e.*, all its eigenvalues have a negative real part.

This theorem applies when we have a unique control model. However, in this case, we have a switched linear system that is composed of a piecewise signal that we want to make stable although the model switching. For this reason, it is necessary to apply a variation of this theorem and find a Common Quadratic Lyapunov function for all the

subsystems. In this case, we look for a positive definite symmetric matrix P such that

$$\mathcal{A}_i^T P \mathcal{A}_i - P = -Q_i < 0, i \in \mathcal{I} \tag{4.9}$$

This condition means that it is required to find one matrix P capable of fulfilling this property for all the subsystems. If this condition is met, the system will be asymptotically stable under arbitrary switching, *i.e.*, there is no restriction on the switching signal [80].

The open-loop transfer function for the approach is determined by the equation $Q_i(s).D_i^{-1}(s)$. These matrices have been built according to three decomposition techniques. The first one is separate $G(s)$ in distributed controllers, this transformation is given in Equation (4.5) where it is possible to verify that the obtained matrices $Q_i(s)$ and $D_i(s)$ are equal to the original transfer function $G(s)$. The other applied transformation is the serial decomposition given by Equation (4.6) where it is also possible to verify that the product of the obtained components respect also the original transfer function. Finally, the same occurs with the decomposition of parallel-serial function whose equation is Equation (4.7). For this reason, it is possible to conclude that $Q_i(s).D_i^{-1}(s) = G(s)$, $\forall i \in \mathcal{I}$. This means that the open-loop transfer function of the approach depends only of the original transfer function $G(s)$ which we know is stable due to the initial assumptions made in Section 4.3. Hence, exists a matrix P solution for condition (4.8). Finally, since all the subsystems are equivalents to $G(s)$, the same solution holds for condition (4.9) and the switched system is stable under arbitrary switching.

In conclusion, and from the physical point of view, there are no restrictions for the switching signal that compromise the stability of the system. However, in this type of system, the physical part is coupled with the cyber components and for this reason, the switching must be done considering the correct behavior of the cyber layer.

4.5 Experimental Results

Testbed. We simulate a CPS using a simplified version of the Tennesse Eastman (TE) control challenge problem [46] showed in Figure 4.2, already used in the related literature [260].

The system is described by the following matrix of transfer functions:

$$y = \begin{bmatrix} F4 \\ P \\ yA3 \\ VL \end{bmatrix} = G(s)u = \begin{bmatrix} g_{11} & 0 & 0 & g_{14} \\ g_{21} & 0 & g_{23} & 0 \\ 0 & g_{32} & 0 & 0 \\ 0 & 0 & 0 & g_{44} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \tag{4.10}$$

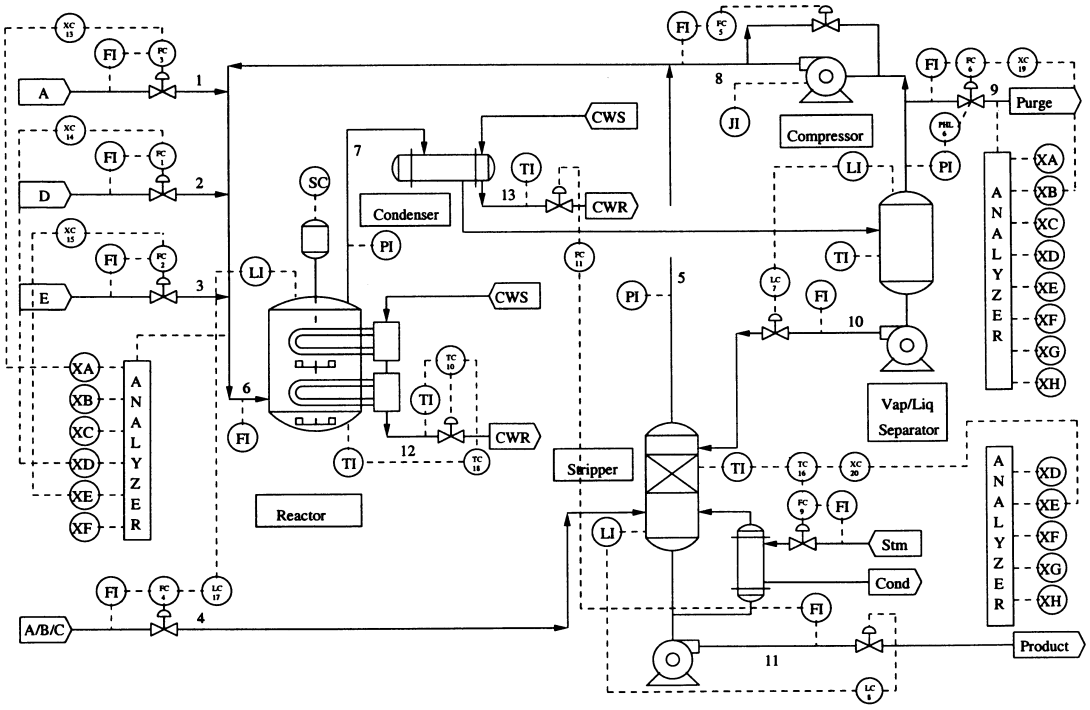


Figure 4.2 – Tennessee Eastman system [305].

where the monitored variables are the production rate ($F4$), the pressure (P), the amount of reactant A in the purge flow ($yA3$) and the liquid inventory (VL). The individual transfer functions are given below (the unit of s is seconds).

$$g_{11} = \frac{0.02833}{45s+1} \qquad g_{21} = \frac{45(340s+1)}{9000s^2+615s+1}$$

$$g_{23} = \frac{-900s-11.25}{9000s^2+615s+1} \qquad g_{32} = \frac{1.5}{600s+1}e^{-6s}$$

$$g_{14} = \frac{-3.4s}{360s^2+66s+1} \qquad g_{44} = \frac{1}{60s+1}$$

Design Procedure. Given the system transfer function G(s) in Equation (4.10), we apply the proposed MTD approach to obtain a resilient design to control the CPS.

Step 1.1 — Firstly, it is necessary to calculate $adjG(s)$ the adjudged matrix of G. In this particular case, we can observe that the output $yA3$ does only depend on variable u_3, *i.e.*, row 3 and column 2 have all zeros except for the element g_{32}. Hence, Steps 1.2 and 1.3 will give as a result the same function. To simplify the calculations, we will remove this row and column to obtain a G' matrix. We will add the component g_{32} again later in

the process. The adjugate matrix for G' is as follows:

$$adjG' = \begin{bmatrix} g_{23}g_{44} & 0 & -g_{14}g_{23} \\ -g_{21}g_{44} & g_{11}g_{44} & g_{14}g_{21} \\ 0 & 0 & g_{11}g_{23} \end{bmatrix} \tag{4.11}$$

Step 1.2 — We calculate the matrix $D(s)$ column by column choosing a position for the unity value. Here, we will show the process just for the first column. Hence, we will design only the first controller, obtaining the controller Q_{11} and four compensators $D_{i1}, i = 1...4$. This process is repeated with the other columns to obtain the other controllers.

To build the first column of matrix $D(s)$, we place the unit value in positions d_{11} or d_{21}. Notice that d_{31} equals 1 is a non-realizable configuration due to $adjG'_{31} = 0$. We obtain two different physical models for Controller 1:

- Model (a): If we choose the option $d_{11} = 1$ then $d_{21} = adjG'_{21}/adjG'_{11} = -g_{21}/g_{23}$.
- Model (b): If we choose the option $d_{21} = 1$ then $d_{11} = adjG'_{11}/adjG'_{21} = -g_{23}/g_{21}$.

Step 1.3 — After this, we can calculate matrix $Q(s)$ for the calculated matrix $D(s)$ in the previous step. In this case, we are just doing one column of the complete matrix, *i.e.*, we can calculate the corresponding controller Q_{11} by multiplying the first row of $G(s)$ and the first column of $D(s)$ to obtain Model (a) as $q_{11} = g_{11}$; and Model (b) as $q_{11} = -g_{11}g_{23}/g_{21}$.

Step 1.4.1 — In the previous steps, we obtained two models (a and b) for the distribution of controller Q_{11}. However, this does not generate enough models to create variability. In addition, the structure we want to build is formed for n controllers D_{ij} connected to each Q_{ii} controller. For this reason, we will apply series decomposition to generate more models and then parallel decomposition to generate four parallel controllers D_{ij}.

In a serial decomposition, we express the global transfer function $G(s)$ as a product of different factors that are executed in the different controllers obtained from the transfer functions $D(s)$ and $Q(s)$.

Table 4.1 summarizes the generated models using this technique. Model (c) has been generated starting from Model (a). According to Equation (4.5), we have that $G(s) = Q(s) \cdot D^{-1}(s)$. At this point, we are creating both matrices and we have not applied the inverse operation to matrix $D(s)$ yet.

Hence, to create Model (c), we part from Model (a) and we move the factor $1/g_{23}$ from the transfer function d_{21} to the transfer function q_{11} as the inverse operation due to $G(s)$

will use the inverse of $D(s)$ in a future step. However, if we observe Figure 4.1, when we change controller q_{11}, we also affect the result that goes to the transfer function d_{11} that is the reason why we have to multiply this controller for g_{23} also. As $G(s)$ uses the inverse of $D(s)$, we get that the changes of the entry d_{11} and q_{11} get compensated and the overall transfer function $G(s)$ does not change.

Similarly, we generate Model (d) from Model (b) using the factor g_{11}/g_{21} in q_{11} and moving its inverse to entries d_{11} and d_{21}.

More models can be generated if we apply this same technique but at the level of factors of the original transfer function. For example, we can obtain model (e) from model (b) in the following manner. The transfer function g_{23} can be expressed using the poles and zero representation as follows.

$$g_{23} = \frac{-900s - 11.25}{9000s^2 + 615s + 1} = -0.1\frac{s + 0.0125}{(s + 0.0667)(s + 0.0017)}$$

The poles and zeros representation can be calculated using *tf2zp* function in Matlab. Hence, it is possible to rewrite $g_{23} = g_{23}^1.g_{23}^2$ where g_{23}^1 and g_{23}^2 are any combination of the previous factors, for example, one of them may be as follows.

$$g_{23}^1 = -0.1\frac{s + 0.0125}{(s + 0.0667)} \quad g_{23}^2 = \frac{1}{(s + 0.0017)}$$

It is possible to move factors from the transfer function q_{11} to d_{11} and d_{21} by applying the inverse operation as in the previous examples. This way, it is possible to obtain even further models, e.g., Model (e).

Table 4.1 – Models generated with series decomposition

Model	q_{11}	d_{11}	d_{21}
(a)	g_{11}	1	$-g_{21}/g_{23}$
(b)	$-g_{11}g_{23}/g_{21}$	$-g_{23}/g_{21}$	1
(c)	$-g_{11}g_{23}$	$-g_{23}$	g_{21}
(d)	$-g_{23}$	$-g_{23}/g_{11}$	g_{21}/g_{11}
(e)	$-g_{11}g_{23}^2/g_{21}$	$-g_{23}^2/g_{21}$	$1/g_{23}^1$

Step 1.4.2 — After the previous step, we have many different models for q_{11}. However, we have just two d_{ij} because d_{31} and d_{41} are zero in $D(s)$. To improve this, we can apply partial fraction decomposition. We will show the procedure for Model (c). The component d_{11} can be separated in the following transfer functions:

$$d_{11} = \frac{900s + 11.25}{9000s^2 + 615s + 1} = \frac{0.0833}{s + 0.0667} + \frac{0.0167}{s + 0.0017}$$

Hence, we can divide d_{11} as the addition of two transfer functions:

$$d_{11}^1 = \frac{0.0833}{s + 0.0667} \text{ and } d_{11}^2 = \frac{0.0167}{s + 0.0017}$$

Similarly, we can transform d_{21} using the partial fraction decomposition as follows.

$$d_{21}^1 = \frac{1.6667}{s + 0.0667} \text{ and } d_{21}^2 = \frac{0.0333}{s + 0.0017}$$

With this procedure, we found the four compensators $d_{i1}, i \in 1..4$. The partial fraction decomposition can be found using the *residue* function in Matlab using the transfer functions.

Step 1.5 — Matrices A, B and C, for each transfer function can be easily obtained using Matlab functions *ss*, *c2d* and *ssdata* from the transfer function.

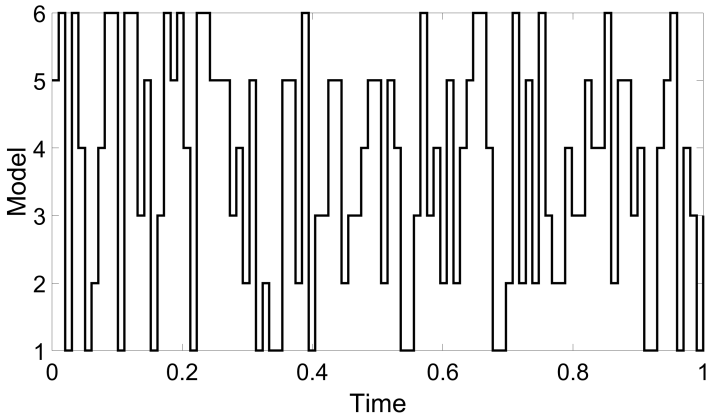
Step 2.1 and 2.2 — The control theory diagram of the obtained system is similar to the one showed in Figure 4.1(a) where there are four Q_{jj} boxes that execute the transfer function in the position (j, j) of the matrix $Q(s)$ and each one is connected to four D_{ij} that execute the transfer function in the position (i, j) of the matrix $D^{-1}(s)$.

Step 2.3 — This point describes the controller's dynamic behavior.

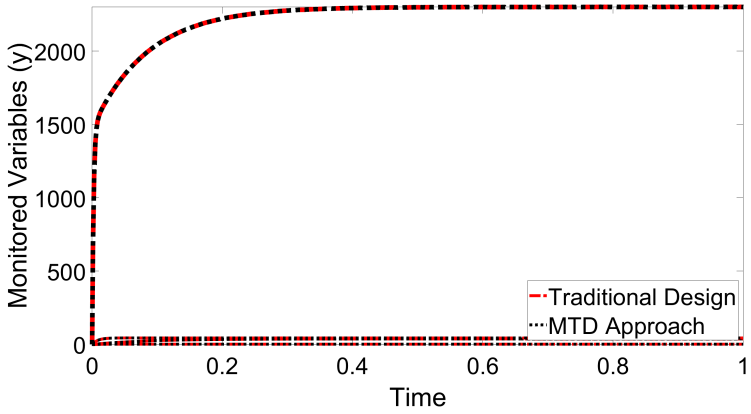
Step 3 — Using the Matlab function *dlyap* it was verified that condition ((4.8)) is met and in consequence, the system will be stable under unrestricted switching.

