REPUBLIC OF IRAQ

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC

RESEARCH

AL-FURAT AL-AWSAT TECHNICAL UNIVERSITY

ENGINEERING TECHNICAL COLLEGE- NAJAF

ADAPTIVE GAIN SCHEDULING BASED CONTROLLER FOR NETWORK CONTROL SYSTEMS

A THESIS
SUBMITTED TO THE COMMUNICATION TECHNIQUES ENGINEERING DEPARTMENT
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE TECHNICAL MASTER DEGREE
BY

Rasha Shamit Salman

(B. Sc. Communication Techniques Engineering.2009)

Supervised by

Asst. Prof. Dr. **Ahmad Taha Abdulsadda**

September/2020

Acknowledgments

In the name of Allah, the Most Merciful, the Most Beneficent. I would like to thank Mr. Supervisor Dr. Ahmad Taha Abdulsadda for time ,attention he gave to me, for his effort, and guidance that helped me in every step to complete my work. Many thanks to my whole family for their support, and people close to me, especially thanks to my husband Qais for his great helping.

Supervisor Certification

I certify that this thesis titled **“Adaptive Gain Scheduling Based Controller for Network Control Systems"** submitted by **Rasha Shamit Salman** prepared under our supervision at the Communication Techniques Engineering Department, Engineering Technical College-Najaf, AL-Furat Al-Awsat Technical University, as partial fulfillment of the requirements for the degree of Master of Technical in Communication Engineering.

Signature:

Name: **Asst. Prof. Dr. Ahmad T. Abdulsadda**

(Supervisor)

Date: / / 2020

In view of the available recommendation, I forward this thesis for debate by the examining committee.

Signature:

Name: **Dr. Salim Muhsen Wadi**

(Head of comm. Tech. Eng. Dept.)

Date: / / 2020

Committee Report

We certify that we have read this thesis titled **"Adaptive Gain Scheduling Based Controller for Network Control Systems "** which is being submitted by **Rasha Shamit Salman** and as Examining Committee, examined the student in its contents. In our opinion, the thesis is adequate for the award of the degree of Master of Technical in Communication Engineering.

Signature:

Name: Asst. Prof. Dr. Mohammed Nasser Hussain AL-Turfi

(Chairman)

Date: / / 2020

Signature:

Name: Asst. Prof. Dr. Wafaa M. R.Shakir

(Member)

Date: / / 2020

Signature:

Name: Lecturer. Dr. Alyaa A. Al-Joda

(Member)

Date: / / 2020

Signature:

Name: Asst. Prof. Dr. Ahmad T. Abdulsadda

(Supervisor)

Date: / / 2020

Approval of the Engineering Technical College- Najaf

Signature:

Name: Asst. Prof. Dr. Hassanain G. Hameed

(Dean of Engineering Technical College- Najaf)

Date: / / 2020

Abstract

Wireless Network Control Systems (WNCS) offers many advantages, including flexibility, ease of movement, and maintenance. Which leads to The wireless network certified transfer the data correctly between the controller, actuator, and sensor nodes.

In this thesis, standard ZigBee wireless protocol has been used as a communication medium between the NCS nodes. Three controllers have been designed and simulated in this thesis: proportional-integral-derivative PID controller and the Fuzzy PID controller and the Particle Swarm Optimization algorithm (PSO) with the Fuzzy PID controller (PSO Fuzzy PID) to determine the stability of the stepper motor speed wirelessly via the ZigBee network. The controllers are tested with different load cases to measure the effect of time delay on the system when the number of nodes increases. The stepper motor is taken as a case to test system performance. The response of the stepper motor is compared with the three proposed controllers. The most problems of WNCS are time delays and handling the largest number of nodes and packet loss that will test in this work proposal. MATLAB 2018 is used with TrueTime simulation tools in WNCS to apply in this work proposal. The simulation results illustrate the PSO Fuzzy PID controller is more robust compared to other controllers with the choice of sampling time is 0.08 seconds and the sampling time in the interference transmitter node is one second. The BWshare 0.4 of PSO Fuzzy PID controller can handle 500 nodes at medium load, while the BWshare 0.9 of PSO Fuzzy PID controller could handle approximately 300 nodes at a high load. However, the system keeps going stable. The PSO optimization algorithm is used to find the optimal fuzzy rules. The Fuzzy logic has been tuning adaptive gain parameters for the PID controller. Also, the packet loss is tested for the PSO Fuzzy PID controller. It is noted that when the packet loss is 10% and 20% in the network, the system remains stable. While the packet loss is 40% in the network, the stability of the system would be affected slightly by 25%.

Table of Contents

Table of Symbols

Symbol	Definition	Unit
CE	Change Error	
C1	The Individual Coefficient	----
C2	The Social Coefficient	----
e(t)	System Error	Volt
G (t)	The Best Solution Obtained by all A Particle Entire Swarm at t th	-----
G(s)	Plant or System Transfer Function	---
I (t)	The Current in The Coil	Amp
J	The Inertia of The Rotor and The Load	----
Kd	Derivative Gain	---
Ki	Integral Gain	---
Km	The Motor Constant	---
Kp	Proportional Gain	----
L	Inductance	H
M	The Number of Stator Phases	
Ms	Fuzzy Logic Membership Functions	-----
N	The Number of Rotor Pole Pairs	
Pi (t)	The Best Solution Acquired by i th Particle at t th iteration	-----
R	Resistance	Ω

r(t)	Input Signal	Volt
r1 and r2	The Random Numbers	-----
S	The Number of Steps	
TF	Frictional Load Torque	N-M
Ts	Sampling Period	Sec
τ_{ca}	The Delay from The Controller to The Actuator	Sec
τ_{sc}	The Delay from The Sensor to The Controller	Sec
τ	The Control Delay	Sec
u (t)	The Output Signal of The PID Controller	Volt
U	A Function of The Supplied Voltage	V
Vi (t)	The Velocity of PSO	-----
w	The Inertia Weight	-----
y(t)	The Output of The System or Plant	----
Φ	The Stepping Angle	Rad
Φ0j	The Location of The Coil j in The Stator	Rad
Xi (t)	The Position of PSO	-----
$X\hat{}$	The Crisp Output	-----
Xi	FL Outputs	-----

Table of Abbreviations

Symbol	Description
AP	Access point
ACO	Ant Colony Optimization
BLER	Block Error Rate
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
CSMA/CA	Carrier Sense Multiple Accesses with Collision Avoidance Mechanism
CPU	Central Processing Unit
CDMA	Code Division Multiple Access
ESS	Error Steady State
FDMA	Frequency Division Multiple Access
FIS	Fuzzy Inference System
FLC	Fuzzy logic Controller
GA	Genetic Algorithm
IEEE	Institute of Electrical and Electronics Engineers
ISE	Integral Square Error
MAC	Media Access Control Layer
MF	Membership function
NB	Negative Big
NM	Negative Medium
Z	Zero
PM	Positive Medium
PB	Positive Big
NCS	Network Control System

PSO	Particle Swarm Optimization
PHY	physical Layer
PID	Proportional-Integral- Derivative
RTOS	Real-Time Operating System
TR	Rise time
TS	Settling time
MP	Max Overshoot
RTT	Round Trip Time
SNR	Signal-to-Noise Ratio
SISO	Single Input and Single Output
TDMA	Time Division Multiple Access
TCP	Transmission Control Protocol
UoD	Universe of Discourse
WI-FI	Wireless Fidelity
WLAN	Wireless Local Area Network
ZOH	Zero-Order-Hold

Table of Figures

Table of Tables

Table of Publication

[1] Rasha S. Salman, and Ahmad T. Abdulsadda. "Adaptive Gain Scheduling Based controller for Networking Control Systems" Al-Furat Journal of Innovations in Electronics and Computer Engineering, 1. doi:10.46649/110420-01, Aprial,2020.

[2] Rasha S. Salman,, and Ahmad T. Abdulsadda. "Network Scheduling By using Expert Nonlinear Controller." Test Engineering and Management Vol.83, No.1708 .1716,pp: 0193-4120 May ,June (2020).

Chapter One

Introduction and Literature Review

1.1 Background

The development of control and networks technology rises to the network control system (NCS). The NCS is a combination of control, communications, and computer technology. It has received much attention in recent years. NCS is a spatially distributed feedback control system. It consists of controllers, sensors, and actuators, and system components that are connected using a network. Figure 1.1 shows the typical structure of the NCS [1].

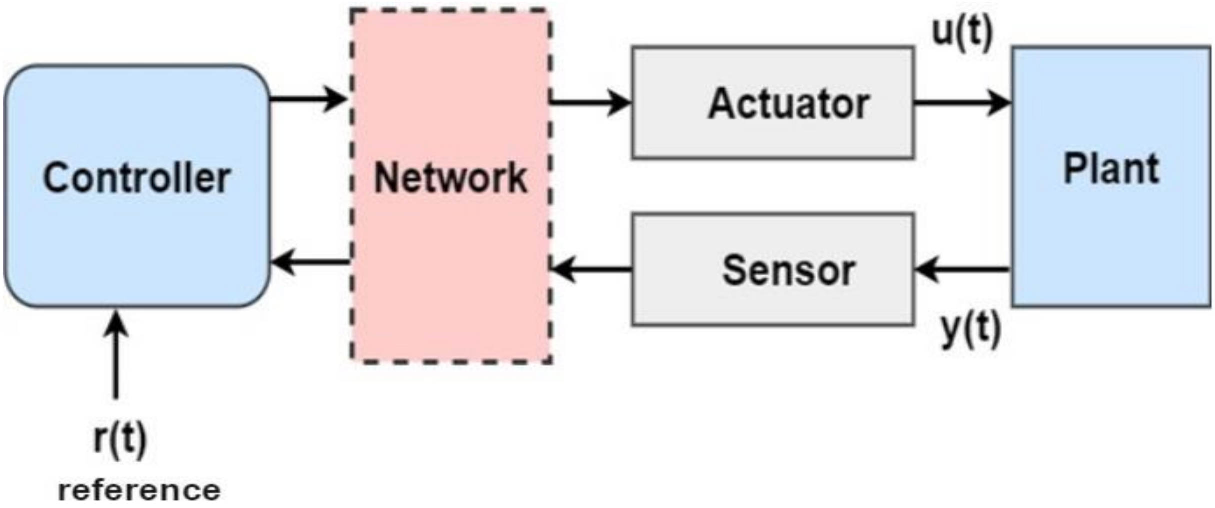


Fig 1.1: The typical structure of the NCS [1].

In figure 1.1 the input reference is r (t), the system response output is y (t), and the control signal is u (t) [1].

Network control systems suffer from many problems, including limitations of flexibility and difficulty in mobility. The wireless networks are replaced instead of chaotic wired cables, which led to the appearance of wireless network control systems (WNCS). Wireless technology in network control systems aims to reduce of the need for cables, maintenance, and obstacle removal, making it easy for cable-less sites to reduce costs, deployment, maintenance, and great flexibility [2].

Although the great advantages that WNCS provides in terms of applications and manufacturing, however, it suffers from some problems. One of these problems is the lack of appropriate modeling of network behavior. The wireless network suffers from [3]: random delays ,packet loss due to the harsh nature of the wireless channel,interference from other wireless devices,limited bandwidth and packet dropouts that can produce serious errors, and that these errors in control systems are considered unfavorable [3]. Some advantages and disadvantages of wireless networks are shown below:

A-Advantages of Wireless Network

The main advantages of WNCS are listed as follows:

1- **Reduced Costs**

In projects automation installation and cables are run at a very high rate of the total system cost. Consequently, the use of wireless installation removes the cost of installation and wiring, when necessary the shutdown mode for the sensing nodes is provided, a large amount of energy will be saved compared to wired devices and therefore requires a power source permanent.

2- **Easy Maintenance**

After completing the installation of a wireless device, the control engineers deal with various problems in the maintenance of wires such as water in the conduit, corrosion, freezing, burning cables, wild animal damage. In addition to the unexpected power cut, so it is necessary to change the battery after a long time of operation.

3- Wiring restrictions

Without the use of wire restrictions, devices can be used within applications that are difficult to access physically or costly. Thus, the wireless transmission reduces the complexity of controllers. Moreover, industrial process systems become very flexible and scalable, that is,easily reorganized for moving devices without hard work to install new cables and remove old cables [28].

B- Disadvantages of Wireless Network

The issues that affect the wireless network are explained below:

1- It can consider air to be a common medium, and therefore it is necessary to consider interference from other terminals and taken in to account the simulation process.
2- During the simulation of wireless networks, the path loss must be taken into consideration. The signal power in the receiver nodes is always much lower than the sender node. And the wireless devices do not have a protected medium and cause the radio signal strength to spread in some directions or all. Obstacles such as buildings, trees, etc. reduce the signal power level of the receiver.
3- In the case that the receiver node and the sender node moving, over time it causes the channel properties to change. Sometimes the receiving signals may have various phases and because of the multipath propagation as they cancel each other, which leads to a clear change in the energy level even if the nodes are moved to a very small distance. It is called short-term fading. The long-term fading is in the fact the signal fading with distance.
4- Must be taken into consideration large obstacles, resulting in the presence of the attenuation or so-called blocking or shadowing, where

made it impossible to receive radio signals. Sometimes the signals may be reflected in mountains and buildings. As this reflection makes wireless signals it is possible to take many various paths from the sender node to the receiver node.[29].

Recently, wireless network control systems have attracted many researchers as it has become an important topic for research and development. WNCS is used in several applications, including home automation and monitoring and industrial automation because it contributes to easy maintenance and provides flexibility, also to other applications such as domestic robots, remote monitoring for systems, and in nursing homes or hospitals [4].

1.2 Wireless Network Technology

Due to the fast development in the field of communication technologies represented using IEEE 802.11 wireless standard, and Ethernet wired network in the modern NCS design, however, the wireless provides more bandwidth, lower cost, and better scalability compared to NCS wired networks[5]. A wireless network control system is very commonly used in industrial applications [5]. The types of wireless networks are explained below:

1. The wireless (Wi-Fi) accuracy contains IEEE 802.1l a / b / g standards for wireless local area networks (WLANs). Wi-Fi networks are designed to enable users to surf the internet at broad speeds using mobile wireless devices and via an access point (*AP*), or ad hoc mode. The IEEE 802.11 Wi-Fi standard covers both the physical layer (*PHY*) and Media Access Control (MAC)[6].
2. The ZigBee specifications are designed according to the IEEE 802.15.4 ZigBee standard as the IEEEE 802.15.4. The IEEE 802.15.4 is responsible for the PHY and MAC layers and ZigBee covers the top layers of the protocol stack as it is intended for low-power,

low-cost, and low-productivity applications so it is simpler than other protocols such as a protocol Wi-Fi.

ZigBee provides three types of network topology, and star topology and also supports the most complex tree and mesh topology. ZigBee offers many applications such as automatic meter reading, home automation and control, personal health care, hospitals, institutional and cable replacement, and communication services [7].

1.3 The Controllers

The choice of the WNCS design control unit is an important part. The goal of choosing an appropriate control system for a closed control system, to obtain suitable system performance, where most problems are addressed and information are identified in the contract or network for the control system, to ensure system stability[8]. The principle of modern control theories mainly depends on feedback, where feedback is used in closed-loop systems. The term "closed-loop" means that the information path in the system process, where the inputs have an impact on the process outputs are calculated or measured through sensors processed in the controller, the result (the control signal) is used as an input to a process in the closed-loop. All operations and algorithms used in the controller, perform tasks in the real-time operating system (RTOS), which is executed through the CPU where so many benefits are available such as start time, priority, period time, and dead time [9],[10].

The Proportional, Integral, Derivative controller (PID) is widely used in industrial processes systems; the PID controller is characterized by simplicity and effectiveness in control. PID controller is widely used in many systems such as temperature, cars, and electric motors as well as the level and flow systems.

The design of the control device PID is easy to implement as it only needs to set the controller constant gain parameters, which can be done automatically.

There are several ways to set the traditional PID controller are Cohen, Conn, Ziegler, and Nichols, in addition to manual and automatic tuning [11].

Fuzzy logic uses a control method that is based on human knowledge in nonlinear systems. Fuzzy logic can deal with nonlinear and complex systems.

The concept of fuzzy logic was introduced by Zadeh at the beginning, and then many studies theoretical and practical, studies have demonstrated strict fog performance. The main difference between conventional control and fuzzy control is that conventional control needs a mathematical model of the process and also the controllers are designed for the model, while the principle of fuzzy control depends on human experience and inference (in terms of fuzzy rules IF-THEN)[12].

The Particle Swarm Optimization algorithm (PSO) provides random research in the relevant field for optimal solutions. PSO technology was introduced by Kennedy and Eberhart in 1995. The PSO technique is mainly inspired by the social behaviors of animals such as fish schools and flocks of birds, and bees [13].

1.4 Literature Survey

Wireless network control systems have become more interesting for many researchers in recent years. New control strategies are being developed to solve most of the WNCS problems by compensating packet loss, and time delay caused by the network. The research papers 23, 24, and 25 are considered the closest to this work. As below some of the published related works are summarized for WNCS:

C.Peng, et al .in 2010[14]: The authors proposed a new methodology for discrete-time models, in which the NCS is studied with various problems experienced by remote control systems, including the network-induced delay effect and packet dropouts. The goal of the integrated design of NCS control and control networks is to ensure stability, which was established by relying on the lower and upper bounds of the network time delay. The stability analysis is applied to the output feedback of the network control system.

A. Cuenca, et al .in 2010[15]: The delay-based gain scheduling law is used in the design of a controller capable of dealing with various load conditions using an Ethernet network. The loading conditions cause variable time delays between control and measurements. The design is simulated using TrueTime tools. The network setup is taken in two aspects: the remote side includes the controller, and the local side includes the plant and sensor and actuator, which the network nodes are shared with variable and random payloads, an event-dependent dual-rate PID controller is used to solve the network traffic problem and to improve control performance, and also avoid most synchronization problems.

H .Zhang, et al .in 2011[16]: The network control system (NCS) is designed to solve losing random data problems in each of the communication components from the controller to the actuator, and sampler to the controller and times delays occur in a network. The digital PID controller methodology was used as a synthesis problem for the static output feedback SOF control in designing systems for remote control. Moreover, the proposed model is applied to a DC motor connected to it and simulated by the network.

