UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA
FACULTY OF ENGINEERING

DOCTORAL THESIS

Cyber-Physical Security of SCADA Systems Against Physical Faults, Cyber Threats and Generic Malicious Attacks

A thesis submitted in fulfilment of the requirements
for the degree of
Doctor of Philosophy
in
Biomedical Engineering

Author:
Eng. Estefanía ETCHEVÉS MICIOLINO

Supervisor:
Prof. Roberto SETOLA

Coordinator:
Prof. Giulio IANNELLO

Reviewers:
Prof. Stephen WOLTHUSEN
Department of Mathematics
Royal Holloway, University of London, UK

Prof. Javier LOPEZ
Head of Computer Science Department
University of Malaga, Spain

March 21, 2016

Estefania Etcheves Miciolino

Firmato digitalmente da Estefania Etcheves Miciolino
ND: cn=Estefania Etcheves Miciolino, o, ou, email=e.etcheves@unicampus.it, c=IT
Data: 2017.03.27 11:01:06 +02'00'

UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

Abstract

Faculty of Engineering

Doctor of Philosophy

Cyber-Physical Security of SCADA Systems Against Physical Faults, Cyber Threats and Generic Malicious Attacks

by Eng. Estefanía ETCHEVÉS MICIOLINO

As Critical Infrastructures are becoming more complex and vital for modern societies, their management, monitoring and protection becomes of paramount importance, specially in safety-critical situations. Several studies are being carried out in the last years, having the objective of enhancing the security level and highlighting any vulnerability of these. It is clear that tests cannot be performed directly on real infrastructures due to security and safety issues, hence the development of realistic emulated environments becomes essential. Moreover, the scientific community has often considered and studied separately the cyber and physical domains constituting these complex systems, whilst it is essential to consider the overall environments, for example analyzing how cyber events may affect the operative condition of the physical infrastructure, as well as how anomalies in the physical dimension may generate a critical situation with respect to the Industrial Control Systems' monitoring architecture.

Having this problem in mind, a wide study on Industrial Control Systems is carried out, analyzing their constituting components, evolution and vulnerabilities. It is then followed by an analysis of the state-of-the-art on the diagnosis of physical faults and an overview of the cyber threats that these systems face. Making good use of these studies and being aware of the security concerns of cyber-physical systems, an innovative testbed reproducing the operation of a water infrastructure has been developed. In the designed system it is possible to introduce several types of physical faults and/or cyber anomalies, as well as to implement different configurations in order to test several scenarios and control strategies. The testbed is here described in detail, and its effectiveness regarding cyber-physical concerns is illustrated via several experimental tests.

Contents

List of Figures

List of Tables

List of Abbreviations

AD	**A**verage **D**egree of links in a network
ADU	**A**pplication **D**ata **U**nit
AICS	**A**utomatic **I**ndustrial **C**ontrol **S**ystems
ARP	**A**ddress **R**esolution **P**rotocol
BA	**B**arabási-**A**lbert Network Topology
BDD	**B**ad **D**ata **D**etection
CC	**C**lustering **C**oefficient
CERT	**C**yber **E**mergency **R**esponse **T**eam
CI	**C**ritical **I**nfrastructure
CPS	**C**yber-**P**hysical **S**ystem
DCS	**D**istributed **C**ontrol **S**ystem
DFDI	**D**istributed **F**ault **D**etection and **I**solation
DHM	**D**epartment of **H**omeland **S**ecurity
Dm	Network **D**ia**m**eter
DMS	**D**ecision **M**aking **S**ystem
DoS	**D**enial **o**f **S**ervice
DS	**D**ominating **S**et
EPA	US **E**nvironmental **P**rotection **A**gency
ER	**E**rdös-**R**ényi Network Topology
ES	**E**xpert **S**ystem
EWS	**E**arly **W**arning **S**ystem
FD	**F**ault **D**etection
FDAE	**F**ault **D**etection and **A**pproximation **E**stimator
FDI	**F**ault **D**etection and **I**dentification
FIT	**F**ault **I**solation **E**stimators
GOS	**G**eneralized **O**bserver **S**cheme
HMI	**H**uman-**M**achine **I**nterface
ICMP	**I**nternet **C**ontrol **M**essage **P**rotocol
ICS	**I**ndustrial **C**ontrol **S**ystems
ICT	**I**nformation and **C**ommunications **T**echnology
IDS	**I**ntrusion **D**etection **S**ystem
IED	**I**ntelligent **E**lectronic **D**evice
LAN	**L**ocal **A**rea **N**etwork
LFD	**L**ocal **F**ault **D**iagnoser
LTI	**L**inear **T**ime-**I**nvariant
MAC	**M**edia **A**ccess **C**ontrol
MBAP	**M**odbus **A**pplication **P**rotocol
MFAE	**M**inimum **F**unctional **A**pproximation **E**rror
MITM	**M**an-**I**n-**T**he-**M**iddle
MTU	**M**aster **T**erminal **U**nit
NICC	**N**ational **C**yber **C**rime **I**nfrastructure
NIST	**N**ational **I**nstitution of **S**tandards and **T**echnology

OS	**O**perating **S**ystem
PDE	**P**artial **D**ifferential **E**quation
PDS	**P**ower **D**ominating **Set**
PDU	**P**rotocol **D**ata **U**nit
PLC	**P**rogramable **L**ogic **C**ontroller
PLOD	**P**ower-**L**aw **O**ut-**D**egree Network Topology
RISI	**R**epository for **I**ndustrial **S**ecurity **I**ncidents
RMS	**R**oot-**M**ean-**S**quare
RP	**R**isk **P**redictor
RTU	**R**emote **T**erminal **U**nit
SCADA	**S**upervisory **C**ontrol and **D**ata **A**quisition **S**ystem
TCP/IP	**T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol
WS	**W**atts-**S**trogatz Network Topology

To my ohana…

Introduction

Critical Infrastructures have become an essential asset in modern societies and our everyday tasks heavily depend on their reliable and secure operation. These are expected to continuously and efficiently operate 24/7 for months or even years, and their overall life expectancy may be measured in decades.

Due to their vital importance and potentially devastating consequences in case of malfunctioning or unavailability, Critical Infrastructures have become attractive targets for both malicious agents and researchers in hunt of new challenges in security.

One strategy to enhance the protection of these systems, largely exposed in the further sections, consists in the employment of emulated environments where the different threats can be tested and their effects on the system can be therefore observed and analyzed, so to develop improved and innovative early detection and protection methods.

With this objective in mind, the effort of this work was focused on the study and implementation of methods and techniques to detect, generate and analyze the effects of attacks, faults and malicious events on a cyber-physical system. Specifically, the target system is an emulator of a water network, on which a proactive approach has been exploited to both the physical and cyber domains from a double point of view, the attacker and the protector, to highlight the main vulnerabilities, weaknesses and strong points of this specific type of Critical Infrastructure, many of which still hold true for other types of systems or in a more general context.

1.1 Critical Infrastructures

Part of the complexity of the modern social organization is due to the fast and continuous development of new technologies, mainly in the communications and information fields. These led to the creation of infrastructures always more complex and sophisticated. Critical Infrastructures (CIs) are defined in [1] as systems, resources and processes, whether physical or virtual, so vital to the countries that the disruption, damage or unavailability of such systems and assets, even partial or temporary, would have a debilitating impact on a nation's security, wellness, economy, public health or safety, or any combination of these matters. These are safety critical systems performing complex operations, spatially distributed, dynamic, time-varying, and uncertain. Examples of these infrastructures include the electrical power plants and the national electrical grid, oil and natural gas systems, telecommunication and information networks, transportation networks, water systems, banking and financial systems, health-care services and security services. Thereby, the term "Critical" is well tailored for such infrastructures, as a malfunction or service unavailability may carry huge social and economical consequences with it, and can reveal to be catastrophic in the worst cases for both human beings and the environment.

During the last decades, the services provided by such infrastructures have increased, enhancing life standards and quality of the population, simplifying and accelerating the main daily activities, and becoming of vital importance in modern societies.

In addition, citizens expect their continuous availability, 24 hours a day, 7 days a week, but, at the same time, the services provided are wanted to be low cost. Although these requests are most often met, as any other physical system the components of these processes may occasionally fail, compromising their normal operation and reducing their performance and reliability. Such failures might be due to environmental or natural disasters, accidental failures, human errors or malicious attacks. Moreover, failures may also take place in the monitoring and control systems, as a consequence of hardware or software faults, cyber-attacks, power outages, etc. [2]. Such events are unlikely to occur, whilst may have a huge impact on everyday life and well-being.

An additional complication for the CIs management and protection has arisen as a consequence of their increasing size and geographical distribution. Moreover, the development of new technologies, employed in combination with outdated infrastructures, has led to an actual revolution of the way CIs are projected and managed, to which the performance of new and more complex tasks is required. On the other hand, the pervasive introduction of Industrial Control Systems (ICSs) components and the consequences of globalization, together with the large increase of infrastructures interdependencies, have given rise to new problems in the detection of criticalities in the security. As a consequence, unprecedented and dangerous vulnerabilities that may harm the whole infrastructure system are taking place. In addition, such Critical Infrastructures interact in ways that appear to be clear, but generally hide very complex interactions which may give rise to dangerous domino effects, with related cascading failures in the whole infrastructures network [3]. Thereby, a failure, both accidental or malicious, may spread in an unpredictable way, amplifying its negative consequences and damaging an unpredictable number of citizens.

The importance of the physical protection of such infrastructures and their assets has been highlighted in [4]. However, their protection should focus not only on the physical elements constituting the CI, but also on their cyber layers (virtual elements). In addition, their correlation is to be taken into account, as the consequences of one aspect would largely influence the proper operation of the other. According to the European Programme for Critical Infrastructure Protection (EPCIP) [5], these infrastructures can be categorized into thirteen main strategic sectors, which must function 24/7:

- **Water:** provision, quality control, stemming and quantity control.
- **Energy:** oil and gas production, refining, treatment and storage, electricity generation, transmission and distribution.
- **ICTs:** information system and network protection, the Internet, provision of fixed and mobile telecommunication, radio communication and navigation, satellite communication, broadcasting.
- **Food:** provision, safety and security.
- **Health:** medical and hospital care, medicines, serums, vaccines and pharmaceuticals, bio-laboratories and bio-agents.
- **Financial systems:** banking, payment services and government financial assignment.
- **Civil administration:** government facilities and functions, armed forces, civil administration services, emergency services, postal and courier services.

- **Public, legal order and safety:** maintaining public and legal order, safety and security, administration of justice and detention.
- **Transport:** road and rail transport, air traffic, border surveillance, inland waterways transport, ocean and short-sea shipping.
- **Chemical industry:** production and storage of dangerous substances, pipelines of dangerous goods.
- **Nuclear industry:** production and storage of nuclear substances.
- **Space:** communication and research.
- **Research facilities**.

Moreover, it is requested they are opportunely monitored and supervised by control systems to ensure the correct performance of the processes and operations. The functional and operational features of control systems, better exposed in the following sections, let us define them Critical Infrastructures, as any physical or virtual disruption may have devastating consequences for continuity and availability of service and business.

1.1.1 CIs Interdependencies

Critical Infrastructures are becoming more and more interoperable and interdependent as technology progresses and the demand for better services increases. Specifically, utility sectors deeply rely on one another to operate, and in same cases are co-located at the same geographic location. In contrast, the knowledge of human operators and stakeholders is becoming more and more sector-specific, knowing primarily their own system, with little or no knowledge of the systems interconnected/interdependent with their own. Thus, representing the behavior and the characteristics of interdependencies between Critical Infrastructures is a must, in order to assess the risk of multiple disruptions and domino effects, and in order to provide adequate policies and countermeasures to react to vulnerabilities, failures, or even intentional attacks.

Individual Critical Infrastructures can be described as complex adaptive systems, since they are complex collections of interacting components in which change often occurs as a result of a learning process [6]. However, Critical Infrastructures do not exist in isolation of one another since they are highly interconnected and interdependent in complex ways, such as physically and through data and information sharing. For example, telecommunications systems require electricity for their operation while for the generation of electricity, power plants requires fuel supplied from oil and natural gas systems, and so forth. In the general case, multiple infrastructures are connected as a "system of systems" and interdependencies are considered between them in order to capture and understand their operational characteristics. The importance of understanding interdependencies between infrastructures is highlighted most often when infrastructures are experiencing catastrophic natural disaster or terrorist attacks and the countries (or major cities) attempt to respond and recover from severe disruptions [7]. Because of the interdependencies between Critical Infrastructures potential failures in one infrastructure may lead to unexpected cascade failures to other infrastructures that may have severe consequences and apart from property damage, may even lead to loss of life. Indeed, as stated in [8], interdependecy among CIs is very much about the security and assurance of a given CI, and as a result information about interdependency is

not commonly available and is considered highly sensitive by the people who possess it. Moreover, little to none evidence has been reported by the infrastructure owners themselves, and most work on this area has been done by academia or not-for-profit associations or government entities.

Because of the large scale and the great importance of the various Critical Infrastructures, real practical solutions, such as introducing scenarios and triggering events for analyzing the interactions between Critical Infrastructures, are not feasible. Thus, CIs (and their interdependencies) models are largely employed to understand how the infrastructures interact with each other, as well as for vulnerability assessment, failure analysis, information generation, anomalous event mitigation/prevention and to develop self-healing strategies.

In recent years, infrastructures have reached a high degree of interoperability, mainly due to the pervasiveness of ICT technologies; in fact, cyber interdependency potentially couples an infrastructure with every other one, in spite of their nature, type, or geographical location [6]. Moreover, because of the enormous growth of the complexity of each infrastructure, the skills of technicians, operators, and stakeholders are becoming more and more sector-specific. Therefore, while it is often possible to retrieve exhaustive information about the behavior of any single infrastructure and its elements, cross-infrastructure interdependencies are often implicit, hidden, or not well understood by the same stakeholders. These interdependencies are mostly highlighted during catastrophic natural disasters or under terrorist attacks, when an attempt to respond and recover from severe disruptions takes place [7].

Due to the intrinsic characteristics of the various CIs, their large scale, complexity and vital importance, real practical tests and solutions for the analysis of the effects are not feasible. Thereby, the modeling, emulation and analysis of these systems are the best and safest methods that can be deployed nowadays to study, develop and provide solutions for their monitoring and control, as well as to understand their functioning in the presence of anomalous events, their influence on other systems, and how they can recover from such conditions.

1.2 The FACIES Project

Considering all the mentioned criticalities in CIs, the principal objective for their protection is the identification and assessment of possible vulnerabilities in such systems, by carrying out tests in operative environments related to CIs. As this cannot be carried out in real CIs due to the high security risks that may arise, the cyber-physical emulation of a CI has been included in the FACIES EU Project, which aims at reproducing its operation, characteristics and interdependencies with other CIs. As a consequence, a complex cyber-physical system has been developed to manage all the different scenarios needed to perform a realistic analysis of the behavior of the CI, and is deeply exposed in Chapter 6. As requested by the EU Commission, this testbed has been developed as a cyber-physical experimental resource for research, academia and educational activities.

Although most of the studies on CIs have mainly focused on infrastructures related to the generation and distribution of electricity, FACIES has turned its attention to water supply and distribution systems. The reason for such a choice ranges from the intrinsic relevance played by the water system in daily life, to the peculiarities of the

water domain, which is characterized by highly non-linear behavior. The same reasons have recently driven several research teams to focus their attention on water systems, e.g. [3], [9].

It is well known that water infrastructure may suffer structural and hydraulics faults and failures in the constituent components, which can result in water leakages and pressure loss compromising the normal operation of the network [10], [11]. Moreover, there is an increasing awareness about criticalities related to malicious contamination events, as highlighted in [12]. To conclude this worrying landscape, such infrastructures are more and more exposed to cyber-attacks, which aim at compromising their normal operation and reducing their performance and reliability. This has been demonstrated, for instance, by the malicious action performed in 2000 against the water infrastructure of Maroochy Shire (Australia) [2], [13], and the 82 incidents against the water infrastructures in the US during the 2011, declared to the Industrial Control Systems Cyber Emergency Response Team (ICS-CERT), which represent the 41% of all the reported cyber-attacks against ICS [14].

On the other hand, failures may also occur in the monitoring and control systems, as a consequence of hardware or software faults, cyber-attacks, power outages, etc. Considering the utmost importance of the water infrastructure for modern societies, it is necessary to implement efficient failure detection schemes, which aim to fasten then time for detection and identification to reduce the consequences of such circumstances.

1.3 System Emulators

As deeply analyzed in [15], strong effort has been made to develop new algorithms for monitoring, control, and security of Critical Infrastructures, generally based on computational intelligence techniques and the real time processing of data received by networked embedded systems and sensor/actuator networks located throughout the system. Despite the utmost importance of these systems and their emerging criticalities, most studies approach the cyber and physical domains separately, not considering the relationships that may arise among them. Therefore, there is the need to develop a new class of simulation environments, specifically designed to reproduce the complex phenomena at the base of infrastructures interdependencies and vulnerabilities, as highlighted by the DIESIS EU Project [16]. The concept of *cyber-physical system* arises, which are the result of the interconnection and interaction of cyber (computation) and physical elements in a system. Sensors, processors and networks monitor and collect data from the physical processes, which can be controlled through the analysis and use of such data, taking appropriate actions to guarantee on one hand the service to be provided, and on the other the stability and security of the system. Unfortunately, there are some limitations which are implicit in the use of non-real environments. Hence, there is the need to develop testbeds able to reproduce with adequate accuracy the peculiar phenomena and properties of the systems being studied. Such testbeds represent a trade-off between a wide number of parameters, among which fidelity, complexity, reliability, repeatability, accuracy, safety, feasibility and development costs. In fact, the main drawback when studying CIs is the actual complexity of such systems and the difficulty to create a scaled-down reproduction, which generally leads to the emulation of the communication and control networks and the large use of software simulation to reproduce the behavior of the infrastructures to be studied. This becomes particularly evident when analyzing the core of the water infrastructure related to which, in addition to the presence of complex and non-linear dynamics, very few studies have

Testbeds	*Simulation*	*N. Tanks*	*Flexibility*	*Control*	*Monitoring*	*PLC*	*SCADA*	*Fault generation*	*Fault Diagnosis*	*Cyber Security*	*IC Interdependence*
Testbed 1 [17]	✓		✓	✓	✓	✓				✓	✓
Testbed 2 [18]	✓	1		✓	✓	✓	✓				
Testbed 3 [19]		2	✓	✓							
Testbed 4 [20]		4	✓		✓						
Testbed 5 [21]	✓		✓	✓		✓					
Testbed 6 [22]	✓	3			✓			✓	✓		
Testbed 7 [23]				✓		✓	✓	✓	✓		
Testbed 8 [24]					✓		✓		✓		
Testbed 9 [9]	✓			✓	✓	✓	✓	✓		✓	
FACIES	✓	5	✓	✓	✓	✓	✓	✓	✓	✓	✓

TABLE 1.2: Testbeds characteristics comparison

approached the problem with a cyber-physical perspective, taking into account healthy and anomalous situations. For this reason, the FACIES testbed presented in this work represents a significant advance towards such direction, as it comprises not only an emulator of a real water transmission system, but also the related control architecture, constituted of sensors, actuators, PLCs, SCADA, the underlined communication network, and other modules for the monitoring and analysis of several cyber-physical aspects.

An analysis of a wide number of emulated environments in literature has been carried out, which main results are summarized in Table 1.2. More specifically, the study was focused on systems which implement simulated parts, considered the presence of water tanks in the specific network and the flexibility in the configuration of the architecture, the use of control strategies in the system and the presence of interfaces for data reporting and monitoring, or the implementation of a SCADA system for higher complexity, the use of control devices, specifically PLCs. Lastly, an analysis of the capability to introduce faults in the system (manually or by means of exogenous signals) and to eventually detect them has been carried out, as well as the presence of cyber security resources and the consideration of the interdependencies that may rise with other CIs.

In [17] the main requirements for developing a useful testbed are exhaustively described, and a comparison between a number of emulated and simulated environments proposed in literature is carried out. The testbed there described, indicated as *Testbed 1* in Table 1.2, has been developed by the authors to carry out realistic real-time experiments on a real network with heterogeneous simulated infrastructures, which facilitates the study of faults propagation and service disruptions caused by cyber-attacks. An ICS testbed, based on the Emulab software, is combined to software simulators for the physical components (e.g. Simulink models), allowing to carry out a simplified

study of the combined effects of cyber-attacks for interdependent Critical Infrastructures. The proposed models for CIs are simplified and an analysis on physical faults is not considered.

On the topic of water systems, several studies have been carried out using experimental setups, many of which focus on the support for control design, with specific attention on the water level control in single or multi-tank systems. A first step is made in [18], where the development of a testbed for the single water tank system control is described (*Testbed 2* in Table 1.2), with the aim of reproducing an automatic continuous cycle of filling and draining. In [19], the control of a couple of tanks is addressed, first exposing the nonlinear model for the dynamics of water level in the single and double tank systems. For the experimental part, the CE105 coupled tanks system (exhaustively described in [25]) is then introduced, and experimental results for level control are provided (*Testbed 3* in Table 1.2).

In [20], a quadruple-tank system is proposed as an experimental framework for performing multivariable control design, highlighting the physical interpretation of the zero positions in the real axis, which is related to the position of the valves. This testbed is referred to as *Testbed 4* in Table 1.2. However, such system does not include ICS components and it is focused only on the case of healthy operation. A step forward, is taken in [21], indicated as *Testbed 5* in Table 1.2, where a number of technologies to perform water level control is studied, and a solution is proposed in which wireless sensors and a PLC are used, even if sensors and actuators are simulated. This work is focused on the control problem, and an algorithm for filling and emptying control is proposed.

Adaptech has developed Amira DTS200 [26], a laboratory setup consisting of a three-tank system, which has been widely used for experimental purposes of different kinds, among which level control and fault detection and identification. In the wake of the latter, [22] presents an observer with a corrective term based on a second-order sliding mode control algorithm, developed for fault detection and identification purposes. Such model is then tested both in a simulated environment and in an experimental set-up of a three-tank system, which is indicated in Table 1.2 as *Testbed 6*. The PI control implemented for water level control in the tanks has been developed in Simulink, and faults reproducing malfunctionings of the valves are induced by the use of adjustable manual valves. However, it is unable to manage cyber events, and the water system studied is considered as stand-alone, i.e. the interdependencies that may rise with other infrastructures are not considered.

A more complex analysis of a water system as a Critical Infrastructure is carried out in [23]. A multi-agent architecture for the simultaneous detection, isolation and estimation of water withdrawal and hardware faults in a water canal network is developed and tested on the experimental water delivery canal held by the Hydraulics and Canal Control Center (NuHCC) of the Évora University in Portugal. The canal network is divided into several subsystems constituted by the single canal pools and the corresponding gate, and single distributed agents for fault diagnosis are then assigned. The residuals obtained from the comparison between the sensor measurements and the output of a model of the system are evaluated for fault diagnosis purpose.

Similarly, Whittle *et al.* [24] carry out studies of a wireless sensor network for near-real-time monitoring of a water distribution network in Singapore (*Testbed 8* in Table 1.2). Such network is developed to monitor hydraulic and water quality parameters, as well as water levels and sewer out flows. The data gathered by sensors is transmitted to a lab-based server through the Internet, where it is stored, processed

and displayed on a web platform for on-line hydraulic modeling, leak and burst detection and operational event analysis.

To conclude, in [9] a performance analysis of a PI controller and a model-based diagnostic scheme under a class of stealthy deception attacks is carried out. Such attacks aim to remotely withdraw water from automated canals systems. Some assumptions on the knowledge possessed by the attacker are made, and it is shown how the existing diagnostic tools for random fault detection are not generally capable to detect cyber-attacks. A test is then carried out in the Gignac canal system in Southern France, in which the sensor measurements sent from the SCADA to the PI controller were opportunely tampered. It is referred to as *Testbed 9* in Table 1.2.

As previously highlighted, the availability of testbeds reproducing ICS and physical components is of paramount importance. However, few have been designed to analyze the effects of faults and attacks in the system, and even less deploy the effectiveness of automatic diagnosis tools. Indeed, to the best of our knowledge, no testbeds have been designed for the analysis of situations in which anomalous events take place concurrently in the physical and cyber dimension, and the evaluation of the impact of dependencies and interdependencies with other ICs has been lightly faced.

After this first overview about Critical Infrastructures, their importance and vulnerabilities, the main issues and threats these cyber-physical systems face nowadays, and how emulation environments may help to improve their security and protection, the next chapters focus on a theoretical approach related to the most recent studies and well-known problems/solutions of the different areas of interest presented in this dissertation. Starting from a general outlook about the Industrial Control Systems in Chapter 2, the focus is then moved to the more specific aspects related to the physical and cyber domains. Specifically, Chapter 3 deals with the Fault Diagnosis Problem, while in Chapter 4 the most known cyber-attacks and their impact in SCADA systems are faced. Then, an original study about controllability of complex networks undergoing attacks in various scenarios is carried out in Chapter 5. The presented methodology is considered a useful approach for both enhancing the protection by highlighting the main vulnerabilities of a network and for evaluating the effects of different types of attacks. The core of the dissertation is developed in Chapter 6 and Chapter 7. Firstly, the FACIES testbed is exposed in detail in Chapter 6, starting from its design and presenting a detailed description about the main features and configurations implemented. This is then followed by the presentation of the experiments carried out and their results in Chapter 7. Again, both the physical and cyber domains have been singularly explored, together with the responses from the detection modules developed, and an analysis about their interactions. Finally, this dissertation ends with some concluding remarks and reflections on the improvements and further developments that could be taken into account as future work.

Chapter 2

Automatic Industrial Control Systems

Industrial networks, aside their large diversity and the wide number of markets they serve, are mainly composed of several distinct areas, among which the business network, the business operations, the supervisory network, and the process and control networks. Each area has its own physical and cyber security considerations, policies and concerns. Although similar to standard information networks, industrial network security presents several unique challenges. Thereby, to study such systems and enhance their protection, it is necessary to gain deep knowledge on how they have evolved in time, how they are currently composed, and the main vulnerabilities that have risen in the last years.

2.1 Industrial Control Systems

In the last decades, the complexity and automation degree of industrial processes has been largely enhanced, due to the increasing demand for system performance, product quality and cost efficiency. Such development required the improvement of system security, reliability and dependability. Industrial Control Systems (ICSs) are complex systems that perform defined tasks as part of an industrial production process, deployed for the monitoring and supervision of remote sensors, managing automation operations and recording sensitive data measurements.

2.2 SCADA Systems

As previously mentioned in Section 1.1, it is mandatory for the CIs to operate 24/7, providing the required services, and their operation is monitored and supervised by systems commonly known as Supervisory Control and Data Acquisition (SCADA) systems, which are part of the ICSs category. Monitoring and control is generally achieved deploying networked intelligent agent systems with sensing and actuator capabilities, besides communication, computing and data processing skills. Specifically, sensors are used to collect monitored data related to physical quantities (measurements), while actuators are used to perform the desired actions based on the sensor measurements and specific control algorithms.

SCADA systems are composed of hybrid integral systems in which a set of control processes may be widely distributed over large geographic locations, but any information has to be centralized at a single point, the SCADA Centre. These networks perform specific functions related to automation, supervision and management of sensitive information. Actually, they are used to gather real-time data, monitoring equipment and

controlling processes in most of the public utilities. In addition, in the SCADA Centre are located all the databases that serve as main units for logging, the main processors, servers and operator consoles or Human-Machine Interfaces (HMIs), which allow operators to visualize processes, read specific physical parameters or alarms received from the substations, control operations and data streams, and transmit commands to field devices. Such HMIs are opportunely designed to offer the suitable overview representation, through map-boards, of the entire system and its network architecture, and offer the means to interact with the system through commands or to retrieve measurements from sensors.

The SCADA Centre is responsible for processing the information received by the local and remote substations, as well as for issuing control commands to the controllers. Thereby, all the control commands and messages are opportunely sent to specific field devices, which are usually located in the remote substations or industrial plants. The remote substations are control sub-networks which comprise field devices, collectors and communication interfaces able to interpret ingoing and outgoing data, execute control actions in the field, and to send information to the SCADA Centre. These devices, among which the most known are the Programmable Logic Controllers (PLCs), the Remote Terminal Units (RTUs) and the Intelligent Electronic Devices (IEDs), are directly connected to sensors and actuators which perform the required actions of the process. In addition, they are endowed with interfaces able to establish connections with other substations, RTUs and field devices through serial or TCP/IP communication systems. More specifically, PLCs and RTUs are able to interface physical sensors, monitoring devices, stations and special purpose systems towards Distributed Control Systems (DCSs) or SCADA systems, by receiving, storing and transmitting telemetry data (usually measures of physical quantities) to a master system. The main difference between PLCs and RTUs is that RTUs use wireless communication and is therefore more suited for wide geographical telemetry, whereas PLCs are generally preferred for local control. Moreover, while RTUs do not usually provide any control logic for the process, as algorithms or control loops, PLCs contain software for controlling the process, allowing to act on physical devices according to defined algorithms. RTUs and PLCs acquire measurements and send them to a centralized computer system for further analysis. Both deal with raw data, usually real-time physical measurements and control information as commands, devices information, and similar.

Based on the data received from the sensors and communicated to the controllers, appropriate data processing methods are employed to obtain meaning and knowledge, which is analyzed to control the system and perform fault diagnosis and accommodation as soon as possible. Due to the huge volume of data obtained from these large-scale CIs and to its different characteristics it may have, such task has become more and more complex. The controller module receives information from the system, which is then processed and exploited to generate commands, warnings, alarms, and communicates with remote agents to control the system via specific actions.

2.2.1 SCADA Evolution

Since their first introduction in the 60s, three main architecture categories have been defined to highlight the technological evolution of control systems, namely *monolithic*, *distributed* and *networked* systems [27]. The step was to exploit the existing technologies and communications systems to open the isolated process networks, thereby providing

better monitoring capabilities, performance in the control and availability of controlled infrastructures.

The first *monolithic* SCADA networks were characterized by a centralized control in a mainframe system, configured as the primary node, while a second redundant mainframe acted for operability recovery needs. Both systems had to register critical data streams, manage the system and make decisions to efficiently coordinate the monitoring processes developed in the whole network. Thereby, in the substation a number of RTUs had to perform measurements and control actions over the field devices through the available input/output interfaces, exchanging data with the central system deploying serial automation protocols, e.g. Modbus serial.

The second SCADA generation, defined as *distributed*, was characterized by the integration of new IP-based technologies for the monitoring of distributed processes in different network components. Thereby, distributed database servers where deployed to store measurements and alarms, and the latency to enter in recovery mode of the devices was largely improved as any active device in the network could immediately cover the functionality of another one. The communication to remote substations was established with large Local Area Networks (LANs), controlled by the Master Terminal Units (MTUs) in the central system. The RTUs employed in the substations were more evolved from a technological point of view, more intelligent and autonomous with respect to the previous.

The *networked* generation was born with the aim of breaking with the isolation of the previous generations, achieved with the inclusion in the network of open connections through the TCP/IP protocol. These connections allow remote monitoring in near real-time, peer-to-peer communication, multiple sessions, concurrency, maintenance, redundancy, security services and connectivity. At the same time, the evolution of RTUs provided hierarchical and inter-RTU communications using the TCP/IP protocol, through both wired and wireless interfaces, web services, management and forwarding to other remote points. Moreover, the RTUs became data concentrators for large data streams, and gained the ability of autonomously and remotely reconfigure/recover parts of the system. This evolution led also to the standardization and implementation of several IP-based communication protocols, as Modbus/TCP, which allowed almost real-time control and the use of public infrastructures, such as the Internet, for the SCADA transmissions and its commands and data streams.

Although the positive business benefits gained from such development, this evolution brought with it two main concerns:

- Process control systems were traditionally closed systems designed for for functionality, safety and reliability, where the main concern was the physical security. Increased connectivity via standard IT technologies has exposed them to new threats, which they are no properly equipped to deal with (e.g., worms, viruses, hackers). As these process control networks continue to increase in number, expand and connect, so the risks from cyber threats continue to escalate.
- Commercial off the shelf software and general purpose hardware is being used to replace property process control systems. Such technologies often do not match the uniqueness, complexities, real-time and safety requirements of the process control environment. Many of the standard IT security protection measures normally used with these technologies have not been adopted into the process control environment. Consequently, there may be insufficient security measures available to protect control systems and keep the environment secure.

Clearly, the architecture largely influences how the data obtained from the sensors can be used to meet the objectives of the system. In the centralized control architecture, all monitored information is processed at a single central entity, whereas in the distributed approach local networked controllers process the information themselves and communicate with neighboring agents so as to formulate the control action. Although centralized systems have complete information of the state of the network and can provide optimal solutions for some of the network functionalities, they do not scale, require large databases to store historical data, and process a very large amount of data.

As observed in [27], for the recent future ICSs might largely demand the use of wireless technologies and the Internet for control. Wireless technologies allow operators in the field to locally manage substations, providing mobility and coexistence with low installation and maintenance costs. Conversely, the Internet allows to remotely control the substations, so the SCADA Centre and the operators in the field can interact despite the geographical distances. The public communication infrastructure of the Internet offers Web solutions together with the flexibility for data acquisition and management, data dissemination, maintenance, diagnosis and interfaces to visualize data streams and resources in real-time [28]. Moreover, costs in terms of hardware, software, time, personnel and field operations can be significantly reduced by deploying open standards and open Web protocols, as HTML, HTTP or HTTPs.

Nevertheless, the use of the Internet gave rise to new security threats and reliability problems for the industrial systems, as intercepted communications channels, disruption of services, isolation or data alteration.

2.2.2 Vulnerabilities and Security in SCADA Systems

Critical Infrastructures suffer from several faults and security risks to the plant or to the control infrastructure. Specifically, the SCADA systems are particularly susceptible to attacks. As better exposed in [15], a *failure* in a CI, whether accidental or intentional, is a negative event influencing the inability to perform the intended function of infrastructures and subsystems. On the other hand, a *fault* in a component denotes an event that may be impossible to avoid, but can be dealt with by exploiting redundancy, so as to guarantee continuity in the infrastructure operation. In effect, the overall system requirement is its effective operation, although suboptimal, even when one or more of its constituent components have failed. However, the suboptimal functioning of a system may potentially lead to waste of energy or resources, or to high risks. Thereby, autonomous methods for the early detection, isolation and recovering from faulty conditions are desirable, especially in the Critical Infrastructures domain.

Over the last decade a wide number of threats have been registered in public databases, most of which have been carried out by malicious insiders, e.g. disgruntled or malicious members of an organization. The consequences could be devastating, since a failure or attack in CIs may trigger massive deficiencies in essential services, which may affect a city, a region, or even a country. As highlighted in [29], our inability to understand complex systems and modeling them through conceptualizing their component parts and security domains at the required decomposition level in which they can be described, evaluated and assessed, has led to the lack of understanding these systems and to cope with their risks.

In addition, a high number of misconceptions regarding SCADA systems lead to security problems. For example, these systems are still considered an isolated and

standalone network. Thereby, many system engineers have simply integrated the Internet or other shared communication mediums' components into the SCADA system with little or any regard on how to expand the network or how a node connected to the Internet could affect the security of the whole system. Moreover, it is largely believed that connections between SCADA systems and corporate networks are secure, whereas they are often linked, as largely described in [30]. Thus, access controls designed to prevent unauthorized access are minimal, and often inadequate, and members of the organization obtain access to unauthorized areas and email servers, and use insecure web services and protocols for the remote control. A security failure in the corporate network may lead to significant security risks in the whole system. Furthermore, deregulation has led to the a wide deployment of open access facilities, which led to a rapid rise in the potential vulnerabilities in corporate networks. Finally, useful information about the corporate network of a utility company is often available on the web, and may be used to trigger a more focused attack against the system, as exposed in [31].

It is also assumed that, in order to successfully perform an attack, an extensive knowledge on the SCADA system is required. On the contrary, due to the lacks in the security, it has been demonstrated that any individual with moderate computer programming skills and a network access has the sufficient means to break into a SCADA system. In fact, due to the primitive nature of SCADA systems, it is likely that these are way more vulnerable that a state-of-the-art personal computer. In addition, it is not to be neglected that companies employing SCADA technologies are usually desirable targets for cyber terrorists, who are more motivated, well organized and better skilled than a random individual.

Section 5.2 provides a method to represent complex large-scale networks as distributed systems, and the study carried out in [32], [33] is reported, which focuses on robustness for different network topologies undergoing attacks. It shows how an attacker with sufficient knowledge on the network can detect the nodes which attack would lead to the most damaging consequences. Conversely, it could be interpreted as an useful method to determine the critical nodes on the network to carefully protect.

As highlighted by [27], a major issue in the implementation of security systems is the lack of guidelines regarding such measures, and the fact that it is generally not cost-effective to actually implement them, they are often non-scalable and may negatively impact the operation of the infrastructure. Thereby, it is necessary to rigorously define security and access control policies, properly configure traditional security mechanisms, frequently carry out auditing and maintenance processes, authentication, authorization, and provide proper training to operators and personnel. However, this may not be enough, and it is necessary to configure intelligent management mechanisms to take over alarms and incidences efficiently, as well as to configure status management and anomaly prevention mechanisms. These preventive proactive mechanisms could feed Early Warning Systems (EWSs) to help systems to appropriately react to an anomalous event, and in the worst case to provide useful information for forensic procedures and recover protocols, based on specific methodologies, techniques, policies and standards.

To such end, it is necessary to consider the deep differences between ICT and SCADA systems, based on their security properties as described in ANSI/ISA-99.00.01-2007 standard. Specifically, SCADA systems have to provide hard real-time responses which are critical, generally ranging in milliseconds, whereas ICT systems have permissible time responses of seconds. Moreover, changes on the SCADA systems do not

happen often, while the opposite is verified in ICTs. Classic Intrusion Detection Systems (IDSs) are to be rethought so as not to disturb normal operations by increasing delays in the communications. Thereby, high-speed traffic analysis is to be achieved, as well as their functionality needs to be extended to monitor SCADA specific protocols and to take into account the specific operational context where they are employed. Moreover, SCADA systems present more physical security concerns due to the isolation and proprietary protocols historically used. Several similarities and common applicable security processes exist that can be exploited to enhance the security of the entire system, but the knowledge acquired in managing and securing ICT systems may be not so straightforward to apply on SCADA systems, and integration efforts and specific adaptations are required to employ security tools and best practices management. Indeed, the security issue should be enforced with good security policies, knowing the risks and threats, a security plan and implementation guidelines, enforcing the principles of least privilege, need to know and segregation of functions, the opening and sharing of security design instead of continuing to rely on security by obscurity, the classification of information, the implementation defense-in-depth, the exploiting of cryptographyc algorithms, protocols and products, and the consciousness of the human factor needs, especially behavior, awareness and formation. Security management is a continuous improvement process that for SCADA systems requires an extended and complementary approach beyond traditional ICT security processes.

As stated in [27], there are four areas of security controls need to be improved through further research and development:

- communications in SCADA systems, which need lower costs increased efficiency;
- enhanced SCADA protocols and networks strengthened by cryptography;
- monitoring and detection controls through firewalls, intrusion detection systems and other security modules, set up to ensure access policy compliance and detect suspicious behaviors;
- SCADA information classification.

In addition, intelligent response mechanisms to incidents are to be provided, so to avoid further damages due to improper collateral impacts. Specifically, the alarm management mechanism needs to be effectively improved. These shall assure reliability, identifying the most suitable operator for performing specific activities, and security, providing input information associated to operators and activities to other security mechanisms, such as auditing and forensic modules. Similarly, it is necessary to deploy detection mechanisms to alert security operators when an attack or anomaly is taking place on some of the components of the SCADA networks. For example, on the cyber domain three general approaches for attack detection are employed by the current IDS solutions [34]:

- *Signature based:* a set of *rules* based on know attacks is developed to find suspicious activity in the current data traffic of the SCADA network. A complete and updated knowledge on attacks behavior is needed, and so has to be the related set of rules.

- *Anomaly based:* the developed rules are based on the normal behavior of the system, obtained by models of the traffic, events, applications or messages transmitted. As a consequence, a good training on a stable scenario is required. In addition, these techniques are able to detect unknown attacks.
- *Specification based:* exploit the known vulnerabilities of the communication protocols by validating each submitted command to detect misbehaviors.

Generally, a combination of such methods is used to take advantage of the benefits of each. Actually, SCADA systems usually have a small set of specific applications to be performed, most of which characterized by long lifetime and regular and predictable communication protocols. Examples of events include the opening or closure of a connection, a protocol request on a specific device or the arrival of a new packet over a certain protocol. Thereby, these can be easily modeled for anomaly detection during operation, while signature-based algorithms are exploited to detect known attacks.

Aside the cyber domain, security in SCADA systems is also related to physical fault management. As previously mentioned and analyzed in [15], CIs are large-scale systems consisting of a wide number of components as sensors, actuators, controllers, communication links, etc. These shall fail at any time, due to component aging, degradation, human error or malicious attacks, among others. As such faults or attacks have a negative impact on the overall system performance and security and may be undetected, the management of early diagnosis and efficient accommodation is required.

One strategy for fault diagnosis is to exploit redundancy, either hardware (from extra sensors or software) or analytic (from mathematical or statistical models), useful to increase the available information by combining additional data and information. For critical scenarios, often hardware redundancy is preferred. In the case of CIs, despite the intrinsic criticality of the operations, analytic models are exploited to avoid excessive cost rising due to installation and maintenance, and most of them use centralized state-estimation techniques. Actually, the state estimator is the main tool in the SCADA systems which provides snapshots of the operating condition in a periodic time basis. Specifically, it provides the state of the system by processing redundant information obtained from customized models or measurements sent from various sensors placed in strategic locations of the system networks. Thereby, Bad Data Detection (BDD) schemes are employed to generate alarms when incongruous data is revealed during transmission. More specifically, these are used as filters against faulty measurements introduced by malfunctions or malicious attacks, ensuring the integrity of the state estimations.

Considering the critical environments in which these systems operate, one shall consider that one or more sensors may provide erroneous information or the output data from the state estimators may have been tampered, degrading the efficiency of the system or leading to instability of the system. Indeed, a fault-free system operation cannot be guaranteed. As a consequence, process monitoring and fault diagnosis are becoming essential in modern automatic control systems. In addition, an usual approach to overcome the intrinsic nonlinearities that often characterize the models of the processes to be controlled and monitored is linearization, what leads to additional errors affecting the fault detectability.

2.3 Water Networks

As any other modern Critical Infrastructure, the drinking water sector heavily rely on automated technologies to manage and supervise the whole process. The evolution of ICS has truly improved the reliability, quality and efficiency of water services. Not only people rely on the constant delivery of drinking water, which is used for the most basic human needs, but also businesses, vital networks, industries, hospitals, agriculture, and other utilities depend on water systems. For instance, widespread illness or casualties may be the result of a significant attack on a water system. Moreover, they are essential for natural disaster recovery, and a denial of service could affect critical services such as firefighting. As a consequence, the water sector has been recognized by homeland security experts as one of the 18 Critical Infrastructure sectors considered vital systems and networks to be protected [35]. As stated in [36], its goal is to recognize and reduce risks to infrastructure and support practices that build and maintain system resiliency.

The water supply system can be divided into two main parts: the transmission network and the water distribution network. The former transports raw water from the sources, e.g. rivers or dams, to the water treatment plants where it is purified and quality is improved. Treated water is then transported along the pipelines to the water storage facilities, and the distribution system, which consists of a large number of underground pipelines that run through the city, providing good quality water to the points where it is delivered to the consumers, based on the daily water demand.

Specifically, a water distribution system is a complex system composed of two main categories of interconnected devices. Pipelines, tanks and reservoirs constitute the *passive* elements of the system, as they cannot be acted but receive the effects of the operations of the active components in terms of pressure and flow. On the other hand, the *active* elements are pumps, valves, regulators (hydraulic control elements) and other components, which are operated to control the water flow and/or the pressure in different parts of the network. All these are properly combined to reliably satisfy the demand of hydric resources from different types of consumers (civil, industrial, etc.), guaranteeing the required water quantities, pressure and quality. The operation of a water distribution system is mainly regulated by:

- The laws which describe its physical behavior, i.e. the flow relationships in the pipes and the hydraulic control elements;
- The water demand from the consumers;
- The system's layout.

As stated in [15], [37], water supply systems planning, optimization and management problems are characterized by a high number of constraints and decision variables, and the equations describing head, flow and water quality are highly non-linear and non-smooth. Such problems can be mainly classified in:

(i) System topology;

(ii) System design;

(iii) System operation.

The supervisory control system actuates the active elements and controls their performance, and periodically updates the information gathered from remote stations and substations, specifically from a set of passive elements and all (or most of all) the active ones, so to monitor the operating conditions of the water network. Moreover, when dealing with the hydric system management one shall also take into account problems related to aggregation, maintenance, reliability, unsteady flow and security.

It is worthy to highlight that many utilities still exploit very old and outdated systems, which result to be extremely vulnerable from both the physical (structural) and cyber (control) points of view. The materials used may not be as resilient as the current day materials, and the aging of components contributes to their weakening. Old ICS equipment is indeed hard to update, due to the out-of-service time for maintenance operations and high costs. In addition, it is not uncommon to find products in use that are no longer being maintained by vendors and for which patches are not released. These constitute vulnerabilities that are often exploited by aware spiteful agents. Moreover, security was not a primary objective during their design and building, and a wide number of access points are available.

For what concerns security aspects, three main classes of threats on the water systems can be distinguished:

- physical disruptions and attacks against the infrastructure (dams, treatment plants, storage reservoirs, pipelines, hydraulic actuators, etc.);
- cyber-attacks against the water SCADA system;
- deliberate chemical or biological contaminant injection on the water network.

Water utilities have always dealt with the consequences of extreme weather conditions, earthquakes, aging infrastructure and equipment failures causing utility outages. Since service interruptions are not uncommon, most organizations are able to face and manage small scale problems effectively. More significant system failures of every nature are way more challenging to prevent and mitigate.

Despite what most people may think, water networks and utilities are undergoing an increasing number of cyber attacks in the last years. These are generally carried out by individuals or groups to remotely corrupt or take control of data and information essential to system operations. Indeed, the increasing of consequences due to outages, the cascading impacts on other systems, the effects on publich health, the ability of first responders to provide emergency services, the massive economic lossess, and the damage to the population, are all due to the wide range of dependencies on water systems. For instance, the 9 incident tickets reported in 2009 to the ICS-CERT managed by the Department of Homeland Security (USA), became 198 in 2011, the 41% of which involved the water-sector utilities [14]. Then, this percentage has unfortunately risen to the 60%, as reported by the Repository for Industrial Security Incidents (RISI) in the Report on Control System Cyber Security Incidents of 2013 [38].

As declared by Reuters in [39], the 8th November 2011 hackers managed to remotely shut down a pump in central Illinois (USA). The attack has been achieved from a computer located in Russia, exploiting stolen credentials from a company which provides software for ICS. The company declared that login credentials were kept by the software developers to allow efficient customers' support. Several cyber-attacks are actually traced to outside companies that provide services to the involved utilities, what focuses there the question of compromises. Fortunately, no service interruption for the 2200 customers of the district was reported.

The increasing threats have been further witnessed by Network World in 2012 [40], where it was stated that "water and energy utilities face constant cyber-espionage and denial-of-service attacks against industrial-control systems". As declared by the ICS-CERT, most of the incidents were originated by well-organized threat actors through spear-phishing attacks via e-mail against utility personnel. What is worthy to highlight is that most of these proven attacks could have been avoided or detected if the compromised utility would have exploited basic network security in their systems.

Aside the clear effects on the systems' components, a further goal of adversaries is to steal sensitive information, ranging from knowledge about a system's vulnerabilities, site security plans, response and recovery plans, description of processes, asset specifications and detailed maps, to customer records and financial data. With a wider knowledge about the whole target utility, more complex attacks can be planned, administrator functions may be accessed, or cyber and physical attacks could be combined.

The US Environmental Protection Agency (EPA) urges water utilities to implement, update and strengthen their cyber security systems [41]. As a consequence, a great effort has been made for the water utilities interests on the development of the most recent NIST's cyber security framework [42]. Similar endeavors have been made in Europe, as witnessed by the definition of 39 Good Practices for SCADA security for the drinking water sector, defined in the TNO report released in 2009 [43], under request of the Dutch ICTU (ICT Implementation Organization) programme NICC (National Cyber Crime Infrastructure). These are based on international standards, de facto standards, and useful security measures employed by industries worldwide.

Without wishing to belittle their importance, while physical and cyber-attacks could be minimized by improving the system's physical security and by increasing the security measures through hardware or/and software protection techniques, respectively, the contamination problem is the most difficult to deal with, due to the variety and uncertainty of the type of contaminant injected and its effects, the injection location and time. In addition, considering the wide number of potential targets, an intentional attack (or its solely attempt or menace), may lead the population to panic. For instance, in July 2005 a false alarm about terrorists having poisoned the water network of Rome has been triggered by a local radio station [44], which assured to have received the information from a reliable source, despite it has never been revealed. Such hoax rapidly spread through every communications medium and psychosis became viral in the Italian capital. Similarly, the 11th February 2015, few months before the Universal Exposition EXPO 2015 in Milan, Panorama, a well-known Italian weekly magazine, published an article with a shocking header: "Two people and 7000 Euros are enough to poison Milan" [45]. Despite the author's intention was only to make people aware of the actual risks and the feasibility of such type of attacks, a big effort was needed from the government to mitigate its repercussions on the media and the population. However, chemical and biological hazards go beyond the goals of this work and are not addressed here. Further details can be found in [15].

Another concern related to the efficient water distribution are the unavoidable leaks that take place along these networks, which are to be minimized as may cause significant economical losses for both service providers and final consumers. Moreover, such leaks may damage physical components of the infrastructure, third-party damages and healthy risks. Thanks to the evolution of monitoring technologies, the possibilities for controlling and managing water networks arose. As largely described in [46], current real-time monitoring of water networks is based on the comparison between telemetric sensor data and predictions obtained through well-calibrated hydraulic models of the

normal operation of the network. By analyzing their difference, it is possible to detect and diagnose faults and possible abnormal situations.

In the following sections, the Fault Diagnosis problem is studied, and the specific case of leaks in a water system is further studied and experimentally tested.

Chapter 3

Fault Diagnosis Techniques

Faults in the systems are often characterized by critical changes in the system's parameters or in its dynamics. Thereby, Fault Diagnosis systems have been developed to automatically detect, diagnose such events, and to eventually perform the proper corrective actions to guarantee safe and efficient system operation. Due to the high level of automation of current industrial systems and the unavoidable error circumstances that the physical components may encounter, applied research on the Fault Diagnosis field is gaining increasing interest for the implementation of academic techniques on always more complex engineering systems. Such technology is becoming essential in the design of intelligent and autonomous control systems, as is guarantees enhanced reliability, security, system availability, as well as maintenance costs reduction.

3.1 Fault Diagnosis in ICS

The main idea for Fault Diagnosis is to monitor the state of the system and to evaluate its operating conditions, preferably in real-time, in order to be able to perform early actions and countermeasures if necessary to prevent the propagation of faults, a disruption in the system or a domino effect on other interdependent systems. Such faults may be caused by design or implementation errors in the system, human errors, unappropriated use, ageing or deterioration, damages, etc. Thereby, *Fault Diagnosis* techniques are gaining attention because of the increasing level of automation in the management of processes, and have been exploited as a decision support for the operators in the control rooms.

The Fault Diagnosis problem can be roughly divided into four main phases:

- **Fault Detection:** detection of the faults present in a system, and the specification of the time at which the detection takes place;
- **Fault Isolation:** indicates the determination of the type and location of the detected faults;
- **Fault Identification:** determination of the size and the evolution in time of the faults. It is carried out in order to quantify the extent to which a fault is present in the system.
- **Fault Accommodation:** considers the reconfiguration of the system to guarantee the service. Depending on the gravity of the fault, to a certain extent this phase aims to permit the system to provide its service, although degraded.

To a fault tolerant system it is required that both false alarms (detected fault when actually no fault occurred in the system) and missed detection (a fault not detected) are minimized.

3.2 Fault Diagnosis Techniques

The Fault Diagnosis problem has been largely addressed in the literature [47], [48], [49] and a wide number of approaches and techniques for Fault Detection and Identification have been developed in the last decades [50], [51]. The Fault Diagnosis techniques have been mainly classified into three groups as sketched in Figure 3.1:

Model-free: these methods require an adequate database of historical data collected in normal operating conditions. The *statistical methods* perform statistical tests on the measured data in order to detect any abnormal behavior. In such a case, the fault diagnosis can be reformulated as the problem of detecting changes in the parameters of a stochastic variable that is monitored. The *knowledge-based methods* require the acquisition of a certain knowledge of the process, a suitable choice of its representation and encoding, and the development of inference procedures for diagnostic purposes. Both these techniques rely on the hypothesis that the distribution of the observed variables in normal conditions is different from the one obtained in case of fault. As a consequence, the main drawback consists on the uncertain behavior of the fault diagnosis system outside its domain of expertise.

Model-based: the models of the process that are deployed can be *quantitative* or *qualitative*, and are based on a different representation of the available *a priori* knowledge of the system and the potential faults that could take place in it. In the first case, qualitative functions centered on different units in a process are used, and mainly cause-effect relations are considered [52], while for the quantitative techniques the relationships are expressed in terms of analytic functions that relate the inputs and outputs of the system, generally reflecting the physical laws describing the system's dynamics. In such case, the estimations of a set of variables are compared to what is considered to be the normal operation of the system, in order to detect and identify the faults. In both cases, appropriate information and knowledge is needed for the estimation of some of the parameters in the model.

Soft computing: these techniques are based on the integration of knowledge-based and analytic methods, exploiting artificial intelligence algorithms [53].

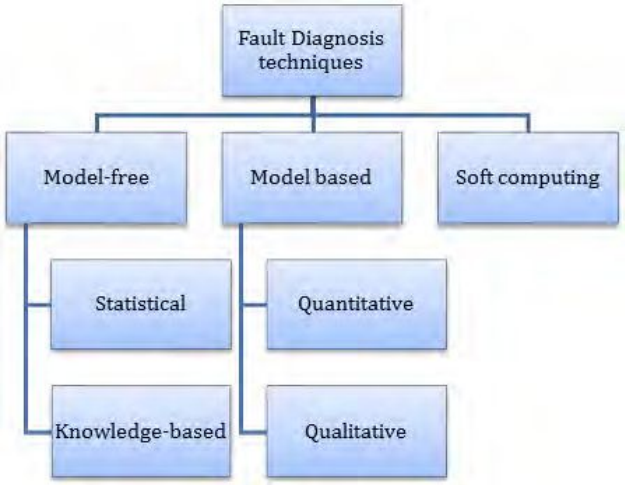

FIGURE 3.1: Fault Diagnosis techniques

Despite the type of method considered, some common characteristics are required for the implementation:

- Robustness to disturbances, noise and modelling errors;
- Sensitivity to different types or locations of the fault;
- Detectability;
- Small or acceptable detection time;
- Limited computation effort;
- Reduce the diagnosis errors.

3.2.1 Quantitative Model-Based Methods

The quantitative model-based techniques rely on the hypothesis that the process outputs can be measured and the system being studied can be represented by a quantitative model. In such a case, the diagnosis can be performed by identifying the current process parameters [54] or by observing the current outputs or states of the system [55], [56]. More specifically, the model explicitly represents the expected behavior of the system considered. On this framework, most of the fault diagnosis and identification techniques are based on the comparison between the observed behavior of the system and the predictions obtained from the model of the process, as depicted in Figure 3.2. Hence, the main elements required for a model-based diagnosis are:

- A model of the system, containing a description of its structure and the expected behaviors for each component;
- A set of observations on the current behavior of the system, obtained from the measurements;
- A set of symptoms.

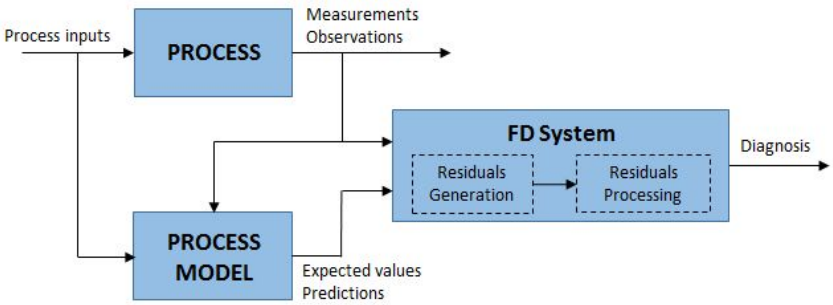

FIGURE 3.2: Model-based falut diagnosis

As previously described, the Fault Diagnosis problem consists of four main phases, the first three of which constitute the *Fault Detection and Identification* (FDI) problem, and are explicitly defined as:

- **Fault Detection problem:** process by which deviations from the normal behavior of the system are detected. Given a mathematical model of the system to be monitored and a sequence of measured inputs and outputs, it is tested whether the hypothesis "the system is healthy" is true or false.

- **Fault Isolation problem:** process of localizing the physical region or component where the fault took place. Given a mathematical model of the system to be monitored, a mathematical model of N possible faults that may occur and a sequence of measured system inputs and outputs, test for the N hypothesis whether "the system is affected by the i-th fault" is true or false.
- **Fault Identification problem:** determination of the size of the fault. It generally involves the estimation of process parameters or the implementation of fault models representing the faulty behavior of the system.

An acceptable solution to the FDI problem should be such that the fault decision can be provided in real-time, so that the largest possible amount of available time is left to the Fault Accommodation phase, before the fault event leads to a failure.

The most frequently used FDI approaches include the diagnostic observers [55], [56], parity relations [57], Kalman filters [58] and parameter estimation [54]. Generally, a set of *residuals* is generated, signals calculated from the difference between measurements and predictions about the state of the system. It is worthy to highlight that the main assumption when using quantitative model-based techniques for Fault Diagnosis is that a precise mathematical model of the plant and faults is to be available. Residuals should ideally be zero in normal operation (in the absence of faults), and differ from zero when a fault occurs. Actually, model uncertainties and physical disturbances usually make residuals different from zero even during normal operation and when no fault is present. Moreover, false detections are to be avoided, thus the residual generator should be able to distinguish a real fault from unmodeled dynamics and uncertain knowledge of the system parameters, and to identify values of the residual that do not indicate a fault. In addition, it would be desirable to obtain information on the fault from the evaluation of the residual, as the time when it occurred, the specific affected component, etc.

The main drawbacks of pure quantitative methods are:

- It is not possible to build exact quantitative models, as there is no enough information about the process, uncertainty in the parameters is to be considered and cannot be properly modeled or avoided, and noise cannot be exactly estimated;
- The computation of quantitative models in real-time may be intractable for complex industrial systems.

In general, the residuals r are a function of the input u and the output y of the system, the noise n, the disturbances ν and the uncertainties d:

$$r = \phi(u, y, n, \nu, d).$$

When dealing with thresholds, two building approaches are possible: the assumption that model uncertainties and disturbances are structured, and the use of *adaptive thresholds*. The latter is the most promising solution for guaranteeing robustness without relying on very conservative fixed thresholds, though the knowledge of a bound on the uncertainties and disturbances is needed.

A more successful approach, deployed in Soft Computing techniques, is based on the use of adaptive on-line approximators, such as neural networks, to learn on-line the unknown or uncertain parts of the dynamical model of the system, or the fault model

if the fault accommodation problem is considered. This learning approach enables the implementation of robust FDI schemes for non-linear uncertain systems.

Limit checking

Such method relies only on the knowledge of a range in which each measured variable is allowed to vary. The usability and success depends on the process working around a well-known set-point.

Signal-based

It relies on fact that the behavior of the measured variables can be precisely analyzed in the time and frequency domains. Known features of measured signals, such as spectral components or peculiar transients, are then compared to the nominal ones. Hence, the method requires specific knowledge of the system behavior during healthy operation.

Parity relations

For this technique a reliable mathematical model of the healthy behavior of the process is built, using which some estimations of the measured variables are computed. Such estimations are then compared to the actual measurements, so as to detect the deviations due to a fault in the system. Hence, the residuals obtained are compared to suitable thresholds by detection and isolation logic, in order to provide a fault decision framework. Specifically, greater-than-zero thresholds must be considered due to model uncertainties, noise and physical disturbances.

Diagnostic observers

A state-space model of the system to be monitored is used in this technique, so that estimations of the state and output can be computed. The estimation errors are then used as residuals and compared to a suitable threshold for detection and isolation purposes. Also in this case noise, uncertainties and disturbances are to be considered, especially for non-linear systems.

Parameter estimation

For this method a parametric model of the system is used. An on-line learning technique is used to adapt the model parameters to the observed measurements. A fault is said to be *detected* when the parameters diverge from their allowed nominal region corresponding to the healthy system behavior.

3.2.2 Redundancy

Two types of redundancy are considered to make the Fault Diagnosis system more robust: *hardware* and *analytic*. The former requires the use of redundant sensors and it is generally used for the control of safety-critical systems. Its main limitation is related to the cost and additional space required for their placement. On the other hand, analytic redundancy is obtained from the functional dependence between the process variables and is generally provided by a set of algebraic or temporal relationships among the inputs, the state and the outputs of the system. Hence, any kind of inconsistency expressed as residuals can be used for detection and isolation purposes.

Analytic redundancy methods do not need to add physical instruments in the plant, and make use of the dependencies among different measured signals in order to detect faults in the process, actuators or sensors that compose the system. Such relations are represented by mathematical models, and the main drawbacks consist on the non-linearity that characterizes the systems considered and the modeling errors. Hence, the final goal of residuals evaluation is to obtain a symptom vector which is the less possible sensitive to noise and false alarms.

Methods based on analytic redundancy derive residuals which are insensitive to uncertainties, but are sensitive to faults. In order to avoid such inconvenient, disturbance decoupling is used. Hence, all uncertainties are treated as disturbances and filters are designed to decouple the effect of faults and unknown inputs, so that they can be properly differentiated.

3.2.3 Types of Faults

When deploying realistic models of a system, two different types of faults can be distinguished considering the way they affect its behavior:

- **Additive faults:** appear as additional terms in the process model and are independent on the value of the observed variables. This type of faults are modeled as unknown functions of time, multiplying known matrices.
- **Multiplicative faults:** lead to changes in the parameters and depend on the actual values of the observed variables. They are represented in the model as known observable functions of time, multiplying unknown matrices.

Considering the behavior of the fault, they can be defined as:

- **Abrupt:** sudden and considerable;
- **Incipient:** slowly affect the system;
- **intermittent:** no specific pattern is observed.

Moreover, faults can be *external*, when the interactions between system and environment are not compatible with the goals, or *internal*, when they take place in one or more of the components of the systems.

3.2.4 Soft Computing Methods

As any modelling error affects the performance of the FDI scheme, especially if dealing with non-linear systems, more abstract models, based on other methods, have been developed. *Soft computing* is the term used to classify all methods which employ computational intelligence algorithms as neural networks, fuzzy logic, and their combinations into hybrid methods [53].

The fuzzy-logic rules aim to assist or replace the use of a model for diagnosis, as they describe the system behavior by the use of "if-then" relations.

On the other hand, neural networks have been studied as they could be trained to reproduce a specific system behavior from given data sets. Hence, to overcome some of the difficulties of the model-based techniques and make FDI algorithms more reliable for real systems, neural networks can be used to both generate residuals and isolate faults [56], as one of their main abilities is to learn from examples. Indeed, they can

be trained to represent relationships between residuals generated from historic data and those identified with some known fault conditions. The framework developed for such approach involves a multi-layer feed-forward network configuration, as depicted in Figure 3.3. The main drawback of this configuration is that symbolic knowledge from experts cannot easily be incorporated, whilst it can be well trained on numerical data if the output is known. More specifically, neural networks can be employed for the Fault Diagnosis problem by the use of different approaches, as pattern recognition and residual generation-decision making. In any case, they provide a "black box" signal processing structure, as the rules governing their operation are hidden and from an input-output point of view the real behavior is unknown. Moreover, the required training time and the complexity of the training algorithm present huge limitations, and state variables with additional terms can be used in training in order to accelerate convergence.

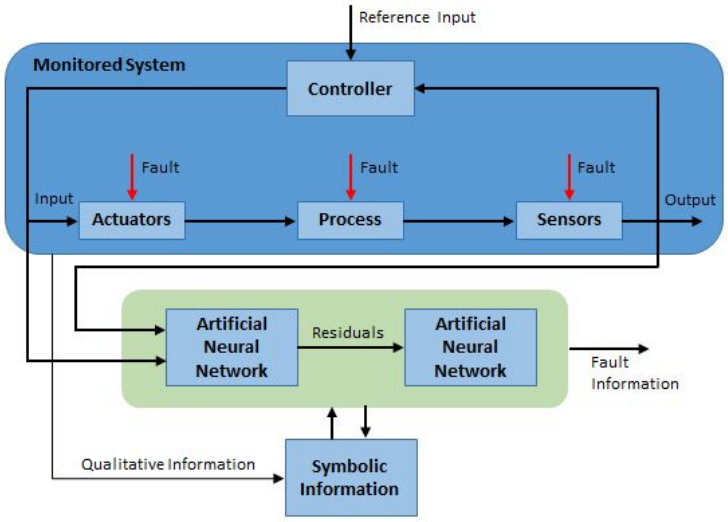

FIGURE 3.3: Neural networks scheme or FDI [53]

3.3 Analyzed Methods

Two model-based Fault Detection and Identification methods have been deeply analyzed, inspired on which a specific algorithm for the detection of physical faults in the experimental testbed has been developed.

3.3.1 Linearized Observer-Based Method

Consider the open-loop schema of a generic system as depicted in Figure 3.4, composed of sensors, actuators and the process. For the sensors, a technique based on physical redundancy will be exposed, while for actuators and process the analytic redundancy will be deployed, by the use of a set of observers both for detection, isolation and identification of the particular fault [59].

A non-linear, time varying, dynamic model can be used for such system:

$$\dot{x} = g(x, u, t) + \eta(x, u, t)$$

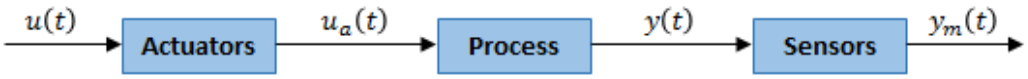

FIGURE 3.4: Open-loop schema of a generic system

$$y = h(x, u, t)$$

where x, u and y are the state, input and output vectors, respectively, g describes the nominal dynamics of the system and η considers the system uncertainties. Neglecting the dynamics of actuators and sensors, the fault-free conditions can be described by the following relations:

$$u_a(t) = u(t), \quad y_m(t) = y(t) + n(t).$$

There, the input signal for the actuators is transmitted as it is, while the signal provided by the sensors is the actual measurement of process outputs, influenced by the measurement noise $n(t)$. The actuator and sensor fault can be modeled as:

$$u_a(t) = u(t) + f_a(t), \quad y_m(t) = y(t) + n(t) + f_s(t),$$

where f_a and f_s are the actuator and sensor fault vectors, respectively. Similarly, process faults can be modeled via an unknown additional term $f_p(x, u, t)$ affecting the state equation of the system dynamics.

The residuals measuring the discrepancies between the behavior of the real system and the model predictions, can be obtained as functions of the measured output y_m and its estimate $\hat{y}$:

$$r = \phi(y_m, \hat{y}, t).$$

Observer-based approaches estimate the output of the system based on the available measurements, and the estimation error is deployed as residual. In the case of methods that rely on parameter estimation, they are based on the hypothesis that a fault may lead to a sensible variation of the physical parameters of the system. Thereby, it is necessary while not trivial to build particularly accurate models.

After being generated, residuals must be processed in order to detect and isolate faults. The decision process usually comes down to the selection of suitable nonzero thresholds, specifically designed to maximize detection and avoid false alarms due to model errors, disturbances, uncertainties, noise, etc. The simplest strategy is to set fixed thresholds, chosen so as to find a compromise between sensitivity to faults and the need to minimize false alarms. Sensitivity to faults can be improved by using adaptive thresholds, adjusted on-line on the basis of measurements.