Results. To validate the approach, we implemented a numeric simulation with Simulink. From all the possible derived models, we choose V of them in an aleatory way to create a reduced proof of concept of the resilient CPS and analyze how the system reacts to adversaries with different capabilities. In this case, V is set up to six. However, in Section 4.6, we analyze the possible model generation for a process $n \times n$. The feedback loop was implemented using a Linear Quadratic Gaussian (LQG) approach.

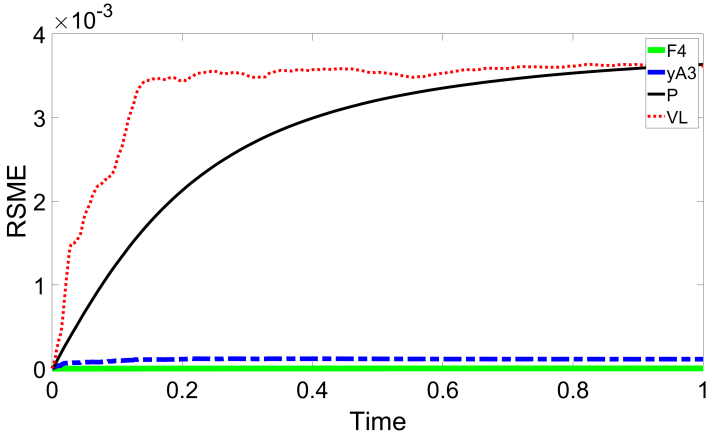
Results are showed in Figure 4.3. All the plots assume that time (cf. x-axis) is normalized between 0.0 and 1.0, representing the temporal percentage of multiple experimental runs. Figure 4.3(a) shows the MTD switching signal that selects the model to execute over time. The switching signal is configured at a frequency of 1 over 10. To simplify the simulation, the switching time was set up periodically. However, this is not necessary and it is possible to use non-periodic signals. Figure 4.3(b) shows the evolution of the system states in normal behavior, *i.e.*, without malicious actions, applying the proposed



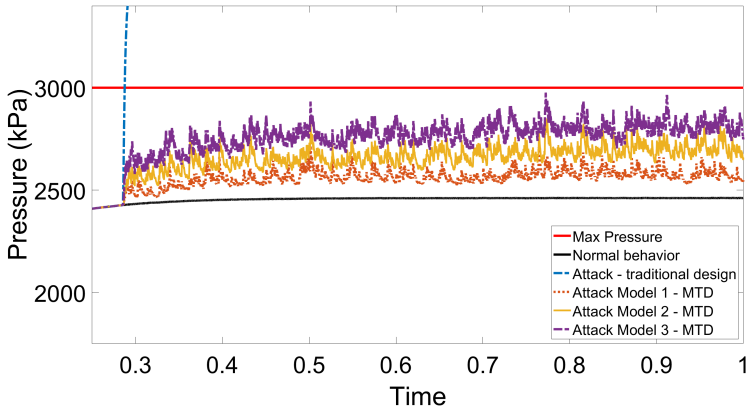
(a) MTD switching signal over time.



(b) Evolution of the system states for the traditional design without MTD and with our MTD approach.



(c) Optimality loss. Root Mean Square Error of the MTD approach with respect to the traditional design.



(d) Pressure evolution under attack with and without the MTD approach.

Figure 4.3 – Experimental results for the Moving-Target Approach. Time (x-axis) is normalized between 0.0 and 1.0, representing the temporal percentage of multiple experimental runs.

MTD approach and the traditional design without the MTD approach. It is possible to verify that the system remains stable and equal to the traditional system design although the model switching. In Figure 4.3(c), we present the Root Mean Square Error (RMSE) to analyze the optimality loss due to the new design. We can observe that the error between both signals, the one with the traditional system design and the proposed MTD approach, is in the order of 10^{-3}.

The actuators are valves that should operate in the range 0-100% which corresponds to the saturation limits. The process has to operate under certain safety constraints. One of them is that the reactor pressure should not exceed 3000 kPa [306].

The adversary aims at damaging the physical process. Hence, his objective will be to make the process pass the pressure limit to damage the system pipes. The pressure is monitored by the output P and in Equation (4.10), it is possible to see that it depends on control inputs u_1 and u_3 since $P = g_{21}.u_1 + g_{23}.u_3$. In addition, g_{21} has a positive sign. So, if we increase u_1, we will increase the pressure. On the contrary, g_{23} has a negative sign, so we need to decrease u_3 value to increase the pressure.

These control inputs are managed by the controllers Q_{11} and D_{1j} for u_1, and Q_{33} and D_{3j} for u_3 with $j \in 1..4$. The most efficient and powerful adversary is the one capable of compromising, in the case of u_1 the outputs from the D_{1j} controllers and the input of Q_{11} and analogously for u_3. This adversary is the one that we implemented to test the approach since it is the worst-case.

In addition, we defined adversaries with different capabilities in terms of the number of models that they are capable of learning for those compromised controllers and the saturation level of the valves, *i.e.*, how much the valves are opened. It can variate from 0% to 100% that represents the close and fully-open states respectively. We consider adversaries that are capable of learning 15% of the models (Model 1), 30% (Model 2) and 50% (Model 3). Also, for the saturation level, we consider u_1 and u_3 completely saturated at 100% and 0% respectively. As a reference point, the saturation level for the valves at the normal case and in stability conditions are 60.95% for u_1 and 25.02% for u_3.

Table 4.2 shows the maximum pressure increase for the worst-case adversary with respect to the normal case. In the case of an attack without a resilience approach, the system's pressure increases 21.95% reaching the maximum possible and damaging the pipes.

Figure 4.3d shows the pressure threshold, the system pressure in normal conditions and under attack considering a traditional design and a MTD design facing adversaries Model 1, 2 and 3. The attack starts when the system is already stable. It is possible to observe that the traditional design is not resilient and the adversary is able to make the system moves to an unsafe condition passing the threshold. In the system designed with the MTD approach, the adversaries are not able to make the system exceed the

Reference	Known Models	Max. Pressure Increase
M1	15%	10.41%
M2	30%	15.93%
M3	50%	20.98%

Table 4.2 – Tested malicious scenarios and pressure increase.

maximum pressure. The process signal presents little oscillations due to the correct models that compensate the actions of the adversary that tries to move the process out of stability.

4.6 Discussion

The example presented in Section 4.5 is a simplified version of a whole chemical process. The complete TE system has 50 states, 41 measured variables and 12 control inputs. Hence, it is possible to generate 12^{41} models (*i.e.*, approximately 2^{147} models). If we switch the model every 30 seconds, the adversary will need 1.6×10^{38} years to learn all models. Another well-known testbed such as the Secure Water Treatment (SWaT) system has 51 devices including sensors and actuators. The Vynil Acetate Monomer (VAM) Process has 246 states, 43 measured variables and 26 control inputs. In addition, a real industrial system may have even more devices. Hence, when applying this technique to bigger processes, it is possible to derive more models and get quite robust designs.

We have provided a concrete case showing how to apply the MTD approach where all the generated models are equivalents to the original one. The fact of building them through equivalences makes it easier to ensure the stability of the process but it may limit the number of models that we can generate to apply the approach. However, this mechanism can go further since the equivalence of the models is not a strict requirement. It is possible to switch different stable or unstable subsystems and ensured the stability of the global system if the switching signal is designed properly.

The control theory community has mathematically proved different switching stabilization methods for both stable and unstable subsystems [76, 80]. For example, in [307], the authors prove that if the total activating period of unstable modes is small enough compared with that of stable modes, the stability of switched linear systems is guaranteed. In addition, as showed in [308] and [309], it is also possible to design the system to switch all unstable subsystems and get as a result a stable system. In consequence, the model generation can be much wider than the one presented here, but to do so it is required to determine in a practical manner how to build the models starting from the

initial transfer function and how to design the proper switching function to control the physical process while guarantying the global stability of the system.

In addition, the system configuration switching should be done with enough regularity to make any information collected for reconnaissance purposes expire quickly. It aims at developing a mechanism that continually and unpredictably changes the parameters of the system to increase the cost of attacking, limit the exposure of vulnerable components and deceive the opponent.

The proposed approach aims at changing the attack surface to protect the system. The attack surface of a system can be seen as the subset of resources that an adversary can use to attack the system. This includes the entry and exit points of the system, its channels and any untrusted data items exchanged with the system.

According to the adversary defined in Section 4.3 and the attack surface defined in [310], the relevant resources that we have to protect are the system measurements (entry points), the command inputs (exit points) that are exploited using the data network packet payloads (channel) and that are generated exploiting the knowledge that the adversary has about the controller model (untrusted data items).

The cyber-physical attacks start with a reconnaissance phase to gather intelligence about the system. This requires time and effort for the adversary. The resilience approach attempts to render the adversary's intelligence invalid by switching the used physical model and remapping the network addresses. Our strategy protects the resources by continuously shifting the attack surface using defenses at two levels: node and network. At the physical level, the approach converts a centralized Linear Time-Invariant system into an equivalent distributed Linear Time-Variant system using switched rules, *i.e.*, from a unique controller represented as in Equation (2.1) by $G(s)$, we obtain a distributed design determined by matrices $Q(s)$ and $D(s)$ which provides $n(n+1)$ controllers. In addition, these $n(n+1)$ controllers switch the models over time and in consequence, they modify the logic for creating the data payloads of the packet since the commands respond to a new distributed way of calculating them. At the network level, the devices change the endpoint information to deceive the adversary.

Despite the promising preliminary results showed in this chapter, it is required a deeper assessment of the approach to evaluate the resilience improvement. However, most of the existing metrics, already presented in Chapter 2, Section 2.5.2 are not adequate to evaluate the resilience of a CPS. For this reason, in the next chapter, we propose metrics to evaluate cyber-resilience and we apply them to assess our proposed moving-target approach. This resilience evaluation includes an analysis of the adversary required effort to compromise the system and how the approach improves the resilience considering factors such as the system stability, performance, design and structure.

4.7 Summary

In this chapter, we proposed a resilient-by-design approach for CPS to face cyber-physical adversaries. We have followed the accepted decentralized architecture for CPS and we model the physical process as a switched system. A technique that already existed for other purposes, but was not explored for security improvement. This technique allows to create a Moving Target Defense that changes periodically the system model and the network configuration to self-heal the system. In this way, the knowledge that the adversary gathered about the system is no longer valid and to create a new attack is required to initiate a new learning process. In addition, with the new physical model, the system is able to compensate malicious actions and recover the stability.

We have validated the approach by simulating an industrial system with multiple actuators and sensors, showing that the strategy is able to build a resilient system capable of recovering from cyber-physical attacks. In the next chapter, we presented metrics to quantify the system resilience and we show how our approach improves it to validate the feasibility of our proposal.

5 Cyber-Resilience Evaluation

5.1 Introduction

Historically, malicious actions have not been a concern in control theory systems since this problem appeared with the introduction of computing resources to control the physical processes. However, control systems have a failure-resilience mechanism from their beginnings. It allows to detect and correct non-malicious disturbances, such as sensor or actuators small errors, process noises, etc. The objective of cyber-resilience is to tolerate attacks against a computational system to keep working and providing the essential services under attack. In the current context, failure-resilience is not enough, it is necessary to have mechanisms to provide cyber-resilience that goes beyond the traditional failure resilience and deal with malicious actions. In addition, it is also necessary to have mechanisms to evaluate at design time the resilience of a CPS from a cyber point of view to determine its capability to face cyber-physical adversaries.

Based on this, stability and performance are important factors to accept or reject a system design. Stability refers to the ability of a system to return to the equilibrium point after system disturbances, including the ones generated by malicious actions that move the system from stable states to unstable ones. The performance aims at working at the desired dynamic response and in a control mode that optimizes the objective function that minimizes costs and maximizes revenues. The control theory community has provided different criteria to analyze them, such as Lyapunov theory, root-locus, Routh-Hurwitz, Bode, or Nyquist methods. These mechanisms are prepared to take into account failure or process errors, but they are not prepared for malicious actions that may perturb the system.

It is not easy to predict at design time if the system will be stable when facing unknown malicious actions that will be introduced at runtime. However, it is possible to provide reference points to evaluate whether the system is better prepared or not to face adversaries.

This chapter aims at providing a set of cyber-resilience metrics that measure at design time the system behavior to determine whether or not it will be acceptable during an attack. We consider both issues: performance and stability. To do so, it may be acceptable to work in a graceful degradation mode while facing an attack, but it must be ensured at least the stability and a minimum performance threshold. We also analyze the internal structure of the system, identifying the critical components that are required to provide its fundamental functions and the capability of the system to restore the crucial components in case of damage due to attacks.

5.2 Contributions

In this chapter, we provide metrics to evaluate cyber-resilience in CPS that takes into account the temporal dimension of resilience. It is based on the stability and performance of the physical process to guarantee that the required safety properties are met. The objective is to provide a mechanism to assess the resilience of the system at design time.

In addition, we analyze the resilience of the system as a whole and not just components of the system. The objective is to quantify whether a proposed approach improves resilience or not and be able to compare different system designs to determine which one is the best from a resilience point of view.

The main contributions can be summarized as follows: (1) we provide a mechanism to evaluate at design time the resilience of a CPS in the presence of cyber-physical adversaries considering both the performance and stability of the system and the design and its structure; (2) we sum up guidelines to improve the resilience-by-design of a CPS; and (3) we provide experimental work to validate the approach. Parts of the contributions explained in this chapter were published in [311].

The outline of this chapter is summarized as follows. Section 5.3 presents the problem formulation and system assumptions. Section 5.4 presents the proposed metrics to evaluate the system's cyber-resilience and Section 5.5 presents the experimental work to validate the proposal. Finally, Section 5.6 discussed the obtained results and Section 5.7 summarizes this chapter.

5.3 Preliminaries

We provide in this section our assumptions about the system and the adversary model as well as some initial preliminary concepts.

System Model

We assume a system modeled as described in Chapter 2, Section 2.1.2. We also consider the actuator saturation that is a phenomenon normally simplified in the system model.

