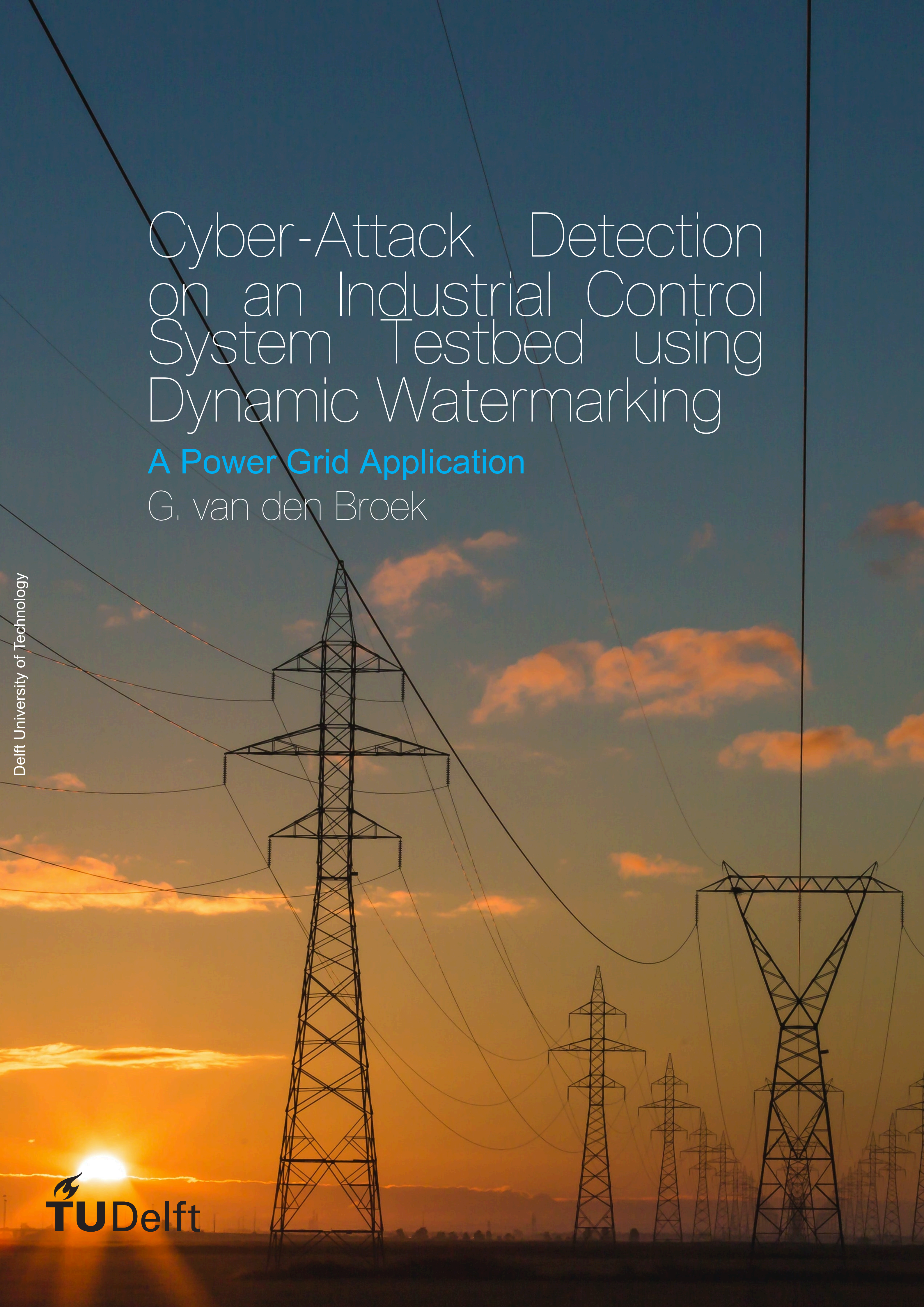
Cyber-Attack Detection on an Industrial Control System Testbed using Dynamic Watermarking

A Power Grid Application

G. van den Broek

Delft University of Technology



Cyber-Attack Detection on an Industrial Control System Testbed using Dynamic Watermarking

A Power Grid Application

by

G. van den Broek

to obtain the degree of Master of Science
of Systems and Control at the Delft University of Technology,
to be defended publicly on Thursday August 25, 2022 at 10:00 AM.

Student number:	4380541	
Project duration:	September 1, 2021 – August 25, 2022	
Thesis committee:	Dr. R. M. G. Ferrari,	TU Delft, supervisor
	Ir. T. Keijzer,	TU Delft, supervisor

Cover image is “Power pylons at sunset” by Matthew Henry on Unsplash.

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Abstract

An Industrial Control System (ICS) is used to monitor and control industrial processes and critical infrastructure, and is therefore crucial to modern society. This makes them attractive targets for malicious cyber-attacks, which have become more advanced and abundant in recent history. To properly defend ICSs from these cyber-attacks, appropriate cyber-defensive mechanisms should be continuously designed and updated, cyber-attack detection mechanisms included. These mechanisms should undergo sufficient testing before being implemented in actual ICSs to minimise unforeseen consequences. Existing literature indicates that Dynamic Multiplicative Watermarking (DMWM) is a promising form of cyber-attack detection, which could improve overall detection performance. Thus far, this technique has not yet been applied to Automatic Generation Control (AGC) (a prominent form of Load Frequency Control (LFC) in power grids) to detect data integrity attacks (specifically scaling and replay attacks).

Ergo, this research aims at testing the performance of DMWM against data integrity attacks on AGC. To perform attack detection, a Luenberger observer it utilised. This observer generates a residual, which is compared to a robustly designed threshold. For the purpose of adequate testing, the HILDA (Hardware-In-the-Loop Detection of Attacks) testbed is designed and constructed. By using this testbed, more realistic scenarios can be simulated than with regular desktop simulations. After verifying the correct construction of the testbed, the DMWM performance is examined both on a desktop simulation environment using `MATLAB & Simulink`, and on the HILDA testbed. It is shown that the addition of DMWM increases the detection performance in the context of both scaling and replay attacks. For replay attacks, this performance increases notably, while for scaling attacks the improvement is more modest. It is shown that, overall, the attacks are detected more quickly when simulated on the HILDA testbed compared to simulations performed on the `MATLAB & Simulink` environment. On the other hand, the overall detection ratio was better when simulated on the `MATLAB & Simulink` environment. This discrepancy in detection performance demonstrates the added value of the HILDA testbed.

”Don’t hope that events will turn out the way you want,

welcome events in whichever way they happen;

this is the path to peace.”

— Epictetus, Enchiridion

Preface

Combining the worlds of information and operational technology: for me, the thesis would have to revolve around this highly pressing topic (or ”something with music”). Dr. Ferrari was kind enough to inform me that, coincidentally, another MSc Systems and Control student (Ir. V. Ranade) was writing his thesis on an industrial control system testbed (i.e. operational technology), which was specifically designed to be subjected to cyber-attacks (i.e. information technology). If I so desired, I would be the student further designing, constructing and utilising this testbed. And so it happened.

As a student of Systems and Control, the theoretical topic on industrial control systems was inside my comfort zone. However, practically building a testbed and involving information technology were not. Exploring these subjects was highly challenging and entertaining at the same time. In the end, I hope to have done a sufficiently successful job in developing the HILDA testbed which, together with her sisters, will provide a platform for plentiful more research. One solemn remark: Due to the ’special military operation’ in Ukraine, this research has obscurely become more relevant than it already was one year back in time. Hopefully the developments regarding cyber-attacks will not outperform those of cyber-security measures in the ever-lasting cat-and-mouse game between the two.

I would like to sincerely thank my head supervisor, Dr. R. M. G. Ferrari. Despite teaching courses, conducting research, attending conferences, and supervising PhD candidates and countless other thesis students, he managed to provide wise guidance in this project. During meetings, we enjoyed talking about the HILDA testbed so much that we often lost track of time. I would also like to express my appreciation towards my daily supervisor, Ir. T. Keijzer. Sometimes literally on a daily basis, we would squabble about the progress and game plan of this project. Despite busy or difficult periods he encountered himself, he was always up for providing feedback, without which this thesis would not have been the same. The joint coffee breaks were welcome intermezzos as well. On top op that, I wish to thank the lab personnel, W.H.A. Wien, W.J.M. van Geest and D. Kromm, who offered their help more often than was previously acknowledged. Last but not least, allow me to convey my gratitude towards my family and friends, whom never lost faith in the prosperous completion of this project.

Geert van den Broek
July 14, 2022

Contents

List of Figures

List of Tables

1 Introduction

An Industrial Control System (ICS) is used to monitor and control processes across the whole spectrum of industry, including critical infrastructure [145], even on a global scale [210]. Examples of ICS domains are manufacturing, energy production and distribution, (waste)water processing, and transportation. As a consequence of their principal role in society, failure of these systems can have devastating economical, social and even fatal consequences [193]. This makes them attractive targets for malicious attackers, cyber-attackers included. According to the World Economic Forum, threats to cybersecurity belong to the top five of global risks confronting worldwide nations [97]. To illustrate the imminent danger of cyber-attacks on ICSs, a handful of cyber incidents is provided in Table 1.1, which is just a microscopic selection of all cyber-attacks conducted on ICSs in the last few decades [85, 93, 133, 134, 135, 159, 179]. Though the 'Stuxnet' cyber-attack is the most widely known, it is not the first, and definitely was not the last: in fact, the overall amount of cyber-attacks on ICSs is growing [97]. The Global Risk Report 2018 stresses that the ability to manage hostile events is developing, but that the cyber-attack capabilities are developing at an even higher pace [54].

#	Ref.	Year	Attack	Impact
1	[188]	2000	A former employee hacked the control system of the Maroochy Shire sewage treatment plant.	800 kilolitres of untreated sewage water were released in a nearby river.
2	[50] [115]	2010	The infamous Stuxnet worm infected the Iranian Natanz nuclear-enrichment facility, where it covertly attacked the controller of the rotational speed of the uranium enrichment centrifuges.	The rotational speed was successfully increased, destroying the enrichment facility.
3	[117] [120]	2015	A foreign attacker remotely controlled the distribution management system of an electricity company in Ukraine, as part of a larger coordinated attack on Ukrainian electricity distribution.	Approximately 225.000 customers lost power in multiple regions.
4	[135]	2020	An attack on globally distributed Honda factories with ransomware designed to disrupt ICSs.	Honda had to freeze the entire global production.

Table 1.1: Timeline of a selection of industrial control system cyber-incidents

The consequences of cyber-attacks can be minimised when attacks are detected accordingly [193]. In other words, adequate detection mechanisms should be implemented in the form of an Intrusion Detection System (IDS). The amount of theoretical contributions on these systems has been abundant (see for example [64], which is a 'survey of surveys' on cybersecurity). However, in practice, these detection techniques fall short, as is for example shown in [207] for healthcare infrastructure. This indicates that theoretical contributions alone are not sufficient in protecting ICSs from cyber-attacks. Conceivably, the main reason for this insufficiency is that only few of the theoretical contributions are actually implemented in real ICSs, which is a consequence of inadequate verification and validation [121, 195]. This is because performing tests on live ICSs is not possible, due to the potentially devastating impact of a wrongful experiment [11]. Hence, to nevertheless achieve sufficient testing, some form of testbed (either a simulation or a physical scaled down version of an ICS) has to be implemented [166].

Consequently, *testing ICS cyber-defences against cyber-attacks* embodies the leading subject of this thesis. From this subject, four core topics can be distinguished: ICSs, IDSs, cyber-attacks, and testbeds. These four topics operate as the structural building blocks throughout this report. Regarding ICSs, arguably power grids are the most pressing ICSs to protect against cyber-attacks, since these are perceived as the source from which most disruptions can occur [193]. When it comes to controlling the stability of a power grid, Automatic Generation Control (AGC) is regarded as one of the most vulnerable but also commonly deployed control algorithms [32, 192, 211, 174]. AGC regulates the electricity flow between multiple geographically distant Load Frequency Control (LFC) areas. It hence depends on long-distance communication, thereby being particularly susceptible to cyber-attacks [27, 203]. One attack type which has proven to be potentially harmful to power grids is the data integrity attack (the third attack from Table 1.1) [117]. These attacks inject malicious data in the communication channels between a plant and its controller, which in the case of power grids can lead to frequency instability [197]. To cope with these attacks, the promising technique of Dynamic Multiplicative Watermarking (DMWM) can be deployed [51, 55, 200]. DMWM is an active signal authentication method which increases control system integrity through organised modification of the data which is sent over possibly compromised communication channels [138]. The increased integrity should then result in increased detection performance of observer-based cyber-attack detection schemes, such as Kalman filters or Luenberger observers [100]. To properly test DMWM on a power grid with AGC, a real-time Hardware-In-The-Loop (HIL) experimental testbed, entitled Hardware-In-the-Loop Detection of Attacks (HILDA), consisting of multiple industrial standard devices, is systematically constructed in the `Delft Centre for Systems and Control (DCSC)` lab, located at the `Delft University of Technology`. Overall, this thesis aims at answering the following research question:

Can the inclusion of dynamic multiplicative watermarking improve detectability accuracy and speed for data integrity attacks using an observer-based detection scheme on an automatic generation controller, simulated real-time on the HILDA testbed?

Through this research question, this thesis harbours three contributions. Firstly, to the author's knowledge, this is the first research about DMWM in the context of AGC, let alone while subject to data integrity attacks. Secondly, the testbed is designed and constructed not only for the specific purpose of this thesis, but also to facilitate future research on ICS operation, not limited to cyber-attack applications. Thirdly, the first validation results are extracted from the testbed by analysing the detection performance of DMWM on the HILDA testbed. These contributions are visualised in Figure 1.1.

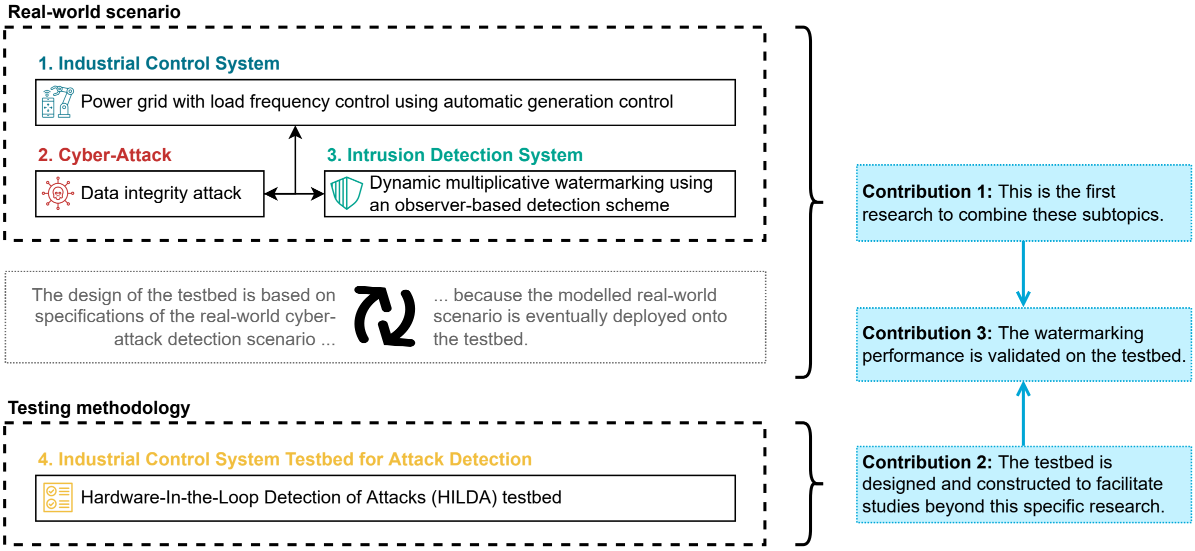

Real-world scenario
1. Industrial Control System
Power grid with load frequency control using automatic generation control
2. Cyber-Attack
Data integrity attack
3. Intrusion Detection System
Dynamic multiplicative watermarking using an observer-based detection scheme
The design of the testbed is based on specifications of the real-world cyber-attack detection scenario ...
... because the modelled real-world scenario is eventually deployed onto the testbed.
Testing methodology
4. Industrial Control System Testbed for Attack Detection
Hardware-In-the-Loop Detection of Attacks (HILDA) testbed
Contribution 1: This is the first research to combine these subtopics.
Contribution 3: The watermarking performance is validated on the testbed.
Contribution 2: The testbed is designed and constructed to facilitate studies beyond this specific research.

Figure 1.1: Visualised thesis contributions

To achieve these contributions, this report is organised as visualised in Figure 1.2. Here, *CA* stands for cyber-attacks, *DIA* for data integrity attacks, while LFC resembles load frequency control using automatic generation control, *WM* resembles DMWM with observer-based detection, and *TB* resembles ICS testbeds. As the outline shows on the left, chapter 2 covers the essential background information on the previously stated subtopics. Subsequently, the focus is put on the mathematical models of the LFC and AGC algorithms and the data integrity attacks they are subjected to, which are provided in chapter 3. This is followed by the mathematical design of the observer-based detection scheme with DMWM in chapter 4. With the mathematical model in place, the hardware and software designs of the HILDA testbed are elaborated in chapter 5. The hardware design should meet certain requirements to be able to deploy the mathematical model and design on the HILDA testbed, while a `MATLAB & Simulink` model is deployed on the involved testbed software. Both this `MATLAB & Simulink` model and the functionality of the HILDA testbed are verified in chapter 6, followed by a performance validation of DMWM when applied on the HILDA testbed in the same chapter. A discussion on the verification and validation results is offered in chapter 7. Finally, the report is concluded in chapter 8.

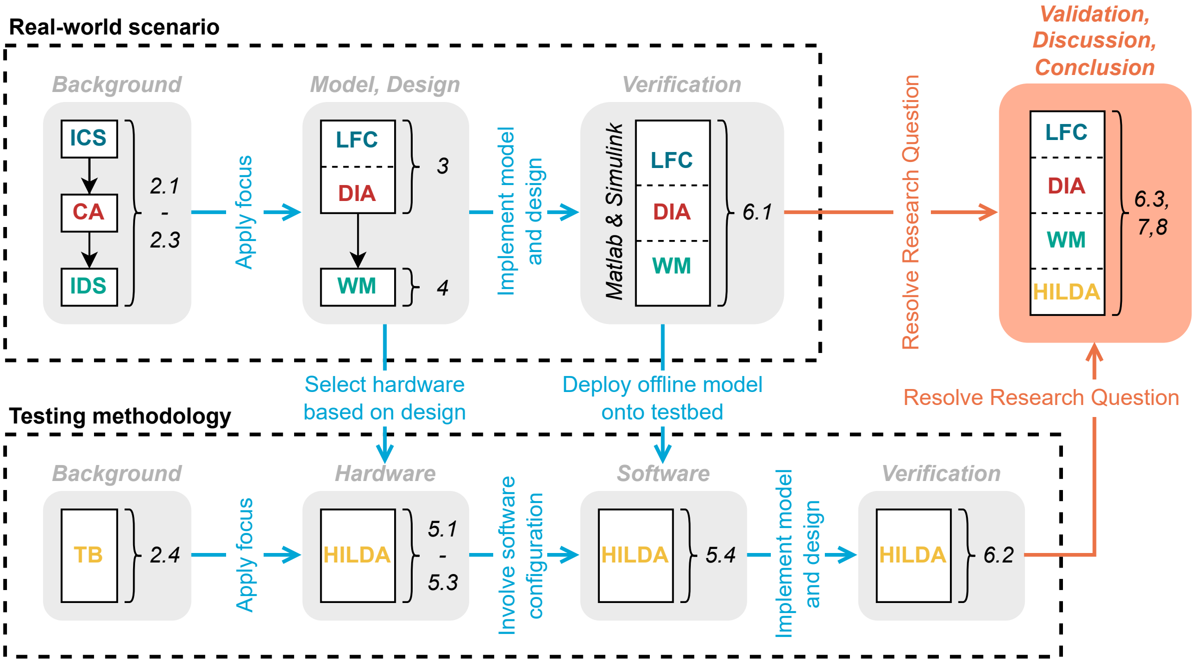


Figure 1.2: Visualised thesis outline

2

Background and Related Work

This chapter provides background information on the subject of *testing ICS cyber-defences against cyber-attacks*, followed by the available literature most relevant for this research. This information and the accompanying literature are deemed necessary by the author for complete comprehension of the research question introduced in chapter 1. This chapter is structured as follows: first, the four distinguished subtopics (ICSs, cyber-attacks, IDSs and ICS testbeds) are elaborated in separate sections. Each section is divided further into prominent features regarding the affiliated subtopic. Finally, of each feature, a specific scope is discussed. For example, a prominent feature of ICSs is their communication protocols, of which the `EtherCAT` protocol is the considered scope. On that account, section 2.1 discusses ICSs by elaborating the dominant architectures, the components relevant for this study, the deployed communication protocol, and the system applications. A brief overview of cyber-attack vulnerabilities of AGC is provided in section 2.2, followed by an introduction to the deployed cyber-attacks in this study, while section 2.3 discusses passive and active IDSs. The final subtopic of ICS testbeds is discussed in section 2.4, with testbed approaches and technologies as its dominant features. After these four subtopics, section 2.5 presents the available research and surveys closest to this study.

2.1. Industrial Control Systems

According to the extensively cited paper [193], ”an ICS consists of combinations of control components (e.g., electrical, mechanical, hydraulic, pneumatic) that act together to achieve an industrial objective”. This industrial objective can be summarised as production, transportation and transformation of assets [185]. Initially, ICSs were isolated from the outside world (i.e. monolithic systems), using only proprietary communication and control protocols [13, 48, 185]. Nowadays, ICSs rarely operate as segregated systems anymore, being strongly connected with remote control entities and corporate networks [170]. The inclusion of enterprise networks into ICSs show a rising trend [57]. The increased connectivity has multiple potential benefits: reducing cost due to optimised processes [29]; flexibility due to versatile control management [25]; and scalability due to wireless technologies [149, 214]. However, the ”cyber-attack surface” also grows once devices become more networked [8, 60, 135]. In other words, by introducing Information Technology (IT) to ICSs, the threat of cyber-attacks has increased [21]. And while the threats are easily inherited, it is not so easy to copy the same ”commercial-off-the-shelf IT cybersecurity solutions” [193]. This is mainly because of a distinction in security objectives [2, 29, 79, 126, 187]. The priority of IT systems is to protect the confidentiality of the data, while that of ICSs is to ensure availability and integrity of real-time data [26, 35, 125] (for a complete overview of differences, the reader is referred to [2, 29, 81, 96, 131, 193, 221]). Consequently, it is necessary to design cybersecurity measures specifically for ICSs, which requires knowledge on their architecture, components, communication protocols and applications, as all discussed in this section. Figure 2.1 provides an explanatory example of an ICS structure, of which some but not all components are treated here.

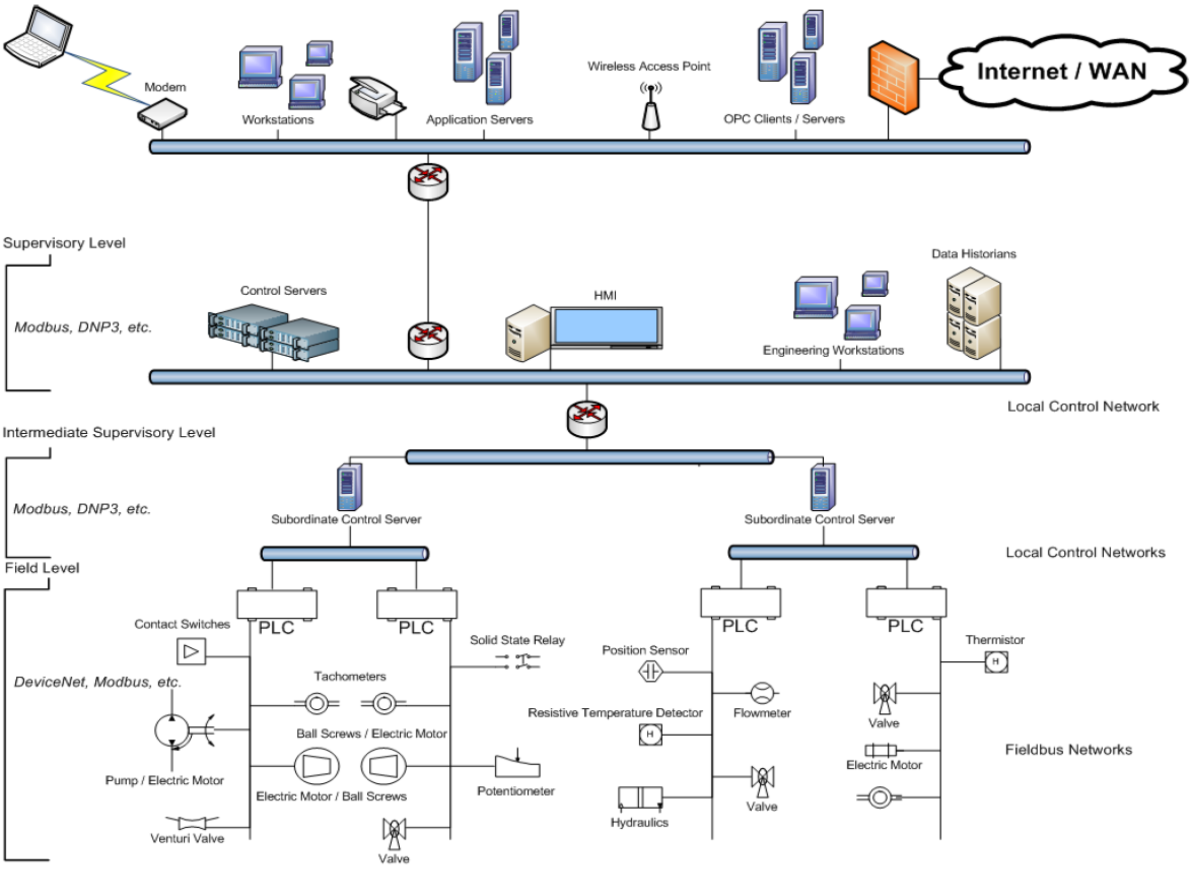


Figure 2.1: Explanatory example of an industrial control system [171]

2.1.1. Dominant Architectures

There are two dominant types in the ICS sector: Supervisory Control And Data Acquisition (SCADA) systems, and Distributed Control System (DCS). SCADA systems can hierarchically be above one or multiple DCSs, but not the other way around. Both types use Process Control System (PCS) for lower-level control actions (usually an industrial computer) [30]. At the highest level, both systems can be coupled to a corporate network [30, 48, 58, 112].

Supervisory control and data acquisition: SCADA systems generally assemble data to a centralised location, from which geographically dispersed assets are controlled [193]. This allows for control of multiple distributed systems from a single location, which is done in real-time [19]. SCADA systems are event-driven, meaning that actions are only triggered by certain changes in values. This lightens the load of the host, since scans are only performed in the occurrence of certain events. Because events are directly recorded when certain values change, event-driven systems are usually also quicker than process-driven systems (such as DCSs) [57]. SCADA systems are composed of three main sections [9, 96, 193]: a control server, an area network, and field sites. In control servers, the potentially distributed industrial processes are controlled. Often, it is possible to remotely access this control server, usually via separate modems (communication gateways) or Wide Area Network (WAN) connections. Area networks enable the control server to communicate with the distinct physical processes. Deployed techniques include physical cables, radios and satellites. The physical processes themselves take place at the field sites. PCSs monitor and control the local processes. Similar to the control server, these are connected with the network via communication gateways or WAN interfaces.

Distributed control systems: Like SCADA systems, DCSs control physical processes of possibly multiple subsystems [47]. However, whereas in SCADA these subsystems could be distributed over long distances, the subsystems of DCSs are located on a single site [193]. Multiple PCSs can be used for monitoring and processing [194]. In DCSs, the communication occurs in the form of process-driven polling between a main control entity and PCSs [57]. Process-driven polling implies that the polling occurs in a sequential manner, and that events are only recorded when requested by the control server.

2.1.2. Relevant Components

ICSs achieve their architectures through a broad variety of components. The most relevant ones for this research are listed below. Other prominent components (which are left out of the scope) are Remote Terminal Unit (RTU) [35, 194], Intelligent Electronic Device (IED) [86, 194], Front End Processor (FEP) [39, 205], and data historians [194].

Programmable logic controller: This is the most dominant representation of a PCS. A Programmable Logic Controller (PLC) is an electronic device, controlled by microprocessors, which can execute programmed instructions and create appropriate outputs, either based on input signals from sensors, or based on commands from supervisory controllers [194]. This operation is made possible through a power supply, Central Processing Unit (CPU), communication interface, and Input/Output (I/O) module(s), which can be analogue or digital. These systems are often designed to last $10-15$ years under harsh environments while in continuous operation. PLCs read inputs, execute logic and write outputs in real-time with the help of their real-time operating systems. This is usually done with an update frequency of $100-1000$ Hertz. In small scale industrial processes, PLCs can operate as the highest level controllers [193]. On larger scaled industrial applications, PLCs are subjective to some form of supervisory control, executing commands from a hierarchically higher control device.

Engineering workstations: These represent the main controllers of an industrial process, usually in the form of desktop computers. These components host the programming software of the PCSs, through which their control logic can be altered [194].

Human-machine interfaces: Not all functionalities and parts of ICSs are controllable or even accessible by all human operators [30]. Those parts that are can be accessed through a Human-Machine Interface (HMI). HMIs provide insight into proceedings of the automation processes. Examples of these proceedings are sensor values and data trends. Though these are a type of component in ICSs, they can take many forms in the likes of computers, laptops, smart-phones and tablets. HMIs can be programmed to assert control over the controller, or they can be limited to monitoring only [194].

Communication gateways: Not all ICS components can operate though all available communication protocols. It can hence be required to transform signal data to other protocols, which can be done through communication gateways.

Industrial control system field devices: These are the sensors, transducers and actuators on the processing sites. They form the bidirectional transmission between physical proceedings, such as the movement of machines, and digital or analogue signals used to control the physical processes.

2.1.3. Communication Protocols

ICSs rely heavily on communication. To achieve this communication, simply put, the involved devices need to speak the same language. These languages are referred to as communication protocols, or fieldbus protocols. The IEC standard 61158 defined fieldbus protocols as “a digital, serial, multidrop, data bus for communication with industrial control and instrumentation devices such as - but not limited to - transducers, actuators and local controllers” [57]. These protocols are developed to ensure real-time response, high availability and reliability [46]. Some popular protocols in the market are `MODBUS`, `Process Field Net (PROFINET)` and `Distributed Network Protocol (DNP3)` [173]. It is out of the scope of this study to provide a detailed description of every protocol in industry. For an overview of the protocols, the reader is referred to [213], though this overview misses the dominant `Process Field Bus (PROFIBUS)` and `PROFINET` protocols, which can be further investigated in [164] and [23]. Instead, the following section will only focus on the protocol used in the remainder of this study: the `EtherCAT` Automation Protocol (EAP), where `EtherCAT` stands for ’Ethernet for Control Automation Technology’.

`EtherCAT` automation protocol: EAP is an enhancement of the `EtherCAT` technology (IEC 61158, Part 12) [76]. EAP is an open protocol, connoting that its specifications are published. These specifications are set by the `EtherCAT Technology Group (ETG)` [76] in ETG.1005 [77]. The protocol is at the core of all `Beckhoff Automation GmbH` equipment. Proper understanding of the protocol requires basic knowledge on telecommunication systems, such as the Open Systems Interconnection (OSI) model (ISO/IEC 7498-1) [193]. Here, only the basic functionalities are discussed. For more details, the reader is referred to the [68], or other sources from `Beckhoff` or `ETG` [67, 75, 76, 77].

EAP operates using the Ethernet protocol (IEEE 802.3) [116]. Unlike most other ICS communication protocols, it does not rely on protocols at higher OSI layers (such as Transmission Control Protocol (TCP)/Internet Protocol (IP)) to establish data transfer or device connection [68]. It works according to the Publisher/Subscriber principle, instead of the traditional master-slave configuration [68]. The master-slave configuration is based on a parallel network configuration (i.e. a master can send a request to a slave, which optionally reports back to the master). This configuration comes with two delays: messages still have to be interpreted by the receiver before responding, causing stack delays; and dependence on protocols at higher OSI layers (such as TCP/IP) result in switching delays between these protocols and the Ethernet protocol.

EAP circumvents these delays by using cyclic 'on the fly' communication over Ethernet. This means that EAP devices read and write data on the message while it is passing by. It does so in a serial configuration, instead of parallel. To this end, it uses a distributed clock system: every EAP node in the network is equipped with a clock, and for every data frame transmission, the delay per node (i.e. the time the message left the node minus the time it arrived) is added to the data frame so that other nodes can calibrate their clock accordingly. These features make EAP 'hard real-time' [76], implying it is specifically designed for real-time physical operations. All EAP devices can operate both as (multiple) Publishers and Subscribers. Hence, EAP uses Master-Master communication [76]. This results in flexible network topology, as routing to any device connected to the network is possible. This can be done through a unicasting (point-to-point messaging), multicasting (sending messages to multiple defined points), or broadcasting (sending messages to all accessible end points) [68].

2.1.4. System Applications

ICSs are used to facilitate the functionality of (vital) industries, such as communication, electricity, water and wastewater, material energy, pharmaceutical, chemical, manufacturing and transportation, either locally or distributed [2, 193]. An entire list of industries is provided in [7]. For critical infrastructures, electric power is usually perceived as the source from which most disruptions can occur [193]. Practically all infrastructures depend on electricity, from transportation to communication. In an era with such dependency on electricity, cyber security of power grids can hence be considered as perhaps the most critical area of research compared to other critical infrastructures [206]. Major blackouts presented in [152] demonstrate the devastating impact of power grid failure on society. In addition to this already present danger, the number of cyber-security challenges for power grids is growing, arguably due to the imminent growth in connectivity, the increasing number of stakeholders [89] and the inclusion of modern technologies, such as Renewable Energy System (RES) technologies [31]. Therefore, the cybersecurity of electrical systems should be of primary concern [192, 132].

A power grid (also known as electrical grid) is a (cyber-)physical network which interconnects generation units to a load through transmission and distribution units [174]. They are controlled by an Energy Management System (EMS) (systems controlling the energy process by processing the sensor data into control input [103]) at an Energy Control Centre (ECC) [174]. These algorithms aim at balancing generated and loaded electricity in an optimal fashion [31], referred to as Optimal Power Flow (OPF) (of which [140] provides a comprehensive overview). Through a SCADA structure, an ECC can communicate with PCSs from both substations and generation units [62]. To this end, power grids rely heavily on wireless communication [127] and other 'intelligent' devices [22].

To achieve grid stability, three quantities need to be actively controlled: frequency, voltage and rotor angle [141, 174]. Understanding these quantities requires understanding of synchronous machines. In brief, synchronous machines are electromechanical transducers that convert mechanical energy, usually generated by a steam or hydro turbine, into electrical energy [174]. It does so by rotating a 'rotor' - which is equipped with field winding - inside a 'stator' - which is equipped with armature winding. By the laws of electromagnetic induction and law of interaction, this rotation produces electrical energy. Further details are out of the scope of this research. The reader is referred to [4, 82, 174]. It is important to note that cross-coupling between the frequency and other regulations is negligible [174]. Therefore, it is possible to perform studies on frequency regulation independently of the other quantities.

Of the three quantities, the frequency is the most time consuming to control [141]. The frequency refers to the utility/mains frequency of Alternating Current (AC), which flows in wide area electrical networks [174]. It is designed to perform around a stable operating point (50Hz in Europe, 60Hz in the United States). Its deviation should be minimised, which is done through LFC. A typical LFC control loop is provided in Figure 2.2. LFC consists of three sequential control phases: primary (or generation) control, secondary (or supplementary) control, and tertiary (or load) control [155]. These are presented in Table 2.1. For a detailed general dynamic model of a power system, the reader is referred to [175]. The secondary control mechanism is more vulnerable to cyber-attacks, since it is most concerned with wide-area control [141, 175]. Hence, the remainder of this research is first and foremost concerned with the secondary control mechanism, specifically the AGC algorithm.

Control Step	Trigger	Initiation	Description
Primary	Automatic	Instantly	Each generator unit is equipped with a ’speed governor’, which controls the amount of fuel going into the turbine driving the electromechanical transducer. This governor bases its control input to the turbine on local frequency measurements. For most generators, it is regulated by a function called ’speed droop’ [31, 95].
Secondary	Automatic	A matter of seconds	If the local primary control has not resolved the frequency deviation, other algorithms (usually controlling multiple interconnected generators) manipulate the set point of the respective governors. For this, AGC is markedly the most used algorithm [31, 191, 192].
Tertiary	Human operator	A matter of minutes	If both primary and secondary control have not resolved the frequency deviation, human power grid operators can manually adjust the dispatch of the involved generator units and loads [174]

Table 2.1: Frequency regulation sequence in power grids

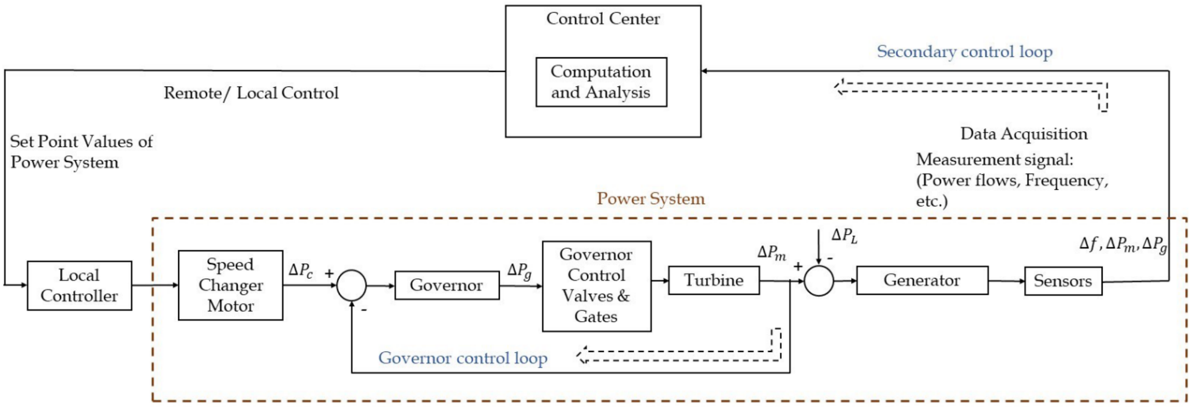


Figure 2.2: Explanatory example of an load frequency control loop [141]

Load frequency control using automatic generation control: AGC is not a new technology: is has been standardised by the IEEE since 1970 [18, 40, 95]. It has defined AGC as ”the regulation of the power output of electric generators within a prescribed area in response to changes in system frequency, tie-line loading, or the regulation of these to each other, so as to maintain the scheduled system frequency and/or the established interchange with other areas within predetermined limits.” In other words, AGC is a multi-variable feedback loop used to stabilise interconnected power areas by controlling their net interchange and the local area production [91, 211]. For this, it relies on measurements presented by telemetry systems (system frequency and tie-line loading) [40, 192]. These measurements, and their resulting control inputs produced by the AGC algorithm, travel over a WAN, making it one of the fundamental wide-area control applications within power grids [14]. Nowadays, AGC is considered crucial to the reliability and stability of bulk power systems [14]. Therefore, in most practical interconnected power system applications, AGC is used [103]. It is also possible to include accompanying economical dispatch algorithms. These take into account the efficiency and sustainability of the linked generators, and deploy their control accordingly [118].

2.2. Cyber-Attacks

The past two decades, the skill and resources of the attackers have grown faster than the increased system complexity and security [13, 135]. An entire cyber-attack business has even developed [91]. Combining this with an increase of ICS vulnerabilities causes a growing number of attacks being conducted. An overview of recent cyber-attacks on power grids is presented in [146, 148]. [7, 15, 32, 45, 89, 94, 107, 119, 156, 221] all provide cyber-attack taxonomies used to execute these attacks. In all taxonomies, two aspects are recurring: vulnerabilities (what to attack), and attack vectors (what cyberweapon(s) to use). This section will therefore analyse the vulnerabilities of the AGC mechanism, together with the attacks used in this research.

2.2.1. Exploitable Automatic Generation Control Vulnerabilities

Cyber-attacks on AGC have the potential to cause large scale stability issues [32, 148]. An attack on a single area can create a blackout of the entire power grid [211]. In other words, an increase in connectivity leads to an increase in potential damage. An increase in connectivity additionally leads to an increase in cyber-attack surface [60]. The main vulnerability of AGC mechanisms is their need for long distance communication [114, 214]. Most communication protocols in industry feature cleartext transmission of data, which lacks authentication and is therefore easy to manipulate [46, 121]. This study therefore focuses on attacks resulting from communication manipulation.

