Detection of attacks against cyber-physical industrial systems

Jose Rubio-Hernan

THÈSE DE DOCTORAT CONJOINT TELECOM SUDPARIS et L'UNIVERSITE PIERRE ET MARIE CURIE

Spécialité : Informatique et Réseaux

École doctorale : Informatique, Télécommunications et Électronique de Paris

Présentée par

Jose Manuel RUBIO HERNAN

Pour obtenir le grade de
DOCTEUR DE TELECOM SUDPARIS

Detection of Attacks against Cyber-Physical Industrial Systems

---

Soutenue le 18 juillet 2017 devant le jury composé de :

Yves ROUDIER
Professeur HDR, I3S-CNRS-Université de Nice Sophia Antipolis / *Rapporteur*
Pascal LAFOURCADE
Maître de Conférences HDR, Université d'Auvergne / *Rapporteur*
Frédéric CUPPENS
Professeur HDR, Télécom Bretagne / *Examinateur*
Ana CAVALLI
Professeur HDR, Télécom SudParis / *Examinateur*
Urko ZURUTUZA
Maître de conférences, Université de Mondragon / *Examinateur*
Jean LENEUTRE
Maître de Conférences, Télécom ParisTech / *Examinateur*
Pierre SENS
Professeur HDR, LIP6/Inria Paris Rocquencourt / *Examinateur*
Joaquin GARCIA-ALFARO
Professeur HDR, Télécom SudParis / *Directeur de thèse*
Luca DE CICCO
Maître de Conférences, Politecnico di Bari / *Co-encadrant*

Thèse No : 2017TELE0015

All our dreams can come true
if we have the courage to pursue them.
— Walt Disney

To my family, my future wife and all the people
that have supported me during this adventure.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. Joaquin Garcia-Alfaro and to my co-advisor Dr. Luca De Cicco for their support during my PhD. Their motivation, patience and knowledge, as well as, theirs questions and advices helped me to carry out my PhD research and to write this thesis.

Besides my advisors, I would like to thank Prof. Yves Roudier and Dr. Pascal Lafourcade for being the reviewers of my thesis. Moreover, I want to thank the rest of my jury committee: Prof. Fréderic Cuppens, Prof. Ana Cavalli, Dr. Urko zurutuza, Dr. Jean Leneutre, and Prof. Pierre Sens. Likewise, I am grateful to Sandra Gchweinger, Veronique Guy and Francoise Abad, who helped me with all the administrative tasks.

I want to thank also my colleagues of the Réseaux et Services de Télécommunications (RST) department at Telecom SudParis. In special to my office colleagues, Dr. Gustavo Gonzalez Granadillo for his advices and discussions about our collaborative work, Rishikesh Sahay and Mohammed El Barbori. I also want to express my gratitude to Dr. Nesrine Kaaniche for her advices during my PhD. I am also sincerely thanking Jonathan Yung for the time he spent in collaborative research work with me and the engineering students.

I cannot forget to acknowledge the support from the Cyber CNI Chair of Institut Mines-Télécom. The chair is held by Télécom Bretagne and supported by Airbus Defence and Space, Amossys, EDF, Orange, La Poste, Nokia, Société Générale and the Regional Council of Brittany.

Last but not the least, I would like to thank my family: my parents, and my brothers and sisters, which have supported me during these three years, and my future wife who also supported me during this adventure. Additionally, I want to thank all friends that I have met in Paris for the great moments we spent together.

Paris, le 18 juillet 2017 T.D.

Abstract

We address security issues in cyber-physical industrial systems. Attacks against these systems shall be handled both in terms of safety and security. Networked control technologies imposed by industrial standards already cover the safety dimension. From a security standpoint, the literature has shown that using only cyber information to handle the security of cyber-physical systems is not enough, since physical malicious actions, that can threaten the correct performance of the systems, are ignored. For this reason, cyber-physical systems have to be protected from threats to their cyber and physical layers. Some authors handle the attacks by using physical attestations of the underlying processes. For instance, the use of physical watermarking can complement the protection techniques at the cyber layer, in order to ensure the truthfulness of the process. These detectors work properly if the adversaries do not have enough knowledge to mislead cross-layer (e.g., cyber and physical) data. Nevertheless, if the adversary is able to acquire enough knowledge from both layers, attacks will not be detected.

This dissertation focuses on the aforementioned limitations. It starts by testing the effectiveness of a stationary watermark-based fault detector, in order to detect, as well, malicious actions produced by adversaries to disrupt the system. We show that the stationary watermark-based detector is unable to identify cyber-physical adversaries. We show that the approach only detects adversaries that do not attempt to get any knowledge about the system dynamics. We analyze the detection performance of the original design under the presence of adversaries that infer the system dynamics to evade detection with high probability. We revisit the original design, using a non-stationary watermark-based design, to handle those adversaries. We also propose a novel approach that combines control and communication strategies. We validate our solutions using numeric simulations and training cyber-physical testbeds.

Résumé

Nous abordons des problèmes de sécurité dans des systèmes cyber-physiques industriels. Les attaques contre ces systèmes doivent être traitées à la fois en matière de sûreté et de sécurité. Les technologies de contrôle dirigés à travers du réseau, imposés par les normes industrielles, couvrent déjà la sûreté. Cependant, du point de vue de la sécurité, la littérature a prouvé que la simple utilisation de techniques cyber pour traiter la sécurité d'un système cyber-physique n'est pas suffisante, car les actions physiques malveillantes, qui peuvent menacer la performance des systèmes, seront ignorées. Pour cette raison, ont a besoin de mécanismes pour protéger les deux couches (cyber et physique) à la fois. Certains auteurs ont traité des attaques de rejeu et d'intégrité en utilisant, par exemple, une attestation physique pour détecter des attaques cyber-physiques à partir d'un tatouage des paramètres physiques du système. Ce type de détecteurs fonctionnent correctement si les adversaires n'ont pas assez de connaissances pour tromper les deux couches. Néanmoins, si l'adversaire est en mesure d'acquérir les connaissances physiques requises pour induire en erreur la couche cyber, l'attaque ne sera pas détectée.

Cette thèse porte sur les limites mentionnées ci-dessus. Nous commençons en testant l'efficacité d'un détecteur qui utilise une signature stationnaire afin de détecter les actions intentionnelles produites pour perturber le système. Nous montrons que le détecteur stationnaire est incapable d'identifier les adversaires cyber-physiques, car l'approche ne détecte que les adversaires qui ne tentent pas de connaître la dynamique du système. Nous analysons le ratio de détection de la conception originale sous la présence de nouveaux adversaires capables de déduire la dynamique du système et d'échapper au détecteur. Nous revisitons le design original, en utilisant un approche de tatouage non stationnaire, afin de gérer les adversaires visant à échapper à la détection. Pour gérer de tels adversaires, nous proposons également une nouvelle approche qui combine des stratégies de contrôle et de communication. Toutes les solutions son validées à l'aide de simulations et bancs de test.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

Symbol	Description
ARMAX	Autoregressive Moving Average Exogeneous.
ARX	Autoregressive Exogeneous.
BJ	Box-Jenkins.
CDF	Cumulative Distribution Function.
CPS	Cyber-Physical System.
HMI	Human Machine Interfaces.
ICS	Industrial Control Systems.
ICT	Information and Communications Technology.
IDS	Intrusion Detection System.
LTI	Linear Time Invariant.
MIMO	Multiple Inputs Multiple Outputs.
MISO	Multiple Inputs Single Outputs.
MTU	Master Terminal Units.
NCS	Networked Control System.
PLC	Programmable Logic Controllers.
RTU	Remote Terminal Units.
SCADA	Supervisory Control and Data Acquisition technology.
SIMO	Single Inputs Multiple Outputs.
SISO	Single Inputs Single Outputs.

Notations

Symbol	Description
$(d_0 \ldots d_n)$	Weight of the polinomial $\mathcal{D}(z)$.
$(n_0 \ldots n_m)$	Weight of the polinomial $\mathcal{N}(z)$.
ΔJ_m	Increment of quadratic cost due to the multi-watermark.
ΔJ_s	Increment of quadratic cost due to the single-watermark.
Δu_t	Single-watermark.
$\Delta u_t^{(i)}$	Multi-watermark.
γ	Detection threshold.
Γ and Ω	Ponderation matrices.
$\hat{T}$	Samples eavesdropped by the adversary.
$\hat{x}_{t\|t-1}$	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.
$\mathcal{W}$	LMS weight matrix.
A	State matrix.
AD	Samples detected.
B	Input matrix.
C	Output matrix.
DR	Detection ratio.
$E[\Delta u]$	Offset of Δu.
FN	False negatives.
FP	False positives.
g_t	Alarm signal.
$H(z)$	Model of the system, using frequency domain.

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.
SA	Samples under attack.
$U(z)$	Control input vector, using frequency domain.
u_t	Control input vector.
u'_t	Control inputs injected by the adversary.
u^*_t	Optimal control input vector.
$V(z)$	Output noise, using frequency domain.
v_t	Output noise.
$Var[\Delta u]$	Variance of Δu.
w_t	Process noise.
x_t	Vector of state variables.
$Y(z)$	Vector of the sensors measurements, using frequency domain.
$y^{\Delta u_t}$	Output due to the watermark.
y_t	Vector of the sensors measurements.
y'_t	Measurements injected by the adversary.

1 Introduction

1.1 Cyber-Physical Industrial Security

Current industries need to have permanent access to their data and related processes. Ensuring the intrinsic control of such exchanges is a challenging problem. A combination of both network and industrial control security needs to be enforced. This shall include: 1) traditional Information and Communications Technology (ICT) security, in order to guarantee appropriate control over computer and communication networks; (2) traditional cyber security solutions, to help at creating adapted detection techniques and countermeasures; (3) Industrial Control System (ICS) resilience, focused on the performance and optimization of industrial physical processes; and (4) safety techniques, in order to provide control over faults and accidents at the process level.

From the aforementioned areas, this dissertation mainly focuses on security challenges to detect attacks against cyber-physical systems. The kind of cyber-physical systems assumed in our work are assumed to be a combination of cyber and physical components working together under discrete and continuous industrial domains [1]. In turn, we devote our work on protection techniques addressing networked control systems, i.e., a subset of cyber-physical systems dedicated to industrial control processes. As many other technologies under control system theories, networked control systems aim at managing system configurations (often referred to as 'plants') that should drive desired responses upon reception of control commands. We focus specifically on closed-loop networked control systems, where the control system handles a dynamic feedback to maintain a certain relationship among the different variables of an industrial system.

Security of cyber-physical industrial systems is drawing a great deal of attention after the infamous StuxNet malware [2, 3] uncovered the potential of successful security attacks carried out against such systems. Several authors have studied the requirements to take into account the new security issues when designing security mechanisms for cyber-physical systems. In [4], Cardenas *et al.* define the issue of secure control by analyzing separately the problem first from an information security point of view and then by looking at specific control issues. In [5], Cardenas *et al.* also outline for the first time the difference between corporate ICTs security and cyber-physical

systems security. Figure 1.1 shows the way how adversaries conducting a cyber-physical attack can be represented through a block diagram, a representation typically used by the control system community. 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. The figure represents the control loop of a monitored system, and how adversaries succeed at modifying some of the readings, by recording and replicating previous measurements corresponding to normal operation conditions. Then, the adversaries modify the control input u_t (using u'_t) to affect the system state and disrupt normal operation conditions. If, on the one hand, adversaries are not required to have the knowledge of the system process model, on the other hand, access to all sensors (i.e., they have access to all components of the vector y_t) or insecure communication protocols are required to carry out a successful attack (using the correct vector y_t to modify the disrupted vector y'_t). This type of adversaries are undetectable with a monitor detector which only verifies faulty measurements.

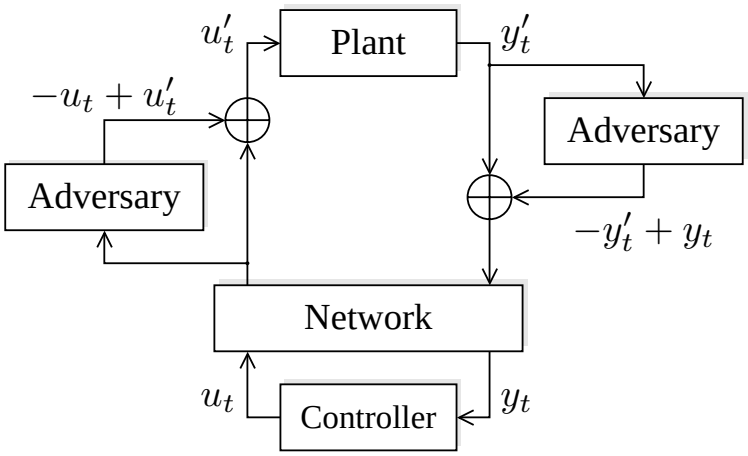


Figure 1.1 – Representation of a cyber-physical industrial attack against a networked control system. u_t and y_t represent the correct input and output vectors of the system. u'_t and y'_t represent the attack vectors.

From a cyber perspective, it is worth mentioning that control of industrial environments through technologies such as SCADA (Supervisory Control and Data Acquisition) is now present in most critical infrastructures (e.g., energy distribution and transport systems). Furthermore, protocols built upon networked control systems must cover regulation rules such as delays and faults [6]. Indeed, most industrial SCADA protocols (e.g., Modbus, DNP3, AGA-12, PROFINET and Ethernet/IP), are not designed to provide security from a traditional information or network perspective. Nevertheless, there are some protocols with security extensions. AGA-12 uses cryptography to add integrity and confidentiality protection, but with high deployment cost [7]. DNP3 has an extension named DNP3-SA (fifth version IEEE-1815-2012), adding new security features to DNP3, ensuring integrity and authentication of messages. Most industrial systems use these protocols over TCP/IP or UDP/IP communications (e.g., Modbus and DNP3 over TCP, PROFINET over TCP, Ethernet/IP over TCP or UDP); or directly over Ethernet communications, then using traditional ICT security mechanisms at the application layer.

At the application layer, we find protocols which have evolved in terms of protection. For instance, PROFINET has a new layer, PROFIsafe, designed to ensure safety, hence protecting the PROFINET protocol against malfunction (e.g., transmission errors). It does not ensure security against intentional malicious acts [8]. It is worth noting that most of the protocols at the application layer are modifications of serial protocols and do not provide security. Although transport and network layers can provide some security elements, these mechanisms are not sufficient to ensure control-data protection [9]. Indeed, to fully address the problem of control-data protection, cyber-physical solutions need to be added over such protocols.

In the literature, some authors propose the use of a physical attestation at the cyber layer [10], e.g., a physical watermark sent by the cyber layer to the physical layer, to verify the correct behavior of the physical processes [9]; or a watermark over the physical data to avoid identifying the real control values to secure the communications [11]. In [12], Arvani *et al.* describe a signal-based detector method, using discrete wavelet transformations. Do *et al.* study in [13] strategies for handling cyber-physical attacks using statistical detection methods. These propositions are only valid when the adversaries carry out attacks without the capability of acquiring the physical process knowledge. The work described in this dissertation aims at dealing with this issue.

1.2 Objectives and Contributions

The recently coined *cyber-physical system* term integrates a physical infrastructure and a cyber framework, in an effort to reducing complexity and costs of traditional control systems in, e.g., industrial environments. In turn, industrial control systems are composed of sensors, actuators, and other field devices that interact with the physical processes. Technology evolution brings these systems towards a combination between a physical layer which encompasses the physical framework; and a cyber layer that encompasses the communication and computation framework [14] via, e.g., SCADA (Supervisory Control and Data Acquisition) protocols.

In this dissertation, we address security issues in cyber-physical industrial systems. We focus on the protection between the cyber and physical layer of these systems. We started with a security analysis of previous theoretical detection mechanisms based on Mo and Sinopoli [15] and Chabukswar et *al.* [16] work, studying stationary watermarks in cyber-physical industrial systems. Following such watermark-based detection approaches, we uncovered some limitations and revisited their underlying constructions, by proposing the use of non-stationary watermarks to cover a larger number of threats. The revisited approach increases the detection ratio while keeping the same performance costs of the original designs. Furthermore, the revisited approach was complemented by an enhanced mechanism that combines control and security strategies, in order to handle some more powerful adversary models. Our solutions were validated by using numeric simulations and real world cyber-physical testbed. In the sequel, we summarize the complete list of objectives and contributions related to the work reported in this dissertation.

Problem statement: Current cyber-physical system security is focused on either cyber adversaries or physical adversaries, but not both at the same time.

Thesis statement: New security challenges in cyber-physical systems make necessary to analyze current control-strategies and related protection mechanisms to properly detect stealthy attacks, i.e., attacks by powerful adversaries aiming to escaping detection. The analysis shall allow us to create new control and security strategies, improving the detection mechanisms existent in the literature, in order to secure the cyber and the physical layer to handle cyber-physical adversaries.

To address the above problem and thesis statements, we establish the following objectives:

- **Objective 1.1:** Analyze potential threats existent in cyber-physical systems.
- **Objective 1.2:** Study how existent detection mechanisms are able to handle cyber-physical threats, as well as to identify shortcomings and limitations.
- **Objective 1.3:** Create new detection mechanisms in order to handle those uncovered shortcoming and limitations.
- **Objective 1.4:** Analyze the robustness and limitations of the new mechanisms.
- **Objective 1.5:** Integrate the detection mechanisms that may mitigate the threats.
- **Objective 1.6:** Validate the proposed mechanisms via simulation and testbeds.

Contributions: The mechanisms proposed in this dissertation allow us to detect threats carried out by cyber-physical adversaries. Adversarial models are proposed and classified, with regard to their capabilities (e.g., a priori knowledge about system dynamics, to evade detection). We define two novel adversarial models: parametric and non-parametric cyber-physical adversaries. We also address the shortcomings of centralized detection mechanisms and define a decentralized detection strategy that increases the robustness against attacks. The complete list of contributions, with regard to the aforementioned objectives, is provided below.

- Definition and classification of cyber-physical adversary models (*Objective 1.1*).
- Evaluation of the detection performance using existent detection mechanisms with respect to the list of adversaries (*Objective 1.2*).
- Revisited detection mechanism providing a cyber-physical detector covering the different adversary models (*Objective 1.3*).
- Integration of the revisited mechanism with a novel control-communication strategy. The resulting detector allows the correlation of events between cyber and physical layers, in order to increase the detection performance (*Objectives 1.3, 1.4 and 1.5*).
- Construction of numeric simulations and training cyber-physical testbeds, in order to validate the new detection mechanisms (*Objective 1.6*).

1.3 Publications

Early versions of the results covered in this dissertation have been successfully reported in the following peer-reviewed publications.

- 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, ISSN: 2161-3915, DOI: 10.1002/ett.3209, August 2017.
- 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, ISSN: 2510-523X, DOI: 10.1186/s13635-017-0060-9, June 2017, p. 8.
- J. Rubio-Hernan, J. Rodolfo-Mejias and J. Garcia-Alfaro, *Security of Cyber-Physical Systems: From Theory to Testbeds and Validation,* 2nd Workshop on the Security of Industrial Control Systems and Cyber-Physical Systems (CyberICPS 2016), Heraklion (Greece), DOI: 10.1007/978-3-319-61437-3_1, June 2017, pp. 3–18.
- J. Rubio-Hernan, L. De Cicco and J. Garcia-Alfaro, *Event-Triggered Watermarking Control to Handle Cyber-Physical Integrity Attacks,* 21st Nordic Conference in Secure IT Systems (NordSec 2016), Oulu (Finland), DOI: 10.1007/978-3-319-47560-8_1, November 2-4, 2016, pp. 3–19.
- J. Rubio-Hernan, L. De Cicco and J. Garcia-Alfaro, *Revisiting a Watermark-Based Detection Scheme to Handle Cyber-Physical Attacks,* 11th International Conference on Availability, Reliability and Security (ARES 2016), **(Best Paper Runner-Up Award)**, Salzburg (Austria), DOI: 10.1109/ARES.2016.2, September 2016, pp. 21–28.
- J. Rubio-Hernan, J. Garcia-Alfaro, *On the Adaptation of Physical-layer Failure Detection Mechanisms to Handle Attacks against SCADA Systems,* Symposium on Digital Trust in Auvergne (SDTA), *Extended Abstract*, Clermont-Ferrand (France), December 2014, pp. 1–2.
- J. Garcia-Alfaro, C. Romero-Tris, J. Rubio-Hernan, *Simulaciones Software para el Estudio de Amenazas contra Sistemas SCADA,* 13th Spanish Meeting on Cryptology and Information Security (RECSI), Alicante (Spain), September 2014, pp. 151–156,

Two additional publications, not directly related to the results of this dissertation, are listed below.

- G. Gonzalez-Granadillo, J. Rubio-Hernan and J. Garcia-Alfaro, *Towards a Security Event Data Taxonomy,* 12th International Conference on Risks and Security of Internet and Systems (CRISIS), Dinard (France), September 2017.

- G. Gonzalez-Granadillo, J. Rubio-Hernan, J. Garcia-Alfaro and H. Debar, *Considering Internal Vulnerabilities and the Attacker's Knowledge to Model the Impact of Cyber Events as Geometrical Prisms,* IEEE Trustcom/BigDataSE/ISPA, Tianjin (China), August 2016, pp. 340–348.

1.4 Organization

The remainder chapters of this dissertation are structured as follows.

- **Chapter 2, State of The Art.** This chapter contributes to *Objective 1.1*. It provides the background of the dissertation, including an analysis of related work.
- **Chapter 3, Dynamic Challenge-Response Authentication Scheme.** This chapter develops our analysis of the watermark-based detector mechanism existent in the literature, reporting shortcoming and limitations. It presents a revisited version of the analyzed detector. The chapter contributes to *Objectives 1.2* and *1.3*.
- **Chapter 4, Adaptive Control-Theoretic Detection.** This chapter provides a distributed detection mechanism, as an evolution of existing watermark-based detectors. The chapter contributes to *Objectives 1.3*, *1.4*, and *1.5*.
- **Chapter 5, Experimental Testbed for the Detection of Cyber-Physical Attacks.** This chapter presents a training cyber-physical testbed to validate the mechanisms proposed in this dissertation. The chapter fulfills *Objective 1.6*.
- **Chapter 6, Conclusion and Future Research.** This chapter concludes the dissertation and provides some future research lines that may be undertaken.
- **Appendix A, Cyber-Physical Countermeasure.** This appendix defines a sample countermeasure approach to drive the response of a cyber-physical system protected by our detection mechanisms, to continue working in a degraded mode, after the detection of cyber-physical attacks.

2 State of The Art

2.1 Literature Definitions

2.1.1 SCADA Technology

Supervisory Control and Data Acquisition (SCADA), is a technology to monitor industrial and critical infrastructures based on cyber-physical systems. In other words, SCADA technologies allow to take into account control industrial environments of critical cyber-physical infrastructures (e.g., energy distribution and transport systems). The SCADA technology was conceived for centralized and isolated processes. Nowadays, it is more distributed and vulnerable to cyber attacks. SCADA systems are typically composed of three well-defined types of field devices: (1) Master Terminal Units (MTUs) and Human Machine Interfaces (HMIs), located at the topmost layer and managing all communications; (2) Remote Terminal Units (RTUs) and Programmable Logic Controllers (PLCs), which control and acquire data from remote equipment and connect with the master station; and, finally, (3) sensors and actuators.

The MTUs of a SCADA system are located at the control center of the organization. The MTUs give access to the management of communications, collection of data (generated by several RTUs), data storage, and control of sensors and actuators connected to RTUs. The interface to the administrators is provided via the HMIs.

RTUs are stand-alone data acquisition and control units. They are generally microprocessor-based devices that monitor and control the industrial equipment at the remote site. Their tasks are twofold: (1) to control and acquire data from process equipment (at the remote sites), and (2) to communicate the collected data to a master (supervision) station. Modern RTUs may also communicate between them (either via wired or wireless networks).

PLCs are small industrial microprocessor-based computers. Most significant differences with respect to an RTU are in size and capability. An RTU has more inputs and outputs than a PLC, and much more local processing power (e.g., to postprocess the collected data before generating alerts towards the MTU via the HMI). In contrast, PLCs are often represented by pervasive

sensors with communication capabilities. PLCs have two main advantages over commercial RTUs: (1) they are general-purpose devices enforcing a large variety of functions, and (2) they are physically compact.

Finally, sensors are monitoring devices responsible for retrieving measurement related to specific physical phenomena and feed them to the controller. Sensors typically convert a measured quantity to an electrical signal, which is later converted and stored as data. Sensors can be seen as the input function of a SCADA system. The data produced by sensors are sent to the upper layers via RTUs/PLCs. Actuators are control devices, in charge of managing some external devices. Actuators translate control signals to actions that are needed to correct the dynamics of the system, via the RTUs and PLCs.

2.1.2 Networked-Control Systems

Networked-Control Systems (NCSs) are spatially distributed systems whose control loops are connected through a communication network. The communication network connects the different components of the control system, i.e., the controller, sensors, and actuators. Examples include smart grids, smart vehicles, and water distribution systems. The use of communication networks to connect the different components of control systems adds more flexibility in the systems and reduces the implementation cost of new installations.

However, the use of communication networks to decentralize traditional control systems comes at the price of an increased control design complexity. For instance, the analysis and design of the overall system has also to deal with new theoretical challenges due to loss of measurements and time-varying sampling [17]. The integration of the control system (often referred as physical-space) and the communication network (cyber-space) creates a new degree of interaction between these two domains [18].

As will be discussed in Section 2.2, the communication protocols used in traditional control systems are required to comply with the constrains imposed by industrial standards (e.g., to cover regulation roles such as delay and faults). Some of the studied protocols (e.g., Modbus, DNP3, and Profinet), are not designed to provide security from a traditional information or network perspective. However, current NCSs use these protocols over TCP/IP or UDP/IP communications (e.g., Modbus over TCP, DNP3 over TCP or UDP, and Profinet over TCP). Although such combinations can provide some security elements at either their transport or network layers, this is not enough to ensure control-data protection. At the same time, traditional control systems come with already existing mechanisms to handle failures. Such mechanisms are expected to detect faults and avoid accidents. Nevertheless, traditional control mechanisms cannot detect intentional actions from malicious adversaries holding enough knowledge about the systems. We present some representative examples in the following section.

2.2 SCADA Protocols for Networked-Control Systems

Industrial control protocols for SCADA environments shall comply with the constraints imposed by industrial standards, in order to cover regulation rules such as delays and faults [6]. However, few of them provide security features in the traditional ICT security sense. In this section, we provide a brief summary of some representative protocols. We summarize some details about the security features that they may offer.

Modbus

The Modbus protocol was created in the 70s by Modicon, an American company created in 1968 and absorbed by Schneider Electric. Nowadays, it is one of the most spread protocols, probably due to its simplicity and its free license.

The Modbus protocol was initially conceived for serial communications. Since 1999, it has been adapted to work over TCP/IP as well. The use of Modbus over TCP/IP allows using SCADA components in heterogeneous environments (i.e., working over IP or serial networks). Moreover, it is possible to use gateways to convert Modbus/TCP messages to/from Modbus/ASCII. This comes with some disadvantages, since the heritage from Modbus/ASCII to Modbus/TCP imposes restrictions, such as restricted length of messages, necessity of identifier units, reduced size of fields, etc.

Modbus defines a data exchange between a Master and a Slave. Typically, the Master is deployed over an MTU (Master Terminal Unit) and the Slave over an RTU (Remote Terminal Unit). The role can also be exchanged, so that RTUs can also interrogate MTUs. In the end, the protocol relies on a Master querying a Slave, and the responses of the Slave to the Master. This query/answer scheme is the core of the protocol.

From a security standpoint, Modbus does not integrate traditional ICT protection features. Some exceptions are listed below:

- *Availability*: Modbus/TCP may use some exception codes (e.g., ECO4: Server Failure, EC06: Server Busy) as the response of a query from an unavailability Slave. For instance, a Master can point out to availability issues in the absence of responses from one or several Slaves, or if their responses are error codes. Error handling is performed at the application layer. The availability of a given equipment is also related to the implementation of the layers below Modbus (e.g., TCP/IP layers) and the nature of the media shared for the exchange of data.
- *Integrity*: The integrity of a Modbus message is validated using the TCP layer for Modbus/TCP or by adding a control field (e.g., Cyclic Redundancy Check or CRC) for Modbus/Serial. Nevertheless, without authentication of the message, malicious actions can modify the message and recalculate valid CRCs. This kind of validation must be seen only

as a protection against transmission errors. Malicious modification of registers, e.g., time windows, is complex but possible. Replay attacks and, in general, integrity attacks, are still possible.

- *Confidentiality*: The Modbus protocol does not implement encryption. Nevertheless, it is possible to implement encryption by encapsulating Modbus/TCP messages under TLS or IPsec tunnels.

We recall that confidentiality is not considered as a crucial property in industrial environment. In fact, the deployment of encryption solutions can be seen as detrimental given their complexity (e.g., Public Key Infrastructures, manual deployment of keys, etc.) and induce to unnecessary latencies.

AGA-12

Few days after the September 11 attacks, the American Gas Association (AGA) decided to design a standard using cryptography for SCADA to fight against cyber-attacks. Referred to as AGA-12, the objective was to add integrity and confidentiality protection to SCADA communications. The standard also aims at developing and standardizing a cryptography protocol [19], named Serial SCADA Protection Protocol (SSPP). The SSPP protocol is designed as an encapsulation protocol for serial RTU (Remote Terminal Unit) links, and shall be compatible with existing SCADA serial communication protocols (e.g., Modbus/ASCII). A free implementation for serial protocols exists (cf. http://scadasafe.sourceforge.net/). Despite the publication of a proof-of-concept [20], the protocol has not been fully adopted by the community. Indeed, despite a conclusive test in real conditions in an industrial plant [7], AGA-12 comes with high deployments costs, estimated to be over $500 extra per equipment. Another drawback is its dependency to serial technologies, given the wide spreading of other technologies in SCADA environments, such as Ethernet communications.

PROFINET and PROFISafe

Created by PROFIBUS and PROFINET International, and mainly used by Siemens products, PROFINET is a set of protocols operating at different levels of the ISO layers. For instance, PROFINET IO is one of the Ethernet-based protocols associated to PROFINET. It is implemented over TCP/IP layers, and allows real-time communication and self-configuration. All equipment implementing PROFINET IO must be certified by the PROFIBUS organization. This certification monitors compliance of software, data model and integrity in a PROFINET IO environment.

In 1999, the first security extension of PROFINET was released. Referred to as PROFIsafe, it leverages from PROFINET IO, acting as one of its upper layers. This allows its deployment over less secure networks maintaining acceptable error rate such as WIFI and Bluetooth, while ensuring high availability and backwards compatibility for legacy equipment. Legacy operations

can yet use the standard layer, called Black channel, while other operations requiring safety properties, can use the new layers. Such new layers include:

- A 24-bit incremental counter to control the continuity of messages to the destination. This value is not transmitted; only one counter increment bit is set in the message.
- A *timing out* to monitor the acknowledgment.
- A 16-bit codename between transmitter and receiver for the authentication of the peer-to-peer connection. This value is fixed and is not transmitted.
- An integrity check on 24 or 32 bits calculated using the payload of the message and the value quoted above. This CRC adds a new control, allowing independence from the lower layers.

A study by Åkerberg and Björkman [8] describes some flaws in the protocol routines associated to the generation of CRCs. Indeed, PROFIsafe meet standards where intentional attacks are not considered a risk. The protocol does not integrate cryptographic features. It only considers protection to cover from unintentional faults. It should not be considered a protection layer against cyber attacks. Indeed, the PROFINET Safety Guide [21] indicates the use of VPNs whenever ICT security is required.

It is important to emphasize that PROFIsafe has been designed to ensure safety, hence protecting the PROFINET protocols against malfunction (e.g., transmission errors). It does not ensure security against intentional malicious acts.

DNP3 and DNP3-SA

Created in 1990 by Westronic, Inc., DNP3 (DNP version 3) is open-source since 1993 and property of DNP3 User Group. DNP3 is presented as a robust, efficient and modern SCADA protocol. It can send and receive multiple data types in a single message, segment the messages into fragments to handle error detection and enable effective error recovery, transmit data without solicitation, synchronize clocks, etc. It includes a security extension, referred to as DNP3-SA (DNP3 Secure Authentication).

The first version of the DNP3-SA extension, published in 2007, adds new security features to DNP3, such as protection to replay attacks by ensuring message integrity and authentication. It does not provide protection against confidentiality attacks. DNP3-SA is nowadays in its fifth version (IEEE-1815-2012), which is not backward compatible. Nevertheless, this is the only recommended version.

Regardless of the type of underlying transport layer (TCP/IP, UDP/IP or Serial), the security features in DNP3-SA are implemented at the application level, and designed as a protocol

extension. In other words, the new features are defined as new function codes of the original DNP3 protocol suite. No previous function codes are modified. In this way, all legacy monitoring and diagnostic tools for DNP3 are still valid. In addition, DNP3-SA is compatible with legacy devices that do not require from security support.

The DNP3-SA is expected to be highly scalable. It allows changing the algorithms, keys sizes, and others parameters to meet future conditions of state-of-the-art installations. However, both DNP3 and DNP3-SA are relatively complex protocols. In 2013, a study carried out by Automatak [22], revealed many implementation problems within DNP3 equipment. The DNP3-SA is a relatively young extension of the protocol, and the first DNP3-SA products may present vulnerabilities. It has not been largely deployed yet and, given the pace of industrial systems (whose upgrades are often superior to decades), it may take quite long before DNP3-SA is fully tested over large environments.

IEC-60870-5-104

Created in 2000 by IEC (International Electrotechnical Commission), IEC-60870-5-104 is an international standard protocol used for monitoring energy systems among other control systems as petrochemical or water treatment. This protocol is the extension of the *60870-05-101* protocol over TCP/IP. The main differences between both protocols are: 1) the number of functions supported (IEC 60870-5-104 supports less functions than IEC 60870-5-101); and 2) data transmission (IEC 60870-5-104 allows to transmit data simultaneously between servers and devices). Despite these differences, both protocols are very often combined, since they can be easily synchronized.

From an ICT security standpoint, IEC-608-5-104 builds upon IEC 62351-5. In turn, IEC 62351-5 provides different solutions for serial and Ethernet communications. More specifically, the security measures described in *IEC 62351-3:2014* [23], are used by *IEC-608-5-104* in order to provide integrity, confidentiality and authentication. Security features are based on TLS (Transport Layer Security) security, e.g., to handle eavesdropping and replay attacks, as well as to providing integrity and confidentiality properties. It also uses message authentication against man-in-the-middle attacks, and X509 certificates for node authentication, e.g., to handle spoofing attacks against node authentication.

EtherNet/IP

EtherNet/IP is an industrial Ethernet protocol, using CIP (Common Industrial Protocol) over standard Ethernet Frames (Ethertype 0x80E1). It is nowadays maintained by ODVA Inc. (Open DeviceNet Vendors Association, Inc.). EtherNet/IP was reported as the most widely deployed protocol over industrial Ethernet environments, according to IMS research [24] — about 30% of the actives nodes in their study, i.e., about five million industrial nodes [25]).

Communication over EtherNet/IP can be either in an unconnected mode, using TCP/IP in a client/server model, or they can be in connected mode. In that case, resources are reserved to create a link between two users; UDP/IP and multicast transmissions are employed to make latency as small as possible to enforce real-time constraints.

In EtherNet/IP, a device will send cyclical messages to a controller. These messages are made of several objects defined by EtherNet/IP, which model the sending device [26]. Examples of objects are: motor, analog input, etc. A vendor can create a custom set of objects for its device. However, each device must maintain the following three main parameters: *identity object* (information about a device e.g. vendor/device ID), *router object* (information about the destinations/equipment the device know in the network) and *network object* (it describes the network interfaces of an object).

EtherNet/IP inherits from Ethernet all its security issues. Besides, the connected mode using UDP loses all the mechanisms of TCP for reliability, ordering and integrity. Thus, we can consider some concerns such as [27]: the identity, router and network objects disclose several details regarding the devices and the network configuration, which can be leveraged by an attacker to devise an attack strategy; an attacker can inject frames in the cyclical real-time messages as it is mainly in connected mode; by doing this, the attacker can easily control actuator thanks to the application objects, or interrupt the cycle by sending for instance wrong timestamps, maybe putting a device in safety mode (DOS); the fact that most objects are pre-defined is convenient but can lead to massive attacks (e.g., worms) as most factories work similarly. As a consequence, attack automation in this context is easier to be implemented. However, some safety modules can be used with EtherNet/IP, such as opensafety [28] and CIPSafety [29]. And recently, CIP has implemented *CIPSecurity Phase 1* based on TLS (Transport Layer Security) and DTLS (Datagrams Transport Layer Security) [30]. The goal of CIP Security is to allow connected devices protecting themselves from malicious manipulations in communication.

Ethernet Powerlink

Ethernet Powerlink is an open protocol released by Austrian company *B&R* in 2001, and standardized by the Ethernet POWERLINK Standardization Group (EPSG) in 2003. This protocol, based on the ISO/OSI layer model, supports Master/Slave communications relationships.

This protocol is enforced over Real-Time Ethernet (RTE) environments. It also provides a mechanism in order to manage network traffic [31]. The mechanism, called *Slot Communication Network Management* (SCNM) has a Managing Node (MN) and Controlled Nodes (CN). It allows to separate the schedule into an isochronous phase and an asynchronous phase. In the isochronous phase, the protocol transmits time-critical data and reserves the asynchronously phase to transfer IP-base protocols like TCP or UDP. Moreover, this division provides transmission of isochronous and asynchronous data without interference and guaranteeing the communication timing.

The aforementioned mechanism, on the one hand, provides to the protocol a higher synchronization among the networked nodes, solving a lot of real time transmission problems. On the other

hand, its isochronous phase prevents certain attacks, such as attacks which flood the network with malicious data messages. That is possible thanks to its ability to avoid the collisions. Otherwise, from a security standpoint, this protocol has not security mechanism against denial of services, impersonation attacks, and neither against some data injection attacks [32].

2.3 Detection and Mitigation of Cyber-Physical Attacks

2.3.1 Cyber-Physical Attacks

The use of communication networks and IT components in traditional control systems opens new vulnerability issues. The attacks against these setups are named cyber-physical attacks. These threats could harm the physical processes through the network. Teixeira et al. propose a taxonomy of cyber-physical attacks, based on the following three dimensions: *a priori* knowledge of the adversary about the system, degree of disruption, and degree of disclosure. The first dimension, *a priori* knowledge of the adversary, is used to represent powerful attacks that allow evading adversaries. *Disruption resources*, shows the capability of an adversary in affecting the target system by violating its integrity or availability. Finally, *disclosure resources* represents the capability of the adversary to obtaining sensitive information during the attack, e.g., by violating data or control confidentiality.

Some representative attacks are: (i) replay attacks, not needing previous knowledge but needing the capability to violate the integrity, the availability and the confidentiality of the system; (ii) injection attacks with *a priori* knowledge and the capability to violate the integrity and the availability; and (iii) covert attacks with full capability and knowledge of the system.

Next, we list a more elaborated list of cyber-physical attacks reported in the literature. Table 2.1 summarized the list and provides the appropriate references.