Actuator saturation is an inherent non-linearity feature in dynamic systems caused by constraints that reflect bounds or limits in actuators. The saturation function $sat : \mathbb{R} \rightarrow \mathbb{R}$ is defined as follows: $sat(u_i) = sign(u_i)min\{|u_i|, S_{max}\}$ where u_i is one entry of the command input u that is calculated as $u = Kx$ with K the feedback gain matrix and S_{max} is the maximum saturation level. For a vector $u \in \mathbb{R}^m$ we define $sat(u)$ as $sat(u) = [sat(u_1)sat(u_2)...sat(u_m)]$.

Actuators can not inject arbitrarily large amounts of energy into the system since there are always physical limitations and the saturation arises from these limits. For example, in the Tennessee Eastman problem, the actuators are valves that can be opened in a range from 0% to 100%. If the calculations from the equation $u = Kx$ give as a result a number out of this range, then the system executes the maximum saturation levels, *i.e.*, 0% for results lower than zero and 100% for results above this value. Other actuator saturation examples are, for instance, the maximum power that can be injected into an electrical system or the maximum acceleration possible by an engine due to limited torque or the maximum flow rate of a pipe.

The saturation limits the maximum command that may be executed at every time step. Hence, it is important to consider the actuator saturation for resilience because it limits the impact of the adversary on the system [312], but it also limits the response of the system to recover due to its implications on the stability and reachability of control. A saturated cyber-physical system can be mathematically modeled as follows:

$$x_{k+1} = Ax_k + Bsat(u_k) + w_k \tag{5.1}$$

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.* $w_k \sim N(0, Q)$. 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.

A static relation maps the state x_k to the system output $y_k \in \mathbb{R}^m$:

$$y_k = Cx_k + v_k \tag{5.2}$$

where $C \in \mathbb{R}^{m \times n}$ is the output matrix. The value of the output vector y_k represents the measurement produced by the sensors that are affected by a noise v_k assumed as a zero-mean Gaussian white noise and covariance R, *i.e.* $v_k \sim N(0, R)$.

We assume a stable system that shows optimal control under normal conditions (i.e. in the absence of malicious actions).

Adversary Model

The objective of the adversary is to cause a malfunction in the system by performing actions that affect the control system. The adversary is situated in a remote location but gained access to the internal network exploiting some cyber vulnerabilities and uses the network traffic to perform the attack as an insider.

A cyber-physical adversary can be modeled mathematically, as detailed in a previous section, with the following equations.

$$x'_{k+1} = Ax_k + B'sat(u'_k) + w_k \tag{5.3}$$

$$y'_k = C'x_k + v_k \tag{5.4}$$

where $B'sat(u'_k)$ represents an attack to the control input. The matrix B' is estimated by the adversary for the system model matrix B and u'_k is a malicious command. C' represents an adversary that is able to create a malicious sensor output y'_k. These malicious actions may be done by compromising sensors, actuators, controllers or network links.

5.4 Resilience Metrics

This section presents the proposed metrics to evaluate the system resilience at design time. First, we analyze the resilience considering the performance and stability of the system. And second, we analyze the resilience considering the design and structure of the system.

As explained previously, the power of the adversary lies in the knowledge gathered from the system, i.e, in the ability to learn the matrices A, B, C, and deceiving the system using malicious matrices B' and C' instead.

When an adversary learns the system model, it may turn it into an unstable system capable of violating safety restrictions. When an attack occurs, the system switches its behavior according to the models described in Section 5.3 due to malicious actions. In order to be resilient, the system should be capable of switching its behavior again during the attack to work in a new configuration without the exploited vulnerabilities. It should be able to ensure the stability and the minimum required performance of the physical process to keep working under safety conditions even in the presence of an attack. For this reason, the resilience definition we propose is as follows.

Definition 5.4.1 *[Cyber-Physical Resilient System] A Cyber-Physical Resilient System (CPRS) can be modeled as a switched control system that consists of a finite number of subsystems and a logical rule that orchestrates the switching between the subsystems:*

$$\begin{aligned} x_{k+1} &= A_{\sigma(k)}x_k + B_{\sigma(k)}sat(u_k) \\ y_k &= C_{\sigma(k)}x_k \end{aligned} \tag{5.5}$$

where $k \in \mathbb{Z}^+$ is the time interval, $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^p$ is the control input and σ is the logical rule that orchestrates the switching between the subsystems. It means that σ is a function $\sigma : \mathbb{Z}^+ \to \mathcal{I}$, where $\mathcal{I} = \{1, ..., N\}$ contains the indexes of the subsystems. The subsystems are determined by the set $\mathcal{M}$ where $\mathcal{M} = \{A_i, B_i, C_i : i \in \mathcal{I}\}$ is the set of physical system models. Hence, σ defines a piece-wise switching signal that is a time-varying definition of the process model that is activated at time k.

In addition, the set $\mathcal{M} = \mathcal{M}_s \cup \mathcal{M}_{us}$, where $\mathcal{M}_s$ denote the set of stable models, i.e, the normal behavior models and the models corresponding to resilience mechanism to recover from the attack. The set $\mathcal{M}_{us}$ contains the unstable models that are used by the adversary to damage the system. It is worth noting that we do not have any previous information about the models $\mathcal{M}_{us}$ since it depends on the adversary's decisions.

Hence, a system is CPRS if the overall system described in equation 5.5 is stable and meets the minimum performance threshold despite the malicious unstable models $\mathcal{M}_{us}$.

Definition 5.4.1 opens the following question to determine whether a system is a Cyber-Physical Resilient system.

1. What is the effort that the adversary should do to build a set of unstable models $\mathcal{M}_{us}$ capable to rend unstable the overall state, *i.e.*, a set of malicious models that can not be stabilized by the set of correct models $\mathcal{M}_s$. This question will be addressed in Section 5.4.1.

2. How do we know if the system will remain stable and meet the minimum performance threshold under the unknown adversary models $\mathcal{M}_{us}$. This question will be addressed in Section 5.4.2.

3. How do we build the models $\mathcal{M}_s$ to face the adversary. This question will be addressed in Section 5.4.3.

5.4.1 Attack Effort Analysis

In this section, we analyze the effort that the adversary should do to successfully attack the moving-target approach presented in Chapter 4. In cyber-physical exploits, the adversary payload contains a set of instructions that manipulate the process and the choice of instructions depends on the specific impact the adversary wants to have on the process. Hence, we consider different strategies an adversary may employ against the system defenses.

According to [313], the phases to achieve a cyber-physical attack are as follows. First, it is the *Access* phase which is the traditional hacking that gives the adversary an entry point to be inside the system. This part of the attack is not relevant for our analysis since it is related to classical cybersecurity problems. Then it is the *Discovery* phase where the adversary tries to learn how the system was designed and built. The next phase is the *Control* where the adversary tries to discover the dynamic behavior of the process that can be described by the transfer function or state-space model which are related by cause and effect relationships of the process. Finally, is the *Damage* phase where the adversary performs the attack itself.

Next, we discuss two adversary types with different knowledge capabilities and their strategies to overcome the attack phases described previously. One of the adversaries has no knowledge and performs a *brute force* attack. The other has detailed knowledge of the system and performs an *efficient targeted* attack.

Discovery. The brute force adversary starts with access to a CPS network but he has no knowledge about the system. During the discovery, the adversary collects information about the system to learn about its structure, how it works and how it was built. It is necessary to learn which are the components and how they are interrelated from analyzing network traffic. In this stage, the adversary faces the first and simplest barrier which is the network MTD that modifies the device's IP addresses. If there are K devices and each of them has a range R of available IP addresses, the adversary has to recreate

the network topology without any knowledge about how many real devices there are. The adversary needs to learn which type of sensors and actuators are involved in the process, and guess which is the function of the physical process, how it works and which may be the safety conditions to be exploited.

The efficient targeting adversary has much more detailed knowledge about the physical process which is more difficult and time-consuming to obtain. This adversary may be an ex-employee or someone who has access to the system management documentation.

First, he studies general information about the physical part such as chemistry, kinetics, thermodynamics, etc. This can be done by consulting open literature as well as proprietary information of process design companies.

The adversary may have typical company internal documents about system design, such as the ones described in [313]. For example, Piping and Instrumentation Diagrams which contains the system layout and physical structure, One-Line Diagrams which often contain information on safety conditions, Cause and Effect Diagrams with the behavior of the system, Cable Diagrams with the physical network topology, Instrument input/output Lists contain a list of instruments which serve as input or output of the control system, among others.

Hence, this adversary has precise knowledge about how the system carries out its functions, how it was built and the conditions that can put the system in danger.

For the efficient targeting adversary, the network MTD should not be a major problem since the adversary knows that exist K devices and their functions. Hence, performing some network analysis, the adversary may guess it.

Control. In the CPS resilient design, there are $n(n+1)$ controllers and n^n possible physical models available for the factorization in the matrices $D(s)$ and $Q(s)$. However, some of these configurations will not be realizable and the number of available factorization is given by:

$$\prod_{j=1}^{p}(\#TF \times \#DS \times \#DP)$$

where p is the number of actuators, i.e., the number of columns in the matrix $G(s)$, $\#TF$ corresponds to the number of transfer functions different from zero in the column j of the adjudged matrix, $\#DS$ corresponds to the number of possible series decomposition of a transfer function to generate two new transfer function which is C_w^2 combination of two taken from w, where w is the transfer function polynomial grade. In addition, $\#DP$ corresponds to the number of parallel decompositions which are $\sum_{j=1}^{p} C_w^j$ where p is the number of control signals.

Remark 1: To be successful, i.e., to go unnoticed during the attack, the adversary has to learn the models of the other cascade-dependent controllers. For example, in figure 4.1, if the adversary manages to learn the model of controller Q_{11} and deceive it. The value that D_{11} and D_{21} receive will not be correlated with what they expect and it is possible to know that something is not working properly in the system.

Similarly, if the adversary learns the model of D_{11} and manages to insert malicious messages, those commands will be executed by the actuators that affect G_{11} and G_{21}, which will modify the measures y_1 and y_2. Hence, Q_{11} and Q_{22} will receive values that are not the expected values.

For this reason, it is not enough to learn just one model, in every switching period the adversary has to gain a position in the required network links to learn the models of all the correlated close-loops to go unnoticed. Hence, in this phase, both adversaries have to do the same work.

However, the efficient targeted adversary can perform a smarter strategy. Since he knows which safety condition he wants to exploit and which are the controllers involved in controlling that variable, he needs to compromise those involved controllers. However, the cascade effect in correlated close-loops will force him to consider also the other controllers too and as a result, his work will not be easier than the brute force adversary.

On the contrary, the brute force adversary has no knowledge about the safety condition he may exploit. So, he has to learn all the controllers' models and start doing small probes. This way, by injecting smart disturbances he needs to understand how all the components work together and the cause-effect of the system variables to create a strategy to damage the system. If the switching time is big enough, such a learning process may be practical. Estimating the time required for an adversary to gather sufficient knowledge during the control phase is critical to assess the adversary's ability to successfully compromise the system and allow us to disrupt the adversary's reconnaissance effort. This way, it is possible to set up the switching time to avoid learning.

Remark 2: The adversary, in the most efficient scenario, has to (1) rebuild the network topology, (2) collect network traffic and (3) use this data to learn the model, for example, using machine learning. The time required for (1) can be depreciated for the efficient target adversary. However, Tasks (2) and (3) involve tasks that require in the order of several minutes to be performed.

Remark 3: Learning one model for just one controller involves learning many independent variables, i.e., the system parameters mentioned in Chapter 2, Section 2.1.2, matrices A, B, C, Q and R. Hence, the complexity of learning one model increases significantly with the complexity of the physical process, i.e., if the system has more sensors and actuators.

Remark 4: The time required for a model switching can be in the order of the seconds to leave enough time to converge the network devices in charge of the packets forwarding. Hence, this can make the task of the adversary hard to achieve.

Each time the adversary learns a model it gains some knowledge that can be used when the same model is executed again. In this case, the adversary has to guess the switching signal or he needs to gather data from each switching period to test if the current model fits with one of the previously learned models. Hence, the models already learned in the previous periods reduce the required effort for the adversary. However, the time required to learn each new model is not reduced because of the knowledge of previous models.

Damage. Even if the adversary learns the models and injects malicious packets during a switching period, i.e., the adversary turns that period into an unstable one, the system can still ensure stability as demonstrated in [307]. To be successful, the adversary has to compromise more than 50% of the physical models. If the adversary learns less than 50%, the stable model is activated sufficiently long (i.e., it is possible to absorb the state divergence made by unstable modes).

5.4.2 Performance and Stability Analysis

This analysis determines whether the system will remain stable and meet the minimum performance threshold under attack. It allows quantifying the maximum time that the system can resist in the absorb phase under attack, i.e., the maximum time that it has to react and stop the malicious actions. It also allows determining the states that the system may reach during the malicious action and estimate the maximum performance damage that the adversary may cause.

Traditionally, performance is used to measure the deviation between the process dynamics and the models to control it. In addition, it can be used to evaluate the resilience of a system by analyzing the capacity to absorb and recover from malicious action. In this section, we evaluate the underlying physical model to dimension the maximum performance loss during the worst attack scenario.

Thresholds and Setpoints: The performance must be defined according to the defined process operating objectives. For example, some possible objectives are safety conditions, product quality, environment protection, equipment protection, quality control, profit, among others. These established objectives will define process constraints that can be expressed as restrictions over the state of the system and they can be controlled through the monitored process variables. These restrictions define the performance thresholds (TS) that must be satisfied even when the system is working under attack.

Hence, the first step is to establish the minimum performance threshold that is required, and the setpoint (SP) for the normal system behavior.

In addition, the performance should be evaluated over a period of time which we will divide into the absorb and recover phase. The absorb phase starts with the attack in time k_0 and finishes in time k_a when the system reaches its minimum performance. The recovery phase starts in k_a and finishes in k_r when the system recovers its normal performance in the setpoint SP. This allows estimating the maximum derivation during the attack to evaluate if the performance threshold will be ensured. A small state variation during the attack is desirable so that the process variable remains close to its equilibrium state.

Resilience is based on absorbing and recovering potential. The absorbing property of a system is the degree to which challenges can be handled even with performance degradation. The recovery potential describes a system's ability to restore normal operation in the face of challenges. To estimate the performance, we will evaluate the system evolution during the absorb and recover phases.

Absorb Phase Time: The absorb time (KA) corresponds to the time required for the resilience approach to start working. In particular, the system is defined as resilient if for any adversarial input in the absorb phase the resulting state is within the threshold range.