The dynamic model can be rewritten in order to obtain the following linearized form, where two sensors are considered:

$$\dot{x} = A_d(y)x + b(y, u) + \eta(x, u, t)$$
$$y = Cx + n$$

where matrices A_d and C are defined as:

$$A_d = \begin{bmatrix} A_M(y) & 0_{2\times N_C} \\ A_{M,E}(y) & A_E \end{bmatrix}, \quad C = \begin{bmatrix} 0_{2\times N_C} & I_{2\times 2} \end{bmatrix}$$

and b, N_C, and matrices $A_M, A_{M,E}, A_E$ contain parameters of the specific process. All the model uncertainties are in $\eta \in \mathbb{R}((N_C + 2) \times 1)$. It is assumed that both uncertainties and measurement noise are norm-bounded:

$$\exists \bar{\eta}, \bar{n} > 0 : ||\eta(x, u, t)|| \leq \bar{\eta}, ||n(t)|| \leq \bar{n}.$$

In such linearized framework, sensor faults could be again modeled as an unknown additive term in the output equation.

$$y(t) = Cx(t) + f_s(t) + n(t).$$

Actuators faults may be modeled as an unknown additive term affecting the state equation, due to unexpected variations of the input with respect to its nominal value. The effects of both process and actuators faults on the system dynamics can be modeled via an additive term in the state equation:

$$f_{a,i}(y, u, t) = \varphi_i(y, u, t)\theta_{f,i},$$

The function f_a is assumed to belong to a finite set of N_F functions F_a, each of which is assumed to have a linear-in-the-parameters structure, where φ_i s a known regression matrix that takes into account the structure of the fault and $\theta_{f,i}$ is an unknown vector of constant parameters that characterizes its magnitude. The regression matrix φ_i is assumed to be norm-bounded for all fault types:

$$\exists \bar{\varphi}_i > 0 : ||\varphi_i|| \leq \bar{\varphi}_i.$$

Therefore, in the presence of faults in different points of the system, the state-space model becomes:

$$\dot{x} = A_d(y)x + b(y, u) + C^T f_{a,i}(y, u, t) + \eta(x, u, t)$$
$$y = Cx + f_s + n$$

where it is assumed that f_s and $f_{a,i}$ are null before the occurrence of a fault, and the occurrence of multiple faults of the same nature is not considered. More specifically, multiple process or multiple actuator faults can be detected but not correctly isolated and identified.

For the fault detection problem only a single residual is necessary, whereas a vector of residuals is usually required for fault isolation. This because it is considered that each residual is affected only by a specific subset of faults, and each fault only affects a specific subset of residuals. As a consequence, it is assumed that only N_F different types of faults can occur, and each isolation observer is designed so that its output is sensitive to all faults in the set but one. Thus, a suitable designed diagnostic system, together with a Decision Making System (DMS), declares the occurrence of a fault, isolates the possible faulty sensor, and outputs a healthy signal. Then, the latter is used to feed a bank of $N_F + 1$ non-linear adaptive observers. The first is used to detect the occurrence of a fault, while the other N_F observers, each corresponding to a particular type of process/actuator fault, achieve fault isolation and identification.

Sensor Fault Diagnosis

The observer for the sensor fault diagnosis has the following form:

$$\dot{\hat{x}}_{SM} = A_d(y_{SM})\hat{x}_{SM} + b(y_{SM}, u) + L_S\tilde{y}_{SM}$$
$$\hat{y}_{SM} = C\hat{x}_{SM}$$

where $\hat{x}_{SM}$ denotes the vector of the state estimates, $\hat{y}_{SM}$ and $\tilde{y}_{SM} = y_{SM} - \hat{y}_{SM}$ are the vectors of output estimates and output estimation errors, respectively, and L_S is the gain matrix modifying the state estimate as a function of the output estimation error.

The dynamics of the state estimation error $\tilde{x}_{SM} = x_{SM} - \hat{x}_{SM}$ can be modeled as:

$$\dot{\tilde{x}}_{SM} = A_S(y_{SM})\tilde{x}_{SM} + \eta_S(x, u, t) + L_S f_s(t)$$
$$\tilde{y}_{SM} = C\tilde{x}_{SM} + f_s(t) + n$$

where $A_S(y_{SM}) = A_d(y_{SM}) - L_S C$ and $\eta_S = \eta + L_S n$.

In [59] it has been demonstrated that, for ideal conditions, i.e. in absence of faults, uncertainties and errors, if the system parameters are bounded, there exists a set of observer gains for which the state estimation error is uniformly globally convergent to 0 for $t \to \infty$, with exponential velocity. On the other hand, in the absence of faults, but in the presence of bounded uncertainties and sensor noise, the evolution of $\tilde{x}_{SM}$ starting from the initial time instant t_0, given the initial state estimation error $\tilde{x}_{SM}(t_0)$, can be expressed as follows:

$$\tilde{x}_{SM}(t) = \phi_S(t, t_0)\tilde{x}_{SM}(t_0) + \int_{t_0}^{t} \phi_S(t, \zeta)\eta_S(x(\zeta), u(\zeta), \zeta)d\zeta,$$

where ϕ_S denotes the state transition matrix corresponding to A_S. As this converges asymptotically, it can be demonstrated that the norm of the output estimation error can be upper bounded by:

$$\begin{aligned} ||\tilde{y}_{SM}(t)|| &= ||C\tilde{x}_{SM}(t) + n(t)|| \\ &\leq ||\tilde{x}_{SM}(t)|| + ||n(t)|| \\ &\leq k_S\left(||\tilde{x}_{SM}(t_0) + \tfrac{\bar{\eta}_S}{\lambda_S}||\right) + \bar{n} = \mu\eta_S \end{aligned}$$

where $\bar{\eta}_S = \bar{\eta} + ||L_S||\bar{n}$. The bound $\mu\eta_S$ could be known or, at least, estimated with reasonable accuracy.

For the sensors fault detection problem, the residuals can be defined as:

$$r_{SM} = \frac{\tilde{y}_{SM}}{\mu_S},$$

where μ_S is a normalization factor. If a fault occurs, the absolute value of r_{SM} is expected to exceed a certain threshold.

On the other hand, in order to isolate the fault, it is necessary to define a number of residuals considering the physical redundancy, hence the number of sensors for which the isolation problem is to be studied. In such case, the residuals are computed considering that the output of a particular observer is affected only by a fault in a particular sensor to which it is associated, hence the fault is declared in a particular sensor based on which residual exceeds the threshold.

When a sensor fault takes place at $t = t_f$, the following equality holds:

$$\tilde{y}_{SM}(t) = C\tilde{x}_{SM}(t) + f_s(t) + n(t),$$

and the following inequality can be derived:

$$||\tilde{y}_{SM}(t)|| \geq ||f_s(t)|| - \bar{\mu}_S.$$

Therefore, a sufficient condition ensuring isolation of a fault affecting a sensor is:

$$||f_s(t)|| > \bar{\mu}_{S,i} + \mu_{S,i} \text{ and } ||\tilde{y}_{SM,l}(t)|| \leq \bar{\mu}_{S,l} \text{ for } l \neq i.$$

In other words, a fault can be detected and isolated only if its magnitude overcomes the effect of the uncertainties and disturbances.

Actuator and Process Fault Diagnosis

In the case of the actuators, the observer for fault detection has the following structure:

$$\dot{\hat{x}}_a = A_d(y)\hat{x}_a + b(y,u) + L_a\tilde{y}$$
$$\hat{y}_a = C\hat{x}_a$$

where $\tilde{y}_a = y - \hat{y}_a$ and L_a is the gain matrix similar to L_S.
The state estimation error dynamics is given by:

$$\dot{\tilde{x}}_a = A_a(y)\tilde{x}_a + \eta_a(x,u,t) + C^T f_a(y,u,t)$$
$$\tilde{y}_a = C\tilde{x}_a + n$$

where $\tilde{x}_a = x - \hat{x}_a$, $A_a = A_d - L_aC$ and $\eta_a = \eta + L_an$.

Also in this case the asymptotic convergence of the state estimation error can be demonstrated in ideal conditions. In the absence of faults and in the presence of uncertainties and disturbances, a bound can be defined for the output estimation error which is given by:

$$\begin{aligned} ||\tilde{y}_a(t)|| &= ||C\tilde{x}_a(t) + n(t)|| \\ &\leq ||\tilde{x}_a(t)|| + ||n(t)|| \\ &\leq k_a\left(||\tilde{x}_a(t_0) + \tfrac{\bar{\eta}_a}{\lambda_a}||\right) + \bar{n} = \mu\eta_a \end{aligned}$$

where $\bar{\eta}_a = \bar{\eta} + ||L_a||\bar{n}$. Hence, a fault is declared when the norm of the residual vector:

$$r_a = \frac{\tilde{y}_a}{\mu_a}$$

exceeds a suitably defined threshold. As before, the factor μ_a is introduced for normalization.

In the presence of a fault occurring at time $t = t_f$, the state estimation error can be expressed as:

$$\tilde{x}_a(t) = \phi_a(t,t_0)\tilde{x}_a(t_0) + \int_{t_f}^{t}\phi_a(t,\zeta)C^T f_a(y(\zeta),u(\zeta),\zeta)d\zeta + \int_{t_0}^{t}\phi_a(t,\zeta)\eta_a(x(\zeta),u(\zeta),\zeta)d\zeta,$$

from which it can be derived:

$$\begin{array}{rcl} ||\tilde{y}_a(t)|| & = & ||C\tilde{x}_a(t) + n(t)|| \\ & \leq & ||\int_{t_f}^{t} C\phi_a(t,\zeta)C^T f_a(y(\zeta),u(\zeta),\zeta)d\zeta|| > \bar{\mu}_a - \mu_a. \end{array}$$

Hence, a sufficient condition for correct detection of the fault is given by:

$$\exists t > t_f : ||\int_{t_f}^{t} C\phi_a(t,\zeta)C^T f_a(y(\zeta),u(\zeta),\zeta)d\zeta|| > \bar{\mu}_a + \mu_a,$$

what means, as before, that a process/actuator fault can be detected only if its effect on the estimation error dynamics has a magnitude larger that the effect of the uncertainties.

Once a process/actuator fault has been detected, isolation and identification can be achieved via N_F non-linear adaptive observers, each of which is sensible to a particular type of fault. In such a case, when a fault occurs which is not included in the N_F types considered in the design of the bank of observers, it can be only detected but not isolated or identified. Hence, considering this analytic redundancy, the i-th observer has the form:

$$\dot{\hat{x}}_i = A_d(y)\hat{x}_i + b(y,u) + L_{a,i}\tilde{y}_i + C^T\hat{f}_{a,i}(y,u,t)$$
$$\hat{y}_i = C\hat{x}_i$$

where $L_{a,i}$ is the gain matrix, $\tilde{y}_i = y - \hat{y}_i$ and $\hat{f}_{a,i}$ is an estimate of the i-th fault, expressed with a linear structure with respect to the parameters:

$$\hat{f}_{a,i}(y,u,t) = \varphi_i(y,u,t)\hat{\theta}_{f,i},$$

where $\hat{\theta}_{f,i}$ is an estimate of the unknown vector of fault parameters, which adaptive law is derived by using the Lyapunov synthesis approach:

$$\dot{\hat{\theta}}_{f,i} = \gamma_i^{-1}\varphi_i^T(y,u,t)\tilde{y}_i, \gamma_i > 0.$$

In the presence of the i-th fault, the state estimation error of the i-th observer is given by:

$$\dot{\tilde{x}}_i = A_{a,i}(y)\tilde{x}_i + C^T\varphi_i(y,u,t)\tilde{\theta}_{f,i} + \eta_{a,i}(x,u,t)$$
$$\tilde{y}_i = C\tilde{x}_i + n$$

where $\tilde{x}_i = x - \hat{x}_i, \tilde{\theta}_{f,i} = \theta_{f,i} - \hat{\theta}_{f,i}, A_{a,i} = A_d - L_{a,i}C$ and $\eta_{a,i} = \eta + L_{a,i}n$.

It can be demonstrated that if the i-th fault takes place, in absence of uncertainties and measurement errors, if the system parameters are bounded, a set of observer gains exist such that the state estimation error $\tilde{x}_i$ is uniformly globally convergent to 0 for $t \to \infty$, and the parameters estimation error $\tilde{\theta}_{f,i}$ is uniformly bounded for every t. On the other hand, when uncertainties and noise are limited but non-zero, the boundedness of $\tilde{\theta}_{f,i}$ is no longer guaranteed. As an appropriate projection operator is considered, the boundedness of $\tilde{\theta}_{f,i}$ is assumed hereafter, i.e. $||\tilde{\theta}_{f,i}(t)|| \leq \bar{\theta}_{f,i}$.

In order to achieve fault isolation, the following residuals are computed:

$$r_{a,i} = \frac{\tilde{y}_i}{\mu_{a,i}}.$$

If the i-th fault occurs, the norm of all residuals but $r_{a,i}$ exceeds its threshold. The norm of the output estimation error can be upper bounded by:

$$\begin{array}{rcl} ||\tilde{y}_i(t)|| & = & ||C\tilde{x}_i(t) + n(t)|| \\ & \leq & ||\tilde{x}_i(t)|| + ||n(t)|| \\ & \leq & k_{a,i}\left(||\tilde{x}_i(t_0) + \frac{\bar{\varphi}_i\bar{\theta}_{f,i}}{\lambda_{a,i}} + \frac{\bar{\eta}_{a,i}}{\lambda_{a,i}}||\right) + \bar{n} = \bar{\mu}_{a,i} \end{array}$$

where $\bar{\eta}_{a,i} = \bar{\eta} + ||L_{a,i}||\bar{n}$.

Hence, a sufficient condition for isolability for the i-th type of process/actuators faults is given by the two inequalities:

$$\exists t > t_f : ||\int_{t_f}^{t} C\phi_{a,l}(t,\zeta)C^T(\varphi_i(y(\zeta),u(\zeta),\zeta)\theta_{f,i} - \varphi_l(y(\zeta),u(\zeta),\zeta)\hat{\theta}_{f,l})d\zeta|| > \bar{\mu}_{a,l} + \mu_{a,l} \forall l \neq i,$$

and $||\tilde{y}_i(t)|| \leq \bar{\mu}_{a,i}$. These guarantee that all the residuals $||r_{a,l}||$, $l \neq i$ exceed the threshold at least for a time instant, while the i-th residual keeps below its threshold.

To make the sensor observer insensitive to process/actuator faults, the following modified dynamics can be adopted:

$$\dot{\hat{x}}_{SMi} = A_d(y_{SMi})\hat{x}_{SMi} + b(y_{SMi},u) + L_S\tilde{y}_{SMi} + C^T\hat{f}_a(y,u,t)$$
$$\hat{y}_{SMi} = C\hat{x}_{SMi}$$

where $\hat{f}_a$ is an estimate of the isolated process/actuator fault. This guarantees that the output estimation error $\tilde{y}_{SMi}$ is only marginally influenced by the process/actuator fault, provided that a bounded error on the fault estimation is achieved.

3.3.2 Centralized Model-Based Fault Diagnosis

Another approach that can be followed is the adaptive one, as proposed by Ferrari *et al.* in [60]. Consider a generic dynamic, discrete, non-linear system, modeled as:

$$x(k+1) = f(x(k),u(k)) + \eta(x(k),u(k),k) + \beta(k-k_0)\phi(x(k),u(k)),$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ represents the nominal healthy dynamics, $\eta : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{N} \rightarrow \mathbb{R}^n$ is the uncertainty in the model, $\beta(k-k_0)\phi(x(k),u(k))$ are changes in the system dynamics due to the occurrence of a fault, with $\phi(x(k),u(k))$ representing the functional structure of the deviation in the state equation due to the fault occurring at the unknown time k_0 and $\beta(k-k_0)$ characterizes the time profile of the fault.

For isolation purposes it is assumed that there are N_F types of possible non-linear fault functions, described as $\phi(x,u)$. Let F be a class function defined as

$$F := \phi_1(x,u), \ldots, \phi_{N_F}(x,u),$$

and more specifically, each fault function that describes the dynamical model of the faulty behavior is assumed to be in the form:

$$\phi_l(x(k),u(k)) = \left[(\theta_{l,1})^T H_{l,1}(x(k),u(k)), \ldots, (\theta_{l,n})^T H_{l,n}(x(k),u(k))\right]^T$$

where the structure of the fault is provided by known functions $H_{l,i} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n q_{l,i}$ and the unknown parameter vectors $\theta_{l,i} \in \Theta_{l,i} \subset \mathbb{R}^n q_{l,i}$ provide its magnitude. For simplicity, some assumptions have been made for the use of this model:

- the parameter domains $\Theta_{l,i}$ are assumed to be origin-centered hyper-spheres, with radius M_{Θ_0};
- at time $k = 0$ no faults take place on the system;
- state and control variables, i.e. $x(k)$ and $u(k)$ are always bounded;
- the model uncertainty η are bounded by some known functional $\bar{\eta} > 0$, for which:

$$|\eta^{(i)}(x(k), u(k), k)| \leq \bar{\eta}^{(i)}(x(k), u(k), k).$$

The Fault Detection and Approximation Estimator (FDAE) is based on a model of the healthy system and provides an estimate $\hat{x}_0(k)$ of the system state. Based on this, a residual and a threshold vector are computed that guarantee the absence of false-positive alarms. An online adaptive approximator is turned on in order to learn the possibly unknown fault functions ϕ. Then the isolation service is provided, which aim is to find which fault function does better represent the actual behavior of the system after a fault has been detected, and to estimate the parameters vector of the fault function. Each of the N_F Fault Isolation Estimators (FIE) is activated after a fault has been detected, and is tuned to a specific element of the fault class F, yielding a state estimate $\hat{x}_j(k) \in \mathbb{R}^n$, thus a residual and a threshold vector.

Healthy Behavior and Fault Detection and Approximation Estimator (FDAE)

Until a fault is detected, the FDAE estimator is the only module to be enabled and provides a state estimate $\hat{x}_0$ of the state x. An estimation error will be computed as a difference between such estimate and the actual measured state:

$$\tilde{x}_0 = x - \hat{x}_0,$$

which is compared, component by component, to a suitable detection threshold $\bar{x}_0 \in \mathbb{R}^n_+$. The system is considered to be healthy (fault hypothesis) if the following condition is met:

$$|\tilde{x}_0^{(i)}(k)| \leq \bar{x}_0^{(i)}(k) \quad \forall i.$$

If such condition is unmet at some time instant k, the fault hypothesis is falsified and a fault signature will be noticed. Formally, it is the index set of the state components for which the fault hypothesis did not hold for at least one time instant, i.e. $S := \left\{ i : \exists k_1, k \geq k_1 > 0, |\tilde{x}_0^{(i)}(k_1)| \leq \bar{x}_0^{(i)}(k_1) \right\}$. Thus, a fault affecting the system will be detected at the first time instant such that S becomes non-empty. Such time is called the fault detection time k_d:

$$k_d := min \left\{ k : \exists i, i \in \{1, \ldots, n\} : |\tilde{x}_0^{(i)}(k)| \leq \bar{x}_0^{(i)}(k) \right\}.$$

Before the detection of a fault, for $0 \leq k < k_0$, the dynamics of the FDAE are selected as:

$$\hat{x}_0(k+1) = \lambda(\hat{x}_0(k) - x(k)) + f(x(k), u(k)),$$

with $0 \leq \lambda < 1$. The state estimation error dynamics is:

$$\tilde{x}_0(k+1) = \lambda\tilde{x}_0(k) + \eta(x(k), u(k), k) + \beta(k - k_0)\phi(x(k), u(k)).$$

By choosing $\hat{x}_0(0) = x(0)$, before the occurrence of a fault, i.e. for $0 \leq k \leq k_0$, the solution of the equation is:

$$\tilde{x}_0(k) = \sum_{h=0}^{k-1} \lambda^{k-1-h}\eta(h).$$

In order to guarantee no false-positive alarms, the threshold on the FDAE estimation error is to be defined as:

$$\bar{x}_0^{(i)}(k) := \sum_{h=0}^{k-1} \lambda^{k-1-h}\bar{\eta}^{(i)}(h) \geq |\tilde{x}_0^{(i)}(k)| \quad \forall k \leq k_0.$$

This certainly implies that faults whose amplitude is comparable with the bound $\bar{\eta}$ are impossible to detect.

Faulty Behavior and Fault Detectability

If there exist two time indexes $k_2 > k_1 \geq k_0$ such that the fault ϕ fulfills the following inequality for at least one component i,

$$\left| \sum_{h=k_1}^{k_2-1} \lambda^{k_2-1-h}(1 - b^{-(h-k_0)}\phi^{(i)}(k)) \right| > 2\bar{x}_0^{(i)}(k_2),$$

then it will be detected at k_2, that is $|\tilde{x}_0^{(i)}(k_2)| > \bar{x}_0^{(i)}(k_2)$. In this case, an exponential profile for the fault is considered, and the parameter b is lower bounded by a known constant $\bar{b}$.

After the detection of a fault at time $k = k_d$, the FDAE approximator is turned on in order to learn the fault function, and the dynamics of the state becomes:

$$\hat{x}_0(k+1) = \lambda(\hat{x}_0(k) - x(k)) + f(x(k), u(k)) + \hat{\phi}_0(x(k), u(k), \hat{\theta}_0(k)),$$

where $\hat{\phi}_0$ is an adaptive approximator and $\hat{\theta}_0 \in \hat{\Theta}_0 \subset \mathbb{R}^{q_0}$ denotes its parameters vector. In order for $\hat{\phi}_0$ to learn the fault function ϕ, its parameter vector is updated according to the following learning law:

$$\dot{\hat{\theta}}_0(k+1) = P_{\hat{\Theta}_0}(\hat{\theta}_0(k) + \gamma_0(k)H_0^T(k)r_0(k+1)),$$

where $H_0(k) := \frac{\partial\hat{\phi}_0(x(k),u(k),\hat{\theta}_0(k))}{\partial\hat{\theta}_0}$ is the gradient matrix of the on-line approximator with respect to its adjustable parameters, $r_0(k+1) = \tilde{x}_0(k+1) - \lambda\hat{x}_0(k)$, γ_0 is the learning rate, and $P_{\hat{\Theta}_0}$ is a projection operator, needed to counter the effects of measuring or modeling uncertainties that make the approximation error be non-zero even when the parameter estimation error is zero or close to zero, and cause the parameter drift. Such projector is defined as:

$$P_{\hat{\Theta}_0}(\hat{\theta}_0) := \begin{cases} \hat{\theta}_0 & \text{if } |\hat{\theta}_0| \leq M_{\hat{\Theta}_0} \\ \frac{M_{\hat{\Theta}_0}}{|\hat{\theta}_0|}|\hat{\theta}_0| & \text{if } |\hat{\theta}_0| > M_{\hat{\Theta}_0} \end{cases}$$

where $M_{\hat{\Theta}_0}$ is the radius of the hyper-sphere constituting the parameter space. It works by projecting at each time the updated parameter vector inside its allowable domain $\hat{\Theta}_0$, without sacrificing the convergence speed or the parameter estimation accuracy.

The learning rate γ_0 is computed at each step k as:

$$\gamma_0(k) := \frac{\mu_0}{\varepsilon_0 + \|H_0(k)\|_F^2}, \quad \varepsilon_0 > 0, \quad 0 < \mu_0 < 2,$$

where $\|\cdot\|_F$ is the Frobenius norm and ε_0, μ_0 are design constants that guarantee the stability of the learning law.

Fault Isolation Logic

After a fault has been detected at time $k = k_d$, the N_F FIEs are activated in parallel in order to isolate the fault, each of which is related to a specific fault function in a defined fault class F. Which fault is the root cause cannot be discerned, as in general a signature would be such that more than a diagnosis can explain it. That is, more than one fault function in F may influence the variables referenced by the signature. Furthermore, even if theoretically the fault candidates would present unique signatures, there is no guarantee that at k_d all the analytic symptoms distinctive of a fault would be present. To make a robust and correct fault decision, the FDI scheme needs to conduct further tests that may lead to fault isolation, by mutually excluding the available fault candidates.

The difference between the measured state x and the estimate $\hat{x}_l$ will yield to the estimation error:

$$\tilde{x}_l = x - \hat{x}_l,$$

which is compared, component by component, to a suitable isolation threshold $\bar{x}_l \in \mathbb{R}^n_+$:

$$|\tilde{x}_l^{(i)}(k)| \leq \bar{x}_l^{(i)}(k) \quad \forall k.$$

If this condition is unmet at some time instant k, the hypothesis that the system is affected by the l-th fault is falsified and such fault will be excluded as a possible cause of the fault signature, at the exclusion time $k_{e,l}$, defined as:

$$k_{e,l} := min\left\{k : \exists k, k \in \{1, \ldots, n\}, |\tilde{x}_l^{(i)}(k)| > \bar{x}_l^{(i)}(k)\right\}.$$

Thereby, the goal of the isolation logic is to exclude every but one of the faults belonging to the fault class F. A fault $\phi_p \in F$ is isolated at time k if and only if $\forall l, l \in \{1, \ldots, N_F\} - p, k_{e,l} \leq k$ and $\nexists k_{e,p}$. Furthermore, $k_{is,p} := min\{k_{e,l}, l \in \{1, \ldots, N_F\} - p\}$ is the fault isolation time.

If a fault is isolated, it can only be concluded that it actually occurred if it is *a priori* assumed that only faults belonging to the class F may occur. Moreover, if every fault in F is excluded, it will be said that the proposed FDI architecture has isolated an unknown fault. In order to possibly add this fault to the class F of known faults, the FDAE on-line approximator is designed in order to be capable of learning any fault

that can reasonably occur.

The dynamics of the state estimation of the l-th FIE is:

$$\hat{x}_l(k+1) = \lambda(\hat{x}_l(k) - x(k)) + f(x(k), u(k)) + \hat{\phi}_l(x(k), u(k), \hat{\theta}_l(k)),$$

where $\hat{\phi}_l(x(k), u(k), \hat{\theta}_l(k))$ is a linearly-parameterized function whose i-th component $\hat{\phi}_l(i)(x(k), u(k), \hat{\theta}_l(k)) := (\hat{\theta}_{l,i})^T H_{l,i}(x(k), u(k))$ matches the structure of $\phi_l(i)$. The learning law for $\hat{\theta}_{l,i}$ is analogous for the FDAE, and for each component it is defined as:

$$\hat{\theta}_{l,i}(k+1) = P_{\hat{\Theta}_{l,i}}(\hat{\theta}_{l,i}(k) + \gamma_{l,i}(k) H_{l,i}^T(k) r_{l,i}(k+1)),$$

where $r_{l,i}(k+1) = \tilde{x}_l^{(i)}(k+1) - \lambda \tilde{x}_l^{(i)}(k)$, $P_{\hat{\Theta}_{l,i}}$ is the projection operator on $\hat{\Theta}_{l,i}$ and the learning rate $\gamma_{l,i}(k)$ is computed as:

$$\gamma_{l,i}(k) := \frac{\mu_{l,i}}{\varepsilon_{l,i} + \|H_{l,i}(k)\|_F^2}, \quad \varepsilon_{l,i} > 0, \quad 0 < \mu_{l,i} < 2.$$

Assuming a matched fault, and with the initial condition $\hat{x}_l(k_d) = x(k_d)$, the solution to the i-th component of the estimation error dynamics equation is:

$$\tilde{x}_l^{(i)}(k) = \sum_{h=k_d}^{k-1} \lambda^{k-1-h} \left(\eta^{(i)}(h) + (1 - b^{-(h-k_0)})(\tilde{\theta}_{l,i})^T H_{l,i}(h) - b^{-(h-k_0)} (\hat{\theta}_{l,i})^T H_{l,i}(h) \right),$$

where $\tilde{\theta}_{l,i}(k) := \theta_{l,i}(k) - \hat{\theta}_{l,i}(k)$ is the parameter estimation error. The norm of the state estimation error can be appropriately upper bounded, allowing the definition of an upper bound:

$$\bar{x}_l^{(i)}(k) = \sum_{h=k_d}^{k-1} \lambda^{k-1-h} \left(\bar{\eta}^{(i)}(h) + g_{l,i} \|H_{l,i}(h)\| + \bar{b}^{-(h-k_d)} \|\hat{\theta}_{l,i}\| \|H_{l,i}(h)\| \right) \geq |\tilde{x}_l^{(i)}(k)|.$$

Any useful solution to the FDI problem must be such that the fault decision d_I^{FD} can be provided in real-time. More specifically, in the span of the sampling time T_s all the measurements and the computations needed to evaluate the $N_F + 1$ estimates of the next system state must be carried on.

3.3.3 Distributed Fault Diagnosis

The task of subdividing the FDI problem in order to let more than one agent solve it is called the *decomposition problem*. In such case, the decomposition of the estimation task will lead to have more than one fault detection estimator, each one estimating only a subset of the state vector, thereby each agent is devoted to monitoring a single subsystem. To such end, consider the system decomposition exposed in Section 5.1, to which the centralized Fault Detection technique is extended.

More specifically, the Distributed Fault Detection and Isolation (DFDI) architecture consists of a network of N agents called Local Fault Diagnosers (LFD), denoted by L_I and dedicated to monitor each of the subsystems $\mathcal{S}_I$ and provide a fault decision d_I^{FD} regarding their health. Each LFD will be able to directly measure the local state x_I

and the local input u_I. LFDs will communicate with their neighbors, exchanging the requested local measurements in real-time, obtaining the interconnection vector z_I. The generic I-th LFD will provide a local detection service through a local FDAE estimator, used to compute a local state estimate $\hat{x}_{I,0}$, a corresponding state estimation error $\tilde{x}_{I,0}$ to be used as residual, and a threshold $\bar{x}_{I,0}$. In addition, a local fault class F_I is defined. The isolation is carried on thanks to N_{F_I} FIE estimators that compute N_{F_I} further local state estimates $\hat{x}_{I,l}$ along with as many state estimation errors $\tilde{x}_{I,l}$ and thresholds $\bar{x}_{I,l}$.

As overlapping decompositions are allowed, shared variables belonging to more than one subsystem, will be monitored by more than one LFD. In particular, all the LFDs in the overlap set $\mathcal{O}_s$ of a shared variable $x^{(s)}$ will collaborate on the task of estimating it, and detecting and isolating faults by which it may be affected. This may be achieved by employing consensus techniques.

An LFD for a physical component can possess a nominal local model of its behavior when the component is operating in some known configuration, but the effect of interconnecting the component to other pieces in a complex system cannot be completely modeled *a priori*. For this reason, the FDAE and FIE estimator will use an on-line adaptive approximator $\hat{g}$ in *lieu* of the uncertain interconnection function g. The learning is then stopped as soon as a fault is detected in order to avoid the approximator learning the fault function as if it were part of the interconnection function.

If the generic I-th LFS detects a fault, it can be concluded that a fault did affect some components of its local state, causing a non-empty signature. Nevertheless, nothing can be deduced about what the fault is doing to other subsystems. One can only hope that if the fault is itself distributed and is affecting other subsystems, other LFDs will detect it. Anyway, there may be situations in which the effect of the fault are widespread, but do not fulfill the detectability condition on other LFDs so that the fault goes unnoticed by them.

The subsystem S_I is said to be healthy if the following condition is met:

$$|\tilde{x}^{(i)}_{I,0}(k)| \leq \bar{x}^{(i)}_{I,0}(k) \quad \forall i$$

.

The *local* signature shown by the subsystem $\mathcal{S}_I$ at time $k > 0$ is the index set $\mathcal{S}_I := \{i : \exists k_1, k \geq k_1 > 0, |\tilde{x}^{(i)}_{I,0}(k_1)| > \bar{x}^{(i)}_{I,0}(k_1)\}$ of the local state components for which the condition for healthiness did not hold for at least one time instant. Similarly, the *global* signature is the index set $\mathcal{S} := \{i : \exists k_1, k \geq k_1 > 0, \exists I \in \{1, \ldots, N\}, |\tilde{x}^{(j)}_{I,0}(k_1)| > \bar{x}^{(j)}_{I,0}(k_1), i$ is the j-th element of $I_I\}$ for at least one LFD.

A fault affecting subsystem $\mathcal{S}_I$ will be detected at the first time instant k_d such that $\mathcal{S}_I$ becomes non-empty. It is assumed that as soon as a LFD detects a fault, it will communicate it to all the other $N-1$ LFDs and the whole system will be declared as faulty, although it does not necessarily influence the whole system. In addition, as soon as the LFD detects or is informed about a fault, its FDAE approximator is stopped and its bank of FIEs is turned on in order to isolate the fault.

In this framework, three fault scenarios can be defined:

- **Local fault:** the local and global signatures are the same and thus it can be isolated by the corresponding LFD, realizing a local Generalized Observer Scheme (GOS)

isolation scheme without further communication between neighboring LFDs, except for the exchange of interconnection variables measurements.

- **Distributed fault, non-overlapping signature:** links and variables in more than one subsystem are affected by the same single fault, so that the resulting global signature is distributed. Shared variables are not affected. In the DFDI isolation scheme it is assumed that every LFD concerned will have in its fault class a local fault function able to explain its local signature. Every LFD will then implement a GOS fault isolation scheme. Only when all the LFDs involved will be able to isolate at some time their local part of the fault, then by communicating their successful diagnosis to each other they will collectively make a correct fault decision. The scalability of this solution holds as every LFD does need to know only local nominal models and local fault models related to its subsystem, and needs only to communicate limited information to other LFDs.

- **Distributed fault, overlapping signature:** as some variables interested by the signature are shared, more than one LFD possesses a local model of the fault influence on those variables. This will enable the use of consensus techniques, and the distributed fault will be isolated only if all the LFDs will isolate their local part of the fault.

Consider the non-linear, discrete-time dynamic system model:

$$x(k+1) = f(x(k), u(k)) + \eta(k) + \beta(k-k_0)\phi(x(k), u(k)),$$

where the function η includes external disturbances, as well as modeling errors and possibly the discretization error. The state equation of subsystem S_I with local state vector x_I can be modeled as:

$$x_I(k+1) = f_I^*(x(k), u(k)) + \eta_I(k) + \beta(k-k_0)\phi_I(x(k), u(k)),$$
$$x_I(k+1) = f_I(x_I(k), u_I(k)) + g_I(x_I(k), z_I(k), u_I(k)) + \beta(k-k_0)\phi_I(x(k), u(k)),$$

where f_I is the *local* nominal function and g_I is the interconnection function. It has been supposed that the uncertainty term η_I affects only the interconnection part of the model, and has been included in the function g_I. This is assumed to be an unstructured and uncertain nonlinear function, bounded by some known non-negative function:

$$|g_I^{(i)}(k)| \leq \bar{g}_I^{(i)}(k) \quad \forall i, k \geq 0.$$

For the sake of simplicity, it will be assumed that the local fault functions from different LFDs that will try to isolate the same distributed fault will be given the same index. As the known faults are assumed to retain the same structure of the healthy subsystem, their fault functions can be written in the form $\phi_I(x_i, z_I, u_I)$. Moreover, for each subsystem S_I the state variables $x_I(k)$ and control variables $u_I(k)$ remain bounded before and after the occurrence of a fault. As a consequence, it is possible to define some stability regions R_I^z for the interconnecting variable z_I.

As previously mentioned, the DFDI architecture is made of N communicating Local Fault Diagnosers (LFDs) L_I, which monitor each of the N subsystems the global system has been decomposed into, by using a Model Based Analytical Redundancy Relation approach. The FDAE estimator will be based on the nominal healthy model and will provide the detection capability. The remaining FIE estimators provide the isolation

capability, and are based on models matched to each one of the N_{F_I} elements of the fault class F_I. Each LFD is allowed to take only local measurements of the local state and the local control vectors, and to communicate with the neighboring LFDs in $\mathcal{I}_I$ in order to populate the interconnection vector z_I.

It will be assumed that a noisy version of the local state x_I is sensed:

$$y_I(k) := x_I(k) + \xi_I(k),$$

where $\xi_I(k)$ is an unknown function that represents the uncertainty associated to the process of measuring $x_I(k)$ by each LFD. Thus, a LFD will receive from its neighbors the vector:

$$v_I(k) := z_I(k) + \zeta_I(k),$$

with $\zeta_I(k)$ made with the components of $\xi_J, J \in \mathcal{I}_I$ affecting the relevant components of the measurements y_I. It is assumed that the measuring uncertainties ξ_I and ζ_I are unstructured and unknown, but are bounded by some known quantity:

$$|\xi_I^{(i)}(k)| \leq \bar{\xi}_I^{(i)}, \quad |\zeta_I^{(i)}(k)| \leq \bar{\zeta}_I^{(i)}.$$

A shared variable $x^{(s)}$ will be measured by distinct LFDs in the overlap set $\mathcal{O}_s$ with different uncertainties, as well as the interconnection part of the local model. As a consequence, it will be convenient for LFDs in the overlap set to employ consensus techniques when implementing their FDAE and FIE estimators, in order to reduce the effect of the unfavorable conditions.

Healthy Behavior and Fault Detection Estimator

The FDAE monitors the subsystem $\mathcal{S}_I$, providing a local state estimate $\hat{x}_{I,0}$ of the local state x_I. Thus, the estimation error $\tilde{x}_{I,0} := y_I - \hat{x}_{I,0}$ is compared to a suitable detection threshold $\bar{x}_{I,0}$. If the following condition is met, the system is said to be healthy:

$$|\tilde{x}_{I,0}^{(i)}(k)| \leq \bar{x}_{I,0}^{(i)}(k).$$

A fault affecting the I-th subsystem will be detected by its LFD at the first time instant such that $\mathcal{S}_I$ becomes non-empty, called the *fault detection time* k_d.

The estimator dynamics for the component $\hat{x}_{I,0}^{(s_I)}$ computed by the I-th LFD, $I \in \mathcal{O}_s$, for $k < k_d$ is:

$$\begin{aligned}\hat{x}_{I,0}^{(s_I)}(k+1) &= \lambda\Big\{\hat{x}_{I,0}^{(s_I)}(k) - y_I^{(s_I)}(k) + \textstyle\sum_{J\in\mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}[\hat{x}_{J,0}^{(s_J)}(k) - \hat{x}_{I,0}^{(s_I)}(k)]\Big\} \\ &\quad + \textstyle\sum_{J\in\mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}[f_J^{(s_J)}(y_J(k), u_J(k)) + \hat{g}_J^{(s_J)}(y_J(k), v_J(k), u_J(k), \hat{\theta}_{J,0})]\end{aligned}$$

where $0 < \lambda < 1$ and $W_{\mathcal{S}}^{(I,J)}$ is a weighted doubly-stochastic adjacency matrix that implements the consensus protocol on x_s. The term $\hat{g}_J^{s_J}$ is the s_J-th output of an adaptive approximator meant to learn the interconnection function g_J, and $\hat{\theta}_J$ denotes its adjustable parameters vector, which is updated according to the following learning law:

$$\hat{\theta}_{J,0}(k+1) = P_{\hat{\Theta}_{J,0}}(\hat{\theta}_{J,0}(k) + \gamma_{J,0}(k) H_{J,0}^T(k) r_{J,0}(k+1)),$$

where $H_{J,0}(k) := \frac{\partial \hat{g}_J(k)}{\partial \hat{\theta}_{J,0}}$ is the gradient matrix of the on-line approximator with respect to its adjustable parameters, $P_{\hat{\Theta}_{J,0}}$ is a projection operator, and $r_{J,0}(k+1)$ is the signal:

$$r_{J,0}(k+1) = \tilde{x}_{J,0}(k+1) - \lambda \tilde{x}_{J,0}(k).$$

The learning rate $\gamma_{J,0}(k)$ is computed at each step as:

$$\gamma_{J,0}(k) := \frac{\mu_{J,0}}{\varepsilon_{J,0} + \|H_{J,0}^T(k)\|_F^2}, \quad \varepsilon_{J,0} > 0, \quad 0 < \mu_{J,0} < 2.$$

For $k < k_0 < k_d$, the dynamics of the LFD estimation error component $\tilde{x}_{I,0}^{(s_I)}$ can be written as:

$$\begin{aligned}\tilde{x}_{I,0}^{(s_I)}(k+1) &= \lambda\left\{\tilde{x}_{I,0}^{(s_I)}(k) + \textstyle\sum_{J\in\mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}[\tilde{x}_{J,0}^{(s_J)}(k) - \tilde{x}_{I,0}^{(s_I)}(k) + \xi_I^{(s_I)}(k) - \xi_J^{(s_J)}(k)]\right\} \\ &\quad + \textstyle\sum_{J\in\mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}[f_J^{(s_J)}(x_J(k), u_J(k)) - f_J^{(s_J)}(y_J(k), u_J(k)) + g_J^{(s_J)}(k) - \hat{g}_J^{(s_J)}(k)] \\ &\quad + \xi_{I,0}^{(s_I)}(k+1)\end{aligned}$$

As by assumption $\sum_{I\neq J} W_{\mathcal{S}}^{(I,J)} = 1 - W_{\mathcal{S}}^{(I,I)}$, it holds:

$$\begin{aligned}\tilde{x}_{I,0}^{(s_I)}(k+1) &= \textstyle\sum_{J\in\mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}\{\lambda[\tilde{x}_{J,0}^{(s_J)}(k) - \xi_{J,0}^{(s_J)}(k)]\Delta f_J^{(s_J)}(k) + \Delta g_J^{(s_J)}(k)\} \\ &\quad + \lambda\xi_I^{(s_I)}(k) + \xi_I^{(s_I)}(k+1)\end{aligned}$$

where Δf_I and Δg_I are defined as:

$$\Delta f_I(k) := f_I(x_I(k), u_I(k)) - f_I(y_I(k), u_I(k))$$
$$\Delta g_I(k) := g_I(x_I(k), z_I(k), u_I(k)) - g_I(y_I(k), v_I(k), u_I(k), \hat{\theta}_{I,0}),$$

both generally assuming non-zero values due to uncertainties. To formalize them, an optimal weight vector $\hat{\theta}_{I,0}^*$ and a Minimum Functional Approximation Error (MFAE) v_I are introduced:

$$\hat{\theta}_{I,0}^* := \arg \min_{\hat{\theta}_{I,0}\in\Theta_{I,0}} \sup_{R_I} \|g_I(x_I(k), z_I(k), u_I(k)) - \hat{g}_I(x_I(k), z_I(k), u_I(k), \hat{\theta}_{I,0})\|,$$

$$v_I(k) := g_I(x_I(k), z_I(k), u_I(k)) - \hat{g}_I(x_I(k), z_I(k), u_I(k), \hat{\theta}_{I,0}).$$

Thus, the parameter estimation error is defined as $\tilde{\theta}_{I,0} = \hat{\theta}_{I,0}^* - \hat{\theta}_{I,0}$, and the function is introduced:

$$\Delta\hat{g}_I(k) := \hat{g}_I(x_I(k), z_I(k), u_I(k), \hat{\theta}_{I,0}) - \hat{g}_I(y_I(k), v_I(k), u_I(k), \hat{\theta}_{I,0}),$$

so as to obtain:

$$\Delta g_I(k) = H_{I,0}\tilde{\theta}_{I,0} + v_I(k) + \Delta\hat{g}_I(k).$$

Considering the estimator dynamics of the component $\hat{x}_{I,0}^{(s_I)}$, the dynamics of the LFD estimation error component $\tilde{x}_{I,0}^{(s_I)}$ before the occurrence of a fault can be written as:

$$\tilde{x}_{I,0}^{(s_I)}(k+1) = \sum_{J \in \mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}[\lambda \tilde{x}_{J,0}^{(s_J)}(k) + \chi_J^{(s_J)}(k)] + \lambda \xi_I^{(s_I)}(k) + \xi_I^{(s_I)}(k+1),$$

where an uncertainty term $\chi_I^{(s_I)}$ has been introduced, defined as:

$$\chi_I^{(s_I)}(k) := \Delta f_I^{(s_I)}(k) + \lambda \xi_I^{(s_I)}(k) + \Delta g_I^{(s_I)}(k).$$

In order to study the behavior of $\tilde{x}_{I,0}^{(s_I)}(k)$ and define the threshold $\bar{x}_{I,0}^{(s_I)}(k)$, the following vectors are introduced, related to the detection estimator of all the LFDs sharing the variable $x^{(s)}$:

$$\tilde{x}_{s,0}(k) := \text{col}(\tilde{x}_{I,0}^{(s_I)}, I \in \mathcal{O}_s),$$
$$\chi_s(k) := \text{col}(\chi_I^{(s_I)}, I \in \mathcal{O}_s),$$
$$\xi_s(k) := \text{col}(\xi_I^{(s_I)}, I \in \mathcal{O}_s).$$

The FDAE estimation error dynamics of all the LFDs in $\mathcal{O}_s$ can then be written as:

$$\tilde{x}_{s,0}(k+1) = W_{\mathcal{S}}[\lambda \tilde{x}_{s,0}(k) + \chi_s(k)] + \lambda \xi_s(k) + \xi_s(k+1).$$

This represents the dynamics of a stable Linear Time Invariant (LTI) discrete-time system with all the eigenvalues inside a circle of radius $\lambda < 1$, which component-wise solution is:

$$\begin{aligned} \tilde{x}_{I,0}^{(s_I)}(k) &\equiv \tilde{x}_{s,0}^{(s_I)}(k) \\ &= w_{\mathcal{S},I}^T \Big\{ \lambda \Big[\textstyle\sum_{h=0}^{k-2} (\lambda W_{\mathcal{S}})^{k-2-h} (W_{\mathcal{S}} \chi_s(h) + \lambda \xi_s(h) + \xi_s(h+1)) + \lambda^{k-1} W_{\mathcal{S}}^{k-1} \tilde{x}_{s,0} \Big] \\ &\quad + \chi_s(k-1) \Big\} + \lambda \xi_s(k-1) + \xi_s^{(I)}(k) \end{aligned}$$

where $w_{\mathcal{S},I}$ corresponds to the I-th row of matrix $W_{\mathcal{S}}$.

The absolute value of the estimation error for $k < k_0$ can be upper bounded by relying on the triangular inequality:

$$\begin{aligned} |\tilde{x}_{I,0}^{(s_I)}(k+1)| &\leq \textstyle\sum_{J \in \mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}[\lambda |\tilde{x}_{J,0}^{(s_J)}(k)| + |\chi_J^{(s_J)}(k)|] + \lambda |\xi_I^{(s_I)}(k)| + |\xi_I^{(s_I)}(k+1)| \\ &\leq \textstyle\sum_{J \in \mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}[\lambda |\tilde{x}_{J,0}^{(s_J)}(k)| + \bar{\chi}_J^{(s_J)}(k)] + \lambda \bar{\xi}_I^{(s_I)}(k) + \bar{\xi}_I^{(s_I)}(k+1) \end{aligned}$$

where it is defined:

$$\bar{\chi}_J^{(s_J)}(k) := \max_{\xi_J} |\Delta f_J^{(s_J)}(k)| + \|H_{J,0}\| \kappa_{J,0}(\hat{\theta}_{J,0}) + \bar{v}_J(k) + \lambda \bar{\xi}_J^{(s_J)}(k) + \max_{\xi_J} \max_{\zeta_J} |\Delta \hat{g}_J(k)|,$$

with $\kappa_{J,0}(\hat{\theta}_{J,0}) \geq \|\tilde{\theta}_{J,0}\|$.

Thus, considering

$$|\tilde{x}_{s,0}| \equiv \text{col}(|\tilde{x}_{I,0}^{(s_I)}| : I \in \mathcal{O}_s),$$
$$|\tilde{x}_{s,0}(k+1)| \leq W_{\mathcal{S}}[\lambda |\tilde{x}_{s,0}(k)| + \bar{\chi}_s(k)] + \lambda \bar{\xi}_s(k) + \bar{\xi}_s(k+1).$$

The absolute value of each component of $\tilde{x}_s$ can be bounded by the corresponding component of $\bar{x}_s$, defined as the solution of the following equation:

$$\bar{x}_s(k+1) = W_{\mathcal{S}}[\lambda \bar{x}_s(k) + \bar{\chi}_s(k)] + \lambda \bar{\xi}_s(k) + \bar{\xi}_s(k+1),$$

with initial conditions:

$$\bar{x}_s(0) := \text{col}(\bar{\xi}_I^{(s_I)}(0) : I \in \mathcal{O}_s).$$

The main property of the threshold is it robustness with respect to all the modeling and measuring uncertainties, so that the absence of false-positive fault detections is guaranteed.

For a non-shared component $x_{I,0}^{(j)}$ the estimator equation, the error equation and the threshold become:

$$\hat{x}_{I,0}^{(j)}(k+1) = \lambda[\hat{x}_I^{(j)}(k) - y_I^{(j)}(k)] + f_I^{(j)}(y_I(k), u_I(k)) + \hat{g}_I^{(j)}(k)$$

$$\tilde{x}_{I,0}^{(j)}(k+1) = [\lambda \tilde{x}_{I,0}^{(j)}(k) - \chi_I^{(j)}(k)] + \lambda \xi_I^{(j)}(k) + \xi_I^{(j)}(k+1)$$

$$\bar{x}_{I,0}^{(j)}(0) := \bar{\xi}_I^{(j)}(0), \quad \bar{x}_{I,0}^{(j)}(k+1) := \lambda \bar{x}_{I,0}^{(j)}(k) + \bar{\chi}_I^{(j)}(k) + \lambda \bar{\xi}_I^{(j)}(k) + \bar{\xi}_I^{(j)}(k+1).$$

Faulty Behavior and Fault Detectability

For $k \geq k_0$, the error dynamics equation for a shared component becomes:

$$\tilde{x}_{s,0}(k+1) = W_{\mathcal{S}}[\lambda \tilde{x}_{s,0}(k) + \chi_s(k)] + (1 - b^{-(k-k_0)})\phi_s(k) + \lambda \xi_s(k) + \xi_s(k+1).$$

If there exist a time index $k_1 > k_0$ and a subsystem $\mathcal{S}_I$ such that the fault ϕ_I fulfills the following inequality for at least one component s_I:

$$\left| \sum_{h=k-0}^{k_1-1} \lambda^{k_1-1-h}(1 - b^{-(h-k_0)})\phi^{(s)}(h) \right| > 2\bar{x}_{I,0}^{(s_I)}(k_1),$$

then it will be detected at k_1, that is $|\tilde{x}_{I,0}^{(s_I)}(k_1)| > \bar{x}_{I,0}^{(s_I)}(k_1)$.

Faulty Isolation

After a fault has been detected at time k_d, the learning of the FDAE interconnection adaptive approximator $\hat{g}_I(k)$ of every LFD is stopped, i.e. $\hat{\theta}_{I,0}(k) = \hat{\theta}_{I,0}(k_d), \forall k \geq k_d$, to prevent the interconnection approximator to learn part of the fault function ϕ_I. The N_{F_I} FIEs allow to test in parallel the N_{F_I} fault hypotheses. Thus, the l-th FIE will provide its own local state estimate $\hat{x}_{I,l}$ of the local state x_I. The difference between the estimate and the local measurements y_I will yield the following estimation error:

$$\tilde{x}_{I,l} = y_I - \hat{x}_{I,l},$$

which will be compared, component by component, to a suitable detection threshold:

$$|\tilde{x}_{I,l}^{(i)}(k)| \leq \bar{x}_{I,l}^{(i)}(k), \quad \forall i.$$

Should this condition be unmet at some time instant k, the fault hypothesis will be falsified and the corresponding fault will be excluded as a possible cause of the signature, at the exclusion time $k_{e,I,l}$, defined as:

$$k_{e,I,l} := \min\left\{k : \exists i, i \in \{1,\dots,n_I\}, |\tilde{x}_{I,l}^{(i)}(k)| > \bar{x}_{I,l}^{(i)}(k)\right\}.$$

Again, the goal of the isolation logic is to exclude every but one fault, which may be said to be isolated. If a fault is local, then having the corresponding LFD exclude every but that fault is sufficient for declaring it isolated. But, for distributed faults, the isolation needs that all the LFDs having a local part of it in their fault class did exclude all their other faults. Formally, we say that a fault $\phi_{I,p} \in F_I$ is locally isolated at time k if and only if $\forall l,\ l \in \{1,\dots,N_{F_I}\} - p,\ k_{e,I,l} \leq k$ and $\nexists k_{e,I,p}$. Furthermore, $k_{lis,I,p} := \min\{k_{e,I,l},\ l \in \{1,\dots,N_{F_I}\} - p\}$ is the local fault isolation time. A fault $\phi_{I,p} \in F_I$ is isolated if for each LFD the corresponding local functions $\phi_{J,p}$ either have been isolated or do not exist. Moreover, $k_{is,I,p} := \min\{k_{lis,J,l},\ J \in \{1,\dots,N\}\}$ is the fault isolation time.

If every fault in F_I is excluded, the following explanations may be given:

- an unknown fault, either local or distributed, has been isolated;
- the fault detection was triggered by a local or distributed fault of another subsystem.

After the fault $\phi(k)$ has occurred, the state equation of the s_I-th component of the I-th subsystem becomes:

$$x_I^{(s_I)}(k+1) = f_I^{(s_I)}(x_I(k), u_I(k)) + g_I^{(s_I)}(k) + \beta(k-k_0)\phi^{(s)}(x(k), u(k)).$$

The l-th FIE estimator dynamic equation for a shared variable is defined as:

$$\begin{aligned}\hat{x}_{I,l}^{(s_I)}(k+1) \;=\; & \lambda\left\{\hat{x}_{I,l}^{(s_I)}(k) - y_{I,l}^{(s_I)}(k) + \textstyle\sum_{J\in\mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}[\hat{x}_{J,l}^{(s_J)}(k) - \hat{x}_{I,l}^{(s_I)}(k)]\right\} \\ & + \textstyle\sum_{J\in\mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}[f_J^{(s_J)}(y_J(k), u_J(k)) + \hat{g}_J^{(s_J)}(k) + \hat{\phi}_{J,l}^{(s_J)}(y_J(k), v_J(k), u_J(k), \hat{\theta}_{J,l})]\end{aligned}$$

where $\hat{\phi}_{J,l}^{(s_J)}(y_J(k), v_J(k), u_J(k), \hat{\theta}_{J,l}) := (\theta_{J,l,s_J})^T H_{J,l,s_J}(y_J(k), v_J(k), u_J(k))$ is the s_J-th component of a linearly-parametrized function that matches the structure of the l-th fault function $\phi_{J,l}$. The parameters vectors are updated according to the following learning law:

$$\hat{\theta}_{J,l,k}(k+1) = P_{\hat{\Theta}_{J,l,k}}(\hat{\theta}_{J,l,k}(k) + \gamma_{J,l,k}(k) H_{J,l,k}^T(k) r_{J,l,k}(k+1)),$$

where $r_{J,l,k}(k+1) = \tilde{x}_{J,l,k}(k+1) - \lambda\tilde{x}_{J,l,k}(k)$ and $P_{\hat{\Theta}_{J,l,k}}$ is the projection operator:

$$P_{\hat{\Theta}_{J,l,k}} := \begin{cases} \hat{\theta}_{J,l,k} & \text{if } |\hat{\theta}_{J,l,k}| \leq M_{\hat{\Theta}_{J,l,k}} \\ \frac{M_{\hat{\Theta}_{J,l,k}}}{|\hat{\theta}_{J,l,k}|}\hat{\theta}_{J,l,k} & \text{if } |\hat{\theta}_{J,l,k}| > M_{\hat{\Theta}_{J,l,k}} \end{cases}$$

The learning rate is computed at each step as:

$$\gamma_{J,l,k}(k) := \frac{\mu_{J,l,k}}{\varepsilon_{J,l,k} + \|H_{J,l,k}^T(k)\|^2}, \quad \varepsilon_{J,l,k} > 0, \quad 0 < \mu_{J,l,k} < 2.$$

The corresponding estimation error dynamic equation is:

$$\begin{aligned}\tilde{x}_{I,l}^{(s_I)}(k+1) &= \lambda\left\{\tilde{x}_{I,l}^{(s_I)}(k)+\textstyle\sum_{J\in\mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}[\tilde{x}_{J,l}^{(s_J)}(k)-\tilde{x}_{I,l}^{(s_I)}(k)+\xi_I^{(s_I)}(k)-\xi_J^{(s_J)}(k)]\right\}\\ &\quad+\textstyle\sum_{J\in\mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}[\Delta f_J^{(s_J)}(k)+\Delta g_J^{(s_J)}(k)+(1-b^{-(k-k_0)})\phi^{(s)}(k)-\hat{\phi}_{J,l}^{(s_J)}(k)]\\ &\quad+\xi_I^{(s_I)}(k+1)\\ &= \textstyle\sum_{J\in\mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}[\lambda\tilde{x}_{J,l}^{(s_J)}(k)+\chi_J^{(s_J)}(k)+(1-b^{-(k-k_0)})\phi^{(s)}(k)-\hat{\phi}_{J,l}^{(s_J)}(k)]\\ &\quad+\lambda\xi_I^{(s_I)}(k)+\xi_I^{(s_I)}(k+1)\end{aligned}$$

Considering a matched fault $\phi^{(s)}(k) := \phi_{J,l}^{(s_J)}(x_J(k), z_J(k), u_J(k), \theta_{J,l})$:

$$\begin{aligned}\tilde{x}_{I,l}^{(s_I)}(k+1) &= \textstyle\sum_{J\in\mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}[\lambda\tilde{x}_{J,l}^{(s_J)}(k)+\chi_J^{(s_J)}(k)\\ &\quad+(1-b^{-(k-k_0)})(H_{J,l,s_J}^T(k)\theta_{J,l,s_J}+\Delta H_{J,l,s_J}^T(k)\theta_{J,l,s_J})-H_{J,l,s_J}^T(k)\hat{\theta}_{J,l,s_J}]\\ &\quad+\lambda\xi_I^{(s_I)}(k)+\xi_I^{(s_I)}(k+1)\end{aligned}$$

where:

$$\Delta H_{J,l,s_J}^T(k) := H_{J,l,s_J}(x_J(k), z_J(k), u_J(k)) - H_{J,l,s_J}(y_J(k), v_J(k), u_J(k)).$$

By introducing the parameter estimation error $\tilde{\theta}_{J,l,s_J} := \theta_{J,l,s_J} - \hat{\theta}_{J,l,s_J}$, the FIE estimation error equation for a matched fault becomes:

$$\begin{aligned}\tilde{x}_{I,l}^{(s_I)}(k+1) &= \textstyle\sum_{J\in\mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}[\lambda\tilde{x}_{J,l}^{(s_J)}(k)+\chi_J^{(s_J)}(k)+(1-b^{-(k-k_0)})H_{J,l,s_J}^T(k)\tilde{\theta}_{J,l,s_J}\\ &\quad+(1-b^{-(k-k_0)})\Delta H_{J,l,s_J}^T(k)\theta_{J,l,s_J})-b^{-(k-k_0)}H_{J,l,s_J}^T(k)\hat{\theta}_{J,l,s_J}]\\ &\quad+\lambda\xi_I^{(s_I)}(k)+\xi_I^{(s_I)}(k+1)\end{aligned}$$

so that its absolute value can be bounded by a threshold that is solution of the following:

$$\begin{aligned}\bar{x}_{I,l}^{(s_I)}(k+1) &= \textstyle\sum_{J\in\mathcal{O}_s} W_{\mathcal{S}}^{(I,J)}[\lambda\bar{x}_{J,l}^{(s_J)}(k)+\bar{\chi}_J^{(s_J)}(k)+\|H_{J,l,s_J}(k)\|\kappa_{J,l,s_J}(\hat{\theta}_{J,l,s_J})\\ &\quad+\|\Delta H\|_{J,l,s_J}(k)\bar{\theta}_{J,l,s_J})-\bar{b}^{-(k-k_0)}\|H_{J,l,s_J}(k)\|\|\hat{\theta}_{J,l,s_J}\|]\\ &\quad+\lambda\bar{\xi}_I^{(s_I)}(k)+\bar{\xi}_I^{(s_I)}(k+1)\end{aligned}$$

The error and threshold solutions can be conveniently written by introducing the vectors:

$$\begin{aligned}\tilde{x}_{s,l}(k) &:= \mathrm{col}(\tilde{x}_{I,l}^{(s_I)}, I\in\mathcal{O}_s),\\ \chi_s(k) &:= \mathrm{col}(\chi_I^{(s_I)}, I\in\mathcal{O}_s),\\ \bar{x}_{s,l}(k) &:= \mathrm{col}(\bar{x}_{I,l}^{(s_I)}, I\in\mathcal{O}_s),\end{aligned}$$

so that it holds:

$$\begin{aligned}\tilde{x}_{s,l}(k+1) &= W_{\mathcal{S}}\Big[\lambda\tilde{x}_{s,l}(k)+\chi_s(k)+\text{col}((1-b^{-(k-k_0)})H^T_{I,l,s_I}(k)\tilde{\theta}_{I,l,s_I}\\ &\quad +(1-b^{-(k-k_0)})\Delta H^T_{I,l,s_I}(k)\theta_{I,l,s_I}-b^{-(k-k_0)}H^T_{I,l,s_I}(k)\hat{\theta}_{I,l,s_I})\,\Big]\\ &\quad +\lambda\xi_s(k)+\xi_s(k+1)\end{aligned}$$

which component-wise solution is:

$$\begin{aligned}\tilde{x}^{(s_I)}_{I,l}(k) &= w_{\mathcal{S},I}\textstyle\sum_{h=k_d}^{k-1}(\lambda W_S)^{k-1-h}\Big[\chi_s(h)+\text{col}((1-b^{-(h-k_0)})H^T_{I,l,s_I}(h)\tilde{\theta}_{I,l,s_I}\\ &\quad +(1-b^{-(h-k_0)})\Delta H^T_{I,l,s_I}(h)\theta_{I,l,s_I}-b^{-(h-k_0)}H^T_{I,l,s_I}(h)\hat{\theta}_{I,l,s_I})\,\Big]\\ &\quad +\lambda w_{\mathcal{S},I}\textstyle\sum_{h=k_d}^{k-2}\big[(\lambda W_{\mathcal{S}})^{k-2-h}(\lambda\xi_s(h)+\xi_s(h+1))\big]+\lambda\xi^{(s_I)}_s(k-1)\\ &\quad +\xi^{(s_I)}_s(k)+\lambda w_{S,I}(\lambda W_{\mathcal{S}})^{k-1-k_d}\tilde{x}_{s,l}(k_d)\end{aligned}$$

$$\begin{aligned}\bar{x}^{(s_I)}_{I,l}(k) &= w_{\mathcal{S},I}\textstyle\sum_{h=k_d}^{k-1}(\lambda W_S)^{k-1-h}\Big[\bar{\chi}_s(h)+\text{col}(\|H_{I,l,s_I}(k)\|\kappa_{I,l,s_I}(\hat{\theta}_{I,l,s_I})\\ &\quad +\Delta\bar{H}^T_{I,l,s_I}(k)\bar{\theta}_{I,l,s_I}-\bar{b}^{-(h-k_d)}\|H_{I,l,s_I}(k)\|\|\hat{\theta}_{I,l,s_I})\|\,\Big]\\ &\quad +\lambda w_{\mathcal{S},I}\textstyle\sum_{h=k_d}^{k-2}\big[(\lambda W_{\mathcal{S}})^{k-2-h}(\lambda\bar{\xi}_s(k)+\bar{\xi}_s(k+1))\big]+\lambda\bar{\xi}^{(s_I)}_s(k-1)\\ &\quad +\bar{\xi}^{(s_I)}_s(k)+\lambda w_{\mathcal{S},I}(\lambda W_{\mathcal{S}})^{k-1-k_d}\bar{x}_{s,l}(k_d)\end{aligned}$$

Considering an unmatched fault $\phi^{(s_I)}_I(x_I(k),z_I(k),u_I(k))=\phi^{(s_I)}_{I,p}(x_I(k),z_I(k),u_I(k),\theta_{I,p})$, with $p\neq l$, the dynamics of the s_I-th component of the estimation error of the l-th FIE of the I-th LFD is:

$$\begin{aligned}\tilde{x}^{(s_I)}_{I,l}(k+1) &= \textstyle\sum_{J\in\mathcal{O}_s}W^{(I,J)}_{\mathcal{S}}[\lambda\tilde{x}^{(s_J)}_{J,l}(k)+\chi^{(s_J)}_J(k)+(1-b^{-(k-k_0)})\phi^{(s_I)}_{I,p}(x_I(k),z_I(k),u_I(k),\theta_{I,p})\\ &\quad -\hat{\phi}^{(s_J)}_{J,l}(y_J(k),v_J(k),u_J(k),\hat{\theta}_{J,l})+\lambda\xi^{(s_I)}_I(k)+\xi^{(s_I)}_I(k+1)\end{aligned}$$

A mismatch vector, defined as follows, is introduced:

$$\Delta_{s,l}\phi_{I,p}(k):=\text{col}((1-b^{-(k-k_0)})\phi^{(s_I)}_{I,p}(k))-\hat{\phi}_{s,l}(k),$$

the dynamics of vector $\tilde{x}_{s,l}$ can be written as:

$$\tilde{x}_{s,l}(k+1)=W_{\mathcal{S}}[\lambda\tilde{x}_{s,l}(k)+\chi_s(k)+\Delta_{s,l}\phi_{I,p}(k)]+\lambda\xi_s(k)+\xi_s(k+1),$$

which solution is:

$$\begin{aligned}\tilde{x}_{s,l}(k) &= \textstyle\sum_{h=k_d}^{k-1}(\lambda W_{\mathcal{S}})^{k-1-h}W_{\mathcal{S}}[\chi_s(h)+\Delta_{s,l}\phi_{I,p}(h)]\\ &\quad +\textstyle\sum_{h=k_d}^{k-1}[(\lambda W_{\mathcal{S}})^{k-1-h}(\lambda\xi_s(h)+\xi_s(h+1))]+(\lambda W_{\mathcal{S}})^{k-k_d}\tilde{x}_{s,l}(k_d)\end{aligned}$$

Component-wise it is:

$$\begin{aligned}\tilde{x}^{(s_I)}_{I,l}(k) &= w_{\mathcal{S},I}\textstyle\sum_{h=k_d}^{k-1}(\lambda W_{\mathcal{S}})^{k-1-h}[\chi_s(h)+\Delta_{s,l}\phi_{I,p}(h)]\\ &\quad +\lambda w_{\mathcal{S},I}\textstyle\sum_{h=k_d}^{k-2}[(\lambda W_{\mathcal{S}})^{k-2-h}(\lambda\xi_s(h)+\xi_s(h+1))]+\lambda\xi^{(s_I)}_s(k-1)\\ &\quad +\xi^{(s_I)}_s(k)+\lambda w_{\mathcal{S},I}(\lambda W_{\mathcal{S}})^{k-1-k_d}\tilde{x}_{s,l}(k_d)\end{aligned}$$

Given a fault $\phi_{I,p} \in F_I$, for each $l \in \{1, \dots, N_{F_I}\} - p$ there exists some time instant $k_l > k_d$ and some $s_I \in \{1, \dots, n_I\}$ such that the following inequality holds:

$$\begin{aligned} w_{\mathcal{S},I} \sum_{h=k_d}^{k_l-1} (\lambda W_{\mathcal{S}})^{k-1-h} \Delta_{s,l} \phi_{I,p}(h) \quad > \quad & w_{\mathcal{S},I} \sum_{h=k_d}^{k_l-1} [\bar{\chi}_s(h) + \text{col}(\|H_{I,l,s_I}(k)\| \kappa_{I,l,s_I}(\hat{\theta}_{I,l,s_I}) \\ & \Delta \bar{H}^T_{I,l,s_I}(k) \bar{\theta}_{I,l,s_I} - \bar{b}^{(-(h-k_d)} \|H_{I,l,s_I}(k)\| \|\hat{\theta}_{I,l,s_I}\|)] \\ & + 2 \Big\{ \lambda w_{\mathcal{S},I} \sum_{h=k_d}^{k-2} [(\lambda W_{\mathcal{S}})^{k-2-h} (\lambda \bar{\xi}_s(k) \\ & + \bar{\xi}_s(k+1))] + \lambda \bar{\xi}_s^{(s_I)}(k-1) \\ & \bar{\xi}_s^{(s_I)}(k) + \lambda w_{\mathcal{S},I} (\lambda W_{\mathcal{S}})^{k-1-k_d} \tilde{x}_{s,l}(k_d) \Big\} \end{aligned}$$

Then, the p-th fault will be isolated. Furthermore, the local isolation time is upper-bounded.

3.4 Threshold Computation and Residual Evaluation

A further problem in the Fault Diagnosis techniques is to establish the thresholds which overcome determines the occurrence of a fault. Such problem becomes more complex when taking into account that residual signals are generally corrupted with disturbances and uncertainties.

As exposed in [49], two residual evaluation strategies have been developed: the *statistic testing*, and the *norm-based residual evaluation*. The latter allows a systematic threshold computation exploiting the robust control theory, and is here analyzed.

For a given signal Υ, the upper limit monitoring can be performed on the signal itself, on its trend, i.e. by studying $\dot{\Upsilon}$, or on the Root-Mean-Square (RMS) $\|\cdot(t)\|_{RSM} = \left(\frac{1}{T}\int_t^{t+T} \|\cdot(\tau)\|^2 d\tau\right)^{\frac{1}{2}}$:

$$\begin{aligned} \Upsilon \leq \Upsilon_{max} \quad &\Longrightarrow \quad \text{fault-free} \\ \Upsilon > \Upsilon_{max} \quad &\Longrightarrow \quad \text{a fault is detected} \end{aligned}$$

$$\begin{aligned} \dot{\Upsilon} \leq \dot{\Upsilon}_{max} \quad &\Longrightarrow \quad \text{fault-free} \\ \dot{\Upsilon} > \dot{\Upsilon}_{max} \quad &\Longrightarrow \quad \text{a fault is detected} \end{aligned}$$

$$\begin{aligned} \|\Upsilon\|_{RMS} \leq \|\Upsilon\|_{RMS,max} \quad &\Longrightarrow \quad \text{fault-free} \\ \|\Upsilon\|_{RMS} > \|\Upsilon\|_{RMS,max} \quad &\Longrightarrow \quad \text{a fault is detected} \end{aligned}$$

where Υ_{max}, $\dot{\Upsilon}_{max}$ and $\|\Upsilon\|_{RMS,max}$ denote the maximum value for Υ, $\dot{\Upsilon}$ and $\|\Upsilon\|_{RMS}$, respectively, in the fault-free conditions, and are thereby the *thresholds*.