- *Stealth Attack:* Stealth attack [33, 34], also called false-data injection attack [35]. To carry out this attack, the adversaries modify some sensors reading by physical interferences, at individual meters or by the communication channel. The adversaries in order to carry out this attack, have to know the behavior of the system (system dynamic, command signal), and the control detection threshold. The goal of this type of attacks is to disrupt the behavior of the system (the states of the system, cf. Section 2.4). They use false-data injection in order to generate a wrong control decisions able to cause a malfunction in the system. To accomplish this objective, the adversaries have to inject false-data able to not affect the residue (cf. Section 2.4), i.e., data which not altering the sensor measurement variations, drive slowly the control decisions out of the correct behaviour. This type of attacks are undetectable with a monitor detector which verify only the sensor measurements. They are detected if the monitor detector verifies the sensor measurements and the compatibility of the measurements with the system dynamic. Adversaries able to conducting this type of attacks, as depicted in Figure 2.1, may use the following attack techniques:

- *Surge Attack:* Surge attacks are stealth attacks used by adversaries in order to maximize the damage at the earliest. The only constraint of this type of attacks are that the data sent to the controller cannot generate a residue which exceeds the threshold of the detector. This is necessary in order to keep the characteristic of stealthy attack.
- *Bias-Injection Attack:* Adversaries conducting this attack, as depicted in Figure 2.1, are expected to modify some sensors readings by physical interferences, in the individual sensors or in the communication channel [36]. The use of the $\bigoplus$ symbol in the figure shall read as *sum of signals*. The adversaries inject a bias in the sensors readings, y_t. The goal is to lead to wrong control decisions and cause large-scale malfunction. A sophisticated variation of the bias-injection attack is the *geometric-injection attack*.
- *Geometric-Injection Attack:* The geometric-injection attack is an intelligent *bias-injection attack* whose bias-injection is gradual. This type of attacks are the mixture between surge attack and bias attack. The attacks may remain undetected when data compatible with system dynamics are injected, potentially leading to irreversible damages. This type of attacks are also undetectable in a system with unstable modes, since they make unobservable the disruptions carry out in these modes.

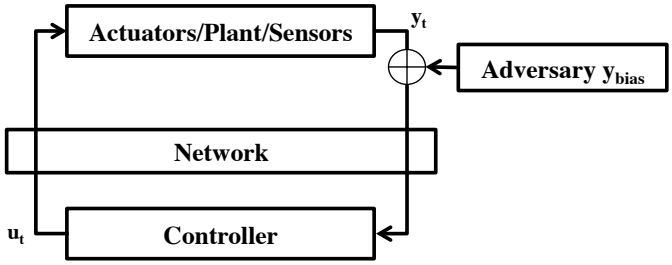


Figure 2.1 – Stealth attack using either surge, bias or geometric attack techniques.

- *Replay Attack*: Figure 2.2 shows adversaries conducting the attack by modifying some sensors readings (e.g., by replicating previous measurements, corresponding to normal operation conditions). Then, the adversaries modify the control input to affect the system state. These adversaries are not required to have the knowledge of the system process model, but the access to all the sensors is required to carry out a successful attack. This type of adversaries are undetectable with a monitor detector which only verifies sensors measurements. To detect the attack, it is required to add some protection to the input control signal u_t [37].
- *Covert Attack*: Adversaries, depicted in Figure 2.3, read and add to both, the physical system and the controller output, the control data and the sensors measurements. The adversaries need *a priori* knowledge about the system process, i.e., the behaviour of the physical system as well as the behaviour of the feedback control. This type of adversaries are considered undetectable, if measurements are compatible with the physical process.

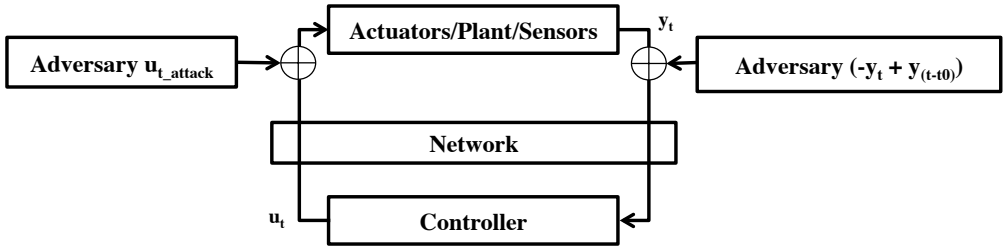


Figure 2.2 – Replay attack.

In other words, the attack cannot be distinguished from the regular system operation [38]. Hoehn *et al.* propose in [39] the insertion of a periodical modulation matrix in the path of the control variables to detect these adversaries. Nevertheless, this detection technique is not able to detect the attacks, if the adversaries are able to know the period when the modulation change.

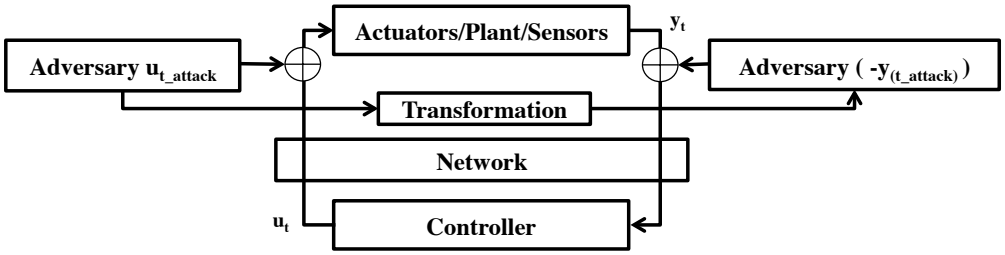


Figure 2.3 – Covert attack.

- *DoS Attack*: The goal of a Denial of Service (DOS) attacks is to disrupt the communication between the MTU and the RTUs or between the RTUs and sensors or RTUs and actuators. This type of attacks break the link between the different parts of the system in order to disrupt the feedback control [40]. These attacks are able to disconnect the controller from the physical device. The isolation avoids to monitor the process and it leaves the system vulnerable against a possible attacker's action.

- *Zero Dynamics Attack*: This attack uses vulnerabilities present in the dynamics of the system with respect to properties used to be able to monitor and control the behavior. More specifically, making an unobservable state unstable (cf. Section 2.4). This vulnerability allows the attacker to disrupt the unobservable part of the system without being detected by the controller [41, 42]. A solution to avoid this kind of attacks is to update the architecture of the system in order to make all the states observable, e.g., deploying more sensors in order to avoid unobservable situations into the system.

- *Dynamic false data injection attacks*: These attacks focus on systems with unstable states. They aim to modify the measurements of the sensors to make some unstable states unobservable [43, 44]. These attacks exploit a vulnerability of the systems as *Zero Dynamic*

attacks. The solution to avoid these attacks is to update the architecture in order to remove the vulnerabilities.

- *Command injection Attacks*: These attacks use the protocols and devices vulnerabilities in order to inject false commands into the control systems, e.g., overwriting remote terminal programs or registers. Signature based IDSs, such as SNORT [45], are the detection techniques used against these attacks.

Attack Name	**Summary**	**References**
Replay	Adversaries replay previous measurements (corresponding to normal operation conditions) and modify control inputs to disrupt the system.	[9], [37], [36]
Dynamic false-data injection	Adversaries drive the system to unstable states, by using system vulnerabilities.	[43], [44]
Stealth or false-data injection, using Bias, Surge, or Geometric attack techniques	Adversaries disrupt the behavior of the system, by injecting faulty data constructed to evade feedback-control detectors.	[33], [34], [36], [46], [47]
Covert	Adversaries hold complete knowledge about the system dynamics, to impersonate the feedback controller and evade fault detection.	[38], [39], [48]
Zero Dynamics	Adversaries make unobservable an unstable state of the system using controller vulnerabilities.	[41], [42]
Denial of Service	Adversaries disrupt communication links, to later control the system.	[40], [49]
Command injection	Adversaries inject false control commands, to disrupt control actions or system settings.	[45], [49], [50]

Table 2.1 – Description of representative cyber-physical attacks reported in the literature.

2.3.2 Detection and Countermeasures

In this section, we enumerate different detection methods and countermeasures. We focus on injection attacks from a control point of view. In control theory, methods as fault detection mechanisms, isolation techniques or reconfiguration methods, as well as hardware redundancy (e.g., adding more sensors), or software redundancy [51] to identify the problems in the system

are not new. The research in this field until recently has focused on the detection and response of accidents, random faults or equipment failures, but not attacks. Events like Stuxnet [52] have activated again this research area into security topics. Recently, the control community started to address, as well as, detection of attacks in industrial systems. We present in the following lines a survey of works about detectors and countermeasures from the control community. Table 2.2 summarizes the survey.

In [9, 53] Mo *et al.* propose the use of watermark-based detection by adapting traditional failure detection mechanisms in order to detect replay attacks. The watermark is added to the control measurements to verify using the detection mechanism that the sensor measurements are not replayed measurements, i.e., the control measurements with the watermark have to be correlated with the sensor measurements. Miao *et al.* in [54] improve the performance of this detection mechanism. They present a suboptimal algorithm against replay attacks, using a stochastic game approach. This suboptimal algorithm combines the watermark-based detector proposed by Mo *et al.* and the zero-sum stochastic game proposed by Zu *et al.* in [55]. In the same way, Do *et al.* [13] formulate the attack detection problem as a transient changes detection problem in stochastic-dynamical systems. This detector is based on the knowledge of the system's behaviour and its stochastic variations to detect data manipulation. Using this detection control algorithm allows to protect the system against attacks able to carry out malicious actions in a short period of time whose goal is to disrupt the safety-critical applications.

A hybrid game-theoretic framework is presented by Zhu *et al.* in [60, 61], where a cross-layer coupled design is created between physical and cyber detection layers, to maximize the chances of identifying unexpected (security) events. Authors emphasize that control and defense strategies depend on both cyber and physical states. Authors also conclude that the effects of the network can manage the performance in the interdependence between cyber and physical layers. Rakesh *et al.* [46] present the idea of protecting only a set of *basic measurements*, validating that it is

Detection and Countermeasures	**Description**	**References**
Watermark-based detector	Adaptation of traditional fault detection mechanisms to detect, as well, attacks.	[13], [37], [46], [54], [55]
Signal-based and model-based detector	Use of signal statistical properties and system behavior to detect attacks.	[12], [56]
State relation-based detector	Correlation of system states together with system behavior, to identify anomalies.	[57], [58], [59]
Cross-layer based resilient detector	Combination of control and cyber techniques in a single cross-layer intrusion detection system.	[43], [46], [60], [61]

Table 2.2 – Detection and countermeasures in cyber-physical systems.

enough to detect the attacks. This protection has to be against physical and network tampered actions. This set, composed of the minimum number of measurements, is not unique in a system, and it could be dynamic if the topology changes. Arvani *et al.* [12] describe a signal-based and model-based intrusion detector model to detect and identify random signal data-injections attacks using Wavelet analysis in order to exploit the statistical properties of the signal as well as a dynamic model of the system with a chi-square detector to identify the anomalies. And Lokhov *et al.* [56] propose a matrix to manage the correlation between the different signals. The target of this matrix is: (i) detecting the anomalies using spectral methods; (ii) localizing the nodes where the anomalies happened; and (iii) identifying the functional role of the anomaly. Another combined technique is presented by Wang *et al.* in [57]. They propose a relation-graph-based detector scheme in order to detect false data injection attacks. A correlation model extracts the relation among the different variables of the system in order to create a graph model with the possible valid system states. The correlation model uses a forward correlation which is a static structure not affected by the time and a feedback correlation which is a dynamic structure depending on time. This correlation model allows to create a more complete correlation graph. Dehghani *et al.* [58] present an static state estimation algorithm able to detect the anomalies in the states under integrity attacks. Chen *et al.* [59] present an anomaly detection distributed algorithm using the spatiotemporal correlation existent in the physical systems and a Gaussian distribution to measure the correlation error.

Pasqualetti *et al.* [43] introduce the use of geometric control theory to optimize cross-layer resilient control systems. They conclude that by using a geometric model of the system is possible to solve problems such as non-interacting control, fault detection or the estimation of the state in the presence of unknown inputs. From a response standpoint, Cardenas *et al.* [33] study the possibility to creating an automatic response mechanism based on feedback control estimation. Authors propose an anomaly detection module able to use the aforementioned detector techniques to generate an automatic response. Due to the problem between the usability and security in these systems, authors conclude that this automatic mechanism would be the first step before a human intervention. The required balance between *security* and *usability* metrics in anomaly detection algorithms, is discussed in a survey about physics-based attack detection techniques proposed by Urbina *et al.* [62]. While their security metric provides the ability to detecting the attack, the usability metric provides the ability to labeling correctly normal events in order to not reduce the number of false alarms.

2.4 Control Theory in Industrial Control Systems

Industrial Control System (ICS) is the term used to define the hardware and the software employed to administer and command industrial systems. For instance, devices as sensors and actuators, as well as, networks and feedback control systems. This complex framework allows to create a model able to manage and control the physical evolution of the states of a system. It is worth noting that to control these states is a challenge, since these systems follow the laws of nature, e.g., energy, water or oil systems [62].

The target of this kind of complex systems is to use the physical properties of the system in order to create a model used as 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. Figure 2.4 shows an ICS. The *system* element represents the physical layer; the devices used to modify the physical parameters; and the *actuators* and *sensors* allow to 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 from the actuator.

Reaching the model used by the feedback control, in order to obtain the reference and generate the control error at each time, is a very well-known problem in the control domain, where techniques have been developed in order to obtain these models [63, 64, 65, 66] and to create feedback control [67, 68, 69]. These systems are not centralized, since the feedback controller is not placed in *sensors*, *actuators* and neither in the *physical system*. The loop generated by these systems is closed through the network.

New industrial control systems frameworks have a lot of advantages that we have enumerated before in this chapter, but new interdisciplinary challenges have been opened. Distributed framework and transport of the control data through the network have generated the necessity to revisit the control strategies as well as a security challenge because of the threats that someone is able to capture the data in order to violate the confidentiality, the availability or the integrity of the system. In these systems the main difference between faults and attacks is that attacks are intentional actions generated by adversaries. These actions can be carefully designed in order to disrupt the system without being detected. Otherwise, faults are phenomena happening randomly in each component of the system [13].

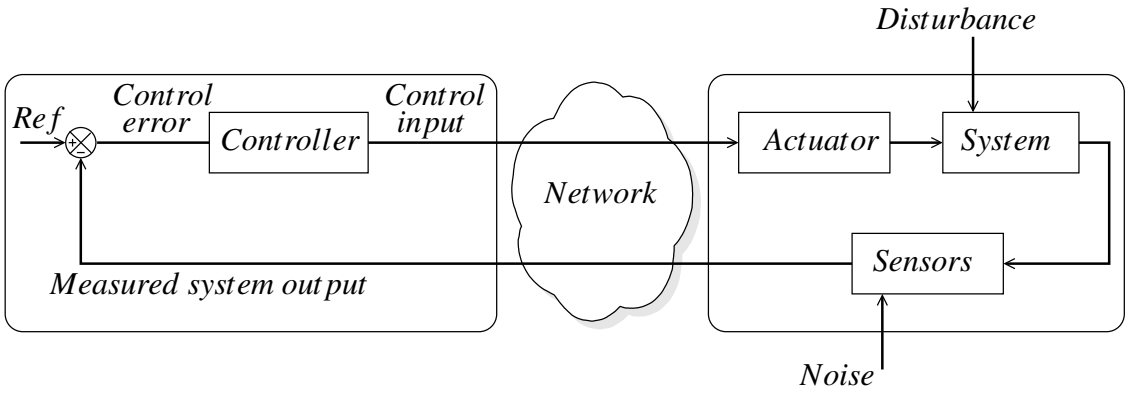


Figure 2.4 – Diagram of a networked feedback control system.

2.4.1 System Dynamics

This section is focused on the definition and description, using mathematical expressions and physical properties, of the behavior of the system. We use in this dissertation the discrete-time approach of the networked control systems, following the definition proposed by Heemels *et al.* in [70], assumed relevant from a practical point of view, since the controllers are normally implemented in discrete-time form, i.e., digital form. This discrete-time approach has to be applied in a *linear context* [70]. Taking into account the above, we can define mathematically these systems as a *linear time-invariant* (LTI) discrete system given by:

$$x_{t+1} = Ax_t + Bu_t + w_t \tag{2.1}$$

$$y_t = Cx_t + v_t \tag{2.2}$$

where $x_t \in \mathbb{R}^n$ is the vector of the state variables (or state) at the t-th time step, $u_t \in \mathbb{R}^p$ is the control signal, and $w_t \in \mathbb{R}^n$ is the *process noise* that is assumed to be a zero mean Gaussian white noise with covariance Q, *i.e.* $w_t \sim N(0,Q)$. 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. In Equation 2.2: $C \in \mathbb{R}^{m\times n}$ is the output matrix. The value of the output vector y_t represents the measurement produced by the sensors that is affected by a noise v_t assumed as a zero mean Gaussian white noise and covariance R, i.e., $v_t \sim N(0,R)$.

The aforementioned equations define mathematically the behavior of a *physical system*. These equations are used by the feedback control to generate a closed-loop system. Closed-loop systems, in opposition of the *open-loop systems*, are systems whose output has an influence to 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, based on three simple functions: 1) a proportional function which controls the *system input*, when there are no errors; 2) an integral function, used in order to eliminate steady state offsets; and 3) a derivative function, which permits to compute deviation errors [71].
- Linear-quadratic-Gaussian (LQG) controllers, a well-known technique for designing optimal dynamic feedback control laws. This optimal solution combines a Kalman filter (linear-quadratic estimation) with a linear-quadratic regulator. These two components are independent, but work together taking into account the measurement noise and process disturbance.

In our work, we assume LQG controllers, since the feedback provided by this type of controllers holds better results than PID controllers [72]. In other words, we assume the modeling of cyber-physical systems as linear time-invariant (LTI) discrete systems, whose feedback control mechanisms are regulated by LQG controllers. To build these systems, it is mandatory to accomplish the properties that we detail in the following section.

2.4.2 Properties

Cyber-physical systems have to follow some properties in order to be managed and controlled. The three most important properties are the following:

- Stability: without loss of generality, a system is stable if the output signal response to a bounded input signal is also bounded. Otherwise, the system is unstable. In a cyber-physical system, the open-loop of the physical system can to be unstable, nevertheless, it is necessary that the closed-loop is stable, i.e., the overall system composed of physical elements and the feedback controller has to be stable. Time delays and packet dropout have to be considered to handle the stability in a cyber-physical system [70, 73].
- Controllability: is the property of a system that allows to know if we have a full reachability map of the system. This map concerns all the states that we can steer.
- Observability and output controllability: a system is called observable, if it is possible to create a map of their states from the output of the system without knowing the initial state (often refereed to as *behavior*). Otherwise, the system is unobservable. Observability and output controllability are dual properties.

2.4.3 Control Strategies

Control theory is a well-known topic, where the evolution of the technology, i.e., the networked control systems and the cyber-physical industrial systems, has been the main motivation to create new control policies to manage these systems, keeping the control features. A wide range of research has been reported in the literature focusing on managing these new technologies in order to preserve the control properties of the systems. They have generated new challenges in control/estimation, signal processing, and communication in order to solve the new performance problems as limited power transmission, bandwidth constrains, packet drop, delay or security. The networked control systems have motivated to consider control, estimation and communication in a unified way [74], in order to solve problems as performance or security.

Among control strategies in cyber-physical systems, we focus on the strategies depending on the transmission policy that we use and modify in this thesis in order to create a more secure policy, considering, not only the possible faults of the control system and networks, but also the attacks that are also faults but carried out by an external entity whose goal is to disrupt the system. We can classify the transmission policies in: sampled-data control or event-triggered control. Into the sampled-data policies, we find mono-frequency sampling, i.e., the same sampling frequency for all the channels (in Figure 2.5, case 1), or multi-frequency sampling (in Figure 2.5, case 2), i.e., different sampling frequencies depending on the channel (sensor/controller or controller/actuator) [75]. Event-Triggered Control (ETC) strategy has been also studied depending on the policy to send the events; 1) Continuous Event-Triggered Control (CETC), [76], represented in Figure 2.5 with the case 3. In this control strategy, the event-triggering condition is monitored continuously,

and sent to the controller; 2) Periodic Event-Triggered Control (PETC) [77], shown in Figure 2.5, case 4. This strategy allows to use the communication resources only when they are needed to assure the stability or the performance of the system. This strategy creates a balance between a policy which monitors continuously and the traditional event-triggered policy, where the sampled-data sent are generated by abnormal events; or 3) Stochastic Events-Triggered Control (SETC), [78], reported in Figure 2.5, case 5. The schedule of this strategy is to improve the previous one, reducing the frequency of communication between the sensors and the controller. This strategy guarantees the stability and performance of the system obtaining a desirable trade-off with the communication rate. Figure 2.5 shows a system diagram where f_s and f_c are the frequencies used by the sensors and controller respectively in order to process data. f_{ca} is the sampling frequency used in the communication channels between the controller and the actuators. f_{sc} is the sampling frequency used in the communication channels between the sensors and the controller. And $p_a, p_s \in \mathbb{N}$.

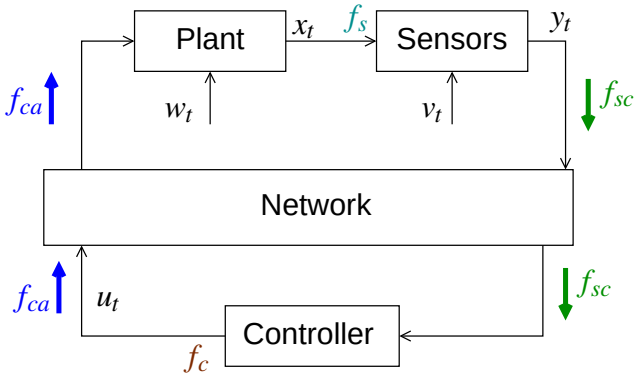


Figure 2.5 – **Cyber-physical System diagram**: *Case 1*: $f_{ca} = f_{sc}$. *Case 2*: $f_{ca} \neq f_{sc}$. *Case 3*: $f_{ca} = f_c$, $f_{sc} = f_s$ and $f_s = f_c$. *Case 4*: $f_{ca} = \frac{f_c}{p_a}$, $f_{sc} = \frac{f_s}{p_s}$. And *case 5*: f_{sc} and f_{ca} →stochastic events.

Control strategy	Description	References
CETC	CETC (Continuous Event-Triggered Control) strategy: sensors sent all the data to the controller, in a continuous manner.	[76]
PETC	PECT (Periodic Event-Triggered Control) strategy: sensors send only data updates, in a periodic manner.	[77]
SETC	SETS (Stochastic Event-Triggered control) strategy: sensors send only data updates, whenever they detect variations in the data they obtain.	[78]

Table 2.3 – Control strategies.

This topic is in line with our research topics, since security in networked control systems includes the management of the control properties through the network to avoid that external entities, e.g., adversaries, may have the capacity to control such properties to harm the system.

2.4.4 Identification Control System Theory

Identification control theory studies how to design and validate models from measured data [65, 66, 68]. System identification methods, which follow this well-known theory, are applied in the industry in order to obtain the model used in the feedback control. We can classify the identification control system methods as follows [79, 80]:

- **Non-parametric identification methods:** Such identification methods are applied to obtain models of the systems (curves or functions), which are not necessarily parametrized by a finite-dimensional parametric vector. There are different methods to carry out this target: transient analysis, frequency analysis, correlation analysis, and spectral analysis [81]. Among the aforementioned methods, we focus on correlation analysis, since is the method used in this thesis. This method lets create a normalized cross-covariance function between output and input employing an estimate of the weighting function. For instance, models as FIR (Finite Impulse Response) models or FSR (Finite Step Response) models, used in signal processing applications.
- **Parametric identification methods:** Such identification methods are characterized as a mapping of the system. The system map is represented by a modeled parameter vector obtained from the recorded data [81]. A non-exhaustive list of parametric identification models found in the literature is following [80]:
  - AutoRegressive with exogenous input (ARX) model
  - Output error (OE) model
  - Autoregressive moving average with exogenous input (ARMAX) model
  - Box-Jenkins (BJ) model

It is worth mentioning that the most popular models used in industrial system identification are: 1) the FIR model, in the non-parametric identification methods; and 2) the ARX model, in the parametric identification methods, since both have the properties to be simple and reliable enough, to predict the behavior of the system, with a lineal error [82, 83].

2.5 Cyber-Physical Training Testbeds

From a performance and a security point of view, cyber-physical training testbeds are necessary to validate existing and experimental new features designed to improve these systems. Chabukswar et al. [84] discuss about potential constraints and characteristics that it is necessary to consider in order to create a valid testbed. In [84], they show that these systems share the network with different kind of applications. This adds a challenge both to the security and to the real-time aspect of SCADA networks. Graham et al. [85] highlight how most of the developed solutions are isolated and individualistic, meaning they can only be used in very specific applications. At

the same time, research on cyber-physical systems has progressed substantially resulting in a large number of testbeds developed and established in the literature. In the sequel, we present a non-exhaustive list of solutions reported in the literature.

Myat-Aung present in [86] a Secure Water Treatment (SWaT) simulation, using Labview and Simulink, and a testbed assuring safety properties, i.e., non damage to the physical system. The testbed and simulation, based on a security standard (ISA-99) proposed by Industrial Automation and Control Systems (IACS), permit to test defense mechanisms against a variety of attacks in order to improve the standard security. Siaterlis et al. [87] define a cyber-physical Experimentation Platform for Internet Contingencies (EPIC) that is able to study multiple independent infrastructures and to provide information about the propagation of faults and disruptions. More specifically, they propose an Emulab-based testbed created from two different parts, the cyber part and the physical part. The target of this separation is to generate accurate assessments of the effect that an attack can produce in both parts of the system.

Green et al. [88] focus their work on an adaptive cyber-physical testbed where they include different equipments, diverse networks, and also business processes. They use this testbed to analyze the remote devices using techniques such as, fuzzing technique in the PLCs in order to discover vulnerabilities that an adversary can harness to disrupt the system. Or, in the control center they carry out test against adversaries able to modify the data memory, more specifically the history of data received from the sensors, modifying at the same time the future decisions based on the historical data and sent to the physical system. Yardley reports in [89] a cyber-physical testbed based on commercial tools in order to experimentally validate emerging research and technologies. The testbed combines emulation, simulation, and real hardware to experiment with smart grid technologies. In the same way, Sanchez Arago et al. present in [90] a framework able to assess remotely the security of these systems. To obtain this target, they have developed testing methodology to provide a guideline of security assessments considering: 1) the architecture of the networked control systems regarding real system requirements, in order to make the testbed as similar as possible to a real system; 2) Taking into account the existent standards and documentation about security in these systems; and 3) using three different types of security assessments: black box, gray box, and white box.

Krotofil and Larsen analyze in [91] the security in cyber-physical systems, focusing their work on the possibility that a cyber attack causes tangible effects in the physical system. They show several testbeds and simulations concluding that just like a successful attack against their envisioned systems has to manage cyber and physical knowledge, the security in this system has to handle the set of cyber threats and physical threats together, i.e., intrusion detections have to consider cyber layer, physical layer as well as the interactions between the layers. Likewise, Zhang et al. in [92] propose a CPS visualization framework, using QEMU system emulator [93] as visualization machine, and Matlab and Simulink in order to emulate physical components. This framework allows to exploit easily the synergy of cyber and physical layers. They do not propose any security mechanism or discussion, but the possibility to study the synergy between cyber and physical layers is interesting from security point of view.

From a more control-theoretic standpoint, Candell et al. report in [94] a testbed to analyze the performance of security mechanisms for cyber-physical systems. Authors enumerate also the different set of metrics that may be cover the multiple control processes (continuous, discrete, and hybrid processes), as well as the different existent security metrics in order to generate an open discussion from control and security practitioners. McLaughlin et al. analyze in [95] different testbeds to collect the vulnerabilities and the threats of cyber-physical systems. They summary that the public information about working system and configuration, combined with defaults in the framework or in the configuration, may make possible that a resource-rich adversary carries out an attack. Concluding that it is necessary to use pathways between cyber and physical components of the system in order to detect attacks. Also, Koutsandria et al. [96] implement a real time testbed for cyber-physical systems security on the power grid, where the data are cross-checked using cyber and physical elements. For this purpose, they add to the testbed a network intrusion detection system (NIDS) which uses knowledge about cyber and physical parameters in order to launch countermeasures consistent with a cross-validation between the cyber and physical events.

Holm et al. survey, classify and analyze in [97] several cyber-physical testbeds proposed for scientific research. They sum up that in order to obtain a high-fidelity security analysis, the testbeds should to follow requirements as: 1) Defining the objectives of the testbed as well as the relation between these objectives and the configuration of the testbed; 2) Employing virtualization framework to assert the simulation and hardware approach used in the testbed; and 3) Providing a list of requirements accomplished by the testbed and their empirical tests in order to verify these requirements as well as create a comparison with other testbeds.

2.6 Summary

This chapter has provided the background and related work of the dissertation. It has introduced related technologies and surveyed threats, detection techniques and countermeasures analyzed in the literature. It has also surveyed some efforts in the literature in terms of control theory solutions and cyber-physical training testbeds. The aim has been to emphasize the main challenges and problem domains, as well as to anticipate the underlying elements of the contributions that will be presented in the following chapters.

3 Dynamic Challenge-Response Authentication Scheme

3.1 Introduction

In an effort of reducing complexity and costs, traditional industrial control systems are being upgraded with novel computing, communication and interconnection capabilities. Industrial control systems that close the loops through a communication network are hereinafter referred to as *cyber-physical systems*. The adoption of new communication capabilities comes at the cost of introducing new security threats that are required to be holistically handled, both in terms of safety and security (in the traditional ICT sense). The use of inadequate *cyber-physical security* mechanisms may have an adverse effect on a vast number of resources, including assets of government networks and mission critical infrastructures [2]. The associated costs, especially in terms of loss of business opportunities and the expenses for fixing the incidents, are expected to be reduced. As a consequence, the issue of the assessment of *cyber-physical security* mechanisms is a hot research topic.

In this chapter, our focus is centered on integrity issues due to the interconnection between *cyber* and *physical* control domains in networked control systems. More specifically, we focus on the adaptation of physical-layer failure detection mechanisms (e.g., systems for the detection of faults and accidents) to handle, as well, attacks (e.g., replay and integrity attacks conducted by malicious adversaries). The Mo *et al.* scheme [53] relies on the adaptation of a real-time failure detector based on a *linear time-invariant* model of the system. Built upon *Kalman filters* and *linear-quadratic regulators*, the scheme produces authentication watermarks to protect the integrity of physical measurements communicated over the cyber and physical control domains of a networked control system. Without the protection of the messages, malicious actions can be conducted to mislead the system towards unauthorized or improper actions and affect the availability of the system services. We show that the Mo *et al.* detection scheme only works against some integrity attacks. We present two new adversary models that can evade the Mo *et al.* detector. These adversary are classified based on the algorithm used to obtain the knowledge of the system dynamics in order to carry out the attack.

3.2 Contributions

In [9, 53], a watermark-based strategy is proposed to detect replay against cyber-physical systems. This chapter reviews the mechanism proposed in [9, 53] and assesses its performance when a new adversary model, that we name *cyber-physical adversary*, is employed.

The outline of this chapter are summarized as follows. Section 3.3 describes the class of control systems considered in this thesis. Section 3.4 describes the attack detection scheme proposed in [9, 53], while Section 3.5 proposes and define the aforementioned cyber-physical adversaries. Section 3.6 shows methods employed by the adversary to mislead the watermark-based detector. Section 3.7 adapts the detection scheme in [9, 53] to handle the uncovered limitations, and Section 3.8 and 3.9 validate the resulting approach via numerical simulations. Section 3.10 discusses the results. And, Section 3.11 concludes the chapter.

3.3 Problem Formulation

We consider plants of industrial control systems that can be mathematically modeled as discrete linear time-invariant (LTI) systems. It is worth mentioning that a mathematical model provides a rigorous way to describe the dynamical behaviour of a given system. Such class of systems can be described as follows:

$$x_{t+1} = Ax_t + Bu_t + w_t \tag{3.1}$$

where $x_t \in \mathbb{R}^n$ is the vector of the state variables (or state) at the t-th time step, $u_t \in \mathbb{R}^p$ is the control signal, and $w_t \in \mathbb{R}^n$ is the *process noise* that is assumed to be a zero mean Gaussian white noise with covariance Q, *i.e.* $w_t \sim N(0, Q)$. 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_t to the system output $y_t \in \mathbb{R}^m$:

$$y_t = Cx_t + v_t \tag{3.2}$$

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

For such a class of systems defined above, a widely used control technique is the *Linear Quadratic Gaussian* (LQG) approach. The overall goal of an LQG controller is to produce a control law u_t such that a quadratic cost J, that is function of both the state x_t and the control input u_t, is minimized:

$$J = \lim_{n\to\infty} E\left[\frac{1}{n}\sum_{i=0}^{n-1}(x_i^T \Gamma x_i + u_i^T \Omega u_i)\right] \tag{3.3}$$

where Γ and Ω represent positive definite cost matrices [98].

It is well-known that such a control problem has, under some technical conditions, an optimal solution that, thanks to the separation principle, is made of two components that can be designed independently:

1. a *Kalman filter* that, based on the noisy measurements, produces an optimal state estimation $\hat{x}_t$ of the state x;
2. a *Linear Quadratic Regulator* (LQR) that, based on the state estimation $\hat{x}_t$, provides the control law u_t that solves the LQR problem (cf. Equation (3.3)).

Let us briefly illustrate how these two components are designed. The Kalman filter estimates 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:

$$u_t = L\hat{x}_t \tag{3.4}$$

where L denotes the feedback gain of a linear-quadratic regulator (*LQR*) which minimizes the control cost (cf. Equation (3.3)) and it is defined as follows (cf. [9, 53] for further details):

$$L = -(B^T SB + \Omega)^{-1} B^T SA,$$

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

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

3.4 Single Watermark-based Detector

This section briefly describes the detection scheme proposed in [9, 53]. The procedure is applicable to discrete LTI plants controlled by a LQG controller as detailed in Section 3.3.

Before presenting the detection scheme, we provide a definition of the adversary model considered in [9, 53]:

Definition 3.1. *An attacker that has the ability to eavesdrop all the messages containing the sensor outputs y_t and to inject messages with a signal y'_t to conduct malicious actions is defined as a cyber adversary.*

Remark 3.4.1. *It is important to notice that the definition given above does not suppose that the attacker possesses (or makes attempts to gather) any knowledge about the system model, reason why we name such attacker a cyber adversary.*

In the following, we will denote with u_t^* the output of the LQR controller given by Equation (3.4) and with u_t the control input that is sent to the plant (cf. Equation (3.1)). The idea is to superpose to the optimal control law u_t^* a watermark signal $\Delta u_t \in \mathbb{R}^p$ that serves as an authentication signal. Thus, the control input u_t is given by:

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

The watermark signal is a Gaussian random signal with zero mean that is independent both from the state noise w_t and the measurement noise v_t. Such an authentication watermark is expected to detect replay and integrity attacks modeled by the cyber adversary defined above. Now that the optimal control law u_t^* is equipped with the authentication signal Δu_t, a *detector* – physically co-located with the controller – can be designed having the goal of generating alarms when an attack takes place. Towards this end, [9, 53] propose to employ a χ^2 detector, a well-known category of real-time anomaly detectors classically used for fault detection in control systems [99], for the purpose of attack detection.

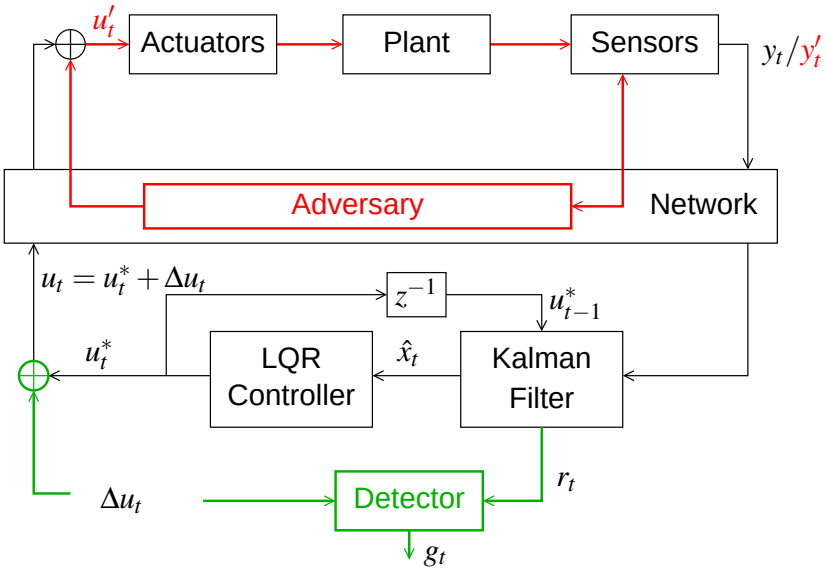


Figure 3.1 – **Watermark-based protection in cyber-physical systems [9].**

Figure 3.1 shows the overall control system equipped with the attack detector proposed in [9, 53].

An *alarm signal* g_t is computed based on the residues $r_t = y_t - C\hat{x}_{t|t-1}$ generated by the estimator. Then, g_t is compared with a threshold γ to decide whether the system is in a normal state. The threshold is tuned to minimize false alarms [9, 53]. The alarm signal g_t is computed as follows:

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

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 (i.i.d.) Gaussian input signal from the sensors.

The system is considered not under attack if $g_t < \gamma$, otherwise if $g_t \geq \gamma$ the system is considered to be under attack and the detector generates an alarm.

3.5 Cyber-Physical Adversary

Let us assume the system employs the detector described in Section 3.4, so that the controller superposes its output with an authentication watermark Δu_t. At steady-state, i.e. after the transient has been exhausted, the output of the system can be considered as the sum of its steady-state value and a component that is due to watermark signal that shall be only known by the controller.

Let us now introduce an enhanced adversary that is aware of the fact that the system employs the χ^2 detector presented above. Since the detector is based on a stationary watermark signal Δu_t, we will show that an adversary that is able to extract the model of the system from the control law u_t and the sensor measurement y_t, is able to conduct an attack while remaining undetected.

Definition 3.2. *An attacker that, in addition to the capabilities of the cyber adversary, is also able to eavesdrop the messages containing the output of the controller u_t with the intention of improving its knowledge about the system model using a parametric or non-parametric identification model, is defined as a cyber-physical adversary.*

Based on the way to model the system's behaviour, two different cyber-physical adversaries can be defined.

Definition 3.3. *An attacker that only uses the previous input and output of the system to identify the system model is defined as a non-parametric cyber-physical adversary.*

Remark 3.5.1. *A non-parametric cyber-physical adversary can use, e.g., a Finite Impulse Response (FIR) model identification tool to identify the system model [100]. In Figure 3.1, the signals u'_t and y'_t are assumed to be respectively the output of the controller and the output of the measurement when an attack is taking place. We denote with $\Delta u'$ the watermark guessed by the non-parametric cyber-physical adversary.*

Definition 3.4. *An attacker able to estimate the parameters of the system using input and output data to mislead the controller detector is defined as a parametric cyber-physical adversary.*

Remark 3.5.2. *A parametric cyber-physical adversary is able to estimate the parameters of the system using input and output data to mislead the controller detector. This adversary can use, for instance, an ARX (autoregressive with exogenous input) model or an ARMAX (autoregressive-moving average with exogenous input) model to estimate the dynamics of the system [66].*

We assume that the main constraint of this adversary is the energy spent to eavesdrop and analyze the communication data, i.e., the number of samples eavesdropped to obtain the system model parameters.