M. Urban, et al. in 2011[17]: The authors focused on the methodology of simulating a network control system using TrueTime function tools. TrueTime tools are used in MATLAB as an easy and simple method that provides many types of networks. The impact of two types of different networks are tested: the first type is the local network standard CAN and the second type is the wireless standard ZigBee network, the DC motor factory process control process is also simulated after setting the system basic parameters.Also, telecommunications standards are compared and their work with the network control system is compared.

J. Wang, et al. in 2012[18]: The authors suggested a method for designing a discrete-time switching system with a time-varying delay model, which is used to describe the WNCS wireless network control system, also providing stabilize closed-loop system installation status. The designed controller relies on both time delay and packet loss while providing a sufficient condition for the system to remain stable, given an explanation of the numerical example of the proposed design effectiveness.

Z.Song, et al. in 2013[19]: The authors used the method of designing the wireless network control system by using the sensor and the actuator to obtain a new model design is WSANCS. A comparison is made between the two controllers for the proposed design, which is the fuzzy PID controller algorithm and the PID control algorithm, and the work presents a WSANCS simulation model using MATLAB-based TrueTime simulation tools. The WSANCS model simulation is performed using a ZigBee network for communication between nodes. Besides, the plant (DC servo) can be controlled wirelessly. The results show that the fuzzy PID controller algorithm has performed well compared to the PID controller algorithm.

D. Peng, et al. in 2014[20]: The network control system is designed with the use of a fuzzy adaptive PID controller. The fuzzy rules are combined with the network control system PID controller, the plant is controlled across the network, and where the plant is represented by the second-order typical inertia. The communication between the components of the control system is made using an Ethernet network. A study is presented by simulating transmission rates and variable parameters in addition to increasing network delay.

D. Shah and A. J. Mehta.in 2014[21]: The proposed discrete PI controller is used in the design of the network control system. In order to improve performance, strengthen, and system stability, also various delays are analyzed. The control law is introduced to recompense for the network-induced delay for a networked control system. The network is simulated using the True Time function for you to demonstrate the effectiveness of the controller and the use of Ethernet as a connection network therefore the position of the DC motor is controlled.

N. Aung , et al.in 2016 [22]: The authors presented a study of the problems of designing the wireless network control system depends on the TrueTime simulation in MATLAB. The actuators and wireless sensors are also used in the design of WNCS, with the application of the PD controller. In addition, the communication and simulation network is discussed, focuses on the interactions between the control system and the communication network. Therefore, stability and time delays of the system are analyzed.

M. **Salman, et al.in 2017 [23]:** The authors suggested the wireless network is used in the design of the network control system (NCS). The speed of the DC motor is controlled by the standard of the ZigBee network. The system is tested with most of the problems that occur in the wireless network control system, especially the problems of packet loss, traffic load and sampling period, that relate to the network data rate, two controllers have compared are the classic PID control and the Sliding Mode Control (SMC). Different scenarios of the proposed ZigBee NCS model are simulated using the TrueTime function in MATLAB. The ZigBee network parameters are used to test the DC motor response, in which PID and SMC controller have tested with three cases of a data rate of ZigBee network are 40 kbps, 20 kbps, and 250 kbps.

I. Laith , et al in 2019 [24]: The authors suggest designing a controller consisting of a conventional Proportional-Integral-Derivative (PID), Fuzzy logic (FL), and gain scheduling (GS) in the design of the network control model NCS. TrueTime tools are used to simulate the Ethernet network used for communication between system nodes. The F-PID-GS controller design is tested with different loading conditions as well as the loss of random packets. A second-order stepper motor is used to represent the plant.

R. Nasser, et al .in 2019[25]: The Particle Swarm Optimization technology (PSO) is used in the design of two controllers, for the WNCS model. The PSO algorithm is applied to the classic PID and Fractional Order Proportional Integral Derivative (FOPID) controller. The Wi-Fi wireless standard is used as a network between the controller and the plant, the stepper motor is used at the plant side. The WNCS model is built using MATLAB and TrueTime simulator. The WNCS model is tested with three cases for the sampling period.

In this thesis, the PSO optimization technique is used to obtain the optimal rules for fuzzy logic that is used to tuning the proposed PID controller parameters with the wireless network control system model. MATLAB R2018b and True Time tools are used to simulate the ZigBee network to control the speed of the stepper motor wirelessly. The results obtained showed that the proposed PSO Fuzzy PID controller has more robust compare to PID and Fuzzy PID controllers.

1.5 Problem Definition

Network control systems suffer from many problems such as:

1- Sampling period
2- Time delays
3- Packet dropouts

Figure 1.2 shows most of the problems facing a WNCS system. Therefore, to overcome these problems and reduce the effects of time delay usually occurring between the sensor, actuator nodes, and the controller node. Thus, it would improve the output response of the stepper motor and reduce the possibility of packet loss, as the system can remain stable with system scalability in case of network load.

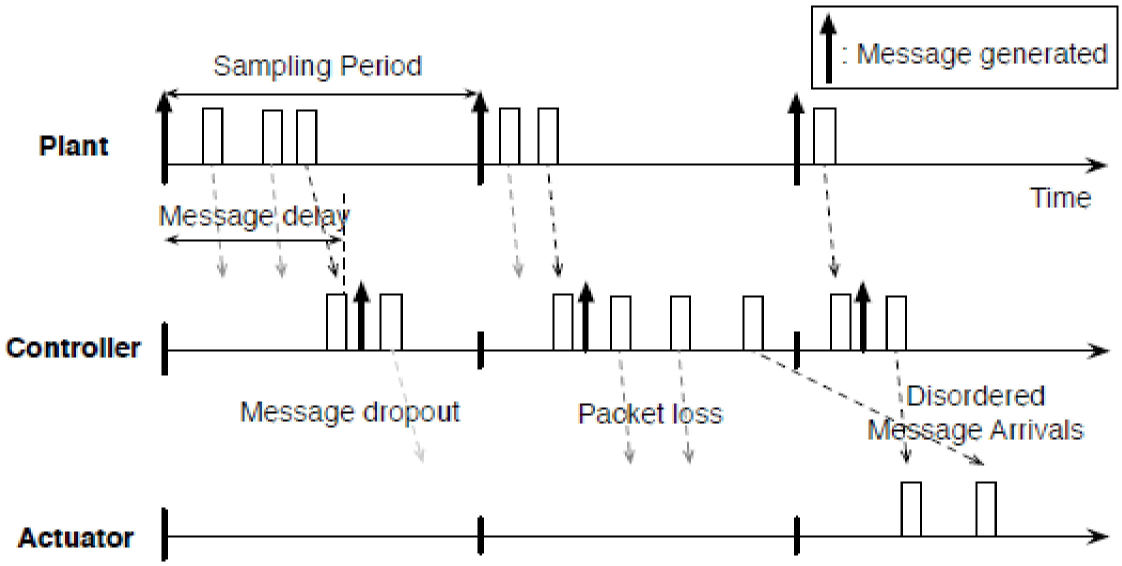


Fig 1.2: The problems facing a WNCS system

1.6 Aim of the Thesis

This work aims to design a robust controller for the wireless network control system that can deal with a large number of nodes, time delays, packet losses, and sampling periods so that the system response remains stable. The following objectives should be achieved:

1- Design the NCS model with a ZigBee wireless network
2- Simulate the ZigBee network on MATLAB based on TrueTime tools
3- Implement a Fuzzy adaptive PID controller.
4- Use the particle swarm optimization algorithm (PSO)to find the optimal rules for a Fuzzy PID controller.
5- Evaluation of Round Trip Time (RTT) to calculated time delays for each packet.
6- Finding the possible limit of packet loss while using the ZigBee network without affecting the stability of the system.

7- The performance test of the WNCS proposed in case of increasing the number of the nodes.

1.7 Outline of the Thesis

This thesis is divided into five chapters, and a detailed description for each chapter is as follows**:**

- **Chapter two:** Brief explanation of a typical structure of the wireless network control system WNCS, As the PID controller, Fuzzy logic system, and the PSO optimization algorithm are shown, the wireless standard ZigBee and the true time tools used in the WNCS model are discussed.

- **Chapter three:** This chapter shows the design of the proposed WNCS model that is used in this work, and the study of different network cases using the proposed fuzzy PID controller is clarified, as well as the implementation of the PSO optimization technique with the proposed controller.

- **Chapter four:** This chapter presents the WNCS simulation method proposed in this work with various cases of load on the wireless network as well as for the analysis of time delay for each case. Also, the proposed PSO Fuzzy PID controller is compared to the fuzzy PID and PID controllers. These controllers are tested for system response in the case of an increased number of nodes, increased network load, and packet loss.

- **Chapter five:** The results extract is clarified, and drawing on the results of the simulation of the proposed model, also provides a proposal for some educational points for future work.

Chapter Two

ZigBee Network Control System

2.1 Introduction

In this chapter, an overview of the structure of WNCS is provided. Major issues such as packet loss, sampling period, and time delay are illustrated. ZigBee Network, PID Controller, Fuzzy Logic, and PSO algorithm are explained. A brief description of the TrueTime tools used to design the WNCS model is given.

2.2 WNCS Structure

The presence of wireless network technology along with control systems has led to the emergence of the WNCS. A wireless network control system has been a growing focus of attention due to their flexible structure that reduced maintenance and installation costs.

The wireless network control system is a spatially closed- loop in which nodes are connected wirelessly using a network of communication between controllers, actuators, and sensors. The controller node works as an event-driven implemented using mathematical algorithms, also works to calculate the control signal immediately across the network when the sensor data is reached. The actuator is based an event-driven also depending on the data received over the network to actuate the plant. The sensor nodes are clock-driven, which periodically takes samples during the sampling period Ts. A continuous-time signal is sent to the controller over a network [26].

The process in it is physically controlled by the computer via a digital to analog conversion (*D / A*), so it is kept constant and during the sampling period, so the Zero-Order-Hold (*ZOH*) component is used enter the system [27].

2. 3 Main Challenges of WNCS

The most important challenges facing the work of WNCS are sampling period, time delay, and, packet dropouts, it is considered the most important challenges found in control networks, whether wireless or wired. According to these challenges, the NCS and WNCS recently have faced many problems in terms of failure and errors compared to other control systems, these problems are overcome and reduced, as various methods are used over the years. One of these methods is the time stamp for data packets so that the control system uses the new data despite the time delay. Despite these various solutions presented to be overcoming important problems, however, research must be done on various aspects of performing networks [30].

2. 3.1 Sampling Period

The first challenge is the sampling period. The sensor in the NCS takes the output or processing values with periodic times, to send continuous-time across the network used in the system, then samples are taken to be encoded in digital format to send over the network. On the other hand, this coding is decoded in the receiver. Despite the increase in the number of samples, it will provide better control of the system. It must be taken into consideration that the increase in the number of samples may negatively affect the NCS system due to the increased system load, consequently, network delays occur, this leads to degraded system performance [26].

2. 3.2Packet Dropout

Another problem that occurs in wireless networks more than wired networks is packet dropout. The Packet dropout results in the event of transmission error at the actual connections of the wired network or it may lose, delay, or reach these packets, but they arrive out of order. The Packet dropouts may be taken into as a particular case of an infinite time delay [31].

2. 3.3 Time Delay

The stability is an important case in the design of NCS. The main reason for the lack of stability of control systems is the time delay. Stability analysis is a very complex process in the presence of time delay. Therefore, its induced delay impact must be taken into account. These time delays depend on several factors including bandwidth and type network, a protocol used, and packet size. The control loop diagram with feedback and time delay caused by the network is called the Single Input and Single Output (SISO) system [32]. When data is transferred between the remote system and the controller. These network delays can be classified in the network control system as well as time delays depending on the direction of data transmissions such a controller to the actuator τ_{ca} and a sensor to a controller τ_{sc} . Both the network delay is collected with the controller processing delay τ_c as the control delay (τ) or Round Trip Time (RTT) and for a facility of analysis is given as [7], [33]:

$$\tau = \tau_{ca} + \tau_{sc} + \tau_c \tag{2.1}$$

This approach has been used in most network control methodologies[33]. Although there is always a delay in controller processing τ_c however, this delay is usually small compared to the network delays τ_{ca} and τ_{sc} and therefore can be neglected, the τ is given approximately by [7], [33]:

$$\tau = \tau_{ca} + \tau_{sc} \tag{2.2}$$

The control system setup with τ_{ca} and τ_{sc} delays are shown in Fig 2.1.

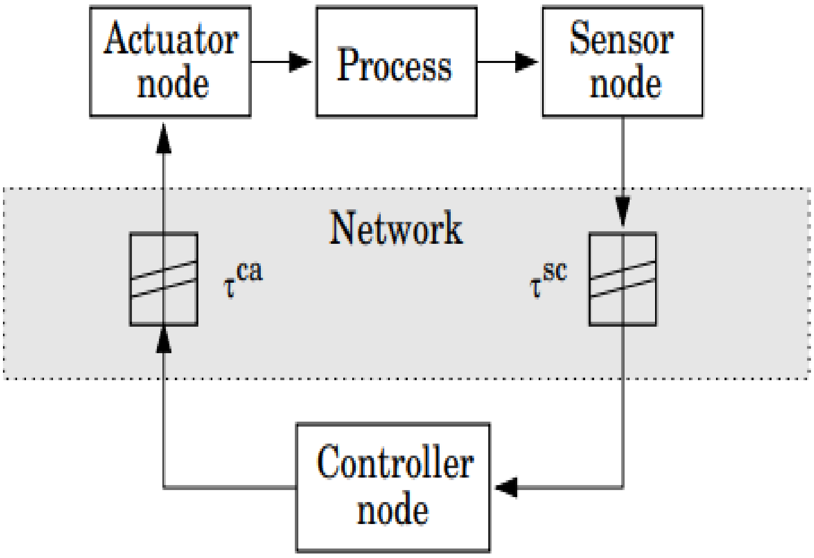


Fig 2.1: The control system setup with τ_{ca} and τ_{sc} delays [7].

There are two types of time delays when talking about length delays in NCS, and they are long time delay and short time delay [33]. At a long time delay the total time delay between the controller/actuator and the sensor/controller is greater than the sampling period $\tau > T_s$. For the time delay Short [7],[33], the total time delay is in which the total time delay between the controller/actuator and the sensor /controller is less than the sampling period (T_s), $\tau < T_s$. In Fig 2.2 the long time delay is represented by orange lines and the short time delay is represented by a green line.

It may require retransmission in the TCP network protocol when an error occurs in a packet or the router, so the packet drops in this case, and therefore the retransmission may affect the delay of the extension [34].

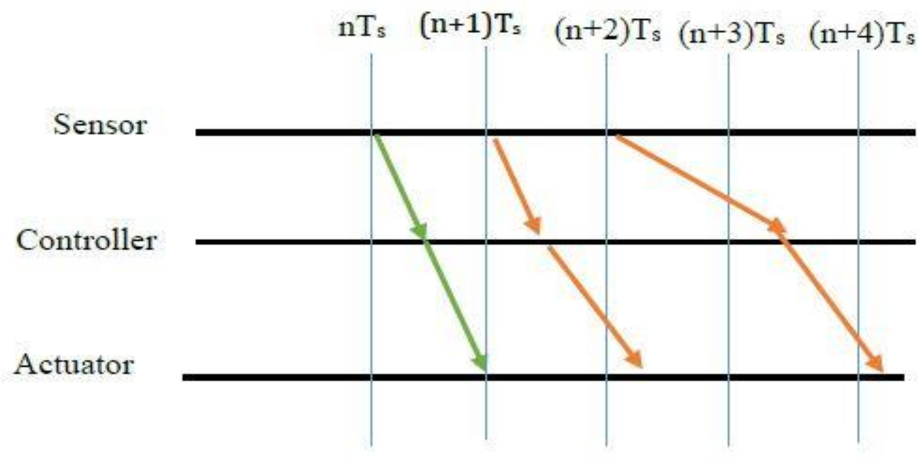


Fig 2.2: long Time Delay and Short Time Delay [7]

2.3 ZigBee Technology

The ZigBee network is designed by the ZigBee alliance and standardized according to the IEEE 802.15.4 specifications, which are designed for the upper layers (network layers, security, and application). It works with a three-frequency band with twenty-seven channels. The ZigBee designed to consume less energy than the rest of the wireless networks such *WI-MAX*, *WI-FI*, and Bluetooth, ZigBee provides a maximum data rate of 250 Kbps and within a range of 10 to 100 meters. ZigBee network is used carrier sense multiple accesses with collision avoidance mechanism (*CSMA-CA*) to improve the possibility of successful data transmission, the maximum packet size in the ZigBee protocol is 133 bytes [6],[35].

ZigBee provides many applications, including remote control, building automation, security systems, remote meter readings, computer accessories, and other applications. The ZigBee protocol is developed to provide the following functions [7]:

1- Low cost to install and maintain.
2- ZigBee device battery does not need a charger for one year.

3- An about 70m indoors and 400m outdoors are range ZigBee wireless network.

The ZigBee structure consists of three layers which are the top layer (the network layer, the application layer, media access control layer (*MAC*), and the physical layer as shown in Fig 2.3.

1- **Physical layer:** In IEEE 802.15.4, the physical layer (hardware) dealing with the reception and transmission of data is defined as shown in table 2.1 that shows the frequency band used and specified for the physical layer. The highest frequency band is 2.4 GHz, which is a global frequency band that uses 11 channels [35].