Manipulation of communication data: Specifically for AGC, this study is concerned with the manipulation of the measurements and control inputs which are being exchanged between the AGC algorithm and the affiliated generators. In the remainder of this report, it is assumed the attacker has access to reading and manipulating both measurement and control data. This could for example occur if the attacker intrudes in the corporate network by bypassing the first layers of security (firewalls). Also, the attacker could intrude in the control centre through the WiFi-network [10]. An example of a successful intrusion in a power grid system is provided in [73].

2.2.2. Types of Cyber-Attack

The available literature is full of cyber-attack descriptions, too numerous to all be discussed. To make a selection, multiple papers which address cyber-attacks on specifically ICSs [58, 59, 89, 92, 121, 144, 143] were investigated. Three resulting attack types are considered further: a Reconnaissance/Eavesdropping attack, a Man-In-The-Middle (MITM) attack, and a data integrity attack. This study aims at implementing the third attack, for which prior execution of the first two attacks is common (though not the only possibility, see [59, 125] for more options). All three attacks are elaborated below, where most emphasis is put on the data integrity attack.

Reconnaissance/eavesdropping attack: Usually, after somehow having intruded the system, the attacker performs a Reconnaissance/Eavesdropping attack [214]. This attack focuses on gathering system information (deployed equipment, used algorithms etc.) [121]. They are frequently used as preparation for more malicious attacks [135]. To execute such an attack, multiple scan types can be inserted, either actively or passively [92]. In an active reconnaissance attack, the attackers sends messages to system devices, provoking a response containing system information. Examples of active scans are address scans [121], function code scans [121], point scans [59, 121] and device identification scans [59]. Passive scans are conducted by somehow intercepting the communication data between two or more devices, usually through MITM attacks.

Man-in-the-middle attack: In MITM attacks, the attacker is nestled between two communication targets. It does so by exploiting the active communication protocols [91]. The communication targets are tricked into sending the communication to the attacker, after which the attacker can read, adjust and replace the message before sending it through to its original destination [5, 20]. the attacker achieves this by identifying itself as the communication targets, also called ’spoofing’ [78, 89, 91]. One manifestation is Address Resolution Protocol (ARP) spoofing (also called ARP cache poisoning), where the attacker’s Media Access Control (MAC) address is associated with the IP address of another host in the network [221]. Once a MITM attack is successful, the communication between the two targets is intercepted. The attacker can then forward, ’kill’ (e.g. interrupt), reconstruct or replace communication messages in real-time.

Data integrity attacks: As stated, this study aims at implementing a data integrity attack. The motivation is plural: firstly, it is an attack on communication data and the control operation specifically [191, 179]; secondly, since power grids are able to (less optimally) operate without AGC, the primary security concern is not data availability, but integrity, which is the primary target of data integrity attacks [141, 172]; thirdly, research has shown the vulnerability of power grids to data integrity attacks [20, 124, 129, 148, 191, 192, 197, 216], of which the 2015 Ukraine blackout is a bedevilling illustration [117, 120].

Data integrity attacks are executed on the communication between a physical process and a controller [89]. On one hand, the measurements which are being sent from the plant to the controller can be manipulated, which is referred to as a False Data Injection (FDI) [137] or deception attack [128]. This is markedly the most studied form of data integrity attacks on cyber-physical systems [125]. The goal is to maliciously influence the decision-making of the controller [121]. In other words, the input of the control server is tempered with by falsifying the data from the physical layers [149]. On the other hand, the signal from the controller to the process (so control signals) can be tempered with, which is referred to as a command injection attack [59]. These attacks can cause malicious control actions in the lower levels of the systems, such as modification of process set points.

Data integrity attacks can disparate in their level of knowledge on the targeted system. Some attacks lack process knowledge about the attacked system, referred to as Naive Malicious Response Injection (NMRI) attacks [59, 144]. Complex Malicious Response Injection (CMRI) attacks are the sophisticated counterpart of NMRI attacks, having (almost) full knowledge of the system at hand, thereby being able to inject data in a more sophisticated fashion.

The two specific types of data integrity attacks which are considered in the remainder of this study are scaling attacks and replay attacks. A scaling attack, mathematically, is a multiplication of the true signal by an arbitrary scale [100]. Replay attacks [128, 138] consist of two main steps: firstly, some output of the system is recorded for a certain time period, through which it is tried to replicate the system dynamics in nominal behaviour; then, as actual attack, the recorded data is injected into the system as false input, replacing the original data, thereby disrupting the system behaviour [214]. Both are of multiplicative nature, which are lesser studied than attacks of additive nature [52, 53]. The attacks are mathematically modelled in section 3.3.

2.3. Intrusion Detection Systems

Because of the potential impact of cyber-attacks as discussed in chapter 1, cybersecurity of ICSs is already a substantial research area. By means of a structured approach, a multitude of numerously cited ICS cyber-defence publications [29, 44, 64, 84, 93, 99, 104, 111, 131, 146, 151, 193, 195, 201] were comprised to two subsequential cyber-defence measures: IDSs and mitigation strategies. IDSs are concerned with extracting useful information from the system, and using this to detect anomalies [29, 131]. Mitigation strategies are concerned with the aversion of future intrusions through risk analyses [13, 36, 142, 193], and with the diffusion of anomalies after detection [158, 201]. An overview of mitigation strategies specifically for smart grids is provided in [163]. The remainder of this thesis is concerned with IDSs. Based on multiple surveys on IDS research [39, 63, 84, 88, 148, 195, 198, 214, 217, 221], a dominant distinction can be made between passive and active IDS approaches, which are elaborated below.

2.3.1. Passive Intrusion Detection Systems

Most anomaly detection mechanisms are passive [63]. Passive IDSs aim at detecting attacks based on some statistical hypothesis tests without adding an excitation to the signal, such as the χ^2 (chi-squared) detection method, which uses Gaussian distribution to detect attacks [128, 141, 179], or an unaccompanied observer-based detection mechanism.

Observer-based detection: In observer-based detection mechanisms, the detection is performed through correlation techniques [201]. Actual system behaviour is compared to nominal system behaviour [39]. This nominal system behaviour is determined through the use of observers. When these behaviours do not match to a certain extend (beyond the impact of irregularities such as noise and disturbance), a fault or attack could be the cause [44]. Observer-based detection is commonly classified as ’anomaly-based’ or ’model-based’ detection [148]. The main drawback of observer-based detection is the need for either a large list of known anomalies, or an accurate and possibly computationally intensive model. Both could be time-consuming to obtain. A drawback of passive IDSs overall is that they especially are vulnerable to attackers which have acquired system information through for example a reconnaissance attack. In this case, a stealthy attack can be designed to bypass the IDS. More sophisticated algorithms could be deployed to increase estimation performance. However, for AGC, this is not an option [141]. This is because of its closed-loop runtime of mere seconds. Therefore, less sophisticated (usually linear [16]) estimation algorithms are implemented, decreasing fault/intrusion detection performance in the case of cyber-attacks [32]. Nonetheless, [108] shows that through specific bounds on data integrity attacks, the attacker can avoid conventional detection methods of AGC systems.

2.3.2. Active Intrusion Detection Systems

Active IDSs, on the other hand, increase detectability by adding excitation signals to the system. These techniques are also often referred to as active monitoring [63]. Watermarking is an effective example of such an active detection technique [218]. This technique superimposes a watermark signal on certain system signals, originating from either sensors or controllers. Another way of actively monitoring the system is by randomly changing its topology, as discussed in [63] for power grids. The downside of active detection is that it could increase computational intensity of the system [56]. Nonetheless, this study will implement active detection through a watermarker signal. This is mainly due to the promising research results [84], and because physical watermarking has not yet been extensively tested on complex systems [122], such as power grids.

Dynamic multiplicative watermarking: A watermark (also called an authentication signal [136, 138, 139]), was already introduced as an effective example of active detection mechanisms. The main concept of watermarking is to superimpose an artificial signal with an authentication signal before sending it through a possibly compromised network. The specifications of this authentication signal are only known to the operator, thereby increasing the integrity of the signals. There are two dominant distinctions in the field of watermarking: additive versus multiplicative, and static versus dynamic [173]. The first distinction touches on the mathematical technique utilised to superimpose the authentication signal: additive watermarking uses addition, while multiplicative watermarking uses multiplication [139]. The second distinction is concerned with the potential adaptation of the parameters of which the watermarker is constructed: static watermarking does not change the watermarking parameters, while dynamic watermarking does. Additive watermarking is commonly executed by means of additive Gaussian noise [89, 138, 173]. The disadvantage of this is that is imposes an additional burden on system performance [3, 200]. On top of that, more recent publications on watermarking have shown additive and static watermarking to be flawed due to the possibility of attackers to identify and copy the watermark parameters, thereby losing signal integrity [173]. These flaws are solved by DMWM. This is because multiplicative watermarking allows for the use of an inverse watermarker, which is applied to the watermarked signal after is has passed the possibly compromised network, thereby neutralising the watermark effect in the absence of an attack [52, 53]. On top of that, by adjusting the artificial watermark parameters over time, it becomes harder for malicious attackers to identify these parameters [173]. The more dynamically (more frequently and unpredictably) the parameters are altered, the harder it becomes for attackers to alter the watermarked signal unnoticed. The implementation of DMWM on AGC is a contribution of this thesis, and is further discussed in chapter 3 and chapter 4.

2.4. Industrial Control System Testbeds

Experimenting on live critical infrastructure is generally impractical or even impossible due to the potential consequences of failure [166, 199]. For this reason, testbeds are developed. The goal of testbeds is to come as close to real-world scenarios as possible to counter unforeseen consequences [87, 110, 166]. Next to that, testbeds should be isolated from other networks to exclude undesired external influences [60]. This section discusses the relevant approaches (i.e. different categories) of testbeds, followed by a brief description on existing testbeds.

2.4.1. Testbed Approaches

Different approaches of testbeds are appropriate for different situations. Five groups can be distinguished [166], which are represented and evaluated in Table 2.2. Physical replication testbeds exist only out of components which would be used in a real-world ICS as well, thereby cloning the real system. Simulated testbeds run only on computer software (in Table 2.2, the accuracy is denoted to be poor, though arguably enough processing power and system knowledge could simulate a plant perfectly, although this knowledge about real-world effects is very hard to achieve). Virtualisation testbeds also run exclusively on software, but in an environment which minimises or eliminates the software's dependence on the hardware which it runs on. Virtual-physical replication testbeds have (a part of) the ICS process level replaced with a real-time computer model, while deploying real-world components (such as PLCs). These testbeds are therefore also known as HIL setups [6]. Finally, hybrid testbeds are some combination of the other categories.

Testbed Approach	Fidelity	Repeatability	Accuracy	Safety	Cost-effective	Reliability	Scalability
Physical replication	Excellent	Poor	Moderate	Poor	Poor	Excellent	Poor
Simulated	Low	Moderate	Poor	Excellent	Excellent	Poor	High
Virtual	Moderate	High	Moderate	Excellent	Moderate	Moderate	Moderate
Virtual-Physical	High	High	Excellent	Excellent	Low	High	Moderate
Hybrid	High	High	Excellent	High	Moderate	High	Moderate

Table 2.2: Evaluation of testbed categories, based on [166]

Virtual-physical testbed at the Delft Centre for Systems and Control lab: The testbed designed and constructed in this thesis is of the virtual-physical kind. The ICS process level is replaced with a real-time simulator from `dSPACE GmbH` (more on this in subsection 2.4.2). The first steps in designing this testbed were performed in the work of the author's predecessor, V. Ranade, as part of his Systems and Control MSc thesis [168]. The continued design and eventual construction of the testbed are a contribution of this thesis, and are further discussed in chapter 5. For ease of notation and communication, the testbed has been given a name: the HILDA testbed.

2.4.2. Existing Testbeds

[37] lists a total of 36 cyber-physical smart grid testbeds published in the period of 2008 till the end of 2015. [195] also lists a few cyber-physical power grid testbeds. [121] lists a total of five testbeds for ICSs in general. [34] discusses five testbeds in relative detail, three of which run on the `MODBUS` communication protocol. Specific relevant testbed implementations include [5], which designs a cyber-physical power system testbed and conducts an IDS experiment on it. [80] and [202] provide testbeds for simulating smart grids, which are not designed specifically for cyber-attacks. [14] simulates a 9-bus power system on a testbed composed of SCADA hardware and software and subjects it to cyber-attacks, without involving any cyber-security mechanism. [110] focuses on building a testbed which includes properly validating detection methods, combining real network traffic with simulated physical models in real-time. Finally, [177] builds a testbed on which the IEEE 30-bus power system model is simulated, and on which a co-simulated `MATLAB & Simulink` based detection model is applied. For details about the structures and components of these testbeds, the reader is referred to the citations.

2.5. Related Work

To emphasise on the research gap which this study tends to, this section elaborates on the most relevant available literature in relation to the scoped topics AGC, data integrity attacks, DMWM with observer-based control, and ICS testbeds. Additionally, it states the surveys most regularly used as background information (all of which are therefore also cited in the preliminary sections of this chapter).

2.5.1. Preliminary Research Closest to this Study

The research closest to this study was performed in [90]. As in this study, [90] implements dynamic watermarking on AGC and subjects it to data integrity attacks (more specifically, replay and noise injection attacks). However, two substantial differences exist: firstly, [90] implements additive watermarking as opposed to multiplicative watermarking; secondly, [90] does not validate the hypotheses through a virtual-physical testbed, but through non-real-time software simulations. [173] also implements dynamic additive watermarking and does validate the hypotheses through a virtual-physical testbed, but not specifically for AGC (but for cyber-physical systems in general). Another approach was taken by [197], which performs a rigorous analysis of the mathematical relationship between FDI attack and

AGC and validates in on a testbed, but does not involve watermarking of any kind.

Regarding data integrity attacks, this study largely builds on the work of [17, 24, 52, 100, 141, 157]. The IDS model applied in this study is mainly adopted from [52], which uses DMWM complementary to observer-based detection (similar to the techniques used in [51, 53, 200]). [52] also provides mathematical proofs of relevant detection and stability guarantees. For modelling the AGC mechanism, a combination of models and findings from [10, 12, 100, 141] was comprised. As watermarking is the technique which is principally reviewed in this study, auxiliary care has been taken by the author to include as much of the available literature on this topic as possible. A table of all considered literature on watermarking - not restricted to multiplicative or dynamic nature - can be found in Appendix A.

2.5.2. Most Relevant Surveys

A selection of surveys were used extensively to properly ground this research. For ICS as a whole, [193] serves as an appropriate survey. More specifically for AGC, a recent comprehensive overview is provided by [84, 203] on both conventional and modern power systems, with economic dispatch as additional objective. [32, 204, 212] are comprehensive overviews of cyber-attacks (including data integrity attacks) on power grid functionalities (including AGC). Similar overviews, but more focused on cyber-security measures, are [141, 148, 192].

2.6. Conclusion

This chapter provided background information on ICSs, cyber-attacks, IDSs and ICS testbeds (in that order). More specifically, it scoped down to specific issues in these areas relevant for the remainder of this report. Additionally, the available research and surveys closest to this study were emphasised. The next chapters dive deeper into the models of this study and the design of the testbed. After that, the results, discussion and final conclusions and recommendations are presented.

3

Modelling of Load Frequency Control and Data Integrity Attack

This chapter provides the mathematical formulations of the LFC mechanism - the AGC algorithm included - and the data integrity attacks it is subjected to. These formulations can then serve as a foundation for the IDS design, which follows in the next chapter. For the LFC and AGC models, first some important dynamics are clarified and substantiated mathematically. Subsequently, these dynamics are combined in state-space models, followed by appropriate completion of the relevant parameters. The modelling of AGC is primarily based on [141], but considers the models from [1, 12, 10, 16, 17, 27, 90, 100, 186] as well. Finally, the data integrity attacks are also represented in suitable mathematical models. This is done primarily based on [17, 24, 52, 100, 141, 157].

3.1. Load Frequency Control Dynamics

This research endeavours to subject an AGC algorithm to a data integrity attack. However, as was clarified in Table 2.1 of section 2.1, AGC is a secondary LFC mechanism, which usually does not operate in absence of a primary control mechanism. Consequently, in order to implement a model of AGC, a model of primary LFC has to be implemented as well. The mathematical dynamics of both implementations are discussed in this section.

3.1.1. Load Frequency Control Modelling Approaches

Primary LFC operates locally at a generator, while multiple generators can be bundled together in LFC areas. These areas can then be physically connected through tie-lines. In other words, the scale of power grid models can vary drastically through their number of areas and generators. Consequently, for the purpose of harbouring potential future expansion, LFC models should be scalable also.

There are three principal approaches of modelling the interaction between a LFC area and an AGC algorithm: combined LFC areas communicating with a centralised AGC algorithm (see Figure 3.1) [16, 17, 27, 100]; split LFC areas communicating with a centralised AGC algorithm (see Figure 3.2) [1, 141]; and split LFC areas communicating with split AGC algorithms (see Figure 3.3) [12, 90] (it should be noted that systematically separating these models as such is not common in the available literature, as other studies merely consider one or the other; also, for simplicity of notation, continuous and discrete time indicators (t) and $[k]$, respectively, are omitted in all schematics). No process or measurement noise are yet considered. The total amount of LFC areas is N. Each area i generates n_y outputs,

$$y_{p_i}(t) = \begin{bmatrix} \Delta f_i(t) & \Delta P_{tie_i}(t) \end{bmatrix} \in \mathbb{R}^{n_y}, \tag{3.1}$$

where $\Delta f_i(t)$ is the local frequency deviation and $\Delta P_{tie_i}(t)$ is the sum of all tie-line power deviations between area i and other connected areas j. The combined measurements of all areas can then be formulated as a stacked vector,

$$y_p(t) \coloneqq \left[y_{p_1}(t)^T \quad y_{p_2}(t)^T \quad \dots \quad y_{p_N}(t)^T\right]^T \in \mathbb{R}^{n_y N}. \tag{3.2}$$

When these measurements are sent over the network, it is possible that their integrity is compromised, resulting in $\tilde{y}_p(t) \in \mathbb{R}^{n_y N}$. The AGC algorithm uses the potentially compromised plant measurements to generate area control inputs $u_{c_i}(t) \in \mathbb{R}$. In Figure 3.3, it is worth noting that split AGC algorithms use the measurements of other areas, instead of just the area they control [12]. Like the measurements, these control signals are sent over the network, resulting in $\tilde{u}_{c_i}(t) \in \mathbb{R}$. There, they are stacked together with an unknown input $u_{u_i} \in \mathbb{R}$, which is the load of the areas. This results in n_u inputs of the areas,

$$u_{p_i}(t) = \left[\tilde{u}_{c_i}(t) \quad u_{u_i}(t)\right]^T = \left[\Delta\tilde{P}_{c_i}(t) \quad \Delta P_{L_i}(t)\right]^T \in \mathbb{R}^{n_u}, \tag{3.3}$$

where $\Delta\tilde{P}_{c_i}(t)$ is the possibly compromised AGC reference setpoint steering the amount of electricity generation, and $\Delta P_{L_i}(t)$ is the local load change. Similarly to $y_p(t)$, $u_p(t)$ can be formulated as a stacked vector,

$$u_p(t) \coloneqq \left[u_{p_1}(t)^T \quad u_{p_2}(t)^T \quad \dots \quad u_{p_N}(t)^T\right]^T \in \mathbb{R}^{n_u N}. \tag{3.4}$$

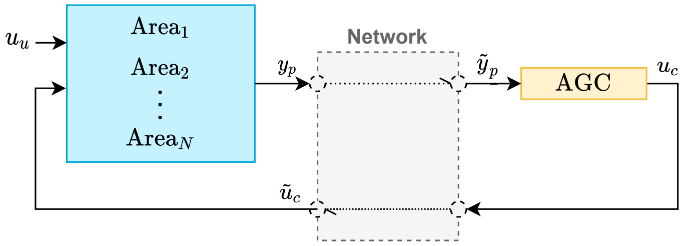


Figure 3.1: Load frequency control with combined area and automatic generation control model

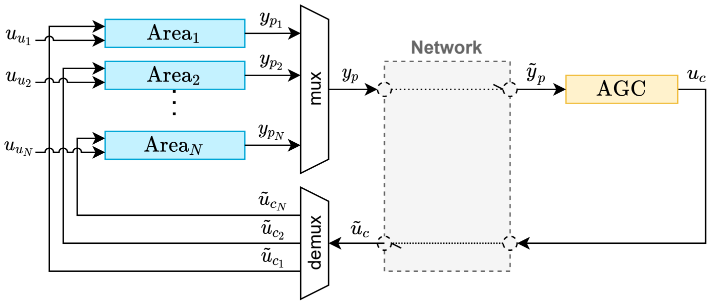


Figure 3.2: Load frequency control with split area and combined automatic generation control model

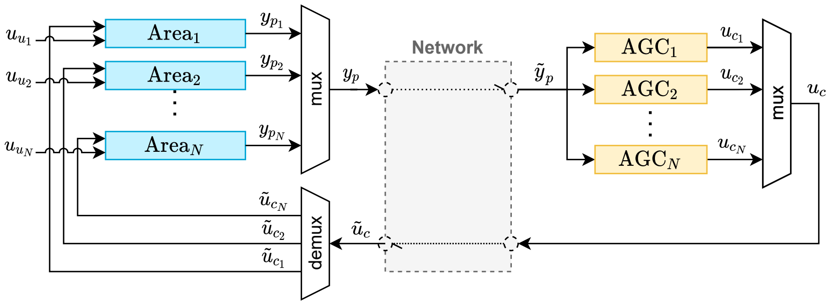


Figure 3.3: Load frequency control with split area and automatic generation control model

Only one of the three models as presented in Figure 3.1, Figure 3.2 and Figure 3.3 is selected for the remainder of this research. To reflect the scenario where multiple energy management companies cooperate to ensure grid stability (which is the status quo), it is best to use the split area and combined

AGC model from Figure 3.2. This is because, due to the split area approach, each energy management company is able to alter the configuration of its own area without having to change that of the other areas. Also, the combined AGC approach represents the close cooperation between the companies. Regarding the application of this research, it would also not be practical to implement a split AGC model, since only a single PLC is used as the hierarchically highest controller. Based on findings of [1, 10, 16, 17, 27, 90, 100, 141, 186] and as explicitly stated by [12], this model is assumed to be sufficiently accurate for LFC studies.

Assumption 3.1. *The Load Frequency Control modelling approach with split area and combined AGC is sufficiently accurate for LFC studies.*

Regarding the amount of areas, a two-area model is selected ($N = 2$). This is to not divert the attention away from the three thesis contributions. For the same reason, only a single generator per area is considered. This is also because, commonly, it is assumed that a multitude of synchronous generators within a single area react the same to frequency fluctuations and load changes as a single generator. This would make it suitable to represent a real-world power grid area with only one generator [141, 174]. Besides, in order to actually increase the accuracy of the AGC system, RES, economic dispatch models, market dynamics etc. would have to be included, but this is all beyond the scope of analysing the prior functionality of AGC subject to cyber-attacks, and the added value of DMWM in this scenario. The eventual model used in this study is represented in Figure 3.4. Using this figure as foundation, the specific dynamics regarding the LFC model are clarified in the following subsections. The considered parameters are provided in Table 3.1, where subscript i denotes the various LFC areas i. In real world scenarios, certain generation control devices, such as valves and motors, would be subjected to nonlinear fatigue [90]. Also, some generators operate more efficiently or sustainably than others. This asset can be used by deploying economic dispatch algorithms, which distribute electricity production across generators accordingly [90]. However, both implementations would drastically increase model complexity, and are hence not considered in this study.

Assumption 3.2. *No fatigue of any generation control devices takes place, despite potential changes in workload over time.*

Assumption 3.3. *All considered generators operate equally efficiently and effectively, regardless of the area they operate in or the amount of electricity they generate.*

			Value	
Parameter	Description	Unit	Area 1	Area 2
$\Delta f_i(t)$	Frequency deviation of power system	[Hz]	-	-
$\Delta P_{m_i}(t)$	Mechanical power change	[p.u.]	-	-
$\Delta P_{g_i}(t)$	Governor output change	[p.u.]	-	-
$\Delta P_{tie_i}(t)$	Tie-line active power deviation	[p.u.]	-	-
$\Delta P_{L_i}(t)$	Load change	[p.u.]	-	-
$\Delta P_{c_i}(t)$	Control signal	[p.u.]	-	-
$ACE_i(t)$	Area Control Error	[p.u.]	-	-
R_i	Speed droop characteristic	[Hz / p.u.]	0.05	0.0625
D_i	Frequency sensitivity load coefficient	[p.u./Hz]	0.6	0.9
H_i	Inertia constant	[p.u.-s]	5	4
T_{g_i}	Governor time constant	[s]	0.2	0.3
T_{t_i}	Turbine time constant	[s]	0.5	0.6
$P_{s_{i,j}}$	Synchronising power coefficient	[p.u.]	2	2
K_i	AGC integrator gain	[p.u.]	0.3	0.3
β_i	Frequency bias factor	[p.u./Hz]	$\frac{1}{R_1} + D_1$	$\frac{1}{R_2} + D_2$
N	Number of control areas	[p.u.]	2	

Table 3.1: Load frequency control parameters

3.1.2. Generator-Load Dynamics

In Figure 3.4, the generator-load dynamics involve the generators and the preceding summations. As was discussed in subsection 2.1.4, generators convert mechanical energy $P_{m_i}(t)$ into electrical energy $P_{el_i}(t)$ by rotating a rotor. This results in a certain frequency, which is the LFC area frequency $f_i(t)$. The famous *swing equation* [113] denotes this relationship as

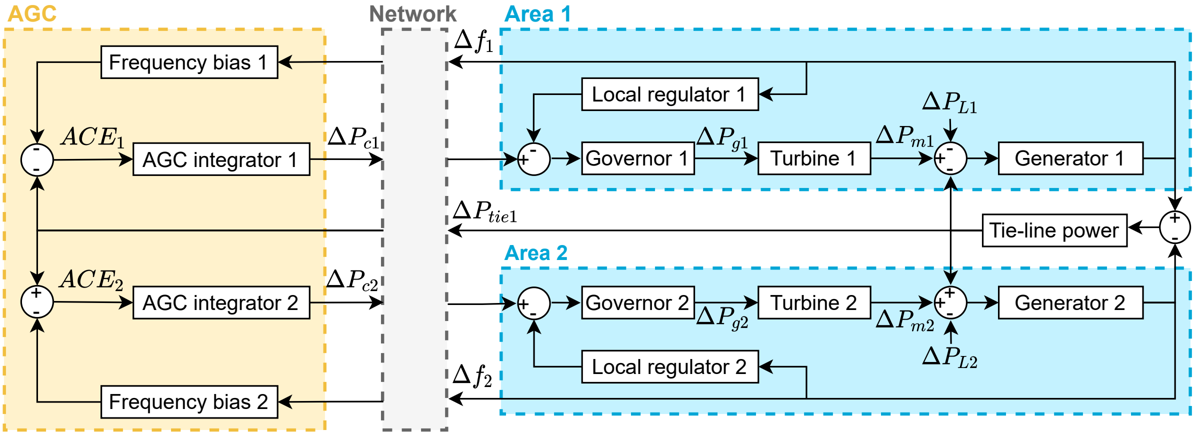


Figure 3.4: Schematics of a two-area load frequency control system

$$\Delta P_{m_i}(t) - \Delta P_{el_i}(t) = 2H_i \Delta \dot{f}_i(t) \tag{3.5}$$

where $\Delta \dot{f}_i(t)$ is the rotational acceleration and H_i is the inertia constant. Together with the mechanical power change, $\Delta P_{m_i}(t) - \Delta P_{el_i}(t)$ represents the overall power deviation. When looking at Figure 3.4, the electrical energy can also be formulated as $\Delta P_{L_i}(t) + D_i \Delta f_i(t) + \Delta P_{tie_i}(t)$, where $\Delta P_{L_i}(t)$ represents the electrical load change, D_i the frequency sensitivity coefficient and where $\Delta P_{tie_i}(t)$ is the sum of all the tie-line power deviations between area i and areas $j \in \delta_i$, with δ_i being the set of areas connected to area i (more on this in subsection 3.1.3). Consequently, according to [141], the fundamental dynamics of the rotational acceleration can be modelled as

$$\Delta \dot{f}_i = -\frac{1}{2H_i}\left(D_i \Delta f_i - \Delta P_{m_i} + \Delta P_{tie_i}(t) + \Delta P_{L_i}\right). \tag{3.6}$$

3.1.3. Tie-Line Power Dynamics

As visualised in Figure 3.4, the tie-lines connect multiple LFC areas to one another. This way, power can be exchanged, and overall electricity production can be made more optimal. The active tie-line power exchange is represented by Equation 3.7, where $P_{s_{i,j}}$ is the synchronising power coefficient. Following Equation 3.7, the dynamics of Equation 3.8 and the relationship of Equation 3.9 are established [141].

$$\Delta P_{tie_i} = \sum_{j \in \delta_i}^{N} \Delta P_{tie_{i,j}}(t) = \sum_{j \in \delta_i}^{N} 2\pi P_{s_{i,j}} \left(\int \Delta f_i(t) - \int \Delta f_j(t) \right), \tag{3.7}$$

$$\Delta \dot{P}_{tie_{i,j}} = P_{s_{i,j}} \left(\Delta f_i - \Delta f_j \right) \tag{3.8}$$

$$\Delta P_{tie_{i,j}} = -\Delta P_{tie_{j,i}}. \tag{3.9}$$

3.1.4. Governor Dynamics

In Figure 3.4, the governor dynamics encapsulate the local regulators, the governors and their preceding summation. The governor controls the feed of fuel (e.g. water in a hydraulic power plant, or steam in a steam turbine). For this, it relies on the control input from the local regulator (which in turn relies on the local frequency measurements), the global AGC control input $\Delta P_{c_i}(t)$, and the state $P_{g_i}(t)$ of the governor itself [141]. Mathematically formulated, the governor dynamics are represented as

$$\Delta \dot{P}_{g_i}(t) = -\frac{1}{T_{g_i}} \left(\frac{1}{R_i} \Delta f_i(t) + \Delta P_{g_i}(t) - \Delta P_{c_i}(t) \right) \tag{3.10}$$

where T_{g_i} represents the governor time constant, and R_i the speed droop characteristics (as was introduced in subsection 2.1.4).

3.1.5. Turbine Dynamics

The governors feed their regulation signal to the turbines, which in this case are steam turbines. The turbines produce the mechanical power which, through the swing equation, can compensate load frequency fluctuations [141]. Mathematically, the turbine dynamics are represented as

$$\Delta \dot{P}_{m_i}(t) = \frac{1}{T_{t_i}}\left(\Delta P_{g_i}(t) - \Delta P_{m_i}(t)\right) \tag{3.11}$$

where T_{t_i} is the turbine time constant.

3.1.6. Automatic Generation Control Feedback

As final individual dynamics, the AGC algorithm is displayed on the left hand side of Figure 3.4. AGC uses ramping adjustments for the generators in each balancing area, also called Area Control Error (ACE), which is comprised of a linear equation [203] represented in Equation 3.12, where β_i is the frequency bias factor. This ACE is then multiplied with the negative of the AGC integrator gain K_i as in Equation 3.13, after which it is sent as control input to the respective LFC areas.

$$ACE_i(t) = \beta_i \Delta f_i(t) + \Delta P_{tie_i}(t) \tag{3.12}$$

$$\Delta \dot{P}_{c_i} = -K_i \times ACE_i. \tag{3.13}$$

This algorithm runs every two to four seconds [12, 14, 32, 90, 192], henceforth denoted by τ. The AGC algorithm uses only the measurements it reads at the time of execution. This way, sequences of control inputs are created, denoted below, where $\lfloor\cdot\rfloor$ is the floor function.

$$u_{c_i}^t := \left\{u_{c_i}(0), \quad u_{c_i}(\tau), \quad \dots, \quad u_{c_i}\left(\left\lfloor\frac{t}{\tau}\right\rfloor\tau\right)\right\}. \tag{3.14}$$

3.2. Load Frequency Control State Space Model

From the discussed dynamics, the state space representation of the LFC model can be derived in the vicinity of a stable operating point. The LFC areas are separately modelled from the AGC algorithm. In this section, first the state space model of the areas is composed, followed by AGC. The specific simplified block schematics for this research is visualised in Figure 3.5.

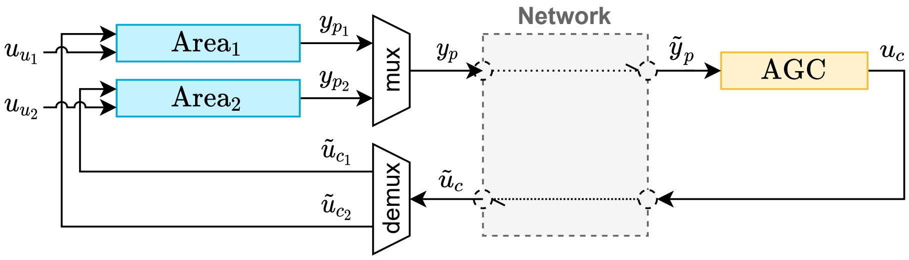


Figure 3.5: Simplified schematics of a two-area load frequency control system

3.2.1. Continuous-Time Area Models

As derived in subsection 3.1.1, the LFC areas are modelled individually. It is possible to incorporate the tie-lines in the state space either separately or in a combined fashion, of which the latter is chosen here. In the absence of an attack, the continuous-time state space representation for each area is:

$$\text{Area}_{c,i} : \begin{cases} \dot{x}_{p_i}(t) = A_{c,p_i} x_{p_i}(t) + B_{c,p_i} u_{p_i}(t) + \eta_i(t) \\ y_{p_i}(t) = C_{c,p_i} x_{p_i}(t) + \xi_i(t) \end{cases} \tag{3.15}$$

where

$$x_{p_i}(t) = \begin{bmatrix} \Delta f_i(t) & \Delta P_{m_i}(t) & \Delta P_{g_i}(t) & \Delta P_{tie_i}(t) \end{bmatrix}^T \tag{3.16}$$

$$u_{p_i}(t) = \begin{bmatrix} \tilde{u}_{c_i}(t) & u_{u_i}(t) \end{bmatrix}^T = \begin{bmatrix} \Delta \tilde{P}_{c_i}(t) & \Delta P_{L_i}(t) \end{bmatrix}^T \tag{3.17}$$

$$y_{p_i}(t) = \begin{bmatrix} \Delta f_i(t) & \Delta P_{tie_i}(t) \end{bmatrix}^T \tag{3.18}$$

are the states, inputs and outputs, respectively, and

$$A_{c,p_i} = \begin{bmatrix} \frac{-D_i}{2H_i} & \frac{1}{2H_i} & 0 & \frac{-1}{2H_i} \\ 0 & \frac{-1}{T_{t_i}} & \frac{1}{T_{t_i}} & 0 \\ \frac{-1}{R_{1_i} T_{g_i}} & 0 & \frac{-1}{T_{g_i}} & 0 \\ \sum_{j \in \delta_i} P_{s_{i,j}} & 0 & 0 & 0 \end{bmatrix}, \quad B_{c,p_i} = \begin{bmatrix} 0 & \frac{-1}{2H_i} \\ 0 & 0 \\ \frac{1}{T_{g_i}} & 0 \\ 0 & 0 \end{bmatrix}, \quad C_{c,p_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.19}$$

are the fixed continuous-time state space matrices, and $\eta(t)$ and $\xi(t)$ are the process and measurements noises, respectively, which both are of appropriate dimensions. These noises are modelled as identically distributed random sequences with zero mean. It is assumed these noises are unknown to the operator, but bounded [53]. Larger disturbances, such as entire disconnections of areas or mechanical failure, are not considered in this research.

Assumption 3.4. *Uncertainties $\eta(t)$ and $\xi(t)$ are unknown by the system operator, but their norms are bounded by known sequences $\bar{\eta}(t)$ and $\bar{\xi}(t)$.*

3.2.2. Continuous-Time Automatic Generation Control Model

For the modelling of AGC, a centralised approach was selected in subsection 3.1.1. In the absence of an attack, the continuous-time state space representation is:

$$\text{AGC}_c : \begin{cases} \dot{x}_c(t) = A_{c,c} x_c(t) + B_{c,c} \tilde{y}_p(t) \\ u_c(t) = C_{c,c} x_c(t) \end{cases} \tag{3.20}$$

where

$$x_c(t) = \begin{bmatrix} \Delta P_{c_1} & \Delta P_{c_2} \end{bmatrix}^T \tag{3.21}$$

$$\tilde{y}_p(t) = \begin{bmatrix} \Delta \tilde{f}_1(t) & \Delta \tilde{f}_2(t) & \Delta \tilde{P}_{tie_1}(t) \end{bmatrix}^T \tag{3.22}$$

$$u_c(t) = \begin{bmatrix} \Delta P_{c_1} & \Delta P_{c_2} \end{bmatrix}^T \tag{3.23}$$

are the states, outputs and inputs, respectively, and

$$A_{c,c} = \mathcal{O}_{2\times 2}, \quad B_{c,c} = \begin{bmatrix} -K_1\beta_1 & 0 & -K_1 \\ 0 & K_2\beta_2 & K_2 \end{bmatrix}, \quad C_{c,c} = I_2. \tag{3.24}$$

are the continuous-time state space matrices. $\mathcal{O}_{2\times 2}$ represents a null matrix and I_2 the identity matrix, both of dimension (2×2). Since AGC is a purely algorithmic process, no physical process or measurement noise are applicable.

Remark. *Due to the relationship of Equation 3.9, only $\Delta P_{tie_1}(t)$ is required to be sent across the network. Hence, $y_{p_2}(t)$ is reduced to containing $\Delta f_2(t)$ only, i.e. $y_{p_2}(t) = \Delta f_2(t)$, and $y_p(t), \tilde{y}_p(t) \in \mathbb{R}^{n_y N - 1}$.*

3.2.3. Discrete-Time Models

For numerical simulation on the HILDA testbed, the continuous-time models are discretized. This is due to the discrete nature of the power system data (which are first sampled before being transferred over telemetry links) [18]. The discrete-time model is achieved according to the zero-order hold (ZOH) method with a sample time of $t_s = 0.01$ seconds. The detailed derivation process resulting in the discrete-time models is not discussed in this thesis. For this, the reader is referred to [109]. The updated notations for the discrete-time state space models are provided in Equation 3.25 and Equation 3.26, where A_{d,p_i}, B_{d,p_i}, C_{d,p_i}, $A_{d,c}$, $B_{d,c}$ and $C_{d,c}$ are the newly discretized state space matrices, while $\eta_i[k]$ and $\xi_i[k]$ represent the discretized noise profiles. The rank of the controllability matrices is equal to the number of states, indicating that both (A_{c,p_i}, B_{c,p_i}) and (A_{d,p_i}, B_{d,p_i}) are controllable pairs.

$$\text{Area}_{d,i} : \begin{cases} x_{p_i}[k+1] = A_{d,p_i} x_{p_i}[k] + B_{d,p_i} u_{p_i}[k] + \eta_i[k] \\ \quad y_{p_i}[k] = C_{d,p_i} x_{p_i}[k] + \xi_i[k] \end{cases} \tag{3.25}$$

$$\text{AGC}_d : \begin{cases} x_c[k+1] = A_{d,c} x_c[k] + B_{d,c} \tilde{y}_p[k] \\ \quad u_p[k] = C_{d,c} x_c[k] \end{cases} \tag{3.26}$$

3.3. Data Integrity Attack Modelling

Most of the available research models data integrity attacks through some form of addition or subtraction applied to the original communication signal [12, 148, 173, 197]. However, modelling in this fashion would not include reconnaissance or MITM attacks: to read the original signal and replace it with malicious data, the modelling approach should allow communication interruption and substitution as well. To this end, the attacker $\mathcal{A}$ is modelled as MITM between the LFC areas and the AGC mechanism, as visualised in Figure 3.6. In the case of an attack, $\tilde{y}_p[k] = y_p'[k]$ and/or $\tilde{u}_c[k] = u_c'[k]$ (depending on which signal is attacked), where $y_p'[k]$ and $u_c'[k]$ represent the maliciously altered measurements and control inputs, respectively. The data integrity attack takes place during the attack time period κ_a, which is initiated at timestep k_0. For the attack scenario to occur, Assumption 3.5 is made.