Proposition 3.5.1. *A cyber-physical adversary that is able to exactly estimate the system controlled by the controller cannot be detected by the χ^2 detector (cf. Equation 3.6).*

Proof. Without loss of generality, we assume an attack is started at time T_0 and we compute the residues r_t for $t \in [T_0, T_0 + T - 1]$:

$$r_t = y'_t - C\hat{x}_{t|t-T} \tag{3.7}$$

Moreover, it is easy to show that the following holds:

$$\begin{aligned} \hat{x}_{t|t-T} &= \hat{x}'_{t|t-T} + \mathcal{A}^{t-T_0}(\hat{x}_{T_0|T_0-1} - \hat{x}'_{T_0|T_0-1}) \\ &\quad + \sum_{i=0}^{t-T_0-1} (\mathcal{A}^i B(\Delta u_{t-1-i} - \Delta u'_{t-1-i})) \end{aligned} \tag{3.8}$$

where $\hat{x}'$ is the estimated state when the system is under attack and $\mathcal{A} = (A+BL)(I-KC)$ is a stable matrix [9, 53]. Substitution of (3.8) in (3.7) yields:

$$\begin{aligned} r_t &= \underbrace{y'_t - C\hat{x}'_{t|t-T}}_{\text{First term}} \\ &- \underbrace{C\mathcal{A}^{t-T_0}(\hat{x}_{T_0|T_0-1} - \hat{x}'_{T_0|T_0-1})}_{\text{Second term}} \\ &- \underbrace{C\sum_{i=0}^{t-T_0-1}(\mathcal{A}^i B(\Delta u_{t-1-i} - \Delta u'_{t-1-i}))}_{\text{Third term}} \end{aligned}$$

Let us consider separately the three terms in the equation written above: the first term follows the same distribution of $(y_t - C\hat{x}_{t|t-1})$; since $\mathcal{A}$ is asymptotically stable – i.e. all its eigenvalues are inside the open unit disk of the complex plane – the second term converges exponentially fast to zero. In fact, the entries of $\mathcal{A}^{t-T_0}$ converge exponentially fast to zero. Now, if the third term would be equal to zero, the dynamics of r_t would recover the dynamics of the residues when no attack is undergoing and thus, the attack would not be detected. Under the hypothesis of this proposition, the adversary knows exactly the watermark signal and thus $\Delta u_t = \Delta u'_t$ which makes the third term equal to zero and concludes the proof. □

3.6 Acquiring the Watermark Signal Model

Motivated by Definition 3.3, we now show a practical method that can be used to acquire the watermark signal Δu_t. In particular, we propose an adversary that employs a Least Mean Square (LMS) filter, a non-parametric identification model, with the purpose of running an online identification of the system model. With the identified model, it is possible to extract the watermark and, finally, using it to authenticate messages with the aim of driving the system to an undesired state.

The LMS filter algorithm was created by Widrow and Hoof (1960) [101]. LMS algorithms are modeled without statistical assumptions, that means, without knowledge about the input and output signals of the physical systems. Their computational complexity is linearly scalar with the dimension of the FIR filter used by the algorithms. Another important characteristic, is their robustness, i.e., their capability to bring a satisfactory performance in the face of unknown disturbance [101]. The algorithm is composed of three components: 1) A FIR filter, which generate an *estimate* response of the desired response; 2) A comparator, which generate the *estimation error* between the estimate and the desired response; and 3) An adaptive weight-control mechanism, which adjusts the FIR filter vector weights based on the *estimation error*.

In Algorithm 1, we denote with p the LMS filter order and with μ its step size. The step size μ is upper bounded by $2/\lambda_{max}$, where λ_{max} is the maximum eigenvalue of the auto-correlation

matrix $R = E[XX^H]$, where X is the input signal, and X^H is the Hermitian transpose, or conjugate transpose, of X. Observe that if μ is chosen too small, the time to converge to optimal weights tends to be large [102]. The adversary initializes the weight matrix $\mathcal{W}$ to be equal to the zero matrix. Then, the adversary's algorithm shown in Algorithm 1, is run online. It is worth noting that in this algorithm $X[x(t-p+1),...,x(t)]$ is the input signal, $e(t)$ is the error, $\bar{e}(t)$ is its complex conjugate, and $d(t)$ is the desired output signal.

Algorithm 1 Non-parametric Cyber-Physical Adversary Algorithm.

```
1: procedure ADVERSARY ALGORITHM
2:     k ← length of eavesdropped data
3:     p ← filter order
4:     j ← p
5: top:
6:     if j < k then i ← 1.
7: loop:
8:     if i ⩽ p then
9:         ini ← j − p + 1.
10:        e(ini) ← d(ini) − W^T X[x(ini), ... x(j)].
11:        W ← W + μē(ini)X[x(ini), ... x(j)].
12:        j ← j + i.
13:        i ← i + 1.
14:        goto loop.
15:        close;
16:    goto top.
```

Once the system model has been identified, the adversary is able to extract the watermark and to carry out the attack. In particular, the adversary follows the steps described below:

1. *Eavesdropping of u_t and y_t and decomposition*: The adversary captures both the control law u_t and the sensors output y_t to make the decomposition between the information data and the watermark using the LMS filter as a noise cancellation adaptive filter. With this first step, we are able to separate u_t^* and the watermark Δu_t starting from u_t. Notice that, since the system is linear, it follows from the superposition principle that $y_t = y_t^* + y_t^{\Delta u}$, being y_t^* the output due to u_t^* and $y_t^{\Delta u}$ the output due to the watermark Δu_t.

2. *Acquiring the weight matrix, $\mathcal{W}$*: The adversary uses the LMS adaptive filter described before, as a system identification method.

3. *Computing the attack sensor measurement y_t'*: The adversary attacks the system by sending fake sensor measurements y_t', where $y_t^{\Delta u}$ is computed using the watermark Δu_t as follows:

$$y_t^{\Delta u} = \mathcal{W}^T \Delta u_t$$

and $y_t' = y_{t-1}^* + y_t^{\Delta u}$.

In the remainder of this section, we show via numerical simulations that the detection mechanism proposed in [9, 53] is not sufficiently robust and is not able to detect cyber-physical adversaries (cf. Section 3.5) that are able to identify the system model by eavesdropping the data channel.

In order to simulate a networked control system, we have employed a simplified version of the Tennesse Eastman control challenge problem [67], shown in Figure 3.2, and also used as a benchmark in [103]. The simplified version of this system simulates a MIMO system of order $n = 7$ with $p = 4$ inputs and $m = 4$ outputs. In particular the model of the discrete LTI system described by Equations (3.1)-(3.2) is defined by the following matrices:

$$A=\begin{bmatrix} 0.987 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.895 & -0.025 & 0 & 0 & 0 & 0 \\ 0 & 0.036 & 0.999 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -0.008 & 0 & 0 \\ 0 & 0 & 0 & 0.005 & 0.960 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.999 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.990 \end{bmatrix}, B=\begin{bmatrix} 0.149 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.071 \\ 0 & 0 & 0 & 0.001 \\ 0.380 & 0 & -0.096 & 0 \\ 1.000 & 0 & -0.096 & 0 \\ 0 & 0.038 & 0 & 0 \\ 0 & 0 & 0 & 0.075 \end{bmatrix},$$

$$C=\begin{bmatrix} 0.151 & -0.076 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.040 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.133 \end{bmatrix}.$$

The co-variance matrices are equal to $Q = 0.01I$ and $R = I$, whereas the cost matrices are $\Gamma = 1.5I$ and $\Omega = 10I$.

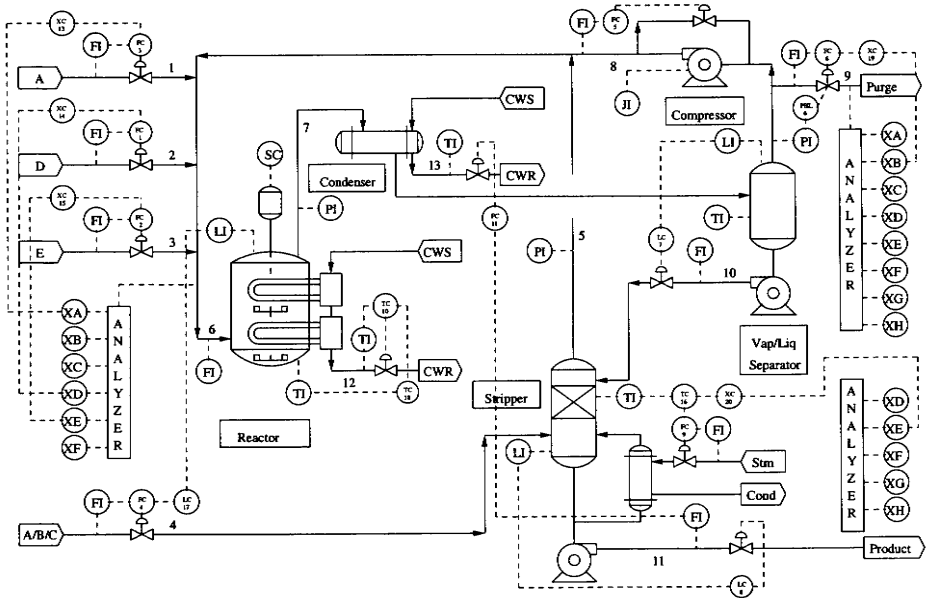
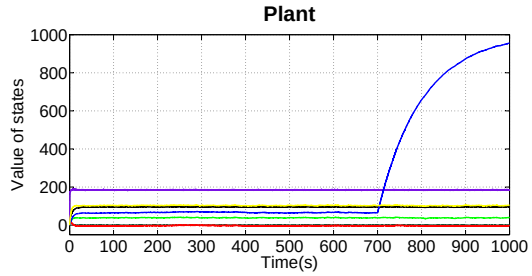


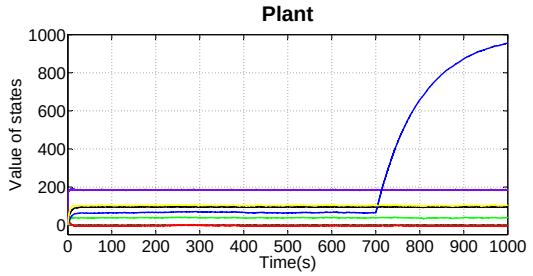
Figure 3.2 – Tennessee Eastman model [104].

To validate our approach, we compare the system dynamics considering the two adversaries described above. Figures 3.3(a) and 3.3(c) show the plant dynamics and the state estimated by the

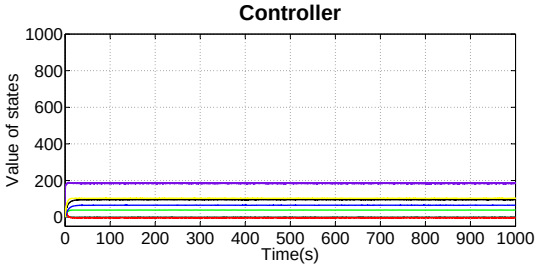
controller in the case of a cyber adversary. Figure 3.3(a) shows that the adversary is able to drive the state components to an undesired value. Nevertheless, the controller, misled by the adversary, does not perceive such situation (cf. Figure 3.3(c)). Figure 3.3(b) and Figure 3.3(d) show the dynamics of the plant and the ones of the controller under a non-parametric cyber-physical adversary model which exhibits the same behavior described above for the case of the cyber adversary.



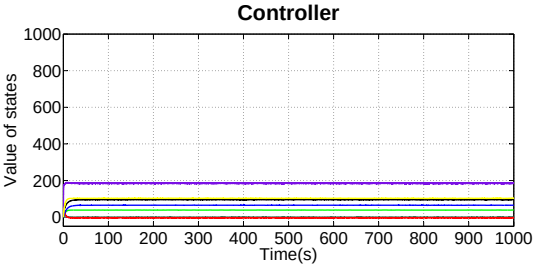
(a) Plant states, cyber adversary attack.



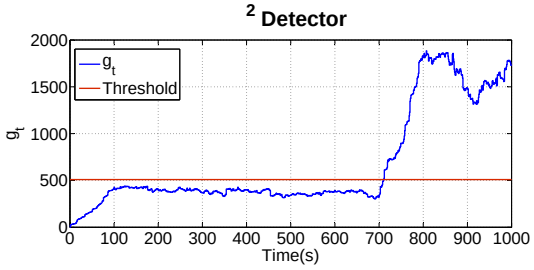
(b) Plant states, non-parametric cyber-physical adversary attack.



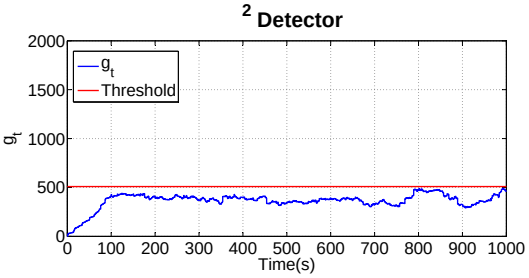
(c) Estimated states in the controller, cyber adversary.



(d) Estimated states in the controller, non-parametric cyber-physical adversary.



(e) Detector results, cyber adversary.



(f) Detector results, non-parametric cyber-physical adversary.

Figure 3.3 – **Numeric simulation results.** Attacks start at $t = 700s$. (a)(c) The dynamics of the state vector in the plant and in the controller under the cyber adversary attack. (b),(d) The dynamics of the state vector in the plant and in the controller under the non-parametric cyber-physical adversary attack. (e),(f) χ^2 detector results under the two aforementioned attack scenarios.

Let us now contrast the performance of the detector described in Section 3.4 when detecting either the cyber or the non-parametric cyber-physical adversary. Towards this end, Figure 3.3(e) and Figure 3.3(f) show the value of the alarm signal g_t produced by the same χ^2 detector in the case of the cyber adversary (cf. Figure 3.3(e)) and the non-parametric cyber-physical adversary (cf. Figure 3.3(f)). Figure 3.3(e) shows that the detector is able to detect the cyber adversary thanks to the added watermark signal as soon as the attack starts at $t = 700$s. However, Figure 3.3(f) shows that the same detector is not able to detect the non-parametric cyber-physical adversary since g_t does not exceed the threshold γ during the attack. In order to quantify the detector performance, we define the *DR* (*Detection Ratio*) metric as follows:

$$DR = \frac{\sum_{t=T_0}^{T_0+T_a} \mathbb{1}_{g_t \geq \gamma}}{T_a} \tag{3.9}$$

where T_a is the attack duration, and $\mathbb{1}$ is the indicator function whose output is equal to 1 if the Boolean condition given as its argument ($g_t \geq \gamma$) is true; or it is equal to 0 otherwise. In a nutshell, $DR \in [0,1]$ can be considered as an efficiency index for the detector: *DR* is equal to one when the attack is always detected; and it is equal to zero when the attack is always undetected.

Figure 3.4 shows the CDF (Cumulative Distribution Function) of the detection ratio obtained by measuring *DR* for 200 simulations both in the case of the cyber-physical and the cyber adversary. The figure shows that the detection scheme proposed in [9, 53] is able to provide a median detection ratio that is larger than 0.9 when a cyber adversary attacks the system. However, using a cyber-physical adversary that acquires the watermark, the median detection ratio drops to around 0.2. This quantitatively shows that the detection strategy proposed in [9, 53] is not sufficiently robust for security.

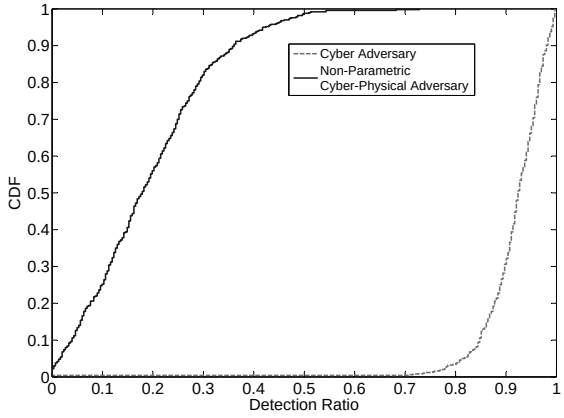


Figure 3.4 – Cumulative distribution function (CDF) of the detection ratio associated to the χ^2 detector (cf. Section 3.4), obtained by measuring the *DR* metric (cf. Equation (3.9)) for 200 simulations (both cyber and non-parametric cyber-physical adversary cases).

3.7 Detecting the Non-parametric Cyber-Physical Adversary

In the previous sections, we have defined three different kinds of adversaries who use different vulnerabilities of a control system to carry out attacks; the cyber-adversaries, the non-parametric cyber-physical adversaries, and the parametric cyber-physical adversaries. In this section, we propose a detection scheme that extends the one presented in [9, 53], in order to detect non-parametric cyber-physical adversaries. We also study the performance loss of the new detection scheme with regard to the one presented in [9, 53].

3.7.1 Multi-Watermark based Attack Detection

The goal of our new detection scheme is to increase the difficulty in retrieving the authentication watermark Δu_t from the control signal u_t, so that the probability of detecting an attack from a non-parametric cyber-physical adversary can be increased. We assume that the control system under attack employs exactly the same type of controllers and the same detection strategy presented in Section 3.4. The only difference in the proposed detection scheme is the way that the watermark signal Δu_t is generated. The control input u_t, as in the case of the detection scheme presented in Section 3.4, is computed as the superposition of the optimal control signal u_t^* produced by the LQR controller and a given multi-watermark signal Δu_t. The idea is to construct the authentication watermark signal by switching between N different and independent processes with different co-variance and average (offsets). More precisely, the non-stationary watermark Δu_t is obtained by periodically switching, with a period T, between N signals $\Delta u^{(i)}$, with $i \in I = \{0, 1, \ldots, N-1\}$, extracted by different stochastic processes. Hence, the watermark signal Δu_t can be formalized as follows:

$$\Delta u_t = \Delta u_t^{(s(t,T))} \tag{3.10}$$

where $s : \mathbb{N} \times \mathbb{R} \rightarrow I$ is a static function that maps the time sample t and the switching period T to an element of the index set I, defined as follows:

$$s(t,T) = \left\lfloor \frac{1}{T} \mod (t, NT) \right\rfloor \tag{3.11}$$

where $\mod(x,y)$ is the modulo operator and $\lfloor \cdot \rfloor$ is the floor function.

By using the proposed watermark (cf. Equation (3.10)), we now have a proper adaptive protection mechanism with two main configurable parameters, i.e., the number of distributions N and the switching frequency $f = 1/T$. Notice that the original watermark signal described in Section 3.3 is recovered when $f \rightarrow 0$ and when $\Delta u_t^{(0)}$ being a stationary zero mean Gaussian process.

3.7.2 Single-watermark LQG Structure Performance Loss

In this section, we compute the increment of cost in the LQG structure due to the single-watermark added to the control input. This supplementary cost is the degradation in the performance of the system, as shown in [9], and can be defined as follows:

$$J = J^* + \Delta J_s \tag{3.12}$$

where J^* is the optimal cost of the system described in Section 3.3; and ΔJ_s is the increment of cost due to the use of the single-watermark-based detector. In the following, we develop the cost of the system.

$$\begin{aligned} J &= \lim_{n\to\infty} \frac{1}{n} \sum_{j=0}^{n-1} E\left[(x_j^T \Gamma x_j + u_j^T \Omega u_j)\right] \\ &= \lim_{n\to\infty} \frac{1}{n} \sum_{j=0}^{n-1} \left[tr(\Gamma Cov(x_j)) + tr(\Omega Cov(u_j))\right] \end{aligned} \tag{3.13}$$

where x_t is defined as:

$$x_t = \mathcal{L}_x(w_t, v_t) + \sum_{j=t-1}^{\infty} (A+BL)^j BVar(\Delta u_{-j}) \tag{3.14}$$

and u_t as follows:

$$u_t = \mathcal{L}_u(w_t, v_t) + \sum_{j=t-1}^{\infty} (A+BL)^j BVar(\Delta u_{-j}) + Var(\Delta u_t) \tag{3.15}$$

where $\mathcal{L}_x$ and $\mathcal{L}_u$ are linear functions. Their definition is not relevant for the target of this dissertation, but the reader can find the details in [9]. Assuming that the noise of the system and the watermark are independent, we can define the increment of cost due to the single-watermark as [9]:

$$\begin{aligned} \Delta J_s &= \Gamma tr\left[Cov\left(\sum_{j=t-1}^{\infty} (A+BL)^j BVar(\Delta u_{-j})\right)\right] \\ &\quad -L\Omega tr\left[Cov\left(\sum_{j=t-1}^{\infty} (A+BL)^j BVar(\Delta u_{-j}) + Var(\Delta u_t)\right)\right] \end{aligned} \tag{3.16}$$

3.7.3 Multi-watermark LQG Structure Performance Loss

Let us now evaluate the increment of the cost generated by the multi-watermark based detector, ΔJ_m, and next compare the cost generated by the single-watermark and the multi-watermark. The equation of ΔJ_m, is given by:

$$\begin{aligned}\Delta J_m(t) &= tr[\Gamma Cov(\sum_{j=t-1}^{\infty}(A+BL)^j B(Var(\Delta u_{-j}^{(i)})+E[\Delta u_{-j}^{(i)}]))] \\ &\quad +tr[L\Omega Cov(\sum_{j=t-1}^{\infty}(A+BL)^j B(Var(\Delta u_{-j}^{(i)})+E[\Delta u_{-j}^{(i)}]) \\ &\quad +(Var(\Delta u_t)+E[\Delta u_t])] \end{aligned} \tag{3.17}$$

where $Var(\Delta u_t^{(i)})$ and $E[\Delta u_t^{(i)}]$ are respectively the variance and the mean of the watermark sent at time t. The performance loss of the LQG structure depends linearly on the variance and the mean of the multi-watermark $\Delta u_t^{(i)}$ for each T samples.

The following theorem shows the difference between the performance loss due to the single-watermark, Δu, and the performance loss due to the multi-watermark, $\Delta u^{(i)}$.

Theorem 3.7.1. *Let us assume that a watermark is a Gaussian signal with a couple of parameters to be characterized, the mean and the variance. The multi-watermark distribution is defined as* $M_w = N(E[\Delta u^{(i)}], Var(\Delta u^{(i)}))$*, and the single-watermark distribution is defined as* $S_w = N(E[\Delta u], Var(\Delta u))$*. If we define for the multi-watermark* β *as:*

$$\beta = E[\Delta u^{(i)}] + Var(\Delta u^{(i)}) \qquad \forall i \in I \tag{3.18}$$

and for the single-watermark ε *as:*

$$\varepsilon = E[\Delta u] + Var(\Delta u) \tag{3.19}$$

where ε *and* β *are constant for single and multi-watermark respectively. Then, we can conclude that the performance loss of both approaches is equal if* $\varepsilon = \beta$*.*

Proof. If we assume that $E[\Delta u] = 0$ for the single-watermark, we can prove the theorem as follows:

$$\begin{aligned} Diff(\Delta J_m(t), \Delta J_s(t)) &= \Delta J_m(t) - \Delta J_s(t) \\ &= \Gamma tr\left[Cov\left(\sum_{j=t-1}^{\infty}(A+BL)^j B(\beta_{-j}-\varepsilon_{-j})\right)\right] \\ &\quad +L\Omega tr\left[Cov\left(\sum_{j=t-1}^{\infty}(A+BL)^j B(\beta_{-j}-\varepsilon_{-j})+(\beta_t-\varepsilon_t)\right)\right] = 0 \end{aligned}$$

□

Remark 3.7.1. *Note that using Theorem 3.7.1, the performance loss due to the multi and the single-watermark is equal. The assumption of the equal performance loss allows compare both approaches under the same conditions. This can be formally stated as follows:*

$$E[\Delta u^{(i)}] + Var(\Delta u^{(i)}) = \varepsilon = \beta. \tag{3.20}$$

3.8 Numerical Validation of the Multi-Watermark Detector against non-Parametric Cyber-Physical Adversaries

This section validates through numerical simulations the detection scheme proposed in Section 3.7.1. In particular, we aim at showing that the proposed watermark signal is able to detect non-parametric cyber-physical adversaries (cf. Section 3.5) with a higher detection ratio with respect to the one obtained with the watermark proposed in [9, 53]. Towards this end, we employ a MIMO system with four inputs and four outputs described by the following matrices:

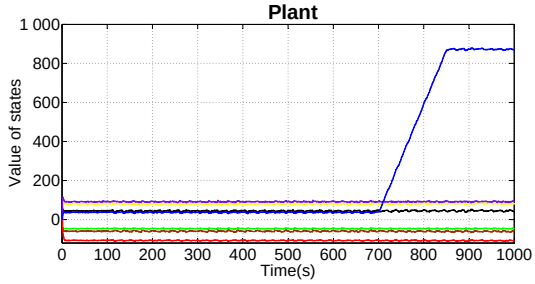
$$\text{A}=\begin{bmatrix} 0.3991 & 0.07113 & 0.1573 & -0.1274 & 0.0226 & -0.0225 & 0.001 \\ 0.003 & -0.07588 & -0.005092 & -0.03893 & 0.09917 & -0.0168 & 0 \\ -0.1974 & -0.01849 & 0.0453 & 0.1579 & -0.1597 & 0.1405 & -0.002 \\ -0.1246 & -0.0726 & 0.1515 & -0.1148 & 0.5156 & -0.0665 & 0 \\ 0.4309 & -0.1204 & 0.09715 & 0.055 & 0.2406 & 0.2812 & 0.0001 \\ -0.0827 & -0.01092 & 0.1234 & -0.1318 & 0.0348 & 0.469 & 0 \\ 0.08312 & -0.0829 & 0.081 & 0.0358 & 0.1124 & 0.02475 & 0.4469 \end{bmatrix},$$

$$\text{B}=\begin{bmatrix} 0.947 & 0 & 0.002 & -0.021 \\ -0.0086 & 0 & -0.406 & 0.02829 \\ -0.8708 & 0.0011 & 0.0011 & -0.106 \\ 0.4872 & 0.002 & 0.188 & -0.041 \\ 0.1233 & 0 & 0.01 & -0.9344 \\ 0 & 0 & 0 & 0.521 \\ 0 & 0.7658 & 0 & 0 \end{bmatrix}, \text{C}=\begin{bmatrix} -1.102 & 0.302 & -0.1004 & 0.0386 & 0.053 & 0.0891 & 0 \\ 0 & 0.114 & -0.0132 & -1.087 & 0.116 & 0.051 & 0.905 \\ 0.0003 & 1.593 & -0.002 & 0 & 0.093 & 0 & 0.0428 \\ -0.163 & -0.0712 & -0.1074 & 0 & 0 & -0.7443 & 0.089 \end{bmatrix}.$$

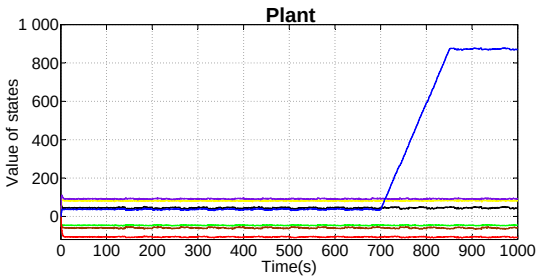
and co-variance matrices equal to $Q = 0.2I$ and $R = I$. The positive definite cost matrices Γ and Ω are both equal to the identity matrix. The simulation is based on Matlab and Simulink models of the plant, as well as the models of the non-parametric cyber-physical adversaries. The attacks start at $t = 700$s. We use three different distributions (i.e., $N = 3$) switched at random: a Gaussian, a Rician and a Rayleigh distribution. Table 3.1 shows the co-variance and offset configured in the simulations for each distribution.

To validate the proposed attack detection scheme, we compare the system dynamics considering two different switching frequencies. We have simulated a high frequency switching watermark configured to switch each seven time samples, and a low frequency switching configured to switch each 20 time samples. Figures 3.5(a) and (c) show the plant dynamics and the dynamics of the states estimated by the controller in the case of a switching frequency watermark configured to seven time samples and a cyber-physical adversary attack. Figure 3.5(a) shows that the adversary is able to drive the state to an undesired value. Nevertheless, the controller misled by the adversary, does not perceive such situation (cf. Figure 3.5(c)). Figures 3.5(b) and (d) show the

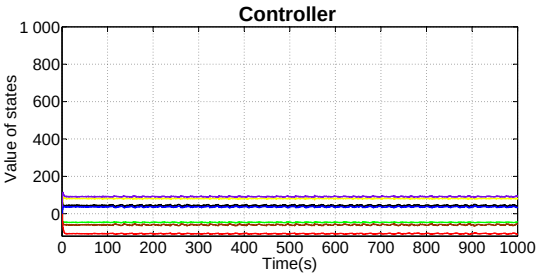
plant dynamics and the dynamics of the states estimated by the controller when the watermark is switched each 20 time samples. The dynamics show exactly the same behavior described above.



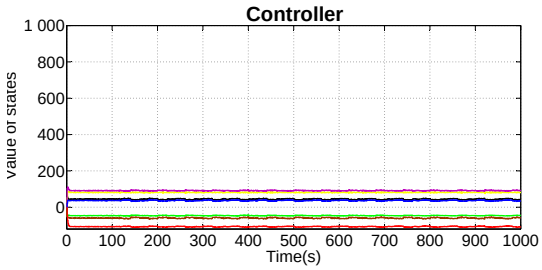
(a) Plant states under a non-parametric cyber-physical adversary attack and *switching frequency* set to 0.14Hz, i.e. every 7 time steps, the controller changes the distribution associated to the watermark.



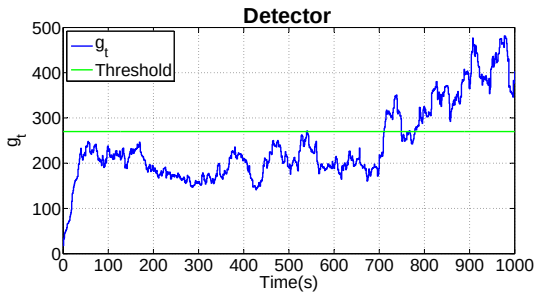
(b) Plant states under a non-parametric cyber-physical adversary attack and *switching frequency* set to 0.05Hz, i.e. every 20 time steps, the controller changes the distribution associated to the watermark.



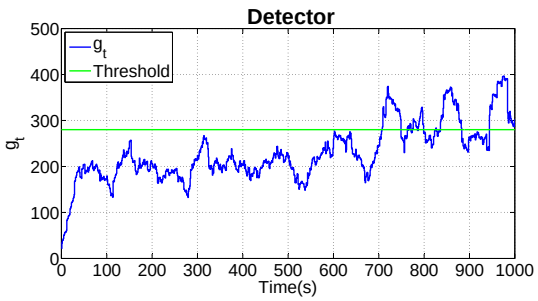
(c) Estimated states in the controller under a non-parametric cyber-physical adversary attack and *switching frequency* set to 0.14Hz.



(d) Estimated states in the controller under a non-parametric cyber-physical adversary attack and *switching frequency* set to 0.05Hz.



(e) Detector results, *switching frequency* set to 0.14Hz.



(f) Detector results, *switching frequency* set to 0.05Hz.

Figure 3.5 – **Numeric simulation results.** Attacks start at $t = 700s$. (a),(b) The dynamics of the states vector in the plant under a non-parametric cyber-physical adversary attack and switching frequency configured with two different configurations (0.14Hz and 0.05Hz). (c),(d) The dynamics of the states vector estimated in the controller, under the same scenarios. (e),(f) The dynamics of the alarm signal g_t produced by the multi-watermark based detector, under the same scenarios.

Figures 3.5(e) and 3.5(f) show the dynamics of the alarm signal g_t produced by the detector, respectively in the case of high and low switching frequency. Notice that switching the watermark distributions at a high frequency provides better detection performances compared to the case of a low switching frequency.

Distribution	**Variance (σ^2)**	**Offset**
Gaussian	5.9536	0.0
Rician	3.8870	3.7106
Rayleigh	3.0581	2.5553

Table 3.1 – Sample parameters used in the multi-watermark Matlab/Simulink implementation.

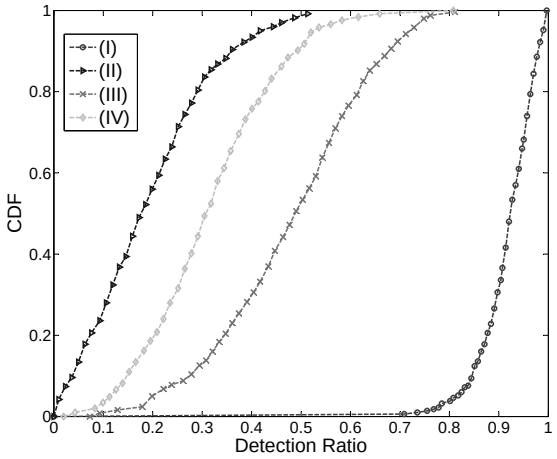


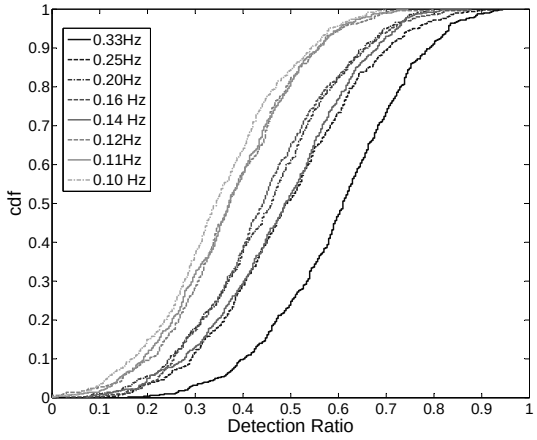
Figure 3.6 – **Numeric simulation results using the single and multi-watermark detection scheme.** Confronting the performance of the two detectors. (I) χ^2 detector in [9, 53], and cyber adversary. (II) χ^2 detector and non-parametric cyber-physical adversary. (III) Multi-watermark detector with switching frequency set to 0.14Hz, and non-parametric cyber-physical adversary. (IV) Multi-watermark detector with switching frequency set to 0.05Hz, and non-parametric cyber-physical adversary.

To quantify the effectiveness of the proposed detection scheme, we compute the detection ratio *DR* as a function of the switching frequency. In particular, for each considered frequency f we run 200 Monte Carlo simulations (with randomly generated system parameters) both in the case of the cyber-physical and the cyber adversary, and we compute the CDF (Cumulative Distribution Function) of the detection ratio.

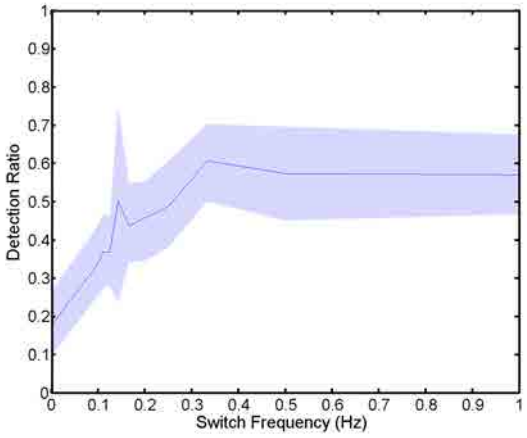
We start by confronting the performance obtained with the detection strategy based on multiple watermark signals proposed in this chapter with that proposed in [9, 53] in both the case of a cyber-physical and a cyber adversary. In the case of the proposed multi-watermark strategy we consider two switching frequencies $f_L = 0.05$Hz (switching watermark each 20 time steps) and $f_H = 0.14$Hz (switching watermark each seven time steps). The results of this comparison

are shown in Figure 3.6. Let us focus on the detection strategy proposed in [9, 53]: as shown before, the detector is able to consistently detect a cyber attack but it performs poorly when a cyber-physical adversary attacks the system. Nevertheless, the proposed detection strategy based on multiple watermarks is able to provide a higher detection ratio. In particular, we notice that the detector employing a higher switching frequency f_H provides better performances with respect to the case of using the lower switching frequency f_L.

In the following we are interested in analyzing in more details the performance of the proposed detection strategy when the switching frequency f is varied, to give a more in depth explanation of the anecdotal evidence shown above. Towards this end, Figure 3.7(a) shows the CDF of the detection ratio obtained when the switching frequency varies in the range $[0.10, 0.33]$Hz. In this case, we only consider cyber-physical adversaries. The CDFs shown in Figure 3.7(a) confirm that when the switching frequency increases, the detection ratio is also increased.



(a) CDF and detection ratio per switching frequencies.



(b) Detection ratio function with respect to the switching frequency.

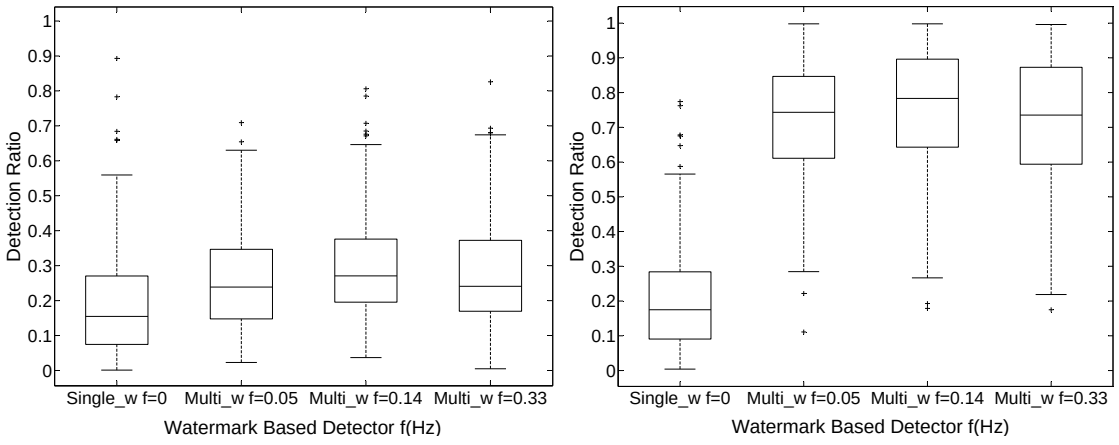
Figure 3.7 – **Numeric simulation results using the multi-watermark detection scheme.** (a) Cumulative distribution function (CDF) of the detector under different switching frequencies. (b) Median detection ratio function per switching frequency, in which the shaded area corresponds to the 25-th and the 75-th percentiles (i.e., confidence intervals).

To finish this section, we provide in Figure 3.7(b) the median detection ratio function of f. The figure also contains the case $f = 0$ that corresponds to the detection strategy proposed in [9, 53] – used as a baseline. The shaded area in the figure corresponds to the values of the detection ratio between the 25-th and the 75-th percentile for each f. As expected, the figure shows that by increasing the frequency we are able to obtain a detection ratio that goes from around 0.2 in the case of the baseline approach of [9, 53] to around 0.6 in correspondence of $f = 0.33$Hz.

Observe that the probability of false alarms without attack (often referred in the related literature as false positives) is fixed, $\xi = 1\%$. Notice as well that false negatives (i.e., undetected real attacks), are inversely proportional to the detection ratio for each switching frequency.

