Vers l'autodétection des attaques cyber-physiques dans les systèmes de contrôle

Amer Atta Yaseen

Numéro d'ordre : 107574

UNIVERSITÉ DE LILLE

École doctorale **Sciences Pour l'Ingénieur**

Unité de recherche **CRIStAL** UMR 9189

Thèse présentée par

Amer Atta Yaseen

Soutenue le 20 mars 2019

En vue de l'obtention du grade de docteur de l'Université de Lille

Discipline **Automatique et Génie Informatique**

Toward Self-Detection of Cyber-Physical Attacks in Control Systems

Membres du jury :

Rapporteurs

Dominique Sauter	Professeur à L'université de Lorraine
Frédéric Kratz	Professeur à l'INSA Centre Val de Loire

Examinateurs

Nicolaï Christov	Professeur à l'Université de Lille
Abdouramane Moussa Ali	Maître de Conférences à l'Université de Toulon

Directrice de thèse

Mireille Bayart	Professeur à l'Université de Lille

ABSTRACT

A networked control system (NCS) is a control system in which the control loop is closed over a real-time network. NCSs are used in many industrial applications, and also in applications such as remote control, unmanned aerial vehicles or surgical teleoperation, ... The major advantages of NCS are a flexible architecture and a reduction of installation and maintenance costs, the main disadvantage of NCS is the network effects, such as time-delays, that influence the performance and stability of the control loop. These systems are also vulnerable to cyber-attacks.

This thesis makes some contributions regarding the detection of cyber-physical attacks as well as the development of a controller which capable of dealing with the other the bad effects of the network like time-delays.

To achieve this goal, the proposed approach is to adapt model-free controller and to improve its use in NCS. The main idea is based on mutual benefit between Smith predictor and the basic model-free controller. Then, the intelligent structure of model-free control is applied along with Generalized Predictive Controller (GPC) to achieve the Intelligent Generalized Predictive Controller (IGPC) as an enhancement for the standard GPC. The IGPC is designed along with two different methods for cyber-attack detection.

Moreover, a new security mechanism based on the deception for the cyber-physical attacks in NCS is proposed, this mechanism can allow to stop the cyber-attacks by providing the last line of defense when the attacker has an access to the remote plant.

Finally, two detectors for controller hijacking attack are introduced. The objective is to be able to detect an attack such as the Stuxnet case where the controller has been reprogrammed and hijacked. The advantage of these proposed detectors is that there is not necessary to have a priori mathematical model of the controller.

Keywords: Networked control system; Cyber-attacks; Time-delay; Model free control.

List of Figures

List of Tables

Notations

The following table describes the different acronyms used in this manuscript.

Acronym	Description of the acronym
NCS	Networked Control System
SCADA	Supervisory Control and Data Acquisition
ICS	Industrial Control Systems
IT	Information Technology
DES	Data Encryption Standard
MD5	Message Digest 5
IP	Internet Protocol
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
FIFO	First In First Out
GPC	Generalized Predictive Control
iPID	intelligent Proportional–Integral–Derivative controller
IGPC	Intelligent Generalized Predictive Controller
SISO	Single-Input, Single-Output

TABLE OF CONTENTS

CHAPTER 1

General Introduction

Contents

1.1 Context

The emergence of communication networks, led to the concept of remote-control system, which resulted in the birth to Networked Control Systems (NCS). Basically, the NCS can be defined as follows: When a traditional feedback control system is closed via a communication channel, which may be shared with other nodes outside the control system, then the control system is called NCS [1]. Also, NCS can be defined as a control system in which the control loop is closed over a real-time network.

NCS have been finding in several applications such as remote control [2-3], remote surgery [4] and unmanned aerial vehicles [5].

Generally, remote control systems and shared-network control systems are the two major methods to apply the communication networks in control systems [6].

Utilizing shared-network resources to transfer the signals, from controllers to actuators and from sensors to controllers, can significantly reduce the difficulty of connections.

A remote-control system can be explained as a system controlled by a controller placed far away from it. This is sometimes expressed to as tele-operation control. Remote data acquisition systems and remote monitoring systems can also be involved in this type of systems.

The industrial control systems have a multi-layer structure [7]. The global purposes of such a control structure are: (1) to keep the safe operational goals by reducing the possibility of unwanted behavior, (2) to meet the production requirements by maintaining a certain process values within predefined limits, (3) to increase the income of the production.

Control systems have been at the core of critical infrastructures, manufacturing and industrial plants for many decades, and yet, there have been few confirmed cases of cyber-attacks [8-9]. The using of NCS will rapidly increase probability of becoming exposed to hackers and the professional attackers, especially because these systems utilize Internet and wireless networks. The control and sensor data transfer via networks in NCS raise problems of network security [10 -11].

Several NCS applications can be labeled as security-critical. Attacking these systems can cause irreversible harm to the controlled physical system and to the people who use it.

Cyber-attacks can cause, considerable material damages and or human losses, in particular, for the Supervisory Control and Data Acquisition (SCADA) systems which perform important functions in national critical infrastructure systems, such as oil and natural gas distribution, electric power distribution, water and waste-water treatment, transportation systems, weapons systems, and healthcare devices. To go farther and highlight the problem, it's important to understand the weaknesses of NCS against the cyber-attacks through review of recently known attacks, and discussing the efforts which were made by the information technology and/or the theoretical tools of control to protect the NCS.

1.2 Some notable attacks in the control systems

The studies of the effects of various kinds of attacks on NCS are extended more and more during the last years [12-15]. Like other types of cyberthreats, the NCS attacks rapidly increased in sophistication way. This growth is linked to the rising connectivity of industrial systems as well as that of cyber-physical systems. In this section, let us review some profile cases reported from different countries during the last years.

In 2000, the Interior Ministry of Russia informed that attackers had hacked temporary the gas flows control system of natural gas pipelines [16].

Also, during spring of 2000, the sewage control system of the council Shire of Maroochy in Australian had suffered one of the most famous cyber-attacks in SCADA systems. This attack led to discharging about 264,000 gallons of raw sewage into parks and rivers. The attacker was a former employee of an Australian organization who had developed a manufacturing software, and asked for a new job with a local government, but was refused. In 46 occasions during two months, he constantly used a radio transmitter to remotely alter electronic data for particular sewage pumping stations and caused faults in their control systems. At the first time the sewage operators believed there is a leak in the pipes without taking into consideration the probability of an attack to their system. Therefore, the action against this attack has been very slow and they needed months to discover that spoofed controller signals were actuating the valves [17].

In August 2006, the traffic lights of Los Angeles were disturbed and caused traffic jam crowdedness and delays, this was made by two employees hacked into computers controlling the city's traffic lights.

The ability of a cyber-attack to damage a power generator turbine was confirmed by the Aurora generator test [18].

In Poland during 2008, the switch tracks of trams were controlled by a teenager through an improved TV remote control. The results of this attack were four derailments with twelve injured [19].

An investigation report in the Tennessee Valley Authority (TVA) which is one of the major public power company, indicated that the damage of their control systems could have been done by cyber-attacks [20].

The Iranian nuclear facility was attacked by a cyber worm Stuxnet in June 2010 [21-22]. The eventual goal of Stuxnet is to disrupt the ability of NCS by reprogramming the networked controllers to run out of their desired limitations [23-26]. The Stuxnet attack proves that the inspiration and ability are existing for making attacks even with the important goals like military NCS [23].

In December 2015, Reuters reported that a power company located in western Ukraine suffered a power outage that impacted a wide area which contained Ivano-Frankivsk regional capital. The investigators discovered that cybercriminals had facilitated the outage by using Black Energy malware to exploit the macros in Microsoft Excel documents. The bug was planted into the company's network using phishing emails [27].

In January 2016, GitHub provided a penetration testing solution which included a brute-force tool that can be used to attack Modbus, a serial communication protocol. Because the issue is public, it led to use of this tool by numerous unknown actors, and therefore to the rise in malicious activity to attack ICS during 2016 [28]. On 24 March 2016, an Iranian hacker was publicly accused by the officials of Department of Justice. The Justice Department claimed Iran had attacked U.S. infrastructure online, by infiltrating the computerized controls of a New York Dam, heralding a new way of war on American soil. Hackers broke into the command and control system of the dam in 2013, apparently through a cellular modem. This case signals the desire of some foreign nations to infect, and to operate, US infrastructure. Although the attack happened in 2013, it was only in 2016 that the cyber-attack was affirmed and attributed to the hackers in Iran [29]. The SFG malware, detected in June 2016, created backdoor on targeted industrial control systems networks related to a European energy company. According to security researchers at

SentinelOne Labs, this backdoor delivered a payload that was "used to extract data from or potentially to shut down the energy grid,". The Windows based SFG malware was designed to bypass traditional antivirus software and firewalls. It contained all the hallmarks of a nation-state attack, likely of Eastern European origin. During 2016*, the attacks targeting Industrial Control Systems (ICS) increased over 110 percent see Fig. 1.1, this information are given by IBM Managed Security Services (MSS). Particularly, the spike in ICS traffic was resulted from the brute-force attacks into the SCADA systems, by guessing the weak or default passwords. Once broken, the connected SCADA devices can be remotely monitored or controlled by the attackers. The USA was also the largest sources and destinations of ICS based attacks in 2016 as shown in Fig. 1.2, and Fig. 1.3. This is normal, since the USA has the largest presence of internet-connected ICS systems in the world [28].

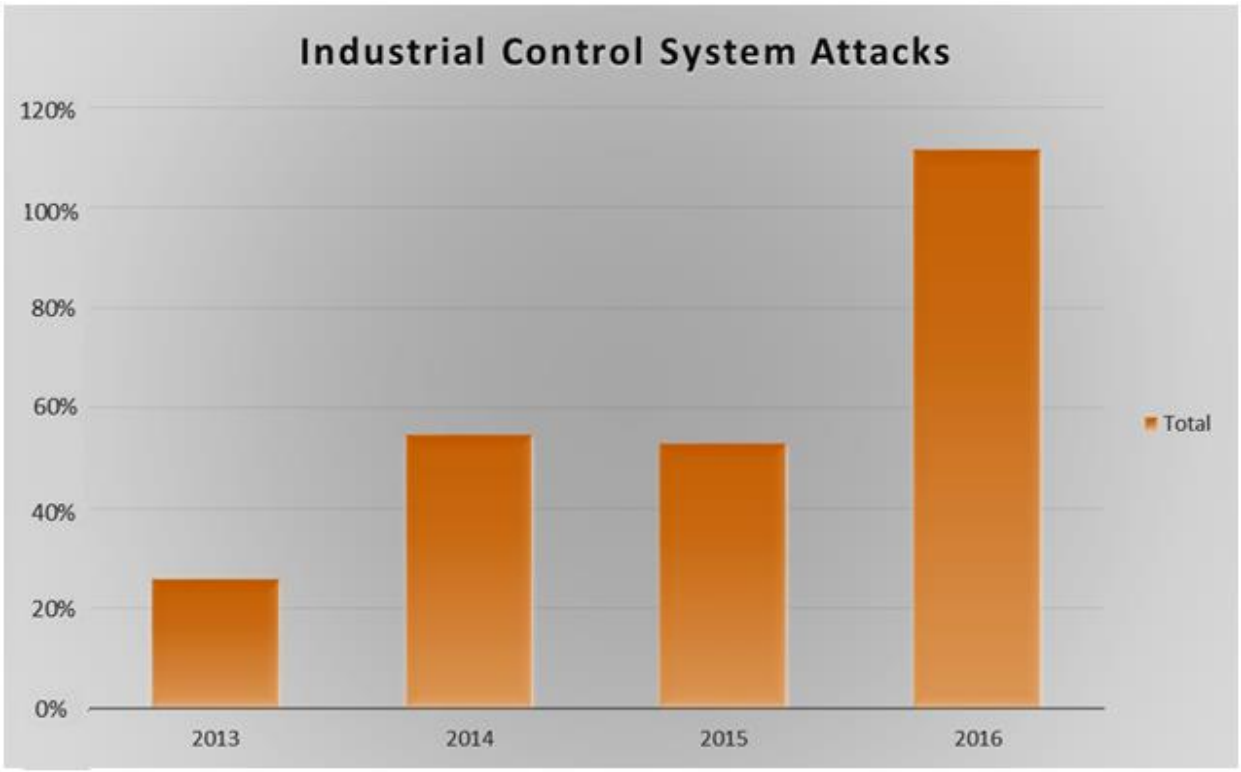

Figure 1.1: Increase of cyber-Attacks targeted industrial control systems between the years 2013 to 2016 [28].

** When we wrote this introduction, there was no information about attacks targeting industrial control systems during 2017 and later.*

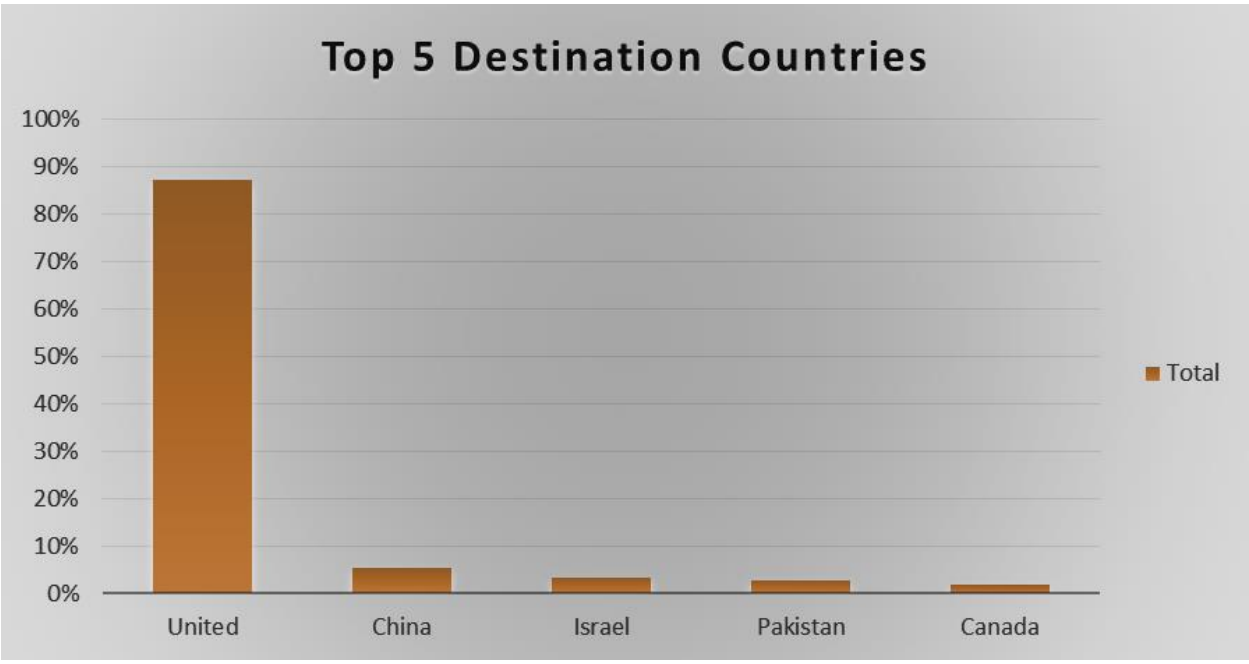

Figure 1.2: Top five countries as a destination of cyber-attacks [28].

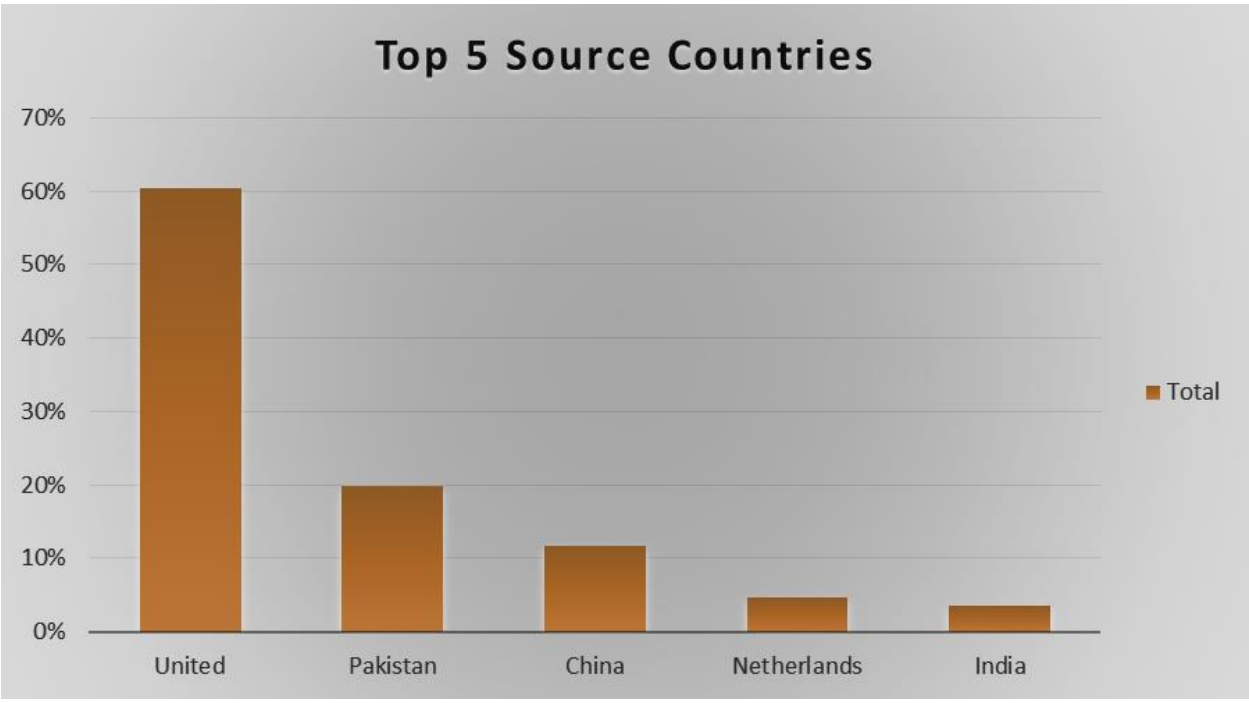

Figure 1.3: Top five countries as a source of cyber-attacks [28].

As we see there are more and more attacks on industrial control systems, therefore its always necessary to develop a secure control system. First idea has been to try to apply the security mechanism by using Information Technology (IT) tools only. After that, there are some differences represented by using a computation tools related to the control system beside IT tools, and then, more approaches are proposed to develop a secure control system without using IT tools.

1.3 Thesis Contributions

To answer the requirements of cybersecurity of network control system, one axis is to deal with this problem from the viewpoint of informatics. This viewpoint is possible to provide integrated solutions against the cyber-attacks in supervisory control layer of NCS/SCADA systems. However, the informatics solution is not able to be integrated within the regulatory control layer of NCS/SCADA.

Moreover, even in supervisory control layer, sometimes the attackers succeeded in hiding a specific programing method to modify the signals of actuation and sensor as well as to reprogram the controllers in NCS; nevertheless, they cannot cover up their ultimate goal [29]. The control behavior of NCS will be affected by the insertion of malicious actuation and sensor signals, or by changing the parameters of controller. These effects can be detected by applying the control-theoretic approach.

In additional and with regard to the works relative to communication system, we find that it is necessary to develop a controller capable to deal with the bad effects of the network like:

- Time-delays.
- Packet dropouts.
- The effects of the faults in the plant, because these effects can lead to a false cyberattacks detection.

This thesis aims to design a complementary solution regarding the cyber-physical attacks in the regulatory control layer, this design is mainly based on the control-theoretic approach as a supplementary part, with the informatics tools. The contributions of this thesis can be listed as follows:

1. A new concept to secure NCS based on the deception for the cyber-attacks.
2. To develop a suitable controller for networked control system applications. This contribution included two parts:

A. The synthesis of model-free control for NCS with time -varying communication delay.

B. The introduction of a strategy with an intelligent generalized predictive control.

3. To develop methods to detect the cyber-attack (by insertion or internal modification) in NCS, this development will be included with the three following approaches:

A. Detection method based on the intelligent generalized predictive control variables.

B. Detection method based on the intelligent generalized predictive control variables with fault accommodation.

C. Detection method based on the behavioral system approach.

4. Detection of controller hijack attack (Stuxnet effect), this includes the following:

A. Detection the controller rough hijacking attack.

B. Detection the controller stealthy hijacking attack.

1.4 Thesis organization

In the next chapters, first we will give a general view on the networked control system, after that the four contributions parts will be presented in details separately as following:

1. The concept of deception the cyber-attack.
2. Design of a suitable controller for the secure networked control system.
3. Detection the cyber-attack in networked control system.
4. Detection of controller hijack attack.

The related theoretical background and the simulation test results will be presented in each of the above four parts, and finally the conclusion and future work will be discussed.

1.5 List of publications

1. A. Yaseen and M. Bayart, " A Model-Free Approach to Networked Control System with Time-Varying Communication Delay," Safeprocess 2018, Warsaw, Poland, 2018.
2. A. Yaseen and M. Bayart, " Synthesis of Model-Free Control for System with Time-Varying Communication Delay," 19th IEEE International Conference on Industrial Technology, Lyon, France, 2018.
3. A. Yaseen and M. Bayart, " Cyber-attack Detection in the Networked Control System with Faulty Plant," 25th Mediterranean Conference on Control and Automation, Valletta, Malta, Forthcoming 2017.
4. A. Yaseen and M. Bayart, "Cyber-Attack Detection with Fault Accommodation Based on Intelligent Generalized Predictive Control," The 20th World Congress of the International Federation of Automatic Control, Toulouse, France, Forthcoming 2017.
5. A. Yaseen and M. Bayart, "Towards distinguishing between faults and cyber-attacks in the networked control system," 2016 World Congress on Industrial Control Systems Security (WCICSS), London, 2016, pp. 1-8.
6. A. Yaseen and M. Bayart, "Attack-tolerant networked control system: an approach for detection the controller stealthy hijacking attack," 13th European Workshop on Advanced Control and Diagnosis (ACD 2016) 17 - 18 November 2016, Lille, France.
7. A. Yaseen and M. Bayart, "Intelligent Generalized Predictive Control strategy for Networked Control System with an internal cyber-attack detector," 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, 2016, pp. 1-8.
8. A. Yaseen and M. Bayart, "Attack-tolerant networked control system in presence of the controller hijacking attack," 2016 International Conference on Military Communications and Information Systems (ICMCIS), Brussels, 2016, pp. 1-8.
9. A. Yaseen and M. Bayart, "SCADA Security: An Attack Tolerant Approach," Journal of Industrial Control Systems Security, Volume 1 (2016), pp.31-39.
10. A. Yaseen and M. Bayart, "Attack-Tolerant networked control system based on the deception for the cyber-attacks," 2015 World Congress on Industrial Control Systems Security (WCICSS), London, 2015, pp. 37-44.

CHAPTER 2

Networked Control System

Contents

2.1 Introduction to Networked Control System

The essential feature of an NCS is that data (reference input, control input, plant output, etc.) are transferred by utilizing a network between control system components (controllers, actuators, sensors, etc., see Fig. 2.1,).

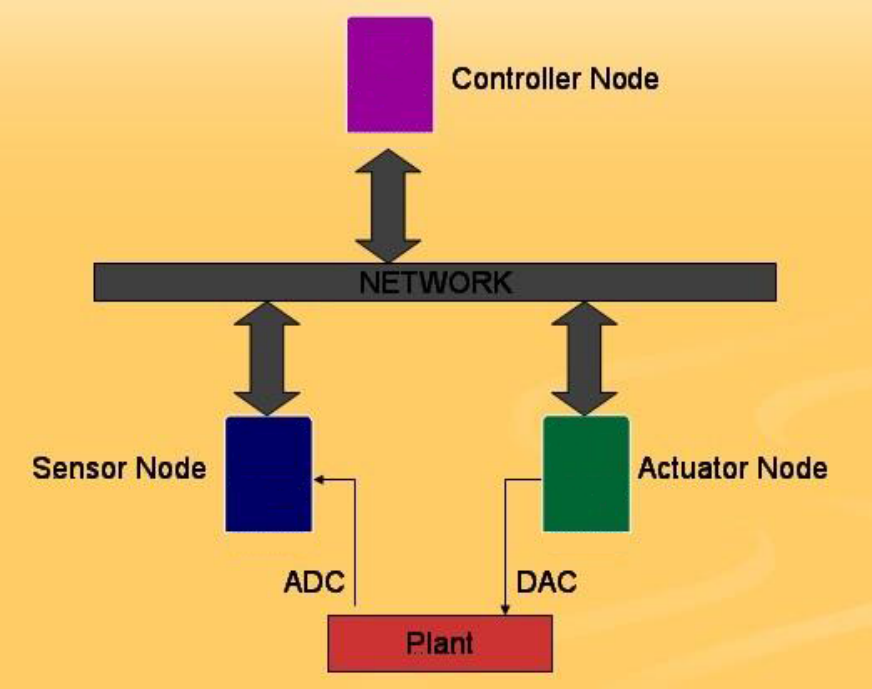

Figure 2.1: The Basic networked control system.

Some of the many advantages of NCS are remote operability, globally optimal solutions, global fusion of data, scalability, etc. NCS can be generally classified, according to the multi-sensors structure and multi-actuators, as a remote-control system and shared network system.

It can also be categorized as a non-real-time/time-insensitive and a time-sensitive/real-time control system.

Human involvement in the feedback loop of the NCS makes it a human supervisory controller which can be applied in many services like remote operation, remote surgery, etc. On the other hand, autonomous NCS take the human operator out of the feedback loop and only system configuration or task associated to inputs from human users are set, and directly putting all the feedback data into the network controller.

NCS combine several research branches affiliated with computer networking, communication, data processing, control theory, sensor fusion, security, etc.
This chapter provides an overview of NCS, its history, issues, architectures, components, methods, and applications.

2.2 Applications and benefits of networked control system

Recently, the technologies of communication networking have been extensively applied in control of industrial and military applications. These applications contain automobiles, manufacturing plants, and aircraft. The connection of control system components in these applications, such as, controllers, actuators, and sensors through a network can decrease the complexity of constructions in an efficient and economical way.

Moreover, with network controllers, the data can be shared effectively. It is easy to integrate the overall information to take intelligent actions over a huge physical environment. They remove avoidable wiring. It is easy to increase the number of controllers, actuators and sensors, with very few expenses and without big structural modifications to the entire system. Furthermore, the connections of physical space to cyber space in NCS facilitate the remotely task execution.

The NCSs are becoming more achievable during the last few years and have a lot of important applications [30-34], including factory automation, manufacturing plant monitoring, remote diagnostics and troubleshooting, experimental facilities, automobiles, domestic robots, aircraft, hospitals or nursing homes, hazardous environments, space explorations, and terrestrial exploration (tele-operation and tele-robotics).

2.3 NCS research developments

A huge base for millions of various fields was created by Internet, such as business, government networks, academic, and smaller domestic, which together provide services and information, such as interlinked web pages, electronic mail, file transfer, online chat and other.

Over the last few years, there has also been a significant increase in the use of wireless systems, which has activated the growth and research of distributed NCS.

Due to its potential in various applications, the notion of NCS ongoing to grow, also this growing brings up many challenges for researchers to reach efficient and reliable control.

Thus, the NCS field has been studied for decades and has given increase to many significant research subjects. A wide branch of these works efforts was on different control strategies and kinematics of the actuators and vehicles appropriate for NCS [35-38].

Additional important research parts regarding NCS are the study of the network structure which is necessary to offer reliable exchanges, the development of data communication protocols for control systems, and secured communication channel with sufficient bandwidth, [35] and [39-40].

Gathering real-time information through a network using distributed sensors and managing the sensor data in an effective method are important research areas accompanying NCS.

Consequently, NCS is not only a versatile area closely related to computer networking, information technology, communication, control theory robotics, signal processing, but it also sets all these together attractively to obtain a single system which can competently work over a network.

2.4 The classification and structures of NCS

In this section, the structure types of the networked control system as well as the methods of classification will be presented.

2.4.1 Architecture of NCS

Usually, the two main kinds of control systems that employ communication networks are (1) remote control systems and (2) shared-network control systems.

Using shared-network resources to transfer control signals from controllers to actuators, and measurements from sensors to controllers, can significantly decrease the difficulty of connections. This technique, as illustrated in Fig. 2.2, gives more flexibility in installation, is efficient and systematic, and easy troubleshooting and maintenance.

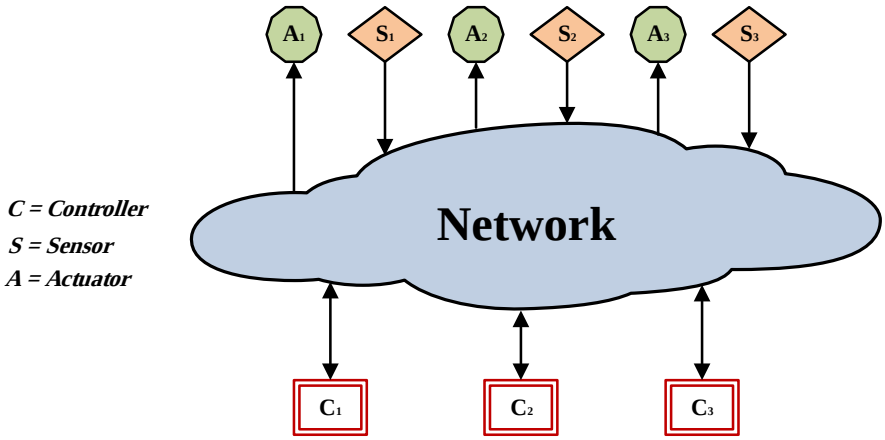

Figure 2.2: Shared-network connections.

Moreover, networks allow communication between control loops. This advantage is tremendously useful when a control loop exchanges information with other control loops to accomplish more sophisticated controls, such as fault tolerant and control. The same constructions for network-based control have been used in industrial plants and automobiles. Also, a tele-operation control can be understood of as a system which controlled by a controller placed far away from it. This is occasionally introduced as a remote-control system. Remote monitoring systems and remote data acquisition systems can also be involved in this type of systems. The location, where a central controller is placed, is normally called a "local side," while the location, where the plant is installed, is named a "Far side". For example, a person sitting in the USA can controlling a robot which is in the eastern part of the world Fig. 2.3, [41].

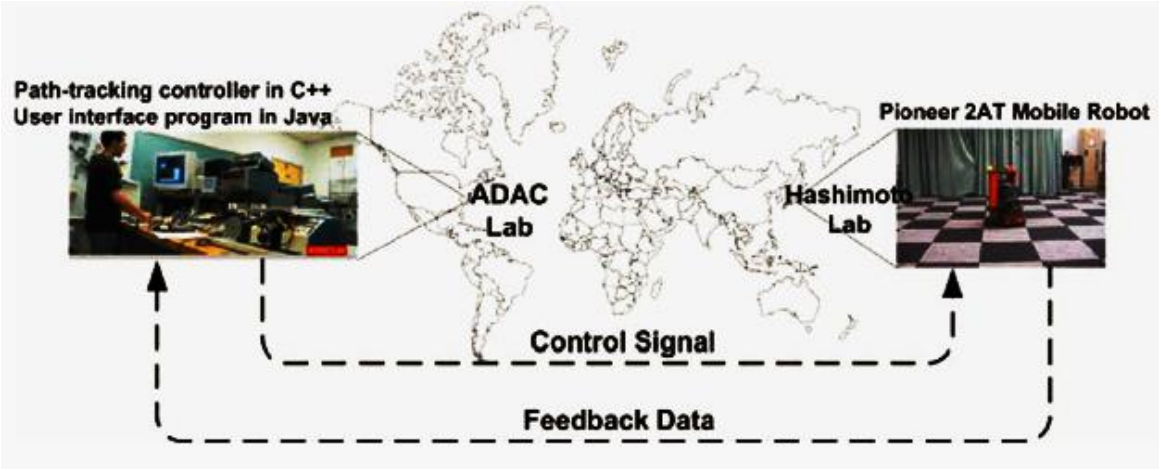

Figure 2.3: Remote mobile robot path-tracking via IP setup between Hashimoto lab (Japan) and ADAC lab (USA) [41].

Generally, there are two methods to design the NCS. The first method is to have multi subsystems arranged as a hierarchical structure, in which each of the subsystems contains a controller, an actuator, and a sensor as illustrated in Fig. 2.4. These system parts are joined to the same control plant. In this approach, the central controller (Cm) send a set point to each subsystem controller.

After that, the subsystem attempts to satisfy this set point individually. The status signal or sensor data is sent back over the network to the central controller.

The second method of networked control is the direct structure, as shown in Fig. 2.5. This structure has a control loop with an actuator and a sensor which are connected directly to a network. In this approach, an actuator and a sensor are directly connected to the plant, while a controller is connected to the plant via a network connection.

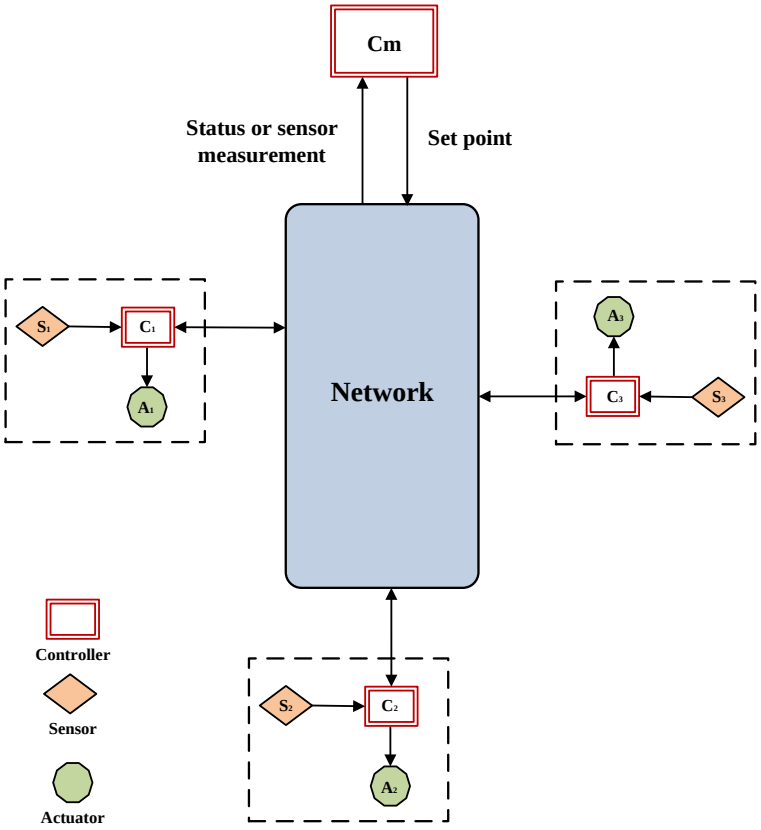

Figure 2.4: Data transfers of hierarchical structure.

