Intrusion Response on Cyber-Physical Control Systems

Luis Francisco Cómbita Alfonso
ca.luis10@uniandes.edu.co

Faculty of Engineering
Electric and Electronic Engineering Department
Bogotá D.C.
2020
Intrusion Response on
Cyber-Physical Control Systems

Luis Francisco Cómbita Alfonso
c.a.luis10@uniandes.edu.co

Doctoral Dissertation

Advisor:
Nicanor Quijano, PhD
Co-Advisor:
Álvaro Cárdenas, PhD

Universidad de los Andes
Faculty of Engineering
Electric and Electronic Engineering Department
Bogotá D.C.
2020
To my mom and dad,
Gladys and Luis Eduardo,
my beloved partner,
Diana Marcela,
and my son,
Daniel Eduardo
I would like to express my deepest gratitude to my advisor, Professor Nicanor Quijano, for his supervision, guidance, and general support during this whole process. I would also like to thank Professor Alvaro Cárdenas, who welcomed me for a research stay during a year at University of Texas at Dallas and shared with me a broader vision of security on industrial control systems. I would like to express my sincere thanks to Professors Eduardo Mojica-Nava and Sandra Rueda for accepting to examine my Ph.D. thesis.

I would like to extend my thanks to my partners at the research group GIAP, from the Universidad de Los Andes, for the frequent talks around the development of my work along these years. I would also like to thank my colleagues at the research group IDEAS, from the Universidad Distrital, for their encouragement at the beginning of this project and the interesting talks at the end of it. A special word of thank should go to my colleague Carlos Suarez for his timely help on prior administrative stuff at beginning of the doctorate, which made it possible.

Many interesting talks I had while I was pursuing this doctorate come back my mind now. However, I have very special memories about a couple of them, they were very significant for this project, our talks with Carlos Barreto at several Saturday nights in the laboratory at the University of Texas at Dallas and with Jairo Espinosa at the end the Fourth Conference of Automatic Control at Medellín, Colombia.

Many people helped and gave me support to face this long project, my sincere thanks go for all of them, but without any doubt, I really need to do now a special acknowledgment. First, to my beloved partner, Diana, for her patience, support and permanent company along whole project, and to my family for its permanent encouragement, which at many times was needed.

Finally, but no less important, I gratefully acknowledge the Universidad Distrital Francisco José de Caldas for providing me with financial support through the Comisión de Estudios No. 015 de 2014. I would like to thank Colciencias for providing me with financial support for my research stay through the Convocatoria 727 Doctorados Nacionales 2015.
## Contents

List of Figures iii  
List of Tables vii  
Introduction 1  

1 Literature Review 11  
1.1 Cyber-Physical Control Systems 12  
1.2 Attacks on Cyber-physical systems 13  
1.2.1 Types of attacks 14  
1.3 Attack Detection 15  
1.4 Defense mechanisms 17  
1.4.1 Preventive mechanisms 18  
1.4.2 Resilient mechanisms 19  
1.4.3 Detection and Isolation based mechanisms 21  

2 Problem Formulation 27  
2.1 Existing System Setup 27  
2.2 Cyber-Attacks in Control Systems 30  
2.3 Attack Mitigation Mechanism Proposed 33  
2.3.1 Anomaly Detection 33  
2.3.2 Anomaly Isolation 35  
2.3.3 Attack Mitigation 37  

3 Noiseless Linear Case 39  
3.1 Existing System Setup 39  
3.2 Unknown Input Observers 41
## Contents

3.3 Detection, Isolation and Mitigation ................................................. 48

3.4 Numerical Results - Four Tanks System ........................................... 48

  3.4.1 System Model ............................................................................. 49
  3.4.2 Closed-loop system behavior ..................................................... 51
  3.4.3 UIOs bank design ....................................................................... 52
  3.4.4 Attack on Level 1 - Impact on the system ................................ 53
  3.4.5 Attacks on both levels - Impact on the system .......................... 58

4 Stability Analysis of Attacked Systems ............................................... 65

  4.1 Attacked System ............................................................................ 65

    4.1.1 Additive attack ......................................................................... 65
    4.1.2 Multiplicative attack ............................................................... 66

  4.2 Stability Analysis ........................................................................... 68

    4.2.1 Quadratic Lyapunov Stability for Discrete-time Systems ........ 68
    4.2.2 Quadratic Lyapunov Stability for the Attacked System .......... 69

  4.3 Numerical Results - Three Tanks System ....................................... 73

    4.3.1 Additive attacks ....................................................................... 75
    4.3.2 Multiplicative attacks .............................................................. 76
    4.3.3 Mitigation process .................................................................... 77

  4.4 Appendix ......................................................................................... 79

5 Noisy Linear Case .............................................................................. 83

  5.1 Existing System Setup ................................................................. 83

  5.2 Optimal Disturbance Decoupling Observers ................................... 84

  5.3 Optimal Disturbance Decoupling Observers Design ....................... 88

  5.4 Detection, Isolation and Mitigation ............................................... 89

  5.5 Numerical Results - Three Tanks System ....................................... 89

    5.5.1 Attacks Definition .................................................................... 90
    5.5.2 Mitigation Approach Implementation Results ....................... 90

6 Conclusions and Prospective Work .................................................... 99

Bibliography ......................................................................................... 103
List of Figures

1.1 Typical cyber-physical control system, where the system interacts with the PLC through a network. ................................. 12
1.2 Anomaly detection architecture for a feedback control system, with attack over sensors and actuators. ................................. 12
1.3 Model-based response general scheme. An adversary compromises sensor signals and the ADM compares the received $y_k^a$ with an estimated $y_k$ based on a model of the system. The ADM chooses a better option for the sensor measure $y_k^a$ that will maintain an adequate performance of the system. Adapted from [Cárdenas et al., 2011] .................................................. 21
2.1 Block diagram of a cyber-physical control system. ............... 29
2.2 closed-loop control system with sensor attack. ....................... 30
2.3 Control systems with mitigation of sensor attacks mechanism included. ................................................................. 34
3.1 closed-loop control system with sensor attack. ....................... 41
3.2 Four tanks benchmark system schematic, adapted from [Johansson, 2000]. ................................................................. 49
3.3 System response with change in reference of Level 1. Top figure shows the behaviour of level 1 and the bottom figure the behavior of level 2. The reference is shown in red, the measured output in blue, the attacked output in dotted black line, and the estimation of the full-order current observer in dashed green. .................. 51
3.4 Residue using full-order current estimation and its associated threshold. ................................................................. 52
3.5 System full-order current observer and UIOs outputs estimation. Top figure shows the behaviour of level 1 and the bottom figure the behavior of level 2. The estimation of the full-order current observer in dash-dot green, and the estimations of UIO 1 and 2 in dashed cyan and magenta, respectively. .................. 54
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.6</td>
<td>Residues using UIOs estimation. UIO 1 related residue in cyan, its threshold in dashed cyan, UIO 2 related residue in magenta and its threshold in dashed magenta.</td>
<td>54</td>
</tr>
<tr>
<td>3.7</td>
<td>Attack on sensor 1.</td>
<td>55</td>
</tr>
<tr>
<td>3.8</td>
<td>System response with attack on sensor 1. Dashed: sensor value. Blue: Physical variable (unknown for the controller). Red: Reference.</td>
<td>56</td>
</tr>
<tr>
<td>3.9</td>
<td>Residue using full-order estimation, when an attack is applied to output 1.</td>
<td>56</td>
</tr>
<tr>
<td>3.10</td>
<td>Detection and isolation variables. The detection signal from the full-order current observer is shown in green, the isolation in output 1 from UIO 1 in dashed cyan and, the isolation in output 2 from UIO 2 in dashed magenta.</td>
<td>57</td>
</tr>
<tr>
<td>3.11</td>
<td>System response with mitigation of an attack on sensor 1.</td>
<td>57</td>
</tr>
<tr>
<td>3.12</td>
<td>Control signals for the system without attack in blue, with attack in red and with attack and mitigation in dashed black. Top figure shows $u_1[k]$ and bottom figure shows $u_2[k]$.</td>
<td>58</td>
</tr>
<tr>
<td>3.13</td>
<td>System response with change in both reference inputs. Top figure shows the behaviour of level 1 and the bottom figure the behavior of level 2. The reference is shown in red, the measured output in blue, the attacked output in dotted black line, and the estimation of the full-order current observer in dashed green.</td>
<td>59</td>
</tr>
<tr>
<td>3.14</td>
<td>System response with attacks on both sensors. Top figure shows the behaviour of level 1 and the bottom figure the behavior of level 2. The reference is shown in red, the measured output in blue, the attacked output in dotted black line, the estimation of the full-order current observer in dashed green, and the estimations of UIO 1 and 2 in dashed cyan and magenta, respectively.</td>
<td>60</td>
</tr>
<tr>
<td>3.15</td>
<td>Residue using full-order estimation, when an attack is applied to output 1.</td>
<td>61</td>
</tr>
<tr>
<td>3.16</td>
<td>Detection and isolation variables. The detection signal from the full-order current observer is shown in green, the isolation in output 1 from UIO 1 in dashed cyan and, the isolation in output 2 from UIO 2 in dashed magenta.</td>
<td>61</td>
</tr>
<tr>
<td>3.17</td>
<td>System response with mitigation of attacks on both sensors. Top figure shows the behaviour of level 1 and the bottom figure the behavior of level 2. The reference is shown in red, the measured output in blue, the attacked output in dotted black line, the estimations of full-order current observer, UIO 1 and 2 in dashed green, cyan and magenta, respectively.</td>
<td>62</td>
</tr>
</tbody>
</table>
List of Figures

3.18 Control signals for the system without attack in blue, with attack in red and with attack and mitigation in dashed black. Top figure shows $u_1[k]$ and bottom figure shows $u_2[k]$. ............................... 63

4.1 Schematic diagram of three tanks system. ................................. 74
4.2 Response of the closed-loop control system without attacks. .... 74
4.3 Response of the closed-loop control system with additive attacks of magnitude $a$. ................................................................. 75
4.4 Response of the closed-loop control system with multiplicative attacks of magnitude $b$. ......................................................... 77
4.5 Response of the closed-loop control system with mitigated additive attacks of magnitude $a$. ......................................................... 78
4.6 Response of the closed-loop control system with mitigated multiplicative attacks of magnitude $b$. ............................................. 79
4.7 Response of the closed-loop control system with multiplicative attack of magnitude $b = 0.8$ with and without mitigation. ........ 80

5.1 Response of closed loop control system without attacks. ......... 90
5.2 Effect of the attack # 5 in the response of the control system. ... 93
5.3 Isolation of the attack # 5, red line denotes isolation on Level 1, and blue line denotes isolation on Level 2. ......................... 94
5.4 Attack detection, computed using of the Kalman filter, that is a part of the original control system, under attack # 5. ................. 94
5.5 Definitive attack isolation for attack # 5, red line denotes the existence and duration of an attack on the sensor of the Level 1. ..... 94
5.6 Mitigation response to sensor of Level 1 attack # 5 without mitigation response and with two different mechanisms of reconfiguration. 95
5.7 Effect of the attack # 8 in the response of the control system. .... 96
5.8 Detection and isolation of the attack # 8, red line denotes isolation on Level 1, and blue line denotes isolation Level 2. ............... 96
5.9 Attack detection, computed using of the Kalman filter, that is a part of the original control system, under attack # 8. ................. 97
5.10 Definitive attack isolation for attack # 8, red line denotes the existence and duration of an attack on the sensor of the Level 1. .... 97
5.11 Mitigation response to sensor of Level 1 attack # 5 without mitigation response and with two different mechanisms of reconfiguration. 98
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Four tanks system parameters.</td>
<td>50</td>
</tr>
<tr>
<td>5.1</td>
<td>Bias attacks applied on the system.</td>
<td>91</td>
</tr>
<tr>
<td>5.2</td>
<td>Static attacks applied on the system.</td>
<td>91</td>
</tr>
<tr>
<td>5.3</td>
<td>Key Performance Index Comparisons of different attacks applied on system sensors</td>
<td>92</td>
</tr>
</tbody>
</table>
Cyber-physical control systems are control systems of physical systems in which sensors, actuators, and controllers exchange information over networks, to maintain a pre-established performance of control systems even in the presence of disturbances. *Cyber* refers to remote sensing and activation, digital signal processing and computing, and network technologies. *Physical* systems commonly allude to the resources and structures which are necessary for a well-working society and economy. The goal of a this kind of cyber-physical system (CPS) is the improvement of the efficiency and reliability of the controlled physical systems.

The inclusion of information technology capabilities in control systems has some advantages as reduced system wiring, low installation and maintenance costs, and increased flexibility and adaption capability. However, using computer systems with a richer set of these functionalities also introduces potential new vulnerabilities, that can be exploited by intruders to cause an incorrect performance of the physical system and even an unsafe condition. A cyber-attack takes place when intruders tamper the information flowing inside a industrial CPS to deviate the nominal operation of the control loops and cause misbehavior, with effects ranging from simple performance degradation to those that can produce critical safety problems.

Several incidents have been reported in the literature about cyber-attacks on critical infrastructures, e.g., power plants, water treatment systems, pipelines and transportation systems. One of the most mentioned attacks on critical infrastructure was the breach occurred at Maroochy Water Services in Australia in March 2000, in this incident there were serious problems with the wastewater system. Control systems in the energy sector have been affected by malware, e.g., in 2003, the Slammer worm entered the network of the Davis-Besse nuclear power plant in Ohio; the worm launched a denial of service attack against the control network and the resulting overload of the network resources rendered the system for monitoring radiation and core temperatures inaccessible for five hours. In 2010, a substantial damage to the nuclear program of Iran was achieved using Stuxnet, a new worm with the capacity to modify the PLC commands causing an inappropriate operation of the centrifuges, while the normal operation values of the centrifuges were reported to the users. Since Stuxnet, reports on cyber-physical attacks have increased. Causing physical damage or injury with a cyber-attack is now seen as a reality, not merely a possibility. The occurrence of these attacks has drawn the attention of researchers, first, trying to utilize developments from the information technology community and, more recently, from the automatic
control community.

The information technology community has explored cyber threats on industrial CPSs addressing different approaches like: i) assessing the impact of cyber-attacks, ii) identifying the vulnerabilities, the models of the attacks, and the kinds of adversaries, and iii) improving the techniques of intrusion detection. Detection techniques include knowledge and behavior-based techniques. Knowledge-based intrusion detection approaches look for run-time features that match a specific pattern of misbehavior. Behavior-based intrusion detection approaches look for run-time features that are out of the ordinary. Ordinary can be defined concerning the history of a test signal or concerning a collection of training data. However, these techniques are very weak to face attacks on CPSs and prompted the exploration of new approaches to mitigate the effect of cyber-attacks.

Traditional protection mechanisms used in information technology systems can be used in cyber-physical control systems, but with different levels of effectiveness. These tools need to be developed with specific manners that try to adapt them to cyber-physical control systems features. Some of these features are, for instance, that in industrial CPSs the protection of integrity and availability may have higher priority than the protection of confidentiality. A difference between information technology systems and industrial CPSs is that in the former, trust data satisfy specific metrics, and in the last, the data comes from physical sensors of dynamical systems and, hence, satisfy physical first principles. Software patching very frequently in enterprise systems is not well suited for control systems that usually work in a continuous manner. And, finally, control systems are making decision systems working in real-time. Above mentioned limitations of applying traditional mechanisms of protection encourages the inclusion of the control theory to overpass these restrictions and achieve more secure and resilient control systems.

From the automatic control theory, significant research has been done. For instance, some works develop models of the industrial CPS to describe the tight coupling and high coordination between cyber and physical worlds, the high degree of automation, and the pervasive use of sensors interconnected inside centralized or distributed control systems. Industrial CPS are alluring targets for malicious intruders mainly because their security mechanisms are relatively immature. Due to this, there is significant work that deals with the mathematical model of cyber-attacks like denial of service, replay attacks, and deception attacks. These models have a purpose to analyze the system performance in the presence of cyber-attacks. The performance, analyzed in those works, is strongly related to stability, robustness, security, and resilience of industrial CPSs.
Attack detection is another topic that has been reported in the literature. In relation with attack detection, published literature can be grouped as: i) Bayesian detection with binary hypothesis methods, (ii) weighted least square (WLS) approaches, (iii) $\chi^2$ detector based on Kalman filters, and (iv) quasi fault detection techniques. Although the detection of cyber-attacks is an important step for mitigating attacks, such a step is not sufficient because in addition to detecting the presence of an attack, it is necessary to carry out actions that decrease or even cancel out the effect of the attack on control systems.

Detection and isolation mechanisms address the anomaly detection problem from the fault-tolerant control perspective. These mechanisms are activated after an attack is acting on a system. The main idea on these mechanisms is to turn on an alert when an attack is detected, to find the components where the attack is placed, and afterward to develop strategies as reconfiguration or accommodation to re-adjust the controller to produce a response in a manner that the effect of the attack is mitigated. Mechanisms on this issue can be grouped in five groups: i) analytical consistency, ii) water-marking, iii) baiting, iv) learning-based anomaly detection, and v) observer-based mechanisms.

Analytical consistency is named to the attack detection mechanisms on the communication layer based on the correlation and physical coupling between state variables and control decisions on the subsystems of a CPS. In there, the reconstruction of a physical signal can be done using related measurements, and in this manner redundancy relationships are established. This redundancy is used to detect if data have been manipulated in the communication layer. This method is very common in spatially distributed CPSs. Water-marking mechanisms are based on twofold: i) an injection of a known disturbance, and ii) the construction of a suitable metric, that signals the presence of an attack when the known disturbance does not have the established behavior in the vulnerable components of the system. These mechanisms have shown their effectiveness mainly to detect and isolate replay attacks. Baiting mechanisms and moving target mechanisms are based on the same principle. It consists in revealing worst-case scenarios of attacks on sensors and/or actuators. A manner to attain this objective is the addition of virtual state variables to prevent the attacker having full knowledge of the system. This lack of knowledge of the system prevents the attacker from achieving stealthy attacks. Similarly, moving target mechanisms are based on a strategy where the defender varies system attributes to cause the attacker losing the prediction capability which is needed to remain stealthy. This method has shown its effectiveness in the detection and isolation of attacks on sensors and actuators. Detecting the presence of suspicious data using machine learning is another technique useful to detect attacks on industrial CPSs. This technique is mainly based on signal processing methods, and its main weakness is that al-
though the anomaly is detected, true information usually cannot be recovered.

Observers-based mechanisms make a comparison between the healthy estimation resulting in the absence of attacks and the variables of the attacked system. Residues of this comparison trigger an alarm when the system is under attack. Using observers in the presence of attacks requires unknown input observers because under attack the information available for the estimation is not trustworthy. There are two kinds of estimation, static estimation where the relationship between variables is algebraic, and dynamic estimation where differential equations describe the relationship between the state variables. Industrial controllers usually include a dynamic estimation of the system variables. Several mechanisms based on observers are reported in the literature to detect and isolate attacks in a similar manner that faults are detected in fault-tolerant control systems. However, to the best of our knowledge, not enough work deploying observers-based mechanisms has been reported addressing automatic responses that can diminish the performance degradation which takes place in control systems under cyber-attacks. Bearing in mind the need to take actions to face cyber-attacks, in this work we deploy an architecture to mitigate attacks on sensors on cyber-physical control systems.

Some works about intrusion response mechanisms or reconfiguration mechanisms on industrial CPSs can be found in the literature. From them, two main perspectives are defined: i) preventive, and ii) reactive. Preventive responses (also called proactive) can be described as the identification of vulnerabilities of the system and improve its robustness by changing its physical structure, e.g., increasing the number of sensors, by changing parameters of the system, or by designing new control methodologies. Methods in this category are also designed to protect systems against disclosure of critical information. The mechanisms that can be gathered as proactive, can help to prevent that attacks happen or to postpone the onset of an attack, but it is well known that an intruder with enough budget may overpass these barriers and cause damage in the attacked system. Two typical methods of prevention are cryptography and randomization. Cryptography is the science of constructing and analyzing protocols that prevent third parties from reading private messages. The idea behind cryptography is to make sure that the data between a sender and a receiver cannot be revealed via an unauthorized user. Randomization as a defensive tool is utilized to confuse the potential attacker. Most of the randomization techniques aim to provide a confidentiality service using the randomization of data. The idea behind this is to use an alternative randomized data set to maintain the main data set free from confidentiality breaches.

The reactive intrusion responses are the actions whose purpose is to decrease
the maximum impact of an attack or to increase the resilience of the system. In concordance with the manner how the attack impact diminishes, the reactive response can be grouped as: i) mitigation or resilience mechanisms, and ii) detection and isolation mechanisms. Related to resilience-increasing, mechanisms can be grouped as (i) game theory, (ii) event-triggered control, (iii) mean sub-sequence reduced algorithms, and (iv) trust-based approaches. A game-theoretic approach that provides resilience consists of trying to maximize the price of attacking a system or minimize the damage that an attacker can apply to the system. Based on how frequent the attacks occur, event-triggered control schemes instead of time-triggered schemes emerged as appropriate tools to increase the resilience of control systems. Mean sub-sequence reduced (MSR) algorithms is a resilient control approach in which at each time of the updates, the controller, in order to not get affected by the attacks, ignores the suspicious values and computes the control input. One of the well-known applications of MSR algorithms is against Byzantine threats. Byzantine nodes are the computational nodes that, in an adversarial manner, send inconsistent information to their neighbors. Trust-based approaches are equivalent to redundancy-based approaches in graphs and are based on the assumption that if the number of attacks is sufficiently small, correct information can flow through the paths formed by trusted nodes.

More resilience methods can be found in the literature, but here we will only refer to a group of them that is strongly related to our work. In these works, secure estimation is defined and the conditions to recover the true state are also given, i.e., the minimum number of trustworthy sensors or actuators needed to attain the recovering of the full state. Scenarios noiseless and noisy have been considered in these works. The two main assumptions on these works are: i) there is a set of sensors that are impossible to be accessed by intruders, and ii) satisfying the controllability conditions of the system is enough for the system to work without losing the performance. These assumptions open a gap that we tackle in our work, because sensors of low-level controllers are not always inaccessible for malicious intruders, and an attack in a single sensor can degrade the controllability and stability conditions of the global system.