Efficiently Validation

We have validated above the multi-watermark detector using a static function, I, to define the multi-watermark and different performance loss between single and multi-watermark. Hereinafter we present the results and validations obtained for a system with the same performance loss between single and multi-watermark detector and where the multi-watermark is generated from a non-static function, I_d. Figure 3.8 shows the result obtained after running 200 Monte Carlo simulations of a system with single and multi-watermark detector against a non-parametric cyber-physical adversary. In this simulation, single and multi-watermark detector have 30% performance loss, ΔJ, respect to the optimal cost. Moreover the watermark uses a dynamic function to define the multi-watermark. In Figure 3.8(a) we show the result of single and multi-watermark for a system of order 4. We can confirm that the multi-watermark detector, with the same performance loss as the single-watermark detector, has a higher detection ratio. Figure 3.8(b) shows the result of single and multi-watermark for a higher system of order, 25. On one hand, these results confirm that multi-watermark detector is able to detect properly non-parametric cyber-physical adversary. Additionally, we can conclude also that the detection ratio increase with the complexity of the system. On the other hand, Figure 3.9 depicts that using multi-watermark, with same performance loss as single-watermark, the ratio of detection increases when the switching frequency varies in the range $[0, 0.14]$Hz, where $f = 0$ is the single-watermark detector. We confirm that the multi-watermark performance increases until $f = 0.14$Hz which obtains a peak, before the steady ratio of detection. In the following section, we extend the analysis to the case of parametric cyber-physical adversaries.



(a) Detection ratio for a system of order 4, using single- and multi-watermark.

(b) Detection ratio for a system of order 25, using single- and multi-watermark.

Figure 3.8 – **Simulation results using the single and multi-watermark detection scheme with the same performance loss.** (a) Detection ratio function with respect to the single- and multi-watermark with different switch frequencies for four order systems. (b) Detection ratio function with respect to the single- and multi-watermark with different switch frequencies for twenty five order systems.

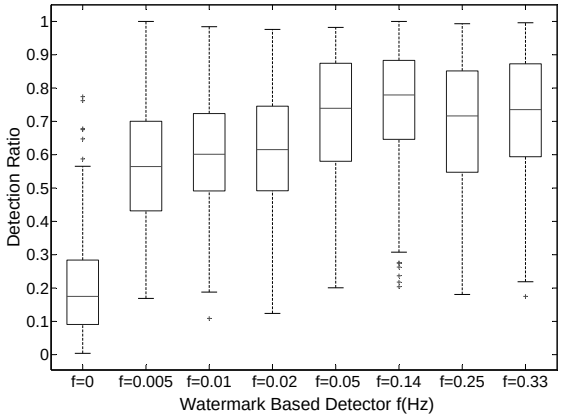


Figure 3.9 – Single ($f = 0$) and multi ($f = [0.005, 0.01, 0.02, 0.05, 0.14, 0.25, 0.33]$) watermark detector against a non-parametric cyber-physical adversary (the order of the system is 25).

3.9 Numerical Validation of the Multi-Watermark Detector against Parametric Cyber-Physical Adversaries

In the previous sections, we have seen how the multi-watermark detector is able to detect both cyber and non-parametric cyber-physical adversaries. In this section, we extend the study to the case of parametric cyber-physical adversaries (cf. Definition 3.4). We recall that parametric cyber-physical adversaries are able to identify the system model parameters from the input and the output plant signals. In fact, a parametric cyber-physical adversary can obtain the system model with great accuracy, if control commands and sensor measurements are accessible.

Let us first show how a parametric cyber-physical adversary acquires the watermark signal presented in [9, 53]. Remember that such a watermark is modeled as a Gaussian signal with zero-mean. Its variance is represented by $\mathcal{U}$, *i.e.* $\Delta u_t \sim N(0, \mathcal{U})$. The variance modifies the control inputs and propagates the modification to the system outputs. However, it does not modify the system dynamics. Control inputs are represented by:

$$u_t = u_t^* + \Delta u_t \tag{3.21}$$

and the system outputs are represented by:

$$y_t = C(Ax_t + B(u_t^* + \Delta u_t) + w_t) + v_t \tag{3.22}$$

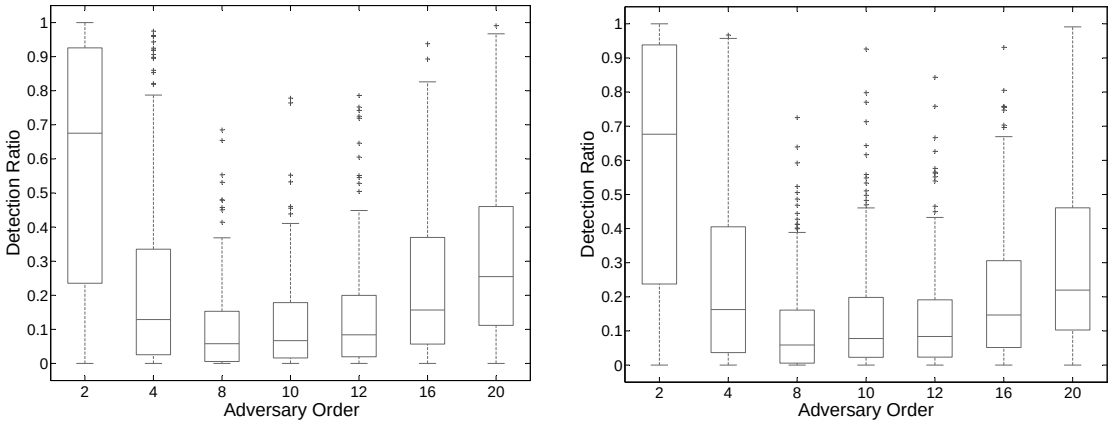
Using the aforementioned characteristic of the watermark, a parametric cyber-physical adversary can use an ARX (autoregressive with exogenous input) model to define the system defined in Equations (3.1) and (3.2), as follows [66]:

$$Y(z) = H(z)U(z) + V(z) \tag{3.23}$$

where $U(z)$ and $Y(z)$ represent the inputs and the outputs of the plant, respectively; $V(z)$ represents the external noise which affects the outputs of the plant; and $H(z)$ is another way to describe the model of the system presented in Section 3.3, using frequency domain, such that:

$$H(z) = \frac{\mathcal{N}(z)}{\mathcal{D}(z)} = \left(\frac{n_0 z^m + n_1 z^{m-1} + ... + n_m}{d_0 z^n + d_1 z^{n-1} + ... + d_n} \right) \tag{3.24}$$

where $\mathcal{N}(z)$ and $\mathcal{D}(z)$ are the polynomial functions which build the model of the system.



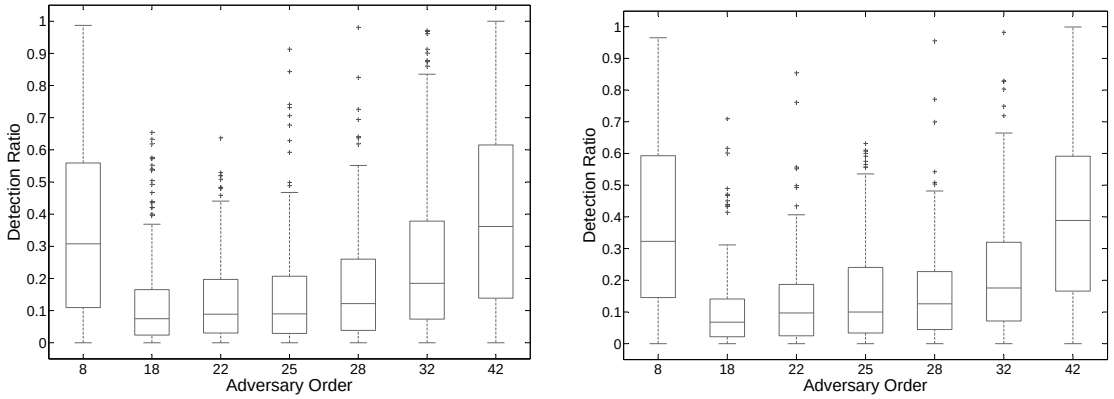
(a) Detection ratio function with respect to the single-watermark against a parametric cyber-physical adversary using different adversary system order.

(b) Detection ratio function with respect to the multi-watermark against a parametric cyber-physical adversary using different adversary system order.

Figure 3.10 – **Numeric simulation results using the single and multi-watermark detection scheme with the same performance loss, different adversary system order, and a window size equal to** 200**.** (a) Detection ratio regarding single-watermark for systems of order ten. (b) Detection ratio regarding multi-watermark for systems of order ten.

Following the same simulation setup introduced in Section 3.8, Figures 3.10 and 3.11 show the detection ratio of the watermark detector against a parametric cyber-physical adversary. Figure 3.10 shows the results of 200 Monte Carlo simulations using systems of order ten, against this type of adversaries. The results present the ratio of detection if these adversaries use a window size equal to 200 and different system orders for the model. If the attackers choose the correct system order for the model, the ratio of detection is around 7%. Nevertheless, if the order chosen by the adversaries varies in the range $[8, 12]$, the detection ratio is not higher than 10%. Out of

this range, the ratio of detection increases drastically. Figure 3.11 shows the ratio of detection for 200 Monte Carlo simulations using systems of order 25, against seven different parametric cyber-physical adversaries. The assumed window size is settled to $\hat{T} = 300$. If the adversaries use a model of the system with the correct order, the ratio of detection is around 8%. The range of orders where the ratio of detection does not increase drastically is $[18, 28]$. If the adversaries use an order in this range, the ratio of detection is not higher than 10%. Otherwise, the likelihood to detect the adversary is high.



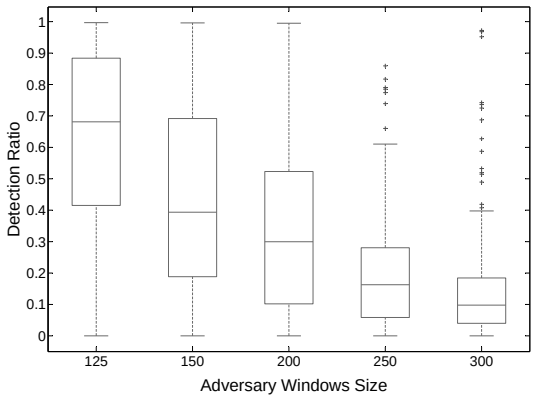
(a) Detection ratio function with respect to the single-watermark for systems of order 25, against a parametric cyber-physical adversary using different adversary system order.

(b) Detection ratio function with respect to the multi-watermark for systems of order 25, against a parametric cyber-physical adversary using different adversary system order.

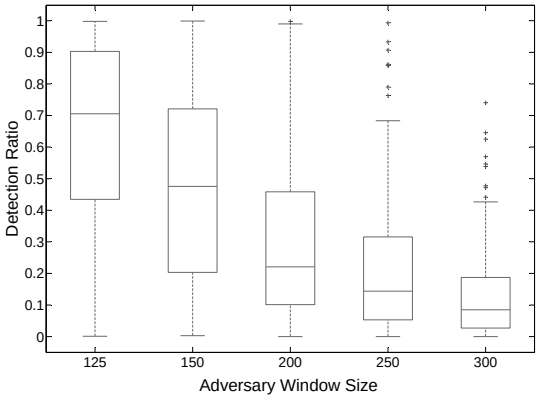
Figure 3.11 – **Numeric simulation results using the single and multi-watermark detection scheme with the same performance loss, different adversary system order, and a window size equal to** 300**.** (a) Detection ratio regarding single-watermark for systems of order 25. (b) Detection ratio regarding multi-watermark for systems of order 25.

Figure 3.12 shows the detection ratio of the same system, against a parametric cyber-physical adversary with different window sizes (125, 150, 200, 250, and 300), and the correct system order. It is worth noting that this type of adversaries need a bigger window size in order to attack a system using a higher order, with a detection ratio less than 10%. Following the previous results, we can conclude that a parametric cyber-physical adversary, who is capable to eavesdrop and analyze a large number of samples from the communication channel, and using an equivalent order system, is capable of evading detection.

Remark 3.9.1. *A parametric cyber-physical adversary is able to obtain the system model, $H(z)$, and mislead the controller, by eavesdropping the control inputs and the sensor measurements. The probability of being detected is equivalent to the probability of obtaining an erroneous model. This probability is directly proportional to the order of the system; and inversely proportional to the window size to eavesdrop the data channel.*



(a) Detection ratio function with respect to the single-watermark against a parametric cyber-physical adversary using different adversary window size for eavesdropping the data of the channel before the attack.



(b) Detection ratio function with respect to the multi-watermark against a parametric cyber-physical adversary using different adversary window size for eavesdropping the data of the channel before the attack.

Figure 3.12 – **Numeric simulation results using the single and multi-watermark detection scheme with the same performance loss, and different adversary window size for eavesdropping the data channel before the attack.** The order used by the adversaries is the correct system order, $p = 25$. (a) Detection ratio for the single-watermark approach. (b) Detection ratio for the multi-watermark approach.

Following the Remark 3.9.1, and under the hypothesis of considering the real system like a black box, erroneous system identification depends on the order selected by the adversaries to recreate the system model, as well as the number of eavesdropped samples and the window size used by the adversaries to recompute the parameters of the target system. This situation can be quantified using Mean Square Error (MSE) [63, 68]. In a nutshell, the likelihood to obtain the correct model of the target system is directly proportional to the order chosen by the adversaries to generate the model, and inversely proportional to the number of samples eavesdropped (cf. Figures 3.10, 3.11, and 3.12). The computational cost for the adversaries is directly proportional to the system order, since this type of adversaries need to increase the order of the model, as well as the window size in order to minimize the MSE. Therefore, the number of samples eavesdropped before conducting the attack, together with the order system chosen by the adversaries, are the two main parameters to evade detection.

3.10 Discussion

In Section 3.4, we have reviewed the watermark-based detector proposed in [9, 53]. We have shown that the detector fails at properly handling attacks carried out by *cyber-physical* adversaries. In particular, we have shown that an adversary that learns about the system model is able to separate the watermark from the control signal and succeeds at attacking the system without being detected. Then, we have presented an enhanced detection scheme. The main idea of the new scheme, is to use multiple watermark distributions and non-stationary identification signals. The resulting approach is able to detect both cyber- and cyber-physical adversaries.

To summarize, the detector in [9, 53] fails at detecting cyber-physical adversaries. The multi-watermark proposal succeeds at properly detecting such adversaries under the assumption that the watermark distributions change quite frequently. The rationale is that, the non-parametric adversary has little chances of acquiring the necessary information to acquire the watermark and bypass the detector. Moreover, the detector performance loss in the multi-watermark approach is equivalent to the performance loss in the case of the single-watermark approach. We have also shown that a smarter parametric cyber-physical adversary is able to attack the system without being detected in case of detecting the correct parameters. We have detailed the strategy of this type of adversaries in Section 3.6. It is worth noting that the detection ratio increases with respect to the lack of accuracy of the adversary.

3.11 Summary

In this chapter, we have focused on the adaptation of failure detection mechanisms. The goal is to handle, in addition to faults and errors, the detection of cyber-physical attacks. Cyber-physical attacks refer to malicious activities conducted over industrial control systems with upgraded computing, communication and interconnection capabilities. In other words, they refer to threats against industrial environments that close their loops through networked control systems.

We have revisited a watermark-based attack detection scheme. The approach relies on the adaptation of a failure detector, by adding a complementary authentication watermark signal for the detection of the malicious activities. The approach only requires to inject the watermark from the system controller. The monitored system continues to work regardless of the added watermark signal. This way, the strategy is free from desynchronization. Nevertheless, we have shown that the detection strategy is not sufficiently robust from a security standpoint. Indeed, we have quantitatively shown that the approach only detects *cyber adversaries*, i.e., attackers with the ability to eavesdrop information from the system, but that do not attempt to get any knowledge about the system model itself. We have validated that the detector fails at covering *cyber-physical adversaries*, i.e., attackers that, in addition to the capabilities of the cyber adversary, are also able to infer the system model to evade the detection.

We have then presented a multi-watermark based adaptive detection scheme with two main configurable parameters: number of distributions and switching frequency. The novel multi-watermark proposal succeeds at properly detecting both cyber and cyber-physical adversaries under the assumption that the watermark distributions change frequently. The rationale is that, even under the presence of adversaries with knowledge about the system dynamics, the detector succeeds at reducing their chances of acquiring the authentication watermark and bypass the detector. In the next chapter we present a new strategy in order to reach a great reduction of bypass the detector for a parametric cyber-adversary. Furthermore, in Chapter 5, we present the results obtained from a training cyber-physical testbed used to validate the detection performance of the constructions provided in this chapter.

4 Adaptive Control-Theoretic Detection

4.1 Introduction

As we have shown in Chapter 3, the use of inadequate cyber-physical security mechanisms can have an adverse effect in cyber-physical industrial systems [2, 105, 106]. These new defined systems need the collaboration amongs a very wide number of disciplines to solve the challenges in term of autonomy, reliability, usability, functionality, and cyber security [107]. Hereinafter, we focus on the use of control-theoretic solutions to detect attacks against cyber-physical systems. Traditional literature proposes the use of control strategies to retain, f.i., satisfactory closed-loop performance, as well as safety properties, when a communication network connects the distributed components of a physical system (e.g., sensors, actuators, and controllers). However, the adaptation of these strategies to handle security incidents, is an ongoing challenge.

Given the control-theoretic nature of cyber-physical industrial systems, the control community is actively working to adapt traditional control strategies to detect faults and errors, towards detectors of malicious attacks [75, 77, 78]. Motivated by the same objectives, we present a solution that complements the watermark-based detector in order to cover these weaknesses. More specifically, the new solution combines event-triggered control strategies together with the challenge-response watermark-based detector analyzed and improved in the previous chapter. This combination allows to handle integrity attacks against cyber-physical systems.

4.2 Contributions

The related literature has reported several research works about control strategies that improve the stability and performance of cyber-physical systems. The use of such strategies to cover, as well, security issues has only been imposed in the last decade. In this chapter, we propose combining control and communication strategies to distribute the security process presented in Chapter 3. As a result, we can successfully decentralize the detection process in order to scale and improve the detection performance of our construction.

The outline of this chapter is summarized as follows. Section 4.3 describes the problem analyzed in Chapter 3 to detect a parametric cyber-physical adversary. Section 4.4 proposes a theoretical solution to cover security system weaknesses against cyber-physical adversaries. Section 4.5 validates the solution via numerical simulations. Section 4.6 presents some applications our the proposed strategies developed in this chapter. Section 4.7 discuses the results. And Section 4.8 concludes the chapter.

4.3 Problem Formulation

Considering the same system model used in Chapter 3, we now focus on adversaries able to use parametric identification tools to escape detection by hiding their system manipulations. These adversaries can use the same identification techniques used by the industrial operators to create the control feedback used at their control centers [62, 108, 109]. A non-exhaustive list with traditional techniques used to identify and model physical systems follows.

Some simple and early models are based on FIR (finite impulse Response) and ARX (Autoregressive Exogeneous) identification. Some more advanced and popular models include as well ARMAX (Auto-Regressive Moving Average Exogeneous), which models also a unknown external input; and LDS (Linear Dynamical state-Space), which uses a dynamic state equation to model the dynamics of the physical system [62]. The aforementioned techniques use the knowledge of the input and output data of the system in order to create an *estimate model*. There are also other regression techniques, such as AR (Auto-Regressive), ARMA (Auto-Regressive Moving Average) or ARIMA (Auto-Regressive Integrated Moving Average) that use only the output data to create correlations among observations. Taking into account that the systems addressed in our work, which uses a feedback control to manage the physical system, we limit our focus to models which use the input and the output data of the systems in order to create the *estimate models*.

Regarding the capability of the adversaries to identify the system using well-know identification tools, and the correct parameters analyzed in Chapter 3, it is worth noting that the only constraint found by the adversary could be the synchronization between the input and the output data. For this reason, it is necessary to improve the physics-based attack detection algorithms [109, 110], in order to avoid that these adversaries bypass the detectors (e.g., by sending the expected behavior of the system while driving the system to unsafe states). A limitation to develop a detector against these powerful adversaries is the impossibility to modify the physical behavior of the system. In other words, the behavior between the actuator and the sensors cannot be modified. However, the data sent from the controller to the actuators, or the data sent from the sensors to the controller, can be modified. Taking this limitation into account, as well as the powers of cyber-physical adversaries, we develop in the rest of this chapter a new control strategy for decentralizing the watermark-based detector approach presented in Chapter 3.

4.4 Detecting Parametric Cyber-Physical Adversaries

In Chapter 3, we have seen that watermark-based detection schemes are able to handle attacks carried out by adversaries with limited knowledge about the system dynamics, e.g., the ones defined as either cyber adversaries or non-parametric cyber-physical adversaries (cf. Chapter 3, Sections 3.4 and 3.5). Nevertheless, it fails at detecting those adversaries with enough knowledge about the system dynamics, defined as parametric cyber-physical adversaries in Chapter 3. In this section, we present a detector strategy, hereinafter denoted as Periodic and Intermittent Event-Triggered Control Watermark Detector (PIETC-WD), that aims at detecting both cyber and cyber-physical adversaries.

Our scheme consists of a local controller located in the sensors and a remote controller creating a distributed controller (cf. Figure 4.1). The cooperation between the local and the remote controller allows us to create an intrusion detection policy to capture integrity attacks (cf. Definition 4.1). The local controllers manage the dynamics of the plant, and the remote controller manages the system closed-loop in order to ensure the system against integrity attacks. Notice that our new scheme requires an additional controller together with the sensors, that must have enough computation power to process data estimations, e.g., to predict errors between environmental and estimated data. The actuators do not require additional computational power. Nevertheless, during the time between two consecutive events, they must keep the last data received from the remote controller.

To carry out our scheme, it is necessary to define communication policies among the sensors, the actuators and the remote controller. We define two communication policies for ensuring the system: (i) *periodic communication policy*, which the communication from the sensors to the remote controller is periodic, with a $T_{sc} = 1/f_{sc}$ period, and also from the remote controller to the actuators, with a $T_{ca} = 1/f_{ca}$ period; and, (ii) *intermittent communication policy*, which allows for sending data from the sensors to the remote controller if the local controller produces an alarm. Notice that T_{sc} cannot be equal to T_{ca} to avoid that an intermittent communication takes place while the periodic communication is being sent.

Definition 4.1. *Periodic and Intermittent Event-Triggered Control Watermark Detector (PIETC-WD) is a detector strategy with distributed control tasks. On the one hand, the sensors control the system periodically, using their local controllers and a local watermark-based detector [9]. On the other hand, the remote controller uses the estimation error received from each sensor to periodically generate the control inputs. The remote controller also controls the closed-loop communication with an intermittent watermark.*

The communication algorithms, used to carry out the PIETC-WD strategy, is shown in Algorithm 2 and 3. Algorithm 2 shows the remote controller implementation whose input is the data sent by the sensors, $data_{sc}$, and its output is the control inputs sent by the controller, $data_{ca}$, and the alarm value, $alarm_c$. Algorithm 3 shows the local controller implementation, placed in the sensors, whose input is the data obtained from the physical system, y_t^i, and its output is: (i) the

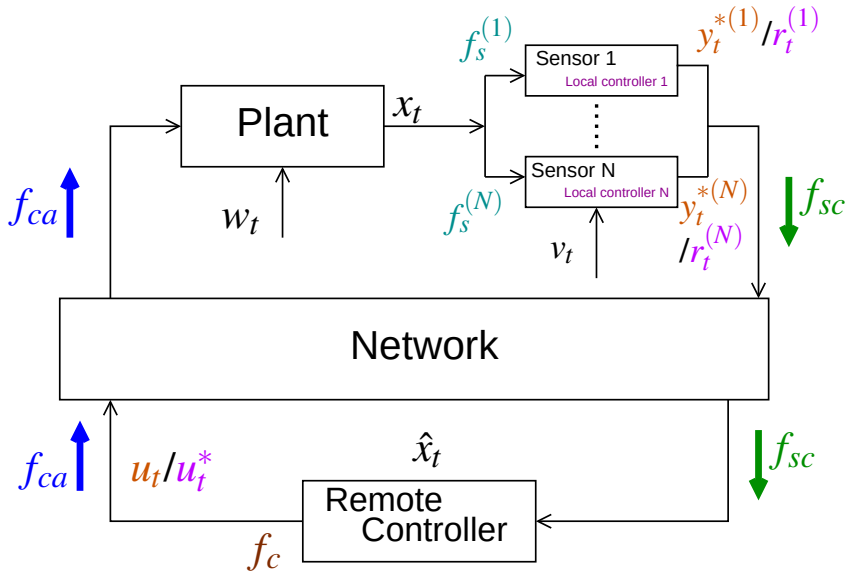


Figure 4.1 – **Cyber-physical system diagram with the new control security strategy.**

residue of the local controllers, r_t^i (with a challenge response watermark), if the alarm is not activated; or (ii) the value obtained by the sensors from the physical system, y_t^i, if the alarm is activated. Briefly, in these algorithms we have implemented how the remote controller manages the data in order to increase the alarm if the data sent from the local controllers is not correct, or data are dropped out. And how sensors modify the data sent to the controller, if an alarm is activated by sensors. We provide more information about the controllers and the communication policies in the following subsections. Before starting with the explanation of the controllers and the communication policies, it is necessary to define some parameters:

$$\begin{aligned} \mathsf{U} &= \{u_0, u_{T_{ca}}, \ldots, u_{\kappa T_{ca}}\} \quad \text{control input variables} \\ \mathsf{Y} &= \{y_0, y_{T_{sc}}, \ldots, y_{\eta T_{sc}}\} \quad \text{observed measurements} \end{aligned}$$

where $\alpha T_{ca} = T_{sc}$ with α, η and $\kappa \in \mathbb{N}$.

4.4.1 Local Controller Design

The local controller is located in the sensors and uses a watermark in order to verify that the dynamics of the system is correct. Each sensor has a local controller with a LQG approach (cf. Section 3.3). We denote the local controller in each sensor by $i \in \{0, 1, \ldots, N-1\}$, where N is the number of sensors in the system. This controller adds a watermark to the sensor measurement before sending the residue to the remote controller:

$$y_t^{(i)} = y_t^{*(i)} + \Delta y_t^{(i)} \tag{4.1}$$

Algorithm 2 Communication policies in the controller.

```
1: procedure CONTROLLER ALGORITHM
2:     T_sc ← sensors/remote controller cycle
3:     T_ca ← remote controller/actuators cycle
4:     data_ca ← data sent by the remote controller
5:     data_sc ← data sent by the sensor
6:     w_c ← controller alarm window
7:     r_data ← reception data
8: top_c:
9:     if r_data ≠ True & mod(t, T_sc) == 0 then
10:        alarm_c ← alarm_c + 1.
11:        goto send_c.
12:    else
13:        if r_data == True then
14:            if mod(t, T_sc) == 0 then
15:                g_t^i ← χ²(r_t^(i)).
16:                if g_t^i > Threshold then
17:                    alarm_c ← alarm_c + 1.
18:                goto send_c.
19:            else
20:                if Δu == 0 then
21:                    alarm_c ← alarm_c + 1.
22:                else
23:                    r_t^(i) ← data_sc^(i) − C_i x̂_{t|t−1}^(i).
24:                    g_t^i ← χ²(r_t^(i)).
25:                    if g_t^i > Threshold then
26:                        alarm_c ← alarm_c + 1.
27:        goto send_c.
28: send_c:
29:    if alarm_c == w_c then
30:        attack ← True.
31:        alarm_c ← 0.
32:    if mod(t, T_ca) == 0 then
33:        data_ca ← u_t.
34:    t ← t + 1.
35:    goto top_c.
```

$$r_t^{(i)} = y_t^{(i)} - C_i \hat{x}_{t|t-1}^{(i)} \tag{4.2}$$

where $y_t^{*(i)}$ is the sensor measurement, $\Delta y_t^{(i)}$ is the watermark added by the local controllers, and $r_t^{(i)}$ is the residue sent to the remote controller to compute the control input $u_t = [u^{(0)}, ..., u^{(N-1)}]$. Notice that the new sensor measurement $y_t^{(i)}$ is computed after verifying that $y_t^{*(i)}$ is correct.

Algorithm 3 Communication policies in the sensors.

procedure SENSOR ALGORITHM
  $T_{sc} \leftarrow$ *sensors/remote controller cycle*
  $w_s \leftarrow$ *sensor alarm window*
top_s:
  $r_t^* \leftarrow y_t^i - \hat{y}_t^i$.
  $g_t^i \leftarrow \chi^2(r_t^*)$.
  if $g_t^i > Threshold$ **then**
    $alarm\_s \leftarrow alarm\_s + 1$.
  if $alarm\_s == w_s$ **then**
    $data_{sc} \leftarrow y_t$.
    $alarm\_s \leftarrow 0$.
  else
    if $mod(t, T_{sc}) == 0$ **then**
      $r_t^{(i)} \leftarrow y_t^{(i)} - C_i \hat{x}_{t|t-1}^{(i)}$.
      $data_{sc} \leftarrow r_t$.
  $t \leftarrow t + 1$.
  $\hat{y}_t^i = Local\_control\_feedback(r_{t-1}^*)$.
  goto top_s.

Remote Controller Design

The remote controller receives periodically the residue of each sensor, $r_t^{(i)}$, and computes these residues using the *LQG* approach (cf. Section 3.3) to obtain the state estimation:

$$\hat{x}_t = \hat{x}_{t|t-\rho} + K_t(r_t) \tag{4.3}$$

and

$$r_t = r_\tau \qquad \forall t \in [\tau, \tau + (\alpha - 1)T_{ca}] \tag{4.4}$$

where r_t is a vector generated by all the residues of the sensors and r_τ with $\tau \in [t : mod(t, T_{sc}) = 0]$ is the periodic vector of the residues. In Equation (4.3), $\hat{x}_{t|t-\rho}$ is defined as:

$$\hat{x}_{t|t-\rho} = A^\rho \hat{x}_{t|t-\rho} + \sum_{j=1}^{\rho} A^{j-1} B u_{\rho-j+1} \tag{4.5}$$

where $\rho \in \{1, ..., \alpha\}$.

Equation (4.3) works properly if the data used by each sensor are independent among them or the system has only one sensor. Otherwise, if the data received by sensors are correlated, it is necessary to add a rectification for each sensor's residue in order to consider the cross-correlation among the watermarks added by local controllers. We can compute this rectification

for $t \in [T_0, T_0 + T - 1]$ as follows:

$$r_t^{rect}(i) = r_t(i) - C_i \Delta \hat{x}^i_{t|t-T} \tag{4.6}$$

Where $r_t^{rect}(i)$ is the rectified value respect to different sensors. And $\Delta \hat{x}^i_{t|t-T}$ is the rectification.

$$\Delta \hat{x}^i_{t|t-T} = \sum_{j=0}^{t-T_0-1} ((A_i + B_i L_i)^{j+1} K_i(\varepsilon_r)) \tag{4.7}$$

where $\varepsilon_r = r_{t-j-1} - r^{(i)}_{t-j-1}$.

We can define the control inputs vector, u_t, as follows:

$$\begin{aligned} u_t &= L(\hat{x}_{t|t-\rho} + K_t r_t^{rect}) \\ &= L(\hat{x}_{t|t-\rho} + K_t(r_t^* + \Delta y_t)) \end{aligned} \tag{4.8}$$

where r_t^* is the vector of the residues before adding the watermark and their respective rectification, and Δy_t is the vector generated by all the sensors' watermarks.

The watermark used intermittently by the remote controller is added to the control inputs. The controller adds a watermark with probability β. Denoting $\lambda_t = 1$ or 0 as indication function whether the watermark is added or not, we assume that $\lambda' s$ are independent and identically distributed (iid.) Bernoulli random variables with $E[\lambda_t] = \beta$.

The intermittence of the watermark communication allows us to define the watermark behaviour as a non-stationary distribution. This watermark, Δu_t (cf. Equation (3.5)), permits us to detect if the closed-loop is being manipulated. It is worth noting that Δu_t is a stochastic signal with the same variance as Δy_t.

4.4.2 Periodic Communication Policy

The periodic communication policy is managed by the sensors. The sensors add the watermark in the measurements received by the plant and send the residue r_t^i to the remote controller. The remote controller uses these residues to generate the control inputs sent to the actuators. The actions of these actuators produce change in the state of the plant that are captured by the sensors. If the real state differs from the state estimated by the sensors, then the sensors will switch from periodic communication policy to intermittent communication policy (cf. Section 4.4.3).

In order to validate the proposal, let us assume that an attack is started at time T_0 and we compute the residue $r_t^{(i)}$ for $t \in [T_0, T_0+T-1]$:

$$r_t^{(i)} = y_t'^{(i)} - C_i \hat{x}_{t|t-T}^{(i)} \tag{4.9}$$

where $y_t'^{(i)}$ is the sensor measurement sent to the controller by the adversary. Moreover, it is easy to show that the following holds:

$$\begin{aligned} \hat{x}_{t|t-T}^{(i)} &= \hat{x}_{t|t-T}'^{(i)} + \mathcal{A}_i^{t-T_0}(\hat{x}_{T_0|T_0-1}^{(i)} - \hat{x}_{T_0|T_0-1}'^{(i)}) \\ &\quad + \sum_{j=0}^{t-T_0-1} (\mathcal{A}^j (A_i + B_i L_i) K_i(\varepsilon_{\Delta y})) \end{aligned} \tag{4.10}$$

where $\varepsilon_{\Delta y} = (\Delta y_{t-1-j}^{(i)} - \Delta y_{t-1-j}'^{(i)})$, $\hat{x}'^{(i)}$ is the local estimated state for each sensor when the system is under attack and $\mathcal{A}_i = (A_i + B_i L_i)(I_i - K_i C_i)$ is a stable matrix [9]. Substitution of (4.10) in (4.9) yields:

$$\begin{aligned} r_t^{(i)} &= \underbrace{y_t'^{(i)} - C_i \hat{x}_{t|t-T}'^{(i)}}_{\text{First term}} \\ &\quad \underbrace{- C_i \mathcal{A}_i^{t-T_0}(\hat{x}_{T_0|T_0-1}^{(i)} - \hat{x}_{T_0|T_0-1}'^{(i)})}_{\text{Second term}} \\ &\quad \underbrace{- C_i \sum_{j=0}^{t-T_0-1} (\mathcal{A}^j (A_i + B_i L_i) K_i(\varepsilon_{\Delta y}))}_{\text{Third term}} \end{aligned}$$

Let us consider separately the three terms in the equation written above: the first term follows the same distribution of $(y_t - C_i \hat{x}_{t|t-1}^{(i)})$; since $\mathcal{A}_i$ is asymptotically stable – i.e. all its eigenvalues are inside the open unit disk of the complex plane – the second term converges exponentially to zero. In fact, the entries of $\mathcal{A}_i^{t-T_0}$ converge exponentially fast to zero. The third term, under attack, is not equal to zero, since $\Delta y_t^{(i)} \neq \Delta y_t'^{(i)}$, and the adversary is detected; for a cyber adversary viewpoint, the measurements of the sensors change all the time and replay measurements are not accepted; likewise, a cyber-physical adversary is not able to obtain the system model using the methodology proposed in Section 3.6. The parametric cyber-physical adversary model, H_{at}, using the ARX (Autoregressive with exogenous input) identification approach [66], is computed as follows:

$$H_{at} = f_{at}(u_t, r_t, y_t, v_t) \tag{4.11}$$

where f_{at} is a linear function that map the input u_t, as a function of t to the output (residue r_t, or measurement y_t) using a process noise, v_t, to measure a stochastic error.

Assuming that the real model is, $H = f(u_t, y_t, v_t)$, a linear function. And the model shown by the system is: $H = f_s(r_t, y_t, v_t, u_t, \Delta u_t, \Delta y_t)$, a non linear function, because we add Δy_t in the measurements of the sensors periodically, and Δu_t in control input, as stochastic event, avoiding to follow a linear function to describe the system. For this reason, a cyber-physical adversary is detected.

4.4.3 Intermittent Communication Policy

The aforementioned periodic communication policy is managed by the sensors. The sensors produce an alarm if $g_t \geq \gamma$. When a sensor produces an alarm, this information is sent immediately to the remote controller. The affected sensor sends the real measurement to the remote controller in order to carry out a second verification. This alarm is activated in a sensor if the control input has been manipulated by an external entity, a problem occurs in the system or the remote controller adds the watermark in the control input.

As we have defined before, remote controller can generate an intermittent communication to verify that the closed-loop is not broken. For an SISO (single-input, single-output) system or a SIMO (single-input, multiple-output) system, the remote controller follows the next steps:

- The controller uses a low-pass filter after computing the control input, u_t, in order to remove the variations, and generate an optimal control input, u_t^*.

$$u_t^* = L(\hat{x}_{t|t-\rho} + K_t r_t^*) \tag{4.12}$$

- The remote controller adds the own watermark.

$$u_t = L(\hat{x}_{t|t-\rho} + K_t r_t^*) + \Delta u_t \tag{4.13}$$

The local controllers located in the sensors receive the modification. They generate an alarm and send the measurement of the sensors to the remote controller. Then the remote controller verifies if the measurement of the sensors is correct using the χ^2 detector. If the detector value is under the threshold the system works correctly. Otherwise, if this value is over the threshold, the remote controller generates an alarm. In both case, the next control input is computed as follows:

$$u_{t+T_{ca}} = L(\hat{x}_{t|t-\rho} + K_t r_t) \tag{4.14}$$

where the residue r_t follow the Equation (4.4).

For a MISO (multiple-input, single-output) system or a MIMO (multiple-input, multiple-output) system the remote controller follows the next steps:

- The controller computes the control input:

$$u_t = L(\hat{x}_{t|t-\rho} + K_t r_t) = [u^{(1)}, ..., u^{(i)}, ..., u^{(n)}] \tag{4.15}$$

  where n is the number of control inputs.

- Then the new control input vector is generated as follows:

$$u_t \begin{cases} u_t = [u^{(1)}, ..., u^{(i)}, ..., u^{(n)}] & \forall i \neq j \\ u_t^{(i)} = u_t^{*(i)} + \Delta u_t^{(i)} & i = j \end{cases}$$

  where $u_t^{*(j)}$ is the control input of the actuator, j, filtered in order to remove the variations. After removing the variation, the remote controller adds its watermark. It is worth noting that j is the actuator chosen by the controller in order to send the watermark.

As we have shown before, if the detector value, generated from the real data sent by the sensor, is under the threshold the next control input is computed as follows:

$$u_{t+T_{ca}} = L(\hat{x}_{t|t-\rho} + K_t r_t) \tag{4.16}$$

Otherwise, the remote controller generates an alarm.

When the remote controller receives a measurement from a sensor, if a watermark, Δu, has not been sent, then the remote controller creates an intrusion alarm. Otherwise, if a watermark has been added to the control input, the controller verifies if this alarm is produced by the watermark. If the residue generated between the real measurements of the sensors and the estimation is under the threshold, the remote controller sends the control input generated before adding the watermark. However, if the residue is over the threshold, it means that an external entity is into the closed-loop, and an alarm is activated.