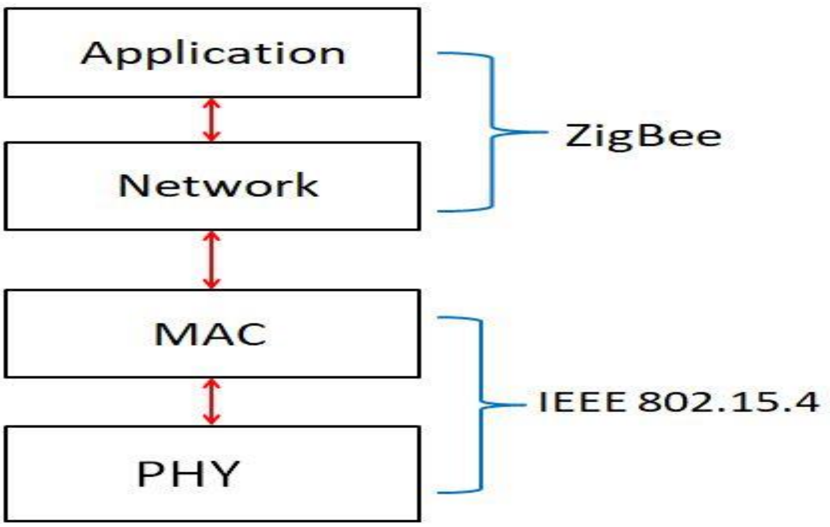


Fig 2.3: the architecture of ZigBee technology [6]

2- **MAC layer:** This layer is used to share a required medium, to transmit data, then to communicate. In broadcasting when the devices send information (data) at the same time. A collision problem may occur. Therefore Code Division Multiple Access (*CDMA*), Time Division Multiple Access (*TDMA*), and Frequency Division Multiple Access (*FDMA*) methods are used to create a channel and avoid the collision problem.

Table-2.1: The frequency band used in the physical layer [35]

Frequency Range	Coverage	Data Rate (kbps)	Channels
2.4 GHz	Worldwide	250	11-26
915 MHz	America	40	1-10
866 MHz	Europe	20	Not used

3- **Network Layer:** The primary functions of the network layer are routing, inter-networking, congestion control, and also handling end to end or point to point packet delivery.
4- **Applıcatıon Layer:** It is the last and important layer that acts as an intermediary between users and other layers. The most important basic functions that this layer performs are the synchronization of communication, determining the availability of resources, and identifying communication partners [35].

2.4 Controller Methodologies

Modern control methodologies and type of algorithm that used for a WNCS as follow:

2.4.1 Proportional Integral Derivative Controller (PID)

PID controller is widely used in many industrial applications. PID controller is one of the previous control strategies that were introduced at the beginning of the previous century. PID controller is used as a standard controller in industrial settings and applications because it is characterized by flexibility, reliability, stability, ease of operation, and simplicity.

The aim of using the PID controller in the closed-loop for control systems is to keep the output response as possible to the required response.

The PID controller works to error calculation and correction. The error represents the difference between the plant output and the set point. The block diagram of the PID controller is shown in Fig 2.4.

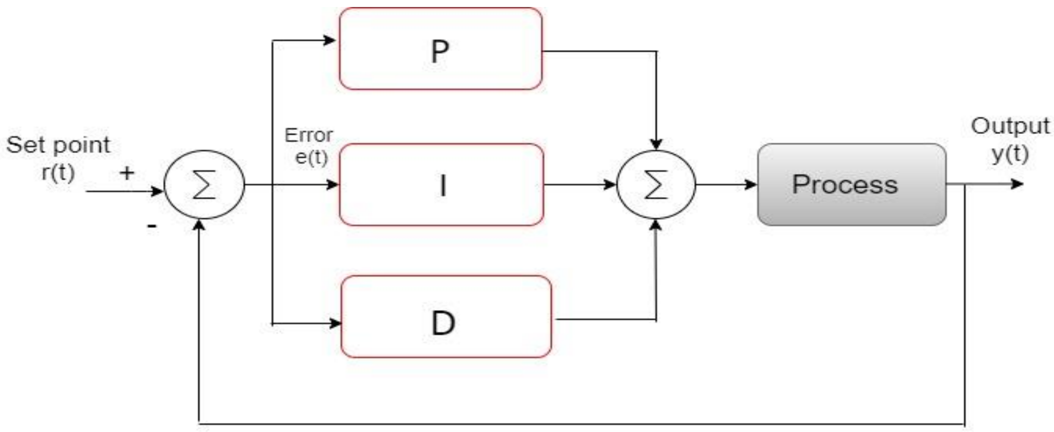


Fig 2.4: The block diagram of a PID controller

PID controller is used to controlling variables such as frequency or current in an electrical system, position, or velocity in mechanical systems or temperature in thermal systems [36]. PID controller consists of three separate constant parameters, proportional, integral, derivative values which denoted P, I, and D terms. The first part is proportional (P), which is dependent on the present error. The second part is integral (I), which is dependent on the accumulation of past errors worked when it is desired controller corrects for any fixed offset from a steady reference point value. The integral part beats the fault of the proportional control through eliminating offset without the use of extremely large controller gain. The derivative part (D) is dependent on a prediction of future errors, and it is based on the current rate of change [37]. The output signal of the PID controller *u (t)* in continuous time is generally written in the "parallel form" which is given by equation 2.3:

$$u(t) = K_p \, e(t) + K_i \int_0^t e(t)dt + K_d \frac{d}{dt} e(t) \qquad (2.3)$$

Where the system error is $e(t)$ and equal: $e(t) = r(t) - y(t)$, and the controller output is $u(t)$, and kp represents a proportional gain, ki represents an integral gain, and kd represents a derivative gain. The reference setpoint is $r(t)$, and the process output is $y(t)$. In table 2.2 is a summary of the individual impacts of adjusting these three-term gain parameters on the PID controller [36].

Table-2.2: Impacts of *P, I, D* terms tuning [36]

System Response	Rise Time	Overshoot	Settling Time	Steady State Error	Stability
Increasing Kp	Decrease	Increase	Small Increase	Decrease	unstable
Increasing Ki	Small Decrease	Increase	Increase Large	Decrease	unstable
Increasing Kd	Small Decrease	Decrease	Decrease	Minor Change	stable

The following equation determines the rise time *TR*, maximum overshoot *MP*, and settling time *TS* of the second-order system and these values will be obtained in terms of damping ratio ζ and ω_n natural frequency [50].

1- Rise time,TR = $[\frac{\pi-\beta}{\omega d}]$, $\omega d = \omega_n\sqrt{1-\zeta^2}$

2- Maximum overshoot, *MP* = $[e^{\frac{-\zeta\pi}{\sqrt{1-\zeta 2}}}]$

3- Settling time,*TS*=$[\frac{4}{\zeta\omega_n}]$,or *TS*=$[\frac{3}{\zeta\omega_n}]$,

Where angle β is defined in Fig.2.5.Clearly, for a small value of *TR*, and ωd must be large [50].

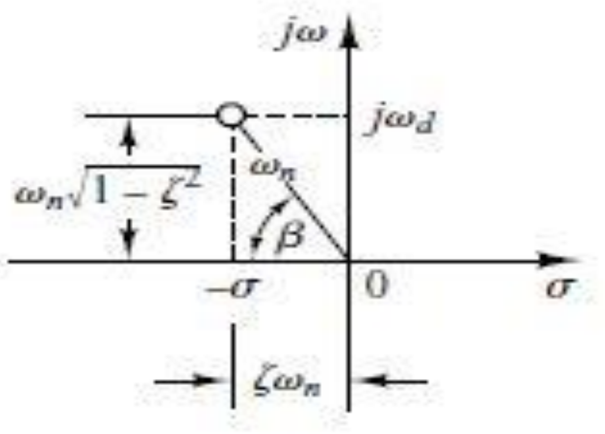


Fig.2.5: Definition of the angle β [50].

The PID output at equation (2.3) after taken Laplace Transformation (S-domain) to obtain an equation (2.4).

$$PID(s) = kp + ki\frac{1}{s} + kd\,S \tag{2.4}$$

Analog controllers are replaced by digital controllers where inputs and outputs are determined in discrete-time states. The digital control system is used as a heart in the controller, which is usually in the form of a programmable computer.

The continuous-time signals cannot be directly processed in digital computer devices that deal with discrete-time signals. In practice, a continuous-time quantity is converted identical sampled version which synchronizes with the authentic quantity at some certain in time. For example, the continuous-time signal samples can be taken as a sequence of numbers, each number is separated from the next number in time by sampling period Ts [27],[38]. The main goal of the transformation process is to decrease complex systems models such as differential equations to algebraic equations.

The Z-Transform supplies a mechanism to plan a discrete-time signal across a complex plane, which is also called a z-domain. Equation 2.4 has to be discretized. The discretization can be implemented in many methods with the application of Laplace Transform [39].

1- Backward Euler *Method* $S = \frac{1-Z^{-1}}{Ts}$

1. Forward Euler Method $S = \frac{1-Z^{-1}}{Ts.Z^{-1}}$
2. Trapezoidal Method $S = \frac{2}{Ts}\frac{1-Z^{-1}}{1+Z^{-1}}$

Where Ts is the sampling time of the analog to digital (A/D)

The PID controller suffers from some of the problems including undesirable overshoot and sensitivity in controlling KI and KP gains and slow operation, and where its performance depends on the parameters of the control system and model accuracy and these problems are solved using some section of artificial intelligence (AI) techniques or soft computing in controlling PID controllers such as Fuzzy Logic (FLC), particle swarm optimization (PSO), and Genetic Algorithm (GA) [36].

2.4.2 Fuzzy System

The complexity or uncertainty of physical systems or processes often leads to difficulty in developing accurate analytical models. However, humans and through their cognitive processes have been able to describe the behavior of these systems from a qualitative point of view, the expert may be able to control the outcome of the process and based on his knowledge and without any mathematical model.

Fuzzy set theory is "a set of concepts and techniques provide precision mathematical solutions to human cognitive processes that are inaccurate and vague by classical mathematics standards in many aspects". Fuzzy logic is used

Fuzzy set membership functions that have values ranging from 0 to 1, which allow for their capturing linguistic representations to knowledge. The inaccuracy of knowledge is dealt with using membership functions associated with linguistic variables. In other words, fuzzy logic is a logical system, which is a generalization of multi-valued logic. The fuzzy logic is proposed by Lotfi Zadeh in 1965, where the fuzzy theory was applied to many complex and unspecified engineering problems with mathematical or conventional analysis [40].

2.4.2.1 Fuzzy Sets and Fuzzy Operations and Membership Functions

The concept of the fuzzy set is an extension of the concept of the crisp or the classic set. The elements are represented ambiguously using a fuzzy set, and by taking elements from the Universe of Discourse (UoD), which represents the domain of range fuzzy sets, where each set in UoD is determined using the "Membership function ", which is often described by "linguistic variables".

The classical set has sharp boundaries, which means that it either belongs to the set or not, unlike the fuzzy set that allows the member to belong to a set to some extent [41].

As previously mentioned, the fuzzy sets are an extension of the classical sets. Therefore, the fuzzy set and the fuzzy control use the basic theoretical processes which are: union, intersection, complement. There are six forms of Membership function (MF) and are the following: Triangular MF, Gaussian MF, Trapezoidal MF, Sigmoidal MF, Z-shaped MF, and Generalized bell MF [42].

2.4.2.2 Structure of Fuzzy Logic Controller (FLC)

After clarifying the Membership functions and the fuzzy set of the fuzzy logic system, the fuzzy reasoning system process is applied. The main idea of the fuzzy inference system is to integrate human knowledge with a set of fuzzy as IF-THEN rules, in which include operations on linguistic variables where the general structure of the fuzzy inference system consists of four important elements are:

1- Fuzzification,
2- Fuzzy inference engine
3- Rule base
4- Defuzzification process.

The inference system for the Fuzzy Logic controller is generally shown in Fig 2.6. The numerical inputs in the fuzzification process are converted to variables that can be recognized for a fuzzy set, whatever the numerical inputs are determined by two or more functions using the MF that describe the degree of the Membership value within the range of [0, 1] [40],[41].

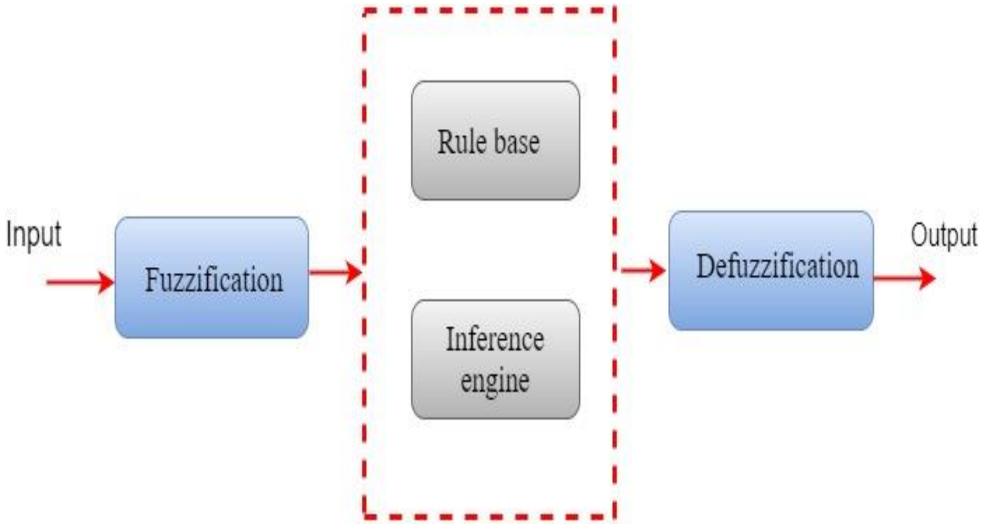


Fig 2.6: Typical structure of a Fuzzy Logic controller

The Fuzzy Inference System (FIS) is depending on a knowledge base, the fuzzy rule base contains a set of IF-THEN fuzzy rules that are established and extracted based on expert experience and knowledge. There are two methods of Fuzzy Inference System FIS are :

1- Mamdani Fuzzy Inference System: It was first introduced as a method to create a control system by synthesizing a set of linguistic control rules obtained from experienced human operators. In a Mamdani system, the output of each rule is a fuzzy set. Mamdani method was well-suited to expert system applications where the rules are created from human expert knowledge. Mamdani is a scheme that is widely used, it depends on the Min-Max function. The Maximum is used if there are the same linguistic variable and the minimum process for results or comparison of fuzzy numbers.

2- Takagi-Sugeno Fuzzy Model (TS Method): It uses singleton output membership functions that are either constant or a linear function of the input values. The defuzzification process for a Sugeno system is more computationally efficient compared to that of a Mamdani system since it uses a weighted average or weighted sum of a few data points rather than compute a centroid of a two-dimensional area

The Defuzzification process converts the fuzzy sets obtained from the inference mechanism to numerical values (crisp value), as the task of the defuzzifier is the process of determining the best point that represents the fuzzy set and is similar to the average value about the random variable.

The most commonly used defuzzification method is the center of gravity method (*CoG*), also commonly referred to as the centroid method. This method determines the center of the area of the fuzzy set and returns the corresponding crisp value. It is given by equation [41]:

$$\mathcal{X}^{\wedge} = \frac{\sum_{i=1}^{n} Xi\,\mathcal{M}s(Xi)}{\sum_{i=1}^{n} \mathcal{M}s(Xi)} \tag{2.5}$$

Where the crisp output value is indicated by ($\mathcal{X}^{\wedge}$),$\mathcal{M}s(Xi)$, and are the FL Membership functions and their outputs respectively, and the total number of output variables is indicated by n.

On the other hand, there is a lot of computer-based software tools designed to implement FL control systems. These tools provide the possibility to correct errors and also improve FL control systems easily and simply with an environment and graphical interface used which is characterized by ease of use examples of these tools are the Fuzzy logic, the functions available in the MATLAB program which provide an environment for designing, simulating, and analyzing FL control systems.

2.4.3 Particle Swarm Optimization (PSO)

The particle swarm optimization (PSO) algorithm is one of the powerful algorithms in the literature of random improvement approaches. The particle swarm optimization technique belongs to the family of swarm-based algorithms and it is a population-based technology as in the ant colony optimization algorithm (ACO) and genetic algorithms (GA), it is the most well-known in the literature compared to algorithms, where appear the advantages of the PSO algorithm in solving problems of different types. However different versions of the PSO algorithm are cited in the literature that is used to solve restricted, multimodal, multi-target, dynamic, and distinct problems based on the behavior birds in nature, the main inspiration was proposed in the form of simple equations, by Raynold in 1987. The concept of an algorithm for particle swarm optimization was developed by electrical engineer Eberhart and social psychologist Kennedy, in 1995, where the idea of computational intelligence was developed through the existing interactive natural systems by simulating the behavior of searching for biological swarms [43].

2.4.3.1 Mathematical model of PSO

In the PSO technique, each solution to a specific problem is considered a "particle" that can move within the search field. For the position of each "particle" to be updated, each particle has two vectors, which are the velocity vector and the position vector, the position vector determines the value of the problem parameters, the particle location can be considered the area of the dimensions “d”, where “d” represents the number of variables, also that the velocity vector determines the speed, intensity, and direction of movement in each step of improvement. The particle is updated position $\mathrm{Xi}(\mathrm{t}+1)$ by using the following equation [43], [44]:

$$Xi(t+1) = Xi\ (t) + Vi\ (t+1) \qquad (2.6)$$

Where Xi (t) indicates the position of i^{th} at t^{th} iteration and Vi (t) indicates the velocity of i^{th} at t^{th} iteration. The formula shows the position update with the velocity vector, which is the main component, where the velocity updates $Vi\ (t+1)$is defined by the following formula:

$$Vi\ (t+1) = \ w\, Vi\ (t) + \ C1\, r1\ \left(Pi\ (t) - Xi\ (t)\right) + \ C2\, r2\ \left(G - \ Xi\ (t)\right) \qquad (2.7)$$

Where

$Xi\ (t)$: indicates the position of $i^{th}\ at\ t^{th}\ iteration$

$Vi\ (t)$: indicates the velocity of $i^{th}\ at\ t^{th}\ iteration.$

W: signifies the inertia weight.

$C1$: denotes the individual coefficient.

$C2$: shows the social coefficient and usually equal to 2.

$r1;\ r2$: are represents a random number in [0; 1].

$Pi\ (t)$: signifies the best solution acquired by i^{th} particle at t^{th} iteration.

$G\ (t)$: denotes the best solution obtained by all a particle entire swarm at t^{th} iteration.