The main purpose of this work is to enhance the resilience to non-simultaneous sensor attacks of legacy industrial CPSs, that is, aged working cyber-physical control systems. Such enhancing is achieved without modifying the main tuned controller, nor the physical structure of the system (without physical sensor redundancy). In order to enhance resilience on legacy industrial CPSs, we propose an observer-based methodology to detect, isolate and mitigate the attacks, through filters and decision making systems that can be programmed in the same PLC in which the main controller is running. The detection mechanism is developed from the state estimator the system already has, assuming the control law depends on
the system state, generating residues that alert when the estimated outputs differ sufficiently from the sensor values. The isolation mechanism utilizes a bank of disturbance decoupler observers, where an observer is designed for each output in order to estimate the system state without the information of such output; again, we generate residues for each observer and, since there is only one attack at each time, only one observer triggers an alarm that the associated output has been attacked. Due to system noise, model nonlinearities and uncertainties, wearing of mechanical parts and so on, there might be false alarms related to non-existent attacks, a logic is develop to reduce false alarms based in the fact that only one attack can occur at some specific time. Finally, the state estimation of the disturbance decoupling observer related to the attacked output is used to feed the control signal calculation, which does not reflect the effect of the attack on the system, and, therefore, mitigates its effect in the overall system behavior.

On the road to fulfill the objective of this work, we study the stability effects on the system of very simple integrity attacks, showing that even when the attack is very simple and detectable, if its magnitude depends on the system state, it can destabilize the system, the proposed approach cannot recover the system to normal functioning.

We have proven the proposed methodology for linear systems with and without noise. For each case, the differences are in the needed system model nature and, therefore, the nature of the controller, the observer and the disturbance decoupler observer; keeping in mind that always we start from a closed-loop functioning system. We show numerical results for different type of controllers/observers for the three and the four tanks benchmark.

**Structure of the PhD Thesis**

This manuscript is organized as follows. Chapter 1 is dedicated to present a literature review about the response and reconfiguration of CPSs. Chapter 2 defines clearly the context of the work and gives a general description of the proposed methodology. Chapter 3 describes the strategy to mitigate the effect of attacks on sensors of low-level controllers for noiseless linear discrete-time systems. Chapter 4 shows the stability implications of integrity attacks and the impossibility of system recovery with the proposed approach. Chapter 5 focuses on the mitigation strategy for noisy linear discrete-time systems. Results presented in chapters 3, 4, and 5 are verified using the numerical examples of standard testbeds. Finally, in Chapter 6, several concluding remarks and prospective work are sketched.

The details of each chapter are presented in the following. The literature review
is presented in Chapter 1. First, a brief description of the main concepts of cyber-physical control systems is given. Second, some works and examples of incidents of cyber-attacks on industrial control systems are given. Third, mechanisms of detection and isolation of cyber-attacks are explored. Finally, an extended revision of work which can be classified as model response to attacks is done.

A very general problem formulation of our work is given in Chapter 2. In there, the setup system utilized across this dissertation is explained. The modeling of the cyber attacks in control systems is given and the mitigation scheme is also shown. This mechanism is based on three stages: i) attack detection, ii) attack isolation and iii) attack mitigation. Conventional tools of fault isolation produce false alarms when attacks happen in multi-variable systems, here we present a mechanism that reduces this amount of false-alarms. Finally, as the mitigation scheme, we present the control action compensation that is designed to reduce the effect of the attacks on sensors of the control systems.

Then, in Chapter 3 a deep analysis of our proposal is done. Starting from an existing control system based on state feedback, we propose to include a complementary algorithm to the existing controller to mitigate the effect produced by attacks on sensors. The complementary algorithm can be separated in two blocks: i) an unknown input observer bank and ii) a decision making mechanism. The unknown input observers are analyzed and designed with the purpose of recovering the state of the system with a significant reduction of the effect of the attack on sensors. Then, this recovered state is utilized to replace the original estimation that is corrupted with the effect of the attack. The decision making mechanism receives the information from sensors, the unknown input observer bank, and the state estimator that is part of the existing controller. This block generates as output the recovered information from sensors, nullifying the effect of the attack on a sensor. At the end of the chapter, some compelling results are used to show the effectiveness of the mechanism.

Chapter 4 shows the study of additive and multiplicative integrity attacks and their effects on system stability. These two attacks are bias attacks, where one system output presents a bias that can be arbitrary (additive attack), or proportional to the state variables (multiplicative attack). First, we show that no matter how big the additive attack is, it has no implication in the system stability whatsoever. For the case of the multiplicative attack, we show it can affect not only the control loop, but the estimation loop as well. Given that an attack that destabilize the system can be very harmful, we formulate an LMI problem to find a conservative bound on the attacks value that allows either, the control engineer or the intruder to know what sensors are more sensitive to attacks to protect or harm the system. At the end of the chapter we show some simulations on the
three tanks system showing compelling results that illustrate the developed theory and, moreover, that show that even with the mitigation mechanism proposed is not possible to recover the system once is destabilized.

Chapter 5 is an extension of Chapter 3. In this chapter we consider the measurement noise in sensors and actuators. Therefore, unknown input observers would not give adequate system state estimates, instead we use optimal disturbance decoupling observers. This kind of observer has the same philosophy than the Kalman filter in their design, and its finality, same as the unknown input observers, is to reconstruct the state vector without the effect of the attack. The decision making mechanisms used is exactly the same one used in Chapter 3, since it is independent of the nature of the estimators. At the end of the chapter, we show an extensive exploration through simulation of different attacks acting on the three tanks benchmark, and the mitigation is analyzed from the figures of the error integral obtained from the system without attack, with attack and no mitigation and with attack and mitigation.

At the end of the document, some conclusions are drawn and some future work is proposed, in order to continue with this research.

**Publications of the PhD Thesis**

Next, we present the publications result of this work.

<table>
<thead>
<tr>
<th>Publication in peer-reviewed international journals</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Book chapters</th>
</tr>
</thead>
</table>
Publication in international conferences


Publication in national conferences


Chapter 1

Literature Review

Widespread growth of new computing and network technologies has permeated industrial control systems (ICS), facilitating the pervasive use of remote sensors, and their interconnection with centralized control systems. These cyber infrastructures (including remote sensing and activation, digital signal processing, and computing) interact with physical industrial systems, creating a cyber-physical industrial control systems (CP-ICS). The goal of these CP-ICS is to improve the efficiency and reliability of these critical infrastructures; however, the inclusion of these technologies also opens the opportunity for cyber-attacks. The main purpose of these attacks is to modify the control loops to cause transgressions, with effects ranging from simple degradations on the performance of the control systems to those that can produce safety critical problems.

Over the years, several cyber security incidents affecting critical infrastructures have been reported [Turk, 2005], [Miller and Rowe, 2012], including security problems in power plants, water treatment systems, pipelines, and transportation systems. As the threats to these system continue to increase, the research community has been developing solutions in a variety of fields [Cárdenas et al., 2011, Hu et al., 2016], [Humayed et al., 2017], [Wolf and Serpanos, 2018], [Giraldo et al., 2017], [Altawy and Youssef, 2016], [He and Yan, 2016].

Cyber-attacks to control systems have been widely studied in the information technology (IT) community and, most recently, in the control community, where they have been focused on identifying the impact of adversaries [Genge et al., 2015], [Wang et al., 2010]; and on improving detection and isolation mechanisms [Pasqualetti et al., 2013], [Mitchell and Chen, 2014], [Cárdenas et al., 2011]; however, how to respond to attacks have attracted less attention.

In this chapter, we survey the literature describing ideas on cyber-attacks classification, detection, isolation, and attack mitigation, also known as response to cyber-attacks.
1.1 Cyber-Physical Control Systems

Some general principles on dynamic systems and cyber-security that will be used in this work are summarized. Figure 1.1 shows a typical cyber-physical control system, where the system sensors and actuators communicate with the Programmable Logic Controller (PLC) through a network. Therefore, an outsider can access the network to tamper sensor and actuators information in order to change system behavior, i.e. the outsider executes a cyber-attack on the control system.

![Typical cyber-physical control system](image1)

**Figure 1.1:** Typical cyber-physical control system, where the system interacts with the PLC through a network.

A feedback control system that includes anomaly detection and reconfiguration mechanisms is depicted in the Figure 1.2. In this architecture, the attacker may have access to sensor and/or actuators signals. We will describe the different elements illustrated in Figure 1.2 as follows.

![Anomaly detection architecture](image2)

**Figure 1.2:** Anomaly detection architecture for a feedback control system, with attack over sensors and actuators.

We will consider a plant or physical system which could be represented, in general, as nonlinear models with or without noise. Even though the plant can be modeled...
as a nonlinear system, it can be controlled by noiseless linear controllers and robust linear (or nonlinear) controllers including noise. Since those types of controllers need to be implemented in a PLC, they will be in discrete-time form.

Sensors are devices used to measure a particular physical quantity in terms of electrical variables, such as voltage or current. For instance, a temperature sensor will provide a voltage signal that evolves in time as the temperature varies in time. Therefore, if a sensor signal reaches a PLC through the network, it might be possible for an outsider to hack the network in order to modify the sensor signal, which will constitute a cyber-attack on the sensor.

Actuators are devices capable of generate changes in a physical variable of a system, which needs a source of power and a control signal. For instance, in a temperature system an actuator will be a source of heat; in a liquid level system an actuator will be a pump; in an inertial object moving in space actuators will be motors, that power wheels, or articulations, or rotors, or propellers, etc. Therefore, if a PLC sends a signal to an actuator through the network, an outsider could hack the network in order to modify the actuator signal, which will constitute a cyber-attack on the actuator.

1.2 Attacks on Cyber-physical systems

Attacks are considered as provoked events that affect the physical behavior of a system. Cyber-attacks are events that happened directly related to the communication network used by sensors, actuators and controllers to interact. Cyber-attacks may be performed over a significant number of attack points in a coordinated fashion and have a malicious intent.

The information technology community has explored cyber-attacks in industrial CPSs paying particular attention to ensure confidentiality, integrity, and availability of information. The confidentiality of data is related to the non-disclosure of information to unauthorized parties. The integrity of data refers to the trustworthiness of data. The availability of data is concerned with the utilization of information or resources when needed. [Do, 2015] This approach has been addressed using different perspectives like: i) assessing the impact of cyber-attacks, ii) identifying the vulnerabilities, the models of the attacks, and the kinds of adversaries, and iii) improving the techniques of intrusion detection. Detection techniques include knowledge and behavior-based techniques. Knowledge-based intrusion detection approaches look for run-time features that match a specific pattern of misbehavior. Behavior-based intrusion detection approaches look for
run-time features that are out of the ordinary. Ordinary can be defined concerning the history of a test signal or concerning a collection of training data. However, these techniques are very weak to face attacks on cyber-physical control systems and prompted the exploration of new approaches to mitigate the effect of cyber-attacks.

In control theory community the effort of the work has been targeted to analyze the impact of attacks on networked control systems, i.e. assessing the negative impact of the potential actions that intruders can deploy on industrial control systems. Different types of the attacks are explored, designing attacks which can partially or completely bypass traditional anomaly detectors, and proposing countermeasures for revealing undetectable attacks [Do, 2015]. Some works develop models of the CPS to describe the tight coupling and high coordination between cyber and physical worlds, the high degree of automation, and the pervasive use of sensors interconnected inside centralized or distributed control systems. Industrial CPSs are alluring targets for malicious intruders mainly because their security mechanisms are relatively immature. Due to this, there is significant work that deals with the mathematical model of cyber-attacks like denial of service, replay attacks, and deception attacks. These models have a purpose to analyze the system performance in the presence of cyber-attacks. The performance, analyzed in those works, is strongly related to stability, robustness, security, and resilience of Industrial CPS [Dibaji et al., 2019].

1.2.1 Types of attacks

Cyber attacks are classified in two general groups [Cardenas et al., 2008], [Pasqualetti et al., 2013], [Cheolhyeon Kwon and Hwang, 2013]: i) denial-of-service (DoS); and ii) integrity attacks. The main purpose of DoS attacks is to deny access to sensor or actuator information; mathematical models for these kind of attacks are summarized in [Do, 2015]. Integrity attacks are characterized by the modification of sensor and/or actuator information, compromising their integrity.

There are different types of cyber-attacks, which may be physical, where an adversary gain access to sensors or PLCs [Teixeira et al., 2014] and modify them, or that affect the communication layer (e.g., tamper sensor information). As depicted in Figure 1.2, attacks may affect directly the sensor measurement or the controller command. In [Teixeira et al., 2015], the authors summarized different types of attacks depending on the capabilities and resources of the attacker. Attackers may, or may not, have full knowledge about the system and its detection mechanisms. However, it is common to assume the worst case scenario, where the attacker knows how the system behaves (i.e., the attacker knows the model
of the system). If a sophisticated adversary (e.g., a state-sponsored attack) wants to harm a control system (e.g., Stuxnet [Langner, 2011]), they will spend a lot of time and resources in order to understand well the target system and the impact of their attacks.

- **Attacks on sensors** consist on replacing $y[k]$ (the real sensor measurement) it with $\tilde{y}^a[k]$. Replay attacks are a typical example of attacks over sensors.
- **Attacks on actuators** consist on replacing the vector $u[k]$ coming from the controller with $\tilde{u}^a[k]$, affecting directly the action that the actuators may execute.

A fault can be defined as a non permitted deviation of at least one parameter, or characteristic property, of the system from the acceptable condition [Hwang et al., 2010]. Deception attacks affect the data integrity of physical variables of the system and they have a similar effect as a fault on a control system.

### 1.3 Attack Detection

Interest on detection of cyber-attacks on physical systems has increased. In [Amin et al., 2013], the authors describe the dynamical model of an irrigation channel system. This model is used to design a bank of observers, which purpose is the detection and isolation of attacks on the irrigation control system. The detection and isolation of the attacks are based on analytic redundancy. This work utilizes traditional methods of fault detection and isolation (FDI) to face deception attacks on actuators/sensors. However, they are mainly focused on stealthy attacks, and they do not work on the reconfiguration of the control action. They also describe the critical measurements required for FDI.

A unified modeling framework for cyber-physical systems and attacks is proposed, and failure-sensitive filters to identify attacks are proposed in [Pasqualetti et al., 2013]. Conditions on limitations of monitoring of attacks, and characterization of undetectable and unidentifiable attacks are also provided.

Integrity attacks on sensors of supervisory control and data acquisition (SCADA) systems are explored in [Mo and Sinopoli, 2009], where the authors establish a feasibility condition for replay attacks and how to detect them with a noisy control authentication signal, but they do not discuss how to respond once an attack is identified. The work in [Mo et al., 2014] is specifically related with integrity attacks on sensors of SCADA systems. They establish a condition for the feasibility of replay attacks. Their proposal is based on the utilization of a noisy control authentication signal. Most of findings exposed in [Amin et al., 2013],
[Pasqualetti et al., 2013], and [Mo et al., 2014] are devoted to detection and isolation of integrity attacks on CPSs. One reason for expanding that work is to search ways for mitigating the effects of attacks with the use of control reconfiguration [Cómbita et al., 2015]. It is important to mention that fault-tolerant control (FTC) technology is being used as a tool for dealing with cyber-attacks [Li et al., 2018], [Ameli et al., 2018]; however, even these works that leverage the fault-tolerant literature focus on detection and isolation mainly using unknown input observers (UIOs), but they do not state how to mitigate these attacks.

Stochastic control has also being used to design resilient systems or secure control. In [Garcia et al., 2015], recently developed techniques to ensure resiliency of a control system are outlined. The specific techniques developed are: a method for sensor trustworthiness evaluation (using the probing signals approach); a non-classical statistical procedure for process variable assessment (using the sensors data and their trustworthiness); a method for sensor network adaptation to the optimal state resulting in the minimum entropy of process variable assessment (using the rational controllers approach); a method for sensor network decomposition (as a means for combating the curse of dimensionality and achieving scalability); and a method for subplant condition assessment (using the Jeffrey’s rule). Based on these techniques, a five-layer resilient monitoring architecture is proposed and analyzed under various cyber-physical attack scenarios. As quantified by the Kullback-Leibler divergence, the system offers effective protection against misleading information and identifies the plant conditions - normal or anomalous - in a reliable manner.

A proposal of a method to authenticate and detect the correct operation of sensor outputs in control systems is described in [Mo et al., 2015]. The concept of watermarking is laid on the validation of physical components with the injection of a noisy input to the physical system and the verification of its effect on the system outputs. When the system outputs agree with the response of the dynamical model to the mentioned input, the sensor output is considered genuine. In this way, if an attacker does not know the specifications of the noisy input, an injection is easily detected.

Another work based on stochastic control is presented in [Satchidanandan and Kumar, 2016]. The authors of this work, present a technique which ensures that malicious sensor nodes cannot introduce any significant distortion without being detected. This technique is based on the superimposing of a random signal on the actuator node, whose realization is unknown to the sensor, on the control law-specified input. With this fact, the technique ensures that either the malicious sensor is detected or it is restricted to add distortion, that is only of zero-power to the noise entering the system.
1.4. Defense mechanisms

Estimation of the state of a noisy linear dynamical system when an unknown subset of sensors is tampered by an attacker is outlined in [Mishra et al., 2017]. The described algorithm restricts the number of attacked sensors with an upper bound, and limits the scope to sensor attacks on presence of Gaussian noise. Techniques with the stochastic control as theoretic framework usually implies a expensive computational load that is not always compatible with legacy systems. For this reason, our first approach is limited to deterministic systems, and this work is mainly focused on the use of conventional tools of fault-tolerant control, taking into account the differences between random faults and malicious attacks.

Reconfiguration control actions are described as the mechanisms to trigger the response for maintaining the system stability or ensuring that the system remains in a safe zone, perhaps with some performance degradation. However, attacks and faults have significant differences, which complicates the use of reconfiguration control to face deception attacks [Teixeira et al., 2015]. This fact opens a gap to adapt the reconfiguration control tools in response to the distinctive features of the attacks. For instance, the adaptation of the controller in networked control systems to prevent and overcome current and future time delay switch attacks is presented in [Sargolzaei et al., 2017]. Another strategy for the attack mitigation is based on adaptive control techniques. In [Jin et al., 2017], the authors propose an adaptive controller able to deal with sensor and actuator attacks, which guarantees stability of the closed-loop dynamical system.

1.4 Defense mechanisms

To analyze the work related to the increase of security of CPSs, we group the strategies that are included in systems to face cyber-attacks and denominate them as defense mechanisms. These defense mechanisms are the actions that have the purpose of trying to maintain the CPS operating within a safe zone still in the presence of attacks on the system. We group all these actions in function of their occurrence time versus the occurrence time of the attack. Prevention mechanisms are targeted to postpone the onset of an attack and are executed prior to the operation of the system. Resilient mechanisms are targeted to decrease the maximum impact of the attack and are executed all of the time the system is in operation, with or without the presence of attacks. Detection and isolation mechanisms are focused on identifying the unreliable system and, after that, restore the normal operation as soon as possible [Sánchez et al., 2019]. All of these actions are also known as the response of security mechanisms to the attacks. There are two types of responses: i) preventive, or proactive, where the architecture of the system and
its controller are improved offline; and ii) reactive response, which corresponds to
the type of response where the control input (or a set of control inputs) is modified
online in such a way that the impact of the attack is mitigated.

1.4.1 Preventive mechanisms

Preventive actions have as a purpose to difficult disclosure attacks. Disclosure
attacks are usually focused on collecting information of the system, allowing an
intruder, to perform actions that degrade the normal operation of the system.
These preventive mechanisms can be grouped into two kinds that are cryptography
and randomization [Sánchez et al., 2019].

Cryptography refers to communication techniques derived from mathematical con-
cepts and a set of rule-based calculations, called algorithms, to transform messages
in ways that are hard to decipher, for secure communication. The idea behind
cryptography is to make sure that the data between a sender and a receiver cannot
be revealed via an unauthorized user.

A secure and private framework for networked control systems with encrypted
sensor measurements is analyzed in [Farokhi et al., 2017]. The conditions on
the parameters of the encryption technique that guarantee the stability and the
performance of the closed-loop system are provided. The approach presented is
also proved for distributed systems. Similarly, an encrypted implementation of an
MPC scheme for linear systems with polytopic constraints is analyzed in [Schulze
Darup et al., 2018]. Proper operation of the encrypted controller is guaranteed
setting conditions about the underlying quantization and encryption techniques.
However, this work remarks that the proposed approach is only practical for sys-
tems with a few states and short prediction horizons. Privacy schemes for SCADA
systems are also explored in several works. For instance in [Rezai et al., 2013] a
cryptographic key management scheme which increases the efficiency and security
of SCADA systems is presented and evaluated. The proposed key management
not only supports the required speed in the MODBUS implementation but also
has several advantages compared to other key management schemes for secure
communication in SCADA networks. However, the required hardware and com-
putational cost in the master station are increased. The actors in [Pramod and
Sunita, 2015] propose a key establishment scheme for SCADA systems based on
a polynomial key distribution scheme. In the proposed approach, the secret key
is not transmitted in the network for any device to device communications. The
scheme considers the constraints such as communication overhead, performance,
bandwidth and accuracy to a key establishment scheme for SCADA systems which
is an interesting mechanism that increases the security of critical infrastructures.
Randomization on security is used to induce confusion to potential attackers that look for key information of a system. Randomization is an important mechanism because it could prevent attackers from gathering information that can be used to predict exploitable system behavior [Sánchez et al., 2019].

Randomization in control of networked systems can be used to improve the overall performance of the system. In [Frasca et al., 2015] randomization and time-averaging are used together with a local gossip communication protocol, to obtain convergence of distributed algorithms to the global synchronous dynamics. In this work, distributed randomized algorithms are used to solve estimation problems in large-scale power systems. An example of masking the private data in the presence of an adversarial agent is shown in [Mo and Murray, 2017].