In order to validate our claims, let us assume the following attack in the communication channel between the sensor and the controller after the controller sends a control input with a watermark. It is started at time T_0, and the remote controller includes the remote watermark, Δu_t at time $T_1 \in [T_0, T_0 + T - 1]$. We compute the residues r_t for $t \in [T_0, T_0 + T - 1]$:

$$r_t \begin{cases} r_t = [r^{(1)}, ..., r^{(i)}, ..., r^{(n)}] & t \in [T_0, T_1] \\ r_t = y'_t - C\hat{x}_{t|t-T} & t \in [T_1, T_0 + T - 1] \end{cases} \tag{4.17}$$

On the one hand, it is worth noting that the attack is not detected between T_0 and T_1, since the adversary is able to replay or insert a correct residue without being detected. On the other hand, for $t \in [T_1, T_0 + T - 1]$, it is easy to show that the following holds:

$$\begin{aligned} \hat{x}_{t|t-T} &= \hat{x}'_{t|t-T} + \mathcal{A}^{t-T_1}(\hat{x}_{T_1|T_1-1} - \hat{x}'_{T_1|T_1-1}) \\ &\quad + \sum_{j=0}^{t-T_1-1} (\mathcal{A}^j B(\Delta u_{t-1-j} - \Delta u'_{t-1-j})) \end{aligned} \tag{4.18}$$

Substitution of (4.18) in (4.17) yields:

$$\begin{aligned} r_t &= \underbrace{y'_t - C\hat{x}'_{t|t-T}}_{\text{First term}} \\ &\quad \underbrace{- C\mathcal{A}^{t-T_1}(\hat{x}_{T_1|T_1-1} - \hat{x}'_{T_1|T_1-1})}_{\text{Second term}} \\ &\quad \underbrace{- C \sum_{j=0}^{t-T_1-1} (\mathcal{A}^j B(\Delta u_{t-1-j} - \Delta u'_{t-1-j}))}_{\text{Third term}} \end{aligned}$$

The first term follows the same distribution of $(y_t - C\hat{x}_{t|t-1})$; the second term converges exponentially to zero. Since the third term is not equal to zero, $\Delta u_t \neq \Delta u'_t$, the adversary is detected; from the cyber adversary viewpoint, the measurements of the sensors change all the time and replay measurements are not accepted; likewise, the cyber-physical adversary is not able to obtain the system model.

4.4.4 New Parametric Cyber-Physical Adversary

In this section we present a new parametric cyber-physical adversary with the knowledge about the new detector strategy, in order to evaluate the new detection strategy. This attacker has knowledge about the new communication policies and the existence of the local and the remote watermarks. Nevertheless, the new adversary does not know the watermark co-variances, the controller's parameters used to obtain the correct error between data, and neither the moment when the remote controller forces an intermittent communication.

The new adversary could be able to detect the correlation model between the inputs and the outputs of the plant. This adversary can force the intermittent communication of the sensors with malfunction control inputs, and mislead the controller with replay error data to obtain the model. Nevertheless, this adversary is not able to know when the communication is periodic or intermittent, since the attacker does not know when the remote control sends the watermark added to the control inputs which generates the intermittent communication. The intermittent

communication does not change the communication frequency between the remote controller and the actuators, but produces an intermittent communication between the sensors and the remote controller, necessary to verify the closed-loop.

Briefly, the new adversary is able to attack the integrity of the system. Nevertheless using the PIETC-WD strategy, this kind of adversaries are detected by the controllers of the sensors. The remote controller detects the attack when it verifies the behaviour of the closed-loop. These adversaries cannot avoid the alarm in the sensors (local controller). Nevertheless, the attackers can cut off the communication between the sensors and the remote control misleading the remote controller with correct residues (e.g. replay residues). Moreover, in order to avoid the alarm in the remote controller, the adversaries can switch between sending the measurement of the sensors or the residues, but they have a great probability to be detected. We validate the PIETC-WD strategy against the new parametric cyber-physical adversaries in the next section.

4.5 Numerical Validation

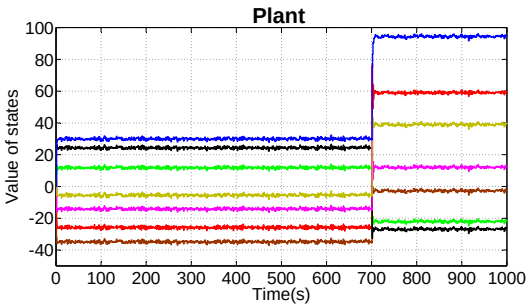
This section presents a numerical validation of the strategy. We have simulated a MIMO system which represents a cyber-physical system with four inputs and four outputs. To define mathematically this system using the model shown in Section 3.3, we use the following matrices[1]:

$$A=\begin{bmatrix} -0.199 & 0.145 & -0.01 & -0.119 & 0.062 & 0.134 & 0.001 \\ 0.0413 & -0.189 & -0.124 & 0.037 & 0.167 & 0.059 & 0 \\ 0.031 & -0.097 & -0.344 & -0.055 & -0.138 & 0.213 & -0.002 \\ -0.156 & -0.030 & 0.042 & -0.014 & 0.005 & 0.143 & 0 \\ -0.016 & 0.162 & -0.129 & 0.118 & -0.199 & 0.005 & 0.001 \\ 0.096 & 0.082 & 0.203 & 0.042 & -0.028 & -0.147 & 0 \\ -0.250 & 0.087 & -0.064 & -0.134 & -0.093 & 0.075 & 0.447 \end{bmatrix}, \quad B=\begin{bmatrix} -1.739 & 0 & 0 & 0 \\ -0.214 & 0 & 0.702 & 0 \\ 1.437 & 0.001 & -1.6 & -0.711 \\ -1.107 & 0.002 & 0 & 1.548 \\ 0 & 0 & -0.952 & 0 \\ 1.197 & 0 & -1.01 & 0 \\ 0.953 & 0.765 & -1.44 & 0 \end{bmatrix},$$

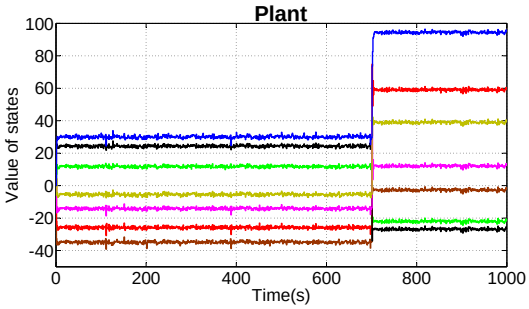
$$C=\begin{bmatrix} -2.261 & 1.112 & -0.09 & 0 & 0.457 & 0.829 & -1.137 \\ 0.091 & 0.259 & 0.876 & -0.273 & 2.236 & -1.484 & -1.091 \\ 0 & 0.689 & -0.074 & 0 & -0.111 & 0 & -0.310 \\ 0 & 0.524 & -0.195 & 0 & 0 & -0.138 & 0 \end{bmatrix}.$$

We present in Figure 4.2 the dynamics of the plant and how the adversary can modify this dynamics in order to disrupt the system. Figure 4.2(a) shows the dynamics of the plant using the periodic communication policy detailed in Section 4.4.2. The different between this control/security policy, used to detect attacks, and the detector used in Chapter 3, is that this policy places the detection and the alarm in the sensors, moving from a centralized detection policy to a distributed detection policy. Otherwise, this policy is not enough to detect the attack, since the adversary can avoid alarm signals arriving to the remote controller. Figure 4.2(b) shows the dynamics of the plant using a combination between periodic and intermittent policies which allow transmit the alarm until the remote controller without be intercepted by the adversary. It is worth mentioning that the dynamics of the plant does not change if we use periodic or periodic an intermittent policies, accomplishing the physical system constrains defined before in Section 4.3.

[1] The co-variance matrices are equal to $Q = 0.2I$ and $R = I$.

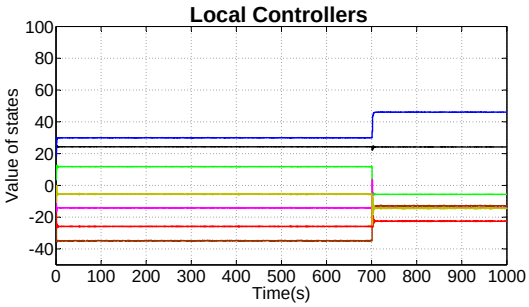


(a) Dynamics of the plant using only periodic policy.

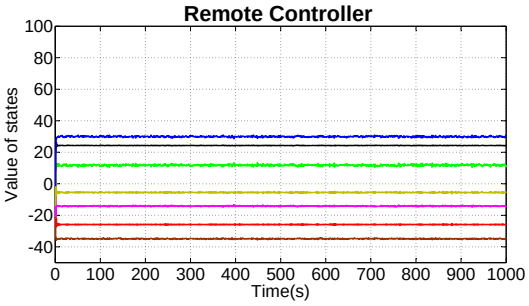


(b) Dynamics of the plant using periodic and intermittent policies.

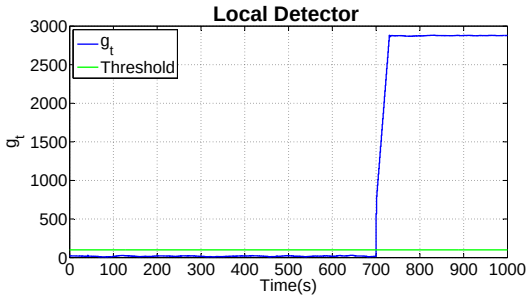
Figure 4.2 – **The dynamics of the states vector in the plant under a parametric attack.** The attack starts at $t = 700s$. (a) using only periodic policy; and (b) using periodic and intermittent policies, i.e., the PIETC-WD control security strategy.



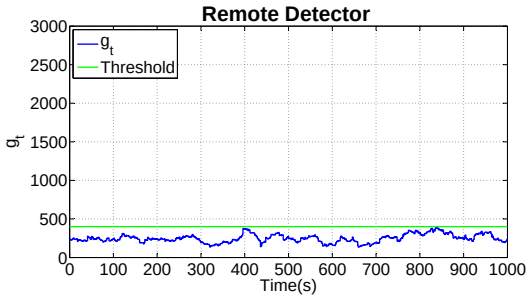
(a) Estimated states in the local controller using only periodic policy.



(b) Estimated states in the remote controller using only periodic policy.



(c) Sensor detector results using only periodic policy.

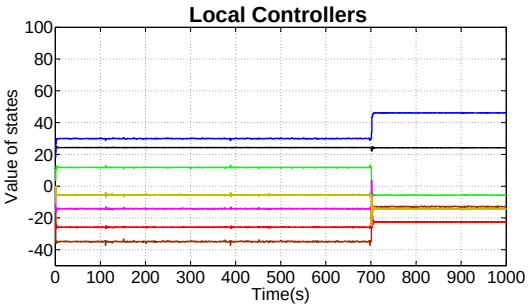


(d) Control center detector results using only periodic policy.

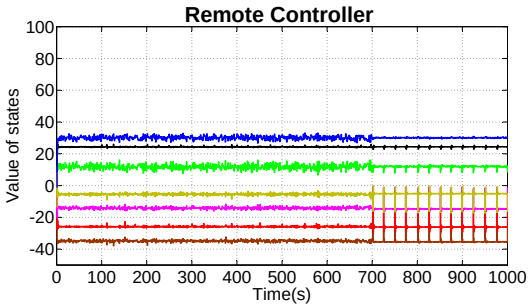
Figure 4.3 – **Numeric simulation results.** Attacks start at $t = 700s$. (a),(b) The dynamics of the states vector estimated in the local and remote controller respectively, under the same scenarios. (c),(d) The dynamics of the alarm signal g_t produced in the local and remote detector respectively, under the same scenarios.

Figures 4.3 shows how the detectors present in the sensors are able to detect attacks using the periodic communication policy and a watermark managed by sensors. Figure 4.3(a) and 4.3(b) show the dynamics of the states estimated by the local controllers placed in sensors and the

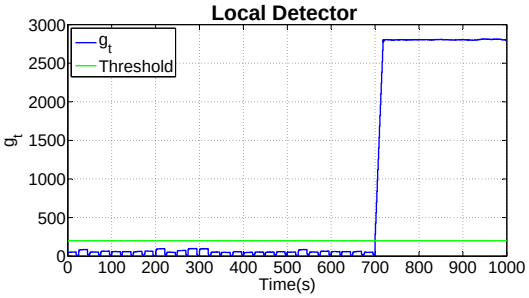
remote controller placed in control center respectively. And Figure 4.3(c) and 4.3(d) show the dynamics of the alarm in sensors (local detectors) and in the control center (remote detector). We show in these figures an attack which is detected by the local detectors, which control the dynamics of the system, but whose alarm's dynamics is not transmitted to the control center. That happens due to the adversary capability to intercept the data and supplant the correct behaviour. The alarms of the sensors are stopped by a cyber-physical adversary which takes the control of the system and sends to the remote controller a response correlated with the data sent from the remote controller (control center) to the actuator. This action allows adversary attacking the system without being detected.



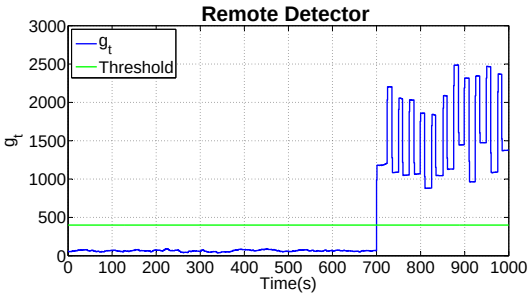
(a) Estimated states in the local controller using periodic and intermittent policies.



(b) Estimated states in the remote controller using periodic and intermittent policies.



(c) Sensor detector results using periodic and intermittent policies.



(d) Control center detector results using periodic and intermittent policies.

Figure 4.4 – **Numeric simulation results.** Attacks start at $t = 700s$. (a),(b) The dynamics of the states vector estimated in the local and remote controller respectively, under the same scenarios. (c),(d) The dynamics of the alarm signal g_t produced in the local and remote detector respectively, under the same scenarios.

Figure 4.4 presents the control/security strategy using the periodic and intermittent communication policies. As shown in Figure 4.4, it is visible that using both policies allows the system to detect the attack. Figure 4.4(c) shows the local detector that is able to detect a disruption in the dynamics of the plant, thanks to the periodic communication policy. In Figure 4.4(d) we can see that the remote detector (placed in the control center) is able to detect that the closed-loop is broken. This detection is possible, thanks to the remote controller's watermark and the intermittent communication policy. It is worth noting that the state estimated by the local and remote controller are not modified by our strategy (cf. Figures 4.4(a) and 4.4 (b)).

4.6 Use Case

This section presents a practical use case, where the PIETC-WD strategy proposed in previous sections, could be used in the real-world. The use case is based on a chemical plant. This plant has multiple sensors with local controllers, actuators and a remote controller, which manages all the measurements of the sensors and actuators. The sensors used in this use case send information about pressure, temperature, and density. This information is produced when there is an alarm, and also periodically to indicate the behaviour of the system to the controller. This plant has to be controlled periodically since, if during ten consecutive periodic samples, the system receives wrong or malicious control inputs able to disrupt the system, a critical state might be reached.

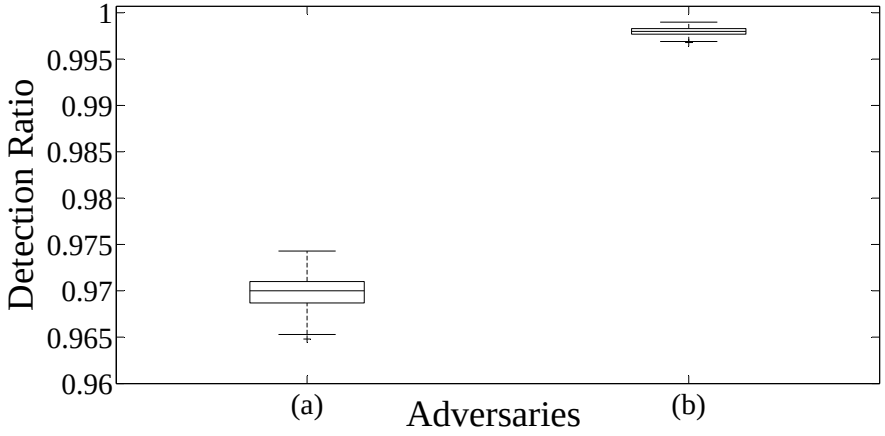


Figure 4.5 – Detection ratio function with respect to the PIETC-WD strategy with a defined controller's watermark policy; (a) against a parametric cyber-physical adversary; and (b) against cyber or other cyber-physical adversaries.

To avoid that an adversary gets the system into a critical state, we use our detector strategy (PIETC-WD), with a policy for the remote controller's watermark defined as follows:

- The controller's watermark uses a policy based on a probability to add the watermark in a specific window of samples. In this use case, the window of samples is assumed equal to five. For each sequence of five control input samples, the probability to add the watermark at each sample is $\zeta = 50\%$. The system is able to produce $2^5 = 32$ different sequences with the same probability to be generated, $\theta = 1/2^5$. Nevertheless, if among these five samples, the system does not send any watermark, three more samples are used to add a watermark to the control input until a new control sequence starts. These three samples added to the original control sequence add $2^3 = 8$ more sequences where the five first samples have not watermark, and the three last samples have the following probability to add the watermark:
  - The probability to add the watermark in the sixth sample is 60%.
  - The probability to add the watermark in the seventh sample is 50% if the watermark is added in the sixth sample. Otherwise, if the watermark is not added, the probability is 60%.

– The probability to add the watermark in the eighth sample is 50%, if the watermark is added in the sixth or seventh sample. Otherwise, the probability is 60%.

Figure 4.5 shows the results of 200 Monte Carlo simulations using the above use case and controller's watermark policy, against the cyber and cyber-physical adversary. These results present that the ratio of detection is around 97% against the new parametric cyber-physical adversary and more than 99% against the other cyber and cyber-physical adversaries using the PIETC-WD strategy with a correct policy for the remote controller's watermark.

4.7 Discussion

In the previous chapter we have improved a challenge-response authentication scheme using a non-stationary watermark. This improvement allows us to move from a static detection scheme [9, 53] to a dynamic detection scheme. Using the new dynamic scheme, we have verified that we are able to detect cyber adversaries, as well as non-parametric cyber-physical adversaries. Nevertheless, if adversaries are able to use parametric identification tools, the dynamic detection scheme is not enough, since these adversaries are able to generate a correlation matrix between the input and the output data of the physical system. To address this limitation, and properly cover all kinds of cyber-physical adversaries, we can decentralize the detection process to other elements of the system.

Based on aforementioned remark, we have proposed to combine control strategies [75] with the dynamic challenge-response scheme proposed in Chapter 3. The combination creates a decentralized detector, able to identify non-parametric cyber-physical adversaries. We have seen that the resulting challenge-response scheme handles the adversaries by adding correlation complexity. Indeed, in a MIMO system (i.e., a system with multiple sensors), local controllers may use independent watermarks unknown by the remote controllers, as well as the potential adversaries. The use of independent watermarks, increasing the difficulty of correlating input and output data, makes even harder the generation of correct residues. The central controller, in charge of receiving all the measurements from the system sensors, computes a *rectification error* that increases the chances of properly detecting the actions of parametric cyber-physical adversaries.

4.8 Summary

In this chapter, we have focused on designing an adaptive control-theoretic strategy that detects parametric cyber-physical adversaries, i.e. adversaries that are able to acquire knowledge about the system dynamics prior starting their attacks, in order to successfully get control over the inputs and measurements of the system. We have followed the idea of decentralizing the detection strategy proposed in Chapter 3, by combining the protection scheme directly with control strategies. By adding security at the *architectural level* of the system [111], we can successfully

preserve the stability and performance of the system, as well as its safety and security properties. The precise architectural design presented and developed in this chapter successfully decentralizes the watermark-based detector. We have validated the approach by simulating an industrial system with multiple actuators and multiple sensors, and high correlation levels between them. We have also presented a practical use-case of the approach, showing that the strategy is able to detect cyber-physical attacks conducted by parametric cyber-physical adversaries. In the next chapter, we show a testbed implementation of our strategy, as well as practical results that validate the feasibility of our proposal.

5 Experimental Testbed for the Detection of Cyber-Physical Attacks

5.1 Introduction

Experimental testbeds are crucial for the study and analysis of ongoing threats against cyber-physical systems. The research presented in this chapter discusses some actions towards the development of a replicable and affordable cyber-physical testbed for training and research. Within this scope, our goal is to put in practice the theoretical solutions developed in previous chapters. To get this target, we have modeled and implemented the solutions under realistic scenarios, in order to analyze their effectiveness against intentional attacks. More precisely, we assume cyber-physical environments operated by SCADA (Supervisory Control And Data Acquisition) technologies and industrial control protocols. We focus on two representative protocols, which are widely used in the industry: Modbus and DNP3 [112, 113]. Both protocols have TCP enabled versions. This allows us the emulation of cyber-physical environments under shared network infrastructures. We assume a Master-Slave design, which mainly dictates that slaves would not initiate any communication unless a given master requests an initial operation. One of our objectives has been to combine these two protocols, both to allow the flexibility and support of several devices with Modbus as well as the security enhancements that DNP3 could provide as one of its features. Furthermore, the cyber-physical detection mechanisms based on challenge-response strategies proposed in Chapter 3 are included in our SCADA testbed. Likewise, we have integrated the control strategy proposed in Chapter 4 to experiment and analyze its real-world performance. To complement the testbed, a set adversarial scenarios are designed and developed to test attacks against the emulated environment. These scenarios focus on attacking the Modbus segments of the SCADA architecture. The final goal is to analyze the effectiveness of novel security methods implemented upon the emulated environment, and under the enforcement of some attack models.

5.2 SCADA Testbed Environment

Different hardware and software tools are used in related works and also are part of this research effort. More details about the most relevant tools are specified in this section.

5.2.1 Lego Mindstorm EV3

The Lego Mindstorm kit includes a variety of hardware and software tools to build personalized robots. This kit consists of a set of modular sensors and actuators with a range of different Lego parts to build mechanical structures. This structure can be powered by the modular actuators which are controlled by an EV3 brick[1]. This brick is the core of the system, being able to trigger the actuators and read data from the sensors. From a research standpoint, Lego Mindstorm kit allows high versatility at the moment of building testbeds. We have chosen this kit for our testbed because the EV3 brick uses an ARM chip that runs a Linux operative system. This allows using complete programming languages such as Java, Python, and C. Despite some limitations, such as the precision of the sensors, the whole package represents a robust tool for research that allows giving a step forward from simulations.

5.2.2 Raspberry Pi

The Raspberry Pi[2] is single-board computer, appeared in 2006, and based on ATmega644 microcontrollers. Build upon a USB-size package, offering USB port and HDMI ports [114], the Raspberry Pi series offers a robust package for embedded computing, making it outstanding for SCADA node roles, such as RTUs and PLCs. It also provides a set of General Purpose Input/Output pins (GPIO), which adds significant compatibility with low-level hardware, such as prevision sensors and actuators. Research results in [115] report the use of this type of devices to control a small water treatment plant having a role as PLC.

5.2.3 Software Libraries

Scapy — Scapy[3] is a powerful packet manipulation library, able to decode network packets from a large number of protocols — Modbus and DNP3 included. It offers tools to handle packets either to send forged packages or to capture them from the network. It also has the ability to inject 802.11 frames or handle VLAN hopping. It can perform ARP cache poisoning, something that is essential for the development of the attackers. The usefulness of this library mostly consists that it is made to handle many different tasks without limitation to only make sniffing for instance. In fact, this library is the single used library to develop the whole range of attackers, supporting to forge responses to deceive TCP.

[1] https://en.wikipedia.org/wiki/Lego_Mindstorms_EV3
[2] https://en.wikipedia.org/wiki/Raspberry_Pi
[3] http://www.secdev.org/projects/scapy/

OpenDNP3 and Jamod — Open-sourced implementations of protocols like DNP3 and modbus exist in the related literature. OpenDNP3 [4], referred by its authors as the de facto reference implementation of IEEE-1815 (DNP3), and offering APIs written in C and Java, provides features such as optimization to run on resource-constrained devices, conformance tests, as well as non-blocking and multi-core support. As for the Modbus protocol, we use the Jamod library [5]. Jamod is an object-oriented implementation of the Modbus protocol completely done in Java. All the specifications, API and Modbus support can be found in [116].

Remainder libraries — Other software libraries used in our testbed include the **Jkalman** Java library[6], used to implement the feedback control, fault detection and watermark-based detection mechanisms; the **Adapfilt** and **Matlab System Identification Toolbox** [117], to implement the adversaries; and **GTK**, in order to implement the graphic user (GUI) interfaces.

5.3 Testbed Architecture

5.3.1 Architecture

Closed-loop systems are systems which rely upon internally gathered information to perform, correct, change or even stop actions. This kind of systems are important in the control theory branch, known to have two-way communication, one to read data and the other to forward commands.

We can observe three important block elements: the controller, the system itself, and the sensors. The controller reads data from the sensors, computes new information and transmit new commands to the system (i.e., the system control input). The system control input is generated by the controller with the purpose of correcting the behavior of the system, under some previously established limits. The system is what we normally see as the entity under control. The sensors are the feedback link between the system and the controller. Their purpose is to quantify the output and provide the necessary information to the controller, in order to compare and, if necessary, correct the behavior of the system.

The architecture proposed for our training cyber-physical testbed works as follows. All the aforementioned elements can be distributed across several nodes in a shared network combining DNP3 and Modbus protocols (cf. Figure 5.1). Likewise, one or various elements can be embedded into a single device. From a software standpoint, the controller never connects directly to the sensors. Instead, it is integrated in the architecture as a SCADA PLC node, with eventual connections to some other intermediary nodes. Such nodes are able to translate the controller commands into SCADA (e.g., either Modbus or DNP3) commands. As depicted in Figure 5.1, the architecture is able to handle several industrial protocols and connect to complementary SCADA

[4] https://www.automatak.com/opendnp3/
[5] https://sourceforge.net/p/jamod/
[6] https://sourceforge.net/projects/jkalman/

elements, such as additional PLCs and RTUs. To evolve the architecture into a complete testbed, new elements can be included in the system, such as additional proxy-like RTU nodes.

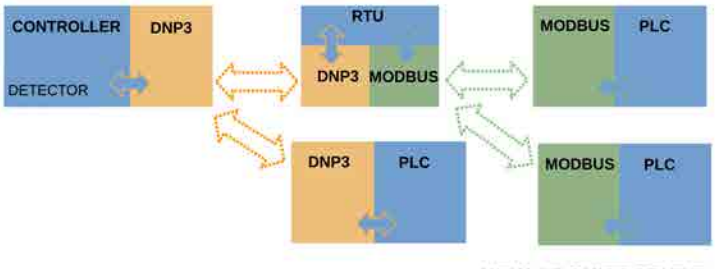


Figure 5.1 – Abstract architecture overview.

From a data transmission standpoint, we include in our testbed the possibility of using different sampling frequencies, in order to cover a larger number of experimental scenarios. The implementation is based on the control theory strategies defined in Chapter 4, i.e., it supports the use of different frequencies when performing read and write operations. This narrows to the sampling frequency to either build a system with mono-frequency sampling — same frequency is used for all the channels — or with multi-frequency sampling — where different sampling frequencies are used in each channel.

The architecture is able to handle many PLCs. To avoid overloading one channel with all the possible registers of the PLCs, separate ports are designated in order to isolate the communication between separated PLCs. DNP3 commands perform an Integrity Scan which gathers all the data from the PLCs in case several PLCs were being handled in the same channel, all variables of the a PLC would be fetched causing overhead in the communication.

5.3.2 Implementation Design

The implementation of our SCADA testbed consists on *Lego Mindstorms* EV3 bricks [118] and Raspberry Pi [115] boards as PLCs to control some representative sensors (e.g., distance sensors) and actuators (e.g., speed actuators). We refer the reader to http://j.mp/legoscada for additional information and video captures of the testbed. Figure 5.2 shows an object-oriented representation of the controller implementation, along with connection control classes, exception classes and also graphical interface classes at the controller side. In Figure 5.3, we can see all the classes that have been created in order to achieve the DNP3-Modbus combination, at the RTU side. A proxy-like behavior has been also implemented allowing to translate the commands in both directions for both protocols. The testbed components are defined as follows.

Controller Design

The controller has a graphical interface to show the behavior of the system to an operator. It is orchestrated by the *ControlCenter* class (cf. Figure 5.2). This class handles the graphical

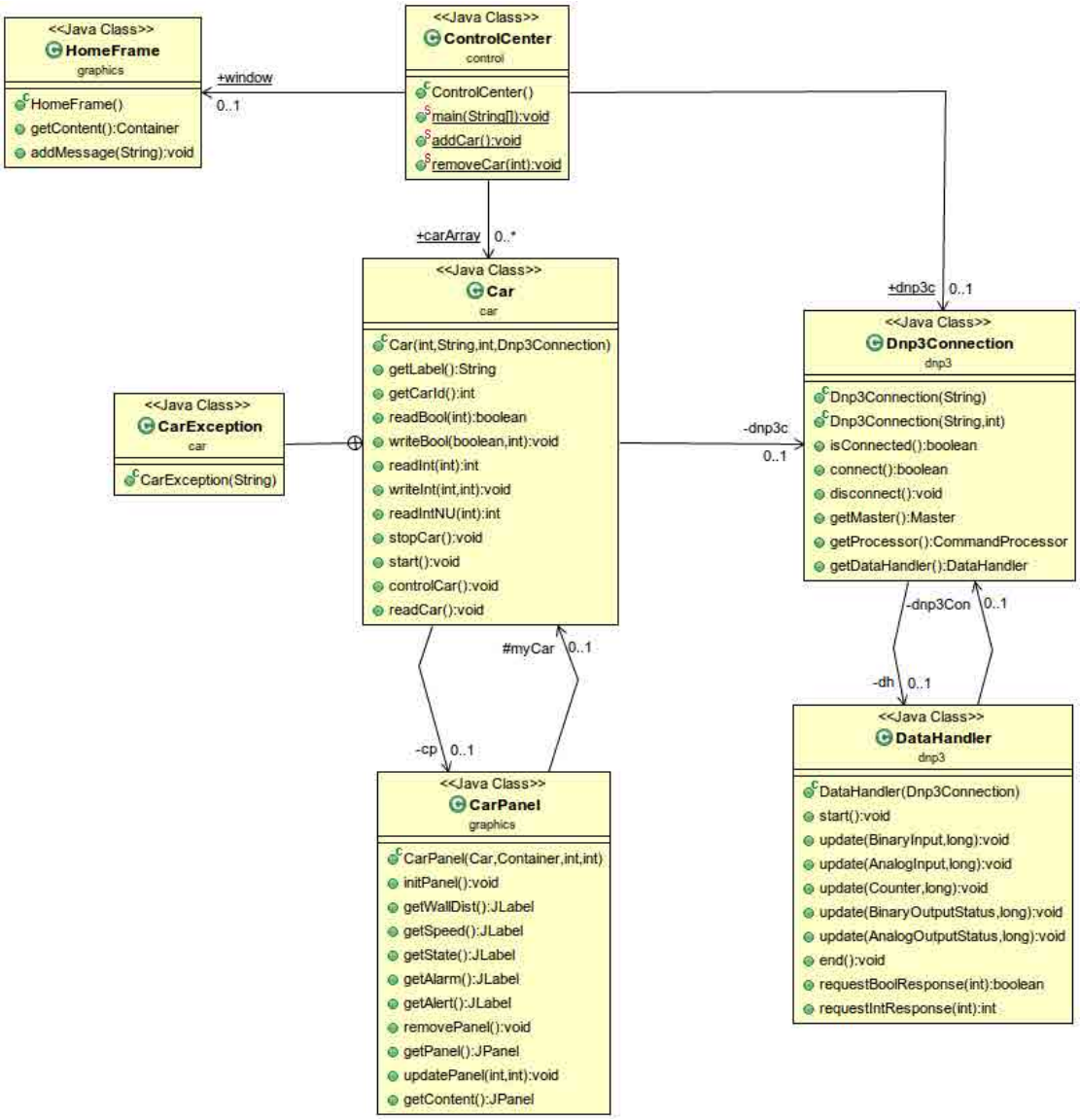


Figure 5.2 – Implementation overview, controller side.

interface (cf. *HomeFrame* class) representing the HMI (Human Machine Interface) of the SCADA architecture. Some PLC instances, e.g., the *Car instances*, subsequently create DNP3 connections under a *DataHandler*, which is in charge of managing the communications between RTUs and PLCs. Finally, some of the instances (e.g., the *Car instance*) implement a graphical component to provide additional information to the operator.

RTU Design

In the implementation, it is possible to have control of one or more PLC instances. For such a task, a dedicated thread manages the translations and constant polling of each PLC. Everything

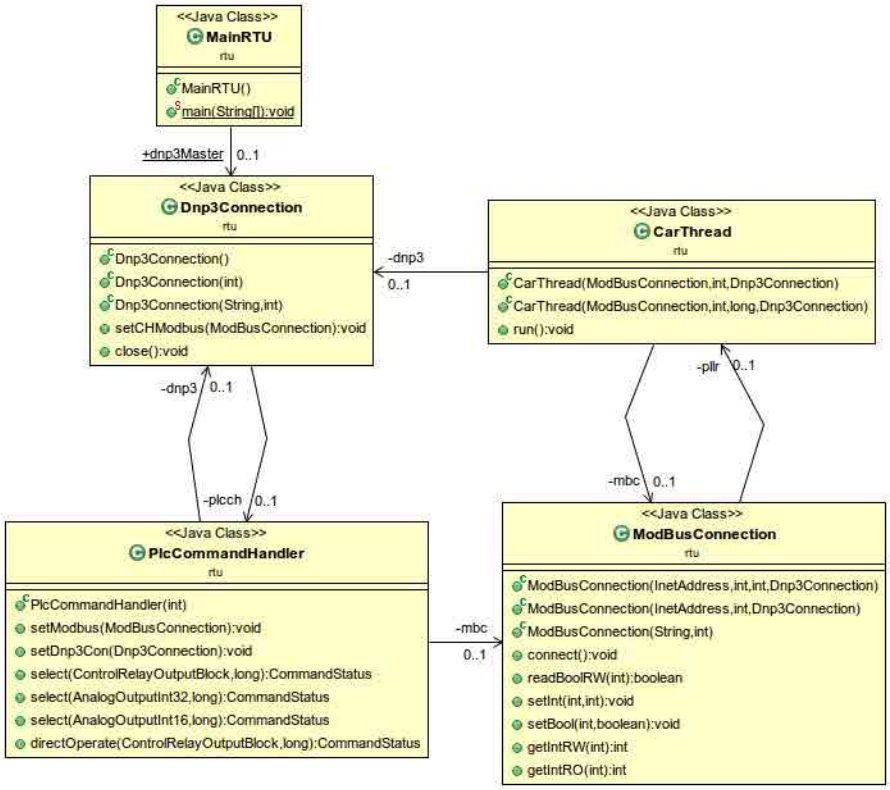


Figure 5.3 – Implementation overview, RTU side.

starts with the *MainRTU* class (cf. Figure 5.3), which opens the main DNP3 connection to expect the controller. Once the controller connects, the RTUs exchange information of the PLCs to add, and create all the respective classes in order to handle each PLC individually and with dedicated ports.

External Tools

Apart from the architecture implementation, other tools have been implemented in order to facilitate the aggregation process of new SCADA nodes. Specific custom scripts have been made to install the OpenDNP3 libraries either compiling them from source or use pre-compiled binaries for the case of the Raspberry Pi. Compiling source is time-consuming if it is done directly at the Raspberry Pi boards. Therefore, cross-compiling or pre-compiled libraries are recommended to avoid long compilation times. The Raspbian scripts give the choice to use pre-compiled libraries.

5.3.3 Test Scenario Description

The architecture proposed before, can be used to generate different testbeds. The most significant difference among the possible testbeds generated using this architecture is the physical process. For instance, the sensors and actuators are not the same in the different testbeds. Nevertheless,

the principle to generate the control feedback and the network architecture is the same in all of them. Figure 5.4 shows four scenarios created using the aforementioned architecture. In the sequel, we focus on the scenario represented in Figure 5.4(d). In this scenario, we implement the control feedback as well as the detection propositions developed in previous chapters, in order to validate the work with real measurements and actions.

(a) Bridge and toll testbed.

(b) Industrial chain testbed.

(c) Railway control testbed.

(d) Autonomous industrial agents testbed.

Figure 5.4 – Different testbed scenarios created with the architecture exposed in this chapter (cf. http://j.mp/legoscada for some video captures).

Figure 5.5 shows the components of the autonomous industrial agents testbed, shown in Figure 5.4(d). This scenario is a simple representation of the architecture proposed in this chapter. It consists of a controller (Personal Computer), an RTU (Raspberry Pi) and a PLC (Lego EV3 Brick). The controller corrects the speed of the car by polling the distance between the car and an obstacle. One single controller and one single RTU can control various PLCs. To start the testbed, it is necessary to execute the Java automaton deployed over each EV3 bricks [116], as well as the automata deployed over the Raspberry Pi boards. Once started the controller and the cars, the controller verifies and controls the dynamics of the car, i.e., the car behavior is continually modified by the controller, hence varying its speed according to the controller's commands.

5.4 Implementing the Adversarial Models

From all the attacks defined in Chapter 2, we focus on attacks which try to mislead the controller using replay data or correct dynamic system's data. These attacks need two entry points (or one entry point if the network is shared among all the devices), in order to disrupt the system. On one side, they mislead the controller in order to bypass the detection, and on the other side, they take the control of the communication between the controller and the actuators, or directly the control of the physical system to harm it. Briefly, all these attacks can be classified, using a very general classification as *Man-in-the-middle (MiM)* attacks. These attacks are very common type of attacks, controlling the communication in both ends, especially if the communication is not properly protected, as is the case with many SCADA implementations. Otherwise, attacks such as the *stealth attacks* whose goal is forcing the controller to drive the physical system to an unsafe mode; *denial of service attacks*, that aim at exploiting cyber vulnerabilities to break the closed-loop communication; or *Zero dynamic attacks* that use controller's vulnerabilities to disrupt the system, use only an entry point in order to attack the system. Nevertheless, it is worth noting that, for instance, the *stealth attacks* and the *Zero dynamic attacks* need to know the system dynamics to carry out the attack.

The testbed proposed in this chapter is expected to provide complementary validation of the watermark detectors reported in Chapters 3 and the detection strategy defined in Chapter 4. After having the entire architecture working, the next requirement is to implement the adversarial scenarios reported previous chapters.

5.4.1 Adversaries

In order to develop the scenarios, a common attacker model is used. It implements most of the underlying capabilities of the opponents, and can be extended in order to implement most specific adversaries. For instance, we assume that attackers can intercept any communication exchanges between ends, and thus the attacker can alter, store, analyze replay and forge false data from and towards the communication channels. Since this is done using a testbed instead of numeric simulations, all real-life limitations are applied to the attacker. ARP poisoning [119] is used by the attacker to intercept the channels and eavesdrop the communications. The attacker has a passive and active mode of operation. The *passive mode* is where the attacker only eavesdrops,

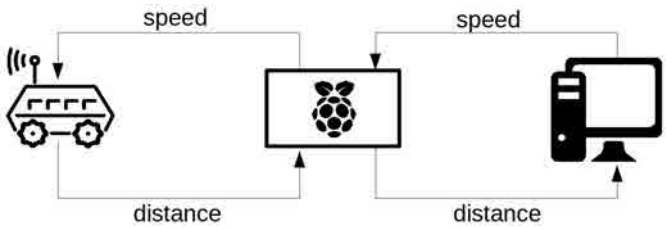


Figure 5.5 – Test scenario overview.

processes, and analyzes the data without modifying the information contained in the payload of the messages. Nevertheless, Ethernet header data, such as the hardware addresses, are modified since ARP tables are poisoned. During the *active mode*, the attacker starts injecting data to the hijacked communication. This injection, depending on the pattern of the attacker, can be a generated response or replayed packets.