Recover Phase Time: The recovery time (KR) which corresponds to the period $k_r - k_a$ depends on how fast the system can be stabilized. It can be estimated with the settling time of the control system. This is defined as the time taken for the process response to settle within near a constant value, usually in some band within 2% around the equilibrium state [314].

Maximum Deviation: The maximum deviation (MD) of the controlled variable from the SP is an important measure of the process degradation. We assume that in normal behavior the system is in the SP. Hence, the maximum deviation MD corresponds to the difference between the SP and the possibles deviations Ψ during the attack.

$$MD = max\{|\Psi - SP|\}$$

Given the time KA, it is possible to calculate the states that can be reached in the worst-case scenario where the adversary takes the system to its saturation level.

Hence, the maximum reachable state in time KA is calculated by substituting recursively the state x_k in the period k_0 and k_a as follows

$$\chi = x_{k_a} = Ax_{(k_a-1)} \pm BS_{max}$$

$$\chi = A^{KA}SP \pm \sum_{i=1}^{KA} A^{(KA-i)}BS_{max} \tag{5.6}$$

where χ indicates the maximum and minimum reachable states using the saturation level S_{max}. The set Ψ can be determined as $\Psi = C\chi$ using Equation 5.2.

Resilience Loss: The resilience loss (RL) is the sum of the differences between SP and the actual performance of the monitored variables during the absorb and recovery phase, i.e., RL is the sum of areas above and below the setpoint.

$$RL = (\sum_{j=k_0}^{k_r} |y_j - SP|) \tag{5.7}$$

Performance and Stability Resilience: The Performance and Stability Resilience PR can be estimated as the area defined within the thresholds TS during the absorb and recovery phase less the resilience loss RL.

$$PR = (NR - RL)/NR \tag{5.8}$$

where $NR = (TS_{sup} - TS_{inf}) \times (KA + KR)$.

General Performance and Stability Evaluation: The *performance and stability resilience analysis* quantify the impact of the attack in one of the monitored variables. For this reason, it is desired to have an overall metric that contemplates the global state of the system.

Not all the components contribute equally to develop the system's crucial functions. Hence, not all resources are equally likely to be used by an adversary. The resource's contribution to a system's attack surface depends on the resource's damage potential, i.e., the level of harm the adversary can cause to the system in using this resource in an attack. The higher the damage potential, the higher the contribution to the attack surface.

In addition, resilience must be evaluated considering the process operation objectives. As we mentioned previously, these objectives establish the process constraints that create state and monitored variables restrictions. For this reason, we need to evaluate the resources that are part of the system's attack surface to determine whether they

are critical from the objectives point of view. Then, it is possible to define c_j^{AT} as the contribution of the monitored variable j to the attack surface according to the defined objectives.

We calculate the global performance and stability resilience (GR) index by pondering the performance and stability resilience PR evaluation of each measured variable j according to their contribution to the attack surface as follows:

$$GR = \sum_{j=1}^{m} c_j^{AT} \times min(PR_j) \tag{5.9}$$

5.4.3 Design and Structure Analysis

To create a resilient CPS, it is required to build a set of stable models to activate when facing an attack. In this section, we review the techniques proposed in the literature to achieve resilient designs and how to evaluate its structure according to the adversaries the system can recover from.

The cyber-physical adversaries compromise the process by affecting the ability to maintain situational awareness of the process (i.e. affecting the observability) or by reducing the ability to bring the process to the desired state (i.e. affecting the controllability), or a combination of both.

Definition 5.4.2 (Controllability [315, 316]) *A system is controllable if every state vector x_k can be transformed into the desired state in finite time by the application of control inputs u_k. The controllability depends only on matrices A and B since a necessary and sufficient condition for a system to be controllable is that the controllability matrix $\mathfrak{C}(A, B)$ has n linearly independent columns.*

$$rank\ \mathfrak{C}(A, B) = rank[B|AB|...|A^{n-1}B] = n \tag{5.10}$$

Definition 5.4.3 (Observability [315, 316]) *The system is observable in n time-steps when the initial state x_0 can be recovered from a sequence of observations $y_0, ..., y_{n-1}$ and inputs $u_0, ..., u_{n-1}$. The observability depends only on matrices A and C since a*

necessary and sufficient condition for a system to be observable is that the observability matrix $\mathfrak{O}(A, C)$ has n linearly independent rows.

$$rank\ \mathfrak{O}(A, C) = rank \begin{bmatrix} C \\ CA \\ ... \\ CA^{n-1} \end{bmatrix} = n \tag{5.11}$$

As mentioned previously, four components in the attack surface may be attacked: sensors, actuators, controllers, and network traffic. For this reason, the generated models should address the vulnerabilities exploited in one or more of these components. The resilience design of a CPS can be characterized by the actuator resilience R_A, the sensor resilience R_S, the control resilience R_C, and the communication resilience R_N.

Definition 5.4.4 (Actuator Resilience R_A) *A CPS is t-actuator resilient if $rank\ \mathfrak{C}(A, B^{\Gamma}) = n$, i.e. the system is controllable for all possible subset Γ, where Γ is the set of all possible combinations of actuators removing t critical compromised actuators.*

Definition 5.4.5 (Sensor Resilience R_S) *A CPS is t-sensor resilient if $rank\ \mathfrak{O}(A, C^{\Delta}) = n$, i.e. the system is observable for all the subsets in Δ, where Δ is the set of all possible combinations of sensor removing t critical compromised sensors.*

This means that the system will be resilient if the controller can take action despite the compromised parts of the system. The definition of the matrices B^{Γ} and C^{Δ} depends on the particular resilience strategy applied to improve the actuator or the sensor resilience.

Definition 5.4.6 (Control Resilience R_C) *A CPS is t-control resilient if $rank\ \mathfrak{C}(A, B^{\Lambda}) = n$ and $rank\ \mathfrak{O}(A, C^{\Lambda}) = n$, i.e, the system is controllable and observable for all possible subset in Λ which is obtained by removing t possible compromised critical controllers.*

This definition means that if t-critical-controllers are compromised, the system can keep working and recover the state to an equilibrium point without these controllers working.

Definition 5.4.7 (Communication Resilience R_N) *A CPS is t-communication resilient if $rank\ \mathfrak{C}(A, B^{\Gamma}) = n$ and $rank\ \mathfrak{O}(A, C^{\Gamma}) = n$, i.e, the system is controllable and observable for all possible subset in Γ removing t compromised network links in which the adversary has the ability to recover the system model from collected data.*

Next, we review different strategies that can be used to improve the resilience of a CPS in each of its dimensions. We start from a minimum CPS with no resilience and progressively increase it with different techniques.

The minimal possible configuration is a CPS with the minimum amount of actuators and sensors to work, an automated controller capable of correcting errors in the process, and a non-redundant network that provides connectivity. This basic system provides observability and controllability to ensure fault correction. However, it is not resilient to attacks.

To build a resilient CPS is required to assess the system design including, f.i., techniques as the following ones.

The *actuators resilience* R_A can be improved by adding diversified actuators to perform the control actions over the system. Another proposal to improve the actuator resilience is presented in [205] which provides a resilient approach based on moving target defense techniques that use this principle to protect CPS from actuator and sensor attacks. In addition, in [136], authors define a decoder that can also correct attacks in actuators or sensors that have been corrupted. The strategies to improve actuator resilience require adding extra hardware devices that help to compensate for the incorrect function of the affected ones.

The *sensor resilience* R_S can be improved using different techniques. Similarly to the previous case, it is possible to add a diversified sensor. In addition, sensor resilience can be improved using software approaches that do not require adding extra hardware devices. For example, the techniques proposed in [138, 317–319] provide resilient state estimation and reconstruction in the presence of integrity attacks.

Another software approach to improve sensor resilience is to use an auxiliary system with Luenberger observers [320]. Observers have been proposed as detection mechanisms. For example, Shoukry and Tabuada [321] describe an algorithm for state reconstruction from sensor measurements that are corrupted using a Luenberger observer. Also, Schellenberger *et al.* [322] extend the 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 with similar dynamics of the original system and capable of 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 an attack.

The *controller resilience* R_C can be improved by adding local capabilities in the devices, for example, a smart actuator with an embedded local controller that can take control decisions outside the domain of the adversary. Another option is to implement distributed controllers that implement voting techniques to reach consensus to avoid malicious nodes. This problem has been studied extensively in distributed computing [236, 237]. Also, techniques such as secret sharing [242–244] and distributed trust [245, 246] may be used to implement, for example, mechanisms that divide the control into shares, such that the system needs to reach a given threshold prior to granting control. Below the threshold, the information gets concealed from the eyes of the adversary.

The *communication resilience* R_N can be addressed as a problem of transmitting information in the presence of misbehaving nodes has been widely studied in communication networks [238, 323]. To improve the network resilience one possibility is to add redundant physical or virtual independent networks. Other mechanisms such as the presented in [223] showed that linear iterative strategies are able to achieve the minimum bound required to disseminate information reliably, so malicious nodes will be unable to prevent from calculating any function (under a broadcast model of communication). Finally, in [286], it is proposed a mechanism that dynamically creates auxiliary controllers that help the switches to sanitize the traffic modified by the adversary in the network exchange.

5.5 Experimental Results

In this section, we analyze whether a system is resilient using the proposed metrics. To validate the approach, we estimate the defined metrics using the same testbed as in Chapter 4, a simplified version of the Tennessee Eastman (TE) control challenge problem [46] in a Matlab numeric simulation. The physical process consists of an isothermal reactor with a separation system. In it occurs an irreversible reaction where the reactants A_{TE} and C_{TE} generate the product D_{TE}. The reaction rate depends only on the partial pressures of A_{TE} and C_{TE}.

Manipulated Variables: The control objective is to maintain the product flow rate at a specified value by manipulating the flows of two feed steams, one purge stream, and the liquid holdup volume.

The two controlled feeds to the reactor chamber are Feed 1 and Feed 2. Feed 1 (u_1) consists of the reactants A_{TE} and C_{TE}, and traces of an inert gas B. Feed 2 (u_2) consists of pure A_{TE}, which is used to compensate for disturbances in the partial pressures of A_{TE} and C_{TE} in Feed 1.

The purge rate (u_3) depends on the pressure in the vessel and the position of the purge control valve. The vapor phase can be assumed to consist only on A_{TE}, B_{TE}, and C_{TE}, and the liquid, pure D_{TE}.

The product flow rate (u_4) is adjusted using a proportional feedback controller that responds to variations in the liquid inventory. The regulatory control problem is to maintain a specified product rate by manipulating flows of streams 1, 2, and 3.

Controlled variables: The monitored variables are the production rate (F4), the pressure (P), the liquid inventory (VL) and the amount of reactant A_{TE} in the purge flow (yA3).

Physical Model: The system model was defined with the matrix of transfer functions in Equation 4.10.

The first step to evaluate the resilience of a system is to determine the system threshold, the setpoints, and the saturation limits for its variables. These parameters are determined by the restrictions from the physical aspects of the plant. We used the data provided in the TE problem [46]. Tables 5.1 and 5.2 summarize the manipulated and measured variables.

Variable	Input for setpoint	Description	Saturation
u1	60.95327313484253	Feed 1 valve position	0–100%
u2	25.02232231706676	Feed 2 valve position	0–100%
u3	39.25777017606444	Purge valve position	0–100%
u4	44.17670682730923	Liquid inventory setpoint	0–100%

Table 5.1 – Manipulated variables [46].

Variable	setpoint	Description	Units	Threshold
F4	100.00	Product flow	kmol/hr	-
P	2700.00	Pressure	kPa	2k - 3k
VL	44.18	Liquid inventory	%	0 – 100
yA3	47.00	Amount of A_{TE} in purge	mol %	0 – 100

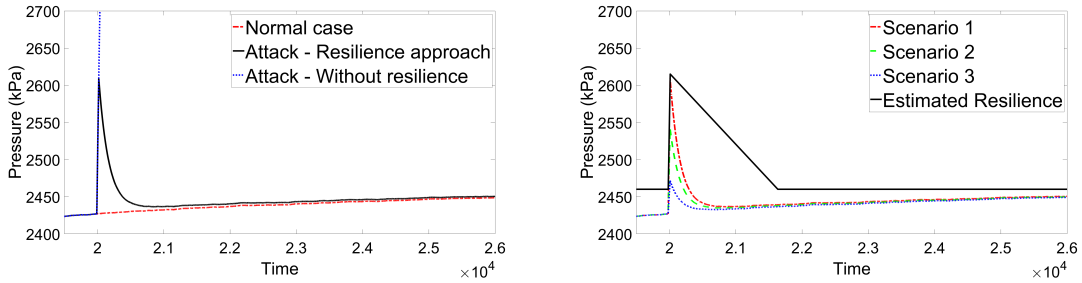
Table 5.2 – Controlled variables [46].

The physical process objective is to maximize the production rate while keeping a safe state.

Thresholds: The system thresholds are expressed in Table 5.2. In particular, the operating pressure must be kept below 3k Pa due to safety restrictions. Otherwise, the system should be shutdown.

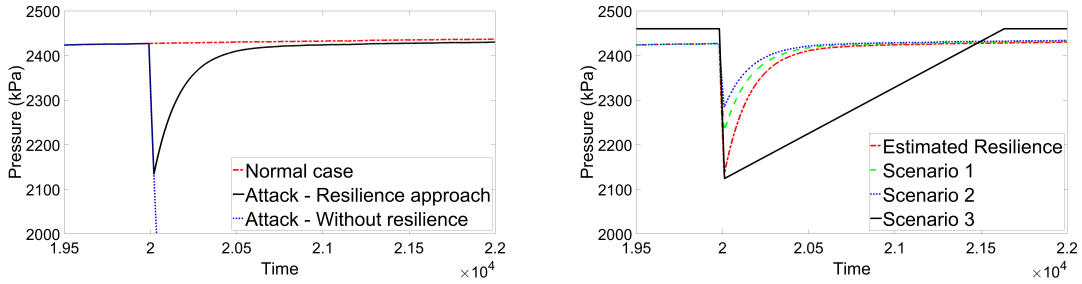
Saturation limits: The limits for each actuator are in Table 5.1. The flow rates saturate at some point and each valve can variate in a range of 0 to 100 % open to variate the flow rate.

Setpoints: The setpoints are in Table 5.2. In addition, in the column Input for SP in Table 5.1 are expressed the input associated with those setpoints.