Assumption 3.5. *Attacker $\mathcal{A}$ is able to infiltrate the network on which communication data between the LFC areas and the AGC mechanism are exchanged.*

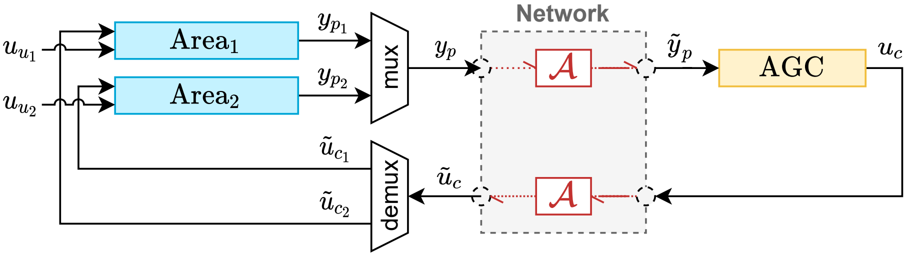


Figure 3.6: Simplified schematics of a two-area load frequency control system under a data integrity attack

Mathematically, the data integrity attack problem is formulated by Equation 3.27 and Equation 3.28. The attack functions for the scaling and replay attacks specifically are discussed in the next two subsections.

Assumption 3.6. *When $k \notin \kappa_a$, signals are not changed by the communication network in any way, i.e. $\tilde{y}_p[k] = y_p[k]$ and $\tilde{u}_c[k] = u_c[k]$.*

$$\tilde{y}_p[k] = \begin{cases} y_p[k] & \text{for } k \notin \kappa_a \\ y_p'[k] & \text{for } k \in \kappa_a \end{cases} \tag{3.27}$$

$$\tilde{u}_c[k] = \begin{cases} u_c[k] & \text{for } k \notin \kappa_a \\ u_c'[k] & \text{for } k \in \kappa_a \end{cases} \tag{3.28}$$

3.3.1. Scaling Attack

A scaling attack was already introduced in subsection 2.2.2. This is a 'naive' attack [59, 144], implying that it does not require any process knowledge of the attacked system. A scaling attack simply intercepts certain communication variables, after which the variables are altered through multiplication by an arbitrary scale and sent to their original destination without additional time delay [17, 24, 100]. This is mathematically formulated as

$$\begin{aligned} y_p'[k] &= \Gamma_y y_p[k], \\ u_c'[k] &= \Gamma_u u_c[k] \end{aligned} \tag{3.29}$$

where $\Gamma_y \in \mathbb{R}^{3\times3}$ and $\Gamma_u \in \mathbb{R}^{2\times2}$ are mapping matrices which can direct data corruption to respective signal transmission channels (e.g. for a multiplication of all accessory measurements by 2, $\Gamma_y = 2I_3$).

3.3.2. Replay Attack

Like the scaling attack, replay attacks were already introduced in subsection 2.2.2. Though preliminary research shows that these attacks are more difficult to detect [17, 24, 52], they too do not require any system knowledge. In replay attacks, the attacker $\mathcal{A}$ records a sequence of transmitted signals during period T, starting at $k_r = k_0 - T$. Then, starting at k_0, the original communication signal is replaced with the recorded sequence. This is mathematically formulated as

$$\begin{aligned} y_p'[k] &= \Psi_y y_p[k-T], \\ u_c'[k] &= \Psi_u u_c[k-T] \end{aligned} \tag{3.30}$$

where $\Psi_y \in \mathbb{R}^{3\times3}$ and $\Psi_u \in \mathbb{R}^{2\times2}$ are binary incidence mapping matrices that determine which variable sequence is replayed on which channel (e.g. for replaying all measurements on their accessory channels, $\Psi_y = I_3$). Only identity mapping matrices are considered in this study.

3.4. Conclusion

In this chapter, multiple LFC modelling approaches were presented, after which the split area with combined AGC approach was selected. This approach was used to model a two-area LFC system with a single generator per area. To this end, first the LFC dynamics were discussed, after which these dynamics were combined in state space representations. Finally, based on this LFC model, data integrity attacks were treated, more specifically scaling and replay attacks. In the next chapter, the IDS mechanism is discussed mathematically in the context of the LFC and attack models.

4

Design of Observer-Based Detection with Watermarking

This chapter describes the mathematical design of the implemented IDS mechanism. As already introduced, this is a combination of observer-based detection and active DMWM. The mathematical formulation of the observer-based detection mechanism is introduced first, including the criteria for proper functionality. This is followed by the formulation of the active DMWM mechanism. In the final section of this chapter, the mathematical proof behind detecting data integrity attacks through the priory designed IDS mechanism is discussed. The design is largely based on [1, 52, 53, 51, 55, 200].

4.1. Observer Design

To determine if an attack (or fault for that matter) has occured, observer-based IDS mechanisms estimate the normal physical behaviour of a process. If the measured values deviate too extensively from the estimated ones, an attack could be the cause. Together with the Kalman filter, the Luenberger observer is the best-known technique to dynamically create these estimations [125]. However, Kalman filters estimate the system states with noise [204], which in this study is not possible due to Assumption 3.4. Hence, this section clarifies the design of a Luenberger observer $\mathcal{D}$ (an abbreviation of 'Detector'). First, its placement/location in the system is discussed, followed by the state estimation leading to residual generation. The detailed functionality of a Luenberger observer is omitted here. For this, the reader is referred to [189].

4.1.1. Placement of Luenberger Observer

Similar to [55], $\mathcal{D}$ is located only at the control site of the LFC system (e.g. next to the AGC mechanism). This is visualised in Figure 4.1, where y_r is the combined residual vector of both LFC area 1 and 2, i.e. $y_r = \begin{bmatrix} y_{r_1} & y_{r_2} \end{bmatrix}^T \in \mathbb{R}^{n_y N-1}$. This residual will be used for the actual detection.

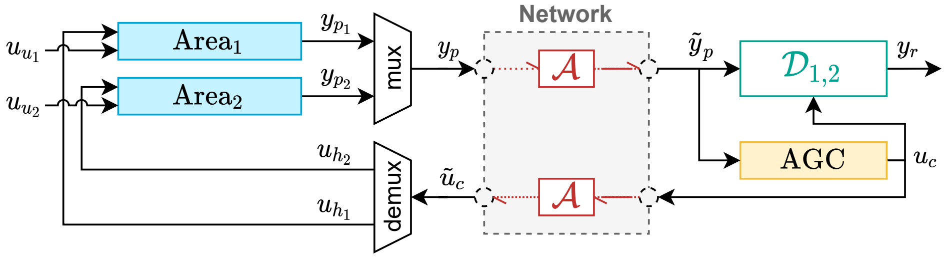


Figure 4.1: Simplified schematics of a two-area load frequency control system under a data integrity attack with detection

4.1.2. State Estimation and Residual Generation

By supplying the Luenberger observer with the control input generated by the AGC mechanism and the (possibly compromised) output measurements, it generates an estimated state and accompanying estimated output. It does so using the closed loop system model. The estimated output is subtracted from the measured output, resulting in $y_{r_i}[k] = |y_{p_i}[k] - \hat{y}_{p_i}[k]|$. $y_{r_i}[k]$ serves as observer output, but it also is fed back to the observer in a closed-loop fashion. In canonical form, $\mathcal{D}_i$ is:

$$\mathcal{D}_i : \begin{cases} \hat{x}_{p_i}[k+1] = A_{d,p_i}\hat{x}_{p_i}[k] + B_{d,p_i}u_{p_i}[k] + L_{d_i}y_{r_i}[k] \\ \quad \hat{y}_{p_i}[k] = C_{d,p_i}\hat{x}_{p_i}[k] \end{cases} \tag{4.1}$$

where $\hat{x}_{p_i}[k]$ and $\hat{y}_{p_i}[k]$ are the estimated states and outputs (of equal dimensions as the actual states and outputs), $\hat{x}_{p_i}[0] = 0$ is the initial state, and $L_{d_i} \in \mathbb{R}^{n_y \times n_u}$ is the static discretized Luenberger observer gain. The state space matrices are equal to those in Equation 3.25. The observer computes output residuals of the two LFC areas separately because otherwise, when scaling up the amount of LFC areas, the observer model might become too computationally expansive to handle real-time operation. For the observer to function, (A_{d,p_i}, C_{d,p_i}) should be a detectable pair [52, 53, 51, 55].

Assumption 4.1. (A_{d,p_i}, C_{d,p_i}) *is a detectable pair.*

4.2. Dynamic Multiplicative Watermarking Design

The idea of watermarking is to actively superimpose a signal (i.e. a watermark) on the original signal which is to be communicated over the network. This is done by sending the original signal through a watermarking filter. Intuitively, such a superimposition empowers system operators to arbitrarily control the transmitted signals, thereby increasing signal integrity control. In this study, this superimposition is done in a multiplicative manner, i.e. such that a watermark equalising filter on the other side of the network can revoke (i.e. equalise) the effect of the watermarker, thereby retrieving the original signal. Also, the watermarker and its equaliser are designed such that their parameters can switch over time. The state of the watermarker and its equaliser are parameterised by

$$\theta[k] := \theta_s \in \Theta \quad \text{for } k_s \le k < k_{s+1} \tag{4.2}$$

where $\Theta := \{\theta_1, ..., \theta_\Omega\}$ is the set of possible parameters (Ω being the total amount of parameters), and $\mathcal{K}_\theta := \{k_1, ..., k_s, ...\}$ is the set of switching times which trigger a watermarking state change [52].

4.2.1. Watermarker Placement

In this study, both the measurements and the control inputs are watermarked. This is similar to the approach in [55]. For the LFC model, this approach is visualised in Figure 4.2. Here, $\mathcal{W}(\theta[k])$ and $\mathcal{G}(\theta[k])$ represent the watermarking filters. These are able to individually watermark all communication signals. $\mathcal{Q}(\theta[k])$ and $\mathcal{H}(\theta[k])$ represent their respective equalising filters. The outputs of the watermarking and equalising filters are $y_w[k]$, $y_q[k]$, $u_g[k]$ and $u_h[k]$ for $\mathcal{W}(\theta[k])$, $\mathcal{Q}(\theta[k])$, $\mathcal{G}(\theta[k])$ and $\mathcal{H}(\theta[k])$, respectively. For the LFC model of this study, the watermarking and equalising filter sets would consist of $n_yN - 1$ and N filters, respectively. However, for simplicity of notation, the remainder of this section considers singular measurements and control inputs. Also for simplicity of notation, $(\theta[k])$ is omitted when deemed possible.

After denoting $\Sigma \in \{\mathcal{W}, \mathcal{G}\}$ and $\Phi \in \{\mathcal{Q}, \mathcal{H}\}$ as the sets of watermarking and equalising filters (i.e. the filters before and after the signal is sent over the network, as depicted in Figure 4.2), respectively, a combined notation of the filter state space model is composed:

$$\Sigma : \begin{cases} x_\sigma[k+1] = A_\sigma\left(\theta_\sigma[k]\right)x_\sigma[k] + B_\sigma\left(\theta_\sigma[k]\right)y_p[k] \\ \quad y_\sigma[k] = C_\sigma\left(\theta_\sigma[k]\right)x_\sigma[k] + D_\sigma\left(\theta_\sigma[k]\right)y_p[k] \end{cases} \tag{4.3}$$

$$\Phi : \begin{cases} x_\phi[k+1] = A_\phi\left(\theta_\phi[k]\right)x_\phi[k] + B_\phi\left(\theta_\phi[k]\right)y_p[k] \\ \quad y_\phi[k] = C_\phi\left(\theta_\phi[k]\right)x_\phi[k] + D_\phi\left(\theta_\phi[k]\right)y_p[k] \end{cases} \tag{4.4}$$

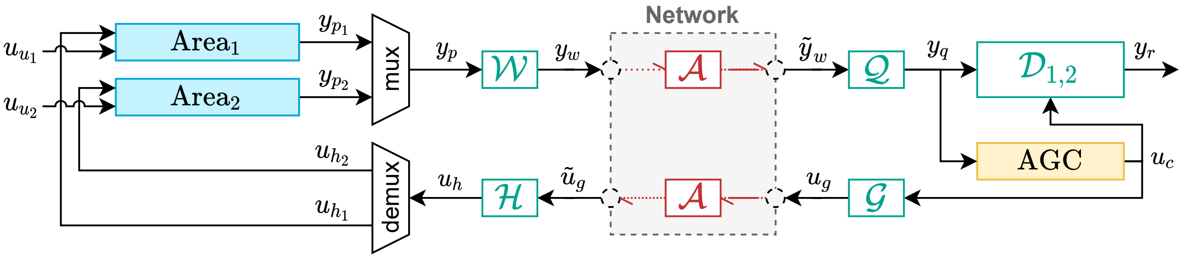


Figure 4.2: Simplified schematics of a two-area load frequency control system under a data integrity attack with detection and watermarking

where $\sigma \in \{w, g\}$ and $\phi \in \{q, h\}$ define what filter the states, inputs and outputs belong to [55]. It should be noted that despite their identical mathematical notation, the watermark parameters within the sets Σ and Φ do not have to be the same (i.e. $\mathcal{W}(\theta_s) \neq \mathcal{G}(\theta_s)$ and $\mathcal{Q}(\theta_s) \neq \mathcal{H}(\theta_s)$). In fact, when generating the results of this study (which are discussed in chapter 6), they are not.

4.2.2. Watermarking and Equalising Filter Design

$\mathcal{G}$ and $\mathcal{H}$ are designed to be the stable inverses of $\mathcal{W}$ and $\mathcal{Q}$, respectively, i.e. $\mathcal{Q} := \mathcal{W}^{-1}$ and $\mathcal{H} := \mathcal{G}^{-1}$. For this, the inverse of a system is to be defined as in Lemma 3.15 from [220]. This gives

$$\mathcal{Q}\mathcal{W} = I_{n_y N-1}, \quad \mathcal{H}\mathcal{G} = I_N. \tag{4.5}$$

As a result, Assumption 3.5 can be extended by stating that $y_q[k] = y_p[k]$ and $u_h[k] = u_c[k]$ in the absence of an attack. This way, system performance is not decreased in the absence of cyber-attacks. Consequently, no additional hardware updates are needed on existing generation units [90].

To design such appropriate filters, this study leverages the work of [52, 53]. The watermark generator is designed to be an Infinite Impulse Response (IIR) filter of order M. In canonical form, where $x_\sigma[k] \in \mathbb{R}^M$ and for $m = \{1, ..., M\}$, the watermarking filters of Equation 4.3 are represented by

$$A_\sigma = \begin{bmatrix} \mathcal{O}_{M-1,1} & I_{M-1} \\ \multicolumn{2}{c}{w_A^\top} \end{bmatrix}, \quad B_\sigma = \begin{bmatrix} \mathcal{O}_{M-1,1} \\ 1 \end{bmatrix}, \quad C_\sigma = \begin{bmatrix} \dots & w_{B,(m)} + w_{B,(0)} w_{A,(m)} & \dots \end{bmatrix}, \quad D_\sigma = w_{B,(0)} \tag{4.6}$$

where $w_A = \begin{bmatrix} w_{A,(1)} & \dots & w_{A,(M)} \end{bmatrix}^T \in \mathbb{R}^M$ and $w_B = \begin{bmatrix} w_{B,(0)} & \cdots & w_{B,(M)} \end{bmatrix}^T \in \mathbb{R}^{M+1}$ represent the filter parameters which are switchable over time. Correspondingly, in canonical form, where $x_\phi[k] \in \mathbb{R}^M$, the equalising filters of Equation 4.4 are represented by

$$A_\phi = \begin{bmatrix} \mathcal{O}_{M-1,1} & I_{M-1} \\ \multicolumn{2}{c}{\frac{-1}{w_{B,(0)}} w_B^T} \end{bmatrix}, \quad B_\phi = \begin{bmatrix} \mathcal{O}_{M-1,1} \\ \frac{1}{w_{B,(0)}} \end{bmatrix}, \quad C_\phi = \begin{bmatrix} \dots & -w_{A,(n)} - \frac{w_{B,(n)}}{w_{B,(0)}} & \dots \end{bmatrix}, \quad D_\phi = \frac{1}{w_{B,(0)}}. \tag{4.7}$$

Assumption 4.2. *$\mathcal{W}(\theta[k]), \mathcal{Q}(\theta[k]), \mathcal{G}(\theta[k])$ and $\mathcal{H}(\theta[k])$ are stable for all $\theta[k] \in \Theta$.*

Now, according to Theorem 2 from [52], using Assumption 4.2, it can be stated that there indeed is no performance loss when involving the watermark filters if, and only if, the states of Σ and Φ are such that $x_\sigma[k] = x_\phi[k]$ for all $k \in \mathcal{K}_\theta$. Also, [52] concludes that the proposed scheme can be used in a modular fashion because there is no performance reduction in the absence of attacks.

4.3. Detection of Data Integrity Attacks

To perform attack detection, some alerting mechanism needs to be constructed. In this case, this is the observer residual $y_r[k]$ which is being compared to a robust detection threshold $\bar{y}_r[k]$, both being produced by $\mathcal{D}_{1,2}$ from Figure 4.2. The alerting criterion is formalised as

$$y_r[k] \geq \bar{y}_r[k]. \tag{4.8}$$

The residual was already introduced earlier in subsection 4.1.2. Its dynamics, which are derived from [52, 53], are formalised as

$$y_{r_i}[k+1] = \left(A_{d,p_i} - L_{d_i} C_{d,p_i}\right) y_{r_i}[k], \tag{4.9}$$

where L_{d_i} should be designed such that $A_{r_i} := \left(A_{d,p_i} - L_{d_i} C_{d,p_i}\right)$ has all eigenvalues inside the unit circle, i.e. is Schur. This is achieved through the algorithm of [98], which places all closed-loop poles at certain specified locations. Through this placement, $y_r[k]$ converges to zero [189].

As detection threshold, the design of [52] is used. The residual is designed in a robust fashion, i.e. such that it can cope with the unknown disturbances of $\eta[k]$ and $\xi[k]$ (in the upcoming equations, the subscript i is omitted for simplicity of notation):

$$\bar{y}_{r,(n)}[k] = \alpha_n \left[\sum_{h=0}^{k-1} (\delta_n)^{k-1-h} \left(\bar{\eta}[h] + \|L_d\| \bar{\xi}[h]\right) + (\delta_n)^k \, \bar{x}_r[0] \right] + \bar{\xi}[k] \tag{4.10}$$

where subscript n denotes the various measurement signals, $\bar{\eta}, \bar{\xi}$ and $\bar{x}_r[0]$ are the upper bounds of the norms of η, ξ and $x_r[0]$, respectively, and α_n and δ_n are constants which meet the requirement that

$$\|C_{d,p,(n)} \, (A_r)^k \, \| \leq \alpha_n \, (\delta_n)^k \leq \|C_{d,p,(n)}\| \cdot \| \, (A_r)^k \, \| \tag{4.11}$$

with $C_{d,p,(n)}$ being the n-th row of the matrix $C_{d,p}$. Ones $y_{r,(n)}[k] \geq \bar{y}_{r,(n)}[k]$, a detection alarm is triggered and the detection step k_d is saved for later inspection. According to Theorem 5 from [51], if a time index $k_d > k_0$ and a component $n \in 1, \cdots, n_y$ exist such that, during a MITM attack, the inequality from Equation 4.12 holds, then detection at k_d is established.

$$\begin{aligned} | \, C_{d,p,(n)} & \left[\sum_{h=k_0}^{k_d-1} (A_r)^{k_d-1-h} \left(B_{d,p} \Delta u[h] - L_d \Delta y_p[h]\right) \right] + \Delta y_p[k] \\ & > 2\alpha_n \sum_{h=0}^{k_d-1} (\delta_n)^{k_d-1-h} \left(\bar{\eta}[h] + \|L_d\| \bar{\xi}[h]\right) + \\ & (\delta_n)^{k_d-k_0} \left(\alpha_n \bar{x}_r \, [k_0] + \bar{y}_{r,(i)} \, [k_0]\right) + 2\bar{\xi} \, [k_d] \end{aligned} \tag{4.12}$$

4.4. Conclusion

This chapter showed the design of the IDS applied in this study, which is a combination of a Luenberger observer and an active DMWM mechanism. The Luenberger observer is located next to the AGC mechanism, outputting a residual $y_r[k]$. The placement of the DMWM mechanism is done such that it increases the integrity of both the measurements and the control inputs travelling over the network. By designing them as switching IIR filters, is was made sure that no additional performance burden is imposed on the system, while theoretically increasing detection performance. This detection is to be executed through a comparison between the observer residual and a robustly designed threshold. This concludes the modelling and design of the LFC system, data integrity attacks and IDS. The next chapter is concerned with the design and configuration of the HILDA testbed.

5 Design and Configuration of the Testbed

In this chapter, the setup of the HILDA testbed is discussed. This HILDA testbed is to serve as prototype, based on which four similar testbeds (i.e. HILDA's sisters) will be deployed in the Systems and Control lab. As a result, this chapter serves two purposes: a handbook for the deployment of HILDA's sisters; and an explanation on how the specific preliminary models and designs of this study are numerically deployed on the HILDA testbed. To this end, this chapter first discusses the preliminaries (i.e. the motivation and constraints) regarding the testbed. This is followed by a description on the appliances. In the section thereafter, the configuration of this hardware is discussed through testbed schematics. Finally, the configuration of the network and the accompanying software is explained.

Remark. *This is a thesis for the programme of Systems and Control, in which IT is not actively studied. In order to facilitate future projects of Systems and Control, the required design and configuration steps regarding IT are elaborated equally deeply as Systems and Control related topics.*

5.1. Testbed Preliminaries

Before discussing the physical aspects of the HILDA testbed, the motivation for a testbed is emphasised. Also, there were some constraints the author had to cope with regarding the design and configuration of the testbed, which are discussed thereafter.

5.1.1. Testbed Motivation

Regarding cyber-security measures (or all technical measures for that matter), decidedly more studies validate performance using desktop simulations than testbeds, mainly because they are more expensive. However, this research argues that there are a few essential limitations to desktop simulations.

First of all, desktop simulations do not include natural system delays as they occur in real-world systems [164]. Examples of these delays are: communication delays, when messages need to travel between geographically distant locations; stack delays (as explained in subsection 2.1.3); and switching delays, both for the switching between different protocols (as also explained in subsection 2.1.3) and for the switching between analogue and digital signals. Due to the real-time property of ICSs, the control algorithm needs to be executed within a certain sample time. Whether or not this is properly possible despite delays should be taken into account in the validation. Though these delays can be simulated in desktop simulations, it self-evidently is more accurate to involve real-world delays.

Additionally, real-world ICS application have to cope with increased signal distortions. For example, when signals are sent over communication cables, a voltage drop occurs, resulting in a loss of signal accuracy [33]. Also, all physical signal transmissions are prone to noise, interference and crosstalk as a consequence of surrounding magnetic fields. Overall, testbeds form a unified environment in which communication, physical and control characteristics are accurately captured [14].

5.1.2. Testbed Constraints

Now that it is clear why cyber-security measures should be validated on testbeds, some constraints for this specific HILDA testbed design are delineated. First of all, the testbed was to be stationed at the `TU Delft` lab for Systems and Control. More specifically, it had to be placed on the 'computer wall', i.e. the desks and accompanying frame directly next to the hallway outside the lab. It was a strict requirement to design the testbed such that the visibility of the rest of the lab was kept as wholesome as possible. Also, the frame of this 'wall' consists mostly of fixed dimensions, so the dimensions of the testbed had to be designed appropriately. Next to physical lab constraints, the author would have to cope with constraints regarding PC administrator rights. For cyber-security and administrative reasons, these rights are limited for `TU Delft` students.

Also, it was already settled which hardware the testbed would consist of, without partaking of the author. This was done in the thesis of the author's predecessor, Ir. V. Ranade. Originally, the idea was that he would also implement the hardware, but due to the COVID-19 crisis, the last essential hardware devices were not delivered until six months after his graduation. The HILDA testbed in this study had to be designed in accordance with the already acquired hardware.

Lastly, the HILDA testbed had to be designed such that it could be extended to larger-scale experimentation, with five times as much PLCs and management PCs. The design of the large-scale setup is provided in Appendix C. This design involves a central Ethernet switch, which therefore also already had to be included in the HILDA testbed design.

5.2. Testbed Appliances

In this section, the most dominant control appliances used in the HILDA testbed are canvassed. First, the required appliances are derived from the model constructed in chapter 3 and chapter 4. Then, for each required appliance, the chosen hardware is discussed together with its functionalities in brief. The reader is expected to have a basic understanding of electrical engineering. For more detailed information, the reader is referred to the cited documentations or the respective product pages.

5.2.1. Required Appliances

To be able to simulate the constructed LFC model with dynamic watermarking being subjected to a data integrity attack, all model segments have to somehow be represented. This representation is visualised in Figure 5.1. The physical process of both LFC areas is simulated on a real-time simulator from `dSPACE`. This simulator outputs analogue 'measurements', which are transformed to an `EtherCAT` protocol by I/O modules. A PLC then watermarks the measurements with filter $\mathcal{W}$ and sends them to another PLC over an Ethernet switch. This switch is simulating a network, and is hence also the connection point for the MITM attack PC. The hierarchically highest PLC, i.e. PLC 1, removes the watermark through filter $\mathcal{Q}$ and executes the AGC and observer algorithms. The control input is filtered by $\mathcal{G}$ and sent to PLC 2 via the same (compromised) switch as before. PLC 2 removes the watermark through filter $\mathcal{H}$, after which the `EtherCAT` messages are transformed to analogue control signals which are provided as input to the simulator. The simulator, both PLCs, the I/O modules and the Ethernet switch are all configured with a management PC. This management PC also serves as HMI by providing analysis and control tools to system operators. These control appliances all depend on proper communication and power wiring, which is discussed in further detail in section 5.3.

5.2.2. Real-Time Simulator: `dSPACE` MicroAutoBox II 1401/1513/1514

Real-time simulators accurately mimic physical ICS processes. To this end, real-time analogue or digital in- and outputs can be generated and communicated to other hardware through connection pins. The specific simulator model used in this study is the MicroAutoBox (MAB) II 1401/1513/1514. Its PPC750 GL Power PC processor with 900 MHz clock frequency is capable of real-time simulations. The base board is further equipped with 8 MB of global RAM, 16 MB of local RAM, and 16 MB flash memory. Whether or not a simulation is running is indicated through a status Light-Emitting Diode (LED). As software interface, it uses the `Real-Time Interface (RTI)` library, which can be integrated into `Simulink`. `ControlDesk` Version 6.3 is optionally used as development environment. Its power input is a 7-pin male Sub-D connector, through which the device is also grounded and which is accompanied by a status LED. Flexible I/O extensions make the MAB II suited for many simulation varieties [72].

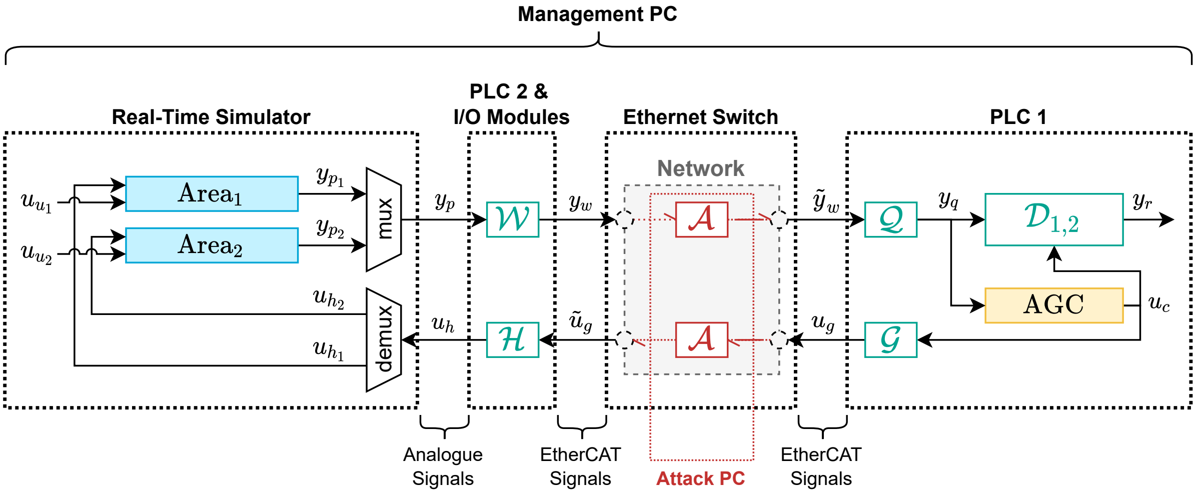


Figure 5.1: Deployment of the simplified schematics on each testbed component

A few items come with the MAB II. These are all for connecting it to a power source and to external I/O. The dominant I/O modules of the MAB II are two female 156-pin Zero-Insertion Force (ZIF) connectors (of type DS1513 and DS1514), which are complemented with included male connectors. Through the connection between the two, various types of communication signals can be exchanged. This study exercises the analogue inputs and outputs of type 1 only, which use an Analogue-to-Digital Converter (ADC) and a Digital-to-Analogue Converter (DAC), respectively, to communicate with the MAB II base board. Each individual analogue signal requires a dedicated ground wire (i.e. reference potential wire) from the external I/O to the dSPACE I/O connector. The Hardware Installation and Configuration documentation is only available to customers of dSPACE.

Remark. *Two male ZIF connectors have been ordered separately for this thesis due to loss of the ones that came with the original package.*

5.2.3. Programmable Logic Controller: Beckhoff C6030-0060

Beckhoff is a developer and distributor of industrial control equipment, such as PLCs and HMIs, and the accompanying technologies (softwares). It uses TwinCAT (which stands for 'The Windows Control and Automation Technology') as configuration and computation software. To execute the control and watermarking algorithms of this study, the Beckhoff C6030-0060 Industrial PC (IPC) is used in a dual fashion. An IPC is similar to a regular PC, except for the fact that it is designed to be deployed in harsher industrial environments (i.e. it is more compact, able to handle large temperature variations, easily mountable etc.). The C6030-0060 type are equipped with two Intel Core processors, (evidently) capable of handling real-time control. The C6030-0060 offers four Ethernet RJ45, four Universal Serial Bus (USB) and two DisplayPort (DP) connections, of which this study only uses the first type. It runs on a rated voltage of 24V Direct Current (DC) power. This is supplied through 2x2-pin voltage sockets, which accommodate wire cross sections up to 1.5mm^2. Correct power supply is verified through the PWR LED, while activity on the hard drive is indicated by the HDD LED.

The IPC runs on a Windows Operating System (OS). In the case of this study, the TwinCAT 3 automation software platform is used to configure the IPC. TwinCAT 3 consists of two essential environments: eXtended Automation Engineering (XAE) and eXtended Automation Runtime (XAR). A simplified view of their functionalities is depicted in Figure 5.2. The XAE environment is the programming environment, running either on the IPC or a linked PC. As development environment, it either uses Microsoft Visual Studio or a stripped version of it, referred to as the TwinCAT XAE Shell (or TcXaeShell). In this environment, TwinCAT projects can be constructed, in which developers can write code based on the object-oriented IEC 61131-3 standard or C/C++ [71]. TwinCAT is accompanied by a large amount of additive products which expand its capabilities (e.g. by integrating MATLAB & Simulink, enabling HMI integration etc.). These products fall into three categories: TExxxx (engineering extensions); TC1xxx (forming the basis of XAR); and TFxxxx (additional functions for XAR).

All can be deployed onto XAE through libraries. Accessing these libraries requires licences, which can be acquired separately, or appraised through one-week trial licences (though these trail versions often come with certain restrictions). The XAE environment interacts with the XAR environment in a duplex fashion. The runtime environment is where the programmes are executed by a real-time kernel. This environment can communicate with sensors, actuators and other TwinCAT devices through its real-time network drivers. Developers can adjust the configuration of the IPC when TwinCAT is in 'Configuration Mode', while the programme can run only when in 'Run Mode'. Which mode the IPC is currently in is indicated by the TC LED on the C6030-0060. More detailed information, including links to full documentations, is found in the information system (or INFOSYS) database [65].

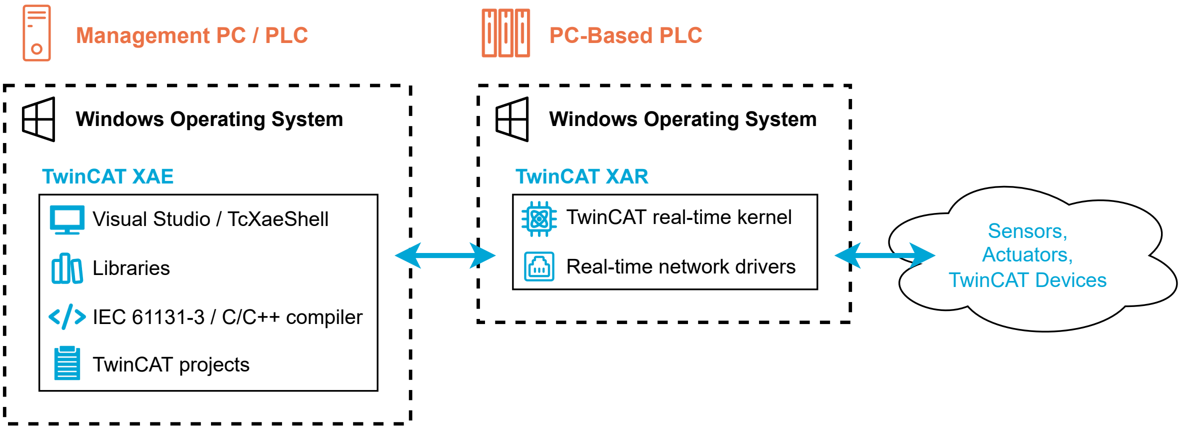


Figure 5.2: Simplified schematics of TwinCAT environments

Remark. *When ordering the C6030-0060 IPCs, three licences were acquired per device (next to the essential TE1000 (TwinCAT 3 Engineering) licence): TC1200 (TwinCAT 3 PLC), enabling the creation of PLC projects, thereby using the IPC as PLC; TF1800 (TwinCAT 3 PLC HMI), allowing for visualisation integrated in the PLC project; and TF6250 (TwinCAT 3 MODBUS TCP), empowering the use of the MODBUS communication protocol.*

Remark. *The licences and separate voltage sockets can be found in their respective envelopes.*

Remark. *A total of ten C6030-0060 IPCs were available to the author.*

5.2.4. Input/Output Modules: Beckhoff EK1100 and ELx004

To establish communication between the simulator and the PLCs, Beckhoff I/O modules are utilised. These modules can assemble communication of a certain type, transform them into another type and/or combine them, and subsequently send them through to a destination. For example, multiple analogue measurement signals from sensors can be combined and send through as MODBUS protocol.

The central element of the I/O module used in this study is the EK1100 EtherCAT Coupler. This device forms the link between an EtherCAT Device Protocol [76], which is sent over an EtherCAT network, and EtherCAT Terminals, which can be of type ELxxxx, ESxxxx or EMxxxx. For connection with the EtherCAT network, it has two Ethernet RJ45 connections (one for regular in- and output, and one for connecting further EtherCAT devices in the same strand). On the other side, a multitude of terminals can be added, which can process different kinds and amounts of communication signals. The whole module (i.e. the EK1100 and its hooked terminals) is powered by two rated voltages of 24V DC power: one is used to power the EK1100; the other is used as power source for the communication outputs (in wired communication, the physical transmission of messages is done using current or voltage).

In this study, only analogue signals have to be translated to the EtherCAT protocol, which is done by the EL30xx and EL40xx lines for inputs and outputs, respectively. More specifically, the EL3004 and EL4004 are employed, which are capable of handling four in- and outputs, respectively. These in- and outputs are single-ended, meaning that they only travel over a single conductor. This is opposed to differential signalling, in which two conductors carry signals of equal magnitude but opposite polarity (to reduce noise impact) [153]. The EL3004 can process signals in a range between -10V and $+10\text{V}$. It does so with a resolution of 12 bits. The same resolution is applied by the EL4004, although its output range is between 0V and +10V. All signals are accompanied by a reference voltage signal.

Remark. *Next to the EL3004 and EL4004, EL1008 and EL2008 terminals have been acquired, which are capable of handling digital in- and outputs, respectively. A total of ten EK1100 Couplers, EL1008 terminals, EL2008 terminals, EL3004 terminals and EL4004 terminals were available to the author.*

5.2.5. Ethernet Switch: HP E3800 48G-4SFP+ (J9574A)

Ethernet switches are capable of connecting the Ethernet adaptors of devices such as computers, thus creating a Local Area Network (LAN). They manage communication traffic between the connected devices by directing incoming and outgoing data. This data is encapsulated in Ethernet frames. The switch used in this study, the HP E3800 48G-4SFP+ (J9574A), contains a total of 48 ports. It can handle up to 10 Gigabit Ethernet protocol, although for this study only 1 Gigabit is required. To implement the model on the testbed as in Figure 5.1, it needs to connect both PLCs and form a node for the MITM attack PC. Also, it needs to connect the management PC to all devices (this could also be done through direct connections, but this would require installing additional Ethernet adaptors on the PC).

5.2.6. Management and Attack PC

The lab PC used for this study is the one stationed at Computer Wall Desk 3 in the lab for Systems and Control. This PC has an `Intel Xeon` processor with eight cores. It houses both the management PC and the malicious attack PC. The management PC performs multiple tasks: developing and deploying the LFC areas onto the MAB II via the `ControlDesk` software; developing the PLC programme in `TwinCAT` which then can be deployed to the `XAR` environments of the PLCs; and serving as HMI during the simulations. It would also have been possible to perform the PLC configuration directly on the C6030-0060 IPCs by connecting a mouse, a keyboard and a monitor, or via a Remote Desktop Connection (RDC) on the management PC. For both options, `TwinCAT XAE` would need to be additionally installed on the IPCs, since by default only `TwinCAT XAR` is installed. A disadvantage of such a setup is the local directory, so projects would have to be saved externally to prevent loss.

The attack PC runs on a Virtual Machine (VM) from `Oracle Corporation`. A VM is capable of realising an additional or replacement OS on a host PC. It does so as a hypervisor. This is an application which either shares or takes over the host PC hardware capabilities, such as CPU, on which a different operating system can be run. `Oracle VM VirtualBox` is a type 2 hypervisor. This type of hypervisor is installed on an already existing OS (`Windows` in this case), called the Host OS, which allows for the usage of its resources to realise a second OS (such as `Kali Linux` or `Windows`), called the Guest OS. A type 1 hypervisor would actually replace the existing OS entirely and directly control all the hardware, instead of having to share it.

Remark. *A deliberate choice was made to use the Computer Wall Desk 3, as this is one of the desks which is best visible from the hallway, thereby having the most exposure.*

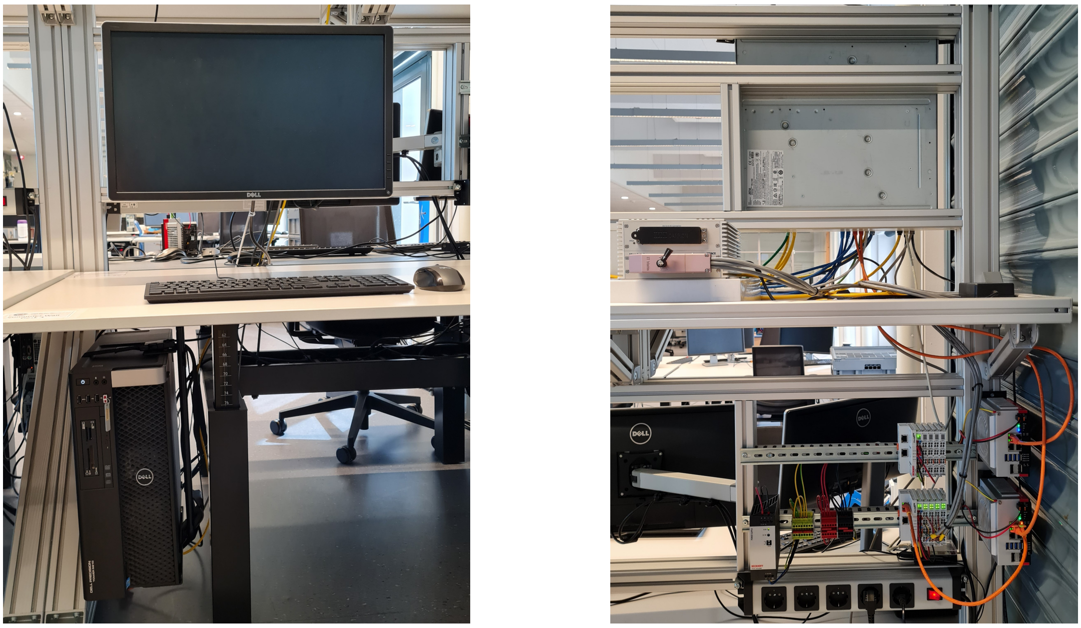
5.2.7. Power Supply: Beckhoff PS1011-2410-0000

Both the C6030-0060 and the I/O modules require a 24V DC power source. The source is provided by a power supply, in the case of this study a `Beckhoff` PS1011-2410-0000 [66]. It contains a full bridge rectifier. This transforms a $100 - 240$V wide-range AC input into a 24V 10A DC output with an efficiency of up to 95.2%. Proper DC output (i.e. a voltage above 18V) is verified through the DC OK status LED.