Replay Attack

The attackers use ARP poisoning to start eavesdropping the connection (passive mode, from the physical-layer standpoint). After capturing enough data, the *active mode* starts. The attackers inject the old captured data following the stream of packets of the previous capture. Before starting to disrupt the system, the attacker conducts the attack between the sensors and the controller, forging only the TCP headers that correspond to the opened TCP sessions. Once replayed the packets, the system gets disrupted by forging data between the controller and the PLCs.

Injection Attacks

Prior to starting the attacks, the attacker eavesdrops connections using the *physical-layer passive mode*, and analyze the data in order to infer the dynamics of the system. This is used to evade the authentication watermark detector. Once inferred the model of the system, the attacker starts injecting correct data in the communication channel, in order to defeat the watermark countermeasure. To evade the detector, the attacker calculates the effect of the watermark in the system and tries to cancel the ability of the detector to sense the changes in the feedback signal. Two different techniques are implemented: 1) a non-parametric adaptive filter, in order to implement the evasion technique presented in Chapter 3, *a non-parametric cyber-physical attack*; and 2) autoregressive methods, such as ARX and ARMAX, in order to implement the evasion technique presented in Chapter 4, *a parametric cyber-physical attack*.

The challenge to implement these two adversaries is to synchronize the output of the adversaries when starting the attacking phase. Since the target of the adversaries is to take the control of the system, it is necessary that the data sent to the controller are able to match the current state of the system and with the correct watermark correlation to avoid being detected at the beginning.

5.4.2 Attack and Fault Detection

The adaptation of a fault detector in order to detect attacks using an authentication watermark is a valid technique that has been introduced in [9], improved in Chapter 3, and then used in the detection strategy proposed in Chapter 4. These techniques have been implemented in our testbed, in order to assess and analyze their performance using real hardware components. The testbed controllers implement the detector with different types of watermarks (cf. Chapter 3), and

control strategies to reinforce the detector (cf. Chapter 4) . The implementation uses the *JKalman* library [120], with some light modifications to parameterize the system and detect the effect of the watermark in the system's output. The detector estimates the next output and then compares it to the value returned by the physical system. The process uses the χ^2 detector proposed in [9]. The detector returns a metric, g_t, which increases rapidly when the output of the system starts to move away from the estimation. The metric is posteriorly used to generate alerts.

The g_t metric is an in-code operator that quantifies the difference between the parametric model output and the actual system output. An increase of g_t means that the system is not behaving or reacting to the watermark as expected. Therefore, the system is likely to be under attack. The value of g_t is calculated for each iteration and compared with the values of some previous iterations. In order to discard false positives, the controller implements Algorithm 4, to separate normal faults from attacks or severe failures. The algorithm alerts the operator only when real intervention is required, separating faults, e.g., latency or inaccuracy events at the sensor; and intentional attacks. For every feedback sample, the controller analyzes g_t. If g_t consecutively bypasses a given threshold (more than *window* times), then it triggers an $alert$.

Algorithm 4 is composed of four main functions, to differentiate between faults, accidents or attacks. They work as follows.

- *alarm_propagation*: this function creates a potential alarm if the detector value, g_t, is over the threshold. This alarm is potential because it could be a fault peak or an attack. The set of potential alarms is analyzed by the following function (*alert_propagation*), in order to estimate if it is an exceptional fault or a potential attack[7].
- *alert_propagation*: this function creates an alert if the *DR_value* is below a minor threshold, *min_R*, or above a maximum threshold, *max_R*. The value of *DR_value* is the difference between the detector's value at instant t, and the detector's value at instant $t-1$. The function also verifies if the system has generated faults (alerts or potential alarms), during a precise period of time (denoted by *window size*). In fact, this *window size* is the number of samples chosen from the physical system, in order to settle the detector value, g_t. This window has to be chosen following certain conditions, such as the sensibility of the system to modify its stability. The precise value has to be: 1) big enough to minimize the number of false positives; and 2) small enough to have enough time to react under critical situations, e.g., when errors or fault shall be handled, in addition to attacks.
- *potential_risk*: this function presents a given system risk level, taking into account alerts and potential alarms; it follows traditional qualitative risk values [111], such as (1) *slow*, (2) *medium*, (3) *high*, (4) *critical*, and (5) *very critical*.
- *real_risk*: this function warns the system under the presence of a real risk, analyzing the conditions received from the other function, i.e., number of potential alarms and alerts.

[7]Notice that we expressly use the term *alarms* to point out towards suspicious events; and *alerts* to point out to events likely to be associated to malicious attacks.

Algorithm 4 Fault and attack differentiation.

```
1:  procedure DETECTOR
2:      alert, alarm ← false
3:      potential_alarm, potential_attack ← false
4:      window ← detector_window
5:      risk, potential_risk ← 0
6:  alarm_propagation:
7:      if detector_value > threshold then
8:          potential_alarm ← true
9:      else
10:         potential_alarm ← false
11:     old_detector_value ← detector_value
12:     goto alert_propagation
13: alert_propagation:
14:     DR_value ← detector_value / old_detector_value
15:     if DR_value < min_R or DR_value > max_R then
16:         alert ← true
17:         if 0 < account_fault ⩽ window then
18:             risk_level ← risk_level + 1
19:         else
20:             potential_attack ← true
21:             alarm_attack ← true
22:     else
23:         alert ← false
24:     goto potential_risk
25: potential_risk:
26:     switch risk_level do
27:         case window/4: potential_risk ← potential_risk + 1
28:         case 3window/4: potential_risk ← potential_risk + 1
29:         case window: potential_risk ← potential_risk + 1
30:     goto real_risk
31: real_risk:
32:     if potential_attack = true then
33:         potential_attack ← false
34:         alarm_attack ← true
35:         risk ← risk + potential_risk
36:     if alarm = true or alert = true then
37:         account_fault ← account_fault + 1
38:     else
39:         account_fault ← 0
40:     if alarm_attack = true then alarm ← true
41:     goto alarm_propagation
```

The detector can now report potential risks, following qualitative impact values [111]. In parallel to these values, it triggers alerts to the operator, whenever events are much more likely to be intentional attacks. The reported alerts, settled with appropriate window size values, are assumed to be triggered soon enough, e.g., before reaching the *critical level*, to allow security operators to deal with the information prior taking the necessary countermeasures — i.e., safety of the systems is assumed to have higher priority w.r.t. security.

5.5 Experimental Results for the Watermark-based Detectors

Based on the test scenario described in Section 5.3.3 — remote controller supervising autonomous industrial agents in movements — and using some identification tools to analyze the data collected from the testbed, we obtain the following system matrices (cf. Section 3.3) for each mobile agent:

$$A = \begin{bmatrix} 0.0821 & 0.0551 \\ 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 0.0551 \\ 0.02 \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 0.8386 \end{bmatrix}. \tag{5.1}$$

The co-variance matrices are settled as $Q = 0.2I$ and $R = 0.8I$. The cost matrices are $\Gamma = 2I$ and $\Omega = 3I$. These matrices have been considered to create an LQG controller in order to estimate the physical system actions at each moment. The χ^2 detector is implemented under a given threshold. The threshold is computed taking into account the sensibility of the data coming from the physical

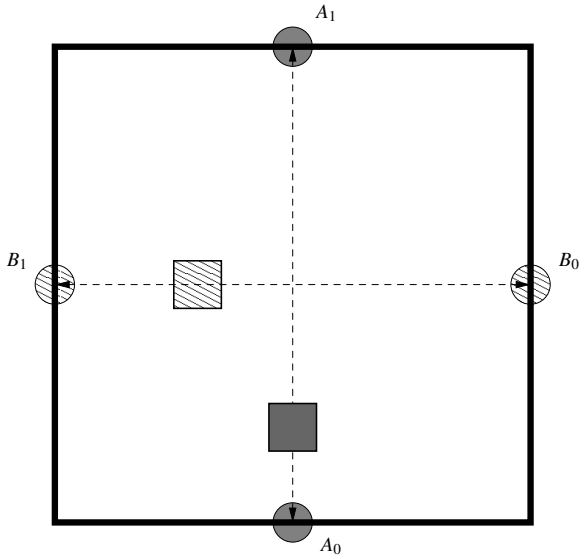
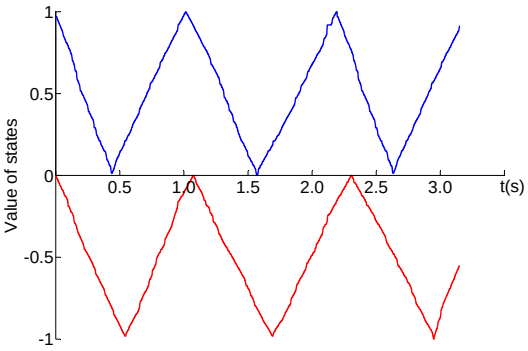


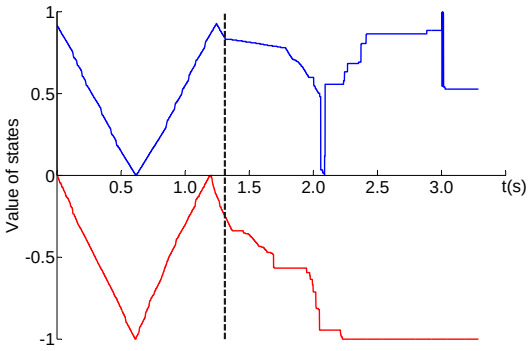
Figure 5.6 – **Cyber-physical industrial scenario implemented in our experimental testbed.** Two mobile agents, represented by solid and pattern gray squares, move from and towards two spatial coordinates, represented by solid and pattern gray circles. Some live demonstration videos of this setup are available at http://j.mp/legoscada.

system, and the number of faults and false positives accepted by the system. The watermark is added to the data sent to control the mobile agents. For the time being, the watermark is known only by the remote controller — the actuators and sensors are seen as passive devices with respect to the watermark-based detector.

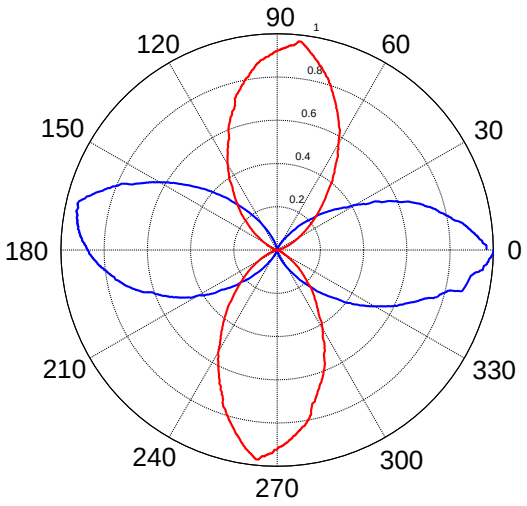
Figure 5.6 shows the cyber-physical industrial scenario used during the experiments. The solid and pattern gray squares represent two mobile agents that move from two spatial coordinates (represented by the the solid and pattern gray circles), and vice versa. The movement dynamics of the two agents is controlled and coordinated by the controller, in order to avoid spatial collisions.



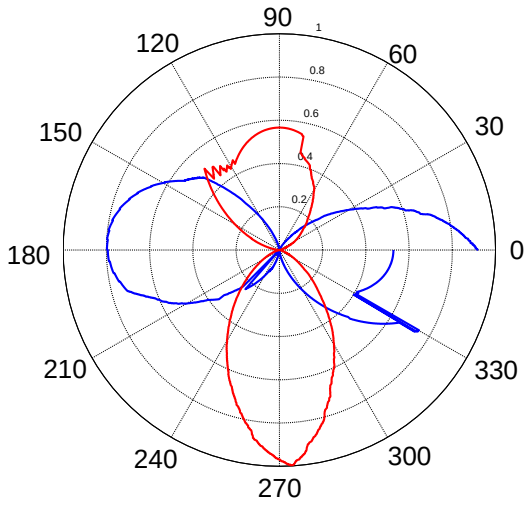
(a) Temporal representation of the dynamics of the agents in the testbed under normal mode.



(b) Temporal representation of the dynamics of the agents in the testbed under attack.



(c) Polar representation of a complete cycle of the dynamics of the agents in the testbed under normal mode.



(d) Polar representation of a complete cycle of the dynamics of the agents in the testbed under attack.

Figure 5.7 – **System dynamics under normal mode vs. attack mode.** The vertical dotted line represents the moment when the attack starts. (a),(c) show the state of the system in proper mode. (b),(d) show the state of the system under attack.

Figure 5.7 shows the dynamics of the system in normal mode (cf. Figures 5.7(a) and (c)) and in attack mode (cf. Figures 5.7(b) and (d)). Figures 5.7(b) and (d) show the moment at which the adversaries take control over the system. We can appreciate how the system moves to unstable states, disrupted by the adversaries. Some live demonstration video captures showing the spatial collision that cause the disruption represented in Figures 5.7(b) and (d) are available at http://j.mp/legoscada. In the sequel, we analyze some experimental results obtained using the aforementioned scenarios, driven by the attacks and adversaries defined in Section 5.4.1.

5.5.1 Experimental Rounds and Data Collection

Following the previous section, we discuss here about the collection of data applying the watermark authentication techniques proposed in Chapter 3. Hereinafter, we denote — due to their physical nature — *stationary watermark* to the single watermark; and *non-stationary watermark* to the multi-watermark. Several repetitions of the experiments are orchestrated using automated scripts handling the elements of some representative scenarios. A set of attacks and detectors are used and posteriorly analyzed. The combinations, attack–detector, are the following:

- *Replay Attack–Watermark Disabled:* In this scenario, the attackers are likely to evade the detector, since no watermark is injected into the system.
- *Replay Attack–Watermark Enabled:* In this scenario, the attackers are likely identified by the detector, since the attack is not able to adapt to the current watermark.
- *Non-parametric Attack–Stationary Watermark:* Using this scenario, the attackers and the detector have equal chances of success.
- *Non-parametric Attack–Non-stationary Watermark:* Using this scenario, the non-stationary watermark changes the distribution systematically, hence preventing the attack to adapt to such changes. The expected results are an increase of the detection ratio.
- *Parametric Attack–Stationary Watermark:* In this scenario, the attackers are likely to evade the detector when the attack properly infers the system parameters.
- *Parametric Attack–Non-stationary Watermark:* The attacker are also likely to evade the detector when the system parameters are properly identified.

The cyber-physical implications of the testbed hinder the experimentation process especially when several repetitions are required in order to obtain statistical results, contrary to simulations where only the code is executed. The creation of the orchestration script, which automates the test, is necessary to simplify the experimentation tasks. The sequel presents the results using the testbed for the aforementioned attacker-detector combinations.

5.5.2 Data Analysis

After collecting data from different devices across the testbed, the information is analyzed accordingly to interpret the performance of the detector with regard to the attack scenario. Since the stationary and non-stationary watermark detectors were correctly refined for each test scenario, we are able to analyze in depth the results through a statistical evaluation of the data. Figure 5.8 shows the detector values, g_t, for all the attack-detector combinations defined before.

For all the plots, the solid horizontal line represents the threshold; and the vertical dotted line represents the moment when the attacker starts injecting malicious data. The short peaks on the left side of the plots, those bypassing the threshold line before the start of the attacks, are counted as false positives or system faults.

Figures 5.8(a) and 5.8(b) show the experimental results of the replay attack. When the watermark is disabled (cf. Figure 5.8(a)), the attacker properly evades the detector. Since the controller is not inserting the protection watermark, it does not detect the attack. On the contrary, the results in Figure 5.8(b) show that the activation of the watermark under the same scenario allows the controller to alert about the attack almost immediately. Based on these results, we can conclude that the stationary watermark based detector properly works out to detect the replay attack.

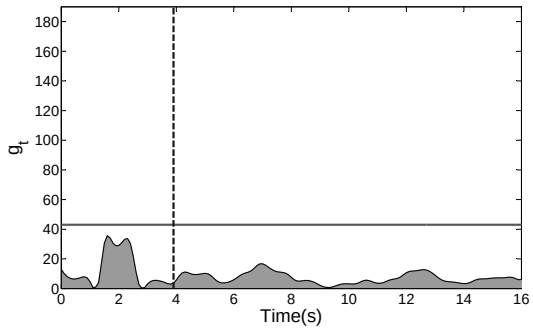
Figure 5.8(c) represents the non-parametric attacker against the previously tested stationary watermark. The detector is now unable to detect the attacker. Figure 5.8(d) shows the case where the non-stationary watermark is enabled. Under this situation, the detector has lightly more chances of detecting the attack. This shows how the non-stationary watermark mechanism does improve the detection abilities compared to the stationary watermark approach.

Figures 5.8(e) and 5.8(f) evaluate the scenario associated to the parametric attacks. Theoretically, the attacker is expected to evade the detector when the attack succeeds at properly identifying the parameters of the system dynamics. Figure 5.8(e) represents the experiments where the parametric attack is executed under the stationary watermark scenario. The figure shows that the detector value, g_t, remains most of the time below the detection threshold. Figure 5.8(f) shows the behavior of the detector under the non-stationary watermark scenario. This time, the detector has slightly more chances of detecting the attack.

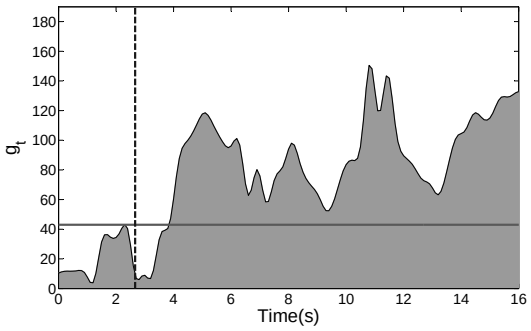
5.5.3 Statistical Data Evaluation

Using the watermark-based detection mechanism, we run for each attack scenario 75 automated rounds (about four hours of data collection processing). In order to evaluate the results, we use the following metrics:

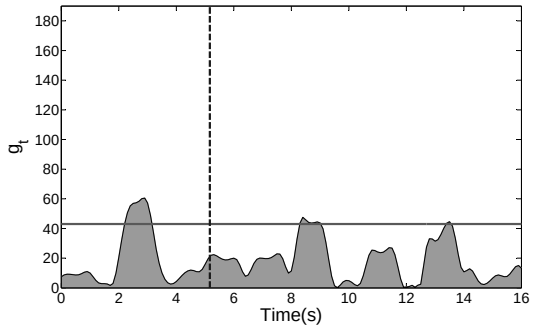
1. *Detection Ratio*, associated to the success percentage of the detector, calculated with regard to the time range after each attack starts.



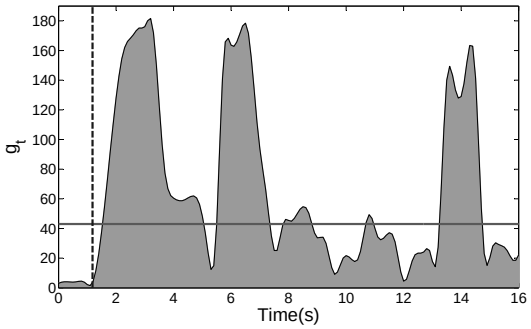
(a) No watermark under replay attack.



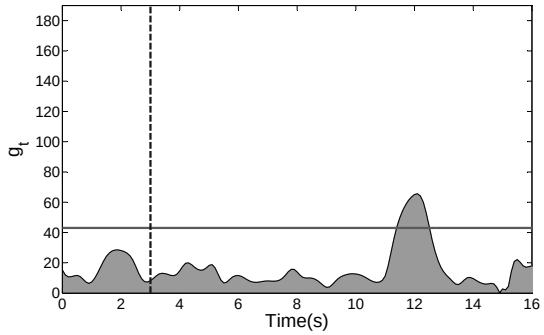
(b) Stationary watermark under replay attack.



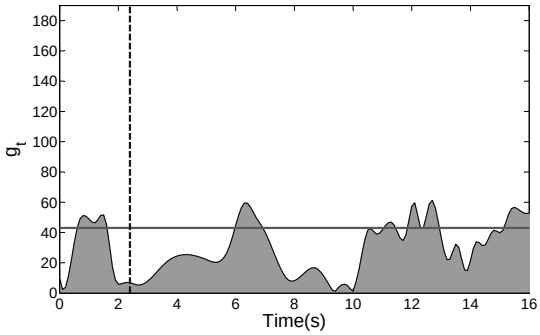
(c) Stationary watermark under non-parametric attack.



(d) Non-stationary watermark under non-parametric attack.



(e) Stationary watermark under parametric attack.



(f) Non-stationary watermark under parametric attack.

Figure 5.8 – **Experimental testbed results.** The horizontal solid line represents the threshold. The vertical dotted line represents the moment when the attack starts. Peaks on the left side of the vertical dotted line represent false positives. (a),(b) detection values of g_t, without and with stationary watermark under replay attack. (c),(d) detection values with stationary and non-stationary watermark under non-parametric attack. And (e),(f) detection values with stationary and non-stationary watermark under parametric attack.

2. *Average Detection Time*, determining the amount of time needed by the detector to trigger the attack alert.

3. *False Negative (FN)* ratio, determining the number of samples where the detector fails at successfully alerting about the attacks. The ratio is calculated as follows.

$$FN = \frac{SA - AD}{SA} \tag{5.2}$$

where SA represents the values of the samples under attack, and AD the samples detected as an attack.

4. *False Positives (FP)* ratio, calculated as the number of samples where the detector signals benign events as attacks. The ratio is calculated as follows.

$$FP = \frac{AD}{SN} \tag{5.3}$$

where SN represents the number of samples under normal operation, and AD the number of samples detected as attack by mistake.

Stationary Watermark-based Detector

Table 5.1 shows the performance results of the stationary watermark-based detector, based on the *Detection Ratio* and the *Average detection Time* metrics.

Regarding the results shown in Table 5.1, we can emphasize that the replay attack is the most detectable scenario, with a detection ratio of about 40%. The non-parametric attacker has a lower detection ratio, of about 18%. This result is expected, as suggested by the theoretical and simulation-based conclusions presented in Chapter 3. The parametric attack has the most robust system identification approach. The attacks can evade the detection process if they succeed at properly identifying the system attributes. In terms of results, they lead to the lowest detection rate of about 12%.

During the replay attack, the *Average Detection Time* is the slowest of all the adversarial scenarios. This behavior is due to the watermark distribution properties (cf. Section 3.4). At the same time, the injection attacks (either the parametric or the non-parametric version) are detected much faster than the replay attack. This is due to the transition period needed by the attackers to estimate the correct data prior misleading the detector. For this reason, if the attacker does not choose the precise moment to start the attack, the detector implemented at the controller side is able to detect the injected data, right at the beginning of the attack. Furthermore, the attackers shall also synchronize their estimations to the measurements sent by the sensors. In case the synchronization process fails, the detector identifies the uncorrelated data and reports the attack.

Table 5.2 shows that the detection of the replay attack has the lowest false negative ratio, 64.06%, hence confirming that this adversarial scenario is the most detectable situation with regard to the detection techniques reported in [9, 53]. The detection of the non-parametric attacks has a higher false negative ratio, 85.20%, confirming the theoretical and simulation-based results

	Replay Attack	**Non-parametric Attack**	**Parametric Attack**
Detection Ratio	40.00%	18.00%	12.00%
Average Detection Time	10.01s	4.89s	6.08s

Table 5.1 – Detector performance results using a stationary watermark.

	Replay Attack	**Non-parametric Attack**	**Parametric Attack**
False Negatives	64.06%	85.20%	88.63%
False Positives	0.98%	1.66%	1.35%

Table 5.2 – Experimental results using a stationary watermark.

reported in Section 3. The detection of the parametric attacks also confirms the results obtained via numeric simulations, leading to the highest false negative ratio (about 88.63%). In terms of false positive ratio, the three adversarial scenarios show a very low impact of our detection approach (on average, about 1.33% false positive ratio). With regard to the ratio of false positives, notice that we use a modified fault detector to detect attacks. Hence, the percentage of false positives is closely bounded to the sensibility of the detector, i.e., the number of false positives shall be seen as an intrinsic property of the detector.

Non-Stationary Watermark-based Detector

Table 5.3 shows the performance results of the non-stationary watermark-based detector, based on the *Detection Ratio* and the *Average detection Time* metrics. Regarding the results shown in the table, we can verify that the performance obtained with the detection strategy based on the non-stationary watermark is consistent to the results obtained from the numerical validation in Chapter 3. We show that the replay attack and the non-parametric attackers have a higher detection ratio with this strategy, of about 60% and 56% respectively. Then, the parametric

	Replay Attack	**Non-parametric Attack**	**Parametric Attack**
Detection Ratio	60.00%	56.00%	16.00%
Average Detection Time	9.26s	6.27s	5.63s

Table 5.3 – Detector performance results using a non-stationary watermark.

	Replay Attack	**Non-parametric Attack**	**Parametric Attack**
False Negatives	62.03%	54.24%	84.61%
False Positives	5.10%	3.30%	4.63%

Table 5.4 – Experimental results using a non-stationary watermark.

attackers have a little increase in the detection ratio, from 12% to 16%. It is worth noting that the *Average Detection Time* decrease respects to the stationary watermark-based detector.

Table 5.4 shows for the non-stationary watermark-based detector, that the number of false negatives decrease, increasing the detection accuracy of this strategy against these adversaries. Otherwise the false positives with this strategy increase with respect to the stationary watermark-based detection. That means, in the real-world testbed, the loss of the system performance increase, since with the same sensibility that the previous strategy, and with a non-stationary watermark computed as detailed in Section 3.7.3 the number of false positives increase from 1.35% to 4.63%.

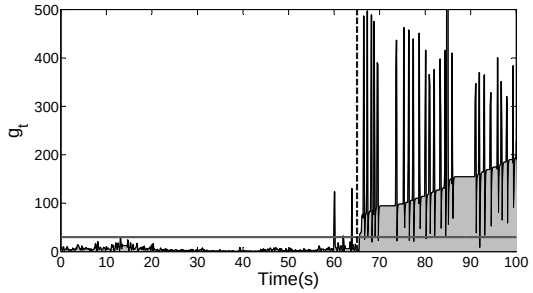
5.6 Experimental Results for the PIETC-WD Strategy

Based on the same scenario used in the previous section, we create a second testbed for validating the PIETC-WD strategy presented in Chapter 4. We have considered the same communication protocols and the same physical scenario. The control and security strategy has been modified as follows:

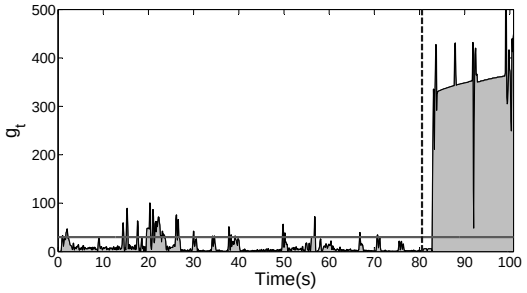
- *Control strategy implementation*: to implement the decentralized strategy proposed in Chapter 4, we create a distributed control system where local controllers at the system sensors have the capability to compute the estimation of the dynamics of the system. The remote controller can create an *LQG controller* using the inputs and the outputs of the system. Otherwise, sensors have only the data received from the physical system to estimate the next behaviour of the system and to compute the *LQG Controller*. Actuators in this strategy work, as we have described in the previous testbed, as passive devices — from either the control and security standpoint.
- *Watermark-based detector implementation*: the watermark detector is also distributed in this strategy. There are two different watermarks. First, a watermark managed by each sensor which allows controlling the dynamic of the system. Each sensor knows its own watermark, independent of the other watermarks. These watermarks are added to the *residues* sent to the remote controller periodically, after the system dynamics verification. Otherwise, if the residues generated by the local controllers located at the mobile agent sensors generate a detector value over the threshold, the local controllers send to the remote controller the real distance without watermarks. In addition, the remote controller holds its own watermark, which is stochastically added to the measurements (e.g., speed measurements) and sent to the controllers located at each mobile agent, to verify the closed-loop.

Figure 5.9 shows two different scenarios. In a first scenario, the system uses only a periodic communication policy using the watermark detector placed at the local controllers (cf. Figure 5.9(a) and 5.9(c)). When the attacker successfully infers the system parameters, the attack is not reported

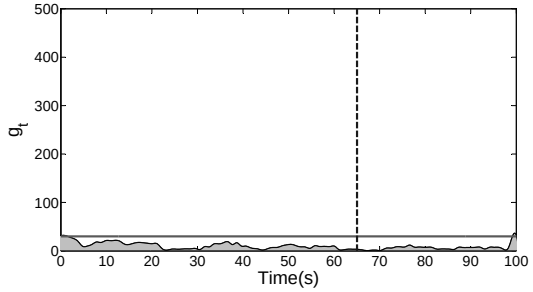
by the remote controller. Nevertheless, the local alarms, placed at the sensor controllers, report the attack. In the second scenario, the system uses also the intermittent policy. In such a case, the attack is detected by the remote controller (cf. Figure 5.9(b) and 5.9(d)). In Figure 5.9(b), we may observe some peaks at time $t = [2.1s; 14.4s; 15.3s; 17.6s; 20.4s; 21.0s; 22.3s; 26.2s; 37.8s; 56.8s; 66.8s]$. These peaks represent the local controllers' reaction under the presence of the watermarks sent by the remote controller — e.g., used to verifying that the closed-loop works properly, as defined in Section 4.4.3.



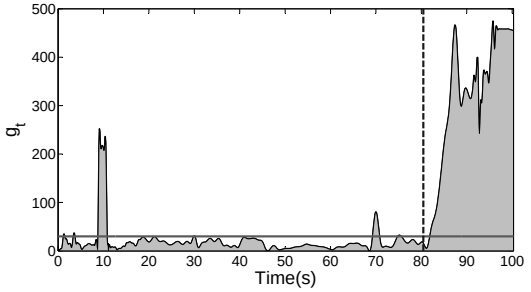
(a) Sensor detector results not using intermittent policy.



(b) Sensor detector results using intermittent policies.



(c) Control center detector results not using intermittent policy.



(d) Control center detector results using intermittent policy.

Figure 5.9 – **Experimental testbed results.** The horizontal solid line represents the threshold. The vertical dotted line represents the moment when the attack starts. Peaks on the left side of the vertical dotted line represent false positives. (a),(b) represent local detector values g_t (placed within the sensors) without and with intermittent policies (i.e., remote watermarks) under a parametric attack. (c),(d) represent remote detection values (placed at the control center) without and with intermittent policies under a parametric attack.

To compare the watermark-based detector mechanism proposed in Chapter 3 and the PIETC-WD strategy proposed in Chapter 4, against parametric cyber-physical adversaries, several repetitions of the experiment were orchestrated. Using the results obtained from the testbed experiments, we report in Table 5.5 the percentage of false positives, false negatives, and detection ratio. We also show the average time that the detector placed at the (remote) control center, takes to launch the alerts.

	Using only the watermark-based detector	**Using as well the PIETC-WD detector**
Detection Ratio	12.00%	75.25%
Average Detection Time	6.08s	6.20s
False Negatives	88.60%	38.66%
False Positives	1.35%	5.23%

Table 5.5 – **Detection performance results using the PIETC-WD detection strategy.**

Regarding the results shown in Table 5.5, we can emphasize that: (1) a system which uses only the watermark-based detector mechanism against parametric attackers has a lower detection ratio, of about 12%. That is possible since the attackers can evade the detection process if they succeed at properly identifying the system attributes. This result is expected, as suggested by the theoretical and simulation-based conclusions presented in Chapter 3; and (2) a system which uses the strategy proposed in Chapter 4 has a higher detection ratio, of about 75.25%. In this scenario, the detection ratio increases, confirming the theoretical and simulation results reported in Chapter 4. The false negative ratio decreases, from 88.60% to 38.66%. In terms of false positives, both scenarios show similar results, but the PIETC-WD strategy generates about 3.9% more. The time between the beginning of the attack and the moment when the attack is detected by the remote controller, takes longer with the PIETC-WD strategy, since the detection watermark managed by the remote controller follows a stochastic law. Therefore, we confirm that the PIETC-WD strategy increases the detection performance, at the cost of increasing the detection time.

5.7 Summary

This chapter has presented a cyber-physical training platform to test defense techniques. The architecture of the testbed is based on real-world components, in order to emulate cyber-physical systems commanded by SCADA (Supervisory Control And Data Acquisition) technologies. Two real-world protocol implementations (DNP3 and Modbus) are included within the platform. Some adversarial scenarios were also integrated in our testbed. These scenarios enforce different types of attackers, incrementing the usability of the testbed to experiment novel security methods against a wider variety of malicious intents. We have presented results based on three main adversarial scenarios. The scenarios were confronted against the defense techniques defined in Chapters 3 and 4. Experimental results confirm the previous theoretical and simulation-based work provided in previous chapters.

6 Conclusion and Future Work

The thesis of this dissertation is that, in a cyber-physical system, adversaries can eavesdrop and manipulate information in order to disrupt availability and integrity properties of the system. Imagine, for instance, energy distribution infrastructures, with adversaries manipulating both cyber and physical layers. Cyber-physical attacks, initiated right after the infection of ICT resources of their associated corporate networks, may affect a wider diversity of users and processes, likely targeting and getting control over critical system processes. Adversaries may use cross-layer techniques, first to get control over the network layers, then to disrupt physical field devices. The combination of such techniques may generate stealthy attacks, in order to evade detection — the final goal being the disruption of, e.g., national critical infrastructures. Attacks against these systems may definitely affect people and physical environments.

In terms of contributions, we have started this dissertation by surveying existing technologies underlying cyber-physical environments. More specifically, we have surveyed control-feedback theories in terms of importance and complexity, as well as described some of the characteristics and properties of industrial SCADA (Supervisory Control and Data Acquisition) protocols and networked control systems. We have listed detection and mitigation techniques to protect the resulting cyber-physical systems. Our focus has been the presentation of theories and techniques from a traditional ICT security standpoint.

The state-of-the art and related work has been complemented by three main contributions, properly disseminated to relevant media in the field. First of all, a first contribution has consisted on revisiting stationary watermark-based protection approaches, transforming them towards an adaptive process able to covering a much wider number of adversarial models. Second, we have extended the resulting watermark-based detector, used as a physical attestation at the cyber layer, by adding a decentralized strategy to scale the approach to multiple elements of a cyber-physical environments (not only controllers, but also sensors and actuators). The idea is to distribute the detection process across all those elements with enough capabilities to identify and handle system dynamics, in order to counter malicious actions, in addition to faults and errors. Third, we have validated all our findings by integrating them on an experimental training SCADA testbed.

The testbed, implemented by using real-world SCADA protocols (e.g., Modbus and DNP3) and training linux-based embedded devices, has allowed us to test and validate the security performance of our proposals. In addition, several adversaries able to attack representative scenarios were provided to complement the numeric simulations reported in previous versions of the work.

In terms of perspectives for future research (as a result of the work initiated in this dissertation), several actions remain to be done. It is worth noting that cyber-physical systems have a vast number of challenges to be addressed. This dissertation has handled, with a limited scope, some of the protection challenges in the topic, paying special attention to detection of malicious actions hidden or combined with faults and accidents. Nevertheless, cyber-physical systems encompass many other fields that have to be handled together in order to improve their resilience to attacks and misuse.

With this in mind, a first perspective would include a more thorough analysis of the performance impact of our decentralized protection process. Indeed, the performance of cyber-physical systems is an important issue that is necessary to handle. In this dissertation, we have revisited and expanded some watermark-based approaches to identify and alert about cyber-physical attacks. We expanded the original design based on a centralized watermark-based detection process, in two extended approaches that increase the detection efficiency with regard to the original designs, while providing an equivalent impact in terms of performance and communication overhead.

However, an issue not properly handled by our work is the analysis of the performance loss when using the decentralized version of the approach, presented as a combination of control-protection strategies. Indeed, our decentralized process moves detection from local to remote controllers, increasing the number of adversarial models detected by the protection system. Further research remains to be conducted in order to analyze, in addition to the security level of the new approach, the impact of the novel construction in terms of performance and overhead. Likewise, novel research in order to further decentralize the control-protection strategy initiated in this dissertation, as well as a proper combination of the cyber and control-physical layers suggested in our work, could be expanded towards next-generation cyber-physical of SIEM (Security Information and Event Management) detectors, to properly correlate cross-layer security incidents.

Bibliography

[1] Yu Zhang, Fei Xie, Yunwei Dong, Gang Yang, and Xingshe Zhou. High Fidelity Virtualization of Cyber-physical Systems. *International Journal of Modeling, Simulation, and Scientific Computing*, 4(2), 2013.

[2] Nicolas Falliere, Liam O Murchu, and Eric Chien. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, 5:6, 2011.

[3] David Corman, Victoria Pillitteri, Scott Tousley, Mark Tehranipoor, and Ulf Lindqvist. NITRD Cyber-Physical Security Panel. 35th IEEE Symposium on Security and Privacy, IEEE SP 2014, San Jose, CA, USA, May 18-21.

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

[5] Alvaro A. Cardenas, Saurabh Amin, Bruno Sinopoli, Annarita Giani, Adrian Perrig, and Shankar Sastry. Challenges for securing cyber physical systems. In *Workshop on Future Directions in Cyber-Physical Systems Security*, page 7. DHS, July 2009.

[6] Simon Brown. Functional safety of electrical/electronic/programmable electronic safety related systems. *Computing & Control Engineering Journal*, 11(11):14, February 2000.

[7] Mark Clayton. Cybersecurity: How US utilities passed up chance to protect their networks, May 2012, http://www.csmonitor.com/USA/2012/0517/Cybersecurity-How-US-utilities-passed-up-chance-to-protect-their-networks, Last access: October 2016.

[8] Johan Åkerberg and Mats Björkman. Exploring Network Security in PROFIsafe. In *Computer Safety, Reliability, and Security: 28th International Conference, SAFECOMP 2009, Hamburg, Germany, September 15-18, 2009. Proceedings*, pages 67–80, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

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

[10] Thomas Roth and Bruce McMillin. Physical Attestation in the Smart Grid for Distributed State Verification. *IEEE Transactions on Dependable and Secure Computing*, PP(99), 2016.

[11] Pedro Barbosa, Andrey Brito, Hyggo Almeida, and Sebastian Clauß. Lightweight Privacy for Smart Metering Data by Adding Noise. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, pages 531–538, New York, NY, USA, 2014. ACM.

[12] Ata Arvani and Vittal 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.

[13] Van Long Do, Lionel Fillatre, and Igor Nikiforov. A statistical method for detecting cyber/-physical attacks on SCADA systems. In *2014 IEEE Conference on Control Applications (CCA)*, pages 364–369, Juan Les Antibes, France, Oct 2014.

[14] Béla Genge, István Kiss, and Piroska Haller. A system dynamics approach for assessing the impact of cyber attacks on critical infrastructures. *International Journal of Critical Infrastructure Protection*, 10:3–17, 2015.

[15] Yilin Mo, Tiffany Hyun-Jin Kim, Kenneth Brancik, Dona Dickinson, Heejo Lee, Adrian Perrig, and Bruno Sinopoli. Cyber-Physical Security of a Smart Grid Infrastructure. *Proceedings of the IEEE*, 100(1):195–209, Jan 2012.

[16] Rohan Chabukswar. *Secure Detection in Cyberphysical Control Systems*. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, May 2014.

[17] Fabio Pasqualetti. *Secure control systems: A control-theoretic approach to cyber-physical security*. PhD thesis, Department of Mechanical Engineering, University of California, Santa Barbara, September 2012.

[18] Quanyan Zhu and Tamer Başar. *A hierarchical security architecture for smart grid*, pages 413–438. Cambridge University Press, Cambridge, May 2012. Cambridge Books Online.