(a) (b)

Figure 5.1 – (a) Resilient response vs normal case and attack without resilience for an adversary exploiting the maximum pressure threshold, (b) Resilience estimation using the proposed metrics vs. Montecarlo simulation for adversaries in Table 5.3



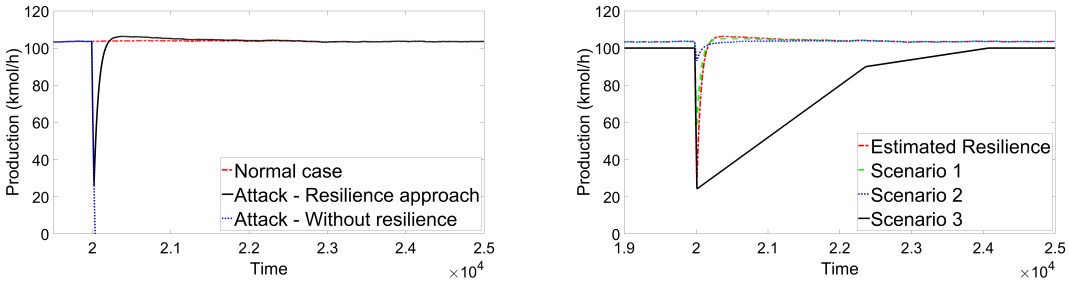
(a) (b)

Figure 5.2 – (a) Resilient response vs normal case and attack without resilience for an adversary exploiting the minimum pressure threshold, (b) Resilience estimation using the proposed metrics vs. Montecarlo simulation for adversaries in Table 5.4.

Valve	**Saturation**		
Scenario	**# 1**	**# 2**	**# 3**
u_1	100%	85%	70%
u_3	0%	5%	12.5%

Table 5.3 – Malicious saturation level scenarios to exceed the system maximum pressure.

The thresholds are essential to evaluate whether a system will be resilient or not. For each monitored variable with threshold restrictions, we should evaluate if the system will



(a) (b)

Figure 5.3 – (a) Resilient response vs normal case and attack without resilience for an adversary exploiting the minimum production rate threshold, (b) Resilience estimation using the proposed metrics vs. Montecarlo simulation for adversaries in Table 5.5.

Valve	**Saturation**		
Scenario	**# 1**	**# 2**	**# 3**
u_1	0%	20%	30%
u_3	100%	75%	50%

Table 5.4 – Malicious saturation level scenarios to exceed the system minimum pressure.

Valve	**Saturation**		
Scenario	**# 1**	**# 2**	**# 3**
u_1	0%	20%	30%
u_4	100%	75%	50%

Table 5.5 – Malicious saturation level scenarios to decrease the production rate.

	Performance & Stability				
Scenario	**KA**	**KR**	**MD**	**RL**	**PR**
# 1	30	126	182	13640	99.93%
# 2	30	53	115	4517	99.98%
# 3	30	0	44	442	100%
Resilience Estimation	30	1623	188	128240	99.80%

Table 5.6 – Resilience evaluation for Tennessee Eastman problem. Scenarios described in Table 5.3.

meet them or not considering the worst-case adversary scenario. In this experimental work, we present the evaluation considering only the system pressure as the monitored variable. However, the process should be also repeated for the other variables.

	Performance & Stability				
Scenario	**KA**	**KR**	**MD**	**RL**	**PR**
# 1	30	491	293	66327	99.68%
# 2	30	405	196	45900	99.78%
# 3	30	339	143	34050	99.83%
Resilience Estimation	30	1623	302	277910	98.72%

Table 5.7 – Resilience evaluation for Tennessee Eastman problem. Scenarios described in Table 5.4.

	Performance & Stability				
Scenario	**KA**	**KR**	**MD**	**RL**	**PR**
# 1	30	132	77	4512	99.78%
# 2	30	113	46	2489	99.88%
# 3	30	50	10	299	99.99%
Resilience Estimation	30	4060	79	155140	93.56%

Table 5.8 – Resilience evaluation for Tennesse Eastman problem. Scenarios described in Table 5.5.

Performance and Stability Metrics: For the evaluation, we consider the resilience approach explained in Chapter 4 and we want to measure how much this approach improves resilience by calculating the defined metrics. To be resilient the system has to remain within the threshold for any adversarial input. In the case of the system pressure, we have a minimum and maximum threshold.

In this experimental work, we consider two adversaries that want to exploit the pressure threshold. The first adversary makes the system exceed the maximum value and the second one the minimum. The configuration parameters for these adversaries are detailed in Tables 5.3 and 5.4 respectively. The scenarios use different saturation levels to represent adversaries' aggressiveness levels in the adversary model.

As showed in Equation (cf. 4.10), the pressure can be obtained as $P = g_{21}.u_1 + g_{23}.u_3$. Hence, it depends on command inputs u_1 and u_3. In addition, g_{21} has a positive sign, so, if we increase u_1, we will increase the pressure. On the contrary, g_{23} has a negative sign, so we need to decrease u_3 value to increase the pressure.

Figures 5.1a and 5.2a compare the behavior of the system with the resilience approach facing both adversaries. In addition, it compares this response with the normal case behavior without attack and the attack case without resilience.

To evaluate the resilience, we used the metrics defined in Section 5.4.2 and we compare the behaviors against the defined threshold. The results for the maximum and minimum pressure threshold are showed in Tables 5.6 and 5.7 respectively.

The estimated resilience is obtained with the proposed metrics and it shows how the system will react during the absorb and recovery phase considering the worst-case adversary. We can observe also that all the adversaries scenarios are included within the resilience estimation and more aggressive adversaries, such as scenario $\#1$, produce a bigger decrease in resilience than a less aggressive such as scenario $\#3$. We can observe this in Figures 5.1b and 5.2b that compare the estimated resilience with experimental Monte Carlo simulations for the scenarios in Tables 5.3 and 5.4 respectively.

Up to this point, we have analyzed the safety objective. However, it is necessary to guarantee also a minimum production. Otherwise, we will be wasting the input reactants without producing a useful product and in this case, shutting down the system will be a better option.

Figures 5.3a and 5.3b show the resilience evaluation for the production rate (F4) according to the simulation scenarios in Table 5.5.

After we have evaluated the critical controlled variables, we can calculate the global resilience. In this case, P and F4 are the critical variables due to the selected plant objectives: safety and maximize production. In addition, the restrictions on variable P are more critical considering the safety risks. For this reason, we will weigh the contribution of P on the system resilience as 70% and F4 as 30%. These values can be chosen arbitrarily according to the importance of each objective from the business point of view, in order to reflect these aspects in the process design. Then, the resilience achieved with the approach in Chapter 4 is

$$GR = 0.70 \times min(PR_P) + 0.30 \times min(PR_{F4}) = 97,17\%$$

Design and Structure Resilience: Thereinafter, we discuss how to incrementally design a resilient TE system. Possible designs are summarized in Table 5.9.

Design 1: The most basic design is a system with no automated controller feeding inputs to actuators. It is controlled, for example, manually by an operator or the actuators operate in a fixed way.

Design 2: Another option to create a basic design is a system that has no sensors and it works at open-loop since the controller is not getting feedback from the physical process.

Design		RA	RS	RC	RN
1	No controller. Fixed inputs.	0	0	0	0
2	No sensors. Open-loop control.	0	0	0	0
3	No redundant sensor, actuators, or network. Automated controller.	0	0	0	0
4	Design 3 with resilient control proposed in chapter 4 .	0	0	1	0
5	Design 4 with redundant sensors.	0	1	1	0
6	Design 5 with redundant actuators.	1	1	1	0
7	Design 6 with redundant forwarding paths.	1	1	1	1

Table 5.9 – Design and structure resilience evaluation for Tennesse Eastman problem.

Designs 1 and 2 are not resilient to attacks. They are even not capable of correcting system failures because there is no controllability and no observability. For this reason, the metrics R_A, R_S, R_C and R_N are all zero.

Design 3: The previous design can be improved by providing basic observability and controllability with an automated controller capable of correcting errors in the process, non-redundant actuators, sensors and network.

This design is better than the previous ones because it ensures fault correction, i.e, it is capable of correcting non-malicious errors in the physical process. However, this design is still not resilient to attacks and the metrics R_A, R_S, R_C and R_N are all zero.

To improve the resilience, it is required to contemplate mechanisms to face the compromise of sensors, actuators, controllers or network links. If a system has more capabilities to restore the critical components than other systems, then its more resilient. In the sequel, we provide examples to do this. These metrics do not represent system security. Instead, a better resilience measure indicates that the system will react in a stable manner, recover with less effort and with less damage after an attack.

Design 4: We can improve Design 3 by adding a resilience approach (e.g., the one explained in Chapter 4), to increase R_N in one.

Designs 5 and 6: Adding diversified sensors and actuators it is possible to improve R_S and R_A respectively. For instance, a system with 4 actuators can get 2-actuator resilient after removing any combination of two actuators and still be able to find a control input that can take the system to an equilibrium state. This means that if the set Γ which contains any combination of two not compromised actuators, i.e. $\{(a1, a2)$, $(a1, a3)$, $(a1, a4)$, $(a2, a3)$, $(a2, a4)$, $(a3, a4)\}$ will be 2-R_A if all the systems defined for this set Γ are controllable, i.e., $(A, B^{(a1,a2)})$, $(A, B^{(a1,a3)})$, $(A, B^{(a1,a4)})$, $(A, B^{(a2,a3)})$, $(A, B^{(a2,a3)})$ and $(A, B^{(a3,a4)})$ are all controllable.

Design 7: We can improve R_C by changing the valves for smart valves with an embedded controller integrated in the device. This option increases the resilience by adding redundant control outside the adversary domain, for example as in [10]. This will increase metric R_C in one unit.

5.6 Discussion

We showed an application case of the proposed metrics using the moving-target approach presented in Chapter 4. Also, this approach can be used for assessing the improvement generated with the detection-reaction approach presented in Chapter 3, considering the adversary effort, the performance, stability, design and structure of the system.

An issue not properly handled by our evaluation is the analysis of the performance and overhead generated by the proposed moving-target approach. Indeed, the performance of a cyber-physical system is an important issue that is necessary to be handled and analyzed. In this line, we highlight the need for better CPS testing and validation environments. Numeric simulation tools, such as Matlab/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. We tested our approach in Matlab/Simulink and we analyzed the performance loss in the physical part. It is still necessary to analyze it considering the cyber and network components. We will aboard this as future work.

The ideal validation option is to use testbeds integrating physical components. However, to test the performance correctly, we should consider that CPS may scale to hundreds or thousands of devices. 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. For example, the Tennessee Eastman problem or the Vynil Acetate Monomer scenario present an interesting diversity of devices. However, these systems are unstable. For this reason, the existing validations based on these scenarios normally use a reduced version of the problem, considering only a subsystem of the whole plant that is stable. This is the strategy that we applied in our tests.

The lack of realistic scenarios is mainly due to the complexity of creating plant 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 [14]. Designing such a

system is a huge effort and insights into real industrial systems are not possible due to justified confidentiality issues. In addition to the previous challenge, proposals address a wide range of application domains, system architectures and problem formulations [4]. The lack of common formulation criteria and validation scenarios makes it difficult to compare different solutions to similar problems.

Another point to evaluate in a resilience approach is how to manage the complexity of the proposal and how to anticipate the impact it may have on the system resilience. All the cybersecurity strategies can cause an unanticipated negative side effect in resilience. For example, to enhance 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. These agents may be able to use additional links to different elements or find replacement resources to ultimately restore its 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. Therefore, the performance impact analysis is not enough. The resilience proposal may also have hidden impacts on the system behavior and complexity, reducing the overall resilience. The quantification and evaluation of this aspect is not trivial.

In addition, regression test and automation testing techniques help to verify that new code or new components do not change the existing functionality and do not generate side effects on the existing functionalities. Such techniques are based on a full or partial re-execution of existing test cases. Then, the obtained output is compared with the predefined expected output. However, the concept of regression testing should be re-evaluated and adapted to be applied in CPS. In an IT system, normally an input value gives a determined output value that is correlated with the received data. In a CPS, the system does not exist isolated, it is coupled with the physical environment. As a consequence, the same inputs normally provide different outputs, since the response depends on many factors that influence the result. For example, plant disturbances, sensor noise, previous executions of the control loop and internal parameters that evolve at each execution cycle. For this reason, traditional automation testing can not be applied to control systems. Indeed, regression testing would facilitate the validation of new functionalities and approaches, in particular for this type of system that traditionally has had difficulties even applying software updates and patches. For that, automation testing should evolve to consider also the physical model and the system interactions with the surrounding environment.

5.7 Summary

In this chapter, we have also presented a definition of a cyber-physical resilient system. The definition models the system as a switched linear system to contemplate the system evolution during the absorb and recovery phases.

We have also presented metrics to evaluate the cyber-resilience of a CPS. The proposed resilience evaluation methodology considers a stability and performance analysis that aims at determining whether a system is resilient and the required conditions. In addition, a design and structure evaluation studies the type of adversaries the system can resist. We have also reviewed different strategies that can be used to improve the design and structure of the system.

These metrics can be analyzed at design time to evaluate if the system will have an acceptable behavior even in case of attack. The metrics also provide a mechanism to compare the resilience achieved by a particular approach or a whole system design, giving tools to evaluate during the system conception the best techniques to create a resilient design.

Finally, we have showed an application scenario using the Moving-Target approach proposed in Chapter 4 and using an industrial system scenario.

6 Conclusion and Future Work

6.1 Conclusion

In Cyber-Physical Systems (CPS), adversaries may disrupt the physical process by injecting malicious traffic, *i.e.*, cyber-physical attacks may use coordinated cross-layer techniques, to get control over the cyber or network layers, then to disrupt physical devices. For this reason, attacks over critical processes may have a catastrophic result, 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 mitigate or attenuate them, and resilience to contain the impact of the attack while keep providing the essential services and restore the normal operation if possible.

Resilience is essential for critical systems which monitor industrial and complex infrastructures based on Networked Control Systems (NCSs) [324]. If the defense strategy relies only on detection and reaction, the system is not protected in case of false negatives, undetectable attacks or extremely rare events that are not considered in the risk assessment. Also, attacks might come from inside, for example, from high skilled employees. The knowledge that insiders possess about the system gives them unrestricted access to steal or modify data or even deactivate functionalities. It is important to have a CPS capable of maintaining the stability of the system during such an attack. Also, the system should be protected at all times including the time required for detecting and responding to the attack. Otherwise, the system could experience damage.

In terms of contributions, we have started this dissertation with a global overview of the existing CPS security-related surveys. Then, we went further by surveying control theory formalities for CPS, the system architecture and cyber-physical attacks. We found

out that CPS are vulnerable to advanced adversaries that may be undetectable if we consider only cyber focused security solution.