To reduce the influence of noise in the continuous case, the average value of the signal over a time interval $[t,\ t+T]$ may also be used:

$$\bar{\Upsilon}(t) = \frac{1}{T}\int_t^{t+T} \Upsilon(\tau) d\tau$$

$$\begin{aligned} \bar{\Upsilon} \leq \bar{\Upsilon}_{max} \quad &\Longrightarrow \quad \text{fault-free} \\ \bar{\Upsilon} > \bar{\Upsilon}_{max} \quad &\Longrightarrow \quad \text{a fault is detected} \end{aligned}$$

where $\bar{\Upsilon}_{max}$ is the maximum admissible value of $\bar{\Upsilon}$.

Thereby, an evaluation function is firstly defined to study the desired signal, and on such basis a threshold is defined. For the Fault Diagnosis purpose in discrete-time systems, two features can be isolated from the residual signal $r(k) \in \mathcal{R}^{k_r}$:

The *peak value*:

$$J_{peak} = \|r\|_{peak} := \sup_{k\geq 0} \|r(k)\|, \quad \|r(k)\| = \sqrt{\sum_{i=1}^{k_r} r_i^2(k)}$$

For which the limit monitoring problem becomes:

$$\begin{array}{rcl} J_{peak} \leq J_{th,peak} & \Longrightarrow & \text{fault-free} \\ J_{peak} > J_{th,peak} & \Longrightarrow & \text{a fault is detected} \end{array}$$

where

$$J_{th,peak} = sup_{fualt-free} \|r(k)\|_{peak}.$$

Similarly, considering the residuals' trend, it is obtained:

$$J_{trend} = \|\Delta r(k)\|_{peak} := \sup_{k\geq 0} \|\Delta r(k)\|, \quad J_{th,peak} = sup_{fualt-free} \|\Delta r(k)\|_{peak},$$

then

$$\begin{array}{rcl} J_{trend} \leq J_{th,trend} & \Longrightarrow & \text{fault-free} \\ J_{trend} > J_{th,trend} & \Longrightarrow & \text{a fault is detected} \end{array}$$

and for the average value evaluation:

$$J_{average} = \|r(k)\|_{average} := \sup_{k\geq 0} \|\bar{r}(k)\|_{peak}, \quad \bar{r}(k) = \frac{1}{N}\sum_{j=1}^{N} r(k+j)$$

$$J_{th,average} = sup_{fualt-free} \|r(k)\|_{average},$$

with

$$\begin{array}{rcl} J_{average} \leq J_{th,average} & \Longrightarrow & \text{fault-free} \\ J_{average} > J_{th,average} & \Longrightarrow & \text{a fault is detected} \end{array}$$

The *RMS value*, which for the discrete case is computed as

$$J_{RMS} = \|r(k)\|_{RMS} = \sqrt{\frac{1}{N}\sum_{j=1}^{N} \|r(k+j)\|^2}$$

and measures the average energy of signal r over a discrete-time interval $(k, k+N)$. Thereby, the threshold is defined as $J_{th,RMS} = sup_{fault-free} \|r\|_{RMS}$, as the following inequality is always valid:

$$\|r(k)\|_{RMS}^2 \leq \frac{1}{N} \|r(k)\|_2^2.$$

Thus,

$$\begin{aligned} J_{RMS} \leq J_{th,RMS} &\implies \text{fault-free} \\ J_{RMS} > J_{th,RMS} &\implies \text{a fault is detected} \end{aligned}$$

As forementioned, the thresholds are to be tailored on specific features of the residuals, making it possible to distinguish between actual faults and disturbances or uncertainties. Although, this problem is not here addressed as it is beyond the scope of this study, and further details are provided in [49].

Chapter 4

Cyber Threats and Malicious Attacks

The aim of a secure system is to guarantee access to information to authorized users, ensuring availability conditions, preventing unauthorized access by outsiders and avoiding any alteration of the data in the system. On the other side, the aim of an attacker is to take control or manipulate the system parameters in order to generate threats or reduce the availability of data and/or resources to the operators. The effects of a cyber-attack on the physical system can be several and may lead to the degradation of the operability, performance and efficiency of the targeted system, or even to its damage or destruction.

4.1 The Cyber Domain in Industrial Control Systems

In the recent years, the focus on the Information and Communications Technology (ICT) security has grown rapidly due to the evolution of the computer systems and the significant benefits they offer to improve the efficiency of the elements in our society, the cost reduction and the enhancements of the quality of life. The fusion between old monolithic and recent networked systems has allowed the development of new scenarios of remote monitoring and control. Technological evolution with new security challenges has also involved the industrial facilities and Critical Infrastructures. After the first SCADA monolithic systems, conceived in the 1960s, the networked generation was born, where designs for open connections using TCP/IP (Transmission Control Protocol/Internet Protocol) have been deployed in order to monitor and control in real-time the processes from remote stations [27]. Despite the advantages of such integration, new vulnerabilities have arisen in these systems, as they were originally characterized by standalone or isolated configurations and the security systems were designed according to specific needs, generally well known [61], [43]. Indeed, the evolution of the CIs until their current networked status, in addition to their - even unknown - linking to general business networks, has made these infrastructures exposed to threats at the cyber level that characterized only the conventional information networks [27]. Indeed, it is well-known that the whole SCADA system is as secure as the weakest point of the entire network. As a consequence, serious accidents in CIs due to cyber-attacks took place as reported by the ICS–CERT (Industrial Control System – Cyber Emergency Response Team) in the US in the last years [14], according to which the number of threats, faults and errors has increased, especially related to cyber-attacks. The most famous example was Stuxnet, the first ever PLC rootkit, targeting specific devices [62]. After Stuxnet, several cyber security events took place. For instance, DuQu, another malware suite designed to exploit the vulnerabilities of ICSs [63], and its recent evolution DuQu 2.0 [64]. Despite the impact of such "alarms", the importance of

the safeguarding of the information traveling in the SCADA system networks is still underestimated.

Several efforts have been carried out for the sake of cyber security. Specifically, two open standards have been developed to provide encryption and authentication for SCADA communications: the IEEE 6189 suite (also known as AGA-12 [65], then included in IEEE 1711 [66]), and the IEC 62351. Moreover, the standard ISA99 establishes best practices, technical reports and related information to define procedures for securing systems [67]. The most recent contribution focused in ICS has been provided by the National Institute of Standards and Technology (NIST), in their second revision to the Guide to Industrial Control Systems (ICS) Security (NIST SP 800-82 [42]. As declared in its publication in June 2015, it includes "new guidance on how to tailor traditional IT security controls to accommodate unique ICS performance, reliability and safety requirements, as well as updates to sections on threats and vulnerabilities, risk management, recommended practices, security architectures and security capabilities and tools".

Nevertheless, the proposed methodologies and solutions are not always feasible, especially in some specific time-critical contexts, and security solutions are still a wide ongoing field of research. Moreover, ICSs are mainly Cyber-Physical Systems (CPSs), since they encompass physical devices having specific hard real-time and bandwidth constraints (Figure 4.1). Therefore, classic tools for cyber security no longer guarantee a total coverage against both malicious attacks and anomalous events. On the other hand, monitoring systems deployed for the detection and identification of physical faults and failures do not take into account events that may take place or may arise from the cyber side. To the best of our knowledge, these problems are generally addressed separately. From a cyber perspective IDS are generally adopted to discover malicious attacks [68]. From a physical perspective fault detection procedures based on system models are adopted [69]. In CPSs this approach is prone to fail. Therefore, it is worthy to analyze the effects of cyber-attacks on ICSs by means of emulated environments, that may provide accurate information on the complex behavior of cyber-physical systems during anomalous or malicious events.

In addition, it is worthy to highlight that access controls in control networks, either wired or wireless, field devices, engineering devices and information systems usually lack suitable authentication mechanisms, mainly in the industrial domains. They are often inadequate, ineffective or non-existent.

For what concerns the motivations an attacker may have to perform a malicious action against any layer of a CI, these can be various and of different nature, with a variety of goals and interests, and act with adequate modus-operandi. In [70], these are classified as:

- Competition, personal challenge or curiosity: there is no specific reason for harming the system, but the attacker desires to show off its skills.
- Revenge or personal issue: generally carried out by disgruntled insiders in response to a negative work-related event.
- Political or financial gain: aim at obtaining benefits of theft information or research, for example by prevailing in a given sector or selling such information to third parties.

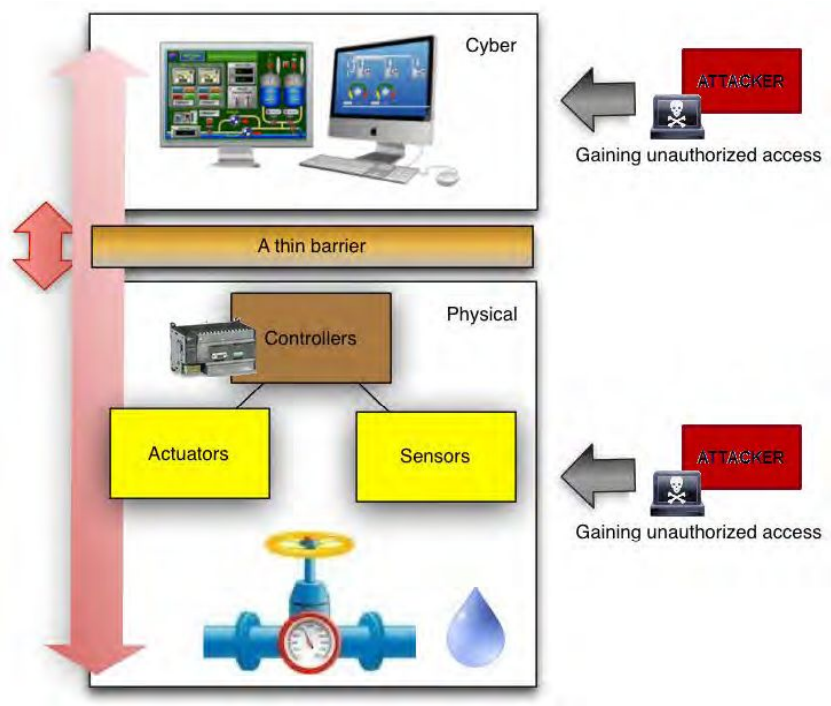

FIGURE 4.1: Vulnerability of Cyber-Physical systems

- Information theft related to critical data to be used for other purposes, attacks or dissemination.
- Harm: aims to cause distortion and destruction of parts in the infrastructure.

According to the aforementioned intentions, the attacker can be classified as:

- **Cyber terrorists:** their goal is to destabilize, disrupt and destroy cyber assets and critical data belonging to an industry, economy, government or nation.
- **White hackers:** their intention is to show off their skills and meet intellectual challenges or test a system to enhance its protection.
- **Black hackers:** their explicit goal is to harm a system, obstructing or corrupting its security or safety.
- **Insiders** or **disgruntled (ex-)employees:** authorized and authenticated personnel with specific ideas of revenge.
- **Hacktivists:** adversaries motivated by a social or political cause, which threats focus primarily on denial of service and defacement attacks.
- **Crashers, spammers, crackers:** these attackers exploit their ability to develop code and engage in malicious activities.
- **Novices:** actors with limited ability to disrupt the system operation or modify its critical information, which impact on the CI is generally low or moderate.

4.2 Overview of Cyber-Attacks against ICS

When approaching the problem of enhancing the cyber security of ICS/SCADA systems, it is important to take into account the significant differences between these systems and traditional ICT systems, considering also the constraints in terms of hard-real time, bandwidth and legacy. As exposed in [27], [32] one of the major differences is expressed in the specific priority order of the CIA (Confidentiality, Integrity, Availability) model. While ICT systems stress network data confidentiality as the most important aspect, followed by integrity and availability as last, the foremost priority in ICS is represented by availability, followed by system integrity and then confidentiality. This means that the unavailability of critical data (e.g., alarms, measurements or commands) or assets (e.g., field devices, servers, databases), or the violation of their integrity may trigger changes in the system, producing severe damage to its operation and security.

Defining an *anomaly* as "something deviated from what is standard, normal or expected", in the CIs field three main classes can be identified:

- **Infrastructural anomalies:** related to physical events taking place in the CI and its components.
- **Control anomalies:** related to any unexpected alteration in the systems control due to hardware/software errors, fault or malicious tampering.
- **Intrusion anomalies:** related with malicious actions performed against the physical or control system to cause unforeseen incidents.

As a consequence, different threat classes have been defined in [33], [68], considering such anomalies:

- **Availability:** threats related to the absence of accessibility and availability of resources and data when needed. Is further divided into **resource availability** and **information availability**.
- **Integrity:** associated to the attacker's ability to manipulate or destroy the authenticity of the information (**information integrity**) or a resource (**resource integrity**). In addition, if the adversary is able to manipulate security credentials and roles to impersonate a user's identity or an administrator of the system, a threat against **user integrity** and **host-user integrity** arise, respectively.
- **Confidentiality:** threats related to the attacker's ability to eavesdrop or expose sensitive information concerning configurations or critical data.

All the exposed threats may be caused by the (unappropriated or undesired) use of a set of specific tools or by human errors, and the consequences may lead to a wide number of security problems. The attacks addressed can be single or non-interactive, or concurrent and/or coordinated. Single attacks consist on performing a set of sequential actions against a single target for a determined interval of time and with a specific degree of intensity. On the other hand, concurrent attacks have a number of assets as targets, generally implying a strategic attack on several nodes/devices of the system.

For the attacks specialized tools and applications are generally exploited, through which the attackers can act on the system or take over integral parts of it. For example, through command-line or console interfaces it is possible to perform specific actions

on the network, adapted to the features of the system and its level of security so as to bypass them. In addition, there are software packages composed of various tools able to perform vulnerability detection, penetration testing and carry out attacks. An example of useful tool are the *sniffers*, i.e. packet analyzers with the ability to intercept and log traffic in a communications network.

For this work we consider four categories, related to the main cyber-attacks that generally target the ICS:

- **Denial of Service (DoS) Attacks:** class of attacks that aim to reduce the accessibility and disposition of information and/or resources of the system, exhaust an asset, disrupt operations or reduce its functionalities. The vulnerabilities of the TCP/IP protocol is the basis for several kinds of DoS attacks, among which we distinguish:

  Bandwidth consumption: consists in the transmission of a high traffic load which consumes all the bandwidth available on the target machine, cutting it out from the network, e.g., through packet flooding attacks. It provokes a degradation of the communication between devices, impeding to obtain data and information, and can be performed in several ways in terms of the kind of packet deployed to flood the channel. It is possible to modulate the flooding in order to introduce delays or, eventually, totally block the communication for a time interval. Also a broadcast attack can be carried out, compromising the communication in the whole network.

  Resource starvation: the aim is to saturate not the network capacity but other system resources (e.g. CPU time, system memory).

  Bug software: the attacker performs stress situations to the system in order to generate incorrect management of the system software.

  Poisoning techniques: the attacker changes destination of incoming and outgoing transmissions on the target machine to disrupt the communication or reduce the efficiency of the system.

- **Man-In-The-Middle (MITM) attacks:** consists in the hijacking of the traffic generated during the communication between two hosts, and deals with subverting the communication path between devices and the malicious node. The attacker places itself between the target machines, therefore all the traffic generated by the victims passes through it, which is eventually able to modify the data and conveniently forward the packets to the right destination without being recognized. Thereby, a compromise is to corrupt the identity of the communicating parties, as both victims have to consider legitimate the source and destination of the data stream.

- **Packet modification:** the Modbus communication can be altered in several ways while still being considered authentic, for example by changing values in the data field of the packet, the device involved, among others. As this communication takes place mainly between two devices, both of them can be considered as targets for this type of attacks, obtaining totally different results.

- **Corruption attacks:** the Modbus packets can be minimally altered, provoking the discard of such packets, which are read as corrupted. Hence, it is possible to selectively mask some physical phenomena or to alter the sampling rate of an event.

- **Replay attacks:** in this case the attacker records the communication flow over the network for a time interval and, at a later stage, substitutes the transmitted packets with the previously recorded ones.

All the previous mentioned classes of cyber-attacks do not require any previous specific knowledge about the plant, as it would be enough to monitor the data flow on the communication channel to implement them.

If some information about plant is available to the attacker, more effective and elusive attacks can be designed. From a cyber point of view, this could be considered as a sophisticated variant of the packet modification strategy but, as a deeper knowledge of the effects can be gained, more effective actions could be performed, as:

- **Stealth attacks:** in this case the modifications in the packets coming from the field are performed in such a way that the Bad Data Detector or Monitor modules (systems encharged of preventing the tampering of signals) are eluded or unable to detect the presence of manipulated data. Thus, this type of attack is able to be hidden, fooling the security systems based on state estimation.

- **Covert attacks:** the attacker is able to inject malicious control inputs in order to induce anomalous dynamics to the plant and, at the same time, to "cover" such dangerous behaviors by manipulating the packets from the field in order to mask them.

- **Fake attacks:** the attacker manipulates the packets from the field in order to generate false alarms, inducing the operators to erroneously perform recovery or emergency tasks and, consequently, degrade the plant performance or lead to dangerous situations.

At the physical or lower level, a malicious attack can be carried out by tampering or introducing false data from the field devices, i.e., in the sensors-actuators environment. Within this level, it is possible to distinguish three types of stealth attacks as in [71], namely:

- **Surge attacks**: aim to maximize the damage on the network, and maintains the intensity threshold for the whole duration of the attack.

- **Bias attacks**: modify the system in a moderate way through small perturbations occurring for long time intervals, or through big perturbations in reduced time intervals.

- **Geometric attacks:** aim to change the behavior of the system in a very discrete way at the beginning of the attack, and when the system is in a more vulnerable state, they maximize the damage.

From another point of view, two main goals can be considered against the state estimates:

1. *casual* injection of false data, where the aim is to find any attack vector that can carry an erroneous estimate of the state variables;

2. *targeted* injection of false data, which aims to find an attack vector that could add arbitrary errors in specific state variables.

As described in [72], an attack can be carried out in three critical points of the network (see Figure 4.2): the RTUs, the communication channels, and the SCADA system. Generally, the measured data is sent to the state estimator through other substations, thus an attacker that gains access (physical or remote) to a substation can modify all the data that passes through it, unless protected by physical means, data authentication or multipath routing [73]. Due to the fact that most of the detection the methods are based on the square of the difference between the observed measures and their estimates, an attacker can bypass the detection techniques if it knows the configuration of the network to compromise, achieving the desired attack.

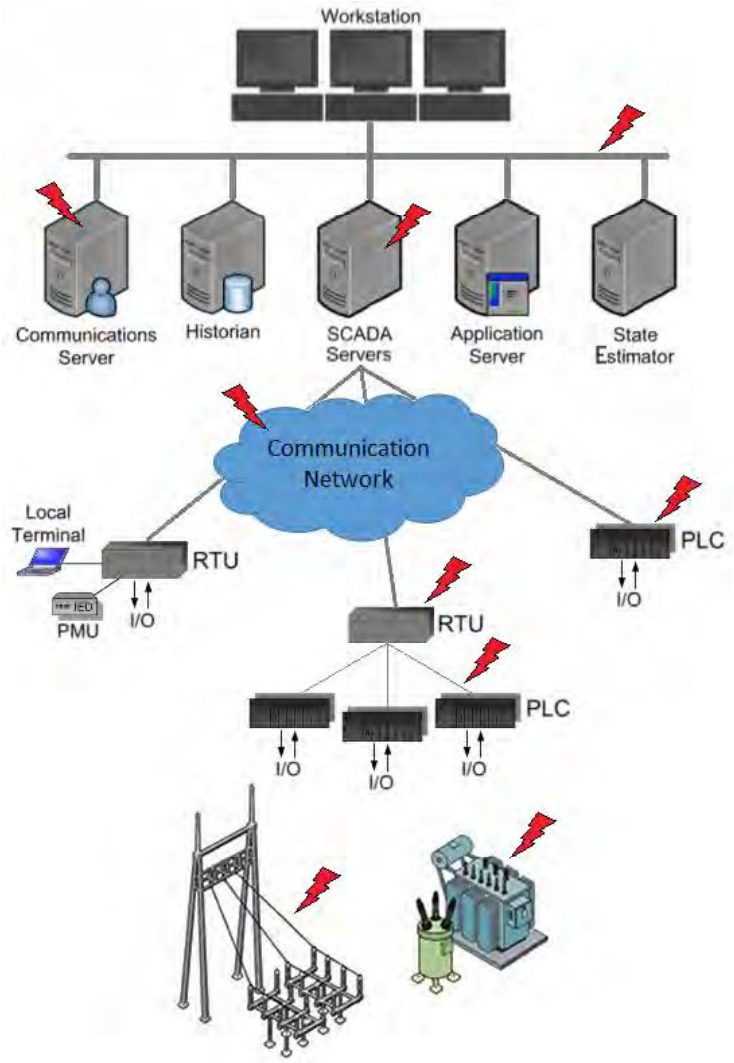

FIGURE 4.2: Schema of attacks at various levels

4.3 Study on Stealth Attacks

The *stealth attacks* have been defined in [74] as those attacks that allow a skilled but not very powerful attacker to target communication networks in a way that makes it unlikely that it gets traced and caught.

Following on such description, the stealth attacks here exposed and deeply studied in [68] correspond to those low level attacks taking place in a system, aiming to introduce false data from the field devices in order to force the control system to perform erroneous actions. On the specific case of power networks, these act specifically against the state estimation system, making the monitoring and auditing tasks of the targeted CI more difficult for the managers of the infrastructure, and in particular for those entities in charge of the security. To be successful, this kind of threat generally requires that a number of combined attacks is carried out, which varies depending on the specific target. Thereby, the term "stealth attack" defines the result of a set of actions performed to make so that the Bad Data Detection systems are properly avoided, by introducing false data that is consistent with the one related to the process. To be successful, the attacker needs to gather a wide knowledge on the operation of the target system. Three phases are to be fulfilled during a cyber-attack in order to achieve the malicious objective, and each one is based on the previous one, as depicted in Figure 4.3:

- stealthiness of the communication;
- stealthiness of the execution;
- stealthiness of the propagation.

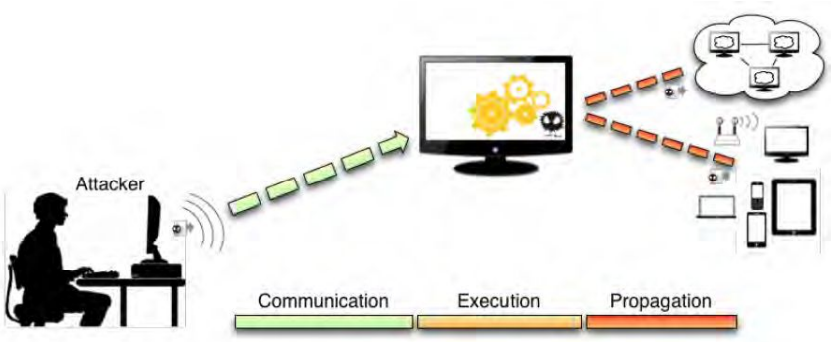

FIGURE 4.3: Stages of a stealth attack [68]

Every single attack is different in nature, and can be conformed by one or more of the three stages mentioned, always following the established order and with the main objective of avoiding being detected. However, it is important to note that the success of a stealth attack depends on the intention of the adversary, since its objective might be to achieve only one or two of the stages, and it does not care about being detected afterwards.

4.3.1 Stealth Attacks Against the State Estimation System of Power Grids

The state estimation problem consists in finding an estimate of the state of the system based on the measurements obtained from the field, exploiting the physics and the peculiarities of the process dynamics. As the measures are taken and sent to the SCADA system at a low frequency, the state estimator used is generally a linearized function that represents the system behavior.

As shown in [75] for the case of power networks, information on the system topology and power states can be obtained when the system dynamics are slow and can be

linearly approximated by the observation of power flow measurements. A hacker can exploit such information to achieve attacks where faulty, but coherent, data is introduced, with the aim of eluding the BDD systems in order not to be noticed. If these corrupted data affects the state estimator output, they can mislead the control algorithms of the network, achieving a stealthy attack. It is worthy to highlight that, as the control systems interact directly with the physical world, when an attack succeeds to bypass the security systems it may be able to provoke serious damages to the physical world.

A first analysis on false data injection attacks against the state estimator for a power grid has been carried out by Liu *et al.* in [76], where it is described how, under specific assumptions, arbitrary errors can be introduced into the state variables without being detected by the BDD systems. Two illustrative scenarios are presented, where also some constraints or limitations the attacker may encounter when projecting such attack are considered. With the same hypotheses, a characterization of the stealthy deception attack as an optimization problem is carried out in [77], where the objective is specified through a security metric and the constraints are related to the attack cost. By means of a number of realistic experiments, it is shown how such metrics can be used to enhance the security of the system by exploiting sensor redundancy. In [78], the problem of finding the optimal attack strategy is formulated, as well, which are then tested on different IEEE standard buses. Moreover, two defense mechanisms are proposed: one based, on protection, aims to the identification and protection of critical sensors, making the system more resilient to attacks; the second one, based on detection, considers schemes based on spatial and temporal distributions to identify data-injection attacks.

In [79], the authors assume that the attacker possess a limited knowledge of the network, more specifically the information regarding the model is partial or outdated. Stealthy deception attacks are proposed for both linear and nonlinear estimators, and a quantitative analysis of the trade-offs between model accuracy and possible attack impacts for different BDD schemes is carried out. The different constraints an attacker may have to perform a malicious action, related to knowledge on the system, capabilities, resources and attack targets, are analyzed in [80]. A formal model for attacks verification is proposed, formalizing grid information and different constraints by means of satisfiability modulo theories. The goal is to provide security analytics for power grid state estimation. Thus, the solution consists of an attack vector of potential stealth attacks, and a number of experiments with different IEEE networks has been carried out to evaluate the scalability of the method. Going further, Yu *et al.* deploy the Principal Component Analysis (PCA) approximation method to study and generate false data injection attack vectors [81], considering no explicit knowledge of the network topology and of the distribution of state variables, but exploiting inferences from the correlations of the line measurements, considering both DC and AC power flow models.

In [71] the problem of detecting different types of cyber-attacks is addressed, which aim to change the behavior of the targeted control system by deploying the knowledge of such system, and neglecting how vulnerabilities are exploited. In addition, they consider that the automatic response mechanisms implemented, based on estimates of the state of the system, are to keep the system in a safe state. Even though the experimental results provided are based on a chemical reactor system, it is shown how they can be extended to other type of processes. For the case of uncertain discrete time-systems, the problem of detecting false data generated by tampered sensors, which compromise the operation of the state estimator, is faced in [82]. A set of measurements consistent with the modeling assumptions is computed, which is then used to provide a measure

of the resilience of the system to false-data attacks. Moreover, they propose a robust approach for detection, which implies a reduced sensitivity to attacks. The case of the AC state estimation of a power grid system is studied in [83], as well as the effects of stealth false data injection on the SCADA system, describing how its physical properties, as the nonlinearities, can be deployed to enhance the system protection against such attacks. Moreover, the authors expose the differences between performing an attack to an AC and to a DC system. In [84] a zero-dynamics attack to the control system is studied, where no online information is available. The stealthiness properties are analyzed, and the detection problem is faced through the modification of the system's structure in terms of the respective outputs, inputs and dynamics, demonstrating the capability to reveal attacks while not affecting the system performance when no attack takes place.

On the other hand, a protection strategy against such stealth attacks is proposed in [85]. To this end, a specific set of sensor measurements is protected and a set of state variables is monitored, hence defining a set of *basic measurements* which allow to detect the attack, also in case of changes in the topology. Other strategies for detection are exposed in [86], by the use of two injection meters and a number of line power meters. Moreover, they demonstrate how phasor measurement units could be used as countermeasures against different cyber-attacks to the power network, if properly located.

Similarly, in [87] the authors analyze both the attacker and defender point of view. For the former, an algorithm with polynomial complexity is obtained using a graph theoretic approach, which provides minimum size stealth attacks against the state estimator of a smart grid when possible, otherwise the obtained attack aims at minimizing the energy while increasing the error, i.e. maximizing the attack impact. On the other hand, an algorithm for the attack detection and localization is developed, by deploying an optimized generalized likelihood ratio test. Moreover, the authors develop an heuristic able to detect different types of malicious attacks for any set of compromised meters, considering both the undetectable ones, as well as the most damaging [88].

An L_∞ norm detector is introduced, which performs better than the L_2 detector for certain parameters, as it takes advantage of the sparsity of the false data injection, not yet providing an optimal solution for each problem. Graph theory is deployed also in [83], where the authors present a technique to determine the minimum number of measurements to be tampered in order to perform stealthily the attack, and the vulnerabilities on the measurements, which consequently represent the target to be protected. The effects of false data injection attacks considering unfixed topologies of the target network are studied in [89]. More specifically, such strategy is considered to be exploited by the system operator to reduce or eliminate the possibility of attacks. Indeed, they demonstrate that if the graph has connectivity greater or equal than 2 and is non-circular, it is always feasible to eliminate the possibility of attack by selecting a set of links forming a spanning tree. This work is lately extended in [90] to consider attacks that are stealthy only in some topologies. In this framework, the authors formulate the consistent deviations of the estimated states, which flexibility linearly decrease if the attack is stealthy in more topologies. Thus, there is a trade-off between the possibility to manipulate the estimation deviation and the possibility for the attack of not being detected.

Always focusing on the operator's point of view, Dan and Sandberg introduce a security index in [91] allowing to locate vulnerable measurements and a protection cost, and an algorithm for their computation is described. Moreover, two algorithms to

place encrypted devices in the system with the aim of maximizing the system security are proposed for different IEEE benchmark networks.

Another analysis on the vulnerabilities of the power system state estimators is carried out by Vukovic *et al.* in [73], where mainly the communication infrastructure is studied. Some security metrics that quantify the importance of each substation are defined, as well as the cost of attacking measurements. Moreover, different network layer and application layer mitigation strategies are described, as modified routing and authentication, with the aim of decreasing the vulnerability of the state estimator.

As highlighted in [86], an attack may become problematic also because of the harmful actions taken by the grid operator as a response. Hence, it may be worthy to carry out an analysis on the relation between the cyber-security threats and the current operating practice under such events.

Although most of the studies previously mentioned are focused on power networks, a totally different scenario is proposed by Amin *et al.* in [92], where a linearized shallow water partial differential equation (PDE) system is presented, modeling water flow in a network of cascade canal pools. With such dynamics, a deception attacks scheme is developed, based on switching the PDE parameters and proportional boundary control actions, to withdraw water from the pools through offtakes. A well known formulation is used, based on a low frequency approximation of the PDE model and an associated proportional integral controller, in order to create a stealthy deception scheme capable of compromising the performance of the closed-loop system. It is hence demonstrated that the attack scheme presented allows to steal water stealthily from the canal until the end of the attack, by manipulating specific sensor measurements.

4.3.2 Stealth Attacks Models

Let $y \in \mathbb{R}^m$ be a measurement vector, $x \in \mathbb{R}^n$ the state vector describing the instantaneous state of a non-linear system, which nominal behavior is described by a function $h(x)$. The state estimation problem is introduced to compute the estimate $\hat{x}$ of the unknown state x, related to the measurements y. Considering that the number m of measures is greater than the number n of state variables, an over determined system of non-linear equations is solved with the unconstrained *weighted least-squares* (WLS) method [77]. This is a particular case of the well-known least-squares method, where the overall solution minimizes the sum of the squares of the errors made in the results of every single equation and all the elements in the off-diagonal of the correlation matrix of the residuals are 0.

As previously mentioned, in critical contexts the measurements are taken and sent to the SCADA systems at a low frequency, what allows to employ a stationary state estimator. The WLS estimator minimizes the weighted sum of the squared residuals $r = y - h(x)$, thus the objective function is:

$$J(x) = \sum_{i=1}^{m} \frac{(y_i - h_i(x))^2}{R_{ii}} = (y - h(x))^T R^{-1} (y - h(x))$$

$$\min_{x \in \mathbb{R}^N} J(x) = \frac{1}{2} (y - h(x))^T R^{-1} (y - h(x))$$

The necessary condition at the first order for a minimum is:

$$\frac{\partial J(x)}{\partial x} = -H(x)^T R^{-1} (y - h(x)) = 0,$$

where $H(x)$ is the $m \times n$ Jacobian matrix of the measures, representing the linearized model of the network, i.e.:

$$H(x) = \frac{\partial h(x)}{\partial x}$$
$$y = Hx + \epsilon$$

An adversary may exploit the configuration of a network to launch attacks where faulty data is introduced. If these corrupted data affects the state estimator output, they can mislead the control algorithms of the network, eluding the fault detection systems and causing serious consequences in field. Indeed, as the control systems interact directly with the physical world, when an attack succeeds to bypass the security systems and varies the integrity level of the automatization and supervision tasks, it can then be able to provoke damages to the physical world.

As previously mentioned, in order to perform a stealth attack over a specific measurement m with value y_m the attacker may need to simultaneously compromise a number of substations in order to prevent the triggering of the BDD system alarms, which make the operation non trivial. This is due to the fact that the BDD systems generally check the error signal $\|e\| = \|(CC^T)^\dagger(y - \hat{y})\|$, where $\hat{y}$ is the estimated measurement vector, obtained by means of a mathematical model of the system, C is a linearized measurement matrix and $(\cdot)^\dagger$ represents a pseudo-inverse matrix. In other words, the BDD system evaluates if the actual measurements from the field are consistent with the estimated values of the process model.

Suppose that the attacker knows the matrix H of the target system and wants to modify the measurement from y to $y_a := y + a$, where y_a represents the vector of the observed measurements that may contain the tampered data and $a = (a_1, \ldots, a_m)^T$ is the *attack vector* that represents the changes made to the vector y of the actual measurements. The i-th element a_i is not null if the i-th measurement has been compromised by the attacker [76].

Let $\hat{x}$ and $\hat{x}_{bad}$ be the estimations of the state x obtained using the actual and tampered measurements, respectively. $\hat{x}_{bad}$ can be expressed as $\hat{x}_{bad} = \hat{x} + c$, where c is a non-zero vector of size n that represents the estimation error induced by the attack. If the attacker is able to choose a as a linear combination of the columns of H, y_a can elude the BDD system as y does. Thus, such vector has to satisfy the following relation:

$$a = Hc \text{ with } c \in \mathbb{R}^n$$

in order to avoid to increase the risk of triggering an alarm. In fact, y succeeds to avoid the BDD system if $\|y - H\hat{x}\| \leq \tau$, where τ is the threshold for the triggering of an alarm. Hence, it can be demonstrated that $a = Hc$ implies $\|y_a - H\hat{x}_{bad}\| \leq \tau$. Specifically,

$$\begin{aligned} \|y_a - H\hat{x}_{bad}\| &= \|y + a - H(\hat{x} + c)\| \\ &= \|y - H\hat{x} + (a - Hc)\| \\ &= \|y - H\hat{x}\| \leq \tau \end{aligned}$$
$$\implies \|y_a - H\hat{x}_{bad}\| \leq \tau$$

Thus, if the L_2 norm of the residual of y_a is lower than the threshold τ, the tampered measurements y_a are considered an admissible state of the system and an alarm would not be triggered by the BDD system.

The goal of a stealth attack is to compromise the measurements available for the state estimator [77], in order to:

i) make the algorithm converge, assuming that no protected measurements are attacked;

ii) make the attack undetectable by the BDD system schema;

iii) make the estimated values of the target measurements $\hat{y}_a$, at the limit, near to the values y_a injected by the attacker, i.e., $\hat{y}_a := h(\hat{x}_{bad}(y_a))$ converges to y_a, thus $|y_a - \hat{y}_a| \to 0$.

Thereby, the overall problem for the attacker is to find an attack vector a which guarantee the accomplishment of the aforementioned conditions, forcing the state estimator to "estimate" a configuration that is different from the actual one, and making the fake measurements unrecognizable by the BDD system.

One strategy to perform such attack is to initially inject tampered measurements that are very close to the actual values, and gradually make the state estimator derive to an erroneous state configuration.

Nevertheless, obtaining an attack vector such that $a = Hc$ is not the only way to perform a successful stealth attack. In fact, if $a \neq Hc$, it is also possible to introduce tampered data in the network without making them detectable if the attack vector a satisfies the following inequality:

$$\|y - H\hat{x} + (a - Hc)\| \leq \tau.$$

However, in this case, besides knowing the linearized model H of network, the attacker needs to possess the values of all the measurements y obtained from the sensors and the estimates of the state variables $\hat{x}$, For this reason, this type of stealth attack is much more difficult to perform. Moreover, it is possible to protect the system from these attacks by increasing the level of confidentiality of the measurements from the sensors, hence preventing the attacker from knowing the vector y. Thus, integrating these confidentiality requirements in the considerations for the protection of the measurements makes possible to detect generic stealth attacks.

Another inconvenience that the adversary may have during an attack, as highlighted in [77], is the variation of the operative conditions. As the state of the studied system is continuously evolving, the attacker has to evaluate the linearized model H for different operational conditions. More specifically, the attacker may obtain a linear model $\tilde{H}$ for a certain state $\tilde{x}$, but the attack may certainly occur only when the system is in a different state x^*, to which corresponds a different linear approximation H^*. Thus, for small attacks or in case the states are actually similar, i.e. $x^* \approx \tilde{x}$, the residual is small and the attack will be unnoticed. The same does not hold for bigger attacks.

On the other hand, under some assumptions, the attack vector a could be kept unaltered regardless the state of the system. As a consequence, if the disturbances on the state variables do not condition the measurements that have been compromised by the attack before the change of the state, the same attack vector a remains valid. This is guaranteed if the measurements conditioned by the change of the state are far from the region of the network where the attack takes place, implying that the attacks can be locally performed in the network and stay unaltered for both the linearized models.

4.3.3 Stealth Attacks Indexes

As proposed in [73], the importance of a substation s can be quantified by its *attack impact index* I_s, which indicates the number of measurements m' that have to be compromised in order to achieve a stealth attack by gaining access to the only substation s. By definition, $I_s = 0$ if the substation s is protected, i.e. if $s \in \mathcal{P}$, where $\mathcal{P}$ is the set of protected substations. A measurement can be tampered if and only if the non-encrypted parts of the lines from the substation $s(m)$ where the attack takes place to the control center pass through the substation s. Thus, the challenge for the attacker is to find how many, and at which cost, other measurements m' have to be corrupted together with m in order to avoid the triggering of the BDD alarm.

Let $M_s \subset \{1, \dots, M\}$ be the set of indexes of all the measurements that can be attacked. Hence, the measurements $m \in M_s$ may undergo a stealth attack if and only if there exists a solution for the following system of equations with respect to the unknown variables $a \in \mathbb{R}^M$ and $c \in \mathbb{R}^{n+1}$, as stated in [73]:

$$\begin{cases} a &= Hc \\ a(m') &= 0, \ \forall m' \notin M_s \\ a(m) &= 1 \end{cases}$$

Hence, the attack impact I_s represents the cardinality of the set of measurements for which the system has a solution, and depends on the set of lines r, the encrypted substations $\mathcal{E}$ and the protected substations $\mathcal{P}$. Thus, the challenge is to find how many and at which cost other measurements m' have to be corrupted together with m in order to avoid the triggering of the BDD alarm.

A *cost index* of the attack Γ can be defined in order to quantify the minimum effort needed to achieve the goals of the attack, avoiding the alarms of the BDD system in the control center of the particular system under attack [73]. This index depends on the topology of the network and the measurements that are available, and can help the security staff to identify data corruption schemes. Similarly, the *protection cost* β for the operator corresponds to the number of measurement devices that are protected.

If the substation where the considered measurement m transits is protected and encrypted, i.e. $s(m) \in \mathcal{P} \cap \mathcal{E}$, the measurement is not vulnerable and the cost of the attack is defined as $\gamma_m = \infty$. Such condition is called *perfect protection* and is obtained if the number of protected and encrypted substations equals the number of state variables of the system, i.e. if $s(m) \in \mathcal{P} \cap \mathcal{E}$, $|s(m)| = n$. Otherwise, for a measurement m, the cost of the attack γ_m represents the minimum number of measurements that have to be corrupted in order to achieve a stealth attack. Specifically, γ_m is defined as the cardinality of the smallest set of substations $w \subseteq S$ such that a stealth attack performed on m involves some measurements m' in substations $s(m')$ and the non-encrypted part of every line of the substations $s(m')$ involved in the stealth attack transits through at least one substation in w:

$$\gamma_m = \min_{\substack{w \subseteq S \\ w \cap \mathcal{P} = \emptyset}} |w| : \exists a, c \,|\, a = Hc, \ a(m) = 1 \text{ and } a(m') \neq 0$$

$$\implies w \cap \sigma_{\mathcal{E}}(r^i_{s(m')}) \neq \emptyset, \ \forall r^i_{s(m')} \in \mathcal{R}_{s(m')}$$

To obtain γ_m a mixed integer linear programming problem is to be solved. The attack vector a has to be a stealth attack addressed to the measurement m, and in order to

have a unique solution it is necessary that the amplitude of the attack on m is unitary, i.e., $a(m) = 1$.

In order to describe the connection between the choice of which substation to affect and the set of measurements that may be corrupted after the attack of the substation, two decisional binary vectors are necessary. Let the first be $\mu \in \{0,1\}^{n+1}$ where $\mu(s) = 1$ if and only if the substation s is attacked. Hence, for the protected substations it will be $\mu(s) = 0\, \forall s \in \mathcal{P}$. Let the second binary vector be $\nu \in \{0,1\}^{M}$ where $\nu(m) = 1$ indicates that the measurement m may be corrupted due to an attack to a main substation, and $\nu(m) = 0$ if m cannot be corrupted. The measurement m can be attacked if and only if the non-encrypted part of every line between $s(m)$ and the control center goes through at least one of the affected substations. If the mixed integer linear programming algorithm is not executable, the cost of the attack is defined as $\gamma_m = \infty$, otherwise γ_m is the optimal solution to the problem.

4.4 Stealth Attacks Detection and Network Protection

Generally, the detection of this type of attacks can be expressed as a problem of anomaly detection due to intrusions [71]. Thus, knowing how the output of the physical system should react to the control input sequence, any attack to the measured data may potentially be detected through the comparison between the expected output with the actual - potentially compromised - signal received. Therefore, to formalize the anomalies detection problem it is necessary to have: (i) the physical system behavioral model and (ii) the anomalies detection algorithm.

While research efforts focus on prevention and detection, in practice the response mechanisms are few. A straightforward response strategy is the use of the linear estimation when an anomaly is detected: if the threshold value τ imposed is exceeded, the anomaly detection module discards the measurements of the sensor, which are eventually replaced by measurements generated by an internal model, as depicted in Figure 4.4. Otherwise, the information obtained from the field is considered correct. Every time the system introduces an automatic response due to an alarm, it is necessary to take into account the presence of an eventual false alarm, and its related costs.

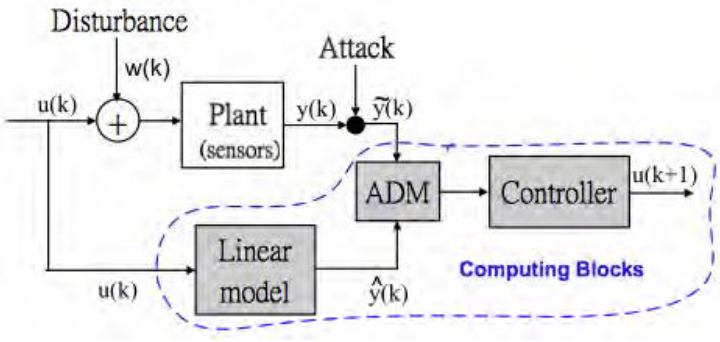

FIGURE 4.4: The role of the anomaly detection module in detection [71]

Two main strategies exist to protect the control systems and the state estimator, namely: deploying robust control algorithms able to detect or tolerate the modifications of the data, and protecting sensor measurements from tampering. Both approaches are complementary to increase security of the data transferred; the former requires high

level techniques which may incur in high developing costs, efficiency reduction, and personnel training. However, the second approach, based on the lowest level layers, is a simpler alternative with better performance, and may be carried out while developing the protection algorithms.

More specifically, as proposed in [85], it is possible to reduce the risk of stealth attacks that affect the state estimator by protecting an accurately chosen set of measurements, defined as the *basic measurement set*, which consists of the minimum number of measurements that ensure the observability of the network, i.e., such that the state variables can be estimated from the measurements. Generally, the cardinality of such set is equal to the number of state variables that have to be estimated, while the number of measurements is generally higher. The remaining measurements constitute the *redundant measurements* and are useful for the traditional BDD systems and for other detection methods [93]. Therefore, by independently verifying the basic measurement set estimations, it is possible to limit the capacity of an attack to manipulate the measurements remaining unnoticed.

Thereby, for a certain matrix H the purpose is to identify the smallest set of sensors to protect and a set of state variables that can be independently verified, so as to to make it impossible to create an attack vector that is not detected by the security systems. A further reduction of such sets could be obtained if the operators are able to perform independent verifications over some particular state variables. This may imply a certain degree of indirect protection for the measurements that are more influential for the state variables.

The probability of success of an attack has been experimentally obtained by making different attempts, considering a fixed number k of random measurements to tamper. If such probability is lower than 1 for a certain k, there exist sets of $m - k$ measurements that, if protected, impede the attacker to insert false data and be undetected. If k measurements can be compromised, with $k \geq m - n + 1$, there exists at least one attack vector that will not be detected, also if the attacker cannot control which of the k sensors have to be tampered. This constitutes a lower limit on the number of sensors that have to be protected in order to prevent an attack, i.e., protect at least n sensors represents a necessary but not sufficient condition. In fact, it does not always guarantee the detection of a stealth attack, and it is sometimes necessary to protect more than n sensors.

When an attack is limited over a certain set of measurements I_m, it can be assumed that the remaining measurements are protected by the operator of the network, who can independently verify the values of some chosen state variables. These constitute the set $I_{\bar{\nu}}$, which tampering has to be obviously avoided by the attacker in order to be undetected. Being $I_{\bar{m}}$ the set of indexes of the measurements protected by the human operator of the network, these unassailable too, the relative elements $a_i,\ i \in I_{\bar{m}}$ of the attack vector are zero. Hence, in order to perform an attack that is unnoticed, it is necessary to find an attack vector that satisfies the three following conditions:

$$\begin{cases} a &= Hc \\ a_i &= 0 \text{ for } i \in I_{\bar{m}} \\ c_j &= 0 \text{ for } j \in I_{\bar{\nu}} \end{cases}$$

From the operator's point of view, in order to ensure that the stealth attacks are always detected, it is necessary to identify a set $I_{\bar{m}}$ of sensors and a set $I_{\bar{\nu}}$ of state variables

which make it impossible to build an attack vector satisfying the three aforementioned conditions.

A/N. A survey about stealth attacks and further contributions exposed in this Chapter have been published in:

L. Cazorla, **E. Etchevés Miciolino**, C. Alcaraz, R. Setola, J. Lopez, F. De Cillis. *Injection-based Stealth Attacks in Critical Infrastructures*. Journal of Computer and Systems Sciences (JCSS), 2015. (*Submitted*)

Chapter 5

Complex Networks

5.1 Distributed Systems and Architectures

Many real systems can be modeled as networks, many of which are large in components number, geographically distributed, and complex. In these representation the elements of the system are *nodes* and interactions between elements are *edges*. An even larger set of systems can be modeled using dynamical processes on networks, which are in turn affected by the dynamics. The structure of distributed systems can be analyzed as being constituted by multiple subsystems that interact with neighboring subsystems, as sketched in Figure 5.1. In such case the number of computing nodes equals the number of present physical subsystems, which are allowed to interact with each other in a pattern that exactly mimics the pattern of physical interactions between subsystems. Thereby, nodes can exchange useful information and measurements for implementing in their models the effect of the physical interconnections.

FIGURE 5.1: Types of architectures

This will certainly need a higher communication capacity, but the needed computation power decreases and the reliability and applicability of the architecture is increased, in comparison to centralized and decentralized architectures. Hence, the idea is that an excessively difficult problem is decomposed into smaller subproblems simpler enough to be solved with the existing computation and communication infrastructures.

5.1.1 The Decomposition Problem

To such end, an implementation involving multiple *agents*, i.e. computation nodes, is needed, particularly a hardware/software combination capable of:

- directly measure physical variables;
- process locally available information;
- communicate with other agents.

Moreover, each resulting subproblem should fulfill the following constraints:

- **Computation constraint:** the computation power needed to execute the task of each subproblem should not exceed the computation power affordable by any single agent to which the subproblem may be assigned.
- **Communication constraint:** the communication capacity needed to convey the information needed for the task execution of each subproblem should not exceed the communication capacity affordable by any single agent to which the subproblem may be assigned.

The outcome of the decomposition of a large-scale system will be a description of its dynamic behavior in terms of the dynamics of a multiset of N subsystems $\mathcal{S}_I, I \in \{1,\dots,N\}$ [60]. Initially the structure of the system is decomposed, in order to select the components of the state and input vectors that will be assigned to each subsystem. Then, the dynamic model of the system is decomposed, so as to derive the dynamic equation of each subsystem.

The *structural graph* of a dynamical system, having a state vector $x \in \mathbb{R}^n$ and input vector $u \in \mathbb{R}^m$, is the directed graph $G := \{V, E\}$ having the vertices set $V := \{x^{(i)} : i \in \{1,\dots,n\}\} \cup \{u^{(i)} : i \in \{1,\dots,m\}\}$ and the system structure Σ as the edges set $E = \Sigma$.

A multiset D is created, $\mathcal{D} := \{\mathcal{S}_1,\dots,\mathcal{S}_N\}$ of $N \geq 1$ subsystems, defined through a multiset $\{I_1,\dots,I_N\}$ of index sets, each one having a *local* state vector $x_I \in \mathbb{R}^{n_I}$ and a *local* input vector $u_I \in \mathbb{R}^{m_I}$. The latter contains all the input components that affect at least one component of the local state vector. As a consequence, the structural graph of the I-th subsystem can be defined as the subgraph G_I induced on G by the subset made of all the components of x_I and u_I. For each $I \in \{1,\dots,N\}$, the following holds:

1. $I_I \neq \emptyset$;
2. $I_I^{(j)} \leq n$;
3. the subdigraph G_I must be weakly connected;
4. $\cup_{I=0}^{N} I_I = \{1,\dots,n\}$, so that the decomposition covers the whole original monolithic system, allowing for a state component to be assigned to one or more subsystems (*overlapping decomposition*), i.e. it is not required that $I_I \cap I_J = \emptyset$.

A shared state variable $x^{(s)}$ is a component of the state x such that $s \in I_I \cap I_J, I \neq J$. The overlap index set of subsystems sharing a state variable is the set $\mathcal{O}_s := \{I : s \in I_I\}$, whose dimension is $N_s := |\mathcal{O}_s|$.

The external variables influencing the dynamics of local state components of subsystem $\mathcal{S}_I$ will make up the vector of interconnection variables $z_I \in \mathbb{R}^{p_I}, p_I \leq n - n_I$, defined as:

$$z_I := \text{ col } (x^{(k)} : (x^{(k)}, x^{(j)}) \in V, j \in I_I).$$

The set of subsystems acting on a given subsystem $\mathcal{S}_I$ is the set

$$I_I := \{K : \exists (x^{(k)}, x^{(j)}) \in V, k \in I_K, j \in I_I).$$

The main idea in model decomposition is to rewrite the model equations in order to separate the effect of the local variables from that of the interconnection variables.

5.2 Controllability in Complex Networks

Controllability theory offers a general, rigorous, and well-understood framework for the design and analysis of networks in which a control relation between nodes is required [94]. In [95] a graph-theoretical formulation is provided, which is then largely exploited in [96] to define the so-called *driver nodes* using non-rigorous maximum matching (to find subset of driver nodes that do not share input vertices). The *Power Dominating Set* (PDS) approach, derived from the Power Grids control domain, provides an equivalent formulation for identifying minimum driver node subsets sufficient to reach a desired configuration from an arbitrary configuration in a finite number of steps, which can be employed for large-scale networks. In [96] and [97] the effects of attacks (edge and vertex removal) on the network and the subgraph representing the controlling structures have been studied, identifying the relation between impact and network topology, while the robustness in large networks has been analyzed in [98].

The PDS proble has been firstly introduced in [99] as an extension of the domination concept, focusing on the structure of electric power networks and the need for its efficient monitoring.

Here, the effects of different non-interactive attack patterns (i.e. attackers are assumed to choose only a single set of vertices) resulting in vertex and edge removal from control graphs are studied, as well as the influence of equivalent PDS choice.

For Critical Infrastructure networks, several topologies are of interest in which the controllability concept results essential for protection. Therefore, elementary random (Erdös-Rényi), small-world (Watts-Strogatz) and scale-free (Barabási-Albert) graphs are studied, providing simulation results for different parameter settings.

5.2.1 Structural Controllability and Power Domination

Consider the general dynamic linear time-invariant (LTI) system:

$$\dot{x}(t) = Ax(t) + Bu(t),\ x(t_0) = x_0,$$

where $x \in \mathbb{R}^n$ is the time-dependent state vector of the system composed of n nodes, $u \in \mathbb{R}^m$ is the input vector forcing the system to a desired state, A is the $n \times n$ matrix defining the network topology, and B is the $n \times m$ matrix, with $m \leq n$, identifying the set of nodes controlled by the input vector u.

For the well-known Kalman's rank criterion, the LTI system is said to be *controllable* if and only if

$$\text{rank}[B,\ AB,\ A^2B, \ldots, A^{n-1}B] = n.$$

To provide a graph-theoretical formulation, let matrix B be the set of nodes capable of driving control. As suggested in [95], one can study the system as a digraph $G(A, B) = (V, E)$, where $V = V_A \cup V_B$ is the set of vertices (or nodes) and $E = E_A \cup E_B$ is the set of edges (or links). Thereby, V_B are the nodes able to inject control signals into the entire network. In the case of SCADA systems, verteces may represent control terminal units, servers, PLCs, etc, while the links constitute the communication lines.

As direct computation of the PDS is undesirable, problem that has been shown to be NP-hard for generic graphs in [99], two rules, exposed in [100], are employed to determine such set, indicated as N_D:

OR1 A vertex in the N_D observes itself and all its neighbours.

OR2 If an observed vertex v of degree $d \geq 2$ is adjacent to $d-1$ observed vertices, the remaining unobserved vertex becomes observed as well.

By omitting OR2, the Dominating Set (DS) is obtained. For a generic undirected graph $G = (V, E)$, the goal to build the PDS is to find the set $N_D \subseteq V$ with $|N_D| < k$, $k \geq 0$ desired integer, and N_D observes all vertices in V satisfying OR1 and OR2. Aside the computational issues arising for the PDS determination, it has arosen to be a non-local problem, as it correctness cannot be checked by considering only a selected neighborhood in the graph. A further analysis on the computational complexity for the DS and PDS is carried out in [32]

In Algorithm 1 and 2 the pseudocodes for determining the PDS based on OR1 and OR2 are provided.

Algorithm 1 OR1 $(G(V, E))$

output $(DS = \{v_i, \ldots, v_k\}$ *where* $0 \leq i \leq |V|)$

Choose vertex $v \in V$
$DS \leftarrow \{v\}$ *and* $N(DS) \leftarrow \{v_i, \ldots, v_k\}\ \forall i \leq j \leq k - (v, v_j) \in E$
while $V - (DS \cup N(DS)) \neq \emptyset$ **do**
    Choose vertex $w \in V - (DS \cup N(DS))$
    $DS \leftarrow DS \cup \{w\}$
    $N(DS) \leftarrow N(DS) \cup \{v_i, \ldots, v_k\}\ \forall i \leq j \leq k - (w, v_j) \in E$
end while
return (DS)

Algorithm 2 OR2 (DS)

output $(N_D = \{v_i, \ldots, v_k\}$ *where* $|N_D| \geq |DS|)$

$N_D \leftarrow DS$
$i \leftarrow 1$
while $i \leq |N_D|$ **do**
    Choose vertex $w \in N_D$ *with* $d \geq 2$
    if ($d - 1$ *vertices* $\in N(w)$ *and* $\subseteq N_D$) **and** ($\exists$ *vertex* $w_1 \in U,\ w_1 \in N(w)$) **then**
        $N_D \leftarrow N_D \cup \{w_1\}$
        $U \leftarrow U - \{w_1\}$
        $i \leftarrow 1$
    else
        $i \leftarrow i + 1$
    end if
end while
return (PDS)

5.2.2 Network Models

As previously mentioned, the Erdös-Rényi (ER) random graphs class [101], [102], [103] is firstly employed, identifying them as $ER(n, p)$, with n number of vertices and edges independently determined with probability p. Then, the Watts-Strogatz (WS) graph

model [104], [105], [106] is considered, which starting point for construction is a simple ring lattice of n vertices connected to k neighbours with independent probability p. These networks, also referred to as "small-world", are connected and have well-defined vertex distance, but are characterized by significant clustering. In addition, graphs showing a power-law degree distribution as the Barabási-Albert (BA) model [107] have been employed. These are obtained from an initial random graph composed of at least 2 vertices with degree $d \geq 1$, to which vertices are added, linked with probability proportional to the existing vertices' degree. The resulting degree distribution follows a power law, and a small number of nodes present high degree. To conclude, the power-law out-degree graph model (PLOD) [108] with lower clustering coefficient is studied.

As empirically shown in [109], a wide number of actual networks are characterized by an exponent between 2 and 3, a small diameter an a low vertex clustering coefficient, characteristics that are taken into account for the graph generation. Additional requirements are acyclical connected graphs, without self-loops, which follow the structural controllability presented in [95]. For the graph generation, the methods and implementations presented in [110] have been used.

5.2.3 Vertex Choice

By implementing rules OR1 and OR2, not a single and unique PDS is obtained for a given graph. Therefore, three generation strategies have been chosen:

1. Beginning with the vertex with *maximum out-degree*;
2. Beginning with the vertex with *minimum out-degree*;
3. Randomly choosing an initial vertex.

For simplicity, *strategies* based on Algorithm 1 satisfying OR1 are described. For a given strategy, assume that an instance $N_D^{strategy}$ is represented by a partial order given by the out-degree ($\leq$ or $\geq$) in case of N_D^{max} or N_D^{min}, respectively. In case of N_D^{rand}, no such relation exists. However, for the sake of simplicity, vertices are assumed to be enumerated.

N_D^{max} Obtains N_D based on vertices with the maximum out-degree, defining a vertex choice sequence generating the DS for OR1. All vertices with maximum degree d_{max} are considered before those with lower degree, as exposed in Algorithm 3.

N_D^{min} Generates N_D using vertices with the minimum out-degree until these are exhausted, before identifying nodes with higher degree (Algorithm 3).

N_D^{rand} Obtains N_D satisfying OR1, in which the DS is randomly generated choosing an arbitrary vertex $v \in V$ in each iteration.

Algorithm 3 Maximum/Minimum strategies for OR1 ($G(V, E)$)

output ($DS = \{v_i, \ldots, v_k\}$ *where* $0 \leq i \leq |V|$ *with max/min out-degree*)

d ← *Find max/min out-degree in V*
$DS \leftarrow \{\}$
$N(DS) \leftarrow \{\}$
while $V - (DS \cup N(DS)) \neq \emptyset$ **do**
    $W \leftarrow$ *Obtain the set of* $v \in V$ *with degree* d
    for each *random* $w \in W$ **do**
        if $w \notin (DS \cup N(DS))$ **then**
            $DS \leftarrow DS \cup \{w\}$
            $N(DS) \leftarrow N(DS) \cup \{z_i, \ldots, z_k\}\ \forall i \leq j \leq k - (w, z_j) \in E$
        end if
    end for
    $d \leftarrow$ *Update* d *with next smaller/larger out-degree in* V
end while
return (DS)

5.2.4 Attack Models

The previous strategies for the PDS determination are analyzed according to five attack models $AM - i$. It is assumed that the attacker has full knowledge of both the network and the PDS, and aims to remove vertices in N_D, i.e. in the PDS itself, without being able to remove an arbitrary number. This type of attacker could correspond to disgruntled insiders, ex plant operators, or outsiders who observe and learn from the topology to later damage a part or the entire system. On an infrastructure network, this may lead to DoS attacks to communication lines, leaving parts of the system uncontrolled, unprotected or isolated. The studied attack models are:

$AM - 1$ The first driver node in a given ordered set $N_D^{strategy}$ is attacked.

$AM - 2$ The attacker aims to delete vertices in the PDS located in the middle of the ordered set obtained by a given $N_D^{strategy}$.

$AM - 3$ The last element in the ordered set given by $N_D^{strategy}$ is removed.

$AM - 4$ Removes vertices $v \in V$ with the highest *betweenness centrality* of the graph.

$AM - 5$ Randomly deletes a vertex $v \in V$, $v \notin N_D$.

In each case the edges of the target vertex are removed until its complete isolation from the network, what may also result in the isolation of other dependent vertices or in graph partition. The following Algorithm 5 describes the different attack models:

5.2.5 Structural Controllability under Vertex Removal

To evaluate the three forementioned types of structural controllability strategies, several attack patterns were studied for the selected graph topologies. As large-scale networks are of particular interest, networks with a number of vertices ranging from 50 to 2000 where studied. In order to generate sparse graph, which best represent Critical Infrastructures, a suitable parameters choice have been performed for each topology,

Algorithm 4 Attack models ($G(V,E),\ AM,\ N_D^{strategy}$)

output (*Isolation of a vertex for a given* $G(V,E)$)
local *target* $\leftarrow 0$

if $AM == AM-1$ **then**
  target $\leftarrow N_D^{strategy}[1]$
else if $AM == AM-2$ **then**
  target $\leftarrow N_D^{strategy}[|N_D^{strategy}|/2]$
else if $AM == AM-3$ **then**
  target $\leftarrow N_D^{strategy}[|N_D^{strategy}|]$
else if $AM == AM-4$ **then**
  target $\leftarrow$ BETWEENESS CENTRALITY($G(V,E)$)
else
  target $\leftarrow$ OUTSIDE $N_D^{strategy}(G(V,E), N_D^{strategy})$
end if
ISOLATE VERTEX($G(V,E)$,*target*)
return $G(V,E)$

e.g., $p = 0.3$ for ER and WS graphs, $d = 2$, $\alpha \approx 3$ for BA graphs. Under these conditions, robustness is evaluated from two perspectives:

- degree of connectivity;
- degree of observability.

In the first case, the diameter (Dm), density and average clustering coefficient (CC) are considered. These should be kept low in proportion to the graph growth and the average degree of links (AD), especially after an attack. For observability, the remaining observable network is considered as a percentage of the initial graph.

The diameter for ER graphs remains broadly stable for larger numbers of nodes, while the density and CC is significantly reduced for small and medium networks (composed of 50 to 500 nodes) after the attack, as esposed in Figures 5.2 and 5.3. This reduction becomes more evident when the perturbation is targeted, i.e. for $AM-1$ on N_D^{max} and $AM-2$ on N_D^{min}, or when the highest betweenness centrality of the network is compromised ($AM-4$ on every strategy for the initial vertex choice).

Similar results have been observed for WS small networks, where the graph parameters evolution may appear somewhat confusing as it does not fully capture the normal small-world behavior, in which the network diameter is expected to be significant [106]. This is due to the small connectivity probability chosen for the graph generation, as the average degree of links reaches small values (≈ 2) regardless the network dimension. Despite the appreciable diminishment of Dm and CC with respect to the initial network shown in Figure 5.3, the global network density is not affected. Therefore, the WS topology can be considered resilient to PDS vertex isolation, particularly when targeting the maximum out-degree vertices or those with the highest betweenness centrality.

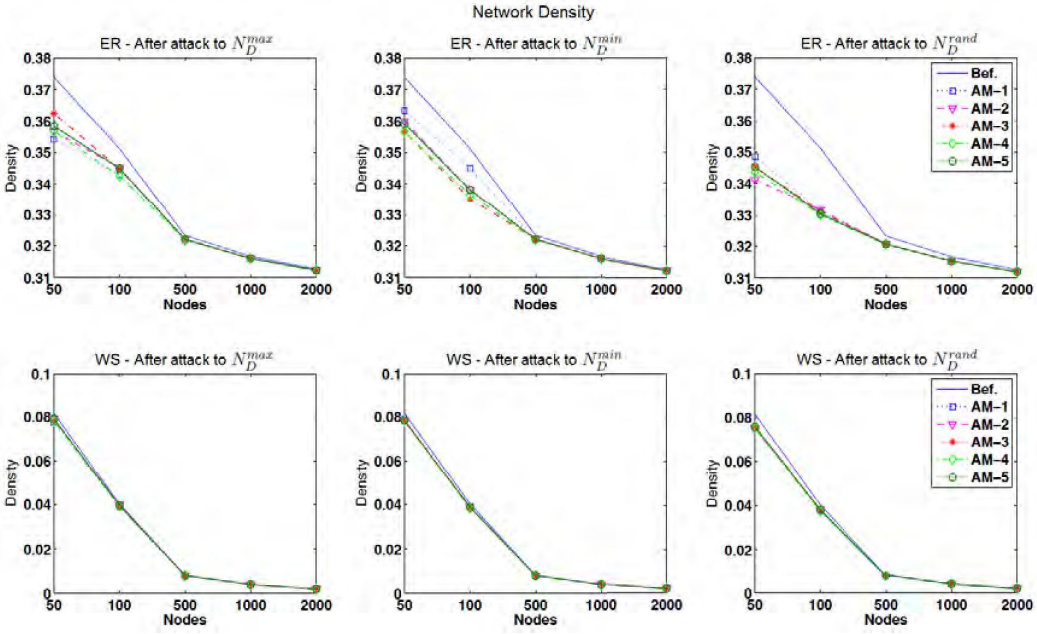

FIGURE 5.2: Global density after attack in ER and WS networks

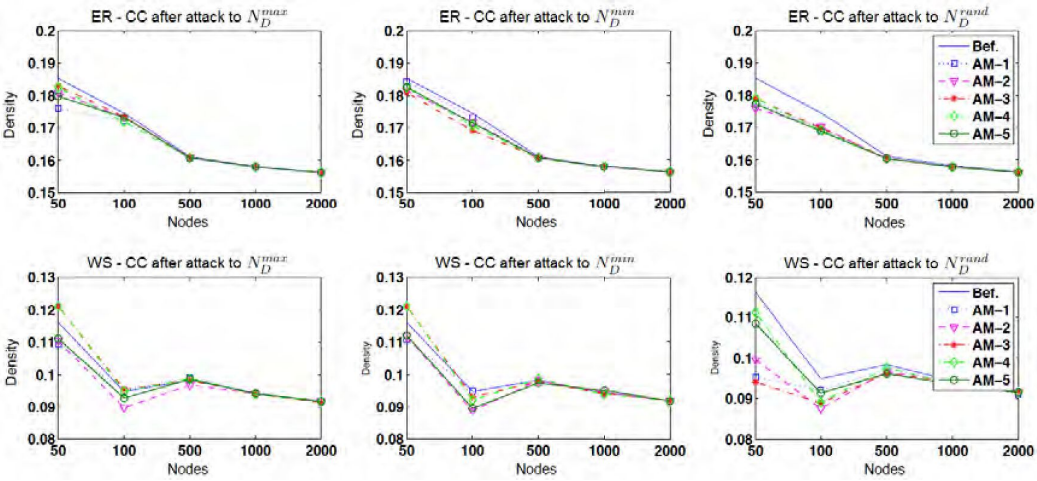

FIGURE 5.3: Cluster coefficient after attack in ER and WS networks

As shown in Figure 5.4 and Figure 5.5, for power-law distributions the parametes remain almost invariant for both small and large networks, confirming their overall resilience. Nevertheless, some sensitivity in observability has been obverved when attacking N_D^{max} in small networks (50 nodes).