The authors show a privacy preserving average consensus algorithm to guarantee the privacy of the initial state and asymptotic consensus on the exact average of the initial values, by adding and subtracting random noises to the consensus process. Moreover, they prove that the algorithm is optimal in the sense that it does not disclose any information more than necessary to achieve the average consensus. Similarly, in [Nozari et al., 2017] the problem of multi-agent average consensus under the requirement of differential privacy of initial states of the agents against an adversary than has access to all the messages is explored. They present and prove the convergence of a novel differentially private Laplacian consensus algorithm in which agents linearly disturb their state-transition and message-generating functions with exponentially decaying Laplace noise. As a partial result in [Dibaji et al., 2018], an algorithm to overcome confidentiality attacks of the underlying control gains is presented. With such an algorithm, the closed-loop performance of the power system is preserved, while each generator keeps its control action private. Recently, in [Katewa et al., 2020] a noise adding differential privacy mechanism to protect the privacy of the sensitive parameters associated with the dynamics of multi-agent LTI systems is shown. The purpose of the privacy mechanism is to prevent that an intruder gains access to measurements from the agents and may estimate the values of sensitive model parameters, and then launch more severe attacks.

1.4.2 Resilient mechanisms

Preventive actions are related to mechanisms that can be deployed before the system is running, or before/after it has been a target of attackers. One of the main difficulties about this type of strategy is that designers are required to have a prior knowledge of some specific types of attacks that the system can suffer. Works in preventive actions have mainly focused on modifying the system structure or
designing more robust controllers.

In [Vaidya and Fardad, 2013] a new measure of network vulnerability is proposed based on controllability and observability Gramians. The mitigation strategy is obtained by using convex optimization to minimize the vulnerability measure. As a result, an optimal location of phasor measuring units (PMU) is obtained. Solving this problem needs to establish a trade-off between vulnerability minimization and the penalization of the number of sensors used.

Attacks against state estimators in the power grid have received significant attention. In [Vukovic et al., 2012], the vulnerabilities of the power system state estimator are explored. The authors define a measure that quantifies the importance of individual substations and the cost of attacking individual measurements. The goal is to mitigate integrity attacks by making a modification of routing and data authentication. The authors highlight the importance of analyzing the physical system and network topology as a whole to enhance the security of the state estimator against attacks.

Another set of operations on electric power systems where it is important to mitigate the impact of integrity attacks are power system stability and electricity market operation; for instance, attacks against the automatic generation control. In [Sridhar and Govindarasu, 2014], a methodology based on a composition of smart attack detection and mitigation for AGC is presented. The proposed mechanism has two objectives: i) to detect the presence of attacked measurements and prevent from performing incorrect area control error computations; and ii) to maintain the balance between generation and demand in the presence of unreliable measurements by using a model-based approach.

Authors in [Mo and Sinopoli, 2015] addressed the estimation problem for a more general framework, where an state $x \in \mathbb{R}$ is estimated based on $m$ sensor measures. An attacker can compromise $l < m$ sensors and the objective of the system operator is to find the optimal estimator that minimizes the estimation error knowing the number of sensors that are under attack (i.e., the operator knows or assumes $l$). They have shown that if $l > m/2$ then the optimal worst case estimator should ignore all $m$ measurements and be based solely on the apriori information. On the other hand, if $l < m/2$ it is possible to obtain an optimal estimator.

Clearly, proactive actions facilitate the detection and mitigation of cyber-attacks before they occur, which offers important benefits. However, it is impossible to configure a system a priori to defend against all possible attacks. Therefore, reactive response is necessary in order to identity what type of sensors, or controllers, are compromised and react as soon as an attack is detected. The response aims to minimize the impact of attacks by intelligently modifying the control actions.
1.4. Defense mechanisms

1.4.3 Detection and Isolation based mechanisms

Model-based response uses a model that generates an approximation of the normal behavior of the system without attacks. A schematic diagram of a basic reconfiguration system is presented on Figure 1.3, where \( y_k \) represents the true value of the system output, \( y^a_k \) represents the resulting value after a sensor attack, and \( \hat{y}_k \) represents an estimation of \( y_k \), based on a model of the plant. The anomaly detection module (ADM) has two inputs: \( \hat{y}_k \) and \( y^a_k \), and compares the measured signal with the estimation provided by the system model in order to take a decision. Thus, the ADM produces \( y^s_k \) that represents the safe signal that is sent to the controller to produce an adequate control action.

Several kinds of models are used to compare deviations from the system normal behavior. Some of them are based on finite state machines (FSM) and on a linearized model of the plant. Some of them show a hierarchical structure to make a good detection of attacks. In [Hartmann et al., 2014] a methodology based on reactive security is introduced, which combines a Model@run.time based simulation and a reasoning engine. This methodology monitors a smart grid that suffers an attack and adapts in a continuous way. The purpose of this approach is to make the smart grid self-adaptive and self-healing. In order to reach this goal, they build a model to give support to the adaptive system. This methodology needs the construction of a model, that must contain the static structure and the dynamic behavior of the system. However, there are some restrictions in the manner the system dynamics are included. In this work, they use FSM to model the dynamic of the system. The use of FSM makes impossible the use of this method in some applications as synchronization of generators in a smart grid, because in this kind of applications it is necessary to model the system dynamics using an ordinary differential equation. However, in the cases where the most
important behavior is related to steady state, a model based on a FSM seems to be a good choice.

In [Cárdenas et al., 2011], a linear model is used to estimate the state of the system, where the ADM compares the estimated measure $\hat{y}_k$ with the real one $y_k$. After an attack is detected, the ADM replaces the value of the sensor attacked and sends instead the value the linear model produces $y_k^s = \hat{y}_k$. With this value, the controller calculates the control action required for the system. In the case that no attack is detected the value of the sensor is sent to the controller in the same way that when there is no an ADM, $y_k^s = y_k$.

Another scheme based on a model is proposed in [Sha, 2001]. In this case, the response to an attack is based on a backup controller. This controller has two subsystems: the high assurance control (HAC) and the high performance control (HPC). To choose which subsystem is active, a trade-off between performance for stability and simplicity is established. The high-assurance and high-performance systems run in parallel, but remain separate. The HPC can use the signals from HAC, but the HAC does not use signals from the HPC. In normal conditions, with no attacks detected, the complex controller governs the plant. The decision logic ensures that states of the plant under any of the controllers remain inside an established stability envelope. This approach satisfies that the impact caused by incorrect actions must be tolerable and recoverable.

In [Chen and Abdelwahed, 2014], the authors propose an approach to design a self-protecting SCADA system based on the developing of a process control system model. One of the most important features of this proposal is that it is based on autonomic computing with minimal human intervention. The autonomic system makes an estimation of forthcoming attacks, gives the system administrator a fast detection signal, and executes the response to remove the cyber-attack. This elimination can be autonomous or semi-autonomous. Self-protection is based on several blocks: system model, intrusion estimation, intrusion detection, live forensics analysis and intrusion response. The estimation module predicts future safe states for the system based on historical measurements of controlled variables and selected security features of the SCADA system. The intrusion detection module, through anomaly and signature detection techniques, can detect known and unknown attacks in real time. Live forensics analysis module monitors and analyses network traffic and, in addition, audits files using forensics tools and statistical theories. New signatures are used to update the intrusion module and to feed the multi-criteria analysis controller. After an abnormal behavior is detected, the intrusion response system selects an appropriate response to recover the physical system behavior from abnormal to normal. This response is based on multi-criteria analysis, that takes into account four criteria: enhancement security, operational
costs, maintenance of normal operations and impacts on properties, finance and human safety. This multi-criteria controller is based on fuzzy logic theory.

The author in [McLaughlin, 2013] tackles the problem of how to verify control signals initiated by operators and controllers are allowed by a policy defined previously. The security policies contain three basic primitives: subjects, objects, and operations. The policy establishes whether each subject is allowed to perform each operation on each object. The proposed approach depends on controller of controllers (C\(^2\)), which mediates all control signals sent by operators and embedded controllers to the physical system. In particular, there are three main properties that C\(^2\) attempts to hold: 1) safety (the approach must not introduce new unsafe behaviors, i.e., when operations are denied the ‘automated’ control over the plant should not lead the plant to an unsafe behavior); 2) security (mediation guarantees should hold under all attacks allowed by the threat model); and 3) performance (control systems must meet real time deadlines while imposing minimal overhead).

Industrial Control Systems (ICS) nowadays use many embedded devices and diverse kinds of control systems, including Supervisory Control and Data Acquisition (SCADA) systems, Programmable Logic Controllers (PLC), Distributed Control Systems (DCS) and other types of controllers. These controllers are the core of miscellaneous industrial sectors and critical infrastructures. These control components are designed and linked to work together to achieve an industrial objective. ICS have evolved to incorporate or improve specifications of their products. This evolution rarely implies total replacement of controllers and technologies. Conversely, some times it is necessary to add new devices that must work together with legacy systems to produce new or improved goods. Substitution of such legacy ICS is expensive, requires extensive planning, and many times cannot be developed because the cost of stopping the production is expensive as well. Legacy ICS usually cannot support computationally expensive security operations while maintaining real-time control.

Most of wireless communication and other IT developments have permeated control systems devices. These improvements on control systems have implications on security of ICS. While the increasing connectivity enables connection of devices physically located in distant places, it also opens up a backdoor to cyber-attacks on control systems. For this reason, it is necessary to develop some tools that help to mitigate the effect of attacks on control systems.

ICS have as a fundamental requirement to maintain system and information integrity, and to assure that sensitive data have not been modified or deleted in an unauthorized and undetected manner. Integrity attacks denote cyber-attacks,
where the true information of sensors and/or actuators is tampered. When the information about the physical variables of the system from sensors is modified, the controller executes the predefined control law with erroneous information and, as a consequence, the output variables do not maintain the required performance. In a similar way, if the control action computed by the controller is modified, the behavior of the system is different from the designed one. These two kinds of actions are integrity attacks on the information of the physical variables of the system. If the parameters of the controller are altered, there is a disruption on the behavior of the system because, in this case, the control law does not match with the originally designed law.

Conventional tools from Fault Detection, Isolation, and Reconfiguration (FDIR) can be adapted to face integrity attacks on sensors of industrial CPS. Fault Detection and Isolation (FDI) methods can be divided into two groups [Zhang and Jiang, 2008]: i) model-based methods, and ii) data-based methods. Model-based methods can be quantitative or qualitative. Quantitative methods include state estimation, which include observers-based methods, for deterministic systems, and Kalman filters based methods, for stochastic systems. Model-based methods need an accurate model of the plant. With this FDI method, it is required to design an observer or a bank of observers based on a dynamical model of the physical system. The quality of the results is strongly linked with the accuracy of the mentioned model.

Several applications of FDI can be found in the literature. Most of them have implicitly the fault detection procedure inside of the fault isolation procedure. There are numerous works that use different kinds of observers to detect and isolate faults on sensors and actuators. In [Bateman et al., 2007] a method for fault detection, isolation, and estimation applied on actuator failures for an UAV is described. The method is based on the use of unknown input decoupled functional observers and it is focused on constant and variable failures.

Mitigation of attacks in control systems means to develop modifications on the control action in order to reduce the severity of the effects caused by the mentioned attacks. The aim of Fault Tolerant Control (FTC) is to prevent a fault, from becoming a failure, i.e. the system is not able to perform its mission, and/or surrounding people or resources are at a high risk. The purpose of an attacker of a control system is to try to modify the system behavior, and to cause a strong failure on the system. These facts are our motivation to utilize techniques from FTC to mitigate the effect of integrity attacks on sensors.

Several ways have been developed to enhance the performance of the control systems where there is a fault. After the detection and isolation of the fault are
achieved, a control re-design is required. The new goal of the controller is to keep the system operable despite the fault causes a wrong control action. This process is known as control reconfiguration and, in this work, deals with the situation where an important signal from the system variables is tampered. If a tampered signal is fed to the controller, the effect could be that the system reaches an unsafe operation.
In this chapter, we start considering our object of study: legacy industrial control systems (ICSs) that have migrated to become industrial cyber-physical control systems. Such systems are physical plants, with nonlinear behavior, that exhibit satisfactory performance in a closed loop with a particular discrete-time control system implemented on a PLC, which can be a linear controller with or without consideration of noise effects on the system. The consideration or not of noise effects on the system does not affect the controller itself, but the kind of state estimator that it is needed to be implemented, i.e., for linear systems, we use full-order current observers; and for linear systems with noise, we use Kalman filters. Now, the main issue when legacy ICSs migrate to cyber-physical control systems is that they are unprotected to cyber-attacks once the network security has been breached. The core of the chapter is a proposal of architecture to help mitigate the attack effects in cyber-physical control systems, which can be implemented in the same PLC the original control system is running. In the proposed architecture, three stages are included: anomaly detection, done with the state estimator already implemented together with the controller; anomaly isolation, done with a bank of decoupling disturbance observers - which have to be design according the nature of the estimator; and attack mitigation, where the state without the effect of the attack is recovered to recompute the control action and mitigate the effect of the attack in the control system.

2.1 Existing System Setup

We consider a physical dynamical system that works with a digital controller in a closed loop manner through a network, i.e., an existing cyber-physical control system, like the one shown in Figure 2.1. The controller allows the system to maintain a specific behavior, where normally the system is able to follow a reference input and to maintain specific characteristics in the transient response. Since the real system is considered to be, in general, nonlinear, it would have a behavior
that can be modeled as

\[ \dot{x}(t) = f(x(t), u(t), t) + \eta(t), \]

\[ y(t) = g(x(t), u(t), t) + \zeta(t), \]

(2.1)

where \( x(t) \in \mathbb{R}^n \), \( u(t) \in \mathbb{R}^m \), and \( y(t) \in \mathbb{R}^p \) are the system state, input, and output, respectively. The functions \( f : \mathbb{R}^{n+m+1} \rightarrow \mathbb{R}^n \) and \( g : \mathbb{R}^{n+m+1} \rightarrow \mathbb{R}^p \) associate, with each value of state, input and time variables, a corresponding vector in \( \mathbb{R}^n \) and \( \mathbb{R}^p \), respectively. Vectors \( \eta(t) \) and \( \zeta(t) \) represent noise, perturbations, or variations on the model parameters, related to the system state in the first equation, and to the system output in the second equation. Since the closed loop system works with a digital controller, we consider the controller is designed with a discrete-time approximation of the system, which can be expressed as

\[ x[k+1] = f_d(x[k], \tilde{u}[k], k) + \eta[k] \]

\[ y[k] = g_d(x[k], \tilde{u}[k], k) + \zeta[k] \]

(2.2)

where \( x[k] \in \mathbb{R}^n \), \( u[k] \in \mathbb{R}^m \), and \( y[k] \in \mathbb{R}^p \) are the discrete-time system state, input, and output, respectively. Notice that the system input is not \( u[k] \) but \( \tilde{u}[k] \), which represents \( u[k] \) after passing through the network. Functions \( f_d \) and \( g_d \) represent the discrete-time equivalences of functions \( f \) and \( g \). In the same way as \( \eta[k] \) and \( \zeta[k] \) are the discrete-time equivalences of \( \eta(t) \) and \( \zeta(t) \), representing independent zero mean noise vector sequences, with covariance matrices \( Q \) and \( R \), respectively. This kind of system model defined by (2.2) can be obtained from either: i) doing some modeling of the system, and discretizing it; or, ii) learning the discrete-time model from input-output data, using an adequate sampling time according to the closed loop system dynamical behavior [Franklin et al., 1997].

The controller that works with the system is in general represented as

\[ u[k] = h(\hat{x}[k], \tilde{y}[k], y^*[k]), \]

(2.3)

where \( \hat{x}[k] \) is the estimated state, \( \tilde{y}[k] \) represents \( y[k] \) after passing through the network, \( y^*[k] \in \mathbb{R}^p \) is the system reference input (the one the system is desired to follow), and \( h : \mathbb{R}^{n+p} \rightarrow \mathbb{R}^m \) is a function of the state and the reference input.

As usual, we consider not all of the system states are available for implementing the part of the controller related to the state feedback. In order to estimate system states, we use a linear current observers with and without noise, depending on the specific case, with the following dynamics

\[ \hat{x}[k] = l(\hat{x}[k-1], u[k-1], \tilde{y}[k]), \]

(2.4)
2.1. Existing System Setup

where $\hat{x}[k]$ is a estimate of the state system, and $l : \mathbb{R}^{(n+m+p)} \to \mathbb{R}^n$ is a function of the system estimated state, input, and output. Notice that (2.4) is only a straightforward formulation to show that the estimated state is calculated using its previous estimation and previous system input, and also the current system output. However, there is always the presence of the observer gain and, depending on the estimator, the use of matrices $Q$ and $R$, and also more than one stage for the estimation process, as in the Kalman filter (see chapter 5).

Since the system and the controller are coupled by a network, the control signal received by the system is not $u[k]$ but $\tilde{u}[k]$, and the output signal received by the controller is not $y[k]$ but $\tilde{y}[k]$, where

$$\tilde{u}[k] = \sum_{i=0}^{q} \delta[\tau_k - i] u[k - i]$$  \hspace{1cm} (2.5)

and

$$\tilde{y}[k] = \sum_{i=0}^{q} \delta[\tau_k - i] y[k - i].$$ \hspace{1cm} (2.6)

The Kronecker delta function $\delta[\tau_k - i]$ is used to represent the random communication delays and stochastic data missing. The time delay $\tau_k$ is a random variable considered to be an integer multiple of the sampling time, $T_k$, introduced to describe the possibility of data missing as well as the size of the delay occurred at time instant $k$, and $q$ represents the maximum delay allowed before the network produces a timeout error. For the ideal case, there is no communication delay, i.e., $\tau_k = 0$, then $\delta[0 - i] = 1$ only for $i = 0$ and hence $\tilde{u}[k] = u[k]$. For a com-
A typical networked tracking control system is shown in Figure 2.1, in the ideal case (without communication delays) \( \tilde{y}[k] = y[k] \) and \( \tilde{u}[k] = u[k] \). When the system is under attack, as in Figure 2.2, equation (2.2) can be extended to include attacks on sensors as follows

\[
\begin{align*}
    x[k+1] &= f_d(x[k], \tilde{u}[k], k) + \eta[k], \\
    y[k] &= g_d(x[k], \tilde{u}[k], k) + \zeta[k], \\
    \tilde{y}_a[k] &= \tilde{y}[k] + F_a a[k],
\end{align*}
\]

(2.7)

where \( a[k] \in \mathbb{R}^p \) represents external attack signals in each of the outputs, keeping in mind that for time \( k \) only one of the signals can be different from 0, and \( F_a \in \mathbb{R}^{p \times p} \) is a matrix that indicates how the attacks signals affect each output.

In general, in this work, we are considering attacks on sensors, which consist on replacing \( \tilde{y}[k] \) (the real sensor measurement after passing through the network) with
with any consistent data value. This kind of attack is known as *integrity*, *false data injection*, or *deception* attack; it can be achieved by, either: i) adding a value to the real measurements (*bias* attacks), or ii) replacing directly the measurement (*static* attacks). Depending on whether or not there is a relationship between the added value with the measurement, *bias* attacks can be considered as *additive*, when the added value to the sensor measurement is, either, constant or random; or *multiplicative*, when the added value is proportional to the measurement.

For integrity attacks, we assume the attacker can alter the true information sent by sensors with the goal of deceiving the controller and, therefore, computing a control action that drives the control system to an unsafe or undesired behavior. These attacks can be achieved when the actual system measurements are replaced by data that are compatible with the measurement equation of the system [Dan and Sandberg, 2010, Liu et al., 2009]. We assume the attacker knows the valid range of the measurement of sensors, then, he will produce an attack vector not trivially detectable. In [Liu et al., 2009], it is shown that an attacker can manipulate these measurements without being detected. In this attack, the attacker does not require knowledge about the model of the system, but the knowledge about current values of the measurements is enough. With current values and the span of the measurements it is easy to compute an attack vector.

It is important to point out that in this work we consider attacks in every output, but those attacks can not be simultaneous, i.e., for time $k$ there is only one active attack. Integrity attacks considered in this work are of the form

$$a[k] = m(k, y[k]), \quad (2.8)$$

where $m(k, y[k])$ varies depending on the type of attack applied to the system sensors.

For *additive bias* attacks, the $j^{th}$ component of $a[k]$ is mathematically defined as

$$a_j[k] = \begin{cases} 
0, & t < t_1, \\
f \cdot f_{s_1}, & t_1 \leq k \leq t_1 + 20, \\
f, & t_1 + 21 \leq k \leq t_2 - 21, \\
f \cdot f_{s_2}, & t_2 - 20 \leq k \leq t_2, \\
0, & t > t_2,
\end{cases} \quad (2.9)$$

where $j = 1, 2, ..., p$, $f$, $f_{s_1}$, and $f_{s_2}$ are functions to soften the initial and final portions of the attacks, as

$$f_{s_1} = \frac{1 + \tanh \left(0.1[k - t_1] - 1\right)\pi}{2} \quad (2.10)$$
and

\[ f_{s_2} = \frac{1 - \tanh \left( \frac{(0.1[k - (t_2 - 20)] - 1)\pi}{2} \right)}{2}, \quad (2.11) \]

$t_1$ and $t_2$ are the initial and final times of the attack, $f$ is the function that shapes the attack itself, which is suitably defined to affect only one sensor. The purpose of smoothing functions is to produce a soft transition to the sensor data, in order to try to avoid that a detector of abrupt changes can easily detect the attack. We use (2.10) when the change is the addition of a positive value and, (2.11) is used when the change is the subtraction of a positive value.

For multiplicative bias attacks, the $j^{th}$ component of $a[k]$ is defined as

\[ a_j[k] = \alpha_j[k] y_j[k], \quad (2.12) \]

with

\[ \alpha_j[k] = \begin{cases} 0, & \text{for } k \text{ when there is not attack}, \\ \alpha_j, & \text{for } k \text{ when there is attack}, \end{cases} \]

where $\alpha_j \in \mathbb{R}$ indicates an attack proportional to the sensor value.

For the static attack, $a[k]$ is such that $\tilde{y}_a[k]$ becomes a static value with some noise for the duration of the attack, and the $j^{th}$ component of $a[k]$ is defined as

\[ a_j[k] = \begin{cases} -y[k] - \eta[k] + k_s + \eta_s[k], & t_1 \leq k \leq t_2, \\ 0, & \text{otherwise}, \end{cases} \quad (2.13) \]

where $\eta_s[k]$ is an independent zero mean noise signal, with the standard deviation equal to the sensor signal characteristics.