As a result, we emphasize that control theory and cybersecurity are research areas that provide significant contributions to solve security issues in CPS, one contributes with insights about the physical process and the other with the cyber perspective. As a consequence, both research domains are complementary and working together have the potential to provide better solutions. In particular, because there are problems that are not possible to solve considering only the cyber or the physical perspective without considering the other dual part. In this line, we have reviewed the research efforts to integrate both areas of knowledge to create a synergy capable of providing new solutions to the new challenges created by cyber-physical adversaries.

For this reason, we analyzed detection and mitigation techniques to protect CPS. We surveyed some current trends in terms of mitigation techniques aiming to optimize the recovery response of a system under attack. We also presented techniques to build resilient systems. The proposals to build 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. However, most of the proposals consider mainly the cyber aspects and they still forget the physical part. We showed 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 detection-reaction and resilience is not clearly defined in the literature, and often, the two concepts are confused. We also discussed why these two concepts are different.

As a result of the literature analysis, we identified that plenty of research effort has been done in detection techniques and state estimation to maintain an awareness of the system state despite an attack. However, much less attention has been paid to create reaction approaches to mitigate or attenuate the attacks. Also, we identified a lack of adapted resilience techniques for the CPS particular needs.

Finally, we reviewed existing validation approaches considering existing testbeds and simulation tools. Also, we analyzed metrics and strategies to evaluate the cyber-resilience of a system.

The systematic review of the state of the art was complemented with three main contributions, properly disseminated in relevant publications in the field.

Our first contribution presented an attenuation approach driven by reflective programmable networking actions, in order to take control of adversarial attacks against CPS. We considered an adversary that injects malicious traffic in the network and is able to acquire knowledge about the system dynamics prior to starting the attack.

The proposed approach works in a detection-reaction manner and the resulting CPS satisfies self-healing, i.e., during adversarial situations, it continues working autonomously. We assumed cooperation between two different families of controllers, the CPS controller and the reflective programmable networking controller. In case of attack, new CPS controllers are created on-the-fly in the network domain and they help the forwarding devices to repair the malicious traffic. Network and CPS controllers cooperate to reach a common goal (e.g, to ensure system stability) even when they have their individual objectives.

We showed and validated the approach via experimental work using a testbed dataset and Omnet++ simulations. We argued that the use of software reflection, in addition to traditional techniques such as redundancy, diversity and automated recovery is a promising way to enable an efficient response under the presence of cyber-physical attacks. However, we identified some concerns and limitations in our approach. For this reason, we discussed new strategies to overcome the issues and we identified new research directions to apply in our next proposal.

In the second contribution, we presented a Moving Target Defense (MTD) approach to design resilient CPS. The objective was to design a system that without using a detection mechanism had the ability to restore the functions of the system by turning into useless knowledge that the cyber-physical adversary may have gathered about the system.

The approach modeled the system using switching linear control. This way, a series of decentralized controllers periodically modify the underlying physical and network configuration models of the CPS, satisfying self-healing properties. At the same time, the approach makes more complex the tasks that the adversary should do to be successful at performing new attacks. The system configuration switching should be done with enough regularity to make any information collected for reconnaissance purposes expire quickly. It aims at developing a mechanism that continually and unpredictably changes the parameters of the system to increase the cost of attacking, limit the exposure of vulnerable components and deceive the opponent.

The approach was validated using numeric simulations with the Tennessee Eastman challenge problem [46]. The obtained results are very promising and we proved that the system is stable using Lyapunov stability theory [56]. The resulting design is capable of absorbing and recovering from attacks. We also discussed how the attack surface changes and the potential of the approach considering different knowledge levels of the adversary. In each configuration change period, the adversary with the higher knowledge level, has to (1) rebuild the network topology, (2) collect network traffic and (3) use this data to learn the system model, for example, using machine learning. The time required for Task (1) can be depreciated. However, Tasks (2) and (3) require in the order of several minutes to be performed. Nevertheless, the time required for a model switching can be in the order of seconds to leave enough time to converge. Hence, this can make

the task of the adversary hard to achieve. We also highlighted the complexity in terms of variables required to learning one model. Finally, we also discussed how to improve the model generation and we provided some future lines to improve.

Our third contribution presented metrics to evaluate the resilience of a CPS. The proposed metrics are based on control theory performance and stability concepts, and the design and structure of the system. A system with better resilience indicates that it can react stably, recover with less effort and with less damage after an attack. We modeled a cyber-physical resilient system as a switched linear system to contemplate the system evolution during the absorb and recovery phases and the models turned unstable because of the malicious actions. Finally, we also reviewed different strategies that can be used to improve the design and structure of the system.

We evaluated the proposed metrics using the Moving-Target approach. We validated the capabilities of the metrics to provide an upper bound for the worst-case damage that an adversary may cause. Our metric proposition is innovative due to these metrics provide a mechanism to compare the resilience achieved by a particular approach or a whole system design, giving tools to evaluate during the system conception which are the best techniques to improve the resilient.

Research in resilience for CPS has still several actions to be done. Hence, there are wide opportunities for future research perspectives to extend and improve the existing field knowledge. In the next section, we point out several promising directions.

6.2 Future Work

In terms of perspectives for future research, as a result of the work initiated in this dissertation, there are several directions for improvement. This section discusses the limitations related to the existing resilience methods and the tools to evaluate them for CPS. This creates a great opportunity for researchers to find solutions to address such limitations and reduce the number of open problems.

System Modelization

- *More interaction between cyber components and physical components*

  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. However, as showed in this dissertation, in a CPS these elements work together and coordinated actions to attack vulnerabilities in the different components 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.

- *Resilient control and attack models*

  The control theory domain is 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 better characterize 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, to improve the defense mechanisms and to compare with other security mechanisms.

- *Digital twins*

  In this dissertation, we design the control loops using Kalman filters that are estimators used for stochastic cases, *i.e.*, when there is randomness involved in the development of the states of the system. Kalman filters explicitly use a noise model for both state and output processes considering the stocastic nature of the dynamical system. Thus, it is more appropriate for CPS and, in general, perform better for stochastic systems. Conversely, an observer, such as the Luenberger observer [320], 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 as explained in Chapter 5, Section 5.4.3. 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 plant, and produces an estimate of the state of the plant as output. The Kalman filter considers noisy measurements as inputs and produces an optimal state estimation.

Part of our future work, is to investigate how Luenberger observers can be used to improve the security of a CPS. For example, Luenberger observers may be used to create a digital twin of a plant. A digital twin is a virtual representation of a physical process which can be used to simulate, predict and optimize physical characteristics and system behavior. As a virtual copy of a process, a digital twin may allow to detect malicious behavior in the system when the virtual representation and the real process do not behave in the same manner. Also, it may be interesting to investigate whether a system damaged by malicious action, may be repaired using information from its digital twin to continue working in a safe mode.

Metrics and Evaluation Methods

- *Performance impact*

  As discussed in Chapter 5, Section 5.6, our work still needs to improve the analysis of the performance loss and overhead using the moving-target approach. With this in mind, a future perspective shall include a more thorough analysis of the performance impact of our resilient design using an experimental testbed. The performance of cyber-physical systems is an important issue that is necessary to handle. For the time being, we evaluated the overhead from a control-theoretic perspective. We need to assess the impact in an integrated manner considering also the network overhead.

- *Complexity management to anticipate impacts on resilience*

  The resilience of a system is influenced by several factors that can be managed or exploited in order to enhance resilience [12, 325]. All resilience-enhancing measures can also cause a negative effect leading to an overall reduction in resilience. As discussed in Section 5.6, greater complexity may also reduce resiliency. For example, due to unanticipated effects in the restoration work induced by hidden behaviors within 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 also 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.

  Hence, the different strategies and techniques used to improve the system resilience, such as the system topology, diversity of the resources, and others, may also have hidden impacts on the system behavior and also in the overall resilience.

How to evaluate this, is not an easy task. However, 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 revel potential negative impacts and systemic effects is another future research work.

- *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 with 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. As mentioned in Section 5.6, regression test and automation techniques should be evaluated and adapted for CPS. Since control systems are coupled with the physical environment, the outputs depend on factors such as plant disturbances, sensor noise, previous executions and evolving internal parameters that make it hard to apply traditional automation testing techniques. As a result, it is required to adapt them to consider the physical model and the system interactions with the surrounding environment.

Testing and Validation Environments

- *Scalability validation*

  CPS may scale into networks with hundreds or thousands of devices. As a result, it is important to test scalability aspects which 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 as commented in Section 5.6. Thus, testing scalability while combining physical process, network and software components is still a challenge.

- *Benchmarking*

  There are no common criteria and scenarios to compare approaches. Lun *et al.* [4] provide a quantitative analysis of the proposals in CPS and from the elaborated statistical data, we can appreciate that proposals address a wide range of application domains, system architectures, problem formulations and theoretical foundations. For this reason, it is difficult to compare different solutions to similar

problems. As a result, benchmarks and unified testbeds are required to improve this issue.

To conclude, we advocate that critical infrastructure may benefit from the improvements in the emerging research area that combines control-theory and cybersecurity to improve the system defense strategies. In the literature, it has been highlighted that a common limitation in CPS is how to react when detecting an attack. Triggering a different behavior may put in danger the process stability and the continuous operation. Our research focused on resilience for CPS, addressing this open issue and proposing new lines for improvement. As a result, we have identified further open research directions with promising future perspectives to develop resilience in CPS and complement the work initiated in this thesis.

Bibliography

[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] V. L. Do, L. Fillatre, I. Nikiforov, and P. Willett. Feature article: security of scada systems against cyber-physical attacks. *IEEE Aerospace and Electronic Systems Magazine*, 32(5):28–45, May 2017.

[8] R. Alguliyev, Y. Imamverdiyev, and L. Sukhostat. Cyber-physical systems and their security issues. *Computers in Industry*, 100:212–223, September 2018.

[9] 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.

[10] 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.

[11] 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.

[12] A. Kott and I. Linkov. *Cyber Resilience of Systems and Networks*. 01 2019.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] P. Cholda, J. Tapolcai, T. Cinkler, K. Wajda, and A. Jajszczyk. Quality of resilience as a network reliability characterization tool. *IEEE Network*, 23(2):11–19, March 2009. Conference Name: IEEE Network.

[21] I. Linkov, D. A. Eisenberg, K. Plourde, T. P. Seager, J. Allen, and A. Kott. Resilience metrics for cyber systems. *Environment Systems and Decisions*, 33(4):471–476, December 2013.

[22] 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.

[23] D. J. Bodeau, R. D. Graubart, and E. R. Laderman. Cyber Resiliency Engineering Overview of the Architectural Assessment Process. *Procedia Computer Science*, 28:838–847, 2014.

[24] 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.

[25] S. Hosseini, K. Barker, and J. E. Ramirez-Marquez. A review of definitions and measures of system resilience. *Reliability Engineering & System Safety*, 145:47–61, January 2016.

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

[27] A. Gholami, T. Shekari, M. H. Amirioun, F. Aminifar, M. H. Amini, and A. Sargolzaei. Toward a Consensus on the Definition and Taxonomy of Power System Resilience. *IEEE Access*, 6:32035–32053, 2018. Conference Name: IEEE Access.

[28] 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.

[29] P. Jain, R. Mentzer, and M. S. Mannan. Resilience metrics for improved process-risk decision making: Survey, analysis and application. *Safety Science*, 108:13–28, October 2018.

[30] M. S. Mahmoud, M. M. Hamdan, and B. U. A. Modeling and control of Cyber-Physical Systems subject to cyber attacks: A survey of recent advances and challenges | Elsevier Enhanced Reader, 2019.

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

[32] N. Bhusal, M. Abdelmalak, M. Kamruzzaman, and M. Benidris. Power System Resilience: Current Practices, Challenges, and Future Directions. *IEEE Access*, 8:18064–18086, 2020. Conference Name: IEEE Access.

[33] D. A. Sepúlveda Estay, R. Sahay, M. B. Barfod, and C. D. Jensen. A systematic review of cyber-resilience assessment frameworks. *Computers & Security*, 97:101996, October 2020.

[34] J. P. A. Yaacoub, O. Salman, H. N. Noura, N. Kaaniche, A. Chehab, and M. Malli. Cyber-physical systems security: Limitations, issues and future trends. *Microprocessors and Microsystems*, 77:103201, 2020.

[35] S. Mohebbi, Q. Zhang, E. Christian Wells, T. Zhao, H. Nguyen, M. Li, N. Abdel-Mottaleb, S. Uddin, Q. Lu, M. J. Wakhungu, Z. Wu, Y. Zhang, A. Tuladhar, and X. Ou. Cyber-physical-social interdependencies and organizational resilience: A review of water, transportation, and cyber infrastructure systems and processes. *Sustainable Cities and Society*, 62:102327, November 2020.

[36] D. K. Mishra, M. J. Ghadi, A. Azizivahed, L. Li, and J. Zhang. A review on resilience studies in active distribution systems. *Renewable and Sustainable Energy Reviews*, 135:110201, January 2021.

[37] T. Clédel, N. Cuppens, F. Cuppens, and R. Dagnas. Resilience properties and metrics: how far have we gone?, 2020.

[38] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn. Guide to industrial control systems (ics) security. In *NIST Special Publication 800-82*, volume revision 2, 2015.

[39] A. A. Cardenas, S. Amin, and S. Sastry. Secure control: Towards survivable cyber-physical systems. In *2008 The 28th International Conference on Distributed Computing Systems Workshops*, pages 495–500, June 2008.

[40] 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.

[41] 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.

[42] L. Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1–12, 2010.

[43] 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.

[44] L. Ljung. *System identification: Theory for the User*. Prentice-Hall, Inc., 1987.

[45] H. Natke. System identification: Torsten Söderström and Petre Stoica. *Automatica*, 28(5):1069–1071, 1992.

[46] N. L. Ricker. Model predictive control of a continuous, nonlinear, two-phase reactor. *Journal of Process Control*, 3(2):109–123, 1993.

[47] M. Barenthin Syberg. *Complexity Issues, Validation and Input Design for Control in System Identification*. PhD thesis, KTH School of Electrical Engineering, Stockholm, Sweden, 2008.

[48] 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.

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

[50] 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.

[51] 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.

[52] Y. Mo and B. Sinopoli. Secure control against replay attacks. In *Communication, Control, and Computing. 47th Annual Allerton Conference on*, pages 911–918. IEEE, Sept 2009.

[53] 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.