Remark. *A total of five PS1011-2410-0000 power supplies were available to the author. Each supply is capable of delivering DC power to (at least) two PLCs and two I/O modules simultaneously.*

5.3. Hardware Schematics

A goal of this chapter is to provide a handbook for the deployment of HILDA's sisters. To this end, this section discusses the mounting and wiring schematics in which HILDA is physically configured. The result, i.e. the mounted and wired HILDA testbed, is visualised in Figure 5.3. In Figure 5.3, the HILDA testbed is divided into two segments, a left on and a right one.



(a) Photograph of the left hand side of the testbed, with the desktop monitor above, and the management PC (which also houses the attack PC) below

(b) Photograph of the right hand side of the testbed, with the Ethernet switch above, the dSPACE on the left, and the Beckhoff equipment on the bottom right

Figure 5.3: Photographs of the testbed inside the lab for Systems and Control

Remark. *Next to the already discussed appliances, more ICS parts were required for the construction of the HILDA testbed. An ordering list with products from mainly RS Components is provided in Appendix B. This list also clarifies what spare parts still are available for future use of the testbed. On top of that, multiple useful tutorials for testbed construction are provided in Appendix D.*

5.3.1. Mounting Schematics

As already introduced, the testbed is placed on the computer wall at the TU Delft lab for Systems and Control. This is done in such a way that the visibility of the rest of the lab remains as good as possible. The frame of the computer wall consists of custom sized beams and other parts from Item Industrietechnik GmbH. For proper mounting, 35mm DIN rail (conform to the EN 60715 standard) is used. Regarding spacing, the dimensions of free space from Table 5.1 should be taken into account. This table also summarises the tools with which the devices can be mounted, and the orientation in which they should be mounted. The Ethernet switch and lab PC are not considered, since these were already properly mounted before the start of this study.

			Minimal Free Space [mm]			
Devices	Mounting	Orientation	Left	Right	Top	Bottom
dSPACE Simulator	M5 screws	Not specified	- - - - - - Not specified - - - - - -			
Beckhoff PLC	M4 screws	Vertical only	50	50	50	50
Beckhoff I/O Modules	DIN rail	Any orientation	20	20	35	35
Beckhoff Power Supply	DIN rail	Not specified	15	15	40	20

Table 5.1: Summary of device mounting requirements

The dSPACE simulator is designed such that it can be used for in-vehicle simulations, which explains the mounting holes at the sides of the simulator. For this study, it suffices to simply place the simulator on the shelf of the computer wall without securing it. The PLCs, I/O modules and power supply are mounted below this shelf, right above a power socket dedicated to the HILDA testbed. The layout is designed as in Figure 5.4 (the dimensions of the devices themselves can be found in [65]). The PLCs are placed on the outer vertical bars of the computer wall frame, horizontally centred. To properly secure the PLCs on the frame, M4 screws and specially designed Item nuts are required. The screws need to be tightened after hanging up the PLCs. The DIN rail is secured on the inner vertical bars of the frame, also with M4 screws and the same specially designed Item nuts. The power supply is placed on the left hand side of the lower DIN rail, which is closest to the power socket to which is should be connected. The I/O modules are also placed on the DIN rail, but then on the right side, closer to the PLCs they should eventually be connected to.

Remark. *For redundancy and future testbed usage, the designed layout consists of two DIN rail and I/O module combinations. This study only requires the combination next to PLC 2.*

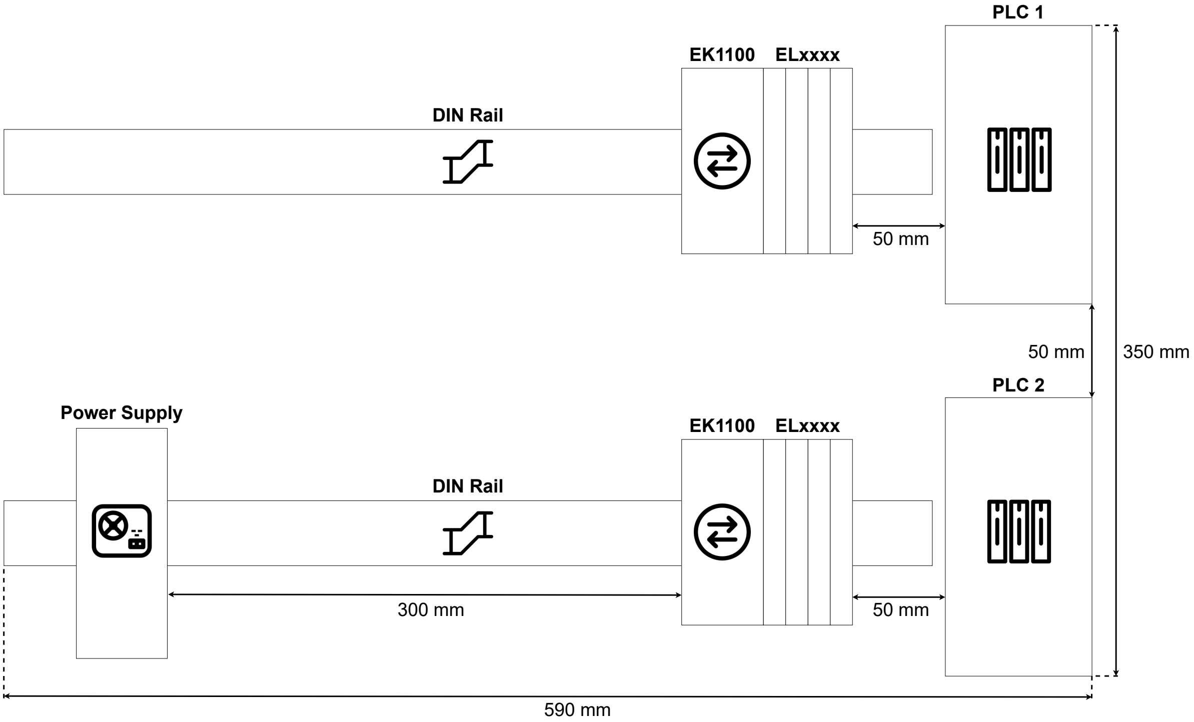


Figure 5.4: Front perspective of the Beckhoff equipment and DIN rail placement (on scale)

5.3.2. Power Wiring Schematics

The power and communication wiring schematics are designed in accordance with Kirchhoff's Circuit Laws (which state that the sum of currents in a node is zero, and that the sum op potential differences in a closed loop should also be zero) [130]. Two important aspects are discussed in this section: wire criteria and grounding. The entire wiring schematics is visualised in Figure 5.5.

Remark. *Because the HILDA testbed is also designed for future use, extra care was taken to ensure orderly installation. To this end, the author attached a wiring duct to the back of each DIN rail. Also, power terminal blocks and their appropriate jumpers, both from Phoenix Contact [42], are installed to properly coordinate DC and ground wiring. Lastly, all wire ends are encapsulated by ferrules.*

Wire criteria: The flow of electrons causes heat due to the present resistance (resulting in electrical energy being transformed to heat energy). To be able to have a sufficiently low cable resistance and prevent hazardous situations, a few wire aspects are crucial: American Wire Gauge (AWG) (which is a measure of wire cross-section), wire length, ampacity (the maximum current a conductor can carry continuously), resistivity (how strongly does the conductor resist electrical current) and wire temperature (since resistance increases with wire temperature). The required wire diameter is calculated by

$$A = \frac{2 \cdot l \cdot I}{\gamma \cdot U_a} \text{ for DC, and } A = \frac{2 \cdot l \cdot I \cdot \cos(\phi)}{\gamma \cdot U_a} \text{ for single-phase AC,}$$

where A is the cross-sectional area in square meters, l is the cable length in meters, I is the rated current in amperes, γ is the conductivity of the conductor in Siemens/meter (S/m), U_a is the permissible voltage drop in %, and $\cos(\phi)$ is the electrical efficiency of the system. Single phase AC is considered

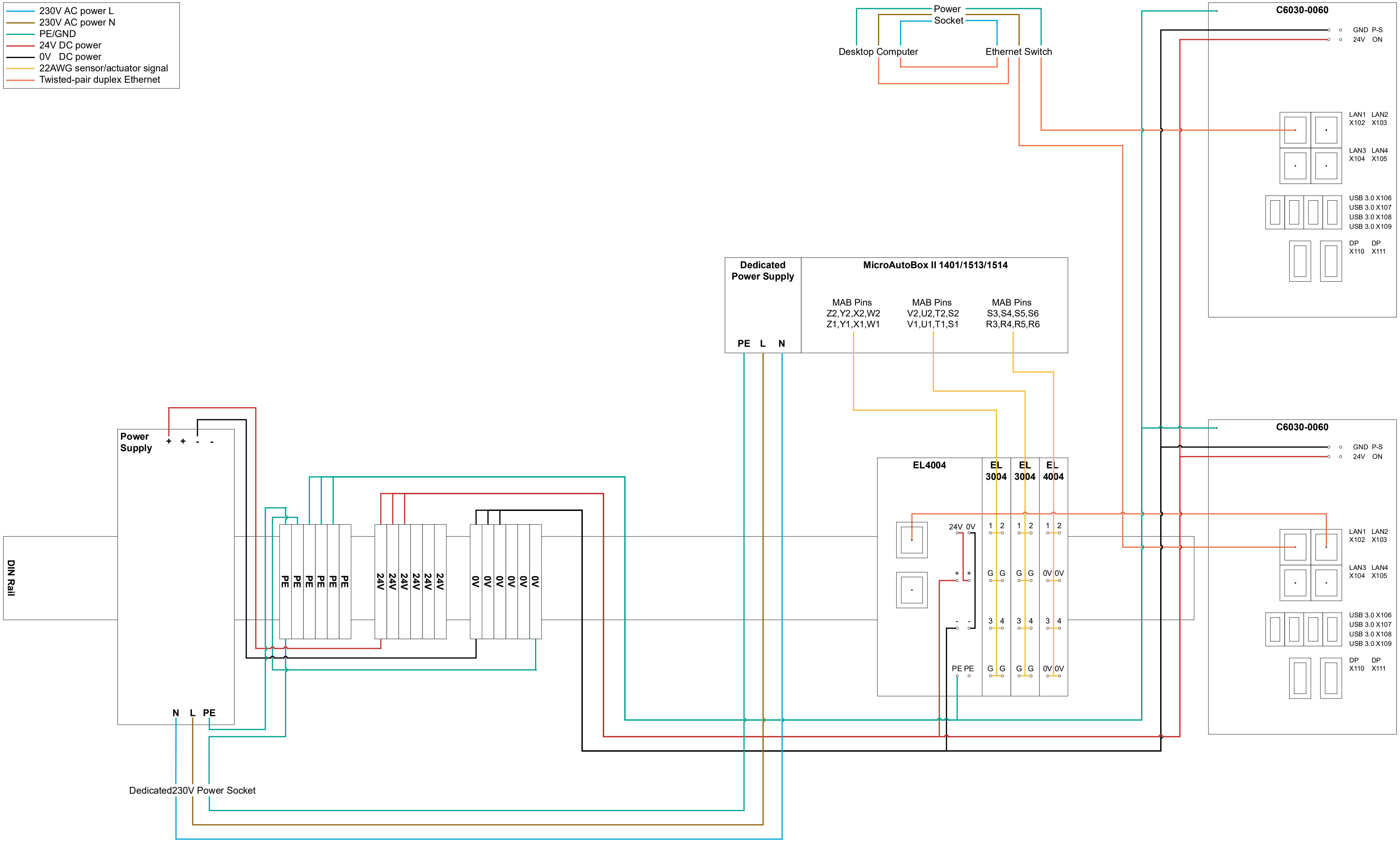


Figure 5.5: Wiring schematics of the testbed

since this is the type of current coming out of the power sockets in the lab. This study uses *flexible plain annealed copper* as conductor for the wiring, which has a conductivity of around 58S/m and a maximum rated current of 17A. For DC wiring, with a rated voltage of 24V and current of 10A, while considering a permissible voltage drop of 8.3% (the PLCs require at least 22V, which means a 8.3% drop) and a cable length of 2m (which is more than sufficient looking at Figure 5.4), the minimum cross-section is 0.35mm^2 (or 21 AWG). For AC wiring, with a rated voltage of 230V and current of 16A, while considering a voltage drop of 1% and a cable length of 2m, the minimum cross-section is 0.48mm^2 (or 20 AWG). However, less voltage drop is preferred, and the cables might need to be used more flexibly in the future, so eventually a cross-section of 1mm^2 is selected. All devices allow this cross-section at their respective input and output slots. The wiring placement is visualised in Figure 5.5.

Grounding: To ensure safety and minimise signal distortion, proper electrical grounding should be arranged. First of all, all electrically conducting chassis need to be able to rid any potential differences to prevent shock hazards. Also, the electrical circuit should have a central gateway, which is done by involving interconnected ground terminal blocks. These are connected directly to the dedicated testbed power socket, which supplies power to the power supply and the dSPACE simulator. Figure 5.5 involves another power socket, which supplies power to the Ethernet switch and the lab PC. Both power sockets are connected to a central lab power socket, which is consequently also the central ground gateway for the entire HILDA testbed. Figure 5.5 shows the grounding cable placement.

5.3.3. Communication Wiring Schematics

As shown in Figure 5.1, two types of communication need to be established: EtherCAT and analogue. EtherCAT travels over Ethernet cables. In this configuration, Cat6 cables are used, which supports communication speeds up to 1 Gbps and 250 MHz up to 100m according to the TIA 568.2-D standard. All devices of the HILDA testbed are capable of handling this speed, so any lesser capable Ethernet cable (such as Cat5) would be at the expense of testbed performance. The selected cables are RJ45 to RJ45. Their deployment is visualised in Figure 5.5.

As was briefly mentioned in subsection 5.2.4, the I/O modules use single-ended analogue communication. This implies that a voltage signal travels over a single conductor, which should be accompanied by a reference potential conductor. (This is opposed to differential communication, for which the reader is referred to [130]). The wires should be shielded to minimise interference from external electromagnetic fields of nearby devices. To connect all eight ports of each individual ELxxxx terminal, a 30m shielded Alpha Wire cable is selected with eight 0.35mm^2 (22 AWG) cores. The pins of the dSPACE ZIF connector demand a cross-section between 20 and 22 AWG, so this cable suffices (an even lower AWG would further decrease the cable flexibility). After consultation from the ”ideal grounding” schematics in the MAB II Hardware Installation & Configuration manual, and after e-mail contact with dSPACE, the analogue communication wiring is configured as in Figure 5.5. An overview of the wired connections between the ZIF connector and the ELxxxx terminals is provided in Table 5.2, where the Analogue Output (AO) and Analogue Input (AI) are viewed from the perspective of the two-area LFC system simulated on the MAB II. At the ZIF connector, all signals are referenced to the GND pin closest to the DAC or ADC pins. Eventually, no shield grounding is applied, based on the tailored advice from dSPACE.

5.4. Software Configuration

Thus far, this chapter has provided information on the individual control appliances and the hardware schematics. With everything in place physically, proper software configuration remains. For this configuration, mainly the respective installation manuals were utilised. For some occasions, third party guides were consulted. This section walks through the software configuration in a minimalistic but comprehensive fashion. For more detailed information, the reader is referred to either the respective manuals, or the third party tutorials, of which an overview is provided in Appendix D.

Remark. *The model is first implemented in MATLAB & Simulink before being deployed on the HILDA testbed. This is due to two reasons: 1) because the author was familiar with MATLAB & Simulink as computing environment, and 2) because especially the dSPACE software is designed to be easily integrated with MATLAB & Simulink. MATLAB & Simulink is therefore used as basis, though it would also be possible to implement the model on the testbed directly.*

#	Signal	Type	Source	Destination	Description
1	$\Delta f_1[k]$	AO	dSPACE DAC pin Z2	EL3004 1 input 1	Frequency variation of area 1
2	$\Delta f_2[k]$	AO	dSPACE DAC pin Y2	EL3004 1 input 2	Frequency variation of area 2
3	$\Delta P_{tie_1}[k]$	AO	dSPACE DAC pin X2	EL3004 1 input 3	Tie-line power variation of area 1
4	-	-	dSPACE DAC pin W2	EL3004 1 input 4	Signal connection for future use
5	-	-	dSPACE DAC pin V2	EL3004 2 input 1	Signal connection for future use
6	-	-	dSPACE DAC pin U2	EL3004 2 input 2	Signal connection for future use
7	-	-	dSPACE DAC pin T2	EL3004 2 input 3	Signal connection for future use
8	-	-	dSPACE DAC pin S2	EL3004 2 input 4	Signal connection for future use
9	-	-	dSPACE DAC pin Z1	EL3004 1 GND 1	Reference potential for signal 1
10	-	-	dSPACE DAC pin Y1	EL3004 1 GND 2	Reference potential for signal 2
11	-	-	dSPACE DAC pin X1	EL3004 1 GND 3	Reference potential for signal 3
12	-	-	dSPACE DAC pin W1	EL3004 1 GND 4	Reference potential for signal 4
13	-	-	dSPACE DAC pin V1	EL3004 2 GND 1	Reference potential for signal 5
14	-	-	dSPACE DAC pin U1	EL3004 2 GND 2	Reference potential for signal 6
15	-	-	dSPACE DAC pin T1	EL3004 2 GND 3	Reference potential for signal 7
16	-	-	dSPACE DAC pin S1	EL3004 2 GND 4	Reference potential for signal 8
17	$\Delta P_{c_1}[k]$	AI	EL4004 output 1	dSPACE ADC pin S3	Control input area 1
18	$\Delta P_{c_2}[k]$	AI	EL4004 output 2	dSPACE ADC pin S4	Control input area 2
19	-	-	EL4004 output 3	dSPACE ADC pin S5	Signal connection for future use
20	-	-	EL4004 output 4	dSPACE ADC pin S6	Signal connection for future use
21	-	-	EL4004 0V 1	dSPACE ADC pin R3	Reference potential for signal 17
22	-	-	EL4004 0V 2	dSPACE ADC pin R4	Reference potential for signal 18
23	-	-	EL4004 0V 3	dSPACE ADC pin R5	Reference potential for signal 19
24	-	-	EL4004 0V 4	dSPACE ADC pin R6	Reference potential for signal 20

Table 5.2: Overview of all analogue outputs, analogue inputs and remaining wired connections from the perspective of the dSPACE simulator

Remark. *The used version of MATLAB & Simulink is R2020b. This is compatible with the Beckhoff TE versions 2.x.xxxx.x and higher, instead of 1.2.xxxx.x and lower. According to the dSPACE documentation of version 2018-A, which is the one used in this study, the MATLAB & Simulink version R2018a or older should be used. However, the author did not encounter any issues with the configuration of the MATLAB & Simulink model on the dSPACE simulator.*

Remark. *Since this is a PC from the TU Delft, special administrator rights to install this software has to be requested. The lab technicians (specifically Will van Geest during the time of writing) can ensure these rights are granted. For this, a NetID and the PC Asset Tag are required.*

5.4.1. Local Area Network Configuration

For devices to be able to find each other, IP addresses and MAC addresses should be defined. Intuitively, in the eyes of Ethernet devices, IP addresses are the device names, while MAC addresses are their residences. To be able to transport a message across a network to a specific receiving device, a sending device needs to know both the IP and the MAC address. All relevant addresses for this study are provided in Table 5.3. In this table, it is also indicated whether the addresses are adjustable in this setup, or that they are fixed. Table 5.3 also includes AMS NetIDs. These are additional identifications for Beckhoff equipment. As was explained in subsection 2.1.3, EAP works according to a Publisher/-Subscriber relationship. A TwinCAT device can have multiple Publishers and Subscribers, which share the same IP address and possibly also the same MAC address. AMS NetIDs are used to make a distinction between these Publishers and Subscribers. Each Publisher/Subscriber receives an own AMS NetID, which only slightly differs from the base AMS NetID (e.g. if the base ID is xxx.xx.xxx.xxx.1.1, the first default Publisher ID is xxx.xx.xxx.xxx.2.1). The AMS NetID is also used to configure Subscribers to only receive information from a specific Publisher and reject all others. All Beckhoff devices have a MAC address which starts with the same Organisationally Unique Identifier (OUI), which is 00:01:05. To accustom the Attack PC to the Beckhoff equipment, it is given the same OUI. The default gateway should be the same for all devices, though is will not be used for the experiments since the information does not have to leave the local network (172.19.3.1-254).

Remark. *All MAC addresses of the connected devices have to be registered on the switch. Otherwise, the messages originating from these devices are rejected. The registration can be performed by the lab technicians.*

	IP Address		MAC Address		AMS Net ID	
Network Device	Address	Adjustable	Address	Adjustable	Base ID	Adjustable
Management PC	172.19.3.242	✓	64:00:6A:74:1E:81	X	172.19.3.51.1.1	✓
Attack PC	172.19.3.239	✓	00:01:05:4B:5B:D7	✓	172.18.237.169.1.1	✓
MAB II Simulator	172.19.3.44	✓	64:4D:70:00:97:C3	X	-	-
PLC 1 Adaptor 1	172.19.3.240	✓	00:01:05:66:D5:FC	X	172.18.237.177.1.1	✓
PLC 2 Adaptor 1	172.19.3.241	✓	00:01:05:66:DC:1C	X	172.18.237.162.1.1	✓
Default Gateway	172.19.3.1	X	E8:1C:BA:35:F9:EF	X	-	-

Table 5.3: Testbed devices and their addresses

5.4.2. Simulator Software: `Real-Time Interface` and `ControlDesk`

To get the LFC model from the `MATLAB & Simulink` simulation and run it on the `dSPACE` MAB II simulator, two softwares are used: `RTI` and `ControlDesk`. The specific versions for this study are `RTI1401` and `ControlDesk` 6.3. `RTI` forms the link between the `dSPACE` HIL applications and the `MATLAB & Simulink` development environment, while `ControlDesk` is the simulation and control environment of the MAB II in which experiments can be initiated and tracked. The softwares can be installed using the accessory `dSPACE` installation disks. To perform the installation process, the reader is referred to the installation manuals. One note: to execute the installation application on this disk, the *WSUS* server should temporarily be bypassed.

Ones installed, the appropriate `Simulink` model can be extended with `RTI` ADC and DAC blocks (which can be found in the `Simulink` library, or by typing in `rti` at the `MATLAB` prompt). The extension of the LFC two-area model with these blocks is visualised in Figure 5.6 (without AGC, as this will be performed by other hardware), where the AO and AI are configured according to Table 5.2. In between the ADC and DAC blocks, the two LFC areas are modelled together with their interconnecting tie-line dynamic. The `dSPACE` blocks in `Simulink` transform ± 10V signals into ± 1.0 values. For the DAC outputs $\Delta f_i[k]$ and $\Delta P_{tie_1}[k]$ this is acceptable, since all measurements stay within the ± 1.0 range with high probability, and hence also within the ± 10V range of the EL3004. However, due to choice of the EL4004 with a signal range between 0V and $+10$V, translating the `Simulink` model to the `dSPACE` simulator requires some rescaling, as these should also be able to take on negative values. Hence, upon receiving the analogue signal, the `dSPACE` model performs the rescaling

$$\Delta P_{c_i,dSPACE}[k] = \begin{cases} 0 & \text{if } \Delta P_{c_i,analogue}[k] = 0 \\ \Delta P_{c_i,analogue}[k] * 2.0 - 1.0 & \text{if } \Delta P_{c_i,analogue}[k] \neq 0 \end{cases} \tag{5.1}$$

where $\Delta P_{c_i,dSPACE}[k]$ is the rescaled signal and $\Delta P_{c_i,analogue}[k]$ the analogue signal inputted in the ZIF connector. After this rescaling, all control inputs stay within the ± 1.0 range with high probability.

After proper construction of the model, it can be 'built' to a *rti1401.tlc* target file using *Simulink Coder* (with the installation of RTI1401, *Simulink Coder* is extended with this option). This requires a right set of configuration parameters, which can be set under *Settings* in the *Simulink Coder* environment and are summarised in Table 5.4. The parameters omitted in this table can be left in their default setting.

Tab	Parameter	Value	Elucidation
Solver	Simulation time	0.0 to inf	Simulation starts with same initial values and runs continuously without a self-governing stop.
Solver	Solver selection	Fixed-step	`RTI` only supports fixed-step solvers, so here a sample time of 0.01s is selected with the default `Simulink` ode3 solver.
Simulation Target	Block reduction	Unchecked	Block reduction is not supported by `RTI`.
Code Generation	System target file	*rti1401.tlc*	The target file which is supported by `ControlDesk`.
`RTI` Simulation Options	Initial simulation state	RUN	Run model as soon as it is loaded.
`RTI` Load Options	Load application after build	Unchecked	More easily configured in `ControlDesk` manually.

Table 5.4: Non-default *Simulink Coder* configuration parameters

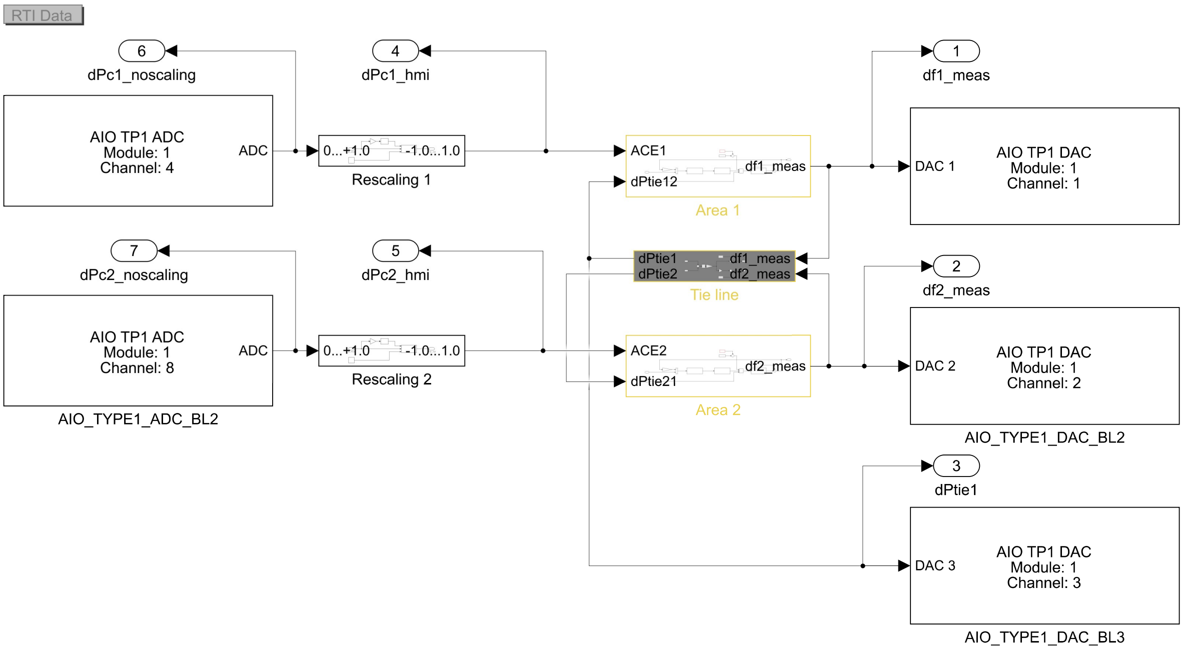


Figure 5.6: Simulink load frequency control two-area model with dSPACE blocks

With the proper configuration parameters, the target file is accompanied by an Structured Data File (SDF). This file can be deployed in ControlDesk. To do so, start a new project and experiment in ControlDesk. Select the dSPACE MAB II as target and the newly created *rti1401.tlc* target file as model. The additional outputs from Figure 5.6 have enabled the system operator to track the values in the layout of the ControlDesk interface, though these values can also be tracked in TwinCAT.

5.4.3. Programmable Logic Controller Software: TwinCAT 3

First, the TwinCAT 3 XAE software should be installed on the management PC. It is recommended to install the TwinCAT XAE Shell, which is the new default environment, replacing the Microsoft Visual Studio versions. Once TwinCAT is installed and launched, there is the option in *Real-Time* settings to isolate one of the eight PC cores, thereby dedicating it to run TwinCAT software. This cancels out the 'Base Time', i.e. the time required to jump from Windows to TwinCAT. Isolating a core is recommended, as it is one of the key advantages of using Beckhoff and TwinCAT. Due to the real-time requirement, is has to be made sure that the programme can run within the selected cycle time (in this case set to the default $0.01s$). This can be verified in the *Online* tab. The subsequent configuration steps are to add the PLC as target systems on the management PC, to establish communication as designed in Figure 5.1, to deploy the MATLAB & Simulink AGC and IDS functionality on the PLCs, and to add a form of HMI to keep track of the relevant parameters.

Add PLCs as target systems: The HILDA testbed is designed such that the PLCs can be configured by the management PC. To enable this, the PLCs should be added as target systems in TwinCAT. For this, the real-time drivers have to be installed on the PLCs. This is done by login in on a RDC, running \C:\TwinCAT\3.1\System\TcRteInstall.exe, and installing all adaptors (though this will cause the connection between the management PC and the PLC to be lost temporarily). If the LAN configuration is set according to subsection 5.4.1, the PLCs should be visible in the TwinCAT software when broadcast searching for new target systems. The PLCs can then be 'added' to the TwinCAT 'route', which means that the management PC now has access to the XAR environment of the PLCs. The licences can then be activated as documented on [65] under *TwinCAT 3* and then *Licensing*. Successful activation can be checked under *Licences* in TwinCAT.

Establish communication: As has already been made clear in subsection 5.3.3, two types of communication need to be established. For the communication between multiple TwinCAT devices (in this case the two IPCs), it is important to note that there are two approaches: using EAP or using Automatic

Device Specification (ADS). As the system is better compatible with the former (since EtherCAT is the standard protocol of Beckhoff), EAP is used here. To establish an EAP connection, the reader is referred to the third party manual in Appendix D. Keep in mind that the correct Ethernet Adaptor has to be selected after adding a new EAP item under *Devices*. The LAN port numbers are reversed with respect to their adaptor numbers (e.g. for this setup, LAN1 is connected to the switch, which means 'Ethernet 4' is to be selected for proper connection, which can be checked through the MAC address). For the analogue communication, a link (or a 'route') has to be created between the PLC 2 and the I/O modules. For this, too, the reader is referred to the tutorials in Appendix D. Similarly to adding PLCs as target systems, the correct Ethernet adaptor has to be selected and properly installed. Keep in mind that all TU Delft administrative rights should be in order as well.

Deploy AGC and IDS functionality on PLCs: As discussed in subsection 5.2.3, the PLCs are programmed using the object-oriented IEC 61131-3 standard. More specifically, Structured Text (ST) is used in this study. To translate the MATLAB & Simulink model to ST, *Simulink PLC Coder* can be used. This works quite similar to *Simulink Coder* as discussed in subsection 5.4.2. To use this, a specific Simulink model functionality (e.g. the AGC) has to be put in a single Simulink subsystem, which has to be transformed to an *Atomic Subsystem*. Then, a ST code can be exported, which again requires a right set of configuration parameters, this time summarised in Table 5.5. One important note is that the entire Simulink model should be operational, not just the affiliated subsection.

Tab	Parameter	Value	Elucidation
Solver	Solver selection	Fixed-step	RTI only supports fixed-step solvers, though here the solver and step size can be left by default (i.e. "auto").
Code Generation	System target file	*grt.tlc*	The target file which is supported by Microsoft Visual Studio.
PLC Code Generation	Target IDE	Beckhoff TwinCAT 3	Dedicated TwinCAT 3 IDE.

Table 5.5: Non-default *PLC Coder* configuration parameters

The resulting ST code requires some modifications before it is implementable in TwinCAT, for which some ST skills are recommended. In case the reader were to use ST and does not have (enough) experience with it, he/she is referred to Appendix D for either a brief or an extended tutorial. Eventually, the PLCs are programmed as in Appendix E. The communication between PLCs and within PLC programmes themselves is configured to be of type *REAL*, while the analogue communication is configured to be of type *INT*. When the input and output signals are configured as such in ST, they can be linked to the I/O device variables after activating the TwinCAT configuration.

Once configured correctly, the HILDA testbed consists of three time zones: one for the dSPACE simulator, and one for each PLC. Each of these devices has its own cycle time of $0.01s$ to run its programme. These cycle times are not necessarily aligned. The real-time process is visualised in Figure 5.7. The process is initiated with a cycle from the simulator, resulting in the analogue $y_p[k]$ measurement signals. A new cycle is initiated on PLC 2, in which the analogue measurement signals are red, watermarked and published. This happens up to one cycle time after the cycle of the dSPACE has finished. In the third cycle time of the process, the published measurements are red by a subscriber at PLC 1 and are subsequently equalised, after which the AGC and IDS mechanisms perform their function, generating control inputs u_c. In the same cycle, these control inputs are watermarked and published. In the next cycle, at PLC 2, the watermarked control inputs are equalised and sent through as analogue signals to the simulator, completing the real-time process. Because the watermarking functionality now runs on two different (real-time) cores, special care has to be taken to align the switching times $\theta[k]$ of the watermarkers and their respective equalisers. Even a minor discrepancy in these switching times can cause a large residual spike due to a mismatch in watermarking parameters. To prevent this, the switching parameters of $\mathcal{W}$ and $\mathcal{G}$ are also published so that $\mathcal{Q}$ and $\mathcal{H}$ can subscribe to them. This way, the watermarking-equalising pairs $\mathcal{W}, \mathcal{Q}$ and $\mathcal{G}, \mathcal{H}$ remain aligned. Note that the watermarking functionality of $\mathcal{W}$ and the equalising functionality of $\mathcal{H}$, both in PLC 2, are executed in a different cycle. This means that the switching times of the watermarking-equalising pair $\mathcal{W}, \mathcal{Q}$ should not be synchronous to that of $\mathcal{G}, \mathcal{H}$, i.e. $\mathcal{W}(\theta_s) \neq \mathcal{G}(\theta_s)$ and $\mathcal{Q}(\theta_s) \neq \mathcal{H}(\theta_s)$. Simultaneous switching would cause $\mathcal{H}$ to equalise a u_g originating from $\mathcal{G}$ which has not yet switched to the new watermarking parameters.

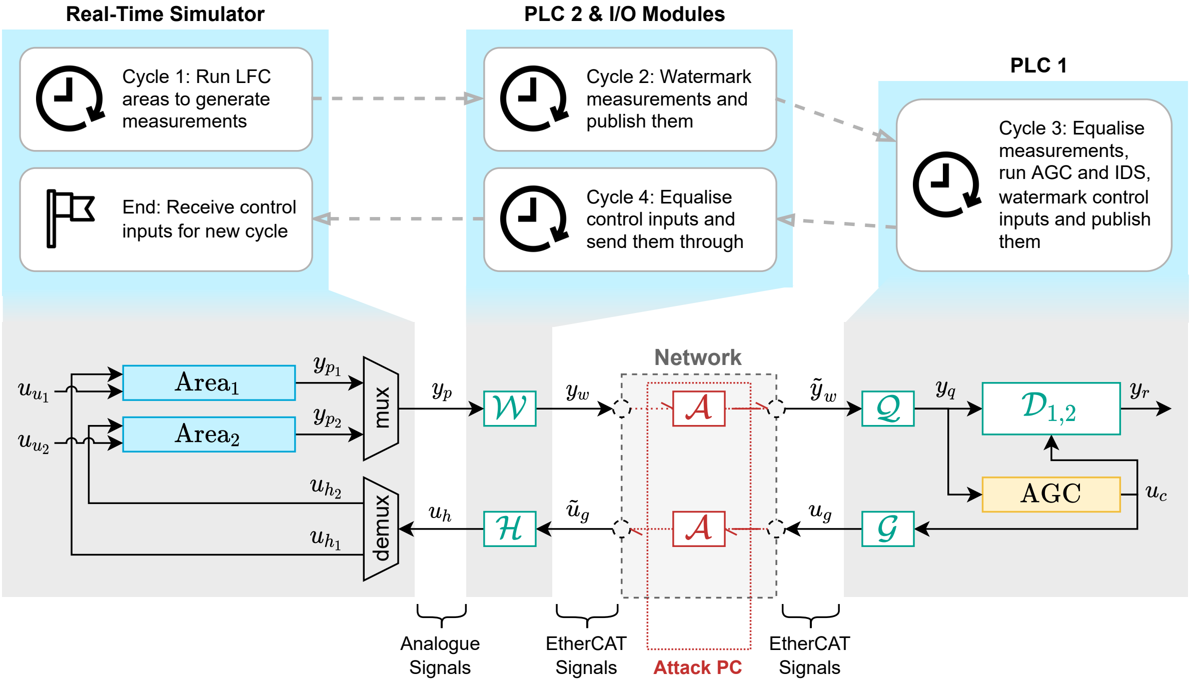


Figure 5.7: Real-time process of the testbed programmes, compared to the simplified load frequency control model

Remark. *Beckhoff offers dedicated licences to translate MATLAB & Simulink models to TwinCAT programmes (namely TE1401 and TE1400 for MATLAB and Simulink respectively). However, when acquiring the Beckhoff equipment, these were deemed superfluous.*

HMI configuration: To visualise the value progression of the relevant variables (i.e. the control inputs, the measurements and their estimates and residuals), this study uses a TwinCAT visualisation (*VISU*) page. Here, live values and value progressions can be visualised. Also, TwinCAT TE13xx Scope View [69] is used to view and export measurements to csv-files so that they can be compared to the results of the desktop simulations. Another option would be TwinCAT TF1800 PLC HMI [70]. A licence has already been acquired for this option when ordering the Beckhoff equipment.

5.4.4. Attack PC Operating System: Oracle VM VirtualBox

To be able to run a VM on the Windows OS, the Intel Virtualisation Technology has to be enabled. This can be done in the Basic Input/Output System (BIOS) settings, which is the very first piece of software run by computers when they are powered on, establishing communication with the operating system and attached devices (hard disk, USB etc.). After this, the VirtualBox installation guide [43] can be followed. Once installed, a few non-default parameters have to be set according to Table 5.6, which are based on tutorials from Appendix D and experience of the author himself. VirtualBox can then be used to realise a second OS, which can on its turn be used to execute a data integrity attack. The remainder of this section discusses the option of deploying Kali Linux on the VM.

Kali Linux on VM: Kali Linux is an operating system designed specifically for cyber-security applications. It contains preinstalled tools such as *Ettercap*, *Hping3* and *Wireshark*. These tools can be used to implement cyber-attacks such as MITM and DDoS. To install Kali Linux, the reader is referred to the installation guide [183]. Once installed, it can be deployed on the VM as in [182].

To conduct MITM attack, the Kali Linux *Ettercap* [154] tool can be used. This tool is equipped with instruments for reconnaissance/eavesdropping attacks, as well as data integrity attacks. To conduct these attacks, first ARP spoofing is executed, thereby fooling the sending devices to send the communication to the MAC address of the attacker and placing the attack PC 'in the middle'. The default setting of Ettercap is to send the communication through to its original destination in real-time, without any packet alteration but with sole sniffing. Now, any packet analyser (in this study this is *Wireshark*)

Setting	Tab	Parameter	Value	Elucidation
System	Motherboard	Base Memory	3500 MB	More than 2048 MB of basic memory, otherwise startup issues might occur.
System	Motherboard	Boot Order	Hard Disk, then Optical	Resulted in the best booting experience.
System	Processor	Processor(s)	3 Cores	Minimally 3 (but preferably 4) cores for proper functionality, otherwise startup issues might occur.
Display	Screen	Video Memory	128 MB	Resulted in the most smooth VM operation.
Network	Adaptor 1	Attacked to	Bridged Adaptor	This way, the host network adaptor enables the VM to connect to the LAN that the host system uses. This is also where the MAC address can be configured manually.

Table 5.6: Non-default `VirtualBox` configuration parameters

should be able to read the data which is being send between the devices. Ettercap works in a duplex fashion, implying that it affects the communication in both ways.

Ettercap also offers instruments for packet alteration, so called ’filters’. Through these filters, specific bits of all messages (the data fields) can be altered such that the content of the message changes before sending the message through to its original source. Attacks can be made stealthy by not only applying an *Ettercap* filter to the communication from the plant to the controller, but also from the controller to the HMI. This way, a healthy plant operation can be mimicked, while the controller is forced to operate with falsified measurements. By multiplying the appropriate bits, a scaling attack could be conducted. It is also possible to stop the flow of messages entirely. By stopping the flow and inserting a recorded data flow, a replay attack can also be conducted through *Ettercap*. Flows of data can be recorded using most packet analysers, including *Wireshark*, which stores the communication in *pcap*-files. Another option would be to use the `Kali Linux` tool *tcpreplay* [184], which allows for classification of traffic as client or server, rewrite OSI Layer 2, 3 and 4 packets, and finally replay the traffic back onto the network and through other devices, including switches.