The PSO algorithm is easily implemented in most programming languages, so the algorithm can be written in one line of code, as it has proven fast, effective, and especially for its application to various groups of most optimization problems, PSO technique is useful in the process of improving parameters during multiple search spaces dimensional and persistent [13].

2.5 Simulation Tools of WNCS

The WNCS structure consists of the network system and control system, which are both taken into consideration during the design of the network control system or during the testing of the control algorithm, that the most frequently used simulation of control systems is MATLAB, where control systems are simulated by MATLAB through invoking control algorithms either by using functions MATLAB code (m file) or via Simulink models (as block diagrams).

The plant can also be modeled using the transfer function. Where, the MATLAB contains a comprehensive set of functions to evaluate the performance of the controller, analyze the system response, analyze the system response, and also MATLAB provides huge libraries to test and design different control algorithms, there are many network simulators available today including NS-2, OPNET, and OMNeT ++ which are the most famous wireless and wired network simulators [7].

2.5.1 TrueTime Toolbox

In order to design and simulate distributed and embedded control systems, a simulation platform must be provided, as simulation tools were developed based on MATLAB called the True Time tools [45] at Lund University in Sweden, in the Automatic Control Department in 1999. The major block in the True Time toolbox is the block library (version 2.0, 2016), as shown in Fig 2.7 This block library includes seven blocks which are kernel block, network blocks for a wired network, send block, receive block, battery block, wireless network block, and ultrasound network block. These True Time blocks are variable step, discrete, and MATLAB S-Functions, also it is written in C++ language [46].

On the other hand, communication networks are simulated in the design of the control system using the kernel block, the wireless network block or the wired block, where the kernel block is configured through the communicate code functions which are written in the m file or written with the C ++ language code, TrueTime kernel block is connected to other MATLAB blocks, through two ports, one for input by A/D and the other for output by D/A, whereas the actuator, sensor and other design components inside the blocks, and by organizing them into tasks, TrueTime kernel block similar to the structure of the RTOS kernel, so that tasks that wait for some time or waiting list are ready for ready tasks.

The TrueTime Kernel block supports three scheduling policies are Fixed-Priority scheduling "*prioFP*", or Earliest-Deadline-First scheduling "*prioEDF*", or Deadline-Monotonic scheduling "*prioDM*", where the Kernel block is linked to the local memory and in order to store the necessary data for each task. The task is also created with the following options [7],[46]:

1- The task may be periodic or aperiodic, so each task must be neutralized by the period, priority, and start time that are used by the kernel scheduling policy.

2- The task is connected to other MATLAB blocks through the *A /D* and *D / A* ports in the TrueTime Kernel block, or by sending and receiving the signal via the network.

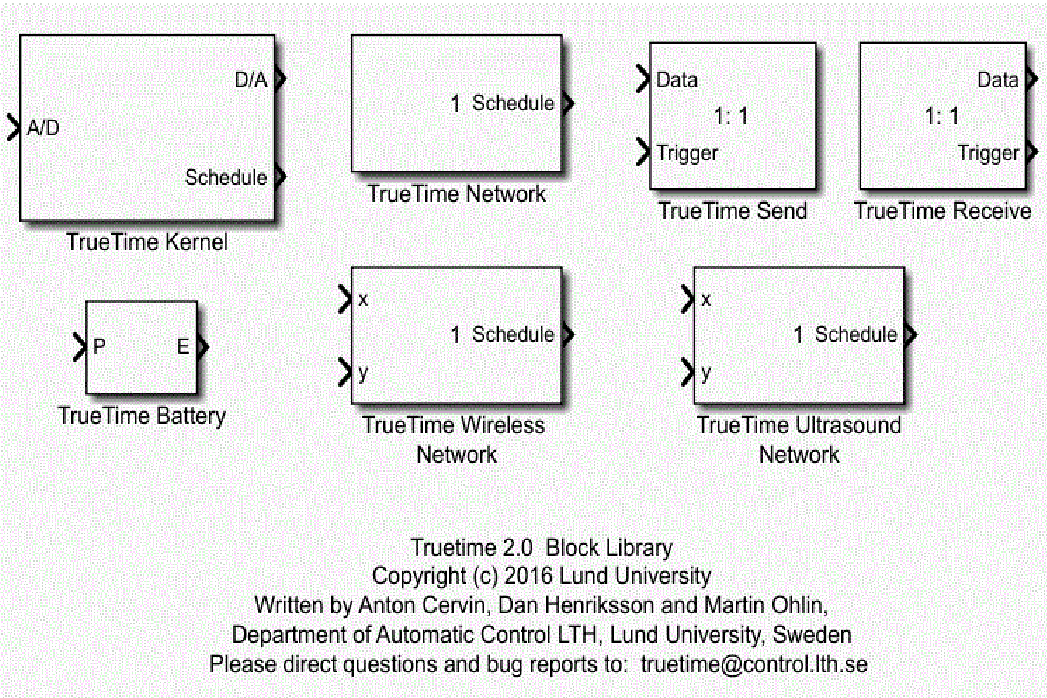


Fig 2.7: The True Time library

The wireless network block simulates the media access control layer and the physical layer of different local area networks in addition to the possibility of retransmission by TCP protocol, and packet loss. The packet sent over the network contains information about sending and receiving addresses as well as user data such as a control signal etc. Besides, the network block in TrueTime supports some network protocols such as CSMA/CD, Switched Ethernet, Round Robin, FDMA, Flex Ray, and TDMA. The wireless network block in TrueTime also supports only two protocols such as IEEE 802.11b / g (Wi-Fi) and IEEE 802.15.4 (ZigBee), since the wireless network block operates in the same way as a network block, but takes into account the location of the nodes by specifying the location of the x and y inputs [25], [29]. Thus it has become important to use a powerful common simulation tool that can efficiently model and simulate communication, calculation, and control aspects together. Therefore, to create a wireless network control system model with TrueTime tools, the following steps are followed [25], [46]:

1- TrueTime simulation tools are used with MATLAB / Simulink to create the WNCS model
2- All nodes like controller and sensor are designed using TrueTime Kernel block
3- The initial values are set for each node and parameters are determined using the m-file functions.
4- Choosing the type of network by using the wireless network block and specifying the appropriate parameters such as data rate, minimum frame, and possible packet loss, number of nodes, transmit power, etc.

2.6 Stepper Motor

The stepper motor is an electromechanical device that converts electrical energy into mechanical energy after receiving the control of pulses, which is represented as linear movements or discrete rotational. The stepper motor consists of two important elements: the rotor, which always contains permanent magnets, and the stator, which includes many windings. The electromagnetic interaction that occurs between these two elements will lead to the rotor making a step movement with each polarity change in the stator windings, where there are direct correspondences in stepper motors between the position of the rotor and energization of a particular stator coil [47].

On the other hand, the stepper motor is one of the best motors, which can be controlled simply, where is considered more reliable compared to dc motors because they do not contain brushes. The stepper motor is best economical in most sensitive applications such as microscopic operations, disk drives, scanners, and printers [47].

The general principle of the two-phase stepper motor is shown in Fig 2.8. The rotor includes a 1-pole pair permanent magnet, at the windings of one phase, are activated, a magnetic dipole will be created on the stator side and for example, if the phase 2 is active (and the phase1 is off), therefore winding3 generates an electrical south pole and also winding 4 an electrical north pole. As shown in Fig 2.7 that rotor in a stable position with phase 2 only operation. The number of steps per revolution of the rotor and the stepping angle is given as follows [44]:

$$S = 2.n.m \tag{2.8}$$

$$\Delta\Phi = \frac{360}{S} \tag{2.9}$$

Where the number of rotor pole pairs is n and the number of stator phases is m, in the hybrid stepper motor, the n is half the number of rotor teeth.

As an example, when n = 1 and m = 2, and when there are four steps for per revolution, as well as the stepping angle of 90 degrees as shown in Fig 2.8.

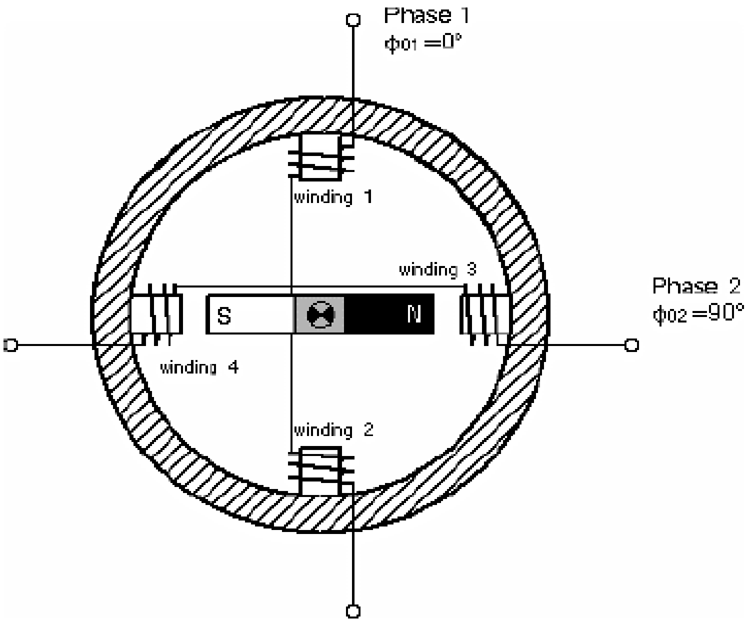


Fig 2.8: The general principle of the two-phase stepper motor [44]

The current I(t) in the coil is the function of the provided voltage U as well as the coil properties. The general equation between U and I(t) is given from:

$$U = emf + R.I(t) + L\frac{dI(t)}{dt} \tag{2.10}$$

Where emf is the electromotive force induced, R is the resistance of the coil, and L is the inductance of the coil. Thus, the differential equation can be written in the Laplace field expression:

$$I(s) = \frac{U(s)}{Ls+R} \tag{2.11}$$

The arrangement of the coils must be taken into consideration for more than one phase, and the location of the coils Φ0j is defined as the stator, as a function of the number of phase m, as in table 2.3

Table-2.3: The location of the coils Φ_{0j} at the stator which a function of the phase number m

No of phase m	2	3	4
$\emptyset_0$ of phase1	0	0	0
$\emptyset_0$ of phase2	90	60	45
$\emptyset_0$ of phase3	-	120	90
$\emptyset_0$ of phase4	-	-	135

The torque in electrical and mechanical for a stepper motor is given as follows [48]:

$$T_{elect} = k \,.\, I \tag{2.12}$$

$$T_{mech} = \mathrm{J}\frac{d\omega}{dt} + \mathrm{D\omega} + \mathrm{TF} \tag{2.13}$$

Where *J* is the inertia of the rotor and the load in (kg-m^2), *D* is indicated the viscous damping constant, and ω is the rotational velocity of the rotor. *TF* is indicated frictional load torque in (N-m) and k is constant depend on a stepper motor.

$$T_{elect} = T_{mech} \tag{2.14}$$

$$k \,.\, I = \mathrm{J}\frac{d\omega}{dt} + \mathrm{D\omega} + \mathrm{TF} \tag{2.15}$$

$T_F = 0$

By taking Laplace Transform for an equation 2.15:

$$\mathrm{k}.\,I(s) = \mathrm{JS}\ \omega(\mathrm{s}) + \mathrm{D}\omega(\mathrm{s}) \tag{2.16}$$

$$\text{k}.I(s) = \omega(\text{s})[\text{ JS} + \text{D}] \tag{2.17}$$

$$I(s) = \frac{\omega(\text{s})[\text{ JS} + \text{D}]}{K} \tag{2.18}$$

$$\frac{U(s)}{R + Ls} = \frac{\omega(\text{s})[\text{ JS} + \text{D}]}{K} \tag{2.19}$$

$$\frac{\omega(s)}{U(s)} = \frac{\text{K}}{[R + Ls].[\text{ J S} + \text{ D}]} \tag{2.20}$$

Assuming D = 0, and therefore the viscous damping is very small, and the equation.2.21 writes:

$$\frac{\omega(s)}{U(s)} = \frac{k/J}{s.[R + Ls]} \quad \text{Divide by L} \tag{2.21}$$

$$\frac{\omega(s)}{U(s)} = \frac{k/LJ}{s^2 + \frac{R}{L}s} \tag{2.22}$$

$\frac{R}{L} = 3 \; and \; \frac{k}{LJ} = 1000$

According to [49] the transfer function of a stepper motor would be :

$$G(s) = \frac{1000}{s^2+3s} \tag{2.23}$$

This model can be used in the case of simulating other types if the sinusoidal feature of the torque angle curves is discarded, for example, brushless DC motor. This system is critical stability because it has a pole at s=0 and this would be a challenge to work with al previous researchers works with a stable system.

Chapter Three

WNCS with PSO Fuzzy PID Controller

3.1 Introduction

The goal of this chapter is to describe the design and simulation of the NCS proposed model with the wireless ZigBee. The MATLAB is used based on a TrueTime tool simulator, and also WNCS simulation is presented across different network conditions. Fuzzy PID controller is used for the WNCS model proposed with the PSO optimization algorithm, also an explanation of their mathematical equations.

3.2 The Simulation Toolbox for WNCS

In order to connect the proposed control system nodes with the ZigBee wireless network, TrueTime simulation tools are used with MATLAB to design a wireless network control system model complete, as shown in Fig 3.1. The local side contains both continuous-time plant and an event-driven actuator and a time-driven sensor with the sampling period Ts, which is assumed that the actuator and sensor have a shared local memory and that both are connected to the plant through the two ports directly of input and output by A/D and D/A, the remote side that includes the controller that is an event-driven, is connected to the reference input through the A/D input port, which is used to implement suggested control algorithm

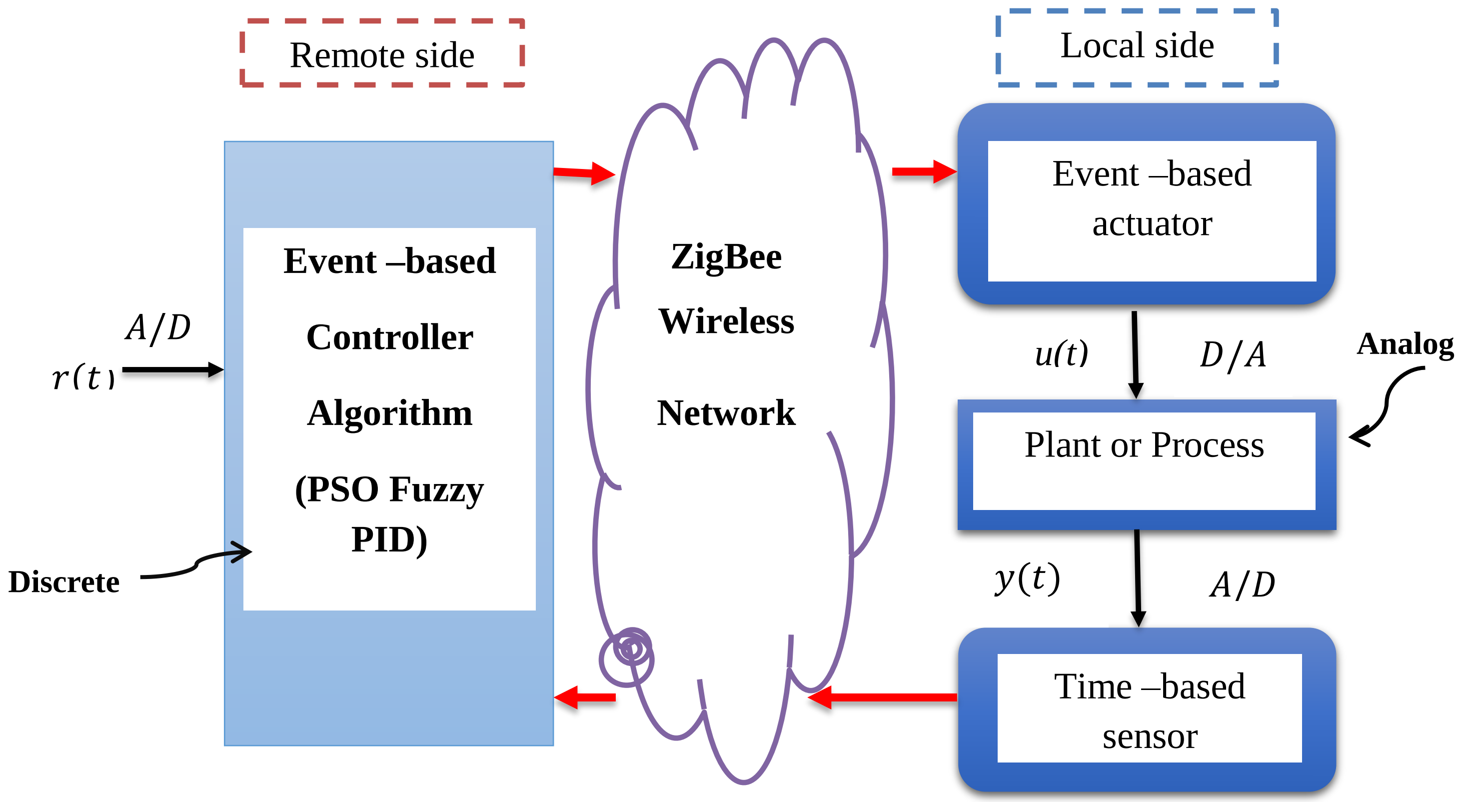


Fig 3.1: The suggested ZigBee NSC model.

The black lines in Fig 3.1 mean direct linked through $A\,/\,D$ and D/A ports, red lines mean the send of packets via a ZigBee wireless network.

The choice of sampling time (T_s) is carefully considered because it will affect the quality of control (QoC). If the sampling time (T_s) is reduced, it will improve QoC and at the same time produce a smaller sampling time to the traffic load on the network, and this leads to longer delays in transmission, which leads to deterioration the QoC, also that affects in the performance of the system.

The value of the sampling period (T_s) is chosen after extracting the time constant value of the second-order equation in (3.1) after compared with the transfer function in equation (2.23):

$$G(s) = \left(\frac{\omega_n^{\,2}}{s^2+2\,\zeta\,\omega_n s+\omega_n^{\,2}}\right) \tag{3.1}$$

Shannon's theory is used which states that the value of the sampling period selection is approximately between one-half to one-tenth of the smallest time constant, to obtaining between 0.066 for minimum and 0.33 for maximum [25]. Therefore in this work, the selection of the sampling time is equal to 0.08 second.