In order to design attack-resilient legacy control systems, we need: i) to detect that an attack is taking place; ii) to isolate (identify) the attacked sensor; and iii) to reconfigure the system and/or change its operation to mitigate the attack (e.g., replace the sensed measurements by a virtual sensor [Cárdenas et al., 2011]). After considering the various existing approaches used to make a system resilient, or at least to detect and isolate attacks, we decide to make a deepen exploration of the use of fault tolerant control (FTC) techniques in order to propose a mechanism that, once an attack has been detected and isolated, is able to mitigate the attack effect on the system.
2.3 Attack Mitigation Mechanism Proposed

In this section, we present a mechanism that generates a response to mitigate the effect produced by a integrity attack in one sensor of a control system. The mitigation response decreases the deviation in the system outputs produced by a sensor cyber-attack. When the controller receives misleading information, it computes an incorrect control action, changing the normal operation of the control system. The attack response is based on the recovering of the state without the effect of the attack, and the posterior computation of the control action using a trustworthy estimation of the state and the output of the system. The computation of the control action with trustworthy information mitigates the effect of the attack.

To obtain trustworthy information, it is necessary to detect and isolate where the anomaly takes place. Some previous works show the use of FTC tools to detect and isolate sensor attacks in control systems [Amin et al., 2013, Teixeira et al., 2015]. In this section, we go a step further by showing a response mechanism after the anomaly detection and isolation is done. The attack-response algorithm computes the required control action with trustworthy information to mitigate the impact that the sensor attack produces in the performance of the control system. The complete scheme for mitigating the effect of the attack in a closed loop control system is shown in Figure 2.3, where the Decision Making Mechanism includes attack detection, isolation, and mitigation, which recovers the state and, therefore, the sensor output, both, without the effect of the attack.

2.3.1 Anomaly Detection

There are a number of anomaly detection mechanisms in the literature, some of them using data driven techniques and some others with model-based structures [Ding, 2008, Hwang et al., 2010]. An anomaly detector is a system in charge to find out when an attack, or anomaly, takes place. However, it gives no information about where the anomaly takes place.

In this work, anomaly detection is done using a residual based on a state estimator, already designed and in use for the feedback control law calculation. To find out when an attack takes place, we define full-order current observer associated residues as

\[
    r[k] = ||\tilde{y}_a[k] - C \hat{x}[k]||_2,
\]

where \( ||x||_2 \) stands for the 2-norm of a vector \( x \in \mathbb{R}^n \), with components \( x_i \) for
$i = 1, 2, ..., n$, i.e., $||x||_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$. An attack is detected when

$$r[k] > \tau_D,$$  

where $\tau_D$ is known as a detection threshold. Due to modeling imperfections of the controlled system and convergence time of the estimator, the residual is not exactly zero when there are no attacks on sensors. The threshold determination is done based on the calculation of the residual with no attacks on sensors of the system, for different situations. Hence, the supremum of $r[k]$, for all $k$, could be chosen as the detection threshold $\tau_D$. If a value smaller than the supremum of $r[k]$, for all $k$, is chosen as the detection threshold some additional false anomalies are detected, and if a value greater than the supremum of $r[k]$, for all $k$, is chosen as the isolation threshold, then some anomalies will not be detected. The binary variable $b[k]$ is used to denote whether or not an attack is active at $k$ on any sensor of the system. In conclusion,

$$b[k] = \begin{cases} 
1, & r[k] > \tau_D \\
0, & \text{otherwise.} 
\end{cases} \quad (2.16)$$
2.3.2 Anomaly Isolation

Anomaly isolation algorithms have the goal of identifying where the attack is located, i.e., which sensor information is false. This, in general, can be achieved by implementing a bank of decoupling disturbance observers (DDOs). In this work, attacks are considered as disturbances that we need to decouple. In order to isolate the anomaly at time \( k \), a bank of DDOs is designed to work without one output (sensor measurement) of the control system. The DDO that is insensitive to anomalies on the \( j^{th} \) sensor has as input all components of the control action vector \( u[k] \) and, all but the \( j^{th} \) component of the attacked output vector, \( \hat{y}^j[k] \); this DDO produces a state estimate vector \( \hat{x}^j[k] \) that does not have the attack effect on the \( j^{th} \) output, if any.

Associated with state estimation and the output estimation from the \( j^{th} \) disturbance decoupling estimator, \( \hat{x}^j[k] \), the \( r^j[k] \) residual vector is computed by

\[
r^j[k] = \| \hat{y}[k] - C \hat{x}^j[k] \|_2 \quad \text{for} \quad j = 1, 2, \ldots, p.
\]

The residual \( r^j[k] \), associated to the \( j^{th} \) sensor, is computed using the information from all inputs \( u[k] \) and all but the \( j^{th} \) component of the outputs \( \hat{y}^j[k] \). When there are no anomalies on actuators, and there is only one attack in the \( j^{th} \) sensor, the residual satisfies

\[
r^j[k] > \tau^j_I,
\]

where \( \tau^j_I \) is known as the isolation threshold for the \( j^{th} \) DDO, which helps to reduce false isolation of attacks. \( \tau^j_I \) must be chosen in a similar way as \( \tau_D \), from the system behaviour without attacks for different inputs. It is worth noticing that when \( \tau^j_I \) is chosen to be smaller than the supremum of \( r^j[k] \), for all \( k \), some additional false anomalies are isolated, and if it is chosen to be larger than this value, some anomalies may not be isolated. The binary variable \( l^j[k] \) is used to denote whether or not an attack is active at \( k \) instant on the \( j^{th} \) sensor, as

\[
l^j[k] = \begin{cases} 
1, & r^j[k] > \tau^j_I \\
0, & \text{otherwise}.
\end{cases}
\]

However, in multivariable systems some (or all) outputs are usually coupled to some (or all) inputs, and for this reason, one sensor attack in the \( i^{th} \) sensor can be wrongly isolated in another sensor, i.e., \( l^j[k] = 1 \) for \( j \neq i \). As a consequence of that, a mechanism to suppress the isolation of false attacks is presented in the next section.
2.3.2.1 Preventing Isolation Because of False Alerts

The isolation mechanism described above produces imperfect results. These imperfections are the result of the coupling between outputs of a multivariable system, i.e., an anomaly/attack on the $j^{th}$ sensor is not just revealed on the residual of the correspond observer, but it is also revealed in the other residuals, usually delayed, and with a smaller amplitude than in the residual $r^j[k]$. Hence, in this section we introduce a mechanism to correct the isolation results based on previous facts.

The false anomaly suppression is done using the previously defined assumption that establishes that only one sensor attack/anomaly can occur simultaneously, and there is no actuator attack/anomaly acting on the system. The first step of false anomalies/attacks suppression is to disable the isolation of more than one attack/anomaly simultaneously. This correction generate $L^j[k]$ variables, using their past values and the values of $l^i[k]$: \[ L^j[k] = (L^j[k-1] \& l^i[k]) \| \left( l^i[k] \& \bar{L}^j[k-1] \& \bar{l}^1[k] \& \cdots \& \bar{l}^{j-1}[k] \& \bar{l}^{j+1}[k] \& \cdots \& \bar{l}^p[k] \right), \text{(2.20)} \]

where $\&$ represents the AND logic operator, $\|$ represents the OR logic operator, and $\bar{a}$ represents the NOT logic operator of $a$. Equation (2.20) means that $L^j[k]$ can be equal to 1, for two different situations:

1. If $L^j[k-1]$ and $l^i[k]$ are both equals to 1, then at the $k-1$ instant an attack was detected in the $j^{th}$ sensor, and the attack remains active at $k$.
2. If $l^i[k]$ is equal to 1, the previous value of $L^j[k-1]$ is equal to 0, and $l^i[k] = 0$, for $i \neq j$, then there is no previous attack in the $j^{th}$ sensor, but now there is one, if and only if there is no attack in other sensors at the same time.

The accurate information about the time duration of the attack on the $j^{th}$ sensor is synthesized in the binary variable $m^j[k]$, using the results from (2.16), (2.19), and (2.20) as \[ m^j[k] = L^j[k] \& b[k], \text{(2.21)} \]

where $m^j[k]$ indicates that at the $k^{th}$ sampling, on the $j^{th}$ sensor, there is an attack if $m^j[k] = 1$, or there is no attack if $m^j[k] = 0$. 
2.3.3 Attack Mitigation

Some previous works have developed similar mechanisms as the ones described above [Teixeira et al., 2015], [Amin et al., 2013], [Mo et al., 2014], [Pasqualetti et al., 2013]. These works have been focused on detection and isolation of cyber-attacks on control systems. In this work, we take an additional step, which consists in the addition of a mechanism with the aim of being able to mitigate the effect produced by a sensor cyber-attack of an ICS. Notice that such mechanism is added to improve the security of an existing networked controller. The proposed mechanism is developed in the same hardware where the remote controller is implemented. The purpose of this mechanism is for the system outputs to avoid having a big deviation with respect to the nominal response, when under attack.

Control action compensation is the last stage in the developed approach to mitigate the effect of a sensor cyber-attack in an ICS. It deals with the stage in which the authentic information of the sensor is recovered. The authentic information of the sensor is the signal before the attacker modifies it. In order to restore the nominal control of the system, it is necessary to find the authentic signal of the sensor using analytical redundancy. As it is explained above, the \( j^{th} \) DDO, is designed to be insensitive to sensor attacks in sensor \( j \), and all other DDOs are sensitive to attacks on sensor \( j \). For this reason, we can define the mitigated state estimation as

\[
\dot{x}_m[k] = \begin{cases} 
\dot{x}[k], & \text{for } b[k] = 0, \\
\dot{x}^j[k], & \text{for } m^j[k] = 1; \ j = 1, 2, \cdots, p. 
\end{cases} \tag{2.22}
\]

That is, if there is no attack, i.e., \( b[k] = 0 \), the mitigated state estimation correspond to the full-order current estimation and, if there is attack, since only one signal \( m^j[k] = 1 \) at time \( k \), we use the estimation of the \( j^{th} \) UIO. From the mitigated state, we also can define a mitigated output as

\[
y_m[k] = C \dot{x}_m[k]. \tag{2.23}
\]

Therefore, the controller that mitigates attacks on the system can be written as

\[
u[k] = h(\dot{x}_m[k], y_m[k], y^r[k]), \tag{2.24}
\]

where we can see that, in the control action, instead of using the attacked output, \( \hat{y}^a[k] \), we use the mitigated output, \( y_m[k] \), and instead of using the estimated state, \( \dot{x}[k] \) (which has effects of the attack, even though it is not attacked directly), we use the mitigated state, \( \dot{x}_m[k] \).
Chapter 3

Noiseless Linear Case

In this chapter, we consider the simpler ideal case where the system is running with a linear controller and the noise in the system signals can be neglected. For the present case, the system model could be obtained from input-output data, knowing the order of the system. We consider as a controller a state-feedback with tracking. As state estimator we select a full-order observer and, in order to isolate the attacks, we implement a bank of unknown input observers (UIOs). The logic for diminishing false alarms, and the attack mitigation mechanism are the same as the ones defined in Chapter 2. We show results of applying the automatic mitigation scheme proposed for the three tank benchmark, assuming the running controller is linear and the signal noise can be neglected.

3.1 Existing System Setup

We consider a physical dynamical system that works with a digital controller in a closed-loop manner through a network, i.e., an existing cyber-physical control system, like the one depicted in Figure 2.1. The controller allows the system to maintain a specific behavior, where normally the system is able to follow a reference input and to maintain specific characteristics in the transient response. Since the real system is considered to be, in general, nonlinear, it would have a behavior that can be modeled as in (2.1). Since the closed loop system works with a digital controller, we consider the controller is designed with a more straightforward approximation, a noiseless discrete-time linear approximation of the system, which can be expressed as

$$x[k + 1] = A x[k] + B \hat{u}[k] + E d[k],$$
$$y[k] = C x[k],$$  \hspace{1cm} (3.1)

where $x[k] \in \mathbb{R}^n$, $u[k] \in \mathbb{R}^m$, and $y[k] \in \mathbb{R}^p$ are the discrete-time system state, input, and output, respectively. The signal $d[k] \in \mathbb{R}$ corresponds to disturbances such as noise, nonlinearities, model inaccuracies, or uncertainties; and the matrix
\( E \in \mathbb{R}^{n \times 1} \) represents how the disturbances affect the system. Notice that the system input is not \( u[k] \) but \( \tilde{u}[k] \), which represents \( u[k] \) after passing through the network. \( A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m} \) and \( C \in \mathbb{R}^{p \times n} \) are the dynamic, input, and output matrices of the system, respectively. This kind of system model defined by (3.1) can be obtained from either: i) modeling the system, linearizing it, and discretizing it; or, ii) learning the discrete-time model from input-output data, using an adequate sampling time according to the closed-loop system dynamical behavior [Franklin et al., 1997].

The controller that works with the system is considered to be a tracking control with state feedback, represented as

\[
\begin{align*}
\tilde{v}[k] &= y^r[k] - \tilde{y}[k] + v[k - 1], \\
u[k] &= K_f \tilde{v}[k] - K_s \hat{x}[k],
\end{align*}
\]  

(3.2)

where \( y^r[k] \in \mathbb{R}^p \) is the system reference input (the one the system is desired to follow), \( \tilde{v}[k] \in \mathbb{R}^p \) is a discrete-time approximation of the error integral, \( e[k] = y^r[k] - \tilde{y}[k]. \) The control signal \( u[k] \) is obtained as a linear combination of the state vector, through the state feedback gain \( K_s \in \mathbb{R}^{m \times n} \), and the error integral vector, through the integral gain \( K_i \in \mathbb{R}^{m \times p} \). The signal vector \( \hat{x}[k] \) represents the estimated state, since usually we do not have available all the state variables for feedback. In order to estimate system states, we use a full-order current observer [Franklin et al., 1997], [Phillips and Nagle, 1995], with the following dynamics

\[
\begin{align*}
\bar{x}[k] &= A \hat{x}[k - 1] + B u[k - 1], \\
\hat{x}[k] &= \bar{x}[k] + L (\tilde{y}[k] - C \bar{x}[k]),
\end{align*}
\]  

(3.3)

where \( \bar{x}[k] \) is the predicted estimate based on a model prediction from the previous time estimate, which is after corrected by the measurement of the output becoming \( \hat{x}[k] \), and \( L \in \mathbb{R}^{n \times p} \) is the observer gain that guarantees \( A - LC \) is Hurwitz, when \( (A,C,A) \) is observable. Notice that, after some manipulations, the full-order current (state) observer can be rewritten as

\[
\hat{x}[k] = [A - LAC] \hat{x}[k - 1] + [B - LBC] u[k - 1] + L \tilde{y}[k],
\]  

(3.4)

which is the equation implemented in the digital device where the controller is running.

It is worthwhile to emphasize that the system, controller and observer described in this section are supposed to already exist. That is, we do not design either the controller or the observer. They are supposed to exist and work reasonably on the
3.2 Unknown Input Observers for the reconstruction of outputs without the effect of the attack

Let us consider the closed-loop control system of the previous section with model as in (3.1), controller as in (3.2) and observer as in (3.3). The system is disturbed with a sensor attack (after passing through the network), with

$$\tilde{y}_a[k] = \tilde{y}[k] + F_a a[k],$$  \hspace{1cm} (3.5)

where $a[k]$ is a vector of $p$ functions that represents the attack signals, $F_a \in \mathbb{R}^{p \times p}$ represents the outputs affected by the attack $a[k]$, i.e., only one output at a time.

In this particular case, for the system in (3.1), we consider the disturbance $d[k]$ to represent the modification on the state due to the change of the control signal by a sensor attack and, given that we assume there is only one attack acting simultaneously, we consider such modifications can be represented by a single signal. Therefore, the matrix $E$ represents the unattacked outputs acting on the system state.

We are interested in estimating the output signals without the effect of the attack, in order to compensate the control action signal and to prevent the attack from
feeding back the system and making it to collapse. We use a bank of $p$ UIOs, one for each sensor that could be attacked, where each UIO does not take into account the $j^{th}$ output. That is

$$\tilde{y}_a^j[k] = M^j \tilde{y}_a[k],$$

where $M^j \in \mathbb{R}^{(p-1) \times p}$ is a transformation matrix equals to an identity matrix without the $j^{th}$ row, with $j = 1, 2, \ldots, p$. The hypotheses behind using UIOs without one of the outputs is that, since there is only one attack simultaneously, when the attacked output is not considered we will be able to estimate accurately the state variables and, therefore, the outputs.

The $j^{th}$ UIO is described using the following state-space representation, inspired by [Chen and Patton, 1999],

$$\begin{align*}
    z^j[k+1] &= F^j z^j[k] + T^j B u[k] + K^j \tilde{y}_a^j[k], \\
    \hat{x}^j[k+1] &= z^j[k+1] + H^j \tilde{y}_a^j[k+1],
\end{align*}$$

(3.6)

where $z^j[k] \in \mathbb{R}^n$ is the dynamic (first) approximation of the estimated state vector, $\hat{x}^j[k] \in \mathbb{R}^n$ is the estimated state vector, which corresponds to the UIO that does not uses the information of the $j^{th}$ output for the estimation process, i.e., $\tilde{y}_a^j[k]$ is the output vector where the $j^{th}$ component is eliminated. $F^j \in \mathbb{R}^{n \times n}$, $T^j \in \mathbb{R}^{n \times n}$, $K^j \in \mathbb{R}^{n \times (p-1)}$ and $H^j \in \mathbb{R}^{n \times (p-1)}$ are design matrices such that the estimated state of the UIO, $\hat{x}^j[k]$, converges to $x[k]$ when there is no attack, i.e., $F_a^j = 0$. When the $j^{th}$ UIO (3.6) is applied to the system (3.2), decomposing $K^j = K^j_1 + K^j_2$, the estimation error ($e^j[k] = x[k] - \hat{x}^j[k]$) is governed by the equation

$$\begin{align*}
    e^j[k+1] &= F^j e^j[k] - K^j_a F^j_a a[k] - H^j F^j_a a[k+1] + \\
    &- \left[ F^j - (I - H^j C^j) A + K^j_a C^j \right] x[k] + \\
    &- \left[ T^j - (I - H^j C^j) \right] B u[k] + \\
    &- \left[ K^j_a - F^j H^j \right] \tilde{y}_a^j[k] + (I - H^j C^j) E^j d^j[k],
\end{align*}$$

(3.7)

notice that we use $E^j$ instead of $E$ and $d^j[k]$ instead of $d[k]$, indicating that each UIO considers its own perturbations. That is because we consider that, for each UIO, a way of approximate the control action modification is to give more weight to the state variables related with the outputs used directly by the $j^{th}$ UIO.

Let us consider for a moment the UIO behavior where we neglect the attack effect on output sensors ($F_a = 0$), in order to see how the UIO estimate the system.
3.2. Unknown Input Observers

For this case, \( \tilde{y}_j[k] = \tilde{y}_j[k] \), and (3.7) becomes

\[
e^j[k + 1] = F^j e^j[k] + (I - H^j C^j) E^j d[k] - \left[ F^j - (I - H^j C^j) A + K_1^j C^j \right] x[k] + \\
- \left[ T^j - (I - H^j C^j) \right] Bu[k] + \\
- \left[ K_2^j - F^j H^j \right] \tilde{y}_j[k].
\]

(3.8)

We know that a proper state estimation is achieved when the estimation error for the \( j^{th} \) UIO has the form

\[
e^j[k + 1] = F^j e^j[k],
\]

(3.9)

and the eigenvalues of \( F^j \) need to be stable in order for the estimation error to converge to zero. That implies for the UIO to estimate the state, we need to make equal to zero all the terms on the right side of (3.8) but the first. That is, we need to guarantee that

\[
E^j = H^j C^j E^j, \quad (3.10)
\]

\[
F^j = (I - H^j C^j) A + K_1^j C^j, \quad (3.11)
\]

\[
T^j = I - H^j C^j, \quad (3.12)
\]

\[
K_2^j = F^j H^j. \quad (3.13)
\]

When we consider the attack effect on the outputs (\( F_a \neq 0 \)), holding (3.10)-(3.13), the estimation error of the \( j^{th} \) UIO will be governed by

\[
e^j[k + 1] = F^j e^j[k] - K_1^j F_a^j a[k] - H^j F_a^j a[k + 1],
\]

(3.14)

where, depending on the form of \( H^j F_a^j \) and \( K_1^j F_a^j \), it will eventually converge proportionally to \( a[k] \) or to zero. That is, if there is an attack on the \( j^{th} \) output and how the \( j^{th} \) UIO does not consider that output then \( e^j[k] \rightarrow 0 \), i.e., \( \tilde{x}_j[k] \rightarrow x[k] \) and we will have the estimation of the outputs without the effect of the attack. On the other hand, if there is an attack on the \( i^{th} \) output and how the \( j^{th} \) UIO considers that output \( e^j[k] \propto a[k] \), and the estimated state will not help to recover the outputs without the effect of the attack.

In order to design the UIOs, we need to solve (3.10)-(3.13). Notice that solving those equations is limited to solve (3.10) for \( H^j \), that allows us to solve the rest of the equations, if we can guarantee that \((A_1^j, C)\) is detectable, with \( A_1^j = (I - H^j C^j) A \). Bellow we introduce a Lemma, addressing the existence of solution of (3.10).
Lemma 3.2.1  Equation (3.10) is solvable if and only if

\[ \text{rank}(C_j^j E^j) = \text{rank}(E^j). \]

If \( E^j \) is full column rank; then a special solution of (3.10) is

\[ H^j = E^j [(C_j^j E^j)^\top] (C_j^j E^j)^{-1} (C_j^j E^j)^\top. \]

Proof: When (3.10) has a solution \( H^j \), we have

\[ (C_j^j E^j)^\top H^j = E^j, \]

i.e., \( E^j \) belongs to the range space of the matrix \( (C_j^j E^j)^\top \) and this leads to

\[ \text{rank}(E^j) \leq \left( \text{rank}(C_j^j E^j) \right), \]

i.e.,

\[ \text{rank}(E^j) \leq \text{rank}(C_j^j E^j). \]

However,

\[ \text{rank}(C_j^j E^j) \leq \min\{\text{rank}(C_j^j), \text{rank}(E^j)\}, \]

and, since \( (E^j) \) is full column rank

\[ \min\{\text{rank}(C_j^j), \text{rank}(E^j)\} \leq \text{rank}(E^j), \]

we have

\[ \text{rank}(C_j^j E^j) \leq \text{rank}(E^j). \]

Therefore, the only way of satisfying (3.16) and (3.17) is that \( \text{rank}(C_j^j E^j) = \text{rank}(E^j) \) and the necessary condition is proved.