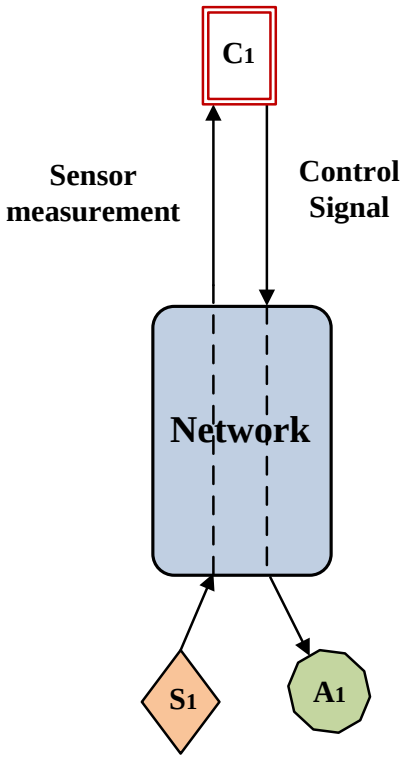

Figure 2.5: Data transfers of direct structure.

Both the direct structure and hierarchical one has their own advantages and disadvantages. Several networked control systems are a mixture of these two methods. The web-based lab is an illustration that utilizes from both structures [42-43]. The applications of networked control can be classified into two categories: (a) time -insensitive applications (b) time-sensitive applications. In time-sensitive applications, time is critical i.e., if the time delay exceeds the specified acceptable value, the plant or the device can be either in failure case or out of the desired performance. Examples of time sensitive applications are automated highway driving, tele-operation for fire-fighting operations over networks, and undersea operations. In additional, the programs or tasks are time insensitive applications that run in real time but the deadlines are not critical, for example http, DNS, email, and ftp, which are not considered our work.

2.4.2 Interaction between NCS and Human operator

The NCS can be categorized according to the rat of human intervention in the loop as following:

A. Tele-operation systems with human involvement

In this category, only one location will be used by a human operator to control the actuators at different locations e.g. unmanned vehicles, arms, or robots. Mostly, the feedback is a

visual information (real-time image or video) received in single operator (or multi operator) side. The performance of the system operation is depending on the skills of the operator and the level of signal distortion, feedback delay, accuracy and precision of system. This can also be named the human supervisory control [44]. Thus, the human operators are required to be trained to operate the system for such type of systems. There are numerous domains of such systems like field robotics, remote surgery systems, and distributed virtual laboratories [45] etc. Normally, these systems are affected by some matters such as network delay, ergonomics, control prediction, human perception accuracy, system portability and security, etc. [46-47]. Also, there are numerous tools designed to accurate the feedback to the operator like three dimensions visualization environment, virtual reality (VR), interactive televisions, etc. [46].

B. Tele-operation without human involvement

For these systems, the intelligence method is implemented within the controller modules. The data to the actuator and from the sensor is directly exchange with the controller via the network. This system is sometime called the autonomous networked control system. In this case, there is no human act as a supervisory controller. The action of human can be only as an outside user which able to specify some manual control commands or to choose tasks. Thus, such systems are independent on the visual perception of the human and don't require training and skill. Nevertheless, for supervisory control it's very important to develop control algorithms, effective data processing and intelligent techniques (e.g. artificial intelligence algorithms, machine learning, and neural networks).

C. Hybrid control

In such system, the efficiency of networked operations will be increased by using the intelligence distribution of controllers and actuators.

In this chapter mostly focus upon the time-sensitive networked control systems.

2.5 NCS Components

Regardless the used connection, hardware, or the software assets configuration to realize the networked control system with specific abilities and modalities arrangement, the four functions which form the basis of NCS components used, should be enabled. These basis functions are information acquisition (sensors), control (actuators), communication and command (controllers).

A. The Acquisition of Data in a Network

In this issue, the sensors study, signal processing, and data processing are the main targets. There is a rising anticipation about the effective application of large-scale sensor networks in several applications such as remote health care, precision agriculture, geophysical and environment monitoring, and security [48]. The quick development in sensing device, low-power computing and the methods of communications lead to multitude of commercially obtainable sensor types. To study the system under control, NCS needs to gather the relevant data using networks distributed sensors.

The sensor data can be represented in many formats, starting from the numerical representation of physical quantities such as flow, pressure, temperature, etc. or in file form such as videos streams, images, arrays, etc. This increases significant enquiries like:

1. Data gathering approaches in the case multi of sensors.

2. The requirements of bandwidth for the data transfer within network.

3. Reliable, low-priced and low energy consumption sensors which can simply be connected to the NCS.

Sensor networks and sensor fusion [49-50] are very important research domains which help to develop sensor data acquisition methods in a network. Evolving some operating systems and middleware for sensor nodes to efficiently send information over network [51-52], efficient energy of sensor nodes [53], information assurance [54], and data sensitivity are the main research trends related to the data acquisition via the network. Networked sensor introduces the facilitates for tele-operation, real-time data processing in difficult environments, large-scale, target tracking [55], and robot navigation [56], etc.

By the improvement in the field of image processing and computer vision, there are numerous advanced algorithms to process images and applied to feature extraction and pattern recognition. Generally, image data is used for applications like surveillance [48]. Various systems and algorithms have been developed using visual and other local distinguishing abilities to control aerial and ground vehicles [57-58].

B. Actuators Control Via a Network

The scalability is one of the main benefits of controlling the system over a network. As its possible to add several sensors connected at different places over the network, it’s possible to have one or more controllers connected to one or more actuators via the network. During

several years, the researchers' studies of researchers have provided us with important and effective control approaches. Extended from the fundamental control theory, starting from PID control, optimal control, robust control, adaptive control, intelligent control and various other unconventional forms of control strategies. The challenges in NCS made the control approaches used to design the controller is very important issue. In this thesis, several suitable approaches will be considered for efficient operation and successful NCS.

C. Communication

The core of NCS is represented by the communication channel. Availability, ease of use, reliability, and security are the main indicators for selecting the communication or data transfer method. Various communication manners are existing from telephone lines, satellite networks, GSM networks and the Internet which is widespread use. Certainly, the selection of network type must be subject the specification of the system under control. Internet is the most appropriate and low-cost network for numerous types of application where the controller and plant are far away from each other. The Controller Area Network CAN is normally used for connecting electronic control modules in industrial applications and automotive, in this network the communication protocol is serial, asynchronous, multi-master. The design of CAN takes into consideration the applications required data rates of up to 1 Mbps with high-level data integrity. Several industrial plants have a wide-ranging line of products allowing industrial designers to integrate CAN into their system.

In parallel with the development of industrial fieldbus, wireless networks are deployed in computer applications. By the time, the applications within any enterprise having been supported by wireless LANs. Homeowners are now installing wireless LANs at a quick pace with lower cost and steady standards. Now, LANs turn out to be approximately universal in organizations to support all sizes of personal computers and workstations. Also, all sites that still utilizing the mainframe have moved most of the load of processing to the personal computers network. The implementation of client/server applications represents prime example of the technique in which personal computers are used for command management and control. Back-end networks are used to connect big systems like supercomputers, mainframes, and mass storage devices. Here, the significant requirements are the high reliability and bulk data exchange within a limited number of devices in a small area.

Global Positioning System GPS can be applied to localize vehicles all over the earth. However, the dedicated optical networks can use to guarantee fast speed and reliable data communication for surgical and other emergency medical applications, and military applications.

There are three basic kinds of medium access control applied in control networks as following:

1. Time-division multiplexing (TDM) as in token-passing or master-slave.

2. Random access (RA) with prioritization for collision avoidance as in Controller Area Network CAN.

3. Random access with retransmission when collisions occur as in the most wireless mechanisms and Ethernet.

The networks are classified according to type: or collision avoidance CA or random-access RA with collision detection CD, or time-division multiplexed TDM using master-slave MS or token-passing TP.

In each of these three classes, there are many network protocols that have been well-defined and utilized. A study of the types of control networks utilized in industry shows an extensive diversity of the applied networks as listed in Table 2.1, which represent the worldwide most popular fieldbuses [59]. For Table 2.1, it’s important to note that the entireties users are greater than 100% since many establishments use more than one type of bus also wireless was not involved in the original survey of [59], but its usage is rising rapidly.

The networks are classified according to type: or collision avoidance CA or random-access RA with collision detection CD, or time-division multiplexed TDM using master-slave MS or token-passing TP.

In each of these three classes, there are many network protocols that have been well-defined and utilized. A study of the types of control networks utilized in industry shows an extensive diversity of the applied networks as listed in Table 2.1, which represent the worldwide most popular fieldbuses [59]. For Table 2.1, it’s important to note that the entireties users are greater than 100% since many establishments use more than one type of bus also wireless was not involved in the original survey of [59], but its usage is rising rapidly.

Table 2.1: Worldwide most popular fieldbuses [59].

Network	***Type***	***Users***	***Application domain***
Ethernet	RA/CD	50%	Various
Profibus	TDM/ (TP and MS)	26%	Process control
CAN-based	RA/CA	25%	Automotive, Process
Modbus	TDM/MS	22%	Various
ControlNet	TDM/TP	14%	Plant bus
ASI	TDM/MS	9%	Building systems
Interbus-S	TDM/MS	7%	Manufacturing
Fieldbus Foundation	TDM/TP	7%	Chemical industry
Wireless (e.g. IEEE 802.11)	RA/CA	Unknown	Various

2.6 Challenges and Solutions in NCS

In the previous sections, different applications, components, and categories of NCS have been introduced. This section will describe the different challenges and matters which must be taken in consideration for a trustworthy NCS. Generally, the applications of NCS can be classified into two groups as (a) Non-real time control applications such as e-mail, sensor data collection, data storage, etc. (b) Real time control application such as military, navigation operations, and space. Nevertheless, for both kinds of systems the reliability of network is a significant issue. The network can give time-dependent and unreliable levels of service like bad connectivity, time delay, packet losses, interference, cyber-attack, congestion, quantization, limited capacity or bandwidth see Fig. 2.6. The Quality of Service QoS can enhanced the real time performance of network, nonetheless the network performance is still affected by interference, redirection effects, and the activities of attackers. Moreover, network abnormality can endanger the performance, safety, and stability of physical environment units [60-61]. The effect of the network time delay is the most important challenge in the network-based control systems.

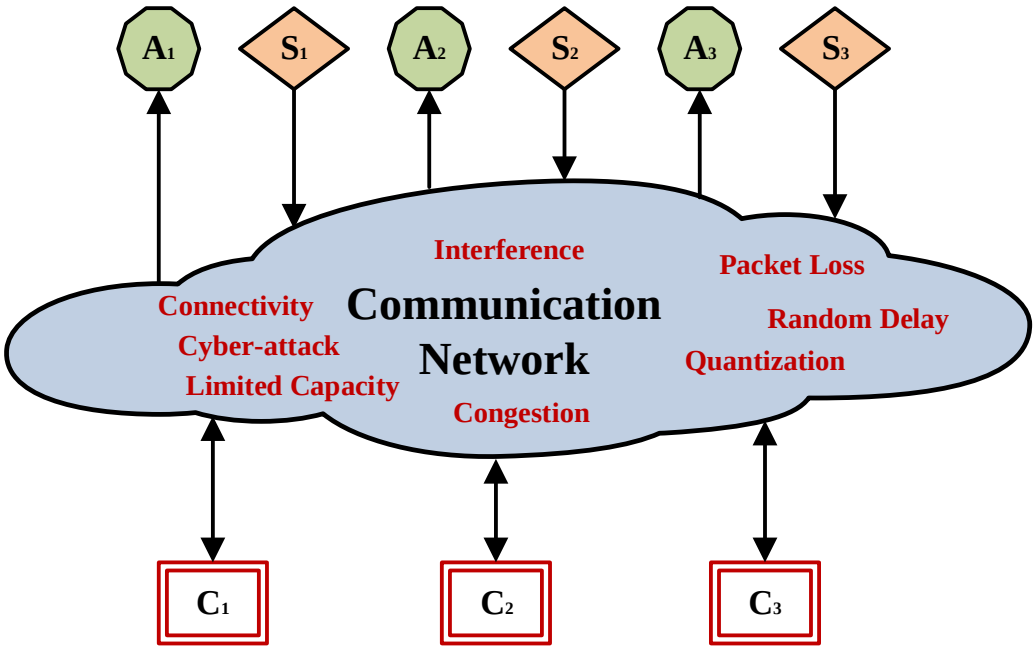

Figure 2.6: Some challenges in networked control system.

The time spend to send a control signal to an actuator and to read the measurement from a sensor through the network depends on network features for example routing arrangements and topology. Consequently, the general behaviour of the NCS can be pointedly affected by network time delays. The seriousness of the time delay problem is increased when the loss of data happens throughout a transmission. Furthermore, the delays do not only reduce the performance of a network-based control system, but they also can be resulted completely unstable system.

2.6.1 Time delay compensation for NCS stabilization

During several years, academics and researchers have suggested specific and optimum control approaches emerging from traditional control theories, starting from PID control, adaptive control, robust control, optimal control, intelligent control and various other unconventional algorithms of control. Nonetheless, it's necessary for these control approaches to be adapted according to reliably action via a network to compensate for time delays and uncertainty as well as to the requirements of various applications. Fig. 2.7, shows the representative model of NCS with considered time delay.

Different heuristic, mathematical, and statistical-based approaches are utilized for time delay compensation in NCS. The authors of [61] have suggested the Gain Scheduler

Middleware GSM to mitigate the effect of the network time delay on the network-based control systems. The technique of GSM is based on network traffic estimations and applying feedback processor to control the gain of the entire system.

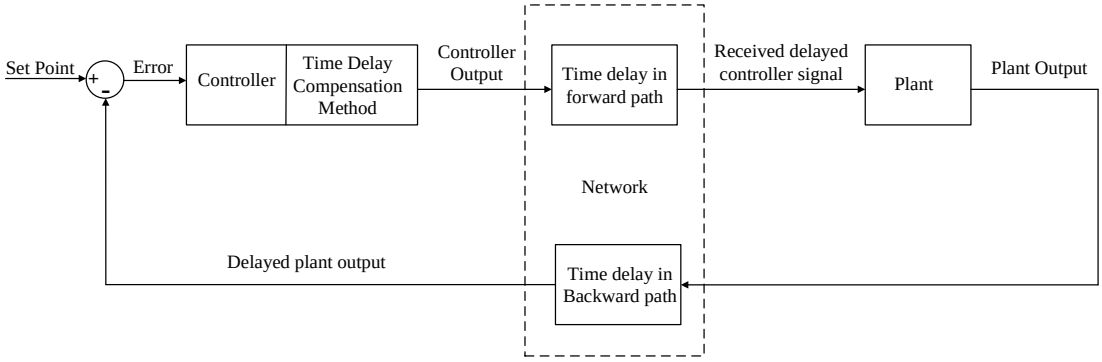

Figure 2.7: NCS plant structure showing network delay.

A new design of model predictive tracking control for a networked control system was introduced by R. Lu and Y. Xu in [62]. In this proposed control approach, a new state space model is presented, where, unlike the conventional state space models, the state variables and the tracking error and are combined and optimized together. Based on the enhanced state space model, additional design degrees can be provided and better performances of the control can be acquired. In [63] Wang and Wang proposed a controller as solution for time delay compensation, this controller is designed with an iteration method of Linear Matrix Inequality LMI, which is extended from the algorithm of cone complementarity linearization.

2.6.2 Scheduling and allocation of NCS bandwidth

As mentioned in the previous sections, the NCS performs the controlling of multi-actuators system and collects the information from multi-sensors via a network. Therefore, the bandwidth available in the network should be taken in consideration. When the availability of bandwidth is limited, it's important to use it in efficient and optimal way. Numerous parameters affect the utilization and availability of the network bandwidth [64]:

- The medium access control protocol that controls the transmission of the data or the size of messages.
- The quantity of units (controllers, actuators, and sensors) that require synchronous operations.
- The sampling rates at which different devices send data through the network.

- Physical factors for example, the method of synchronization between requesters and providers and network length.

These additional parameters increase the necessity for scheduling issues and priority decisions for controlling a series of tasks through a series of actuators [65]. Over the past decade, numerous bandwidth allocation methods and scheduling approaches were established for NCS [66-67]. Also, several tools such as dynamic programming, Petri-net modeling, nonlinear integer, genetic algorithms, and the artificial intelligence, are developed for NCS scheduling. According to the authors of [68], these tools are used to design a technique to achieve a maximum acceptable delay boundary for scheduling NCS with respect to the linear matrix of inequalities. Li and Chow proposed the scheduling of sampling rate to maintain the availability of data transmission and to solve the problem of signal reliability [69-70]. A protocol and Try-once-discard TOD scheduling for MIMO NCS are proposed in [65].

2.6.3 The security of NCS

In all previous discussions related to sending important control commands to actuators and reading sensors, the NCS brings us to an important point, which is the security over the NCS. However, this topic has been detailed in the first chapter of this thesis. In the next chapters, various algorithms and approaches related to detection of security threats on NCS will be described.

2.7 Conclusion

In this chapter, a general presentation of NCS was made, and important challenges were identified. In this thesis, time-sensitive with direct structure NCS will be considered, with, in particular, two challenges which are represented by time delay and the security problems related to cyber and/or physical attacks. The direct structure NCS will be simulated with the Ethernet which is easy to be under attack since its well known for the attacker as will be described in the next chapters.

CHAPTER 3

Deception for the cyber-attacks as an attack-tolerant approach

Contents

3.1 Introduction

The security of control systems has grown to be an effective area in the last few years. Nevertheless, before applying IT approaches to cyber Security, there is an imperative need to express what is new and essentially different in the point of view of security of control systems when compared to traditional IT security.

In this section, some differences that are formerly recognized, are briefly discussed and some new problems will be introduced.

One of the peculiarities of control systems that is most commonly brought up as a distinction with IT security, is that software patching and frequent updates are not well suited for control systems. These solutions are commonly utilized and sometime lead to weaknesses. For example, a number of establishments have proved that an effective antivirus and patching policy can be used successfully by the others [71], thus patching is not an essential constraint in the control systems. Also, security patches can be used to breach the certification of control systems such as previous accidents in control systems [72]. This is a simple reason that prevents the SCADA operators from using a good patch and antivirus programs in their platforms [73]. Moreover, the guidance for management of patch and antivirus in control equipment is offered by most of vendors for all their control products and available for the public.

Sometime, the upgrading of the system is economically difficult because it requires more than one month of preparation to switch to offline mode. Therefore, it's hard to justify the regular stop of the work of an industrial computer on a base to install new security patches.

The Large industrial control systems also included with several old parts not updatable or using only the recent IEEE P1711 standard which is designed for enhancement the security in legacy serial links [74].

Several studies efforts have tried to design lightweight cryptographic mechanisms or/and other IT security tools to ensure data integrity and confidentiality [75-76]. In [77], network security algorithms DES (Data Encryption Standard), 3DES (Triple Data Encryption Standard), and AES (Advanced Encryption Standard) are integrated into the application to secure the sensor as well as control data flow on the network. In addition, 1-D gain scheduler has been designed and implemented to mitigate the negative of security.

In [78], the DES and message digest 5 (MD5) are integrated with the application to encrypt the data transmitted on the network and to detect their integrity. The compromise between NCS security and its real-time performance has been demonstrated in [79]. A quick detection approach against data-injection attack in the smart grid was introduced in [80]. The Named Data Networking (NDN) was used in [81] in order to prevent the most attacks to which IP-based systems are vulnerable to. The authors in [82] have designed a detection module based on implementation of DES algorithm, furthermore, to protect NCSs from getting out of control, the authors have also designed a response module. The DES was adopted in [83] as security solutions for the DC motor networked control system in TrueTime platform algorithm.

However, several control systems are autonomous decision-making agents that need to make decisions in real time. While availability is a very important information security issue, real time system availability adds a constraint to the operating environment relative to most traditional IT systems.

Conventionally, the protection of information has been considered in all computer security studies but these do not take into consideration how the attackers disturb the measurement and control algorithms as well as the physical system as a result.

The main difference between the control systems and the other IT systems is the mutual interaction of the control system with the physical world. In spite of the recent tools of information security, they can provide the essential mechanisms for securing control systems, these mechanisms alone are insufficient to provide the deeply defense in control systems especially.

In all the mentioned works, the injected control data by the cyber-attacks can be only detected and rejected. Therefore, the attacker will continue to improve his strategy to break the NCS security by the method of trial and error. Normally, the method of trial and error is based on the ability of monitoring the actual sensor. This ability is achieved by breaking the encryption related to recent security mechanisms that used to secure the actual sensor readings. Furthermore, the rejection of attacks control data is always carried out on the side of the remote plant without any notification in controller side. As an example, the DES

key, which is used by most of previous researches on NCS security, this DES public key was broken in 22 hours and 15 minutes in January 1999 [84].

3.2 Towards attack-tolerant NCS

Network security can be defined as the measures and policies adopted by a network administrator to prevent and to monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources [85]. Previous researches on NCS security have been applied to the use of security tools designed for all types of networks. The main orientation of these tools is prevention.

However, what will be happened if the prevention is broken? The control system will have an undesired response and here we have a new type of the undesired response due the attack.

Previous researches focused on cryptography (DES, AES, etc.) to improve the security of NCS. In this chapter, we propose a method tolerant to attacks as an action against the attack due to the broken cryptography and, more specifically, it will be called "Attack-Tolerant". The proposed method will stop the spread of the attack by reconfiguring the sensor reading. The proposed method will be described in detail in the following sections.

3.3 Stopping the attack improvement

In general, the attacker's algorithm can be summarized as in Fig.3.1. The logical behavior of all attackers is based on the development of an attack method consisting by reading the sensor information which returned to the network by the attacked plant.

In this chapter, a new scenario for security mechanism will be introduced. The proposed mechanism is capable of making the attacker believe that he has succeeded in injecting deceptive control data to the remote plant as new control signal instead of the original authorized control signal. The proposed method will render the attacker's algorithm imperceptible and weak by disabling the core of its development (The block of correcting and updating in Fig.3.1,). The Fig.3.2, illustrating what will be the status of attacker's algorithm against the proposed security mechanism. Furthermore, an indication about any attempt to take the control of the plant by unauthorized person will be available through the proposed method. This indication will be available in the local controller side.

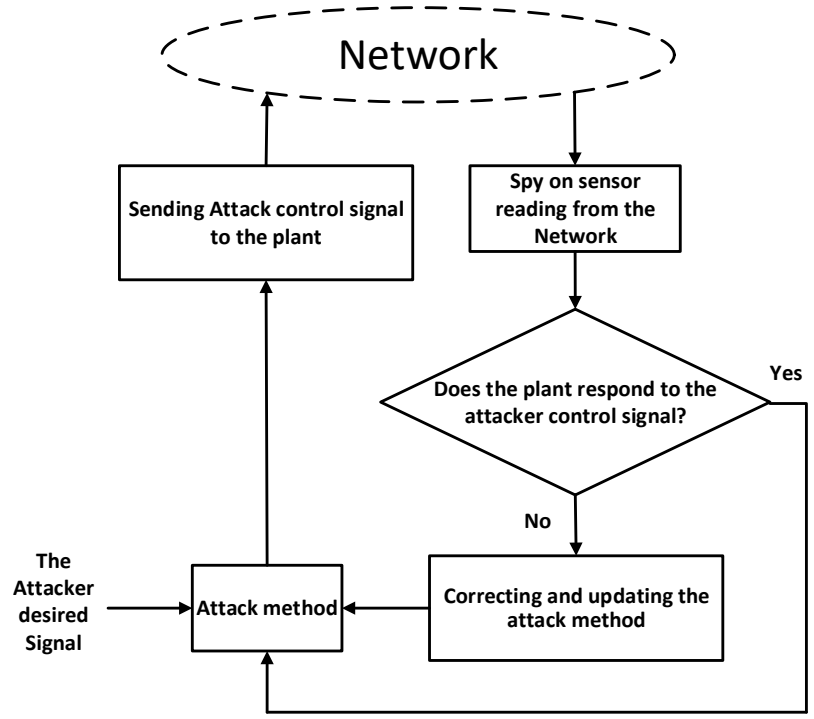

Figure 3.1: The general attacker algorithm against the current mechanism.

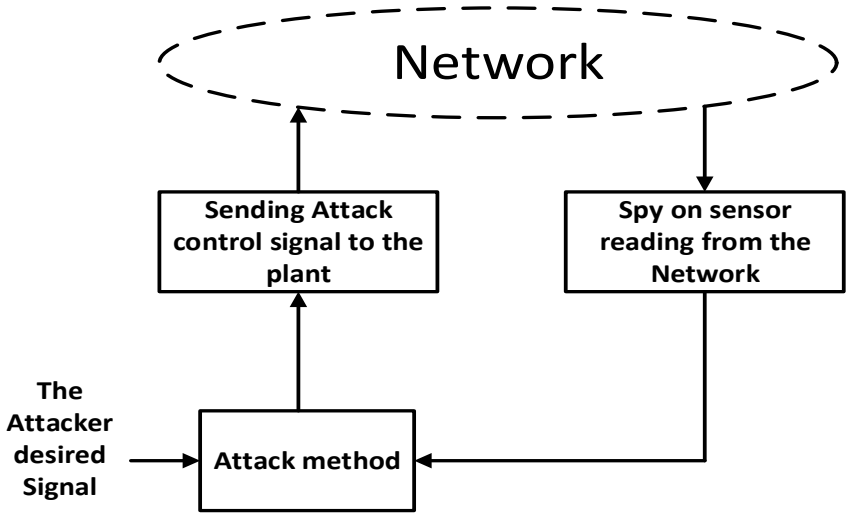

Figure 3.2: The general attacker algorithm against the proposed mechanism.

3.4 System structure and the common security objectives

Fig. 3.3, and Fig. 3.4, illustrate the complete block diagram for the controller side and plant side of the proposed secure method for NCS.

There are three common security objectives in both system sides (i.e. controller & plant); these components are data confidentiality, data integrity, and data replay detection that we describe below. Then, the network protocol UDP (User Data Protocol) will be presented.

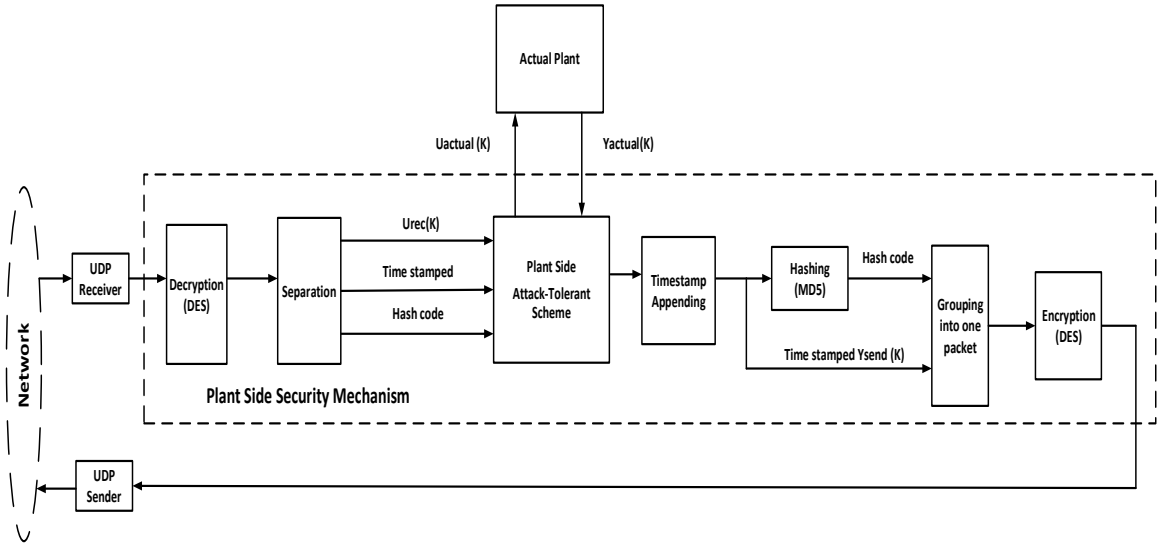

Figure 3.3: The block diagram of plant side.

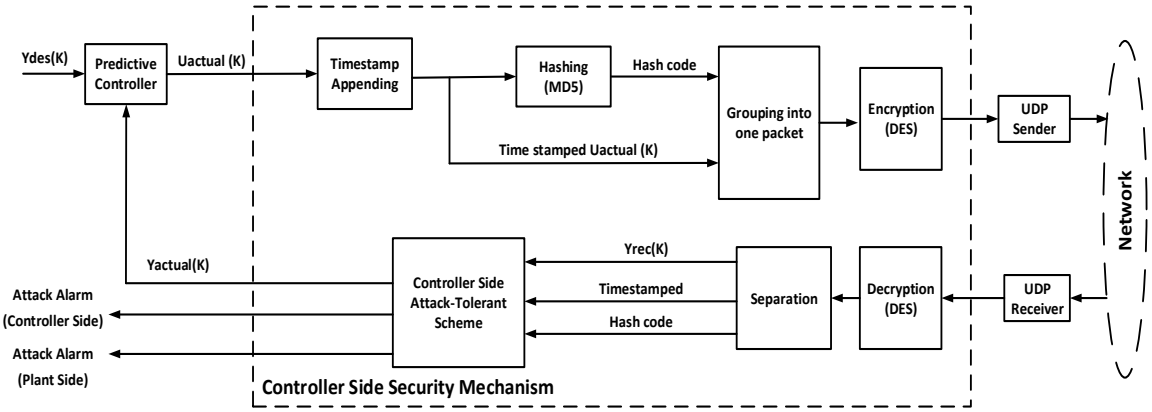

Figure 3.4: The block diagram of the controller side.

3.4.1. Data confidentiality

For the NCS data confidentiality, DES, 3DES, and AES are the typical symmetric ciphers; the total time for encryption as well as decryption depends upon the packet length between controller and the plant. The induced time-delay with respect to DES, 3DES, and AES is

illustrated in Fig.3.5, [86]. DES algorithm is the most commonly used symmetric cipher due to its fast speed. Therefore, the DES algorithm performs data encryption in this thesis.

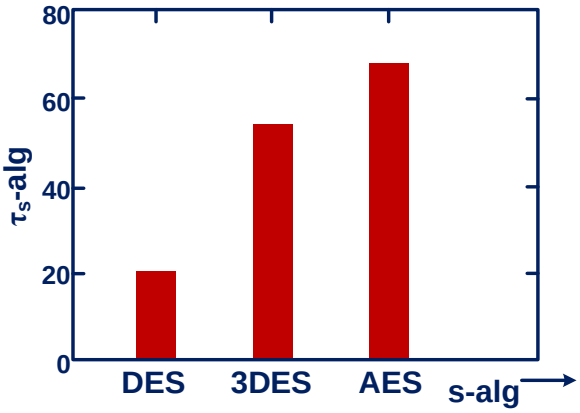

Figure 3.5: Induced time-delay with respect to security algorithms [86].

DES (Data Encryption Standard, that is, Standard Data Encryption) is a world standard since the end of 1970. At the beginning of this decade, the development of communications between computers required the implementation of a data encryption standard to limit the proliferation of different algorithms that cannot communicate with each other. To solve this problem, the US National Security Agency (N.S.A.) has issued tenders. I.B.M. developed then an algorithm named Lucifer, relatively complex and sophisticated. After a few years of discussions and modifications (applications of S-Boxes and reduction to 56-bit keys), this algorithm, now DES, was adopted at the federal level on November 23, 1976.

The DES has several advantages that have made it, for a long time, the standard symmetric encryption algorithm until a few years ago. Here are a few:

- It has a high level of security,
- It is completely specified and easy to understand,
- The security is independent of the algorithm itself,
- It is made available to all, by the fact that it is public,
- It is adaptable to various applications (software and hardware),
- It is fast and exportable,
- It is based on a relatively small key, which is used for both encryption and decryption,
- It is easy to implement.

DES is a cryptosystem acting in blocks. This means that DES does not encrypt the data on the fly when the characters arrive, but virtually divides the plaintext into 64-bit blocks that it encodes separately and then concatenates. A 64-bit block of clear text enters one side of the algorithm and a 64-bit block of encrypted text out the other side. The algorithm is quite simple since it only combines permutations and substitutions. The main DES algorithm is shown in Fig. 3.6.

It is a secret key encryption algorithm. The key serves both to encrypt and to decrypt the message. This key has a length of 64 bits, that is to say 8 characters, but only 56 bits are used. One can therefore imagine a program testing the integrity of the key by exploiting these unused bits as parity check bits.

The entire security of the algorithm is based on the keys since the algorithm is well known to all. The 64-bit key is used to generate 16 other keys of 48 bits each which will be used during each of the 16 iterations of DES. These keys are the same regardless of the block that is coded in a message.

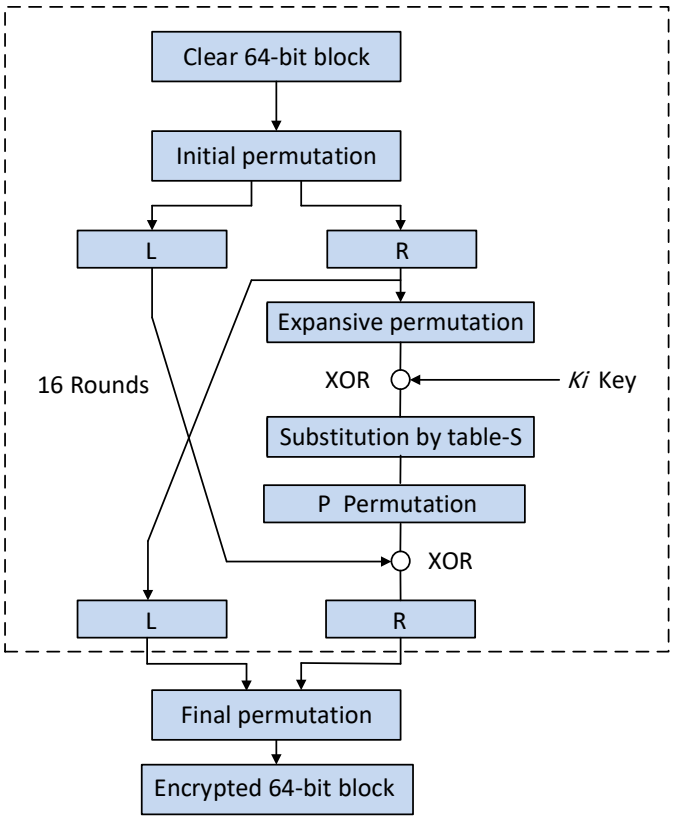

Figure 3.6: DES main algorithm

This algorithm is relatively easy to achieve and some chips encrypt up to 1 GB of data per second. For manufacturers, this is an important point especially in the face of asymmetric, slower algorithms, such as the RSA (Rivest–Shamir–Adleman) algorithm.
The algorithm is mainly based on 3 steps, in addition to the specific management of the key:

1. Initial permutation.
2. Median calculation (16 times): application of a complex algorithm applied according to the key.
3. Final Permutation.

The decryption is obtained by using the same algorithm but inversed. Details of this algorithm are given in Appendix 1.