5.5. Conclusion

This chapter provided a comprehensive overview of the HILDA testbed design, construction and configuration. After discussing the motivation and constraints behind the testbed, the required appliances were derived. These appliances were elucidated, after which their physical schematics were discussed. With the physical schematics in place, the configuration of the testbed was discussed so that the mathematical models from chapter 3 and chapter 4 could be deployed on the testbed. This concludes the modelling and design of the LFC model, the data integrity attack, the IDS mechanism and the HILDA testbed. Next, everything constructed so far is verified and validated.

6 Results

This chapter provides the experimental verification and validation results. The distinction between verification and validation is made according to the IEEE-STD-610 standard [41]: verification answers the question whether the product is built right, while validation answers the question whether the right product is built. The verification and validation steps of this study are provided in Table 6.1. The idea is that if the functionality of both the mathematical model and the HILDA testbed are promptly verified, legitimate results will follow in the validation experiments. First, this chapter discusses the verification results, followed by those of the validation.

			Directly Related Subtopics			
Section	Result Type	Description	AGC	Attack	IDS	HILDA
section 6.1	Verification	Model testing in `MATLAB & Simulink`	x	x	x	
section 6.2	Verification	HILDA testbed design and configuration testing				x
section 6.3	Validation	Validation of DMWM performance on HILDA testbed	x	x	x	x

Table 6.1: Thesis verification and validation process

6.1. Model and Design Testing in `MATLAB & Simulink`

Before deploying the model onto the testbed, it is verified in `MATLAB & Simulink` simulations. The code can be found in Appendix E. Similar to the sequence of chapter 3 and chapter 4, this section first discusses the LFC and AGC functionalities, followed by that of the data integrity attack. This section is closed off by discussing the DMWM verification.

6.1.1. Load Frequency Control and Automatic Generation Control Verification

To simulate the model, the noise and load profiles subdued on the model need to be defined. The LFC area process noise η_i is modelled as independent and identically distributed zero-mean Gaussian noise with covariance matrix Cov_{η_i}, while the measurement noise ξ_i is modelled as zero-mean Gaussian noise with covariance matrix Cov_{ξ_i}. The respective covariance matrices are configured as

$$Cov_{\eta_i} = diag\begin{bmatrix}2.5E^{-3} & \mathcal{O}_{n_x-1}\end{bmatrix} \tag{6.1}$$

$$Cov_{\xi_i} = 8E^{-5}I_{n_y}. \tag{6.2}$$

The variance $2.5E^{-3}$ causes the frequency to fluctuate within the normal range of ± 0.03 Hz with high probability [90, 165], while the measurement noise magnitude $8E^{-5}$ causes the frequency measurement to fall within $\pm 5E^{-4}$ Hz, also with high probability [90, 219]. Cov_{η_i} has direct effect on the load changes $\Delta P_{L_i}(t)$. These are modelled to be constants throughout the simulation times, $\Delta P_{L_1}[k] = 0.08$ and $\Delta P_{L_2}[k] = 0.12$. The load profiles together with the respective noises are visualised in Figure 6.1.

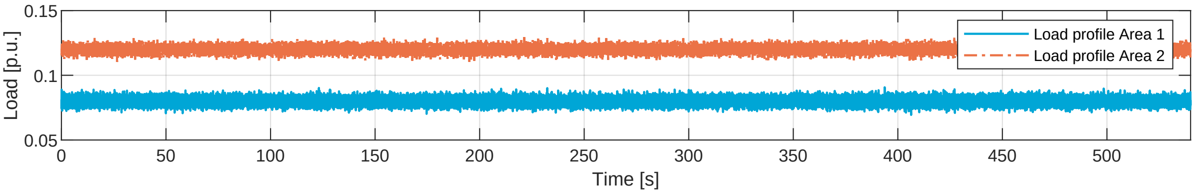
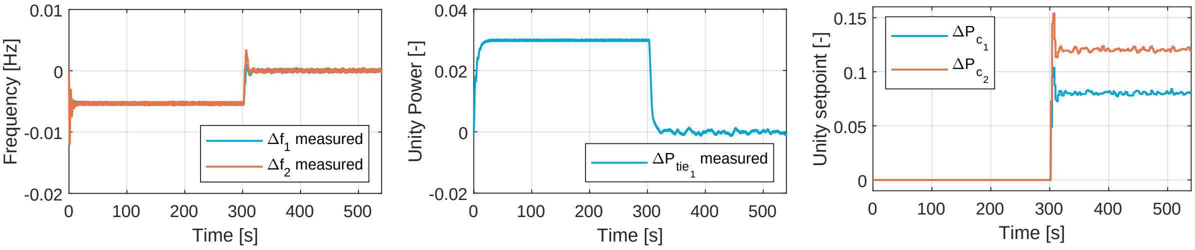


Figure 6.1: Load profiles of both load frequency control areas

The effect of the AGC mechanism on the LFC area frequencies and tie-line power flow is depicted in Figure 6.2. At $t < 300s$, no AGC algorithm is active. Due to the load change, the measurements stabilise around an offset. When AGC is activated at $t = 300s$, this offset is resolved and the stable operating point becomes $\lim_{k\to\infty} \Delta f_i[k], \Delta P_{tie_1}[k] = 0$.



(a) Impact on $\Delta f_i[k]$ measurements **(b)** Impact on $\Delta P_{tie_1}[k]$ measurement **(c)** Impact on control inputs $\Delta P_{c_i}[k]$

Figure 6.2: Impact of the automatic generation control mechanism on the load frequency control input and output values

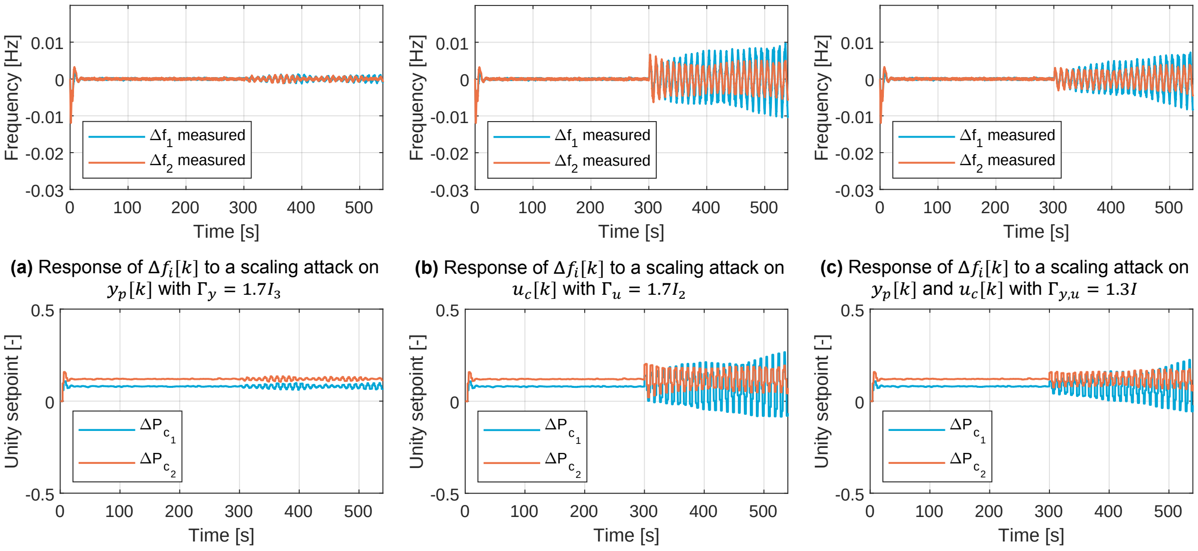
6.1.2. Data Integrity Attack Verification

Now that the proper functionality of the combined LFC and AGC mechanism have been verified, the impact of the scaling and replay data integrity attacks can be analysed. This is done for six attack scenarios in total, as clarified in Table 6.2. Recall that $y_p[k]$ are the plant outputs travelling over the network from plant to controller, while $u_c[k]$ are the control inputs also travelling over the network but from controller to plant. Also recall the mathematical formulation of the scaling attack from Equation 3.29. For attack scenario 1 and 2, the scaling attack is appointed a scaling of $\Gamma_y = 1.7I_3$ and $\Gamma_u = 1.7I_2$ for the measurements and the control inputs, respectively. For attack scenario 3, this is $\Gamma_y = 1.3I_3$ and $\Gamma_u = 1.3I_2$. In all three scaling attack scenarios, the attack is initiated at $t_0 = 300s$. The impact of the attack on the $\Delta f_i[k]$ measurements and $\Delta P_{c_i}[k]$ control inputs is visualised in Figure 6.3. The scaling attack appears to destabilise both LFC areas in scenarios 2 and 3, but not in scenario 1.

For the replay attack, recall the mathematical formulation of Equation 3.30. In scenario 4 from Table 6.2, only the measurement signals are attacked, implying that the control inputs are not maliciously altered (i.e. $\tilde{u}_c[k] = u_c[k]$). For scenario 5, this is the other way around, i.e. the control inputs are replayed while the measurement signals are not maliciously altered (i.e. $\tilde{y}_p[k] = y_p[k]$). Finally, for scenario 6, both the signals are replayed. For all replay attack scenarios, the recording period is set to be $T = 50s$, starting as $t_r = 250s$. At $t_0 = 300s$, the recorded data is replayed, replacing the original communication. The impact on the frequency measurements and the control inputs is plotted in Figure 6.4. Indeed, starting $t_0 = 300s$, the same pattern repeats every $T = 50s$. The destabilising effect appears to be less for replay attacks than for scaling attacks.

		Targeted signals		
Figure	Data integrity attack	$y_p[k]$	$u_c[k]$	$y_p[k]$ and $u_c[k]$
Figure 6.3	Scaling attack	Scenario 1	Scenario 2	Scenario 3
Figure 6.4	Replay attack	Scenario 4	Scenario 5	Scenario 6

Table 6.2: Six simulation attack scenarios



(a) Response of $\Delta f_i[k]$ to a scaling attack on $y_p[k]$ with $\Gamma_y = 1.7I_3$

(b) Response of $\Delta f_i[k]$ to a scaling attack on $u_c[k]$ with $\Gamma_u = 1.7I_2$

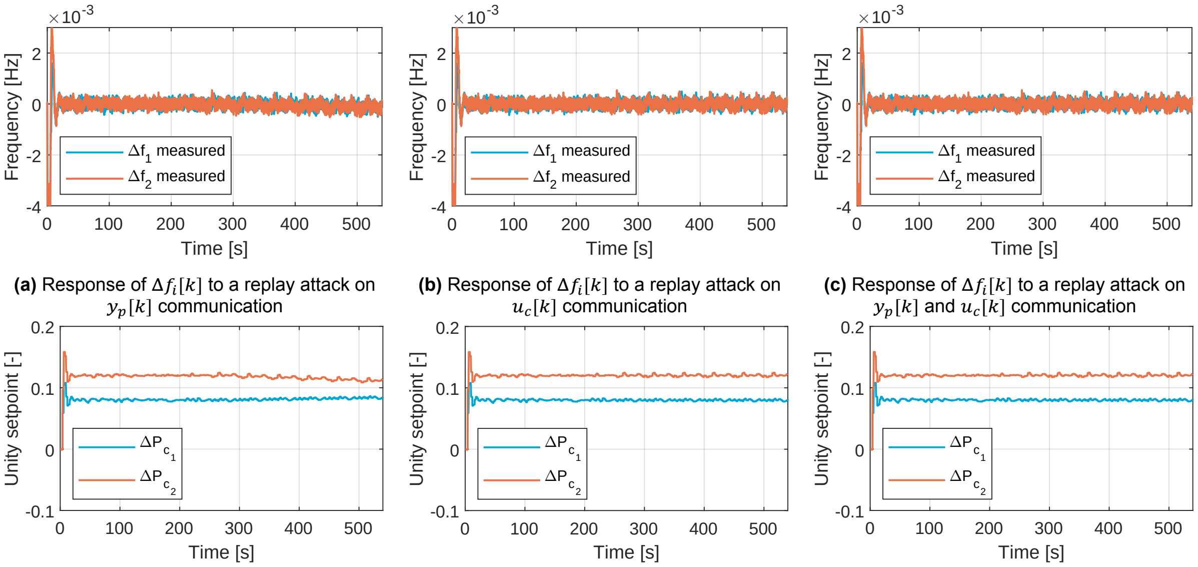
(c) Response of $\Delta f_i[k]$ to a scaling attack on $y_p[k]$ and $u_c[k]$ with $\Gamma_{y,u} = 1.3I$

(d) Response of $\Delta P_{c_i}[k]$ to a scaling attack on $y_p[k]$ with $\Gamma_y = 1.7I_3$

(e) Response of $\Delta P_{c_i}[k]$ to a scaling attack on $u_c[k]$ with $\Gamma_u = 1.7I_2$

(f) Response of $\Delta P_{c_i}[k]$ to a scaling attack on $y_p[k]$ and $u_c[k]$ with $\Gamma_{y,u} = 1.3I$

Figure 6.3: System performance impact of a scaling attack with $t_0 = 300s$



(a) Response of $\Delta f_i[k]$ to a replay attack on $y_p[k]$ communication

(b) Response of $\Delta f_i[k]$ to a replay attack on $u_c[k]$ communication

(c) Response of $\Delta f_i[k]$ to a replay attack on $y_p[k]$ and $u_c[k]$ communication

(d) Response of $\Delta P_{c_i}[k]$ to a replay attack on $y_p[k]$ communication

(e) Response of $\Delta P_{c_i}[k]$ to a replay attack on $u_c[k]$ communication

(f) Response of $\Delta P_{c_i}[k]$ to a replay attack on $y_p[k]$ and $u_c[k]$ communication

Figure 6.4: System performance impact of a replay attack with $T = 50s$, $t_r = 250s$ and $t_0 = 300s$

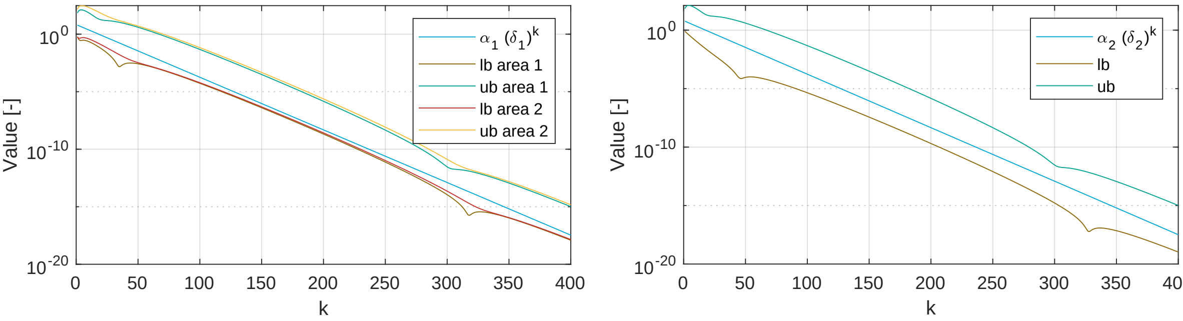
6.1.3. Watermarking and Luenberger Observer Verification

The Luenberger observer is implemented in Simulink through a discrete-time Luenberger observer block, producing $\hat{x}_{p_i}[k]$, which on its turn is used to produce $\hat{y}_{p_i}$ and y_{r_i} per LFC area. The discrete-time observer poles of A_{r_i} are selected to be placed at $\begin{bmatrix} 0.7 & 0.8 & 0.9+0.01j & 0.9-0.01j \end{bmatrix}$ by the Luenberger gains L_i. With this pole placement, it is verified that matrices A_{r_1} and A_{r_2} are both Schur.

For the residual threshold $\bar{y}_r$, α and δ need to be designed according to Equation 4.11. They are configured to be the same for both LFC areas, so

$$\alpha_{1,n} = \alpha_{2,n}, \quad \delta_{1,n} = \delta_{2,n}. \tag{6.3}$$

By selecting $\alpha_{i,1} = 6.8$, $\alpha_{i,2} = 6.5$ and $\delta_{i,1}, \delta_{i,2} = 0.9$, the lower bound (lb) $\|C_{d,p_i,(n)}\,(A_r)^k\|$, upper bound (ub) $\|C_{d,p_i,(n)}\| \cdot \|\,(A_r)^k\,\|$ and $\alpha_n\,(\delta_n)^k$ of Figure 6.5 are obtained. Because of the relationship in Equation 6.3, also $\alpha_1\,(\delta_1)^k$ is the same for both LFC areas, though their bounds are not because of a different C_{d,p_i} matrix. Altering α changes the initial value of $\alpha_n\,(\delta_n)^k$, while altering δ changes its exponential slope. From Figure 6.5, it can be verified that indeed the relationship of Equation 4.11 holds. Only the values up to $k = 400$ are visualised, but due to an identical slope of $\alpha_n\,(\delta_n)^k$ and the bounds, the relationship of Equation 4.11 holds for all subsequential time steps.



(a) Configuration of $\alpha_{i,1} = 6.8$ and $\delta_{i,1} = 0.9$ to obtain the residual thresholds of $\Delta f_1[k]$ and $\Delta f_1[k]$

(b) Configuration of $\alpha_{i,2} = 6.5$ and $\delta_{i,2} = 0.9$ to obtain the residual threshold of $\Delta P_{tie_1}[k]$

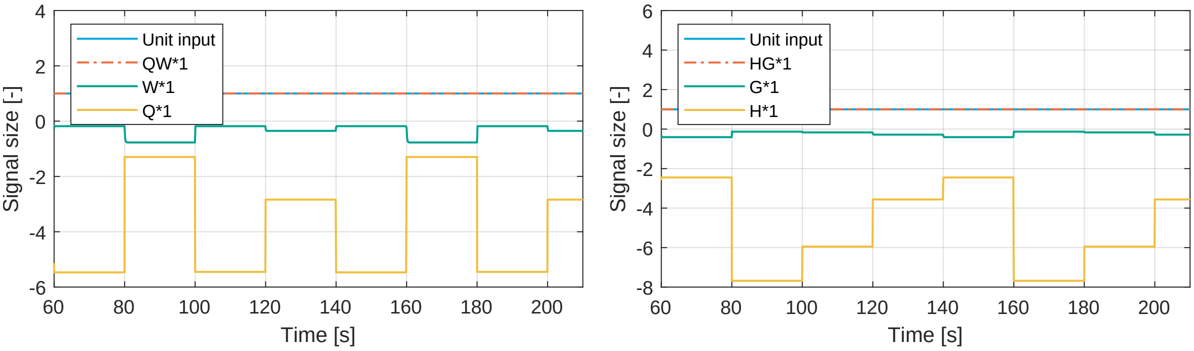
Figure 6.5: Configuration of α and δ to obtain the residual thresholds of the measurements

The watermarkers $\mathcal{W}$ and $\mathcal{Q}$ and their equalisers $\mathcal{G}$ and $\mathcal{H}$ are designed to have a limited set of four predetermined states, i.e. $\Theta := \{\theta_1, \theta_2, \theta_3, \theta_4\}$. A limited set of predetermined states is selected so that the results of the `MATLAB & Simulink` simulations can be compared to those of the testbed simulations. The set of watermarking filter parameters w_A and w_B are constructed *a priori* in `MATLAB & Simulink` and exported to the PLCs. They are designed to be of third order, i.e. $M = 3$. All watermark parameter combinations are constructed as

$$w_A = \begin{bmatrix}-0.1 & 0 & 0.1\end{bmatrix} + 0.35 * rand^{1x3} \tag{6.4}$$

$$w_B = \begin{bmatrix}-0.1 & 0 & 0 & 0\end{bmatrix}. \tag{6.5}$$

where $rand^{1\times3} \in \mathbb{R}^{1\times3}$ represents a row vector of uniformly distributed random numbers in the range of $[-1, 1]$. These random numbers are kept the same for every simulation through the use of random seeds. The states are designed to switch every $20s$, i.e. $\mathcal{K}_\theta := \{20, 40, 60, 80, ...\}$. To illustrate the functionality of the watermarkers, the filter output responses to a constant unit signal input are plotted in Figure 6.6. Both the individual and the combined responses of the watermarking and equalising filter are displayed for a limited time sequence. The resulting pattern due to the switching of the watermark parameters every $20s$ is clearly visible in Figure 6.6.



(a) Output responses of filters $\mathcal{W}$ and $\mathcal{Q}$

(b) Output responses of filters $\mathcal{G}$ and $\mathcal{H}$

Figure 6.6: Watermarking filter output responses to a constant unit signal input

Figure 6.7 provides a zoomed-in plot of the same signals as Figure 6.6a. The emphasis is put on the output response of the filters around a parameter switch. Instead of an immediate step response, the individual outputs of $\mathcal{W}$ and $\mathcal{Q}$ show a transient response. Still, because the watermarker and the equaliser are each others inverse, no decrease in system performance can be detected, as the output $\mathcal{Q}\mathcal{W} * 1$ is the exact same as the unit input.

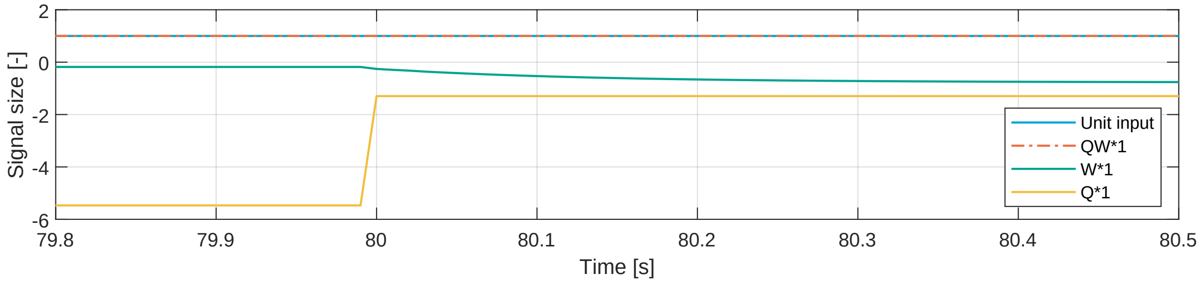


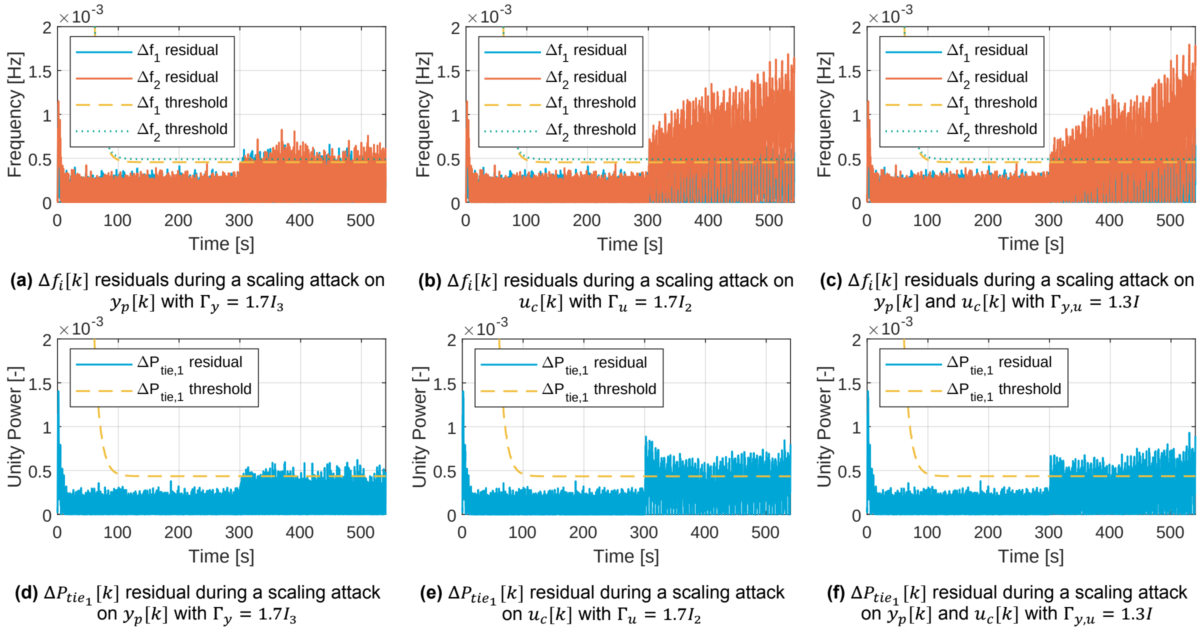
Figure 6.7: Zoomed in transient watermarking filter output responses to a constant unit signal input

To analyse the IDS functionality, the same attack scenarios from Table 6.2 are considered. Now, instead of the responses of $y_p[k]$ and $u_c[k]$, the residuals $y_r[k]$ and their thresholds $\bar{y}_r[k]$ are examined. For both the scaling and the replay attack, the residuals without and with DMWM are evaluated. This leads to a total of twelve simulation scenarios. These scenarios are depicted in Figure 6.8, Figure 6.9, Figure 6.10 and Figure 6.11. It should be noted that, due to the use of random seeds, each distinct scenario produces the same disturbances for all MATLAB & Simulink simulation runs.

The detection performance is summarised in Table 6.3, including the Detection Ratio (DR) metric. This metric indicates to what extend $y_r[k]$ surpasses $\bar{y}_r[k]$ after an attack has occured through

$$DR = \frac{\sum_{k=k_0}^{k_0+\kappa_a} \mathbb{1}_{y_r[k] \geq \bar{y}_r[k]}}{\kappa_a} \in [0, 1] \tag{6.6}$$

where $\mathbb{1}$ is the indicator function which is equal to 1 if the attack is detected, and 0 otherwise [173]. Consequently, $DR = 1$ if the attack is always detected for $k \in \kappa_a$, while $DR = 0$ if it is never detected.



(a) $\Delta f_i[k]$ residuals during a scaling attack on $y_p[k]$ with $\Gamma_y = 1.7I_3$

(b) $\Delta f_i[k]$ residuals during a scaling attack on $u_c[k]$ with $\Gamma_u = 1.7I_2$

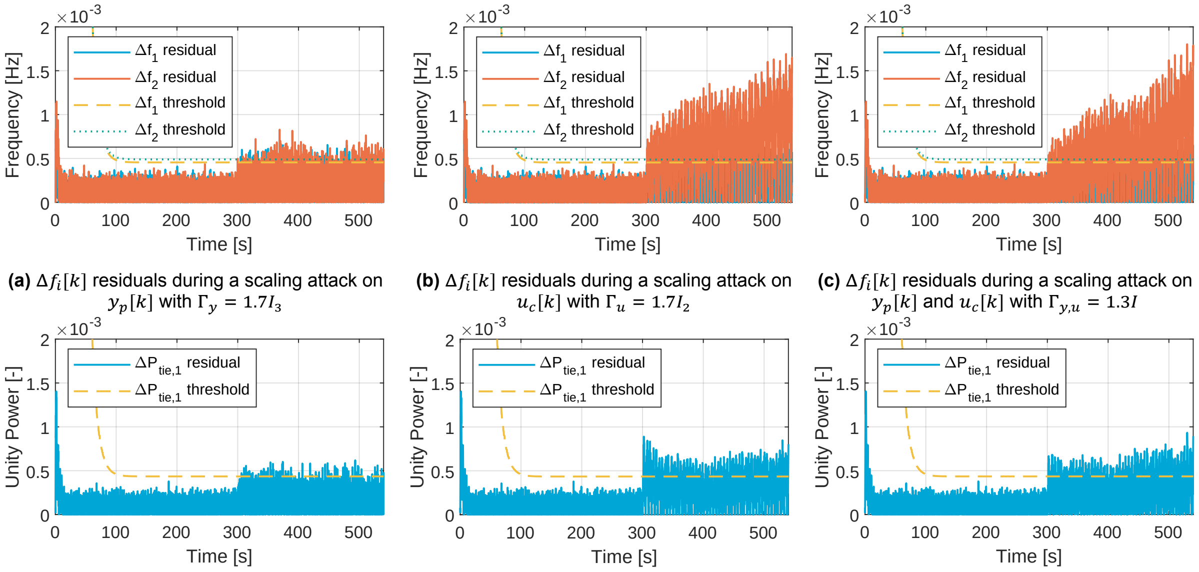
(c) $\Delta f_i[k]$ residuals during a scaling attack on $y_p[k]$ and $u_c[k]$ with $\Gamma_{y,u} = 1.3I$

(d) $\Delta P_{tie_1}[k]$ residual during a scaling attack on $y_p[k]$ with $\Gamma_y = 1.7I_3$

(e) $\Delta P_{tie_1}[k]$ residual during a scaling attack on $u_c[k]$ with $\Gamma_u = 1.7I_2$

(f) $\Delta P_{tie_1}[k]$ residual during a scaling attack on $y_p[k]$ and $u_c[k]$ with $\Gamma_{y,u} = 1.3I$

Figure 6.8: Residuals and thresholds during a scaling attack *without* watermarking and $t_0 = 300s$, simulated on the desktop



(a) $\Delta f_i[k]$ residuals during a scaling attack on $y_p[k]$ with $\Gamma_y = 1.7I_3$

(b) $\Delta f_i[k]$ residuals during a scaling attack on $u_c[k]$ with $\Gamma_u = 1.7I_2$

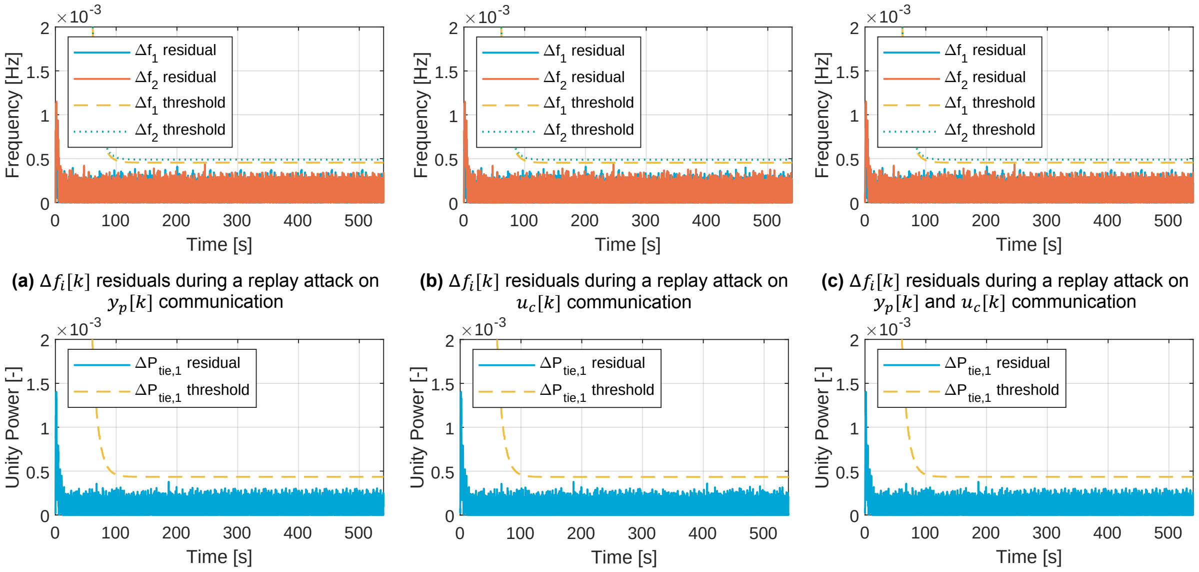
(c) $\Delta f_i[k]$ residuals during a scaling attack on $y_p[k]$ and $u_c[k]$ with $\Gamma_{y,u} = 1.3I$

(d) $\Delta P_{tie_1}[k]$ residual during a scaling attack on $y_p[k]$ with $\Gamma_y = 1.7I_3$

(e) $\Delta P_{tie_1}[k]$ residual during a scaling attack on $u_c[k]$ with $\Gamma_u = 1.7I_2$

(f) $\Delta P_{tie_1}[k]$ residual during a scaling attack on $y_p[k]$ and $u_c[k]$ with $\Gamma_{y,u} = 1.3I$

Figure 6.9: Residuals and thresholds during a scaling attack *with* watermarking and $t_0 = 300s$, simulated on the desktop



(a) $\Delta f_i[k]$ residuals during a replay attack on $y_p[k]$ communication

(b) $\Delta f_i[k]$ residuals during a replay attack on $u_c[k]$ communication

(c) $\Delta f_i[k]$ residuals during a replay attack on $y_p[k]$ and $u_c[k]$ communication

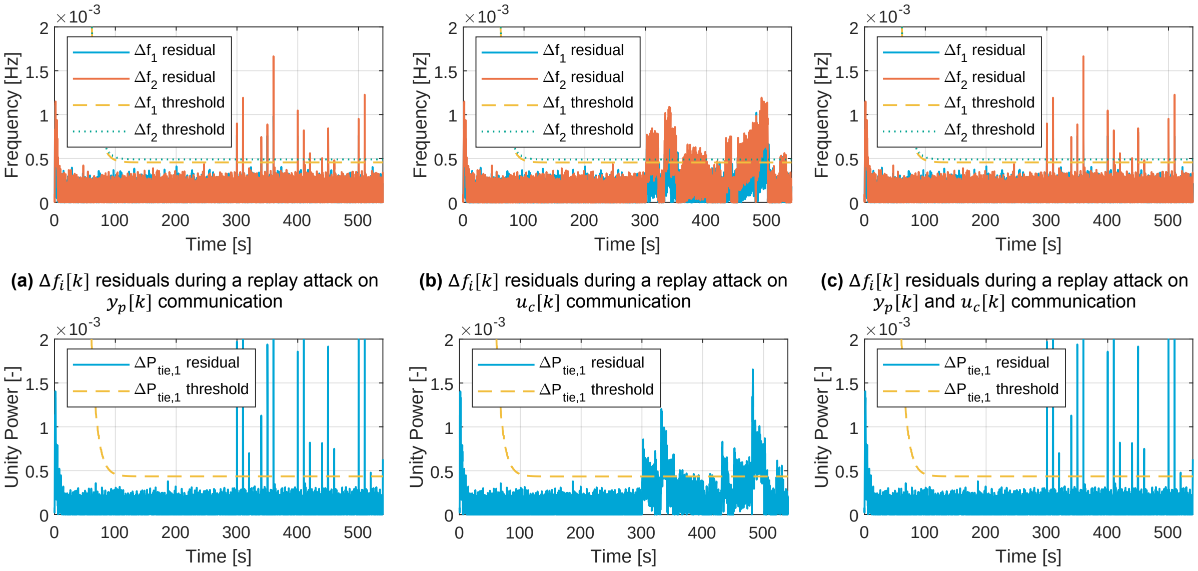
(d) $\Delta P_{tie_1}[k]$ residual during a replay attack on $y_p[k]$ communication

(e) $\Delta P_{tie_1}[k]$ residual during a replay attack on $u_c[k]$ communication

(f) $\Delta P_{tie_1}[k]$ residual during a replay attack on $y_p[k]$ and $u_c[k]$ communication

Figure 6.10: Residuals and thresholds during a replay attack *without* watermarking and $T = 50s$, $t_r = 250s$ and $t_0 = 300s$, simulated on the desktop

A few remarkable findings are distinguished. Firstly, for all scenarios, the False Alarm Rate (FAR) is equal to zero, i.e. no alarms are produced when $k < k_0$. Secondly, there is an almost similar detection performance of the scaling attacks with and without watermarking. Also notable is that only the scenario of the replay attack without watermarking does not detect any attack, i.e. $DR = 0$. The final remarkable finding is the improvement of detection performance when including watermarking in the context of a replay attack. At the very first timestep of the attack, k_0, the residuals of $\Delta f_1[k]$ and $\Delta P_{tie_1}[k]$ surpass their thresholds, followed promptly by $\Delta f_2[k]$.



(a) $\Delta f_i[k]$ residuals during a replay attack on $y_p[k]$ communication

(b) $\Delta f_i[k]$ residuals during a replay attack on $u_c[k]$ communication

(c) $\Delta f_i[k]$ residuals during a replay attack on $y_p[k]$ and $u_c[k]$ communication

(d) $\Delta P_{tie_1}[k]$ residual during a replay attack on $y_p[k]$ communication

(e) $\Delta P_{tie_1}[k]$ residual during a replay attack on $u_c[k]$ communication

(f) $\Delta P_{tie_1}[k]$ residual during a replay attack on $y_p[k]$ and $u_c[k]$ communication

Figure 6.11: Residuals and thresholds during a replay attack *with* watermarking and $T = 50s$, $t_r = 250s$ and $t_0 = 300s$, simulated on the desktop

			Detection Step k_d			Detection Ratio		
#	Cyber-Attack	Watermarking	$\Delta f_1[k]$	$\Delta f_2[k]$	$\Delta P_{tie_1}[k]$	$\Delta f_1[k]$	$\Delta f_2[k]$	$\Delta P_{tie_1}[k]$
1	Scaling of $y_p[k]$	Non-active	30378	30848	30612	0.00546	0.01279	0.00533
2	Scaling of $u_c[k]$	Non-active	30161	30127	30115	0.02917	0.62742	0.19583
3	Scaling of $y_p[k]$, $u_c[k]$	Non-active	30179	30197	30154	0.02588	0.50217	0.11267
4	Scaling of $y_p[k]$	Active	30474	30000	30612	0.00558	0.01367	0.00567
5	Scaling of $u_c[k]$	Active	30158	30127	30115	0.02938	0.62842	0.19758
6	Scaling of $y_p[k]$, $u_c[k]$	Active	30161	30197	30136	0.02642	0.50371	0.11533
7	Replay of $y_p[k]$	Non-active	-	-	-	0	0	0
8	Replay of $u_c[k]$	Non-active	-	-	-	0	0	0
9	Replay of $y_p[k]$, $u_c[k]$	Non-active	-	-	-	0	0	0
10	Replay of $y_p[k]$	Active	30000	30000	30000	0.00163	0.00392	0.00246
11	Replay of $u_c[k]$	Active	30123	30129	30106	0.07588	0.28879	0.20254
12	Replay of $y_p[k]$, $u_c[k]$	Active	30000	30000	30000	0.00163	0.00392	0.00246

Table 6.3: Detection performance for each attack scenario with or without watermarking, simulated on the desktop

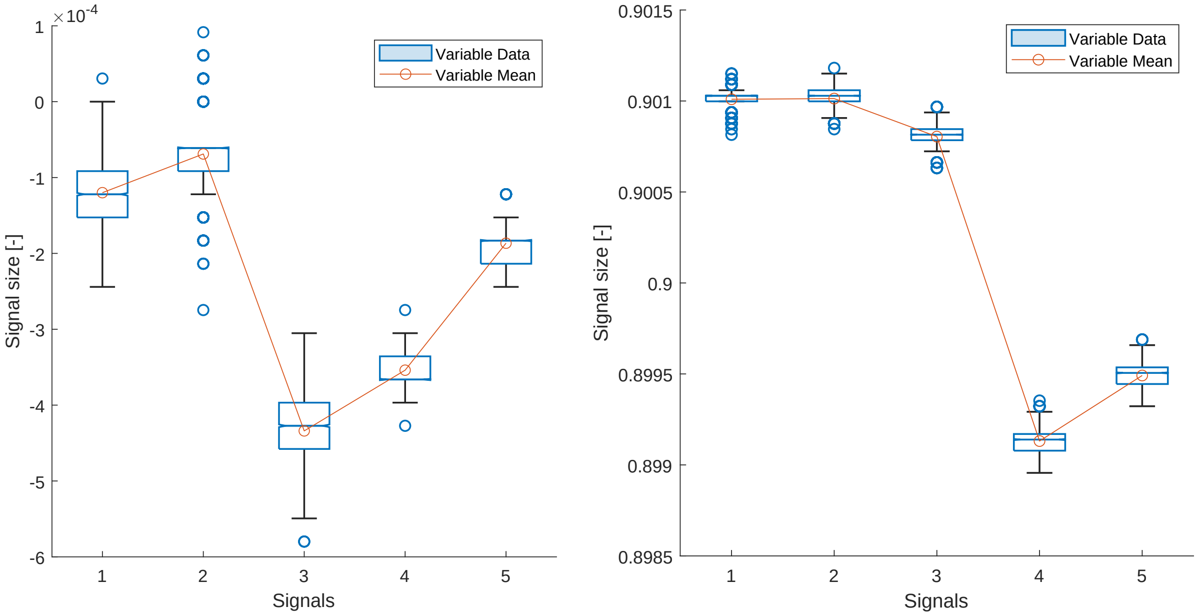
6.2. Design and Configuration Testing of the Testbed

Next to the verification of the model in MATLAB & Simulink, the functionality of the HILDA testbed is to be verified. If some component of the testbed is either designed, constructed or configured incorrectly, the results originating from testbed simulations would be faulty, i.e. not representative to real-world scenarios. Therefore, a systematic sequence of experiments were conducted to mitigate these potential faults. The entire experimentation sequence is provided in Appendix F. This section only discusses a selection of the findings from these experiments. This selection includes the findings regarding the physical wiring, the network configuration, and the cyber-attack configuration of the HILDA testbed.