When \( \text{rank}(C_j^j E^j) = \text{rank}(E^j) \) holds true, \( C_j^j E^j \) is a full column rank matrix, and a left inverse of \( C_j^j E^j \) exists:

\[ (C_j^j E^j)^+ = [(C_j^j E^j)^\top C_j^j E^j]^{-1} (C_j^j E^j)^\top. \]

Clearly, \( H^j = E^j (C_j^j E^j)^+ \) is a solution of (3.10). \[ \square \]

Now, we introduce a lemma to show equivalence of the detectability of an augmented system and the one of the original system.

Lemma 3.2.2  Let \( C_1^j = [C_j^j \quad C_j^j A]^\top \), then the detectability for the pair \((C_1^j, A)\) is equivalent to that for the pair \((C^j, A)\).
Proof: The observability of a system can also be verified if
\[
\text{rank}\left\{\begin{bmatrix} sI - A \\ C \end{bmatrix}\right\} = n,
\tag{3.19}
\]
for any \( s \in \mathbb{C} \) (see Theorem 5-13 in [Chen, 1984]). Therefore, if \( s_1 \in \mathbb{C} \) is an unobservable mode of the pair \((C^j, A)\), we have
\[
\text{rank}\left\{\begin{bmatrix} s_1 I - A \\ C^j \end{bmatrix}\right\} = \text{rank}\left\{\begin{bmatrix} s_1 I - A \\ C^j A \end{bmatrix}\right\} < n.
\]
This means that a vector \( \alpha \in \mathbb{C}^n \) will exist such that
\[
\begin{bmatrix} s_1 I - A \\ C^j \end{bmatrix} \alpha = 0,
\]
which implies that
\[
\begin{bmatrix} s_1 I - A \\ C^j \end{bmatrix} \alpha = 0, \quad \text{or} \quad \text{rank}\left\{\begin{bmatrix} s_1 I - A \\ C^j \end{bmatrix}\right\} < n.
\]
That is to say that \( s_1 \) is also an unobservable mode of the pair \((C^j, A)\). Now, if \( s_2 \in \mathbb{C} \) is an unobservable mode of the pair \((C^j, A)\), we have
\[
\text{rank}\left\{\begin{bmatrix} s_2 I - A \\ C^j \end{bmatrix}\right\} < n.
\]
This means that a vector \( \beta \in \mathbb{C}^n \) can always be found, such that
\[
\begin{bmatrix} s_2 I - A \\ C^j \end{bmatrix} \beta = 0,
\]
which can be rewritten as
\[
(s_2 I - A) \beta = 0 \quad \text{and} \quad C^j \beta = 0. \tag{3.20}
\]
Multiplying on the left by \( C^j \) the first equation in (3.20)
\[
C^j (s_2 I - A) \beta = 0,
\]
that is equivalent to
\[ C^j s_2 \beta = C^j A \beta = 0 \text{ (since } C^j \beta = 0). \]

Hence
\[
\begin{bmatrix}
  s_2 I - A \\
  C^j \\
  C^j A
\end{bmatrix}
\beta =
\begin{bmatrix}
  s_2 I - A \\
  C^j \\
  C^j A
\end{bmatrix}
\beta = 0,
\]
i.e., \( s_2 \) is also an unobservable mode of the pair \((C^j_1, A)\). As the pairs \((C^j_1, A)\) and \((C^j, A)\) have the same unobservable modes, their detectability is formally equivalent.

At this point, it is important to include the necessary and sufficient conditions for UIOs.

**Theorem 3.2.3** The conditions for (3.6) to be an UIO for the system defined by (3.2) are:

(i) \( \text{rank}(C^j E^j) = \text{rank}(E^j) \).

(ii) \((C^j, A^j_1)\) is detectable pair, where

\[ A^j_1 = A - E^j [(C^j E^j)^\top C^j E^j]^{-1} (C^j E^j)^\top C^j A. \]

**Proof:** According to Lemma 3.2.1, (3.10) is solvable when condition (i) holds true. A special solution for \( H^j \) is

\[ H^j_* = E^j [(C^j E^j)^\top C^j E^j]^{-1} (C^j E^j)^\top. \]

In this case, the system dynamics matrix is

\[ F^j = A - H^j C^j A - K^j_1 C^j = A^j_1 - K^j_1 C^j, \]

which can be stabilized by selecting the gain matrix \( K^j_1 \) due to the condition (ii). Finally, the remaining UIO matrices described in (3.6) can be calculated using (3.12) - (3.13). Thus, the observer (3.6) is a UIO for the system defined in the two first rows of (3.2).

Since (3.6) is a UIO for (3.2), (3.10) is solvable. This leads to the fact that condition (i) holds true according to Lemma 3.1. The general solution of the matrix \( H^j \) for (3.10) can be calculated as

\[
H^j = \underbrace{E^j (C^j E^j)^\top}_H + H^j_0 [I_m - C^j E^j (C^j E^j)^\top],
\]
where $H^j_0 \in \mathbb{R}^{n \times (p-1)}$ is an arbitrary matrix and $(C^j E^j)^+$ is the left inverse of $C^j E^j$, defined in (3.18).

Substituting the solution for $H^j$ into (3.11), the system dynamics matrix $F^j$ is

$$F^j = A - H^j C^j A - K^j_l C^j$$

$$= [I_n - E^j (C^j E^j)^+ C^j] A +$$

$$- [K^j_l H^j_0] \left[ I_m - C^j E^j (C^j E^j)^+ \right] C^j A$$

$$= A^j - [K^j_l H^j_0] \left[ C^j A^j_1 \right]$$

$$= A^j - \bar{K}^j_l \bar{C}^j_1,$$

where $\bar{K}^j_l = [K^j_l H^j_0]$ and $\bar{C}^j_1 = \left[ C^j \begin{bmatrix} C^j A^j_1 \end{bmatrix} \right]$. Since the matrix $F^j$ is stable, the pair $(\bar{C}^j_1, A^j_1)$ is detectable, and the pair $(C^j, A^j_1)$ also is detectable according to Lemma 3.2.2.

Once we have verify the necessary conditions to design the bank of UIOs, we need to prove that it is possible to recover the state estimation of the output without the effect of the attack from the $j$th UIO.

**Theorem 3.2.4** Suppose (3.6) is the $j$th UIO for the system defined by (3.2), if there is an attack in the $j$th output, we can get an estimation of the output without the effect of the attack from the $j$th UIO.

**Proof:** When we consider there is an attack on the system (3.2), i.e., $F_a \neq 0$. In fact, since we only consider one simultaneous attack $F_a$ can be written as a set of unitary column vectors with all but one element different from zero. Lets consider the simpler case where $a[k]$ is a scalar function and $F_a$ is a unitary vector. Assuming that attack is affecting the $j$th output, the form of $F_a$ is

$$F_a = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \leftarrow j^{th} \text{ row}$$
then \( F_j^a = 0_{(p-1)\times 1} \) (\( F_j^a \) equals \( F_a \) without the \( j^{th} \) row, in this assumption, the only element different from zero), and clearly (3.14) will become

\[
e^j[k + 1] = F^j e^j[k],
\]

that is equal to the expression obtained when there is no attack considered (i.e., \( F_a = 0 \)). The last implies that \( \hat{x}[k] \to x[k] \) and, therefore, we can find \( \hat{y}[k] = C \hat{x}[k] \), that represents the estimation of the outputs without the effect of the attack.

Now, if we consider the general case where \( a[k] \) is a vector of \( p \) functions and \( F_a \in \mathbb{R}^{p \times p} \), the previous procedure holds but only for the time intervals where each attack is taking place, given that only one attack will occur simultaneously.

\[ \square \]

### 3.3 Detection, Isolation and Mitigation

We use the attack mitigation mechanism proposed and detailed in Section 2.3. For this particular case, we propose the use of the already working full order current observer described by (3.3) to know when an attack takes place, in a process known as detection. Whereas, for the isolation process we use a bank of UIOs, where the \( j^{th} \) UIO is defined as in (3.6), that in the previous section has proven to recover the system state without the attack effect. We maintain the false alarm reduction logic described in Section 2.3 and, given the specific control in use, the calculation of the control action, with the nomenclature given in Subsection 2.3.3, can be written as

\[
\begin{align*}
v[k] &= y^r[k] - y_m[k] + v[k - 1], \\
u[k] &= K_I v[k] - K_S \hat{x}_m[k],
\end{align*}
\]

where instead of using the attacked output, we use the mitigated output (without the attack effect) and, instead of using the estimated state form the full order observer, we use the estimated state from the \( j^{th} \) UIO, which does not have the attack effect.

### 3.4 Numerical Results - Four Tanks System

In this section we consider the four tanks benchmark, initially proposed in [Johansson, 2000] (see Figure 3.2). We use a tracking controller with state feedback,
working with a full-order current observer. The objective of this section is to show: i) the system working in closed-loop before any attack is considered; ii) the effect of an attack in one output; and, iii) the effect on the system of using the mitigation mechanism proposed.

Figure 3.2: Four tanks benchmark system schematic, adapted from [Johansson, 2000].

### 3.4.1 System Model

The system depicted in Figure 3.2 has two inputs related with the pumps, $u_1$ and $u_2$, which allow liquid to be fed in the tanks. The height of the four tanks, $h_i$ for $i = 1, 2, 3, 4$, are the state variables of the system, and $k_c$ times the height of tanks 1 and 2 are the system outputs. The four tanks system model can be written as

\begin{align}
\frac{dh_1}{dt} &= -\frac{a_1}{A_1} \sqrt{2gh_1} + \frac{a_3}{A_1} \sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1} u_1, \\
\frac{dh_2}{dt} &= -\frac{a_2}{A_2} \sqrt{2gh_2} + \frac{a_4}{A_2} \sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2} u_2, \\
\frac{dh_3}{dt} &= -\frac{a_3}{A_3} \sqrt{2gh_3} + \left(1 - \gamma_2\right) \frac{k_2}{A_3} u_2, \\
\frac{dh_4}{dt} &= -\frac{a_4}{A_4} \sqrt{2gh_4} + \left(1 - \gamma_1\right) \frac{k_1}{A_4} u_1,
\end{align}

(3.22)
where $A_i$ is the area of the $i$-th tank, $a_i$ is the outflow section of the $i$-th tank (in the bottom of the tank), $g$ is the gravity acceleration, $\gamma_j \in [0, 1]$ for $j = 1, 2$ is a proportional constant that divides a pump flow $u_j$ in two parts ($\gamma_j$ and its complement, to feed two different tanks), and $k_j$ are the pump gains. The parameters of the system considered are the same as in [Johansson, 2000], showed in Table 3.1.

### Table 3.1: Four tanks system parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1, A_3$</td>
<td>[cm$^2$]</td>
<td>28</td>
</tr>
<tr>
<td>$A_2, A_4$</td>
<td>[cm$^2$]</td>
<td>32</td>
</tr>
<tr>
<td>$a_1, a_3$</td>
<td>[cm$^2$]</td>
<td>0.071</td>
</tr>
<tr>
<td>$a_2, a_4$</td>
<td>[cm$^2$]</td>
<td>0.057</td>
</tr>
<tr>
<td>$k_c$</td>
<td>[V/cm]</td>
<td>0.50</td>
</tr>
<tr>
<td>$g$</td>
<td>[cm/s$^2$]</td>
<td>981</td>
</tr>
<tr>
<td>$\gamma_1, \gamma_2$</td>
<td></td>
<td>(0.70, 0.60)</td>
</tr>
<tr>
<td>$k_1, k_2$</td>
<td>[cm$^3$/V s]</td>
<td>(3.33, 3.35)</td>
</tr>
<tr>
<td>$(h_1, h_2)$</td>
<td>[cm]</td>
<td>(12.4, 12.7)</td>
</tr>
<tr>
<td>$(h_3, h_4)$</td>
<td>[cm]</td>
<td>(1.8, 1.4)</td>
</tr>
<tr>
<td>$(\bar{u}_1, \bar{u}_2)$</td>
<td>[V]</td>
<td>(3.00, 3.00)</td>
</tr>
</tbody>
</table>

The controller used with the system is of the same form as in (3.2) [H. Noura and Chamseddine, 2009], with

$$K_I = \begin{bmatrix} 0.0150 & 0.0037 \\ -0.0016 & 0.0092 \end{bmatrix}$$

and

$$K_S = \begin{bmatrix} 0.0928 & 0.0086 & 0.0056 & -0.0025 \\ -0.0021 & 0.0797 & -0.0002 & -0.0040 \end{bmatrix}.$$  

The full-order current observer for the system is as described in (3.3), with

$$L = \begin{bmatrix} 0.0736 & -0.0002 \\ -0.0000 & 0.0054 \\ 0.0007 & -0.0000 \\ 0.0000 & 0.0054 \end{bmatrix}.$$
3.4.2 Closed-loop system behavior

The closed-loop system behavior with controller and observer is shown in Figure 3.3, considering no delay in the communication network. The figure shows the behavior of the physical outputs (in blue), their measurements (in dotted black) and their estimations (in dashed green) when a change in the reference input (in red) is introduced. The coupling between the system outputs is clear.

The full-order current observer related residue is shown in Figure 3.4, it can be used to determine the detection threshold. From that figure, the threshold value could be set near to 0.008, but that value generates a lot of false alarms for simulations without attacking the system. That threshold should be set considering different situations, such as modifications of both reference inputs simultaneous or not, in order to avoid false alarms in the detection process. In that sense, after various simulations, we found that for the four tanks system $\tau_D = 0.033$.

![Figure 3.3](image-url)  
**Figure 3.3:** System response with change in reference of Level 1. Top figure shows the behaviour of level 1 and the bottom figure the behavior of level 2. The reference is shown in red, the measured output in blue, the attacked output in dotted black line, and the estimation of the full-order current observer in dashed green.
3.4.3 UIOs bank design

Even though we have not attacked the system sensors yet, we are going to show the design process of the UIOs, as well as their behavior, in order to see their associated residuals and how to set their thresholds. In order to design the UIOs bank, we need a linear discrete-time representation of the system. Such representation is obtained by linearization through the Jacobian, around an equilibrium point defined by \( \bar{h}_i \) and \( \bar{u}_i \) from Table 3.1. The linearized model is discretized, using the zero-order hold technique [Franklin et al., 1997], with \( T = 1 \) s, obtaining a model as in (3.1), with

\[
A = \begin{bmatrix}
0.9842 & 0 & 0.0407 & 0 \\
0 & 0.9890 & 0 & 0.0326 \\
0 & 0 & 0.9590 & 0 \\
0 & 0 & 0 & 0.9672 \\
\end{bmatrix},
\]

\[
B = \begin{bmatrix}
0.0826 & 0.0010 \\
0.0005 & 0.0625 \\
0 & 0.0469 \\
0.0307 & 0 \\
\end{bmatrix},
\]

\[
C = \begin{bmatrix}
0.5 & 0 & 0 & 0 \\
0 & 0.5 & 0 & 0 \\
\end{bmatrix}.
\]

Having the linear discrete-time model, we can start the design process for the UIOs bank. Let us start with UIO 1. First, we need to select the value of the \( E^1 \) matrix, which defines the way the perturbation \( d[k] \) is acting on the system. For this case, we have selected

\[
E^1 = \begin{bmatrix}
10^{-4} & 1 & 10^{-4} & 10^{-4} \\
\end{bmatrix}^T,
\]

which represents that we are considering the major effect of the attack is in state 2, which is related with the only output UIO 1 works with (output 2).
After having $E^1$, we have to satisfy (3.10)-(3.13). From (3.10), we can calculate $H^1$. With $H^1$, we can get $T^1$ from (3.12). In order to find $F^1$, using (3.11), first, we have to verify observability or, at least, detectability of the pair $(A^1_1, C^1)$. The rank of the observability matrix of the pair $(A^1_1, C^1)$ is 1, instead of $n = 4$, which means that the system is not completely observable. Moreover, there is only one observable mode. Then, we need to transform the pair $(A^1_1, C^1)$ to its observability canonical form in order to: i) verify if the nonobservable modes of the system are stable and, if that is true ii) define the closed-loop desired mode for UIO 1. Effectively, in this case, the three nonobservable modes are stable and, therefore, we consider a good location for the only observable mode could be in $z = 10^{-3}$, in order to guarantee that the convergence of the estimation error is faster than the closed-loop system dynamic. Given the above, we find

$$K^1_1 = \begin{bmatrix} -1 & -0.002 & 1 & 1 \end{bmatrix}. \quad (3.23)$$

Once we have found $F^1$, from (3.13) we can calculate $K^1_2$. Finally, we get $K^1 = K^1_1 + K^1_2$, which completes UIO 1 design process.

We follow a similar process to design UIO 2, where

$$E^2 = \begin{bmatrix} 1 & 10^{-4} & 10^{-4} & 10^{-4} \end{bmatrix}^T.$$ 

For this case

$$K^1_2 = \begin{bmatrix} -0.002 & 1 & 1 \end{bmatrix}, \quad (3.24)$$

and the design can be completed as before.

State estimation done by the bank of UIOs is very similar to the one obtained with the full-order current observer. In Figure 3.5, we can see UIOs output estimation versus the full-order current observer one. There it is clear the estimation is satisfactory, from both UIOs. After various simulation, we have defined thresholds in the same way as for the full-order current observer, obtaining $\tau^1_I = 0.0315$ and $\tau^2_I = 0.0420$. In Figure 3.6, we can see the shape of the residues and their associated thresholds.

### 3.4.4 Attack on Level 1 - Impact on the system

Initially, we are considering only an attack in the sensor of level 1. In this case, the attack signal is defined by
Figure 3.5: System full-order current observer and UIOs outputs estimation. Top figure shows the behaviour of level 1 and the bottom figure the behavior of level 2. The estimation of the full-order current observer in dash-dot green, and the estimations of UIO 1 and 2 in dashed cyan and magenta, respectively.

Figure 3.6: Residues using UIOs estimation. UIO 1 related residue in cyan, its threshold in dashed cyan, UIO 2 related residue in magenta and its threshold in dashed magenta.

\[ a_1[k] = \begin{cases} 
0, & k < 600, \\
0.5, & 600 \leq k \leq 1000, \\
0, & 1000 < k. 
\end{cases} \quad (3.25) \]

The shape of the attack is depicted in Figure 3.7.

Since we are going to consider only an attack on output 1, we define \[ a_2[k] = \]
0 for all $k$, and

$$a[k] = \begin{bmatrix} a_1[k] \\ a_2[k] \end{bmatrix}.$$ (3.26)

Therefore $F^a = I_p$.

The closed-loop response system with attack is shown in Figure 3.8, where we can see that the attack effect consists in an increment in the value of level 1. It is interesting to see that, since the attack is pulse shaped, its effect involves a sort of pulse response. That is, the effect of the attack has similar shape to the system response to a step variation of the reference. Because of that, the effect of the attack is longer than the attack itself.

In Figure 3.8, we can also see the estimated outputs by the full-order current observer, as well as for the UIOs. It is important to notice that the full-order observer does not do a good job while the attack is taking place. Also, we can see that UIO 1 estimates converges to what really happens with the physical variable, unknown by the controller (only seeing on-site), while UIO 2 estimates estimates converges to the attacked variable. We can see also that due to the coupling in the system variables, there is also an effect on level 2, of similar duration than the effect on level 1.

Figure 3.9 shows the residues obtained from the full-order current observer in green, from UIO 1 in cyan and from UIO 2 in magenta, all of them compared with their associated thresholds plotted in the same colors in dashed lines. From there, we can see that attack detection will take place between 600 s and 1050 s, while an isolation of the attack for output 1 is achieved between 600 s and 1150 s. The last can be verified in Figure 3.10, with variables $b[k]$ for detection, and $l^1[k]$ and $l^2[k]$ for isolation on each output. It is important to point out that, given there is no simultaneous isolation on both outputs, the result of computing $L^1[k] = l^1[k]$ and $L^2[k] = l^2[k]$. Also, we should remark that isolation in output 1 is longer than
Figure 3.8: System response with attack on sensor 1. Dashed: sensor value. Blue: Physical variable (unknown for the controller). Red: Reference.

Figure 3.9: Residue using full-order estimation, when an attack is applied to output 1.

it should be, which evidences the helpfulness of using full-order current observer for the detection process, since it helps to diminish over correction, that could appears when only using UIOs.

3.4.4.1 Attack on Level 1 - Mitigation

In this subsection, we show the results of using UIOs estimations to recalculate the control action and mitigate the attack. Figure 3.11 shows the value of level 1 output, with the reconfiguration mechanism mechanism activated. Notice the difference
3.4. Numerical Results - Four Tanks System

Figure 3.10: Detection and isolation variables. The detection signal from the full-order current observer is shown in green, the isolation in output 1 from UIO 1 in dashed cyan and, the isolation in output 2 from UIO 2 in dashed magenta.

with Figure 3.8, where the deviation for level 1 was even higher than the input change. Whereas in Figure 3.11 the deviation is even smaller than the overshoot. Therefore, we can see that the mitigation scheme proposed it indeed helps.

Figure 3.11: System response with mitigation of an attack on sensor 1.

Regarding to output 1 estimation, when the mitigation process is active, we obtain similar results to the ones in the previous section. For the full-order current observer, the estimation lies between the physical value and the mitigated state, which becomes the system state. For UIO 1, the estimation converge to the mitigated state (since the attack is in the output related to this variable and, when
detected and isolate, they are the same). For UIO 2, as before, the estimation converge to the physical variable.

Finally, in Figure 3.12 control signals for the system without attack are shown in blue, with attack in both outputs en red and with attack with reconfiguration in dashed black. Notice the similarity between the control actions for the system without attack and reconfiguration, where it is clear the usefulness of the proposed mechanism. However, it is also important to notice the difference after the attack, due to the drastic modification that this kind of attack imposes both, when it starts and when it finishes.