3.4.2 Data integrity

In order to checking the Integrity of the NCS data, one-way hash function is used. One-way hash functions accept a variable size message as input and produce a fixed-size output, called hash code. MD5 hashes are used to ensure the data integrity of the received control signals at the plant side and the received sensor signals at the controller side. An MD5 hash is NOT encryption. It is simply a fingerprint of the given input. However, it is a one-way transaction and as such, it is almost impossible to reverse engineer an MD5 hash to retrieve the original string.

Because the MD5 hash algorithm always produces the same output for the same given input. The deception attacks (except the data replay attack) can be detected by comparing the hash of the source data (i.e. actual controller signal or sensor reading) with a newly created hash of the destination side (i.e. controller side or plant side) to check that it is intact and unmodified. Appendix 2 describes the MD5 algorithm.

3.4.3 Data replay detection

In this work, the timestamp will be utilized for detecting the data replay attack. A timestamp is the current time of an event that is recorded by a controller, sensor or even by the plant. This will be accomplished through mechanisms such as the network time protocol, NCS components which maintain an accurate current time, the calibration to minute fractions of a second. Such precision makes it possible for NCS applications to

communicate effectively. The timestamp mechanism is used for a wide variety of synchronization purposes, such as assigning a sequence order for a multi-control transaction so that if a failure occurs the transaction can be avoided.

Another way that a timestamp is used, is to record time in relation to a particular starting control point. In IP network, for example, the Real-time Transport Protocol (RTP) assigns sequential timestamps related to controller and sensor packets so that they can be buffered by the receiver ends, reassembled, and delivered without error. In this thesis, the timestamp will be used to accept only the newer packet from controller side to plant side and vice versa.

3.5 Network protocol

Due to the real time nature of the NCS data, UDP is preferred for use in the NCS over IP network. UDP is the abbreviation for "User Datagram Protocol". UDP is non-reliable protocol, there are no retransmissions in UDP and the packets may arrive in out-of-sequence manner at the receiver end so to use UDP in NCS we must add reliability. This reliability is represented by using the timestamp and the predictive control as will be described during the next sections.

The data encapsulated in a UDP header are UDP packets. UDP runs over IP and relies on the services provided by it and allows applications to directly access a datagram delivery service, such as the IP forwarding service.

3.5.1 Identification of the destination

At the Internet layer, datagrams are routed from one machine to another depending on the bits of the IP address that identify the network number. During this operation, no distinction is made between services or users that issue or receive datagrams, i.e. all datagrams are mixed.

The UDP layer adds a mechanism that allows the identification of the service (Application level). Indeed, it is essential to sort between the various applications (services): several programs of several users can simultaneously use the same layer of transport and there must be no confusion between them.

For the Unix system, programs are uniquely identified by a process number, but this number is ephemeral, unpredictable remotely, it cannot be used for this function.

The idea is to associate the destination with the function it fulfills. This identification is done using a positive integer that we call port.

- The local operating system is responsible for defining the mechanism that allows a process to access a port.
- Most operating systems provide the means for synchronous access to a port. This software must then provide the ability to manage the queue of incoming packets, until a process (Application) reads them. Conversely, the OS (Operating System) blocks a process that attempts to read data that are not yet available.

To communicate with a remote service, you must know the port number, in addition to the IP address of the machine itself. The port number can be predicted according to the service to be reached.

The Fig. 3.7, explicitly explains the concept of port. The Internet Protocol (IP) layer separates SCTP (Stream Control Transmission Protocol), TCP (Transmission Control Protocol), and UDP datagrams through the PROTO field of its header, the combination of transport protocol and the port number identifies an unambiguous service.

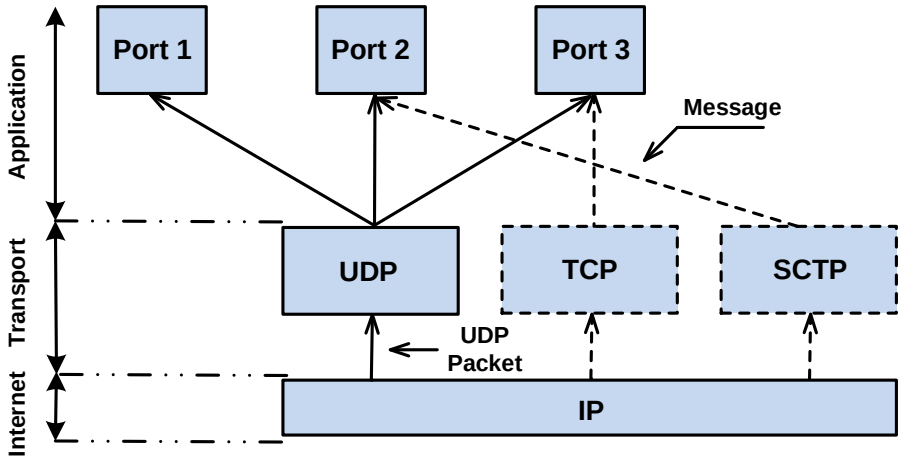

Figure: 3.7 Port number as service number.

3.5.2 Description of the header

A UDP packet is designed to be encapsulated in an IP datagram and allows data exchange between two applications, without preliminary exchange. Thus, if the data to be transmitted does not force IP to fragment, a UDP packet generates an IP datagram and that's it.

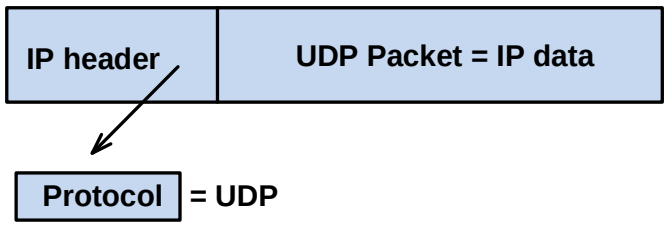

Figure 3.8: UDP encapsulated in IP.

From the above:

- UDP provides a port management mechanism, over the Internet layer.
- UDP is simply an interface over IP, so messages are sent without guarantee of good routing.

More particularly, packets destined for a UDP application are kept in a FIFO (First In First Out) type stack. If the destination application does not consume enough of them quickly, the oldest packets may be overwritten by new ones, an additional risk (compared to already known IP properties) of data loss. Also, it's necessary to note the following:

- There is no feedback at the protocol level to provide any means of control over the proper routing of the data. This deficiency must be taken into account at the application level.
- UDP is also referred to as unconnected transport mode, or datagram mode, as opposed to TCP or SCTP.

Among the most common uses of UDP, we can report the name server and a database distributed worldwide which is very well suited to this mode of transport. In local, other very useful applications like tftp (trivial file transfer protocol) or nfs (network file system) are also likely to use UDP. More details of UDP are given in Appendix 3.

3.6 Attack-Tolerant scheme

In order to improve the NCS security, we develop an attack-tolerant scheme. It is designed to serve the continuity of the NCS works when the data confidentiality is broken. In additional to the attack detection, the attack-tolerant scheme helps to making the attacker believes that he has achieved the desired goal and as a result, the attacker will stop the attack method improvement. Furthermore, an indication about the attacks will be included

within the sensor reading. There are two parts of the attack-tolerant scheme; the first part is located in the plant side and the second part in the controller side.

3.6.1 Plant side attack-tolerant scheme

The internal diagram of "Plant Side Attack-Tolerant Scheme" block in Fig. 3.3, can be illustrated as in Fig. 3.9.

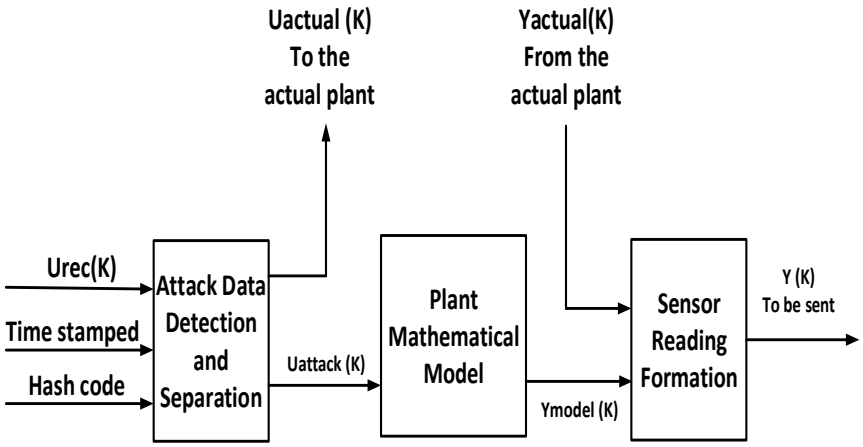

Figure 3.9: The internal diagram of the plant side attack-tolerant scheme.

The attack detection and separation unit will receive three inputs, *Urec(k)* (which represents either authorized, or attacker control signal), timestamp and hash code as mentioned in section 3.4. Attack detection will be utilized from the received MD5 hash codes to check the integrity as well as the originality of the received controller signal. If the received signal is original (i.e. *Uactual(K)*), the timestamp is used to prevent the control signal from accessing the actual plant if it is delayed or replied. If there is, no integrity or replay detected, then the received signal will be classified as an attack control signal *Uattack(K).*

In order to have plant output that related to the attack control signal, *Uattack(K)* will be sent to the plant mathematical model. In the sensor reading formation unit, the sensor reading of the actual plant *Yactual(k)* will be included in the response of plant model in a manner that makes the attacker believe he has achieved the desired goal.

In this thesis, we introduce the sensor reading formation technique for attack-tolerant. But this technique can be implemented with various software algorithms; one of the possible algorithms is shown in. Fig. 3.10, which represents the overall plant side attack-tolerant software algorithm. In case of an attack, the formation technique of sensor reading can be described as following:

Referring to Fig. 3.9, let *X(k)* is the difference between the actual sensor reading and the plant model response or,

X(k) = Yactual(K) - Ymodel(K).

- The absolute value of *X(k)* will be *"Y3 Y2 . Y1 Y0"* .
- The value of *Ymodel (K)* will be *"Y7 Y6 . Y5 Y4"*.

If *X(k)* is positive then *Y4* will be odd and vice versa. The final string of the sensor reading *Y(K)* will be as follows:

"Y7 Y6 . Y5 Y4 Y3 Y2 Y1 Y0"

The uncertainty in actual plant is employed to increase the level of lack of understanding of the value of actual sensor reading.

3.6.2 Controller side attack-tolerant scheme

On the controller side, a timestamp will be appended to the controller output *Uactual(K)*, the Message Digest 5 (MD5) is used to generating the hash code of the *Uactual(K)*. The *Uactual(K)* hash code, and timestamp as a whole are encrypted into one packet and sent to the plant side. The "Controller side Attack-Tolerant Scheme" block in Fig.3.4, performers the following procedures:

- Controller side attack detection and rejection.
- The extraction of the actual sensor reading.
- The plant side attack detection by checking the form of the received sensor reading.

The controller side attack-tolerant scheme will receive three inputs, the *Yrec(k)* which represents either authorized or attacker sensor reading, timestamp, and hash code. The first procedure in attack-tolerant scheme at the controller side is similar to the procedure in the plant side but in controller side, there is only rejection for both of unauthorized and replied data. If there is any attack in the plant side that was not detected during the first procedure, the sensor reading that provided from the procedure in the plant side will be included with this attack information.

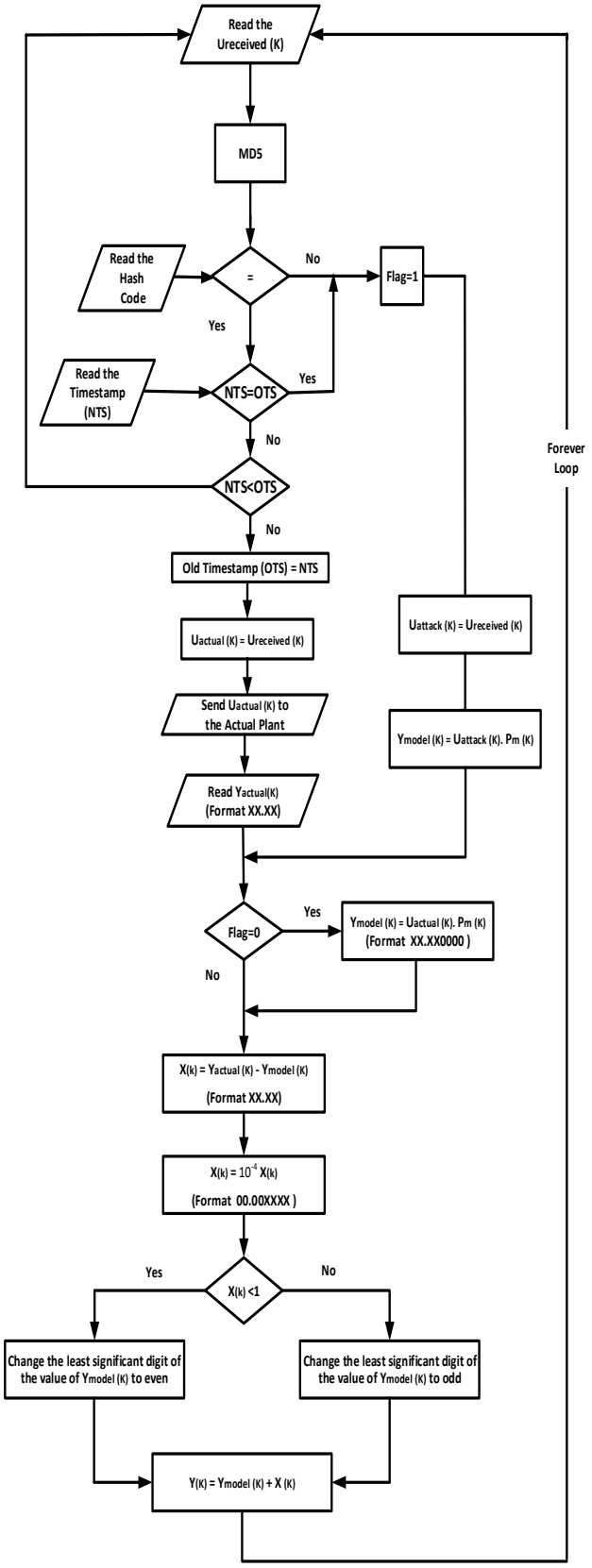

Figure 3.10: Overall plant side attack-tolerant software algorithm.

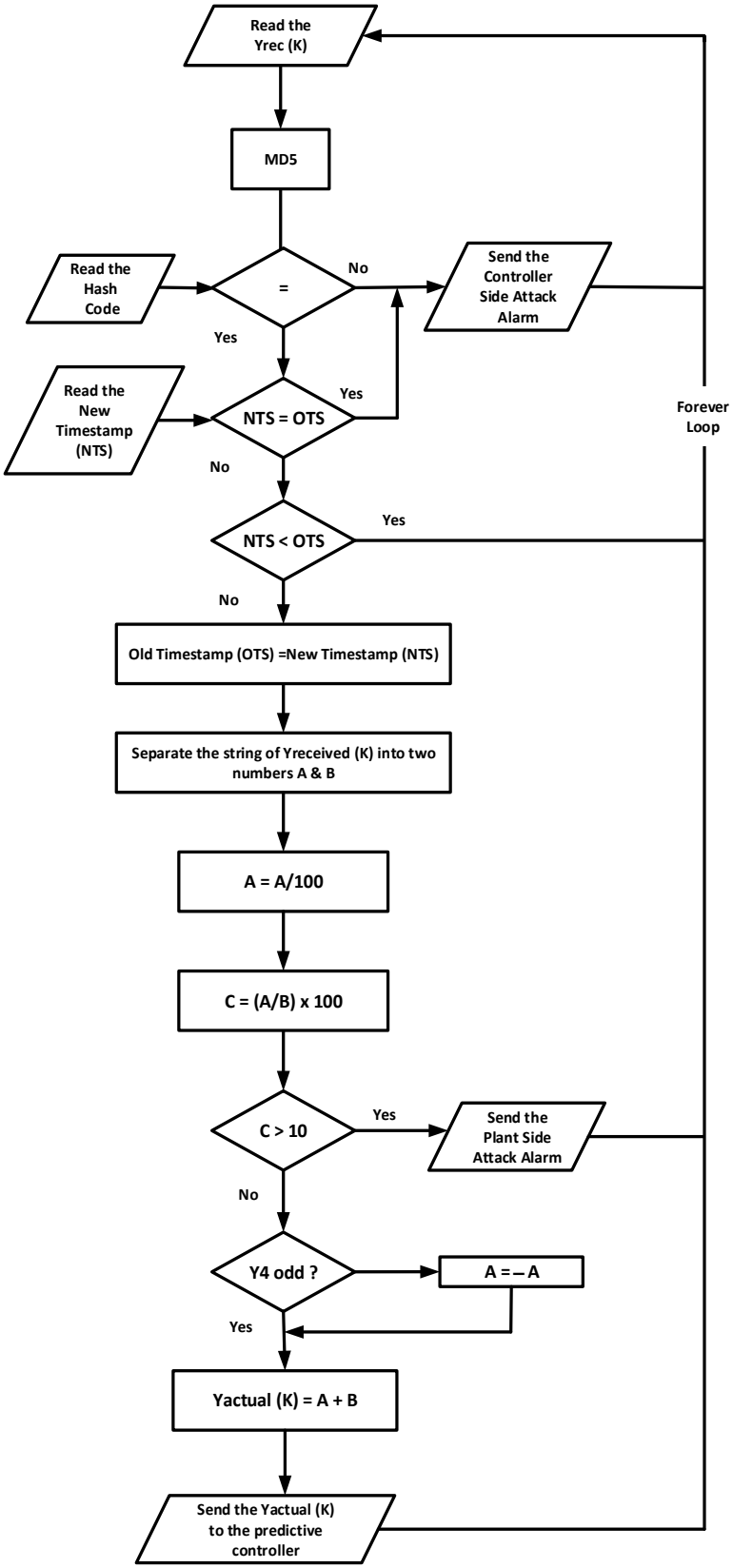

Figure 3.11: Overall controller side attack-tolerant software algorithm.

In order to extracting the actual sensor reading and to perform the plant side attack detection, the follows steps will be applied, these steps are only compatible to the algorithm shown in Fig. 3.10, and Fig. 3.11, illustrates overall controller side attack-tolerant software algorithm which can be described as following:

- Separate the message of *Yrec(K)* into two strings *A= "Y3 Y2 Y1 Y0"* and *B= "Y7 Y6. Y5 Y4"*.
- Convert the two strings into their two equivalent numbers.
- Divide ***A*** by 100 to have number in format of *(Y3 Y2. Y1 Y0)*.
- Calculate the percentage between the ***A*** & ***B*** in order to send the attack alarm if any (up to 10% is accepted in this work).
- Specify the sign of ***A*** according to the value of *Y4* (i.e. positive for odd or negative for even).
- In order to have the actual sensor reading, ***A*** will be added to ***B***.

3.7 Simulation results

The proposed security mechanism for NCS is developed as per the scheme mentioned above with MATLAB. Position control of networked DC servomotor based on generalized predictive control (the details of this control method are available in chapter 4) is selected for testing and for verifying the performance of the proposed security system in this chapter. The parameters and values chosen for motor modeling are given in Table 3.1. By using these parameters, we can get (3.1) which represents the transfer function of DC servo motor for controlling position [87].

Table 3.1: DC servomotor parameters

Parameter	***Nomenclature***	***Value***
Moment of inertia	Jm	0.000052 Kg.m^2
Friction coefficient	*Bm*	*0.01 N.ms*
Back EMF constant	*Kb*	*0.235 V/rad S^{-1}*
Torque constant	*Ka*	*0.235 Nm/A*
Electric resistance	*Ra*	*2 Ohm*
Electric inductance	*La*	*0.23 H*

$$G_{position}(s) = \frac{19640}{s^3 + 201s^2 + 6290s} \tag{3.1}$$

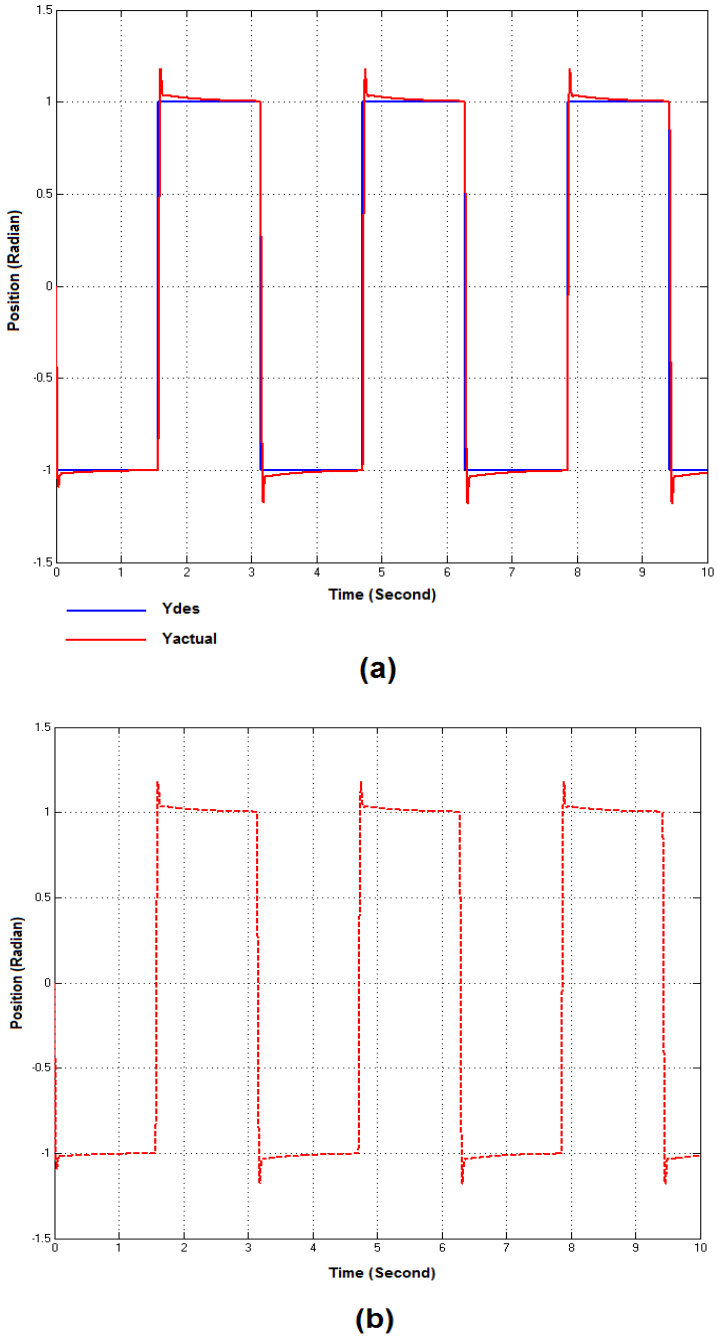

Figure 3.12: Position control over the secure NCS without attacks (a) Actual system response (b) The received sensor signal.

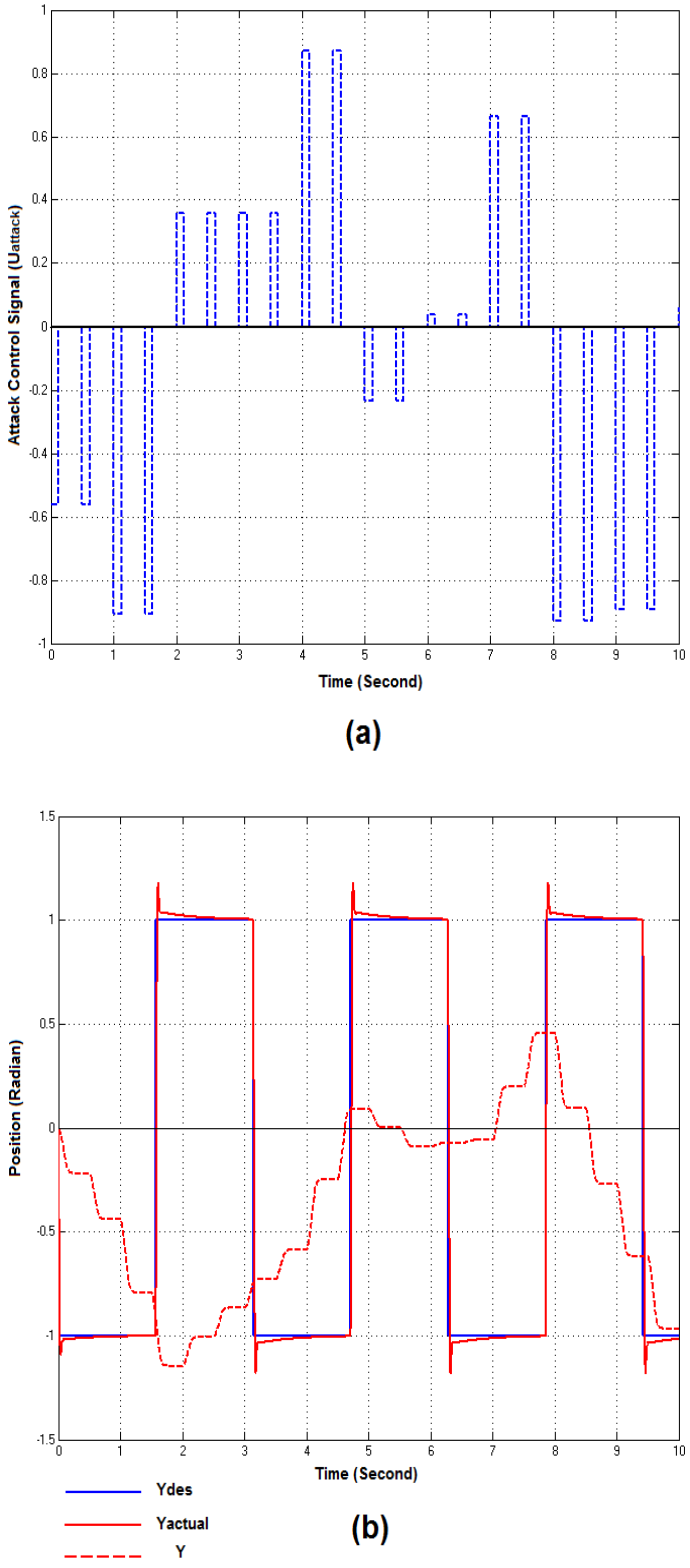

Figure 3.13: Position control over the secure NCS with 25% attacks (a) Attack control signal (b) Actual response and the response to the attacker, desired (blue), actual response(red), and the response to be sent to the attacker (dotted red).

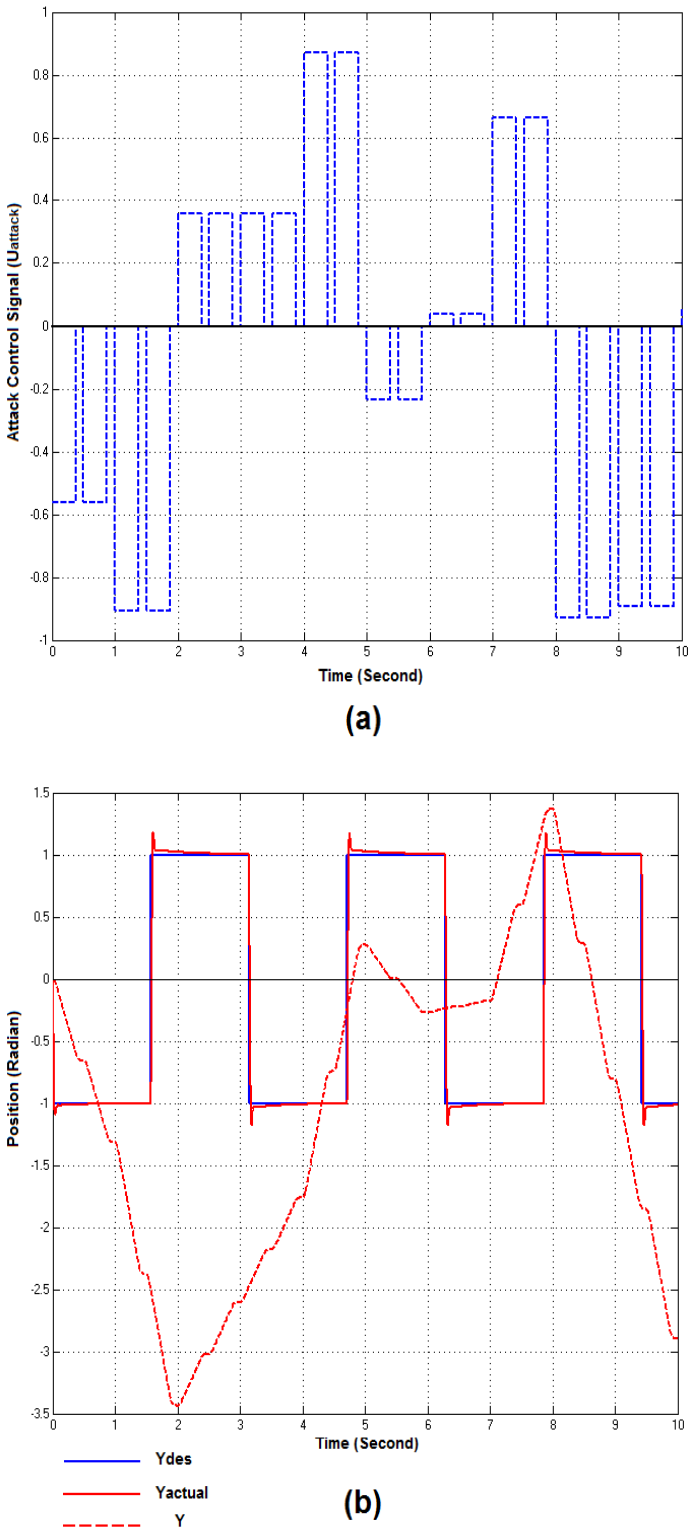

Figure 3.14: Position control over the secure NCS with 75% attacks (a) Attack control signal (b) Actual response and the response to the attacker, desired (blue), actual response(red), and the response to be sent to the attacker (dotted red).

Initially, the system is tested without any attack; the test result is illustrated in Fig.3.12. After that, the simulation is carried out by injecting of external attack control signals *Uattack(K)* to the proposed secure NCS. The *Uattack(K)* is represented by pulses with random values of amplitude (between -1 to 1). At the first time, the rate of data modification attack signal is selected to be 25% as in Fig.3.13, and after that, it is increased to 75% as in Fig.3.14.

We can see that the security mechanism maintains the normal response of the position control system (continue lines in Fig.3.13 (b) and Fig.3.14 (b)).

However, the sensor reading *Y* (dotted red lines in Fig.3.13 (b) and Fig.3.14(b)) follows what the remote plant response to the attacker signal '*Uattack*' should be.

3.8 Conclusion

In this chapter, attack-tolerant scheme for networked control system is introduced. The proposed method is based on deception of the cyber-attack when the current available data encryption method is broken.

In addition to the continuity of the control system work, the proposed technique makes the attacker to stop the development of his attack method. An attack alarm will be sent to controller side and this will give enough time for the authorized person to repair the broken part in the security mechanism.

Certainly, DES is an old encryption method and has been broken before, but it is used in this chapter because it causes a small delay time, and the most important is the objective of this research which is not to know how to break encryption tools, but how to maintain the correct performance of the control system in case of broken encryption.

In order to introduce the proposed technique, simple algorithm for sensor reading formation is described. This algorithm can be extended to be more complex with an advanced encryption tool, which can be specified according to the security requirements of the control system application.

In the following chapters, we look at the control system to see, similarly, how it can help make the system tolerant to attacks when digital interfaces have not made it possible to stop it.

CHAPTER 4

Controller Design for NCS with Time-Varying Delay

Contents

4.1 Introduction

As mentioned in the first chapter, the current objective of control systems and industrial automation is to avert the physical co-location of controllers, actuators, and sensors either by the possibility of modifying wired communication by wireless communication or by necessity in the case of moving parts that cannot be wired.

However, the use of a shared medium like Ethernet or Internet or communication like the wireless as well as the using of IT security tools (as in chapter three) results in random delay between the controller and the actuator also between the sensor and the controller.

Moreover, packets may be lost due to noise, traffic congestion, or even packet refusing process to avoid the wrong sequence or reply attack. Therefore, the main control challenge is to design a controller that able to compensate the effects of random delay and packet loss.

Therefore, several researches were tried to deal with the problem of time-delays by using different methods. The largest proportion of the used tools were relative to predictive and robust control. The authors in [88-92] applied different model predictive control methods to solve mentioned problem. The paper of [88] discussed the methodology to use multi-step predictive control increment with queue sequencing to compensate for Controller-to-Actuator time delay in NCS. The authors in [89] applied the Bilateral Generalized Predictive Controller (BGPC) to ensure the stability in the presence of the environment and transmission time-delays uncertainties. The state-based networked predictive control approach was proposed to compensate for the network communication delay in [90]. In order to overcome the influences of time-varying delays on the NCS performance, the Generalized Predictive Controller GPC was used in [91] while in [92], the implicit GPC was combined with Extreme Learning Machine (ELM) to predict the delay.

Related to the robust control, an analysis method for H∞ performance of NCS with the effects of both network-induced delay and data dropout was provided in [93] by introducing some slack matrix variables and employing information of the lower bound of the network induced delay. A proposal for consensus control in directed networks of agents with time-delay was described in [94], this proposal was based on reduced-order system conditions under which all agents reach consensus with the desired H∞ performance. The

robust H2 and H∞ step tracking control methods for networked control systems subject to random time delays modeled by Markov chains were investigated in [32]. According to the stochastic characteristic of the time delays and packet dropouts, a model based on a Markov jump system structure was proposed in [95] to randomly compensate for the adverse effect of the two channels time delays and packet dropouts. The design problem of optimal robust non-fragile H∞ state feedback controller for NCS with uncertain time delays and controller gain perturbations was detailed in [96].

These various approaches give good results in NCS, but require an accurate modeling effort. Model free control should be use to define a new controller without mathematical model of the plant.

In this chapter, the design of suitable controller for NCS with time-varying delay will be introduced by two methods. The first one is by an improvement of model-free control against the effects of random time delay while the second method is based on applying the intelligent mechanism of model-free control along with the Generalized Predictive Control (GPC). The details of these two methods will be presented in the next sections.

4.2 Model-Free control

The model-free control scheme was introduced in [97-98], this scheme was applied along with PID controller (also known as intelligent PID controller or iPID controller) and has led to numerous convincing applications.

Model-free control is a design to nonlinear control system, this approach was described in more details in [99]. The discrete-time version of model-free control was presented in [100]. Model-free control consists in trying to estimate via the input and the output measurements what can be compensated by control in order to achieve a good output trajectory tracking. This implies the construction of a purely numerical model also called "local model" of the plant that can be written as:

$$y^{(\nu)} = F + \alpha u \tag{4.1}$$

where, u , y are the input and output of the plant respectively, $\alpha \in \mathbb{R}$ is a non-physical parameter, F represents the nonlinear term of the plant and can be compensated from the knowledge of the input-output behavior of the plant. The order $\nu \in \mathbb{N}$ of the numerical

model (4.1) is a necessarily design parameter that can be arbitrarily chosen. However, if we assume that the relative dominant order of the plant is known then *v* will be equal to this order.