Similarly, low-exponent power-low networks appear to be robust, except for $AM-4$ attacks where the network diameter highly varies. Conversely, $AM-5$ attacks do not present major consequences with respect to intentional threats, and may show the larger impact on observability. Actually, no significant changes in global density due to the attacks are to be highlighted for different exponents of the power-law distribution ($\alpha = 0.1,\ 0.3$ and 0.5), even when the diameter varies after an $AM-4$ threat, as

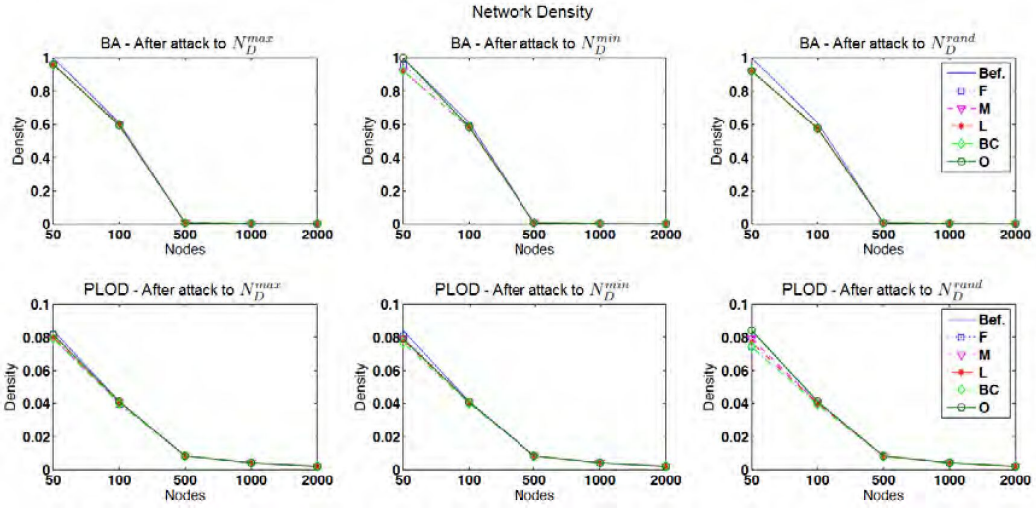

FIGURE 5.4: Global density after attack in BA and low-exponent power-law networks

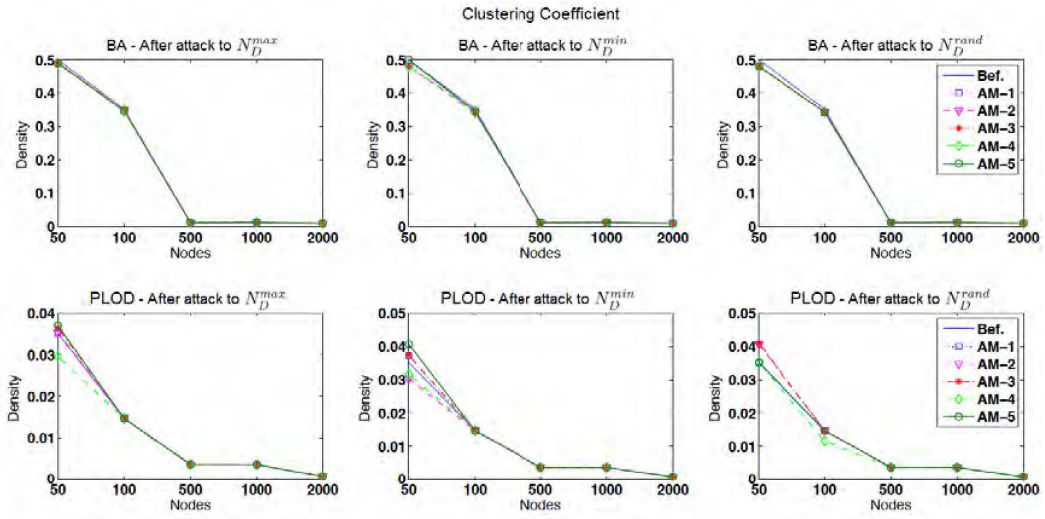

FIGURE 5.5: Local density after attack in BA and low-exponent power-law networks

depicted in Figures 5.4 and 5.6.

For each topology and varying parameters, the fraction of observed nodes is shown in Table 5.4 and Table 5.5, which is kept high despite the attack. Thereby, observability does not only depend on the network topology and construction strategies of the driver nodes set $N_D^{strategy}$, it depends also on the nature of the attack or perturbation, as described in [98], among which the degree-based attacks are the most dangerous.

5.2.6 Multi-Round Threat Model

The single-attack framework proposed has the been extended in [33] to multiple-round attacks, studying again the robustness of controllability when multiple and combined

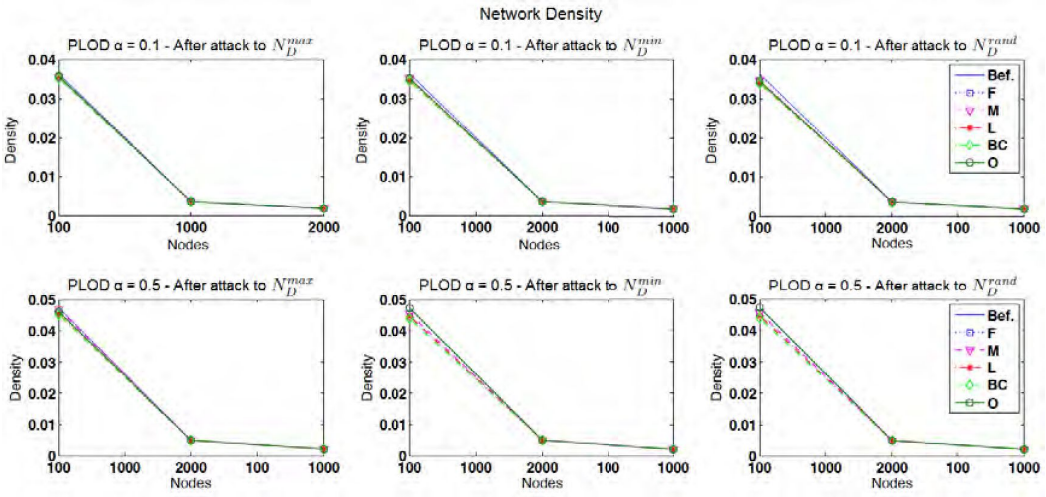

FIGURE 5.6: Global density after attack in low-exponent power-law networks

attacks affect control in different network topologies. In addition, the interaction between different multi-round attack strategies and the underlying control graph topology on robustness, considering three new multi-round attack scenarios, namely:

- The removal of random edges from a single or from several vertices;
- The isolation of some vertices;
- The removal of random edges and vertices from a dense power-law subgraph.

Thereby, three attack scenarios SCN have been defined:

$SCN-1$ A small number of random edges is removed from one or more vertices, chosen depending on the selected AM, but avoiding spurious node isolation;

$SCN-2$ The aim is to totally isolate one or more vertices from the network, by deleting all the links connected to it;

$SCN-3$ One or more vertices on a sub-graph are attacked by randomly deleting part of their links ($SCN-1$), or isolating it ($SCN-2$), so as to assess the effects on the whole graph.

For the latter, the *Girvan-Newman* algorithm [16] has been deployed to detect and obtain specific communities within a complex graph, useful for the sub-graph extraction task. These communities consist of a subset of nodes with dense links within the community itself and few connections to other less dense communities. Thereby, links between communities are sought by progressively calculating the betweenness of all existing edges and removing those with the highest betweenness.

Algorithm 5 is then extended to consider the desired scenario as follows:
For the multi-round attacks, combinations of the previously mentioned AMs, which are only representative of wider classes, result in a set of rounds based on multi-target attacks. For our studies, five target classes have been considered, denoted as $TG-i,\ i=\{1,\ldots,5\}$ and described as follows:

Algorithm 5 Attack models ($G(V,E)$, AM, $N_D^{strategy}$, SCN)

output (*Attack of one vertex for a given* $G(V,E)$)
local i, *target*

if $AM == AM - 1$ **then**
  target $\leftarrow N_D^{strategy}[1]$
else if $AM == AM - 2$ **then**
  target $\leftarrow N_D^{strategy}[|N_D^{strategy}|/2]$
else if $AM == AM - 3$ **then**
  target $\leftarrow N_D^{strategy}[|N_D^{strategy}|]$
else if $AM == AM - 4$ **then**
  target $\leftarrow$ BETWEENESS CENTRALITY($G(V,E)$)
else
  target $\leftarrow$ OUTSIDE $N_D^{strategy}$($G(V,E)$, $N_D^{strategy}$)
end if
if $SCN == SCN - 1$ **then**
  REMOVE SELECTIVE EDGES($G(V,E)$,*target*)
else if $SCN == SCN - 2$ **then**
  ISOLATE VERTEX($G(V,E)$,*target*)
end if
return $G(V,E)$

$TG-1$ Non-interactive scenario in which a single vertex $v \in V$ is attacked according to an AM, being v a driver or and observed node. Such condition corresponds to the one considered in the single-attack case.

$TG-2$ Multi-round scenario based on two AM attacks, $AM-x$ and $AM-y$, with $x, y \in \{1,\ldots,5\}$ and $x \neq y$. Thereby, this represents the case in which one or several attackers compromise two strategic nodes.

$TG-3,4,5$ Multi-round scenario based on 3, 4 or 5 threats, respectively, with analogous goals and similar features as in $TG-2$.

Table 5.1 summarizes these target classes, and the technique is described in Algorithm 6, depending on the type of scenario and the number of targets to be attacked. As previously mentioned, for $SCN-3$ scenarios, a sub-graph is first extracted from $G(V,E)$ using the Girvan-Newman algorithm.

Targets	Combination of $AM-x$	Num. of Attacks
TG-1	F, M, L, BC, O	5
TG-2	F-M, F-L, F-BC, F-O, M-L, M-BC, M-O, L-BC, L-O, BC-O	10
TG-3	F-M-L, F-M-BC, F-M-O, F-L-BC, F-L-O, F-BC-O, M-L-BC, M-L-O, M-BC-O, L-BC-O	10
TG-4	F-M-L-BC, F-M-L-O, F-M-BC-O, F-L-BC-O, M-L-BC-O	5
TG-5	F-M-L-BC-O	1

TABLE 5.1: Five attacks rounds with combined AM

Algorithm 6 Multi- Round Attacks ($G(V,E)$, $N_D^{strategy}$, $TG-x$, SCN)

output (*Attack of several vertices for a given* $G(V,E)$)
local i, *Combination_AM*, AM, SCN

1: **if** $SCN == SCN-3$ **then**
2: $G_{sub}(V,E) \leftarrow$ GIRVAN-NEWMAN($G(V,E)$)
3: $N_D^{strategy} \leftarrow$ EXTRACT DRIVER NODES FROM SUBGRAPH($G_{sub}(V,E)$, $N_D^{strategy}$)
4: $SCN \leftarrow$ DETERMINE NEW SCN-1-2()
5: **end if**
6: *Combination_AM* $\leftarrow$ COMBINE ATTACKS($TG-x$)
7: **for** $i \leftarrow$ SIZE(*Combination_AM*) **do**
8: $AM \leftarrow$ *Combination_AM*$[i]$
9: **if** $SCN == SCN-3$ **then**
10: $G(V,E) \leftarrow$ ATTACK MODELS II($G(V,E)$, $G_{sub}(V,E)$, $N_D^{strategy}$, AM, , SCN)
▷ Algorithm 5 on $G_{sub}(V,E)$
11: **else**
12: $G(V,E) \leftarrow$ ATTACK MODELS($G(V,E)$, $N_D^{strategy}$, AM, , SCN)
13: **end if**
14: **end for**
15: **return** $G(V,E)$

5.2.7 Structural Controllability under Multi-Round Attack Scenarios

A previously done, the robustness of the networks after an attack is studied under two perspectives, the degree of connectivity and of observablity, and the results for the three scenarios SCN are summarized in Table 5.6, Table 5.7 and Tables 5.8, 5.9.

As depicted in Table 5.6 for $SCN-1$, it is observed that ER topologies are sensitive in connectivity terms. The diameter for small networks is variable, especially for those under the control of $N_D^{max,min}$, with a special emphasis in scenario $TG-3$ where a complete break up of the network is verified and the observation rate is largely influenced, reaching null values. Also local and global densities are variable for all network distributions and for all TGs, where the controllability $N_D^{min,rand}$ are mainly affected. Observability is mainly kept high, except for $TG-3$ and networks under the control of $N_D^{max,rand}$.

For WS graphs, the diameter changes for any distributions, particularly for small networks, and the greatest effect is obtained when launching a $TG-3$ attack. For this topology, the density of the network is slightly modified when performing a $TG-2$ attack, whereas no relevant effect has been observed for the other cases. However, this does not hold for local density, since the effects on the network become more and more evident as the number of targets increases, especially when the number of nodes that constitute the network is not high. The impact on the observability is not very accentuated for this topology, and the effect is more evident when performing an attack to the driver node with the maximum out-degree in small networks.

For BA graphs, the diameter shows a small variation for any $N_D^{strategy}$ and for both single and multiple targets. The difference is made by the $TG-3$ strategy, for which the consequences on the network are remarkable in any case. The global density of

the network is influenced mainly in small network, and the links of a N_D^{rand} are damaged. Unlike ER and WS, the CC of BA graphs does not significantly change, but its observability is heavily compromised for any TGs where the control relies on N_D^{max}.

For $SCN-1$, the diameter, density and the CC of BA networks remains almost invariant what shows its robustness degree for all types of AMs. Nonetheless, the densities can suffer some changes when three or more nodes are compromised and these nodes mainly belong to N_D^{rand}. Moreover, the rate reaches $\approx 2\%$ of the observation when driver nodes primarily of the N_D^{max} are compromised.

In contrast, power-topology distributions show a high robustness in connectivity and observability terms, where observation rate reaches values $\approx 100\%$. The global density is not affected even if CC mainly varies for small networks and the diameter specially impacts on both $N_D^{min,rand}$ for dense distributions with $\alpha = 0.5$ in $TG-1$ scenarios and $N_D^{max,min}$ for different exponents in $TG-3$ scenarios. Only for $TG-3$ the observability degree reaches the 90%, in addition to following similar behaviour pattern for any exponentiation value. While no effect is appreciated in diameter, the density decays only in small networks when two or more nodes are excluded from the graph. The consequences on the CC for small networks are not negligible, but the greatest consequences have been observed in observability when 3 nodes are removed.

For $TG-2$ scenarios, which results are summarized in Table 5.7, ER topologies continue to be very sensitive in connection terms where the diameter loosely changes for small distributions, and the global and local density drastically vary for any TGs. The observation rate is moderately high, but it presents certain weaknesses to attack models containing $AM-1$ to $AM-4$ aiming to break down $N_D^{max,rand}$.

The diameter in WS networks slightly changes for any $N_D^{strategy}$, where the global density remains invariant for $TG-1$ and its value notably decreases according to the number of isolated nodes, specifically for small networks, despite the drastic change in CC. The observation rate remains high, except for multi-interactive threat scenarios based on $TG-3$.

Lastly, common behaviors in $SCN-1$ and $SCN-2$ arise. The removal of random links in three vertices or the isolation of three vertices ($TG-3$) using the M-BC-O and L-BC-O combinations can cause the break down of the entire graph. These two configurations seem to be the most threatening, as the observability is largely influenced for any distribution and the diameter is drastically decreased for $N_D^{max,min}$. In addition, $AM-4$t threats stand out from the rest, underlying the importance of protecting to the node with the highest centrality within the network.

Table 5.8 shows the results obtained from attacks against a small number of random edges in a power-law sub-graph. Varying the exponent α it is observed that these types of networks have similar behavioral characteristics to those previously analized, e.g. the density for any distribution and for any TGs remains invariant after a planned elimination of edges ($SCN-1$). This is also seen for the observability which reaches values close to 100% at all times. Unfortunately, this observation degree decays extremely when the graph is subjected to attacks of type M-BC-O and L-BC-O, where two driver nodes and a vertex of the sub-graph outside the $N_D^{strategy}$ are simultaneously attacked. These two attack combinations are also dangerous in connectivity terms. The diameter values radically vary for any $N_D^{strategy}$ and for any distribution, although the global density remains broadly constant.

As expected, when the sub-graph is subjected to massive attacks in its adjacency matrix to isolate a single or multiple nodes, the diameter, density, and CC of the entire network vary, as shown in Table 5.9. The diameter primarily changes for any large distribution, whereas the local and global densities impact on small networks. As in the previous case, the observability is high at all the times, even if insignificant variations caused by attacks in N_D^{max} appear.

For both $SC-1$ and $SC-2$ attack types, the higher the exponent assigned to generate PLOD networks, the more robust network is, i.e. power-law networks with $\alpha = 0.1$ are less robust than networks with $\alpha = 0.5$. Most of the attacks that hamper connectivity are particularly linked to $AM-2$, $AM-3$ and $AM-4$, whereas the observation includes in addition $AM-1$. As previously mentioned, this also means that observability is a factor not only dependent on the network topology and construction strategies of driver nodes, but also on the nature of the perturbation where degree-based attacks and attacks to centrality are the most significant. On the other hand, scale-free and power-law distributions present analogous behaviors with respect to observability. Both are mainly vulnerable to threats given in N_D^{max} for small networks, and they are no only sensitive to $TG-3$ attacks, but also to $TG-4$ based on a planned F-M-BC-O attack.

Thereby, an adversary with sufficient knowledge of the network distribution and its power domination can disconnect the entire network and leave it without observation at very low cost. In this contexts, it may be necessary to construct suboptimal approximations to ensure the reconstruction of a power dominance relationship which might have been partially severed. This procedure involves taking into account the handicap of non-locality of PDS and the NP-hardness in most cases.

Nomenclature	Definition
n_d	Driver node
$AM-x$	Attack model following a particular attack strategy x, $x \in \{1,\dots,5\}$
$TG-x$	Number of target nodes, with $x \in \{1,\dots,5\}$
$N_D^{strategy}$	Set of driver nodes n_d following a particular controllability strategy ($N_D^{max,min,rand}$)
$N_D^{max,min,rand}$	Attack with low impact on structural controllability
$N_D^{max_\dagger,min_\dagger,rand_\dagger}$	Attack with intermediate impact on structural controllability
$N_D^{max_\ddagger,min_\ddagger,rand_\ddagger}$	Attack with high impact on structural controllability
$N_{D_{s,l,*}}^{strategy}$	Representation of small ($N_{D_s}^{strategy}$), large ($N_{D_l}^{strategy}$) and every ($N_{D_*}^{strategy}$) networks
$*, \{AM-x\}$	Influence of all attacks, but with a special vulnerability for $AM-x$
$\{X-AM-x\}$	Any X threat combined with $AM-x$
$x-y\%$	Minimum and maximum rate of observability

TABLE 5.2: List of symbols employed

A/N. The contributions and results exposed in this Chapter have been published in:

C. Alcaraz, **E. Etchevés Miciolino**, S. Wolthusen. *Structural Controllability of Networks for Non-Interactive Adversarial Vertex Removal*. In Critical Information Infrastructures Security – 8th International Workshop, CRITIS 13 – 16th-18th September 2013, Amsterdam (The Nederlands), Lecture Notes in Computer Science, vol. 8328, pp. 120-132, Springer, ISBN: 978-3-319-03963. (2013)

C. Alcaraz, **E. Etchevés Miciolino**, S. Wolthusen. *Multi-Round Attacks on Structural Controllability Properties for Non-Complete Random Graphs*. In Proceedings of the 16th International Conference of Information Security (ISC 2013), 13rd-15th November 2013, Dallas (TX – USA), Lecture Notes in Computer Science, vol. 7807, pp. 140-151, Springer, ISBN: 978-3-319-27658-8. (2015)

		ER				**WS**					**BA** with $\alpha = 0.3$					**PLOD** with $\alpha = 0.3$				
		50	100	500	1000	2000	50	100	500	1000	2000	50	100	500	1000	2000	50	100	500	1000	2000
									Before Attack												
	DA	6.66	17.39	80.03	157.86	312.17	1.66	2.00	1.97	1.99	1.99	24.50	30.16	1.97	1.99	1.99	2.06	2.04	2.08	2.10	2.11
	Dm	3	4	5	5	5	12	14	38	78	78	1	4	9	11	13	6	12	28	35	46
										$AM-1$											
N_D^{max}	Dm	3	4	5	5	5	12	16	38	78	78	1	4	9	11	13	6	12	28	35	46
N_D^{min}	Dm	3	4	5	5	5	12	14	38	78	78	1	4	9	11	13	6	12	28	35	46
N_D^{rand}	Dm	3	4	5	5	5	12	14	39	68	78	1	4	9	11	13	7	12	28	35	46
										$AM-2$											
N_D^{max}	Dm	4	4	5	5	5	12	14	38	78	78	1	4	9	11	13	6	12	28	35	46
N_D^{min}	Dm	3	4	5	5	5	12	14	37	78	78	1	4	9	11	13	6	12	28	35	46
N_D^{rand}	Dm	3	4	5	5	5	12	14	39	78	78	6	6	9	11	13	6	12	28	35	46
										$AM-3$											
N_D^{max}	Dm	3	4	5	5	5	9	14	38	78	78	1	4	9	11	13	7	12	28	35	46
N_D^{min}	Dm	4	4	5	5	5	9	14	38	78	78	1	4	9	11	13	6	12	28	35	46
N_D^{rand}	Dm	3	4	5	5	5	12	16	39	78	78	1	4	9	11	13	6	11	28	35	46
										$AM-4$											
N_D^{max}	Dm	4	4	5	5	5	9	15	45	78	78	1	4	9	11	13	8	11	25	33	51
N_D^{min}	Dm	4	4	5	5	5	9	15	45	78	78	1	4	9	11	13	9	11	25	33	51
N_D^{rand}	Dm	4	4	5	5	5	9	15	45	78	78	1	4	9	11	13	9	11	25	33	51
										$AM-5$											
N_D^{max}	Dm	4	4	5	5	5	12	14	38	78	78	1	4	9	11	13	6	12	28	35	46
N_D^{min}	Dm	4	4	5	5	5	12	14	38	78	78	1	4	9	11	13	6	12	28	35	46
N_D^{rand}	Dm	3	4	5	5	5	12	16	39	78	78	1	4	9	11	13	6	12	28	35	46

TABLE 5.3: Network diameter before and after the attack

		Diameter						CC						Observation Rate [%]					
		PLOD $\alpha = 0.1$			PLOD $\alpha = 0.5$			PLOD $\alpha = 0.1$			PLOD $\alpha = 0.5$			PLOD $\alpha = 0.1$			PLOD $\alpha = 0.5$		
		100	1000	2000	100	1000	2000	100	1000	2000	100	1000	2000	100	1000	2000	100	1000	2000
$AM-1$																			
N_D^{max}	10	36	36	14	25	46	0.0152	0.0028	0.0013	0.0192	0.0036	0.0007	99.0	99.7	99.9	98.0	99.7	100.0
N_D^{min}	10	36	36	14	25	46	0.0162	0.0028	0.0013	0.0169	0.0039	0.0007	100.0	99.8	99.85	100.0	99.8	100.0
N_D^{rand}	9	36	36	14	23	46	0.0180	0.0028	0.0013	0.0176	0.0036	0.0007	100.0	99.9	99.9	100.0	99.8	100.0
$AM-2$																			
N_D^{max}	10	36	36	14	25	46	0.0136	0.0028	0.0013	0.0198	0.0039	0.0007	99.9	99.7	99.9	98.0	99.9	100.0
N_D^{min}	10	36	36	14	25	46	0.0153	0.0028	0.0013	0.0195	0.0039	0.0007	100.0	99.8	99.85	100.0	99.8	100.0
N_D^{rand}	9	36	36	14	25	46	0.0155	0.0028	0.0013	0.0187	0.0039	0.0007	100.0	99.9	99.9	100.0	99.8	100.0
$AM-3$																			
N_D^{max}	10	36	36	14	25	46	0.0123	0.0028	0.0013	0.0179	0.0039	0.0007	99.0	99.7	99.9	98.0	99.9	100.0
N_D^{min}	12	36	36	14	25	46	0.0146	0.0028	0.0013	0.0181	0.0039	0.0007	100.0	99.8	99.85	100.0	99.8	100.0
N_D^{rand}	10	36	36	14	23	46	0.0158	0.0028	0.0013	0.0181	0.0039	0.0007	100.0	99.9	99.9	100.0	99.8	100.0
$AM-4$																			
N_D^{max}	12	36	38	11	26	51	0.0145	0.0028	0.0013	0.0161	0.0037	0.0007	99.0	99.7	99.9	97.0	99.9	100.0
N_D^{min}	12	36	38	11	26	51	0.0146	0.0028	0.0013	0.0151	0.0037	0.0007	100.0	99.8	99.85	100.0	99.8	100.0
N_D^{rand}	11	36	38	11	26	51	0.0148	0.0028	0.0013	0.0151	0.0037	0.0007	100.0	99.9	99.9	100.0	99.8	100.0
$AM-5$																			
N_D^{max}	10	36	36	14	25	46	0.0153	0.0028	0.0013	0.0187	0.0039	0.0007	99.0	99.7	99.85	97.0	99.8	99.95
N_D^{min}	10	36	36	14	23	46	0.0155	0.0028	0.0013	0.0198	0.0039	0.0007	99.0	99.8	99.8	100.0	99.8	99.95
N_D^{rand}	10	36	36	14	23	46	0.0158	0.0028	0.0013	0.0198	0.0039	0.0007	99.0	99.8	99.85	100.0	99.7	99.95

TABLE 5.4: Diameter and observation rate for PLOD with different exponents

		ER				WS				BA with $\alpha = 0.3$				PLOD with $\alpha = 0.3$			
		50	100	500	1000	2000	50	100	500	1000	2000	50	100	500	1000	2000	50	100	500	1000	2000
$AM-1$																					
N_D^{max}		92.0	86.0	99.8	99.5	99.95	96.0	89.0	99.8	99.7	99.9	20.0	97.0	100.0	100.0	100.0	98.0	100.0	100.0	99.6	100.0
N_D^{min}		100.0	100.0	100.0	100.0	100.0	100.0	98.0	100.0	100.0	100.0	100.0	99.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
N_D^{rand}		92.0	96.0	98.4	99.5	99.8	96.0	98.0	96.2	97.8	97.85	94.0	96.0	99.8	99.8	99.9	100.0	100.0	100.0	100.0	100.0
$AM-2$																					
N_D^{max}		100.0	86.0	100.0	99.8	99.9	100.0	90.0	99.8	99.9	99.95	20.0	95.0	100.0	100.0	100.0	98.0	100.0	100.0	100.0	100.0
N_D^{min}		100.0	100.0	100.0	100.0	100.0	100.0	98.0	100.0	99.9	100.0	100.0	99.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
N_D^{rand}		88.0	98.0	98.4	99.3	99.85	96.0	98.0	96.2	97.8	97.85	94.0	96.0	99.8	99.8	99.9	100.0	100.0	100.0	100.0	100.0
$AM-3$																					
N_D^{max}		100.0	91.0	100.0	99.9	100.0	100.0	91.0	100.0	99.9	100.0	20.0	96.0	100.0	100.0	100.0	98.0	100.0	100.0	100.0	100.0
N_D^{min}		100.0	100.0	99.8	99.9	99.95	98.0	98.0	100.0	100.0	100.0	100.0	99.0	100.0	100.0	99.95	100.0	100.0	100.0	100.0	100.0
N_D^{rand}		86.0	96.0	98.0	99.3	99.8	96.0	98.0	96.2	97.8	97.85	92.0	96.0	99.8	99.8	99.9	100.0	100.0	100.0	100.0	100.0
$AM-4$																					
N_D^{max}		98.0	90.0	99.8	99.9	99.95	100.0	91.0	99.8	100.0	99.95	20.0	97.0	100.0	100.0	100.0	98.0	100.0	100.0	100.0	100.0
N_D^{min}		100.0	99.0	99.9	99.9	99.95	98.0	98.0	99.8	100.0	99.85	100.0	98.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
N_D^{rand}		90.0	97.0	98.2	99.3	599.75	96.0	98.0	96.2	97.8	97.85	92.0	95.0	99.8	99.8	99.9	100.0	100.0	100.0	100.0	100.0
$AM-5$																					
N_D^{max}		98.0	90.0	99.8	99.9	99.95	98.0	90.0	99.8	99.9	99.95	50.0	95.0	100.0	100.0	100.0	98.0	100.0	99.8	100.0	99.95
N_D^{min}		98.0	99.0	99.8	99.9	99.95	98.0	98.0	99.8	99.9	99.95	100.0	98.0	100.0	100.0	100.0	98.0	99.0	99.8	100.0	99.95
N_D^{rand}		90.0	96.0	98.4	99.3	99.80	96.0	97.0	96.2	97.8	97.85	96.0	96.0	99.8	99.6	99.85	100.0	100.0	99.8	100.0	99.95

TABLE 5.5: Observation rate after perturbation or attack

TGx	Network	Connectivity: Diameter	Connectivity: Density	Connectivity: CC	Attack	Observability: Observation	Observability: Attack	Rate
TG-1	ER	$\mathbf{N}_{Ds}^{\max_{\ddagger},\min_{\dagger},\mathrm{rand}_{\dagger}}$	$\mathbf{N}_{Ds}^{\max,\min_{\dagger},\mathrm{rand}_{\dagger}}$	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	*	$\mathbf{N}_{Dl}^{\max_{\dagger},\min,\mathrm{rand}_{\dagger}}$	*	96.8-100%
	WS	$\mathbf{N}_{Ds}^{\max_{\dagger},\min_{\dagger},\mathrm{rand}}$	-	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	*,{BC}	$\mathbf{N}_{Ds}^{\max_{\dagger},\min,\mathrm{rand}}$	*	84-99%
	BA	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	-	-	*,{BC}	$\mathbf{N}_{Ds}^{\max_{\dagger},\min,\mathrm{rand}}$	*,{F}	16-100%
	PLOD $\alpha \simeq 0.1$	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	-	$\mathbf{N}_{Ds}^{\max_{\dagger},\min,\mathrm{rand}_{\dagger}}$	*,{BC}	-	-	$\simeq$ 100%
	PLOD $\alpha \simeq 0.3$	$\mathbf{N}_{Ds}^{\max,\mathrm{rand}}$	-	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	*,{BC}	-	-	$\simeq$ 100%
	PLOD $\alpha \simeq 0.5$	$\mathbf{N}_{Ds}^{\max,\min_{\ddagger},\mathrm{rand}_{\ddagger}}$	-	$\mathbf{N}_{Ds}^{\min,\mathrm{rand}}$	*	-	-	$\simeq$ 100%
TG-2	ER	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	$\mathbf{N}_{Ds}^{\max,\min_{\ddagger},\mathrm{rand}_{\dagger}}$	$\mathbf{N}_{Ds}^{\max,\min_{\dagger},\mathrm{rand}_{\dagger}}$	*,{X-BC}	$\mathbf{N}_{Dl}^{\max_{\dagger},\mathrm{rand}_{\dagger}}$	*	96.7-100%
	WS	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	$\mathbf{N}_{Ds}^{\min,\mathrm{rand}}$	$\mathbf{N}_{Ds}^{\max_{\dagger},\min_{\dagger},\mathrm{rand}_{\dagger}}$	*	$\mathbf{N}_{Ds}^{\max_{\dagger},\min,\mathrm{rand}}$	*	88-97.85%
	BA	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	$\mathbf{N}_{Ds}^{\min,\mathrm{rand}}$	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	*	$\mathbf{N}_{Ds}^{\max_{\dagger},\min,\mathrm{rand}}$	*,{F-BC,L-BC}	4-100%
	PLOD $\alpha \simeq 0.1$	$\mathbf{N}_{Ds}^{\max,\min_{\dagger},\mathrm{rand}_{\dagger}}$	-	$\mathbf{N}_{Ds}^{\max_{\dagger},\min,\mathrm{rand}}$	*,{F-BC, M-BC, BC-O}	-	-	$\simeq$ 100%
	PLOD $\alpha \simeq 0.3$	$\mathbf{N}_{Ds}^{\max,\min_{\dagger},\mathrm{rand}_{\dagger}}$	-	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	*,{M-BC, L-BC, BC-O}	-	-	$\simeq$ 100%
	PLOD $\alpha \simeq 0.5$	$\mathbf{N}_{Ds}^{\max,\min_{\dagger},\mathrm{rand}_{\dagger}}$	-	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}_{\dagger}}$	*	-	-	$\simeq$ 100%
TG-3	ER	$\mathbf{N}_{Ds}^{\max_{\ddagger},\min_{\dagger},\mathrm{rand}_{\dagger}}$	$\mathbf{N}_{Ds}^{\max,\min_{\dagger},\mathrm{rand}_{\dagger}}$	$\mathbf{N}_{Ds}^{\max,\min_{\dagger},\mathrm{rand}_{\dagger}}$	*,{M-BC-O, L-BC-O}	$\mathbf{N}_{Ds}^{\max_{\ddagger},\min_{\ddagger},\mathrm{rand}}$	*,{M-BC-O, L-BC-O}	0-100%
	WS	$\mathbf{N}_{Ds}^{\max_{\ddagger},\min_{\ddagger},\mathrm{rand}}$	-	$\mathbf{N}_{Ds}^{\max,\min_{\ddagger},\mathrm{rand}_{\ddagger}}$	*,{M-BC-O, L-BC-O}	$\mathbf{N}_{Ds}^{\max_{\dagger},\min_{\dagger},\mathrm{rand}}$	*,{M-BC-O, L-BC-O}	2-98%
	BA	$\mathbf{N}_{Ds}^{\max_{\ddagger},\min_{\ddagger},\mathrm{rand}}$	$\mathbf{N}_{Ds}^{\mathrm{rand}}$	-	*,{M-BC-O, L-BC-O}	$\mathbf{N}_{Ds}^{\max_{\ddagger},\min_{\ddagger},\mathrm{rand}}$	*,{F-M-L, M-BC-O, L-BC-O}	0-100%
	PLOD $\alpha \simeq 0.1$	$\mathbf{N}_{Ds}^{\max_{\ddagger},\min_{\ddagger},\mathrm{rand}}$	-	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	*,{M-BC-O, L-BC-O}	$\mathbf{N}_{Ds}^{\max_{\ddagger},\min_{\ddagger}}$	*,{M-BC-O, L-BC-O}	0-100%
	PLOD $\alpha \simeq 0.3$	$\mathbf{N}_{Ds}^{\max_{\ddagger},\min_{\ddagger},\mathrm{rand}}$	-	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	*,{M-BC-O, L-BC-O}	$\mathbf{N}_{Ds}^{\max_{\ddagger},\min_{\ddagger}}$	*,{M-BC-O, L-BC-O}	0-100%
	PLOD $\alpha \simeq 0.5$	$\mathbf{N}_{Ds}^{\max_{\ddagger},\min,\mathrm{rand}_{\dagger}}$	-	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}_{\dagger}}$	*,{M-BC-O, L-BC-O}	$\mathbf{N}_{Ds}^{\max_{\ddagger},\min_{\ddagger}}$	*,{M-BC-O, L-BC-O}	0-100%
TG-4	ER	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	$\mathbf{N}_{Ds}^{\max,\min_{\ddagger},\mathrm{rand}_{\dagger}}$	$\mathbf{N}_{Ds}^{\max,\min_{\ddagger},\mathrm{rand}_{\dagger}}$	*	$\mathbf{N}_{Dl}^{\max,\mathrm{rand}}$	*	96.4-100%
	WS	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	-	$\mathbf{N}_{Ds}^{\max_{\ddagger},\min_{\ddagger},\mathrm{rand}_{\ddagger}}$	*	$\mathbf{N}_{Ds}^{\max_{\ddagger},\min_{\dagger},\mathrm{rand}_{\dagger}}$	*	86-97.85%
	BA	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	$\mathbf{N}_{Ds}^{\mathrm{rand}}$	-	*,{F-M-L-O}	$\mathbf{N}_{Ds}^{\max_{\dagger},\min,\mathrm{rand}}$	*,{F-M-L-O, F-M-BC-O}	4-100%
	PLOD $\alpha \simeq 0.1$	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	-	$\mathbf{N}_{Ds}^{\max_{\dagger},\min,\mathrm{rand}}$	*	-	-	$\simeq$ 100%
	PLOD $\alpha \simeq 0.3$	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	-	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	*	-	-	$\simeq$ 100%
	PLOD $\alpha \simeq 0.5$	$\mathbf{N}_{Ds}^{\max,\min_{\dagger},\mathrm{rand}_{\dagger}}$	-	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	*	-	-	$\simeq$ 100%
TG-5	ER	$\mathbf{N}_{Ds}^{\mathrm{rand}}$	$\mathbf{N}_{Ds}^{\max_{\dagger},\min_{\dagger},\mathrm{rand}_{\dagger}}$	$\mathbf{N}_{Ds}^{\max_{\dagger},\min_{\dagger},\mathrm{rand}_{\dagger}}$	*	$\mathbf{N}_{Ds}^{\max,\mathrm{rand}}$	*	96.3-100%
	WS	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	-	$\mathbf{N}_{Ds}^{\max_{\ddagger},\min_{\ddagger},\mathrm{rand}_{\ddagger}}$	*	$\mathbf{N}_{Ds}^{\max_{\dagger},\min_{\dagger},\mathrm{rand}_{\dagger}}$	*	86-97.85%
	BA	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	$\mathbf{N}_{Ds}^{\mathrm{rand}}$	-	*	$\mathbf{N}_{Ds}^{\max_{\dagger},\min,\mathrm{rand}}$	*	14-100%
	PLOD $\alpha \simeq 0.1$	$\mathbf{N}_{Ds}^{\max,\mathrm{rand}}$	-	$\mathbf{N}_{Ds}^{\min,\mathrm{rand}}$	*	-	-	$\simeq$ 100%
	PLOD $\alpha \simeq 0.3$	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	-	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	*	-	-	$\simeq$ 100%
	PLOD $\alpha \simeq 0.5$	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	-	$\mathbf{N}_{Ds}^{\max,\min,\mathrm{rand}}$	*	-	-	$\simeq$ 100%

TABLE 5.6: $SCN-1$ Removal of a small number of edges $\in E$ from one or several vertices $\in V$. Refer to Table 5.2 for symbols.

TGs	Network	Connectivity: Diameter	Connectivity: Density	Connectivity: CC	Observability: Attack	Observability: Observation	Observability: Attack	Rate
TG-1	ER	$N_{Ds}^{max,min,rand}$	$N_{Ds}^{max\uparrow,min\uparrow,rand\downarrow}$	$N_{Ds}^{max\downarrow,min\uparrow,rand\downarrow}$	*, {E, BC}	$N_{Ds}^{max\uparrow,rand\uparrow}$	*, {E,M}	86-100%
	WS	$N_{D*}^{max,min,rand}$	-	$N_{D*}^{max\downarrow,min\uparrow,rand\downarrow}$	*, {BC}	$N_{Ds}^{max,min,rand}$	*	89-100%
	BA	-	-	-	-	$N_{Ds}^{max\downarrow,min,rand}$	*, {E, M, L, BC}	2-100%
	PLOD $\alpha \simeq 0.1$	$N_{D*}^{max,min,rand}$	-	$N_{Ds}^{max\uparrow,min,rand\downarrow}$	*, {BC}	$N_{D*}^{max,min,rand}$	*, {O}	99-100%
	PLOD $\alpha \simeq 0.3$	$N_{D*}^{max,min,rand}$	-	$N_{D*}^{max\downarrow,min\downarrow,rand\uparrow}$	*, {BC}	$N_{D*}^{max,min}$	*	98-100%
	PLOD $\alpha \simeq 0.5$	$N_{D*}^{max,min,rand}$	-	$N_{Ds}^{max\downarrow,min\uparrow,rand\uparrow}$	*, {BC}	N_{Ds}^{max}	*	97-100%
TG-2	ER	$N_{Ds}^{max,min,rand}$	$N_{D*}^{max\downarrow,min\downarrow,rand\downarrow}$	$N_{D*}^{max\downarrow,min\downarrow,rand\uparrow}$	*, {E, BC}	$N_{D*}^{max\downarrow,min,rand\uparrow}$	*,{E-M,F-BC,F-O}	70-100%
	WS	$N_{Ds}^{max,min,rand}$	$N_{D*}^{max,min,rand}$	$N_{D*}^{max\downarrow,min\downarrow,rand\downarrow}$	*	$N_{D*}^{max\downarrow,min,rand\uparrow}$	*,{F-O}	84-98%
	BA	-	$N_{Ds}^{min,rand}$	-	*	$N_{D*}^{max\downarrow,min,rand\uparrow}$	*	2-100%
	PLOD $\alpha \simeq 0.1$	$N_{D*}^{max,min,rand}$	-	$N_{Ds}^{max\uparrow,min,rand\downarrow}$	*, {F-BC, M-BC, L-BC, BC-O}	$N_{Ds}^{max,min,rand}$	*	99-100%
	PLOD $\alpha \simeq 0.3$	$N_{D*}^{max\uparrow,min\uparrow,rand\uparrow}$	$N_{Ds}^{max,min,rand\uparrow}$	$N_{Ds}^{max\uparrow,min\downarrow,rand\downarrow}$	*,{L-BC, BC-O}	$N_{Ds}^{max,min}$	*	98-100%
	PLOD $\alpha \simeq 0.5$	$N_{D*}^{max,min,rand}$	$N_{Ds}^{max,min,rand}$	$N_{Ds}^{max\downarrow,min\downarrow,rand\uparrow}$	*, {F-BC, M-BC, L-BC, BC-O}	N_{Ds}^{max}	*	97-100%
TG-3	ER	$N_{D*}^{max\downarrow,min\downarrow,rand}$	$N_{D*}^{max\downarrow,min\downarrow,rand\uparrow}$	$N_{D*}^{max\downarrow,min\downarrow,rand\uparrow}$	*,{M-BC-O, L-BC-0}	$N_{D*}^{max\downarrow,min\downarrow,rand\uparrow}$	*,{M-BC-O, L-BC-0}	0.15-99.90%
	WS	$N_{D*}^{max\downarrow,min\downarrow,rand\uparrow}$	$N_{Ds}^{max\downarrow,min,rand}$	$N_{D*}^{max\downarrow,min\downarrow,rand\downarrow}$	*,{M-BC-O, L-BC-0}	$N_{D*}^{max\downarrow,min\downarrow,rand}$	*,{M-BC-O, L-BC-0}	2-98%
	BA	$N_{D*}^{max\downarrow,min\downarrow,rand}$	$N_{Ds}^{min,rand}$	-	*,{M-BC-O, L-BC-0}	$N_{D*}^{max\downarrow,min\downarrow,rand\uparrow}$	*,{M-BC-O, L-BC-0}	0-100%
	PLOD $\alpha \simeq 0.1$	$N_{D*}^{max\downarrow,min\downarrow,rand}$	$N_{Ds}^{max,min,rand}$	$N_{Ds}^{max\downarrow,min,rand}$	*,{M-BC-O, L-BC-0}	$N_{D*}^{max\downarrow,min,rand}$	*,{M-BC-O, L-BC-0}	0.15-100%
	PLOD $\alpha \simeq 0.3$	$N_{D*}^{max\downarrow,min\downarrow,rand}$	$N_{Ds}^{max,min,rand}$	$N_{Ds}^{max,min,rand}$	*,{M-BC-O, L-BC-0}	$N_{D*}^{max\downarrow,min\downarrow}$	*,{M-BC-O, L-BC-0}	0-100%
	PLOD $\alpha \simeq 0.5$	$N_{D*}^{max\downarrow,min\downarrow,rand}$	$N_{Ds}^{max,min,rand}$	$N_{D*}^{max,min\uparrow,rand\uparrow}$	*,{M-I-O, L-BC-0} L-BC-O, M-BC-O}	$N_{D*}^{max\downarrow,min\uparrow,rand}$	*,{M-BC-O, L-BC-0}	0-100%
TG-4	ER	$N_{Ds}^{max,min,rand}$	$N_{D*}^{max\downarrow,min\downarrow,rand\downarrow}$	$N_{D*}^{max\downarrow,min\downarrow,rand\downarrow}$	*	$N_{D*}^{max\uparrow,min,rand\downarrow}$	*	66-99.90%
	WS	$N_{Ds}^{max,min,rand}$	$N_{Ds}^{max\downarrow,min\uparrow,rand\uparrow}$	$N_{D*}^{max\downarrow,min\downarrow,rand\downarrow}$	*	$N_{D*}^{max\downarrow,min\uparrow,rand\uparrow}$	*	82-97.85%
	BA	-	$N_{Ds}^{max,min,rand\uparrow}$	$N_{Ds}^{min,rand}$	*	$N_{Ds}^{max\downarrow,min,rand\downarrow}$	*	2-100%
	PLOD $\alpha \simeq 0.1$	$N_{D*}^{max,min,rand}$	$N_{Ds}^{max,min,rand}$	$N_{Ds}^{max\downarrow,min,rand\downarrow}$	*	$N_{Ds}^{max,min,rand}$	*	99-100%
	PLOD $\alpha \simeq 0.3$	$N_{D*}^{max,min,rand}$	$N_{Ds}^{max,min,rand}$	$N_{Ds}^{max,min,rand}$	*	$N_{Ds}^{max,min}$	*	98-100%
	PLOD $\alpha \simeq 0.5$	$N_{D*}^{max,min,rand}$	$N_{Ds}^{max,min,rand}$	$N_{Ds}^{max\uparrow,min\uparrow,rand\uparrow}$	*	N_{Ds}^{max}	*	96-100%
TG-5	ER	$N_{Ds}^{max,min,rand}$	$N_{D*}^{max\downarrow,min\downarrow,rand\downarrow}$	$N_{D*}^{max\downarrow,min\downarrow,rand\downarrow}$	*	$N_{D*}^{max\downarrow,min,rand\downarrow}$	*	68-99.85%
	WS	$N_{D*}^{max,min,rand}$	$N_{Ds}^{max\downarrow,min\uparrow,rand\uparrow}$	$N_{D*}^{max\downarrow,min\downarrow,rand\downarrow}$	*	$N_{D*}^{max\downarrow,min\uparrow,rand\uparrow}$	*	84-97.85%
	BA	-	$N_{Ds}^{max,min,rand\uparrow}$	$N_{Ds}^{min,rand}$	*	$N_{D*}^{max\downarrow,min,rand\downarrow}$	*	2-100%
	PLOD $\alpha \simeq 0.1$	$N_{D*}^{max,min,rand}$	$N_{Ds}^{max,min,rand}$	$N_{Ds}^{max\downarrow,min,rand\downarrow}$	*	$N_{Ds}^{max,min,rand}$	*	99-99.85%
	PLOD $\alpha \simeq 0.3$	$N_{D*}^{max,min,rand}$	$N_{Ds}^{max,min,rand}$	$N_{Ds}^{max,min,rand}$	*	$N_{Ds}^{max,min}$	*	98-100%
	PLOD $\alpha \simeq 0.5$	$N_{D*}^{max,min,rand}$	$N_{Ds}^{max,min,rand}$	$N_{Ds}^{max\uparrow,min\uparrow,rand\uparrow}$	*	N_{Ds}^{max}	*	96-100%

TABLE 5.7: $SCN - 2$: Isolation of one or several vertices $\in V$. Refer to Table 5.2 for symbols.

TGs	Network	Connectivity: Diameter	Connectivity: Density	Connectivity: CC	Observability: Attack	Observability: Observation	Observability: Rate	Attack
TG-1	PLOD $\alpha \simeq 0.1$	$N_{D*}^{max,min,rand\dagger}$	-	$N_{Ds}^{max,min,rand}$	{M, L, BC}	$N_{Ds}^{max,rand}$	{F, M, L, BC}	99,70-100%
	PLOD $\alpha \simeq 0.2$	N_{Dl}^{rand}	-	$N_{Ds}^{max,rand}$	{L}	-	-	$\simeq$ 100%
	PLOD $\alpha \simeq 0.3$	$N_{Dl}^{max,min}$	-	$N_{Ds}^{max,min,rand}$	{L, BC}	-	-	$\simeq$ 100%
	PLOD $\alpha \simeq 0.4$	$N_{D*}^{max,min,rand}$	-	$N_{Ds}^{max,min,rand}$	{M, L, BC}	N_{Ds}^{max}	*	98-100%
	PLOD $\alpha \simeq 0.5$	$N_{Dl}^{max,min}$	-	$N_{Ds}^{max,min,rand}$	{L, BC}	N_{Ds}^{max}	*	98-100%
TG-2	PLOD $\alpha \simeq 0.1$	$N_{D*}^{max\ddagger,min\ddagger,rand\ddagger}$	-	$N_{D*}^{max,min,rand\dagger}$	*,{X-BC}	$N_{Ds}^{max,rand}$	*	99,70-100%
	PLOD $\alpha \simeq 0.2$	$N_{Ds}^{max,min,rand}$	-	$N_{Ds}^{max,min,rand}$	{F-O, F-L, M-L, M-O}	-	-	$\simeq$100%
	PLOD $\alpha \simeq 0.3$	$N_{Dl}^{max\ddagger,min\ddagger,rand\dagger}$	-	$N_{Ds}^{max\ddagger,min\ddagger,rand}$	*	-	-	$\simeq$100%
	PLOD $\alpha \simeq 0.4$	$N_{D*}^{max\ddagger,min\ddagger,rand\dagger}$	-	$N_{Ds}^{max\ddagger,min,rand}$	*	N_{Ds}^{max}	*	97-100%
	PLOD $\alpha \simeq 0.5$	-	-	$N_{Ds}^{max\ddagger,min,rand}$	{L, BC}	N_{Ds}^{max}	*	98-100%
TG-3	PLOD $\alpha \simeq 0.1$	$N_{D*}^{max\ddagger,min\ddagger,rand\ddagger}$	-	$N_{Ds}^{max,min,rand\dagger}$	*,{M-BC-O, L-BC-0}	$N_{D*}^{max\ddagger,min\ddagger,rand\ddagger}$	*,{M-BC-O, L-BC-0}	0-100%
	PLOD $\alpha \simeq 0.2$	$N_{D*}^{max\ddagger,min\ddagger,rand\ddagger}$	-	$N_{Ds}^{max,min,rand}$	{M-L-O,M-BC-O} L-BC-0}	$N_{D*}^{max\ddagger,min\ddagger,rand\ddagger}$	* {M-BC-O, L-BC-0}	0-100%
	PLOD $\alpha \simeq 0.3$	$N_{D*}^{max\ddagger,min\ddagger,rand\ddagger}$	-	$N_{Ds}^{max\ddagger,min\ddagger,rand}$	*,{M-BC-O, L-BC-0}	$N_{D*}^{max\ddagger,min\ddagger,rand\ddagger}$	*,{M-BC-O, L-BC-0}	0-100%
	PLOD $\alpha \simeq 0.4$	$N_{D*}^{max\ddagger,min\ddagger,rand\ddagger}$	-	$N_{Ds}^{max\ddagger,min,rand}$	*,{M-BC-O, L-BC-0}	$N_{D*}^{max\ddagger,min\ddagger,rand\ddagger}$	*,{M-BC-O, L-BC-0}	0-100%
	PLOD $\alpha \simeq 0.5$	$N_{D*}^{max\ddagger,min\ddagger,rand\ddagger}$	-	$N_{Ds}^{max\ddagger,min\ddagger,rand}$	*,{M-BC-O, L-BC-0}	$N_{D*}^{max\ddagger,min\ddagger,rand\ddagger}$	*,{M-BC-O, L-BC-0}	0-100%
TG-4	PLOD $\alpha \simeq 0.1$	$N_{D*}^{max\ddagger,min\ddagger,rand\ddagger}$	-	$N_{Ds}^{max,min,rand\dagger}$	*	$N_{Ds}^{max,rand}$	*	99-100%
	PLOD $\alpha \simeq 0.2$	$N_{Dl}^{max,min,rand}$	-	$N_{Ds}^{max,min,rand}$	{F-L-BC-O,F-M-L-O} F-M-L-BC}	-	-	$\simeq$ 100%
	PLOD $\alpha \simeq 0.3$	$N_{Dl}^{max,min,rand}$	-	$N_{Ds}^{max\ddagger,min\ddagger,rand}$	*	-	-	$\simeq$ 100%
	PLOD $\alpha \simeq 0.4$	$N_{D*}^{max\ddagger,min\ddagger,rand\dagger}$	-	N_{Ds}^{max}	*,{F-M-BC-O}	N_{Ds}^{max}	*,{F-M-BC-O}	97-100%
	PLOD $\alpha \simeq 0.5$	N_{Ds}^{max}	-	$N_{Ds}^{max\ddagger,min\ddagger,rand}$	{M-L-BC-O}	N_{Ds}^{max}	*	98-100%
TG-5	PLOD $\alpha \simeq 0.1$	$N_{Dl}^{max\ddagger,min\ddagger,rand\ddagger}$	-	$N_{Ds}^{max,min,rand\dagger}$	*	$N_{Ds}^{max,rand}$	*	99-100%
	PLOD $\alpha \simeq 0.2$	$N_{Dl}^{min,rand}$	-	N_{Ds}^{rand}	*	-	-	$\simeq$ 100%
	PLOD $\alpha \simeq 0.3$	$N_{Dl}^{max,min,rand}$	-	$N_{Ds}^{max,min,rand}$	*	-	-	$\simeq$ 100%
	PLOD $\alpha \simeq 0.4$	$N_{Dl}^{max,min,rand}$	-	-	*	N_{Ds}^{max}	*	97-100%
	PLOD $\alpha \simeq 0.5$	-	-	$N_{Ds}^{max\ddagger,min,rand}$	*	N_{Ds}^{max}	*	98-100%

TABLE 5.8: $SCN-3$: Removal of a few edges ($SCN-1$) of a given sub-graph $\mathcal{G}_{sub} = (V, E)$. Refer to Table 5.2 for symbols.

TGs	Network	Connectivity: Diameter	Connectivity: Density	Connectivity: CC	Attack	Observability: Observation	Observability: Rate	Attack
TG-1	PLOD $\alpha \simeq 0.1$	$N_{D_s}^{\max_\dagger,\min_\dagger,\text{rand}_\dagger}$	-	$N_{D_s}^{\max,\min,\text{rand}_\dagger}$	{M, L, BC}	$N_{D_s}^{\max,\text{rand}}$	*	99-100%
	PLOD $\alpha \simeq 0.2$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max,\min,\text{rand}}$	*{L,BC}	-	-	$\simeq 100$
	PLOD $\alpha \simeq 0.3$	$N_{Dl}^{\max_\dagger,\min_\dagger,\text{rand}_\dagger}$	-	$N_{D_s}^{\max,\min,\text{rand}}$	{L, BC}	-	-	$\simeq 100$
	PLOD $\alpha \simeq 0.4$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max_\dagger,\min_\dagger,\text{rand}_\dagger}$	*{M, L, BC}	$N_{D_s}^{\max_\dagger,\text{rand}}$	*, {M, BC}	96-100%
	PLOD $\alpha \simeq 0.5$	$N_{D*}^{\max,\min,\text{rand}}$	-	$N_{D_s}^{\max,\min,\text{rand}}$	{M, L}	$N_{DS}^{\max}$	*	99,60-100%
TG-2	PLOD $\alpha \simeq 0.1$	$N_{D*}^{\max_\dagger,\min_\dagger,\text{rand}_\dagger}$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max,\min,\text{rand}_\dagger}$	{F-O, M-L, X-BC}	$N_{D_s}^{\max,\text{rand}}$	*	98-100%
	PLOD $\alpha \simeq 0.2$	$N_{D*}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max_\dagger,\min,\text{rand}_\dagger}$	*{F-O, M-L, L-O}	-	-	$\simeq 100\%$
	PLOD $\alpha \simeq 0.3$	$N_{Dl}^{\max,\min_\dagger,\text{rand}_\dagger}$	-	$N_{D_s}^{\min_\dagger,\text{rand}}$	{M-L, X-BC}	-	-	$\simeq 100\%$
	PLOD $\alpha \simeq 0.4$	$N_{D_s}^{\max_\dagger,\min_\dagger,\text{rand}_\dagger}$	$N_{D_s}^{\max_\dagger,\min_\dagger,\text{rand}_\dagger}$	$N_{D_s}^{\max_\dagger,\min_\dagger,\text{rand}_\dagger}$	*{X-BC}	$N_{D_s}^{\max_\dagger,\text{rand}}$	*, {F-X, X-BC}	97-100%
	PLOD $\alpha \simeq 0.5$	$N_{D*}^{\max_\dagger,\min,\text{rand}}$	-	$N_{D_s}^{\max_\dagger,\min_\dagger,\text{rand}}$	*,{F-L, M-L, X-BC}	$N_{DS}^{\max}$	*,{BC-O}	96-100%
TG-3	PLOD $\alpha \simeq 0.1$	$N_{D*}^{\max_\dagger,\min_\ddagger,\text{rand}_\ddagger}$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max,\min,\text{rand}_\dagger}$	*,{M-BC-O, L-BC-O}	$N_{D*}^{\max_\ddagger,\min_\ddagger,\text{rand}_\ddagger}$	*	0-100%
	PLOD $\alpha \simeq 0.2$	$N_{D*}^{\max_\dagger,\min_\ddagger,\text{rand}_\ddagger}$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max_\dagger,\min,\text{rand}_\dagger}$	*,{M-BC-O, L-BC-O}	$N_{D*}^{\max_\ddagger,\min_\ddagger,\text{rand}_\ddagger}$	*,{M-BC-O, L-BC-O}	0-100%
	PLOD $\alpha \simeq 0.3$	$N_{D*}^{\max_\ddagger,\min_\ddagger,\text{rand}_\ddagger}$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max,\min_\dagger,\text{rand}}$	*,{M-BC-O, L-BC-O}	$N_{D*}^{\max_\ddagger,\min_\ddagger,\text{rand}_\ddagger}$	*,{M-BC-O, L-BC-O}	0-100%
	PLOD $\alpha \simeq 0.4$	$N_{D*}^{\max_\ddagger,\min_\ddagger,\text{rand}_\ddagger}$	$N_{D_s}^{\max_\dagger,\min_\dagger,\text{rand}_\dagger}$	$N_{D_s}^{\max_\ddagger,\min_\dagger,\text{rand}_\dagger}$	*,{M-BC-O, L-BC-O}	$N_{D*}^{\max_\ddagger,\min_\ddagger,\text{rand}_\ddagger}$	*,{M-BC-O, L-BC-O}	0-100%
	PLOD $\alpha \simeq 0.5$	$N_{D*}^{\max_\ddagger,\min_\ddagger,\text{rand}_\ddagger}$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max_\dagger,\min_\dagger,\text{rand}_\dagger}$	*,{M-BC-O, L-BC-O}	$N_{D*}^{\max_\ddagger,\min_\ddagger,\text{rand}_\ddagger}$	*,{M-BC-O, L-BC-O}	0-100%
TG-4	PLOD $\alpha \simeq 0.1$	$N_{D*}^{\max_\ddagger,\min_\ddagger,\text{rand}_\ddagger}$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max,\min,\text{rand}_\dagger}$	*	$N_{D_s}^{\max,\min,\text{rand}}$	*	99-100%
	PLOD $\alpha \simeq 0.2$	$N_{D*}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max,\min,\text{rand}}$	*	-	-	$\simeq 100\%$
	PLOD $\alpha \simeq 0.3$	$N_{Dl}^{\max_\dagger,\min_\dagger,\text{rand}_\dagger}$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\min_\dagger,\text{rand}}$	*	-	-	$\simeq 100\%$
	PLOD $\alpha \simeq 0.4$	$N_{D*}^{\max_\ddagger,\min_\ddagger,\text{rand}_\ddagger}$	$N_{D_s}^{\max_\ddagger,\min_\ddagger,\text{rand}_\ddagger}$	$N_{D_s}^{\max_\ddagger,\min_\dagger,\text{rand}_\dagger}$	*	$N_{D*}^{\max_\dagger,\text{rand}}$	*	96-100%
	PLOD $\alpha \simeq 0.5$	$N_{D*}^{\max_\dagger,\min,\text{rand}}$	$N_{D*}^{\max,\min,\max}$	$N_{D*}^{\max_\dagger,\min,\text{rand}}$	*	$N_{DS}^{\max}$	*	96-100%
TG-5	PLOD $\alpha \simeq 0.1$	$N_{D*}^{\max_\ddagger,\min_\ddagger,\text{rand}_\ddagger}$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max,\min,\text{rand}}$	*	$N_{D_s}^{\max,\min,\text{rand}}$	*	99-100%
	PLOD $\alpha \simeq 0.2$	$N_{Dl}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\max,\text{rand}}$	*	-	-	$\simeq 100\%$
	PLOD $\alpha \simeq 0.3$	$N_{D*}^{\max_\dagger,\min,\text{rand}}$	$N_{D_s}^{\max,\min,\text{rand}}$	$N_{D_s}^{\min_\dagger,\text{rand}}$	*	-	-	$\simeq 100\%$
	PLOD $\alpha \simeq 0.4$	$N_{D*}^{\max_\ddagger,\min_\ddagger,\text{rand}_\ddagger}$	$N_{D_s}^{\max_\ddagger,\min_\ddagger,\text{rand}_\ddagger}$	$N_{D_s}^{\max,\min_\ddagger,\text{rand}}$	*	$N_{D*}^{\max_\dagger,\text{rand}}$	*	96-100%
	PLOD $\alpha \simeq 0.5$	$N_{D*}^{\max_\dagger,\min,\text{rand}}$	$N_{D_s}^{\max,\min,\max}$	$N_{D*}^{\max_\dagger,\min,\text{rand}}$	*	$N_{DS}^{\max}$	*	96-100%

TABLE 5.9: $SCN - 3$: Isolation of vertices ($SCN - 2$) of a given sub-graph $\mathcal{G}_{sub} = (V, E)$. Refer to Table 5.2 for symbols.

Chapter 6

The FACIES Project

As mentioned in the introduction chapter, this work has been carried out in the framework of the European Project FACIES, in which the University Campus Bio-Medico of Rome, University "Roma Tre", University of Cyprus, University of Malaga and Radiolabs were involved, gathering their expertise and experience in different fields. The aim was to develop cooperation strategies for the early automatic detection of failures and attacks in interdependent Critical Infrastructures. To such end, a cyber-physical testbed has been designed and created, in which the studied techniques could be implemented, tested and validated through the induction of physical faults and cyber-attacks to the system.

6.1 The FACIES Testbed

The realization of the testbed aims to emulate the operation of a real CI, to reproduce its main characteristics and the effects of interdependencies with other CIs. Specifically, a scaled version of a water transmission system has been chosen as scenario. From one side, this testbed is physically highly modular, able to reproduce several types of operating scenarios, and its design provided for a high flexibility, allowing to experimentally deploy the system in 14 different configurations. On the other hand, it includes all the elements that characterize a modern control network, i.e. sensors and actuators, Programmable Logic Controllers (PLCs), a SCADA (Supervisory Control and Data Acquisition) system and the communication network, as well as specific monitoring modules to perform physical fault and cyber-attack detection. Hence, the system can be deployed to test the effectiveness of several types of control strategies implemented at different hierarchical levels, and can take into account several water consumption profiles. In addition, it is possible to introduce 38 different and independent physical faults, as water leaks and actuator/sensors faults, and a wide number of cyber events can be emulated on the control architecture, ranging from simple DoS attacks to more sophisticated replay and covert attacks [69]. Thereby, the testbed comprises also a number of modules developed for the automatic detection of physical fault and cyber-attacks. To conclude, the interaction and interdependencies with two other CIs have also been considered, namely a dam, designated for the continuous water supply, and a plant for the electric power generation, which requires an important amount of water in order to launch the refrigerating processes in the cooling towers. Thereby, the FACIES testbed is able to reproduce also consequences of negative events that may take place in other infrastructures (e.g. electricity and telecommunications infrastructures), simulating the domino effects induced by dependencies and interdependencies in a system of infrastructures [111].

6.1.1 HighLake City: the Chosen Scenario

For the EU Project FACIES, the water transmission system of a small fictional city, called "HighLake City", has been chosen as CI case study. Such choice was based on the fact that the temporal variables deployed for modeling the water phenomena are much slower with respect to other systems, and consequently are easier to reproduce and study. On the other hand, the modeled phenomena are highly non-linear, which represents an interesting feature for an emulated system that is pretended to be as real as possible.

A scaled down version has been theoretically designed, as well as its daily operation opportunely scaled over a reduced time range, maintaining a good level of similarity to an actual small city. The scenario, as depicted in Figure 6.1, consists of two *Residential Areas*, each divided in two sub-areas, and one *Industrial Area*, the three of which differ by altitude, characteristics and water demand from the customers. More specifically, the *Industrial Area* is considered to be located in a hilly zone, in a higher position with respect to the *Residential Areas* and the main water supply system.

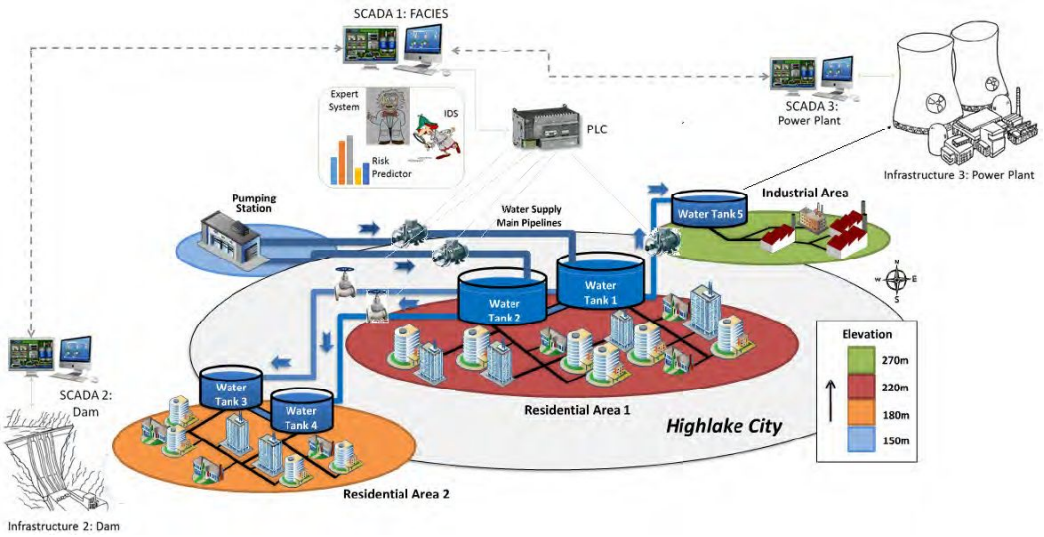

FIGURE 6.1: HighLake City - The chosen scenario

The water distribution system is projected to cover the whole city. It consists of five main water tanks for storage, each one destined to a single (sub-)area and located at its highest point, so as to provide water to the customers primarily by gravity through the water distribution network. *Tanks 1* and *2* are located at an altitude of $220m$, with $15000m^3$ capacity each. They are filled from the main water treatment plant of the country by a pipeline, and provide water through the distribution network to two urban zones in the *Residential Area 1*, as well as to the other three main tanks of the city. The water *Tanks 3* and *4*, receive water by gravity from *Tanks 1* and *2*, respectively, as they are located at a lower position, specifically at $180m$ from the sea level. Their input flow is regulated by a system of valves located along the pipeline. On the other hand, *Tank 5*, with $6000m^3$ capacity, is located at an altitude of $50m$ above *Tank 1*'s position and provides water to the *Industrial Area*. Hence, water is transferred through a pipeline from *Tank 1* to *Tank 5* by means of centrifugal pumps, able to provide the required head.

When all the five tanks are full, the total volume of water that can be provided to the city is $46000m^3$. According to the population considered in this scenario, and the size of the water tanks in each area, both residential areas can have up to 40 hours of continuous water supply [112], whereas the industrial area up to 32 hours [113]. During the normal operation, the system has to satisfy an uncertain evolving demand from the diferent areas, and it is required that the pressure at the output of the main tanks is kept within a specific bound. Pressure should not drop below the lower bound, otherwise part of the areas may suffer a leak of water supply. On the other hand, the upper bound should not be exceeded in order not to induce the water system into physical damage risk. Moreover, leaks should not be present along the water distribution network, as water is not to be wasted. To achieve these objectives, sensor information is collected using the SCADA system, which controls the actuators (pumps and valves), according to the water demand from the consumers. More specifically, the SCADA system is responsible for monitoring and controlling the water supply and distribution system of HighLake City. It receives real-time data from a number of sensors located in different points of the network and controls the actuators, guaranteeing the requested quality of service to the customers in almost every situation.

Besides the aforementioned water distribution system, two other CIs have been considered, which operation is closely related to the former. More specifically, water for HighLake City derives from a nearby lake where a dam – *Infrastructure 2* – is located, and the *Industrial Area* provides water for the cooling of a power plant – *Infrastructure 3*. Hence, notifications and reports about the current operation of the water supply and distribution system have to be exchanged, as well as alarms in case of faults, failures or attacks, in order to guarantee an adequate response and the needed countermeasures for each CI.

6.1.2 The Daily Water Demand

Generally, during a day the water distribution system of a city has to supply an unknown demand from the customers, being efficient enough to satisfy an evolving request from the different areas [114]. For HighLake City a normal daily scenario has been hypothesized and implemented, developing and codifying a a daily water demand curve, so as to perform experimental runs deploying the testbed. Typically, during the first hours of the morning (6-9 am) the demand both from the *Residential* and *Industrial Areas* increases because of the morning activities of the population, which is getting ready to face the day. Then, as depicted in Figure 6.2, the demand in the *Residential Areas* tend to decrease until lunch-time, when a slight growth can be noticed. The demand there decreases again during the afternoon, and after 6 pm it raises, as people goes back home, wash themselves, prepare dinner, make housework, etc. On the other hand, as industries and factories may close, the water demand in the *Industrial Area* sensibly decreases. Finally, during the night, the demand definitively diminishes also in the *Residential Areas*, reaching the lower daily level.