[54] M. Ivanescu. Mechanical Engineer's Handbook - Chapter 9. Academic Press Series in Engineering, pages 611 – 714. Academic Press, San Diego, 2001.

[55] B. Kouvaritakis, M. Cannon, and I. of Electrical Engineers. *Non-linear Predictive Control: Theory and Practice*. Control, Robotics and Sensors. Institution of Engineering and Technology, 2001.

[56] M. Sami Fadali and A. Visioli. *Digital Control Engineering: Analysis and design*. Academic Press, Boston, second edition edition, 2013.

[57] S. Svoronos, D. Papageorgiou, and C. Tsiligiannis. Discretization of nonlinear control systems via the carleman linearization. *Chemical Engineering Science*, 49(19):3263 – 3267, 1994.

[58] A. Rauh, J. Minisini, and H. Aschemann. Carleman linearization for control and for state and disturbance estimation of nonlinear dynamical processes. *IFAC Proceedings Volumes*, 42(13):455 – 460, 2009. 14th IFAC Conference on Methods and Models in Automation and Robotics.

[59] R. Vepa. *Nonlinear Control of Robots and unmanned aerial vehicles: an integrated approach*. CRC Press, 2017.

[60] J. Slotine and W. Li. *Applied Nonlinear Control*. Prentice-Hall International Editions. Prentice-Hall, 1991.

[61] L. Liu, S. Tian, D. Xue, T. Zhang, Y. Chen, and S. Zhang. A Review of Industrial MIMO Decoupling Control. *International Journal of Control, Automation and Systems*, 17(5):1246–1254, May 2019.

[62] J. Garrido, F. Vázquez, and F. Morilla. An extended approach of inverted decoupling. *Journal of Process Control*, 21(1):55–68, January 2011.

[63] L. Bakule. Decentralized control: An overview. *Annual Reviews in Control*, 32(1):87–98, April 2008.

[64] P. Campo and M. Morari. Achievable closed-loop properties of systems under decentralized control: conditions involving the steady-state gain. *IEEE Transactions on Automatic Control*, 39(5):932–943, May 1994.

[65] S. Skogestad and M. Morari. Variable selection for decentralized control. *Modeling, Identification and Control: A Norwegian Research Bulletin*, 13(2):113–125, 1992.

[66] Q. Wang. *Decoupling control*. Number 285 in Lecture notes in control and information sciences. Springer, New York, 2002.

[67] W. L. Luyben. Distillation decoupling. *AIChE Journal*, 16(2):198–203, March 1970.

[68] W. H. Ray. Multivariable process control - a survey. page 28, 1983.

[69] M. Singh, A. Titli, and K. Malinowski. Decentralized Control Design: An Overview. *IFAC Proceedings Volumes*, 16(12):1–15, July 1983.

[70] J. Garrido, F. Vázquez, and F. Morilla. Centralized multivariable control by simplified decoupling. *Journal of Process Control*, 22(6):1044–1062, July 2012.

[71] H. Lin and P. Antsaklis. Stability and persistent disturbance attenuation properties for a class of networked control systems: Switched system approach. *International Journal of Control - INT J CONTR*, 78:1447–1458, 12 2005.

[72] W. A. Zhang and L. Yu. Modelling and control of networked control systems with both network-induced delay and packet-dropout. *Automatica*, 44(12):3206 – 3210, 2008.

[73] A. Kruszewski, W. . Jiang, E. Fridman, J. P. Richard, and A. Toguyeni. A switched system approach to exponential stabilization through communication network. *IEEE Transactions on Control Systems Technology*, 20(4):887–900, 2012.

[74] R. Decarlo, M. Branicky, S. Pettersson, and B. Lennartson. Perspectives and results on the stability and stabilizability of hybrid systems. *Proceedings of the IEEE*, 88(7):1069–1082, July 2000.

[75] D. Liberzon and A. Morse. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine*, 19(5):59–70, October 1999.

[76] H. Yang, B. Jiang, and V. Cocquempot. *Stabilization of Switched Nonlinear Systems with Unstable Modes*, volume 9 of *Studies in Systems, Decision and Control*. Springer International Publishing, Cham, 2014.

[77] D. Liberzon. *Switching in systems and control*. Birkhäuser, Boston, 2012. OCLC: 904247468.

[78] J. P. Hespanha and A. Morse. Switching between stabilizing controllers. *Automatica*, 38(11):1905–1917, November 2002.

[79] S. Pettersson. Synthesis of switched linear systems. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, pages 5283–5288 Vol.5, Maui, HI, USA, 2003. IEEE.

[80] H. Lin and P. J. Antsaklis. Stability and Stabilizability of Switched Linear Systems: A Survey of Recent Results. *IEEE Transactions on Automatic Control*, 54(2):308–322, February 2009.

[81] J. Hespanha and A. Morse. Stability of switched systems with average dwell-time. In *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*, volume 3, pages 2655–2660, Phoenix, AZ, USA, 1999. IEEE.

[82] G. Zhai, B. Hu, K. Yasuda, and A. N. Michel. Qualitative analysis of discrete-time switched systems. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, volume 3, pages 1880–1885 vol.3, 2002.

[83] A. N. Michel. Recent trends in the stability analysis of hybrid dynamical systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 46(1):120–134, 1999.

[84] Y. Hui, A. Michel, and H. Ling. Stability theory for hybrid dynamical systems. *IEEE Transactions on Automatic Control*, 43(4):461–474, April 1998.

[85] N. Falliere, L. O. Murchu, and E. Chien. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, 5:6, 2011.

[86] 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.

[87] D. U. Case. Analysis of the cyber attack on the ukrainian power grid. *Electricity Information Sharing and Analysis Center (E-ISAC)*, 2016.

[88] J. Slay and M. Miller. Lessons learned from the maroochy water breach. In E. Goetz and S. Shenoi, editors, *Critical Infrastructure Protection*, pages 73–82, Boston, MA, 2008. Springer US.

[89] X. Li, C. Zhou, Y. Tian, N. Xiong, and Y. Qin. Asset-based dynamic impact assessment of cyberattacks for risk analysis in industrial control systems. *IEEE Transactions on Industrial Informatics*, 14(2):608–618, 2018.

[90] 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.

[91] 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.

[92] G. Dán and H. Sandberg. Stealth Attacks and Protection Schemes for State Estimators in Power Systems. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 214–219, Oct 2010.

[93] Y. Liu, P. Ning, and M. K. Reiter. False Data Injection Attacks Against State Estimation in Electric Power Grids. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '09, pages 21–32, New York, NY, USA, 2009. ACM.

[94] 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.

[95] Y. Liu, P. Ning, and M. K. Reiter. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security (TISSEC)*, 14(1):13, 2011.

[96] 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.

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

[98] A. Hoehn and P. Zhang. Detection of covert attacks and zero dynamics attacks in cyber-physical systems. In *2016 American Control Conference (ACC)*, pages 302–307, July 2016.

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

[100] 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.

[101] 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.

[102] 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.

[103] 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.

[104] T. H. Morris and W. Gao. Industrial control system cyber attacks.

[105] K. Tierney and M. Bruneau. Conceptualizing and Measuring Resilience. page 5, 2007.

[106] D. Wei and K. Ji. Resilient industrial control system (RICS): Concepts, formulation, metrics, and insights. In *2010 3rd International Symposium on Resilient Control Systems*, pages 15–22, Idaho Falls, ID, USA, August 2010. IEEE.

[107] Z. Wang, M. S. Nistor, and S. W. Pickl. Analysis of the Definitions of Resilience. *IFAC-PapersOnLine*, 50(1):10649–10657, July 2017.

[108] S. Jackson, S. Cook, and T. L. J. Ferris. A generic state-machine model of system resilience. *INSIGHT*, 18(1):14–18, 2015.

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

[110] M. T. Manavi. Defense mechanisms against distributed denial of service attacks : A survey. *Computers & Electrical Engineering*, 72:26 – 38, 2018.

[111] A. Bhardwaj, V. Mangat, R. Vig, S. Halder, and M. Conti. Distributed denial of service attacks in cloud: State-of-the-art of scientific and commercial solutions. *Computer Science Review*, 39:100332, 2021.

[112] S. T. Zargar, J. Joshi, and D. Tipper. A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE Communications Surveys Tutorials*, 15(4):2046–2069, 2013.

[113] G. Loukas and G. Öke. Protection against denial of service attacks: A survey. *The Computer Journal*, 53(7):1020–1037, 2010.

[114] W. P. M. H. Heemels and N. van de Wouw. *Stability and Stabilization of Networked Control Systems*, pages 203–253. Springer London, London, 2010.

[115] W. Zhang, M. S. Branicky, and S. M. Phillips. Stability of networked control systems. *IEEE Control Systems*, 21(1):84–99, Feb 2001.

[116] F. Pasqualetti, F. Dorfler, and F. Bullo. Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729, Nov 2013.

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

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

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

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

[121] 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.

[122] 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.

[123] 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.

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

[125] 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.

[126] 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.

[127] 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.

[128] 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.

[129] 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 M. Kutyłowski and J. Vaidya, editors, *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.

[130] 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.

[131] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen. Cyber security of water scada systems—part i: Analysis and experimentation of stealthy deception attacks. *IEEE Transactions on Control Systems Technology*, 21(5):1963–1970, 2013.

[132] 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.

[133] 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.

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

[135] 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.

[136] 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.

[137] 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.

[138] 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.

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

[140] 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.

[141] 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.

[142] 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.

[143] 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.

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

[145] 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.

[146] 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.

[147] 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.

[148] W. Chen and L. Qiu. Stabilization of networked control systems with multirate sampling. *Automatica*, 49(6):1528 – 1537, 2013.

[149] 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.

[150] 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 H. Jin, Q. Wang, and L. J. Zhang, editors, *Web Services – ICWS 2018*, pages 472–486, Cham, 2018. Springer International Publishing.

[151] 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.

[152] 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.

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

[154] 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.

[155] 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.

[156] 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.

[157] 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.

[158] 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.

[159] 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.

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

[161] 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.

[162] 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.

[163] 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.

[164] 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.

[165] H. Psaier and S. Dustdar. A survey on self-healing systems: approaches and systems. page 31, 2010.

[166] 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.

[167] 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, January 2013. pub_id: 202 Bibtex: nuernberger2013gadge URL date: None Organization: ACM.

[168] V. Pappas, M. Polychronakis, and A. D. Keromytis. Smashing the gadgets: Hindering return-oriented programming using in-place code randomization. In *2012 IEEE Symposium on Security and Privacy*, pages 601–615, 2012.

[169] 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.

[170] E. G. Chekole, S. Chattopadhyay, M. Ochoa, H. Guo, and U. Cheramangalath. CIMA: Compiler-Enforced Resilience Against Memory Safety Attacks in Cyber-Physical Systems. *Computers & Security*, 94:101832, July 2020.

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

[172] T. Jackson, A. Homescu, S. Crane, P. Larsen, S. Brunthaler, and M. Franz. Diversifying the software stack using randomized nop insertion. In S. Jajodia, A. K. Ghosh, V. Subrahmanian, V. Swarup, C. Wang, and X. S. Wang, editors, *Moving Target Defense II*, pages 151–173, New York, NY, 2013. Springer New York.

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

[174] 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.

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

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

[177] 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. Association for Computing Machinery.

[178] 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.

[179] 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.

[180] 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.

[181] 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.

[182] 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.

[183] B. Genge and C. Siaterlis. An experimental study on the impact of network segmentation to the resilience of physical processes. In R. Bestak, L. Kencl, L. E. Li, J. Widmer, and H. Yin, editors, *NETWORKING 2012*, pages 121–134, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

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

[185] 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.

[186] 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.

[187] 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.

[188] 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.

[189] 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.

[190] 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.

[191] 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.

[192] 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.

[193] 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.

[194] 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.

[195] 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.

[196] 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.

[197] 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.

[198] 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.

[199] 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 K.-Y. Lam, C.-H. Chi, and S. Qing, editors, *Information and Communications Security*, pages 39–53, Cham, 2016. Springer International Publishing.

[200] 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.

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

[202] 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.

[203] 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.

[204] 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.

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

[206] 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.

[207] E. Al-Shaer, Q. Duan, and J. Jafarian. Random host mutation for moving target defense. In A. D. Keromytis and R. Di Pietro, editors, *Security and Privacy in Communication Networks*, pages 310–327, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[208] 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.

[209] 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.

[210] 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.

[211] 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.

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

[213] 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.

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

[215] 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.

[216] 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.

[217] 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.

[218] 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.

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

[220] 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.

[221] 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.

[222] 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.

[223] 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.

[224] 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.

[225] 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.

[226] Q. Zhu and T. Başar. Game-Theoretic Approach to Feedback-Driven Multi-stage Moving Target Defense. In S. K. Das, C. Nita-Rotaru, M. Kantarcioglu, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, and G. Weikum, editors, *Decision and Game Theory for Security*, volume 8252, pages 246–263. Springer International Publishing, Cham, 2013.

[227] 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.

[228] 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.

[229] 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.

[230] 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.

[231] 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.

[232] 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.

[233] 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.

[234] P. Veríssimo, N. Neves, and M. Correia. Intrusion-Tolerant Architectures: Concepts and Design. In R. de Lemos, C. Gacek, and A. Romanovsky, editors, *Architecting Dependable Systems*, pages 3–36, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[235] 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.

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

[237] 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.

[238] 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.

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

[240] 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.

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

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

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

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

[245] 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.

[246] 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.

[247] 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.

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

[249] E. F. Brickell. Some ideal secret sharing schemes. In J.-J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology — EUROCRYPT '89*, pages 468–475, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg.

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

[251] 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.

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

[253] 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.

[254] 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 Q. Zhu, T. Alpcan, E. Panaousis, M. Tambe, and W. Casey, editors, *Decision and Game Theory for Security*, pages 159–171, Cham, 2016. Springer International Publishing.

[255] Matpower. https://matpower.org/. Accessed: 2020-09.

[256] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, 2011.

[257] F. Bellard. QEMU, a Fast and Portable Dynamic Translator. In *Annual Technical Conference, ATEC'05, Anaheim, CA, USENIX Association*, pages 41–46, Berkeley, CA, USA, 2005.

[258] Ns3 Network Simulation, Last Access: January 2021. Available at https://www.nsnam.org/.

[259] The OMNeT++ Network Simulation Framework, Last Access: April 2019. Available at http://www.omnetpp.org/.