![Figure 3.12: Control signals for the system without attack in blue, with attack in red and with attack and mitigation in dashed black. Top figure shows $u_1[k]$ and bottom figure shows $u_2[k]$.

3.4.5 Attacks on both levels - Impact on the system

Before even considering the attacks, in Figure 3.13 we show the closed loop behavior of the system, where in this case the simulation lasts 3000 s and there are changes in both reference inputs. In the figure, we can see the same characteristics of the closed-loop behavior that we commented previously for both the system and the observers.
Let us now consider attacks on both output sensors at different times. In this case, the attack signal is defined by

$$a_i[k] = \begin{cases} 
0, & k < t_{1i}, \\
-\frac{1}{m_i} (k - t_{1i}), & t_{1i} \leq k < t_{2i}, \\
-1 + \frac{1}{m_i} (k - t_{2i}), & t_{2i} \leq k \leq t_{3i}, \\
0, & t_{3i} < k.
\end{cases} \quad (3.27)$$

For $a_1[k]$, $m_1 = 200$, $t_{11} = 600s$, $t_{21} = 800s$, $t_{31} = 1000s$, and for $a_2[k]$, $m_2 = -200$, $t_{12} = 2000s$, $t_{22} = 2200s$, $t_{32} = 2400s$.

Therefore $a[k] = [a_1[k] \ a_2[k]]^\top$, and $F^a = I_p$.

The closed-loop response system with attack is shown in Figure 3.14, where we can see that each attack mainly affects one output, i.e., $\tilde{y}_1^a[k]$ is affected directly by $a_1[k]$. However, due to the coupling in the system variables, is also visible the attack effect in the other output.
Figure 3.14: System response with attacks on both sensors. Top figure shows the behaviour of level 1 and the bottom figure the behavior of level 2. The reference is shown in red, the measured output in blue, the attacked output in dotted black line, the estimation of the full-order current observer in dashed green, and the estimations of UIO 1 and 2 in dashed cyan and magenta, respectively.

Figure 3.14, shows also the estimated output variables for the attacked system by the full-order current observer, UIO 1 and UIO 2 in dashed green, cyan and magenta lines, respectively. As we can see, the estimated of the full-order observer is not good during either of the attack times. Notice that UIO 1 estimates output 1 coinciding with the real measurement (blue line), while output 2 trends to the sensor value. Something similar happens for the UIO 2, where the estimation of output 1 coincides with the sensor value, while the estimations of output 2 trends to the real measurement. What we see in Figure 3.14 is exactly what was expected from the UIOs.

Figure 3.15 shows the residues obtained from the full-order current observer in green, from UIO 1 in cyan and from UIO 2 in magenta, all of them compared with their associated thresholds plotted in the same colors in dashed lines. From there, we can see that:

- Attack detection will take place between 610 s and 1020 s, with a discontinuity between 919 s and 968 s, and between 2004 s and 2411 s, with a discontinuity between 2380 s and 2397 s.
3.4. Numerical Results - Four Tanks System

• Isolation of the attack on output 1 is achieved between 610 s and 1150 s, also with a discontinuity between 960 s and 980 s.
• Isolation of the attack on output 2 is achieved between 2004 s and 2603 s, also with a discontinuity between 960 s and 980 s.

The last can be verified in Figure 3.16, with variables \( b[k] \) for detection, and \( l^1[k] \) and \( l^2[k] \) for isolation on each output. It is important to point out that, similar to the previous case of only attack in output 1, given there is no simultaneous isolation on both outputs, the result of computing \( L^1[k] = l^1[k] \) and \( L^2[k] = l^2[k] \). Again, we remark that isolation is longer than it should be, and the full-order current observer for the detection process helps to identify closer the duration of the attacks.

3.4.5.1 Attacks on both levels - Mitigation

Let us show the results of using UIOs estimations to recalculate the control action and mitigate the attacks. Figure 3.17 shows the outputs with the reconfiguration mechanism activated. Notice the difference with Figure 3.14, where the deviation...
for the outputs with attacks was higher than the one due to the input change. Whereas in Figure 3.17 the deviation is even smaller than the overshoot. Therefore, we can see that the mitigation scheme proposed it indeed helps. Notice that the same observations to each estimation in the previous subsection holds, UIO $i$ estimates very closely output $i$ and the full-order current observer lies in the middle, but helps to avoid corrections for longer times.

![Figure 3.17: System response with mitigation of attacks on both sensors. Top figure shows the behaviour of level 1 and the bottom figure the behavior of level 2. The reference is shown in red, the measured output in blue, the attacked output in dotted black line, the estimations of full-order current observer, UIO 1 and 2 in dashed green, cyan and magenta, respectively.](image)

Finally, in Figure 3.18 control signals for the system without attack are shown in blue, with attack in both outputs in red and with attack with reconfiguration in dashed black. Notice, as in the case with only one attack on level 1, the similarity between the control actions for the system without attack and reconfiguration, where it is again clear the usefulness of the proposed mechanism. It is worthwhile to emphasize that, given the nature of the attacks, when they finish the control action is slightly different than from the one when there is no attack from the system; different from the step type attacks, where when the attack finishes, the mitigated control action shows quite an effort for the control system to overcome the effects post-attack.

In this chapter, we have studied the problem of defending low-level controllers
based on state-feedback against sensor attacks. We show that, for an attacked system with only one simultaneous attack in place, using a bank of unknown input observers, we are able to recover the complete state of the system without the effect of the attack and, therefore, we are able to recover the output as well. We also show how the control action can be re-computed with the uncorrupted information, once the attack has been detected and isolated. The proposed mechanism was tested on an existing control system for the four tank system testbed, with no simultaneous attacks on the sensors of the system outputs, with satisfying results. The results in this work show a way to improve the resilience of low-level controllers to make them suitable for more sophisticated mechanisms such as secure estimation, where is assumed that the low-level controller is secure.
Chapter 4

Stability Analysis of Attacked Cyber-Physical Systems

In the previous chapter we showed the effectiveness of the mitigation mechanism proposed in Section 2.3 for the noiseless linear case. In this chapter, we continue doing the same, considering, not only additive attacks, but also multiplicative attacks, which depend proportionally on the state variables. We analyze how this simple false data injection attacks can affect system stability and how the mitigation mechanism proposed can enlarge a little bit the system stability range, but cannot prevent the system to collapse. We show numerical results with the three tanks system benchmark that supports the theory presented.

4.1 Attacked System

Let us consider two different possibilities of false data injection attacks on sensors [Teixeira et al., 2015], an additive one and a multiplicative one, in order to study the stability of the attacked system and to conclude regarding which one would be more harmful.

4.1.1 Additive attack

We first consider closed-loop system working with a controller and an observer, described by

\[
\begin{align*}
x[k+1] &= A x[k] + B \tilde{u}[k], \\
y[k] &= C x[k], \\
\tilde{y}_a[k] &= \tilde{y}[k] + F_a a[k], \\
\tilde{x}[k] &= A \tilde{x}[k-1] + B u[k-1], \\
\hat{x}[k] &= \hat{x}[k] + L (\tilde{y}_a[k] - C \tilde{x}[k]), \\
v[k+1] &= y_r[k] - \tilde{y}_a[k] + v[k], \\
u[k] &= -K_r v[k] - K_S \hat{x}[k],
\end{align*}
\]

with the same variables definition as in Section 3.1.
Let us consider the network has no perceptible effects on the signals whatsoever, that is \( \tilde{u}[k] = u[k] \), \( \tilde{y}[k] = y[k] \) and \( \tilde{y}_a[k] = y_a[k] \). Therefore, the state equation for the control loop can be obtained from the system defined by (4.1a) with the control law in (4.1f)-(4.1g) and the attacked output in (4.1c), defining an extended state vector composed by the state variables and the integrator variables. That is

\[
\begin{bmatrix}
x[k+1] \\
v[k+1]
\end{bmatrix} = \begin{bmatrix}
A - BK & -BK_i \\
-C & I
\end{bmatrix} \begin{bmatrix}
x[k] \\
v[k]
\end{bmatrix} + \begin{bmatrix}
0 \\
I
\end{bmatrix} y^r[k] + \begin{bmatrix}
0 \\
-F_a
\end{bmatrix} a[k].
\] (4.2)

Notice that the dynamic matrix of the equation has no terms related with the attack. In fact, the attack signal acts as an external input. Therefore, it is clear that this kind of attack does not affect the stability of the control loop, since the dynamic matrix remains the same as when there is no attack considered.

Let us consider the state equation of the observer loop that can be obtained from the system defined in (4.1a), with the observer in (4.1d)-(4.1e) and the attacked output in (4.1c). It can be written as

\[
\dot{x}[k] = (A - LC A) \dot{x}[k-1] + (B - LC B) u[k-1] + L y[k] + L F_a a[k].
\] (4.3)

Notice, again, that the dynamic matrix of the equation has no terms related with the attack and it acts as an external input. That is, \( a[k] \) does not affects the stability of the observer loop.

We can conclude from the prior analysis that additive attacks, where the attack signal is external, do not affect the stability of neither the control nor the observer loops and, therefore, do not affect overall system stability.

### 4.1.2 Multiplicative attack

Let us consider the closed-loop control system of the previous section, disturbed with a sensor attack proportional to the state vector. That is, the same set in (4.1a)-(4.1g), but instead of (4.1c), we have

\[
\tilde{y}_a[k] = C_m x[k] + C_a x[k],
\] (4.4)

where \( C_m = C \), which corresponds to sensor gains, \( \tilde{y}_a[k] \in \mathbb{R}^p \) represents a multiplicative attack on the output signal, of a tracking feedback control system, increasing its value in a proportion determined by \( C_a \) (after passing through the network), i.e., only one output at the time. The structure for \( C_m \) corresponds to consider each output associated with a single measured state variable. That is, let
us consider sensor gains $k_i$, for $i = 1, 2, \ldots, p$, for a system with $p$ outputs, each of them related to an output. Then, without loss of generality, we consider that the sensors are related to the first $p$ state variables (which can be easily arranged with an order modification of the state variables, through a basic transformation). Therefore, the structure for $C_m$ can be written as

$$C_m = [\text{diag}[k_1, k_2, \ldots, k_p] \ 0_{p \times (n-p)}].$$

Since we consider only one attack simultaneously, $C_a$ can be considered to have in some time window a similar structure. That is

$$C_a = [\text{diag}[0, \ldots, 0, \alpha_i, 0, \ldots, 0] \ 0_{p \times (n-p)}],$$

where $\alpha_i$ is the constant value that represents the attack on the $i^{th}$ output. With the previous structure for $C_a$, it is easy to see that the attack on the $i^{th}$ output consists in modifying the measurement by the $i^{th}$ sensor in a proportional fashion. That is

$$\tilde{y}_i^a = k_i x_i + \alpha_i x_i. \quad (4.5)$$

Now, let us show how this kind of attack affects the stability of the control and observer loops. In order to do that, we consider, as in the previous case, the network has no perceptible effects on the signals whatsoever, that is $\tilde{u}[k] = u[k]$, $\tilde{y}[k] = y[k]$ and $\tilde{y}_a[k] = y_a[k]$.

For the control loop, the state equation can be obtained in the same way as we obtained (4.2), but with the attack described as in (4.4). Then, the state equation for the control loop can be written as

$$\begin{bmatrix} x[k+1] \\ v[k+1] \end{bmatrix} = \begin{bmatrix} A - B_c K_1 & -B_c K_2 \\ -C_m - C_a & I \end{bmatrix} \begin{bmatrix} x[k] \\ v[k] \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} y^r[k]. \quad (4.6)$$

Notice that different from the additive case, the value of $C_a$ affects the dynamic matrix of the system and, therefore, the stability of the control loop.

For the observer loop, the state equation can be obtained as we obtain (4.3), but with the attack described as in (4.4). Then, the state equation for the observer loop can be written as

$$\dot{x}[k] = [A - L (C_m + C_a) A] \dot{x}[k-1] +$$

$$+ [B_c - L (C_m + C_a) B_c] u[k-1] + L \tilde{y}_a[k], \quad (4.7)$$

where, again, the value of $C_a$ affects the system dynamic matrix and the stability of the observer loop.
After analyzing the stability effects of additive and multiplicative attacks, we can conclude that an attack as an external input does not affect system stability, whereas an attack proportional to the state may destabilize the system, as we will show in the next section.

4.2 Stability Analysis

In the previous section we showed that additive attacks do not affect system stability, since the attack values do not affect the dynamic matrix of neither the control nor the observer loops. Also, we showed that multiplicative attacks do modify the dynamic matrix of both the control and the observer loops, therefore, it can modify the stability of the closed-loop system. In this section, we use Lyapunov’s second method for discrete-time systems and, with a parameterization of the attack, we come up with a way of finding conservative bounds on the attacks values to guarantee closed-loop asymptotic stability of the system.

4.2.1 Quadratic Lyapunov Stability for Discrete-time Systems

First, we are going to consider the classic Lyapunov’s second method to show that the system is asymptotically stable. The next theorem establishes the conditions for the existence of a Lyapunov function, associated with the system state, and the asymptotic stability of the equilibrium point.

**Theorem 4.2.1** Consider a system of the form

\[ x[k + 1] = A \, x[k]. \]

Suppose there exists a scalar function \( V(x) > 0 \), continuous in \( x \) such that

1. \( V(x) > 0 \) for \( x \neq 0 \).
2. \( V(0) = 0 \).
3. \( V(\infty) \to \infty \) as \( ||x|| \to \infty \).
4. \( \Delta V(x) < 0 \) for \( x \neq 0 \), where

\[ \Delta V(x[k]) = V(x[k + 1]) - V(x[k]). \]

Then, the equilibrium state \( x^* = 0 \) is asymptotically stable in the large and \( V(x) \) is a Lyapunov function.
Let us consider the following Lyapunov function candidate,

\[ V(x[k]) = x^\top[k] P x[k], \]

(4.8)

where \( P \) is a positive definite and symmetric Hermitian matrix. Then

\[
\Delta V(x[k]) = V(x[k+1]) - V(x[k]) = x^\top[k+1] P x[k+1] - x^\top[k] P x[k] \\
= (A x[k])^\top P A x[k] - x^\top[k] P x[k] \\
= x^\top[k] (A^\top P A - P) x[k].
\]

For asymptotic stability we require that \( \Delta V(x[k]) < 0 \). Therefore,

\[
\Delta V(x[k]) = x^\top[k] (A^\top P A - P) x[k] < 0.
\]

(4.9)

For this equation to be satisfied, we need to solve the following linear matrix inequality (LMI)

\[
A^\top P A - P < 0.
\]

(4.10)

### 4.2.2 Quadratic Lyapunov Stability for the Attacked System

We need to check the stability for both, the dynamic matrix from the discrete-time closed-loop system and the observer. In order to do that, we use Lyapunov theory, which for this kind of system establishes that the LMI in (4.10) should be satisfied. For the tracking feedback closed-loop system, from (4.6) we can see that

\[
\bar{A} = \begin{bmatrix}
A - B c K_1 & -B c K_2 \\
-C_m - C_a & I
\end{bmatrix},
\]

and, for the observer, from (4.7) we can identify

\[
\bar{A} = A - L(C_m + C_a) A,
\]

each of them designed to be stable for \( C_a = 0 \).

Before introducing the main result of this work, let us introduce a lemma that will help to prove our result [Petersen and Hollot, 1986].

**Lemma 4.2.2** Let \( S \) and \( Z \) be \( q \times q \) symmetric positive-semidefinite matrices and
S a \( q \times q \) symmetric negative-semidefinite matrix. Suppose further that
\[
(w^\top Y w)^2 - 4w^\top S w w^\top Z w > 0,
\]
for all \( w \neq 0 \in \mathbb{R}^q \). Then \( \varepsilon^2 S + \varepsilon Y + Z < 0 \), for some \( \varepsilon > 0 \).

**Theorem 4.2.3 (General Stability)** Let us assume that the system is described by
\[
x[k + 1] = \tilde{A} x[k],
\]
where \( \tilde{A} = \bar{A}_n + \Delta \bar{A} \) and the matrix \( \Delta \bar{A} \) is decomposed as a bounded norm uncertainty, i.e.,
\[
\Delta \bar{A} = \gamma DF E.
\]
\( F \) represents the real unknown parameter, in this case the attack, that satisfies \( F^\top F \leq 1 \) and, \( D \) and \( E \) represent how the unknown value affects \( \bar{A} \).

The equilibrium state of the system in (4.12) is stable if and only if there exist a symmetric positive definite matrix \( P \) and positive scalars \( \alpha > 0 \) and \( \varepsilon > 0 \) such that
\[
\min \alpha \\
\text{s.t.} \\
\begin{bmatrix}
-P & \bar{A}_n^\top P & 0 & E \\
-P & \bar{A}_n^\top P & 0 & E \\
0 & PD^\top & -\varepsilon I & 0 \\
E^\top & 0 & 0 & -\alpha I \\
\end{bmatrix} < 0,
\]

or, equivalently, if and only if there exist a symmetric positive definite matrix \( P \) and positive scalars \( \beta > 0 \) and \( \varepsilon > 0 \) such that
\[
\max \beta \\
\text{s.t.} \\
\beta > 0, \\
\begin{bmatrix}
-P + \beta E^\top E & \bar{A}_n^\top P & 0 \\
-P & DP & 0 \\
0 & PD^\top & -\varepsilon I \\
\end{bmatrix} < 0.
\]

**Proof:** Using the Schur complement [Boyd et al., 1994], (4.10) can be written as
\[
\begin{bmatrix}
-P & \bar{A}^\top P \\
P \bar{A} & -P \\
\end{bmatrix} < 0,
\]
4.2. Stability Analysis

By hypothesis, we will consider $\bar{A}$ as a matrix with a bounded nominal uncertainty, that is $\bar{A} = \bar{A}_n + \Delta \bar{A}$, this representation is the more general one. For the controller we will have that we must have

$$\bar{A}_n = \begin{bmatrix} A - B, K_1 & -B, K_2 \\ -C_m & I \end{bmatrix},$$

with

$$\Delta \bar{A} = \begin{bmatrix} 0_{n \times n} & 0_{n \times m} \\ -C_a & 0_{m \times m} \end{bmatrix}.$$  

(4.17)

For the observer loop, we will have that

$$\bar{A}_n = A - L C_m A,$$

(4.19)

with

$$\Delta \bar{A} = -L C_a A.$$  

(4.20)

Then,

$$\begin{bmatrix} -P & (\bar{A}_n + \Delta \bar{A})^\top P \\ P (\bar{A}_n + \Delta \bar{A}) & -P \end{bmatrix} < 0.$$  

By (4.13), this corresponds to

$$\begin{bmatrix} -P & (\bar{A}_n + \gamma D F E)^\top P \\ P (\bar{A}_n + \gamma D F E) & -P \end{bmatrix} < 0.$$  

Equivalently

$$\begin{bmatrix} -P & \bar{A}_n^\top P \\ P \bar{A}_n & -P \end{bmatrix} + \begin{bmatrix} 0 & \gamma E^\top F^\top D^\top P \\ \gamma P D F E & 0 \end{bmatrix} < 0,$$

which can also be written in quadratic form as

$$X^\top \begin{bmatrix} -P & \bar{A}_n^\top P \\ P \bar{A}_n & -P \end{bmatrix} X + X^\top \begin{bmatrix} 0 & \gamma E^\top F^\top D^\top P \\ \gamma P D F E & 0 \end{bmatrix} X < 0,$$

for all $X \neq 0$. Since we know that $F^\top F \leq 1$, we can write

$$X^\top \begin{bmatrix} -P & \bar{A}_n^\top P \\ P \bar{A}_n & -P \end{bmatrix} X < \max \left\{ X^\top \begin{bmatrix} 0 & \gamma E^\top F^\top D^\top P \\ \gamma P D F E & 0 \end{bmatrix} X : F^\top F \leq 1 \right\},$$
which left hand side is negative. Then, squaring on both sides of the inequality, we have
\[
\left( X^T \begin{bmatrix} -P & \bar{A}_n^T P \\ P \bar{A}_n & -P \end{bmatrix} X \right)^2 > 
\left( \max \left\{ X^T \begin{bmatrix} 0 & \gamma E^T F^T D^T P \\ \gamma P D F E & 0 \end{bmatrix} X : F^T F \leq 1 \right\} \right)^2.
\] (4.21)

In order to rewrite the right hand side of (4.21), we express
\[
X = \begin{bmatrix} x^T \\ y^T \end{bmatrix}^T,
\]
obtaining
\[
\left( \begin{bmatrix} x^T & y^T \end{bmatrix} \begin{bmatrix} 0 & \gamma E^T F^T D^T P \\ \gamma P D F E & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \right)^2 = 4(\gamma^2 y^T P D F E x)^2.
\] (4.22)

Using the triangle inequality, we have
\[
4(\gamma^2 y^T P D F E x)^2 \leq 4 y^T P D D^T P y \gamma^2 x^T E^T E x,
\] (4.23)
which can be rewritten as
\[
4 \gamma (y^T P D F E x)^2 \leq 4 X^T \begin{bmatrix} \gamma^2 E^T E & 0 \\ 0 & 0 \end{bmatrix} X X^T \begin{bmatrix} 0 & 0 \\ 0 & P D D^T P \end{bmatrix} X,
\] (4.24)
which implies that the right hand side is the maximum value that we are looking for. Therefore, we can rewrite (4.21) as
\[
\left( X^T \begin{bmatrix} -P & \bar{A}_n^T P \\ P \bar{A}_n & -P \end{bmatrix} X \right)^2 + 4 X^T \begin{bmatrix} \gamma^2 E^T E & 0 \\ 0 & 0 \end{bmatrix} X X^T \begin{bmatrix} 0 & 0 \\ 0 & P D D^T P \end{bmatrix} X > 0.
\] (4.25)

Using Lemma 1,
\[
\epsilon \begin{bmatrix} \gamma^2 E^T E & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -P & \bar{A}_n^T P \\ P \bar{A}_n & -P \end{bmatrix} + \frac{1}{\epsilon} \begin{bmatrix} 0 & 0 \\ 0 & P D D^T P \end{bmatrix} < 0.
\]
Adding up the matrices, we have
\[
\begin{bmatrix} -P + \epsilon \gamma^2 E^T E & \bar{A}_n^T P \\ \bar{A}_n E & -P + \epsilon^{-1} P D D^T P \end{bmatrix} < 0,
\] (4.26)
which needs to be solved for \( \epsilon \), \( \gamma \) and \( P \) and, therefore, it is not an LMI. Finally,
applying the Schur complement to each of the principal diagonal elements and defining $\alpha^{-1} = \varepsilon \gamma^2$, the problem of finding the upper limit for the attack in order the complete closed-loop system (controller together with observer) remains stable can be formulated as (4.14). If we only apply the Schur complement to the term $-P + \varepsilon^{-1}PD^TP$ in (4.26), the problem can be formulated as (4.15), with $\beta = \varepsilon \gamma^2$.

\[
\begin{bmatrix}
0.9899 & 0.0005 & 0.0098 \\
0.0004 & 0.9804 & 0.0095 \\
0.0108 & 0.0107 & 0.9784
\end{bmatrix},
\]

\[
\begin{bmatrix}
60.1584 & 0.1660 \\
-0.3848 & 60.1895 \\
0.4138 & 0.1935
\end{bmatrix},
\]

and

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix}.
\]

The proposed control for this system given in \cite{H. Noura and Chamseddine, 2009} is a discrete-time controller as the one in the last two lines of (3.5), and feedback gains given by

\[
K_S = \begin{bmatrix} 21.6 & 3 & -5 \\ 2.9 & 19 & -4 \end{bmatrix} \times 10^{-4},
\]

and

\[
K_I = \begin{bmatrix} -0.95 & -0.32 \\ -0.30 & -0.91 \end{bmatrix} \times 10^{-4}.
\]
In order to implement the control law, we design a full order current observer as in lines fourth and fifth of (3.5), with

\[
L = \begin{bmatrix}
0.9899 & 0.0005 & 0.0098 \\
0.0004 & 0.9894 & 0.0095 \\
0.0108 & 0.0107 & 0.9784
\end{bmatrix}.
\]

The behavior of the closed-loop system is shown in Figure 4.2.