There are two approaches to specify the value of α. The first one is represented by considering it as a constant design parameter [99], and the second one is to consider α as time-varying parameter [101].

For more robustness, the second consideration will be used in this thesis. The interested readers might refer to [101] for a complete presentation. The value of α will be calculated at each sample time rather than selecting it as a constant parameter. The $\alpha(k)$ can be calculated using the following equation,

$$\alpha(k) = \frac{u(k-1)[y^{(v)}(k)]_e + \mu(k)}{1 + u^2(k-1)} \tag{4.2}$$

where, $[y^{(v)}(k)]_e$ is the estimation of the *v* derivative of the output that can be laid at time *k*, *u(k−1)* is the control input that has been applied to the plant during the previous sampling period, and,

$$\mu(k) = [u(k-1) - 1]\, y(k) \tag{4.3}$$

In the present approach, the quantity *F* in (4.1) is updated at each sampling time from the measurement of the output and the knowledge of the input. At sampling time *k*, the estimation of *F* is:

$$[F(k)]_e = [y^{(v)}(k)]_e - \alpha(k)u(k-1) \tag{4.4}$$

Close the loop via the intelligent controller,

$$u(k) = -\frac{[F(k)]_e}{\alpha(k)} + \frac{y^{*(v)}(k) - \Omega(\ell(k))}{\alpha(k)} \tag{4.5}$$

where, $y^{*(v)}(k)$ is the *v* derivative of reference trajectory, $\ell(k) = y(k) - y^*(k)$ is the tracking error and $\Omega(\ell(k))$ is closed loop feedback controller.

Based on the numerical knowledge of *F*, the control for sampling period *k* is calculated from (4.5) as a simple cancellation of the nonlinear term *F* plus a closed loop tracking of a reference trajectory.

4.2.1 The derivatives estimation of plant output

In this section, the derivative estimation method, which described in [100] will be used to estimate the derivatives of the actual delay-free plant output. Let y_m be a measured value of the delay-free plant output *y*, y_m is the image of *y* and we consider that *y* is distorted by some of added noise η, therefore we have: $y_m = y + \eta$.

The objective is to estimate the derivatives of the delay-free plant output *y*, up to a finite order of derivation, from its measurement y_m observed on a given time interval. The Taylor expansion of the controller output around 0 is given by

$$y(\tau) = \sum_{n=0}^{\infty} y^{(n)}(0) \frac{\tau^n}{n!} \tag{4.6}$$

By the polynomial we can approximate *y(t)* for the interval *[0,T]*, *T >0*.

$$y_N(\tau) = \sum_{n=0}^{N} y^{(n)}(0) \frac{\tau^n}{n!} \tag{4.7}$$

For *N* degree Φ_N is the operational analogue of y_N and can be written as

$$\Phi_N(s) = \frac{y(0)}{s} + \frac{\dot{y}(0)}{s^2} + \cdots + \frac{y^{(N)}(0)}{s^{N+1}} \tag{4.8}$$

By applying a convenient operator to $\Phi_N(s)$, we can separate each coefficient $y^{(i)}(0)$ appearing in previous expression. Thus,

$$\forall i = 0, \dots, N$$
$$\frac{y^{(i)}(0)}{s^{2N+1}} = \frac{(-1)^i}{N!\,(N-i)!} \cdot \frac{1}{s^{N+1}} \cdot \frac{d^i}{ds^i} \cdot \frac{1}{s} \cdot \frac{d^{N-i}}{ds^{N-i}} (s^{N+1} \Phi_N(s)). \tag{4.9}$$

The expression of $y^{(i)}(0)$ in the time domain can be written as

$$y^{(i)}(0) = \int_0^T P(\delta; T)\, y_N(\delta)\, d\delta \tag{4.10}$$

where $P(\delta;T)$ is polynomial in δ and *T*. Notice that (4.10) gives the calculation of $y^{(i)}(0)$ from an integral on the time interval [0; *T]* for a given small *T > 0*.

As $\frac{d^i y(t-\delta)}{d\delta^i}\Big|_{\delta=0} = (-1)^i y^{(i)}(t)$ it is possible to express $y^{(i)}(t)$ as an integral, which includes values of y_N on the time interval *[t – T, t]*:

$$y^{(i)}(t) = (-1)^i \int_0^T P(\delta;T)\, y_N(t-\delta)\, d\delta \tag{4.11}$$

By using the noisy signal y_m, a simple estimator of the derivative $y^{(i)}(t)$ can be expressed as

$$\langle\langle y^{(i)}\rangle\rangle_T^c\,(t) = (-1)^i \int_0^T P(\delta;T)\, y_m(t-\delta)\, d\delta \tag{4.12}$$

(4.12) is realized from (4.11) by changing y_N by y_m. Note that the integral operation works according to the low-pass filter rule by reducing the noise that distorts y_m. The choice of *T* results in a trade-off, small value of *T* leads to the effect of the noise; the large value of *T* reducing the effect of the noise but there is truncation error.

In practice, the integral expressed in (4.12) is obtained by a numerical integration method; therefore, the estimator $\langle\langle y^{(i)}\rangle\rangle_T^c\,(t)$ will be performed at each sample of *k*.

Let *Ts* is the sampling period, then the discretization of any continuous time function *f* will be denoted by *f* [k] i.e.

$$f[k] = f(k, Ts), k \in Z \tag{4.13}$$

With these notations, the discrete-time approximation of the derivative estimation of the delay-free plant output is simply a discrete sum that can be written as:

$$\langle\langle y^{(v)}\rangle\rangle_{T_s,n_s}^d\,[k] = \sum_{j=0}^{n_s} w(j)\, P(jT_s; n_sT_s) y_m[k-j] \tag{4.14}$$

where, *ns* the number of samples used in the time window $T = n_sT_s, v = i$, and the *w(j)* is the weight related to the used numerical integration method.

4.2.2 Applying model-free control in NCS

The applications of model-free control in Networked Control System (NCS) is still limited because of its weakness to process the time-varying delay which usually appears in this type of networks. This thesis proposes a new method to adapt model-free controller to

improve its use in NCS. This method suggests a new structure of model-free control. The main idea is based on mutual benefit between Smith predictor and the basic model-free controller.

Smith predictor was first introduced in [102] as a dead time compensator. Since then, several important researches have been carried out along with this compensator. The block diagram of Smith predictor is illustrated in Fig.4.1, it consists of a traditional feedback loop plus an internal loop that provides two additional terms directly into the feedback pathway. The first term is an estimate of what the process variable would look like in the time delay free condition. It is generated by passing the controller output through a plant model. If the plant model is accurate then the output will be a delay-free version of the actual plant variable.

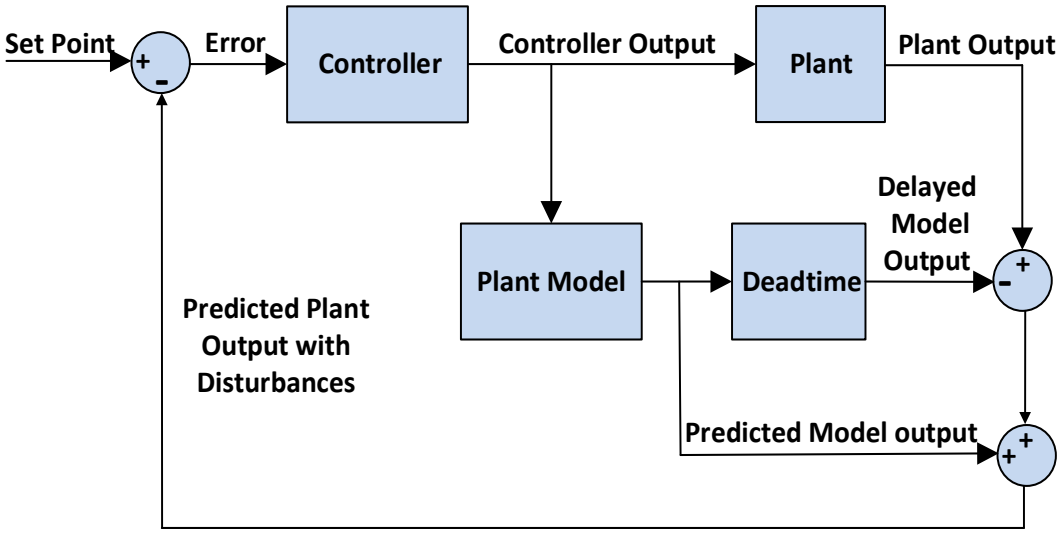

Figure 4.1: The block diagram of classic Smith predictor.

The mathematical model used to generate the estimated delayed plant output consists of two models hooked up in series. The first model represents all of the plant behavior not related to deadtime. The second model represents the deadtime. Subtracting the estimated delayed plant output from the received actual plant output yields an estimate of the disturbances. By adding this difference to the predicted plant variable, Smith predictor will create a feedback variable that includes the effect of disturbances, but not the deadtime. Unfortunately, a simple matter to generate the plant model moreover the time delay is not always constant especially in NCS application. These obstacles made the Smith predictor is useless in the NCS.

The compensation for nonlinear term in the basic model-free controller classic is not sufficient to compensate for time-varying delay in NCS [103]. Therefore, a new corporation will be made in this thesis between the basic-model free controller and the classic Smith predictor, this will be implemented by replacing the mathematical model of the plant in Smith predictor by the integration of ultra-local model (4.1) which is constantly updated within the algorithm of the basic model-free controller. On the other hand, Smith structure will eliminate the effect of time delay from the feedback variable by moving the time delay outside the control loop.

$$\hat{y}(t) = \int_0^t F(\tau) + \alpha(\tau)\, u(\tau)\, d\tau \tag{4.15}$$

where, $\hat{y}(t)$ is the estimated delay-free version of the plant output.

The predefined time delay value which is important requirement in the classic Smith predictor will be replaced by the automatic computation method of time delay which is suggested in [104] and can be summarized as following.

Let,

$$\frac{d\mathcal{t}_d(t)}{dt} = -K_{\mathcal{t}_d.}(y(t) - \hat{y}(t)) \tag{4.16}$$

where, $\mathcal{t}_d(t)$ is the time-varying delay and $K_{\mathcal{t}_d.}$is a constant

Hence,

$$\mathcal{t}_d(t) = \mathcal{t}_d^{\circ} + K_{\mathcal{t}_d} \int_0^t \hat{y}(\xi) - y(\xi)\, d\xi \tag{4.17}$$

where, $\mathcal{t}_d^{\circ}$ is the initial value of the delay which specified during the design time.

Some of corrective actions have to be added because the sign of $K_{\mathcal{t}_d}$ must agree with the modification of the setpoint, also the adaption law has to be disabled when the plant output reaches the steady state. Indeed, after the plant output reaches the steady state, the disturbance may be not understanding as a variation of the controlled variable and so leads to wrong change $\mathcal{t}_d(t)$. In order to prevent this problematic, the following formula [105] will be utilized:

$$sign\left(K_{\mathcal{t}_d}\right) = sign\left[\frac{\frac{2(z-1)}{T(z+1)}}{1 + \frac{2(z-1)}{(z+1)}}\, Y(z)\right] \tag{4.18}$$

By using (4.18), the time delay would be rapidly computed and as a result, improves the robustness of the proposed model-free control structure. The block diagram of the complete proposed system is illustrated in Fig.4.2.

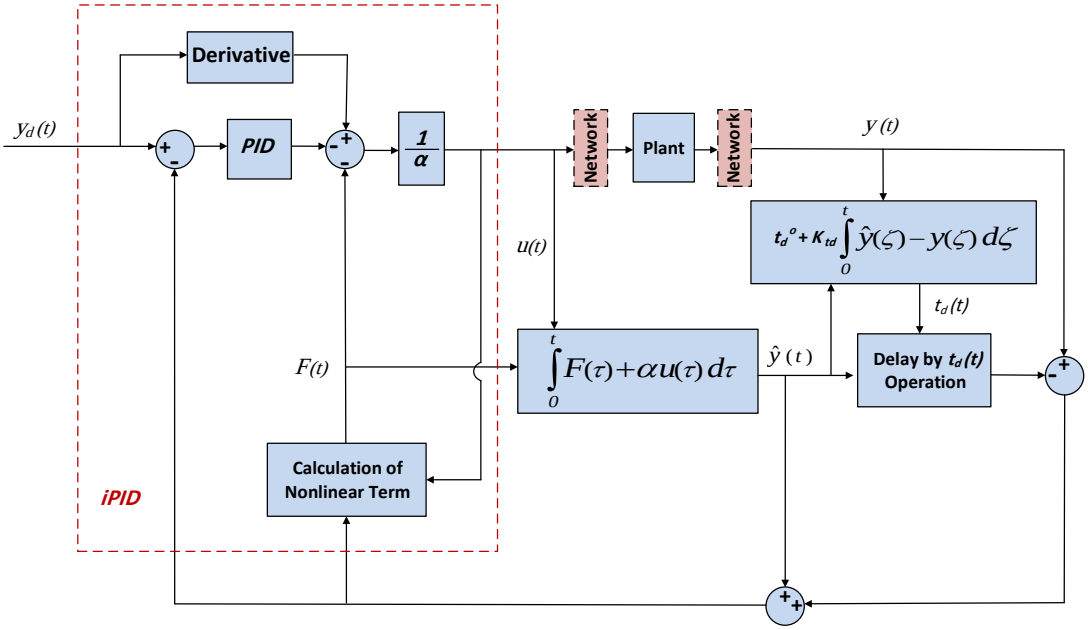

Figure 4.2: The Model-free control and Smith structure.

An application of this proposal will be given at the end of this chapter which shows some obtained results. The mutual benefit between Smith predictor and the basic model-free controller leads to completely model-free Smith predictor. However, the core of this proposal was represented by model-free controller which is utilizing PID algorithm, therefor and for more development, we will try to apply the intelligent structure of model-free control along with predictive control which is much used in NCS.

4.3 Applying the intelligent generalized predictive control in NCS

The generalized predictive control (GPC) belongs to a class of the model-based predictive control (MBPC) methods and was introduced by Clarke and his colleagues in 1987 [106-108]. The MBPC methods have been investigated and applied successfully in industry process control since the end of the 1970s and have continued to gain acceptance with growing computational ability of computers. These applications have confirmed that GPC has a good performance, robustness and efficiency against unmodeled disturbances as compared to traditional control techniques [109-112]. The GPC employs the receding horizon approach. Through the use of plant model, the GPC predicts the output of the plant

over a time horizon based on the assumption about future controller output sequences. An appropriate sequence of the control signals is then calculated to reduce the tracking error by minimizing a quadratic cost function. After which only the first element of the control signals is applied to the system. This process is repeated for each sample interval. Accordingly, new information is updated at each sample interval. Due to this methodology, the GPC gives good rejection against modeling errors and disturbances.

However, NCS will produce time delay and packet dropout. As a result, it becomes much harder to use the regular GPC in this type of networks. In this thesis, a new strategy to eliminate these negative effects will be introduced. The intelligent control scheme which described in section 4.2 will be applied along with GPC. The new controller scheme will be called the Intelligent Generalized Predictive Controller (IGPC) and will be designed to take advantage of the prediction in GPC strategy along with the ability to cancel the nonlinear terms which provided by structure of the intelligent controller.

In this section, a more flexible form of the GPC strategy that was detailed by the author of [113] will be adopted for the networked controller design.

The generalized predictive control algorithm used the Controlled Auto-Regressive Integrated Moving Average model referred to as CARIMA, it is used because the uncertainty is included in a manner that is a good representation of slowly varying disturbances.

$$[a(z)\Delta]\, y(k) = b(z)[\Delta u(k)] \tag{4.19}$$

where, $y(k)$ and $u(k)$ are the system output and input respectively, $\Delta = 1 - z^{-1}$ is the difference operator, and

$$a(z) = 1 + a_1 z^{-1} + \cdots + a_n z^{-n} \tag{4.20}$$

$$b(z) = 1 + b_1 z^{-1} + \cdots + b_m z^{-m} \tag{4.21}$$

Combine a(z) and Δ

$$A(z)y(k) = b(z)\Delta u(k);\ A(z) = a(z)\Delta$$

$$A(z) = 1 + A_1 z^{-1} + \cdots + A_n z^{-n} \tag{4.22}$$

Discrete models are one-step ahead prediction models, that is, given data at sample *k*, data can be determined at sample *k+1*.

$$y(k+1) + A_1\, y(k) + \cdots + A_n\, y(k-n+1) = b_1 \Delta u(k) + b_2 \Delta u(k-1) + \cdots + b_m \Delta u(k-m+1) \tag{4.23}$$

As this prediction model implicit within the use of increments, there is no need for the disturbance estimation. The one-step ahead prediction can be used recursively to find an n-step ahead prediction as follows:

$$\begin{aligned}
&y(k+1) + A_1\, y(k) + \cdots + A_n\, y(k-n+1) \\
&\qquad = b_1 \Delta u(k) + b_2 \Delta u(k-1) + \cdots + b_m \Delta u(k-m+1) \\
&y(k+2) + A_1\, y(k+1) + \cdots + A_n\, y(k-n+2) \\
&\qquad = b_1 \Delta u(k+1) + b_2 \Delta u(k) + \cdots + b_m \Delta u(k-m+2) \\
&y(k+3) + A_1\, y(k+2) + \cdots + A_n\, y(k-n+3) \\
&\qquad = b_1 \Delta u(k+2) + b_2 \Delta u(k+1) + \cdots + b_m \Delta u(k-m+3) \\
&y(k+4) + A_1\, y(k+3) + \cdots + A_n\, y(k-n+4) \\
&\qquad = b_1 \Delta u(k+3) + b_2 \Delta u(k+2) + \cdots + b_m \Delta u(k-m+4)
\end{aligned}$$

Simultaneous equations can be represented using matrix/vector format and the shorthand notation to give a compact description of the entire predictions.

$$C_A \begin{bmatrix} y(k+1) \\ y(k+2) \\ y(k+3) \\ y(k+4) \end{bmatrix} + H_A \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-n+1) \end{bmatrix} = C_b \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \Delta u(k+3) \end{bmatrix} + H_b \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \Delta u(k-m+1) \end{bmatrix} \tag{4.24}$$

$$C_A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ A_1 & 1 & 0 & 0 \\ A_2 & A_1 & 1 & 0 \\ A_3 & A_2 & A_1 & 1 \end{bmatrix};\ H_A = \begin{bmatrix} A_1 & A_2 & \dots & A_{n-4} & A_{n-3} & \dots & A_{n-1} & A_n \\ A_2 & A_3 & \dots & A_{n-3} & A_{n-2} & \dots & A_n & 0 \\ A_3 & A_4 & \dots & A_{n-2} & A_{n-1} & \dots & 0 & 0 \\ A_4 & A_5 & \dots & A_{n-1} & A_n & \dots & 0 & 0 \end{bmatrix}$$

$$C_b = \begin{bmatrix} b_1 & 0 & 0 & 0 \\ b_2 & b_1 & 0 & 0 \\ b_3 & b_2 & b_1 & 0 \\ b_4 & b_3 & b_2 & b_1 \end{bmatrix};\ H_b = \begin{bmatrix} b_2 & b_3 & \dots & b_{n-4} & b_{n-3} & \dots & b_{n-1} & b_n \\ b_3 & b_4 & \dots & b_{n-3} & b_{n-2} & \dots & b_n & 0 \\ b_4 & b_5 & \dots & b_{n-2} & b_{n-1} & \dots & 0 & 0 \\ b_5 & b_6 & \dots & b_{n-1} & b_n & \dots & 0 & 0 \end{bmatrix}$$

Rewrite by using the '$arrow$' notation as follows:

$$C_A \underset{\rightarrow(k+1)}{y} + H_A \underset{\leftarrow(k)}{y} = C_b \underset{\rightarrow(k)}{\Delta u} + H_b \underset{\leftarrow(k-1)}{\Delta u} \tag{4.25}$$

where, right arrow implies prediction and left arrow implies past data. The prediction variables are in compact matrix/vector format, it is easily to re-arrange to find the dependence of the output predictions upon past (or known) data and the decision variables

$$\underset{\rightarrow(k+1)}{y} = C_A^{-1} C_b \underset{\rightarrow(k)}{\Delta u} + C_A^{-1} H_b \underset{\leftarrow(k-1)}{\Delta u} - C_A^{-1} H_A \underset{\leftarrow(k)}{y} \tag{4.26}$$

or,

$$\underset{\rightarrow(k+1)}{y} = H \underset{\rightarrow(k)}{\Delta u} + P \underset{\leftarrow(k-1)}{\Delta u} + Q \underset{\leftarrow(k)}{y} \tag{4.27}$$

where, $H = C_A^{-1} C_b; P = C_A^{-1} H_b; Q = -C_A^{-1} H_A$

The performance index (assuming simple weights) is given as:

$$J = \underset{\rightarrow(k+1)}{e^T} \underset{\rightarrow(k+1)}{e} + \lambda \underset{\rightarrow(k)}{\Delta u^T} \underset{\rightarrow(k)}{\Delta u} \tag{4.28}$$

$$\underset{\rightarrow(k+1)}{e} = \underset{\rightarrow(k+1)}{r} - \underset{\rightarrow(k+1)}{y} \tag{4.29}$$

$$J = \left[r^T_{\rightarrow(k+1)} - y^T_{\rightarrow(k+1)} \right] \left[\underset{\rightarrow(k+1)}{r} - \underset{\rightarrow(k+1)}{y} \right] + \lambda \underset{\rightarrow(k)}{\Delta u^T} \underset{\rightarrow(k)}{\Delta u} \tag{4.30}$$

where, λ is weighting coefficient.

It is essential to select a subset of the future inputs and assuming the other values are known. Practically, this is done by supposing the input within the predictions becomes fixed after *nu* control horizon steps.

$$\underset{\rightarrow(k)}{\Delta u} = \begin{bmatrix} \Delta u(k) \\ \vdots \\ \Delta u(k+n_u-1) \\ 0 \\ 0 \\ \vdots \end{bmatrix} \tag{4.31}$$

The predictions in (4.27) include a square matrix *H*. However, given we know the structure of the $\underset{\rightarrow(k)}{\Delta u}$

$$H \underset{\rightarrow(k)}{\Delta u} = [H_1 \quad H_2] \begin{bmatrix} \Delta u(k) \\ \vdots \\ \Delta u(k+n_u-1) \\ 0 \\ 0 \\ \vdots \end{bmatrix} = H_1 \underset{\rightarrow(k)}{\Delta u} \tag{4.32}$$

For practical convenience, only *H* will be written (not H_1) and assume that this is defined to have the relevant number of columns (equivalent to n_u control horizon).

Substituting predictions in (4.27) into (4.30) give the following:

$$\underset{\rightarrow(k+1)}{y} = H\,\underset{\rightarrow(k)}{\Delta u} + P\,\underset{\leftarrow(k-1)}{\Delta u} + Q\,\underset{\leftarrow(k)}{y} \tag{4.33}$$

where, $H = C_A^{-1}C_b; P = C_A^{-1}H_b; Q = -C_A^{-1}H_A$

The performance index (assuming simple weights) is given as:

$$J = \underset{\rightarrow(k+1)}{e}^T\,\underset{\rightarrow(k+1)}{e} + \lambda\,\underset{\rightarrow(k)}{\Delta u}^T\,\underset{\rightarrow(k)}{\Delta u} \tag{4.34}$$

$$\underset{\rightarrow(k+1)}{e} = \underset{\rightarrow(k+1)}{r} - \underset{\rightarrow(k+1)}{y} \tag{4.35}$$

$$J = \left[\underset{\rightarrow(k+1)}{r}^T - H\,\underset{\rightarrow(k)}{\Delta u}^T - P\,\underset{\leftarrow(k-1)}{\Delta u}^T - Q\,\underset{\leftarrow(k)}{y}^T\right]\left[\underset{\rightarrow(k+1)}{r} - H\,\underset{\rightarrow(k)}{\Delta u} - P\,\underset{\leftarrow(k-1)}{\Delta u} - Q\,\underset{\leftarrow(k)}{y}\right] + \lambda\,\underset{\rightarrow(k)}{\Delta u}^T\,\underset{\rightarrow(k)}{\Delta u} \tag{4.36}$$

$$\begin{aligned} J &= \left[\underset{\rightarrow(k+1)}{r}^T - P\,\underset{\leftarrow(k-1)}{\Delta u}^T - Q\,\underset{\leftarrow(k)}{y}^T\right]\left[\underset{\rightarrow(k+1)}{r} - P\,\underset{\leftarrow(k-1)}{\Delta u} - Q\,\underset{\leftarrow(k)}{y}\right] \\ &+ \left[(H\,\underset{\rightarrow(k)}{\Delta u})^T\right]\left[H\,\underset{\rightarrow(k)}{\Delta u}\right] - \left[2\left(H\,\underset{\rightarrow(k)}{\Delta u}\right)^T\right]\left[\underset{\rightarrow(k+1)}{r} - P\,\underset{\leftarrow(k-1)}{\Delta u} - Q\,\underset{\leftarrow(k)}{y}\right] \\ &+ \lambda\,\underset{\rightarrow(k)}{\Delta u}^T\,\underset{\rightarrow(k)}{\Delta u} \end{aligned} \tag{4.37}$$

$$\begin{aligned} &\underbrace{min}_{\underset{\rightarrow(k)}{\Delta u}}\ J \\ &\equiv \underbrace{min}_{\underset{\rightarrow(k)}{\Delta u}}\ \left[\left(H\,\underset{\rightarrow(k)}{\Delta u}\right)^T\right]\left[H\,\underset{\rightarrow(k)}{\Delta u}\right] - \left[2\left(H\,\underset{\rightarrow(k)}{\Delta u}\right)^T\right]\left[\underset{\rightarrow(k+1)}{r} - P\,\underset{\leftarrow(k-1)}{\Delta u} - Q\,\underset{\leftarrow(k)}{y}\right] \\ &+ \lambda\,\underset{\rightarrow(k)}{\Delta u}^T\,\underset{\rightarrow(k)}{\Delta u} \end{aligned} \tag{4.38}$$

Simplifying the performance index

$$\equiv \underbrace{min}_{\underset{\rightarrow(k)}{\Delta u}}\ \underset{\rightarrow(k)}{\Delta u}^T(H^TH + \lambda I)\underset{\rightarrow(k)}{\Delta u} - \left[2\left(H\,\underset{\rightarrow(k)}{\Delta u}\right)^T\right]\left[\underset{\rightarrow(k+1)}{r} - P\,\underset{\leftarrow(k-1)}{\Delta u} - Q\,\underset{\leftarrow(k)}{y}\right]$$

The performance index is quadratic; therefore, it has a unique minimum, which can be achieved using a gradient operator, that gradient is zero. Therefore, one can to find the gradient of *J* in function of the vector of future input increments.

$$grad\left(\underset{\rightarrow(k)}{\Delta u}^T(H^TH + \lambda I)\underset{\rightarrow(k)}{\Delta u} - \left[2\left(H\,\underset{\rightarrow(k)}{\Delta u}\right)^T\right]\left[\underset{\rightarrow(k+1)}{r} - P\,\underset{\leftarrow(k-1)}{\Delta u} - Q\,\underset{\leftarrow(k)}{y}\right]\right)$$

$$= 2(H^TH + \lambda I)\underset{\rightarrow(k)}{\Delta u} - 2H^T\left[\underset{\rightarrow(k+1)}{r} - P\,\underset{\leftarrow(k-1)}{\Delta u} - Q\,\underset{\leftarrow(k)}{y}\right]$$

But, $grad\,(J) = 0$

Hence,

$$2(H^T H + \lambda I)\underset{\rightarrow (k)}{\Delta u} = 2H^T \left[\underset{\rightarrow (k+1)}{r} - P\,\underset{\leftarrow (k-1)}{\Delta u} - Q\,\underset{\leftarrow (k)}{y}\right] \tag{4.39}$$

And

$$\underset{\rightarrow (k)}{\Delta u} = (H^T H + \lambda I)^{-1} H^T \left[\underset{\rightarrow (k+1)}{r} - P\,\underset{\leftarrow (k-1)}{\Delta u} - Q\,\underset{\leftarrow (k)}{y}\right] \tag{4.40}$$

Even though the optimization sets out a viable long-term plan of future control increments, only the first of these is carried out as need to improve any other options. The first value of the optimum input trajectory will be used to define the control law.

Assume that the vector of future input increments is structured by sample; hence, the first block can be extracted as follows:

$$\underset{\rightarrow (k)}{\Delta u} = \begin{bmatrix} \Delta u(k) \\ \vdots \\ \Delta u(k + n_u - 1) \end{bmatrix} \Rightarrow \Delta u(k) = \underbrace{[I, 0, \cdots, 0]}_{E_1{}^T}\ \underset{\rightarrow (k)}{\Delta u}$$

Hence GPC control law is

$$\Delta u(k) = E_1^T (H^T H + \lambda I)^{-1} H^T \left[\underset{\rightarrow (k+1)}{r} - P\,\underset{\leftarrow (k-1)}{\Delta u} - Q\,\underset{\leftarrow (k)}{y}\right] \tag{4.41}$$

$$\Delta u(k) = P_r\,\underset{\rightarrow (k+1)}{r} - P_r\,P\,\underset{\leftarrow (k-1)}{\Delta u} - P_r\,Q\,\underset{\leftarrow (k)}{y} \tag{4.42}$$

$$\Delta u(k) = P_r\,\underset{\rightarrow (k+1)}{r} - \breve{D}(k)\underset{\leftarrow (k-1)}{\Delta u} - N(k)\,\underset{\leftarrow (k)}{y} \tag{4.43}$$

where,

$$P_r = E_1^T (H^T H + \lambda I)^{-1} H^T$$

$$\breve{D}(k) = P_r\,P$$

$$N(k) = P_r\,Q$$

Expand out each of the vectors,

$$P_r = \left[P_1, P_2, \cdots, P_{n_y}\right]$$

$$\breve{D}(k) = \left[D_1, D_2, \cdots, D_{n_b - 1}\right]$$

$$N(k) = \left[N_0, N_2, \cdots, N_{n_a}\right]$$

Dimensions linked to P, Q which in turn are linked to model transfer function $G = b(z)/a(z)$.

n_y = Number of terms dictated by prediction horizon.

$n_b - 1$ = Number of terms one less then number of terms in $b(z)$ of CARIMA model.

n_a = Number of terms one more than in $a(z)$ in CARIMA model. Hence,

$$\Delta u(k) = \left[P_1, P_{2,\cdots}, P_{n_y}\right] \underset{\rightarrow (k+1)}{r} - \left[D_1, D_{2,\cdots}, D_{n_b-1}\right] \underset{\leftarrow (k-1)}{\Delta u} - \left[N_0, N_{2,\cdots}, N_{n_a}\right] \underset{\leftarrow (k)}{y} \qquad (4.44)$$

Rewrite using difference equations:

$$\begin{aligned} \Delta u(k) = & \left[P_1\, z + P_2\, z^2 + \ldots + P_{n_y} z^{n_y}\right] r(k) \\ & - \left[D_1\, z^{-1} + D_2 z^{-2} + \ldots + D_{n_b-1}\, z^{-n_b+1}\right] \Delta u(k) \\ & - \left[N_0 + N_1\, z^{-1} \right. \\ & \left. + \ldots + N_{n_a}\, z^{-n_a}\right] y(k) \end{aligned} \qquad (4.45)$$

Then the transfer function format is

$$\underbrace{\left[1 + D_1\, z^{-1} + D_2 z^{-2} + \ldots + D_{n_b-1}\, z^{-n_b+1}\right]}_{D_k(z)} \Delta u(k) = \underbrace{\left[P_1\, z + P_2\, z^2 + \ldots + P_{n_y} z^{n_y}\right]}_{P_r(z)} r(k) - \underbrace{\left[N_0 + N_1\, z^{-1} + \ldots + N_{n_a}\, z^{-n_a}\right]}_{N_k(z)} y(k) \qquad (4.46)$$

Or, simplifying

$$D_k(z)\, \Delta u(k) = P_r(z)\, r(k) - N_k(z)\, y(k) \qquad (4.47)$$

A more conventional compensator form could be written as follows:

$$u(k) = [D_k(z)\, \Delta]^{-1}\, [P_r(z)\, r(k) - N_k(z)\, y(k)] \qquad (4.48)$$

The diagram of GPC is illustrated in Fig.4.3.

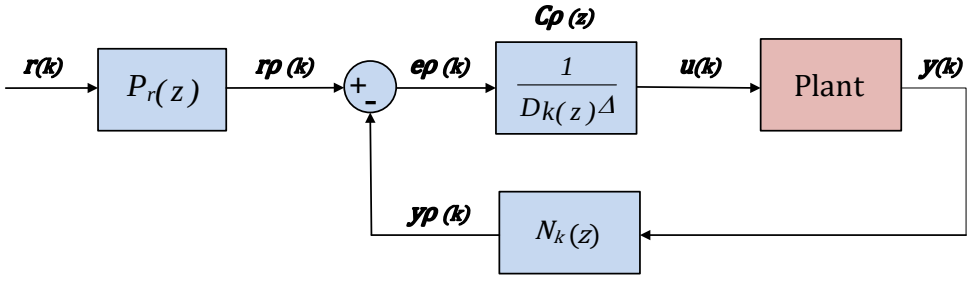

Figure 4.3: The block diagram of GPC.

The controller $\Omega(\ell(k))$ can be of various form (Delaleau (2014)). $\Omega(\ell(k))$ can be called intelligent controller when it is located within (4.5). Referring to Fig.4.3, it is possible to define the following:

The predicted plant output = $y\rho(k) = N_k(z)\, y(k)$ (4.49)

The planned reference trajectory = $r\rho(k) = P_r(z)\, r(k)$ (4.50)

The predicted error = $e\rho(k) = r\rho(k) - y\rho(k)$ (4.51)

The generalized predictive closed loop controller is,

$$C\rho(z) = [D_k(z)\, \Delta]^{-1} \tag{4.52}$$

From (4.1) we can write the predicted local model as

$$y\rho^{(v)}(k) = F\rho(k) + \alpha\rho(k)\, u(k-1) \tag{4.53}$$

and,

$$\alpha\rho(k) = \frac{u(k-1)\, y\rho^{(v)}(k) + \mu\rho(k)}{1 + u^2(k-1)} \tag{4.54}$$

$$\mu\rho(k) = [u(k-1) - 1]\, y\rho(k) \tag{4.55}$$

where, $F\rho(k)$ represents the prediction of nonlinear term of the plant. Given the fact that the prediction is also utilized from the measured plant output. $F\rho(k)$ can be compensated from the knowledge of the input and the derivative of predicted output of the plant.

$$F\rho(k) = y\rho^{(v)}(k) - \alpha\rho(k)\, u(k-1) \tag{4.56}$$

Hence,

$$u(k) = -\frac{F\rho(k)}{\alpha\rho(k)} + \frac{r\rho^{(v)}(k) - e\rho(k)\, C\rho(z)}{\alpha\rho(k)} \tag{4.57}$$

4.4 Simulation Results

The proposed controllers are developed as per the scheme mentioned above with simulation of remotely operated underwater vehicle ROV SUSD-02 connected to the controller over Ethernet. The dynamics of a 6-degree-of-freedom underwater vehicle was described in [114] as following:

$$M\dot{\nu} + Cr(\nu)\nu + D(\nu)\nu + g(x) = \tau \tag{4.58}$$

$$\dot{x} = J(x)\nu \tag{4.59}$$

where,

- $\nu = [u \quad v \quad w \quad p \quad q \quad r]^{Tr}$ is the body-fixed velocity vector in surge, sway, heave, roll, pitch and yaw.