3.3 Designing Model for the Zigbee NCS

Due to wireless networks are used as a communication method in WNCS design, the ZigBee network is used as a wireless communication method. ZigBee network is simulated using True Time wireless block, where it is based on the CSMA/CA protocol with some simulation modifications to ZigBee network parameters which will be explained later. The wireless network block works as follows: when the node wants to send a message, the triggering signal is sent to the wireless network block on the corresponding input channel, upon completion of the simulated transmission of the message, the network block will send a new triggering signal on the corresponding output channel (for the receiving node), where the sent message is placed in the buffer at the receiving computer node, and the message also contains information about the receiving and sending node of the computer node as well as random user data that usually represents the control signal and the measurement signal, the message length, also optional features in real-time such as deadline and priority [29].

3.3.1 ZigBee Wireless Network Node

The wireless network True Time block operates in the same way as the wired True Time block but taking into account the path loss of radio signals.

The wireless network block has two inputs :($X-\text{axis}$) and ($Y-\text{axis}$) that are used to locate the following nodes of the controller node, and sensor/actuator node, as well as the nodes of the interference transmitter and interference recipient, where the location of sensor/actuator node (Xsc, Ysc) is positioned at (0,0), the location of controller node (Xc, Yc) is positioned at (50,0), the location of the interference recipient (Xir, Yir) is positioned at (110,0), the location of the interference transmitter (Xit, Yit) is positioned at (10,0). The parameters of the IEEE 802.15.4 ZigBee network are given in Table 3.1 [29]. The wireless network TrueTime block is configured with the following parameters [7]. The Fig 3.2 shows the parameters of the ZigBee True Time block:

1- **Network type**: This parameter used to define the MAC protocol, the network type can be either IEEE 802.15.4 ZigBee or IEEE 802.11b/g WLAN, where the IEEE 802.15.4 ZigBee network that is used in this work.
2- **Network number**: is a parameter that is used to the identification of the number of the network block, where it is numbered of the block (1) in this work model.
3- **Number of nodes**: This parameter is used to determine the number of nodes that will be linked to a network, where 4 nodes are used in this work.
4- **Data rate (bits/sec)**: It determines the speed of the network.
5- **Minimum frame size**: This parameter is used to determine the minimum frame size of the packet in (bits). The size of the packet is defined through the type of network. If a frame size less than this parameter padding is implemented to fit the minimum length.
6- **Transmit power :**this parameter determines the strength of the radio signal and the distance it will travel.
7- **Receiver signal Threshold:** This parameter is specified for the medium whether busy or not. If the received power is above this threshold.
8- **Path-loss exponent:** This parameter represents the radio signal path loss, which is modeled as $\left(\frac{1}{d^a}\right)$ where: "d" indicates a distance in meters and “a" indicates a parameter that is suitably selected to model the environment, the interval of the parameter "a" is normally having chosen between (2-4).

9- **ACK Timeout:** This parameter specifies the time it takes for the sent node to search for an ACK message, where if the timeout is passed and without an ACK message, the message will be resent again.

10- **Retry limit:** This parameter is used to specify the maximum number of times for retransmitting messages before the sender will be discarded the packet.

11- **Error coding Threshold:** The threshold interval of error coding is between [0, 1], which specifies the proportion of block errors at the message, which the coding can be dealt with. For example, certain coding systems can completely reconstruct a message when the block errors are less than 3%.

Table-3.1: Parameters of the IEEE 802.15.4 ZigBee network [29].

Parameters	**Value & units**
Network type	IEEE 802.15.4 (ZigBee)
Network number	1
Number of nodes	4
Data rate	250KBps = 2 μb/s
Minimum frame size	248 bit=81byte
Transmit power	3 dBm
Receiver signal threshold	-48 dBm
Path loss exponent	3.5 $(\frac{1}{d^a})$
ACK timeout	0.000864 sec =864μ sec
Retry limit	5
Error coding threshold	3%
Data	80 bit=10 Byte

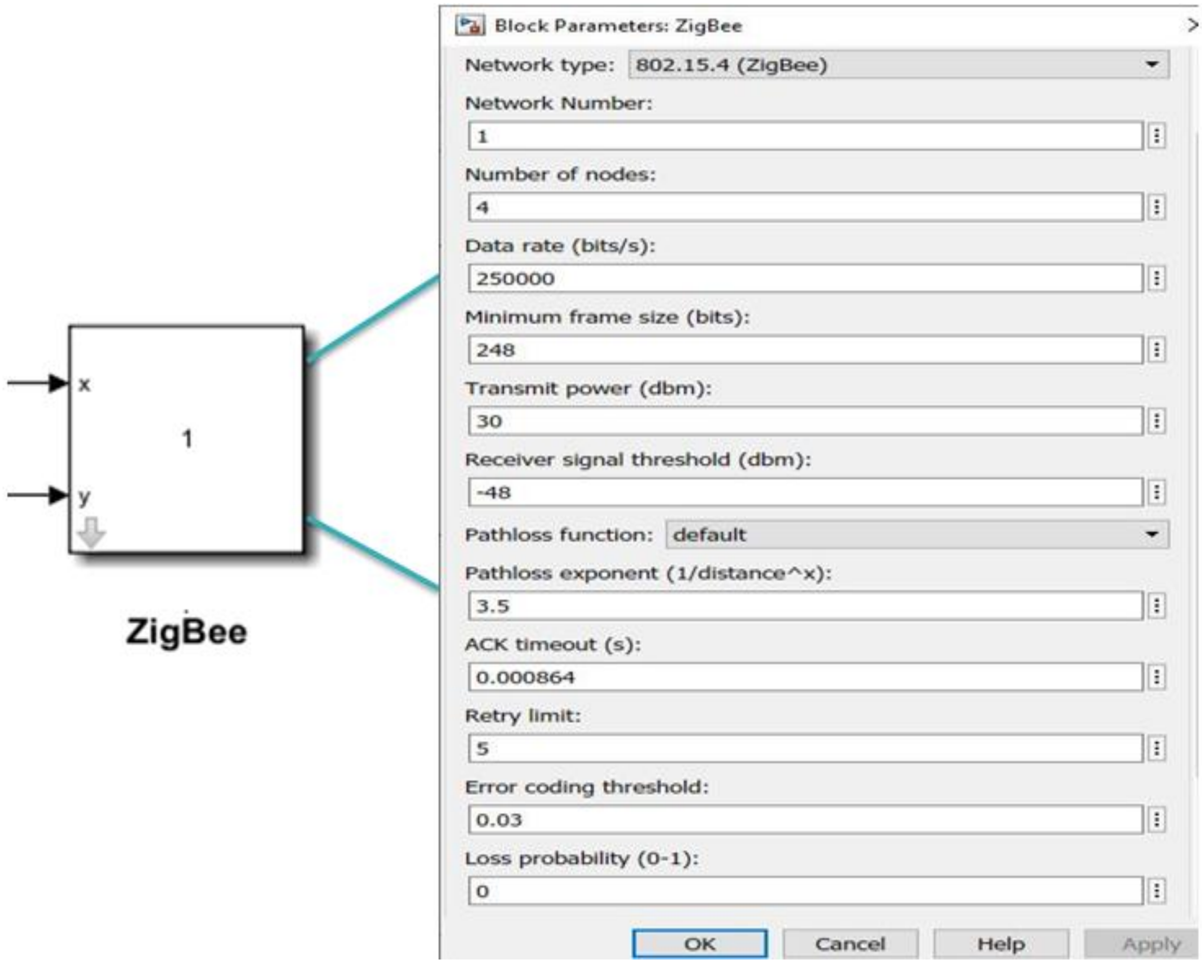


Fig 3.2: The parameters of the ZigBee True Time block

In the case that the signal level at the receiving node is greater than the received signal threshold, where the signal is supposed to be able to detect, then the Signal-to-Noise Ratio (SNR) is used to calculate the block Error Rate (BLER). The BLER with the message size is used for calculating the number of bit errors in the message and when the percentage of bit errors is below the error coding threshold, then the channel coding scheme can be able to completely build the message.

In addition, if it does not receive the ACK message during the ACK timeout, the message will resend after the retrying limit and before the sender abandons the message or ignores it, as the message is not retransmitted at all.

3.3.2 Interference Nodes

The two interference blocks are used in TrueTime to the proposed design. The main goal of the interference node is the process of generating traffic on the wireless network and thus the network will be loaded, which leads to the effect of time delay in the transmission process, as the proposed idea is similar to the experiment based on [15].

The two interference nodes are the interference transmitter node and the interference recipient node, where the interference transmitter node is connected to the MATLAB block BWshare and number nodes through the A / D input port (where BWshare represents the bandwidth). The second node is the interference recipient node. The interference transmitter node will share the control system nodes (as sensor/actuator and the controller nodes) with the same network channel. BW share is set to a constant value and where this value is used to determine the bandwidth through the interference transmitter node. TrueTime Kernel blocks dialogue of interference transmitter and recipient nodes are set with the following parameters:

A- **Interference transmitter node: -**

1- **Name of init function (MEX or MATLAB):** $interference\_transmitter\ \_init.m$. The m.file used to initialize and define the kernel operating system for the True Time kernel block (interference transmitter node).
2- **Init function argument (arbitrary struct):** It is set as [0], not used.
3- **Number of analog inputs and outputs:** It is set as [2, 0], The [2] represents the input A/D port that connects to the MATLAB block is used (BW block and number nodes block)
4- **Number of external triggers**:[0],not used

5- **Network and node numbers**: Is set as [4], this parameter represents the node identifier used for the TrueTime kernel block (Interference transmitter node).

6- **Local clock offset and drift:** Is set as [0,0], not used**.** In Fig 3.3 shows the block parameters of the interference transmitter node

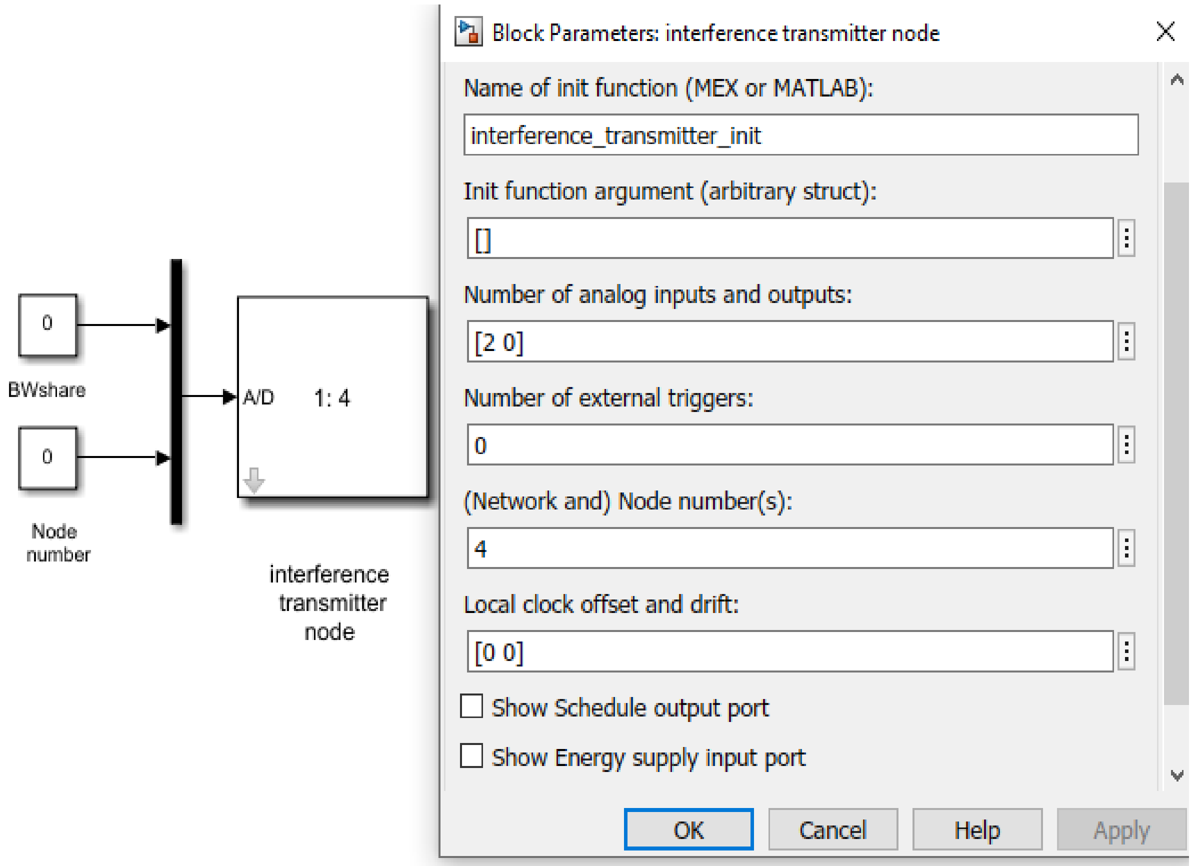


Fig 3.3: The block parameters of the interference transmitter node

B- **Interference recipient node:-**

1- **Name of init function (MEX or MATLAB):**

$interference\_recipient\_init.m$, the m-file is used to initialize and define the kernel operating system for the TrueTime kernel block (interference recipient node).

2- **Init function argument (arbitrary struct):** is set as [0], not used.

3- **Number of analog inputs and outputs:** [0,1] is used, need only $D\backslash A$ output port for shows the received signal from an interference transmitter node

4- **Number of external triggers**:[0],not used

5- **Network and node numbers**: Is set as [3], this parameter represents the node identifier used for the TrueTime kernel block (interference recipient node).

6- **Local clock offset and drift:** Is set as [0,0], not used. In Fig 3.4 shows the block parameters of interference recipient node

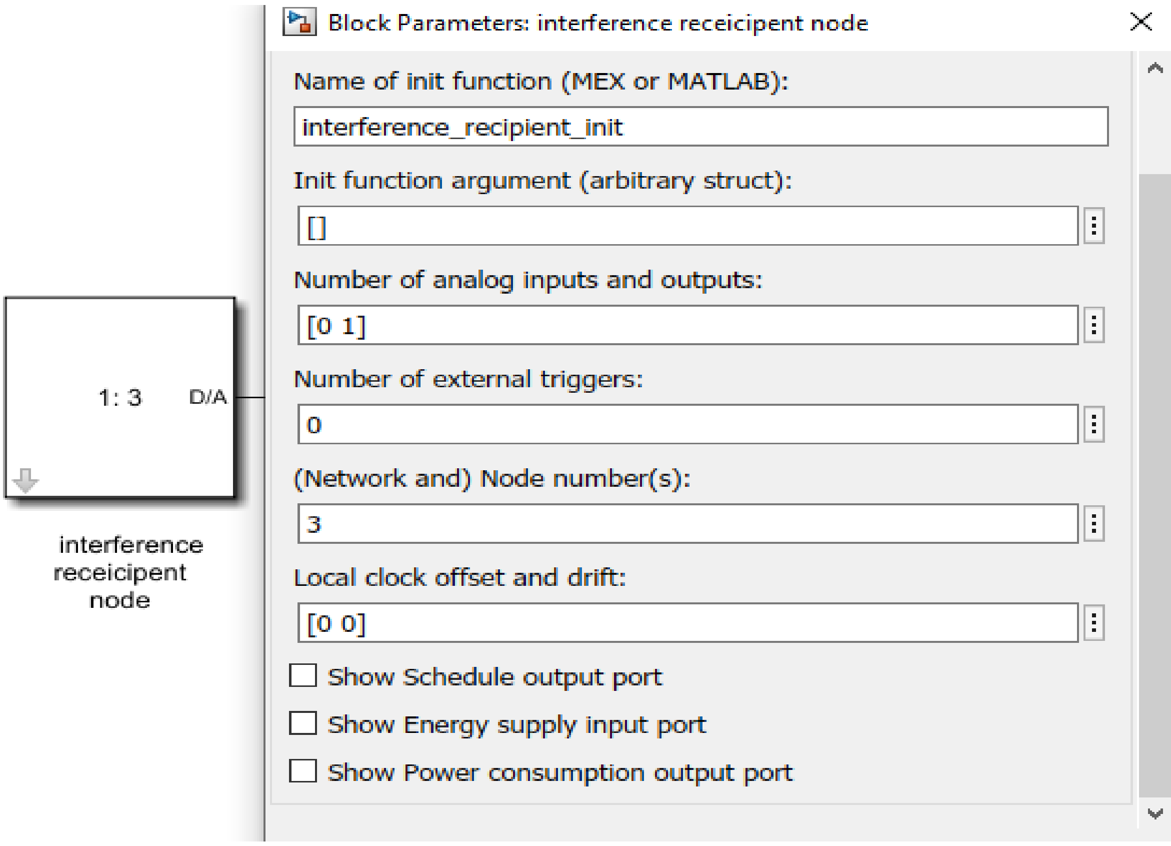


Fig 3.4: The block parameters of interference recipient node

In the interference_transmitter_init. m, the kernel is initialized for the operating system with the setting of fixed-priority scheduling "*prioFP*", by using the command (ttInitKernel), and two mailboxes are created using the command (ttCreateMailbox), the goal of creating the mailbox is for communication between the interference transmitter and recipient nodes, and to verify the transmission of the packet. Therefore the task is created periodically and the priority of the *interference_transmitter_task* is determined using the command(*ttCreatePeriodicTask*).

this task will call (*interference_transmitter_code*) that written in the m file, where the interference_transmitter_code will read the two input values using the command $(ttAnalogIn)$, which is represented by value BW share and node number. The interference transmitter code will be created and sent messages using the command ($ttSendMsg$).

The $interference\_transmitter\_m\ .file$ will create another task with the name $power\_request\_interference\_transmitter\_task$, which calls the file the function code, named ($power\_request\_code$).