[19] American Gas Association. Cryptographic Protection of SCADA Communications. Technical Report 12, 2006, https://www.scadahacker.com/library/Documents/Standards/AGA%20-%20Cryptographic%20Protection%20of%20SCADA%20Communications%20-%2012%20Part1.pdf, Last access: October 2016.

[20] Andrew K. Wright, John A. Kinast, and Joe McCarty. *Low-Latency Cryptographic Protection for SCADA Communications*, pages 263–277. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[21] PROFIBUS and PROFINET International. International Standard, PROFINET Security Guideline, 2013, http://www.profibus.com/download/specifications-standards/, Last access: October 2016.

[22] Automatak, 2013, http://www.automatak.com/robus/, Last access: Octobre 2014.

[23] Frances Cleveland. IEC 62351 Security Standards for the Power System Information Infrastructure. Technical report, 2016.

[24] IMS Research. The World Market for Industrial Ethernet - 2009 Edition. Technical report, 2009.

[25] John Rinaldi. ETHERNET/IP OVERVIEW, 2014, http://www.rtaautomation.com/technologies/ethernetip/, Last access: Octobre 2016.

[26] International Electrotechnical Commission. Industrial communication networks - Fieldbus specifications - Part 6-2: Application layer protocol specification - Type 2 elements, 2014, https://webstore.iec.ch/publication/4695, Last access: Octobre 2016.

[27] Eric D Knapp. *Industrial Network Security: Securing critical infrastructure networks for smart grid, SCADA, and other Industrial Control Systems*. Syngress Publishing, Boston, MA, USA, 1st edition, 8 2011.

[28] EPSG. openSAFETY over EtherNet/IP Version 0.0.1. Technical report, 2010.

[29] IMS Research. CIP Safety: Safety networking for today and beyond. Technical report, https://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00110R1_CIP_Safety_White_Paper.pdf, Last access: April 2017.

[30] Brian Batke, Joakim Wiberg, and Dennis Dubé. CIP Security Phase 1: Secure Transport for EtherNet/IP. In *ODVA 2015 Industry Conference & 17th Annual Meeting October 13-15, Frisco, Texas, USA*, page 15, 2015, https://www.odva.org/Portals/0/Library/Conference/2015_ODVA_Conference_Batke-Wiberg-Dube_CIP-Security-Phase-1.pdf, Last access: April 2017.

[31] Ethernet POWERLINK Standardisation Group. Ethernet POWERLINK. Communication Profile Specification (EPSG DS 301 V1.2.0). Technical report, 2013.

[32] Jonathan Yung, Hervé Debar, and Louis Granboulan. Security Issues and Mitigation in Ethernet POWERLINK. In *2nd Workshop On The Security Of Industrial Control System & Cyber-Physical Systems (CyberICPS 2016). Heraklion, Greece*, September 2016.

[33] Alvaro A. Cárdenas, Saurabh Amin, Zong-Syun Lin, Yu-Lun Huang, Chi-Yen Huang, and Shankar 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.

[34] György Dán and Henrik 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.

[35] Yao Liu, Peng Ning, and Michael 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.

[36] André Teixeira, Daniel Pérez, Henrik Sandberg, and Karl Henrik 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.

[37] Yilin Mo, Rohan Chabukswar, and Bruno Sinopoli. Detecting integrity attacks on SCADA systems. *IEEE Transactions on Control Systems Technology*, 22(4):1396–1407, July 2014.

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

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

[40] Yuan Yuan, Quanyan Zhu, Fuchun Sun, Qinyi Wang, and Tamer 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.

[41] André Teixeira, Iman Shames, Henrik Sandberg, and Karl Henrik Johansson. A secure control framework for resource-limited adversaries. *Automatica*, 51:135–148, 2015.

[42] Yuan Chen, Soummya Kar, and Jose 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.

[43] Fabio Pasqualetti, Florian Dorfler, and Francesco 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.

[44] Yilin Mo, Emanuele Garone, Alessandro Casavola, and Bruno Sinopoli. False data injection attacks against state estimation in wireless sensor networks. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5967–5972, Atlanta, GA, USA, Dec 2010.

[45] Wei Gao and Thomas 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.

[46] Rakesh B Bobba, Katherine M Rogers Qiyan Wang, Himanshu Khurana, Klara Nahtstedt, and Thomas 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, Stockholm, Switzerland, April 2010. Citeseer.

[47] Yao Liu, Peng Ning, and Michael 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.

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

[49] Wei Gao, Thomas Morris, Bradley Reaves, and Drew Richey. On SCADA control system command and response injection and intrusion detection. In *2010 eCrime Researchers Summit*, pages 1–9, Oct 2010.

[50] Thomas H. Morris and Wei Gao. Industrial control system cyber attacks.

[51] Inseok Hwang, Sungwan Kim, Youdan Kim, and Chze Eng Seah. A Survey of Fault Detection, Isolation, and Reconfiguration Methods. *IEEE Transactions on Control Systems Technology*, 18(3):636–653, May 2010.

[52] Ralph Langner. Stuxnet: Dissecting a cyberwarfare weapon. *Security & Privacy, IEEE*, 9(3):49–51, 2011.

[53] Yilin Mo and Bruno Sinopoli. Secure control against replay attacks. In *Communication, Control, and Computing. 47th Annual Allerton Conference on*, pages 911–918, Monticello, IL, USA, Sept 2009. IEEE.

[54] Fer Miao, Miroslav Pajic, and George J. Pappas. Stochastic game approach for replay attack detection. In *52nd IEEE Conference on Decision and Control*, pages 1854–1859, Florence, Italy, Dec 2013.

[55] Quanyan Zhu and Tamer Başar. Dynamic policy-based IDS configuration. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 8600–8605, Shanghai, China, Dec 2009.

[56] Andrey Y. Lokhov, Nathan Lemons, Thomas C. McAndrew, Aric Hagberg, and Scott 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.

[57] Yong Wang, Zhaoyan Xu, Jialong Zhang, Lei Xu, Haopei Wang, and Guofei Gu. SRID: State Relation Based Intrusion Detection for False Data Injection Attacks in SCADA. In Mirosław Kutyłowski and Jaideep 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, Cham, 2014. Springer International Publishing.

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

[59] Po-Yu Chen, Shusen Yang, and Julie A. McCann. Distributed Real-Time Anomaly Detection in Networked Industrial Sensing Systems. *IEEE Transactions on Industrial Electronics*, 62(6):3832–3842, June 2015.

[60] Quanyan Zhu. *Game-theoretic methods for security and resilience in cyber-physical systems*. PhD thesis, University of Illinois at Urbana-Champaign, 2013.

[61] Quanyan Zhu and Tamer 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.

[62] David I Urbina, Jairo Giraldo, Alvaro A. Cardenas, Junia Valente, Mustafa Faisal, Nils Ole Tippenhauer, Justin Ruths, Richard Candell, and Henrik 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.

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

[64] 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.

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

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

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

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

[69] Tae Hoon Lee, Won Sang Ra, Seung Hee Jin, Tae Sung Yoon, and Jin Bae 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.

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

[71] Cheng Ching Yu. *Autotuning of PID Controllers*. Springer London, London, 2006.

[72] Antonio Barrientos, Iñaki 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.

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

[74] Ke-You You and Li-Hua Xie. Survey of Recent Progress in Networked Control Systems. *Acta Automatica Sinica*, 39(2):101 – 117, 2013.

[75] Jos'e Salt, Vicente Casanova, A Cuenca, and R Pizá. Sistemas de Control Basados en Red Modelado y Diseño de Estructuras de Control. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 5(3):5–20, 2008.

[76] W.P.M.H. Heemels and M.C.F. Donkers. Model-based periodic event-triggered control for linear systems. *Automatica*, 49(3):698 – 711, 2013.

[77] WPMH Heemels, MCF Donkers, and Andrew R Teel. Periodic Event-Triggered Control for Linear Systems. *IEEE Transactions on Automatic Control*, 58(4):847–861, April 2013.

[78] Duo Han, Yilin Mo, Junfeng Wu, Sean Weerakkody, Bruno Sinopoli, and Ling Shi. Stochastic Event-Triggered Sensor Schedule for Remote State Estimation. *IEEE Transactions on Automatic Control*, 60(10):2661–2675, Oct 2015.

[79] R.G.K.M. Aarts. System identification and parameter estimation. Technical report, Faculty of Engineering Technology, University Twente, 2012.

[80] Yucai Zhu. New development in industrial MPC identification. In *Proceedings of the International Symposium on Advanced Control of Chemical Processes (ADChEM)*. Hong Kong, China, January 2003.

[81] Torsten Soderstrom and Petre Stoica. *System identification* . New York : Prentice Hall, 1989. Includes indexes.

[82] Yucai Zhu, Firmin Butoyi, and Dow Benelux NV. Multivariable and closed-loop identification for model predictive control. Technical report, 2000.

[83] Yahya Chetouani. Using arx approach for modelling and prediction of the dynamics of a reactor-exchanger. In *INSTITUTION OF CHEMICAL ENGINEERS SYMPOSIUM SERIES*, volume 154, page 297. Institution of Chemical Engineers; 1999, 2008.

[84] Rohan Chabukswar, Bruno Sinopoli, Gabor Karsai, Annarita Giani, Himanshu Neema, and Andrew Davis. Simulation of Network Attacks on SCADA Systems. In *First Workshop on Secure Control Systems, Cyber Physical Systems Week*, April 2010.

[85] James H. Graham and Sandip C. Patel. Security Considerations in SCADA Communication Protocols. Technical Report TR-ISRL-04-01, 2004, http://www.cs.louisville.edu/facilities/ISLab/tech%20papers/ISRL-04-01.pdf, Last access: October 2016.

[86] Aung 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: October 2016.

[87] Christos Siaterlis, Béla Genge, and Marc Hohenadel. EPIC: a testbed for scientifically rigorous cyber-physical security experimentation. *IEEE Transactions on Emerging Topics in Computing*, 1(2):319–330, Dec 2013.

[88] Benjamin Green, David Hutchison, Sylvain Andre Francis Frey, and Awais Rashid. Testbed diversity as a fundamental principle for effective ICS security research. In *Proceedings of the First International Workshop on Security and Resilience of Cyber-Physical Infrastructures (SERECIN)*. Lancaster University, Technical Report SCC-2016-01, pp. 12–15, 2016.

[89] Tim Yardley. Testbed cross-cutting research, 2014, https://tcipg.org/research/testbed-cross-cutting-research, Last access: October 2016.

[90] Antonio Sánchez Aragó, Enrique Redondo Martínez, and Sandra Salán Clares. SCADA Laboratory and Test-bed As a Service for Critical Infrastructure Protection. In *Proceedings of the 2Nd International Symposium on ICS & SCADA Cyber Security Research 2014*, ICS-CSR 2014, pages 25–29, UK, 2014. BCS.

[91] Marina Krotofil and Jason Larsen. Rocking the pocket book: Hacking chemical plants for competition and extortion. *DEF CON*, 23, 2015.

[92] Yu Zhang, Fei Xie, Yunwei Dong, Gang Yang, and Xingshe Zhou. High Fidelity Virtualization of Cyber-Physical Systems. *International Journal of Modeling, Simulation, and Scientific Computing*, 4(2):1–26, June 2013.

[93] Fabrice 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.

[94] Richard Candell, Keith Stouffer, and Dhananjay Anand. A cybersecurity testbed for industrial control systems. In *Process Control and Safety Symposium, International Society of Automation*, Houston, TX, 2014.

[95] Stephen McLaughlin, Charalambos Konstantinou, Xueyang Wang, Lucas Davi, Ahmad-Reza Sadeghi, Michail Maniatakos, and Ramesh Karri. *The Cybersecurity Landscape in Industrial Control Systems*, volume 104, pages 1039–1057. May 2016.

[96] Georgia Koutsandria, Reinhard Gentz, Mahdi Jamei, Anna Scaglione, Sean Peisert, and Chuck 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.

[97] Hannes Holm, Martin Karresand, Arne Vidström, and Erik Westring. A Survey of Industrial Control System Testbeds. In Sonja Buchegger and Mads Dam, editors, *Secure IT Systems: 20th Nordic Conference, NordSec 2015, Stockholm, Sweden, October 19–21, 2015, Proceedings*, pages 11–26, Cham, 2015. Springer International Publishing.

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

[99] 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.

[100] Shikha Tripathi and Mohammad Asif Ikbal. Step Size Optimization of LMS Algorithm Using Aunt Colony Optimization & Its comparison with Particle Swarm optimization Algorithm in System Identification. *International Research Journal of Engineering and Technology (IRJET)*, 2:599–605, October 2015.

[101] Simon S Haykin. *Adaptive filter theory*. Pearson Education India, 5 edition, 2014.

[102] Bernard Widrow, John M McCool, Michael G Larimore, and C Richard Johnson Jr. Stationary and nonstationary learning characteristics of the LMS adaptive filter. *Proceedings of the IEEE*, 64(8):1151–1162, Aug 1976.

[103] Rohan Chabukswar, Yilin Mo, and Bruno Sinopoli. Detecting Integrity Attacks on SCADA Systems. *IFAC Proceedings Volumes*, 44(1):11239 – 11244, 2011.

[104] Leo H. Chiang, Evan L. Russell, and Richard D. Braatz. Tennessee eastman process, 2001.

[105] Stephan Weyer, Mathias Schmitt, Moritz Ohmer, and Dominic Gorecky. Towards Industry 4.0 - Standardization as the crucial challenge for highly modular, multi-vendor production systems. *IFAC-PapersOnLine*, 48(3):579 – 584, 2015.

[106] Jay Lee, Behrad Bagheri, and Hung-An Kao. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3:18 – 23, 2015.

[107] Radhakisan Baheti and Helen Gill. Cyber-physical systems. *The impact of control technology*, 12:161–166, 2011.

[108] Bharadwaj Satchidanandan and Panganamala R Kumar. Dynamic Watermarking: Active Defense of Networked Cyber-Physical Systems. *Proceedings of the IEEE*, 105(2):219–240, Feb 2017.

[109] Dina Hadžiosmanović, Robin Sommer, Emmanuele Zambon, and Pieter H. Hartel. Through the Eye of the PLC: Semantic Security Monitoring for Industrial Processes. In *Proceedings of the 30th Annual Computer Security Applications Conference*, ACSAC '14, pages 126–135, New York, NY, USA, 2014. ACM.

[110] David I. Urbina, Jairo A. Giraldo, Alvaro A. Cardenas, Nils Ole Tippenhauer, Junia Valente, Mustafa Faisal, Justin Ruths, Richard Candell, and Henrik Sandberg. Limiting the Impact of Stealthy Attacks on Industrial Control Systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 1092–1105, New York, NY, USA, 2016. ACM.

[111] Group REI-cyber. La Cybersécurité des Réseaux Electriques Intelligents. White book. La Revue de l'Electricité et de l'Electronique (REE), February 2016.

[112] Modbus Organization. Official Modbus Specifications, 2016, http://www.modbus.org/specs.php, Last access: October 2016.

[113] Ken Curtis. A DNP3 Protocol Primer. A basic technical overview of the protocol, 2005, http://www.dnp.org/AboutUs/DNP3%20Primer%20Rev%20A.pdf, Last access: October 2016.

[114] Sheffield Learning Community. Raspberry Pi, 2012, https://www.sheffieldlearningcommunity.com/sites/default/files/uploads/Raspberry%20Pi.pdf#page=2, Last access: February 2016.

[115] Sonali S. Lagu and Sanjay 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.

[116] Dieter Wimberger and John Charlton. Java Modbus Library, 2004, http://jamod.sourceforge.net, Last access: October 2016.

[117] Ljung Lennart. *System Identification Toolbox: User's Guide*. The MathWorks, Inc., 2012.

[118] Mark Rollins. *Beginning LEGO MINDSTORMS EV3*. Apress, 2014.

[119] Seung Yeob Nam, Dongwon Kim, Jeongeun Kim, et al. Enhanced ARP: preventing ARP poisoning-based man-in-the-middle attacks. *IEEE communications letters*, 14(2):187–189, 2010.

[120] Petr Chmelar. Java Kalman Library, 2014, https://sourceforge.net/projects/jkalman/, Last access: October 2016.

[121] Garrett Hayes and Khalil El-Khatib. Securing modbus transactions using hash-based message authentication codes and stream transmission control protocol. In *2013 Third International Conference on Communications and Information Technology (ICCIT)*, pages 179–184, June 2013.

A Cyber-Physical Countermeasure

A.1 Security Mechanism

In this appendix, we present a security response mechanism based on the management of two different communication protocol modes. These protocol modes are defined as: (i) *Normal protocol mode* which prioritize the real time communication instead of the secure communication; and (ii) *Degraded protocol mode* which has a security mechanism in order to guarantee the security in the communication and the safety in the physical system. Both protocol modes switch depending on the risk level detected in the system. To detect the existent risk level in the system, we use the multi-watermark detector proposed in Chapter 3, as well as the PIETC-WD strategy proposed in Chapter 4. The goal of the solution presented in this appendix is preserving the real-time communication, if the system has not an attack risk; or moving toward a secure communication, if the system has a high attack risk.

The PIETC-WD strategy measures the risk level, and provides the alerts to switch from a normal protocol mode to a degraded protocol mode. It is worth mentioning that the switch from normal mode to degraded mode is carried out by the remote controller or by the sensors. Otherwise, the switch between the degraded mode to normal mode is carried out only by the remote controller and managed by an operator. Moreover, noting also that this mechanism allows to focus on the part of the physical system which has been disrupted. The PIETC-WD strategy permits to know if an attack is happening and to activate the degraded protocol mode to ensure the communication activating also the safety mode at the physical system. There are three different situations: (1) If this mechanism is activated without problems and the attack is avoided, we estimate that the attacker used the network to conduct an attack; (2) If this mechanism manages to secure the communication network, but a part of the physical system is not in a safety mode. In this case, we know the area of the physical system where the attacker is working; and (3) If this mechanism cannot be activated in a part of the communication network. In this case, we know that the attacker uses the communication network to conduct the attack.

The solution has been implemented using the training cyber-physical testbed described in Chapter 5. We use standard Modbus over TCP during the normal protocol mode, and a secure Modbus over TCP implementation that includes keyed-Hash Message Authentication Codes (HMAC) [121], during the degraded protocol mode. The PIETC-WD strategy implemented in the testbed gives the capability to detect the attack risk to the master and slave devices of the testbed. In the next section, we present the details about the implementation of this countermeasure.

A.2 Implementation

We have implemented the countermeasure using a modified version of the Modbus protocol that includes HMAC authentication and control of integrity. By switching from one mode to another may generate other security problems. We have to make sure that the switching mechanism prevents an attacker from reversing or overriding the switching procedure. To solve these problems, we have implemented: (1) A counter, in order to increase the security to avoid replay attacks; (2) Heartbeat messages to ensure authentication; and (3) A transition challenge to make more secure the switching from degraded mode to normal mode. In the rest of this section, we define how we have implemented these mechanisms to integrate the countermeasure.

A.2.1 Two Modes: Normal and Degraded

We use the *Protocol ID* field of MODBUS/TCP protocol in order to warn that the system is using normal mode or degraded mode. The slave can change from normal mode to degraded mode and the master can change from normal mode to degraded mode and vice-versa. We show in Table A.1 a *Protocol ID* implementation.

Modbus application Header			PDU	
Transaction ID 2 bytes	Protocol ID 2 bytes	Length 2 bytes	Unit Identifier 1 byte	Data n bytes
	`0x0000`: normal mode `0x00FF`: Degraded mode			

Table A.1 – Protocol ID implementation on MODBUS/TCP.

A.2.2 Message Counter

We implement a counter in the MODBUS/TCP messages in order to increase the communication security against replay attacks. We use the *Transaction ID* field of the MODBUS/TCP messages to store the counter, as we show in Table A.2. The counter range is from 0 to $2^{16} - 1$. The counter is reset when it has reach the maximum 65535.

Modbus application Header			PDU	
Transaction ID 2 bytes	Protocol ID 2 bytes	Length 2 bytes	Unit Identifier 1 byte	Data n bytes
Counter [0, 65535]				

Table A.2 – Message counter implementation on MODBUS/TCP.

A.2.3 Heartbeat Message

In the system, normal protocol mode allows to keep the real-time communication. However, when this mode is activated, the system is vulnerable to man in the middle and replay attacks. The PIETC-WD strategy allows to detect these attacks in physical system and in the communication network. Nevertheless, in order to ensure authentication, and increase the security against these attacks, we have added a Heartbeat frame to our MODBUS/TCP implementation. This Heartbeat message is sent periodically by the master to the slave. The period of sending can be set by using the message counter or a clock as a token or timestamp. This Heartbeat frame has its own *Protocol ID* and possesses a HMAC field so that the slave can verify the integrity of the frame and authenticate the master. If the slave receives no valid Heartbeat frame within a given period (timewise) then the slave enters in a HMAC mode.

To ensure authentication, the Heartbeat message contains in its data field the IP address, the port and the current mode. In addition, the counter is considered as a *nonce* to avoid replay attacks.

A.2.4 Transition Message Challenge

We implement a Transition message to make more secure the switching from degraded mode to normal mode. We consider that only the master can send this kind of message. We also consider that allowing any device to switch from normal to secure mode is not a real problem because an attacker should not know the secret key to write an HMAC message. Heartbeat messages regularly check that a slave receives order from a legitimate master, even in normal mode.

This Transition challenge really looks like the Heartbeat challenge. The differences are only the *Protocol ID* and the response. The response from a Transition message has no HMAC but contains the Transition *Protocol ID* and the same information in the data field than an Heartbeat response.

A.2.5 Master and Slave Reactions Toward Messages

In this section, we explain how the slave and master react. There are two main situations: (i) if one of the two devices detects an attack, it switches to secure mode; or (ii) if one of the two devices detects a wrong HMAC, a wrong *Protocol ID* or a wrong message counter, it drops the corresponding message. Other important situations are listed in the following paragraphs and summarized in Table A.3.

Master Side

- If the master does not receive any response for a given number of request, the master rises an alert.
- If the master receives a consecutive given number of wrong HMAC messages, the master rises an alert.
- If the master does not receive a consecutive given number of Heartbeat messages, the master tries to use the secure mode.
- If the secure mode is already active, the master rises an alert.
- If the master receives a wrong transition response (any wrong input in the data field), the master switches to secure mode.

Slave Side

- If the slave receives a consecutive given number of wrong HMAC messages, the slave sends a message of alert in normal mode.
- If the slave does not receive a consecutive given number of Heartbeat messages, the slave tries to use the secure mode.
- If the secure mode is already active, the slave puts the work in safety mode.
- If the slave receives a wrong transition response (any wrong input in the data field), the slave switches to secure mode.

MODBUS Mode	Entity	Cause or event	Reaction
Normal	Slave	No Heartbeat received	Switch to HMAC Mode
HMAC	Slave	No Heartbeat received	Alert
Normal or HMAC	Master	No Messages received from slave	Alert
Normal	Master	No Heartbeat received	Switch to HMAC Mode
HMAC	Master	No Heartbeat received	Alert

Table A.3 – Reaction of master and slave to different Events.

B French Summary

B.1 Introduction

De nos jours, de nombreuses entreprises et industries doivent avoir accès aux données critiques depuis n'importe quel endroit, en assurant le contrôle de ces données ainsi que des processus industriels. Cette nécessité rend la combinaison de la sécurité du réseau et de la sécurité du contrôle industriel un domaine de recherche très important. Ce domaine couvre plusieurs champs : (1) les Technologies de l'Information et des Communications (TIC) qui englobent le contrôle des réseaux informatiques et des communications ; (2) la cybersécurité traditionnelle, ciblée sur la création de techniques de détection et de contre-mesures contre les attaques dans le domaine cyber ; (3) les Systèmes de Contrôle Industriel (ICS), axés sur la performance et optimisation des processus physiques de l'industrie ; et (4) la sûreté dans les processus industriels, ciblée sur les moyens d'éviter les pannes et les accidents dans le processus.

Avant de commencer, nous définissons dans ce paragraphe quelques concepts importants : (i) les mots cyber et physique, que nous utiliserons dorénavant pour faire allusion aux technologies de l'information et des communications et aux systèmes de contrôle respectivement ; (ii) les systèmes cyber-physiques, définis comme une nouvelle génération de systèmes qui combinent des composants cybers et physiques en utilisant des données dans le domaine numérique et continu [1] ; (iii) les systèmes de contrôle à travers le réseau (Networked Control System) qui sont un sous-ensemble des systèmes cyber-physiques dédiés aux systèmes de contrôles industriels ; et (iv) les systèmes de contrôle, définis comme une interconnexion de composants qui forment un système physique (souvent appelés usines, et dans cette thèse *environnement physique*) et qui fournissent la réponse souhaitée. Nous nous concentrons spécifiquement sur les systèmes de contrôle en boucle fermée, où le système de contrôle a un retour d'information pour maintenir une certaine relation entre les différentes variables du système.

La sécurité des systèmes cyber-physiques attire beaucoup d'attention [3], notamment après que le malware StuxNet [2] a révélé le potentiel des attaques de sécurité menées contre de tels systèmes. Plusieurs auteurs ont étudié les exigences pour tenir compte des nouveaux problèmes de sécurité

lors de la conception de mécanismes de sûreté pour les systèmes cyber-physiques. Dans [4], Cardenas *et al.* définissent les problèmes de sécurité dans ces systèmes en analysant séparément le problème, d'abord du point de vue de la sécurité de l'information, puis en examinant les problèmes de contrôle spécifiques. Dans [5], Cardenas *et al.* décrivent pour la première fois la différence entre la sécurité des réseaux d'entreprises classiques, et la sécurité des systèmes cyber-physiques. La Figure B.1 montre comment les adversaires qui mènent une attaque cyber-physique peuvent être représentés par un schéma généralement utilisé par la communauté des systèmes de contrôle. Le symbole $\bigoplus$ dans la figure représente une *addition*, c'est-à-dire un élément linéaire qui délivre la somme d'un certain nombre de signaux d'entrée. La figure représente la boucle fermée d'un système de contrôle, et la manière dont les adversaires réussissent à modifier certaines des lectures. Plus spécifiquement, les adversaires modifient l'entrée de contrôle u_t (en insérant à la place u'_t) pour affecter l'état du système et perturber les conditions de fonctionnement normales. Les adversaires n'ont pas besoin de la connaissance du modèle de processus du système. Cependant, l'accès à tous les capteurs (i.e., à tous les composants du vecteur y_t) ou aux protocoles de communication est nécessaire pour effectuer une attaque, c.-à-d. pour pouvoir insérer le bon vecteur y_t à la place du vecteur y'_t généré à cause des données malveillantes u'_t. Ce type d'adversaires est alors indétectable avec un détecteur qui vérifie uniquement les mesures défectueuses.

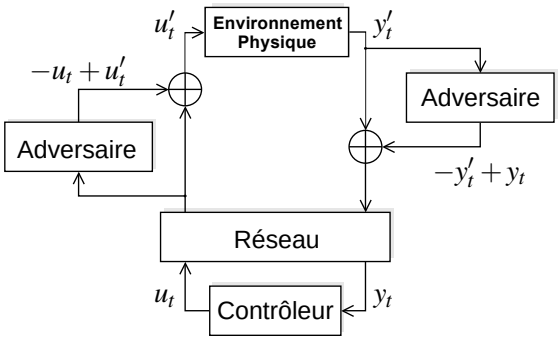


Figure B.1 – Représentation d'une attaque cyber-physique. u_t et y_t représentent les vecteurs d'entrée et de sortie du système. u'_t et y'_t représentent les vecteurs d'attaque.

Du point de vue cyber, les technologies de type SCADA (Supervisory Control and Data Acquisition) sont utilisées pour contrôler les environnements industriels (comme par exemple, la distribution d'énergie, ou les systèmes de transport). En plus, les protocoles basés sur les systèmes de contrôle à travers le réseau doivent couvrir des règles de régulation telles que les retards et les anomalies [6]. En effet, la plupart des protocoles de contrôle industriel (par exemple, MODBUS, DNP3, AGA-12, PROFINET et EtherNet/IP) sont conçus pour assurer la sûreté des systèmes, mais pas la sécurité de l'information à travers du réseau. Néanmoins, il existe des protocoles avec des extensions de sécurité. AGA-12 utilise la cryptographie pour ajouter une protection d'intégrité et de confidentialité, mais avec un coût de déploiement élevé [7]. DNP3 possède une extension nommée DNP3-SA (Secure Authentication, à partir de la cinquième version IEEE-1815-2012), ajoutant à DNP3, l'intégrité et l'authentification des messages. Toutefois, les

systèmes cyber-physiques actuels utilisent ces protocoles sur TCP/IP ou UDP/IP (par exemple, MODBUS, DNP3 et PROFINET sur TCP, EtherNet/IP sur TCP ou UDP). Dans ce cas, il existe juste des mécanismes de sécurité jusqu'à la couche application telle que TLS et IPSec.

Au niveau de la couche application, nous trouvons aussi des protocoles qui ont évolué. Par exemple, PROFINET qui a une nouvelle couche, PROFIsafe, conçue pour assurer la sûreté, protégeant ainsi le protocole contre un dysfonctionnement (par exemple, des erreurs de transmission). Cependant, cela n'assure pas la sécurité contre les actes malveillants intentionnels [8]. Il convient de noter que la plupart des protocoles qui fonctionnent sur Ethernet ou TCP/IP sont des modifications de protocoles séries qui ne fournissent pas de sécurité. Bien que les couches transport et réseau puissent fournir certains éléments de sécurité, ces mécanismes ne suffisent pas à assurer la protection des données de contrôle [9]. Pour résoudre pleinement le problème de la protection des données de contrôle, il est nécessaire d'ajouter des solutions cybers-physiques à ces protocoles.

Dans la littérature, certains auteurs ont proposé l'utilisation d'une attestation physique à la couche cyber [10], une signature physique envoyée par la couche cyber à la couche physique afin de vérifier le bon comportement des processus physiques [9], ou une signature sur les données physiques pour éviter d'identifier la valeur réelle des données et sécuriser la communication [11]. Dans [12], Arvani *et al.* décrivent une méthode de détection en utilisant la transformation des signaux en ondelettes discrètes. Do *et al.* [13] étudient des stratégies pour gérer les attaques cyber-physiques en utilisant des méthodes de détection statistiques. Ces propositions ne sont valides que lorsque les adversaires effectuent une attaque par rejeu ou une attaque d'intégrité sans avoir la capacité d'acquérir des connaissances sur les processus physiques.

B.1.1 Objectifs et contributions

Dans cette thèse, nous nous concentrons sur la sécurité entre la couche cyber et la couche physique des systèmes cyber-physiques. Nous commencons par une analyse de sécurité basée sur des mécanismes théoriques de détection proposés par Mo et Sinopoli [15] et Chabukswar *et al.* [16], qui étudient les signatures stationnaires dans les systèmes cyber-physiques. Poursuivant l'approche des détecteurs basés sur des signatures, nous proposons un nouveau mécanisme de détection utilisant des signatures non stationnaires, afin de couvrir un nombre plus grand de menaces. Ce nouveau mécanisme augmente le taux de détection d'attaque tout en conservant le même coût de performance que l'approche précédente. Ensuite, nous analysons les limites de la nouvelle proposition. Cette analyse nous amène à améliorer le mécanisme de détection, ainsi qu'à créer une nouvelle stratégie de contrôle et de sécurité capable d'éviter les faiblesses de sécurité générées par l'adhésion de la couche cyber au domaine physique et de contrôle.

Motivation : La sécurité actuelle des systèmes cyber-physiques est axée sur les adversaires cybers ou les adversaires physiques, mais pas les deux.

Objetifs de la thèse : Les nouveaux défis de sécurité dans les systèmes cyber-physiques obligent à analyser les stratégies de contrôle et les mécanismes de sécurité afin de détecter les attaques. Cette analyse nous permettra de créer une nouvelle stratégie de contrôle et de sécurité, en améliorant les mécanismes de détection existant dans la littérature, afin de sécuriser la couche cyber et la couche physique contre les adversaires cyber-physiques.

Contributions de la thèse : Les mécanismes proposés dans cette thèse nous permettent de détecter les menaces menées par les adversaires cyber-physiques. En plus, nous analysons les différents adversaires cyber-physiques, et nous classifions ces adversaires en fonction de la capacité d'obtenir le bon comportement du système. Autrement dit, leur capacité de trouver une corrélation entre les entrées et sorties du système. Cette classification nous donne deux types différents d'adversaires cyber-physiques : adversaires cyber-physiques paramétriques et non paramétriques. Nous abordons également les défauts des mécanismes de détection centralisés en proposant une stratégie de detection decentralisée qui permet d'augmenter la robustesse du système contre les attaques. Ensuite, nous définissons un mécanisme de détection distribué qui augmente la robustesse face aux attaques cyber-physiques. Pour finir, nous construisons des simulations et bancs de test SCADA afin de valider les nouveaux modèles de détection.

B.2 Schéma d'authentification défi-réponse dynamique

Dans cette section, notre attention est centrée sur les problèmes d'intégrité en raison de l'interconnexion entre les domaines *cyber* et *physique* dans les systèmes de contrôle à travers le réseau. Plus précisément, nous nous concentrons sur l'adaptation des mécanismes de détection d'anomalies, existant dans le domaine physique, pour gérer également les attaques.

Le schéma d'authentification proposé par Mo *et al.* [37] repose sur l'adaptation d'un détecteur d'anomalies en temps réel basé sur un modèle *linéaire* et *invariant* du système. Ce schéma, construit à partir de *Filtres de Kalman* et *régulateurs linéaires-quadratiques*, génère des signatures d'authentification pour protéger l'intégrité des mesures physiques communiquées à travers le réseau, car si on ne protège pas les messages qui portent ces mesures, des actions malveillantes peuvent être menées pour induire le système en erreur. Cependant, nous montrons que le schéma de détection proposé par Mo *et al.* ne fonctionne que contre certaines attaques d'intégrité. Nous présentons deux nouveaux modèles d'adversaires qui peuvent échapper à ce détecteur. Ces adversaires sont classés en fonction de l'algorithme utilisé pour obtenir la connaissance de la dynamique du système afin de mener à bien l'attaque. Ensuite, nous revisitons le mécanisme proposé dans [9, 53] et évaluons ses performances par rapport aux deux nouveaux modèles d'adversaires presentés dans cette section. Nous adaptons ce schéma de détection pour gérer les limitations non couvertes, en validant l'approche résultante par des simulations numériques.

B.2.1 Formulation du problème

Dans cette thèse, nous considerons les environnements physiques des systèmes de contrôle industriels qui peuvent être mathématiquement modélisés en tant que systèmes discrets linéaires invariables dans le temps (LTI). Il convient de mentionner qu'un modèle mathématique fournit un moyen rigoureux pour décrire le comportement dynamique d'un système donné. Une telle catégorie de systèmes peut être décrite comme suit :

$$x_{t+1} = Ax_t + Bu_t + w_t \tag{B.1}$$
$$y_t = Cx_t + v_t \tag{B.2}$$

où $x_t \in \mathbb{R}^n$ est le vecteur des variables d'état, $u_t \in \mathbb{R}^p$ est le signal de contrôle, $y_t \in \mathbb{R}^m$ est la sortie du système, et $w_t \in \mathbb{R}^n$ et $v_t \in \mathbb{R}^m$ sont le *bruit du processus* et le *bruit des mesures des capteurs* respectivement. Les bruits sont supposés être des bruits blancs gaussiens avec moyenne nulle et covariance Q, *i.e.* $w_t \sim N(0,Q)$ et R, *i.e.* $v_t \sim N(0,R)$. En plus, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$ et $C \in \mathbb{R}^{m\times n}$ sont respectivement la matrice d'*état*, la matrice d'*entrée*, et la matrice de sortie.

Pour la classe de systèmes définis ci-dessus, une des méthodes de contrôle la plus largement utilisée est la commande *linéaire quadratique gaussienne* (LQG). Cette commande est constituée de deux composants qui peuvent être conçus de manière indépendante :

1. Un *filtre de Kalman* qui produit une estimation optimale d'état $\hat{x}_t$ de l'état x_t en fonction des mesures de bruit obtenues.
2. Un *régulateur linéaire quadratique* (LQR) qui fournit la loi de contrôle u_t qui résout le problème LQR, en fonction de l'estimation de l'état $\hat{x}_t$.

B.2.2 Detecteur basé sur une signature stationnaire

Cette section décrit brièvement le schéma de détection proposé par Mo *et al.* [9, 53]. La procédure s'applique aux environnements physiques qui suivent un modèle LTI en temps discret, et sont contrôlés par un contrôleur LQG (cf. section B.2.1).

Avant de présenter le schéma de détection, nous fournissons une définition du modèle d'adversaire considéré en [9, 53] :

Définition B.2.1. *Un attaquant qui a la possibilité d'écouter tous les messages contenant les sorties du capteur y_t, et d'injecter des messages avec un signal y'_t pour mener des actions malveillantes, est défini comme un adversaire cyber.*

Remarque B.2.1. *Il est important de noter que la définition donnée ci-dessus suppose que l'attaquant ne possède pas (ou ne tente pas de rassembler) des connaissances du modèle du système. Pour cette raison, nous désignons un tel attaquant comme un adversaire cyber.*

Dans ce qui suit, nous appellerons u_t^* la sortie du contrôleur LQR et u_t l'entrée de contrôle qui est envoyée à l'environnement physique (cf. Équation (B.1)). L'idée est de superposer à la loi de contrôle optimale u_t^* un signal de signature $\Delta u_t \in \mathbb{R}^p$ qui sert de signal d'authentification. Ainsi, l'entrée de contrôle u_t est donnée par :

$$u_t = u_t^* + \Delta u_t \tag{B.3}$$

Le signal de signature est un signal gaussien aléatoire avec moyenne nulle, qui est indépendant à la fois du bruit du processus w_t et du bruit de la mesure v_t. Cette signature d'authentification devrait détecter les attaques de répétition et d'intégrité générées par l'adversaire cyber défini ci-dessus. Étant donné que la loi de contrôle optimale u_t^* est équipée du signal d'authentification Δu_t, un *détecteur* – co-localisé physiquement avec le contrôleur – peut être conçu avec l'objectif de générer des alarmes lorsqu'une attaque a lieu. Dans ce but, Mo *et al.* [9, 53] proposent d'utiliser un détecteur χ^2, qui est une catégorie bien connue de détecteurs d'anomalies en temps réel classiquement utilisés pour la détection d'anomalies dans les systèmes de contrôle [99], avec pour objectif la détection d'attaque. La figure B.2 montre le système de contrôle global équipé du détecteur d'attaque proposé dans [9, 53].

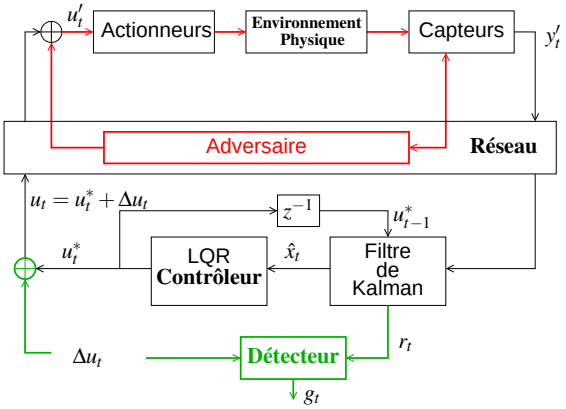


Figure B.2 – **Protection à partir de signatures dans les systèmes cyber-physiques [9].**

Un *signal d'alarme* g_t est calculé en fonction des résidus $r_t = y_t - C\hat{x}_{t|t-1}$ générés par l'estimateur. Ensuite, g_t est comparé à un seuil, γ, pour décider si le système est dans un état normal. Le seuil est réglé pour minimiser les fausses alarmes [9, 53]. Le signal d'alarme g_t est calculé comme suit :

$$g_t = \sum_{i=t-w+1}^{t} (y_i - C\hat{x}_{i|i-1})^T \mathcal{P}^{-1} (y_i - C\hat{x}_{i|i-1}) \tag{B.4}$$

où w est la taille de la fenêtre de détection et $\mathcal{P}$ est la co-variance d'un signal d'entrée gaussien indépendant et identiquement distribué, provenant des capteurs.