- $x=[x \quad y \quad z \quad \phi \quad \vartheta \quad \psi]^{Tr}$ is the corresponding earth-fixed vector.
- M is a 6x6 symmetric positive definite inertia matrix.
- Cr is a 6x6 matrix of Coriolis and centripetal terms.
- D is a 6x6 dissipative matrix of hydrodynamic damping terms.
- $g(x)$ represents the restoring forces and moments.
- $J(x)$ is the transform matrix relating the body-fixed reference frame to the inertial reference frame.
- Tr is the input torque vector.

In this thesis, only ROV depth control will be considered, the ROV depth control equation was described in [115] as: -

$$m(\dot{w} - uq - z_G q^2 - x_G \dot{q} + y_G rq) = Z_u + Z_m + Z_d \tag{4.60}$$

where,

- m is the ROV quality, w is the vertical velocity,
- Z_G, X_G, Y_G are the gravity coordinates in the moving coordinate system,
- Z is the vertical driving force.
- Z_d is the total interference power in the vertical direction,
- Z_m is the summation of the buoyancy and gravity,
- Z_u is the vertical thrust.

The ROV is designed with an independent vertical propeller. The vertical thrust and the simplified underwater vehicle depth control transfer function were expressed in [114] as following.

$$Z_u = Ct \ \partial|\partial| \tag{4.61}$$

where, Ct is the thrust coefficient, ∂ is propeller speed.

$$Pm(s) = \frac{0.2466S + 0.1251}{S^3 + 1.8101S^2 + 0.9412S + 0.1603} \tag{4.62}$$

Firstly, the networked ROV is connected to the basic model-free controller (i.e. without time delay compensation) over Ethernet with added time-varying delay of zero second as minimum value and three seconds as maximum as shown in Fig.4.5. The test result of the ROV depth controlled by the basic model-free controller is illustrated is Fig.4.4, from this result it is easy to detect the instability of the system response.

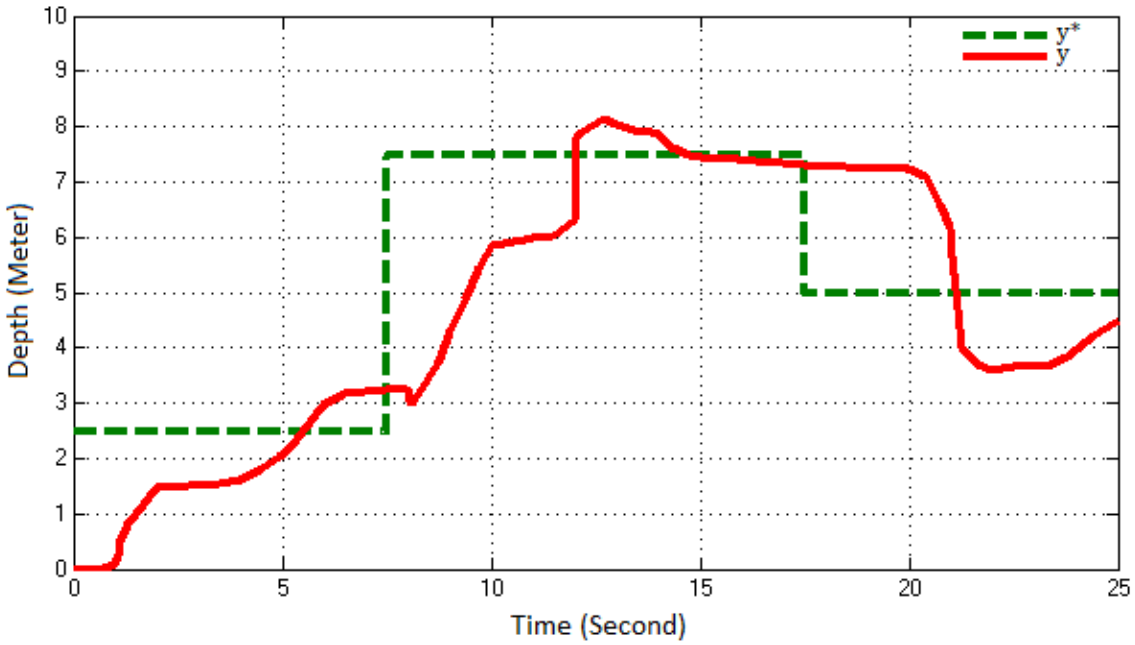

Figure 4.4: The response of networked ROV controlled by the basic model-free controller

After that, and in order to evaluate the performance of the suggested control system in the first part of this chapter, the proposed model-free controller with time delay compensation is connected the ROV over the same Ethernet with same time-varying delay function of Fig.4.5. The test results of model-free controller with time delay compensation are shown in Fig.4.6, from this figure the ROV has an acceptable response even with the presences of the mentioned time delay.

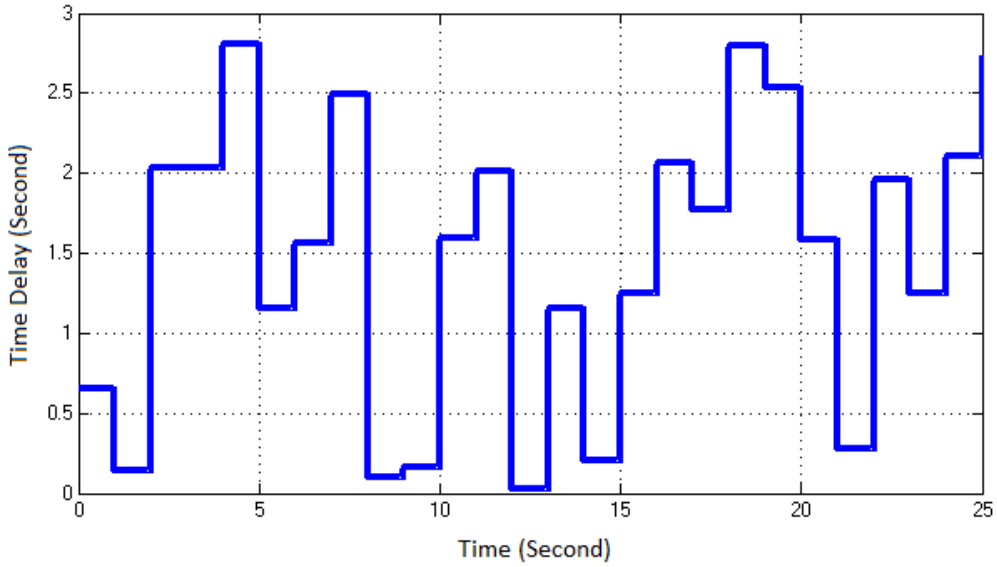

Figure 4.5: The time-varying delay of the network.

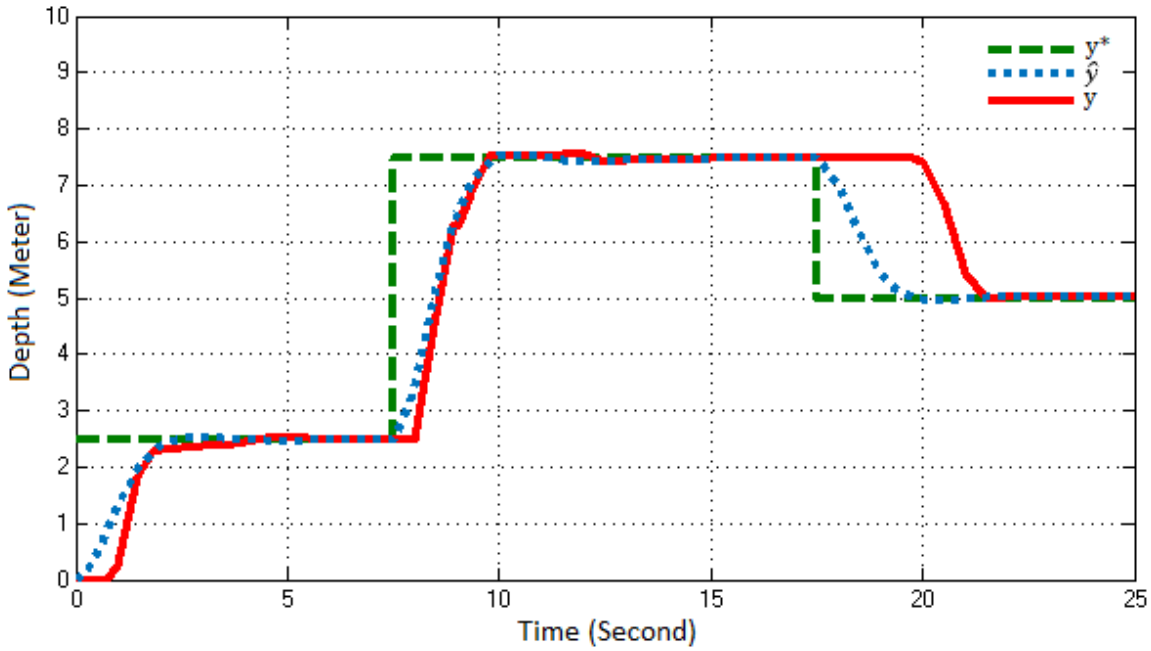

Figure 4.6: The response of networked ROV controlled by the model-free controller with time delay compensation.

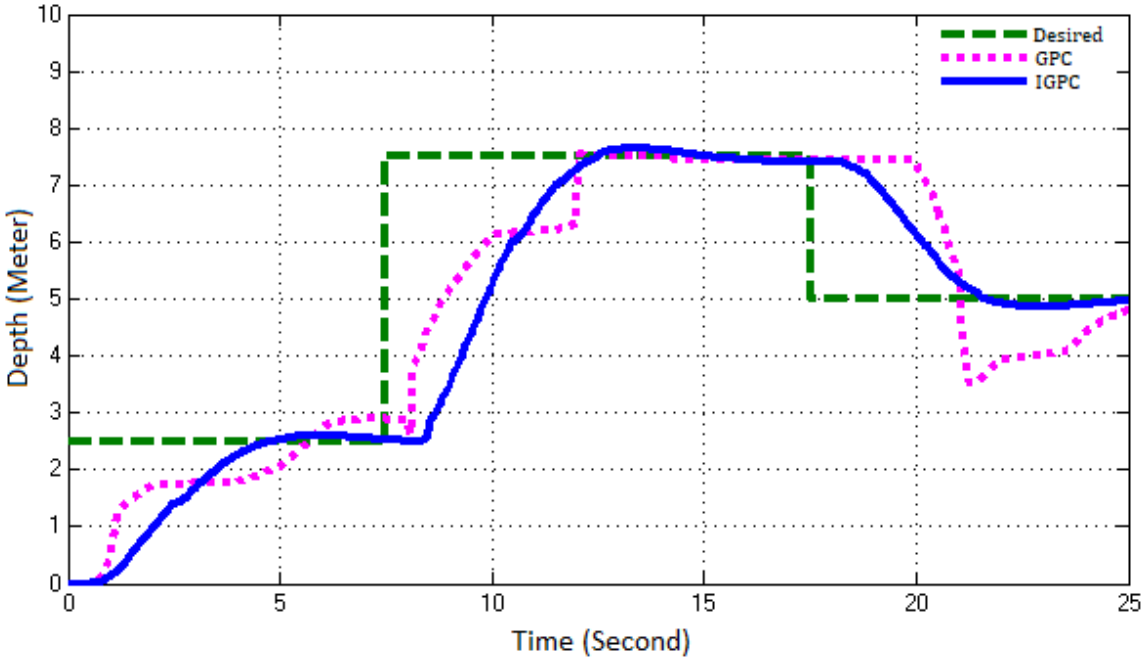

Figure 4.7: The response of networked ROV controlled by GPC and IGPC.

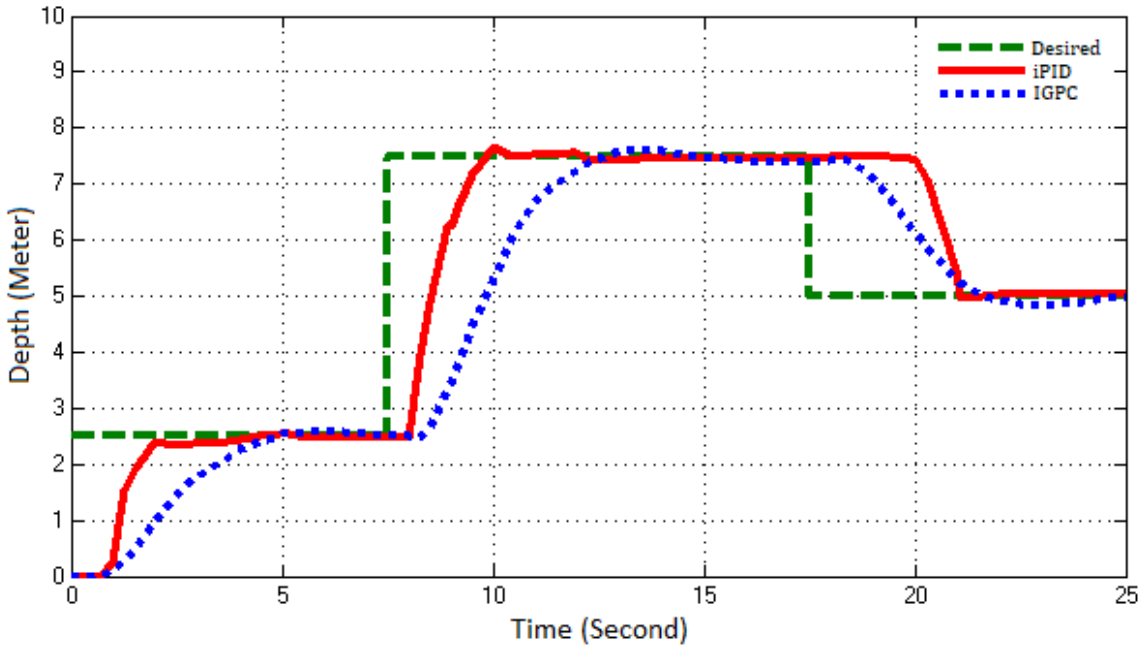

Figure 4.8: The response of networked ROV controlled by IGPC and iPID with time delay compensation.

Finally, the control of networked ROV with GPC and IGPC are tested with the same time delay of Fig.4.5. The benefit of using IGPC over GPC is clearly shown in Fig.4.7, while Fig.4.8, illustrates the advantage of using the intelligent structure along with GPC rather than PID even with time delay compensation.

Referring to Fig.4.7, the depth of ROV which controlled by GPC is swinging during the starting of the operation, and ROV lost the tracking of desired depth for two times, the first one at the time of 10 second and the second one after the time of 20 second, while with IGPC the response of ROV is keeping apart from these perturbation with same work condition.

From Fig.4.8, the response of ROV with GPC is steadier than iPID with compensation but with iPID is faster. Therefore, the selection between these two controllers is depend on the requirements of the application.

4.5 Conclusion

The countermeasure to avoid the effects of time-varying delay in networked model-free control is introduced in the first part of this chapter. The presented method is based on mutual benefit between Smith predictor and the basic model-free controller. The online updated variables within the basic model-free algorithm will be utilized to replace the

mathematical model of the plant in classic Smith predictor by the integration of the ultra-local model. This enhancement for Smith predictor along with the used automatically tuned model for dead time lead to completely model-free Smith predictor. It can be associated with the basic model-free controller in NCS, apart from the effects time-varying delay, and at the same time keeps the advantage of model-free in the control structure. The proposed method can be extended to be used with longer time delay by improving the estimation method of the non-linear term in the ultra-local model.

In the second part of this chapter, the Intelligent Generalized Predictive Controller (IGPC) is introduced as a suitable controller for NCS application. IGPC is designed to take advantage of the predictor and delay compensator of the GPC strategy along with the ability to cancel the nonlinear term which provided by structure of the intelligent controller.

The investigation reveals the proposed control schemes can successfully be used in the NCS with time-varying delay. In spite of the performance of the IGPC is better than that of the model-free controller with time delay compensation, but the availability of the mathematical model of the plant remains the primary factor of choice.

In this chapter, two control schemes for NCS able to compensate time delay are presented. In the next chapter, only the Intelligent Generalized Predictive Controller (IGPC) is retained, and a fault tolerant approach is proposed to detect attack and to compensate it.

CHAPTER 5

Attacks Detection Based on Control-Theoretic Approaches

Contents

5.1 Introduction

The NCS and SCADA (Supervisory Control and Data Acquisition) systems have a hierarchical structure with regulatory and supervisory control layers [115]. This chapter will be focusing on the attack detection at regulatory control layer.

Firstly, only IGPC will be applied to detect the attack, then IGPC along with a fault-tolerant ability which represented by a compensation of fault accommodation will be utilized. Beside the performance enhancement which is normally provided by the fault tolerant scheme, the provided compensation for the plant fault will help to recognize the effect of attacker signal on the plant apart from the influences of the fault, and as a result we should clear detection of the cyber-attack.

Secondly, the detection methods will be presented in a special attack scenario such as Stuxnet case [116] where a controller was reprogrammed and hijacked.

5.2 IGPC with an internal cyber-attack detector

The known loss of the response efficiency of the actuator was used in [117] to calculate the value of *F* which related to the iPID. However, the predicted and actual values *F* will be used in this chapter to calculate the loss of the response efficiency, which is caused by the cyber-attack on the plant side. The equation of the true control variable was given in [117], this equation can be used to express the attack on the plant as follows:

$$u_r(k) = (1 - \psi(k))\, u(k) \tag{5.1}$$

where, $\psi(k), 0 \leq \psi(k) \leq 1$, is the loss of the response efficiency of the plant due to control signal which is sent from the attacker. $u_r(k)$, is the true control variable.

In presence of the attack, the relation between predicted and estimated values of $F(k)$ can be written as:

$$[F(k)]_e = F\rho(k) - \psi(k)\, \alpha(k)\ \ u(k-1) \tag{5.2}$$

Hence,

$$\psi(k) = \left| \frac{F\rho(k) - [F(k)]_e}{\alpha(k)u(k-1)} \right| \tag{5.3}$$

where $[F(k)]_e$ and $F\rho(k)$ can be calculated from (4.4) and (4.56) respectively.

If $\psi(k) = 1$ implies that the plant is completely under attack. Fig.5.1 shows the block diagram of the proposed IGPC and the internal cyber-attack detector.

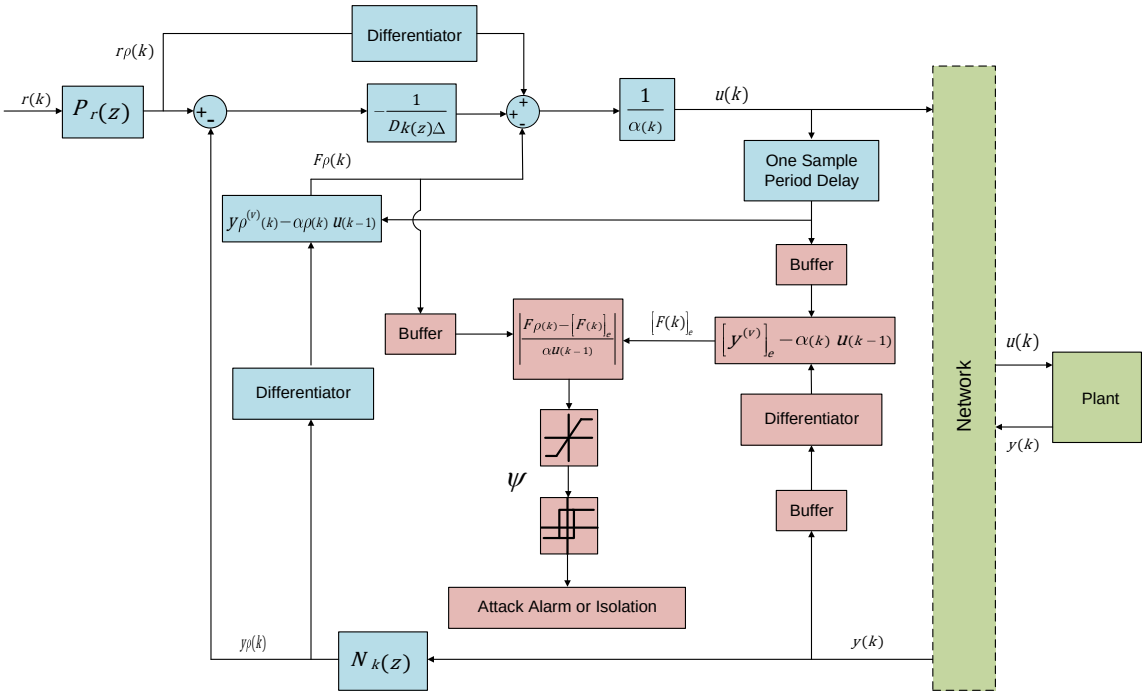

Figure 5.1: The block diagram of the IGPC and the attack detector.

In this section, the attack detection method completely depends on the comparison between the actual and predicted nonlinear terms of the plant. However, the predicted nonlinear term will be accurate for this comparison only if it is included with the effects of fault on the plant side, therefore this method needs more development regarding the presence of a fault in the plant. Moreover, the prediction horizon should be long enough to keep the predicted nonlinear term apart from the effects of the attack, here we should have a good trade-off between the sensitivity of detection and the control requirements.

5.3 Plant side attack detection with fault accommodation

In the previous section, the computation variables of IGPC were utilized to detect the cyber-attack on the plant side. However, this detection is still need for more development regarding the presence of a fault in the plant side. Therefore, the fault accommodation

control is the next design target in the proposed networked controller with attack detection mechanism.

5.3.1 Fault accommodation based on IGPC

The fault accommodation is relying on an on-line control law that maintains the main performances, even though some minor parts of the plant may slightly fail.

It is possible to adapt the self-tuning computation algorithm which presented in [117] to be applicable for IGPC as follows,

The plant fault can be stated by

$$u(k) = (1 - \phi(k))\, u_m(k) \tag{5.4}$$

where, ϕ, $0 < \phi < 1$, is the loss of efficiency of the plant which caused by the fault, and $u_m(k)$ is the normal control variable input to the healthy plant model which associated to the current actual plant output.

The two following cases are not considered:

ϕ=0, means that there is no fault.

$\phi=1$, indicates that the control does not act anymore.

Then (4.53) becomes

$$y\rho^{(v)}(k) = \tilde{F}\rho(k) + \alpha\rho(k)\ u(k-1) \tag{5.5}$$

where,

$$\tilde{F}\rho(k) = F\rho(k) + \alpha\rho(k)\ \phi(k)\ u(k-1) \tag{5.6}$$

and

$$\phi(k) = \left|1 - \frac{u(k)}{u_m(k)}\right| \tag{5.7}$$

$$u_m(k-1) = \frac{y\rho(k)}{P_m(z)} \tag{5.8}$$

where, *Pm(z)* is the healthy plant model.

5.3.2 Attack detection with IGPC

The equation of the true control variable can be used to express the attack on the plant as follows

$$\hat{u}(k) = u(k)(1 - \theta(k)) \tag{5.9}$$

where, $\theta(k)$, $0 \leq \theta(k) \leq 1$, is the loss of the response efficiency of the plant due to control signal which is sent from the attacker, and $\hat{u}(k)$, is the estimated value of the received control signal at the plant side.

To calculate the loss of the response efficiency due to the cyber-attack, we can rewrite (5.9) as follows,

$$\theta(k) = \left| 1 - \frac{\hat{u}(k-1)}{u(k-1)} \right| \tag{5.10}$$

$\hat{u}(k-1)$ will be calculated by (5.11) which is take into consideration the presence of fault.

$$\hat{u}(k-1) = \frac{y\rho^{(v)}(k) - \tilde{F}\rho(k-1)}{\alpha\rho(k-1)} \tag{5.11}$$

When $\theta(k)$ implies that the plant is completely under attack. Fig.5.2 shows the block diagram of the proposed IGPC with fault accommodation.

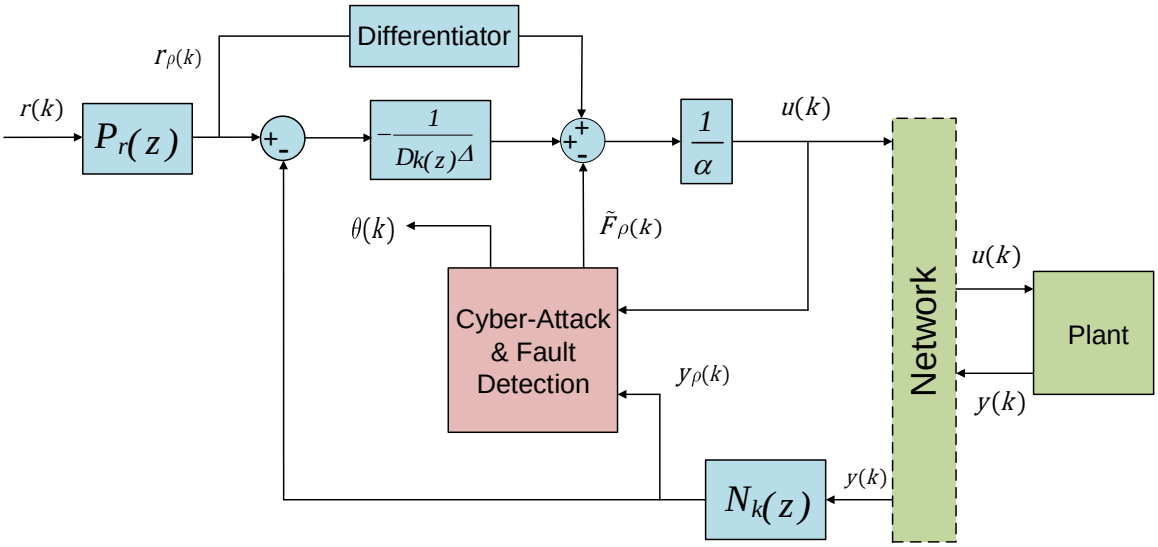

Figure 5.2: The block diagram Cyber-Attack detection with fault accommodation based on IGPC.

In the above attack detector, the predicted nonlinear term is included with the effects of the fault in the plant side. It's important to note that, only the detection of attacks at the plant side is taken into consideration during the two detection methods which are presented in sections 5.2 and 5.3.2. However, there is a possibility for other attacks (the attack on the controller side as an example). Therefore, the overall attack detection will be considered in the next section before the detection methods for controller attack.

5.3.3 Attack detection using the framework of behavioral system

Beside the fault accommodation which based on IGPC, the mathematical framework of behavioral system will be applied for detection the cyber-attack on the received control signal at the plant side.

Firstly, we recall the basic closed loop control in the behavioral context. Interested readers might refer to [118] for a complete presentation. In the behavioral setting, the control problem is viewed as an interconnection of two dynamical subsystems, these subsystems are the plant and the controller.

Related to the plant, a dynamical subsystem Σp is represented by a triple $\Sigma p = (\mathcal{T}, \mathcal{W}, \mathcal{P})$ where $\mathcal{T} \subseteq \mathbb{R}$ is named the time axis, $\mathcal{W} \subseteq \mathbb{R}^{w}$ is named the signal space and $\mathcal{P} \subseteq \mathcal{W}^{\mathcal{T}}$ is named the full behavior of the plant. Similarly, for the controller, a dynamical subsystem Σc is represented by a triple $\Sigma c = (\mathcal{T}, \mathcal{W}, \mathcal{C})$ where $\mathcal{C}$ is the controller behavior.

The interconnection of Σp and Σc shared by control variable c and it represented as $\Sigma p \wedge_{c} \Sigma c$ where,

$$\Sigma p \wedge_{c} \Sigma c := (\mathcal{T}, \mathcal{W}, \mathcal{P} \wedge_{c} \mathcal{C}) \tag{5.12}$$

Consequently, the behavior of $\Sigma p \wedge_{c} \Sigma c$ involves basically of the trajectories $w: \mathcal{T} \rightarrow \mathcal{W}$ which are compatible with the rules of Σp and Σc.

The interconnection between $\mathcal{P}$ and $\mathcal{C}$ outcomes in a manifest controlled behavior $\mathcal{B}$ which can expressed as $\mathcal{B} = (\mathcal{P} \wedge_{c} \mathcal{C})_{w,}$ where w is to be controlled variable.

Thus, $\mathcal{B}$ is always implemented by IGPC and the fault accommodation mechanism, and

$$\mathcal{B}_h \subset \mathcal{B} \subset \mathcal{B}_r \tag{5.13}$$

where, $\mathcal{B}_r$ is *restricted behavior*, and $\mathcal{B}_h$ is the *hidden behavior* which is defined as the behavior containing of plant trajectories with the interconnection variables place equal to zero.

For IGPC with fault accommodation and without any attack, the full behavior of the plant and the controller can be written as in (5.14) and (5.15) respectively,

$$\mathcal{P} := \begin{bmatrix} 1 & -1 & -1 & 0 & 0 \\ 0 & Pm_y & 0 & 0 & Pm_u \end{bmatrix} \begin{bmatrix} r\rho(k) \\ y\rho(k) \\ e\rho(k) \\ \tilde{F}\rho(k) \\ u(k) \end{bmatrix} = 0 \tag{5.14}$$

$$\mathcal{C} := \begin{bmatrix} \frac{D}{\alpha\rho(k)} & 0 & \frac{-C\rho(k)}{\alpha\rho(k)} & \frac{-1}{\alpha\rho(k)} & -1 \end{bmatrix} \begin{bmatrix} r\rho(k) \\ y\rho(k) \\ e\rho(k) \\ \tilde{F}\rho(k) \\ u(k) \end{bmatrix} = 0 \tag{5.15}$$

where, Pm_y and Pm_u are the exist co-prime polynomials such that $Pm = Pm_y^{-1} Pm_u$ and D is a differential operator.

Hence, the controlled behavior $\mathcal{B}$ can be expressed as following

$$\mathcal{B} = (\mathcal{P} \wedge_e \mathcal{C})_w := \begin{bmatrix} 1 & -1 & -1 & 0 & 0 \\ 0 & Pm_y & 0 & 0 & Pm_u \\ \frac{D}{\alpha\rho(k)} & 0 & \frac{-C\rho(k)}{\alpha\rho(k)} & \frac{-1}{\alpha\rho(k)} & -1 \end{bmatrix} \begin{bmatrix} r\rho(k) \\ y\rho(k) \\ e\rho(k) \\ \tilde{F}\rho(k) \\ u(k) \end{bmatrix} = 0 \tag{5.16}$$

IGPC with fault accommodation mechanism, is designed to achieve a desired controlled behavior $\mathcal{B}_d$ such that $\mathcal{B} \subseteq \mathcal{B}_d$,. In presence of cyber-attack, $\mathcal{B} \nsubseteq \mathcal{B}_d$, therefore the cyber-attack can be detected by measuring the controlled behavior using (5.16) and by taking the magnitude of it $\|\mathcal{B}\|$.

5.4 Detection of the controller hijacking attack

From the British Columbia Institute of Technology Industrial Security Incident Database [119], we can note other attacks on process control and industrial networked systems (e.g. Stuxnet case).

In addition to previous sections, this chapter deals with the effects of special attack scenarios such as Stuxnet case [116] where a controller was reprogrammed and hijacked.

The Stuxnet case needs to the implementation of special attack detectors along with backup controller. Until now, a typical approach to implement such attack detectors is based on system models [120-121] and it is inspired by fault diagnosis detectors [122-123]. The Stuxnet attack detector, which not based on controller model was proposed in [124]. However, the restriction of [124] was represented by assuming the value of set point (operator's desired value) is always equal to zero.

This section specially focuses on the controller hijacking attack with the injection of a destructive control signal into the networked control loop. The architecture of the attack-tolerant system developed will be detailed in the next sections.

5.4.1 A technique to detect of the controller hijacking attack

In this section, the ultra-local model approach which introduced in [99] will be used instead of the mathematical model of the controller. Assume we have a controller (see Fig.5.3) and that one wants to monitor the controller behavior by mean of the controller input and output. The controller with a single input variable ***e*** and a single output variable ***u*** is selected. The complex mathematical model of the controller is replaced by (5.17):

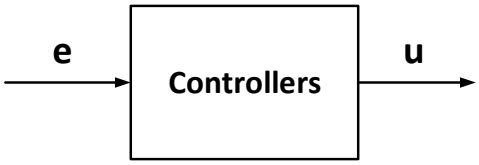

Figure 5.3: SISO Controller.

$$u^{(v)} = \mathcal{S} + \beta e \tag{5.17}$$

where,

- $\beta \in \mathbb{R}$ is a non-physical constant parameter. It is chosen by the designer such that αe and $u^{(v)}$ are of the same magnitude. It should be therefore clear that its numerical value, which is obtained by trials and errors during attack free condition, is not a priori precisely defined.
- $\mathcal{S}$ represents all the unusual input-output behaviour of the controller;
- The order ***v*** is also a design parameter of the numerical model of (5.17) that can be arbitrarily chosen by the designer.
- $e = y_d - y$. y_d, is the operator's desired set point and y is the plant response.

However, (5.17) is not a black-box identified model of the controller of Fig.5.3. In attack detector, (5.17) will be updated at each sampling time from the knowledge of the input-output behavior of the controller in order to estimate the unusual quantity $\mathcal{S}$ and use it in the proposed detection method. $\hat{\mathcal{S}}$ is the estimation of $\mathcal{S}$ at sample *k* and it is given by:

$$\hat{\mathcal{S}}(\mathrm{k}) = \langle\langle u^{(v)} \rangle\rangle_{T_s,n_s}^{d}(k) - \beta e(k) \tag{5.18}$$

where, $\langle\langle u^{(v)}\rangle\rangle_{T_s,n_s}^{d}(k)$ is discrete-time estimation of the derivative of order *v* of the output of the controller, and $e(k)$ the error input applied to the controller at sample time *k.Ts*, *k* = 0, 1,…

Based on the numerical knowledge of $\hat{\mathcal{S}}(\mathrm{k})$ at each sample *k*, the unusual controller behavior can be detected and the attack can be evaluated according to the specific threshold level of $\hat{\mathcal{S}}(\mathrm{k})$.

5.4.2 Derivative estimation of the controller output

The derivative estimation method, which described in [100] will be used to estimate the derivatives of the controller output. Let u_m be a measured value of the controller output u, and u_m is the image of u and we consider that u is distorted by some of added noise ζ, therefore we have: $u_m = u + \zeta$.