Such pattern has been obtained by the sequential opening and closing of the valves and pumps at predefined time instants, making so that the total water demand of the city follows the target water demand curve as close as possible. The simulation has been scaled down to a 6 minutes time interval, representing the 24 hours operation. Moreover, specific rules have been implemented for the supply pumps and valves, considering a lower and upper bound for the water level in each tank, in order to avoid water shortages or overflows by taking the proper actions in each case.

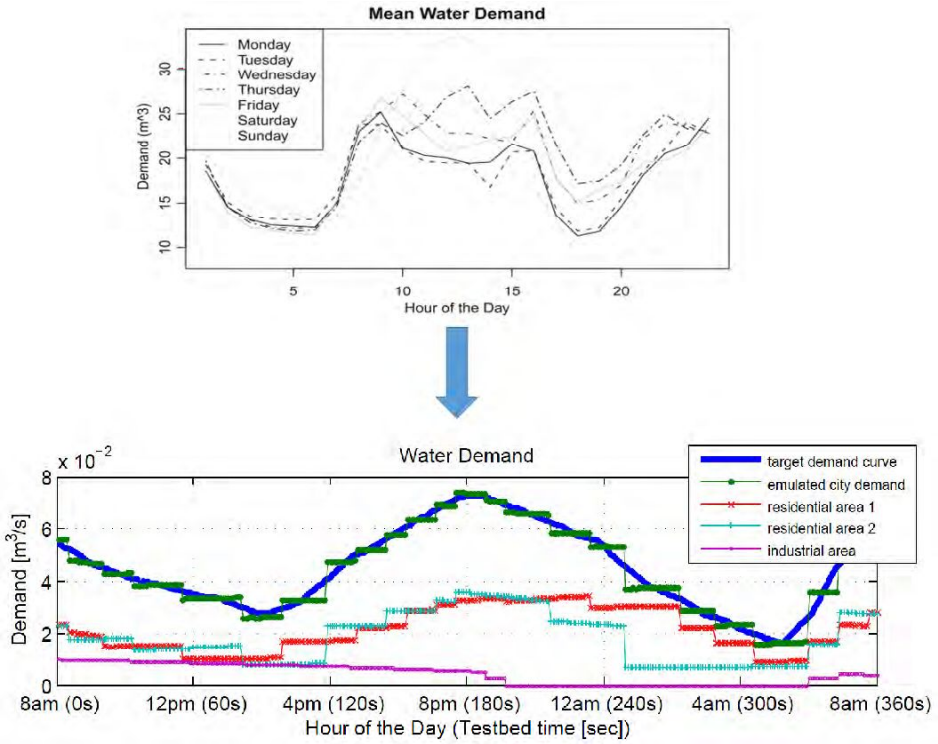

FIGURE 6.2: Example of mean water demand in 24h scaled down to 6 min scenario

6.1.3 Requirements Analysis

The emulator system was designed to reproduce some of the peculiarities of a CI with the corresponding control system. The goal is to test the capability of different techniques to detect and identify faults, failures, physical and/or cyber-attacks. Thereby, the main requirements for the realization of the testbed were:

Complexity emulation:

- Multiple subsystems;
- MIMO subsystems;
- Soft-wired topology, allowing rewiring and reconfiguration;
- Non linearity.

Faults/attacks injection:

- Sensor/actuator fault emulation;
- Leaks in tanks or pipelines induction;

Independent control/supervision routines:

- Unsupervised inputs;
- Unsupervised states.

The presence of the SCADA and the HMI (Human-Machine Interface) shall allow the supervision the system at all times. It is to be connected to the PLCs through a wired communication system, as well as the sensors and actuators distributed in the testbed.

6.2 Testbed Realization

6.2.1 The Physical System

As previously described, the system consists of five tanks of different dimensions and a reservoir. The main *Residential Area* of HighLake City, represented by *Tank 1* and *Tank 2*, is supplied with water thanks to two main sources coming from the pumping station, i.e. the reservoir. These two tanks supply water by gravity to the two tanks representing the secondary *Residential Area*, i.e. *Tank 3* and *Tank 4*, and to the *Industrial Area* represented by *Tank 5*, by means of a pump, as it is located at a higher altitude. The reservoir operates both as the main water source for the pumping station, and as sink for all the tanks. It is hypothetically characterized by a constant head and infinite water capacity. In the studies carried out, only the water levels in each tank and the water demand from the different areas have been analyzed, neglecting the pressure values in the system.

The schema designed for the realization of the testbed is depicted in Figure 6.3, where the black paths describe the nominal/healthy connections among the different tanks, and the grey dashed lines correspond to the leaks that can be induced along the pipelines by the use of manual valves, which flow can be arbitrarily regulated.

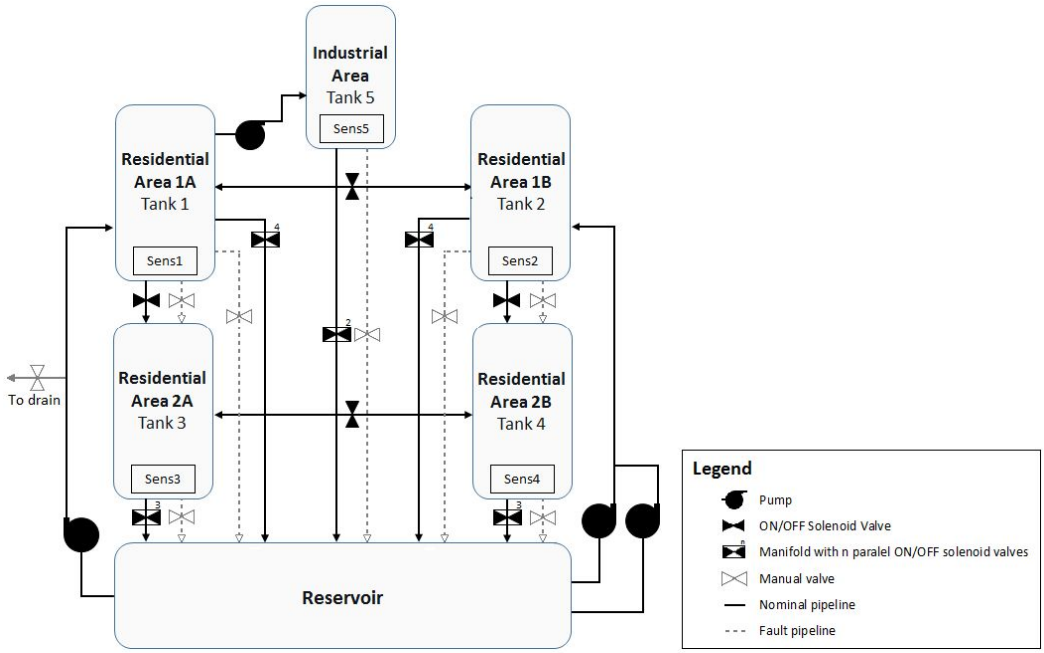

FIGURE 6.3: Testbed schema

The manner in which the different tanks are connected allows to have a large number of different configurations (serial, parallel, crossed-connections and their combinations), which endow the testbed with high flexibility and allow to represent many scenarios. Specifically, 14 different configurations can be obtained by using 1 to 5 tanks, with different connections between them and different input flows, by considering only the nominal connections (i.e. black paths in Figure 6.3). Moreover, for each configuration, the output flow of the single tanks can be discretely modulated by opening a different number of output valves in the manifolds. Thereby, it is possible to perform a huge number of experiments, ranging from the system identification, to the accuracy analysis of models, to simple or more sophisticated control schemes. This feature has been used, as better illustrated in Section 6.2.4, to modulate the tanks output in order to reproduce the daily water demand.

Components

For the structure, the components and parts deployed have been the following:

- $6\times$ polyethylene barrels with different capacities:

  $2 \times 25L$ for *Residential Area 1*, $A = 573cm^2$;
  $2 \times 10L$ for *Residential Area 2*, $A = 346cm^2$;
  $1 \times 5L$ for *Industrial Area*, $A = 707cm^2$.
  $1 \times 125L$ as reservoir.

- Pipes - multilayer system, cross-linked polyethylene, $16mm\emptyset$;

- $3\times$ centrifugal pumps supplying *Residential Area 1*:

  Mini-type pipe pump – Model 151410 - $Q_{max} = 20\frac{L}{min}, H_{max} = 10m, 220V, 0.09W$

- $1\times$ centrifugal pump supplying *Industrial Area*:

  Water pump EK-DCP 2.2 - $Q_{max} = 6\frac{L}{min}, H_{max} = 2.2m, 12V\,DC, 6.5W$

- $5\times$ pressure/level sensors for the tanks:

  GEMS Series 11700 - $1\frac{1}{2}''$ G thread male connector, $0-1m_{H_2O}(0-0.1bar), 4-20mA$

- $16\times$ solenoid valves to represent the water demand:

  Evian Series 263 – Model DVP-72 - $\frac{1}{4}''$ G thread female connectors, $0-1bar$, ON/OFF (nOFF), $24V\,DC$ input
  $8\times$ for demand from *Residential Area 1*;
  $6\times$ for demand from *Residential Area 2*;
  $2\times$ for demand from *Industrial Area*.

- $2\times$ solenoid valves supplying water from *Tanks 1* and *2* to *Tanks 3* and *4*

  ODE Series 21HT - Model 4K0Y160 - $\frac{1}{2}''$ G thread female connectors, $0-14bar$ - ON/OFF (nOFF)

- $2\times$ solenoid valves to represent the crossed-connection between sub-areas

Evian Series 263 – Model DVP-72 - $\frac{1}{4}''$ G thread female connectors, $0 - 1bar$, ON/OFF (nOFF), $24VDC$ input

- $7\times$ manual valves representing leaks along pipes and in tanks
- $1\times$ manual valve to drain
- $1 \times 3m$, $3 \times 2m$ and $1 \times 50cm$ length, $(40 \times 40mm)$ aluminum profiles
- $3 \times (1 \times 1m)$ and $1 \times (40 \times 40cm)$ expanded polyurethane panels

The expanded polyurethane panels have been disposed horizontally, where the reservoir and the tanks have been positioned at $5cm$ (reservoir), $60cm$ (*Tanks 3* and *4*), $150cm$ (*Tanks 1* and *2*) and $178cm$ (*Tank 5)* height, respectively.

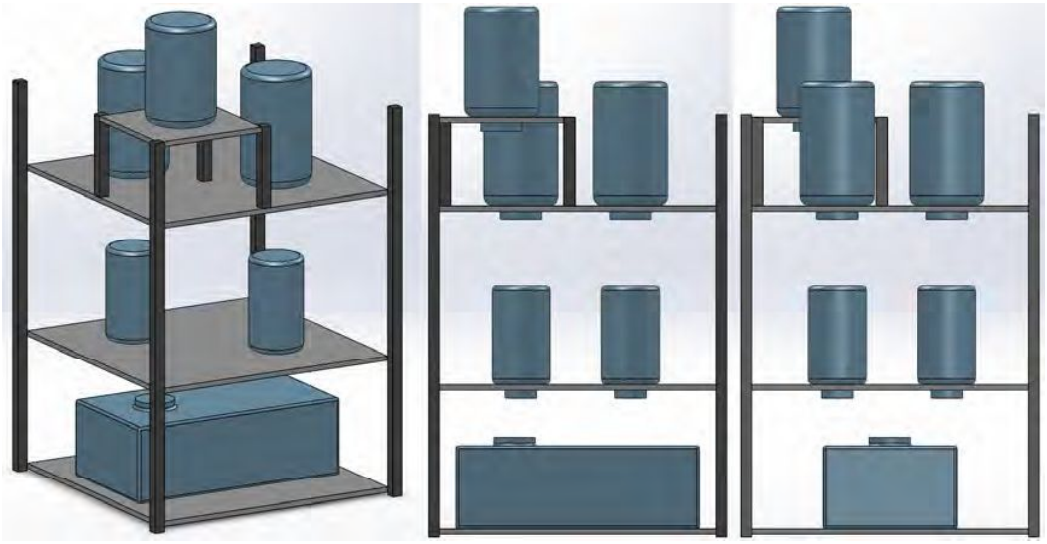

FIGURE 6.4: Testbed structure

The 16 electro-valves simulating the water demand of each area have been disposed in 5 manifolds, one for each tank. The P&ID diagram shown in Figure 6.5 represents the overall implemented structure. There, also the electric connections between the PLC and the different components on the field are depicted, as well as all the communication channels.

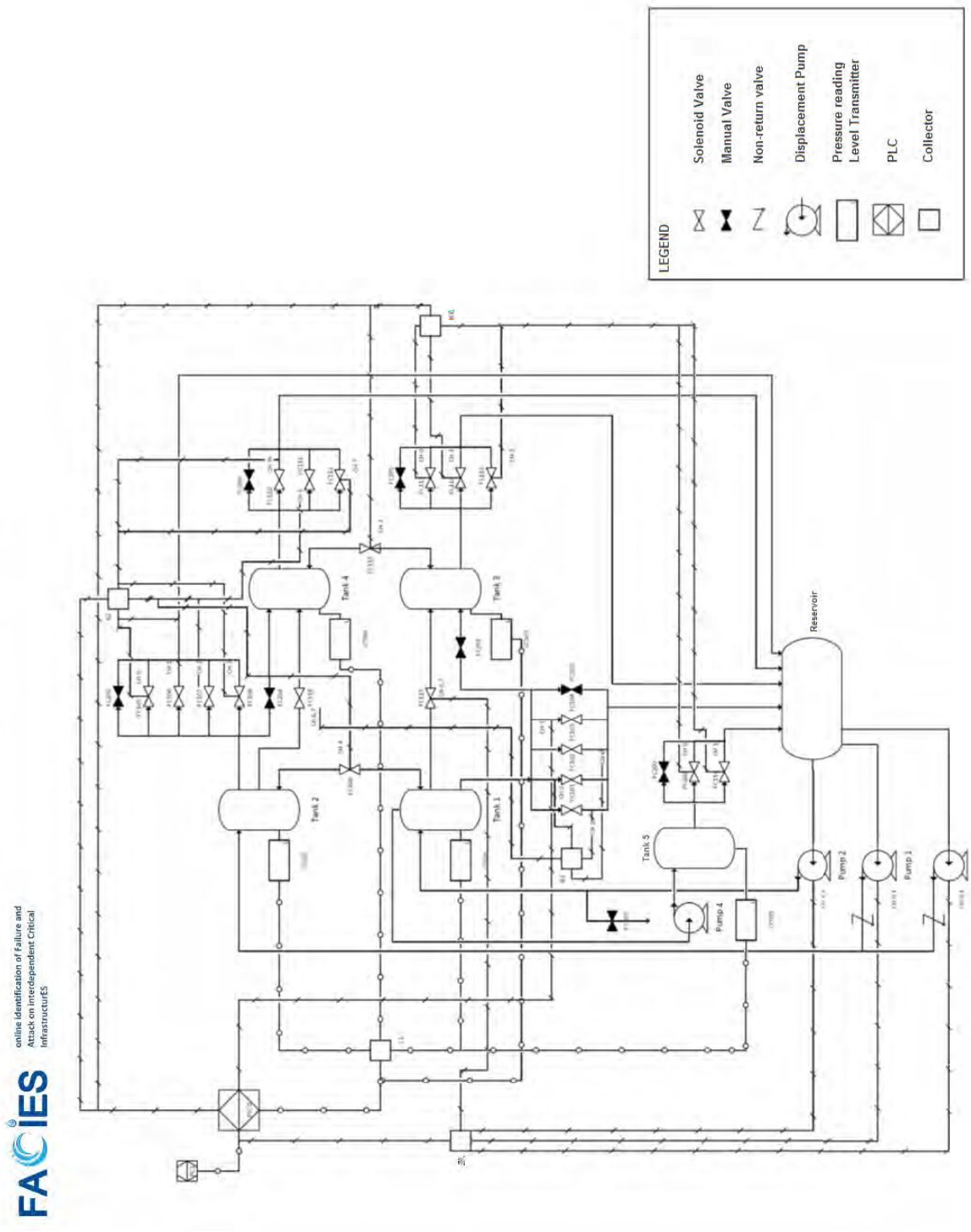

FIGURE 6.5: P&ID diagram of the testbed

For all the actuators, a classification can be found in Tables 6.1, 6.2 and 6.3. In addition, each tank is provided with a level sensor, excluding the reservoir, as summarized in Table 6.4. The current testbed is shown in Figure 6.6.

Electro-Valves			
Connection	**Manifold**	**HMI Name**	**P&ID Name**
Tank 1 - Reservoir	1	V.1.1, V.1.2, V.1.3, V.1.4	FC101, FC102, FC103, FC104
Tank 2 - Reservoir	2	V.2.5, V.2.6, V.2.7, V.2.8	FC105, FC106, FC107, FC108
Tank 4 - Reservoir	3	V.3.10, V.3.11, V.3.12	FC110, FC111, FC112
Tank 3 - Reservoir	4	V.4.13, V.4.14, V.4.15	FC113, FC114, FC115
Tank 5 - Reservoir	5	V.5.19, V.5.10	FC119, FC120
midrule Tank 1 - Tank 3		V.D.18	FC117
Tank 2 - Tank 4		V.D.17	FC118
Tank 1 - Tank 2		V.C.9	FC109
Tank 3 - Tank 4		V.C.16	FC116

TABLE 6.1: Electro-valves classification

Pumps		
Connection	**HMI Name**	**P&ID Name**
Reservoir - Tank 1	Pump 2	Pump 2
Reservoir - Tank 2	Pump 1, Pump 3	Pump 1, Pump 3
Reservoir - Tank 5	Pump 4	Pump 4

TABLE 6.2: Pumps classification

Manual Valves		
Connection	**HMI Name**	**P&ID Name**
Tank 1 - Reservoir	-	FC201
Tank 2 - Reservoir	-	FC202
Tank 3 - Reservoir	-	FC205
Tank 4 - Reservoir	-	FC206
Tank 5 - Reservoir	-	FC208
Tank 1 - Tank 3	-	FC203
Tank 2 - Tank 4	-	FC204
Reservoir - Sink	-	FC207

TABLE 6.3: Manual valves classification

Sensors		
Connection	**HMI Name**	**P&ID Name**
Tank 1	Tank 1	LT101
Tank 2	Tank 2	LT102
Tank 3	Tank 3	LT103
Tank 4	Tank 4	LT104
Tank 5	Tank 5	LT105

TABLE 6.4: Sensors classification

FIGURE 6.6: Current testbed

6.2.2 Dynamic Characterization

A large number of experiments have been carried out in order to study the dynamic behavior of the system for different operating conditions. In Figures 6.7 to 6.17 the results for the filling and emptying of the tanks in various operative conditions are shown, created with the data recorded on the testbed database. Table 6.5 summarizes the results for the different tanks, considering the time required for each one.

The differences between the filling times depend mainly on the length of the pipes connecting the pumps or valves from the source to the specific tank. *Tank 2* presents two values, as it can be supplied by one or two pumps in parallel. As the emptying process that takes place only by gravity, the minimum time has been obtained by opening all the output valves in the manifold connected to the specific tank, while the maximum is calculated by opening only one valve on it.

Tank	Filling Time [s]	Emptying Time	
		min [s]	max [s]
Tank 1	139	285	586
Tank 2	229 | 151	237	549
Tank 3	148	251	577
Tank 4	162	215	497
Tank 5	264	280	528

TABLE 6.5: Filling and emptying times. *Tank 2* can be filled with two different input flow rates, by using one or two pumps, respectively. For the emptying, all the valves of the manifold are opened to obtain the minimum time, while only one valve in the manifold is opened for the maximum emptying time.

In Figure 6.7 the level of *Tank 1* during filling with a constant input flow from Pump 2 is depicted, with all valves closed, and with one or more valves open. The fifth valve opened is V.D.18, connecting *Tank 1* to *Tank 3* and, as it can be seen in the chart, its output flow if much higher than the one in the manifold. In fact, the level in the tank reaches an equilibrium value and is not totally filled.

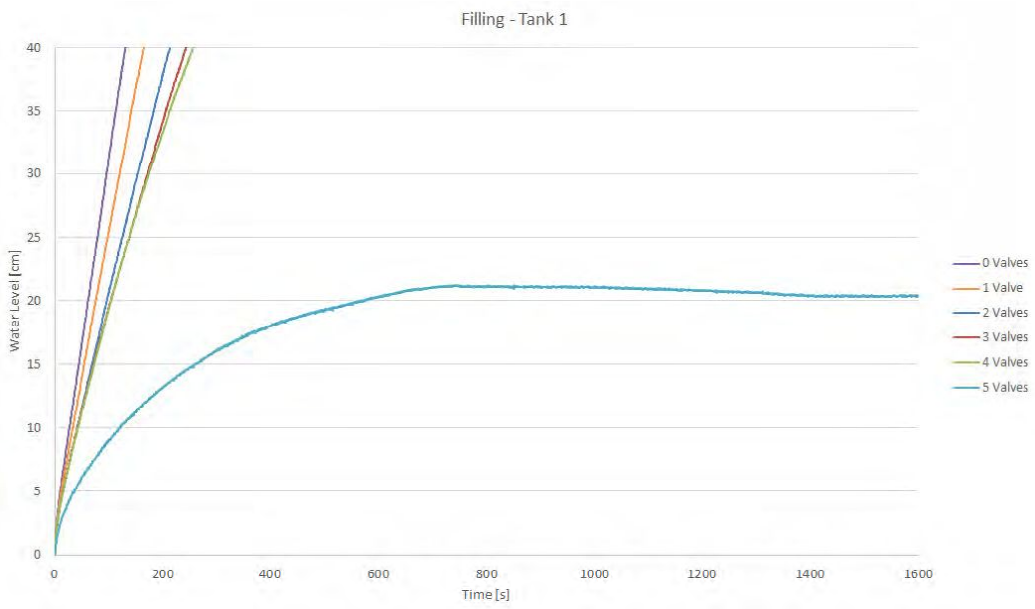

FIGURE 6.7: *Tank 1* filling times with a different number of opened output valves

As *Tank 2* is provided with two pumps, in Figure 6.8 and Figure 6.9 both cases are reported. In the former, no more than 3 valves from the manifold have been opened as the output flow became higher than the input flow. Moreover, it is evident from the two charts how the input flow, hence the filling velocity, grew when deploying both pumps, despite the number of opened valves.

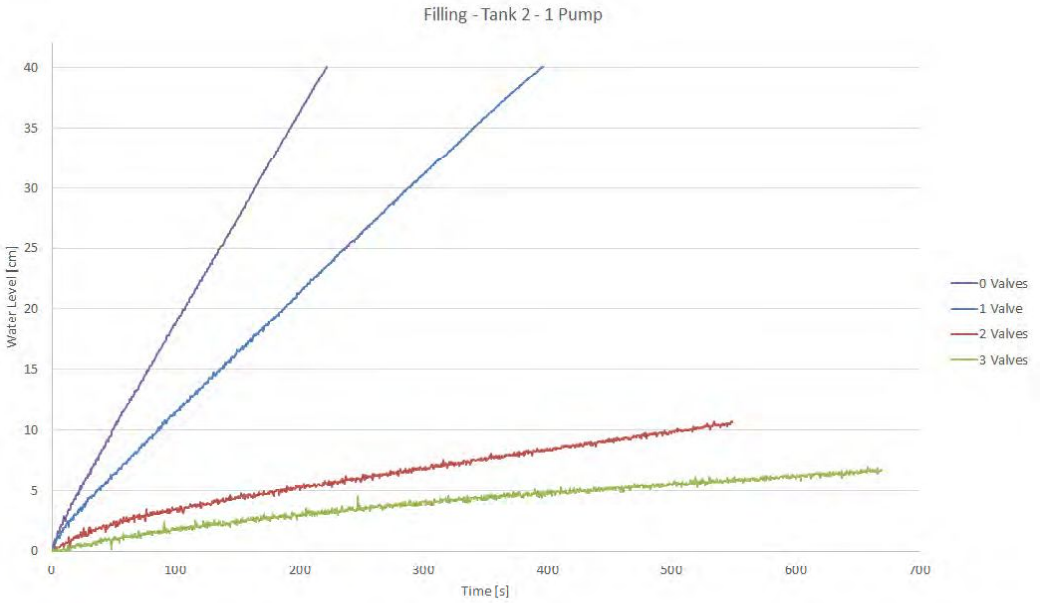

FIGURE 6.8: *Tank 2* filling times with 1 supply pump on and a different number of opened output valves

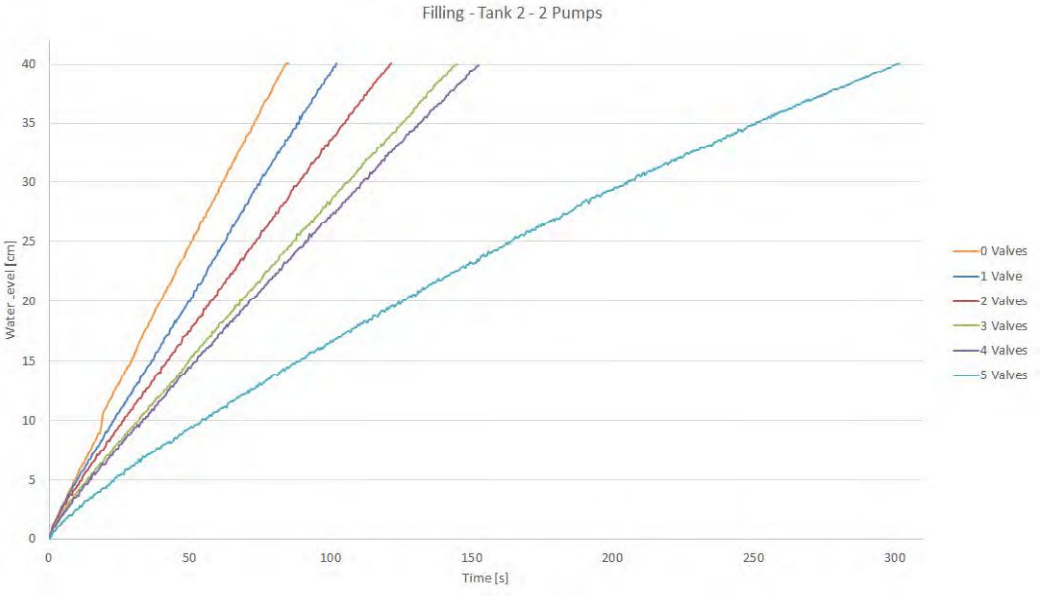

FIGURE 6.9: *Tank 2* filling times with 2 supply pumps on and a different number of opened output valves

In Figure 6.10 the trend of water level in *Tank 2* is shown when supplying water to *Tank 4* by means of V.D.17, with one and two pumps supplying water to *Tank 2*, respectively.

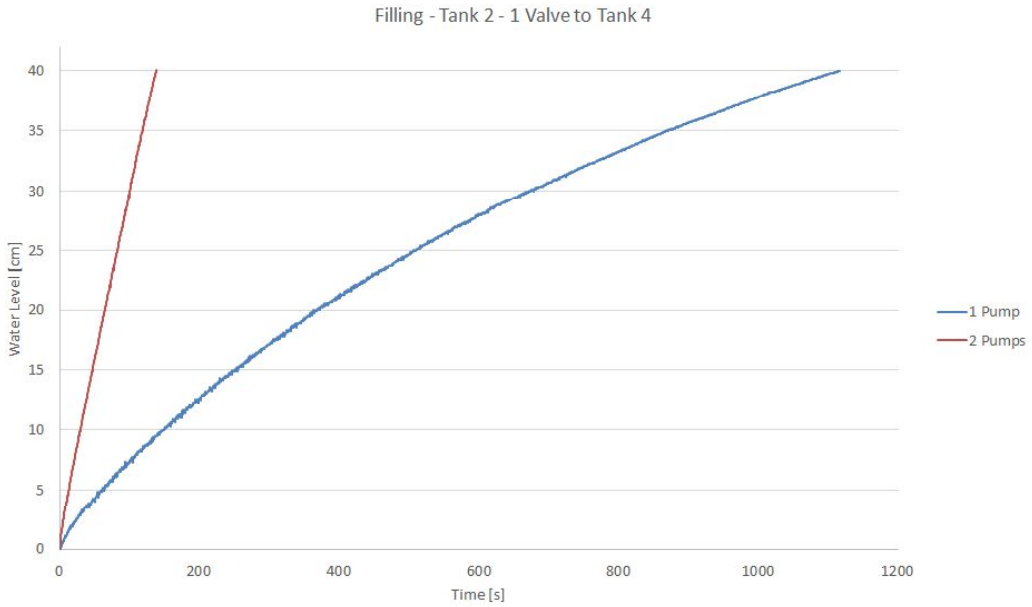

FIGURE 6.10: *Tank 2* filling times with one or two supply pumps on, when valve V.D.17 is open, supplying water to *Tank 4*

In Figure 6.11 and Figure 6.11 the charts show how *Tanks 3* and *4* are filled by means of valves V.D.17 and V.D.18, respectively. The difference with respect to *Tanks 1* and *2* is evident, which are supplied by a pump, rather than with the only effect of gravity. This is reflected by the lower filling speed, hence the longer it takes to provide the same quantity of water to such tanks.

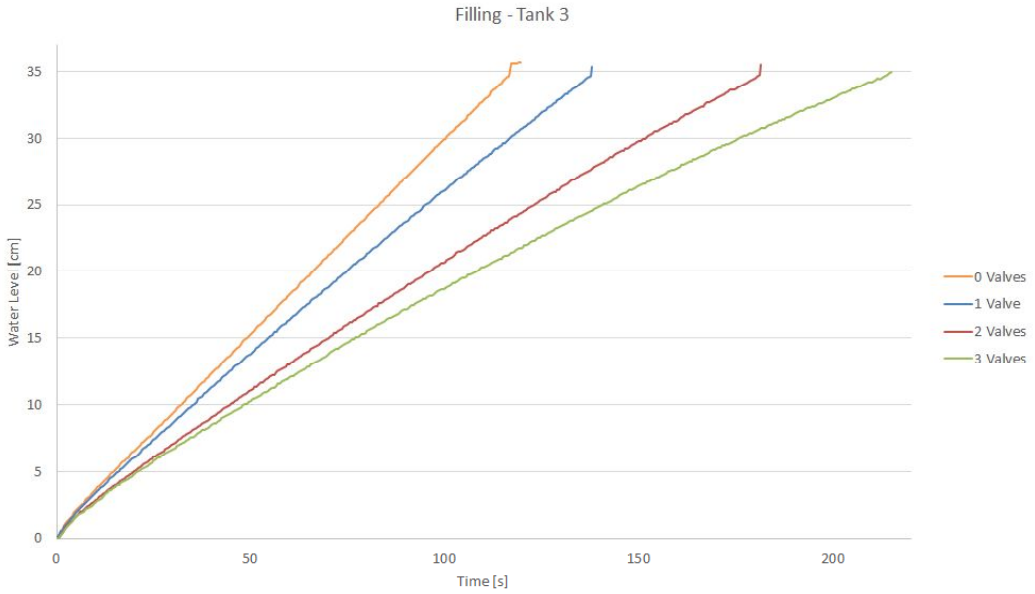

FIGURE 6.11: *Tank 3* filling time, supplied by valve V.D.18

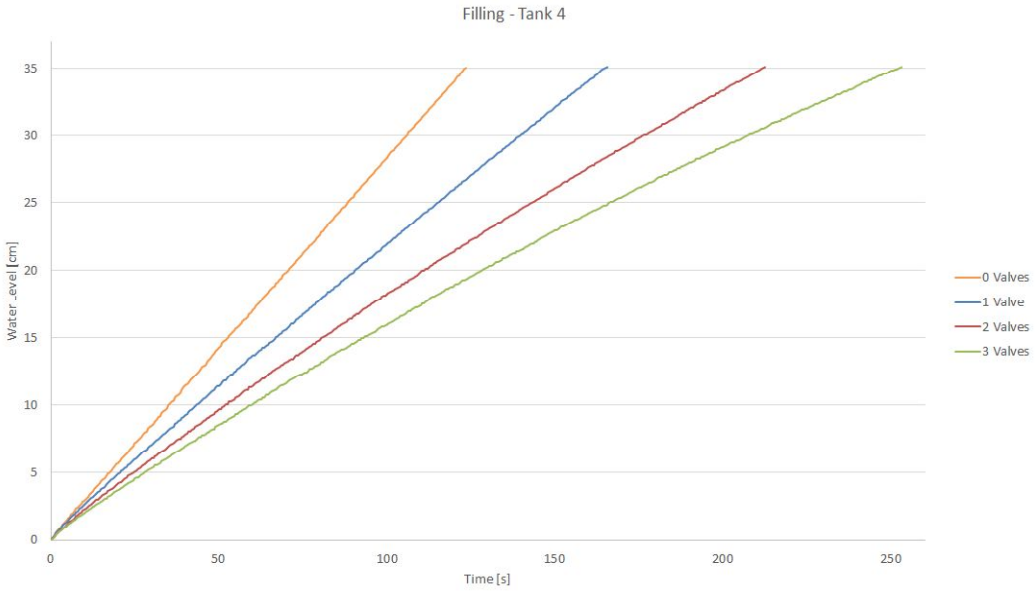

FIGURE 6.12: *Tank 4* filling time, supplied by valve V.D.17

In Figure 6.13 and Figure 6.14 the effect of the cross connections between *Tanks 1* and *2*, and *Tanks 3* and *4*, respectively, is shown. Initially one of the tanks is filled, while the other is empty. When the first one is totally full, the related crossed valve is open. As expected for the communicating vessels principle, the level in the tanks evolves until they reach an equilibrium point, in which the level is almost equal in both tanks. For each pair of tanks both flow directions haven been considered and compared.

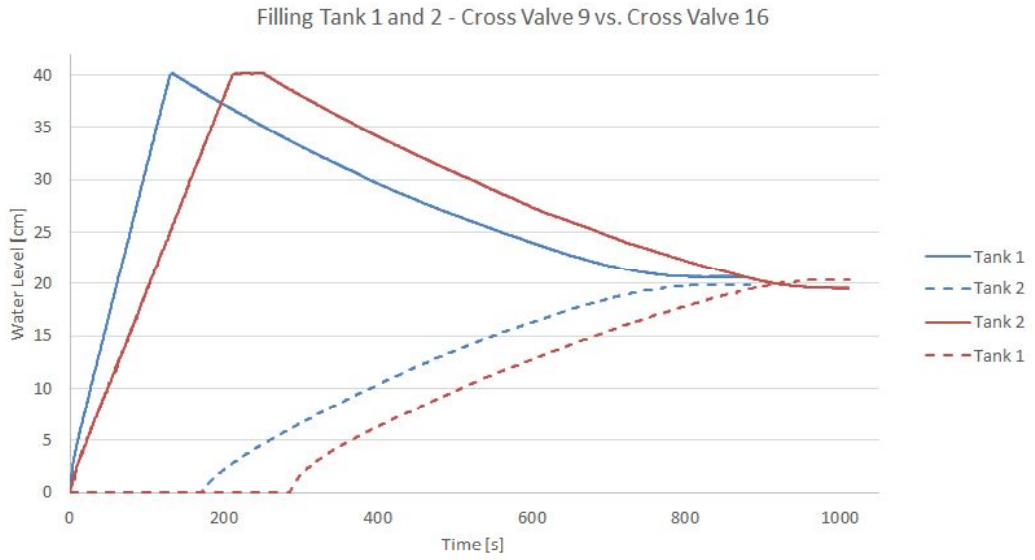

FIGURE 6.13: Effect of the communicating vessels principle between *Tanks 1* and *2*, by deploying the cross-connection valve V.C.16

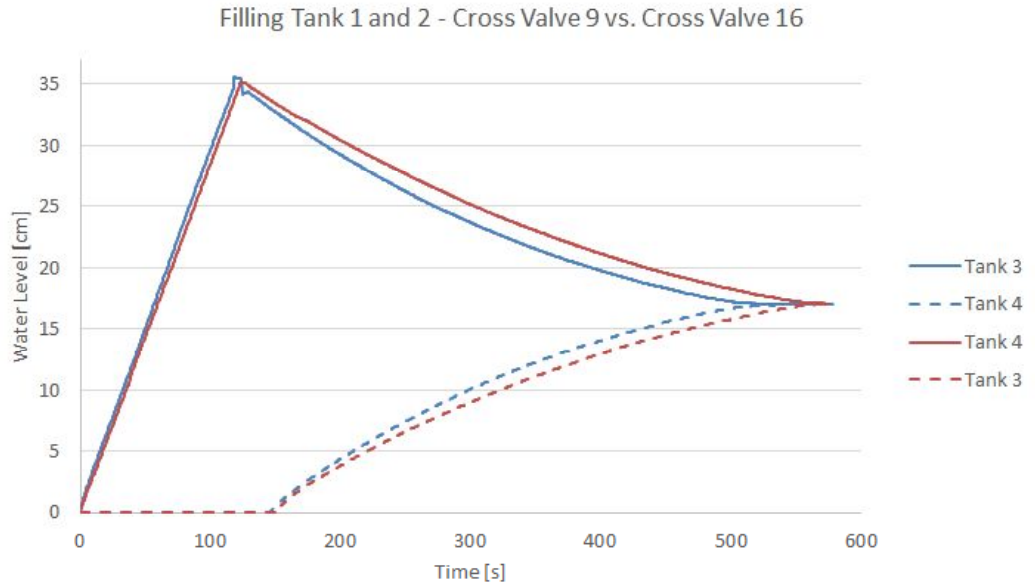

FIGURE 6.14: Effect of the communicating vessels principle between *Tanks 3* and *4*, by deploying the cross-connection valve V.C.9

As previously mentioned, an analysis during the emptying of the tanks has been carried out, as depicted in Figure 6.15 to Figure 6.17. In Figure 6.15 the results for *Tank 2* are shown, obtained by opening one or more valves in the manifold, or the only valve V.D.17 supplying *Tank 4*.

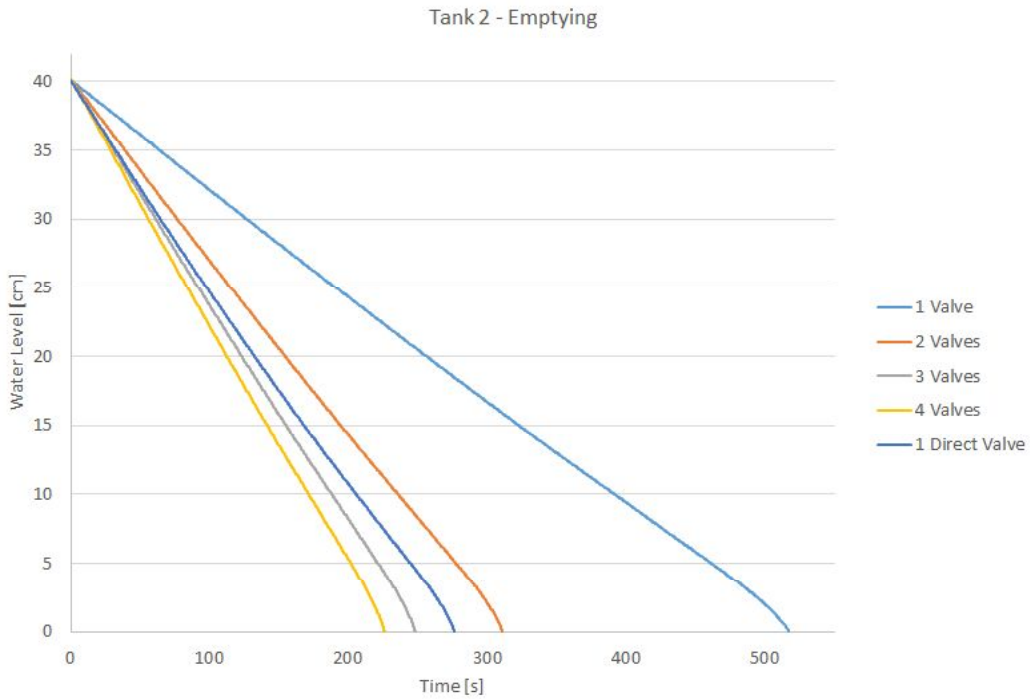

FIGURE 6.15: *Tank 2* emptying times with different open valves

Similarly, Figures 6.16 and 6.17 show the behavior of the water in *Tank 3* and *Tank 5* during emptying, when a different number of valves are open.

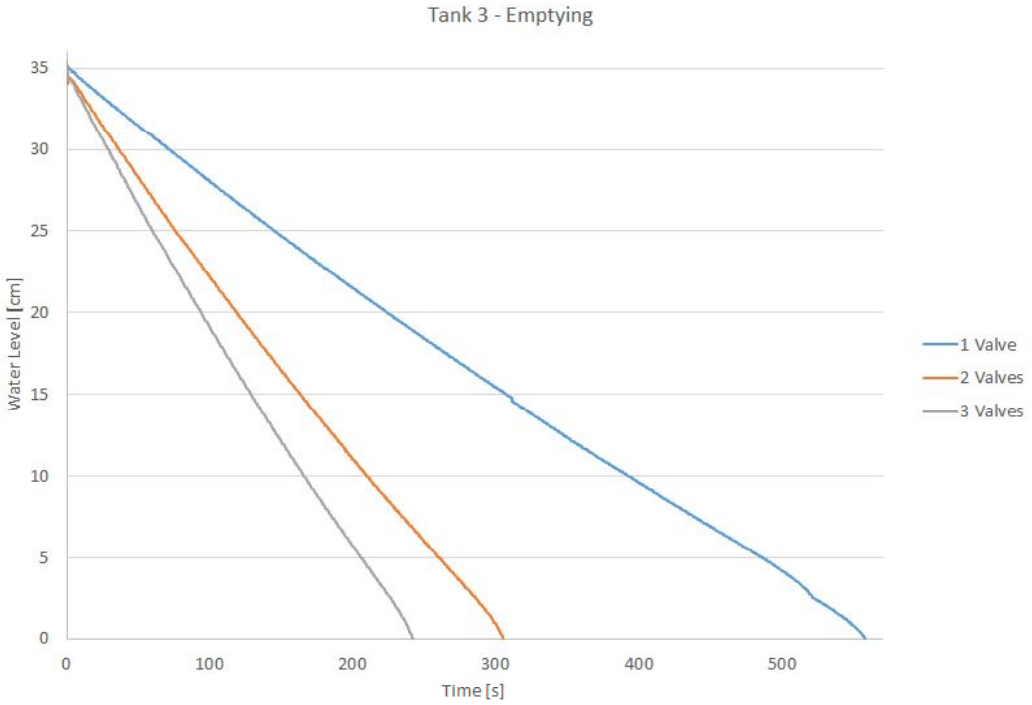

FIGURE 6.16: *Tank 3* emptying times with different open valves

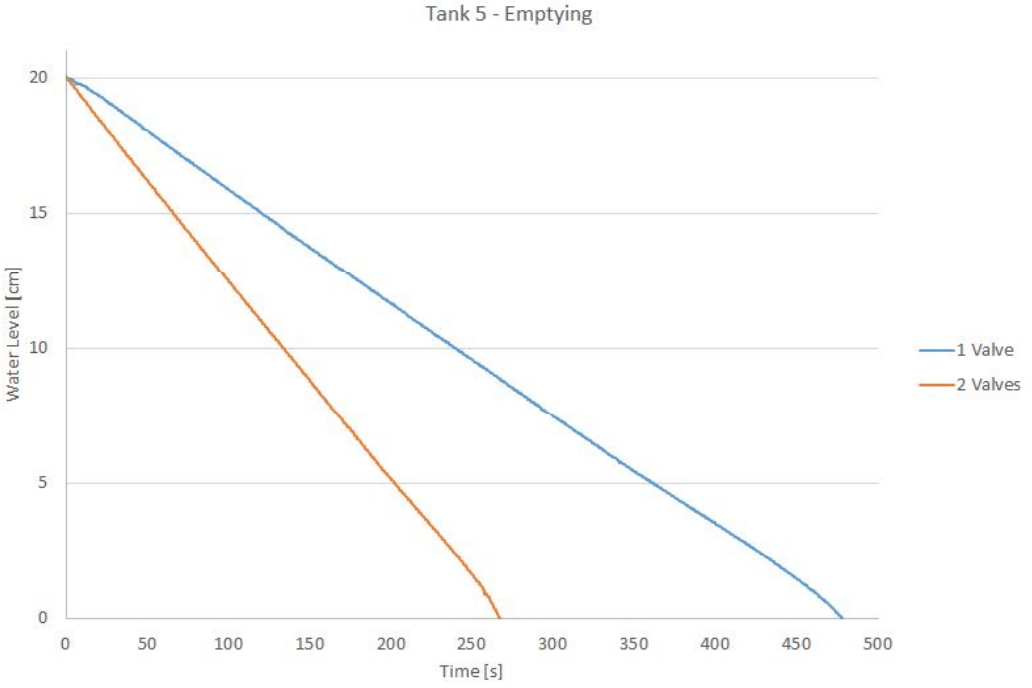

FIGURE 6.17: *Tank 5* emptying times with different open valves

To conclude, the behavior of the system with the presence of leak faults has been studied. In Figure 6.18 to Figure 6.21 the filling of the tanks has been studied when the respective manual valves inducing a leak fault are open to the 100% and 50%. There, it is possible to appreciate how the filling speed is remarkably lower with respect to the healthy case.

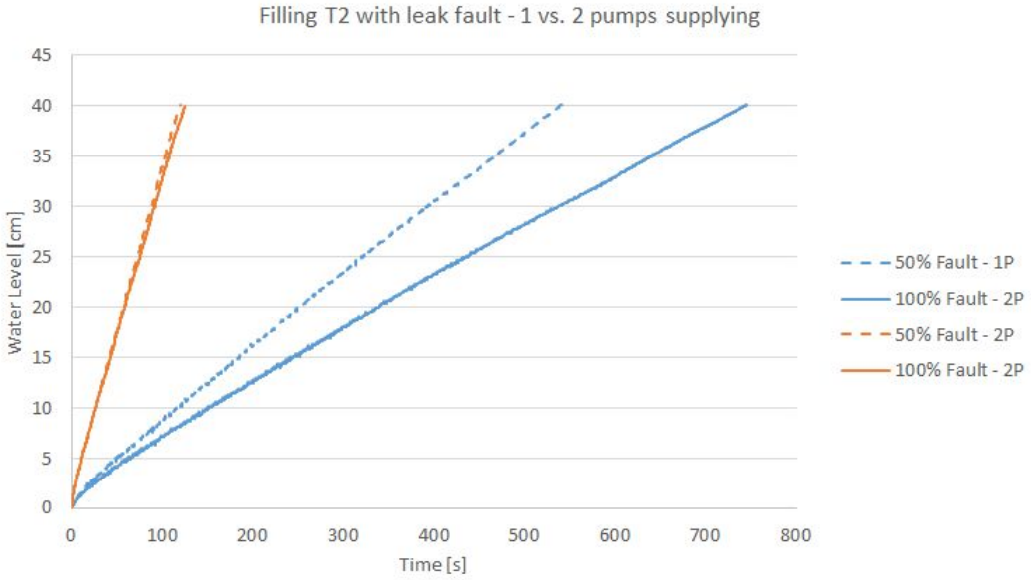

FIGURE 6.18: *Tank 2* filling times with 1 or 2 supply pumps on, with leak fault

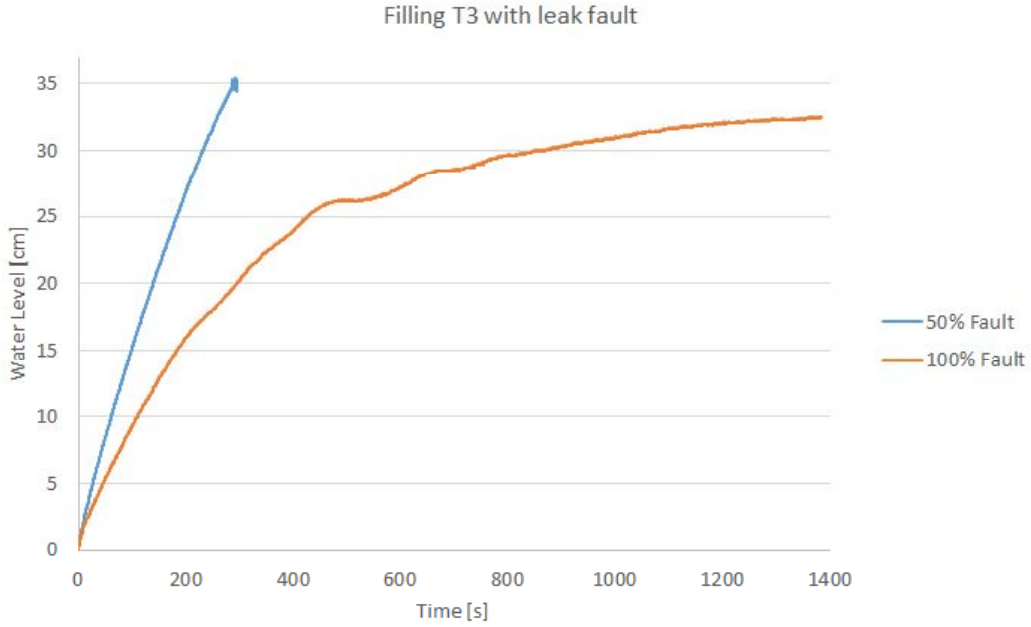

FIGURE 6.19: *Tank 3* filling times with leak fault

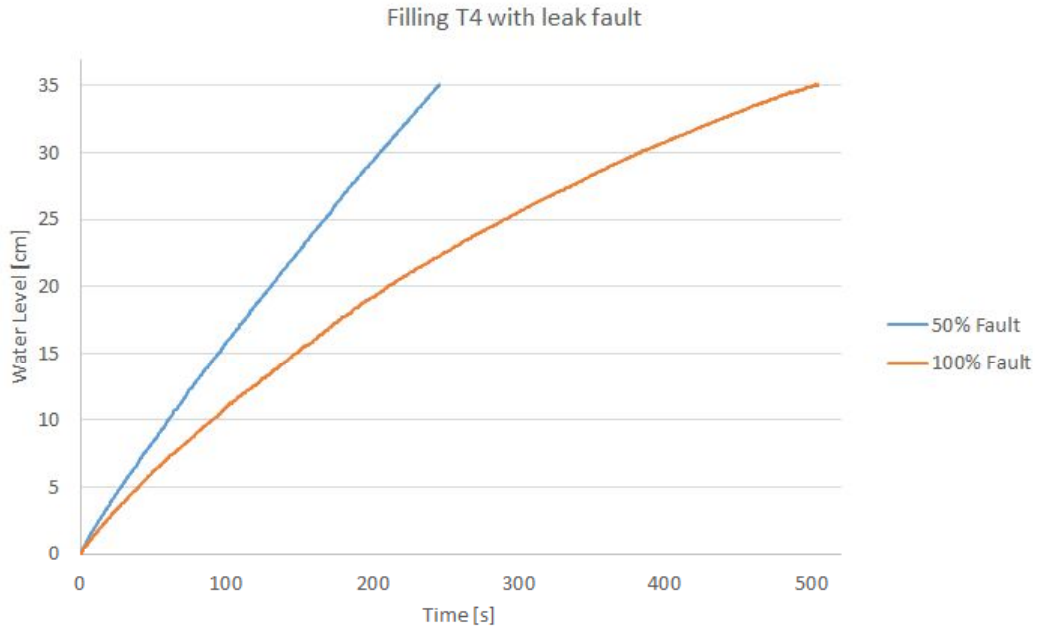

FIGURE 6.20: *Tank 4* filling times with leak fault

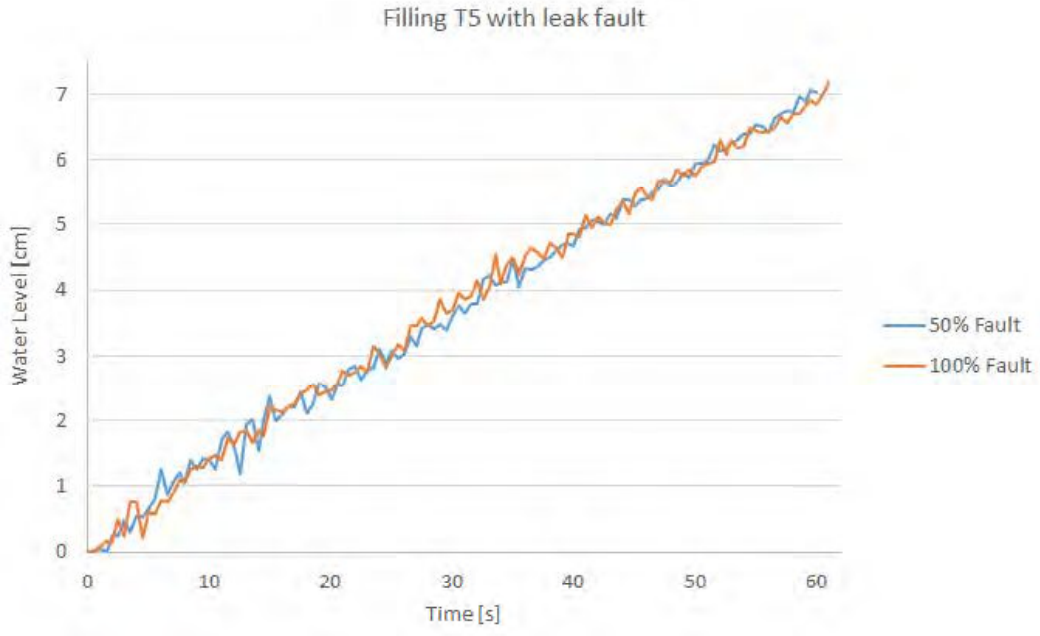

FIGURE 6.21: *Tank 5* filling times with leak fault

In Figure 6.22 to Figure 6.25 a complete cycle of filling and emptying a tank is shown, where the emptying of the tank is due only to the occurrence of the fault, i.e. the complete or half opening of the manual valve connected to the specific tank.

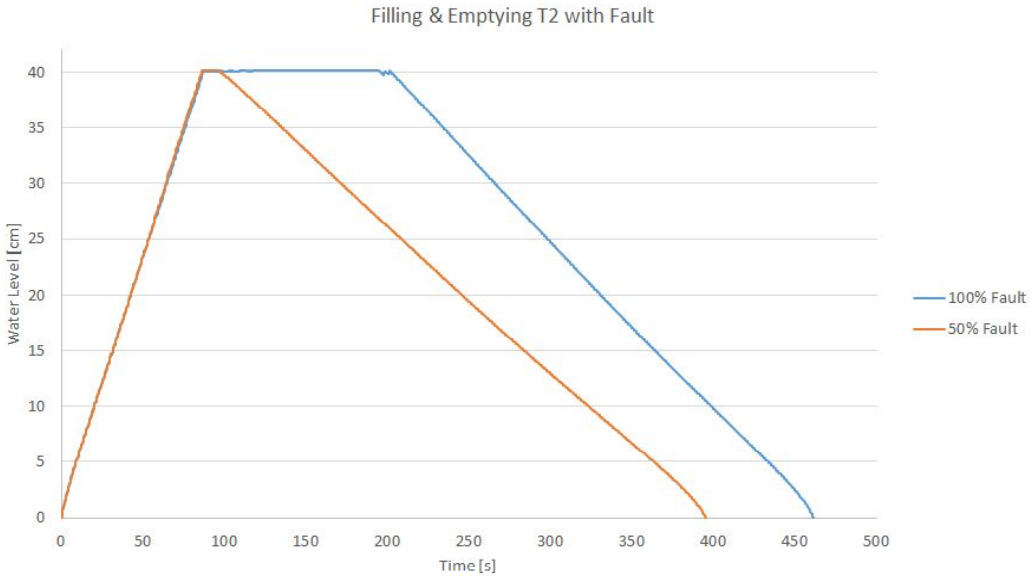

FIGURE 6.22: *Tank 2* filling-emptying cycle times with 1 or 2 supply pumps on, with leak fault

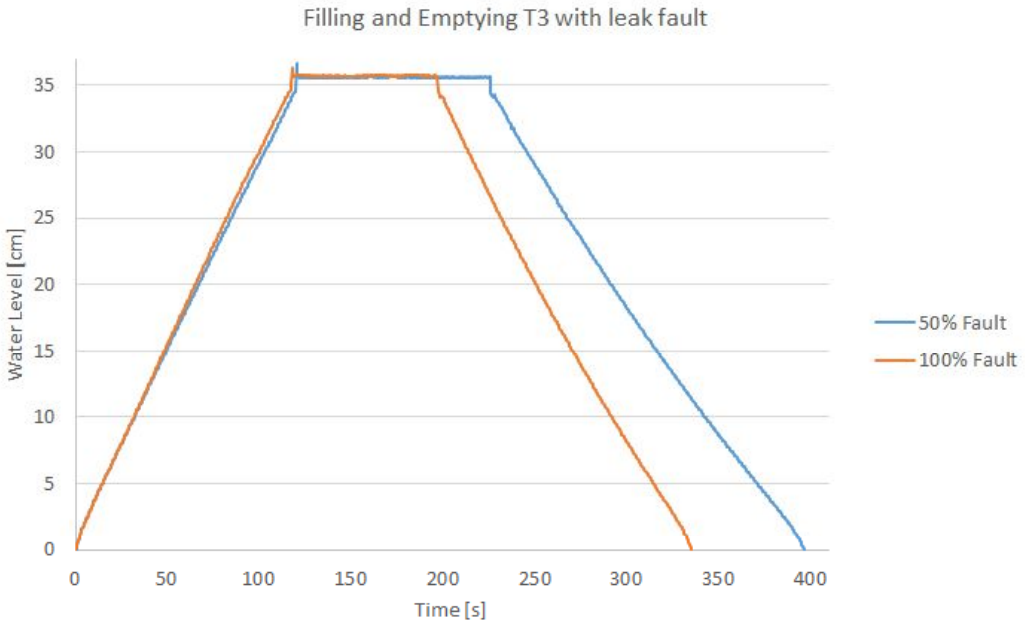

FIGURE 6.23: *Tank 3* filling-emptying cycle times with leak fault

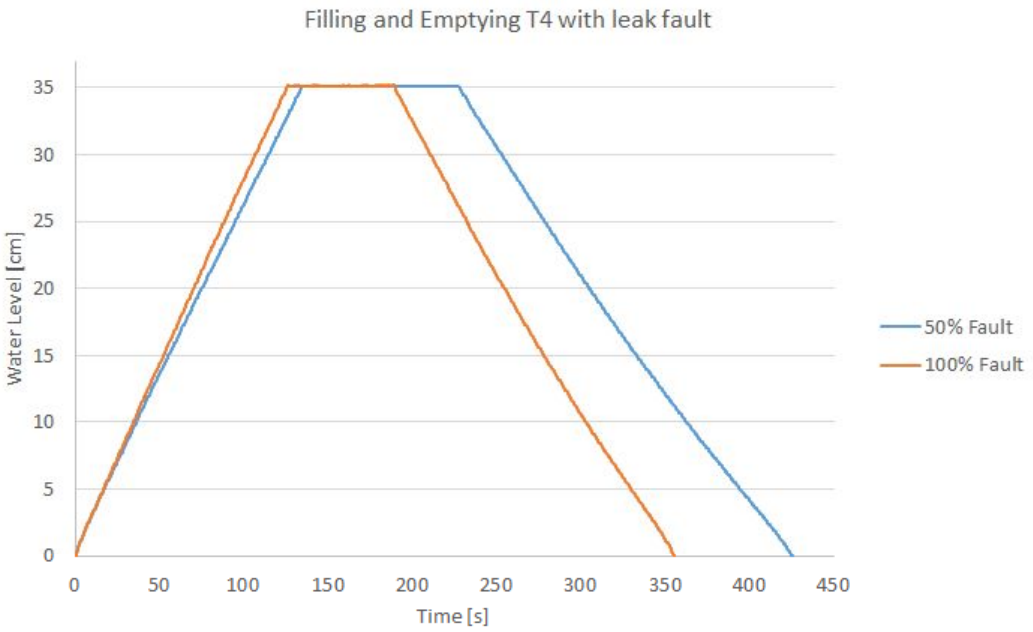

FIGURE 6.24: *Tank 4* filling-emptying cycle times with leak fault

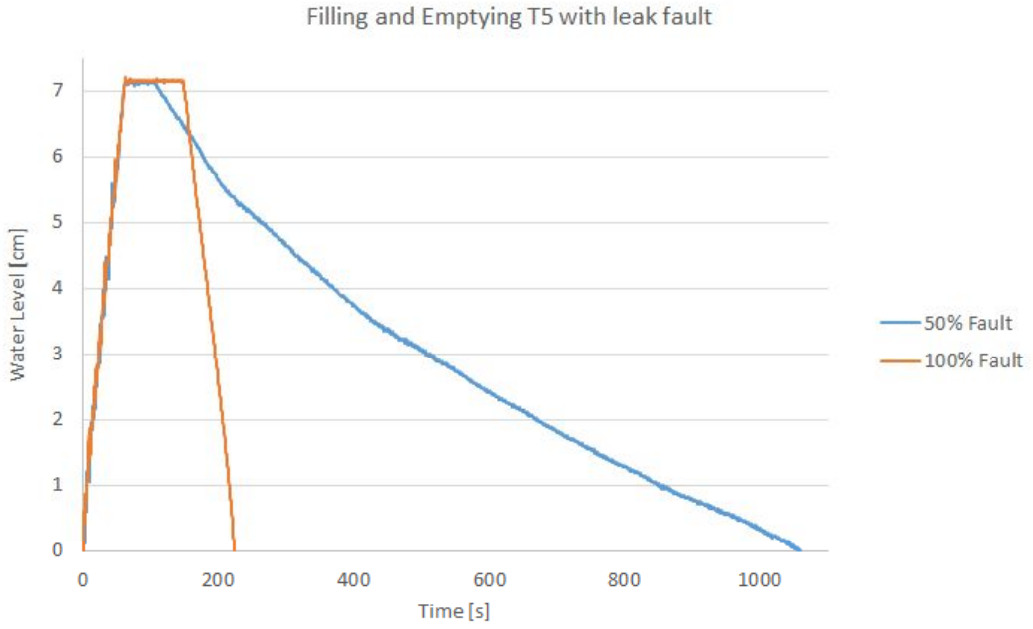

FIGURE 6.25: *Tank 5* filling-emptying cycle times with leak fault

Finally, the emptying of a *Tank 2* with the occurrence of a fault of both the 50% and 100% has been studied, having from one to all the valves in the manifold opened, and the supply valve to *Tank 4*. As expected, the emptying speed is increased due to the presence of the fault.

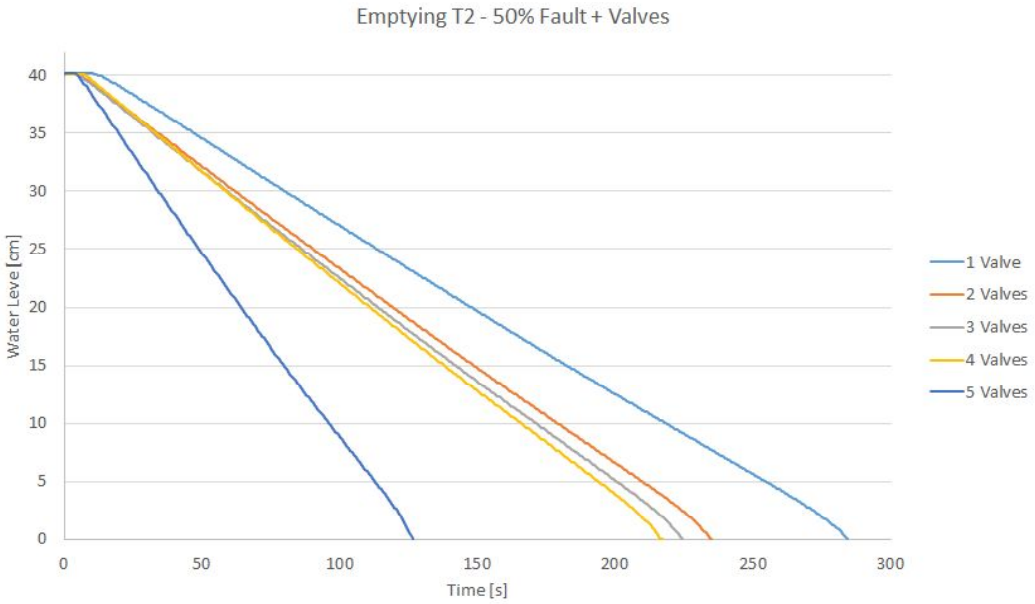

FIGURE 6.26: *Tank 2* emptying times with different open valves, with 50% leak fault

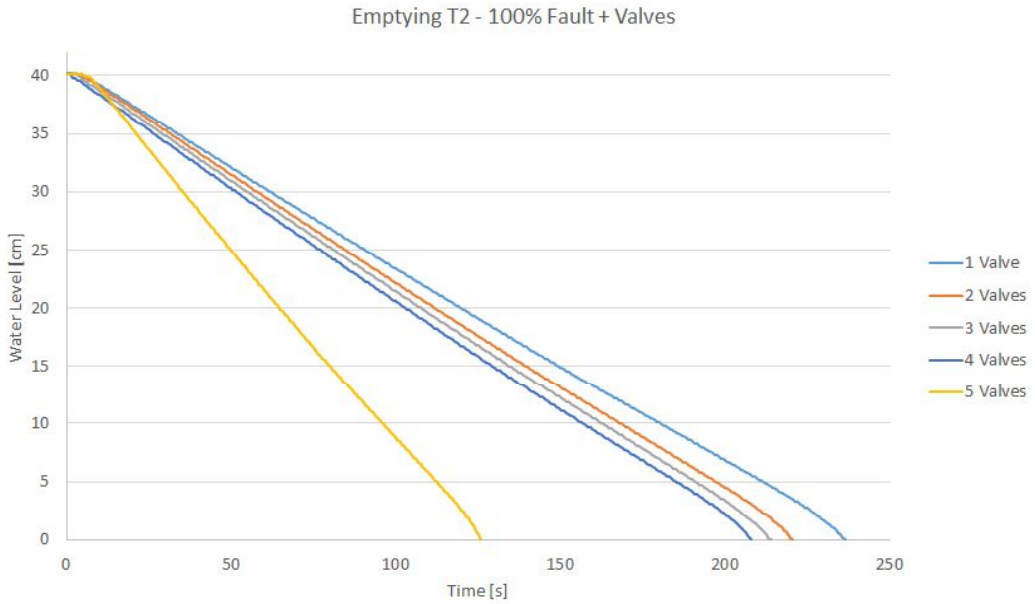

FIGURE 6.27: *Tank 2* emptying times with different open valves, with 100% leak fault

6.2.3 The Control System

In order to monitor and control the water supply and distribution system, a SCADA system has been developed using a commercial framework. All the sensors, pumps and electro-valves in the field are connected to the PLCs, which collect real-time data from the field and perform the control of the testbed. A SCADA and the relative user interface have been built in a dedicated PC, which communicates with the PLC through a Modbus over TCP/IP protocol. For the control system a Modicon M340 PLC (Schneider Electric) has been chosen, with the following modules to perform the different tasks, shown in Figure 6.28

- BMX CPS 2000 - Power supply;
- BMX P34 2020 – Processor and communication module, which supports Modbus serial and Ethernet communication;
- BMX DDO 3202K – Digital I/O to control electro-valves and pumps, with 32 channels;
- BMX AMI 0810 – Analog I/O to read the sensors measurements, with 8 input channels and 16 bits resolution.

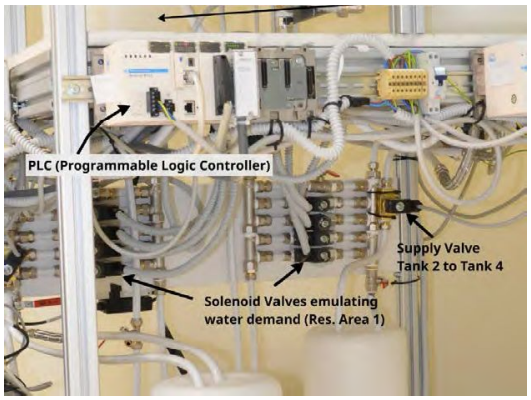

FIGURE 6.28: Control system

Unity Pro XL v7.0 suite (Schneider Electric) has been deployed for the PLC configuration and for the development of the control program. In it, the following tasks have been implemented:

- PLC configuration;
- TCP/IP communication between PLC and PC with the SCADA system;
- Variables creation and addressing;
- Modbus addressing;
- Pumps and electro-valves control;
- Sensors reading and calibration;
- Automatic level control;

- Maximum level control.

All the variables used and their corresponding PLC and Merker addressing is described in Table 6.6, and depicted in Figure 6.29. According to the standard, their structure is the following:

	PLC	**Merker**
Digital outputs	%Qr.s.p	%M#
Analog inputs	%IWr.s.p	%MW#

For the PLC addressing, the letter (Q, I, WI, ...) represents the type of variable that is to be saved in the memory slot (input/output, boolean, word, ...), the first number (r) stands for the rack considered, the second (c) corresponds to the slot where the I/O module is located, and the last (p) indicates the specific pin to which the referred component is connected. Hence, the addressing reflects the physical location for each component/device. For the Merker addressing, the first letter (M, MW) represents, as before, the type of variable that is saved in the register specified by the number (#).