6.2.1. Physical Wiring of the Testbed

Verifying the correct power wiring is done by analysing the various LEDs on the appliances, as was briefly discussed in section 5.2. These are the *DC OK* LED on the power supply, the *Power Status* LED on the MAB II, the *Us 24V* and *Up 24V* LEDs on the EK1100, and the *PWR* LED on the IPCs.

Verifying the correct analogue communication wiring is done by sending through a relatively small and large signal, and subsequently analysing the values at the input terminals of the EL3004 and ADC inputs of the dSPACE ZIF connector. As the original signals are known, the effect of the analogue



(a) Disturbance due to analogue signal wires when transmitting a 0 signal

(b) Disturbance due to analogue signal wires when transmitting a 0.9 signal

Figure 6.12: Disturbance due to analogue signal wires which transport $\Delta f_1[k]$, $\Delta f_2[k]$, $\Delta P_{tie_1}[k]$, $\Delta P_{c_1}[k]$ and $\Delta P_{c_2}[k]$, respectively, as measured at the input terminals when transmitting a 0 and a 0.9 signal

wiring on the signals can be derived. The considered signal magnitudes are 0 and 0.9 on a scale from -1 to 1 (a signal of 1 would be too large, since the input terminal cuts off the input signals larger than $10V$). This results in the noise profiles of Figure 6.12a and Figure 6.12b, respectively. The results for the measurements $\Delta f_i[k]$ and $\Delta P_{tie_1}[k]$ are extracted from the `TwinCAT` software through the *Measurement Export Wizard* of `TwinCAT` Scope View [69], while the results for the control inputs $\Delta P_{c_i}[k]$ are extracted from `ControlDesk` by using the built-in record option and saving them as *mat*-files.

In the box plots of Figure 6.12a and Figure 6.12b, a few observations are worth mentioning. First of all, when sending through a signal of 0, all measurements have an offset lower than zero. When sending through a signal of 0.9, a clear distinction can be identified between the measurement signals (1,2,3) and the control input signals (4,5). Whereas the control inputs seem to have a slightly bigger negative offset as a result of the voltage drop compared to the 0 signal case, the measurements have a positive offset, though the differences between the measurements themselves are approximately the same.

6.2.2. Network Configuration of the Testbed

The network configuration should be configured as in Table 5.3. This can be verified on either the management PC, one of the PLCs, or even the attack PC, as all should be on the same LAN. When executing the `arp -a` command in a command-line interface of the management PC (such as *Windows PowerShell*), indeed, all ”Interface Address[es]” (i.e. IP addresses) and the ”Physical Address[es]” (i.e. MAC addresses) match those of Table 5.3, as visualised by the interface listing below.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32> arp -a