The objective is to estimate the derivatives of the controller output u, up to a finite order of derivation, from its measurement u_m observed on a given time interval.

The Taylor expansion of the controller output around 0 is given by

$$u(\tau) = \sum_{n=0}^{\infty} u^{(n)}(0)\frac{\tau^n}{n!} \tag{5.19}$$

By the polynomial we can approximate $u(t)$ for the interval $[0, T], T > 0$.

$$u_N(\tau) = \sum_{n=0}^{N} u^{(n)}(0)\frac{\tau^n}{n!} \tag{5.20}$$

For *N* degree Φ_N is the operational analogue of u_N and can be written as

$$\Phi_N(s) = \frac{u(0)}{s} + \frac{\dot{u}(0)}{s^2} + \cdots + \frac{u^{(N)}(0)}{s^{N+1}} \tag{5.21}$$

By applying a convenient operator to $\Phi_N(s)$, we can separate each coefficient $u^{(i)}(0)$ appearing in previous expression. Thus,

$$\forall i = 0, \ldots, N$$
$$\frac{u^{(i)}(0)}{s^{2N+1}} = \frac{(-1)^i}{N!\,(N-i)!}\cdot\frac{1}{s^{N+1}}\cdot\frac{d^i}{ds^i}\cdot\frac{1}{s} \cdot \frac{d^{N-i}}{ds^{N-i}}(s^{N+1}\Phi_N(s)) \tag{5.22}$$

The expression of $u^{(i)}(0)$ in the time domain can be written as

$$u^{(i)}(0) = \int_0^T \mathfrak{P}(\delta; T)\, u_N(\delta)\, d\delta \tag{5.23}$$

where $\mathfrak{P}(\delta; T)$ is polynomial in δ and *T*. Notice that (5.23) gives the calculation of $u^{(i)}(0)$ from an integral on the time interval [0; *T]* for a given small *T > 0.*

As $\frac{d^i u(t-\delta)}{d\delta^i}\Big|_{\delta=0} = (-1)^i u^{(i)}(t)$ it is possible to express $u^{(i)}(t)$ as an integral, which includes values of u_N on the time interval *[t – T, t]*:

$$u^{(i)}(t) = (-1)^i \int_0^T \mathfrak{P}(\delta; T)\, u_N(t-\delta)\, d\delta \tag{5.24}$$

By using the noisy signal u_m, a simple estimator of the derivative $u^{(i)}(t)$ can be expressed as follows:

$$\langle\langle u^{(i)}\rangle\rangle_T^c\,(t) = (-1)^i \int_0^T \mathfrak{P}(\delta; T)\, u_m(t-\delta)\, d\delta \tag{5.25}$$

(5.25) is realized from (5.24) by changing u_N by u_m. Noting that the integral operation acting the rule of the low-pass filter which reducing the noise that distorts u_m. The choice of *T* results in a trade-off: small value of *T* leads to the effect of the noise; the large value of *T* leads to better integrals low pass filtering but there is truncation error.

In practice, the integral expressed in (5.25) is obtained by a numerical integration method; therefore, the estimator $\langle\langle u^{(i)}\rangle\rangle_T^c\,(t)$ will be performed at each sample of *k.*

Let Ts is the sampling period, then the discretization of any continuous time function *f* will be denoted by *f* [k] i.e.

$$f[k] = f(k, Ts), k \in Z \tag{5.26}$$

with these notations, the discrete-time approximation of the derivative estimation of the controller output is simply a discrete sum that can be written as:

$$\langle\langle u^{(v)}\rangle\rangle_{T_s,n_s}^d\,[k] = \sum_{j=0}^{n_s} \mathfrak{W}(j)\, \mathfrak{P}(jT_s; n_sT_s) u_m[k-j] \tag{5.27}$$

where, n_s the number of samples used in the time window $T = n_sT_s, v = i$, and the $\mathfrak{W}(j)$ is the weight related to the used numerical integration method.

5.5 Detection of the controller stealthy hijacking attack

The stealthy hijacking attack will be considered in this section rather than the rough attack. Rewrite (5.1) for the attack on the controller (e.g. controller hijacking attack) or,

$$e_r(k) = \big(1 - \varsigma(k)\big)e_n(k) \tag{5.28}$$

where, $e_r(k)$ is the true error variable which is measured at the controller input during all cases including the presence of an attack on controller side, $e_n(k)$ is the nominal error value and $\varsigma(k)$, $0 \leq \varsigma(k) \leq 1$, is the controller performance index.

Rewrite (5.28),

$$\varsigma(k) = 1 - \frac{e_r(k)}{e_n(k)} \tag{5.29}$$

In this section, the controller performance index will be used to detect the controller hijacking attack. Now, the objective is to estimate the nominal error as well as the true error. Recall the ultra-local model of the controller which is given in (5.17), for the attack free controller, the values of $\langle\langle \dot{u} \rangle\rangle^{d}_{Ts,ns}(k)$ and $\beta\left|\langle\langle e \rangle\rangle^{d}_{Ts,ns}(k)\right|$ are approximately equal in magnitude or,

$$\left|\langle\langle e_n \rangle\rangle^{d}_{Ts,ns}(k)\right| = \left|\frac{\langle\langle \dot{u} \rangle\rangle^{d}_{Ts,ns}(k)}{\beta}\right| \tag{5.30}$$

The method of section 5.4.2 can be used to calculate the value of $\langle\langle \dot{u} \rangle\rangle^{d}_{Ts,ns}(k)$.

To eliminate the effects of network which are mentioned before, the true error function values will be integrated as a tabular data. These values are taken at certain discrete points during the same time window of the derivative output estimation.

First is to fit a curve through the true error data, and then integrate the resulting curve. It is possible to use any integration methods, but the most common approach is to use a piecewise polynomial such as a spline as follows:

- Suppose they are given as set of true error samples point $(t, e_r(t))$.
- Fit a spline $Er(t)$, through this samples, keep in mind that $Er(t)$ is a piecewise polynomial on several intervals in particular.

$$Er(t) = \begin{cases} erp_1(t) & t_0 \le t \le t_1 \\ erp_2(t) & t_1 \le t \le t_2 \\ \vdots & \\ erp_{ns}(t) & t_{ns-1} \le t \le t_{ns} \end{cases} \tag{5.31}$$

Then,

$$\langle\langle e_r \rangle\rangle_T^c(t) = \int_{t-nsT}^{t} e(t)\, dt \approx \int_{t-nsT}^{t} Er(t)\, dt = \sum_{j=0}^{ns} \int_{t_{j-1}}^{t_j} erp_j(t)\, dt \tag{5.32}$$

It is possible to use a cubic spline interpolant. In this case, each $erp_j(t)$ is a cubic polynomial, which can be written as

$$erp_j(t) = a_{j,1}(t - t_{j-1})^3 + a_{j,2}(t - t_{j-1})^2 + a_{j,3}\big(t - t_{j-1}\big) + a_{j,4} \tag{5.33}$$

Solving for $a_{j,1}, a_{j,2}, a_{j,3}$ and $a_{j,4}$ using Gauss elimination [125], then we can easily integrate $erp_j(t)$

$$\int_{t_{j-1}}^{t_j} erp_j(t)\, dt = \frac{a_{j,1}}{4}(t_j - t_{j-1})^4 + \frac{a_{j,2}}{3}(t_j - t_{j-1})^3 + \frac{a_{j,3}}{2}(t_j - t_{j-1})^2 + a_{j,4}(t_j - t_{j-1}) \tag{5.34}$$

Let, T = $t_j - t_{j-1}$ then,

$$\int_{t_{j-1}}^{t_j} erp_j(t)\, dt = \frac{a_{j,1}}{4}T^4 + \frac{a_{j,2}}{3}T^3 + \frac{a_{j,3}}{2}T^2 + a_{j,4}T \tag{5.35}$$

Thus, the estimated true error is given by

$$\langle\langle e_r \rangle\rangle_{T_s,n_s}^d(k) = \sum_{j=0}^{ns} \left(\frac{a_{j,1}}{4}T_s^4 + \frac{a_{j,2}}{3}T_s^3 + \frac{a_{j,3}}{2}T_s^2 + a_{j,4}T_s\right) \tag{5.36}$$

Note that a direct implantation of this summation would require 7*ns* multiplications, 3*ns* additions, and 3*ns* exponentiations. An algebraic rearrangement results in a nested approach of this computation [125],

$$\langle\langle e_r \rangle\rangle_{T_s,n_s}^d(k) = \sum_{j=0}^{ns} T_s \left(T_s \left(T_s \left(\frac{a_{j,1}}{4}T_s + \frac{a_{j,2}}{3}\right) + \frac{a_{j,3}}{3}\right) a_{j,4}\right) \tag{5.37}$$

(5.37) requires only 7*ns* multiplications, 3*ns* additions, and no exponentiations.

Rewrite (5.29) in term of estimated values

$$\varsigma(k) = 1 - \left| \frac{\beta \sum_{j=0}^{ns} T_s \left(T_s \left(T_s \left(\frac{a_{j,1}}{4} T_s + \frac{a_{j,2}}{3} \right) + \frac{a_{j,3}}{3} \right) a_{j,4} \right)}{\sum_{j=0}^{n_s} \mathfrak{W}(j)\, \mathfrak{P}(jT_s; n_s T_s) u_m(k-j)} \right| \tag{5.38}$$

If $\varsigma(k) = 0$ means that there is no controller hijacking attack.

$0 < \varsigma(k) < 1$ means that there is stealthy controller hijacking attack.

If $\varsigma(k) = 1$ implies that the controller is completely hijacked.

5.6 Simulation Results

In this section, the test results will be classified into two parts, the first part is related to the detection of the attack on the plant side while, the second part is related to the attack on the controller side.

5.6.1 Detection of the attack on the plant side

The proposed IGPC with cyber-attack detector is simulated as per the scheme mentioned in sections 5.2 and 5.3 with networked servo-pneumatic positioning system.

The main complications in controlling the servo-pneumatic position control are resulting from the undesirable effect of friction and air compressibility, these make the pneumatic-servo drive intrinsically nonlinear as well as the variations of its parameters with time. The network control of this system will increase its nonlinearity and uncertainty and, consequently, the difficulty of detecting cyber-attacks. In this section, a model of pneumatic drive will be used as a plant. Fig.5.4 illustrates the diagram of this plant.

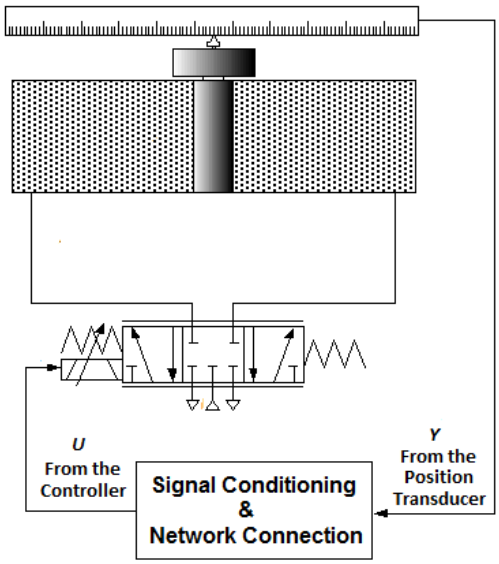

Figure 5.4: Servo-pneumatic positioning system.

This plant model can be expressed as follows

$$G_p(s) = \frac{y(s)}{u(s)} = \frac{K_v}{s(\omega_0^{-2}s^2 + 2\xi\omega_0^{-1}s + 1} \tag{5.39}$$

where, $u(s)$ is the input, $y(s)$ is the position of the load, K_v is the velocity gain, ξ is its damping ratio, and ω_0 is the natural frequency of the loaded cylinder drive. Of course, this model is based on linearization of the nonlinear dynamics around an equilibrium state, however it is still involving with the essential characteristics.

The natural frequency and damping ratio of the cylinder can be expressed as in (5.40) and (5.41), the parameters of (5.40) and (5.41) are given in Table 1 [126].

$$\omega_0 = \sqrt{\frac{4.A^2.\kappa.p}{M.V_c}} \tag{5.40}$$

$$\xi = \frac{1}{2}.k_f.\sqrt{\frac{V_c}{4.p.A^2.\kappa.M}} \tag{5.41}$$

Table 5.1. Numerical value of the servo-pneumatic positioning system

Parameter	Nomenclature	Value
Cross-sectional area	A	$1.767.10^{-4}\ m^2$
Volume of the cylinder	V_c	$8.835.10^{-5}\ m^3$
Supply pressure	p	$5.10^5\ P_a$
Initial load	M	$0.91\ Kg$
Friction coefficient	k_f	$65\ N_s/m$
Spec. heat ratio of air	κ	1.4
Natural frequency	ω_0	$32.97\ rad/s$
Damping ratio	ξ	1.1

Initially, the system is tested without any attack; the test result is illustrated in Fig.5.5, during the steady-state parts of the response, the value of θ is varies between 0 and 0.1 while it will be less than 0.55 during the transit parts of the response, this variation is resulted mainly from uncertainty which is produced from the network behavior and the accuracy of the plant modeling.

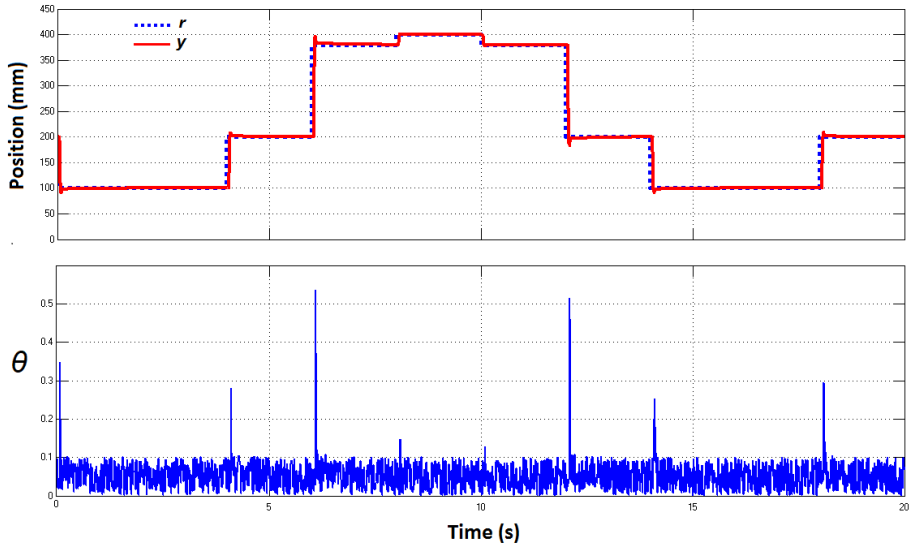

Figure 5.5: The response of servo-pneumatic positioning system and the value of θ without any attack.

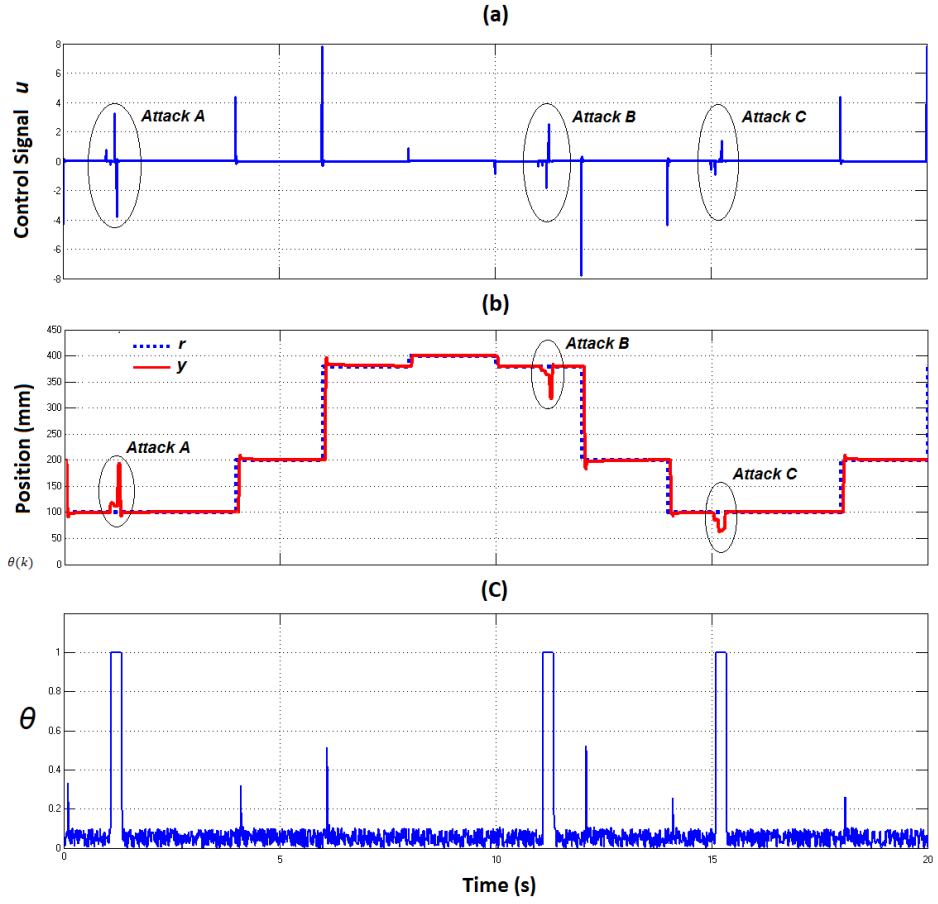

Figure 5.6: The response of servo-pneumatic positioning system and the value of θ in presence of the attacks (A, B, and C)

After that, the simulation is carried out by injecting of external attack control signals to the remote plant. The attack signals are represented by random pulses separated by random time intervals. Fig.5.6 shows the response of the system in presence of three attacks (A, B, and C). The value of θ is quickly increased to one, when there is any attack on the plant side even within a short time.

From above results, the output of the detector can easily provide a logical signal which can be used to alarm or to isolate as a reaction against the cyber-physical attack at the plant side. Also, because this method is based on the online measurements of received control at the plant side, the length of prediction horizon has no effects on the sensitivity of the detector.

However, for the used plant the variation in control signal caused little increasing in detector outputs (during transient time) which does not affect the sensitivity of the detection method. But, for a plant with high level of variation it will be necessary to optimize the ultra-local model.

5.6.2 Behavioral system approach to detect attack on plant side

The proposed mathematical framework of behavioral system in sections 5.3.3 is simulated along with IGPC to detect the cyber-attack on remotely operated underwater vehicle ROV. The model of ROV is given in chapter four. Fig.5.7 illustrates the obtained result.

Referring to Fig.5.7. a, the attacks have been made by adding three different pulses to the control signal which input to ROV.

The magnitude of the desired controlled behavior is selected to be 0.3, this has been made by measuring the maximum value of the controlled behavior during the normal condition i.e. without any attack. The values of $\|B\|$, are exceed the desired level during the attack. These changes in controlled behavior during the attack are used to generate the detection signal as shown Fig.5.7. c.

From the results, we can see that the proposed method detects quickly the presence of attack pulses, but the detection signals are only available during the attack pulses time and disappear after that, even if the impact of the attack is continuing.

The detection method of this section, gives us full detection for the attack at the plant side (detect the attack on the actuating signal or on the sensor) without distinguish between

them. To detect the impact of the attack, it will be necessary to extended the analysis method to take into consideration both, the restricted behavior and the hidden behavior.

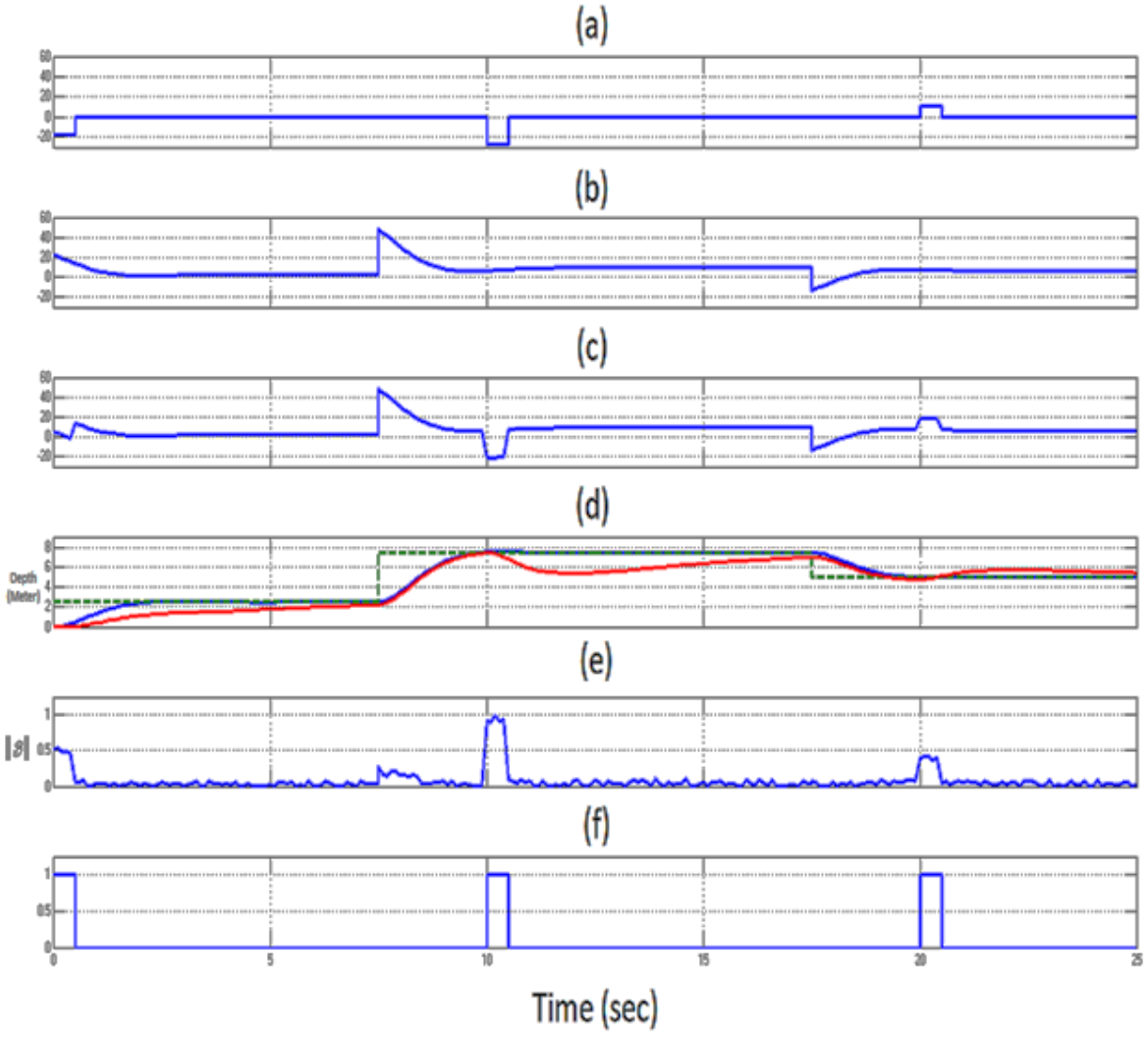

Fig.5.7. The test result of Behavioral system approach, (a) The three different attack pulses (b) Control signal without any attack, (c) Control signal wich is effected by the attacker pulses, (d) Is the response of ROV: doted green line is the desired value, the blue line is normal response without any attack, and red line is the response effected by the attack red (e) The magnitude of controlled behavior, (f) The detection signal

5.6.3 Detection of the controller hijacking attack

The position control of networked DC servomotor is selected to verify the performance of the first technique to detect the controller hijacking attack.The parameters and values chosen for motor modeling are presented in chapter three.

The simulation is carried out by randomly reprogramming the controller. In this case the original settings of the controller will be lost and its behavior will be unknown (i.e. hijacked by the attacker). Fig.5.8 shows the plant response Y when the controller is suddenly hijacked without detection.

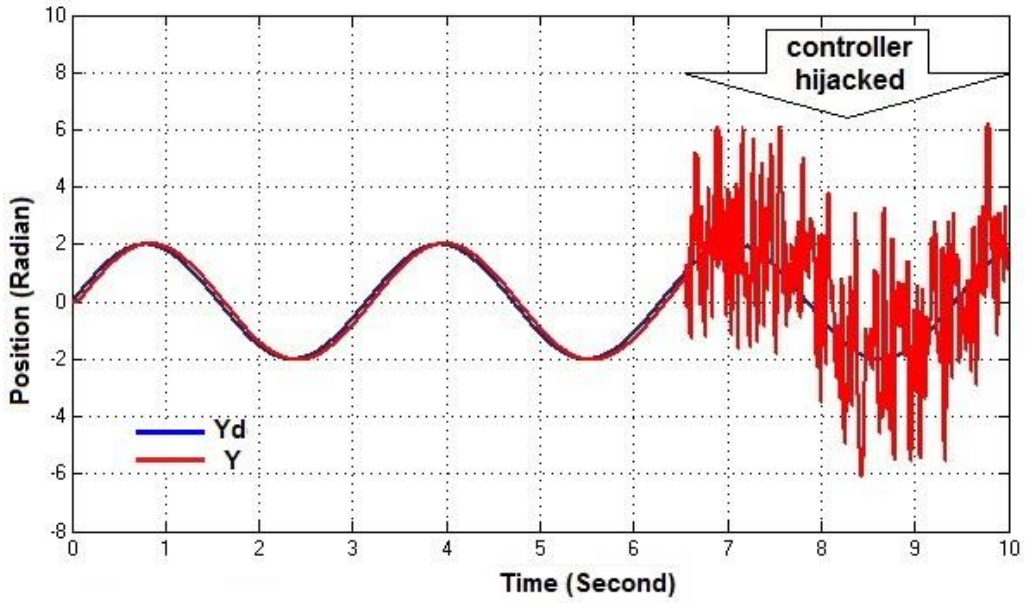

Figure 5.8: The response of servomotor when the controller is suddenly hijacked without detection.

The detection method of the controller hijacking attack which described in section 5.4 is applied to this case of attack. The test results are shown in Fig.5.9, and Fig.5.10. The controller attack is successfully detected. From Fig.5.9-b and Fig.5-10-b, one can note the effect of controller hijacking attack on the value of $\hat{S}$. The abnormal change of $\hat{S}$ will be used to generate controller recovering signal as illustrate in Fig.5.8.c.

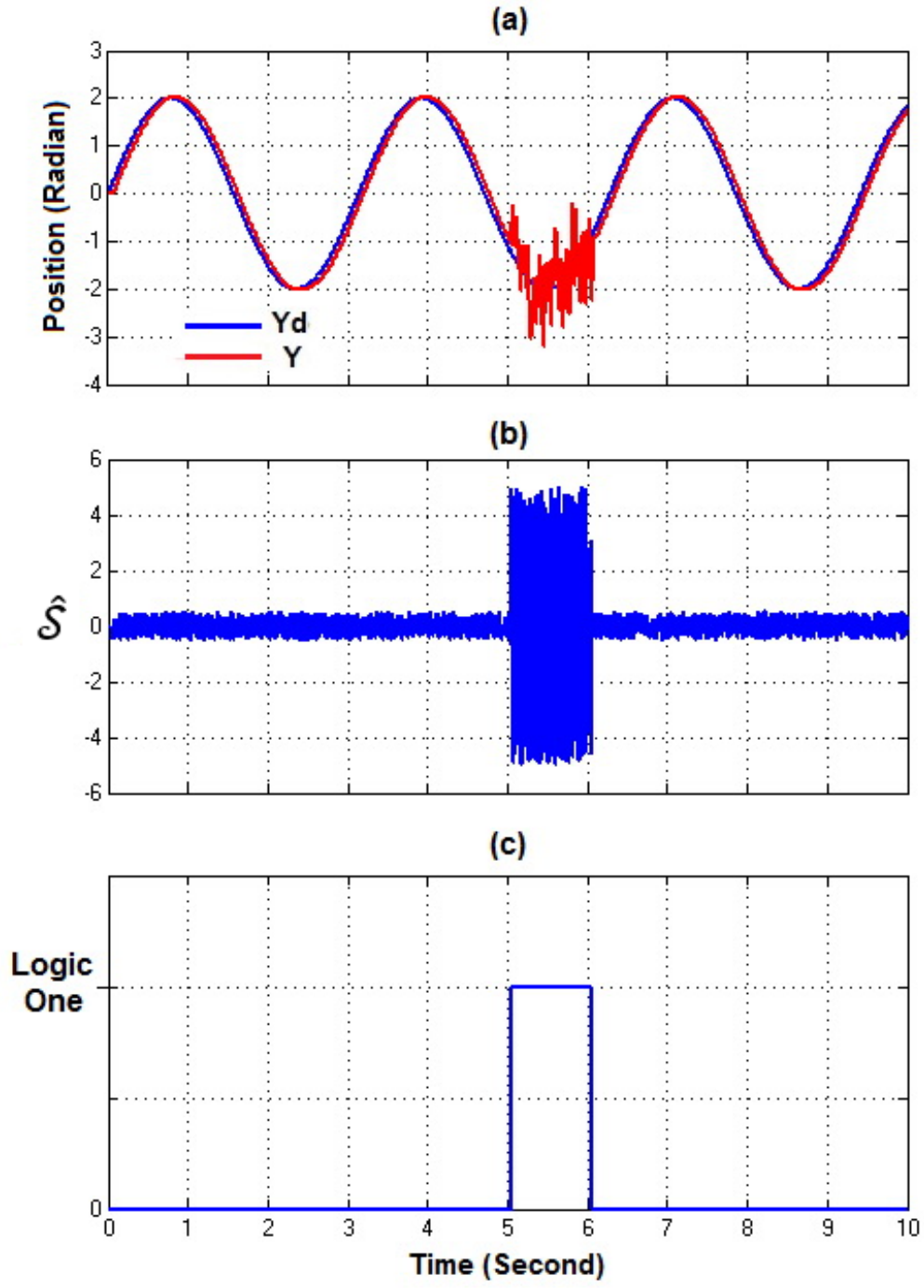

Figure 5.9: (a) The response of the servomotor when the attack is (or has been) detected and the controller is recovered. (b) The effect of the controller hijacking attack on the value of $\hat{\mathcal{S}}$ (c) controller recovering signal.

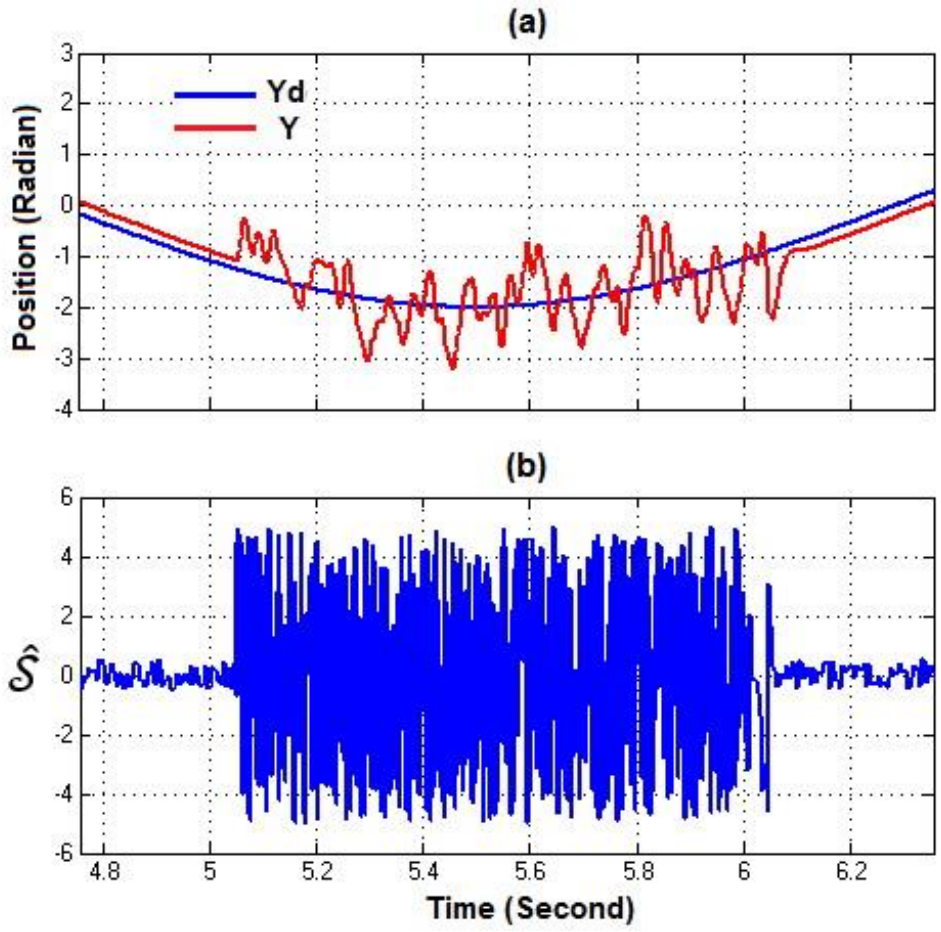

Figure 5.10: (a) The response of the plant when the attack is detected and the controller is recovered (with zoom) , (b) The effect of the controller hijacking attack on the value of $\hat{\mathcal{S}}$ (with zoom).

From the above figures, the proposed method provides an acceptable detection for the controller hijacking attack; however, the main limitation is represented by the sensitivity to the attack. This resulted mainly because of its dependence is only on the nonlinear term of the controller with one method for estimation.

5.6.4 Detection of stealthy hijacking attack

The detection stealthy hijacking attack at the controller side will be considered in this section rather than the rough attack of section 5.4.

For this case, the proposed method is developed as per the scheme mentioned in section 5.5 with a simulation of the local downstream controller at the Partiteur cross-regulator in

the Gignac canal which located 40 km north-west of Montpellier, in the south of France. This example is selected because irrigation main canal pools whose dynamics strongly change with the discharge regime variations and as a result it's difficult to detect the stealthy hijacking attack at the controller side.

The Integrator Delay (ID) model is an approximate representation of the dynamics of canal pool for low frequencies, this model and its parameters were given in [127] as follows,

$$y = \frac{1}{A_d\, s}\left(e^{-\tau_d s}\, u - p\right) \tag{5.42}$$

where,

- y is the downstream water elevation.
- τ_d is the delay of the canal pool.
- p the downstream perturbation.

When the control input u is a discharge, A_d represents the backwater area, and when u is the upstream gate opening, A_d is the inverse integrator gain.

For the Partiteur Left Bank cross regulator, the *Ad* = 25.2113 and *τd* =30 seconds. The PI controller will be used in this test as the downstream controller with *Kp=0.60* and *Ti=96* seconds [127].