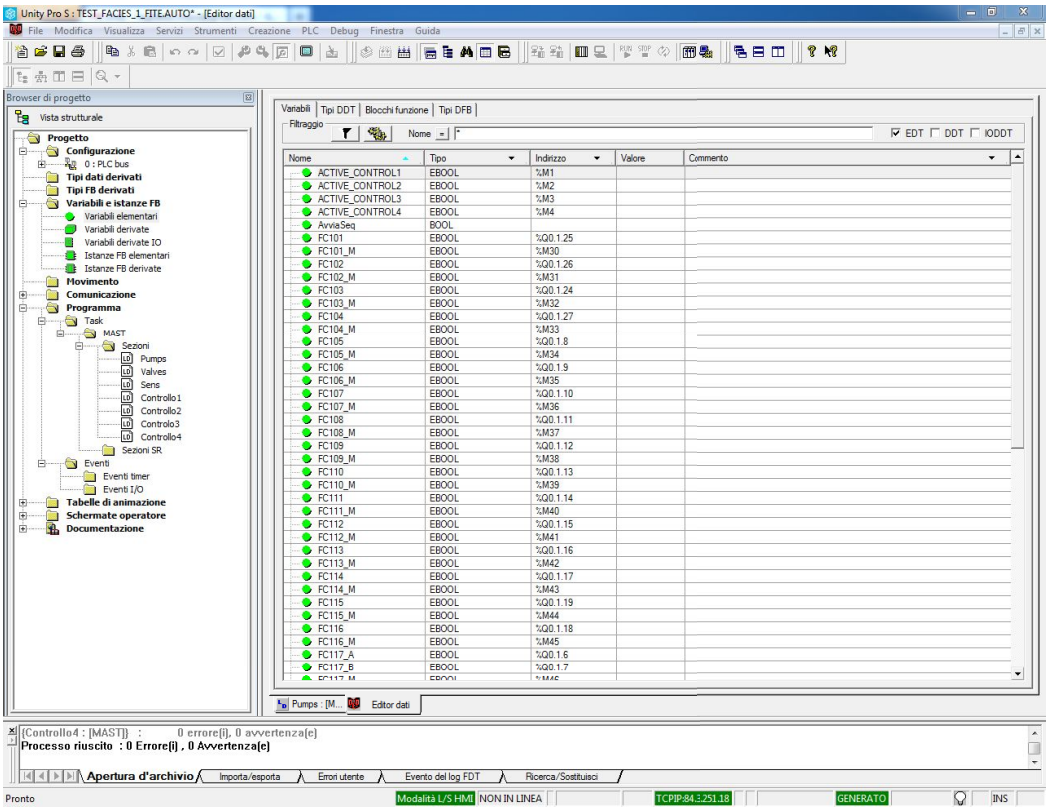

FIGURE 6.29: Implementation of the variables employed

For what concerns the control of pumps and valves, as well as the sensors calibration and measurements, the level control and the maximum level control for the tanks, the code has been developed in Ladder Logic programming language (LD).

Variable	Address			Variable	Address		
	PLC	Merker	Database		PLC	Merker	Database
Valve 1	%Q0.1.25	%M30	000031	Active Control 1	-	%M1	000002
Valve 2	%Q0.1.26	%M31	000032	Active Control 2	-	%M2	000003
Valve 3	%Q0.1.24	%M32	000033	Active Control 3	-	%M3	000004
Valve 4	%Q0.1.27	%M33	000034	Active Control 4	-	%M4	000005
Valve 5	%Q0.1.8	%M34	000035	Active Control 5	-	%M5	000006
Valve 6	%Q0.1.9	%M35	000036	Set Point 1	-	%MW10	400011
Valve 7	%Q0.1.10	%M36	000037	Set Point 2	-	%MW11	400012
Valve 8	%Q0.1.11	%M37	000038	Set Point 3	-	%MW12	400013
Valve 9	%Q0.1.12	%M38	000039	Set Point 4	-	%MW13	400014
Valve 10	%Q0.1.13	%M39	000040	Set Point 5	-	%MW14	400015
Valve 11	%Q0.1.14	%M40	000041	Max Level 1	-	%M6	000007
Valve 12	%Q0.1.15	%M41	000042	Max Level 2	-	%M7	000008
Valve 13	%Q0.1.16	%M42	000043	Max Level 3	-	%M8	000009
Valve 14	%Q0.1.17	%M43	000044	Max Level 4	-	%M9	0000010
Valve 15	%Q0.1.19	%M45	000046	Max Level 5	-	%M10	0000011
Valve 16	%Q0.1.18	%M44	000045	Set Max Level 1	-	%MW20	-
Valve 17	%Q0.1.6 %Q0.1.7	%M46	000047	Set Max Level 2	-	%MW23	-
Valve 18	%Q0.1.30 %Q0.1.31	%M47	000048	Set Max Level 3	-	%MW26	-
Valve 19	%Q0.1.28	%M48	000049	Set Max Level 4	-	%MW29	-
Valve 20	%Q0.1.29	%M49	000050	Set Max Level 5	-	%MW32	-
Sensor 1	%IW0.2.1	%MW1	300002	Sensor 1 Hysteresis	-	%MW15	-
Sensor 2	%IW0.2.2	%MW2	300003	Sensor 2 Hysteresis	-	%MW16	-
Sensor 3	%IW0.2.4	%MW3	300004	Sensor 3 Hysteresis	-	%MW17	-
Sensor 4	%IW0.2.3	%MW4	300005	Sensor 4 Hysteresis	-	%MW18	-
Sensor 5	%IW0.2.5	%MW5	300006	Sensor 5 Hysteresis	-	%MW19	-
Pump 1	%Q0.1.0 %Q0.1.1	%M20	000021	Max Sensor 1 Hysteresis	-	%MW21	-
Pump 2	%Q0.1.2 %Q0.1.3	%M21	000022	Max Sensor 2 Hysteresis	-	%MW24	-
Pump 3	%Q0.1.4 %Q0.1.5	%M22	000023	Max Sensor 3 Hysteresis	-	%MW27	-
Pump 4	%Q0.1.21 %Q0.1.22	%M23	000024	Max Sensor 4 Hysteresis	-	%MW30	-
				Max Sensor 5 Hysteresis	-	%MW33	-

TABLE 6.6: Variables addressing

Level Control and Overflow Prevention

The level control and the maximum level control have been implemented on the "low level" for safety reasons. Indeed, even when the PC on which the SCADA/HMI is installed does not respond or the communication is interrupted for any reason, these two controls are still performed by the controller, guaranteeing that no overflows can have place in normal operating conditions. In addition, a hysteresis has been configured, as depicted in Figure 6.30, so as to avoid the intermittent opening and closing of the pumps/valves when the desired level has been reached. Hence, a lower and upper bound are set in a neighborhood of the the setpoint. During filling, when the water level in a tank overcomes the setpoint and reaches the upper threshold, the relative pump/valve is turned off until the level reaches the lower bound. This allows measurement fluctuations near the setpoint, avoiding high-frequency jitters in the control signal, i.e. high-frequency switching of the actuator. An analogous logic has been implemented for the emptying towards the setpoint. The maximum level threshold is represented by a fixed value, directly configured in a PLC memory register, and suitably chosen for each tank by considering its maximum capacity.

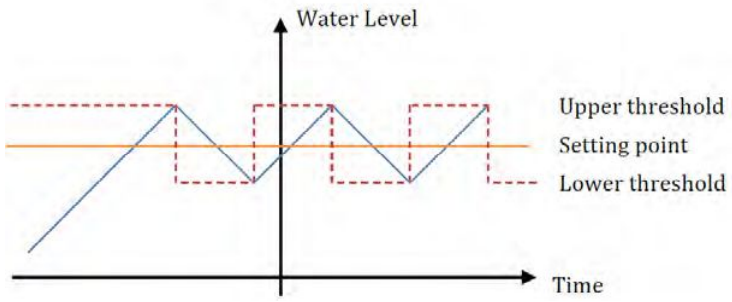

FIGURE 6.30: Water level setpoint hysteresis

The operator has then the possibility to choose if to manually control the water level in the tanks by manhandling the relative pumps and/or valves by means of the HMI, or employing the automatic level control. For the latter, the setpoint to be reached and/or maintained can be manually chosen on the interface, which is then copied to the corresponding Modbus variable to perform such control.

The Human-Machine Interface (HMI)

The SCADA/HMI has been developed using iFIX v5.0 (General Electric) on a Windows XP system, which is shown in Figure 6.31. It has been developed deploying the *Dynamos* available on IFIX, where each element has been logically connected to the relative variable on the database. The design of the interface has been chosen in order to reflect the actual physical configuration of the testbed. Hence, the five tanks and the reservoir are connected by a piping system, along which also the pumps and electric valves are located. Some of the latter have been divided into manifolds as made in the actual testbed.

Moreover, an animation has been added for the pumps and valves, related to their ON/OFF state, and for the tanks, which show their water level according to the measurements from the sensors, and their value in tenths of millimeters, e.g. $1000 = 10cm$. By employing the IFIX Alarm Configurator, for each tank in the interface a low, high and very high water level alarms have been configured, making so the colour of the

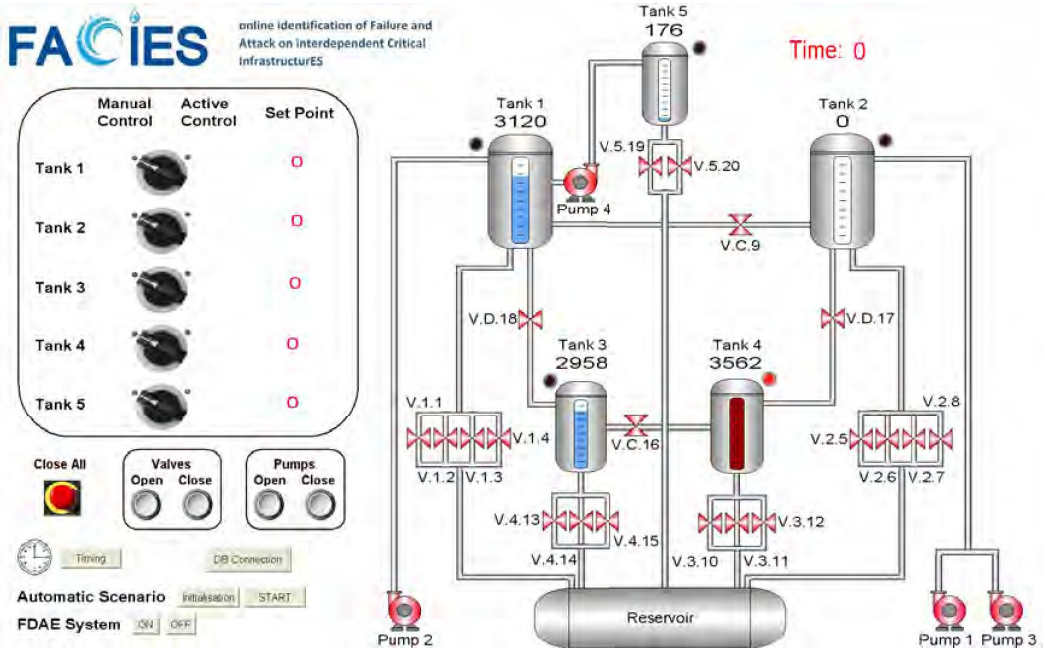

FIGURE 6.31: SCADA/HMI Interface

"water" in the animated tank is blue in the intermediate range, becomes orange when the low or high threshold is reached, and flashes between red and claret for the very high threshold, as depicted in Figure 6.32. The thresholds have been configured based on the specific tank's capacity. In addition, a maximum level indicator has been linked to the tanks, in order to inform the operator that a particular tank is completely full. In such a case, the relative pump/valve that supplies water to the tank is disabled in order to avoid overflows.

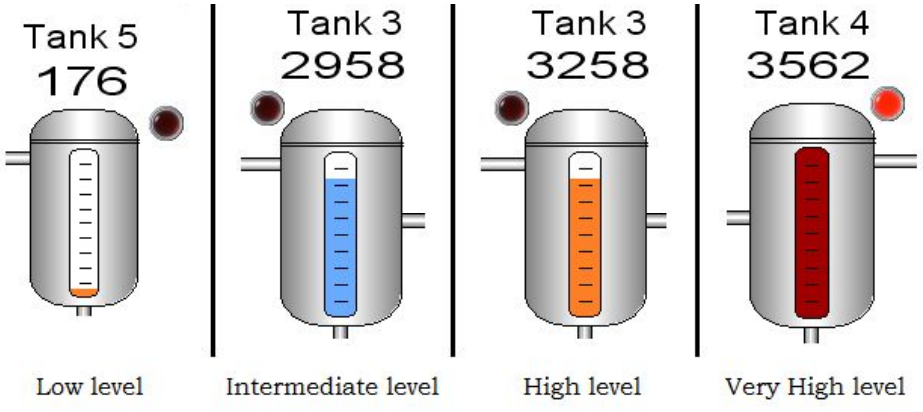

FIGURE 6.32: Low, high and very high alarms examples indicating extreme water level conditions in the tanks.

To this end, the communication between the PC and the PLC has been created, opportunely configuring the network card through which the communication takes place and the addresses for the different variables that have to be transferred. For each group of components and variables, a memory block has been created based on the number and type of variables employed, properly configured to be matched to

the related registers of the PLC. Afterwards, the database has been created, providing to each variable an opportune tag and indicating the type of variable (digital/analog, input/output) and its specific PLC register address (which are Merker addresses), as described in Table 6.6.

Operation Modes

Four different operation modes have been implemented on the operator interface:

- Manual operation;
- Automatic level control;
- Temporized control;
- Daily scenario.

In the *manual control*, the operator is allowed to individually manhandle the different components present on the interface, turning ON/OFF he pumps or valves by just clicking on the corresponding element of the interface. For the *automatic level control*, the desired water level to be reached in a particular tank can be manually set (in tenths of millimeters) in the panel shown in Figure 6.33, considering the capacity of each tank, and the relative pumps/valves are activated or deactivated in order to maintain such level, despite the state of the other components connected to the tank.

FIGURE 6.33: Automatic level control panel

The *temporized control* can be configured on an additional interface made available as a pop-up window, where the operator can set the filling and emptying time intervals desired to perform a *fill-wait-empty-wait* cycle in a chosen tank. It has been implemented exploiting the possibility to directly add Visual Basic (Microsoft) code to the elements in the interface. Hence, the user has the possibility to choose, through the opportune interface shown in Figure 6.34, a pattern or sequence of filling and emptying for one or more tanks. Such interface presents two modes for the pattern configuration. In the *simple mode*, the tank/s can be chosen, and the filling, waiting and emptying times (in seconds) can be set. The sequence followed is filling-waiting-emptying-waiting, and during the task execution, all the pumps and/or valves available for the particular operation are used. On the other side, in the *advanced mode* the specific valves to be used during the emptying phase can be chosen.

An opportune error message is displayed if no tank or component is chosen, or if no timing value has been set for the pattern execution. During the automatic operation of the sequence, the manipulation of the components on the interface is disabled.

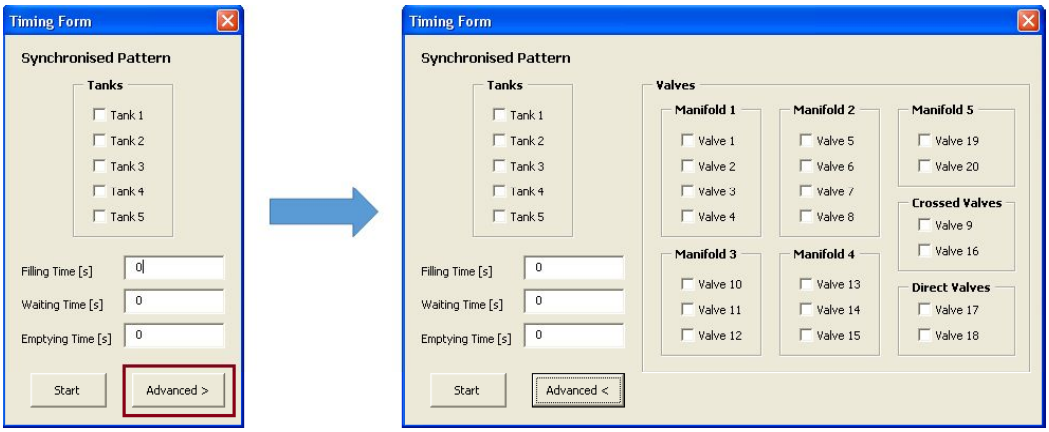

FIGURE 6.34: Temporized control interfaces

Finally, an *automated scenario* has been implemented as described in Section 6.1.2, which aims at reproducing a typical water demand of a small city in 24 hours, scaled down to a 6 minutes emulation. As for the temporized control, it has been developed in Visual Basic, and during its operation the manual operation on the interface is disabled, while the time progression is displayed on the interface. In addition, for the daily scenario, an automatic initialization of the tanks has been implemented. The desired levels can be manually set inside the developed Visual Basic code. It has been configured so as to bring the water level in all the tanks to the specified values. As for the previous case, during the operation, the commands in the interface cannot be manually modified.

To conclude, some additional buttons have been added to the interface, which allow to contemporary open or close all the valves or pumps, as well as an emergency button which turns OFF all the actuators when pressed.

6.2.4 The Daily Scenario

The scenario considered has been a typical daily water demand curve of a small city, opportunely scale down to a 6 minutes interval, as depicted in Figure 6.2. The water level in all the tanks is first initialized to specific pre-set values shown in Table 6.7, and the demand curve is obtained by the sequential opening and closing of the valves disposed in the manifolds, as indicated in Table 6.8.

Tank	Initial Level
Tank 1	$0.2m$
Tank 2	$0.18m$
Tank 3	$0.225m$
Tank 4	$0.225m$
Tank 5	$0.2m$

TABLE 6.7: Initial water level for the different tanks

Time	**Valves State**															
	V1	V2	V3	V4	V5	V6	V7	V8	V10	V11	V12	V13	V14	V15	V19	V20
2s																
20s																
34s																
46s																
66s																
90s																
106s																
124s																
136s																
148s																
156s																
168s																
176s																
188s																
196s																
214s																
230s																
244s																
266s																
280s																
296s																
318s																
330s																
342s																
354s																
360s																

TABLE 6.8: Open (green)/close (red) valves sequence

To avoid any water shortages or the total emptying of the tanks, rules for the activation of pumps, supply valves and cross-connection valves have been configured so as to refill the tanks when a low water level is reached. Moreover, these rules, detailed in Table 6.9, aim to prevent overflows in the tanks.

Component	Start Level	Stop Level
Pump 2	$x_1 \leq 0.15m$	$x_1 \geq 0.39m$
Pump 1	$x_2 \leq 0.15m$	$x_2 \geq 0.35m$
Pump 4	$x_5 \leq 0.02m$	$x_5 \geq 0.19m$
V.D.17	$x_4 \leq 0.02m$	$x_4 \geq 0.34m$
V.D.18	$x_3 \leq 0.02m$	$x_3 \geq 0.34m$
	$\lvert x_1 - x_2 \rvert \geq 0.3m$	
V.C.9	$x_1 > 0.15m$ and $x_2 < 0.05m$	$\lvert x_1 - x_2 \rvert \leq 0.05m$
	$x_2 > 0.15m$ and $x_1 < 0.05m$	
	$\lvert x_3 - x_4 \rvert \geq 0.3m$	
V.C.16	$x_3 > 0.2m$ and $x_4 < 0.05m$	$\lvert x_3 - x_4 \rvert \leq 0.05m$
	$x_4 > 0.2m$ and $x_3 < 0.05m$	

TABLE 6.9: Pumps, supply valves and cross-connection valves control during scenario to avoid shortages

Such sequence has been implemented as a Visual Basic subroutine that can be run from the HMI. During the whole run, the manual control of the interface is disabled.

6.2.5 Monitoring Modules Architecture

For this system, a modular architecture for the detection and identification of cyber-physical anomalies has been developed, which is also able to manage interdependencies with other infrastructures and with the telecommunications network. As depicted in Figure 6.35, it is composed by four main modules, namely:

- **Fault Detection and Approximation Estimator (FDAE):** aims to detect and identify the onset of physical faults and/or attacks. In order to detect anomalous behaviors due to the presence of physical faulty events, this system deploys a model of the plant and is supplied with data collected in the SCADA database. Thereby, it allows to detect and react when the behavior moves from the expected. In such case, the information about occurring or incipient faults in the system is transferred to the RP and the ES.

- **Intrusion Detection System (IDS):** monitors the whole traffic flowing in the network with the aim of detecting malicious cyber activities. Hence, the packets are analyzed to prevent and intervene in case of suspicious or malicious data, generating reports for both the RP and the ES.

- **Risk Predictor (RP):** manages the effects of dependencies and interdependencies with other CIs. It implements a MHR model [115] of the specific scenario (HighLake City in this case), with its different functions and dependencies (physical, geographic, logical and cyber), and exploits the solution proposed in [116] to achieve a distributed estimation of the operative level of the different infrastructures. Such approach allows to limit the quantity of information exchanged between CIs' operators, which are generally very reluctant to share detailed data,

still being able to managefseveral kinds of situations, ranging from cascading effects to phenomena as strikes, etc. More specifically, all the alarms generated by the SCADA, the events from the IDS and the predictions and potential faults detected by the FDAE are sent to the RP, which combines such information with the interdependencies of the system at different levels, implementing a simulation of the entire scenario, and generates a status of the consequences on the components and cascading effects. Hence, it is able to provide early information on the consequences of domino effects in the whole system, and the operators are able to focus their actions to mitigate the future effects in the network, performing the appropriate countermeasures, and to enable the necessary restoring procedures.

- **Expert System (ES):** integrates information from the various modules, performing a cross-validation of the information, and provides a dynamic holistic risk assessment for the actual and near future situation. This information, is transferred to the SCADA operator and shown in a dedicated control panel, mainly exposed as highlighted events and alarms.

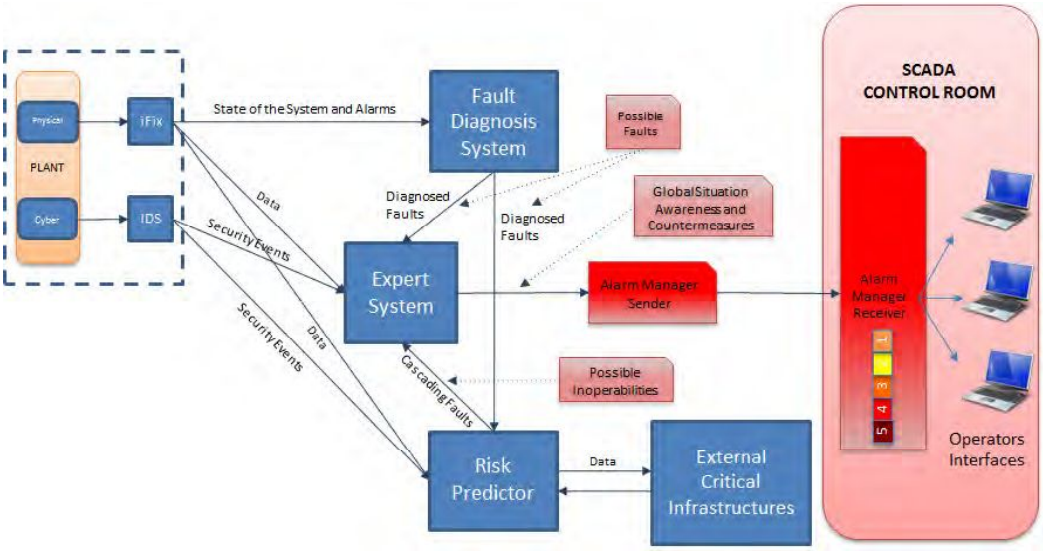

FIGURE 6.35: FACIES architecture

6.2.6 Main Database

During the operation, all the data related to the state of the different components are saved into a database, developed in MySQL Workbench (Oracle). This has been possible by developing a specific Visual Basic program for the communication between Modbus and MySQL, which periodically (every $500ms$) queries the PLC to obtain the data to be saved. The interface for communication can be accessed from the relative button on the operator interface, which leads to the opening of the pop-up window shown in Figure 6.36. In it, the state of the communication and of the connection between Modbus and database is indicated, as well as the current state of the different components.

The MySQL database has been configured as a log, as shown in Figure 6.37, containing all the state variations of the system in the time interval in which the communication between Modbus and database is established. The data recording can be started by

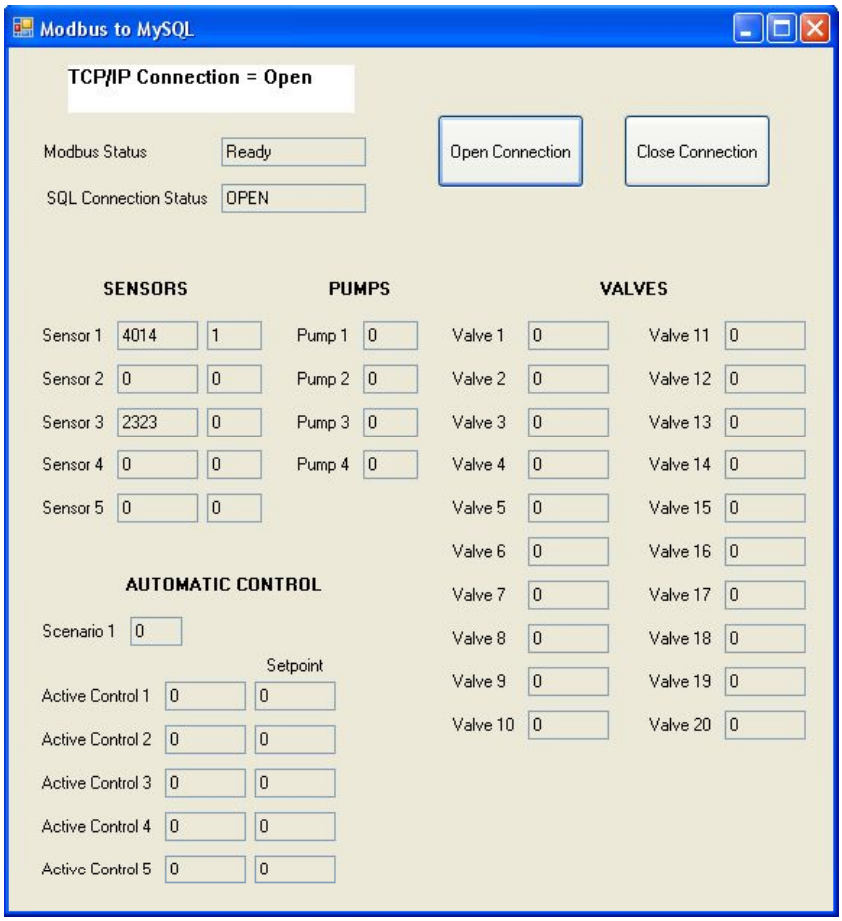

FIGURE 6.36: Modbus-MySQL Database connection

pressing the *Open Connection Modbus/DB* button, and stopped by the *Close DB Connection* button. During the communication, the data transferred on Modbus will be saved on the database each time there is a variation on the state of a variable, i.e. in an event driven fashion, by adding a record in the table, indicating the id of the event (which increases automatically), the name of the specific component, the value assumed and the timestamp in the *YYYY-MM-DD hh-mm-ss ms* format. Then, the database can be interrogated by the use of opportune queries, based on which information is necessary to be extracted, in which case it acts as an *Historian* for the system.

6.2.7 The Local Network

The data transmission between the PLCs, SCADA, and database takes place using the Modbus over TCP protocol, while the remaining modules communicate through a classic TCP/IP protocol on a dedicated standalone local network. On it, the PLCs sends to the SCADA the data gathered from the field regarding the state of the various components and the commands from the SCADA to the actuators. The physical connection between PLCs, SCADA/HMI and the other modules of the architecture (Risk Predictor, IDS, Fault Detection system and Expert System) has been created by means of two TP-LINK TL-SG108E switches, as sketched in Figure 6.38.

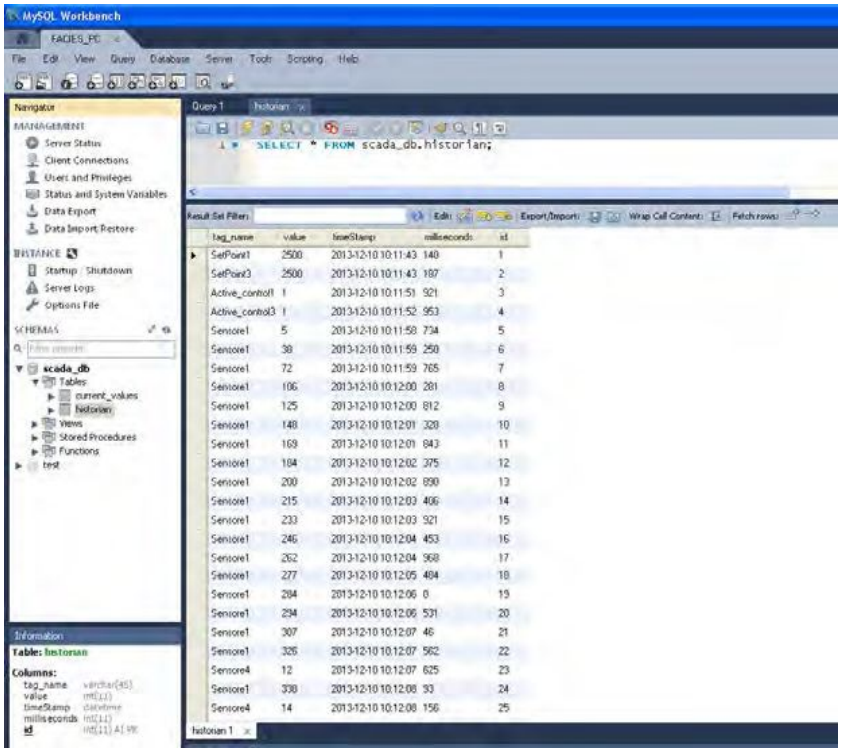

FIGURE 6.37: MySQL Testbed Database

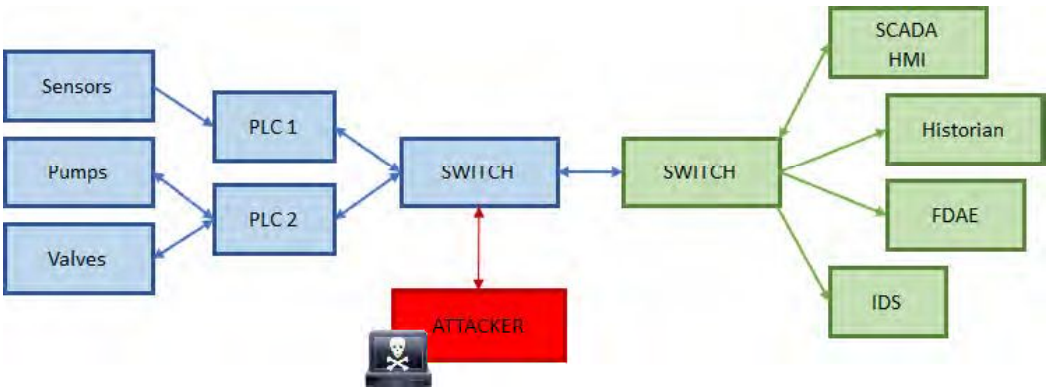

FIGURE 6.38: Local *ad-hoc* network schema

Thus, an *ad-hoc* local network has been created, for which a static IP addressing has been deployed, as described in Table 6.10.

The Modbus Protocol

As largely analyzed in [61], several highly specialized communication protocols for industrial automation and control have been developed in the last decade, most of which have been designed to cope with the economic and operational requirements of large distributed control systems, focusing mainly on efficiency, reliability and real-time operation to support precision operations. Thereby, for the sake of efficiency, any feature or function considered not absolutely necessary has not been foreseen, including often

Device	IP Address
Network	84.3.251.0
Default Gateway	84.3.251.10
Subnet Mask	255.255.128.0
PLC 1	84.3.251.18
PLC 2	84.3.251.19
SCADA/HMI	84.3.251.12
FDAE	84.3.251.13
IDS	84.3.251.15
Risk Predictor	84.3.251.11

TABLE 6.10: IP Addressing of the testbed network

security.

Modbus is the oldest and most widely deployed communication protocol for industrial applications. It is an application layer messaging protocol, developed by Modicon, which became the *de facto* standard since its publication in 1979 [117]. It provides a Client/Server communication between devices connected on different types of networks, with a request/reply setting and offers various services specified by 127 function codes. Generally, this protocol is used for the communication between Remote Terminal Units (RTUs) and supervisory computers in SCADA systems. It can be implemented by asynchronous serial transmission (serial Modbus) or using TCP/IP over Ethernet (Modbus/TCP) and allows an easy communication within all types of network architectures (e.g. the Internet) [118].

The Modbus/TCP protocol is implemented as a Ethernet connection: to this end the TCP frame contains a Modbus packet as payload, as depicted in Figure 6.39. The Modbus payload is divided into two fields: a dedicated header, called Modbus Application Protocol (MBAP), to identify the Application Data Unit (ADU) and map it on the specific network, and a Modbus Protocol Data Unit (PDU), which is independent of the underlying communication layers. This last is further divided into two fields, the *function code*, which indicates in one byte the kind of action to be performed by the server, and the *data field*, containing the information to perform the requested action. Its length is variable from 0 to 252 bytes, and depends on the type of data carried by the specific packet. In the response message from a server to the client the function code is echoed unmodified, while the data field is updated to contain the data requested.

For what concerns the IP layer, the parameters to be set for both client and server are the local IP address, a subnet mask consistent with the class chosen for the local IP address, and the default gateway on the same subnet as the local IP address.

The MBAP header consists of 7 bytes divided into four fields, namely:

- *Transaction identifier*, 2 bytes used for transaction pairing. It couples a response message with the related request, mainly when multiple transactions are to be managed by the devices, thus it must be unique. The number of simultaneous transactions that a device can manage depends on its resource capacity, and can generally range from 1 to 16.
- *Protocol identifier*, 2 bytes set by default to 0 (0x00) identifyng the Modbus protocol.

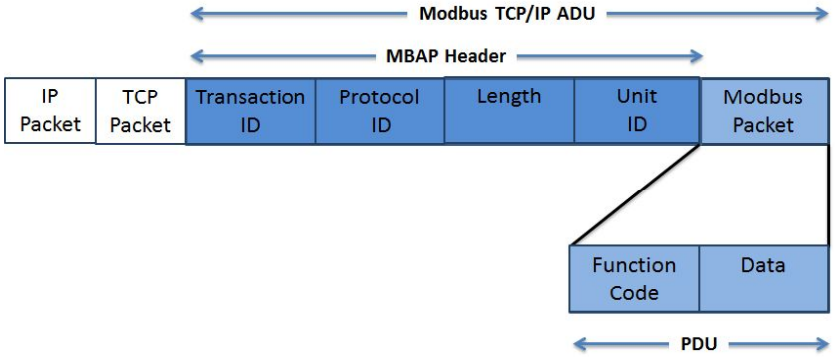

FIGURE 6.39: Local *ad-hoc* network schema

- *Length*, 2 bytes indicating the number of bytes of the following fields in the message.
- *Unit identifier*, 1 byte that carries the remote server address for bridged or routed connection with serial devices. The bridge is identified by the destination IP address, and the unit identifier is used to forward the request to the specific server device. If the server is a Modbus/TCP device, value 0x00 or 0xFF are used as default for the unit identifier, which becomes useless.

Three categories of function codes have been defined: public, which are well defined and unique, user-defined, deployed to implement actions not supported by the specifications, and reserved, currently used for legacy products and not available for public use. The function codes considered in this work are related to the functionalities implemented by control systems, thereby, only write and read commands are taken into account. The read operations apply to actuator status or sensor data through the PLC, while the write commands are used to modify the state of the coils. More specifically, three types of function codes have been employed:

Read coils - 01 (0x01): Used to read from 1 to 2000 contiguous status of coils (e.g. pumps and valves states). In the request PDU the starting address and the number of coils to be read are specified. In the data field of the response message, each bit corresponds to one coil (1 ON/0 OFF). Unused bits in the last data byte are 0-filled.

Read input registers - 04 (0x04): Used to read from 1 to 125 contiguous input registers (e.g. level sensor measurements). In the request PDU the starting register address and the number of registers to be read are specified. The data field in the response message is built with two bytes per register with a binary code of the read value.

Write multiple coils - 15 (0x0F): Used to force a sequence of coils. In the request PDU the starting address and the number of consecutive coils to be written are specified. The data field consists of a binary sequence corresponding to the desired state to be forced on each coil (1 ON/0 OFF). Unused bits in the last data byte are to be set to 0. The normal response echoes the function code, the starting address and the number of coils forced.

In case of error, the function code field of the response message (*exception response*) differs from the request by 0x80 (hex), and the data field contains an exeption code useful to determine the type of error that the server encountered when processing the request.

In order to establish a Modbus/TCP communication, a TCP connection between a client and a server has to be set. The connection is automatically established when the first packet is received from a client to the server port 502, and it is closed if a termination request arrives or if locally decided by the device. Moreover, an access control option can be used to forbid access to the device by unauthorized clients, by checking the client's IP address. Hence, initially a Modbus transaction is instantiated by the client. The request is then encoded by prefixing the PDU with the MBAP header with all the required information provided by the user application which is initiating the transmission demand. Finally, the IP destination address and the request ADU are passed to the TCP management module which sends it to the remote server. On the recipient side, the server has to analyze the received request, processes the required action, and sends back an appropriate response message. More specifically, the MBAP header is first parsed and the protocol identifier is checked, which has to be 0x00 to be correct, otherwise the request is discarded. Afterwards, if a transaction is available, it is initialized and the TCP connection identifier, the Modbus transaction identifier and the unit identifier are stored in memory. If no more transactions are available, a Modbus exception response is sent with exception code 6 (Server busy). Then, the PDU is parsed, starting with the analysis of the function code, and the service processing activity initiates if valid, otherwise an exception response with exception code 1 (Invalid function) is built. Subsequently, the Modbus response is built and sent to the TCP management component, which returns it to the right Modbus client. If the processing of the request was successful, the response is positive, thus the PDU is built with a function code that is the same as the request and the data field containing the results of the processing as requested by the client. If it is not, an exception response is sent providing relevant information of the error encountered during the processing. More specifically, the value 0x80 is added to the request function code, and the data field is filled with an exception code indicating the reason of the error. Then the MBAP header is added as prefix of the PDU, containing the same unit identifier as the Modbus request, the size of the PDU and unit identifies byte indicated in the length field, the protocol identifier set to 0x00 as given in the request, and the transaction identifier associated with the original request. To conclude, when the response is received by the client, it is associated with the original request by analyzing the transaction identifier in the MBAP header. If it corresponds to a pending transaction, it is parsed and a confirmation is sent to the user application, otherwise the message is discarded. Hence, the protocol identifier and unit identifier are first verified (the latter is discarded if the devices are directly connected on the TCP/IP network) and the function code and the response format are verified. If these are correct or an exception code is read, a positive confirmation is sent, as the command has been correctly received (but not necessarily successfully processed by the server), otherwise a negative confirmation signals the error to the user application.

For what concerns the time-out management, no specifications are provided by the Modbus protocol and it is generally left to the TCP defaults, set in the order of several seconds depending on the operating system (e.g. 72s for Microsoft Windows). It can be configured based on the criticality of the particular application, which is generally the case in AICS, taking into account a reasonable response time, the transport delays across the network and its topology. Hence, before initiating a retry, the time-out time

should be larger than the maximum expected reasonable time of the studied application.

On the security side, no specifications are provided or required by the standard, as it is based on the lower layers of the ISO/OSI model. Moreover, the Modbus standard has been first developed considering the serial communication between devices, and the main security issues arised at a later stage, with the introduction of the Ethernet networks. Actually, the data in the Modbus packets is transmitted in clear, without any type of encription or encoding, and no authentication mechanisms are provided as the sessions only verify the validity of certain parts of the message (e.g., address and function code), without ensuring non-repudiation and anti-replay mechanisms. As a fact, it is expected that the security matters are solved on the TCP level, based on the specific transmission support deployed. Thereby, the networks integration has exposed the systems to several cyber vulnerabilities, especially when the communication takes place over open networks, as the Modbus/TCP protocol is characterized by very rudimentary security properties. By exploiting this lack in security, a number of cyber-attacks have been carried out against the data exchanged between PLCs, SCADA system and historian, especially considering that being a master-client protocol every request from the master is authorized, and that timestamps are not considered by the specifications of the standard.

Chapter 7

The Cyber-Physical Problem on the FACIES Testbed

7.1 Testbed Analytic Model

7.1.1 Non-linear Model

The testbed has been mathematically modeled as a non-linear uncertain discrete-time dynamic switching system, for which also the effects derived from physical and/or cyber adverse events and other uncertainties have been considered:

$$x(k+1) = f(x(k), \tilde{q}(k)) + \eta(x(k), d(k), \tilde{q}(k)) + w_s(k)$$
$$y(k) = h(x(k)) + \Phi(x(k), d(k)) + w_o(k)$$

where

- $x \in \mathbb{R}^n$ is the state vector (water levels), with $n \in [1, \ldots, 5]$ number of tanks;
- $\tilde{q} \in \mathbb{Q}^m$ is the input vector (control commands), with $m \in [1, \ldots, 12]$ number of actuators, i.e. pumps and electro-valves, and $\mathbb{Q}$ is the set of values that the inputs can assume, where also the malicious inputs injected by the cyber-attacks are considered;
- $f : \mathbb{R}^n \times \mathbb{Q}^m \to \mathbb{R}^n$ represents the nominal dynamics;
- $d \in \mathbb{R}^p$ represent the physical faults that can affect the system, e.g., introduced by the manual valves, with $p \in [1, \ldots, 30]$;
- $\eta : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}^n$ is the function modeling the cyber-physical attacks/faults influencing the state;
- $y \in \mathbb{R}^\nu$ is the output vector (sensor measurements);
- $h : \mathbb{R}^n \to \mathbb{R}^\nu$ is the nominal output function;
- $\Phi : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^\nu$ is the cyber-physical attack function influencing the output;
- $w_s \in \mathbb{R}^n$ and $w_o \in \mathbb{R}^\nu$ represent the model uncertainties and noise vectors, respectively;
- $k \in \mathbb{N}$ is the time-step variable.

The plant interacts with the SCADA/HMI, which performs the control and monitoring actions. Thus, the general nominal model of the control actions generated by the SCADA can be expressed as:

$$q(k) = \beta(\tilde{y}(k), y_{ref}(k)),$$

where $q \in \mathbb{Q}^m$ is the vector of commands from the HMI to control the plant, $\tilde{y} \in \mathbb{R}^\nu$ is the sensor measurements as received by the SCADA system, $y_{ref}(k) \in \mathbb{B}^m$ are the commands obtained from the HMI and $\beta : \mathbb{R}^n \times \mathbb{B}^m \to \mathbb{Q}^m$ is the control function performed by the SCADA system.

As depicted in Figure 7.1, the interface between the plant and the SCADA system performs the following matching:

$$\tilde{q}(k) = q(k) + \Delta q(k) \qquad \tilde{y}(k) = y(k) + \Delta y(k),$$

where $\Delta q(k)$ and $\Delta y(k)$ are the modifications to the input signals of the plant and SCADA, respectively, due to cyber-physical anomalies. In the nominal operation of the system, $\Delta q(k) = 0$ and $\Delta y(k) = 0$.

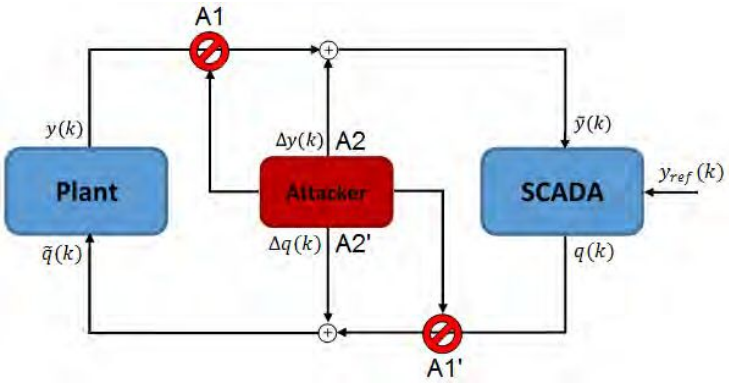

FIGURE 7.1: Diagram of the interaction between SCADA and controlled system. The attacker can interpose in the communication between plant and SCADA, altering the data flow in different ways.

Exploiting the flexibility that characterizes the testbed, different types of analysis can be carried out, starting from the single-tank case, to a more complex multi-tank cyber-physical system, considering serial-, parallel-, crossed-connections, and their combination. For any of the different configurations, one can implement suitable control strategies as state estimation or fault detection algorithms. In each case, the physical quantity of interest is the water level in the tanks.

In Figure 7.2 the nominal model of the testbed is shown. By using the balance equations, Torricelli's rule and the forward Euler discretization, with time step T_s, the following discrete-time state equations for the nominal healthy dynamics $f(\cdot)$ can be obtained for the complete system:

$$x_1(k+1) = x_1(k) + \frac{T_s}{A_1}\Big(u_1(k) - u_5(k) - S \cdot \Big[(c_1^m(v_1^m(k)) + c_1^s(v_1^s(k))) \cdot \sqrt{2gx_1(k)} + c_1^c(v_1^c(k)) \cdot \sqrt{2g \cdot |x_1(k) - x_2(k)|} \cdot \text{sign}(x_1(k) - x_2(k))\Big]\Big)$$

$$x_2(k+1) = x_2(k) + \frac{T_s}{A_2}\Big(u_2(k) - S \cdot \Big[(c_2^m(v_2^m(k)) + c_2^s(v_2^s(k))) \cdot \sqrt{2gx_2(k)} + c_1^c(v_1^c(k)) \cdot \sqrt{2g \cdot |x_1(k) - x_2(k)|} \cdot \text{sign}(x_1(k) - x_2(k))\Big]\Big)$$

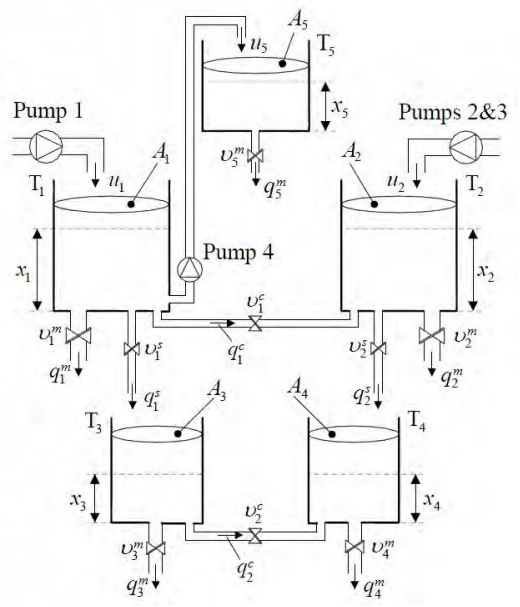

FIGURE 7.2: Testbed nominal diagram

$$x_3(k+1) = x_3(k) + \frac{T_s S}{A_3}\Big(c_1^s(v_1^s(k)) \cdot \sqrt{2gx_1(k)} - c_3^m(v_3^m(k)) \cdot \sqrt{2gx_3(k)} + c_2^c(v_2^c(k)) \cdot \sqrt{2g \cdot |x_3(k) - x_4(k)|} \cdot \text{sign}(x_3(k) - x_4(k))\Big)$$

$$x_4(k+1) = x_4(k) + \frac{T_s S}{A_4}\Big(c_2^s(v_2^s(k)) \cdot \sqrt{2gx_2(k)} - c_4^m(v_4^m(k)) \cdot \sqrt{2gx_4(k)} + c_2^c(v_2^c(k)) \cdot \sqrt{2g \cdot |x_3(k) - x_4(k)|} \cdot \text{sign}(x_3(k) - x_4(k))\Big)$$

$$x_5(k+1) = x_5(k) + \frac{T_s}{A_5}\Big(u_5(k) - S \cdot c_5^m(v_5^m(k)) \cdot \sqrt{2gx_5(k)}\Big)$$

The whole system consists of five cylindrical tanks of cross-section area $A_i, i = 1, \ldots, n, n = 5$, and pipes having cross-section area $S = 3.167 \cdot 10^{-5} m^2$. $g = 9.81 \frac{m}{s^2}$ denotes the gravity acceleration. Specifically, *Tanks 1* and *2* representing the *Residential Area 1* have a cross-section area $A_{1,2} = 5.73 \cdot 10^{-2} m^2$ and the allowed water level is in the interval $0 \leq x_{1,2} \leq 0.4m$, *Tanks 3* and *4* representing the *Residential Area 2* have a cross-section area $A_{3,4} = 3.46 \cdot 10^{-2} m^2$ and the allowed water level is in the interval $0 \leq x_{3,4} \leq 0.35m$, and *Tank 5* representing the *Industrial Area* has a cross-section area $A_5 = 7.07 \cdot 10^{-2} m^2$ and the allowed water level is in the interval $0 \leq x_5 \leq 0.2m$.

The water supply flow rates of the ON/OFF pumps filling tanks T1, T2 and T5 are denoted by u_j, $j = \{1, 2, 5\}$, with $u_i = \{0, 1.5 \cdot 10^{-4} \frac{m^3}{s}\}$, $i = \{1, 2\}$ and $u_5 = \{0, 0.85 \cdot 10^{-4} \frac{m^3}{s}\}$, while the direct ON/OFF electro-valves supplying water from the upper tanks to the lower T3 and T4 are controlled by the input signals $v_i^s = \{0, 1\}$, $i = \{1, 2\}$, representing their open (1) and closed (0) state, and each providing a flow rate indicated as q_i^s, $i = \{1, 2\}$.

To regulate the output flow rates from each tank q_i^m emulating the water demand from the consumers, multiple output on/off valves have been gathered in manifolds. These valves are controlled by the discrete signals $v_i^m \in \mathbb{N}$, assuming discrete values

depending on the number of opened valves in each manifold. More specifically, $v_i^m = \{0,\dots,4\},\ i = \{1,2\}$, $v_j^m = \{0,\dots,3\},\ j = \{3,4\}$ and $v_5^m = \{0,\dots,2\}$.

The tanks at the same level are connected by crossed-connection on/off valves, controlled by signals $v_i^c = \{0,1\},\ i = \{1,2\}$, and the water flow rate between them is denoted by $q_i^c,\ i = \{1,2\}$. In addition, the term $sign(\cdot)$ provides the direction of the water flow between tanks, as it depends on the relative water level.

Hence, the input vector is defined as $q = [u, v^s, v^m, v^c]$, with $m = 12$.

A discharge coefficient vector related to the output flow is associated to each valve and manifold. Their values have been experimentally determined, depend on the number of opened valves, i.e. $c_i^j(v_i^j(k))$, and are indicated as c_i^m for the valves in the manifolds ($c_i^m = \{0, 0.68, 0.34, 0.23, 0.17\},\ i = \{1,2\}, c_j^m = \{0, 0.4, 0.2, 0.13\},\ j = \{3,4\}, c_5^m = \{0, 0.7, 0.35\}$), $c_i^s = 0.62,\ i = \{1,2\}$ for the supply valves and $c_i^c = 0.37,\ i = \{1,2\}$ for the cross-connection valves.

The uncertainty has been considered as unstructured, non-linear and unknown, but bounded at all times by a function $\bar{\eta}(\cdot)$, experimentally determined with data obtained from the healthy operation of the system. More specifically, for each component, the bounding function is $\bar{\eta}_i(x(k), q(k), k) = a_i$, with $a_i \in \mathbb{R}^+$ and

$$|\eta_i(x(k), q(k), k)| \leq \bar{\eta}_i(x(k), q(k), k) \quad \forall k$$

7.1.2 Linearized Model

The previous model's complexity is mainly due to the presence of nonlinearities and switching conditions. In several situations it is convenient to operate some simplifications on it, as linearization could be. For the case of the two-tanks system in the serial configuration depicted in Figure 7.3 a linearized model has been obtained:

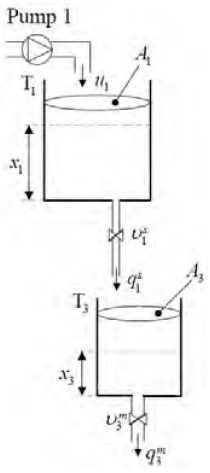

FIGURE 7.3: Two-tanks serial configuration diagram

$$\Delta x(k+1) = A\,\Delta x(k) + B\,\Delta q(k)$$

where,

$$A = \begin{bmatrix} 1 - \frac{T_s}{A_1} \cdot \frac{S \cdot \bar{c}_1^s(v_1^s) \cdot \sqrt{2g\bar{x}_1}}{2\bar{x}_1} & 0 \\ \frac{T_s}{A_2} \cdot \frac{S \cdot \bar{c}_1^s(v_1^s)\sqrt{2g\bar{x}_1}}{2\bar{x}_1} & 1 - \frac{T_s}{A_2} \cdot \frac{S \cdot \bar{c}_3^m(v_3^s)\sqrt{2g\bar{x}_3}}{2\bar{x}_3} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{T_s}{A_1} & -\frac{T_s}{A_1} \cdot S\sqrt{2g\bar{x}_1} & 0 \\ 0 & \frac{T_s}{A_2} \cdot S\sqrt{2g\bar{x}_1} & -\frac{T_s}{A_2} \cdot S\sqrt{2g\bar{x}_3} \end{bmatrix}$$

and $\bar{c}_i^s(v_1^s), \bar{x}_i$ equilibrium levels.

In addition, the linearization of the whole water system has been carried out, whilst its implementation is still missing. Matrix A has been split into 7 sub-matrices, for the sake of space and clarity:

$$A = \left[\begin{array}{cc|c} A_{12} & A_2 & A_{15} \\ A_3 & A_{34} & \\ \hline \multicolumn{2}{c|}{A_{51}} & A_5 \end{array}\right]$$

$$A_{12} = \begin{bmatrix} 1 - \frac{T_s \cdot S}{A_1}\left(\left(\bar{c}_1^s(v_1^s) + \bar{c}_1^m(v_1^m)\right) \cdot \frac{\sqrt{2g\bar{x}_1}}{2\bar{x}_1} - \bar{c}_1^c(v_1^c) \cdot \sqrt{2g|\bar{x}_1 - \bar{x}_2|} \cdot (2\delta(\bar{x}_1 - \bar{x}_2) + \frac{1}{2|\bar{x}_1 - \bar{x}_2|})\right) & -\frac{T_s \cdot S}{A_1}\left(\bar{c}_1^c(v_1^c) \cdot \sqrt{2g|\bar{x}_1 - \bar{x}_2|} \cdot (2\delta(\bar{x}_1 - \bar{x}_2) + \frac{1}{2|\bar{x}_1 - \bar{x}_2|})\right) \\ \frac{T_s \cdot S}{A_2}\left(\bar{c}_1^c(v_1^c) \cdot \sqrt{2g|\bar{x}_1 - \bar{x}_2|} \cdot (2\delta(\bar{x}_1 - \bar{x}_2) + \frac{1}{2|\bar{x}_1 - \bar{x}_2|})\right) & 1 - \frac{T_s \cdot S}{A_2}\left((\bar{c}_2^s(v_2^s) + \bar{c}_2^m(v_2^m)) \cdot \frac{\sqrt{2g\bar{x}_2}}{2\bar{x}_2} + \bar{c}_1^c(v_1^c) \cdot \sqrt{2g|\bar{x}_1 - \bar{x}_2|} \cdot (2\delta(\bar{x}_1 - \bar{x}_2) + \frac{1}{2|\bar{x}_1 - \bar{x}_2|})\right) \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad A_{15} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = {A_{51}}^T \qquad A_3 = \begin{bmatrix} \frac{T_s \cdot S}{A_3} \cdot \frac{\bar{c}_1^s(v_1^s)\sqrt{2g\bar{x}_1}}{2\bar{x}_1} & 0 \\ 0 & \frac{T_s \cdot S}{A_4} \cdot \frac{\bar{c}_2^s(v_2^s)\sqrt{2g\bar{x}_2}}{2\bar{x}_2} \end{bmatrix}$$

$$A_{34} = \begin{bmatrix} 1 + \frac{T_s \cdot S}{A_3}\left(\frac{-\bar{c}_3^m(v_3^s)\cdot\sqrt{2g\bar{x}_3}}{2\bar{x}_3} + \bar{c}_2^c \cdot \sqrt{2g|\bar{x}_3 - \bar{x}_4|} \cdot (2\delta(\bar{x}_3 - \bar{x}_4) + \frac{1}{2|\bar{x}_3 - \bar{x}_4|})\right) & -\frac{T_s \cdot S}{A_3} \cdot \left(\bar{c}_2^c \cdot \sqrt{2g|\bar{x}_3 - \bar{x}_4|} \cdot (2\delta(\bar{x}_3 - \bar{x}_4) + \frac{1}{2|\bar{x}_3 - \bar{x}_4|})\right) \\ \frac{T_s \cdot S}{A_4} \cdot \left(\bar{c}_2^c \cdot \sqrt{2g|\bar{x}_3 - \bar{x}_4|} \cdot (2\delta(\bar{x}_3 - \bar{x}_4) + \frac{1}{2|\bar{x}_3 - \bar{x}_4|})\right) & 1 - \frac{T_s \cdot S}{A_4} \cdot \left(\frac{\bar{c}_4^m(v_4^s)\cdot\sqrt{2g\bar{x}_4}}{2\bar{x}_4} + \bar{c}_2^c \cdot \sqrt{2g|\bar{x}_3 - \bar{x}_4|} \cdot (2\delta(\bar{x}_3 - \bar{x}_4) + \frac{1}{2|\bar{x}_3 - \bar{x}_4|})\right) \end{bmatrix}$$

$$A_5 = \begin{bmatrix} 1 - \frac{T_s \cdot S}{A_5} \cdot \frac{\bar{c}_5^m(v_5^s)\cdot\sqrt{2g\bar{x}_5}}{2\bar{x}_5} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{T_s}{A_1} & 0 & -\frac{T_s}{A_1} & -\frac{T_s \cdot S}{A_1}\sqrt{2g\bar{x}_1} & 0 & -\frac{T_s \cdot S}{A_1}\sqrt{2g\bar{x}_1} & 0 & 0 & 0 & \frac{T_s \cdot S}{A_1}\sqrt{2g|\bar{x}_1 - \bar{x}_2|} \cdot sign(\bar{x}_1 - \bar{x}_2) & 0 \\ 0 & \frac{T_s}{A_2} & 0 & 0 & -\frac{T_s \cdot S}{A_2}\sqrt{2g\bar{x}_2} & 0 & -\frac{T_s \cdot S}{A_2}\sqrt{2g\bar{x}_2} & 0 & 0 & \frac{T_s \cdot S}{A_2}\sqrt{2g|\bar{x}_1 - \bar{x}_2|} \cdot sign(\bar{x}_1 - \bar{x}_2) & 0 \\ 0 & 0 & 0 & \frac{T_s \cdot S}{A_3}\sqrt{2g\bar{x}_1} & 0 & 0 & -\frac{T_s \cdot S}{A_3}\sqrt{2g\bar{x}_3} & 0 & 0 & 0 & \frac{T_s \cdot S}{A_3}\sqrt{2g|\bar{x}_3 - \bar{x}_4|} \cdot sign(\bar{x}_3 - \bar{x}_4) \\ 0 & 0 & 0 & 0 & \frac{T_s \cdot S}{A_4}\sqrt{2g\bar{x}_2} & 0 & 0 & -\frac{T_s \cdot S}{A_4}\sqrt{2g\bar{x}_4} & 0 & 0 & \frac{T_s \cdot S}{A_4}\sqrt{2g|\bar{x}_3 - \bar{x}_4|} \cdot sign(\bar{x}_3 - \bar{x}_4) \\ 0 & 0 & \frac{T_s}{A_5} & 0 & 0 & 0 & 0 & 0 & -\frac{T_s \cdot S}{A_5}\sqrt{2g\bar{x}_5} & 0 & 0 \end{bmatrix}$$

7.2 Fault Detection Module

The Fault Detection module aims to monitor the system, obtaining on-line data from the SCADA system and triggering an alarm when a fault occurs. For this approach, the difference between the sensor measurements x_i and the estimated values for the water level in each tank $\hat{x}_i$ obtained from the system model is calculated for each component [119], [120]. Hence, an anomaly is detected when there is a strong discrepancy between the estimated measurement obtained from the model and the data gathered from the sensor. To this end, two different techniques have been developed based on the linearized and non-linear multi-tank models, respectively.

More specifically, for the linearized two-tanks model, a discrete-time version of the Fault Detection Filter has been deployed as described in [49], which represents a well-known observer-based technique for the generation of residual signals. It is based on the design of a full-state observer,

$$\hat{x}(k+1) = A\hat{x}(k) + Bq(k) + L(y(k) - C\hat{x}(k))$$
$$\hat{y}(k) = C\hat{x}(k)$$

where $\hat{x} \in \mathbb{R}^n$ is the vector of the estimated state, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the state-space matrices described in Section 7.1.2, $C \in \mathbb{R}^{n\times n}$ in the specific case is an identity matrix, $\hat{y} \in \mathbb{R}^n$ are the estimated outputs and $L \in \mathbb{R}^{n\times n}$ is a gain matrix, chosen so that the observer is stable. The residual signal $r \in \mathbb{R}^n$ is thereby defined as:

$$r(k) = |y(k) - \hat{y}(k)|.$$

As previously mentioned, such residual is calculated as the difference between the real measurements from the sensors and the estimated output obtained from the linearized model.

A threshold for the error signal has been experimentally defined and computed with the RMS value method, previously exposed in Section 3.4, and considering the maximum value of the residual signal in nominal operative conditions. Thus, a fault is said to be detected if and only if the error signal exceeds the threshold.

A more sophisticated non-linear fault detection technique, based on the one proposed in [121], has been implemented and deployed for the Fault Diagnosis module of the FACIES architecture. Again, the fault estimator dynamics is based on the difference between the sensor readings of the water level in the tanks and the expected levels obtained from the non-linear model, and the error signal obtained is compared to a fixed threshold. Hence, when the magnitude of the estimation error of at least one component, defined as $e_i(k) := x_i(k) - \hat{x}_i(k)$, exceeds the corresponding threshold, i.e. $|e_i(k)| \geq \bar{e}_i(k)$, a fault is said to be detected in the i-th component.

To this end, Algorithm 7 has been developed and implemented.

When the Fault Detection routine is called, a loop of period $T_s = 0.5s$ is run, which allows to run the adequate query to the SCADA database at every time step k, in order to obtain the last recorded measurements from the sensors and the inputs from the controller to the actuators, transfer the data over the Local Network and execute all the steps inside the periodic loop. Then, the state estimation error $e(k)$ is calculated and its modulus $|e(k)|$ is compared with the estimation error threshold $\bar{e}(k)$. If the state estimation error it is higher, a fault is detected and an alarm is triggered. Then, the state estimations are calculated for the next iteration using the FD estimator dynamics,

Algorithm 7 Online Fault Detection algorithm

procedure FD($\hat{x}(0), \bar{e}(0)$) ▷ as input the initial values for $\hat{x}$ and $\bar{e}$
  $k \leftarrow 0, t \leftarrow timer$ ▷ t denotes a timer that starts from 0 and increases
  repeat every $t = k \cdot T_s$ ▷ repetition of the loop with period T_s
    $x(k) \leftarrow x_{db_{last}}$ ▷ get from database the last recorded sensors values
    $q(k) \leftarrow q_{db_{last}}$ ▷ get from database the last recorded input values
    $e(k) \leftarrow x(k) - \hat{x}(k)$ ▷ state estimation error calculation
    if $|e(k)| > \bar{e}(k)$ **then** ▷ comparison for fault detection
      Alarm - Fault Detected
    end if
    $\hat{x}(k+1) \leftarrow \lambda(\hat{x}(k) - x(k)) + f(x(k), q(k))$ ▷ FD estimator dynamics
    $\bar{e}(k+1) \leftarrow \lambda\bar{e}(k) + \bar{\eta}(x(k), q(k), k)$ ▷ FD estimation error threshold
    $k \leftarrow k + 1$
  until FD turn off
end procedure

where $0 \leq \lambda < 1$ ($\lambda = 0.9$ chosen for the experimental tests). Similarly, the estimation error threshold $\bar{e}(k)$ is calculated, where $\bar{\eta}(\cdot)$ is a known function that bounds the modeling uncertainty $\eta(\cdot)$, that has been found experimentally by gathering data from the healthy scenario. More specifically, for each FD component $i = 1, \ldots, 5$, the bounding function is $\bar{\eta}_i(x(k), q(k), k) = a_i$, where $a_i \in \mathbb{R}^+$ ($a_i = 8 \cdot 10^{-4}$ experimentally tuned and, for the sake of simplicity, equally chosen for each module).

7.2.1 Experimental Results

As for the system models, the exposed Fault Detection techniques have been implemented in Matlab, and different types of faults have been induced in the system at different time instants so as to test their effectiveness. Every time a fault is detected by the FDAE system, the involved tank/s and detection time are indicated in Matlab's Command Window, and a message is sent through TCP/IP to the SCADA, which shows such information in a pop-up window on the HMI and allows the operator to acknowledge the alarm.

The physical faults have been induced by means of the manual valves to simulate leaks in the tanks or along the pipelines, or by virtually modifying the state of the actuators. Specifically, independent leaks of arbitrary magnitude can be induced at different locations along the pipelines, 5 of which directly connect the tanks to the reservoir and reproduce leaks of water, while the other 2 represent an undesired and unknown direct connection between the upper and lower tanks. As previously mentioned, the faulty or undesired flow of each leak can be modulated with the opening of the manual valve. Moreover, 31 different faults involving sensors or actuators, i.e. pumps and electro-valves, can be carried out in the single components both by changing the software configuration, simulating its malfunctioning, disruption or undesired activation, or by physically disconnecting it from the PLC or from the power supply source.

For the linearized two-tanks system both the filling and emptying phases have been considered, during which leak faults have been induced, as depicted in Figure 7.4.

The upper and central charts show the level of *Tanks 1* and *3*, respectively, measured by the sensors (blue) and the results obtained from the linear model (dotted red), while lower charts show the error for each tank, calculated as the difference between the actual and the estimated behavior, and the threshold deployed to determine the presence of a fault. Such thresholds have been computed with the RMS value method, previously exposed in Section 3.4, and are respectively $J_{th,T1} = 0.035m$ and $J_{th,T2} = 0.025m$.

In Figure 7.4 (a) a leak in *Tank 1* has been introduced at time $t = 25s$, which is detected by the linear FD system at about $t = 95s$. The behavior for a leak in *Tank 3* is depicted in Figure 7.4 (b). Such fault takes place at time $t = 30s$, and it is detected at time $t = 40s$. In Figure 7.4 (c) a multiple leak fault takes place in both tanks during emptying from an initial level. More specifically, a fault in *Tank 1* occurs at time $t = 20s$, while a second fault takes place in *Tank 3* at time $t = 60s$, which are detected at about $t = 65s$ and $t = 70s$, respectively.

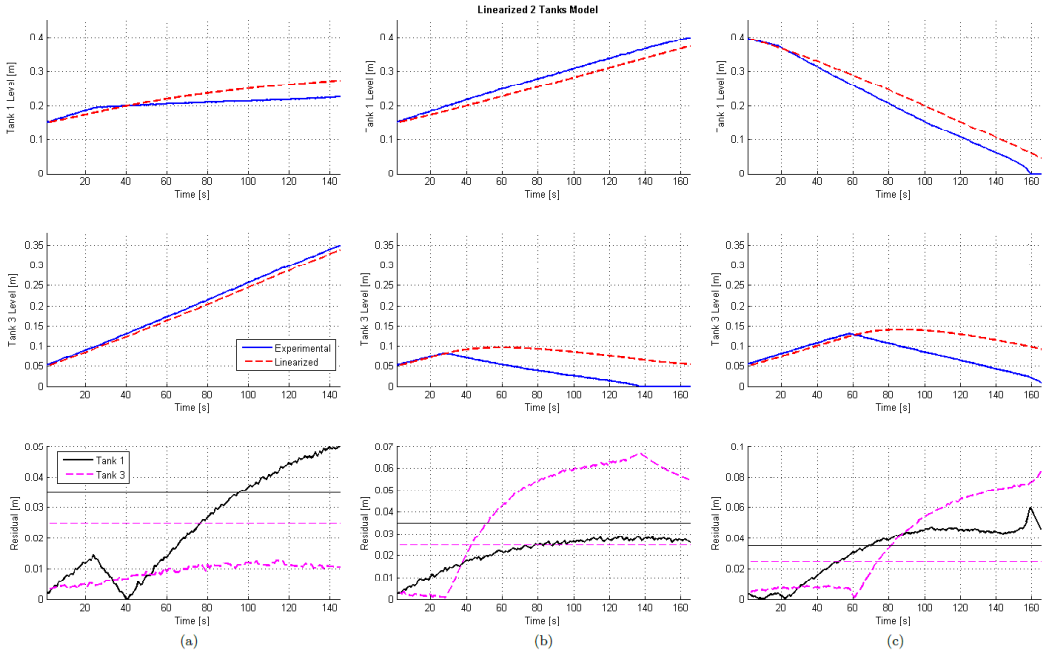

FIGURE 7.4: (a) Leak fault in *Tank 1* @ $t = 25s$ (100% leak) during tank filling. (b) Leak fault in *Tank 3* @ $t = 30s$ during tank filling (100% leak). (c) Multiple leak fault in *Tanks 1* and *3* @ $t = 20s$ and $t = 60s$, respectively, during *Tank 1* emptying and *Tank 3* filling. Notice that the fault in cases (b) and (c) leads to the total emptying of *Tank 3*, at times $t = 139s$ and $t = 165s$, respectively.