Also, the local memory of the interference transmitter is used to store the data needed for the code function, which is retrieved at each function call. When the file $interference\_transmitter\_code.m$ will call in order to perform by the $interference\_transmitter\_task$, firstly the value of the BW share is read since the value of BW has an important impact on network traffic, when the value of BW is high the traffic on the network will be high and vice versa. Second, the code function works by reading or retrieving basic data from the relevant memory, where there is retrieved data. The retrieved data as following:

1- $Max\ P$: The value indicates the maximum packet size, which depends on the type of network used.
2- $Nodes$: represents the number of nodes that are emulated by the kernel block of the interference transmitter node.

in each function call, where BWshare is multiplied by the number of nodes, and by the maximum number of packets and the random weighting parameter (w), (whose value is between 0 and 1), this process is repeated for the nodes time, meaning that from one to nodes a packet will be sent over the network and the following size [15]:

$$Packet\ Length\ = (N \times BWshare\ \times\ MaxP\ \times\ w)byte \quad (3.2)$$

$$(byte)\ = (No.of\ Nodes\ \times (0 \leq\ BWshare \leq 1) \times (MaxP) \times 0 \leq\ w \leq 1)$$

$$(untiless) \quad (untiless) \quad (byte) \quad (untiless)$$

The N indicates the number of nodes, and the w is periodically varied (as random value) to generate various packet sizes, consequently, the various time delays will appear in the communication environment and this will lead to a call to the code function ($interference\_transmitter\_code.m$) to send the packets to the interference receiver node so that the type of these packets is known as "interference".

The interference_recipient_init. m, the kernel is initialized for the operating system with the setting of fixed-priority scheduling "$prioFP$", by using the command (ttInitKernel), and three mailboxes are created using the command (ttCreateMailbox), the goal of creating the mailbox is for communication between the interference transmitter and recipient nodes, and to verify the transmission of the packet. Therefore the task is created periodically and the priority of *interference_recipient_task* is determined using the command *(ttCreatePeriodicTask)*, and this task will call *interference_recipient_code.m*, and the *power_response_code.m* will create another task to check the messages arrive or not that will name *power_response_interference_recipient_task*.
When interference_ recipient_task will call by function $ttCreateTask$ the file interference_recipient_code. m at first, it will receive an interference signal from a network, and then show it in the output port to displays the interference signal.

The main objective of the interference recipient node is to receive the number of packets from the interference transmitter node and to ensure that the network is loaded with a certain number of packets through the nodes that have been emulated by the interference transmitter node and in order to study the impact of this load on the system performance and also know the number of nodes that the network will handle.

3.3.3 Sensor/Actuator Node

As mentioned in the previous chapters, the actuator and sensor are supposed to have a shared clock and local memory, and they are directly connected to the plant by the A/D input port and the D/A output port, and therefore one kernel block is used to design the sensor and actuator of two tasks (sensor/actuator node). The parameters needed to the TrueTime kernel block (sensor/actuator node) dialog are configured as follows:

1- **Name of init function (MEX or MATLAB):** *actuator_initialize* , the m-file is used to define and initialize the kernel operating system for the TrueTime kernel block (sensor/actuator node).

2- **Init function argument (arbitrary struct):** is set as [0], not used.

3- **The number of analog inputs and outputs:** is set to [1,1], one is the A / D input port and the second is the D/A output port which is used for reading and send to and from the connected plant.

4- **Number of external triggers**: is set to[0],not used

5- **Network and node numbers**: is set as [1,1], the network number is 1, and the node number is 1

6- **Local clock offset and drift :** is [0,0], not used. In Fig 3.5 shows the block parameters of Sensor/Actuator Node

The sensor/actuator node is linked to the local memory and contains the necessary data which will be used next to the sensor and actuator. In the actuator_initialize . m, the operating system kernel is initialized as “prioDM" scheduling by using the command (ttInitKernel), also the actuator_initialize . m contains three mailboxes are created using the command (ttCreateMailbox), which is used to create communication with the controller node, *actuator_initialize .m* provided two tasks, named *actuator_task* and *sensor_task*. The sensor_task operates as a time-based when the period sampling time is taken sensor_task calls the m-file code function, which is called *sensor_code.m* also this code function performs the necessarily required operations to sensing the plant output

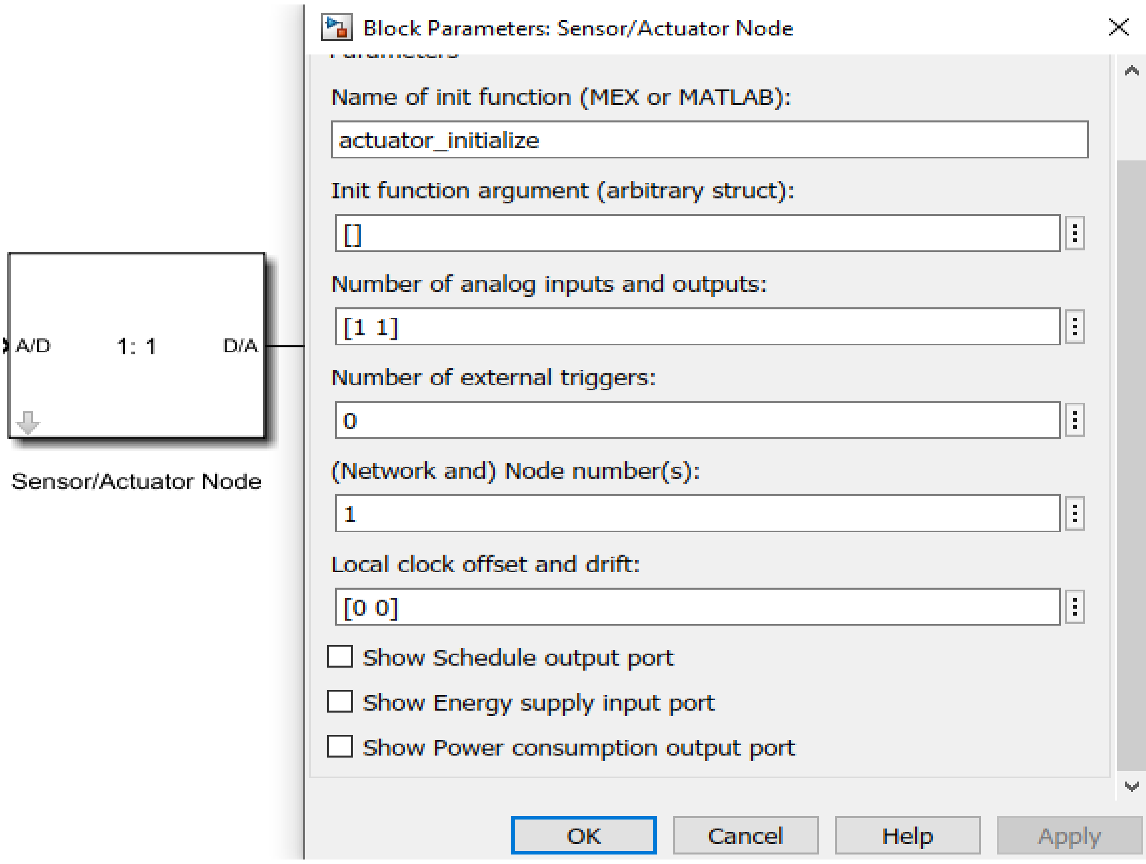


Fig 3.5: the block parameters of Sensor/Actuator Node

In the sensor_code. m two cases are implemented: the first case represents the readable output value of the plant. The second case creates a message named "sensor_signal" that contains the type of output and then the measurements are sent inside (sensor_ packet) to the remote side (regulator node), a call to sensor_task will be performing the following steps in the order:

1- Plant output is directly sensed via the A/D input port.
2- A *sensor_ packet* is prepared with the readable output value of the plant, which is called "*sensor_signal* " and the plant measurement.
3- A *sensor_packet* is sent to the remote side (*controller node*) over the network.
4- The timer starts when the *sensor_packet* is sent

From step four is noticed that *sensor_task* is accountable for time delay computation. When a response of the controller is received in *controller_packet*.

A *controller_packet* would be read from the network interface, where actuator_task is initialized as soon as the new controller_packet is received so that actuator_task is an event-based task. The *actuator_task* will be calling the *m.file* code function called ($actuator\_code.m$), this code function performs the necessary operations for the plant is actuating. A call to actuator_task will implement the following steps in order:

1- A $controller\_packet$ is read or pulled via the network interface called "$control\_signal$".

2- The timer is paused and saved it in an array called RTT.

3- A conversion (D/A) is performed for the control signal to actuate the plant with the newly received value.

The importance of using the sensor and actuator tasks in one kernel is eliminating the need for synchronization between the remote side (controller node) and the sensor/actuator node, also to calculate the value of time delays, and given that the sensor and actuator_task that will share the local clock, there is no necessary need for synchronization. This will reduce the additional packets required for synchronization on the network channel, where actuator_task is assigned the highest priority over the other tasks that may work in this operating system.

In other words, actuator_task cannot be interrupted when it enters into the ready queue or when it continues to operate. Figure 3.6 shows the algorithm steps followed by a sensor /actuator

3.3.4 Controller Node (or Regulator node)

The controller (or regulator) node in the TrueTime kernel block is configured with the following parameters:

1. **Name of init function (MEX or MATLAB)**: $regulator\_initialize$, an m-file is used to define and initialize the kernel operating system for the TrueTime kernel block

2. **Init function argument (arbitrary struct)**: Is set as [0], not used.
3. **The number of analog inputs and outputs**: is set to [1 1], one input A/D port is connected with action control of the controller algorithm, and the second output $D\backslash A$ port is connected to the signal generator of MATLAB block as a feedback system.
4. **Number of external triggers:** is set to[0], not used
5. **Network and node numbers**: is set as [1 2], the network number is 1, and the node number is 2
6. **Local clock offset and drift**: set to [0 0], not used. Fig 3.7 shows the block parameters of the Regulator node.

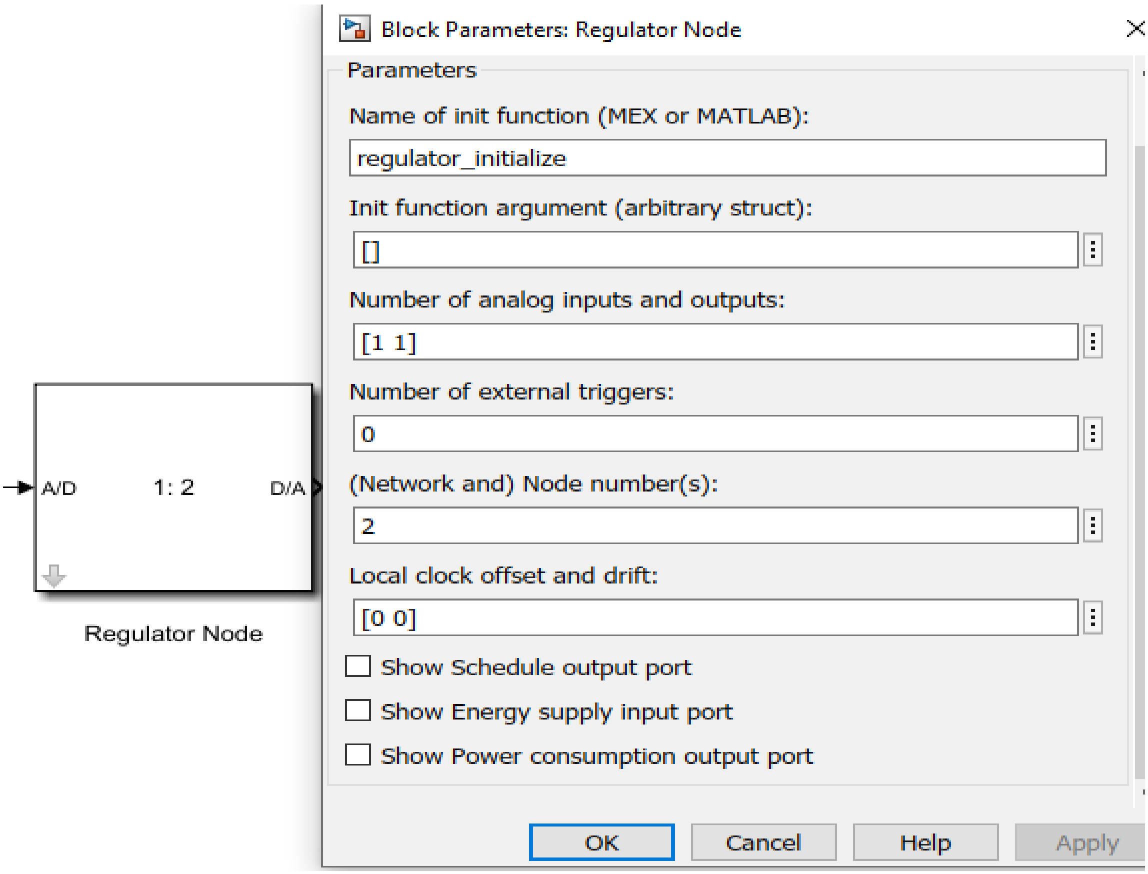


Fig 3.7: the block parameters of the Regulator node.

In the regulator_initialize, the operating system scheduling with this node is configured as the fixed priority "*prioFP*" by using the command (ttInitKernel), with high priority to the controller task and preference, that is, the controller task is not interrupted when it enters the ready queue or when it starts to work (or process).

The controller task is considered as an event-based task. The regulator_initialize includes three mailboxes are created using the command (ttCreateMailbox), that is created for communication between the controller node and the local side (sensor/actuator node), and a task is created as a control_task to calls code function that written in the m file, called control_code and also two other tasks are created with the name *power_response_control_task* and *power_request_control_task* in respectively, where the two tasks call code function that is written in m-files, called power_response_code and power_request_code in respectively.

When the control_task calls to the code function for the control_code in the controller node, In the first, the sensor_signal is received through a network interface, and then this signal is processed by the Fuzzy PID controller with the PSO algorithm, and then this signal will be sent as a control_signal to the destination of the sensor/actuator node. This code function calls and the following steps will be executed in sequence:

1- The received $sensor\_packet$ is read over the network interface.
2- The proposed controller algorithm is implemented to produce a control signal.
3- Control values are read via A/D port.
4- A $controller\_packet$ is prepared, where this packet is designed to provide: controller values and the signal type as "$control\_signal$".
5- A $controller\_packet$ is sent to actuator node via the network.

The MATLAB blocks are used to design and implement controller algorithm in step three, where are used to implement a Fuzzy PID controller with the PSO optimization algorithm is implemented in the form of a programmed code in m-file function.

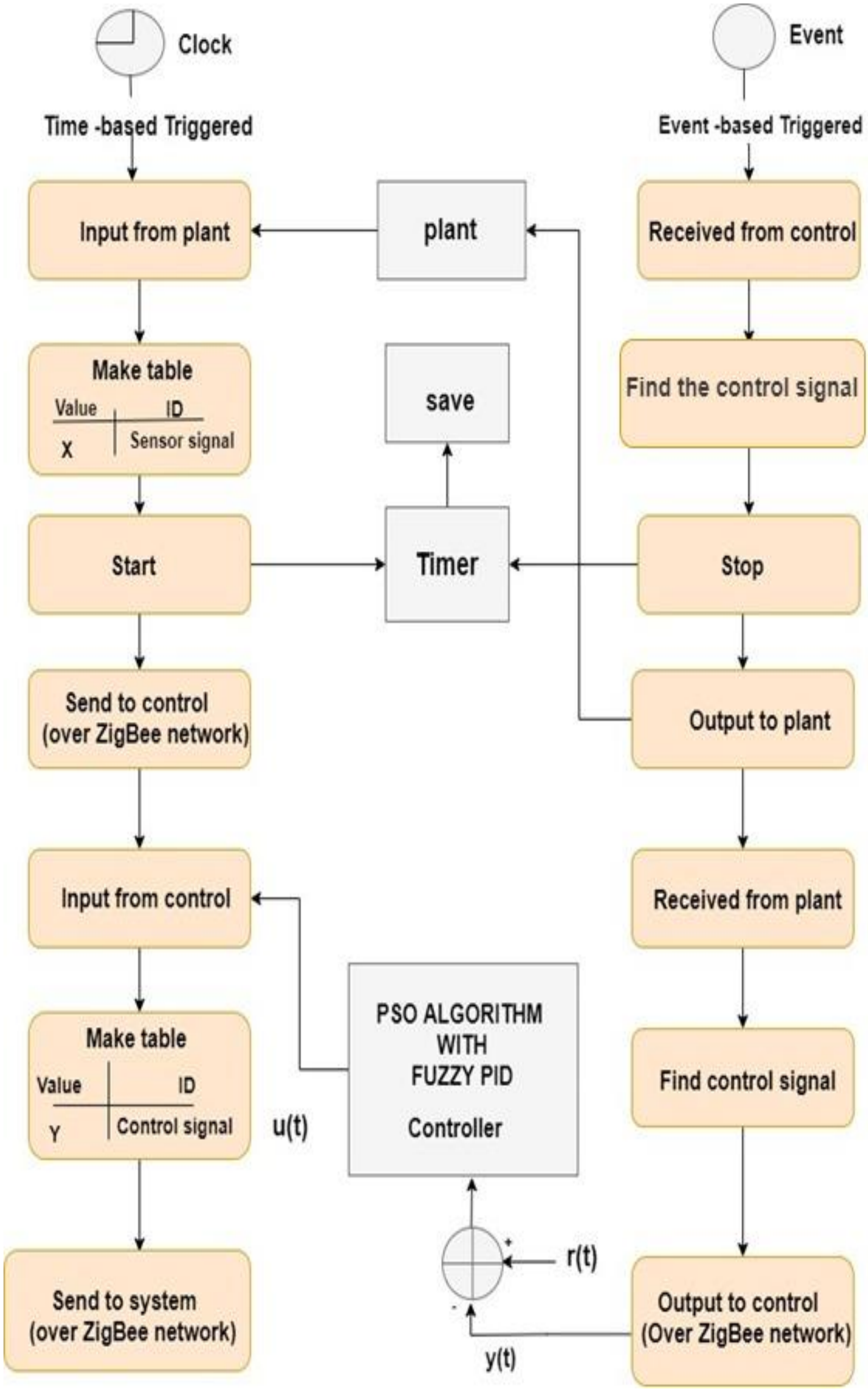


Fig 3.6: The algorithm steps followed by a sensor /actuator

3.4 Controllers Algorithms Simulation

The controller algorithm on MATLAB without True Time is designed with two stages:

1- Algorithm of the Fuzzy PID controller.