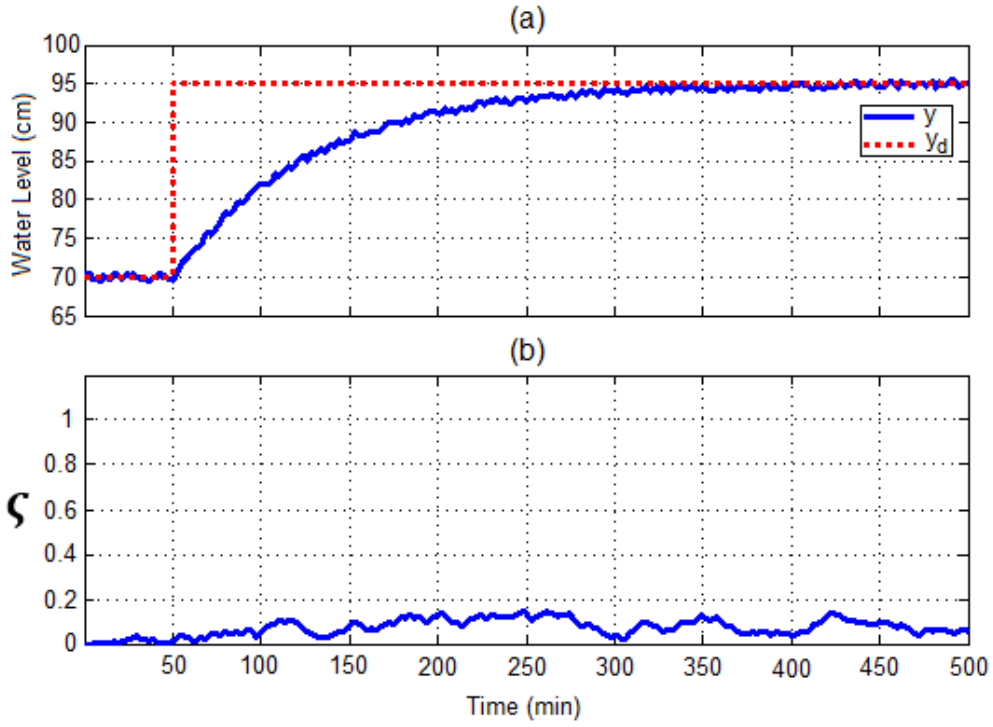

Figure 5.11: The downstream water elevation and ς without any attack.

Initially, the system was tested without any attack; the test result is illustrated in Fig.5.11. After that, the simulation is carried out with stealthy reprogramming of PI controller by the randomly change of *kp* and *Ti* of *PI* but within a very short time.

In this case, the original settings of the controller will be lost and its behavior will be unknown. In order to evaluate the proposed detector, the controller has been reprogramed for three times (attack A, attack B, and attack C). The test results are shown in Fig.5.12. From Fig.5.12-b, one can note the effect of controller hijacking attacks on the value of ς

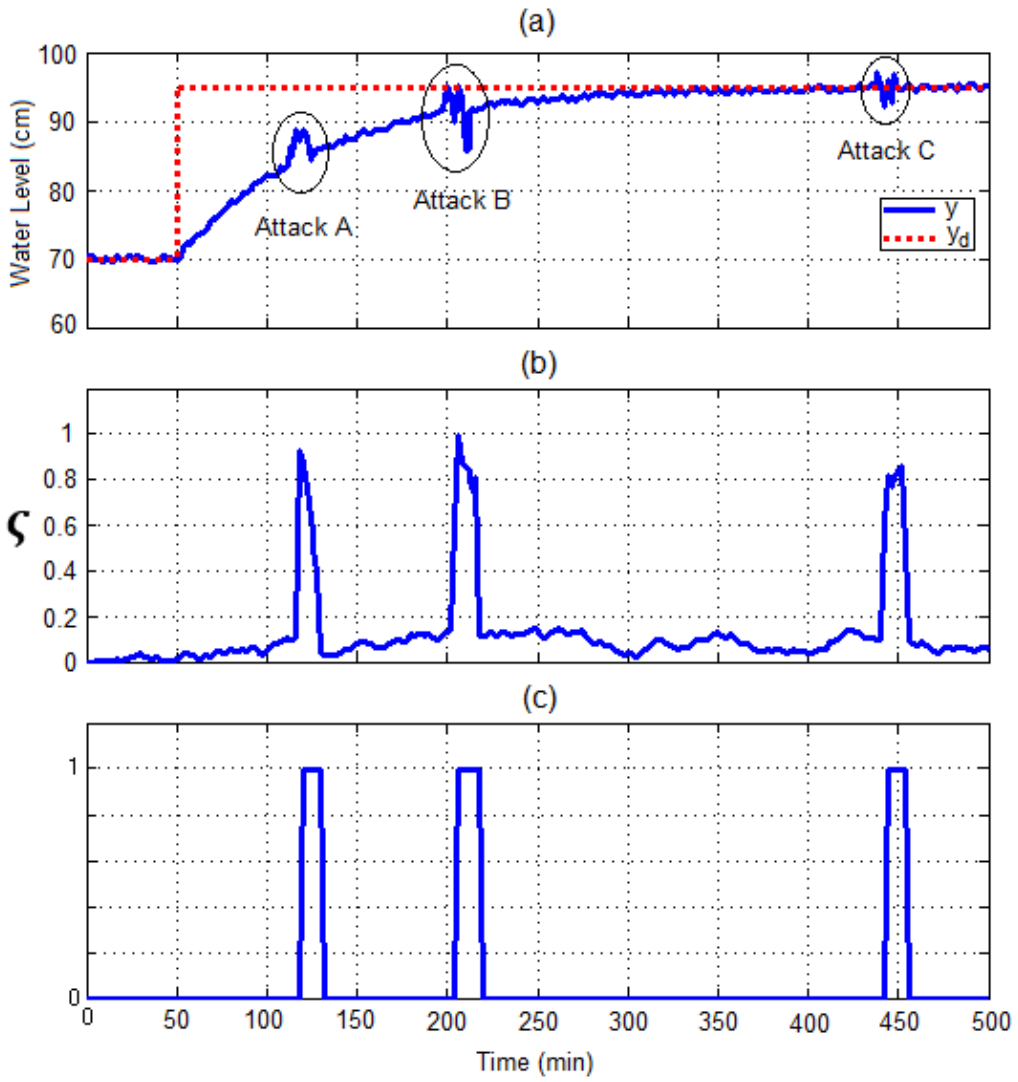

Figure 5.12: (a) The downstream water elevation when the attack is (or has been) detected and the controller is recovered. (b) The effect of the controller hijacking attack on the value of ς (c) controller recovering signal

This detection keeps an acceptable performance even with the short time duration of attacks and the downstream perturbation. The abnormal change of ς will be used to generate controller recovering signal as illustrate in Fig.5.12- c.

From the results, the suggested method in this section can provide a good sensitivity to the controller stealthy hijacking attack in comparison with the proposed method in the previous section. This improvement mainly resulted from the used additional estimation method for the true error.

However, the proposed detectors for the controller hijacking attack take into consideration only Single-Input, Single-Output (SISO) controller because it is commonly used in industrial control as well as used in Programmable Logic Controller (PLC) systems which were attacked by the Stuxnet.

5.7 Conclusion

In this chapter, three methods to detect the attacks on the plant side and two methods to detect the attack on the controller are presented.

Related to the plant side, the predicted nonlinear term of the plant, which achieved from IGPC, and the actually estimated nonlinear term of the plant, which achieved from the knowledge of the plant input and the direct measurement of its output, are used to generate an indication about the loss of the response efficiency of the plant due to the attacker's bad signal.

The comparison between the sent control signal and the estimated arrived value of the control signal at the plant side is also applied to detect the presence of any attack at the plant side.

In the third method, the mathematical framework of behavioral system is applied to detect the attack on the plant.

The above three methods demonstrated that they can be considerably used for detection of the attack on the plant side. The suitability and efficiency of the three methods is depend on the model information and other security supports, for example the precision of plant model and the high-speed dynamics are necessary for the first method while there is no

such constrains for second and third methods. Furthermore, the availability of controller attack detector is very necessary to ensure the type of attack in the third method.

Finally, the two detectors for controller hijacking attack are introduced. The facility of the proposed two methods is that no a priori controller mathematical model is required. The investigation reveals that the stealthy hijacking attack detector scheme can be used successfully in presence of the Stuxnet case.

CHAPTER 6

Conclusions and Future Work

In this thesis, the cyber-physical attacks in NCS have been discussed. The detection methods for these attacks by using some of the control variables are introduced as an alternative or support methods for IT methods which are frequently used in this domain.

Due to the effects of time-delay on the stability of NCS and the resulted effects on the control variables which can be used by the attack detector, the first target in this thesis was to implement a controller that is able to deal with this problem. Firstly, the mutual benefit between the Smith predictor and the basic model-free controller is proposed as a compensation for the variable time- delay. The online updated variables within the basic model-free controller are utilized to replace the mathematical model of the plant in standard Smith predictor by the integration of the ultra-local model. This improvement for Smith predictor along with the used automatically tuned model for dead time lead to complete model-free Smith predictor. Secondly, the IGPC is introduced as a suitable controller for NCS application. IGPC is designed to benefit from the facilities of the prediction and delay compensation in GPC approach along with the capability of the intelligent structure to cancel the nonlinear term of the plant or the network.

After developing the network-appropriate controller in terms of time-delay challenge, three methods are proposed to detect the attacks at the plant side. In the first method, the predicted nonlinear term of the plant, which is normally calculated by IGPC, and the actual nonlinear term of the plant, which is achieved from the information of the plant input and the direct measurement of its output, are used to generate an indication about the loss of the response efficiency of the plant due to the attacker's bad signal. In the second method, the comparison between the sent control signal and the estimated arrived value of the control signal in the plant side are applied to detect the presence of any attack in the plant side. In the third method, the mathematical framework of behavioral system is applied to detect the attack in the control system. For more reliability the fault accommodation along with IGPC is applied during the second and third methods.

Finally, the two detectors for controller hijacking attack are suggested. The facilitate of the proposed two methods is that no a priori controller mathematical model is required because it's based on the variables of ultra-local model for the controller.

Furthermore, the attack-tolerant scheme for networked control system is introduced. The proposed method is based on deception of the cyber-attack when the data encryption method is broken.

The four contributions lines (controller with time-delay compensation, plant side attack detectors, controller side attack detectors, and the attack-tolerant scheme), are simulated with different systems. The investigation reveals that the proposed contributions give good results and can be considered and utilized and successfully for self-detection of cyber-physical attacks in NCS.

Future work

For the future, there is much to do and can be summarized as following:

- Develop the performance of model-free controller by online adaptation of other parameters for more optimization with respect to NCS.
- Make enhancement and investigation of the fault accommodation method.
- Apply a prediction method to stealthy controller hijacking attack detector and to integrate the detector output to SCADA system to make an exchange with security software for better performance against Stuxnet.
- Combine the three attack detection methods of the plant side and the controller side attack detectors in one detection system with analysis algorithm to have more reliable and accurate detection signal.
- Start to implement shadow or parallel secure network to detect and isolate the cyber-physical attack in case of collapse of the security system.

APPENDIX A

DETAILS OF THE DES ENCRYPTION SYSTEM [128-129]

A.1 The initial permutation

The 64 bits of the input block undergo the permutation of Fig. A.1.

IP

58	**50**	**42**	**34**	**26**	**18**	**10**	**2**
60	**52**	**44**	**36**	**28**	**20**	**12**	**4**
62	**54**	**46**	**38**	**30**	**22**	**14**	**6**
64	**56**	**48**	**40**	**32**	**24**	**16**	**8**
57	**49**	**41**	**33**	**25**	**17**	**9**	**1**
59	**51**	**43**	**35**	**27**	**19**	**11**	**3**
61	**53**	**45**	**37**	**29**	**21**	**13**	**5**
63	**55**	**47**	**39**	**31**	**23**	**15**	**7**

Figure A.1: Initial permutation matrix.

This "matrix" allows the internal changes to the block (i.e. there is no external data input).

The first bit will be bit 58, the second bit 50, etc.

A.2 The median calculation

The initial 64 bits of data are divided into 2 blocks (L and R). *Kn* is the subkey.

Iterations:

$$-L_n = R_{n-1} \quad \text{(A.1)}$$

$$-R_n = L_{n-1} \oplus F(R_{n-1}, K_n) \quad \text{(A.2)}$$

$$-K_n = G(K, n) \quad \text{(A.3)}$$

With

$$-T_n = L_n R_n \quad \text{(A.4)}$$

$$-L_n = t_1 \dots t_{32} \quad \text{(A.5)}$$

$$-R_n = t_{33} \dots t_{64} \quad \text{(A.6)}$$

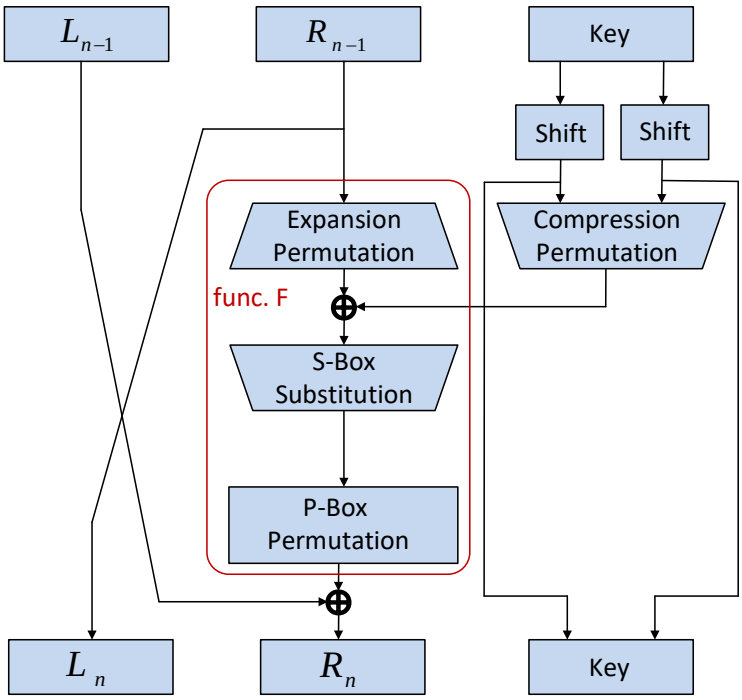

Figure A.2: General stage of the median calculation

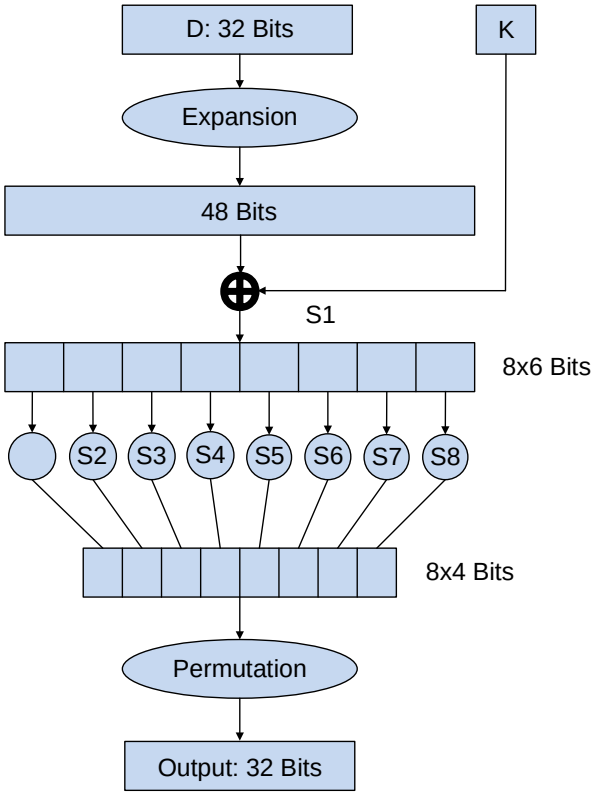

Figure A.3: Detailed F function

The general stage of the median calculation is illustrated in Fig. A.2. The median calculation is done in 16 iterations. The detail of the function F is given in Fig. A.3. Two blocks are processed simultaneously: a block of 32 bits (data) and a block of 48 bits (keys). The result forms a 32-bit block.

1. ***Expansion:*** The 32 bits are extended to 48 bits thanks to an expansion table as in Fig. A.4, (also called extension matrix). Here we find an avalanche effect, the expansion phase is shown in Fig. A.5.

32	**1**	**2**	**3**	**4**	**5**
4	**5**	**6**	**7**	**8**	**9**
8	**9**	**10**	**11**	**12**	**13**
12	**13**	**14**	**15**	**16**	**17**
16	**17**	**18**	**19**	**20**	**21**
20	**21**	**22**	**23**	**24**	**25**
24	**25**	**26**	**27**	**28**	**29**
28	**29**	**30**	**31**	**32**	**1**

Figure A.4: Expansion Matrix.

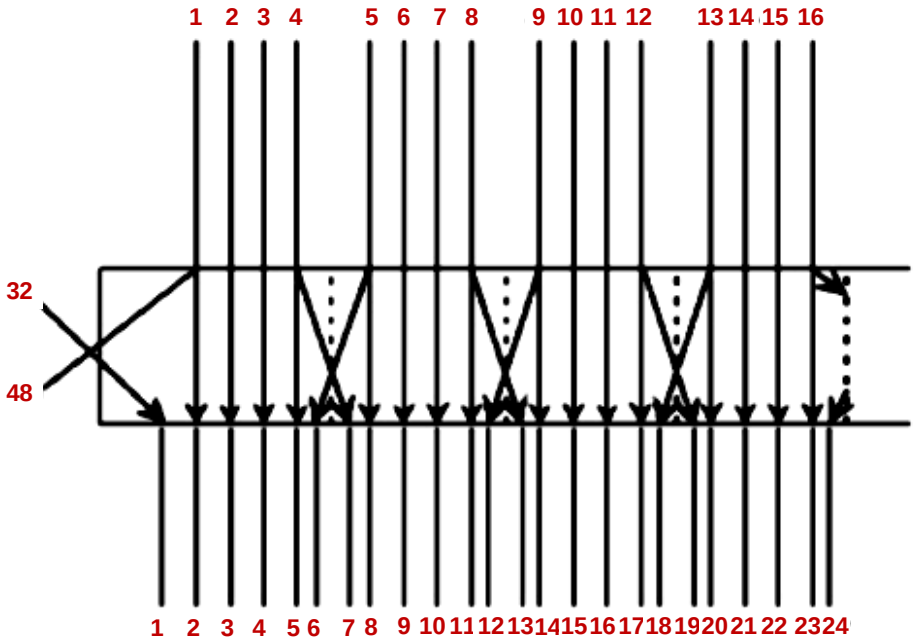

Figure A.5: Expansion phase.

2. ***Addition of the subkey:*** The result of the expansion is added (by an operation $\oplus$) to the subkey K_n corresponding to the iteration according to the formula:

$$E(R_{i-1}) \oplus K_i = B_1 B_2 \dots B_8 \quad \text{(A.7)}$$

The $B_1, B_2, \dots, B_8$ are 6-bit blocks:

$$B_j = b_1 b_2 b_3 b_4 b_5 b_6 \quad \text{(A.8)}$$

3. ***Transformations by S-Boxes:*** Each block B_j then constitutes the input of the substitution operation carried out on the basis of the S-Boxes. Fig A.6, shows the transformation of S-Box while the Special S-Box and the 8 S-Boxes of DES are shown in Fig A.7, and Fig A.8, respectively.

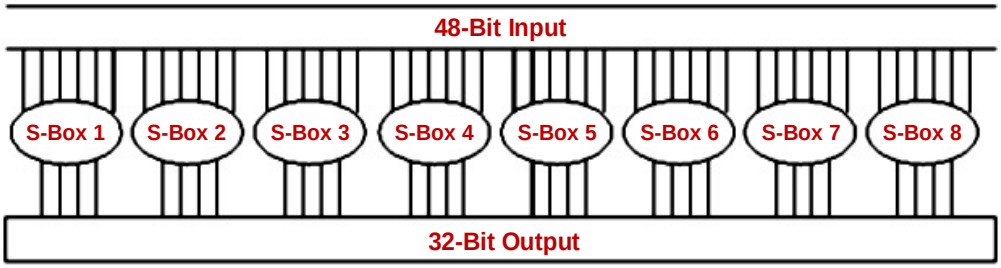

Figure A.6: S-Box Transformations

The substitution operation consists for each S-box to calculate:
$-b_1 b_6 = No.$ line
$-b_2 b_3 b_4 b_5 = No.$ Column

Column No.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Line No.

Figure A.7: Special S-Box

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
0	15	1	2	14	6	11	3	4	9	7	2	13	2	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	1	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	122	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Figure A.8: The 8 S-Boxes of DES

4. ***Transformations by P-Box (permutation of the median calculation):*** The operation of permutation is carried out on the result of the substitution of the S-boxes and is based on the table of the Fig. A.9.

16	**7**	**20**	**21**
29	**12**	**28**	**17**
1	**15**	**23**	**26**
5	**18**	**31**	**10**
2	**8**	**24**	**14**
32	**27**	**3**	**9**
19	**13**	**30**	**6**
22	**11**	**4**	**25**

Figure A.9: Matrix of permutation of the median calculation

The result of this last permutation is noted $F(R_{n-1}, K_n)$.

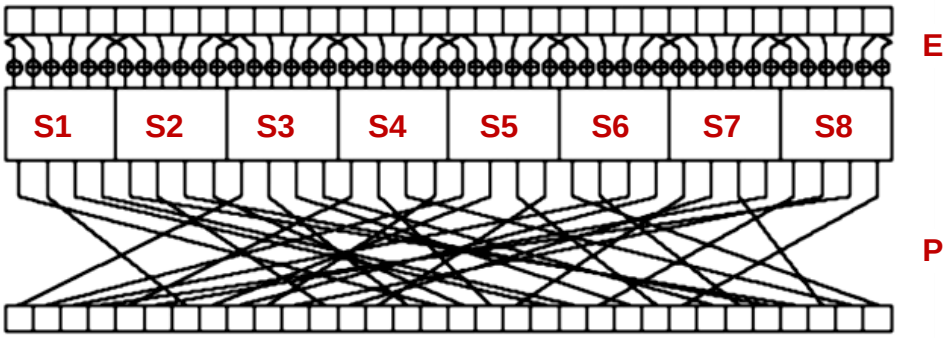

Figure A.10: Detailed round of the median calculation.

A.3 Final permutation

Once the median calculation is complete, the inverse permutation of the initial permutation is performed. However, it is the inverse of the initial permutation, in other words, this table allows to find the starting position. This is not the inverse of the "matrix" of departure!

40	**8**	**48**	**16**	**56**	**24**	**64**	**32**
39	**7**	**47**	**15**	**55**	**23**	**63**	**31**
38	**6**	**46**	**14**	**54**	**22**	**62**	**30**
37	**5**	**45**	**13**	**53**	**21**	**61**	**29**
36	**4**	**44**	**12**	**52**	**20**	**60**	**28**
35	**3**	**43**	**11**	**51**	**19**	**59**	**27**
34	**2**	**42**	**10**	**50**	**18**	**58**	**26**
33	**1**	**41**	**9**	**49**	**17**	**57**	**25**

Figure A.11: Final permutation

A.4 Algorithm for calculating the key $G(k, n)$

The key consists of 64 bits of which 56 are used in the algorithm. The other 8 can be used for error detection where each of these bits will be used as the parity bit of the 7 groups of 8 bits. Thus, the total number of keys is 2^{56}.

The initial key is 64 bits. The calculation takes place in 4 steps:

1. 56-bit reduction: The parity bits are removed. Then proceed to a permutation similar to that of Fig. A.12.

57	**49**	**41**	**33**	**25**	**17**	**9**
1	**58**	**50**	**42**	**34**	**26**	**18**
10	**2**	**59**	**51**	**43**	**35**	**27**
19	**11**	**3**	**60**	**52**	**44**	**36**
63	**55**	**47**	**39**	**31**	**23**	**15**
7	**62**	**54**	**46**	**38**	**30**	**22**
14	**6**	**61**	**53**	**45**	**37**	**29**
21	**13**	**5**	**28**	**20**	**12**	**4**

Figure A.12: Key reduction matrix

2. Division into 28-bit subkeys: The result of the previous step (56 bits) is split into two 28-bit subkeys.

3. Rotation of the key: at each iteration, each 28-bit subkey is rotated 1 or 2 bits to the left according to the Table A.1.

TABLE A.1. Rotating the key

Iteration n	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2

11	2
12	2
13	2
14	2
15	2
16	1

4. Reduction: after concatenation of the two previous subkeys, the resulting key (56 bits) is reduced to a 48-bit subkey based on the matrix of Fig. A.13.

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Figure A.13: Key reduction matrix

The result of this reduction is the subkey K_n added with $E(R_n - 1)$.

APPENDIX B

AN OVERVIEW ON MD5 [129]

B.1 Introduction

MD5 is designed by Ronald Rivest, it's the latest in a series (MD2, MD4). This algorithm produces a 128-bit digest. It was still from some time ago the most widely used hash algorithm. Its specifications are available on the Internet in RFC 1321. The general sequence of the algorithm is shown in Fig. B.1.

- ***Completion:*** addition of padding if necessary, so that the message has a length of 448 mod 512. This addition always takes place.
- ***Add length:*** add the actual length of the message (on 64 bits) after the 448 bits. As a result, the total block size reaches 512 bits. If the length requires more than 64 bits, only the 64 bits of low weight are noted.
- ***Initialization:*** Initialize 4 buffers of 32 bits each (A, B, C, D), which constitutes the IV (Initial Value).
- ***Iterative computation:*** process the message in blocks of 512 bits. There are 4 rounds of 16 operations: block function (512), buffers and primitive functions as in Fig. B.2.

  The final result is obtained by concatenating the results of the additions of registers A, B, C, D with the value of CVq (Chained Value) (see Fig. B.2).

B.2 Algorithm

Fig. B.2 shows the details of the MD5 algorithm, the characteristics of this algorithm are as follows.

$$-CV_O = IV \tag{B.1}$$

$$-CV_{q+1} = \sum_{32} \left[CV_q, RF_I(Y_q, RF_H\left(Y_q, RF_G\left(Y_q, RF_F(Y_q, CV_q)\right)\right)\right] \tag{B.2}$$

$$-MD = CV_L \tag{B.3}$$

where,

IV : initial value of ABCD registers.

Y_q: the 5th block of 512 bits of the message.

L: the number of 512-bit blocks in the message

CV_q: chained variable obtained by the manipulation of the fourth block

RF_x: primitive function dependent on the current round

MD: final result

$\sum_{32}$: addition modulo 2^{32}

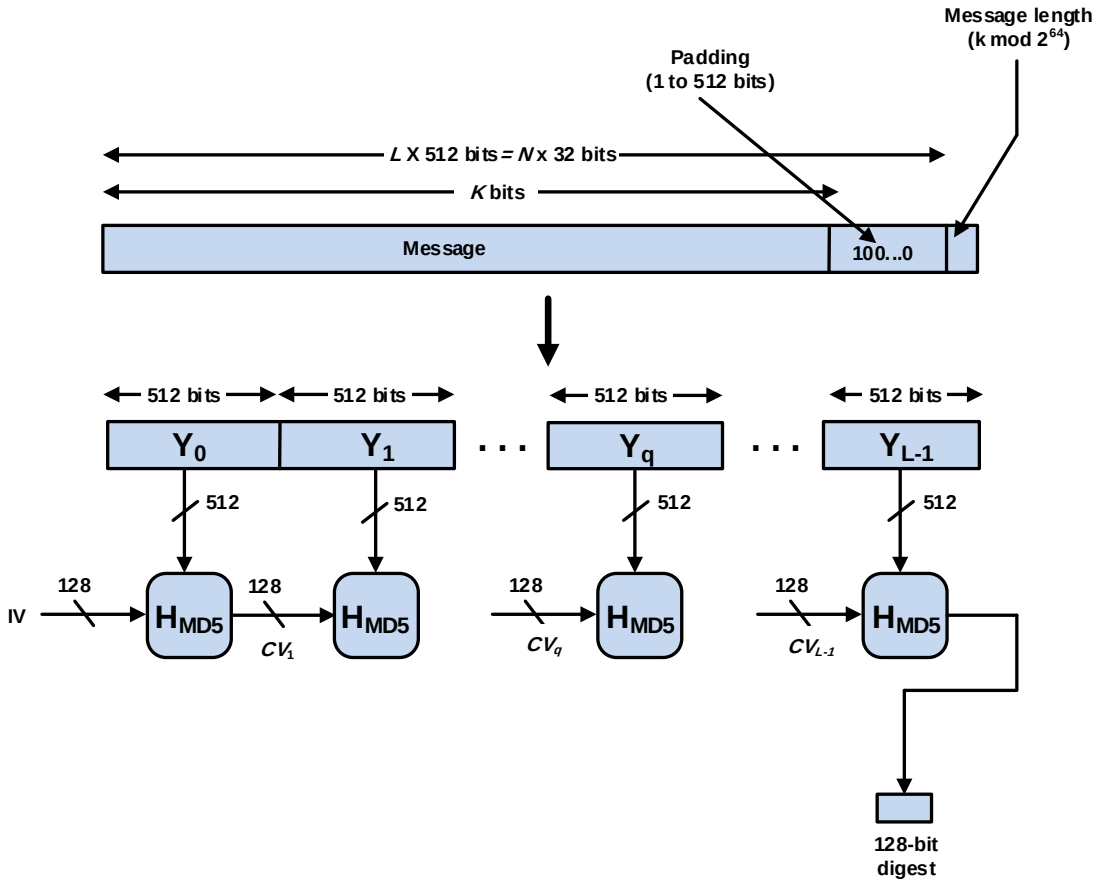

Figure B.1: General sequence of MD5

B.3 Compression function

Each round includes 16 iterations of the form:

$$(A, B, C, D) \leftarrow (D, B + ((A + g\,(B, C, D) + X[k] + T[i]) <<< s), B, C) \qquad \text{(B.4)}$$

The letters *A, B, C, D* refer to the four buffers, but are used according to variable permutations. It should be noted that this only one update of the 32-bit buffers. After 16 steps, each buffer has been updated 4 times. The function *g (B, C, D)* is a different nonlinear

function in each round (denoted F, G, H and I in Fig. B.2). *T[i]* is a constant value derived from the sine function. (<<< s) represents a left circular shift (for each buffer separately) of s bits. The message buffer is represented by *X[k]*.

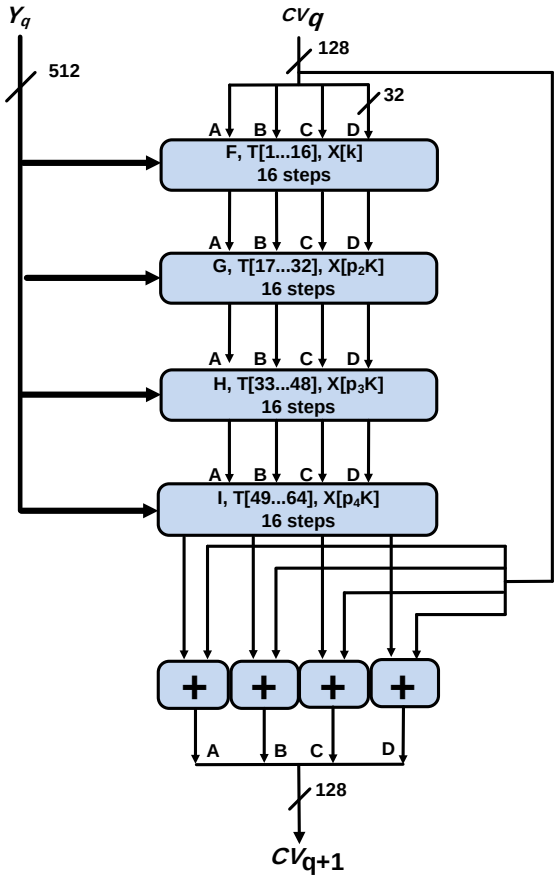

Figure B.2: The rounds of MD5

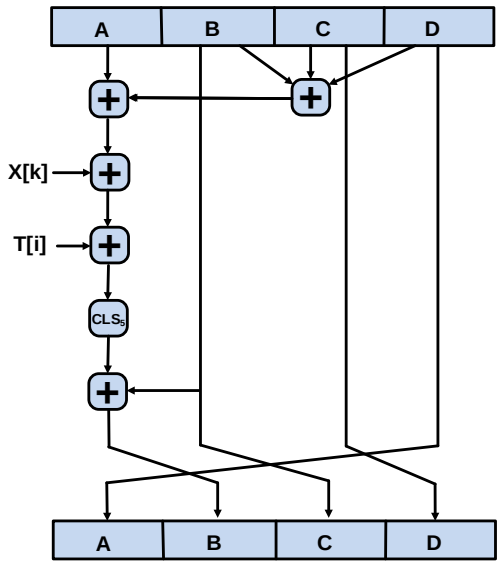

Figure B.3: Elementary Function of MD5

APPENDIX C

DETAILS OF UDP PROTOCOL [130]

C.1 The structure of the UDP

The structure of the UDP header is illustrated in Fig. C.1.

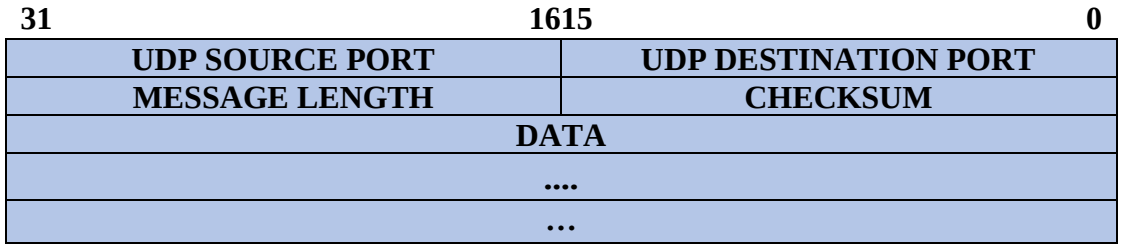

Figure C.1: Structure of the UDP header

1- **UDP SOURCE PORT:** The port number of the packet sender. This field is optional, when specified it indicates the port number that the recipient must use for its response. The value zero (0) indicates that it is unused, so port 0 is not that of a valid service.

2- **UDP DESTINATION PORT:** The port number of the recipient of the packet.