For the non-linear model and FDAE system, a Matlab routine has been specifically developed so as to periodically query the database and run a simulation of the dynamic model, thereby being able to calculate in real-time the error between the current and expected values, and plotting the results in the relative charts.

The results for a healthy daily scenario run are depicted in Figure 7.5. Each pair of charts is related to a specific tank. The upper graphs show the actual water level, as obtained from the sensor measurements during the daily scenario (blue), and the expected behavior, as calculated with the system model (green). The lower charts show the error (residual) calculated as the difference between the actual and the estimated behavior (blue), and the threshold deployed to determine the presence of a fault (green). As it can be seen, during the healthy run no threshold is reached, hence no fault is detected in the system.

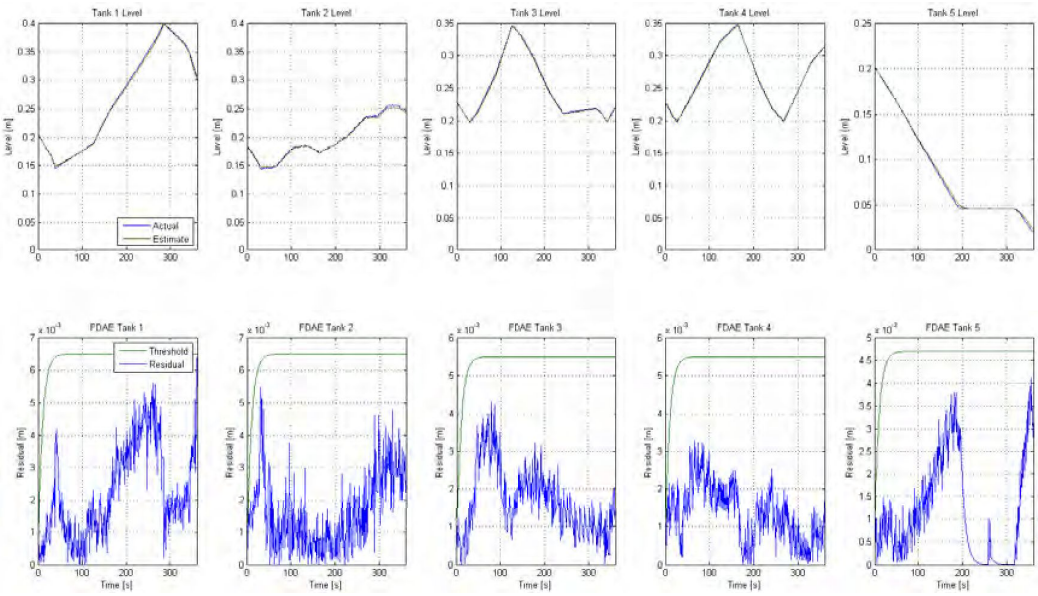

FIGURE 7.5: Healthy behavior of the system during the daily scenario

As mentioned, different types of faults have been induced into the system, as shown in Table 7.1, and the relative results are here exposed.

	Component	Entity of the fault	Fault Time
Leak Fault	Tank 1	100%	$250s$
	Tank 2	100%	$220s$
	Tank 3	100%	$140s$
	Tank 4	50% 100%	$80s$
	Tank 5	100%	$240s$
	Connection Tanks 1 and 3	100%	$275s$
	Connection Tanks 2 and 4	100%	$200s$
Pump Fault	Pump 2	Turns OFF	$225s$
	Pump 1	Turns OFF	$200s$ $250s$
Valve Fault	V.1.3	Remains closed	$150s$
	V.2.5	Remains closed	$65s$
	V.D.17	Remains closed	$200s$
Multiple Fault	Tank 3	100%	
	Pump 1	Turns OFF	$100s$
	V.D.17	Remains closed	
	Tank 3	100%	
	Pump 2	Turns OFF	$260s$
	V.D.17	Remains closed	

TABLE 7.1: Faults induced during the daily scenario

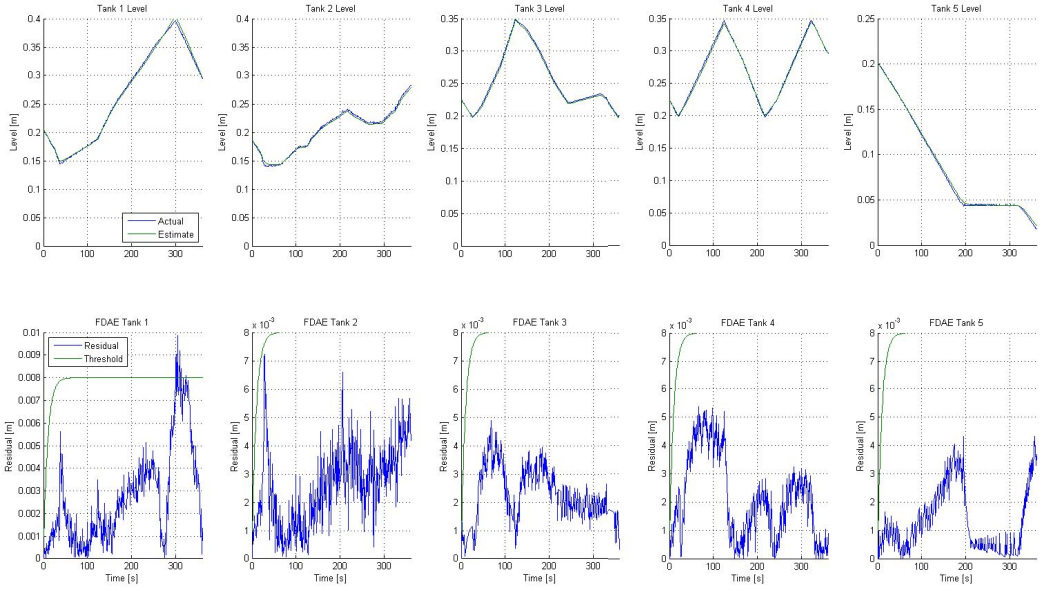

FIGURE 7.6: Leak fault in *Tank 1* @ $t = 250s$ (100% leak)

The leak fault to *Tank 1* occurs during the filling phase, hence the input flow compensates such leak and the fault is detected with a delay of about $50s$. Moreover, no effects can be highlighted on *Tank 3* because of the drop of water in *Tank 1*.

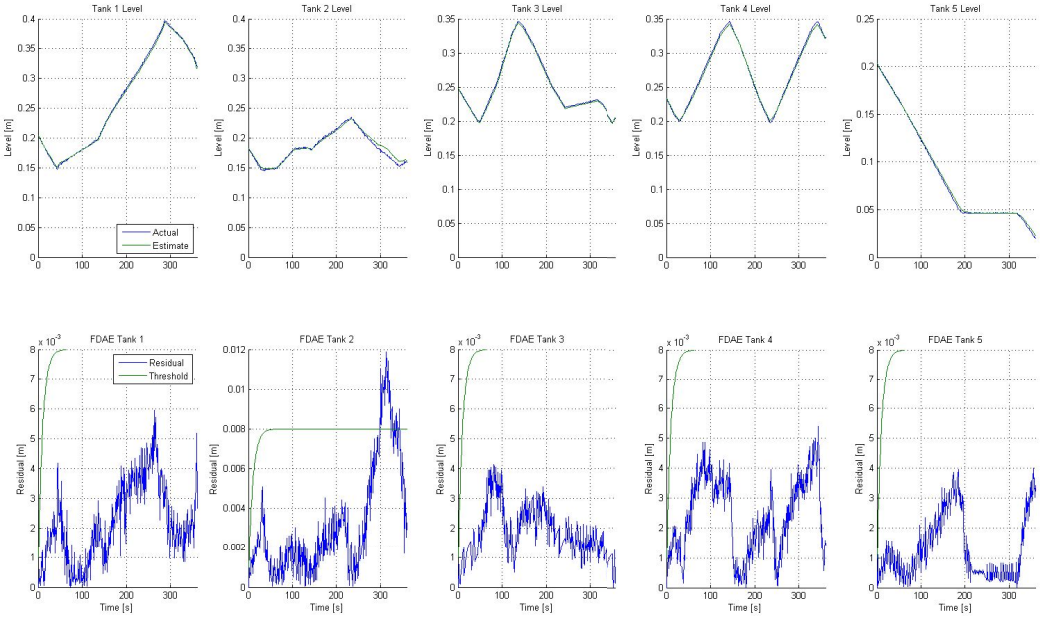

FIGURE 7.7: Leak fault in *Tank 2* @ $t = 220s$ (100% leak)

In Figure 7.7 it can be seen that, as in the previous case, the leak fault takes place during the filling of *Tank 2*, leading to a delay of about $80s$ in the detection, and *Tank 4* is subjected to no effects, keeping its healthy behavior.

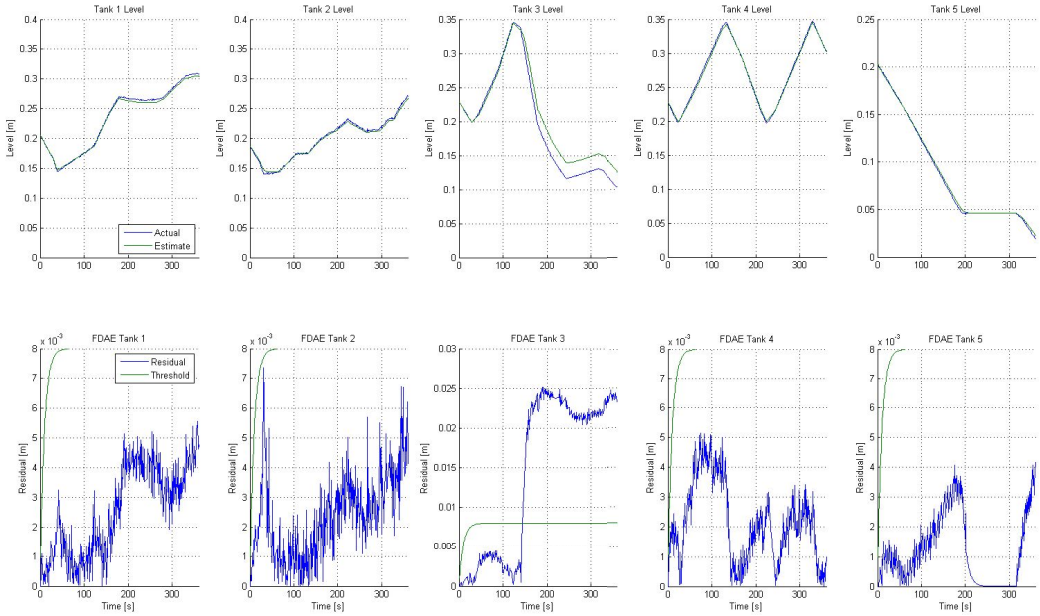

FIGURE 7.8: Leak fault in *Tank 3* @ $t = 140s$ (100% leak)

The leak fault at *Tank 3* tanking place at $t = 140s$ is detected with a very short time delay. Moreover, as the water level goes below the lower limit imposed by the rules, the supply valve opens at about $t \approx 180s$, what leads to a water diminishment in *Tank*

1, but is not enough to compensate the leak in *Tank 3*, which ends up with its emptying.

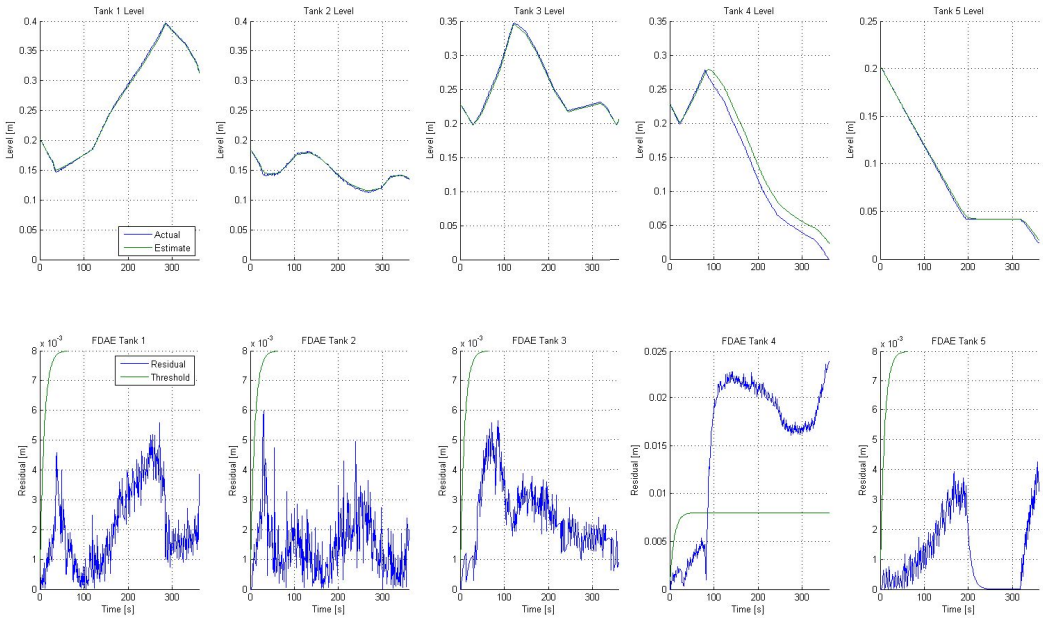

FIGURE 7.9: Leak fault in *Tank 4* @ $t = 80s$ (100% leak)

The leak in *Tank 4* leads to its rapid emptying, triggering the FD system with a short delay, as depicted in Figure 7.9. As in the previous case, the water level goes below the lower threshold, but the supply from *Tank 2* is not enough to compensate the leak.

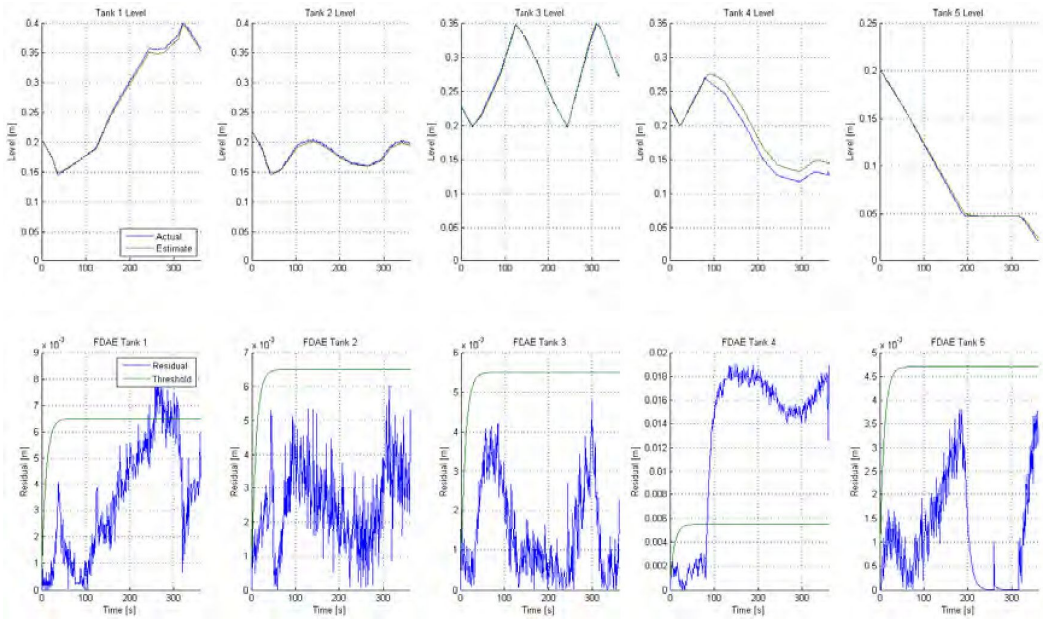

FIGURE 7.10: Leak fault in *Tank 4* @ $t = 80s$ (50% leak)

In this second case, the leak is lower, but the consequences are almost the same. The

fault is detected with a short delay, despite the amplitude of the error signal is inferior, and the water supply does not totally compensate the leak, which leads to the emptying of the tank.

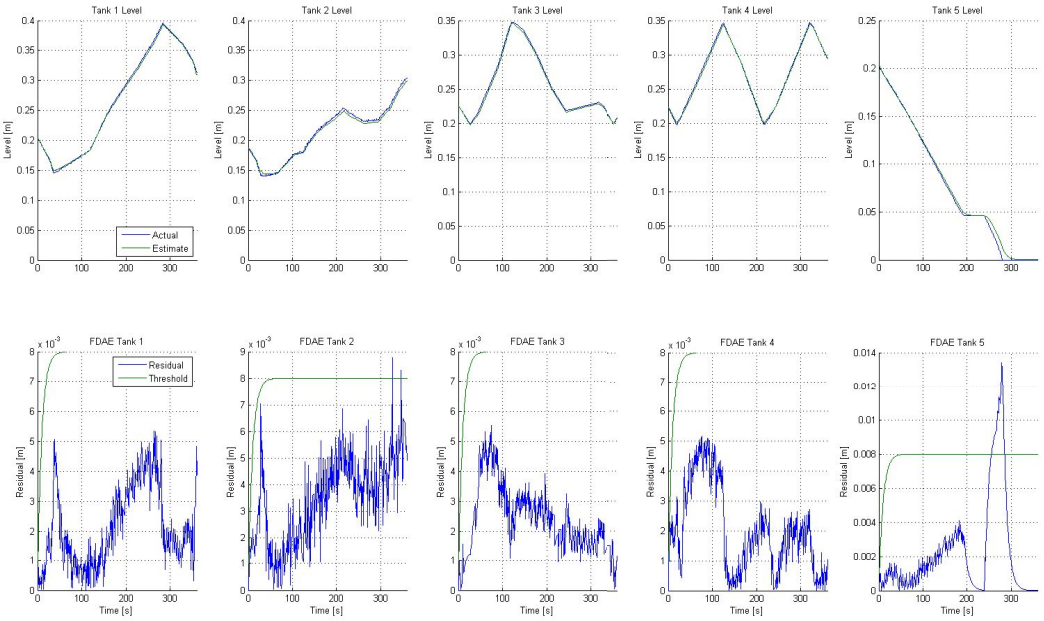

FIGURE 7.11: Leak fault in *Tank 5* @ $t = 240s$ (100% leak)

The leak fault that takes place in *Tank 5* is successfully detected with a delay of about $5s$, with no consequences on the other areas.

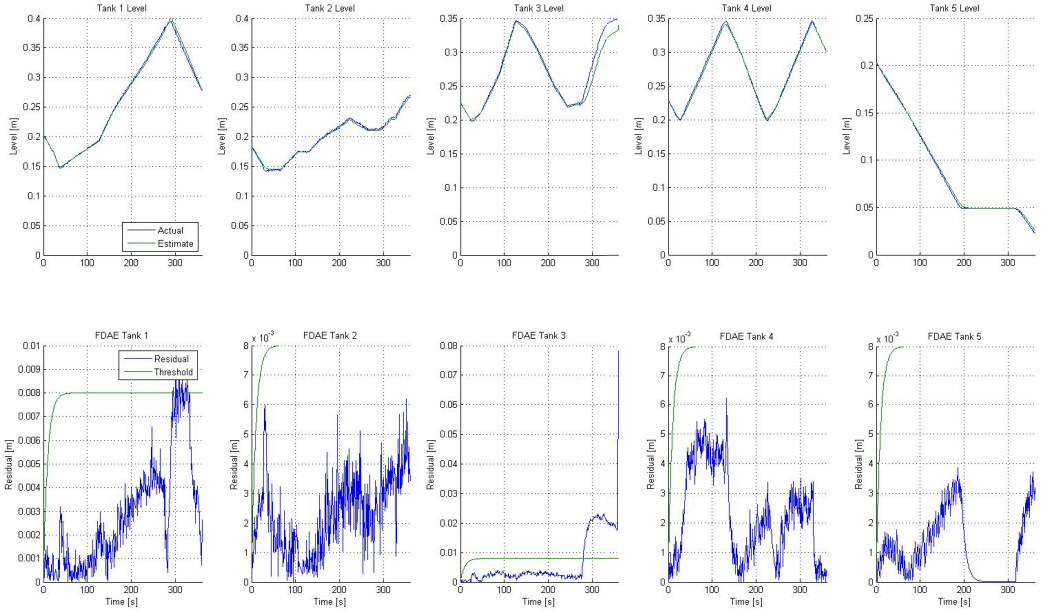

FIGURE 7.12: Leak fault along pipe connecting *Tank 1* to *Tank 3* @ $t = 275s$ (100% leak)

The leak along the pipe connecting *Tank 1* to *Tank 3* triggers a double alarm from the FD system, as expected. The first one is suddenly generated as a consequence of the unexpected water level increasing in *Tank 3*. The second alarm is obtained after about $20s$ of delay due to the water diminishment in *Tank 1*.

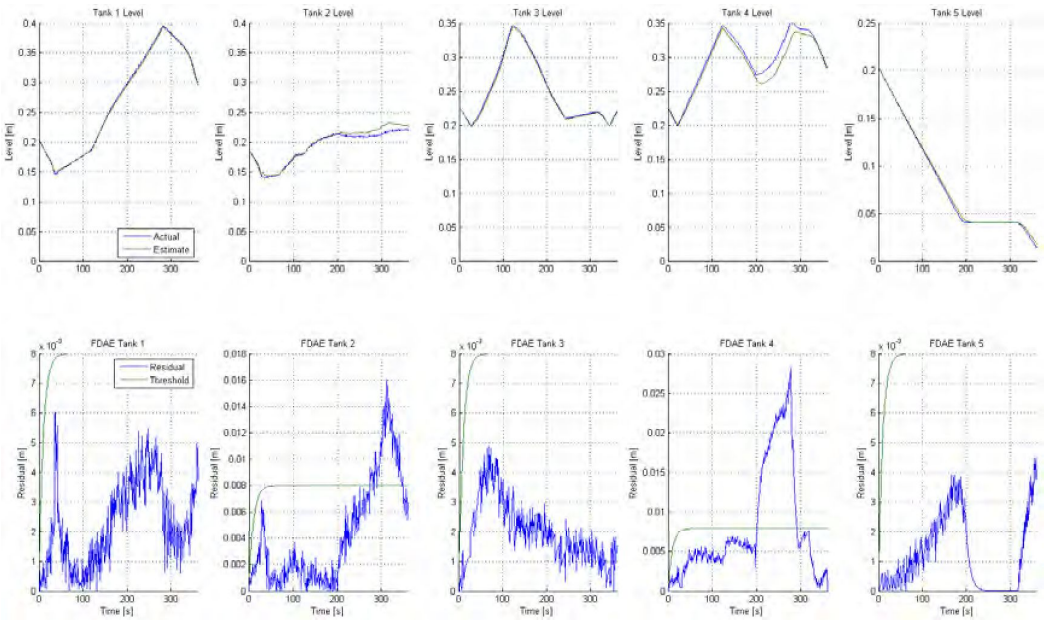

FIGURE 7.13: Leak fault along pipe connecting *Tank 2* to *Tank 4* @ $t = 200s$ (100% leak)

As in the latter case, a water leak from *Tank 2* that pours in *Tank 4* is rapidly detected. Firstly, the unexpected filling of *Tank 4* triggers an alarm, while the residual in *Tank 2* increases due to the diminishment of water in it, until the fault is detected with a delay of about $80s$. Moreover, a manual valve in *Tank 4* is opened at $t = 281s$ in order to avoid its overflow.

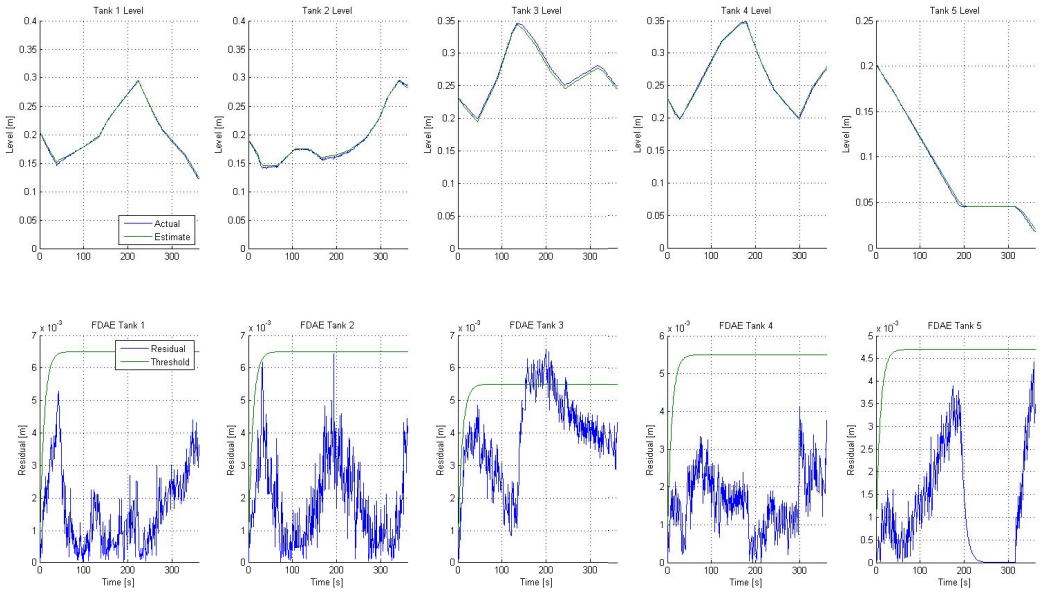

FIGURE 7.14: Fault in Pump 2 which unexpectedly turns OFF @ $t = 225s$

In this case, the pump supplying water to *Tank 1* stops working at $t = 225$, leading to a sudden drop of the water level in such tank. Nevertheless, such fault is successfully detected by the FD system due to the pressure drop in *Tank 1*, which leads to a reduction of the water supplied to *Tank 3*.

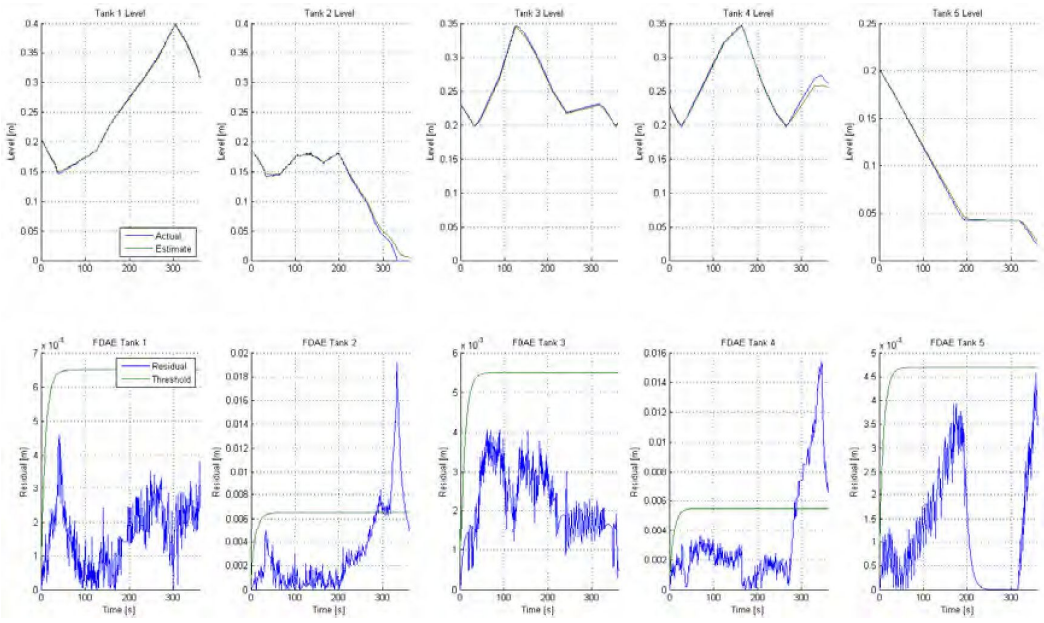

FIGURE 7.15: Fault in Pump 1 which unexpectedly turns OFF @ $t = 200s$

As the pump supplying water to *Tank 2* stops working, its water level rapidly decreases until the fault is detected with a delay of about $80s$. At the same time, a second fault is triggered in *Tank 4* due to the pressure drop during its filling.

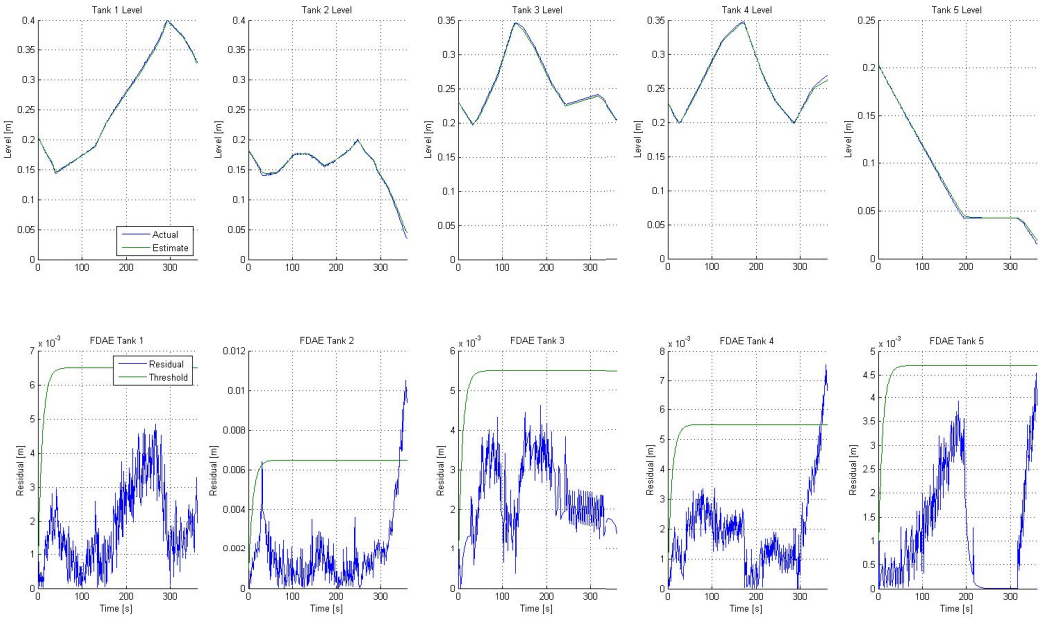

FIGURE 7.16: Fault in Pump 1 which unexpectedly turns OFF @ $t = 250s$

In this second case, the fault in Pump 1 is detected with a large delay in two parts of the FD system. The first is due to the water drop in *Tank 2*, while the second occurs in *Tank 4*, as water is not supplied to it as expected after the fault.

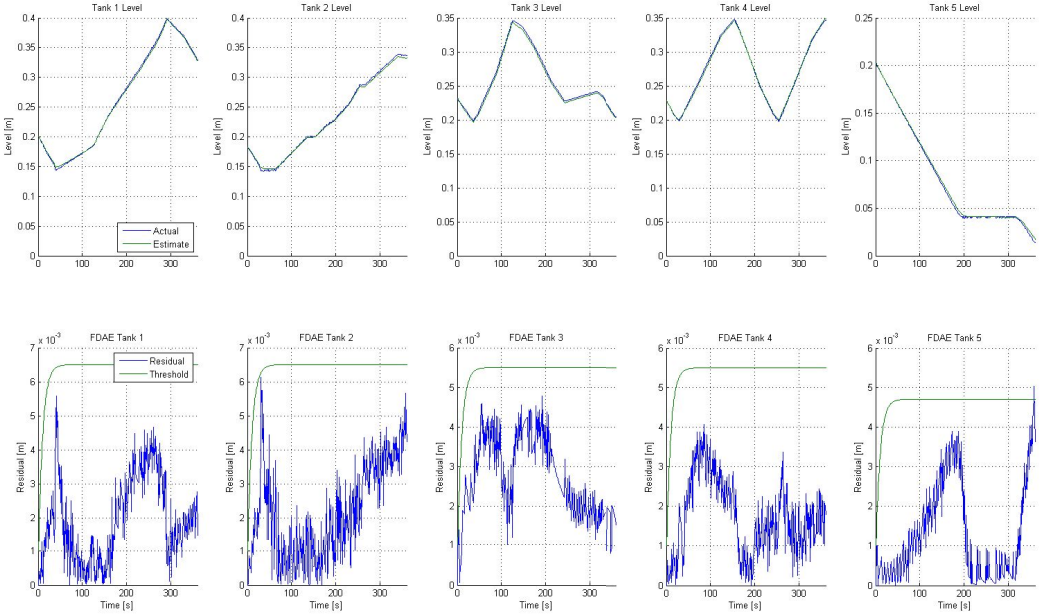

FIGURE 7.17: Fault in Valve V.2.5 which remains closed @ $t = 65s$

As it can be seen in Figure 7.17, the fault of Valve V.2.5 is not recognized by the FD system. Nevertheless, the error signal shows an increasing trend, as the water level in *Tank 2* is more than expected due to the missed opening of the valve for $106s \leq t < 296s$

and $318s \leq t < 360s$.

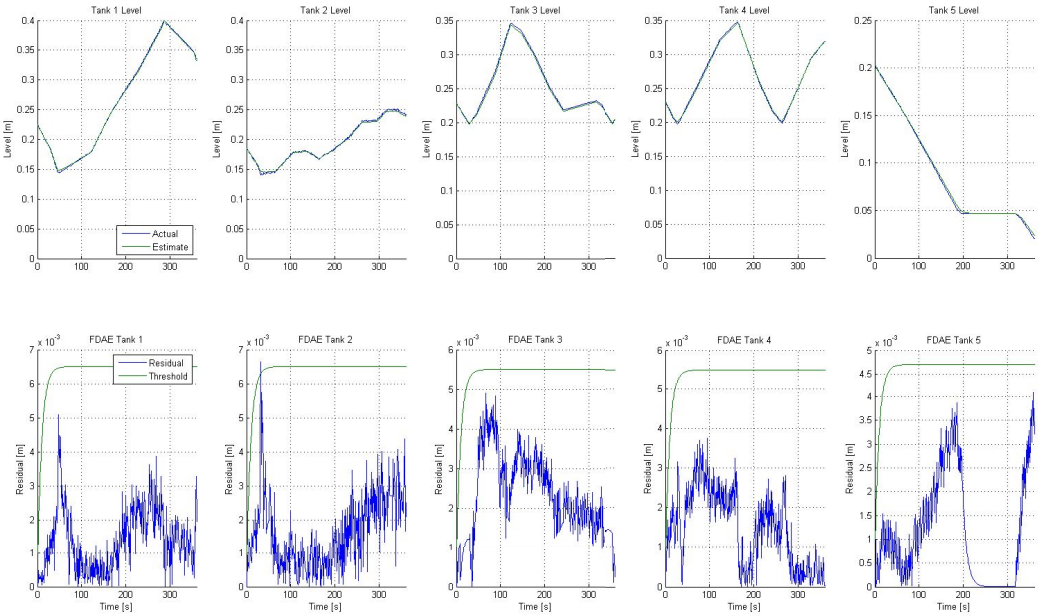

FIGURE 7.18: Fault in Valve V.1.3 which remains closed @ $t = 150s$

Similarly, the fault in Valve V.1.3 is not evident, and no particular trends can be observed in the error signal, as depicted in Figure 7.18.

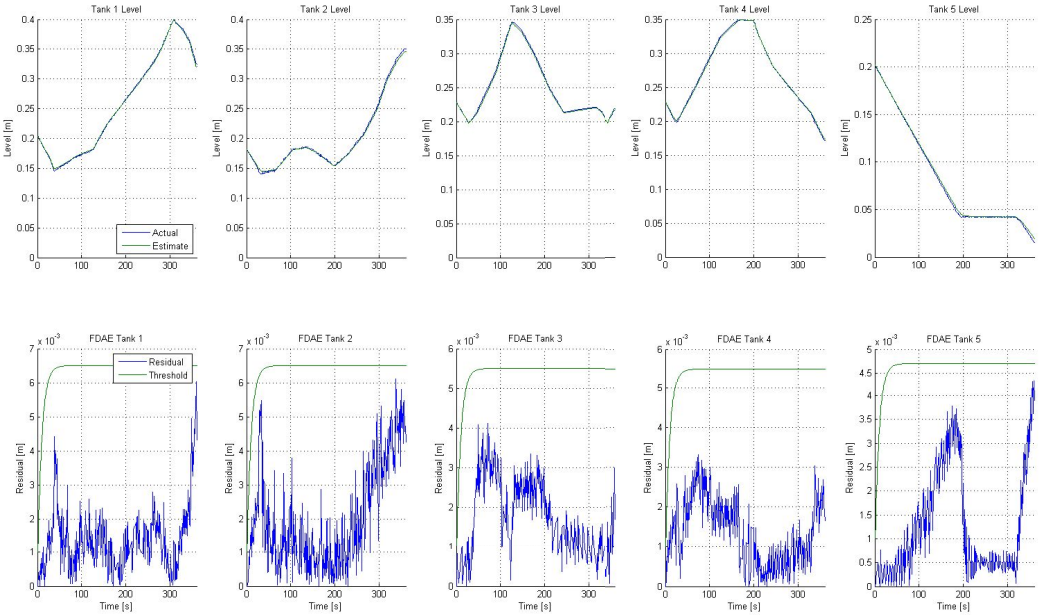

FIGURE 7.19: Fault in Valve V.D.17 which remains closed @ $t = 200s$

As in the latter case, the fault is not detected, but the error shows an increasing trend as the water level in *Tank 2* is higher than expected. Moreover, the water level in

Tank 4 decreases as its supply valve does not open as requested.

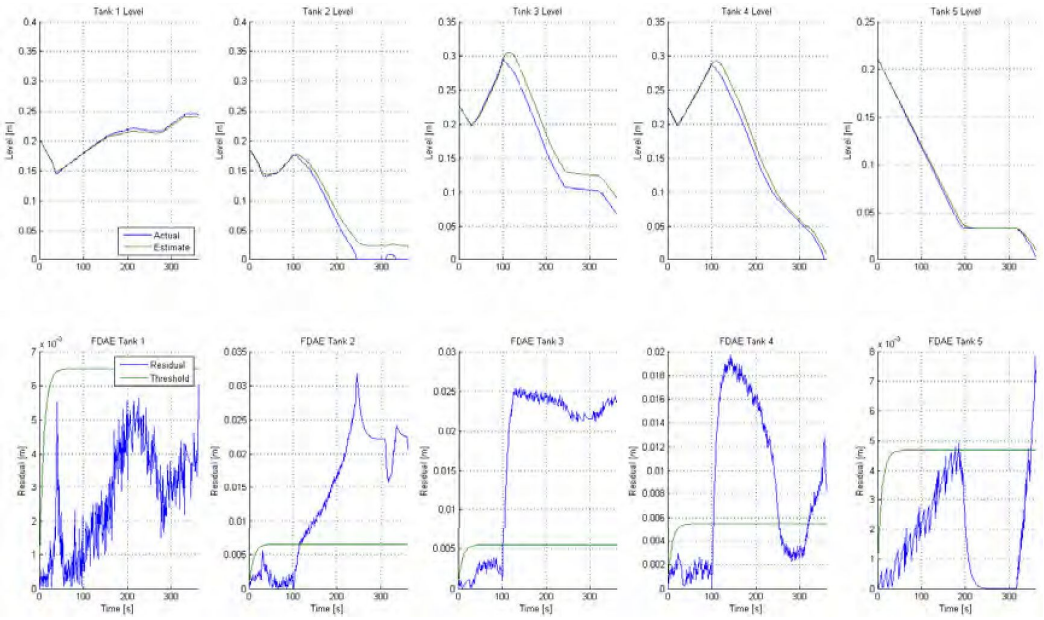

FIGURE 7.20: Multiple fault in *Tank 3*, Pump 1 and Valve V.D.17 @ $t = 100s$

To conclude, two different cases of multiple fault have been induced to the system. The first, which consists in a leak in *Tank 3* and the undesired switch OFF of Valve V.D.17 and Pump 1, is first detected at time $t = 102s$, thanks to the residual associated to *Tank 3*, which almost instantly exceeds the threshold imposed for detection as depicted in Figure 7.20. Moreover, such leak is in part compensated by an additional water supply from *Tank 1*, which shows a lower level of water with respect to the nominal case as a consequence, not yet sufficient to trigger a related fault. At almost the same time the residual associated to *Tank 4* indicates the presence of a fault, which is related to the supply Valve V.D.17 closure. Lately, the residual related to *Tank 2* highlights the presence of a fault after $14s$ of its occurrence.

The effects of these three faults when occur separately, i.e. as single faults, has been analyzed and reported in Figure 7.21. The residual related to the leak in *Tank 3* behaves almost identically as in the multiple-fault case. On the other hand, as *Tank 2* supplies *Tank 4*, a fault in Pump 1 (supplying *Tank 2*) makes it so that *Tank 2* runs out of water very quickly due to the water demand, as depicted in Figure 7.21 (b), and hence is not longer able to supply *Tank 4*. As a consequence, an alarm is first triggered in *Tank* 2 at time $t = 103s$, and after about $56s$ also in *Tank 4*. Similarly, also the fault in Valve D.V.17 (supply valve from *Tank 2* to *Tank 4*) triggers both FDAE modules, but in this case the effect is almost simultaneous, as shown in Figure 7.21 (c). A differentiation (isolation) between the fault in Pump 1 and in Valve D.V.17 could be obtained by a more detailed analysis on the respective error signals, considering duration and amplitude.

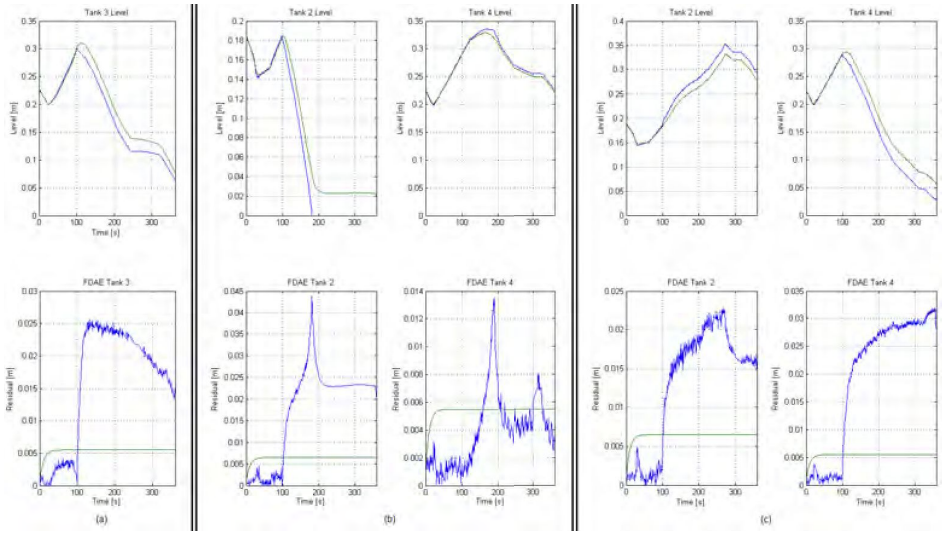

FIGURE 7.21: Faults composing the multiple fault depicted in Figure fig:MultiFaultT3P1V17 when performed separately, all @ $t = 100s$. (a) Leak fault in *Tank 3* (100% fault), (b) Pump 2 fault, (c) Valve V.D.17 fault.

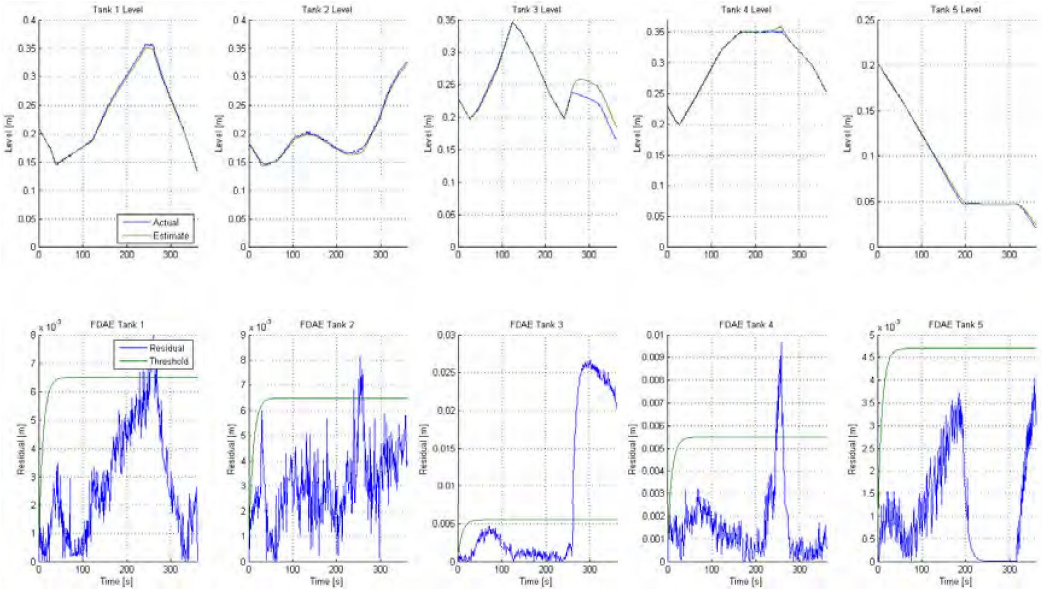

FIGURE 7.22: Multiple fault in *Tank 3*, Pump 2 and Valve V.D.17 @ $t = 260s$

Similarly, the second multiple fault considered is successfully detected almost instantaneously, as the water level in *Tank 1* decreases due to the lack of water supply, as well as in *Tank 3* due to a leak, while *Tank 2* presents a rapid increase of the water level due to the closing of Valve V.D.17.

It is worthy to notice that an increase of the residuals, which in some cases may lead to the triggering of a fault signal, can be highlighted when the level in the tanks reach low values (beneath $5cm$, especially nearby the end of the daily scenario). This

is in part due to the fact that the non-linear model previously described considers a cylindrical section with constant diameter. Such assumption is not real on the lower part of the tanks, where the section shrinks in a nonlinear way. As a consequence, the model does not perfectly reflect the dynamics of the water level in the lower part of the tanks, increasing the discrepancy between actual measurements and level estimations.

7.3 Cyber-Attacks to the Testbed

Aside physical faults, a number of cyber-attacks have been considered to take place in the local network, creating atypical and unexpected situations within the cyber-physical system. These have been performed by means of an attacker PC connected to the control network and deploying the wide number of tools provided together with Kali Linux [122], a Debian-based Linux open-source Operating System (OS) conceived for penetration testing and computer forensics science. Specifically, two main classes of attacks, previously described in Chapter 4 and sketched in Figure 7.23, have been studied and implemented:

- **Denial of Service (DoS):** attacks conceived to reduce or totally disrupt the service capabilities of the target device. The DoS attacks mainly exploit weakness related to the systems' hardware and software design. The vulnerabilities of the TCP/IP protocol often constitute the basis;
- **Man-In-The-Middle (MITM):** consists in the hijacking of the traffic generated during the communication between two hosts. The attacker places itself between the target machines, receiving all the traffic generated by the victims and conveniently forwarding the packets to the right destination.

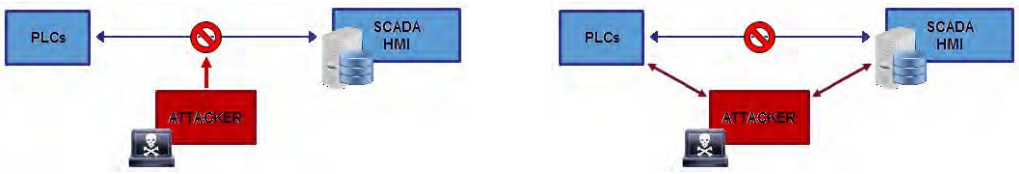

FIGURE 7.23: Denial of Service (DoS) and Man-In-The-Middle (MITM) attacks schema.

Such classification is not mutually exclusive, as some attacks carried out as MITM led to a DoS effect in the system. Therefore, these two sets of attacks may be characterized of some level of overlapping (e.g. to perform a DoS packet deletion attack, a MITM is needed).

As previously explained, by combining various data modification attacks it is possible to perform a more complex one. On one hand, the malicious agent would be prone to send fake healthy information to the operator interface, while attacking the components of the system in order to move the whole plant or a part of it to an unstable state. In this way, the operator would not be able to early recognize the hazard and, therefore, to perform the proper recovery actions, as the HMI would not trigger the required alarms and the situation would be reported as normal. The dramatic consequences of this situation have been experienced during the 2003 US blackout where, due to a fault, the data in the SCADA was not updated for a while, hence the operators neglected the

warning messages from other sources, as they considered them unreliable [123]. On the other hand, it is possible for the attacker to modify the packets so as to emulate the behavior of the system undergoing a fake attack. In this case, the operator interface would present an anomalous behavior of the system, while the plant is actually working properly. As a consequence, the operator would feel in need to perform operations for recovery from such "unstable" state or, in the worst case, to halt the operation of the system. These actually constitute undesired actions, which may drive the plant from a normal state to an unexpected or unstable one.

7.3.1 Cyber-Attacks models

Inspired by the attack models exposed in [69] and considering the system's model previously described, different kinds of attacks have been derived. Indeed, when a malicious agent gains access to the communication network, it becomes able to tamper in different ways the data exchanged between PLC and SCADA. Hence, the original signals $q(k)$ and $y(k)$ turn into attacked signals $\tilde{q}(k)$ and $\tilde{y}(k)$ (A2 and A2' attacks in Figure 7.1), respectively, or will not reach the destination device at all (A1 and A1' attacks).

Ping Flooding

The simplest way to perform a DoS attack is achieved through ping flooding. Specifically, a flooding attack indicates a methodology to disrupt the services of a machine connected to a network by forwarding to it a large number of packets as fast as possible. In the case of ping flooding, the victim is overloaded with Internet Control Message Protocol (ICMP) packets.

If the attack is performed against the PLC between time instants k_s (start of the attack) and k_e (end of the attack), the output from the plant received as input to the SCADA will be:

$$\tilde{y}(k) = \begin{cases} y(k) & \text{for } k < k_s \text{ and } k > k_e, \\ y(k-\tau) & \text{for } k_s < k < k_e \text{ if the attack introduces a delay of } \tau \text{ time steps}, \\ [\,] & \text{if the attack interrupts the communication for } k_s < k < k_e. \end{cases}$$

where [ ] indicates the lack of information due to the communication interruption, represented as A1 attack in Figure 7.1. Similarly, as depicted by the A1' case, an attack performed against the SCADA between time instants k_s and k_e, can be modeled as:

$$\tilde{q}(k) = \begin{cases} q(k) & \text{for } k < k_s \text{ and } k > k_e, \\ q(k-\tau) & \text{for } k_s < k < k_e \text{ if the attack introduces a delay of } \tau \text{ time steps}, \\ [\,] & \text{if the attack interrupts the communication for } k_s < k < k_e. \end{cases}$$

Modbus Flooding

With respect to other types of flooding, the Modbus flooding's goal is not to harm the network communication of the target device but to set out of order its elaboration capabilities. Specifically, when the attack starts, the first set of packets are received by the target device and rejected as they are not consistent with the current operation of the system (e.g., a corresponding request has not been previously received). After a short

time interval, the target device is no more able to cope with the wide number of Modbus packets received in a short time lapse, what leads to a reduction of the elaboration capabilities of the PLC/SCADA and the real commands that are received/sent by the device are no longer properly elaborated.

The specific action coded on the data field was of no actual interest, as long as it was recognized as valid by the destination component. For this reason, no differences are to be made with the ping flooding case from the model point of view.

Packet Deletion

The idea behind this attack is to disrupt the communication between a target machine and the rest of the network by intercepting and deleting all the packets originated by the device under attack. This is possible through a MITM attack that, in this specific case, results in a DoS. By defining both the source and the destination devices to be attacked, it is possible to disrupt not the entire communication service but only the data flow between two specific machines on the network.

The effect on the system depends mainly on the data carried by the specific packets that have been deleted. If the attack is performed against the packets flowing from the PLC to the SCADA (i.e., PLC set as source and SCADA as destination), and the packets contains the sensors measurements, the SCADA would not receive such information, hence the effect is modeled as:

$$\tilde{y}(k) = \begin{cases} y(k) & \text{for } k < k_s \text{ and } k > k_e, \\ [\,] & \text{for } k_s < k < k_e. \end{cases}$$

Similarly, if the deleted packets are related to the modification of the state of the actuators the subsequent required actions will not be performed by the PLC, as the actuators do not receive any command, i.e.,

$$\tilde{y}(k) = \begin{cases} q(k) & \text{for } k < k_s \text{ and } k > k_e, \\ [\,] & \text{for } k_s < k < k_e. \end{cases}$$

These two attacks can be sketched as cases A1 and A1' in Figure 7.1. By observing the model it is not possible to distinguish such attack from a DoS generated by packet flooding.

More complex attacks can be designed depending on the degree of knowledge on the system possessed by the attacker. Specifically, by introducing further information in the filter to be implemented, definite packets could be deleted, e.g. commands to a particular actuator, or incoming data from a given sensor. Evidently, the more the attack is designed to perform specific and limited actions, the more it becomes difficult to detect. As a further example, it is interesting to consider the high effectiveness of such attack when performed against specific signals, as alerts, alarms or the overcome of certain security thresholds. If such signal is intercepted and deleted, no countermeasures are going to be taken by the destination device, probably leading to undesired consequences. For instance, despite the Modbus communication is generally started by the SCADA, in case of anomaly at the lowest level of the plant, the PLC generally sends to the SCADA a request message to trigger the relative alert or alarm.

Data Modification

A malicious actor can manipulate the data field of the Modbus/TCP packets, modifying its content in various ways. If the attacker succeeds to gather the proper information about the system, it may be able to perform undesirable actions on the system, as changing the values shown on the operator interface or to modify the commands sent to the RTUs/PLCs. To do so, the attacker needs specific knowledge about both the protocol deployed on the system by the end user (more specifically the structure of the data field of the packets that travel through the network) and information about the specific plant or system to be compromised. Although such hypothesis may sound quite restrictive, partial but sufficient knowledge could be obtained, e.g., by observing and analyzing the network traffic for an adequate time lapse. For instance, data can be easily accessed and maliciously deployed by a disgruntled insider, as testified by the events that took place in Maroochy Shire (Australia) in 2000 [13], [2].

The ability to modify the actual data exchanged between PLCs and SCADA allows the attacker to perform the attack in two different ways:

- Concealing a real attack performed against the devices interacting with the environment, by modifying the data to make so the user believes that the plant is operating normally;
- Making the operator believe that the plant is undergoing some anomalies, while it is actually working properly. In this way, the operator is "encouraged" to perform actions and countermeasures in order to guarantee the safety of the system and to recover the plant operation, bringing it to an undesired state or provoking a forced shutdown.

During a Data Modification attack the communication seems normal, as the tampered packet is syntactically correct and semantically valid. Due to the MITM configuration, the packets pass through the attacker, who performs the desired data modifications, before reaching the destination device. If an operator is not able to identify such illicit transit of packets, an escalation of threats may cause major problems, mainly on the physical layer, as the attacker would be able to perform actions on the field.

As for the packet deletion attack, the effects depend on which type of packet is tampered and the target of the attack. If it carries data regarding the sensors measurements at time step k_i and is performed against the PLC, the attack model becomes:

$$\tilde{y}(k) = \begin{cases} y(k) + \Delta y(k) & \text{for } k = k_i, \\ y(k) & \text{otherwise.} \end{cases}$$

Analogously, when the attack is performed against the SCADA:

$$\tilde{q}(k) = \begin{cases} q(k) + \Delta q(k) & \text{for } k = k_i, \\ q(k) & \text{otherwise.} \end{cases}$$

Moreover, if the value of only a limited number of sensors or actuators is modified, indicated by j, the model is:

$$\tilde{y}(k_i) = \begin{bmatrix} y_1(k_i) \\ \vdots \\ y_j(k_i) + \Delta y_j(k_i) \\ \vdots \\ y_\nu(k_i) \end{bmatrix}, \qquad \tilde{q}(k_i) = \begin{bmatrix} q_1(k_i) \\ \vdots \\ q_j(k_i) + \Delta q_j(k_i) \\ \vdots \\ q_\nu(k_i) \end{bmatrix}$$

Replay Attack

A particular case of Data Modification attack is the Replay Attack, which can be divided into two main phases. Firstly, the Modbus data flow between target machines is recorded for an arbitrary time lapse. Then, it is repeatedly re-sent to the devices, substituting the actual data flow. Thereby, the plant and/or the SCADA will receive and process old information, repeating the same previous actions. If this attack is carefully studied, the operator may not notice that it is taking place, as the situation would be considered normal, e.g. the operation parameters are not out of the allowed range.

In such a case, the model for the attack having the PLC and the SCADA as target is similar to the data modification attack:

$$\tilde{y}(k) = \begin{cases} y(k) + \Delta y(k) & \text{for } k_s < k < k_e, \\ y(k) & \text{otherwise} \end{cases} \qquad \tilde{q}(k) = \begin{cases} q(k) + \Delta q(k) & \text{for } k_s < k < k_e, \\ q(k) & \text{otherwise} \end{cases}$$

where $\Delta y(k) = -y(k) + y(k-\tau)$ and $\Delta q(k) = -q(k) + q(k-\tau)$, and $y(k-\tau)$, $q(k-\tau)$ are the sensor measurements and the control signals previously recorded, respectively.

Packet Deviation

This last type of attack consists in deviating some or all the packets originated from the target machine, which are thereby sent to a false destination on the network. This approach generates a communication anomaly that results in data loss and a decay on the functionality of the system. Moreover, it is possible to deviate the packets from multiple source machines. In this case, it is possible to identify the attack as it may lead to a DoS scenario, and the communication between machines under attack would be disrupt.

As the effect of this attack is a lack of information to the destination device, it can be modeled as the previously studied packet deletion attack, and sketched as attack cases A1 and A1' in Figure 7.1.

7.3.2 Experimental Results

Extensive experimental tests have been carried out using the FACIES testbed to perform several cyber- and cyber-physical attacks. Specifically, all the five tanks have been involved in the tests considering the daily scenario already explained. Particularly, in most of the tests the attacks start at $t = 100s$, i.e., when the water demand presents a local minimum.

The most interesting analysis that has been performed is related to the cyber-physical crossed effects, particularly, how the events on the cyber domain are interpreted and/or revealed by the monitoring modules related to the physical sphere. Conversely, this allowed to analyze how the anomalous flow generated in the control network due to physical faults could be misunderstood by the IDS, as well as the effects of combined and/or, eventually, coordinated, cyber-physical faults/attacks.

Two different attacker machines have been deployed for the tests, which main features are compared in Table 7.2. As previously mentioned, the cyber-attacks have been carried out by employing the Kali Linux OS and some of its tools.

Wireshark, a free, open-source and widely deployed network protocol analyzer, used for network troubleshooting, analysis, software and communications protocol development, has been employed to capture and browse the traffic running on the

	Attacker 1	**Attacker 2**
CPU	Intel®Core^{TM} i5-3317U @1.7GHz	Intel®Core^{TM} i7-2670QM @2.20GHz
RAM	6 GB	6 GB
Network Card	Atheros AR8166 PCI-E	Controller Realtek PCIe GBE Family
Virtual Machine	VMware® Player v7.1.0	VMware® Workstation 12 Player v12.0.0
OS	Kali Linux 1.0.9a	Kali Linux 1.0.9a
Host OS	Windows® 7	Windows® 10 Pro
IP Address	Win: 84.3.251.21 Kali: 84.3.251.16	Win: 84.3.251.20 Kali: 84.3.251.14

TABLE 7.2: Attacker machines deployed for the cyber-attacks.

network during healthy and attacked operating conditions. Deploying it on a PC connected to a port in the switch configured for *mirroring*, it has been possible to perform online verification of the network state, observing all the packets and analyzing how the injected ones operated, thereby verifying the correct achievement of the attack goals. Wireshark is provided with TCP and IP filters, allowing to track only the machines under attack.

Concerning the MITM attack scenario, the Address Resolution Protocol (ARP) spoofing methodology has been employed. As known, the IP (Internet Protocol) is the standard protocol for the Internet communications and it is used for both local and remote transmissions. When an IP packet arrives at a local subnet, it is handled by the Ethernet protocol that does not identify the nodes on the network via the IP address, but through the Media Access Control (MAC) address. It represents a physical address 48 bit long, and is unique for each physical device. In order to handle the IP packets of the traffic in a local network, the Address Resolution Protocol (ARP) deals with the mapping of the IP addresses (32 bit) into the MAC addresses (48 bit) to which are referred, in order to properly deliver the packets to the target machine. The ARP protocol considers two messages:

- *ARP request:* the MAC address associated to an IP address is requested.
- *ARP reply:* a response to the ARP request is sent, and it contains the information of the MAC address associated with the requested IP.

The information related to the mapping operation from IP to MAC addresses are recorded in the ARP CACHE table. This table stores the information for a variable time period in order to optimize the network performance. The MITM attacks exploit the ARP reply receiving mechanism in order to modify the ARP CACHE of the victims. The weakness of the protocol is exploited by intercepting the communication previously to the handshake process between devices. Therefore, false ARP reply messages are sent to the two machines under attack and the MAC address of the attacker machine is inserted into the ARP replies. Hence, the packets supposed to travel between the victims will be actually sent to the attacker.

For our studies, the MITM attacks have been performed to inject false or tampered data in selected Modbus packets, or to modify the data flow between victims. To this

end, *Ettercap*, an open source suite conceived for MITM attacks and *Etterfilter*, a filter compiler able to manipulate the packets on the network with the content filtering engine, have been deployed to implement network filters for the manipulation of the packets exchanged between the victims. Such filters have been activated for a predefined time interval in order to perform the desired attack for a specific duration. The versatility of the filters used allows a considerable mastery of the ARP spoofing attack methodology.

Network Analysis

Initially, an analysis on the nominal Modbus traffic on the network has been carried out, by observing the number of bytes exchanged over time during normal operation between SCADA and PLC. To the best of our knowledge this parameter is not usually considered in the network analysis due to the unpredictable state of the communication channels. Conversely, in industrial control system the dedicated network experiences the same state during the observation window (i.e., the day cycle). During normal operation, indeed, the control processes perform some multiple reading and writing procedures in a loop fashion. The signal related to the number of bytes exchanged in the network turns to be almost periodic, as it can be appreciated in Figure 7.24, where peaks caused by an increase of the traffic data load are highlighted.

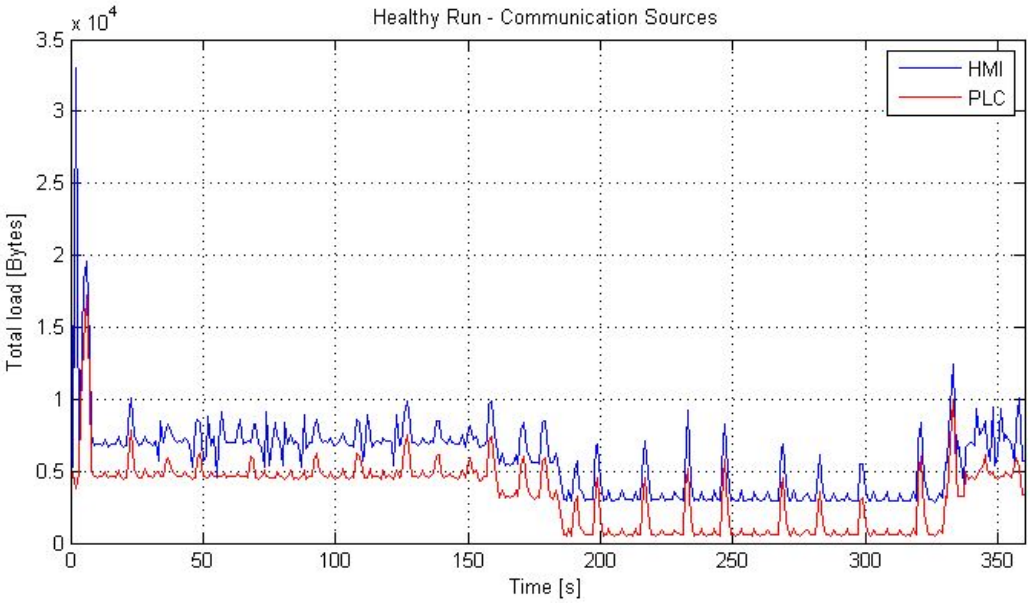

FIGURE 7.24: Nominal network load for SCADA and PLC during daily scenario.

Ping Flooding

This attack can be performed by deploying the *ping* tool, setting the desired size of the packet and other properties which is sent repeatedly to the target machine, without latency between consecutive packets. In this manner, it is possible to degrade or totally disrupt the communication between devices. It is worthy to underline that, in order to successfully perform a packet flooding attack in our scenario with only one operative

station for the attacker, it has been necessary to set to 10 Mbps the switch ports where the PLCs, SCADA and other modules were plugged.

Initially, a number of tests have been carried out to determine the percentage of network usage related to the size of the packets deployed for the ping flooding against the SCADA, considering the two different attacker machines deployed. The results are shown in Figure 7.25, where the average percentage of network utilization is shown, and it is possible to appreciate that a total disruption of the communication in the network can be obtained by setting the packets dimension to at least 8 kBytes. For lower packet sizes a delay in the communication could be appreciated. The difference between the behaviors for sizes greater than 1 kBytes is caused by the better performance of the machine deployed by *Attacker 2*. The destination device was unable to generate the same number of ping responses, hence the number of packet lost increased, and the communication was interrupted. Conversely, all the packets sent by *Attacker 1* received a ping response, thereby it was able only to degrade the communication for such packet dimensions.

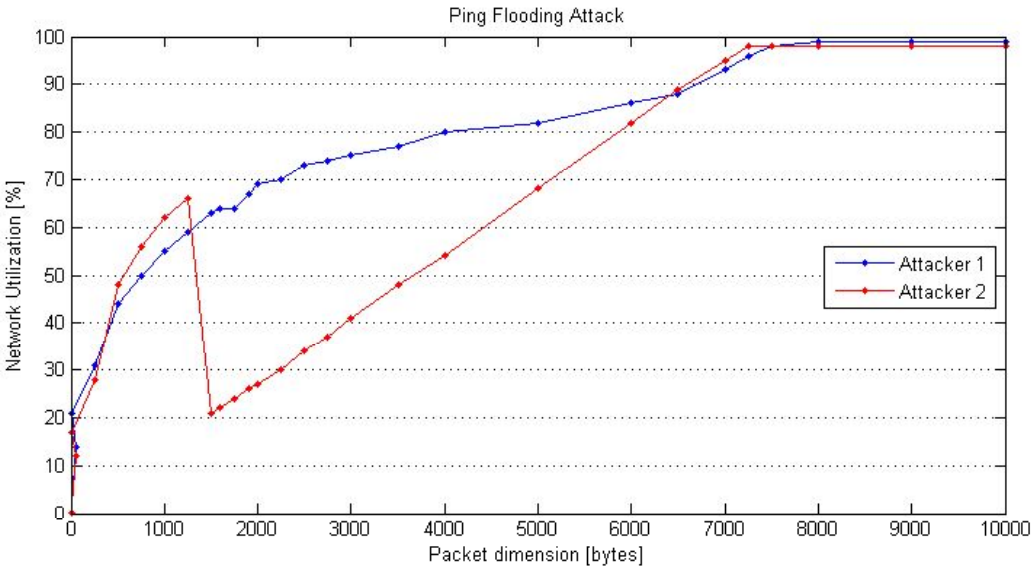

FIGURE 7.25: Average network usage during ping flooding attacks with different packet sizes.