Interface: 172.19.3.242 --- 0x5
  Internet Address      Physical Address      Type
  172.19.3.1            e8-1c-ba-35-f9-ef     dynamic
  172.19.3.44           64-4d-70-00-97-c3     dynamic
  172.19.3.239          00-01-05-4b-5b-d7     dynamic
  172.19.3.240          00-01-05-66-d5-fc     dynamic
  172.19.3.241          00-01-05-66-dc-1c     dynamic
  172.19.3.255          ff-ff-ff-ff-ff-ff     static
```

6.2.3. Cyber-Attack Configuration

The initial plan of this research was to use the MODBUS communication protocol, which requires ARP to determine the MAC address destination of the messages. As clarified in subsection 5.4.4, such a setup would have been hackable with the *Ettercap* tool from Kali Linux. However, the author was not able to implement the MODBUS protocol, or any form of protocol which required TCP for that matter. To make the LFC and AGC mechanisms operational on the HILDA testbed, the EtherCAT protocol was adopted, which does not require ARP. This is because the EtherCAT messages are encapsulated in Ethernet frames before being deployed on the TwinCAT XAR environment. In other words, the respective MAC addresses are already provided to the EtherCAT frames manually. Consequently, no ARP is required, and ARP spoofing does not work. Instead, the attacker would have to either change the configuration of the Ethernet switch so that the messages are rerouted via the attack PC, or intrude the PLC programming environment so that the attacker can change the programme configuration [74].

Configuration of PLC programmes without VM: As alternative, the data integrity attacks from section 3.3 are programmed on the PLCs directly, not via the attack PC. [74] has shown that, indeed, it is possible to execute a MITM attack on a Beckhoff system which uses the EtherCAT protocol, though they have not shared the code with which this was achieved. Nonetheless, the results from [74] show that it makes sense to simulate a MITM attack by manually programming the attack in the PLC programmes. The code to execute the attacks is provided in Appendix E.

6.3. Validation of Watermarking Performance on the Testbed

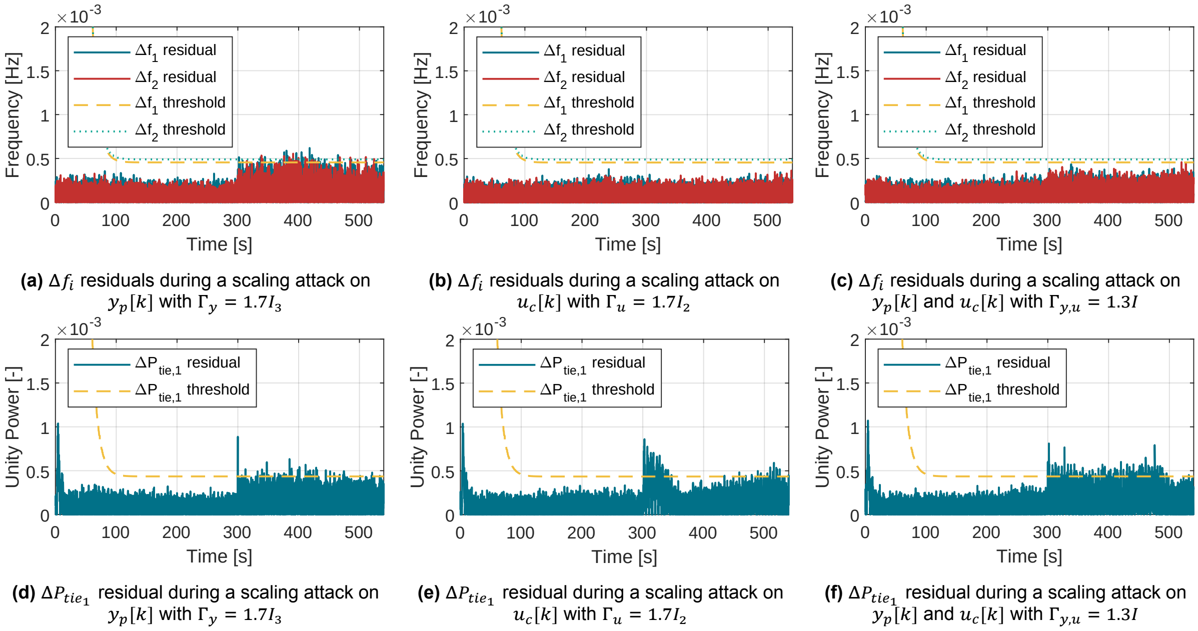
Now that proper functionality of the model, design and configuration of the LFC mechanism, the data integrity attacks, the detection mechanism and the testbed have been verified, it is possible to analyse the performance of the DMWM mechanism when applied on the HILDA testbed. Doing so, and comparing it to the case where there is no watermarking, should answer the thesis research question.

To extract the data from the HILDA testbed, again the *Measurement Export Wizard* of TwinCAT Scope View [69] is utilised. Apart from the disturbances due to the physical wiring and the delays as a consequence of the real-time behaviour, the testbed experiments are configured such that they can be compared with the MATLAB & Simulink experiments: they have the same simulation duration, the same watermarking parameters which switch at simultaneous instances, the same data integrity attack configuration, and the same AGC cycles which are initiated simultaneously to the start of the experiments, i.e. at $t = 0s$. Also, the residual thresholds from the desktop simulations are used in the testbed simulations. The only difference is the initiation of the random Gaussian noises: for the desktop simulations this was identical each simulation run due to the use of random seeds, but for the testbed simulations these are initiated at an arbitrary time instance because they run continuously on the dSPACE simulator. Nonetheless, the desktop and the testbed simulations are comparable.

The same twelve simulation scenarios conducted for the desktop simulations are conducted here for the testbed simulations. The results are plotted in Figure 6.13, Figure 6.14, Figure 6.15 and Figure 6.16. Similar to Table 6.3, the metrics from the testbed simulations are summarised in Table 6.4.

			Detection Step k_d			Detection Ratio		
#	Cyber-Attack	DMWM	$\Delta f_1[k]$	$\Delta f_2[k]$	$\Delta P_{tie_1}[k]$	$\Delta f_1[k]$	$\Delta f_2[k]$	$\Delta P_{tie_1}[k]$
1	Scaling of $y_p[k]$	Non-active	30177	33888	30000	0.00146	0.00054	0.00263
2	Scaling of $u_c[k]$	Non-active	-	-	30116	0	0	0.04196
3	Scaling of $y_p[k]$, $u_c[k]$	Non-active	-	-	30000	0	0	0.02146
4	Scaling of $y_p[k]$	Active	31848	33943	30000	0.00192	0.00083	0.00338
5	Scaling of $u_c[k]$	Active	-	-	30113	0	0	0.05204
6	Scaling of $y_p[k]$, $u_c[k]$	Active	-	-	30000	0	0	0.02290
7	Replay of $y_p[k]$	Non-active	-	-	-	0	0	0
8	Replay of $u_c[k]$	Non-active	-	-	-	0	0	0
9	Replay of $y_p[k]$, $u_c[k]$	Non-active	-	-	-	0	0	0
10	Replay of $y_p[k]$	Active	50008	-	30000	0.00004	0	0.00829
11	Replay of $u_c[k]$	Active	-	-	30110	0	0	0.10379
12	Replay of $y_p[k]$, $u_c[k]$	Active	41005	30001	30000	0.00008	0.00025	0.01233

Table 6.4: Detection performance for each attack scenario with or without watermarking, simulated on the testbed



(a) Δf_i residuals during a scaling attack on $y_p[k]$ with $\Gamma_y = 1.7I_3$

(b) Δf_i residuals during a scaling attack on $u_c[k]$ with $\Gamma_u = 1.7I_2$

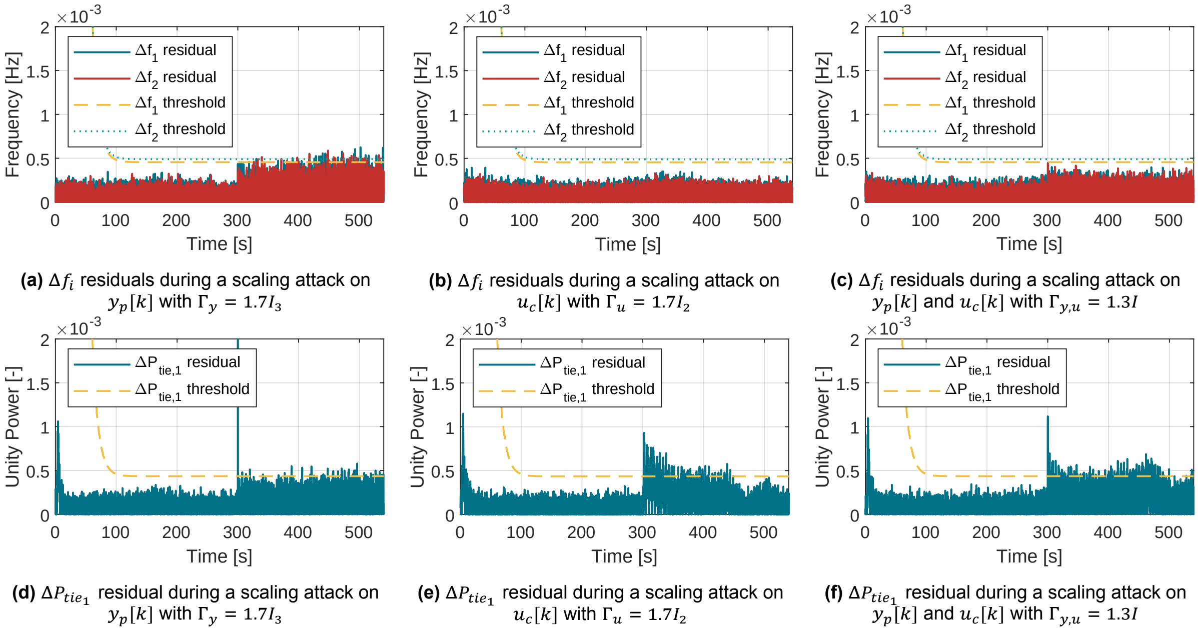
(c) Δf_i residuals during a scaling attack on $y_p[k]$ and $u_c[k]$ with $\Gamma_{y,u} = 1.3I$

(d) ΔP_{tie_1} residual during a scaling attack on $y_p[k]$ with $\Gamma_y = 1.7I_3$

(e) ΔP_{tie_1} residual during a scaling attack on $u_c[k]$ with $\Gamma_u = 1.7I_2$

(f) ΔP_{tie_1} residual during a scaling attack on $y_p[k]$ and $u_c[k]$ with $\Gamma_{y,u} = 1.3I$

Figure 6.13: Residuals and thresholds during a scaling attack *without* watermarking and $t_0 = 300s$, simulated on the testbed



(a) Δf_i residuals during a scaling attack on $y_p[k]$ with $\Gamma_y = 1.7I_3$

(b) Δf_i residuals during a scaling attack on $u_c[k]$ with $\Gamma_u = 1.7I_2$

(c) Δf_i residuals during a scaling attack on $y_p[k]$ and $u_c[k]$ with $\Gamma_{y,u} = 1.3I$

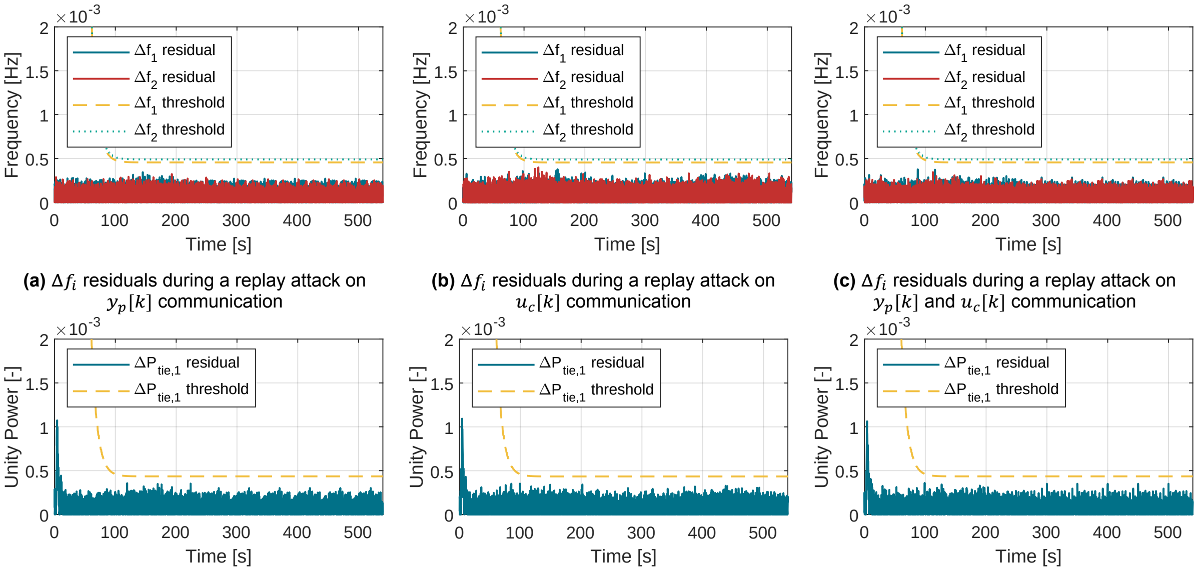
(d) ΔP_{tie_1} residual during a scaling attack on $y_p[k]$ with $\Gamma_y = 1.7I_3$

(e) ΔP_{tie_1} residual during a scaling attack on $u_c[k]$ with $\Gamma_u = 1.7I_2$

(f) ΔP_{tie_1} residual during a scaling attack on $y_p[k]$ and $u_c[k]$ with $\Gamma_{y,u} = 1.3I$

Figure 6.14: Residuals and thresholds during a scaling attack *with* watermarking and $t_0 = 300s$, simulated on the testbed

Again, a few remarkable findings are noted. Like the case for the desktop simulations, the FAR is zero for all scenarios. This is despite the additional noise of the physical wiring. Regarding the replay attack, this again is not detected without watermarking, while it is detected directly at k_0 when an attack on $y_p[k]$ is involved, and after 110 time steps (i.e. $1.1s$) when only $u_c[k]$ is attacked (note that for an entire attack to be detected, only one of the residuals has to have surpassed its threshold). This performance regarding k_d is almost identical for the scaling attacks, both in the scenarios with and without watermarking. Compared to the desktop simulations, this detection time is a big improvement. The major difference with the desktop simulations is the residuals of $\Delta f_i[k]$ when an attack on $u_c[k]$ is involved. In these cases, there is not a single instance where the residual of $\Delta f_i[k]$ surpasses its threshold, though the attack is still detected through the residual of ΔP_{tie_1}.



(a) Δf_i residuals during a replay attack on $y_p[k]$ communication

(b) Δf_i residuals during a replay attack on $u_c[k]$ communication

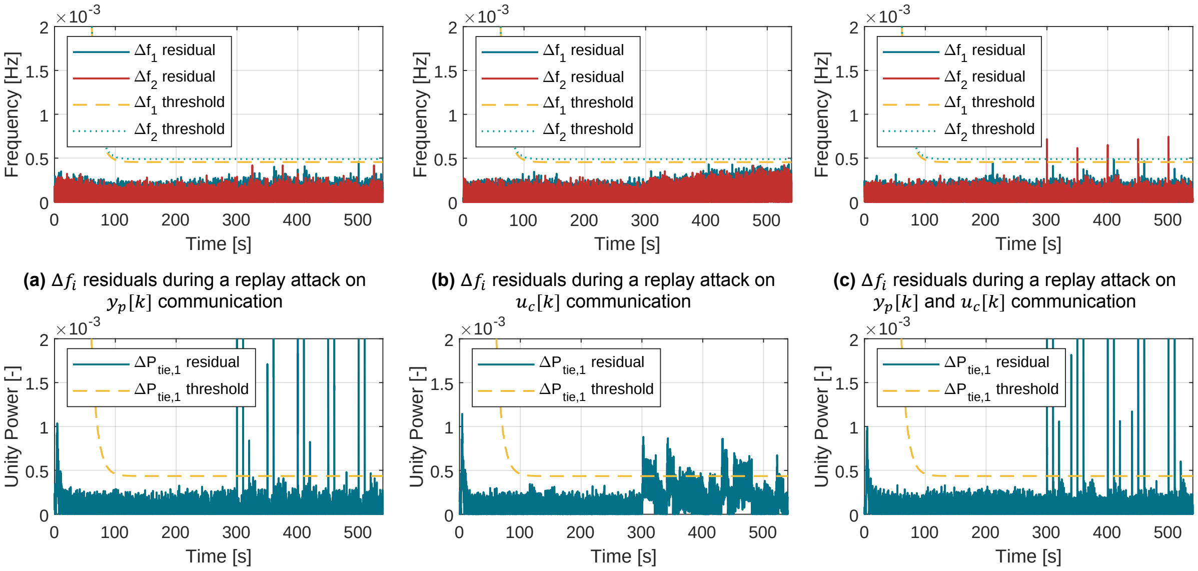
(c) Δf_i residuals during a replay attack on $y_p[k]$ and $u_c[k]$ communication

(d) ΔP_{tie_1} residual during a replay attack on $y_p[k]$ communication

(e) ΔP_{tie_1} residual during a replay attack on $u_c[k]$ communication

(f) ΔP_{tie_1} residual during a replay attack on $y_p[k]$ and $u_c[k]$ communication

Figure 6.15: Residuals and thresholds during a replay attack *without* watermarking and $T = 50s$, $t_r = 250s$ and $t_0 = 300s$, simulated on the testbed



(a) Δf_i residuals during a replay attack on $y_p[k]$ communication

(b) Δf_i residuals during a replay attack on $u_c[k]$ communication

(c) Δf_i residuals during a replay attack on $y_p[k]$ and $u_c[k]$ communication

(d) ΔP_{tie_1} residual during a replay attack on $y_p[k]$ communication

(e) ΔP_{tie_1} residual during a replay attack on $u_c[k]$ communication

(f) ΔP_{tie_1} residual during a replay attack on $y_p[k]$ and $u_c[k]$ communication

Figure 6.16: Residuals and thresholds during a replay attack *with* watermarking and $T = 50s$, $t_r = 250s$ and $t_0 = 300s$, simulated on the testbed

6.4. Conclusion

This chapter provided the results of this research. After verification of the LFC and AGC functionalities, the impact of the data integrity attacks was shown, both on desktop simulations using `MATLAB & Simulink`. Then, also on desktop simulations, twelve simulation scenarios were treated to analyse the behaviour of the IDS. Also, proper functionality of the HILDA testbed was verified by looking at the physical wiring, the network configuration and the cyber-attack configuration. Finally, validation results of the watermarking performance on the HILDA testbed were shown and briefly compared to the desktop simulation results. In the next chapter, the implications of the results are discussed.

7 Discussion

The added value of this chapter is to resolve the research question: *Can the inclusion of dynamic multiplicative watermarking improve detectability accuracy and speed for data integrity attacks using an observer-based detection scheme on an automatic generation controller, simulated real-time on the HILDA testbed?* Now that all results are presented, this question can be answered properly. Though some of the results were already briefly discussed in the respective chapters and sections, this chapter dives deeper into the central discussion points, possibly spanning over multiple results. If fundamental assumptions were made to get to the concerning results, these are clarified as well. Firstly, the results of the desktop simulations from section 6.1 are discussed, including the performance of the frequency control and that of the data integrity attacks, but excluding the IDS performance. Secondly, the testing results of the HILDA testbed from section 6.2 are discussed. Finally, the detection performance results of both the desktop and testbed simulations are covered, closed off by a comparison.

7.1. Performance of Frequency Control and Cyber-Attacks

A multitude of desktop simulation results were shown in section 6.1. This section argues that, based on these results, indeed proper functionality of the LFC, AGC and data integrity attacks can be verified. It does so by first analysing the frequency control mechanisms, followed by the attack impact.

7.1.1. Performance of Frequency Control

The overall performance of the frequency control regards the combined performance of the LFC and AGC mechanisms. First, is it argued that the assumptions on the simplified generator models and those on the state space model properties are legitimate. Then, the offset compensation as a result of the AGC mechanism activation is clarified.

The generator model simplifications are legitimate: In Assumption 3.2 is was assumed that no generator fatigue takes place. This is a reasonable assumption considering the simulations lasted maximally ten minutes, and the loads remained constant. Assumption 3.3 stated that all considered generators operate equally efficiently and effectively. This is a simplification of real-world scenarios, where no generator is exactly identical. This is why real-world applications include economic dispatch models, which can distribute electricity production based on the generator type and the market dynamics. For example, such dispatch models would give priority to generators which are more efficient/more sustainable regarding the environment. Including such environmentally friendly generators, in the likes of solar and wind energy, would have increased the relevance of this study, as these increasingly penetrate the existing power grid infrastructure. These are to a lesser extend controllable compared to the steam turbine generators of this study, because they depend on naturally uncontrollable resources. Logically, control mechanisms which can cope with this dependence are a hot research topic at the moment. However, involving these would considerably have increased the model complexity, loosing the scope of the other thesis contributions.

The state space models are controllable and observable: According to Assumption 4.1, (A_{d,p_i}, C_{d,p_i}) had to be a detectable pair. Also, for proper system operation, it has to be stabilisable. Both the observability and controllability matrices have rank 4. Since the LFC state space model has four states, i.e. $n = 4$, this implies that the system is observable and controllable, and therefore also detectable and stabilisable (which are the inferior mathematical duals). However, this changes when the amount of areas is increased. This resulted in an unstable system, i.e. the discrete eigenvalues fell out of the unit circle. This is because the implemented LFC state space model is designed in a minimalistic fashion: it only uses local states, i.e. $\Delta f_i(t), \Delta P_{m_i}(t), \Delta P_{g_i}(t)$ and $\Delta P_{tie_i}(t)$. When involving the change in frequencies of the other areas, as is done in [12, 90] for example, the instability is solved. This would lead to Assumption 3.1 being more fundamentally grounded. However, this also would increase model complexity, and using measurements from other LFC areas creates additional signals which have to travel over a network, thereby creating another cyber-attack vulnerability.

Automatic generation control compensates offset: As was depicted in Figure 6.2, there is an offset in all measurements $\Delta f_i[k]$ and $\Delta P_{tie_1}[k]$ if no AGC is applied. This is because the local LFC control is not designed to be connected with other LFC areas. In other words, from the perspective of the local controller, there is an unexpected leakage or inflow. Nonetheless, without AGC, the system is still asymptotically stable with a nonzero equilibrium point. This means that, in the case of a cyber-attack on the AGC mechanism, a simple means of mitigation could be to decouple the AGC mechanism from the local LFC area control, though this will lead to a loss of optimality regarding the electricity production.

No altering load profiles because of the `dSPACE` simulator: The reason no altering load profiles were considered is because of the real-time implementation of the `dSPACE` simulator. This is designed to run continuously, which makes it difficult to align the `Simulink` and the testbed simulations. With constant load changes, it becomes irrelevant when an attack is implemented or when a watermark switch occurs. To involve altering load profiles, some form of trigger would have to be involved to start the experiment on the testbed such that it matches the offline simulation. This could potentially be done through the 'start triggered' function in `ControlDesk`.

7.1.2. Performance of Data Integrity Attacks

The LFC mechanism showed various responses to the data integrity attacks. The performance of both the scaling and the replay attack is discussed here. Also, it is argued that both utilised data integrity attacks lack sophistication, and explained what the inclusion of this sophistication could bring about.

Scaling attacks can destabilise the system: It was shown in Figure 6.3 that the scaling attack on the AGC mechanism can destabilise the entire frequency control system. The magnitudes of the attacks (i.e. $\Gamma_y = 1.7 I_3$ and $\Gamma_u = 1.7 I_2$ for the attacks on $y_p[k]$ and $u_c[k]$ individually, and $\Gamma_{y,u} = 1.3I$ for the attacks on $y_p[k]$ and $u_c[k]$ combined) were selected such that the boundary of instability was approached. Indeed, for the attack scenarios 2 and 3, a growing trend of the measurement and control input fluctuation is visible, indicating instability. The scaling attack destabilises the system because, from the perspective of the AGC mechanism, the control inputs ΔP_{c_i} lead to a 'stronger' measurement response than expected. This response is stronger because the control inputs and/or the measurement response increase in magnitude due to the scaling. As a consequence, the AGC mechanism will try to overcompensate these occurrences, leading again to stronger a response, etcetera. For attack scenario 1, however, the scaling magnitude of $\Gamma_y = 1.7 I_3$ does not seem to destabilise the system, though it is decreasing the AGC performance by introducing a fluctuation in the $y_p[k]$ and $u_c[k]$ signals. The destabilising effect is more present for scenarios 2 and 3 compared to scenario 1 because, in steady state, the $u_c[k]$ signals are of magnitude $\lim_{k\to\infty} u_{c_1}[k] = 0.08$ and $\lim_{k\to\infty} u_{c_2}[k] = 0.12$ (the same magnitudes as their respective constant load profiles), while the $y_p[k]$ signals are of magnitude $\lim_{k\to\infty} \Delta f_i[k], \Delta P_{tie_1}[k] = 0$. Because of the greater magnitude of $u_c[k]$, the impact of a scaling multiplication is evidently larger. This effect is clearly visible when comparing the plots from scenario 2, i.e. Figure 6.3b and Figure 6.3e, to those of scenario 1, i.e. Figure 6.3a and Figure 6.3d.

Minor impact of replay attack: In Figure 6.4, the impact of the replay attack was shown with $t_0 = 300s$, $T = 50s$ and $t_r = 250s$. Opposed to the scaling attacks, the replay attacks appear to have the biggest impact on system performance when conducted on the $y_p[k]$ signals, i.e. in attack scenario 4. This can best be clarified by means of an illustration: say that the value of $\Delta f_1[k_r]$ is higher than that of $\Delta f_1[k_0]$. This means that over the replay period of $T = 50s$, the value of $\Delta f_1[k]$ decreases, and it will do so every time the sequence is replayed. The AGC mechanism tries to cope with this decrease by increasing the control input, which it also does every time the sequence is replayed. This leads to the deflection of $\Delta P_{c_i}[k]$ visible in Figure 6.4d. However, this deflection is only minor. This is because the system is attacked in steady state, and the load profiles are configured to be constant. Additionally, each LFC area has its own local controller which can cope with the minor impact of the attack on the AGC mechanism. The impact of the replay attack would be larger were the load profiles to fluctuate, and were the attacker to replay data which is out of sync with the actual data. This is illustrated in Figure 7.1, where a sinusoidal curve (with zero mean and 0.1 amplitude) and linearly increasing (from -0.04 to 0.12) load profile are implemented in LFC area 1 and 2, respectively, and where the same replay attack parameters are used as before. The same deflecting response is visible as in Figure 6.4d, but in this case for an attack on both $y_p[k]$ and $u_c[k]$, which barely resulted in any deflection in the case of constant load profiles, as visible in Figure 6.4f.

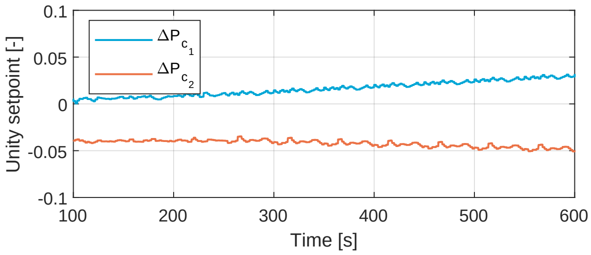


Figure 7.1: Response of $\Delta P_{c_i}[k]$ to a replay attack on $y_p[k]$ and $u_c[k]$ with $T = 50s$, $t_r = 250s$ and $t_0 = 300s$, while non-constant load profiles are inputted into the system, simulated on the desktop

Attacker lacks sophisticated approach: Both the scaling and the replay attack do not require any system knowledge. Though this is convenient, it could also be that the attacker does have this system knowledge to some degree. Once more system knowledge is used by the attacker, its capabilities to remain stealthy increase. In the specific case of this study, the attacker could for example record communication data for $80s$, which is the time it takes for the sequence of four watermarking parameters to start over. The attacker would then exactly mimic the watermarking behaviour, thereby remaining undetected, similar to the case without watermarking at all.

7.2. Performance of Testbed Simulations

Without a properly functioning testbed, all validation results would be dispensable. This section argues that, with the HILDA setup as it was used, the validations are legitimate. This is done by discussing the physical analogue signal wiring performance and the outcome of the attack PC configuration.

7.2.1. Performance of Physical Analogue Signal Wiring

It was argued in subsection 5.1.1 that the inclusion of physical wiring was one of the added values of testbed simulations compared to desktop simulations. This subsection serves as a clarification on the physical wiring impact. It does so by first analysing the signal offsets, followed by a consequence regarding the performance of the AGC algorithm.

Signal offset due to physical analogue wiring: To analyse the impact of the physical analogue communication wiring between the `dSPACE` ZIF connector and the `Beckhoff` I/O terminals, box plots of constant signal values as measured at their input terminals were provided in Figure 6.12. All measurements at the input terminals had an offset, both when sending through a signal of 0 and when sending through a signal of 0.9. For the case of the 0 signal, the offsets are all negative. Considering the signal range of -1 to 1, the smallest offset of the variable means is -0.01% for the $\Delta f_2[k]$ signal, while the largest (in absolute terms) is -0.04% for the $\Delta P_{tie_1}[k]$ signal. It is most likely that these negative offsets are a consequence of the voltage drop. When sending through a signal of 0.9, the offsets grow. The most interesting distinction with the 0 signal case is the sudden positive offset of the $\Delta f_i[k]$ and $\Delta P_{tie_1}[k]$ signals. Relative to each other, these signals remain approximately the same, but not compared to the $\Delta P_{c_i}[k]$ signals, which also have remained approximately the same relative to each other. This indicates that the new offset is most likely the result of the signal treatment in either the `dSPACE` simulator or the `Beckhoff` I/O terminals, and not a consequence of the wiring. Nonetheless, the offsets are only

$+0.10\%$, $+0.10\%$ and $+0.08\%$ for the $\Delta f_1[k]$, $\Delta f_2[k]$ and $\Delta P_{tie_1}[k]$ signals, respectively, and -0.08% and -0.05% for the ΔP_{c_1} and ΔP_{c_2} signals, respectively. These offsets are deemed sufficiently small to still perform experiments on the HILDA testbed.

Running AGC algorithm without having activated the LFC areas: Interestingly enough, due to the offsets resulting from the physical analogue communication wiring, it is not possible to activate the AGC mechanism on PLC 1 without also having activated the LFC mechanism on the MAB II simulator. This is because the AGC mechanism will try to cope with this offset, as it registers the offset as a discrepancy between the LFC areas which needs to be compensated. Therefore, the mechanism will keep on increasing the control inputs, up to the point where they reach the $+10\text{V}$ limit.

7.2.2. Configuration Outcome of Attack PC

As clarified in subsection 6.2.3, the data integrity attacks were simulated on the PLCs. It was already explained that this is a legitimate alternative to the original attack methodology. Nevertheless, this section briefly analysis the potential cause of the original attack methodology not functioning properly, and the subsequential taken steps, so that this can be taken into account by future testbed users.

TCP retransmissions when implementing the `MODBUS` protocol: *Plan A* was to use the `MODBUS` communication protocol, which is widely acknowledged by literature to be hackable. However, as was clarified, an issue occured regarding TCP. A *Wireshark* analysis indicated that *TCP retransmissions* were occurring. These happen when (a part of) a TCP segment gets lost on its way to the receiving end, resulting in a retransmission of the same Acknowledgement (ACK) number, which is used to either confirm or deny correctly receiving the message. In other words, due to some form of congestion on the network (presumably on the Ethernet switch), the `MODBUS` messages were not getting across. *Plan B* was to use the `EtherCAT` protocol, but as was clarified in subsection 6.2.3, this protocol is not susceptible to the original attack methodology of using the *Ettercap* tool in `Kali Linux`. Therefore, the attacks were simulated on the PLCs. Nonetheless, these simulations have been designed such that they mimic an actual attack. Because the attacks are executed real-time on the testbed, they still approach real-world scenarios more effectively than offline simulations.

7.3. Performance of Intrusion Detection System

In this final discussion section, the IDS performance is discussed. The emphasis it put on the added value of DMWM regarding the detection performance. This is firstly done for the remarkable findings which are similar for both the desktop and the HILDA testbed simulations. Secondly and finally, this section provides a discussion on the remarkable findings in which the two simulation types are different.

7.3.1. Similarities of Desktop and Testbed Simulations

There are a few aspects in which both simulation types show similar performances. The first is the stability of the watermarking filters, as the same filter parameters are used for the testbed simulations as for the desktop ones. Secondly, neither of the simulation types produced any false alarms. Finally, regarding the detection performance metrics, multiple similarities exist for scaling and replay attacks.

Stability of watermarking filters: In Assumption 4.2, it was assumed that the watermarking filters remained stable for all $\theta[k] \in \Theta$. Indeed, for the implemented watermark parameters this was the case. However, stability issues occured when selecting the random element of w_A to increase to $0.4 * rand^{1x3}$ or higher. This can be clarified by the fact that the parameter w_A forms a large part of the watermarking state space matrices. In some composition of w_A, it might occur that the state space matrices are configured unstably, which would make DMWM a burden instead of an asset.

Zero FAR for all simulation scenarios: One of the metrics for detection performance is the FAR. For every single simulation of section 6.1 and section 6.3, this was equal to zero. This is a consequence of the careful design of the robust threshold $\bar{y}_r[k]$, as proven with Equation 4.12. This $\bar{y}_r[k]$ is the same for each simulation, also for the testbed simulations. One might expect that the additional disturbance from the physical analogue communication wiring would results in false alarms, since this disturbance is not taken into account in the threshold computation. However, the results show this is not the case. It could even be attempted to lower the threshold by accordingly adjusting the α and δ parameters from Equation 4.11. Lowering the threshold would increase the DR, and possibly bring forward k_d.

Similar detection performance with and without DMWM against scaling attacks: For the scaling attack, the observer is capable of detecting the attack without the assistance of watermarking in all simulations. This is logical, as it is based on nominal performance, which is deteriorated when the communication signals are scaled. The simulations with and without DMWM show similar detection performances. This is because these attack do not influence the watermarking switching times, so Σ and Φ remain aligned. They do influence the signal magnitude. However, in steady state, the magnitude of the measurement signals is close to zero (both the local frequency control and the AGC are designed to aim at measurement magnitudes of zero). Inputting a signal which is close to zero into a multiplicative watermarking filter also results in an output close to zero. As a result, the detection performance barely increases when adding DMWM in the context of a scaling attack. If the watermarking model would be extended with an additive functionality, this would be different. Intuitively, if some value is added to each signal before being sent through the multiplicative watermarking filter, and on the other side of the network is first sent through the multiplicative equaliser before removing the added value, the multiplicative watermark would be more present on the signal [90]. Another option is to include a detector on the plant side of the model, because the control inputs $\Delta P_{c_i}[k]$ do not take on an (almost) zero value, i.e. $\lim_{k\to\infty} u_{c_i}[k] \neq 0$.

Detection of replay attacks involving $y_p[k]$ due to watermarking: Without watermarking, none of the replay attacks are detected. Considering the limited system performance deterioration visualised in Figure 6.4, this is unsurprising: based on the measured values and the control inputs, the Luenberger observer expects a steady state response, which is still the case for $k \in \kappa_a$, so the residual remains smaller than the threshold. When adding DMWM to the system, the detection performance goes from non-detecting to immediate detection for replay attacks involving $y_p[k]$. The detection comes about because different watermarking filter parameters (i.e. those from Σ) are replayed compared to the equalising parameters (i.e. those from Φ). In other words, $\theta_a[k] \neq \theta[k]$, where $\theta_a[k]$ is the set of watermarking parameters of the replayed sequence at timestep $[k]$, and $\theta[k]$ is the actual set of watermarking parameters. Consequently, the inequalities $\tilde{y}_p[k] \neq y_p[k]$ and $\tilde{u}_c[k] \neq u_c[k]$ are initiated with a sharp edge when either the watermarking parameters (of the actual or the replayed data) switch, or a new sequence of data is replayed. For replay attacks involving $y_p[k]$, this edge is most clearly noticable. This is because the measurements flow directly into the observer. The effect of the sharp edge on the residual lasts only a few time steps. This clarifies the low DR for these replay attacks. Fortunately, a single timestep where $y_r[k] \geq \bar{y}_r[k]$ is sufficient to detect the whole attack.

Detection of replay attacks on $u_c[k]$ due to watermarking: Similar to the replay attacks involving $y_p[k]$, the attacks on $u_c[k]$ are detected as a result of DMWM inclusion. This can be clarified by the same sharp edge resulting from the switching mechanism. However, unlike the measurements, the control inputs do not directly flow into the observer. The replayed data is first inputted into the LFC areas. Therefore, it is not directly the sharp edge which causes the residuals to increase, but rather their indirect effect on the LFC area responses. Such a sharp edge causes the $\Delta f_i[k]$ and $\Delta P_{tie_1}[k]$ to adjust sharply as well, which is interpreted by the observer as abnormal system behaviour. This indirect effect is why it takes 106 and 110 time steps before the attack on $u_c[k]$ is detected for the desktop and testbed simulations, respectively. On the other hand, the effect of the sharp edge lasts longer, resulting in a higher DR for both simulation types.

7.3.2. Differences of Desktop and Testbed Simulations

Next to the similarities, there were some discrepancies regarding the IDS performances of both simulation types. These are covered in this section. First, it is debated that the detection occurs more quickly for the testbed simulations. This is followed by a comparison of the overall DR. This section ends with a remark on the additional cyber-security risks regarding the testbed.

Quicker detection of scaling attacks for the testbed simulations: The k_d of all scaling attack scenarios is compared between the two simulation types. It is noticed that, overall, the results for the testbed simulations are better. More specifically, the first detection is performed instantly for the attack scenarios 1, 3, 4 and 6. This instant detection is only the case for scenario 4 on the desktop simulations. The instant detection is thanks to the spike at $t_0 = 300s$ of the $\Delta P_{tie_1}[k]$ residual. This spike is even stronger when DMWM is added to the system. This is because, next to the sudden scaling of the original signal, Equation 4.5 no longer holds.

Higher DR in all simulation scenarios for the desktop simulations: For every single scenario, the DR was higher for the desktop simulations compared to the testbed simulations. This most likely has to do with the real-time capabilities of the testbed. From the perspective of PLC 1, it takes multiple individual cycle times of (other) real-time programmes before a response to a specific control signal is received. This was also visualised in Figure 5.7. This delay, which goes both ways, makes it more difficult for the Luenberger observer to make correct estimations of the measurements. Nevertheless, only a single instance of detection is required, so arguably k_d is the more important metric.

Compromisable DMWM functionality on the testbed: As final note, this study predefined the watermarking parameters in the testbed, and the switching times were communicated over the network as separate signals. These aspects could compromise the added value of the watermarking, because they make it easier for attackers to identify the watermarking configuration. Therefore, the watermarking algorithm should be constructed such that the algorithm can be published, and nevertheless its functionality (the improvement of attack detectability) holds. Examples of this are provided in [51, 55]

7.4. Conclusion

The goal of this chapter was to provide a discussion on the presented results. The performance of the HILDA testbed, the LFC mechanism, the data integrity attacks and the IDS were all discussed. For each, the optional assumptions were defended. After having done so, looking back to the research question, it can be claimed that, indeed, both the detectability accuracy (given by DR) and speed (given by k_d) are improved by including DMWM when simulated on the HILDA testbed. The performance differences between the simulation types demonstrate the added value of the HILDA testbed.

8 Conclusions and Recommendations

The Industrial Control System (ICS) plays a crucial role in today's society. All modern critical infrastructures use some form of ICS to empower them in fulfilling their critical function. History has shown that this has not gone unnoticed for cyber-attackers, which to a rising degree claim ICSs as their target. One of these critical infrastructures is the power grid. The frequency of this power grid is regulated by Load Frequency Control (LFC) mechanisms, in which Automatic Generation Control (AGC) is a dominant algorithm. There are multiple ways to model such a LFC mechanism which involves AGC, of which this study choose a split LFC area and combined AGC model. This model was designed in continuous time, and transformed to discrete time. Popular attacks on control systems specifically are data integrity attacks, which aim at compromising the integrity of communication data being exchanged between a plant site and a (geographically distant) controller. Through a Man-In-The-Middle (MITM) attack, an attacker can read and change this communication data, i.e. compromise the integrity. A basic data integrity attack, in the form of a scaling attack, and a slightly more advanced one, in the form of a replay attack, were considered in this study. Once these attacks are executed, they should be adequately detected so that they can be properly mitigated. This detection is done through an Intrusion Detection System (IDS). This study used an active IDS in the form of Dynamic Multiplicative Watermarking (DMWM). This was combined with a Luenberger observer, which performed the actual detection by comparing a residual with a robustly designed threshold. To verify the performance of the DMWM in the context of LFC and data integrity attacks, `MATLAB & Simulink` simulations were performed. However, this study also argued that such simulations were falling short due to their lack of physical wiring and real-time performance. Therefore, another major part of this study was the design, construction and configuration of the Hardware-In-the-Loop Detection of Attacks (HILDA) testbed. Based on the combined LFC, data integrity attack and DMWM model, it was determined what appliances would carry out what functionality. The testbed was adequately configured, both physically and regarding software. With the HILDA testbed in place, it was possible to conduct experiments on it and properly validate the DMWM performance. In total, this thesis had three research contributions, which are elaborated in section 8.1, after which some central recommendations are provided in section 8.2.

8.1. Conclusions

This thesis aimed at three contributions, which are summarised by the subsequent enumeration:

1. **The first research on DMWM in the context of AGC:** To the author's knowledge, never before had a research been conducted on DMWM in the context of AGC, let alone while being subjected to scaling and replay attacks. It was shown that, also in this context, DMWM was of added value for the detection performance. For the replay attacks, the detection performance showed clear improvement with instant detection. This performance increase was thanks to the dynamic functionality of the watermarker. Due to the near-zero measurements, DMWM had a more modest added value for the scaling attacks, which could be solved by including additive watermarking, or adding an observer at the plant side of the network.

2. **Design and construct a testbed for future research:** The goal was to develop a testbed which not only would be of added value for this specific research, but also for many more researches to come. To achieve this, the design procedure was performed very cautiously. The verification results showed that the design, construction and configuration procedures were performed successfully, resulting in an operational testbed for future use.

3. **Produce the first validation results on the testbed:** With the proper model and functioning HILDA testbed in place, the final goal was to extract the first results from the HILDA testbed which would be of added value to the `MATLAB & Simulink` simulations. Indeed, a discrepancy in detection performance for the different simulation types was shown, which, together with some unexpected design twists, exhibit the added value of the testbed.

8.2. Recommendations

Next to the already reached conclusions, the author has some recommendations for future research. This research touches upon many different (sub)topics, so a lot more is possible. Only the most important recommendations are stated here, which are clustered by the following enumeration:

1. **Preparation for future users of the HILDA testbed:** As the HILDA testbed was also designed for future usage, recommendations are provided on how to best conduct this future usage. First of all, it is important to analyse and understand all testbed components and the combined setup. Also, when not fully acquainted with the programming of the `dSPACE` or `Beckhoff` equipment, the future users are advised to look at some of the tutorials which were provided in this report. This should enable them to properly use the testbed, and possibly use the attack PC.

2. **Enhance LFC model and design:** In some areas, the LFC model could be improved. Mainly the scalability is something the author would recommend researching more deeply. It would be interesting to analyse the effect of LFC model scaling on the detection performance. Also, instead of the actual measurements, the estimated measurements could be used as inputs of the AGC algorithm. The noise of the estimates is lower than that of the actual measurements, so this might lead to increased AGC performance. Finally, it would be very interesting to include Renewable Energy System (RES) technologies.

3. **Dynamic watermarking improvement:** Because of the predefined watermarking parameters and the signal communication of the parameter states, the considered implementation of DMWM is not secure enough for real-world applications. The author would advice on involving mechanisms which take away these elements which could be compromised by attackers. Also, it would be interesting to further analyse the effects of different 1) watermarking periods, 2) watermarking switching frequencies, and 3) residual threshold sizes. Finally, the watermarking mechanism could be accompanied by more advanced observers.

Bibliography

[1] Alireza Abbaspour, Arman Sargolzaei, and Kang Yen. “Detection of false data injection attack on load frequency control in distributed power systems”. In: *North American Power Symposium (NAPS)* (2017), pp. 1–6. DOI: `10.1109/NAPS.2017.8107333`.

[2] Tschroub Abdelghani. “Industrial control systems (ics) security in power transmission network”. In: *Proceedings of Algerian Large Electrical Network Conference, CAGRE* (2019), pp. 17–20. DOI: `10.1109/CAGRE.2019.8713289`.

[3] Ahmed Abdelwahab, Walter Lucia, and Amr Youssef. “Set-Theoretic Control for Active Detection of Replay Attacks with Applications to Smart Grid”. In: *IEEE Conference on Control Technology and Applications (CCTA)* 2 (2020), pp. 1004–1009. DOI: `10.1109/CCTA41146.2020.9206373`.

[4] E. Acha, S. Garci, and A. Go. “Overview of power electronics technology and applications in power generation transmission and distribution”. In: *Journal of Modern Power Systems and Clean Energy* 5 (2017), pp. 499–514. DOI: `10.1007/s40565-017-0308-x`.

[5] Uttam Adhikari, Thomas H. Morris, and Shengyi Pan. “A cyber-physical power system test bed for intrusion detection systems”. In: *IEEE Power and Energy Society General Meeting* (2014). DOI: `10.1109/PESGM.2014.6939262`.

[6] Hossein Ghassempour Aghamolki, Zhixin Miao, and Lingling Fan. “A hardware-in-the-loop SCADA testbed”. In: *2015 North American Power Symposium (NAPS)*. 2015, pp. 1–6. DOI: `10.1109/NAPS.2015.7335093`.

[7] Mohammad Mehdi Ahmadian, Mehdi Shajari, and Mohammad Ali Shafiee. “Industrial control system security taxonomic framework with application to a comprehensive incidents survey”. In: *International Journal of Critical Infrastructure Protection* 29 (2020), pp. 334–356. DOI: `10.1016/j.ijcip.2020.100356`.

[8] Monjur Ahmed and Mohammad Ashraf Hossain. “Cloud computing and security issues in the cloud”. In: *International Journal of Network Security & Its Applications* 6.1 (2014), p. 25. DOI: `10.5121/ijnsa.2014.6103`.

[9] Cristina Alcaraz et al. “Secure Management of SCADA Networks”. In: *Novatica, New Trends in Network Management* 9.6 (2008), pp. 22–28.

[10] Muhammad Qasim Ali et al. “Two-tier data-driven intrusion detection for automatic generation control in smart grid”. In: *IEEE Conference on Communications and Network Security (CNS)* (2014), pp. 292–300. DOI: `10.1109/CNS.2014.6997497`.

[11] Thiago Alves, Rishabh Das, and Thomas Morris. “Virtualization of industrial control system testbeds for cybersecurity”. In: *ACM International Conference Proceeding Series*. 2016, pp. 10–14. DOI: `10.1145/3018981.3018988`.

[12] Amir Ameli et al. “Attack Detection and Identification for Automatic Generation Control Systems”. In: *IEEE Transactions on Power Systems* 33.5 (2018), pp. 4760–4774. DOI: `10.1109/TPWRS.2018.2810161`.

[13] Oxana Andreeva et al. *Industrial Control Systems Vulnerabilities Statistics*. Tech. rep. Kaspersky, 2016.

[14] Aditya Ashok et al. “Experimental evaluation of cyber attacks on Automatic Generation Control using a CPS Security Testbed”. In: *IEEE Power and Energy Society General Meeting*. IEEE, 2015. DOI: `10.1109/PESGM.2015.7286615`.

[15] Michael Assante and Robert Lee. *The Industrial Control System Cyber Kill Chain*. Tech. rep. Sans Institute, 2015.

[16] Abdelrahman Ayad, Mohsen Khalaf, and Ehab El-Saadany. “Detection of False Data Injection Attacks in Automatic Generation Control Systems Considering System Nonlinearities”. In: *IEEE Electrical Power and Energy Conference (EPEC)* (2018), pp. 1–6. DOI: 10.1109/EPEC.2018.8598328.

[17] Tummala S. L. V. Ayyarao and I. Ravi Kiran. “A Two-stage Kalman Filter for Cyber-attack Detection in Automatic Generation Control System”. In: *Journal of Modern Power Systems and Clean Energy* (2021). DOI: 10.35833/MPCE.2019.000119.

[18] Asma Aziz, Amanullah Mto, and Alex Stojsevski. “Automatic Generation Control of Multigeneration Power System”. In: *Journal of Power and Energy Engineering* 2 (2014), pp. 312–333. DOI: 10.4236/jpee.2014.24043.

[19] David Bailey and Edwin Wright. *Practical SCADA for industry*. Elsevier, Amsterdam, 2003.

[20] Christopher Beasley et al. “A survey of electric power synchrophasor network cyber security”. In: *IEEE PES Innovative Smart Grid Technologies Conference Europe* (2015), pp. 1–5. DOI: 10.1109/ISGTEurope.2014.7028738.

[21] Andreea Bendovschi. “Cyber-Attacks – Trends, Patterns and Security Countermeasures”. In: *Procedia Economics and Finance* 28 (2015), pp. 24–31. DOI: 10.1016/s2212-5671(15)01077-1.

[22] Samaresh Bera, Sudip Misra, and Senior Member. “Cloud Computing Applications for Smart Grid : A Survey”. In: *IEEE Transactions on Parallel and Distributed Systems* 26.5 (2015), pp. 1477–1494. DOI: 10.1109/TPDS.2014.2321378.

[23] Dillon Beresford. *Exploiting Siemens Simatic S7 PLCs Prepared for Black Hat USA+2011*. Tech. rep. Black Hat USA, 2011.

[24] Giuseppe Bernieri et al. “Monitoring system reaction in cyber-physical testbed under cyber-attacks”. In: *Computers and Electrical Engineering* 59 (2017), pp. 86–98. DOI: 10.1016/j.compeleceng.2017.02.010.

[25] Deval Bhamare et al. “Cybersecurity for industrial control systems: A survey”. In: *Computers and Security* 89 (2020). DOI: 10.1016/j.cose.2019.101677.

[26] Matt Bishop. “What is computer security?” In: *IEEE Security & Privacy* 1.1 (2003), pp. 67–69. DOI: 10.1109/MSECP.2003.1176998.

[27] Saroj Biswas and Arif Sarwat. “Vulnerabilities in two-area automatic generation control systems under cyberattack”. In: *Proceedings - 2016 Resilience Week (RWS)* (2016), pp. 40–45. DOI: 10.1109/RWEEK.2016.7573304.

[28] David Bombal. *How TCP really works // Three-way handshake // TCP/IP Deep Dive*. 2022. URL: https://www.youtube.com/watch?v=rmFX1V49K8U (visited on 07/09/2022).

[29] Eric Byres and Dan Hoffman. *The Myths and Facts behind Cyber Security Risks for Industrial Control Systems*. Tech. rep. Proceedings of the VDE Kongress, 2004.

[30] Richard Candell, Keith Stouffer, and Timothy Zimmerman. *An Industrial Control System Cybersecurity Performance Testbed*. Tech. rep. NIST, 2016.

[31] Andreia M. Carreiro, Humberto M. Jorge, and Carlos Henggeler Antunes. “Energy management systems aggregators : A literature survey”. In: *Renewable and Sustainable Energy Reviews* 73 (2017), pp. 1160–1172. DOI: 10.1016/j.rser.2017.01.179.

[32] Kaustav Chatterjee, V. Padmini, and S. A. Khaparde. “Review of cyber attacks on power system operations”. In: *TENSYMP 2017 - IEEE International Symposium on Technologies for Smart Cities* (2017). DOI: 10.1109/TENCONSpring.2017.8070085.

[33] Bo Chen et al. “Implementing a real-time cyber-physical system test bed in RTDS and OPNET”. In: *North American Power Symposium (NAPS)*. 2014, pp. 1–6. DOI: 10.1109/NAPS.2014.6965381.

[34] Bo Chen et al. “Implementing attacks for modbus/TCP protocol in a real-time cyber physical system test bed”. In: *Proceedings - CQR 2015: 2015 IEEE International Workshop Technical Committee on Communications Quality and Reliability* (2015). DOI: 10.1109/CQR.2015.7129084.

[35] Yulia Cherdantseva et al. “A review of cyber security risk assessment methods for SCADA systems”. In: *Computers and Security* 56 (2016), pp. 1–27. DOI: 10.1016/j.cose.2015.09.009.

[36] Nabin Chowdhury and Vasileios Gkioulos. “Cyber security training for critical infrastructure protection: A literature review”. In: *Computer Science Review* 40 (2021), pp. 1–20. DOI: 10.1016/j.cosrev.2021.100361.

[37] Mehmet Hazar Cintuglu et al. “A Survey on Smart Grid Cyber-Physical System Testbeds”. In: *IEEE Communications Surveys and Tutorials* 19.1 (2017), pp. 446–464. DOI: 10.1109/COMST.2016.2627399.

[38] PLC Coder. *Communicating between Beckhoff controllers part 2: ADS*. 2020. URL: https://www.plccoder.com/communicating-between-beckhoff-controllers-part-2-ads/ (visited on 07/09/2022).

[39] Edward J M Colbert and Alexander Kott. *Cyber-security of SCADA and Other Industrial Control Systems*. Vol. 63. Springer, Berlin, 2016. DOI: 10.1007/978-3-319-32125-7.

[40] IEEE Standards Coordinating Committee. “IEEE No 94-1970”. In: *IEEE Standard Definitions of Terms for Automatic Generation Control on Electric Power Systems* (1970), pp. 1–12. DOI: 10.1109/IEEESTD.1970.7440744.

[41] IEEE Standards Coordinating Committee. “IEEE Std 610.12-1990”. In: *CA: IEEE Computer Society* 169 (1990), p. 132. DOI: 10.1109/IEEESTD.1990.101064.

[42] Phoenix Contact. *Phoenix Contact Terminal Blocks EPaper Datasheet*. 2020. URL: https://www.phoenixcontact.com/assets/2018/interactive_ed/101_141234/index.html#0 (visited on 06/24/2022).

[43] Oracle Corporation. *Oracle VM VirtualBox Installation Guide*. URL: https://www.virtualbox.org/manual/ch02.html (visited on 06/27/2022).

[44] Derui Ding et al. “A survey on security control and attack detection for industrial cyber-physical systems”. In: *Neurocomputing* 275 (2018), pp. 1674–1683. DOI: 10.1016/j.neucom.2017.10.009.

[45] Mohamed Amine Douad and Youcef Dahmani. “ARTT taxonomy and cyber-attack Framewok”. In: *First International Conference on New Technologies of Information and Communication (NTIC)* (2015), pp. 1–6. DOI: 10.1109/NTIC.2015.7368742.

[46] Zakarya Drias, Ahmed Serhrouchni, and Olivier Vogel. “Taxonomy of attacks on industrial control protocols”. In: *International Conference on Protocol Engineering, ICPE 2015 and International Conference on New Technologies of Distributed Systems, NTDS 2015 - Proceedings* January 2016 (2015). DOI: 10.1109/NOTERE.2015.7293513.

[47] Kelvin T Erickson and John L Hedrick. *Plant-wide process control*. Vol. 4. John Wiley & Sons, Hoboken, 1999.

[48] Xiaohe Fan et al. “Overview of cyber-security of industrial control system”. In: *International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC) - Proceedings* (2015). DOI: 10.1109/SSIC.2015.7245324.

[49] Chongrong Fang et al. “Cost-Effective Watermark Based Detector for Replay Attacks on Cyber-Physical Systems”. In: *Asian Control Conference (ASCC)* 11.1 (2017), pp. 940–945.

[50] James P Farwell and Rafal Rohozinski. “Stuxnet and the future of cyber war”. In: *Survival* 53.1 (2011), pp. 23–40.

[51] Riccardo M.G. Ferrari and André M.H. Teixeira. “A Switching Multiplicative Watermarking Scheme for Detection of Stealthy Cyber-Attacks”. In: *IEEE Transactions on Automatic Control* 66.6 (2020), pp. 2558–2573. DOI: 10.1109/TAC.2020.3013850.

[52] Riccardo M.G. Ferrari and André M.H. Teixeira. “Detection and Isolation of Replay Attacks through Sensor Watermarking”. In: *IFAC-PapersOnLine* 50.1 (2017), pp. 7363–7368. DOI: 10.1016/j.ifacol.2017.08.1502.

[53] Riccardo M.G. Ferrari and André M.H. Teixeira. "Detection and Isolation of Routing Attacks through Sensor Watermarking". In: *American Control Conference (ACC)* (2017), pp. 5436–5442. DOI: 10.23919/ACC.2017.7963800.

[54] World Economic Forum. *The Global Risk Report 2018*. Tech. rep. World Economic Forum Geneva, 2018.

[55] Alexander J Gallo, Francesca Boem, and Thomas Parisini. "Distributed cyber-attack isolation for large-scale interconnected systems". In: *European Control Conference (ECC)* (2021), pp. 48–53. DOI: 10.23919/ECC54610.2021.9655176.

[56] Alexander J. Gallo et al. "Distributed watermarking for secure control of microgrids under replay attacks". In: *IFAC-PapersOnLine* 51.23 (2018), pp. 182–187. DOI: 10.1016/j.ifacol.2018.12.032.

[57] Brendan Galloway and Gerhard P. Hancke. "Introduction to industrial control networks". In: *IEEE Communications Surveys and Tutorials* 15.2 (2012), pp. 860–880. DOI: 10.1109/SURV.2012.071812.00124.

[58] Haihui Gao et al. "The design of ICS testbed based on emulation, physical, and simulation (EPS-ICS Testbed)". In: *Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)* (2013), pp. 420–423. DOI: 10.1109/IIH-MSP.2013.111.

[59] Wei Gao and Thomas Morris. "On Cyber Attacks and Signature Based Intrusion Detection for Modbus Based Industrial Control Systems". In: *Journal of Digital Forensics, Security and Law* 9.1 (2014), pp. 37–56. DOI: 10.15394/jdfsl.2014.1162.

[60] Yangyang Geng et al. "A survey of industrial control system testbeds". In: *IOP Conference Series: Materials Science and Engineering* 569.4 (2019), pp. 1–10. DOI: 10.1088/1757-899X/569/4/042030.

[61] Mohsen Ghaderi, Kian Gheitasi, and Walter Lucia. "A Blended Active Detection Strategy for False Data Injection Attacks in Cyber-Physical Systems". In: *Transaction on Control of Network Systems* 8.1 (2021), pp. 168–176. DOI: 10.1109/TCNS.2020.3024315.

[62] Amrita Ghosal. "Key Management Systems for Smart Grid Advanced Metering Infrastructure : A Survey". In: *IEEE Communications Surveys & Tutorials* 21.3 (2019), pp. 2831–2848. DOI: 10.1109/COMST.2019.2907650.