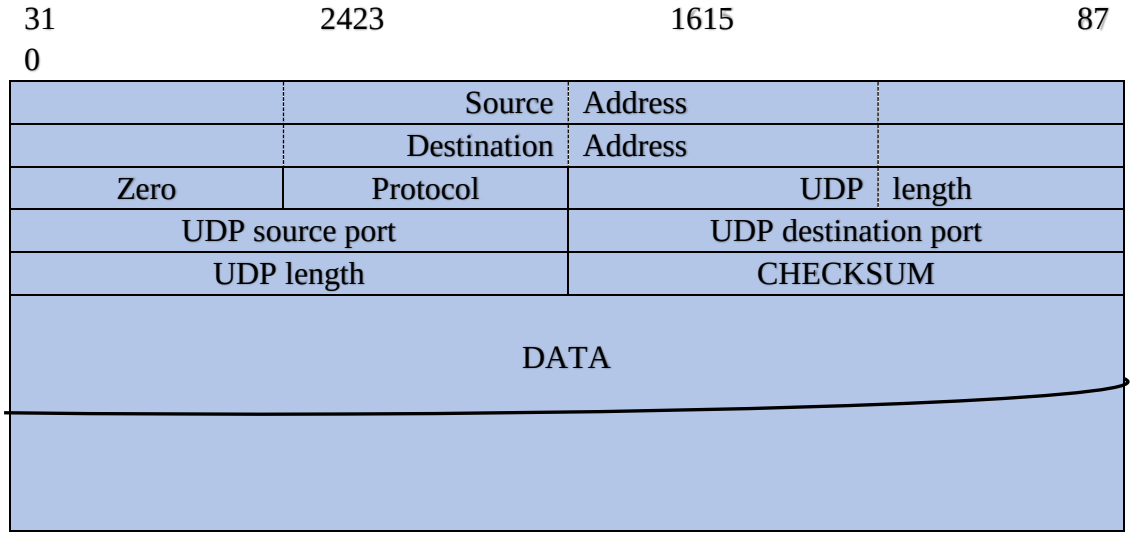

Figure C.2: Case of the non-zero checksum

3- **MESSAGE LENGTH:** This is the length of the packet, so including the header and the message. The minimum length is 8 and the maximum length is 65535 H (IP). In the current case (IP without option) this maximum size is 65 515.

4- **CHECKSUM:** The checksum is optional and not all implementations use it. If it is used, it relates to a pseudo header, Fig. C.2 illustrates it contains. This pseudo header is intended initially to provide protection in case of mis-routed datagrams.

C.2 Reserved / available ports

The port number is an unsigned 16-bit integer, so the range is [0-65535], by construction. We have previously seen that port 0 is not exploitable as a valid service designation, so the actually exploitable segment is [1-65535].

Any machine that uses the TCP / IP stack needs to know a number of well-known services, spotted by a series of well-known ports, to be able to interact with other machines on the Internet.

On a Unix machine, this list of services is placed in the **/ etc / services** file and readable by all users and applications.

Indeed, any service including a program at the application level that starts its network activity, and which is therefore considered to have a server role will assign for itself the port number (s) that it receives according to the Table 2.

This table presents some of the well-known ports, but there are plenty of others. Normally, the Internet Assigned Numbers Authority (IANA) centralizes and diffuses information about all numbers used on the Internet via the Request For Comments (RFC). The latest RFC is over 200 pages. As a consequence, this RFC also concerns the port numbers.

C.3 Port assignment (old method)

Historically, ports from 1 to 255 are reserved for well-known services, more recently this segment has been expanded to [1-1023].

No application can be attributed durably and at the level of the Internet a port number in this segment, without referring to the IANA, which controls the use. From 1024 until 65535, IANA only records the usage requests and reports potential conflicts.

TABLE C.1.: Some of the well-known ports

Name	Port	Protocol	Comment
echo	7	tcp	
echo	7	udp	
ftp-data	20	tcp	#File Transfer [Default Data]
ftp-data	20	udp	#File Transfer [Default Data]
ftp	21	tcp	#File Transfer [Control]
ftp	21	udp	#File Transfer [Control]
ssh	22	tcp	#Secure Shell Login
ssh	22	udp	#Secure Shell Login
smtp	25	tcp	mail #Simple Mail Transfer
smtp	25	udp	mail #Simple Mail Transfer
domain	53	tcp	#Domain Name Server
domain	53	udp	#Domain Name Server
http	80	tcp	www www-http #World Wide Web HTTP
http	80	udp	www www-http #World Wide Web HTTP
pop3	110	tcp	#Post Office Protocol - Version 3
pop3	110	udp	#Post Office Protocol - Version 3
imap	143	tcp	#Interim Mail Access Protocol
imap	143	udp	#Interim Mail Access Protocol
https	443	tcp	#Secure World Wide Web HTTP
https	443	udp	#Secure World Wide Web HTTP

TABLE C.2: An example of the category of segment [1024-49151]

Name	Port	Protocol	Comment
bpcd	13782	tcp	VERITAS NetBackup
bpcd	13782	udp	VERITAS NetBackup

C.4 Port assignment (new method)

With the explosion of the number of registered services, IANA has modified the preceding segmentation. Port numbers are now classified into the following three categories:

a) *The segment [1-1023]* is always reserved for well-known services. Well-known services are designated by IANA and are implemented by applications that run with privileged rights (root on a Unix machine).

b) *The segment [1024-49151]* is that of the registered services. They are listed by the IANA and can be used by processes with ordinary rights, an example is illustrated in Table 3.

c) *The segment [49152, 65535]* is dynamic assignments and private services.

BIBLIOGRAPHY

[1] Hong Y (1998) Networked Control Systems. [Online document] available http://www.enme.umd.edu/ice lab/ncs/ncs.html

[2] Y. Qiao, G. P. Liu, G. Zheng, and W. Hu, "NCSLab: A Web-based global-scale control laboratory with rich interactive features," IEEE Trans. Ind. Electron., vol. 57, no. 10, pp. 3253–3265, Oct. 2010.

[3] H. Yu, Y. Liu, and M. S. Hasan, "Modelling and remote control of an excavator," Int. J. Autom. Comput., vol. 7, no. 3, pp. 349–358, Aug. 2010.

[4] J. Arata, H. Takahashi, P. Pitakwatchara, S. Warisawa, K. Tanoue, K. Konishi, S. Ieiri, S. Shimizu, N. Nakashima, K. Okamura, Y. Fujino, Y. Ueda, P. Chotiwan, M. Mitsuishi, and M. Hashizume, "A remote surgery experiment between Japan and Thailand over Internet using a low latency CODEC system," in Proc. IEEE Int. Conf. Rob. Autom., 2007, pp. 953–959.

[5] T. Samad, J. S. Bay, and D. Godbole, "Network-centric systems for military operations in urban terrain: The role of UAVs," Proc. IEEE, vol. 95, no. 1, pp. 92–107, Jan. 2007.

[6] Gupta R.A., Chow MY. (2008) Overview of Networked Control Systems. In: Wang FY., Liu D. (eds) Networked Control Systems. Springer, London

[7] Quin, S. J. and Badgwell, T. A. [2003], 'A survey of industrial model predictive control technology', Control Engineering Practice 11(7), 733–764.

[8] S. Amin, A. M. Bayen, Alexandre, X. Litrico, S. S. Sastry, "Cyber security of water SCADA systems – Part II: Attack detection using enhanced hydrodynamic models," IEEE Transactions on Control Systems Technology, 21(5):1679-1693, 2013

[9] S. Amin, X. Litrico, S. S. Sastry, A. M. Bayen, "Cyber security of water SCADA systems – Part I: Analysis and experimentation of stealthy deception attacks," IEEE Transactions on Control Systems Technology, 21(5):1963-1970, 2013

[10] A. A. C?rdenas, S. Amin, and S. S. Sastry, "Secure control: Towards survivable cyber-physical systems," in Proc. 28th Int. Conf. Distrib. Comput. Syst. Workshops, 2008, pp. 495–500.

[11] A. Teixeira, H. Sandberg, and K. H. Johansson, "Networked control systems under cyber-attacks with applications to power networks," in Proc. Amer. Control Conf., 2010, pp. 3690–3696.

[12] U. Attorney, "Willows man arrested for hacking into tehama colusa canal authority computer system," 2007.

[13] T. Reed, At the abyss: an insider's history of the Cold War. Random House LLC, 2007.

[14] D. Kravets, "Feds: Hacker disabled offshore oil platform leak-detection system," 2009.

[15] A. Greenberg, "Hackers cut cities power," Forbes, Jaunuary, 2008.

[16] P. Quinn-Judge, "Cracks in the system," TIME Magazine (January 9, 2002), 2002.

[17] J. Slay and M. Miller, Lessons learned from the maroochy water breach. Springer, 2007.

[18] J. Meserve, "Sources: Staged cyber attack reveals vulnerability in power grid," CNN. com, vol. 26, 2007.

[19] J. Leyden, "Polish teen derails tram after hacking train network," The Register, vol. 11, 2008.

[20] A. A. Cardenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems.," in HotSec, 2008.

[21] J. P. Farwell and R. Rohozinski, "Stuxnet and the future of cyber war," Survival, vol. 53, no. 1, pp. 23{40, 2011.

[22] J.Marko, "A silent attack, but not a subtle one," New York Times, vol. 26, p. A6, 2010.

[23] T. M. Chen and S. Abu-Nimeh, "Lessons from Stuxnet," in Computer, vol. 44, no. 4, pp. 91-93, April 2011.

[24] R. Langner, "Stuxnet: Dissecting a Cyberwarfare Weapon," in IEEE Security & Privacy, vol. 9, no. 3, pp. 49-51, May-June 2011.

[25] D. Kushner, "The real story of stuxnet," in IEEE Spectrum, vol. 50, no. 3, pp. 48-53, March 2013.

[26] P. A. Yannakogeorgos and E. Tikk, "Stuxnet as cyber-enabled sanctions enforcement," 2016 International Conference on Cyber Conflict (CyCon U.S.), Washington, DC, 2016, pp. 1-6.

[27] D. E. Whitehead, K. Owens, D. Gammel and J. Smith, "Ukraine cyber-induced power outage: Analysis and practical mitigation strategies," 2017 70th Annual Conference for Protective Relay Engineers (CPRE), College Station, TX, 2017, pp. 1-8.

[28] Dave McMillen, "Attacks Targeting Industrial Control Systems (ICS) Up 110 Percent," IBM SecurityIntelligence, December, 2016.

[29] S. Amin, "On Cyber Security for Networked Control Systems," UC Berkeley, 2011.

[30] Sliwa and E. Benoist, "Wireless sensor and actor networks: e-Health, e-Science, e-Decisions," 2011 International Conference on Selected Topics in Mobile and Wireless Networking (iCOST), Shanghai, 2011, pp. 1-6.

[31] D. J. Antunes, J. P. Hespanha and C. J. Silvestre, "Volterra Integral Approach to Impulsive Renewal Systems: Application to Networked Control," in IEEE Transactions on Automatic Control, vol. 57, no. 3, pp. 607-619, March 2012.

[32] Y. Shi, J. Huang and B. Yu, "Robust Tracking Control of Networked Control Systems: Application to a Networked DC Motor," in IEEE Transactions on Industrial Electronics, vol. 60, no. 12, pp. 5864-5874, Dec. 2013.

[33] Y. A. Harfouch, S. Yuan and S. Baldi, "Adaptive control of interconnected networked systems with application to heterogeneous platooning," 2017 13th IEEE International Conference on Control & Automation (ICCA), Ohrid, 2017, pp. 212-217.

[34] W. Cao, C. Lin, L. Zhang, Y. Ming and H. Liu, "Fuzzy sliding mode control of networked control systems and applications to independent-drive electric vehicles," 2017 IEEE International Conference on Industrial Technology (ICIT), Toronto, ON, 2017, pp. 1397-1402.

[35] K. Yoshizawa, H. Hashimoto, M. Wada and S. Mori, "Path tracking control of mobile robots using a quadratic curve," Proceedings of Conference on Intelligent Vehicles, Tokyo, 1996, pp. 58-63.

[36] L. Litz, O. Gabel and I. Solihin, "NCS-Controllers for Ambient Intelligence Networks - Control Performance versus Control Effort," Proceedings of the 44th IEEE Conference on Decision and Control, 2005, pp. 1571-1576.

[37] N. Lechevin, C. A. Rabbath, A. Tsourdos and B. A. White, "A Causal Discrete-time Estimator-Predictor for Unicycle Trajectory Tracking," Proceedings of the 44th IEEE Conference on Decision and Control, 2005, pp. 2658-2663.

[38] P. Arena, L. Fortuna, M. Frasca, G. Lo Turco, L. Patane and R. Russo, "Perception-based navigation through weak chaos control," Proceedings of the 44th IEEE Conference on Decision and Control, 2005, pp. 221-226.

[39] D. Shah and A. J. Mehta, "Robust controller design for Networked Control System," 2014 International Conference on Computer, Communications, and Control Technology (I4CT), Langkawi, 2014, pp. 388-393.

[40] P. G. Flikkema, K. R. Yamamoto, S. Boegli, C. Porter and P. Heinrich, "Towards Cyber-Eco Systems: Networked Sensing, Inference and Control for Distributed Ecological Experiments," 2012 IEEE International Conference on Green Computing and Communications, Besancon, 2012, pp. 372-381.

[41] S. R. M. Canovas and C. E. Cugnasca, "Implementation of a Control Loop Experiment in a Network-Based Control System With LonWorks Technology and IP Networks," in IEEE Transactions on Industrial Electronics, vol. 57, no. 11, pp. 3857-3867, Nov. 2010.

[42] Mo-Yuen Chow, "Time-sensitive network-control systems and applications," 2010 IEEE International Symposium on Industrial Electronics, Bari, 2010, pp. 4616-4657.

[43] M. Delrobaei and K. A. McIsaac, "Parking control of a center-articulated mobile robot in presence of measurement noise," 2010 IEEE Conference on Robotics, Automation and Mechatronics, Singapore, 2010, pp. 453-457.

[44] Thomas B. Sheridan, "Telerobotics, automation, and human supervisory control," The MIT Press, Cambridge, MA, 2002.

[45] T. Hansen et al., "Implementing force-feedback in a telesurgery environment, using parameter estimation," 2012 IEEE International Conference on Control Applications, Dubrovnik, 2012, pp. 859-864.

[46] J. M. Smith, B. P. DeJong, E. Karadogan and J. Hasbany, "Image display visualization in teleoperation," 2017 IEEE Sensors Applications Symposium (SAS), Glassboro, NJ, 2017, pp. 1-5.

[47] A. Jafari, M. Nabeel, H. Singh and J. H. Ryu, "Stable and transparent teleoperation over communication time-delay: Observer-based input-to-state stable approach," 2016 IEEE Haptics Symposium (HAPTICS), Philadelphia, PA, 2016, pp. 235-240..

[48] Chung-Kuo Chang, J. M. Overhage and J. Huang, "An application of sensor networks for syndromic surveillance," Proceedings. 2005 IEEE Networking, Sensing and Control, 2005., 2005, pp. 191-196.

[49] Z. Xing and Y. Xia, "Distributed Federated Kalman Filter Fusion Over Multi-Sensor Unreliable Networked Systems," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 63, no. 10, pp. 1714-1725, Oct. 2016.

[50] K. Xiao, J. Li and C. Yang, "Exploiting Correlation for Confident Sensing in Fusion-Based Wireless Sensor Networks," in IEEE Transactions on Industrial Electronics, vol. 65, no. 6, pp. 4962-4972, June 2018.

[51] Seungmin Park, Jin Won Kim, Kwangyong Lee, Kee-Young Shin and Daeyoung Kim, "Embedded sensor networked operating system," Ninth IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC'06), Gyeongju, 2006.

[52] Seungmin Park, Jin Won Kim, Kee-Young Shin and Daeyoung Kim, "A nano operating system for wireless sensor networks," 2006 8th International Conference Advanced Communication Technology, Phoenix Park, 2006, pp. 4 pp.-348.

[53] S. Halder and S. DasBit, "Design of a Probability Density Function Targeting Energy-Efficient Node Deployment in Wireless Sensor Networks," in IEEE Transactions on Network and Service Management, vol. 11, no. 2, pp. 204-219, June 2014.

[54] S. Olariu and Q. Xu, "Information assurance in wireless sensor networks," 19th IEEE International Parallel and Distributed Processing Symposium, 2005.

[55] Y. Cui et al., "End-to-End Visual Target Tracking in Multi-robot Systems Based on Deep Convolutional Neural Network," 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, 2017, pp. 1113-1121.

[56] G. L. Mariottini, G. Pappas, D. Prattichizzo and K. Daniilidis, "Vision-based Localization of Leader-Follower Formations," Proceedings of the 44th IEEE Conference on Decision and Control, 2005, pp. 635-640.

[57] D. Folio and V. Cadenat, "A controller to avoid both occlusions and obstacles during a vision-based navigation task in a cluttered environment," Proceedings of the 44th IEEE Conference on Decision and Control, 2005, pp. 3898-3903.

[58] S. Rathinam, Zu Kim, A. Soghikian and R. Sengupta, "Vision Based Following of Locally Linear Structures using an Unmanned Aerial Vehicle," Proceedings of the 44th IEEE Conference on Decision and Control, 2005, pp. 6085-6090.

[59] J. Pinto, "Fieldbus-conflicting 'standards' emerge, but interoperability is still elusive. Design Engineering, UK, October 1999. Available at: http://www.jimpinto.com/writings/fieldbus99.html

[60] Feng-Li Lian, J. Moyne and D. Tilbury, "Network design consideration for distributed control systems," in IEEE Transactions on Control Systems Technology, vol. 10, no. 2, pp. 297-307, March 2002.

[61] Y. Tipsuwan and Mo-Yuen Chow, "Gain scheduler middleware: a methodology to enable existing controllers for networked control and teleoperation-part II: teleoperation," in IEEE Transactions on Industrial Electronics, vol. 51, no. 6, pp. 1228-1237, Dec. 2004.

[62] R. Lu, Y. Xu and R. Zhang, "A New Design of Model Predictive Tracking Control for Networked Control System Under Random Packet Loss and Uncertainties," in IEEE Transactions on Industrial Electronics, vol. 63, no. 11, pp. 6999-7007, Nov. 2016.

[63] Changhong Wang and Yufeng Wang, "Design networked control systems via time-varying delay compensation approach," Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No.04EX788), 2004, pp. 1371-1375 Vol.2.

[64] Y. Koren, Z. J. Pasek, A. G. Ulsoy, and U. Benchetrit “Real-time open control architectures and system performance,” CIRP Annals-Manufacturing Technology, vol.45, no 1, pp. 377-380, 1996.

[65] G. C. Walsh and Hong Ye, "Scheduling of networked control systems," in IEEE Control Systems, vol. 21, no. 1, pp. 57-65, Feb 2001.

[66] A. T. Al-Hammouri, M. S. Branicky, V. Liberatore and S. M. Phillips, "Decentralized and dynamic bandwidth allocation in networked control systems," Proceedings 20th IEEE International Parallel & Distributed Processing Symposium, Rhodes Island, 2006.

[67] M. Velasco, J. M. Fuertes, C. Lin, P. Marti and S. Brandt, "A control approach to bandwidth management in networked control systems," 30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004, 2004, pp. 2343-2348 Vol. 3.

[68] Hong Seong Park, Yong Ho Kim, Don-Sung Kim and Wook Hyun Kwon, "A scheduling method for network-based control systems," in IEEE Transactions on Control Systems Technology, vol. 10, no. 3, pp. 318-330, May 2002.

[69] Z. Li and M. Y. Chow, "Adaptive Multiple Sampling Rate Scheduling of Real-time Networked Supervisory Control System - Part I," IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics, Paris, 2006, pp. 4604-4609.

[70] Z. Li and M. Y. Chow, "Adaptive Multiple Sampling Rate Scheduling of Real-time Networked Supervisory Control System - Part II," IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics, Paris, 2006, pp. 4615-4620.

[71] E. Cosman, "Patch management at Dow chemical," in ARC Tenth Annual Forum on Manufacturing, 2006.

[72] B. Krebs, "Cyber Incident Blamed for Nuclear Power Plant Shutdown," Washington Post, June 5, 2008.

[73] E. Byres, D. Leversage and N. Kube, "Security incidents and trends in SCADA and process industries," The Industrial Ethernet Book 39(2), pp. 12–20, 2007.

[74] S. Hurd, R. Smith and G. Leischner, "Tutorial: Security in electric utility control systems," in 61st Annual Conference for Protective Relay Engineers, pp. 304–309, 2008.

[75] P. Tsang, and S. W. Smith, "YASIR: A low-latency high-integrity security retrofit for legacy SCADA systems," in '23rd International Information Security Conference (IFIC SEC)', pp. 445–459, 2008.

[76] A. K. Wright, J. A. Kinast, J. McCarty, "Low-latency cryptographic protection for SCADA communications," in Applied Cryptography and Network Security (ACNS), pp. 263–277, 2004.

[77] R. A. Gupta and M. Y. Chow, "Performance assessment and compensation for secure networked control systems," in Proc. 34th Annual Conference of IEEE Industrial Electronics (IECON 2008), Orlando, Florid, 10-13 Nov. 2008, pp.2929-2934.

[78] Z. h. Pang and G. Liu, "Secure networked control systems under data integrity attacks," in Proc. 29th Chinese Control Conference (CCC), Beijing, China, 29-31 July 2010, pp.5765-5771.

[79] W. Zeng and M. Y. Chow, "A trade-off model for performance and security in secured Networked Control Systems," in Proc. 2011 IEEE International Symposium on Industrial Electronics (ISIE), Gdansk, Poland, 27-30 June 2011, pp. 1997- 2002.

[80] S. Cui, Z. Han, S. Kar, T. T. Kim, H. V. Poor and A. Tajer, "Coordinated data-injection attack and detection in the smart grid: A detailed look at enriching detection solutions, "IEEE Signal Processing Magazine, vol.29, no.5, pp. 106-115, Sept. 2012.

[81] V. Perez, M. T. Garip, S. Lam and L. Zhang,"Security evaluation of a control system using Named Data Networking," in Proc. 21st IEEE International Conference on Network Protocols (ICNP), Goettingen, Germany, 7-10 Oct. 2013, pp.1-6.

[82] Zhang Liying, Xie Lun, Li Weize and Wang Zhiliang, "A secure mechanism for networked control systems based on TrueTime," in Proc. International Conference on Cyberspace Technology (CCT 2013), Beijing, China, 23-23 Nov. 2013, pp.44-49.

[83] Zhang Liying, Xie Lun, Li Weize and Wang Zhiliang, "Security Solutions for Networked Control Systems Based on DES Algorithm and Improved Grey Prediction Model", IJCNIS, vol.6, no.1, pp.78-85, 2014.

[84] Fedora Documentation Project, "Fedora 13 Security Guide", Fultus Corporation, July 12, 2010.

[85] Simmonds, A; Sandilands, P; van Ekert, L "An Ontology for Network Security Attacks". Lecture Notes in Computer Science, 2004.

[86] W. Stallings, Cryptography and Network Security: Principles and Practice, 4th ed. Englewood Cliffs, NJ: Pearson/Prentice-Hall, 2006.

[87] Munadi; Akbar, M.A., "Simulation of fuzzy logic control for DC servo motor using Arduino based on MATLAB/Simulink," 2014 International Conference on Intelligent Autonomous Agents, Networks and Systems, pp.42-46, 19-21 Aug. 2014.

[88] T. K. Wang, L. H. Zhou, P. Han and Q. Zhang, "Complete Compensation for Time Delay in Networked Control System Based on GPC and BP Neural Network," 2007 International Conference on Machine Learning and Cybernetics, Hong Kong, pp. 637-641, 2007.

[89] T. Slama, D. Aubry, A. Trevisani, R. Oboe and F. Kratz, "Teleoperation systems over the Internet: Experimental validation of a bilateral Generalized Predictive Controller," 2007 European Control Conference (ECC), Kos, pp. 1203-1210, 2007.

[90] J. Zhang, Y. Xia and P. Shi, "Design and Stability Analysis of Networked Predictive Control Systems," in IEEE Transactions on Control Systems Technology, vol. 21, no. 4, pp. 1495-1501, July 2013.

[91] C. F. Caruntu and F. C. Braescu, "Further analysis on network-induced time-varying delay modeling methods used in GPC design," 2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP), Brasov,pp. 893-898, 2017.

[92] S. Chao, Z. Ying, D. Junxiang and L. Jinxing, "Networked control system time delay compensation based on improved implicit GPC," 2017 36th Chinese Control Conference (CCC), Dalian,pp. 4562-4566, 2017.

[93] Dong Yue, Qing-Long Han, James Lam,"Network-based robust H∞ control of systems with uncertainty," In Automatica, vol.41, Issue 6, Pages 999-1007, 2005.

[94] Peng Lin, Yingmin Jia, Lin Li, "Distributed robust H∞ consensus control in directed networks of agents with time-delay," In Systems & Control Letters, Vol. 57, Issue 8, Pages 643-653, 2008.

[95] L. Qiu, Y. Shi, F. Yao, G. Xu and B. Xu, " Network-Based Robust H2/H∞ Control for Linear Systems With Two-Channel Random Packet Dropouts and Time Delays," in IEEE Transactions on Cybernetics, vol. 45, no. 8, pp. 1450-1462, Aug. 2015.

[96] Z. Zhang, H. Zhang, Z. Wang and J. Feng, "Optimal robust non-fragile H∞ control for networked control systems with uncertain time-delays," Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, pp. 4076-4081, 2014.

[97] M. Fliess, C. Join, M. Mboup, and H. Sira-Ramirez, "Vers une commande multivariable sans modèle," in Proc. Conférence internationale francophone d'automatique (CIFA'06), Bordeaux, France, 2006.

[98] M. Fliess and C. Join, "Model-free control and intelligent PID controllers: towards a possible trivialization of nonlinear control?" in 15th IFAC Symposium on System Identification (SYSID 2009). Saint-Malo, France: IFAC, pp. 1531-1550, 2009.

[99] M. Fliess, C. Join, "Model-free control," Int. J. Control, vol.86, pp. 2228–2252, 2013.

[100] E. Delaleau, "A proof of stability of model-free control," in Proc. 2014 IEEE Conference on Norbert Wiener in the 21st Century (21CW), Boston, MA, pp. 1-7, 2014.

[101] Francisco Javier Carrillo, Frédéric Rotella, "Some contributions to estimation for model-free control," In IFAC-PapersOnLine, Vol. 48, Issue 28, pp. 150-155, 2015.

[102] Smith, O.J.M.,"Closer control of loops with dead time," Chemical Engineering Progress, Vol. 53, No. 5, pp.217–219,1957.

[103] H. Thabet, M. Ayadi, "Design of adaptive PID controllers based on adaptive Smith predictor for ultra-local model control," Int. J. Automation and Control, vol.11, No. 2, pp. 222–238, 2017.

[104] Veronesi Massimiliano,"Performance Improvement of Smith Predictor Through Automatic Computation Of Dead Time,"Yokogawa Technical Reports, P.P. 25-30, 2003.

[105] M. Veronesi, A. Visioli, "Controllo di processi industriali affetti da ritardo," Atti del Convegno Automazione e processi decisionali, ANIPLA, 2000.

[106] D.W. Clarke, C.Mohtadi, and P.C. Tuffs, “Generalized predictive control—Part I. The basic algorithm,” Automatica, vol. 23, Issue 2, pp. 137–148, 1987.

[107] D.W. Clarke, C. Mohtadi, and P.C. Tuffs, “Generalized Predictive Control—Part II Extensions and interpretations,” Automatica, vol. 23, Issue 2, pp. 149–160, 1987.

[108] T.W. Yoon and D.W. Clarke, "Advances in Model-Based Predictive Control," chapter Towards Robust Adaptive Predictive Control, Oxford University Press, pp.402-414, 1994.

[109] W. A. Silva, L. L. N. dos Reis, B. C. Torrrico and R. N. de C. Almeida, "Speed control in switched reluctance motor based on generalized predictive control," 2013 Brazilian Power Electronics Conference, Gramado, 2013, pp. 903-908.

[110] W. J. He et al., "Temperature Control for Nano-Scale Films by Spatially-Separated Atomic Layer Deposition Based on Generalized Predictive Control," in IEEE Transactions on Nanotechnology, vol. 14, no. 6, pp. 1094-1103, Nov. 2015.

[111] N. Network, Z. Jiang, Q. Wang and Y. Li, "Generalized predictive control of DEAP actuator based on RBF," 2017 11th Asian Control Conference (ASCC), Gold Coast, QLD, pp. 1632-1637, 2017.

[112] J. Salcedo Hernandez, R. Rivas-Perez and J. J. Sotomayor Moriano, "Design of a Generalized Predictive Controller for Temperature Control in a Cement Rotary Kiln," in IEEE Latin America Transactions, vol. 16, no. 4, pp. 1015-1021, April 2018.

[113] Rossiter, J. A., Model-Based Predictive Control: A Practical Approach, CRC Press, 2003.

[114] Zhijie Tang, PengYan and Luojun, "A novel ROV depth control based on LSM fitting predictor and fuzzy compensation," 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Chengdu, pp. V2-612-V2-614, 2010.

[115] S. Amin, X. Litrico, S. Sastry and A. M. Bayen, "Cyber Security of Water SCADA Systems—Part I: Analysis and Experimentation of Stealthy Deception Attacks," in IEEE Transactions on Control Systems Technology, vol. 21, no. 5, pp. 1963-1970, Sept. 2013.

[116] Nourian, A.; Madnick, S., "A Systems Theoretic Approach to the Security Threats in Cyber Physical Systems Applied to Stuxnet," IEEE Transactions on Dependable and Secure Computing, vol.PP, no.99, pp.1-1. 2015.

[117] Frédéric Lafont, Jean-François Balmat, Nathalie Pessel, Michel Fliess, "A model-free control strategy for an experimental greenhouse with an application to fault accommodation," Computers and Electronics in Agriculture, vol.110, Pages 139-149, January 2015.

[118] Tushar Jain, Joseph J. Yamé, Dominique Sauter, "Model-free reconfiguration mechanism for fault tolerance," Int. J. Appl. Math. Comput. Sci., Vol. 22, No.1, pp. 125–137, 2012.

[119] Gupta, R.A.; Mo-Yuen Chow, "Networked Control System: Overview and Research Trends," IEEE Transactions on Industrial Electronics, vol.57, no.7, pp.2527-2535, July 2010.

[120] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," in First Workshop on Secure Control Systems, Cyber Physical Systems Week 2010, April 2010.

[121] I. Shames, A. M. Teixeira, H. Sandberg, and K. H. Johansson, "Distributed fault detection for interconnected second-order systems," Automatica, vol. 47, no. 12, pp. 2757 – 2764, 2011.

[122] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, Diagnosis and Fault-Tolerant Control, 2nd ed. Springer, 2006

[123] J. Gertler, Fault Detection and Diagnosis in Engineering Systems. New York: Marcel Dekker, Inc., 1998.

[124] A. Rosich, H. Voos, Y. Li and M. Darouach, "A model predictive approach for cyber-attack detection and mitigation in control systems," 52nd IEEE Conference on Decision and Control, Florence, 2013, pp. 6621-6626.

[125] Joe D. Hoffman, Numerical Methods for Engineers and Scientists, Marcel Dekker, Inc, second edition,2001.

[126] Ž. Šitum, B. Novaković, J. Petrić, "Identification and Control of Pneumatic Servodrives," in Proc. 9th Mediterranean Conference on Control and Automation, June 27-29, Dubrovnik, 2001.

[127] Litrico, X., Malaterre, P., Baume, J., Vion, P., and Ribot-Bruno, J., "Automatic Tuning of PI Controllers for an Irrigation Canal Pool," Journal of Irrigation and Drainage Engineering, 2007 133:1, 27-37.

[128] William Stallings, Cryptography and Network Security: Principles and Practice, 3rd ed. Prentice Hall, 2003.

[129] Didier Müller, Les Codes secrets décryptés, City Editions, 2007.

[130] Eric Hall, Internet Core Protocols: The Definitive Guide, O'Reilly Media-California USA, 2000.

Vers l'auto-détection des attaques cyberphysiques dans les systèmes de contrôle

Résumé : Un Système Contrôlé en Réseau (SCR en français, NCS (Networked Control System) en anglais) est un système de contrôle/commande distribué dans lequel les informations sont échangées en temps réel via un réseau reliant capteurs, actionneurs, contrôleurs, …) Ces SCR sont présents dans de nombreuses applications industrielles tels que les systèmes de production, les systèmes contrôlés à distance, les véhicules autonomes, la téléopération, …. Les principaux avantages de ces systèmes sont la flexibilité de leur architecture et la réduction des coûts d'installation et de maintenance, le principal inconvénient est les effets dus au réseau tels que les retards de transmission, qui influencent les performances et la stabilité de la boucle de régulation, ces systèmes sont également vulnérables aux cyber – attaques.

Cette thèse apporte quelques contributions sur la détection des cyber attaques ainsi que le développement d'un contrôleur capable de traiter les effets des retards temporels.

Pour atteindre cet objectif, la méthode proposée est d'adapter une commande sans modèle et d'améliorer son utilisation dans les Systèmes Contrôlés en Réseau. L'idée principale est basée sur le bénéfice mutuel d'un prédicteur de Smith et du modèle de base de la commande sans modèle. Ensuite, la structure intelligente de la commande sans modèle est appliquée avec une commande prédictive généralisée (GPC Generalized Predictive Control) de manière à obtenir une commande prédictive généralisée intelligente, qui est une amélioration du contrôleur généralisé standard. Ce contrôleur est conçu selon deux méthodes différentes pour détecter les cyber attaques.

Parallèlement, un nouveau mécanisme de sécurité basé sur une réponse trompeuse pour les cyber attaques dans les Systèmes Contrôlés en Réseau est proposé. Le mécanisme proposé peut permettre d'arrêter une cyber-attaque en apportant une dernière ligne de défense lorsque l'attaquant a un accès à l'installation distante.

Enfin, deux détecteurs d'attaque de piratage de commande sont introduits. L'objectif est de pouvoir détecter une attaque tel que le cas Stuxnet où le contrôleur a été détourné par reprogrammation. L'avantage des détecteurs proposés est qu'il ne nécessite pas d'avoir a priori un modèle mathématique du contrôleur.

Toward Self-Detection of Cyber-Physical Attacks in Control Systems

Abstract: A networked control system (NCS) is a control system in which the control loop is closed over a real-time network. NCSs are used in many industrial applications, and also in applications such as remote control, unmanned aerial vehicles or surgical teleoperation, ... The major advantages of NCS are a flexible architecture and a reduction of installation and maintenance costs, the main disadvantage of NCS is the network effects, such as time-delays, that influence the performance and stability of the control loop. These systems are also vulnerable to cyber attacks.

This thesis makes some contributions regarding the detection of cyber-physical attacks as well as the development of a controller which capable of dealing with the other the bad effects of the network like time-delays.

To achieve this goal, the proposed approach is to adapt model-free controller and to improve its use in NCS. The main idea is based on mutual benefit between Smith predictor and the basic model-free controller. Then, the intelligent structure of model-free control is applied along with Generalized Predictive Controller (GPC) to achieve the Intelligent Generalized Predictive Controller (IGPC) as an enhancement for the standard GPC. The IGPC is designed along with two different methods for cyber-attack detection.

Moreover, a new security mechanism based on the deception for the cyber-physical attacks in NCS is proposed, this mechanism can allow to stop the cyber-attacks by providing the last line of defense when the attacker has an access to the remote plant.

Finally, two detectors for controller hijacking attack are introduced. The objective is to be able to detect an attack such as the Stuxnet case where the controller has been reprogrammed and hijacked. The advantage of these proposed detectors is that there is not necessary to have a priori mathematical model of the controller.