2- A PSO algorithm is used.

In the first stage, the gain parameters of the PID are adaptive by the fuzzy logic, which in turn gives the parameters automatic adjustment. The second step is to use the PSO algorithm to give the optimum rules for fuzzy logic. The two stages are related to working with each other online and to obtain the gains of the controller to handle the errors.

3.4.1 Algorithm of Fuzzy PID Controller

The fuzzy logic in our work is used for adjusting a gain of PID controller, which is dependent on the error and error change, where fuzzy logic contains two inputs namely error (E) and change error (CE) and three outputs are $\mathrm{Kp}, \mathrm{Ki}, \mathrm{and}\ \mathrm{Kd}$ gains, where these gains parameters are variable over time. Figure 3.8 shows the proposed design of the Fuzzy adaptive PID controller. The fuzzy logic system is designed in MATLAB with the name FIS file, and the following are chosen:

1- The number of two inputs in the range [-2 2] and the number of three outputs (kp, ki, kd) in the range ([0.008 0.01]), [0,0002 0.00025], and [0.0025 0.0036] are determined, the value of the parameter is chosen based on trial and error,

2- The Membership functions (MF) are assigned to each of the inputs and outputs. the Triangular functions are chosen for each (five membership functions for each of the inputs and outputs)

3- Fuzzy set of linguistic variables [NB (Negative Big), NM (Negative Medium), Z (Zero), PM (Positive Medium), and PB (Positive Big)] are set.

4- Mamdani type in the inference process.

5- The Defuzzification method is determined by a centroid (center of gravity).

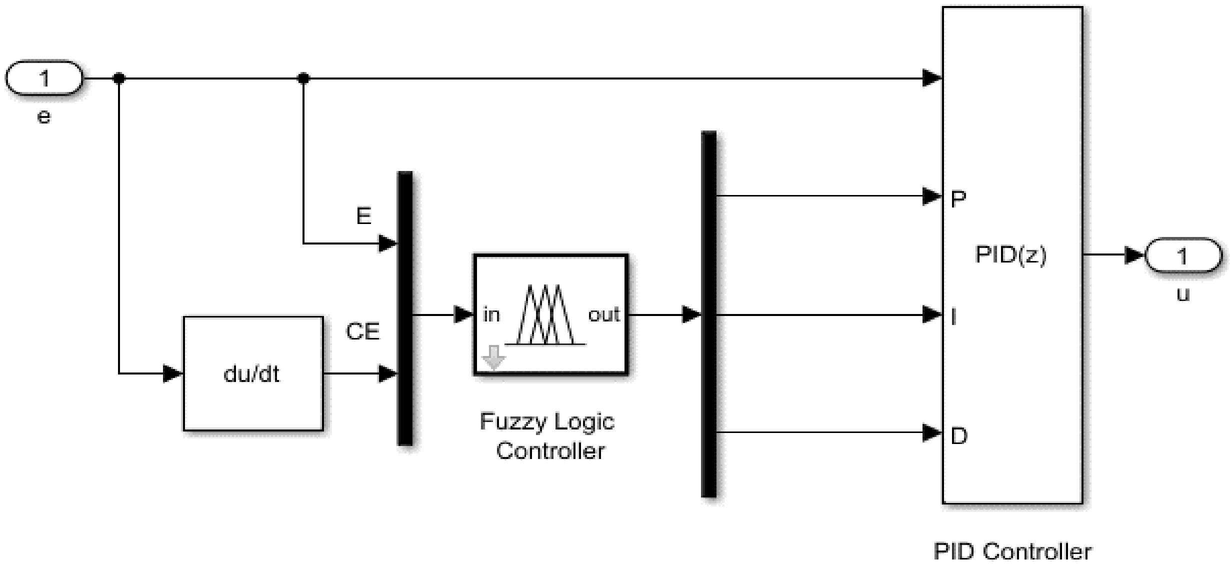


Fig.3.8: The proposed design of Fuzzy Adaptive PID Controller

The Mamdani system has an advantage by the simplicity of understanding and implementing the rules, which mainly depend on expert experience and human smart. Where the number of rules is the number of memberships raised to the exponent of inputs (5^2), so the number of rules in this work 25.

The choice of fuzzy logic rules depends on the experience of the experts or the knowledge of the control expert as the rules have a direct impact on the quality of the control system, therefore it was suggested to use the optimization technique (PSO) to find the optimal rules and suitable for the work of the control system online. Figure 3.9 shows the fuzzy logic designer (FIS), and Fig 3.10 shows the Membership functions for input and output of fuzzy logic: $a.E, b.CE\ , d.kp, c.ki, e.kd$.

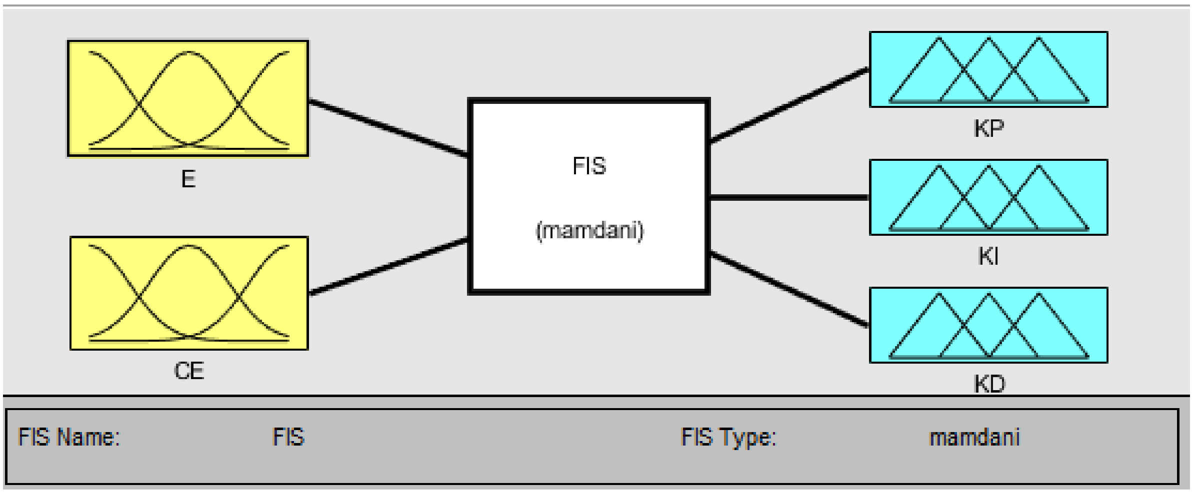
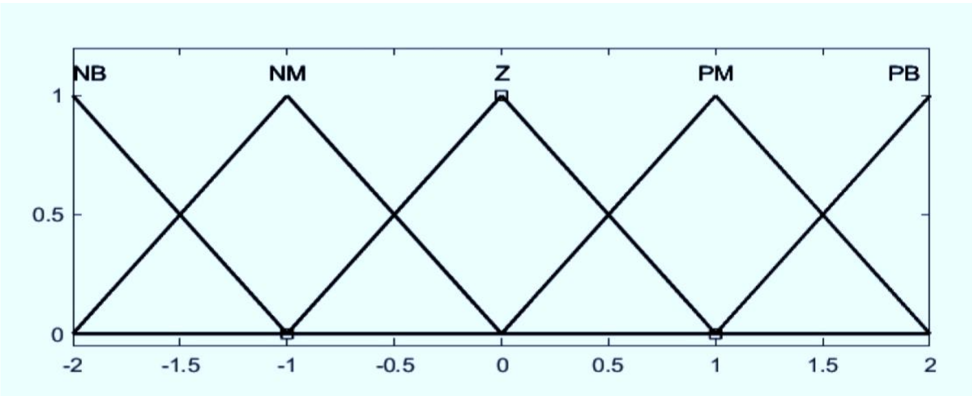
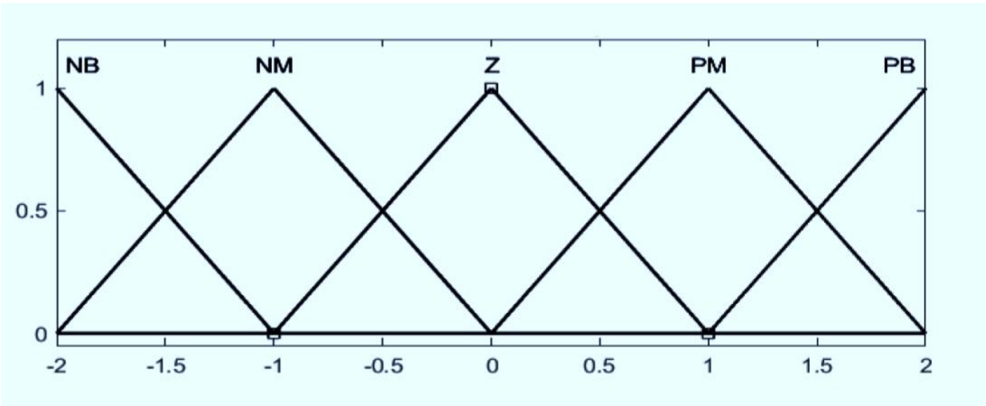


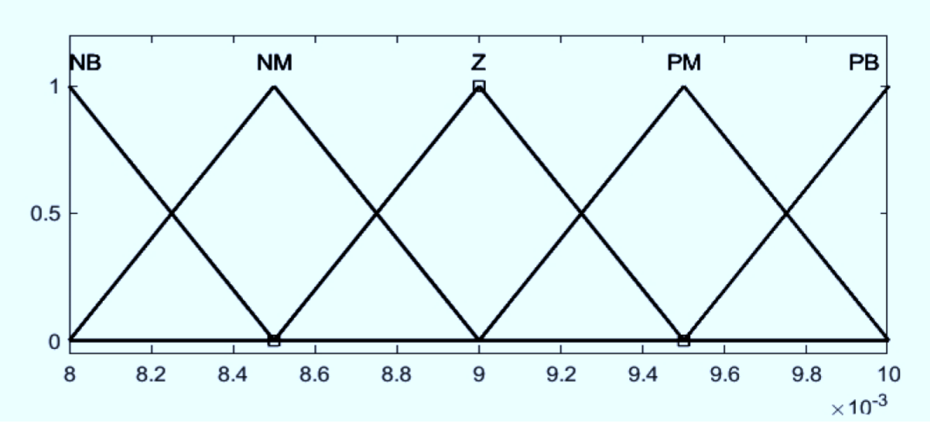
Fig 3.9: shows the fuzzy logic designer (FIS)



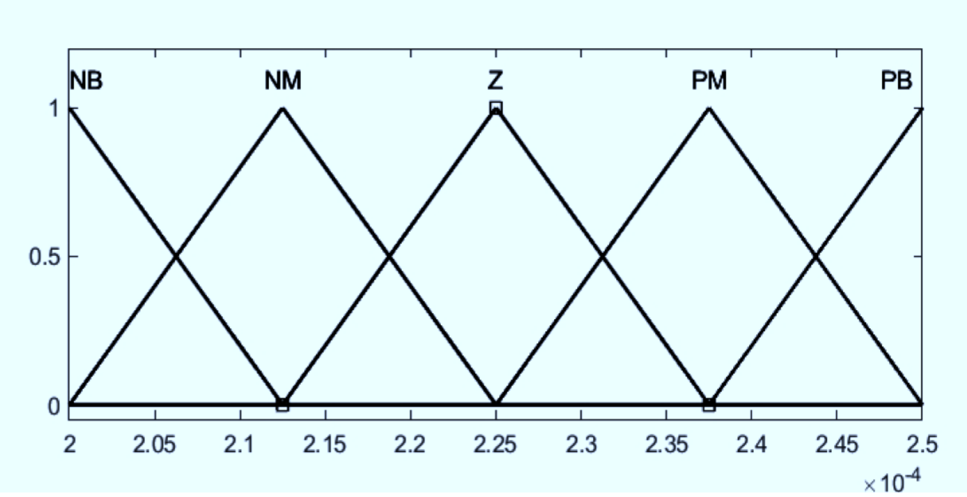
$\boldsymbol{a}$ – Membership functions of E



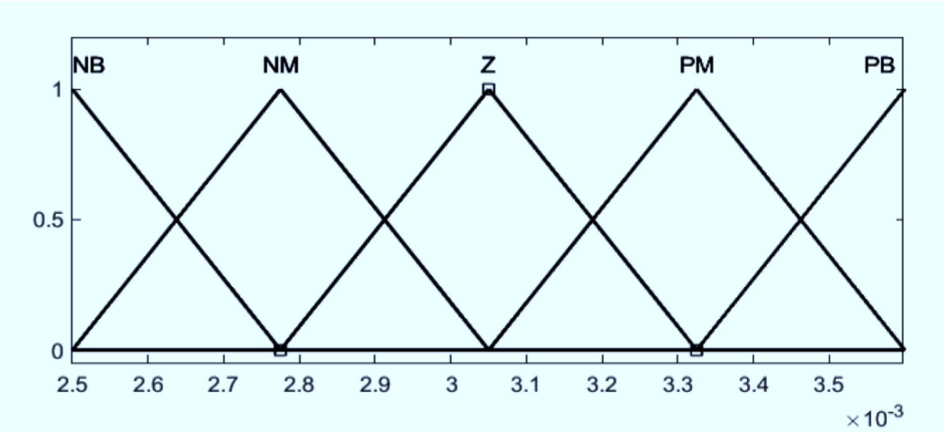
$\boldsymbol{b}$ – Membership functions of CE



$\boldsymbol{c}-$ Membership functions of Kp



$\boldsymbol{d}-$ Membership functions of Ki



$\boldsymbol{e}-$ Membership functions of Kd

Fig 3.10: Membership functions for input and output

3.4.2 PSO Algorithm

The main goal of using the PSO algorithm with the Fuzzy PID controller in WNCS design is to get the optimal rules of fuzzy logic and to improve the system response to the stepper motor. The PSO optimization algorithm is used with the proposed controller to obtain a control algorithm called the PSO Fuzzy PID algorithm. Figure 3.11 shows the flowchart of the PSO with the Fuzzy PID controller in our work.

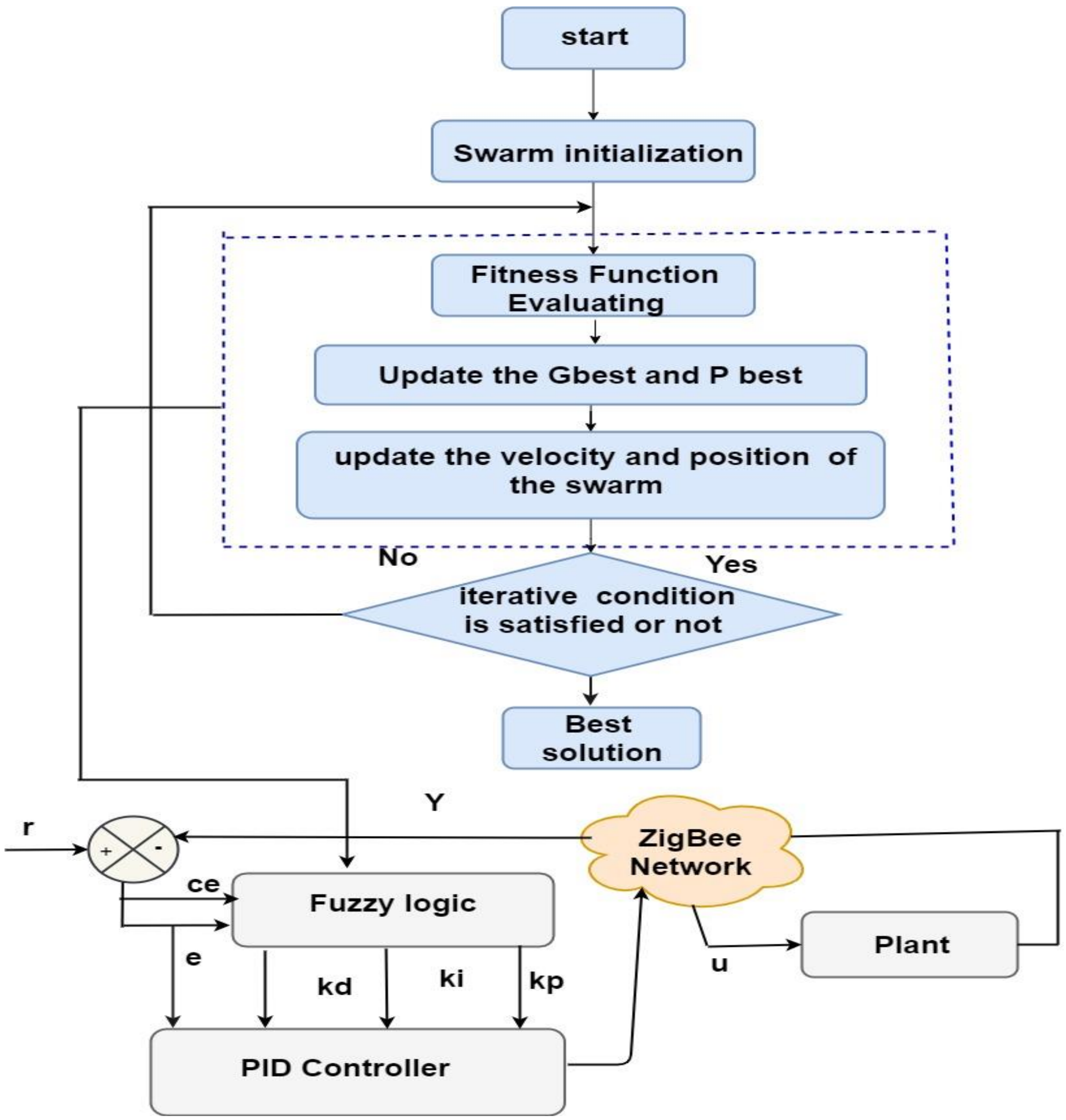


Fig 3.11: The flow diagram of the PSO with the Fuzzy PID controller

The PSO algorithm works to find optimal values rules. The PSO optimization algorithm uses a population of particles, which called a swarm. Swarm consist of seventy-five particles, each three particles represent one rule because Fuzzy logic has three output.