[63] Jairo Giraldo et al. "A survey of physics-based attack detection in cyber-physical systems". In: *ACM Computing Surveys* 51.4 (2018), pp. 1–36. DOI: 10.1145/3203245.

[64] Jairo Giraldo et al. "Security and Privacy in Cyber-Physical Systems: A Survey of Surveys". In: *IEEE Design and Test* 34.4 (2017), pp. 7–17. DOI: 10.1109/MDAT.2017.2709310.

[65] Beckhoff Automation GmbH. *Beckhoff Information System*. URL: https://infosys.beckhoff.com/ (visited on 06/07/2022).

[66] Beckhoff Automation GmbH. *Documentation PS1011-2410-0000 Power Supply*. 2022. URL: https://download.beckhoff.com/download/document/io/power-supplies/PS1011-2410-0000en.pdf (visited on 06/18/2022).

[67] Beckhoff Automation GmbH. *EtherCAT – the Ethernet Fieldbus*. URL: https://www.beckhoff.com/nl-nl/products/i-o/ethercat/ (visited on 06/07/2022).

[68] Beckhoff Automation GmbH. *Manual TE100 TwinCAT 3 | EAP*. URL: https://download.beckhoff.com/download/Document/automation/twincat3/EAP_EN.pdf (visited on 06/07/2022).

[69] Beckhoff Automation GmbH. *Manual TE13xx TwinCAT 3 | Scope View*. 2022. URL: https://download.beckhoff.com/download/Document/automation/twincat3/TE13xx_TwinCAT_3_ScopeView_EN.pdf (visited on 06/15/2022).

[70] Beckhoff Automation GmbH. *Manual TE1800 TwinCAT 3 | PLC HMI*. 2022. URL: https://download.beckhoff.com/download/Document/automation/twincat3/TF1800_TC3_PLC_HMI_EN.pdf (visited on 06/15/2022).

[71] Beckhoff Automation GmbH. *TwinCAT Automation Software Product Page*. URL: https://www.beckhoff.com/en-us/products/automation/twincat/#text_bild_2 (visited on 06/12/2022).

[72] dSPACE GmbH. *MicroAutoBox II Product Information*. 2020. URL: https://www.dspace.com/shared/data/pdf/2020/dSPACE_MicroAutoBox-II-Brochure_2020-08_01_200811_E.pdf (visited on 06/12/2022).

[73] Siobhan Gorman. *Electricity grid in US penetrated by spies*. 2009. URL: https://www-wsj-com.tudelft.idm.oclc.org/articles/SB123914805204099085 (visited on 08/25/2021).

[74] Andreas Granat, Hans Höfken, and Marko Schuba. "Intrusion detection of the ICS protocol EtherCAT". In: *2nd International Conference on Computer, Network Security and Communication Engineering*. 2017, pp. 113–117.

[75] EtherCAT Technology Group. *EtherCAT for factory networking*. 2009. URL: https://www.ethercat.org/download/documents/pcc_0409_etg_e.pdf (visited on 06/07/2022).

[76] EtherCAT Technology Group. *EtherCAT for factory networking: EtherCAT Automation Protocol (EAP)*. 2010. URL: https://www.ethercat.org/download/documents/EtherCAT_EAP_EN.pdf (visited on 06/07/2022).

[77] EtherCAT Technology Group. *EtherCAT webpage*. URL: https://www.ethercat.org/default.htm (visited on 06/07/2022).

[78] M. Zekeriya Gunduz and Resul Das. "Analysis of cyber-attacks on smart grid applications". In: *International Conference on Artificial Intelligence and Data Processing (IDAP)* (2018). DOI: 10.1109/IDAP.2018.8620728.

[79] Adam Hahn. "Operational technology and information technology in industrial control systems". In: *Cyber-security of SCADA and other industrial control systems*. Spinger, Berlin, 2016, pp. 51–68. DOI: 10.1007/978-3-319-32125-7_4.

[80] Adam Hahn et al. "Cyber-physical security testbeds: Architecture, application, and evaluation for smart grid". In: *IEEE Transactions on Smart Grid* 4.2 (2013), pp. 847–855. DOI: 10.1109/TSG.2012.2226919.

[81] Adam Hahn et al. "Development of the PowerCyber SCADA security testbed". In: *ACM International Conference Proceeding Series* (2010), pp. 1–4. DOI: 10.1145/1852666.1852690.

[82] Dong Han et al. "A real application of measurement-based load modeling in large-scale power grids and its validation". In: *IEEE Transactions on Power Systems* 24.4 (2009), pp. 1756–1764. DOI: 10.1109/TPWRS.2009.2030298.

[83] Hayes. *Configuring the TwinCAT I/O System*. 2004. URL: https://instrumentacionycontrol.net/wp-content/uploads/2017/11/IyCnet_Configure_the_TwinCAT_IO_System.pdf (visited on 07/09/2022).

[84] Haibo He and Jun Yan. "Cyber-physical attacks and defences in the smart grid: a survey". In: *IET Cyber-Physical Systems: Theory & Applications* 1.1 (2016), pp. 13–27. DOI: 10.1049/iet-cps.2016.0019.

[85] Kevin E Hemsley and Ronald E Fisher. *History of Industrial Control System Cyber Incidents*. Tech. rep. U.S. Department of Energy, 2018, pp. 1–37.

[86] Leslie Hewitson, Mark Brown, and Ramesh Balakrishnan. *Practical power system protection*. Elsevier, Amsterdam, 2004.

[87] Hannes Holm et al. "A survey of industrial control system testbeds". In: *IOP Conference Series: Materials Science and Engineering* 569.4 (2019), pp. 11–26. DOI: 10.1088/1757-899X/569/4/042030.

[88] Yan Hu et al. "A survey of intrusion detection on industrial control systems". In: *International Journal of Distributed Sensor Networks* 14.8 (2018), pp. 1–14. DOI: 10.1177/1550147718794615.

[89] K. Huang, M. Siegel, and S. Madnick. "Systematically Understanding the Cyber Attack Business: A Survey". In: *Notfall und Rettungsmedizin* 14.4 (2018), pp. 303–304. DOI: 10.1007/s10049-011-1440-1.

[90] Tong Huang et al. “An online detection framework for cyber attacks on automatic generation control”. In: *IEEE Transactions on Power Systems* 33.6 (2018), pp. 6816–6827. DOI: 10.1109/TPWRS.2018.2829743.

[91] Xiaoge Huang, Zhijun Qin, and Hui Liu. “A Survey on Power Grid Cyber Security: From Component-Wise Vulnerability Assessment to System-Wide Impact Analysis”. In: *IEEE Access* 6 (2018), pp. 69023–69035. DOI: 10.1109/ACCESS.2018.2879996.

[92] Peter Huitsing et al. “Attack taxonomies for the Modbus protocols”. In: *International Journal of Critical Infrastructure Protection* 1 (2008), pp. 37–44. DOI: 10.1016/j.ijcip.2008.08.003.

[93] Abdulmalik Humayed et al. “Cyber-Physical Systems Security - A Survey”. In: *IEEE Internet of Things Journal* 4.6 (2017), pp. 1802–1831. DOI: 10.1109/JIOT.2017.2703172.

[94] Eric Hutchins, Michael Cloppert, and Rohan Amin. “Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains”. In: *6th International Conference on Information Warfare and Security (ICIW)* 1.1 (2011), pp. 113–125.

[95] Nasser Jaleeli et al. “Understanding automatic generation control”. In: *IEEE Transactions on Power Systems* 7.3 (1992), pp. 1106–1122. DOI: 10.1109/59.207324.

[96] Stig O. Johnsen, Andreas Aas, and Ying Qian. “Sector-Specific Information Infrastructure Issues in the Oil, Gas, and Petrochemical Sector”. In: *Critical Infrastructure Protection: Information Infrastructure Models, Analysis, and Defense*. Ed. by Javier Lopez, Roberto Setola, and Stephen D. Wolthusen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 235–279. DOI: 10.1007/978-3-642-28920-0_11.

[97] Thomas A Johnson. *Cybersecurity: Protecting critical infrastructures from cyber attack and cyber warfare*. CRC Press, Boca Raton, 2015.

[98] Jaroslav Kautsky, Nancy K Nichols, and Paul Van Dooren. “Robust pole assignment in linear state feedback”. In: *International Journal of control* 41.5 (1985), pp. 1129–1155. DOI: 10.1080/0020718508961188.

[99] Hakan Kayan et al. “Cybersecurity of industrial cyber-physical systems: a review”. In: *ACM Computing Surveys (CSUR)* (2021). DOI: 10.1145/3510410.

[100] Mohsen Khalaf, Amr Youssef, and Ehab El-Saadany. “Detection of False Data Injection in Automatic Generation Control Systems Using Kalman Filter”. In: *IEEE Electrical Power and Energy Conference (EPEC)* (2018). DOI: 10.1109/EPEC.2017.8286194.

[101] Amir Khazraei et al. “A New Watermarking Approach for Replay Attack Detection in LQG Systems”. In: *IEEE 56th Annual Conference on Decision and Control (CDC)* (2017), pp. 5143–5148. DOI: 10.1109/CDC.2017.8264421.

[102] Amir Khazraei et al. “Replay attack detection in a multi agent system using stability analysis and loss effective watermarking”. In: *American Control Conference* (2017), pp. 4778–4783. DOI: 10.23919/ACC.2017.7963694.

[103] Brendan Kirby, Erik Ela, and Michael Milligan. “Chapter 7 - Analyzing the Impact of Variable Energy Resources on Power System Reserves”. In: *Renewable Energy Integration*. Ed. by Lawrence E. Jones. Second Edition. Academic Press, Boston, 2017, pp. 85–101. DOI: 10.1016/B978-0-12-809592-8.00007-X.

[104] William Knowles et al. “A survey of cyber security management in industrial control systems”. In: *International Journal of Critical Infrastructure Protection* 9 (2015), pp. 52–80. DOI: 10.1016/j.ijcip.2015.02.002.

[105] Woo-hyun Ko, Bharadwaj Satchidanandan, and P R Kumar. “Dynamic Watermarking-based Defense of Transportation Cyber-physical Systems”. In: *ACM Transactions on Cyber-Physical Systems* 4.1 (2019). DOI: 10.1145/3361700.

[106] Woo-Hyun Ko, Bharadwaj Satchidanandan, and P.R. Kumar. “Theory and Implementation of Dynamic Watermarking for Cybersecurity of Advanced”. In: *IEEE Conference on Communications and Network Security (CNS): International Workshop on Cyber-Physical Systems Security (CPS-Sec) Theory* (2016). DOI: 10.1109/CNS.2016.7860529.

[107] Charalambos Konstantinou, Anastasis Keliris, and Michail Maniatakos. “Taxonomy of firmware trojans in smart grid devices”. In: *IEEE Power and Energy Society General Meeting (PESGM)* (2016), pp. 1–5. DOI: 10.1109/PESGM.2016.7741452.

[108] Efstathios Kontouras, Anthony Tzes, and Leonidas Dritsas. “Impact Analysis of a Bias Injection Cyber-Attack on a Power Plant”. In: *IFAC-PapersOnLine* 50.1 (2017), pp. 11094–11099. DOI: 10.1016/j.ifacol.2017.08.2493.

[109] Efstathios Kontouras, Anthony Tzes, and Leonidas Dritsas. “Set-theoretic detection of data corruption attacks on cyber physical power systems”. In: *Journal of Modern Power Systems and Clean Energy* 6.5 (2018), pp. 872–886. DOI: 10.1007/s40565-018-0452-y.

[110] Georgia Koutsandria et al. “A real-time testbed environment for cyber-physical security on the power grid”. In: *Proceedings of the 1st ACM Workshop on Cyber-Physical Systems-Security and/or Privacy (CPS-SPC), co-located with CCS* (2015), pp. 67–78. DOI: 10.1145/2808705.2808707.

[111] Siwar Kriaa et al. “A survey of approaches combining safety and security for industrial control systems”. In: *Reliability Engineering and System Safety* 139 (2015), pp. 156–178. DOI: 10.1016/j.ress.2015.02.008.

[112] Maryna Krotofil and Dieter Gollmann. “Industrial control systems security: What is happening?” In: *IEEE International Conference on Industrial Informatics (INDIN)* (2013), pp. 670–675. DOI: 10.1109/INDIN.2013.6622964.

[113] Prabha Kundur, Neal J Balu, and Mark G Lauby. *Power system stability and control*. Vol. 7. McGraw-hill New York, 1994. ISBN: 0-07-035958-x.

[114] Mehmet Necip Kurt, Yasin Yilmaz, and Xiaodong Wang. “Distributed quickest detection of cyber-attacks in smart grid”. In: *IEEE Transactions on Information Forensics and Security* 13.8 (2018), pp. 2015–2030. DOI: 10.1109/TIFS.2018.2800908.

[115] Ralph Langner. “Stuxnet: Dissecting a cyberwarfare weapon”. In: *IEEE Security & Privacy* 9.3 (2011), pp. 49–51. DOI: 10.1109/MSP.2011.67.

[116] David Law et al. “Evolution of Ethernet standards in the IEEE 802.3 working group”. In: *IEEE Communications Magazine* 51.8 (2013), pp. 88–96. DOI: 10.1109/MCOM.2013.6576344.

[117] Robert M. Lee, Michael J. Assante, and Tim Conway. *Analysis of the cyber attack on the Ukrainian power grid*. Tech. rep. Electricity Information Sharing and Analysis Center (E-ISAC), 2016.

[118] Na Li, Changhong Zhao, and Lijun Chen. “Connecting automatic generation control and economic dispatch from an optimization view”. In: *IEEE Transactions on Control of Network Systems* 3.3 (2016), pp. 254–264. DOI: 10.1109/TCNS.2015.2459451.

[119] Xu Li et al. “Securing smart grid: cyber attacks, countermeasures, and challenges”. In: *IEEE Communications Magazine* 50.8 (2012), pp. 38–45. DOI: 10.1109/MCOM.2012.6257525.

[120] Gaoqi Liang et al. “The 2015 ukraine blackout: Implications for false data injection attacks”. In: *IEEE Transactions on Power Systems* 32.4 (2016), pp. 3317–3318. DOI: 10.1109/TPWRS.2016.2631891.

[121] Chih Ta Lin, Sung Lin Wu, and Mei Lin Lee. “Cyber attack and defense on industry control systems”. In: *IEEE Conference on Dependable and Secure Computing* (2017), pp. 524–526. DOI: 10.1109/DESEC.2017.8073874.

[122] Hanxiao Liu, Yilin Mo, and Karl Henrik Johansson. “Active Detection Against Replay Attack: A Survey on Watermark Design for Cyber-Physical Systems”. In: *Safety, Security and Privacy for Cyber-Physical Systems*. Spinger, Berlin, 2021, pp. 145–171. DOI: 10.1007/978-3-030-65048-3_8.

[123] Hanxiao Liu et al. “An On-line Design of Physical Watermarks”. In: *IEEE Conference on Decision and Control (CDC)* (2018), pp. 440–445. DOI: 10.1109/CDC.2018.8619632.

[124] Yao Liu, Peng Ning, and Michael K Reiter. “False data injection attacks against state estimation in electric power grids”. In: *ACM Transactions on Information and System Security (TISSEC)* 14.1 (2011), pp. 1–33. DOI: 10.1145/1952982.1952995.

[125] Yuriy Zacchia Lun et al. “Cyber-physical systems security: An automatic control perspective”. In: *Journal of Systems and Software* 149 (2019), pp. 174–216. DOI: 10.1016/j.jss.2018.12.006.

[126] Tyson Macaulay and Bryan L Singer. *Cybersecurity for industrial control systems: SCADA, DCS, PLC, HMI, and SIS*. CRC Press, Boca Raton, 2011. ISBN: 978-1-4398-0196-3.

[127] Anzar Mahmood, Nadeem Javaid, and Sohail Razzaq. “A review of wireless communications for smart grid”. In: 41 (2015), pp. 248–260. DOI: 10.1016/j.rser.2014.08.036.

[128] Magdi S. Mahmoud, Mutaz M. Hamdan, and Uthman A. Baroudi. “Modeling and control of Cyber-Physical Systems subject to cyber attacks: A survey of recent advances and challenges”. In: *Neurocomputing* 338 (2019), pp. 101–115. DOI: 10.1016/j.neucom.2019.01.099.

[129] Kebina Manandhar et al. “Detection of faults and attacks including false data injection attack in smart grid using Kalman filter”. In: *IEEE transactions on control of network systems* 1.4 (2014), pp. 370–379. DOI: 10.1109/TCNS.2014.2357531.

[130] Isaak D Mayergoyz and Wes Lawson. *Basic electric circuit theory: a one-semester text*. Gulf Professional Publishing, 1997. ISBN: 978-0-12-480865-2.

[131] Stephen McLaughlin et al. “The Cybersecurity Landscape in Industrial Control Systems”. In: *Proceedings of the IEEE* 104.5 (2016), pp. 1039–1057. DOI: 10.1109/JPROC.2015.2512235.

[132] Anthony R Metke and Randy L Ekl. “Security technology for smart grid networks”. In: *IEEE Transactions on Smart Grid* 1.1 (2010), pp. 99–107. DOI: 10.1109/TSG.2010.2046347.

[133] Bruce Middleton. *A history of cyber security attacks: 1980 to present*. CRC Press, Boca Raton, 2017. DOI: 10.1201/9781315155852.

[134] Cristea Lavinia Mihaela. “Current security threats in the national and international context”. In: *Journal of Accounting and Management Information Systems* 19.2 (2020), pp. 351–378. DOI: 10.24818/jamis.2020.02007.

[135] Thomas Miller et al. “Looking back to look forward: Lessons learnt from cyber-attacks on Industrial Control Systems”. In: *International Journal of Critical Infrastructure Protection* 35 (2021), pp. 1–14. DOI: 10.1016/j.ijcip.2021.100464.

[136] Yilin Mo, Rohan Chabukswar, and Bruno Sinopoli. “Detecting integrity attacks on SCADA systems”. In: *IEEE Transactions on Control Systems Technology* 22.4 (2014), pp. 1396–1407. DOI: 10.1109/TCST.2013.2280899.

[137] Yilin Mo and Bruno Sinopoli. “False data injection attacks in control systems”. In: *Preprints of the 1st workshop on Secure Control Systems*. 2010, pp. 1–6.

[138] Yilin Mo and Bruno Sinopoli. “Secure control against replay attacks”. In: *47th Annual Allerton Conference on Communication, Control, and Computing, Allerton* (2009), pp. 911–918. DOI: 10.1109/ALLERTON.2009.5394956.

[139] Yilin Mo, Sean Weerakkody, and Bruno Sinopoli. “To Detect Counterfeit Sensor Outputs”. In: *IEEE Control Systems* 35.1 (2015), pp. 93–109. DOI: 10.1109/MCS.2014.2364724.

[140] Erfan Mohagheghi et al. “A Survey of Real-Time Optimal Power Flow”. In: *Energies* 11.11 (2018), pp. 1–20. DOI: 10.3390/en11113142.

[141] Athira M. Mohan, Nader Meskin, and Hasan Mehrjerdi. “A comprehensive review of the cyber-attacks and cyber-security on load frequency control of power systems”. In: *Energies* 13.15 (2020), pp. 1–33. DOI: 10.3390/en13153860.

[142] Thomas Morris, Rayford Vaughn, and Yoginder Dandass. “A retrofit network intrusion detection system for MODBUS RTU and ASCII industrial control systems”. In: *Proceedings of the Annual Hawaii International Conference on System Sciences* (2012), pp. 2338–2345. DOI: 10.1109/HICSS.2012.78.

[143] Thomas H Morris, Zach Thornton, and Ian Turnipseed. “Industrial Control System Simulation and Data Logging for Intrusion Detection System Research”. In: *Seventh Annual Southeastern Cyber Security Summit* (2015), pp. 1–6.

[144] Thomas H. Morris and Wei Gao. “Industrial Control System Cyber Attacks”. In: *International Symposium for ICS & SCADA Cyber Security Research* (2013), pp. 22–29. DOI: 10.14236/ewic/icscsr2013.3.

[145] John Moteff and Paul Parfomak. *Critical infrastructure and key assets: definition and identification*. Tech. rep. Library of Congress Washington DC Congressional Research Service, 2004.

[146] Zakaria El Mrabet et al. “Cyber-security in smart grid: Survey and challenges”. In: *Computers and Electrical Engineering* 67 (2018), pp. 469–482. DOI: 10.1016/j.compeleceng.2018.01.015.

[147] Devaprakash Muniraj and Mazen Farhood. “Detection and mitigation of actuator attacks on small unmanned aircraft systems”. In: *Control Engineering Practice* 83 (2019), pp. 188–202. DOI: 10.1016/j.conengprac.2018.10.022.

[148] Ahmed S. Musleh, Guo Chen, and Zhao Yang Dong. “A Survey on the Detection Algorithms for False Data Injection Attacks in Smart Grids”. In: *IEEE Transactions on Smart Grid* 11.3 (2020), pp. 2218–2234. DOI: 10.1109/TSG.2019.2949998.

[149] Arunava Naha et al. “Deception Attack Detection using Reduced Watermarking”. In: *2021 European Control Conference (ECC)* (2021), pp. 1–7. DOI: 10.23919/ECC54610.2021.9654843.

[150] Arunava Naha et al. “Sequential detection of Replay attacks”. In: *IEEE Transactions on Automatic Control* (2022). DOI: 10.1109/TAC.2022.3174004.

[151] Sajid Nazir, Shushma Patel, and Dilip Patel. “Assessing and augmenting SCADA cyber security: A survey of techniques”. In: *Computers and Security* 70 (2017), pp. 436–454. DOI: 10.1016/j.cose.2017.06.010.

[152] Ijeoma Onyeji, Morgan Bazilian, and Chris Bronk. “Cyber security and critical energy infrastructure”. In: *Electricity Journal* 27.2 (2014), pp. 52–60. DOI: 10.1016/j.tej.2014.01.011.

[153] Sophocles J Orfanidis. *Introduction to signal processing*. Prentice-Hall, New Jersey, 1995. ISBN: 0-13-209172-0.

[154] Ettercap Project Organisation. *Ettercap Home Page*. URL: https://www.ettercap-project.org/ (visited on 06/28/2022).

[155] Arman Oshnoei et al. “On the Contribution of Wind Farms in Automatic Generation Control: Review and New Control Approach”. In: *Applied Sciences* 8.10 (2018), pp. 1–23. DOI: 10.3390/app8101848.

[156] Yao Pan et al. “Taxonomies for Reasoning About Cyber-physical Attacks in IoT-based Manufacturing Systems”. In: *International Journal of Interactive Multimedia and Artificial Intelligence* 4.3 (2017), pp. 45–54. DOI: 10.9781/ijimai.2017.437.

[157] Fabio Pasqualetti, Florian Dörfler, and Francesco Bullo. “Attack Detection and Identification in Cyber-Physical Systems”. In: *IEEE Transactions on Automatic Control* 58.11 (2013), pp. 2715–2729. DOI: 10.1109/TAC.2013.2266831.

[158] M-Elisabeth Paté-Cornell et al. “Cyber risk management for critical infrastructure: a risk analysis model and three case studies”. In: *Risk Analysis* 38.2 (2018), pp. 226–241. DOI: 10.1111/risa.12844.

[159] Richard Piggin. “Cyber security trends: What should keep CEOs awake at night”. In: *International Journal of Critical Infrastructure Protection* 13 (2016), pp. 36–38. DOI: 10.1016/j.ijcip.2016.02.001.

[160] Matthew Porter et al. “Detecting deception attacks on autonomous vehicles via linear time-varying dynamic watermarking”. In: *4th IEEE Conference on Control Technology and Applications (CCTA)* (2020), pp. 821–826. DOI: 10.1109/CCTA41146.2020.9206278.

[161] Matthew Porter et al. “Detecting Generalized Replay Attacks via Time-Varying Dynamic Watermarking”. In: *IEEE Transactions on Automatic Control* 66.8 (2020), pp. 3502–3517. DOI: 10.1109/tac.2020.3022756.

[162] Matthew Porter et al. “Resilient Control of Platooning Networked Robotic Systems via Dynamic Watermarking”. In: *arXiv preprint* (2021), pp. 1–19. DOI: 10.48550/arXiv.2106.07541.

[163] Maneli Malek Pour, Arash Anzalchi, and Arif Sarwat. “A review on cyber security issues and mitigation methods in smart grid systems”. In: *SoutheastCon 2017* (2017), pp. 1–4. DOI: 10.1109/secon.2017.7925278.

[164] James Powell and P Eng. *Profibus and Modbus : a comparison*. Tech. rep. Siemens AG, 2013.

[165] *Power System Dynamic Tutorial*. Tech. rep. Electric Power Research Institution, 2009.

[166] Qais Qassim et al. “A Survey of SCADA Testbed Implementation Approaches”. In: *Indian Journal of Science and Technology* 10.26 (2017), pp. 1–8. DOI: 10.17485/ijst/2017/v10i26/116775.

[167] Jorge Ramos-ruiz et al. “An Active Detection Scheme for Cyber Attacks on Grid-tied PV Systems”. In: *IEEE CyberPELS* (2020), pp. 1–6. DOI: 10.1109/CyberPELS49534.2020.9311539.

[168] Vedang Suhas Ranade. “A laboratory for cyber-attack generation and testing in Industrial Control Systems: Design and Simulation”. unpublished. 2021. URL: http://resolver.tudelft.nl/uuid:ad554d68-4503-4544-b51b-e48379fc7216.

[169] RealPars. *Electrical Grounding Explained | Basic Concepts*. 2021. URL: https://www.youtube.com/watch?v=YO-Dnk6ZKrI (visited on 07/09/2022).

[170] Steven M Rinaldi, James P Peerenboom, and Terrence K Kelly. “Identifying, understanding, and analyzing critical infrastructure interdependencies”. In: *IEEE control systems magazine* 21.6 (2001), pp. 11–25. DOI: 10.1109/37.969131.

[171] Julian Rrushi. “Composite intrusion detection in process control networks”. In: (2009), pp. 1–205. DOI: 10.13130/rrushi-julian_phd2009-01.

[172] Juan Enrique Rubio et al. “Current cyber-defense trends in industrial control systems”. In: *Computers and Security* 87 (2019), pp. 1–12. DOI: 10.1016/j.cose.2019.06.015.

[173] Jose Rubio-Hernan, Luca De Cicco, and Joaquin Garcia-Alfaro. “On the use of watermark-based schemes to detect cyber-physical attacks”. In: *Eurasip Journal on Information Security* 8 (2017). DOI: 10.1186/s13635-017-0060-9.

[174] Hadi Saadat. *Power system analysis*. McGraw-Hill, New York, 1999. ISBN: 0-07-116758-7.

[175] Tomonori Sadamoto et al. “Dynamic Modeling , Stability , and Control of Power Systems with Distributed Energy Resources”. In: *IEEE Control Systems Magazine* 39.2 (2018), pp. 34–65. DOI: 10.1109/MCS.2018.2888680.

[176] Rasoul Sadeghi. *Communicating between Beckhoff controllers part 1: EAP*. 2020. URL: https://www.plccoder.com/communicating-between-beckhoff-controllers-via-eap/ (visited on 06/22/2022).

[177] Mohammad Ashraf Hossain Sadi et al. “OPNET/simulink based testbed for disturbance detection in the smart grid”. In: *ACM International Conference Proceeding Series* 17 (2015), pp. 1–4. DOI: 10.1145/2746266.2746283.

[178] Jakob Sagatowski. *PLC programming using TwinCAT 3*. 2022. URL: https://www.youtube.com/playlist?list=PLimaF0nZKYHz3I3kFP4myaAYjmYk1SowO (visited on 07/09/2022).

[179] Helem S. Sánchez et al. “Bibliographical review on cyber attacks from a control oriented perspective”. In: *Annual Reviews in Control* 48 (2019), pp. 103–128. DOI: 10.1016/j.arcontrol.2019.08.002.

[180] Bharadwaj Satchidanandan and P. R. Kumar. “Dynamic watermarking: Active defense of networked cyber-physical systems”. In: *Proceedings of the IEEE* 105.2 (2017), pp. 219–240. DOI: 10.1109/JPROC.2016.2575064.

[181] Bharadwaj Satchidanandan and P. R. Kumar. “On Minimal Tests of Sensor Veracity for Dynamic Systems”. In: *International Conference on Communication Systems and Networks (COMSNETS)* (2017), pp. 23–30. DOI: 10.1109/COMSNETS.2017.7945354.

[182] Offensive Security. *Kali inside VirtualBox (Guest VM)*. URL: https://www.kali.org/docs/virtualization/install-virtualbox-guest-vm/ (visited on 06/27/2022).

[183] Offensive Security. *Kali Linux Installation Guide*. URL: https://www.kali.org/docs/installation/hard-disk-install/ (visited on 06/27/2022).

[184] Offensive Security. *tcpreplay tool*. URL: https://www.kali.org/tools/tcpreplay/ (visited on 06/28/2022).

[185] Roberto Setola et al. “An overview of Cyber Attack to Industrial Control System”. In: *Chemical Engineering Transactions* 77 (2019), pp. 907–912. DOI: 10.3303/CET1977152.

[186] Yubin Shen, Minrui Fei, and Dajun Du. “Cyber security study for power systems under denial of service attacks”. In: *Transactions of the Institute of Measurement and Control* 41.6 (2019), pp. 1600–1614. DOI: 10.1177/0142331217709528.

[187] Paulo Simões et al. “On the use of honeypots for detecting cyber attacks on industrial control networks”. In: *12th European Conference on Information Warfare and Security (ECIW 2013)* (2013), pp. 263–269.

[188] Jill Slay and Michael Miller. “Lessons learned from the maroochy water breach”. In: *International conference on critical infrastructure protection* (2007), pp. 73–82. DOI: 10.1007/978-0-387-75462-8_6.

[189] Eduardo D Sontag. *Mathematical control theory: deterministic finite dimensional systems*. Vol. 6. Springer Science & Business Media, 2013. DOI: 10.1007/978-1-4612-0577-7.

[190] SquishyBrained. *Learning PLCs with Structured Text*. 2015. URL: https://www.youtube.com/playlist?list=PLE1CU6EebvTCJCMIUOSWgMseMaW-2k5zH (visited on 07/09/2022).

[191] Siddharth Sridhar and Manimaran Govindarasu. “Model-based attack detection and mitigation for automatic generation control”. In: *IEEE Transactions on Smart Grid* 5.2 (2014), pp. 580–591. DOI: 10.1109/TSG.2014.2298195.

[192] Siddharth Sridhar, Adam Hahn, and Manimaran Govindarasu. “Cyber-physical system security for the electric power grid”. In: *Proceedings of the IEEE* 100.1 (2012), pp. 210–224. DOI: 10.1109/JPROC.2011.2165269.

[193] Keith Stouffer, Joe Falco, and Karen Scarfone. *Guide to industrial control systems (ICS) security*. Tech. rep. National Institution for Standards and Technology (NIST), 2011, pp. 11–158.

[194] Daniel Sullivan, Eric Luiijf, and Edward JM Colbert. “Components of industrial control systems”. In: *Cyber-security of SCADA and other industrial control systems*. Spinger, Berlin, 2016, pp. 15–28. DOI: 10.1007/978-3-319-32125-7_2.

[195] Chih Che Sun, Adam Hahn, and Chen Ching Liu. “Cyber security of a power grid: State-of-the-art”. In: *International Journal of Electrical Power & Energy Systems* 99 (2018), pp. 45–56. DOI: 10.1016/j.ijepes.2017.12.020.

[196] RSP Supply. *RSP Supply Education Channel*. 2022. URL: https://www.youtube.com/c/RSPSupply (visited on 07/09/2022).

[197] Rui Tan et al. “Modeling and Mitigating Impact of False Data Injection Attacks on Automatic Generation Control”. In: *IEEE Transactions on Information Forensics and Security* 12.7 (2017), pp. 1609–1624. DOI: 10.1109/TIFS.2017.2676721.

[198] Sen Tan et al. “Brief Survey on Attack Detection Methods for Cyber-Physical Systems”. In: *IEEE Systems Journal* 14.4 (2020), pp. 5329–5339. DOI: 10.1109/JSYST.2020.2991258.

[199] James M. Taylor and Hamid R. Sharif. “Security challenges and methods for protecting critical infrastructure cyber-physical systems”. In: *International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT)* (2017), pp. 1–6. DOI: 10.1109/MoWNet.2017.8045959.

[200] André M.H. Teixeira and Riccardo M.G. Ferrari. “Detection of Sensor Data Injection Attacks with Multiplicative Watermarking”. In: *European Control Conference (ECC)* (2018), pp. 338–343. DOI: 10.23919/ECC.2018.8550114.

[201] Chee Wooi Ten, Govindarasu Manimaran, and Chen Ching Liu. “Cybersecurity for critical infrastructures: Attack and defense modeling”. In: *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans* 40.4 (2010), pp. 853–865. DOI: 10.1109/TSMCA.2010.2048028.

[202] Eleftherios Tsampasis et al. “Novel simulation approaches for smart grids”. In: *Journal of Sensor and Actuator Networks* 5.3 (2016), pp. 1–22. DOI: 10.3390/jsan5030011.

[203] Kaleem Ullah et al. “Automatic generation control strategies in conventional and modern power systems: A comprehensive overview”. In: *Energies* 14.9 (2021), pp. 1–43. DOI: 10.3390/en14092376.

[204] David I Urbina et al. *Survey and new directions for physics-based attack detection in control systems*. Tech. rep. 2016. DOI: 10.6028/nist.gcr.16-010.

[205] Vincent Urias, Brian Van Leeuwen, and Bryan Richardson. “Supervisory Command and Data Acquisition (SCADA) system cyber security analysis using a live, virtual, and constructive (LVC) testbed”. In: *Proceedings - IEEE Military Communications Conference MILCOM* (2012), pp. 1–8. DOI: 10.1109/MILCOM.2012.6415818.

[206] Erik Van der Vleuten and Vincent Lagendijk. “Transnational infrastructure vulnerability: The historical shaping of the 2006 European “Blackout””. In: *Energy Policy* 38.4 (2010), pp. 2042–2052. DOI: 10.1016/j.enpol.2009.11.047.

[207] Steven Walker-Roberts, Mohammad Hammoudeh, and Ali Dehghantanha. “A systematic review of the availability and efficacy of countermeasures to internal threats in healthcare critical infrastructure”. In: *IEEE Access* 6 (2018), pp. 25167–25177. DOI: 10.1109/ACCESS.2018.2817560.

[208] Sean Weerakkody, Yilin Mo, and Bruno Sinopoli. “Detecting Integrity Attacks on Control Systems using Robust Physical Watermarking”. In: *IEEE Conference on Decision and Control* (2014), pp. 3757–3764. DOI: 10.1109/CDC.2014.7039974.

[209] Sean Weerakkody, Omur Ozel, and Bruno Sinopoli. “A Bernoulli-Gaussian Physical Watermark for Detecting Integrity Attacks in Control Systems”. In: *Annual Allerton Conference* 5 (2017), pp. 966–973. DOI: 10.1109/ALLERTON.2017.8262842.

[210] Joseph Weiss. *Protecting industrial control systems from electronic threats*. Momentum Press, 2010. ISBN: 978-1-60650-199-3.

[211] Guangyu Wu, Jian Sun, and Jie Chen. “A survey on the security of cyber-physical systems”. In: *Control Theory and Technology* 14.1 (2016), pp. 2–10. DOI: 10.1007/s11768-016-5123-9.

[212] Yingmeng Xiang, Lingfeng Wang, and Yichi Zhang. “Adequacy evaluation of electric power grids considering substation cyber vulnerabilities”. In: *International Journal of Electrical Power and Energy Systems* 96 (2018), pp. 368–379. DOI: 10.1016/j.ijepes.2017.10.004.

[213] Yikai Xu et al. “Review on Cyber Vulnerabilities of Communication Protocols in Industrial Control Systems”. In: *IEEE Conference on Energy Internet and Energy System Integration (EI2)* (2017), pp. 1–6. DOI: 10.1109/EI2.2017.8245509.

[214] Jean Paul A. Yaacoub et al. “Cyber-physical systems security: Limitations, issues and future trends”. In: *Microprocessors and Microsystems* 77 (2020), pp. 1–33. DOI: 10.1016/j.micpro.2020.103201.

[215] Bahram Yaghooti, Raffaele Romagnoli, and Bruno Sinopoli. “Physical Watermarking for Replay Attack Detection in Continuous-time Systems”. In: *European Control Conference (ECC)* (2021), pp. 1406–1411. DOI: 10.1016/j.ejcon.2021.06.012.

[216] Samuel Yankson and Mahdi Ghamkhari. “Transactive energy to thwart load altering attacks on power distribution systems”. In: *Future Internet* 12.1 (2020), pp. 1–14. DOI: 10.3390/fi12010004.

[217] Dan Zhang et al. “A survey on attack detection, estimation and control of industrial cyber–physical systems”. In: *ISA Transactions* (2021), pp. 1–16. DOI: 10.1016/j.isatra.2021.01.036.

[218] Jingfan Zhang et al. “A Real-Time Dynamic Watermarking Detection Method of Networked Inverted Pendulum Servo Systems”. In: *Communications in Computer and Information Science* 1303 (2020), pp. 250–263. DOI: 10.1007/978-981-33-6378-6_19.

[219] Yingchen Zhang et al. “Wide-area frequency monitoring network (FNET) architecture and applications”. In: *IEEE Transactions on smart grid* 1.2 (2010), pp. 159–167. DOI: `10.1109/TSG.2010.2050345`.

[220] K. Zhou, J.C. Doyle, and K. Glover. *Robust and optimal control*. Prentice Hall, New Jersey, 1996. DOI: `10.1016/S0005-1098(97)00132-5`.

[221] Bonnie Zhu, Anthony Joseph, and Shankar Sastry. “A taxonomy of cyber attacks on SCADA systems”. In: *IEEE International Conferences on Internet of Things and Cyber, Physical and Social Computing, (iThings/CPSCom)* (2011), pp. 380–388. DOI: `10.1109/iThings/CPSCom.2011.34`.

A

Considered Watermarking Studies

This appendix provides a list of all studies on watermarking which were considered in this study. The amount of citations is based on data from Google Scholar at the time of collection (during the period of April 2021 to May 2022). They are arranged based on their reference entry.

Ref.	Year	Cite	Title
[3]	2020	2	Set-Theoretic Control for Active Detection of Replay Attacks with Applications to Smart Grid
[49]	2017	18	Cost-Effective Watermark Based Detector for Replay Attacks on Cyber-Physical Systems
[52]	2017	31	Detection and Isolation of Replay Attacks through Sensor Watermarking
[53]	2017	17	Detection and Isolation of Routing Attacks through Sensor Watermarking
[51]	2020	1	A Switching Multiplicative Watermarking Scheme for Detection of Stealthy Cyber-Attacks
[56]	2018	11	Distributed Watermarking for Secure Control of Microgrids under Replay Attacks
[61]	2021	4	A Blended Active Detection Strategy for False Data Injection Attacks in Cyber-Physical Systems
[101]	2017	18	A New Watermarking Approach for Replay Attack Detection in LQG Systems
[102]	2017	22	Replay attack detection in a multi agent system using stability analysis and loss effective watermarking
[105]	2019	6	Dynamic Watermarking-based Defense of Transportation Cyber-physical Systems
[106]	2016	27	Theory and Implementation of Dynamic Watermarking for Cybersecurity of Advanced Transportation Systems
[122]	2021	0	Active Detection Against Replay Attack: A Survey on Watermark Design for Cyber-Physical Systems
[123]	2018	8	An On-line Design of Physical Watermarks
[136]	2014	424	Detecting Integrity Attacks on SCADA Systems
[138]	2009	692	Secure Control Against Replay Attacks
[139]	2015	313	Physical Authentication of Control Systems: Designing watermarked control inputs to detect counterfeit sensor outputs
[147]	2019	16	Detection and mitigation of actuator attacks on small unmanned aircraft systems
[149]	2021	0	Deception Attack Detection using Reduced Watermarking
[150]	2020	1	Sequential detection of Replay attacks
[160]	2020	6	Detecting Deception Attacks on Autonomous Vehicles via Linear Time-Varying Dynamic Watermarking
[162]	2021	0	Resilient Control of Platooning Networked Robotic Systems via Dynamic Watermarking
[161]	2020	7	Detecting Generalized Replay Attacks via Time-Varying Dynamic Watermarking
[167]	2020	2	An Active Detection Scheme for Cyber Attacks on Grid-tied PV Systems
[173]	2017	22	On the Use of Watermark-Based Schemes to Detect Cyber-Physical Attacks
[180]	2017	122	Dynamic Watermarking: Active Defense of Networked Cyber–Physical Systems
[181]	2017	22	On Minimal Tests of Sensor Veracity for Dynamic Watermarking-Based Defense of Cyber-Physical Systems
[200]	2018	6	Detection of Sensor Data Injection Attacks with Multiplicative Watermarking
[208]	2014	85	Detecting Integrity Attacks on Control Systems using Robust Physical Watermarking
[209]	2017	16	A Bernoulli-Gaussian Physical Watermark for Detecting Integrity Attacks in Control Systems
[215]	2021	0	Physical Watermarking for Replay Attack Detection in Continuous-time Systems
[218]	2020	0	A Real-Time Dynamic Watermarking Detection Method of Networked Inverted Pendulum Servo Systems

Table A.1: Overview of considered watermarking papers

B

Ordered Testbed Parts

This appendix lists the ordered electrical, IT-related and mounting parts for the HILDA testbed. This excludes the `dSPACE` simulator, `Beckhoff` equipment, management PC and Ethernet switch, which were already in-house. The ordering was done at `RS Components`, a standard supplier to the `TU Delft`. This made it possible to relatively quickly order the items. `RS Components` has a large stock of useful parts for ICSs. The specific items which were ordered by the author can be found in Table B.1 and in the `Microsoft Teams` group of the Computer Wall at the Systems and Control lab. Table B.1 also mentions an extended setup, which is discussed in Appendix C.

		Amount of parts		
Product	Fabrication #	Single setup	Extended setup	Available in lab
Power cord single-phase black 2m	490-239	1	5	2
PE wire grean/yellow 1mm2 100m	180-5934	1	1	1
DC 24V 10A red 1mm2 100m	180-5938	1	1	1
DC 0V 10A black 1mm2 100m	180-5936	1	1	1
Terminal blocks PE max. 2.5mm2	3044092	6	30	10
Terminal blocks 24V max. 2.5mm2	3045062	4	20	10
Terminal blocks 0V max. 2.5mm2	3045088	4	20	10
Jumpers 4	3030190	4	20	10
Cover terminals	3047028	5	25	5
Ferrules 1mm2 8mm red	458-702	50	250	100
Cat6 5m orange	N6PATC5MOR	0	4	0
Cat6 3m orange	N6PATC3MOR	0	4	0
Cat6 1m orange	N6PATC1MOR	2	2	3
Cat6 0.5m orange	N6PATC50CMOR	2	10	3
Signal cable 8 core 0.35mm2 30m	1178C SL005	1	1	1
Ferrules 0.34mm2 8mm	157-1222	100	100	100
Network cards PCs	ST1000SPEX2	1	5	1
ZIF connector kit	MABXII_C	1	2	0
DIN rail 0.5m 35/7.5mm	467-406	2	10	2
Wiring duct 1m 40x25mm (pack of 4)	10430022-4x1m	1	5	4
M4x16mm Bolts (bag of 100)	553-431	1	1	1
M4x12mm Bolts (bag of 100)	553-425	1	1	1
M6x12mm Bolts (bag of 100)	553-504	1	1	1
M6 Nut (bag of 100)	201-0852	1	1	1
T-slot nut 8 Zn M4 bright zinc-plated	0.0.373.58	8	40	10

Table B.1: Overview of ordered parts for the construction of the testbed

C

Extended Testbed Setup

This appendix provides context on the extended testbed setup. Eventually, the extended testbed is designed to include five setups similar to the HILDA testbed. The combination of these setups is visualised in Figure C.1. Instead of only two PLCs and a single management PC from the HILDA setup, the entire testbed includes a total of ten PLCs and five management PCs. All PLCs are connected to the simulator and the main network switch, while all management PCs are connected only to the main network switch. The `dSPACE` simulator and attack PC remain singular. The extended setup also contains a second network switch connecting the (simulated) attack PC to the main network switch. This second network switch can also be simulated by a VM. The combination of multiple setups could be used to for example simulate car platooning or a larger scale power grid.

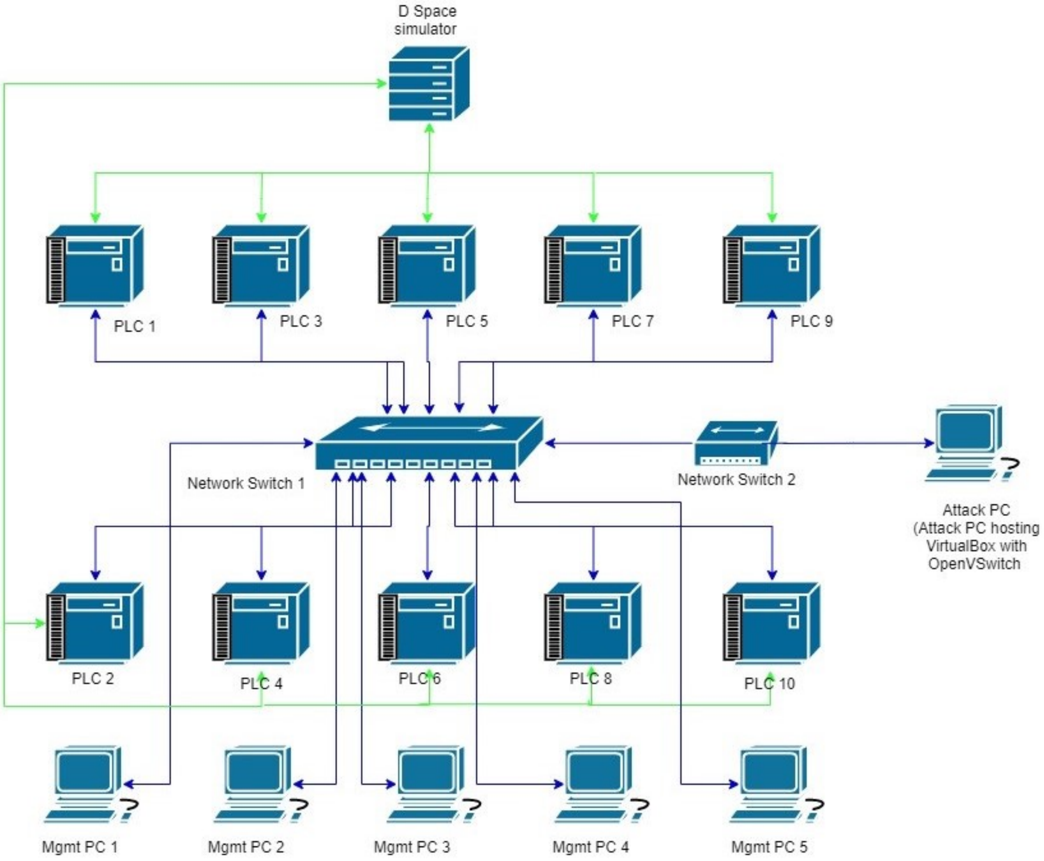


Figure C.1: Setup of the extended testbed [168]

D

Useful Tutorials for Testbed Design, Construction and Configuration

This appendix lists multiple useful tutorials for the design, construction and configuration of an ICS testbed (or ICSs in general) in Table D.1. These were consulted because the development of the HILDA testbed was a very practical undertaking, and they are listed here because they might be of use to future users of the HILDA testbed. Table D.1 is ordered based on the chronological sequence in which the tutorials were consulted.

Ref.	Distributor	Description
[169]	`RealPars`	Basic concepts of electrical grounding
[196]	`RSP Supply`	Practical videos on ICS construction
[83]	`Hayes`	Third party tutorial on configuration of `TwinCAT` I/O modules
[190]	`SquishyBrained`	Brief tutorial on programming PLCs using `TwinCAT 3`
[178]	`Jakob Sagatowski`	Extended tutorial on programming PLCs using `TwinCAT 3`
[176]	`PLC Coder`	Third party tutorial on EAP
[38]	`PLC Coder`	Third party tutorial on ADS
[182]	`Offensive Security`	How to get `Kali Linux` on `VirtualBox` (Guest VM)
[28]	`David Bombal`	TCP/IP deep-dive using *Wireshark*

Table D.1: Overview of the tutorials consulted for the construction of the testbed

E

Simulation Codes

This appendix provides the codes used to generate all results presented in this study. This includes both the `MATLAB & Simulink` and the HILDA testbed simulations. For the `MATLAB & Simulink` simulations, a main `MATLAB` code is run, which calls on multiple functions and the main `Simulink` model. Secondly, there is a separate `MATLAB` code dedicated to the generation of all results, which runs the `Simulink` model multiple times with different parameter settings and calls on certain `csv`- and `mat`-files. Then, each PLC has its own main programme, written in ST, which calls on multiple function blocks. An overview of the codes is provided in Table E.1, while the codes themselves can be found in the `GitHub` repository of the author.

Platform	Language	Main function of the code
`MATLAB`	MATLAB	Main code for desktop simulation
`MATLAB`	MATLAB	Code for desktop simulation analysis
PLC 1	ST	Filters $\mathcal{Q}$ and $\mathcal{G}$, cyber-attack on control inputs, AGC algorithm, and IDS
PLC 2	ST	Filters $\mathcal{W}$ and $\mathcal{H}$, and cyber-attack on measurements

Table E.1: Central codes of this study

F

Testbed Verification Experiments

Power wiring: To check whether the power wiring has been done correctly, analyse all lights on the testbed appliances. These are the *DC OK* LED on the Power Supply, the *Power Status* LED on the MAB II, the *Us 24V* and *Up 24V* LEDs on the EK1100, and finally the *PWR* LED on the IPCs.

EL3004 (input) I/O modules power connectivity: After establishing an Ethernet connection of the EL3004 I/O modules to one of the PLCs (see [65] for the setup manual), connect a power supply (with arbitrary voltage output) to a specific input port and the accompanying *GND* port. Generating a voltage should result in accessory values which can be analysed in `TwinCAT`.

EL4004 (output) I/O modules power connectivity: After establishing an Ethernet connection of the EL4004 I/O modules to one of the PLCs (see [65] for the setup manual), write values on the PLC to a specific output port of the EL4004. Connecting a voltmeter to the port and its accompanying *0V* port should result in accessory measured values.

PLC programme on `TwinCAT` combined with EL3004 and EL4004: Create a new (standard) PLC project. Create new I/O variables in either the Main programme or a Global Variables list (`I/Q` for input/output, `*` for unappointed variables, and `INT` for signed 16 bit values, e.g. `AnIn11 AT \%I*: INT;` and `AnOut1 AT \%Q*: INT;`). Activate the configuration, after which they should appear under *PLC Instance*. Under *Devices*, link each I/O to the PLC project's I/O. Again active the configuration, go to run mode, then go online (green icon) and hit play. All variables can now be checked online in the PLC project. Logout of online mode. Under *POU* (main script), write a simple script to verify if the PLC code is operational (`gvl.AnIn11 := gvl.AnOut11 / 2;`, e.g. the first analogue output should be half of the first analogue input). Go online and hit play if required. The result should now be visible.

`dSPACE` RTI and `ControlDesk` 6.3 operational: Connect a power supply to the power socket and the host-PC to the Ethernet Switch. Make sure the Ethernet connection is set up appropriately. Integrate the MAB into an existing network by giving it the IP address 172.19.3.44. Create a simple `Simulink` model (e.g. add 1 to an analogue input). When configuring ADC and DAC blocks in `Simulink`, select the appropriate type, module and channel. Connect the power supply to the appropriate pins. Build the `Simulink` model appropriately with the correct configuration parameters. Load it onto the `dSPACE` (create a new project, select the MAB II, and select the file you just created), create a time plotter in `ControlDesk` for the variables, then run while tuning the power supply and analyse the results. The `dSPACE` output should change accordingly.

Functionality MAB AO Type 1: Physically connect AIO Type 1 channel 1-4 output pins from `dSPACE` with appropriate inputs from the `Beckhoff` EL3004 analogue input module, and connect the signal ground accordingly. Create a simple `Simulink` model with four constant values in the range of `0` and `1`, which directly feed into 4 DAC blocks. Add an Out per channel so that later in the process the values can be analysed in `ControlDesk`. Make sure the build settings are configured correctly and build an `sdf`-file from the `Simulink` model so that it can be loaded into `ControlDesk`. In `ControlDesk`, start a new project and an experiment. Select the `dSPACE` MAB II as target and the newly created `sdf`-file as model. Then drag the in- and outputs into the Layout, creating a time plotter. In `ControlDesk`, go online and start measuring. In `TwinCAT`, activate the configuration and set the system to Run-mode. When analysing the connected EL3004 ports, the appropriate values should be visible.

No hardware disturbances on MAB: In `Simulink`, replace ΔP_c with a zero reference, terminate the AGC control input, and remove all potential measurement and process noise. Run and plot the resulting measurements (these are not 0, but of approximate magnitude $-5.88E-3$). Build the model and run it on the `dSPACE`, without activating it (so that the input is a zero reference as well). This should result in the same measurement values as in `Simulink`, otherwise there are unwanted hardware disturbances.

MAB AIO and `Beckhoff` I/O connection: Repeat physically connecting the AIO Type 1 channel output pins from `dSPACE` with appropriate inputs from the `Beckhoff` EL3004 analogue input module. Additionally connect the AIO Type 1 `dSPACE` input pins to the appropriate EL4004 outputs. Repeat creating a new `Simulink` model for the `dSPACE`, but now with 8 DAC outputs and 4 ADC inputs. When again building with the appropriate configuration settings, the same result as from the previous test should be visible for the other `dSPACE` outputs. The inputs can be tested by writing a value to the analogue outputs of the EL4004 and analysing the response at the `dSPACE` input end.

LFC areas on `dSPACE` and AGC on PLC 2: Create a new TC project and scan for I/O modules. Create a new PLC project. In the *Main* programme, configure the AGC code. This can be partially developed using the *PLC Coder* function in `Simulink`. However, this does not yet result in an operational code, so it should be rewritten accordingly. Also, an additional counter functionality has to be developed to run the AGC algorithm only once every two seconds, which can be done through the use of a `CASE` function. Link the input and output variables (which can be developed both locally in the Main programme or as global variable) to the appropriate I/O module ports. Activate the configuration. Select the appropriate `dSPACE` project (i.e. the two-area model without AGC, first without any measurement or process noise). Go online and start the measuring. If all is well, this should result in a stable system with slight offset. Logging in on `TwinCAT` and starting the AGC algorithm should decrease this offset to near zero.

Detection on PLC 2: For this, the Luenberger observer has to be translated to ST. This cannot be done directly using a `Simulink` Luenberger block (since this is not included in the *PLC Coder* library), so the internal structure of the block should be exposed and copied to a new `Simulink` file. There, $\hat{y}$ can be added as output (not an option for the `Simulink` block), after which a subsystem can be created. Involve the appropriate in- and outputs to and from the block, make it an autonomic unit, set the configuration settings correctly, and build a PLC code. The resulting ST file can relatively easily be copied into `TwinCAT`. A few adjustments: the initialisation `CASE` is not necessary, all variables are dedicated to a specific area, and the measurements and control inputs have to be converted to an array entry to be able to go through the for-loop. The residual can be generated by subtracting the estimated measurement values from the values coming out of watermark remover $\mathcal{Q}$. The absolute value is generated through a simple `IF`-statement (if lower than zero, multiply with -1).

VISU HMI: For visualisation purposes, an array can be manually constructed which updates its final value each time-step. This array should not update too frequently, as the available plot can only be 500 steps wide (which would mean only 5 seconds for a time step of 0.01s). Hence, a new task has to be created (of 100ms is this case), which has to be added to the PLC project, after which the programme in which the array is created should be linked to it. Create new visualisation in the PLC project. The constructed array can be visualised in a histogram. Insert such a histogram from the *Toolbox*, select the desired data array, select the *Curve* display type, and adjust aesthetically through its position and scale. As additional information point, the actual real-time variable can be displayed in a text box.

EAP connection between the PLCs: For connection verification, a simple counter algorithm is copied from [176]. If the connection is correctly established, the subscribed IPC should be able to read the same counter value as the publisher, after activating and logging in on the publisher side.

AGC on PLC 1: Start new TC project, and select the PLC 2 as target system. Link the I/O module after scanning for new devices. Rewrite these input variables to variables which can be published as output variables. Also rewrite the variables which will be sent back from PLC 1 as output variables. Add an EAP device (which will become *Device 1*), and select the *Ethernet 4* as adaptor. Follow the same steps for a publisher device as the previous test, but then with the rewritten measurements originating from the I/O module. On PLC 1, also properly rewrite the variables. Then, follow the same steps as the previous test to create and appropriately link a subscriber device. Now, back on the PLC 2, add a subscriber device for the returning control inputs, and link the appropriate variables. Set up the HMI as described before. It should now be functional.

Involve watermarking: A sequence of four watermarking state space stages is developed in `MATLAB`, which are tested for their stability. Similar to implementing the observer, now a regular state space system should be implemented in TC. To do so, copy the state space structure of a `Simulink` block, put it in a new model, and insert the correct matrices (the first stage of $\mathcal{W}$ is used as sample). Copy the relevant pieces of the thereafter created PLC code into a TC function. The tricky part is to get all the state space matrices right (four sequences of 16 matrices in total, of which multiple ones are duplicates). Write a `CASE`-function for the switching of the matrices according to the current watermark stage, and add this state as EAP publisher variable. As the watermarking filters have singular inputs and outputs, first try it for a single measurement ($\Delta f_1[k]$ in this case). Involving a function results in the necessity of working with arrays instead of individual values (as ST functions can only have a single output). Hence, rewrite the required variables on both PLCs. To test the functionality, one could overwrite the watermark variables with the unwatermarked variables, to test if the rewriting of variables to arrays was successfully done. Repeat the process for watermarking the input variables. Here, it is important to involve a second counter for the watermarker and remover of the control inputs. This should result in the exact same response as without watermarking.

Involve process and measurement noise: Instead of a model without noise, build a LFC model including the noise, and load it onto the `dSPACE`. Run the model as before. The whole HILDA testbed, without attack PC, should now be operational.

Glossary

List of Acronyms

AC	Alternating Current
ACK	Acknowledgement
ADC	Analogue-to-Digital Converter
ADS	Automatic Device Specification
AGC	Automatic Generation Control
ARP	Address Resolution Protocol
AI	Analogue Input
AO	Analogue Output
AWG	American Wire Gauge
BIOS	Basic Input/Output System
CPU	Central Processing Unit
CMRI	Complex Malicious Response Injection
DAC	Digital-to-Analogue Converter
DC	Direct Current
DCS	Distributed Control System
DCSC	Delft Centre for Systems and Control
DMWM	Dynamic Multiplicative Watermarking
DNP3	Distributed Network Protocol
DP	DisplayPort
DR	Detection Ratio
EAP	`EtherCAT` Automation Protocol
ECC	Energy Control Centre
EMS	Energy Management System
ETG	`EtherCAT` Technology Group
FAR	False Alarm Rate
FEP	Front End Processor
FDI	False Data Injection
HIL	Hardware-In-The-Loop
HILDA	Hardware-In-the-Loop Detection of Attacks
HMI	Human-Machine Interface
ICS	Industrial Control System
IDS	Intrusion Detection System

IED	Intelligent Electronic Device
I/O	Input/Output
IP	Internet Protocol
IPC	Industrial PC
IIR	Infinite Impulse Response
IT	Information Technology
LAN	Local Area Network
LED	Light-Emitting Diode
LFC	Load Frequency Control
MAB	MicroAutoBox
MAC	Media Access Control
MITM	Man-In-The-Middle
NMRI	Naive Malicious Response Injection
OPF	Optimal Power Flow
OS	Operating System
OSI	Open Systems Interconnection
PCS	Process Control System
PLC	Programmable Logic Controller
PROFIBUS	Process Field Bus
PROFINET	Process Field Net
RDC	Remote Desktop Connection
RES	Renewable Energy System
RTI	Real-Time Interface
RTU	Remote Terminal Unit
SCADA	Supervisory Control And Data Acquisition
ST	Structured Text
TCP	Transmission Control Protocol
USB	Universal Serial Bus
VM	Virtual Machine
WAN	Wide Area Network
XAE	eXtended Automation Engineering
XAR	eXtended Automation Runtime
ZIF	Zero-Insertion Force