On the other hand, an analysis has been carried out to study how the delays in the communication between PLC and SCADA due to ping flooding DoS attacks may be erroneously interpreted by the Fault Detection system as physical faults. In particular, the goal was to determine when a cyber-attack would trigger a false FDAE alarm. A number of tests at different instants of the daily scenario has been carried out, performing ping flooding DoS attacks of varying duration and evaluating the number of FDAE modules that detect an anomaly. Specifically, the attacks have been triggered at time $t = \{50, 100, 150, 200, 250, 300\}s$ of the nominal daily scenario, and lasted from $1s$ to $20s$ each. As depicted in Figure 7.26, only a subset of FDAE modules are triggered by such attack, the number increases with the duration of the attack. It is worthy to notice that the effect of the attack not only depends on the duration of the attack, but it is also strictly related to the time instant of the daily scenario at which it takes place, i.e., on the actual plant operative condition, what justifies the variation of the number of FDAE modules involved during the different attacks. Thereby, a proper detection

or identification of the cyber-attack with respect to single or multiple physical faults taking place in the system would not be always guaranteed.

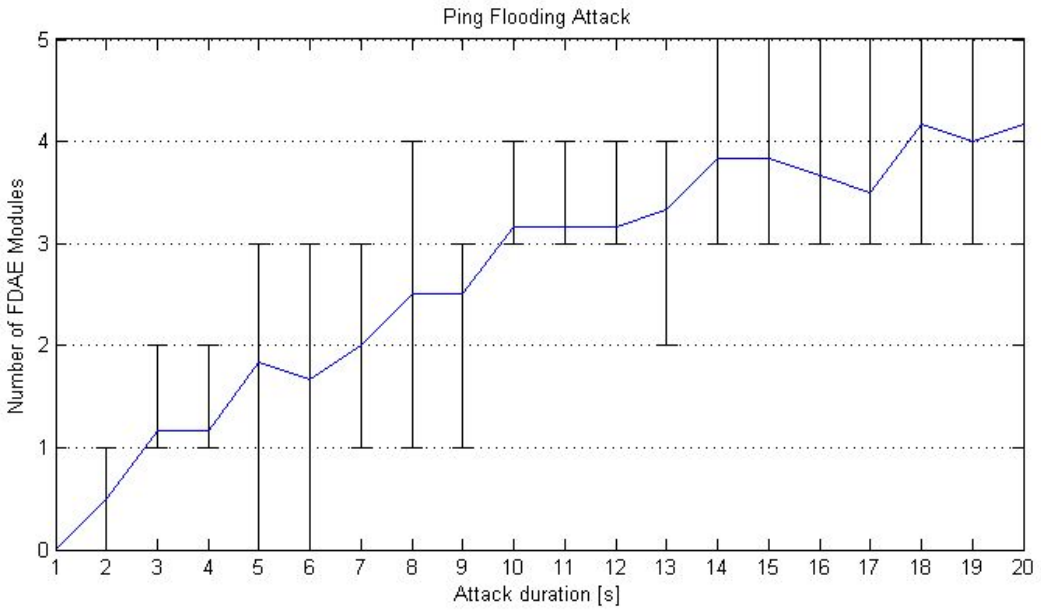

FIGURE 7.26: Average number of FDAE modules triggered by ping flooding DoS attacks tanking place at different instants of the daily scenario, with varying duration. The vertical lines consider the max and min number of FDAE modules triggered for each duration.

Figure 7.27 shows the effects of ping flooding DoS attack against the PLC, taking place at time $t = 100s$ during a healthy daily scenario, and lasting for $15s$. This type of attack compromises the communication, causing the triggering of most of the FDAE modules. This because the model is not updated, due to the interruption in the communication provoked by the attack, and depends on the specific discrepancy between the last received measurement from the sensors and the estimations of the model, which keep evolving dynamically. As can be seen, the updated values from the sensors are sensibly different from the expected data, thereby the error signal generated is a peak revealing the anomaly.

Substantially, the same results are achieved performing a ping flooding DoS attack to the SCADA system. Notice that this particular attack could be quite simple to detect as no information is obtained from the field. However, the same effect could be obtained by performing a simple Replay attack, thanks to which a given sensor measurement is re-sent as current for a desired time interval, and the timestamp and other sequential identification indexes of the packet are automatically updated, making it not trivial to detect.

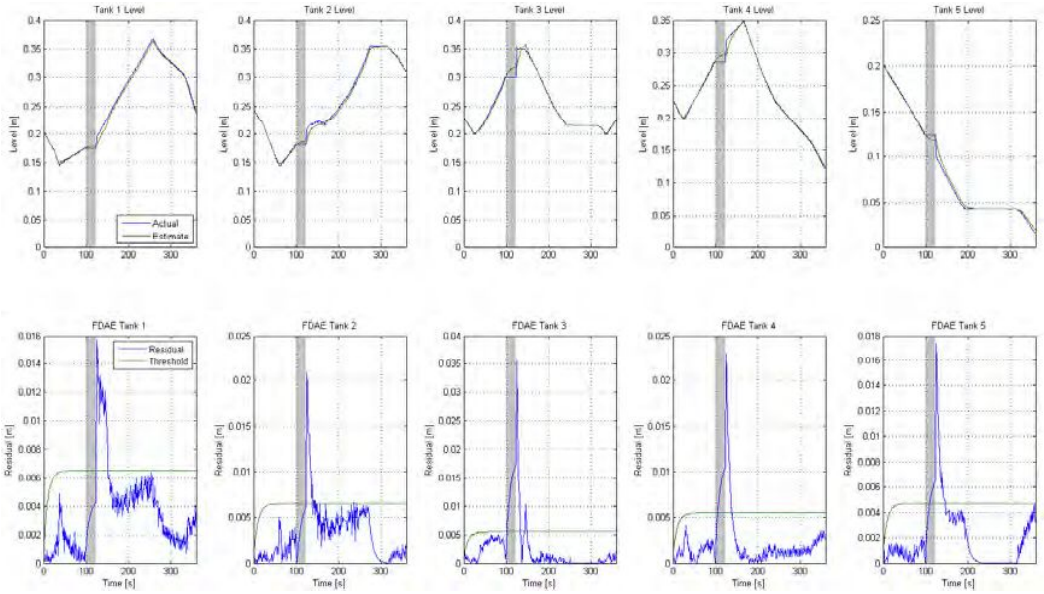

FIGURE 7.27: Ping flooding DoS attack to the PLC at $t = 100s$ for $15s$ during the nominal daily scenario. The duration of the cyber-attack is highlighted.

The concurrent presence of a cyber-attack and a physical fault is more subtle to detect. Figure 7.28 illustrates a fault taking place in *Tank 3* at time $t = 100s$ and a ping flooding DoS against the PLC starting at the same time instant and lasting $15s$. It can be seen that the physical fault influences only the variable related to the particular component involved (*Tank 3* in this case). Moreover, after the DoS stops, the communication is restored and the residuals go down to acceptable (non-faulty) values, except the one related to *Tank 3*, which remains in a faulty state, as expected.

Thereby, if the attacker is able to gain sufficient knowledge of the system and its operation, it would be possible to cover the effects of a cyber-physical attack by designing a proper combination of events and relative duration. Moreover, in many cases it would not be possible to the operator to distinguish whether the system is undergoing a cyber or a physical anomaly or attack.

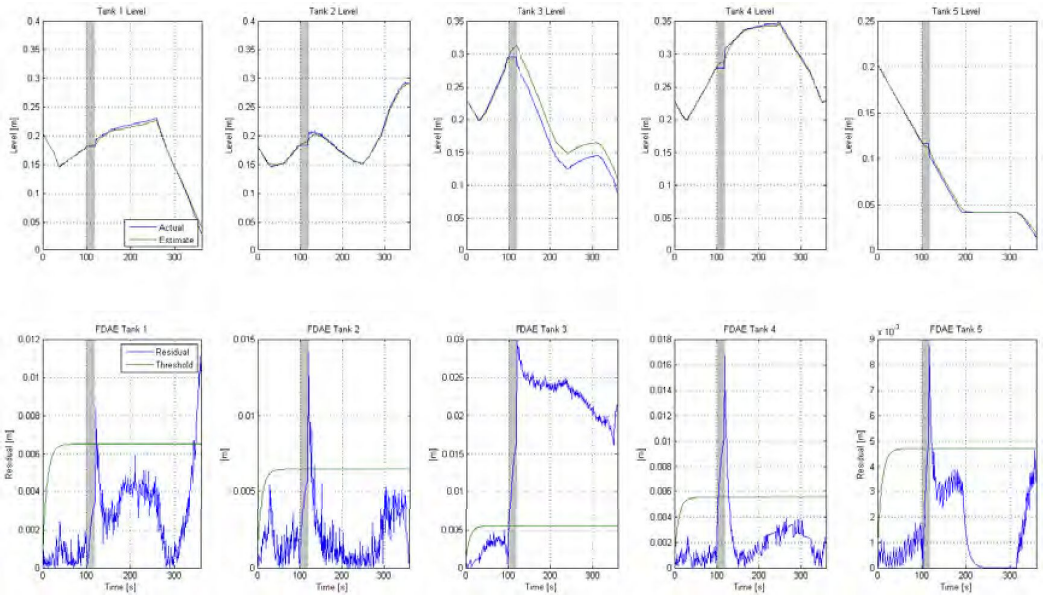

FIGURE 7.28: Ping flooding DoS attack to the PLC at $t = 100s$ for $15s$ - Leak fault in *Tank 3* @ $t = 100s$.

Modbus Flooding

Concerning the Modbus flooding, several tests have been performed by sending consecutive packets to the SCADA system and the PLC with different time delays between every single transmission. This attack has been implemented using the *nping* tool, a network packet generator, able to create packets for many protocols, including Modbus/TCP, and inject them in the communication network. A specific packet of the TCP stream has been selected using the network analyzer and has been deployed as model for the flood replication. The packet consisted in a response message from the PLC to the SCADA containing measurements from the sensors. The tests have been carried out by sending such Modbus packet repeatedly, with time delays ranging from $1000ms$ to $0.01ms$, considering the PLC IP address as (fake) source (port 502) and the SCADA as destination. As depicted in Figure 7.29, where the percentage of network utilization is plotted for different sending frequencies, such attack caused a generalized slack in the communication of the devices under attack for time delays lower than $1ms$ and a raise of the network usage to around the 37% in the single attacker worst case. Such limit is risen to almost the 67% with the contemporary use of two attacker machines. No more effects than a slight delay in the communication could be observed nor in the monitor interface, nor in the system operation, which continued to be normal. Nevertheless, it has been observed that after about $25s$ from the start of the attack and for time delays beyond $0.8ms$, problems in the communication of sensor measurements arose. Such inconvenient was caused because the SCADA was no longer able to properly send the sensor measurements to the operator interface, as it was elaborating - at a high frequency - both actual and fake Modbus replies containing sensor information.

For what concerns the packet flow between SCADA and PLC during such attack and considering the trend in Figure 7.24 as reference, a Modbus flooding attack has been performed against a PLC. Specifically, 100.000 fake Read input registers queries (i.e., requesting the sensor measurements) have been sent from the SCADA to the PLC at time $t = 100s$, with a time delay of $0.1ms$ between consecutive packets. In this case,

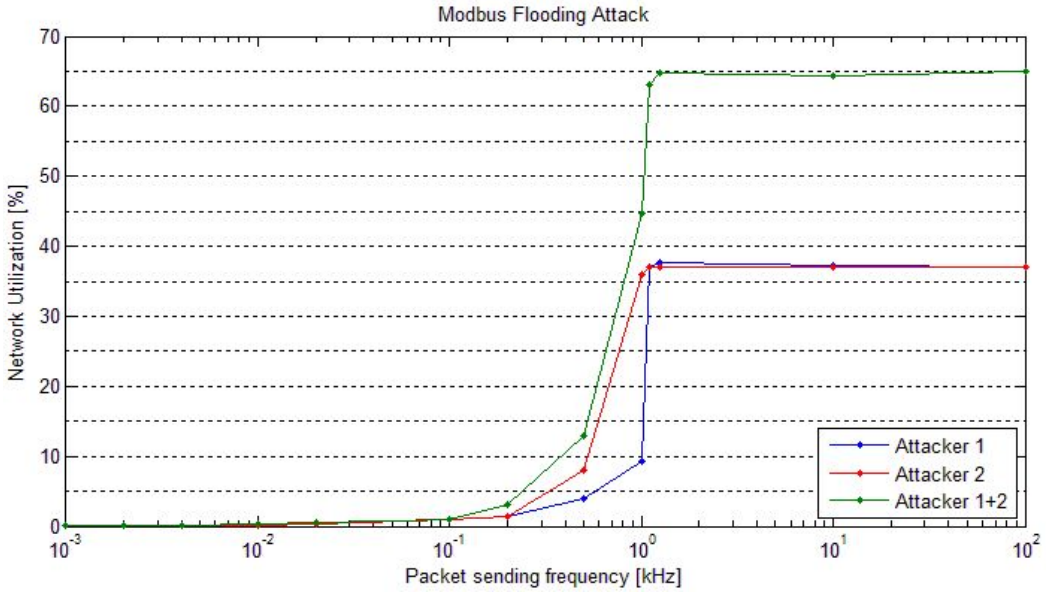

FIGURE 7.29: Network usage during Modbus flooding with different time delays, expressed as sending frequency of the packets.

the SCADA constitutes a fake source of packets, as they are actually sent by the attacker, but contain the SCADA IP as sender. As can be observed in Figure 7.30 through an external network analyzer, this provokes an instantaneous peak in the communication load both on the fake source side (SCADA), and on the PLC, which attempts to reply to every request until saturation. After the attack, which lasts $10s$, the communication is restored, but a slight delay on the PLC replies is verified.

Packet Deletion

Another DoS attack performed consiste in the disruption of the communication flow from PLC to SCADA, obtained by deleting the packets generated by the PLC and directed to the SCADA. This attack has been tested during the healthy daily scenario, starting at $t = 100s$ and ending after $15s$, respectively, and its effects in the FDAE system are shown in Figure 7.31.

It is worthy to highlight how the effects of the two specific DoS attacks depicted in Figures 7.27 and 7.31 cannot be clearly distinguished between them, as both generate very similar inconveniences from the FDAE system point of view. The lack of communication among devices is revealed by the constant value obtained for the measurements during the attack, and the residuals of no specific module but most of them exceed the imposed threshold.

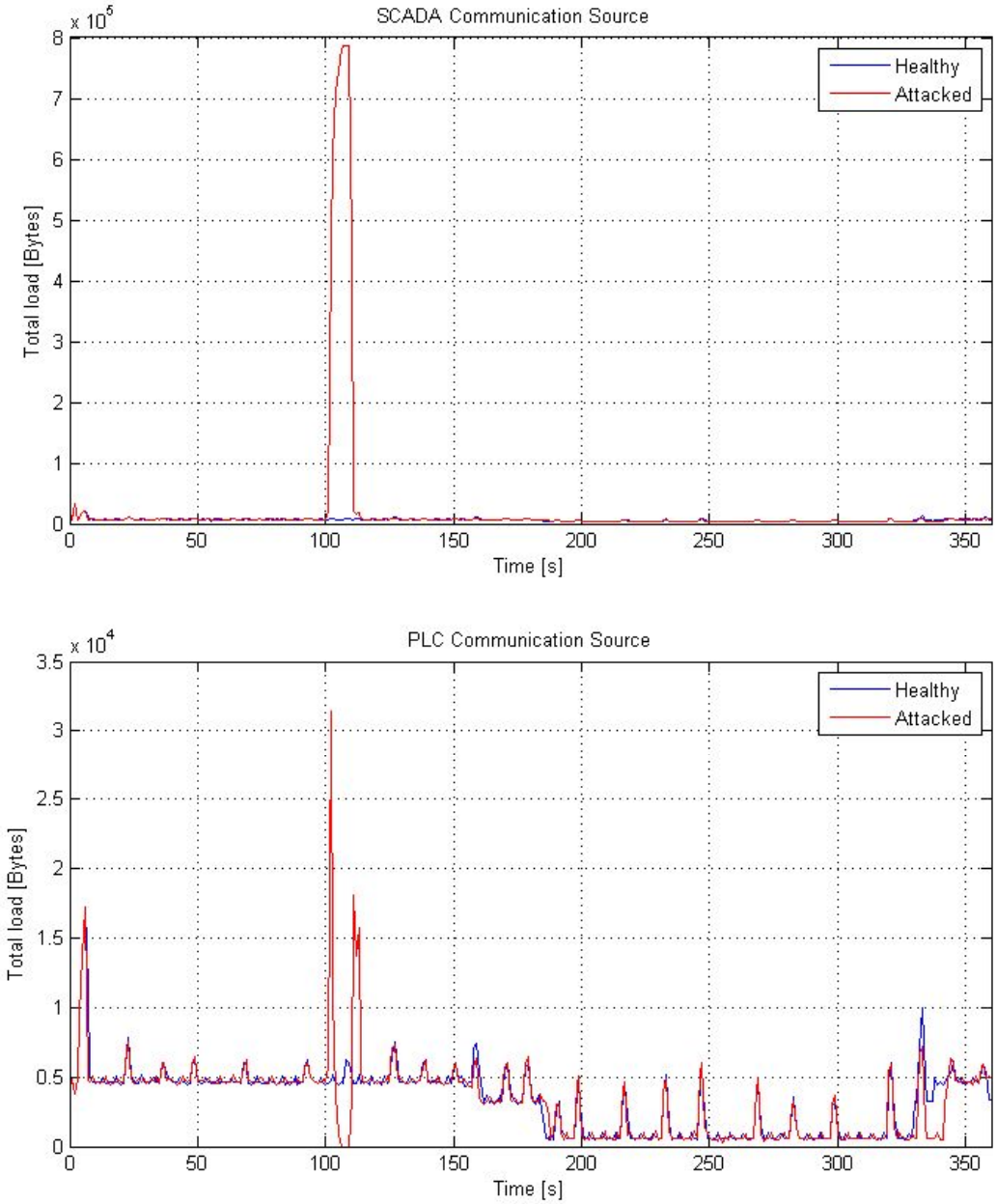

FIGURE 7.30: SCADA and PLC total load (Bytes) during daily run vs. Modbus flooding @ $t = 100s$ for $10s$.

Data Modification

In the context of the data modification attacks, three different cases have been studied.

Coils state modification: the data field of the Modbus packets has been modified to make so that the commands addressed by the SCADA to Pump 2 (id = x15) were instead performed by Pump 4 (id = x17). As these attacks are carried out by intercepting the messages sent by the SCADA before they reach the PLC (MITM attack), no anomalous behavior could be observed in the operator interface, which is actually unable to

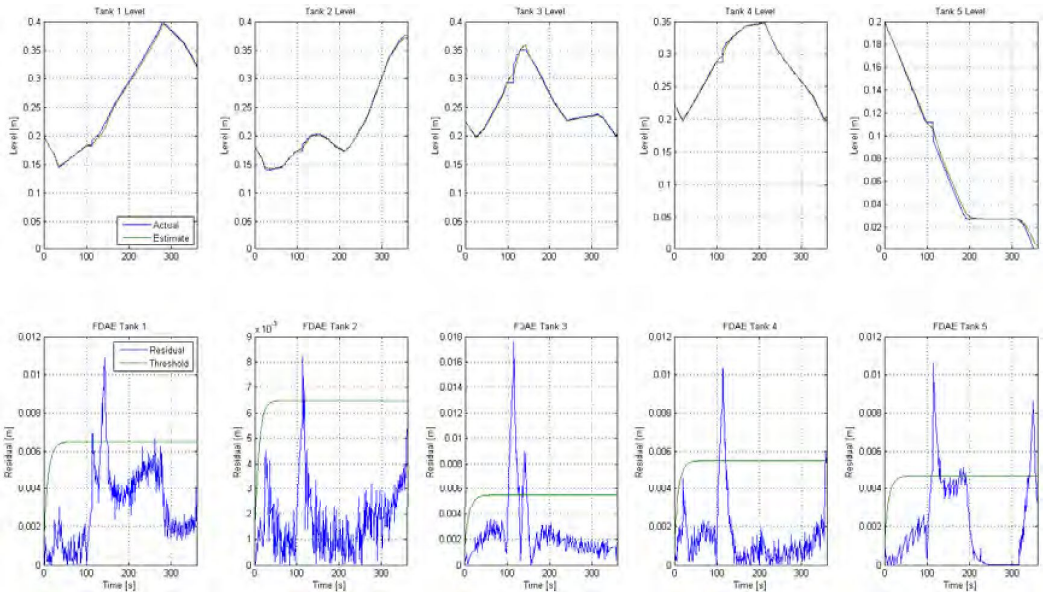

FIGURE 7.31: Packet deletion attack to the PLC at $t = 100s$ for $15s$ during the nominal daily scenario.

operate on the desired actuator. To this end, a filter has been implemented as sketched in Algorithm 8. The conditions are related to the IP source of the packets, the SCADA system in this case, and the function code and actuator to which the modification is referred. The instruction performs a substitution of the code of the target device (Pump 2) with the one corresponding to the desired one (Pump 4).

Algorithm 8 Data modification filter

```
1: while loop do
2:     C1 → IP source = target machine IP
3:     C2 → Modbus.DATA is "\x0f\x00\x15"
4:     if C1 ∧ C2 then
5:         **replace Modbus.DATA**
6:     end if
7: end while
```

Tampering the communication in this way may lead to dangerous consequences, as the operator is unable to stop the tank filling by sending the related command from the HMI, as shown by the evolution of the water level in *Tank 1* depicted in Figure 7.32, and a water overflow may occur. This has been prevented by implementing low-level security controls, as previously described, which stops the operation of the corresponding actuator when the water level measured by the sensor reaches a specific value. These are performed locally on the PLC and do not depend on the communications network.

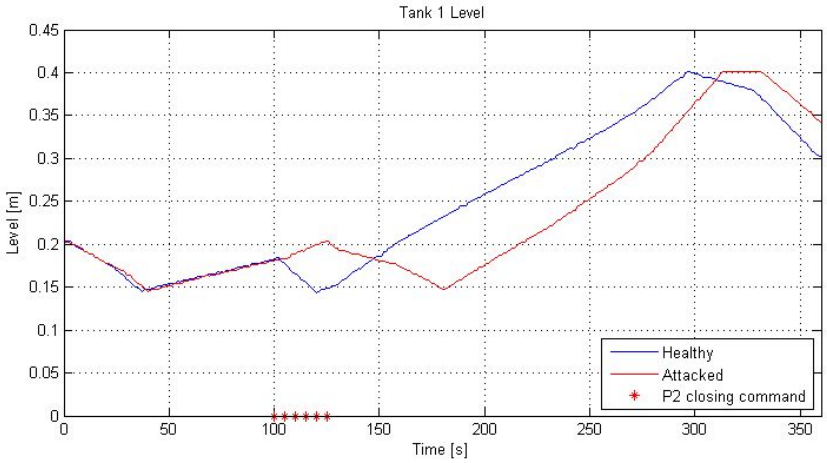

FIGURE 7.32: Data modification attack vs. SCADA @ $t = 100s$ for $25s$ during a healthy scenario - Commands to Pump 2 performed by Pump 4. The healthy and attacked conditions are compared.

On the other hand, as the monitor calculates the state estimates considering the commands sent by the SCADA, a deviation of the estimates from the actual measurements is observed in the water level of the tanks related to the attacked component, as depicted in Figure 7.33, which shows the same attack performed for $15s$ at time $t = 35s$. More specifically, the model computes the estimation considering that Pump 2 is ON, hence *Tank 1* is being supplied, while it actually does not happen (as Pump 1 is OFF due to the attack), as highlighted by the decreasing water level measured by the sensor. Thereby, the rapid deviation increase between the actual and the estimated level leads to the overcome of the fault threshold in *Tank 1*. On the other hand, as Pump 4 is erroneously turned ON by the attack, what is not expected by the model, an anomaly on the FDAE module of *Tank 5* is observed. To conclude, the emptying of *Tank 1* provokes a lack of water supply in *Tank 3*, which expected values diverge from the actual, triggering the related monitoring module.

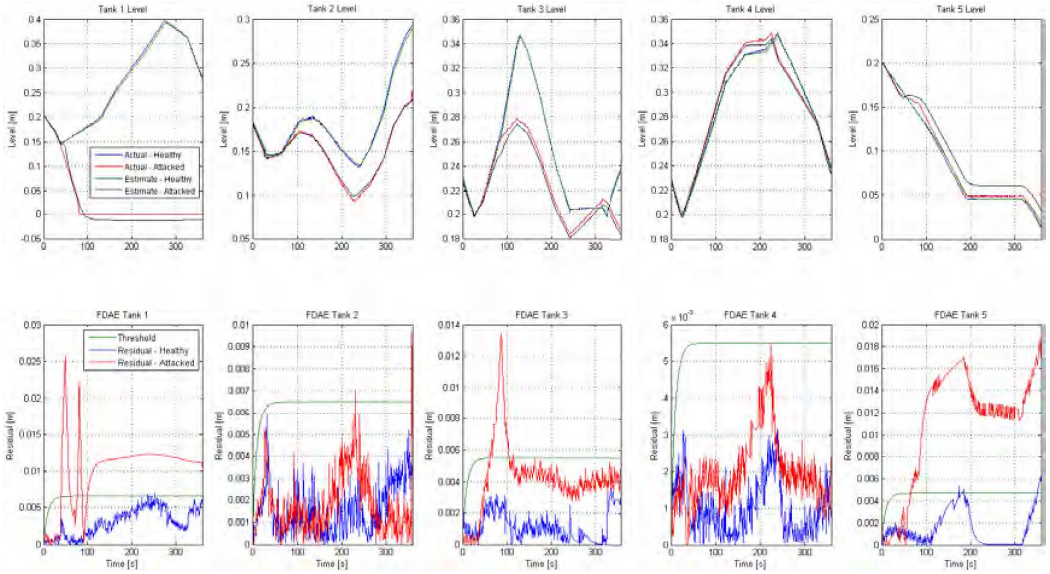

FIGURE 7.33: Data modification attack vs. SCADA @ $t = 35s$ for $15s$ during a healthy scenario - Commands to Pump 2 performed by Pump 4. The healthy and attacked conditions are compared.

Sensor measurements modification: as illustrated in Figure 7.34, the water level of *Tank 2* has been masked on the operator interface between times $t = 100s$ and $t = 130s$, reproducing an empty tank. No alarms are triggered by the monitor in such case, as it properly received the actual sensor measurements. However, the alarm module of the HMI prompts the situation to the operator, which is thereby induced to unnecessarily perform countermeasures, as a forced shutdown of the system, with potential dramatic consequences, as illustrated in the novel "Blackout" [124].

Fake exception response: some response messages from the PLC to the SCADA, have been intercepted and substituted by an exception response, as depicted in Figure 7.35. To such end, the hex value 0x80 has been added to the function code field, and the data was composed by the chosen exception code (01 - Illegal function; 02 - Illegal data address; 03 - Illegal data value; 06 - Slave device busy). On the plant side no effects are due to the attack, as the operation requested by the SCADA is properly processed and carried out by the PLC.

As required by the Modbus protocol, the request packets are automatically re-sent, and after a while an error message is triggered on the operator interface, containing the relative exception code.

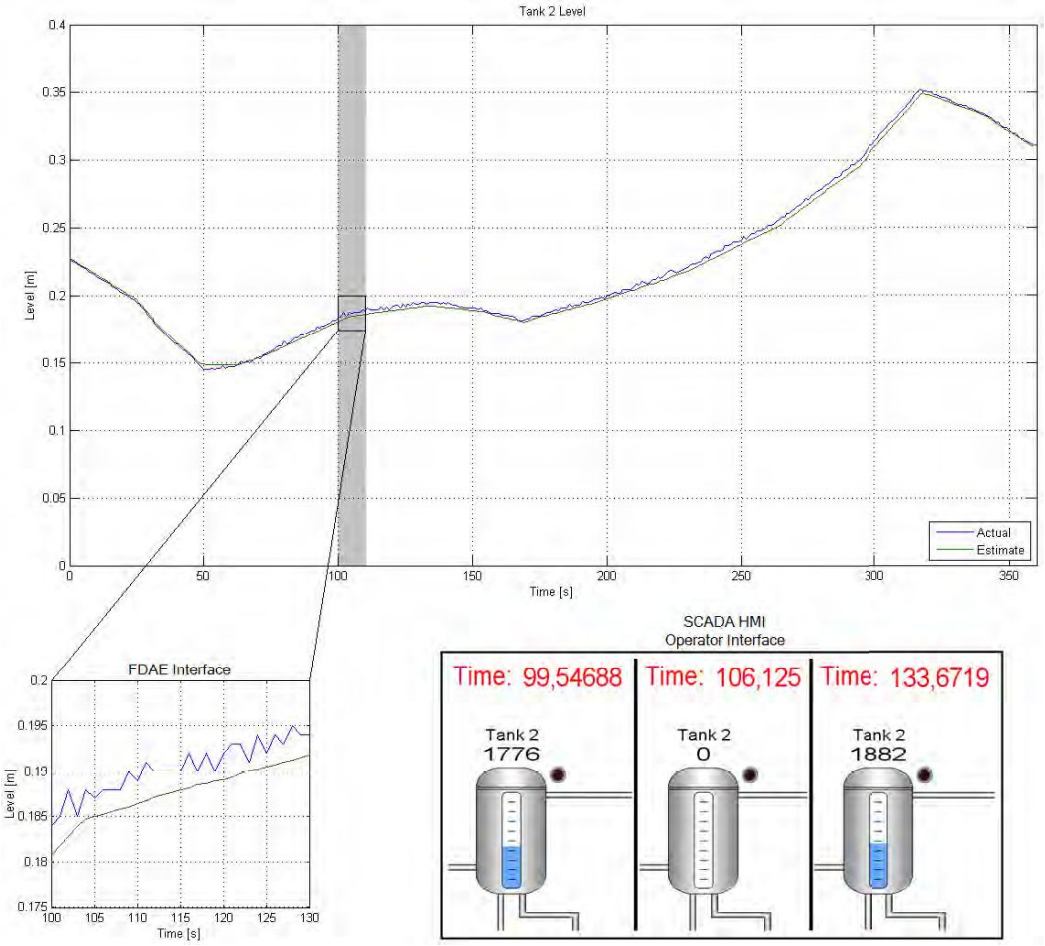

FIGURE 7.34: Data modification attack vs. SCADA @ $t = 100s$ for $30s$ during healthy scenario - Sensor measurements sent to the SCADA are modified with a 0 value, while the physical system is operating properly, as shown on the monitor interface.

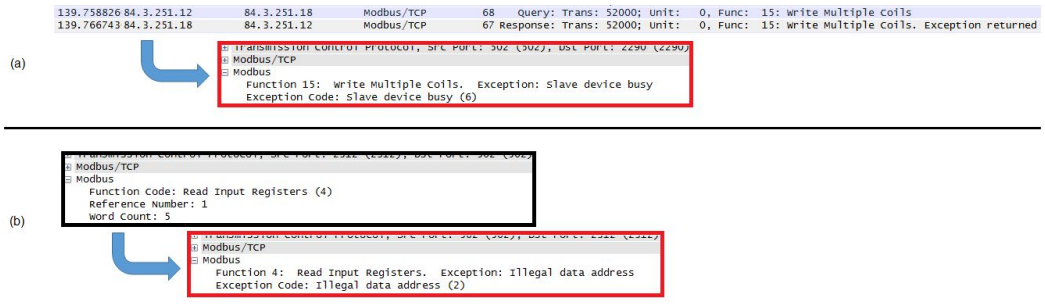

FIGURE 7.35: Data modification attack - Fake exception response - (a) Write multiple coils - (b) Read input registers.

Replay Attack

The values of the water level measurements of *Tank 3* received by the SCADA have been substituted with a previous value for $20s$, starting at time $t = 100s$, by changing the hex value in the data field of the Modbus packet related to the respective register. Thereby, the data modification method has been exploited to perform a replay attack, as previously mentioned. During the attack, the tampered values were visible on the operator and monitor interfaces, and after its end the actual measurements returned to be visible in the HMI. As depicted in Figure 7.36, the attack is revealed by the FDAE module due to the noticeable difference between the fake measurements and the estimations provided by the model.

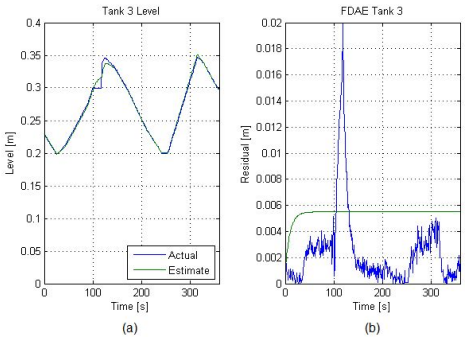

FIGURE 7.36: Replay attack @ $t = 100s$ for $20s$ during healthy scenario. The difference between the expected values and the tampered measurements (a) provoke an instant response of the monitoring system (b).

As previously mentioned, the replay attack is obtained by performing a data modification attack, where the actual measurements and commands are opportunely substituted with values previously recorded by the attacker. Due to the high complexity of this attack, the repetition of the last measured value from the sensors has been successfully implemented, as depicted in Figure 7.36. Aside the similarity with the results obtained for the ping flooding DoS attack, in this case the measurements of only one tank has been compromised and, what is more important, the timestamp of the data received from the PLC is updated, i.e. the values obtained are "current", as it is not obtained by the interruption of communications as in the DoS case.

It is worthy to highlight that during the various data modification attacks it has been verified that different performances were obtained depending on the characteristics of the specific machine used by the attacker. Specifically, Attacker 1 (Table 7.2) was not always able to successfully perform the sensor measurements modification attacks, as an interruption of the communication took place, probably due to insufficient computational capabilities or poorly performing network interface. Conversely, Attacker 2, equipped with more performant resources, was always successful in modifying the Modbus packets' payload without incurring in communication interruptions. Thereby, the attacker has to take particular care on the choice of the devices to exploit, in order to guarantee the achievement of its goals.

A/N. The contributions and results exposed in this Chapter have been published in:

E. Etchevés Miciolino, S. Panzieri, F. Pascucci, M.M. Polycarpou, R. Setola. *A Testbed for Analysis of Cyber-Physical Faults and Attacks in Water Critical Infrastructures*. Control Engineering Practice (CEP), 2015. (*Submitted*)

G. Bernieri, **E. Etchevés Miciolino**, F. Pascucci, R. Setola. *Monitoring System Reaction in Cyber-Physical Testbed under Cyber-Attacks*. Transactions on Control of Network Systems: Special Issue on Secure Control Cyber Physical Systems, 2015. (*Submitted*)

E. Etchevés Miciolino, G. Bernieri, F. Pascucci, R. Setola. *Communications Network Analysis in a SCADA System Testbed under Cyber-Attacks*. TELFOR 2015 – 23rd Telecommunications Forum - 24th-25th November 2015, Belgrade (Serbia). (2015)

C. Heracleous, **E. Etchevés Miciolino**, R. Setola, F. Pascucci, D.G. Eliades, G. Ellinas, C.G. Panayiotou, M.M. Polycarpou. *Critical Infrastructure Online Fault Detection: Application in Water Supply Systems*. CRITIS 2014 – 9th International Conference on Critical Information Infrastructures Security - 13rd-15th October 2014, Limassol (Cyprus). (2014)

E. Etchevés Miciolino, R. Setola, F. Pascucci, J. Lopez, M.M. Polycarpou. *FACIES: a Testbed for Distributed Fault and Attack Identification in Interdependent Critical Infrastructures*. 2nd International SCADA LAB Workshop, 28th May 2014, Seville (Spain). (2014)

Conclusions

8.1 Concluding Remarks and Future Developments

The technological evolution that took place in the last decades allowed to offer new and improved services to the population, most of which constitute the fundamentals of modern societies. Their malfunction, damage or disruption may cause important economical and social losses, consequences that in some cases may turn to be catastrophic, reason why these have been defined *Critical Infrastructures*.

Thanks to the high automation level that characterize such infrastructures multiple advantages have been obtained, reached as a result of the deep melt of the physical and cyber domains. Nevertheless, a wide number of vulnerabilities are revealing, and the high complexity of these systems makes it challenging to effectively and efficiently improve protection, security and eventual countermeasures. Such vulnerabilities arose at every level of the Critical Infrastructures, and a wide diversity of threats arrive from unavoidable errors and physical faults, malicious attackers, human errors, terrorist cells, natural disasters and environmental changes, among others. As a consequence, the security and protection of Critical Infrastructures and the related Automatic Industrial Control Systems are a challenging hot topic for research.

In consideration of that, the purpose of this dissertation thesis was to firstly provide an overview on these systems, their importance, structure, evolution and main risks and vulnerabilities. The study of these complex and large-scale cyber-physical systems was then split in the two main domains. Initially, an analysis of the state-of-the-art of techniques for the diagnosis of physical faults in the process has been carried out. It is then followed by the study of the cyber factors threatening these systems, and how the security problem should be addressed based on both the peculiarities that characterize Industrial Control Systems and their similarities to classic Information and Communications Technologies.

As the simulation of these systems and the proposed security measures provides useful but very restricted results and solutions, researchers are focusing on the development of emulated environments, in which most of (if not the whole) system is physically created, i.e. field components, controllers, communication network, monitoring and supervisory devices, etc. Despite the great step forward represented by the "real" data obtained from this emulators and the possibility to actually perform faults and attacks tests, analyzing the system's response, they certainly constitute a trade-off between scalability, approximations, feasibility and costs.

With this aim, a testbed emulating the water system of a small city has been developed and was here largely described, composed of several of its main constituting parts, including the communication network, the control system and the monitoring and supervisory system. In addition, an efficient Fault Diagnosis technique has been implemented and validated, thanks to which it has been possible to detect and isolate a wide number of cyber-physical faults and attacks induced to the system. The experimental tests comprised a wide number of independent physical faults and different

types of cyber attacks at various levels, and the results shown for which conditions these could be detected and isolated. In addition, their combinations have been considered, demonstrating how it is possible for an attacker to hide the effects on one domain by acting on the other one, e.g., how the corruption of the physical operations could be kept unnoticed and unknown to the operators in the Control Centre, and vice versa.

Despite the efforts of the entire scientific community, utility holders and the government, a long is still to go to reach optimal, reliable and secure solutions, goal that is considered by many almost impossible to actually reach. Specifically for the water domain, cyber-related standards have been developed and private companies are working to minimize their risk of exposure to cyber-related incidents and events. Indeed, public-private partnership models with government and industry stakeholders are striving to implement meaningful cyber protocols and protections so to secure the served communities. This efforts have come with a high cost, as they require both financial and human capital to implement the necessary security measures.

What mostly arises from this study is the need of deeper awareness on the tight relation between the cyber and physical domains in these complex systems, and that the studies and improvements on one sphere cannot neglect the impacts and implications on the other one. Better results could be obtained considering the overall system, oppositely as the current approaches which tend to completely separate the cyber from the physical fields.

The testbed presented and the results here provided constitute a good starting point that requires further developments and analysis. A wider network is to be created to increase the possible scenarios, for example by simulating its dynamics through specialized software (e.g. EPANET), and other physical components are to be added to the system, as flow and pressure meters.

On the other hand, considering that the main assumption for the cyber domain was that the attacker had already gained access to the system, a more sophisticated architecture could be considered (e.g., by employing firewalls, properly configuring switches, etc.), and more secure protocols could be exploited (e.g., OPC UA). The system's security could also be enhanced through the use of authentication and messages encryption, which are totally missing in the testbed here described (as are unfortunately missing in a large number of real ICS systems). Nevertheless, timing is critical in most SCADA systems, what makes the implementation of such techniques nontrivial and, for some specific context, these requirements are actually completely limitating. Moreover, the types of tested cyber-attacks are to be extended, considering the ability of an attacker of exploiting the cyber-physical configuration to conceal its presence. To conclude, the interdependencies and domino effects of the water system with other Critical Infrastructures are to be better studied, so as to provide a wider overview on the situation, whether normal or atypical, and provide useful data for the decision-making process continuously carried out by the operators.

8.2 Summary of the Publications

The following papers have been published (or submitted for publication) as a result of the studies presented in this dissertation:

Papers in international journals

E. Etchevés Miciolino, S. Panzieri, F. Pascucci, M.M. Polycarpou, R. Setola. *A Testbed for Analysis of Cyber-Physical Faults and Attacks in Water Critical Infrastructures*. Control Engineering Practice (CEP), 2015. (*Submitted*)

G. Bernieri, **E. Etchevés Miciolino**, F. Pascucci, R. Setola. *Monitoring System Reaction in Cyber-Physical Testbed under Cyber-Attacks*. Transactions on Control of Network Systems: Special Issue on Secure Control Cyber Physical Systems, 2015. (*Submitted*)

L. Cazorla, **E. Etchevés Miciolino**, C. Alcaraz, R. Setola, J. Lopez, F. De Cillis. *Injection-based Stealth Attacks in Critical Infrastructures*. Journal of Computer and Systems Sciences (JCSS), 2015. (*Submitted*)

G. Oliva, **E. Etchevés Miciolino**, R. Setola. *Distributed Opinion Dynamics with Heterogeneous Reliability*. International Journal of System of Systems Engineering (IJSSE), vol. 4, N. 3/4, pp. 277-290, ISSN: 1748-068X, 2013.

Papers in international conference proceedings

E. Etchevés Miciolino, G. Bernieri, F. Pascucci, R. Setola. *Communications Network Analysis in a SCADA System Testbed under Cyber-Attacks*. TELFOR 2015 – 23rd Telecommunications Forum - 24th-25th November 2015, Belgrade (Serbia). (2015)

C. Heracleous, **E. Etchevés Miciolino**, R. Setola, F. Pascucci, D.G. Eliades, G. Ellinas, C.G. Panayiotou, M.M. Polycarpou. *Critical Infrastructure Online Fault Detection: Application in Water Supply Systems*. CRITIS 2014 – 9th International Conference on Critical Information Infrastructures Security - 13rd-15th October 2014, Limassol (Cyprus). (2014)

E. Etchevés Miciolino, R. Setola, F. Pascucci, J. Lopez, M.M. Polycarpou. *FACIES: a Testbed for Distributed Fault and Attack Identification in Interdependent Critical Infrastructures*. 2nd International SCADA LAB Workshop, 28th May 2014, Seville (Spain). (2014)

C. Alcaraz, **E. Etchevés Miciolino**, S. Wolthusen. *Structural Controllability of Networks for Non-Interactive Adversarial Vertex Removal*. In Critical Information Infrastructures Security – 8th International Workshop, CRITIS 13 – 16th-18th September 2013, Amsterdam (The Nederlands), Lecture Notes in Computer Science, vol. 8328, pp. 120-132, Springer, ISBN: 978-3-319-03963. (2013)

C. Alcaraz, **E. Etchevés Miciolino**, S. Wolthusen. *Multi-Round Attacks on Structural Controllability Properties for Non-Complete Random Graphs*. In Proceedings of the 16th International Conference of Information Security (ISC 2013), 13rd-15th November 2013, Dallas (TX – USA), Lecture Notes in Computer Science, vol. 7807, pp. 140-151, Springer, ISBN: 978-3-319-27658-8. (2015)

G. Oliva, **E. Etchevés Miciolino**, R. Setola. *Criticality Assessment via Opinion Dynamics*. In Proceedings of DHSS 2012 - International Defense and Homeland Security Simulation Workshop - 19th-21st September 2012, Vienna (Austria). pp. 120-132, ISBN: 978-88-97999-08-9. (2012)

Acknowledgements

This dissertation has been supported in part by the Prevention, Preparedness and Consequence Management of Terrorism and other Security-related Risks Programme European Commission – Directorate – General Home Affairs, under the EU project FACIES (HOME/2011/CIPS/AG/4000002115).

Bibliography

[1] USA Patriot Act. *Public Law 107-56*. 2001.

[2] S. Bologna and R. Setola. "The Need to Improve Local Self-Awareness in CIP/CIIP". In: *First IEEE International Workshop on Critical Infrastructure Protection*. IEEE, 2005. DOI: 10.1109/IWCIP.2005.19.

[3] J. Moteff and P. Parfomak. "Critical Infrastructure and Key Assets: Definition and Identification". In: *Congressional Research Service*. 2004.

[4] U.S. Government. *The Physical Protection of Critical Infrastructures and Key Assets*. Last Access on July 2013. The White House, Washington, 2003.

[5] European Commission. *Green Paper on a European Programme for Critical Infrastructure Protection*. COM(2005) 576 final, Brussels, 2005.

[6] S.M. Rinaldi, J. Peerenboom, and T. Kelly. "Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies". In: *IEEE Control Systems* 21.6 (2001), pp. 11–25. DOI: 10.1109/37.969131.

[7] S.M. Rinaldi. "Modeling and Simulating Critical Infrastructures and Their Interdependencies". In: *Proceedings of the 37th Annual Hawaii International Conference on System Sciences, 2004*. IEEE, 2004. DOI: 10.1109/HICSS.2004.1265180.

[8] Tyson Macaulay. *Critical Infrastructure: Understanding its Component Parts, Vulnerabilities, Operating Risk, and Interdependencies*. CRC Press - Taylor & Francis Group, 2009.

[9] S. Amin et al. "Cyber Security of Water SCADA Systems - Part I: Analysis and Experimentation of Stealthy Deception Attacks". In: *IEEE Transactions on Control Systems Technology* 21.5 (2013), pp. 1963–1970. DOI: 10.1109/TCST.2012.2211873.

[10] M. Tabesh et al. "Assessing Pipe Failure Rate and Mechanical Reliability of Water Distribution Networks Using Data-Driven Modeling". In: *Journal of Hydroinformatics* 11.1 (2009), pp. 1–17.

[11] M. Islam et al. "Water Distribution System Failure: a Framework for Forensic Analysis". In: *Environment Systems and Decisions* 34.1 (2014), pp. 168–179. DOI: 10.1007/s10669-013-9464-3.

[12] A. Zaccone. *UpsideDown Project Magazine*. English. Direzione Generale Ambiente, Energia e Sviluppo Sostenibile - Regione Lombardia. 2014.

[13] J. Slay and M. Miller. "Lessons Learned from the Maroochy Water Breach". In: ed. by E. Goetz and S. Shenoi. Vol. 253. IFIP International Federation for Information Processing. Springer US, 2008. Chap. Critical Infrastructure Protection - Part II, pp. 73–82. DOI: 10.1007/978-0-387-75462-8_6.

[14] ICS-CERT. *Incident Response Summary - Report 2009-2011*. Tech. rep. Industrial Control Systems Cyber Emergency Response Team (ICS-CERT), 2012.

[15] E. Kyriakides and M.M. Polycarpou. *Intelligent Monitoring, Control, and Security of Critical Infrastructure Systems*. Ed. by E. Kyriakides and M.M. Polycarpou. 1st ed. Vol. 565. Studies in Computational Intelligence 1. Springer-Verlag Berlin Heidelberg, 2015. DOI: 10.1007/978-3-662-44160-2.

[16] *http://www.cordis.europa.eu/project/rcn/92603/en.html*.

[17] C. Siaterlis, A. Perez Garcia, and B. Genge. "On the Use of Emulab Testbeds for Scientifically Rigorous Experiments". In: *IEEE Communications Surveys & Tutorials* 15.2 (2012), pp. 929–942. DOI: 10.1109/SURV.2012.0601112.00185.

[18] A. Linn. "SCADA Test Bed - Water Tank System". paper 473. Capstone Experience Thesis Project. University of Illinoys Honors College.

[19] E. Laubwald. *Coupled Tanks Systems 1*. Tech. rep. www.control-systems-principles.co.uk.

[20] K.H. Johansson. "The Quadruple-Tank Process: a Multivariable Laboratory Process With an Adjustable Zero". In: *IEEE Transactions on Control Systems Technology* 8.3 (2002), pp. 456–465. DOI: 10.1109/87.845876.

[21] C. Illes, G.N. Popa, and I. Filip. "Water Level Control System Using PLC and Wireless Sensors". In: *9th IEEE ICCC, International Conference on Computational Cybernetics*. IEEE, 2013, pp. 195–199. DOI: 10.1109/ICCCyb.2013.6617587.

[22] N. Orani, A. Pisano, and E. Usai. "Fault Diagnosis for the Vertical Three-Tank System via High-Order Sliding-Mode Observation". In: *Journal of the Franklin Institute, special issue Advances in Nonlinear Observation and Identification for Dynamic Systems* 347.6 (2010), pp. 923–939. DOI: 10.1016/j.jfranklin.2009.11.010.

[23] J. Lemos Nabais, L.F. Mendonca, and M. Ayala Botto. "A Multi-Agent Architecture for Diagnosing Simultaneous Faults Along Water Canals". In: *Control Engineering Practice* 31 (2014), pp. 92–106. DOI: 10.1016/j.conengprac.2013.08.015.

[24] A.J. Whittle et al. "WaterWiSe@SG: A Testbed for Continuous Monitoring of the Water Distribution System in Singapore". In: *Water Distribution Systems Analysis 2010*. Ed. by American Society of Civil Engineers. Water Distribution System Modeling Issues. 2010, pp. 1362–1378. DOI: 10.1061/41203(425)122.

[25] TecQuipment. *CE105 or CE105MV - Coupled Tanks Apparatus*. URL: http://www.tecquipment.com/Control/Control-Engineering/CE105.aspx.

[26] AMIRA. *Three-Tank-System DTS200, Practical Instructions*. Tech. rep. AMIRA, 1996.

[27] C. Alcaraz, G. Fernandez, and F. Carvajal. "Security Aspects of SCADA and DCS Environments". In: ed. by Javier Lopez, Roberto Setola, and Stephen D. Wolthusen. Vol. 7130. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 120–149. DOI: 10.1007/978-3-642-28920-0_7.

[28] M. Smith. "Web-Based Monitoring & Control for OilGas Industry". In: *SCADA's Next Step Forward, Pipeline & Gas Journal* 228.3 (2001).

[29] J. James, J. Graham, and A. Leger. *Gap Analysis for Survivable PCS*. Research Report 14. United States Military Academy, 2009.

[30] E. Byres et al. "Worlds in Collision: Ethernet on the Plant Floor". In: *Proceeding of the ISA Emerging Technologies Conference*. 2002.

[31] J. Pollet. "Developing a Solid SCADA Security Strategy". In: *2nd ISA/IEEE Sensors for Industry Conference*. 2002, pp. 148–156.

[32] C. Alcaraz and S. Zeadally. "Critical Control System Protection in the 21st Century: Threats and Solutions". In: *IEEE Computer* 46.10 (2013), pp. 74–83. DOI: 10.1109/MC.2013.69.

[33] C. Alcaraz and J. Lopez. "Wide-Area Situational Awareness for Critical Infrastructure Protection". In: *IEEE Computer* 46.4 (2013), pp. 30–37. DOI: 10.1109/MC.2013.72.

[34] R. Chandia et al. "Security Strategies ofr SCADA Networks". In: ed. by E. Goetz and S. Shenoi. Vol. 253. IFIP International Federation for Information Processing. Springer Heidelberg, 2008. Chap. Critical Infrastructure Protection III, pp. 117–131. DOI: 10.1007/978-0-387-75462-8_9.

[35] Homeland Security Council. *National Strategy for Homeland Security*. Tech. rep. The White House, Washington DC, 2007.

[36] U.S. Department of Homeland Security. *National Infrastructure Protection Plan - Water Sector*. Tech. rep. URL: https://www.dhs.gov/xlibrary/assets/nipp-ssp-water-2010.pdf.

[37] G. Cembrano et al. "Optimal Control of a Water Distribution Network in a Supervisory Control System". In: *Control Engineering Practice* 8.10 (2000), pp. 1177–1188. DOI: 10.1016/S0967-0661(00)00058-7.

[38] *http://www.risidata.con/index.php/news/2013_report_on_control_system_cyber_security_incidents_rel*

[39] *http://www.reuters.com/article/us-cybersecurity-attack-idUSTRE7AH2C320111121*.

[40] *http://www.networkworld.com/article/2188264/malware-cybercrime/dhs–america-s-water-and-power-utilities-under-daily-cyber-attack.html*.

[41] *https://www.epa.gov/waterresilience*.

[42] K. Stouffer, J. Falco, and K. Scarfone. *NIST SP 800-82 - Guide to Industrial Control Systems (ICS) Security*. National Institution of Standards and Technology (NIST). URL: http://csrc.nist.gov/publications/nistpubs/800-82/SP800-82-final.pdf.

[43] Eric Luiijf. *SCADA Security Good Practices per il settore delle acque potabili - TNO Report*. Franco Angeli.

[44] *http://www.repubblica.it/2005/g/sezioni/cronaca/allertaitalia1/allacqua/allacqua.html*.

[45] *Bastano Due Uomini e 7 Mila Euro per Avvelenare Milano* (2015).

[46] R. Pérez et al. "Leak Localization in Water Networks: A Model-Based Methodology Using Pressure Sensors Applied to a Real Network in Barcelona". In: *IEEE Control Systems Magazine* 34.4 (2014), pp. 24–36. DOI: 10.1109/MCS.2014.2320336.

[47] R. Isermann. *Fault-Diagnosis Systems - An Introduction from Fault Detection to Fault Tolerance*. 1. Springer Berlin Heidelberg, 2006, p. 475. DOI: 10.1007/3-540-30368-5.

[48] R. Isermann. *Fault-Diagnosis Applications - Model-Based Condition Monitoring: Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems*. 1. Springer Berlin Heidelberg, 2011, p. 354. DOI: 10.1007/978-3-642-12767-0.

[49] S.X. Ding. *Model-Based Fault Dignosis Techniques. Design Schemes, Algorithms and Tools*. 2nd ed. Advances in Industrial Control. Springer London, 2013. DOI: 10.1007/978-1-4471-4799-2.

[50] V. Venkatasubramanian et al. "A Review of Process Fault Detection and Diagnosis. Part I: Quantitative Model-Based Methods". In: *Computers & Chemical Engineering* 27.3 (2003), pp. 293–311. DOI: 10.1016/S0098-1354(02)00160-6.

[51] R. Isermann. "Model-Based Fault-Detection and Diagnosis - Status and Applications". In: *Annual Reviews in Control* 29.1 (2005), pp. 71–85. DOI: 10.1016/j.arcontrol.2004.12.002.

[52] V. Venkatasubramanian, R. Rengaswamy, and S.N. Kavuri. "A Review of Process Fault Detection and Diagnosis. Part II: Qualitative Models and Search Strategies". In: *Computers & Chemical Engineering* 27.3 (2003), pp. 313–326. DOI: 10.1016/S0098-1354(02)00161-8.

[53] R.J. Patton, F.J. Uppal, and C.J. Lopez-Toribio. "Soft Computing Approaches to Fault Diagnosis for Dynamic Systems: a Survey". In: *Proceedings of the 4th IFAC Symposium SAFEPROCESS00*. Elsevier, 2000, pp. 298–311.

[54] R. Isermann. "Fault Diagnosis of Machines via Parameter Estimation and Knowledge Processing - Tutorial Paper". In: *Automatica* 29.4 (1993), pp. 815–835. DOI: 10.1016/0005-1098(93)90088-B.

[55] J. Chen and R.J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Ed. by K.Y. Cai. Vol. 3. The Kluwer Internation Series on Asian Studies in Computer and Information Science. Beijing, China: Springer US, 1999. DOI: 10.1007/978-1-4615-5149-2.

[56] R.J. Patton, P.M. Frank, and R.N. Clark. *Issues of Fault Diagnosis for Dynamic Systems*. 1. Springer-Verlag London, 2000. DOI: 10.1007/978-1-4471-3644-6.

[57] J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Electrical Engineering and Electronics. CRC Press, 1998.

[58] R.K. Mehra and J. Peschon. "An Innovations Approach to Fault Detection and Diagnosis in Dynamic Systems". In: *Automatica* 7.5 (1971), pp. 637–640. DOI: 10.1016/0005-1098(71)90028-8.

[59] F. Caccavale et al. *Control and Monitoring of Chemical Batch Reactors*. Advances in Industrial Control 1. Springer-Verlag London, 2010. DOI: 10.1007/978-0-85729-195-0.

[60] R. Ferrari, T. Parisini, and M.M. Polycarpou. "Distributed Fault Detection and Isolation of Large-Scale Discrete-Time Nonlinear Systems: An Adaptive Approximation Approach". In: *IEEE Transactions on Automatic Control* 57.2 (2012), pp. 275–290. DOI: 10.1109/TAC.2011.2164734.

[61] E.D. Knapp. *Industrial Network Security - Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*. Ed. by A. Ward and M. Cater. Syngress, 2011.

[62] N. Falliere, L.O. Murchu, and E. Chien. *W32. Stuxnet Dossier*. Tech. rep. 1.4. Symantec, 2011.

[63] Symantec Security Response. *W32.Duqu - The Precursor to the Next Stuxnet*. Tech. rep. 1.4. Symantec Security Response, 2011.

[64] Kaspersky Lab. *The Duqu 2.0 - Technical Details*. Tech. rep. 2.1. Kaspersky, 2015.

[65] *AGA Report No.12 - Cryptographic Protection of SCADA Communications*. American Gas Association, 2006.

[66] *1711-2010 - IEEE Trial-Use Standard for a Cryptographic Protocol for Cyber Security of Substation Serial Links*. IEEE.

[67] *ISA99 - Industrial Automation and Control Systems Security*. ISA. URL: https://www.isa.org/isa99/.

[68] L. Cazorla et al. "Injection-based Stealth Attacks in Critical Infrastructures". In: *Submitted to Journal of Computer and System Sciences* (2015).

[69] F. Pasqualetti, F. Dörfler, and F. Bullo. "Control-Theoretic Methods for Cyber-physical Security - Geometric Principles for Optimal Cross-Layer Resilient Control Systems". In: *IEEE Control Systems Magazine* 35.1 (2015), pp. 110–127. DOI: 10.1109/MCS.2014.2364725.

[70] C. Meyers, S. Powers, and D. Faissol. *Taxonomies of Cyber Adversaries and Attacks: a Survey of Incidents and Approaches*. Technical Report. U.S. Department of Energy, 2009. DOI: 10.2172/967712.

[71] A.A. Cárdenas et al. "Attacks Against Process Control Systems: Risk Assessment, Detection, and Response". In: *ASIACCS '11 - Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. Hong Kong, China: ACM New York, 2011, pp. 355–366. DOI: 10.1145/1966913.1966959.

[72] H. Sandberg, A. Teixeira, and K.H. Johansson. "On Security Indices for State Estimators in Power Networks". In: *First Workshop on Secure Control Systems (CPSWEEK 2010)*. 2010.

[73] O. Vukovic et al. "Network-Layer Protection Schemes Against Stealth Attacks on State Estimators in Power Systems". In: *2011 IEEE International Conference on Smart Grid Communications*. IEEE, 2011, pp. 184–189. DOI: 10.1109/SmartGridComm.2011.6102314.

[74] M. Jakobsson, S. Wetzel, and B. Yener. "Stealth Attacks on Ad-Hoc Wireless Networks". In: *IEEE 58th Vehicular Technology Conference (VTC 2003-Fall)*. Vol. 3. IEEE, 2003, pp. 2103–2111. DOI: 10.1109/VETECF.2003.1285396.

[75] M. Esmalifalak et al. "Stealth False Data Injection Using Independent Component Analysis in Smart Grid". In: *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2011, pp. 244–248. DOI: 10.1109/SmartGridComm.2011.6102326.

[76] Y. Liu, P. Ning, and M. Reiter. "False Data Injection Attacks Against State Estimation in Electric Power Grids". In: *16th ACM Conference on Computer and Communications Security (CCS '09)*. 2009, pp. 21–32. DOI: 10.1145/1653662.1653666.

[77] A. Teixeira et al. "A Cyber Security Study of a SCADA Energy Management System: Stealthy Deception Attacks on the State Estimator". In: *18th IFAC World Congress* (2011).

[78] Q. Yang et al. "On False Data-Injection Attacks against Power System State Estimation: Modeling and Countermeasures". In: *IEEE Transactions on Parallel & Distributed Systems* 25.3 (2014), pp. 717–729. DOI: 10.1109/TPDS.2013.92.

[79] A. Teixeira et al. "Cyber-Security Analysis of State Estimators in Electric Power Systems". In: *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 5991–5998. DOI: 10.1109/CDC.2010.5717318.

[80] M.MA. Rahman and E. Al-Shaer. "A Formal Model for Verifying Stealthy Attacks on State Estimation in Power Grids". In: *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. Vancouver, BC: IEEE, 2013, pp. 414–419. DOI: 10.1109/SmartGridComm.2013.6687993.

[81] Z.H. Yu and W.L. Chin. "Blind False Data Injection Attack Using PCA Approximation Method in Smart Grid". In: *IEEE Transactions on Smart Grid* 6.3 (2015), pp. 1219–1226. DOI: 10.1109/TSG.2014.2382714.

[82] A.N. Bishop and A.V. Savkin. "On False-Data Attacks in Robust Multi-Sensor-Based Estimation". In: *9th IEEE International Conference on Control and Automation (ICCA), 2011*. Santiago: IEEE, 2011, pp. 10–17. DOI: 10.1109/ICCA.2011.6137928.

[83] G. Hug and J.A. Giampapa. "Vulnerability Assessment of AC State Estimation With Respect to False Data Injection Cyber-Attacks". In: *IEEE Transactions on Smart Grid* 3.3 (2012), pp. 1362–1370. DOI: 10.1109/TSG.2012.2195338.

[84] A. Teixeira et al. "Revealing Stealthy Attacks in Control Systems". In: *50th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2012*. Monticello, IL: IEEE, 2012, pp. 1806–1813. DOI: 10.1109/Allerton.2012.6483441.

[85] R.B. Bobba et al. "Detecting False Data Injection Attacks on DC State Estimation". In: *Preprints of the First Workshop on Secure Control Systems, CPSWEEK*. 2010.

[86] A. Giani et al. "Smart Grid Data Integrity Attacks: Characterizations and Countermeasures". In: *IEEE International Conference on Smart Grid Communications (SmartGridComm), 2011*. IEEE, 2011, pp. 232–237. DOI: 10.1109/SmartGridComm.2011.6102324.

[87] O. Kosut et al. "On Malicious Data Attacks on Power System State Estimation". In: *45th International IEEE Universities Power Engineering Conference (UPEC)*. 2010, pp. 1–6.

[88] O. Kosut et al. "Limiting False Data Attacks on Power System State Estimation". In: *2010 Conference Information Sciences and Systems*. 2010.

[89] S. Wang and W. Ren. "Stealthy False Data Injection Attacks against State Estimation in Power Systems: Switching Network Topologies". In: *American Control Conference (ACC), 2014*. Portland, OR: IEEE, 2014, pp. 1572–1577. DOI: 10.1109/ACC.2014.6858904.

[90] S. Wang and W. Ren. "Stealthy Attacks in Power Systems: Limitations on Manipulating the Estimation Deviations Caused by Switching Network Topologies". In: *2014 IEEE 53rd Annual Conference on Decision and Control (CDC)*. Los Angeles, CA: IEEE, 2014, pp. 217–222. DOI: 10.1109/CDC.2014.7039384.

[91] G. Dan and H. Sandberg. "Stealth Attacks and Protection Schemes for State Estimators in Power Systems". In: *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2010, pp. 214–219.

[92] S. Amin et al. "Stealthy Deception Attacks on Water SCADA Systems". In: *HSCC '10 - Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control*. Stockholm, Sweden, 2010, pp. 161–170. DOI: 10.1145/1755952.1755976.

[93] A. Monticelli. *State Estimation in Electric Power Systems: a Generalized Approach*. 1. Springer US, 1999. DOI: 10.1007/978-1-4615-4999-4.

[94] R.E. Kalman. "Mathematical Description of Linear Dynamical Systems". In: *Journal of the Society of Industrial and Applied Mathematics Control Series A* 1 (1963), pp. 152–192.

[95] C.T. Lin. "Structual Controllability". In: *IEEE Transactions on Automatic Control* 19.3 (1974), pp. 201–208.

[96] Y.Y. Liu, J.J. Slotine, and A.L. Barabási. "Controllability of Complex Networks". In: *Nature* 473 (2011), pp. 167–173.

[97] W.X. Wang et al. "Optimizing Controllability of Complex Networks by Minimum Structural Perturbations". In: *Physical Review E* 85.2 (2012).

[98] C.L. Pu, W.J. Pei, and A. Machaelson. "Robustness Analysis of Network Controllability". In: *Physica A: Statistical Mechanics and its Applications* 391.18 (2012), pp. 4420–4425.

[99] T.W. Haynes et al. "Domination in Graphs Applied to Electric Power Networks". In: *SIAM Journal on Discrete Mathematics* 15.4 (2002), pp. 519–529.

[100] J. Kneis et al. "Parameterized Power Domination Complexity". In: *Information Processing Letters* 98.4 (2006), pp. 145–149.

[101] P. Erdös and A. Rényi. "On Random Graphs I". In: *Publicationes Mathematicae* 6 (1959), pp. 290–297.

[102] P. Erdös and A. Rényi. "On the Evolution of Random Graphs". In: *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* (1960), pp. 17–61.

[103] B. Bollobás. "Random Graphs". In: *Cambridge Studies in Advanced Mathematics - 2nd Edition* 73 (2001).

[104] D.J. Watts and S.H. Strogatz. "Collective Dynamics of 'Small-World' Networks". In: *Nature* 393 (1998), pp. 440–442.

[105] D.J. Watts. "Small Worlds: the Dynamics of Networks Between Order and Randomness". In: *Princeton Studies in Complexity* (2003).

[106] A. Barrat and M. Weigt. "On the Properties of Small-World Network Models". In: *The European Physical Journal B - Condensed Matter and Complex Systems* 13.3 (2000), pp. 547–560.

[107] R. Albert and A.L. Barabási. "Statistical Mechanics of Complex Networks". In: *Reviews of Modern Physics* 74.1 (2002), pp. 47–97.

[108] C.R. Palmer and J.G. Steffan. "Generating Network Topologies That Obey Power Laws". In: *Proceedings of the 2000 IEEE Global Telecommunications Conference (GLOBECOM '00)*. Vol. 1. IEEE, 2000, pp. 434–438.

[109] R. Cohen, S. Havlin, and D.B. Avraham. "Structural Properties of Scale-Free Networks". In: ed. by S. Bornholdt and H.G. Schuster. Wiley-VCH, Weinheim, Germany, 2005. Chap. Handbook of Graphs and Networks: From the Genome to the Internet.

[110] E. Etchevés Miciolino. "Analysis of Dynamic Decision Models in Complex Networks". MA thesis. University Campus Bio-Medico of Rome, 2012.

[111] R. Setola, S. De Porcellinis, and M. Sforna. "Critical Infrastructure Dependency Assessment Using the Input-Output Inoperability Model". In: *International Journal of Critical Infrastructure Protection* 2.4 (2009), pp. 170–178. DOI: 10.1016/j.ijcip.2009.09.002.

[112] United Nations Development Program. *Average Water Use per Person per Day*. URL: http://www.data360.org/dsg.aspx?Data_Set_Group_Id=757.

[113] USGS Water Science School. *Industrial Water Use*. 2005. URL: http://water.usgs.gov/edu/wuin.html.

[114] M. Herrera et al. "Predictive Models for Forecasting Hourly Urban Water Demand". In: *Journal of Hydrology* 387.1–2 (2010), pp. 141–150. DOI: 10.1016/j.jhydrol.2010.04.005.

[115] S. De Porcellinis, G. Oliva, and R. Setola. "A Holistic-Reductionistic Approach for Modeling Interdependencies". In: ed. by C. Palmer and S. Shenoi. Vol. 311. IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, 2009. Chap. Critical Infrastructure Protection III, pp. 251–227. DOI: 10.1007/978-3-642-04798-5_15.

[116] G. Oliva, S. Panzieri, and R. Setola. "Distributed Synchronization Under Uncertainty: A Fuzzy Approach". In: *Fuzzy Sets and Systems* 206 (2012), pp. 103–120. DOI: doi:10.1016/j.fss.2012.02.003.

[117] Modbus Organization Inc. *Modbus Application Protocol Specification v1.1b3*. Modbus Organization Inc. 2012.

[118] Modbus Organization Inc. *Modbus Messaging on TCP/IP Implementation Guide v.1.0b*. Modbus Organization Inc. 2006.

[119] X. Zhang, M.M. Polycarpou, and T. Parisini. "A Robust Detection and Isolation Scheme for Abrupt and Incipient Faults in Nonlinear Systems". In: *IEEE Transactions on Automatic Control* 47.4 (2002), pp. 576–593. DOI: 10.1109/9.995036.

[120] V. Reppa, M.M. Polycarpou, and C.G. Panayiotou. "Distributed Sensor Fault Diagnosis for a Network of Interconnected Cyber-Physical Systems". In: *IEEE Transactions on Control of Network Systems* 2.1 (2015), pp. 11–23. DOI: 10.1109/TCNS.2014.2367362.

[121] R.M.G. Ferrari, T. Parisini, and M.M. Polycarpou. "A Fault Detection and Isolation Scheme for Nonlinear Uncertain Discrete-Time Systems". In: *46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 1009–1014. DOI: 10.1109/CDC.2007.4434848.

[122] *https://www.kali.org/*.

[123] U.S.-Canada Power System Outage Task Force. *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*. Final Report. 2004. URL: http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/BlackoutFinal-Web.pdf.

[124] Marc Elsberg. *Blackout*. Nord, 2013.