Le système est considéré comme non attaqué si $g_t < \gamma$. Le système est sinon considéré comme attaqué et le détecteur génère une alarme.

B.2.3 Adversaires cyber-physiques

Dans cette section, nous introduisons un adversaire amélioré qui est conscient du fait que le système utilise le détecteur χ^2 présenté ci-dessus. Étant donné que le détecteur est basé sur un signal de signature stationnaire Δu_t, nous montrons qu'un adversaire capable d'extraire le modèle du système à partir de la loi de contrôle u_t et de la mesure des capteurs y_t, est capable d'effectuer une attaque tout en restant non détecté.

Définition B.2.2. *Un attaquant qui, en plus des capacités du cyber-adversaire, est en mesure d'écouter les messages contenant la sortie du contrôleur (u_t) dans l'intention d'améliorer ses connaissances sur le modèle du système à l'aide d'un modèle d'identification paramétrique ou non-paramétrique, est défini comme un adversaire cyber-physique.*

En fonction de la manière de modéliser le comportement du système, deux adversaires cyber-physiques différents peuvent être définis comme suit :

Définition B.2.3. *Un attaquant qui utilise uniquement l'entrée et la sortie précédente du système pour identifier le modèle du système est défini comme un adversaire cyber-physique non-paramétrique.*

Remarque B.2.2. *Un adversaire cyber-physique non paramétrique peut utiliser, par exemple, un outil d'identification de modèle basé sur un filtre à réponse impulsionnelle finie (FIR) pour identifier le modèle du système [100]. Dans la figure 3.1, les signaux u'_t et y'_t sont supposés être respectivement la sortie du contrôleur et la sortie des capteurs lorsqu'une attaque est en train de se produire. Nous désignons par $\Delta u'$ la signature estimée par l'adversaire cyber-physique non paramétrique.*

Définition B.2.4. *Un attaquant capable d'estimer les paramètres du système en utilisant les données d'entrée et de sortie pour tromper le détecteur du contrôleur est défini comme un adversaire cyber-physique paramétrique.*

Remarque B.2.3. *Un adversaire cyber-physique paramétrique est capable d'estimer les paramètres du système en utilisant les données d'entrée et de sortie pour induire en erreur le détecteur du contrôleur. Cet adversaire peut utiliser, par exemple, un modèle ARX (autorégressif avec entrée exogène) ou un modèle ARMAX (autorégressif avec moyenne dynamique et entrée exogène) pour estimer la dynamique du système [66].*

Nous supposons que la contrainte principale de cet adversaire est l'énergie dépensée pour écouter et analyser les données de communication, c'est-à-dire le nombre d'échantillons nécessaires pour obtenir les paramètres du modèle du système.

B.2.4 Detecteur basé sur des multi-signatures

Dans les sections précédentes, nous avons défini trois types d'adversaires qui utilisent différentes vulnérabilités des systèmes de contrôle pour effectuer des attaques ; les cyber-adversaires, les

adversaires cyber-physiques non paramétriques et les adversaires cyber-physiques paramétriques. Dans cette section, nous proposons un schéma de détection qui étend celui présenté dans [9, 53], afin de détecter des adversaires cyber-physiques. Nous étudions également la perte de performance du nouveau système de détection par rapport à celui présenté dans [9, 53].

L'objectif de notre nouveau schéma de détection est d'augmenter la difficulté à récupérer la signature d'authentification Δu_t à partir du signal de contrôle u_t, de sorte que la probabilité de détecter une attaque d'un adversaire cyber-physique non paramétrique peut être augmentée. Nous supposons que le système de contrôle attaqué utilise exactement le même type de contrôleurs et la même stratégie de détection présentés dans les section B.2.1 et B.2.2 . La seule différence dans le schéma de détection proposé est la façon dont le signal de signature, Δu_t, est généré. L'entrée de contrôle u_t, comme dans le cas du schéma de détection présenté dans la section B.2.2, est calculée comme la superposition du signal de contrôle optimal u_t^* produit par le contrôleur LQR et un signal de plusieurs signatures, Δu_t. L'idée est de construire le signal de la signature d'authentification en alternant entre N processus différents et indépendants avec une co-variance et une moyenne différentes. Plus précisément, la signature non stationnaire, Δu_t, est obtenue en changeant périodiquement, avec une période T, entre N signals $\Delta u^{(i)}$, avec $i \in I = \{0, 1, \ldots, N-1\}$, extrait par différents processus stochastiques. Par conséquent, le signal de signature Δu_t peut être formalisé comme suit :

$$\Delta u_t = \Delta u_t^{(s(t,T))} \tag{B.5}$$

où $s : \mathbb{N} \times \mathbb{R} \to I$ est une fonction statique qui désigne l'échantillon du temps, t, et la période de commutation T à un élément de l'ensemble d'index, I, défini comme suit :

$$s(t,T) = \left\lfloor \frac{1}{T} \mod (t, NT) \right\rfloor \tag{B.6}$$

où $\mod(x,y)$ est l'opérateur modulo et $\lfloor \cdot \rfloor$ est la fonction partie entière par défaut.

En utilisant la signature proposée (cf. Équation (B.5)), nous avons maintenant un mécanisme de protection adaptative approprié avec deux paramètres principaux configurables ; le nombre de distributions N et la fréquence de commutation $f = 1/T$. Notez que le signal de signature d'origine décrit dans la section B.2.1 est récupéré lorsque $f \to 0$ et lorsque $\Delta u_t^{(0)}$ est un processus gaussien et stationnaire avec moyenne nulle.

Validation contre les adversaires cyber-physiques non-paramétriques

Cette section valide à l'aide des simulations numériques le schéma de détection proposé précédemment. En particulier, nous voulons montrer que le signal de signature proposé est capable de détecter des adversaires cyber-physiques non paramétriques (cf. section B.2.3) avec un taux de détection plus élevé par rapport à celui obtenu avec la signature proposée dans [9, 53]. La simulation est basée sur des modèles Matlab et Simulink d'une usine, ainsi que sur les modèles

des adversaires cyber-physiques non paramétriques. Nous utilisons trois distributions différentes (c.-à-d., $N = 3$) commutées au hasard : une distribution gaussienne, une distribution de Rician et une distribution de Rayleigh.

Pour quantifier l'efficacité du schéma de détection proposé, nous calculons le taux de détection en fonction de la fréquence de commutation. En particulier, pour chaque fréquence f considérée, nous effectuons 200 simulations de Monte Carlo (avec des paramètres de système générés de manière aléatoire) dans le cas d'un adversaire cyber-physique non-paramétrique et d'un adversaire cyber, et nous calculons la fonction de répartition cumulée (CDF) du taux de détection.

Nous commençons par confronter la performance obtenue avec la stratégie de détection basée sur une signature non stationnaire proposée dans cette section par rapport à celle proposée dans [9, 53] dans le cas d'un adversaire cyber et d'un adversaire cyber-physique non-parametrique. Nous considérons ici deux fréquences de commutation $f_L = 0.05$Hz (changer la signature après 20 étapes) et $f_H = 0.14$Hz (changer la signature après 7 étapes). Nous vérifions que la stratégie de détection proposée dans [9, 53], comme indiqué précédemment, peut détecter une attaque cyber, mais fonctionne mal lorsqu'un adversaire cyber-physique attaque le système. Néanmoins, la stratégie de détection proposée basée sur une signature non stationnaire est capable de fournir un taux de détection plus élevé. En particulier, nous remarquons que le détecteur utilisant une fréquence de commutation plus élevée f_H offre de meilleures performances par rapport à l'utilisation de la fréquence de commutation inférieure f_L.

Validation de l'efficacité : Nous avons validé ci-dessus le détecteur de signature non stationnaire à l'aide d'une fonction statique I pour définir la multi-signature. Ci-après, nous présentons les résultats et les validations obtenus pour un système avec la même perte de performance entre le détecteur utilisant une signature stationnaire et celui utilisant une signature non stationnaire où cette signature non stationnaire est générée à partir d'une fonction non statique, I_d. Dans cette simulation, les deux détecteurs ont une perte de performance de 30%, ΔJ, par rapport au coût optimal. De plus, la signature utilise une fonction dynamique pour définir la non-stationarité. Nous constatons qu'en utilisant la multi-signature (ou signature non stationnaire) avec la même perte de performance que la signature stationnaire, le rapport de détection augmente lorsque la fréquence de commutation varie dans la gamme $[0, 0.14]$ Hz, où $f = 0$ est le détecteur de la signature stationnaire. Nous confirmons que la performance de la multi-signature augmente jusqu'à $f = 0.14$ Hz où nous observons un pic avant la stabilisation du taux de détection. Dans la section suivante, nous étendons l'analyse au cas des adversaires cyber-physiques paramétriques. En plus, nous testons des systèmes d'ordre différent concluant que le taux de détection augmente avec la complexité du système.

Validation contre les adversaires cyber-physiques paramétriques

Précédemment, nous avons vu comment le détecteur de multi-signature est capable de détecter des adversaires cybers et cyber-physiques non paramétriques. Ci-après, nous étendons l'étude

au cas des adversaires cyber-physiques paramétriques (cf. définition B.2.4). Nous rappelons que les adversaires cyber-physiques paramétriques sont capables d'identifier les paramètres du modèle du système à partir des signaux d'entrée et de sortie de l'usine (environnement physique). Un adversaire cyber-physique paramétrique peut obtenir le modèle du système avec une grande précision si les commandes de contrôle et les mesures des capteurs sont accessibles. Par exemple, en utilisant la caractéristique de la signature, un adversaire cyber-physique paramétrique peut utiliser un modèle ARX (autorégressif avec entrée exogène) pour définir le système.

De la même manière que pour la validation précédente, nous analysons le ratio de détection pour 200 simulations de Monte Carlo en utilisant des systèmes d'ordre 25, contre sept adversaires cyber-physiques paramétriques différents. La taille de la fenêtre supposée est $\hat{T} = 300$. Si les adversaires utilisent un modèle du système avec le bon ordre, le ratio de détection est d'environ 8%. L'ensemble d'ordres des systèmes où le ratio de détection n'augmente pas de façon drastique est $[18, 28]$. Sinon, la probabilité de détecter l'adversaire est élevée. Ensuite, nous analysons le ratio de détection du même système, contre un adversaire cyber-physique paramétrique avec différentes tailles de fenêtre (125, 150, 200, 250, et 300), et avec l'ordre correct du système. Nous concluons avec les résultats obtenus que la taille de la fenêtre utilisée par l'adversaire et inversement proportionnelle au ratio de détection.

Remarque B.2.4. *Un adversaire cyber-physique paramétrique est capable d'obtenir le modèle du système, $H(z)$, et d'induire le contrôleur en erreur en écoutant les entrées de contrôle et les mesures des capteurs. La probabilité d'être détectée équivaut à la probabilité d'obtenir un modèle erroné. Cette probabilité est directement proportionnelle à l'ordre du système ; et inversement proportionnelle à la taille de la fenêtre pour écouter le canal de données.*

De la remarque B.2.4 découle que, si nous considérons le système réel comme une boîte noire, une identification erronée du système dépend de l'ordre du système choisi par les adversaires pour recréer le modèle du système, ainsi que du nombre d'échantillons écoutés et de la taille de la fenêtre utilisée par les adversaires pour recalculer les paramètres du système cible. Cette situation peut être quantifiée en utilisant l'erreur quadratique moyenne (EQM) [63, 68]. En résumé, la probabilité d'obtenir le modèle correct du système ciblé est directement proportionnelle à l'ordre choisi par les adversaires pour générer le modèle et inversement proportionnel au nombre d'échantillons récupérés. Le coût de calcul pour les adversaires est directement proportionnel à l'ordre du système, car ce type d'adversaires doit augmenter l'ordre du modèle, ainsi que la taille de la fenêtre afin de minimiser le EQM. Par conséquent, le nombre d'échantillons écoutés avant d'effectuer l'attaque, et l'ordre du système choisi par les adversaires sont les deux paramètres principaux pour échapper à la détection.

B.2.5 Discussion

Nous montrons dans la section B.2 que la stratégie de détection avec un signal de signature stationnaire n'est pas suffisamment robuste du point de vue de la sécurité. En effet, nous prouvons quantitativement que l'approche ne détecte que les *adversaires cybers*. Ensuite nous présentons

un schéma de détection adaptatif basé sur plusieurs signatures avec deux paramètres configurables principaux : le nombre de distributions et la fréquence de commutation. La nouvelle proposition *multi-signature* réussit à détecter correctement les adversaires cybers et cyber-physiques non paramétriques, sous l'hypothèse que les distributions de la signature changent fréquemment. Dans la prochaine section, nous présentons une nouvelle stratégie afin de réduire considérablement le contournement du détecteur pour un adversaire cyber-physique paramétrique.

B.3 Détection adaptative basée sur la théorie du contrôle

Comme nous l'avons montré dans la section B.2, l'utilisation de mécanismes de sécurité cyber-physiques inadéquats peut avoir un effet négatif dans les systèmes cyber-physiques industriels [2, 105, 106]. Ces nouveaux systèmes ont besoin de la collaboration d'un très large nombre de disciplines pour résoudre les défis en termes d'autonomie, de fiabilité, de facilité d'utilisation, de fonctionnalité et de cyber sécurité [107]. Ci-après, nous nous concentrons sur l'utilisation de solutions théoriques de contrôle pour détecter les attaques contre les systèmes cyber-physiques. La littérature traditionnelle propose l'utilisation de stratégies de contrôle pour conserver, par exemple, la performance de la boucle-fermée du système ou les propriétés de sûreté d'un réseau de communication reliant les composants distribués d'un système physique. Cependant, l'adaptation de ces stratégies pour gérer les incidents de sécurité est un défi toujours d'actualité.

La communauté de contrôle travaille activement à adapter les stratégies de contrôle traditionnelles utilisées pour détecter les failles accidentelles et les erreurs, vers la détection d'attaques malveillantes [75, 77, 78]. Motivés par les mêmes objectifs, nous présentons une solution qui complète le détecteur de signature afin de couvrir ces faiblesses. Plus précisément, la nouvelle solution combine les stratégies de contrôle avec la stratégie de défi-réponse analysée et améliorée dans la section précédente. Cette combinaison permet de gérer les attaques d'intégrité contre les systèmes cyber-physiques.

B.3.1 Détection d'adversaires cyber-physiques paramétriques

Dans cette section, nous présentons une stratégie de détection, désignée ci-après sous le nom de *détecteur de signature avec événements de contrôle périodiques et intermittents* (PIETC-WD stratégie), qui vise à détecter les adversaires cybers et cyber-physiques en complétant la stratégie proposée dans la section précédente.

Notre stratégie consiste en un contrôleur local situé dans chaque capteur et un contrôleur à distance commun pour tout le système (contrôleur distribué, cf. Figure B.3). La coopération entre les contrôleurs locaux et le contrôleur à distance nous permet de créer une politique de détection d'intrusion pour capturer les attaques d'intégrité (cf. Definition B.1). Les contrôleurs locaux gèrent la dynamique de l'environnement physique et le contrôleur à distance gère la boucle fermée du système afin d'assurer le système contre les attaques d'intégrité. Notez que

notre nouveau système nécessite un contrôleur supplémentaire pour chaque capteur qui doit avoir suffisamment de puissance de calcul pour traiter les estimations de données pour, entre autre, prédire les erreurs entre les données environnementales et les données estimées. Les actionneurs ne nécessitent pas de puissance de calcul supplémentaire. Néanmoins, pendant le temps entre deux événements consécutifs, ils doivent conserver les dernières données reçues par le contrôleur à distance.

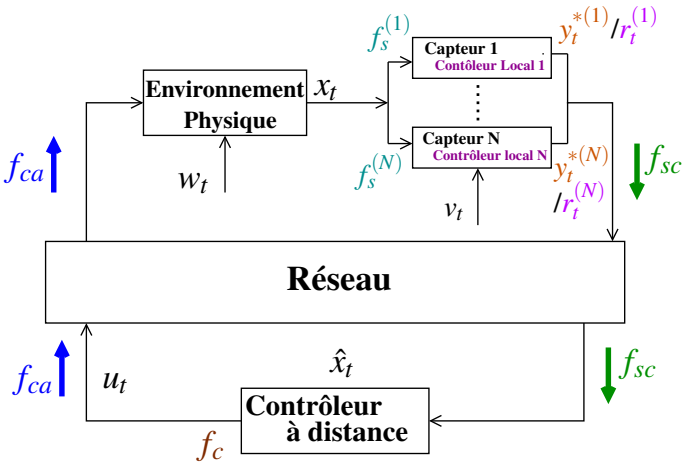


Figure B.3 – **Diagramme d'un système cyber-physique avec une nouvelle stratégie de sécurité basée sur le contrôle du système.**

Un tel système nécessite de définir des politiques de communication parmi les capteurs, les actionneurs et le contrôleur à distance. Nous définissons deux politiques de communication pour assurer le système : (i) *une politique de communication périodique*, avec des communications entre les capteurs et le contrôleur à distance, avec une période $T_{sc} = 1/f_{sc}$, et entre le contrôleur à distance et les actionneurs, avec une période $T_{ca} = 1/f_{ca}$; et, (ii) *une politique de communication intermittente*, qui permet d'envoyer des données des capteurs au contrôleur à distance si un contrôleur local produit une alarme. Notez que T_{sc} ne peut pas être égal à T_{ca} afin d'éviter qu'une communication intermittente ait lieu pendant que la communication périodique est envoyée.

Definition B.1. *Le détecteur de signature avec événements de contrôle périodiques et intermittents (PIETC-WD) est une stratégie de détection avec des tâches de contrôle distribuées. D'une part, les capteurs contrôlent le système de façon périodique, en utilisant leurs contrôleurs et détecteurs de signature locaux. D'autre part, le contrôleur à distance utilise l'erreur d'estimation reçue par chaque capteur pour générer périodiquement les entrées de contrôle. Ce contrôleur surveille également la communication en boucle fermée avec une signature intermittente.*

Pour exécuter la stratégie PIETC-WD, nous developpons deux algorithmes. Le premier définit l'implémentation du contrôleur à distance dont l'entrée est la donnée envoyée par les capteurs et la sortie est l'ensemble des entrées de contrôle envoyées vers le système physique, et la valeur de l'alarme. Le second montre l'implémentation du contrôleur local, placé dans les capteurs, dont l'entrée est la donnée obtenue à partir du système physique, y_t^i, avec $i \in I = \{0, 1, \ldots, N-1\}$, et

la sortie est : (i) le résidu des contrôleurs locaux, r_t^i (avec une signature de réponse au défi), si l'alarme n'est pas activée ; ou (ii) la valeur obtenue par les capteurs du système physique, y_t^i, si l'alarme est activée. En résumé, ces algorithmes définissent comment le contrôleur à distance gère les données afin d'augmenter la probabilité de détecter une attaque, si les données envoyées depuis les contrôleurs locaux ne sont pas correctes, ou si des données ont été perdues. De même, ils déterminent de quelle façon les capteurs modifient les données envoyées au contrôleur à distance si une alarme est activée par les capteurs.

Nouvel adversaire cyber-physique paramétrique : Pour valider cette stratègie, nous présentons un nouvel adversaire cyber-physique paramétrique qui a la connaissance de la nouvelle stratégie de détection, afin de l'évaluer. Cet adversaire connaît les nouvelles politiques de communication et l'existence des différentes signatures des données envoyées depuis les contrôleurs locaux ou le contrôleur à distance. Cependant, il ne connaît pas les co-variances des signatures, les paramètres du contrôleur utilisés pour obtenir l'erreur correcte entre les données ni le moment où le contrôleur à distance force une communication intermittente.

Le nouvel adversaire peut détecter le modèle de corrélation entre les entrées et sorties de l'environnement physique. Il peut forcer la communication intermittente des capteurs avec des mauvaises entrées de contrôle et tromper le contrôleur à distance avec des données d'erreur de lecture pour obtenir le modèle. Néanmoins, cet adversaire n'est pas en mesure de savoir quand la communication est périodique ou intermittente, puisque l'attaquant ne sait pas quand le contrôleur ajoute, aux entrées de contrôle, la signature qui génère la communication intermittente. La communication intermittente ne change pas la fréquence de communication entre le contrôleur à distance et les actionneurs, mais produit une communication intermittente entre les capteurs et le contrôleur à distance, nécessaire pour vérifier la boucle fermée.

En utilisant la stratégie PIETC-WD, ce type d'adversaires est détecté par les contrôleurs localisés dans les capteurs, et par le contrôleur à distance lorsqu'il vérifie le comportement de la boucle fermée. Ces adversaires ne peuvent pas éviter l'alarme dans les capteurs (contrôleurs locaux). Néanmoins, les attaquants peuvent interrompre la communication entre les capteurs et le contrôleur à distance en trompant le contrôleur à distance avec les résidus corrects (par exemple, avec des résidus rejoués). De plus, afin d'éviter de générer une alarme dans le contrôleur à distance, les adversaires peuvent basculer entre l'envoi de la mesure des capteurs ou des résidus. Cependant, ils ont alors une grande probabilité d'être détectés. Nous validons la stratégie PIETC-WD contre les nouveaux adversaires cyber-physiques paramétriques dans la section suivante.

B.3.2 Cas d'utilisation

Cette section présente un cas d'utilisation pratique où la stratégie PIETC-WD proposée dans les sections précédentes pourrait être utilisée dans le monde réel. Ce cas d'utilisation est basé sur une usine chimique. Cette installation possède plusieurs capteurs avec des contrôleurs locaux, des actionneurs et un contrôleur à distance qui gère toutes les mesures des capteurs et des

actionneurs. Les capteurs utilisés dans ce cas d'utilisation envoient des informations sur la pression, la température et la densité. Cet envoi d'information est produit lorsqu'un événement génère une alerte dans un capteur, ainsi que périodiquement pour indiquer le comportement du système au contrôleur à distance. Cette installation doit être contrôlée périodiquement puisque, si le système reçoit des entrées de contrôle incorrectes ou malveillantes capables de perturber le système pendant dix échantillons périodiques consécutifs, il pourrait arriver à un état critique.

Pour éviter qu'un adversaire mette le système dans un état critique, nous utilisons notre stratégie de détecteur (PIETC-WD) avec une politique de gestion de la signature du contrôleur à distance. Ensuite, nous analysons les résultats de 200 simulations de Monte Carlo en utilisant le cas d'utilisation ci-dessus et la politique de signature des contrôleurs (locaux et à distance) contre l'adversaire cyber et cyber-physique. Ces résultats montrent que le ratio de détection est d'environ 97% contre le nouvel adversaire cyber-physique paramétrique et plus de 99% contre les autres adversaires cybers et cyber-physiques en utilisant la stratégie PIETC-WD avec une politique correcte pour la signature du contrôleur à distance.

B.3.3 Discussion

Dans la section B.3, nous nous sommes concentrés sur la conception d'une stratégie de théorie du contrôle adaptatif qui détecte les adversaires cyber-physiques paramétriques, c'est-à-dire les adversaires capables d'acquérir des connaissances sur la dynamique du système avant de commencer leurs attaques, afin de contrôler les entrées et les sorties du système. Nous avons suivi l'idée de décentraliser la stratégie de détection proposée dans la section B.2 en combinant le schéma de protection directement avec les stratégies de contrôle. En ajoutant de la sécurité au *architectural level* du système [111], nous pouvons conserver avec succès la stabilité et les performances du système ainsi que ses propriétés de sûreté et de sécurité. La conception architecturale précise présentée et développée dans cette section décentralise avec succès le détecteur de signature. Après, nous validons cette approche en utilisant Matlab et Simulink et nous présentons également un cas d'utilisation pratique, démontrant ainsi que la stratégie est capable de détecter les attaques cyber-physiques menées par des adversaires cyber-physiques paramétriques. Dans la prochaine section, nous montrons une mise en œuvre de notre stratégie, ainsi que des résultats pratiques qui valident la faisabilité de notre proposition.

B.4 Banc de test pour la détection des attaques cyber-physiques

Les tests expérimentaux sont essentiels pour l'étude et l'analyse des menaces en cours contre les systèmes cyber-physiques. La recherche présentée dans cette section traite de certaines actions visant à développer un banc de test cyber-physique réplicable et abordable pour la formation et la recherche. Dans ce cadre, notre objectif est de mettre en pratique les solutions théoriques développées dans les sections précédentes. Pour atteindre cet objectif, nous mettons en œuvre les solutions dans des scénarios réalistes afin d'analyser leur efficacité contre les attaques

intentionnelles. Plus précisément, nous supposons des environnements cyber-physiques exploités par les technologies SCADA et les protocoles de contrôle industriel. Nous nous concentrons sur deux protocoles représentatifs, largement utilisés dans l'industrie : MODBUS et DNP3 [112, 113]. Les deux protocoles ont des versions sur TCP. Cela nous permet d'émuler des environnements cyber-physiques sur des infrastructures de réseau partagées. Nous supposons une conception Maître/Esclave, qui dicte principalement que les esclaves n'initialisent aucune communication à moins qu'un maître ne le demande. L'un de nos objectifs a été de combiner ces deux protocoles, à la fois pour permettre la flexibilité et le support de plusieurs périphériques avec MODBUS ainsi que les améliorations de sécurité incluses dans les fonctionnalités de DNP3. De plus, les mécanismes de détection cyber-physiques basés sur les stratégies de défi-réponse proposées dans la section B.2 sont inclus dans notre banc de test SCADA. De même, nous intégrons la stratégie de contrôle proposée dans la section B.3 pour expérimenter et analyser ses performances réelles. Pour compléter le banc de test, un ensemble de scénarios d'attaque sont conçus et développés pour tester les attaques contre l'environnement émulé. Ces scénarios se concentrent sur l'attaque des segments MODBUS de notre architecture. Le but final est d'analyser l'efficacité des nouvelles méthodes de sécurité mises en œuvre sur l'environnement émulé et sous l'application de certains modèles d'attaque.

B.4.1 Architecture

L'architecture proposée pour notre banc de test cyber-physique fonctionne comme suit. Tous les éléments du système (contrôleur, capteurs et actionneurs), peuvent être répartis sur plusieurs nœuds dans un réseau partagé combinant les protocoles DNP3 et MODBUS. De même, un ou plusieurs éléments peuvent être intégrés dans un seul périphérique. Du point de vue du logiciel, le contrôleur ne se connecte jamais directement aux capteurs. Au lieu de cela, il est intégré dans l'architecture en tant que PLC, avec des connexions éventuelles à d'autres noeuds intermédiaires. De tels noeuds peuvent traduire les commandes du contrôleur entre différents protocoles (par exemple MODBUS ou DNP3). Cette architecture est capable de gérer plusieurs protocoles industriels et de se connecter à des éléments SCADA complémentaires, tels que des automates et des RTU supplémentaires. Pour faire évoluer l'architecture vers un banc de test complet, de nouveaux éléments peuvent être inclus dans le système, tels que des noeuds de RTU supplémentaires semblables à des proxy.

B.4.2 Mise en œuvre des modèles d'adversaire

Après avoir mis en marche l'architecture, la prochaine exigence consiste à implémenter les adversaires rapportés dans les sections précédentes. Pour développer ces scénarios, nous utilisons un modèle d'attaquant commun. Il implémente la plupart des capacités sous-jacentes des adversaires et peut être étendu pour implémenter les adversaires les plus spécifiques. Par exemple, nous supposons que les attaquants peuvent intercepter tous les échanges de communication entre les extrémités et, par conséquent, modifier, stocker et analyser ce qui peut être rejoué

pour forger de fausses données depuis et vers des canaux de communication. Comme cela se fait à l'aide d'un banc de test au lieu de simulations numériques, toutes les limitations de la vie réelle sont appliquées à l'attaquant. La technique *ARP poisoning* [119] est utilisée par l'attaquant pour intercepter les canaux et écouter les communications. L'attaquant a un mode de fonctionnement passif et actif. Pendant le *mode passif*, l'attaquant ne fait qu'observer, traiter et analyser les données sans modifier les informations contenues dans la charge utile des messages. Les données d'en-tête Ethernet, telles que les adresses MAC, sont néanmoins modifiées du fait de la compromission des tables ARP. Pendant le *mode actif*, l'attaquant commence à injecter des données dans la communication détournée. Cette injection, selon le modèle de l'attaquant, peut être un paquet rejoué ou généré par l'attaquant.

Attaque par rejeu : Les attaquants utilisent la technique d'empoisonnement *ARP poisoning* pour commencer à écouter la connexion (mode passif, du point de vue de la couche physique). Après avoir enregistré suffisamment de données, le *mode actif* commence. Les attaquants injectent les anciennes données capturées. Avant de commencer à perturber le système physique, l'attaquant effectue l'attaque de rejeu entre les capteurs et le contrôleur. Une fois rejoués les paquets vers le contrôleur, le système physique est perturbé en falsifiant des données entre le contrôleur et les automates.

Attaque par injection : Avant de lancer cette attaque, l'attaquant écoute les connexions en utilisant le mode passif *de la couche physique*, et analyse les données afin de déterminer la dynamique du système. Ceci permet d'échapper au détecteur d'authentification basé sur la signature. Une fois le modèle du système déduit, l'attaquant commence à injecter des données correctes dans le canal de communication afin de contourner la signature d'authentification. Pour échapper au détecteur, l'attaquant calcule l'effet de la signature dans le système et tente d'annuler la capacité du détecteur à détecter les changements dans le signal de retour. Deux techniques différentes sont mises en œuvre : 1) un filtre adaptatif non paramétrique, afin de mettre en œuvre la technique d'évasion présentée dans la section B.2, *une attaque cyber-physique non-paramétrique* ; et 2) méthodes autorégressives, telles que ARX et ARMAX, afin de mettre en œuvre la technique d'évasion présentée dans la section B.3, *une attaque cyber-physique paramétrique*.

Le défi de mettre en œuvre ces deux adversaires consiste à synchroniser la sortie des adversaires lors du démarrage de la phase d'attaque. Étant donné que la cible des adversaires est de prendre le contrôle du système, il faut que les données envoyées au contrôleur puissent correspondre à l'état actuel du système et à la corrélation correcte de la signature afin d'éviter d'être détecté.

B.4.3 Détection d'attaques et d'anomalies

Comme expliqué dans la section B.2.2, la métrique g_t est un opérateur qui quantifie la différence entre la sortie du modèle paramétrique et la sortie réelle du système. Une augmentation de g_t signifie que le système ne se comporte pas ni ne réagit pas à la signature comme prévu. Par

conséquent, le système risque d'être attaqué. La valeur de g_t est calculée pour chaque itération et comparée aux valeurs de certaines itérations précédentes. Pour éliminer les faux positifs, nous avons mis en œuvre dans le contrôleur à distance un algorithme pour séparer les failles des attaques ou des pannes graves. L'algorithme avertit l'opérateur uniquement lorsqu'une intervention réelle est requise, en séparant les failles (par exemple les événements de latence ou d'imprécision sur le capteur) des attaques intentionnelles. Pour chaque échantillon reçu, le contrôleur à distance analyse g_t. Si g_t dépasse consécutivement (plus que la durée d'une *fenêtre* pré-définie) un seuil donné, il déclenche une $alerte$.

En utilisant cet algorithme, le détecteur peut signaler les risques potentiels, en fonction des valeurs d'impact qualitatives [111]. Parallèlement à ces valeurs, il déclenche des alertes à l'opérateur chaque fois que les événements sont susceptibles d'être des attaques intentionnelles. Les alertes signalées, en utilisant des valeurs de taille de fenêtre appropriées au système spécifique, sont supposées être déclenchées assez tôt, par exemple, avant d'atteindre le *niveau critique*, pour permettre aux opérateurs de sécurité de traiter les informations avant de prendre les contre-mesures nécessaires – c'est-à-dire la sûreté des systèmes est supposée avoir une priorité plus élevée par rapport à la sécurité.

B.4.4 Résultats expérimentaux

En utilisant le banc de test défini précédement, nous avons analysé le détecteur avec la signature stationnaire, la signature non stationnaire et la stratégie définie dans la section B.3.1. En utilisant la signature stationnaire, nous pouvons souligner que l'attaque de rejeu est le scénario le plus détectable, avec un taux de détection d'environ 40%. L'attaquant non paramétrique a un taux de détection inférieur, d'environ 18%. Ce résultat est attendu, comme le suggèrent les conclusions théoriques et les simulation présentées (cf. section B.2). L'attaque paramétrique utilise l'approche d'identification du système la plus solide. Ces attaques peuvent échapper au processus de détection si elles réussissent à identifier correctement les attributs du système. En termes de résultats, ils conduisent au taux de détection le plus bas d'environ 12%.

Nous devons noter également que le *temps moyen de détection* d'une attaque de rejeu est le plus lent de tous les scénarios analysés. Ce comportement est dû aux propriétés de distribution de la signature (cf. section B.2.2). Parallèlement, les attaques par injection (version paramétrique ou non paramétrique) sont détectées beaucoup plus rapidement que l'attaque de répétition. Ceci est dû à la période de transition requise par les attaquants pour estimer les données correctes avant de tromper le détecteur. Pour cette raison, si l'attaquant ne choisit pas le moment précis pour lancer l'attaque, le détecteur mis en place au niveau du contrôleur est capable de détecter les données injectées au début de l'attaque. En plus, les attaquants doivent également synchroniser leurs estimations avec les mesures envoyées par les capteurs. Dans le cas où le processus de synchronisation échoue, le détecteur identifie les données non corrélées et signale l'attaque.

En ce qui concerne les résultats du détecteur de la signature non stationnaire, nous pouvons vérifier que la performance obtenue avec cette signature est compatible avec les résultats obtenus dans la validation numérique (cf. section B.2). Nous montrons que l'attaque de rejeu et les attaquants non paramétriques ont un taux de détection plus élevé avec cette stratégie, d'environ 60% et 56% respectivement. Ensuite, les attaquants paramétriques ont une petite augmentation du taux de détection, de 12% à 16%. Il est intéressant de noter que le *temps moyen de détection* diminue par rapport au détecteur de signature stationnaire. De même, le nombre de faux négatifs diminue, ce qui augmente la précision de détection de la stratégie contre les adversaires implémentés. Cependant, les faux positifs avec cette stratégie augmentent par rapport au détecteur de signature stationnaire. Cela signifie que, dans un banc de test réel, la perte de performance du système est plus importante, car le nombre de faux positifs augmentent de $1,35\%$ à $4,63\%$, avec la même sensibilité que la stratégie précédente et une signature non stationnaire.

Concernant les résultats obtenus avec la stratégie PIETC-WD, nous pouvons souligner que : (1) un système qui utilise uniquement le mécanisme de détection basé sur la signature contre les attaquants paramétriques a un taux de détection inférieur, d'environ 12%. Cela est possible car les attaquants peuvent échapper au processus de détection s'ils réussissent à identifier correctement les attributs du système ; et (2) un système qui utilise la stratégie proposée dans la section B.3.1 a un taux de détection plus élevé, d'environ 75.25%. Dans ce scénario, le taux de détection augmente, confirmant les résultats théoriques et de simulation rapportés dans la section B.3. Le ratio de faux négatif diminue, passant de $88,60\%$ à $38,66\%$. En termes de faux positifs, les deux scénarios présentent des résultats similaires, mais la stratégie PIETC-WD en génère environ $3,9\%$ de plus. Le temps entre le début de l'attaque et le moment où l'attaque est détectée par le contrôleur à distance prend plus de temps avec la stratégie PIETC-WD, puisque la signature de détection gérée par le contrôleur à distance suit une loi stochastique. Par conséquent, nous confirmons que la stratégie PIETC-WD augmente les performances de détection, au détriment du temps utilisé pour la détection.

B.5 Conclusion et futurs travaux

Cette thèse est basée sur le postulat que, dans un système cyber-physique, les adversaires peuvent écouter et manipuler des informations afin de perturber les propriétés de disponibilité et d'intégrité du système. Les adversaires peuvent utiliser des techniques de la couche cyber et de la couche physique, d'abord pour contrôler les couches du réseau, puis pour perturber les périphériques physiques. La combinaison de ces techniques peut générer des attaques furtives, permettant d'échapper à la détection. Les attaques contre ces systèmes peuvent affecter des personnes et des environnements physiques.

En termes de contributions, nous avons commencé cette thèse en examinant les technologies existantes sur des environnements cyber-physique du point de vue de la sécurité des TIC traditionnels. L'état de l'art a été complété par trois contributions principales. Tout d'abord, une première contribution a consisté à revisiter les approches de protection liées aux signatures stationnaires,

en les transformant en un processus adaptatif capable de couvrir un nombre plus large de modèles d'adversaires. Deuxièmement, nous avons étendu le détecteur de signatures résultant, utilisé comme attestation physique dans la couche cyber, en ajoutant une stratégie décentralisée pour étendre l'approche à plusieurs éléments d'un environnement cyber-physique (pas seulement des contrôleurs, mais aussi des capteurs et des actionneurs). L'idée est de distribuer le processus de détection sur tous ces éléments avec des capacités suffisantes pour identifier et gérer la dynamique du système, afin d'identifier les actions malveillantes en plus des failles accidentelles et des erreurs. Troisièmement, nous avons validé toutes nos propositions en les intégrant à un banc de test SCADA. Ce dernier a été mis en œuvre en utilisant des protocoles SCADA utilisés dans l'industrie (par exemple, MODBUS et DNP3) et des périphériques embarqués basés sur linux. Il nous a permis de tester et de valider les performances de sécurité de nos propositions. De plus, plusieurs adversaires capables d'attaquer des scénarios représentatifs ont été fournis pour compléter les simulations numériques.

En matière de perspectives pour la future recherche, plusieurs actions restent à faire. Cette thèse a traité, avec une portée limitée, certains défis sur la protection dans le domaine en accordant une attention particulière à la détection d'actions malveillantes cachées ou combinées avec des anomalies et des accidents. Néanmoins, les systèmes cyber-physiques englobent de nombreux autres domaines qui doivent être gérés ensemble afin d'améliorer leur résilience aux attaques et aux mauvais usages.

Dans cette optique, une première perspective comprendrait une analyse plus approfondie de l'impact sur la performance de notre processus de protection décentralisée. Sinon, à la suite du modèle décentralisé présenté dans la thèse, d'autres recherches restent à mener afin d'analyser le niveau de sécurité de la nouvelle approche, l'impact de la nouvelle construction en termes de performance du réseau, ainsi que la performance du système de contrôle. Egalement, des recherches afin de décentraliser totalement la stratégie de protection qui a été lancée dans cette thèse, ainsi qu'une combinaison appropriée des couches cyber et contrôle-physique suggérée dans notre travail pourraient être développées vers une nouvelle génération de SIEM (Security Information and Event Management) cyber-physique, capable de corriger correctement les incidents de sécurité entre couches.