Now, we verify that additive attacks do not affect system stability.
4.3.1 Additive attacks

As we have mentioned before, we are going to show the effects of an additive attack on one of the sensors, in this case on output 2. In this case, we are considering $a[k] = [a_1[k] \ a_2[k]]^t$. Since we are only considering an attack on output 2, $a_1[k] = 0$, while $a_2[k]$ is defined as in (2.9), with $t_1 = 900s$ and $t_2 = 1300s$ are the initial and final times of the attack, $f_i$ is the function that shapes the $i^{th}$ attack itself, in this case a pulse (between $t_1$ and $t_2$) of amplitude $a$.

In Figure 4.3 we can see the effect of the attacks defined by (2.9) for different values of $a$. There, we can notice that, no matter the sign or the magnitude of the additive attack, the shape of the attack effect on the outputs is the same and it never compromises system stability, as it was shown in the analysis in Subsection 4.1.1. Also, it is important to mention that there can be attack magnitudes that will take out the variables from its feasible vales, and that can cause malfunction of the closed-loop system. However, that is not an implication related with stability.

![Figure 4.3: Response of the closed-loop control system with additive attacks of magnitude a.](image-url)
### 4.3.2 Multiplicative attacks

In this case, for $k$ when there is attack $C_a = 0$. Notice that we do not consider attacks that last the complete simulation time.

$$C_a = \begin{bmatrix} 0 & 0 \\ 0 & b \end{bmatrix}, \quad (4.27)$$

between $t_1$ and $t_2$, for the same duration of additive attacks, in order to facilitate the comparison.

Now, we are going to use Theorem 4.2.3 to find the stability value interval for $b$. We have to analyze two systems to find such interval, the controller loop and the observer loop.

Let us first consider the controller loop, with $\bar{A}_n$ as in (4.17) and, with $C_a$ defined as in (4.27), we have

$$\Delta \bar{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & b & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} \underbrace{0}_{D} \underbrace{1}_{\sqrt[\gamma F]} \underbrace{0}_{E} \underbrace{0}_{F} \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix},$$

where we can see the parameterization of $\Delta \bar{A}$ as in (4.13). Solving Problem 1b we find that $b \leq 0.8872$.

Now, considering the observer loop, with $\bar{A}_n$ as in (4.19) and, with $C_a$ defined as in (4.27), we have

$$\Delta \bar{A} = -LC_aA = -A(2,:) \underbrace{b}_{D} \underbrace{L(:,2)}_{E},$$

where the notation $A(2,:)$ represents the second row of $A$, and $L(:,2)$ represents the second column of $L$. Solving Problem 1b we find that $b \leq 0.7203$. Therefore, given the separation principle [Phillips and Nagle, 1995, Ogata, 1987], we can guarantee the stability of the system if both loops (controller and observer) are stable. That is, the system is going to be stable for attacks with $b \leq 0.7203$. Equivalent results were found solving Problem 1a.

In Figure 4.4 we can see the effect of the attacks defined in (4.27) for different values of $b$, where we can notice different system behavior depending on the sign.
and magnitude of the attack. For instance, for negative attacks, the larger the attack the larger the response peak time and the overshoot remains approximately the same. On the other hand, for positive attacks, the larger the attack amplitude the larger the overshoot up to achieving an unstable system (see Figure 4.7). For this case, using simulation, we find $b = 0.735$ for critically stable system; a higher value than the one obtained solving Problem 1b, as expected, since the values obtained from that approach are more restrictive.

![Figure 4.4: Response of the closed-loop control system with multiplicative attacks of magnitude $b$.](image)

### 4.3.3 Mitigation process

The mitigation process utilized here is done as in Section 3.3. Therefore, for UIO1 we have

$$E^1 = \begin{bmatrix} 10^{-4} & 1 \end{bmatrix}^\top,$$

in order to decouple the influence of sensor 1 in the system state, to be able to estimate the state only with sensor 2 information. After the decoupling transformation is done on the system, the transformed resulting system turns out to have only one observable mode. Therefore, only one of the close loop UIO1 mode can be located, and we chose to locate it at $p_d = 0.001$. Given that the non observable modes of the UIO are located at 0.9957 and 0.9707, the closed-loops poles of UIO1

$$\begin{bmatrix} 10^{-4} & 1 \end{bmatrix}^\top.$$
will be located at 0.9957, 0.9707 and 0.0010.

Doing something similar for UIO2, we have

\[ E^2 = \begin{bmatrix} 1 & 10^{-4} & 10^{-4} \end{bmatrix}^\top. \]

Similar to the case of UIO1, after performing the decoupling transformation, we found again only one observable mode for the resulting system. That mode will be located at \( p_d = 0.001 \). The remaining non observable modes are located at 0.9667 and 0.9890, since they are inside the unit circle (they are stable) the UIO can be designed. The modes of UIO2 will be located at 0.9667, 0.9890 and 0.0010.

Attack detection, isolation and mitigation processes are done as in Section 3.3. Figures 4.5 and 4.6 show the result of mitigate the attacks shown in Figures 4.3 and 4.4. There we can see that the attack effect on the system has been reduce. Interestingly enough, for the mitigation of both kinds of attacks we get sort of pulse responses in both sensors; obviously, for positive multiplicative attacks we can notice longer oscillations (since it is affecting directly system stability). Also, we can notice that the effect on the non attacked sensor is shorter than for the attacked one, whereas for the additive attacks the mitigated effect lasts almost the same in both sensors, only the magnitude of the overshoot is depending on the magnitude of the attack.

![Figure 4.5](image)

**Figure 4.5:** Response of the closed-loop control system with mitigated additive attacks of magnitude \( a \).
4.4. Appendix

Finally, we show in Figure 4.7 an attack with magnitude bigger than the stability limit (specifically $b = 0.8$), where we can see classic unstable behavior for the attacked system without mitigation, where there are increasing oscillations up to the system collapses. Notice that, in this case, the mitigation of the attack diminish slightly the oscillations amplitude, allowing the system to work for a little bit longer, but ending up collapsing. In any case, we can see that both kinds of attacks can be mitigated, but in the case of multiplicative attacks the mitigation is only possible when the attack magnitude is inside the range that allows the system to keep stability.

4.4 Appendix

Proof: [Lemma 4.2.2] Let us start noticing that (4.11) holds for all $w$ and that the unit ball in $\mathbb{R}^q$ is compact. Since the left side of (4.11) is continuous in $w$, we can write

$$0 < \eta_1 \triangleq \min \left\{ (w^T Y w)^2 - 4w^T S w w^T Z w : \|w\| = 1 \right\}.$$
Let $\eta_2$ be a positive scalar such that

$$\eta_2^2 + \eta_2 (\lambda_{\text{max}}[S] + \lambda_{\text{max}}[Z]) < \eta_1/8.$$ 

Since $\lambda_{\text{max}}[S] + \lambda_{\text{max}}[Z] \geq 0$, the left hand side of the inequality vanishes for $\eta_2 = 0$ and has a non-negative slope. Therefore, we can conclude that such $\eta_2$ exists. Let us define symmetric positive definite matrices

$$\bar{S} \triangleq S + \eta_2 I \quad \text{and} \quad \bar{Z} \triangleq Z + \eta_2 I.$$ 

Now, let us show that

$$(w^T Y w)^2 - 4w^T S w w^T Z w > 0 \quad (4.28)$$

for all $w \neq 0 \in \mathbb{R}^q$.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{closed_loop_response.png}
\caption{Response of the closed-loop control system with multiplicative attack of magnitude $b = 0.8$ with and without mitigation.}
\end{figure}
From the definitions of $\eta_1$, $\bar{S}$ and $\bar{Z}$, we have
\[
\eta_1 ||w||^4 \leq (w^T Y w)^2 - 4w^T S w w^T Z w \\
= (w^T Y w)^2 - 4w^T (S - \eta_2 I) w w^T (Z - \eta_2 I) w \\
= (w^T Y w)^2 - 4w^T S w w^T Z w + 4 \eta_2 ||w||^2 (w^T \bar{S} w + w^T \bar{Z} w) + 4 \eta_2 ||w||^4,
\]
for all $w \neq 0 \in \mathbb{R}^q$. Using the definition of $\eta_2$
\[
4 \eta_2 ||w||^2 (w^T \bar{S} w + w^T \bar{Z} w) \\
= 4 \eta_2 ||w||^2 (w^T S w + w^T Z w + 2 \eta_2 ||w||^2) \\
\leq 4 \eta_2 ||w||^4 (\lambda_{\text{max}}[S] + \lambda_{\text{max}}[Z] + \eta_2) \\
= 4 ||w||^4 (\eta_2^2 + \eta_2 (\lambda_{\text{max}}[S] + \lambda_{\text{max}}[Z])) \\
< \frac{\eta_1}{2} ||w||^4.
\]
From (4.29) and (4.31), we conclude that
\[
(w^T Y w)^2 - 4w^T \bar{S} w w^T \bar{Z} w + \eta_1 ||w||^4 \geq \frac{\eta_1}{2} ||w||^4.
\]
Therefore, (4.28) holds.

From Theorems (12.5) and (13.1) in [Gohberg and Rodman, 2009], we can conclude that $\varepsilon^2 S + \varepsilon Y + Z < 0$ for some $\varepsilon > 0$. As a result, for any $w \neq 0 \in \mathbb{R}^q$
\[
0 > w^T (\varepsilon^2 \bar{S} + \varepsilon \bar{Y} + \bar{Z}) w \\
= w^T (\varepsilon^2 S + \varepsilon Y + Z) w + 2 \eta_2 ||w||^2 \\
> w^T (\varepsilon^2 S + \varepsilon Y + Z) w.
\]
In this chapter, we present an extension of the mechanism proposed in the Chapter 2 to mitigate integrity cyber-attacks on ICS. We extend the work in Chapter 3 in two ways: i) the addition of measurement noise to the readings of the sensors, leading to the use of Kalman filters for optimal state estimation; and ii) the inclusion of optimal disturbances decoupling observers (ODDOs), as the isolation mechanism, with the addition of a false alarms reduction mechanism.

Adding noise is a more realistic example of ICSs, and it is an extension with respect to previous works (e.g., [Cómbita et al., 2017], [Combita et al., 2018], [Fawzi et al., 2014]), where only noiseless scenarios have been considered.

Several attacks have been explored, and the evaluation of our proposal is based on a key performance index such as the integral of the absolute error (IAE), which shows that lower values of IAE are obtained using our proposal instead of the conventional way that reacts to the false alarms.

### 5.1 Existing System Setup

For this case, the linear discrete-time invariant model of the plant is given by

\[ x[k+1] = Ax[k] + Bu[k] + Ed[k] + \zeta[k] \]
\[ y[k] = Cx[k] + \eta[k]. \]

(5.1)

Similar to Chapter 3, we consider the disturbances \( d[k] \) to represent the modification on the state due to the change of the control signal by a sensor attack and, given that we assume there is only one attack acting simultaneously, we consider such modifications can be represented by a single signal. Therefore, the matrix \( E \) represents the unattacked outputs acting on the system state.

In the same way, the remote controller is designed to produce disturbance rejection and zero-steady state error for step inputs and, as in the linear noiseless case, we
use integrators working together with state-feedback. The controller is defined as in (3.2), (shown next for reading easiness)

\[ v[k] = y^r[k] - \tilde{y}[k] + v[k - 1], \]
\[ u[k] = K_I v[k] - K_S \hat{x}[k], \]

As in the case of Chapter 3, the state-feedback requires the estimation of the state variables, from the available measurements. In this case, that also considers noise, for this purpose a Kalman filter is used.

The Kalman filter provides the optimal estimate state \( \hat{x}[k] \). From the initial estimation of the associated error covariance matrix, \( P[k | k - 1] \), the Kalman gain is computed as

\[ K[k] = P[k | k - 1] C^T (C P[k | k - 1] C^T + R)^{-1}. \] (5.2)

After this, an update of the state estimation -using the measurement vector- and of the covariance error matrix is done

\[ \hat{x}[k] = \hat{x}[k | k - 1] + K[k] (\tilde{y}[k] - C \hat{x}[k | k - 1]), \]
\[ P[k] = P[k | k - 1] - K[k] C P[k | k - 1]. \] (5.3)

Finally, the state estimation and the covariance error matrix is given by

\[ \hat{x}[k + 1 | k] = A \hat{x}[k] + B u[k], \]
\[ P[k + 1 | k] = A P[k] A^T + Q. \] (5.4)

As it can be noticed from (5.2), (5.3), and (5.4), the state estimate \( \hat{x}[k] \) obtained using the Kalman filter is computed using the information from all inputs \( u[k] \) and all outputs \( \tilde{y}[k] \).

### 5.2 Optimal Disturbance Decoupling Observers for the reconstruction of outputs without the effect of the attack

Let us consider the closed-loop control system of the previous section with model as in (5.1), controller as in (3.2) and the Kalman filter defined as in (5.2) - (5.4). The system is disturbed with a sensor attack (after passing through the network)
as

\[ \tilde{y}_a[k] = \tilde{y}[k] + F_a a[k] + \eta[k]. \] (5.5)

We are interested in estimating the output signals without the effect of the attack, in order to compensate the control action signal and to prevent the attack from feeding back the system and making it to collapse. Given the fact that in this chapter we are considering noise effects on the control system, we cannot use UIOs as in Chapters 3 and 4. Instead, we use their stochastic equivalent: optimal disturbance decoupling observers (ODDOs). Specifically, we use a bank of \( p \) ODDOs, one for each sensor that could be attacked, where each ODDO does not take into account the \( j^{th} \) output. That is, as in Chapter 3, \( \tilde{y}_a^j[k] = M^j \tilde{y}_a[k] \). The hypotheses behind using ODDOs without one of the outputs is that, since there is only one attack simultaneously, when the attacked output is not considered we will be able to estimate accurately the state variables and, therefore, the outputs.

The \( j^{th} \) ODDO is described using the following state-space representation, inspired by [Chen and Patton, 1999],

\[
\begin{align*}
\dot{z}^j[k+1] &= F^j[k] z^j[k] + T^j B u[k] + K^j[k] \tilde{y}_a^j[k], \\
\dot{x}^j[k+1] &= z^j[k+1] + H^j \tilde{y}_a[k+1],
\end{align*}
\] (5.6)

where, in this case, \( \dot{x}^j[k] \) represents the estimated state by the \( j^{th} \) ODDO, and \( F^j[k] \in \mathbb{R}^{n \times n} \), \( T^j \in \mathbb{R}^{n \times n} \), \( K^j[k] \in \mathbb{R}^{n \times (p-1)} \) and \( H^j \in \mathbb{R}^{n \times (p-1)} \) are design matrices such that the estimated state of the ODDO, \( \dot{x}^j[k] \), converges to \( x[k] \) when there is no attack, i.e., \( F_a = 0 \). When the \( j^{th} \) ODDO (5.6) is applied to the system (5.1), decomposing \( K^j[k] = K^j_1[k] + K^j_2[k] \), the estimation error \( (e^j[k] = x[k] - \dot{x}^j[k]) \) is governed by the equation

\[
\begin{align*}
e^j[k+1] &= F^j[k] e^j[k] - K^j_1[k] \eta^j[k] - H^j \eta^j[k+1] + \\
&- K^j_1[k] F_a a[k] - H^j F_a a[k+1] + (I - H^j C^j) \zeta[k] + \\
&- \left[ F^j[k] - (I - H^j C^j) A + K^j_1[k] C^j \right] x[k] + \\
&+ (I - H^j C^j) E^j d[k] - \left[ T^j - (I - H^j C^j) \right] B u[k] + \\
&- \left[ K^j_2 - F^j H^j \right] \tilde{y}_a^j[k],
\end{align*}
\] (5.7)

notice that we use \( E^j \) instead of \( E \), indicating that each ODDO considers its own perturbations. That is because we consider that, for each ODDO, a way of approximate the control action modification is to give more weight to the state variables related with the outputs used directly by the \( j^{th} \) ODDO.

Let us consider for a moment the ODDO behavior where we neglect the attack effect on output sensors (\( F_a = 0 \)), in order to see how the ODDO estimate the
system state. For this case, \( \tilde{y}_a^j[k] = \tilde{y}^j[k] \), and (5.7) becomes

\[
e^j[k + 1] = F^j[k] e^j[k] - K^j_1[k] \eta^j[k] - H^j \eta^j[k + 1] + (I - H^j C^j) \zeta[k] +
- [F^j[k] - (I - H^j C^j) A + K^j_1[k] C^j] x[k] +
+ (I - H^j C^j) E^j d[k] - [T^j - (I - H^j C^j)] B u[k] +
- [K^j_2 - F^j H^j] \tilde{y}^j[k].
\] (5.8)

Given that we have a stochastic system, a proper state estimation is achieved when the estimation error for the \( j \)-th ODDO has the form

\[
e^j[k + 1] = F^j e^j[k] - K^j_1[k] \eta^j[k] - H^j \eta^j[k + 1] + (I - H^j C^j) \zeta[k],
\] (5.9)

and the eigenvalues of \( F^j \) need to be stable in order to \( \mathcal{E}\{e^j[k]\} \to 0 \), since \( \mathcal{E}\{\eta^j[k]\} \to 0 \) and \( \mathcal{E}\{\zeta[k]\} \to 0 \). That implies for the ODDO to estimate the state, we need to make equal to zero all the terms on the right side of (5.8) but the first. That is, we need to guarantee that

\[
E^j = H^j C^j E^j, \quad \text{(5.10)}
\]
\[
T^j = I - H^j C^j, \quad \text{(5.11)}
\]
\[
F^j[k] = (I - H^j C^j) A + K^j_1[k] C^j, \quad \text{(5.12)}
\]
\[
K^j_2[k] = F^j[k] H^j. \quad \text{(5.13)}
\]

When we consider the attack effect on the outputs (\( F_a \neq 0 \)), holding (5.10)-(5.13), the estimation error of the \( j \)-th ODDO will be governed by

\[
e^j[k + 1] = F^j e^j[k] - K^j_1[k] \eta^j[k] - H^j \eta^j[k + 1] + (I - H^j C^j) \zeta[k] +
- K^j_1 F^j a[k] - H^j F^j a[k + 1],
\] (5.14)

where, depending on the form of \( H^j F^j_a \) and \( K^j_1 F^j_a \), it will eventually converge proportionally to \( \mathcal{E}\{a[k]\} \) or to zero. That is, if there is an attack on the \( j \)-th output and how the \( j \)-th ODDO does not consider that output then \( \mathcal{E}\{e^j[k]\} \to 0 \), i.e., \( \mathcal{E}\{\hat{x}^j[k]\} \to \mathcal{E}\{x[k]\} \) and we will have the estimation of the outputs without the effect of the attack. On the other hand, if there is an attack on the \( i \)-th output and how the \( j \)-th ODDO considers that output \( \mathcal{E}\{e^i[k]\} \to \mathcal{E}\{a[k]\} \), and the estimated state will not help to recover the outputs without the effect of the attack.

In order to design the ODDOs, we need to solve (5.10)-(5.13). Notice that we only need to solve (5.10) for \( H^j \), that allows us to solve the rest of the equations, if we can guarantee that \((A^j_1, C)\) is detectable, with \( A^j_1 = (I - H^j C^j) A \). At this
5.2. Optimal Disturbance Decoupling Observers

point, it would be useful to use Lemma 3.2.1 to address the existence of solution of (5.10). Then, Lemma 3.2.2 can be used to show equivalence of the detectability of an augmented system and the one of the original system. Also, necessary and sufficient conditions for ODDOs can be reconstructed from Theorem 3.2.3, having into account that $F_j[k]$ and $K_j[k] = K_1^j[k] + K_2^j[k]$ are time dependant. Once we have verify the necessary conditions to design the bank of ODDOs, we need to proof that it is possible to recover the state estimation of the output without the effect of the attack from the $j^{th}$ ODDO. Finally, we can construct a similar theorem as Theorem 3.2.4 to prove that the state estimation from the $j^{th}$ ODDO is equivalent to the estimation of the state without the attack effect.