[260] R. Chabukswar, B. Sinopoli, G. Karsai, A. Giani, H. Neema, and A. Davis. Simulation of Network Attacks on SCADA Systems. In *First Workshop on Secure Control Systems, Cyber Physical Systems Week*, April 2010.

[261] C. Queiroz, A. Mahmood, and Z. Tari. Scadasim—a framework for building scada simulations. *IEEE Transactions on Smart Grid*, 2(4):589–597, 2011.

[262] J. Downs and E. Vogel. A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17(3):245 – 255, 1993. Industrial challenge problems in process control.

[263] M. L. Luyben and B. D. Tyréus. An industrial design/control study for the vinyl acetate monomer process. *Computers & Chemical Engineering*, 22(7-8):867–877, July 1998.

[264] R. Chen, K. Dave, T. J. McAvoy, and M. Luyben. A Nonlinear Dynamic Model of a Vinyl Acetate Process. *Industrial & Engineering Chemistry Research*, 42(20):4478–4487, October 2003.

[265] Y. Machida, S. Ootakara, H. Seki, Y. Hashimoto, M. Kano, Y. Miyake, N. Anzai, M. Sawai, T. Katsuno, and T. Omata. Vinyl Acetate Monomer (VAM) Plant Model: A New Benchmark Problem for Control and Operation Study. *IFAC-PapersOnLine*, 49(7):533–538, 2016.

[266] A. Kaung Myat. Secure Water Treatment Testbed (SWaT): An Overview, 2015, https://itrust.sutd.edu.sg/wp-content/uploads/sites/3/2015/11/Brief-Introduction-to-SWaT_181115.pdf, Last access: April 2019.

[267] A. P. Mathur and N. O. Tippenhauer. SWaT: a water treatment testbed for research and training on ICS security. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, pages 31–36, Vienna, Austria, April 2016. IEEE.

[268] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur. A Dataset to Support Research in the Design of Secure Water Treatment Systems. In G. Havarneanu, R. Setola, H. Nassopoulos, and S. Wolthusen, editors, *Critical Information Infrastructures Security*, volume 10242, pages 88–99. Springer International Publishing, Cham, 2017.

[269] X. Yu and J. Jiang. Hybrid Fault-Tolerant Flight Control System Design Against Partial Actuator Failures. *IEEE Transactions on Control Systems Technology*, 20(4):871–886, July 2012.

[270] K. H. Johansson. The quadruple-tank process: a multivariable laboratory process with an adjustable zero. *IEEE Trans. Contr. Sys. Techn.*, 8:456–465, 2000.

[271] Black- i landshark ugv. https://www.blackirobotics.com/landshark-ugv/. Accessed: 2020-09.

[272] J. Rubio-Hernan, J. Rodolfo-Mejias, and J. Garcia-Alfaro. Security of cyber-physical systems — from theory to testbeds and validation. In *Security of Industrial Control Systems and Cyber-Physical Systems – Second International Workshop, CyberICPS 2016, Heraklion, Crete, Greece, September 26-30, 2016, Revised Selected Papers*, pages 3–18. Springer, September 2016.

[273] T. Yardley. Testbed cross-cutting research, 2014, https://tcipg.org/research/testbed-cross-cutting-research, Last access: April 2019.

[274] G. Koutsandria, R. Gentz, M. Jamei, A. Scaglione, S. Peisert, and C. McParland. A real-time testbed environment for cyber-physical security on the power grid. In *1st ACM Workshop on Cyber-Physical Systems-Security and/or Privacy*, pages 67–78. ACM, 2015.

[275] Y. Fang, N. Pedroni, and E. Zio. Resilience-Based Component Importance Measures for Critical Infrastructure Network Systems. *IEEE Transactions on Reliability*, 65(2):502–512, June 2016.

[276] R. Francis and B. Bekera. A metric and frameworks for resilience analysis of engineered and infrastructure systems. *Reliability Engineering & System Safety*, 121:90–103, January 2014.

[277] C. G. Rieger. Resilient control systems Practical metrics basis for defining mission impact. In *2014 7th International Symposium on Resilient Control Systems (ISRCS)*, pages 1–10, August 2014.

[278] K. Eshghi, B. K. Johnson, and C. G. Rieger. Power system protection and resilient metrics. In *2015 Resilience Week (RWS)*, August 2015.

[279] 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.

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

[281] 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.

[282] 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.

[283] European Union Agency for Network and Information Security Agency (ENISA). Methodologies for the identification of Critical Information Infrastructure assets and services. full report at https://www.enisa.europa.eu/, 2015.

[284] Y. Soupionis, S. Ntalampiras, and G. Giannopoulos. Faults and Cyber Attacks Detection in Critical Infrastructures. In *Critical Information Infrastructures Security: 9th International Conference, CRITIS 2014, Limassol, Cyprus, October 13-15, 2014, Revised Selected Papers*, pages 283–289, Cham, 2016. Springer International Publishing.

[285] J. Rubio-Hernan, L. De Cicco, and J. Garcia-Alfaro. Revisiting a watermark-based detection scheme to handle cyber-physical attacks. In *Availability, Reliability and Security (ARES), 2016 11th International Conference on*, pages 21–28. IEEE, August 2016.

[286] M. Segovia, A. R. Cavalli, N. Cuppens, J. Rubio-Hernan, and J. Garcia-Alfaro. Reflective Attenuation of Cyber-Physical Attacks. In S. Katsikas, F. Cuppens, N. Cuppens, C. Lambrinoudakis, C. Kalloniatis, J. Mylopoulos, A. Antón, S. Gritzalis, F. Pallas, J. Pohle, A. Sasse, W. Meng, S. Furnell, and J. Garcia-Alfaro, editors, *Computer Security*, pages 19–34, Cham, 2020. Springer International Publishing.

[287] M. Segovia, A. Cavalli, N. Cuppens, and J. Garcia-Alfaro. A Study on Mitigation Techniques for SCADA-driven Cyber-Physical Systems. In *Foundations and Practice of Security. FPS 2018. Lecture Notes in Computer Science, vol 11358*, pages 257–264. Springer, November 2018.

[288] D. B. West. *Introduction to Graph Theory*. Prentice Hall, 2 edition, September 2000.

[289] D. Jungnickel. *Graphs, Networks and Algorithms*. Algorithms and Computation in Mathematics. Springer Berlin Heidelberg, 2007.

[290] S. Rangarajan, Y. Huang, and S. K. Tripathi. Computing reliability intervals for k-resilient protocols. *IEEE Transactions on Computers*, 44(3):462–466, 1995.

[291] J. Fridman and S. Rangarajan. Maximizing mean-time to failure in k-resilient systems with repair. *IEEE Transactions on Computers*, 46(2):229–234, 1997.

[292] B. Brumback and M. Srinath. A chi-square test for fault-detection in Kalman filters. *IEEE Transactions on Automatic Control*, 32(6):552–554, Jun 1987.

[293] Modbus Organization. Official Modbus Specifications, 2016, http://www.modbus.org/specs.php, Last access: April 2019.

[294] M. Rollins. *Beginning LEGO MINDSTORMS EV3*. Apress, 2014.

[295] S. S. Lagu and S. B. Deshmukh. Raspberry Pi for Automation of Water Treatment Plant. In *Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on*, pages 532–536, February 2015.

[296] A. Varga and R. Hornig. An overview of the OMNeT++ simulation environment. In *1st International conference on Simulation tools and techniques for communications, networks and systems & workshops (Simutools)*, 2008.

[297] The OMNeT++/INET Framework, Last Access: April 2019. Available at http://inet.omnetpp.org/.

[298] T. Elteto and S. Molnar. On the distribution of round-trip delays in tcp/ip networks. pages 172–181, 11 1999.

[299] K. Paridari, N. O'Mahony, A. E.-D. Mady, R. Chabukswar, M. Boubekeur, and H. Sandberg. A Framework for Attack-Resilient Industrial Control Systems: Attack Detection and Controller Reconfiguration. *Proceedings of the IEEE*, 106(1):113–128, January 2018.

[300] P. Kumari and T. Anjali. Symmetric-key generation protocol (sgenp) for body sensor network. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6, 2018.

[301] F. Sivrikaya and B. Yener. Time synchronization in sensor networks: a survey. *IEEE Network*, 18(4):45–50, July 2004.

[302] W. C. Lindsey, F. Ghazvinian, W. C. Hagmann, and K. Dessouky. Network synchronization. *Proceedings of the IEEE*, 73(10):1445–1467, Oct 1985.

[303] S. Bregni. A historical perspective on telecommunications network synchronization. *IEEE Communications Magazine*, 36(6):158–166, June 1998.

[304] O. Simeone and U. Spagnolini. Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators. *EURASIP J. Wireless Comm. and Networking*, 2007, 01 2007.

[305] L. H. Chiang, E. L. Russell, and R. D. Braatz. Tennessee eastman process, 2001.

[306] N. L. Ricker. Decentralized control of the Tennessee Eastman Challenge Process. *Journal of Process Control*, 6(4):205 – 221, 1996.

[307] G. Zhai, B. Hu, K. Yasuda, and A. Michel. Stability analysis of switched systems with stable and unstable subsystems: an average dwell time approach. In *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, pages 200–204 vol.1, Chicago, IL, USA, 2000. IEEE.

[308] W. Xiang and J. Xiao. Stabilization of switched continuous-time systems with all modes unstable via dwell time switching. *Automatica*, 50(3):940–945, March 2014.

[309] X. Mao, H. Zhu, W. Chen, and H. Zhang. New results on stability of switched continuous-time systems with all subsystems unstable. *ISA Transactions*, 87:28–33, April 2019.

[310] S. Jajodia, editor. *Moving target defense: creating asymmetric uncertainty for cyber threats*. Number 54 in Advances in information security. Springer, New York, 2011. OCLC: ocn755905147.

[311] M. Segovia, J. Rubio-Hernan, A. R. Cavalli, and J. Garcia-Alfaro. Cyber-resilience evaluation of cyber-physical systems. In *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, pages 1–8, 2020.

[312] S. H. Kafash, J. Giraldo, C. Murguia, A. A. Cardenas, and J. Ruths. Constraining Attacker Capabilities Through Actuator Saturation. *arXiv:1710.02576 [cs, math]*, October 2017. arXiv: 1710.02576.

[313] M. Krotofil, D., and Larsen. Rocking the pocket book hacking chemical plants for competition and extortion. *DefCon Conference, DEFCON*, page 52, 2015.

[314] T. Marlin and T. Marlin. *Process Control: Designing Processes and Control Systems for Dynamic Performance*. Chemical Engineering series. McGraw-Hill Education, 2000.

[315] R. Kalman. Contributions to the theory of optimal control, 1960.

[316] R. Kalman. On the general theory of control systems. *IFAC Proceedings Volumes*, 1(1):491–502, 1960. 1st International IFAC Congress on Automatic and Remote Control, Moscow, USSR, 1960.

[317] D. Han, Y. Mo, and L. Xie. Towards a unified resilience analysis: State estimation against integrity attacks. In *2016 35th Chinese Control Conference (CCC)*, pages 7333–7340, July 2016.

[318] M. L. Corradini and A. Cristofaro. Robust detection and reconstruction of state and sensor attacks for cyber-physical systems using sliding modes. *IET Control Theory Applications*, 11(11):1756–1766, 2017.

[319] L. F. Combita, A. A. Cardenas, and N. Quijano. Mitigating sensor attacks against industrial control systems. *IEEE Access*, 7:92444–92455, 2019.

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

[321] 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.

[322] 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.

[323] J. Hromkovic, R. Klasing, A. Pelc, P. Ruzicka, and W. Unger. *Dissemination of Information in Communication Networks: Broadcasting, Gossiping, Leader Election, and Fault-Tolerance (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag, Berlin, Heidelberg, 2005.

[324] G. Gonzalez-Granadillo, J. Rubio-Hernán, and J. Garcia-Alfaro. Towards a security event data taxonomy. In N. Cuppens, F. Cuppens, J.-L. Lanet, A. Legay, and J. Garcia-Alfaro, editors, *Risks and Security of Internet and Systems*, pages 29–45, Cham, 2018. Springer International Publishing.

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

ECOLE DOCTORALE

Titre: Cyber-résilience et tolérance aux attaques des systèmes cyber-physiques

Mots clés: Systèmes Cyber-Physiques, Résilience, Tolérance aux Attaques, Réseaux Programmables, Software Reflection, Moving Target Defense, Systèmes Linéaires Commutés.

Résumé: Cette thèse porte sur la résilience des systèmes cyber-physiques. L'objectif principal est de développer une approche qui permet de poursuivre le fonctionnement du système de manière sûre, même en cas d'attaque. Nous abordons la réaction du système en créant une synergie entre l'information de la théorie du contrôle et les méthodes de cybersécurité pour absorber la menace et remettre le système dans son état correcte. Nous proposons deux approches utilisant des paradigmes différents. La première propose une stratégie de détection et de réaction visant à atténuer les attaques cyber-physiques, qui s'appuie sur des actions des *programmable reflective networks* pour prendre le contrôle des actions adverses. La seconde approche propose une stratégie de résilience par conception. L'approche est basée sur un paradigme de *moving target defense*, piloté par une commutation linéaire des matrices d'état-espace, et appliqué à la fois aux couches physique et réseau d'un système cyber-physique. Nous présentons également des mesures pour quantifier le niveau de cyber-résilience d'un système basé sur la conception, la structure, la stabilité et la performance pendant l'attaque. Enfin, nous avons identifié plusieurs possibilités de perspectives de recherche futures pour améliorer les connaissances existantes dans le domaine.

Title: Cyber-Resilience and Attack Tolerance for Cyber-Physical Systems

Keywords: Cyber-Physical Systems, Resilience, Attack Tolerance, Programmable Networking, Software Reflection, Moving Target Defense, Switched Linear Systems.

Abstract: This thesis investigates the resilience of Cyber-Physical Systems (CPS). We aboard the system reaction creating a synergy between control-theoretic information and cybersecurity methods to absorb and recover from the threat. We propose two approaches using different paradigms. The first one is based on a detection and reaction strategy to attenuate cyber-physical attacks driven by reflective programmable networking to take control of adversarial actions. The second approach proposes a resilient-by-design strategy. The approach is based on a Moving Target Defense paradigm, driven by a linear switching of state-space matrices, and applied at both the physical and network layers of a CPS. Also, we present metrics to quantify the cyber-resilience level of a system based on the design, structure, stability, and performance under the attack. Finally, we identified several possibilities for future research perspectives to improve existing knowledge in the field.

Institut Polytechnique de Paris

91120 Palaiseau, France