**Theorem 5.2.1** Suppose (5.6) is the $j^{th}$ ODDO for the system defined by (5.1), if there is an attack in the $j^{th}$ output, we can get an estimation of the output without the effect of the attack from the $j^{th}$ ODDO.

**Proof:** When we consider there is an attack on the system (5.1), i.e., $F_a \neq 0$. In fact, since we only consider one simultaneous attack $F_a$ can be written as a set of unitary column vectors with all but one element different from zero. Lets consider the simpler case where $a[k]$ is a scalar function and $F_a$ is a unitary vector. Assuming that attack is affecting the $j^{th}$ output, the form of $F_a$ is

$$F_a = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \leftarrow j^{th} \text{ row}$$

then $F_a^j = 0_{(p-1)\times 1}$ ($F_a^j$ equals $F_a$ without the $j^{th}$ row, in this assumption, the only element different from zero), and clearly (5.14) will become (5.9). The last implies that $\hat{x}[k] \rightarrow x[k]$ and, therefore, we can find $\hat{y}[k] = C\hat{x}[k]$, that represents the estimation of the outputs without the effect of the attack.

Now, if we consider the general case where $a[k]$ is a vector of $p$ functions and $F_a \in \mathbb{R}^{p\times p}$, the previous procedure holds but only for the time intervals where each attack is taking place, given that only one attack will occur simultaneously. □
5.3 Optimal Disturbance Decoupling Observers Design

Designing the $j^{th}$ ODDO requires a transformation to guarantee the existence of the observer and the anomaly decoupling on the $j^{th}$ sensor, which is done by

$$
H^j = E^j (C^j E^j)^+, \\
T^j = I_n - H^j C^j, \\
\bar{A}^j = T^j A,
$$

(5.15)

where $E^j$ is the matrix used to decouple the effect of the unknown attack on the $j^{th}$ sensor, $C^j$ is the resulting matrix when the $j^{th}$ row is eliminated from the matrix $C$, $I_n$ is an order $n$ identity matrix. Then, a standard Kalman gain is calculated, similarly as in (5.2),

$$
K_1^j[k] = \bar{A}^j P^j[k|k-1] C^j^\top \left( C^j^\top P^j[k|k-1] C^j^\top + R \right)^{-1},
$$

(5.16)

where $P^j$ is the covariance matrix associated to the transformed system after applying (5.15). After that, an update of the estimation of the covariance error matrix is done

$$
P^j[k] = P^j[k|k-1] - K_1^j[k] C^j P^j[k|k-1] \bar{A}^j^\top.
$$

(5.17)

Some other transformation matrices, need to be updated at each iteration, and they are given by

$$
F^j[k] = \bar{A}^j - K_1^j[k] C^j \\
K_2^j[k] = F^j[k] H^j \\
K^j[k] = K_1^j[k] + K_2^j[k] \\
z^j[k+1] = F^j[k] z^j[k] + T^j B \tilde{u}[k] + K^j[k] \tilde{y}^j[k],
$$

(5.18)

where $\tilde{y}^j[k]$ is the vector of sensor measurements for time $k$, when the row correspondent to the $j^{th}$ sensor is suppressed. Finally, the updates for the state estimated and the ahead prediction of the error covariance matrix are performed by

$$
\hat{x}^j[k+1] = z^j[k+1] + H^j \tilde{y}^j[k+1] \\
P^j[k+1|k] = \bar{A}^j P^j[k] \bar{A}^j^\top + T^j Q B^\top T^j^\top + H^j R H^j^\top.
$$

(5.19)
5.4 Detection, Isolation and Mitigation

Very similarly to what is done in Chapter 3, we use the attack mitigation mechanism proposed and detailed in Section 2.3. For this particular case, we propose the use of the already working Kalman filter described by (5.2)-(5.4) to know when an attack takes place, in a process known as detection. Whereas, for the isolation process we use a bank of ODDOs, where the $j^{th}$ ODDO is defined as in (5.15)-(5.19), that in the previous section has proven to recover the system state without the attack effect. We maintain the false alarm reduction logic described in Section 2.3 and, given the specific control in use, the calculation of the control action, with the nomenclature given in Subsection 2.3.3, can be written as (3.21).

5.5 Numerical Results - Three Tanks System

In this section, we show some results from applying the mitigation mechanism proposed in Section 5.4 to the three tanks system, described in Section 4.3, considering noise acting on the system.

In addition to the system model and controller defined in Section 4.3, we need to consider noise characteristics that affect system behavior. For the simulations we include white noise for sensors, $\eta[k] \sim \mathcal{N}(0, 5 \times 10^{-5})$, and actuators, $\zeta[k] \sim \mathcal{N}(0, 5 \times 10^{-6})$. Since the measurements include noise, the estimation of the state variables is done using a Kalman filter, instead of a full order current observers.

The state estimation $\hat{x}[k]$ is obtained using (5.2), (5.3), and (5.4), with

$$\begin{align*}
    P[1|0] &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\
    Q &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.
\end{align*}$$

The behavior of the closed loop system is shown in Fig. 5.1. There, it is seen how the control works properly for both variables $L_1(t)$ and $L_2(t)$, taking them to reach the desired value, every time the set-point varies. It can also be noticed how the system is coupled, because some small changes on the behavior appear when the references are changed.

The IAE between the response without attacks and the set-point input of the control system is used to quantify the impact of the attacks on the sensors of the system. It is important to highlight that the effect of an attack on the sensor
of Level 1 has impact on the response of Level 2, and for this reason, for each attack scenario the IAE, i.e., for Levels 1 and 2 are computed. In the case there is not attacks on the control system, the IAE for Level 1 is: 3.6935, and for Level 2 is: 2.7889. These IAE values without attacks are taking into account as the reference values. Therefore, the bigger the IAE values are the bigger the impact of the attack is.

5.5.1 Attacks Definition

In order to prove the effectiveness of the approach proposed, a set of 8 integrity attacks were applied to the system. As we mentioned before, we consider bias attacks and static attacks. Within the set of the applied attacks, there are six bias attacks, like the ones defined by (2.9), with their specific parameters shown in Table 5.1. The remaining two are static attacks, like the ones defined by (2.13), with their specific parameters shown in Table 5.2. In all of the cases, only one attack in one sensor is applied each time.

5.5.2 Mitigation Approach Implementation Results

We now evaluate our attack mitigation approach as outlined in Section 2.3. The anomaly detection and isolation mechanisms are implemented using the existing Kalman filter used to implement the controller, and two ODDOs. The 1st ODDO
is designed to decouple the effect of the attacks on the sensor of tank Level 1; the estimation of the state $\hat{x}^1[k]$ is obtained using (5.15), (5.16), (5.17), and (5.18); the inputs of this observer are the whole input vector $u[k]$, and only the output $y_2[k]$; the decoupling of attacks on the sensor for Level 1 is achieved using the matrix $E^1 = [10^{-5} \ 1 \ 10^{-5}]^\top$. The design of the 2nd ODDO, to decouple attacks on Level 2 sensor, is similar to the 1st ODDO; in this case, the inputs of the observer are the whole input vector $u[k]$, and only the output $y_1[k]$; the decoupling matrix is $E^2 = [1 \ 10^{-5} \ 10^{-5}]^\top$.

### Table 5.1: Bias attacks applied on the system.

<table>
<thead>
<tr>
<th>Attack #</th>
<th>On Sensor</th>
<th>$t_1$ (s)</th>
<th>$t_2$ (s)</th>
<th>$f_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>600</td>
<td>800</td>
<td>$-20 \times 10^{-3}$</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>900</td>
<td>1100</td>
<td>$-20 \times 10^{-3}$</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>900</td>
<td>1100</td>
<td>$-0.1 \times 10^{-3}(k-t_1)$</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1000</td>
<td>1200</td>
<td>$-0.1 \times 10^{-3}(k-t_1)$</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>800</td>
<td>1200</td>
<td>$(k-t_1) \times 10^{-4}$, $t_1 \leq k \leq \frac{t_1+t_2}{2}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$(k-t_2) \times 10^{-4}$, $\frac{t_1+t_2}{2} \leq k \leq t_2$</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>600</td>
<td>1000</td>
<td>$(k-t_1) \times 10^{-4}$, $t_1 \leq k \leq \frac{t_1+t_2}{2}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$(k-t_2) \times 10^{-4}$, $\frac{t_1+t_2}{2} \leq k \leq t_2$</td>
</tr>
</tbody>
</table>

### Table 5.2: Static attacks applied on the system.

<table>
<thead>
<tr>
<th>Attack #</th>
<th>On Sensor</th>
<th>$t_1$ (s)</th>
<th>$t_2$ (s)</th>
<th>$k_{si}$</th>
<th>$\eta_{ik}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>1</td>
<td>800</td>
<td>900</td>
<td>0.4480</td>
<td>$\mathcal{N}(0, \ 5 \times 10^{-4})$</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>1200</td>
<td>1300</td>
<td>0.2235</td>
<td>$\mathcal{N}(0, \ 5 \times 10^{-4})$</td>
</tr>
</tbody>
</table>

With the state estimation from each ODDO, we can now compute the residuals $r^1[k]$ and $r^2[k]$. For our simulations the thresholds obtained for ODDOs are: $\tau_1 = 3.3 \times 10^{-3}$ and $\tau_2 = 1.5 \times 10^{-3}$. The threshold for residual obtained with the Kalman filter is $\tau_D = 1.5 \times 10^{-6}$.

The effectiveness of our proposal is validated using a set composed by 8 attacks. A summary of the results after applying each of the attacks 1 - 8 are shown in Table 5.3. The first column is utilized to specify the attack number. The
Chapter 5. Noisy Linear Case

Attack Kind column has two possibilities, bias attack or static attack. The sensor data measurement altered by the attacker is in the third column, named Sensor Attacked, and has two options 1 or 2, to show the corresponding level. Columns four to six show IAE values for the two outputs of the system. In these columns there are three cases, the column without reconfiguration labeled w/o.R. that shows the impact of the attack on both outputs. A column with a conventional FTC reconfiguration scheme, which exhibits the reduction of the impact of the attack utilizing FTC techniques directly, and it is labeled C.R. The last column presents the impact of the attack when the mitigation new mechanism proposed is applied, and is labeled N.R.

Table 5.3: Key Performance Index Comparisons of different attacks applied on system sensors

<table>
<thead>
<tr>
<th>Attack Number</th>
<th>Attack Kind</th>
<th>Sensor Attacked</th>
<th>IAE w/o.R.</th>
<th>IAE C.R.</th>
<th>IAE N.R.</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B</td>
<td>1</td>
<td>7.2997</td>
<td>4.6104</td>
<td>4.6315</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.7904</td>
<td>4.0047</td>
<td>2.9526</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>2</td>
<td>3.7306</td>
<td>7.1550</td>
<td>3.8587</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6.3866</td>
<td>3.7127</td>
<td>3.6059</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>B</td>
<td>1</td>
<td>5.5417</td>
<td>4.4990</td>
<td>4.4263</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.8479</td>
<td>3.3428</td>
<td>2.8644</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>B</td>
<td>2</td>
<td>3.7006</td>
<td>5.0913</td>
<td>3.7630</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.6440</td>
<td>3.5105</td>
<td>3.4056</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>B</td>
<td>1</td>
<td>6.7904</td>
<td>4.3649</td>
<td>4.2625</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.8494</td>
<td>3.8014</td>
<td>2.7832</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>B</td>
<td>2</td>
<td>3.6775</td>
<td>6.5832</td>
<td>3.7952</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.8342</td>
<td>3.2639</td>
<td>3.3311</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>S</td>
<td>1</td>
<td>6.5327</td>
<td>4.0082</td>
<td>3.8635</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.8991</td>
<td>2.8715</td>
<td>2.8262</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>2</td>
<td>3.7173</td>
<td>3.7375</td>
<td>3.6946</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.1333</td>
<td>2.9410</td>
<td>2.8781</td>
<td>2</td>
</tr>
</tbody>
</table>

B = Bias attack, S = Static attack, w/o.R. = Without reconfiguration, C.R. = Conventional reconfiguration, N.R. = New reconfiguration.

Results of Table 5.3, show that both reconfiguration mechanism reduce the impact of the attack on the output corresponding to the attacked sensor. However, in bias injection attacks the conventional reconfiguration causes a bigger impact on the opposite output, while the reconfiguration proposed maintains a better behavior in the opposite output while adjusting the attacked output. In static
injection attacks, the result with both mechanisms of reconfiguration produces similar results.

5.5.2.1 Results Discussion - Attack #5

Now, in order to gain a better understanding of the results, we present a detailed description of two of the eight attacks utilized to show the effectiveness of our proposal. The first attack scenario analyzed is related with bias attacks. All of the attacks 1 - 6 have similar behavior, therefore without loss of generality, the attack # 5 is now analyzed. The effect of the attack # 5 in the outputs of the system is shown in Fig. 5.2. When this attack takes place, the IAE for Level 1 increases from 3.6935 to 6.7904, but the IAE for Level 2 has a little deviation from 2.7889 to 2.8494.

![Figure 5.2: Effect of the attack # 5 in the response of the control system.](image)

The next stage on the mitigation process is the detection and isolation of the attack. This process is explained in Subsection 5.3. Fig. 5.3 shows the result of this process. The red continuous line shows a true detection of the attack on Level 1. Due to the soft variation at the beginning and the end of the attack, the attack is detected and isolated with some delay, between 862 s and 1136 s. However, a false isolation of an attack on Level 2 is also obtained (dashed blue line). The last is a consequence of the fact that the output named $l^1$ is obtained without
the tampered information of the sensor of Level 1 (let us remember that the 1st ODDO does not use the information of $y_1$), but the output $l^2$ is obtained using that tampered information.

Figure 5.3: Isolation of the attack # 5, red line denotes isolation on Level 1, and blue line denotes isolation on Level 2.

Figure 5.4: Attack detection, computed using of the Kalman filter, that is a part of the original control system, under attack # 5.

Figure 5.5: Definitive attack isolation for attack # 5, red line denotes the existence and duration of an attack on the sensor of the Level 1.

The correction of the previous results of detection and isolation is done based on two facts. The first fact is that the Kalman filter, used for state feedback, is also useful to extract accurate information about the attack duration. This result is shown in Fig. 5.4. The second fact is that there is no simultaneous attacks on the two sensors of the system. The procedure explained in 2.3.2.1 is used to obtain the definitive attack isolation, and it is shown in Fig. 5.5.

A comparison between the utilization of the conventional FTC tools and the improved response obtained with our proposal is shown in the Fig. 5.6. It is clear that the main problem of the conventional FTC method is the degradation of the output of Level 2, when the attack on the sensor for Level 1 is mitigated. Using IAE values to compare the system behavior for the Level 1, without reconfiguration mechanism the value is 6.7904, with the conventional mechanism the value is 4.3649 and, with the improved mechanism it is 4.2625. In the same way, IAE the value for Level 2 without reconfiguration is 2.8494, with reconfiguration using the
5.5. Numerical Results - Three Tanks System

Figure 5.6: Mitigation response to sensor of Level 1 attack # 5 without mitigation response and with two different mechanisms of reconfiguration.

conventional FTC tools the value is 3.8014, and with our proposal it is 2.8262. IAE values, and visual inspection of Level 2 in Fig 5.6, show that the approach proposed keep the behavior of Level 1, where the attack takes place, and improves the behavior of Level 2, where the conventional reconfiguration process affects the system in a negative way.

5.5.2.2 Results Discussion - Attack #8

Similarly to the first scenario, attacks 7 and 8 have a similar behavior, and the second attack scenario corresponds to the detailed analysis of attack # 8. The effect of the attack # 8 in the outputs of the system is shown in Fig. 5.7, and the results of attack detection and isolation are shown in Fig. 5.8. The attack duration and the attack isolation are shown in Figs. 5.9 and 5.10. Finally, the results of the two mechanisms of reconfiguration are shown in Fig. 5.11. Attack # 8 also has effect on the IAE index; for Level 1 (sensor without attack) there is a little increment from 3.6935 to 3.7173, and for Level 2 (sensor under attack) there is an increment from 2.7889 to 4.1333 (see Fig. 5.7).

It is worthwhile to mention that the effect of attack # 8 is greater than the effect of attack # 5, in the sense that the amplitude deviation accomplished with attack
Chapter 5. Noisy Linear Case

Figure 5.7: Effect of the attack # 8 in the response of the control system.

# 8 is higher that the one with attack # 5. Also, due to the integrators in the controller, the maximum magnitude of attacks like # 7 and # 8 is proportional to the attack duration. It is important to emphasize that attack # 8 requires less resources for the attacker than attack # 5, i.e., in attack # 5 the attacker needs to know the current value of the sensor to add a value and then send the resultant value to the controller; but, in the case of attack # 8, the attacker only needs to know the desired value of the system system, and always sends the same distortion for all the duration of the attack.

Due to the noise included in the attack, as in (2.13), the isolation obtained exhibits an intermittency at the beginning of the attack between 1200 s and 1213 s. However, the end of the attack on Level 2 is well detected at 1300 s. (See Fig. 5.8).

Figure 5.8: Detection and isolation of the attack # 8, red line denotes isolation on Level 1, and blue line denotes isolation Level 2.
5.5. Numerical Results - Three Tanks System

Unlike the results obtained for attack # 5, with the reconfiguration in the case of attack # 8, the effect of the conventional reconfiguration on the opposite output is insignificant. The IAE value in Level 1 for this attack, without reconfiguration mechanism is 3.7173, with the conventional mechanism is 3.7375, and with the improved mechanism it is 3.6946. These IAE values show that in this case there are no significant deviations. Now, an examination of the IAE values for Level 2, without reconfiguration is 4.1333, with reconfiguration using the conventional FTC tools is 2.9410, and with our proposal is 2.8781. It is clear, that both reconfiguration mechanisms produce similar results on the outputs of the systems.

In this chapter, we show the usefulness of the proposed mechanism in a system (a benchmark system designed to prove FTC techniques) facing integrity attacks (false data injection attacks) in noisy environments. However, results for bias attacks are better than the ones for static attacks, mainly because the incidence of the attack in the opposite variables. That is, given a working control system, bias attacks always affect the attacked output as well as the other systems outputs with some delay, while static attacks only affect the attacked output.

We
Figure 5.11: Mitigation response to sensor of Level 1 attack ≠ 5 without mitigation response and with two different mechanisms of reconfiguration.
Conclusions and Prospective Work

Conclusions

We have developed an architecture to mitigate non-simultaneous sensor attacks in cyber-physical control systems. Such architecture consists of detection, isolation and response mechanisms, that working in a coordinated manner, decrease the degradation of the system performance produced by attacks on sensors of cyber-physical control systems. The detection mechanism that signals the occurrence of an attack is attained with the state observer, already working with the system. In addition to the existing state observer, we design a bank of disturbance decoupler based on observers to isolate the attack and to recover the state signals without the effect of the attack. Once the attack has been detected and isolated, a decision-making mechanism triggers an alarm indicating the output under attack, to calculate the control signal with the estimated state signal without the effect of the attack and, therefore, mitigate the effect of the attack on the closed-loop system. The architecture was proven for linear systems with and without noise. The mechanism have been tested in multivariable systems for integrity attacks named as bias (additive), static, and multiplicative attacks.

For linear systems without noise, the state estimator included is a full order current observer and a bank of UIOs is used for decoupling the attack effect on the system. The architecture proposed was simulated for a working closed-loop control system in the four tanks benchmark, where the control system is a tracking controller with state feedback. We also explored the implication that sensor attacks could have, finding that a single attack proportional to the state could destabilize the system and, once the system is unstable, the attack cannot be mitigated. In that sense, we have formulated an optimization LMI problem to find stability limits on attacks value in order to know whether or not a system is too sensitive to sensor attacks and if it is worth to have some redundancy in specific sensors in order to increase the resilience of the system to cyber-attacks.

For the linear system with noise, we use the included Kalman filter to estimate the state of the system and a bank of ODDOs to decouple the effect of the attack on the
system. We simulate the particular scheme on the three tanks benchmark system working with, again, a tracking controller with state feedback. It is worthwhile to mention that, given the presence of the noise, in this kind of systems is where simultaneous isolation often takes place and, it is where the correction mechanism implemented to reduce false alarms comes to be very useful, showing very good results, mainly for the mitigation of the attack effects on the non-attacked outputs.

In general, we can see that the proposed architecture to detect, isolate and mitigated non-simultaneous attacks in legacy systems works properly and it could be expanded to nonlinear systems with and without noise, only using the adequate state estimator and the decoupling observer bank, depending on the nature of the controller that is already working with the system.

In order to deploy our proposed architecture in legacy industrial control systems, it is necessary to develop an additional layer, which would act as a supervisor for the low-level controllers. Such layer will be in charge of verifying the elapsed time, while the mitigation mechanism is acting to decouple an attack, because a prolonged correction should be analyzed in a more detailed fashion. Prolonged corrections of an attack on sensors could be the consequence of the occurrence of system faults, instead of attacks. In some cases, small magnitude faults can also be corrected with our mechanisms but faults that cause bigger deviations cannot be corrected with the proposed mechanism.

**Prospective Work**

There are still several open problems in the mitigation response to cyber-attacks in cyber-physical control systems. Significant work on detection has been developed in recent work, improving our proposal with more sophisticated anomaly detection methods could be an interesting issue to explore.

Our work is addressed to give a response to the above-mentioned deception attacks on sensors of multivariable systems. However, this work has some constraints. First, no more than one sensor can be attacked simultaneously. Second, the considered system should be operating in a region that allows its representation by a linear approximation model. Third, the observability property of the system remains when one of the outputs of the system is not taking into account for the estimation of the state. Surpass these constraints are still open problems. The mitigation mechanism proposed in this work is based on the model of the system. Another interesting way to face the mitigation problem is to use data-driven methods approaches. Similarly, the exploration of different kinds of attacks
could be done as future work.

Attacks on actuators are not tackled in this work, a route to extend the results of the present work is to include this kind of attacks, i.e., extend our proposal to mitigate the effect of attacks on actuators and sensors (not only sensors). Similarly, another extension could be done if our proposal is improved to mitigate a plural number of simultaneous attacks on sensors.

Another interesting topic to explore is the scalability of our work. The extension and validation of the presented approach to a realistic factory with a huge number of low-level controllers is a work that fits well as a prospective work in this field.
Bibliography