The Particle has a value between one into five due to membership functions used in our work are five [NB, NM, Z, PB, PM]. Every value of particle represents one of the Membership functions where one is NB, two is NM, three is Z, four is PM, and five is PB. The first output of fuzzy logic is Kp that take from one to twenty-five of particle. The second output of fuzzy logic is Ki that takes from twenty-six to fifty of particles. Last, the output of fuzzy logic is Kd that takes from fifty-one to seventy-five particles. The swarm can be described in the matrix by the following equation 3.3.

$$\begin{bmatrix} Swarm_{(1)} \\ : \\ Swarm_{(n\ )} \end{bmatrix} =$$

$$\begin{bmatrix} Kp_{(1,1)}\,..\,Kp_{(1,25)} & Ki_{(1,26)}\,..\,Ki_{(1,50)} & Kd_{(1,51)}\,..\,Kd_{(1,75)} \\ : & : & : \\ Kp_{(n,1)}\,..\,Kp_{(n,25)} & Ki_{(n,26)}\,..\,Ki_{(1,50)} & Kd_{(n,51)}\,..\,Kd_{(n,75)} \end{bmatrix} \tag{3.3}$$

The fitness function is defined according to equation 3.4 [44]. It should be noted that the fitness function differs from one researcher to another, and there is no fixed methodology for determining this function. For each iteration in the PSO algorithm, the value of the fitness value for all swarms is obtained to find the best swarm, which represents a set of rules for Fuzzy Logic**.**

$$\mathrm{J} = \int_0^{\infty} [w_1|e(t)| + w_2|e(t)|]dt + w_3 t_s \tag{3.4}$$

Where $e(t)$ indicates a system error, ts indicates settling time of the response. The item w_3t_s is used to reduce the settling time of response and coefficient ($w3$) is important to the smooth of the response curve. The item $w_2|e(t)|$ added to fitness function to avoid overshoot of response, and (w_2) is much larger than (w_1) ($w_2 > w_1$), these parameters selection of $w1, w2, w3$ is more important for the performance of the PSO.

The parameters in this work are used with value for $w3$ is 1000, $w2$ is 1000, and $w1$ is 0.1[44]. The parameters of the PSO algorithm in this work are chosen with the following values in table 3.2

Table-3.2: The PSO optimization algorithm

Parameter	**value**
$C1$	2
$C2$	2
$Wmin$	0.4
$Wmax$	0.8
Number of a particle (n)	75
Number of iterations (itr)	100
population size (swarm)	50

The using research procedures are followed in the design of the PSO Fuzzy PID controller proposed in the algorithm (1).

Algorithm 1: Pseudocode of PSO for Fuzzy PID controller

1: Initialize the number of particles, population size, minimum ($wmin$), and maximum ($wmax$) of inertia weight, acceleration factor, the maximum number of iterations ($bird\_setp$).

2: Determine the lower bounds of particles and upper bounds of variables for NB, NM, Z, PM, and PM.

3: Create a particles population randomly, where each particle is a set of rules for Fuzzy Logic, according to the following formula in equation (3.5) :

$$x(i,j) = LB\ (j)\ + rand() * (UB\ (j) - LB\ (j)) \quad (3.5)$$

$$i = 1\ ,2 \ldots n, and\ j = 1{,}2\ {,}3{,}4 \ldots\ 5.$$

4: The initial velocity of each particle is set according to the following formula in equation (3.6)

$$v(i,j) = 0.1\ *\ x(i,j) \quad (3.6)$$

5: Apply each particle on Fuzzy logic.

6: The fitness function of each particle is calculated by using the fitness function defined in equation 3.4.

7: Get global best particle ($globl\_best\_position$) and local best particle ($local\_best\_position$) is determined in the population, which represents the minimum fitness function.

8: Start loop.

9: Calculate inertia weight according to the following formula in equation (3.7)

$$w = wmax - (wmax - wmin) * ite/bird\_setp \quad (3.7)$$

where the current iteration is ite.

10: Get R1 and R2 randomly

11: The velocity for each particle is updated according to equation (2.6):

$$v(i,j) = w * v(i,j) + c1 * R1 * \{\text{local\_best\_position}(i,j) - x(i,j)\} + c2 * \text{R2} * \{\text{globl\_best\_position}(1,j) - x(i,j)\}$$

12: Each particle position is updated according to equation (2.7):

$$x(i,j) = x(i,j) + v(i,j)$$

13: Apply each new particle on Fuzzy logic.

14: Calculate the fitness function of each particle in the population using the fitness function defined in equation 3.4.

15: Find the new $local\_best\_position$

16: $if\ fitness(local\_best\_position) < fitness(globl\_best\_position)$

$then\quad globl\_best\_position \leftarrow local\_best\_position$

17: Print $fitness(globl\_best\_position)$ and $globl\_best\_position$

18: Stop the loop if the condition is satisfied.

19: Print result.

Chapter Four

Simulations and Results

4.1 Introduction

In this chapter, different cases of the WNCS model are studied and time delay analysis, the number of nodes that the WNCS model handles is calculated to maintain system stability. The WNCS model is simulated with MATLAB 2018b and True Time 2.0 tools in 2016. The PID and Fuzzy PID and PSO Fuzzy PID controllers are tested based on the response of the system. The best controller that can handle the increasing number of nodes is selected at the same time and the control system maintains its stability. Also, the proposed WNCS model is tested with packet losses in the wireless network**.**

4.2 Performance Analysis of Control System

Essentially, the goal of choosing a controller is to achieve the best reference tracking. Consequently, the performance of the controller is studied, which depends on the system response (motor or plant) related to the input reference. These criteria are chosen to test the operation and performance of the controller:

- Rise time (TR)
- Settling time (TS)
- Max Overshoot (MP)
- Integral Square Error (ISE) ,where ISE $=\int_0^t e(t)^2\, dt$

4.3 Response of the Stepper Motor

At first, the stepper motor is represented as a Transfer Function from equation 2.23, which is expressed in a MATLAB block (Transfer Fcn). The open-loop of the stepper motor shows in Fig 4.1. The response of the stepper motor given in Fig 4.2 is shown, whereas the response turns out is unstable.

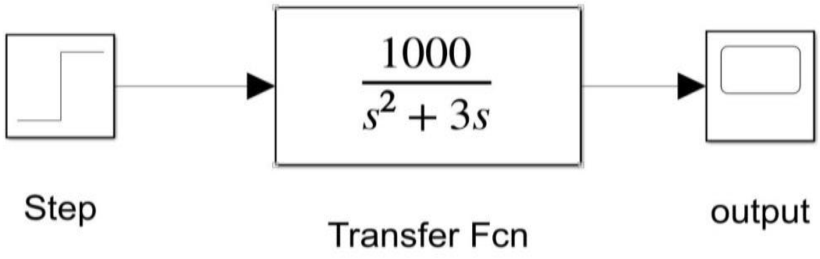


Fig 4.1: Open-loop of the stepper motor

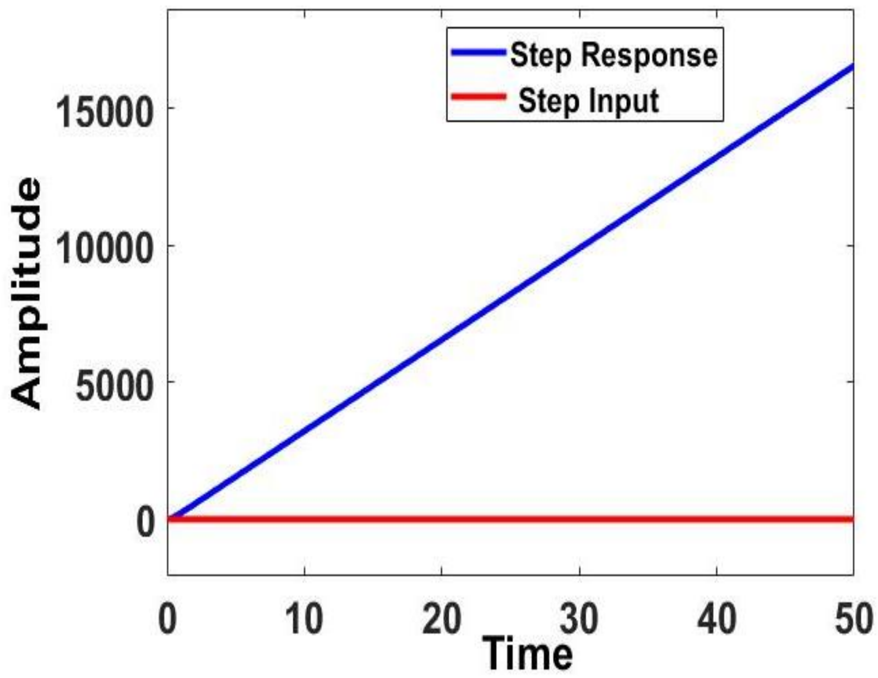


Fig 4.2: The response of the stepper motor

The closed-loop of the stepper motor (Transfer Function) that represented with MATLAB block and input unit step as a reference as shown in Fig 4.3.The output response of this Transfer Fcn as in Fig 4.4.

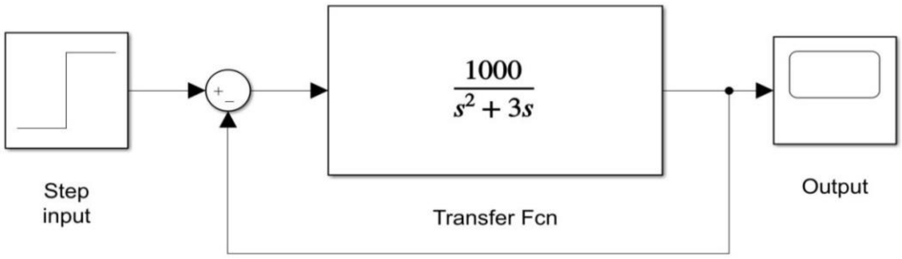


Fig 4.3: closed-loop of Transfer Function

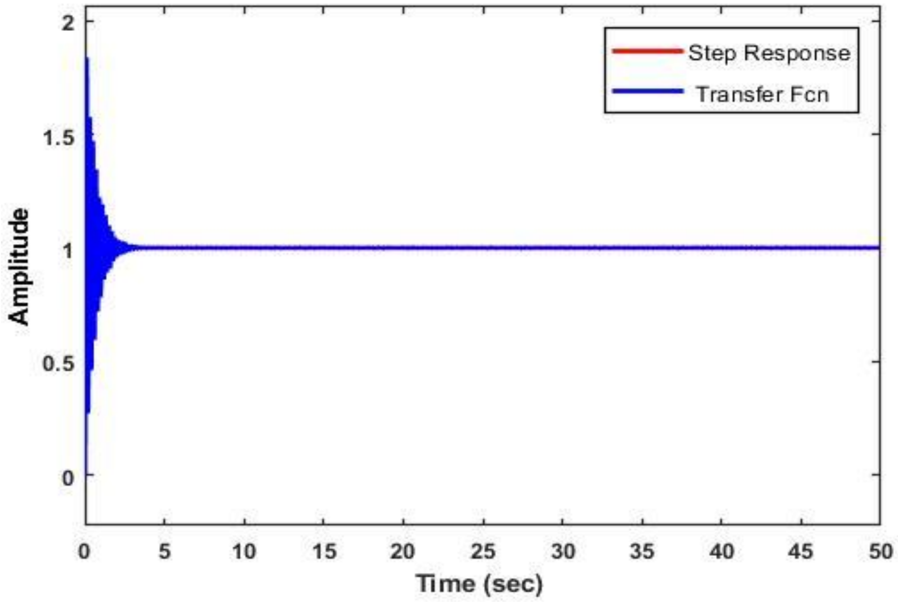


Fig 4.4: The step response of closed-loop

4.4 Simulation Case Studies

The ZigBee network is suggested as a communication protocol in this work. Three controllers are PID, Fuzzy PID, and PSO Fuzzy PID used in this work of the WNCS model. The discrete PID controller and fuzzy logic are designed with MATLAB blocks. These controllers have then been used in the Regulator node (or controller node). The control signal is digitally processed and it converted again from digital to analog at the sensor/actuator node. The simulation of WNCS model will be performed in two ways:

1- No-load: The value of BWshare is set to zero where there are no packets that will be sent from the interference transmitter node. The regulator node and sensor/actuator node remain the only ones active in this model.
2- With load: It is divided into two cases as follows :

A- Medium load when the value of BWshare is set to 0.4
B- High load when the value of BWshare is set to 0.9.

4.4.1 PID Controller with WNCS Model

The PID controller has three fixed gain parameter, these gains are initially set using MATLAB tuning, found as in table 4.1. WNCS is simulated in MATLAB R2018b and using True Time tools. PID controller is used only with the ZigBee network with two interference nodes as shown in Fig 4.5. The reference input is a square waveform (signal generator), it is linked to the regulator node directly via the A/D input port. The WNCS is simulated with execution time 50 sec and the sampling time is 0.08sec.

Table-4.1: Gain parameter of PID

Kp	Ki	Kd
$6.808\ x\ 10^{-3}$	$13.067\ x\ 10^{-4}$	$1.794\ x10^{-3}$

The response of the stepper motor for these parameters in case of no load at BW share is (zero) and the number of nodes is (zero) as shown in Fig 4.6. Figure 4.7 shows the stepper motor response of the PID controller with medium load at BWshare 0.4 and the number of nodes is 100 nodes, and Fig 4.8 shows the stepper response with high load at BWshare 0.9 and number of node is 75 nodes. The Performance of the PID controller shows in table 4.2.

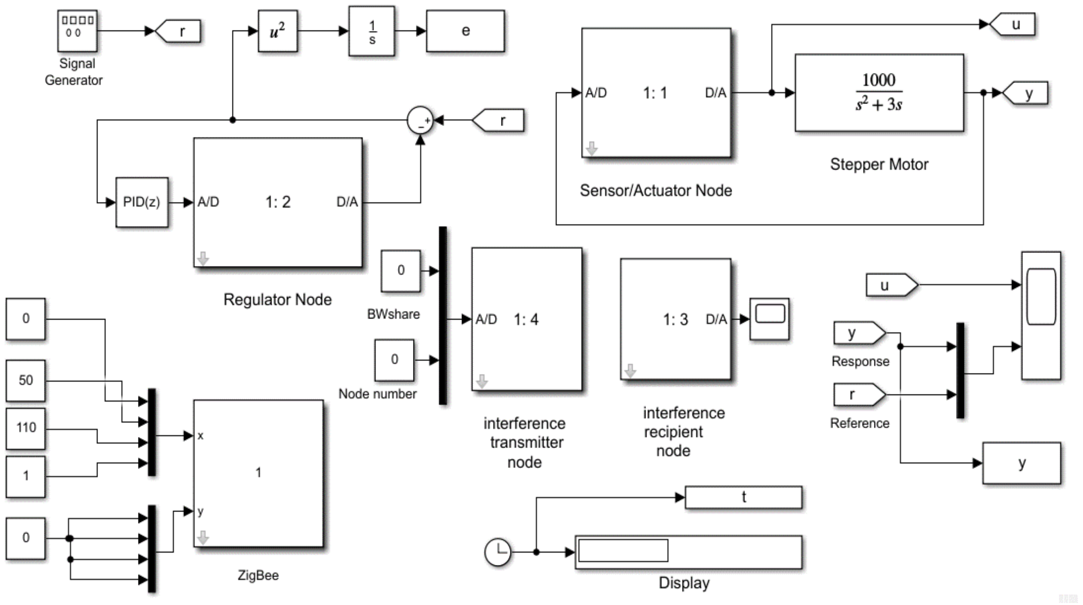


Fig 4.5: WNCS model with PID controller

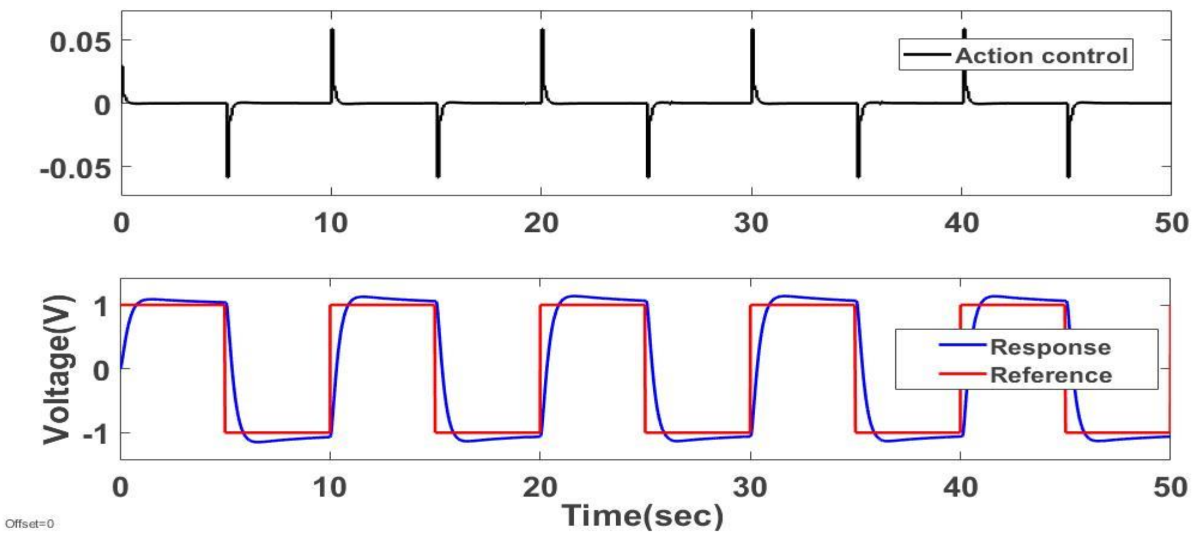


Fig 4.6: Stepper motor response and action control of WNCS with PID controller at BWshare =0, number of nodes=0

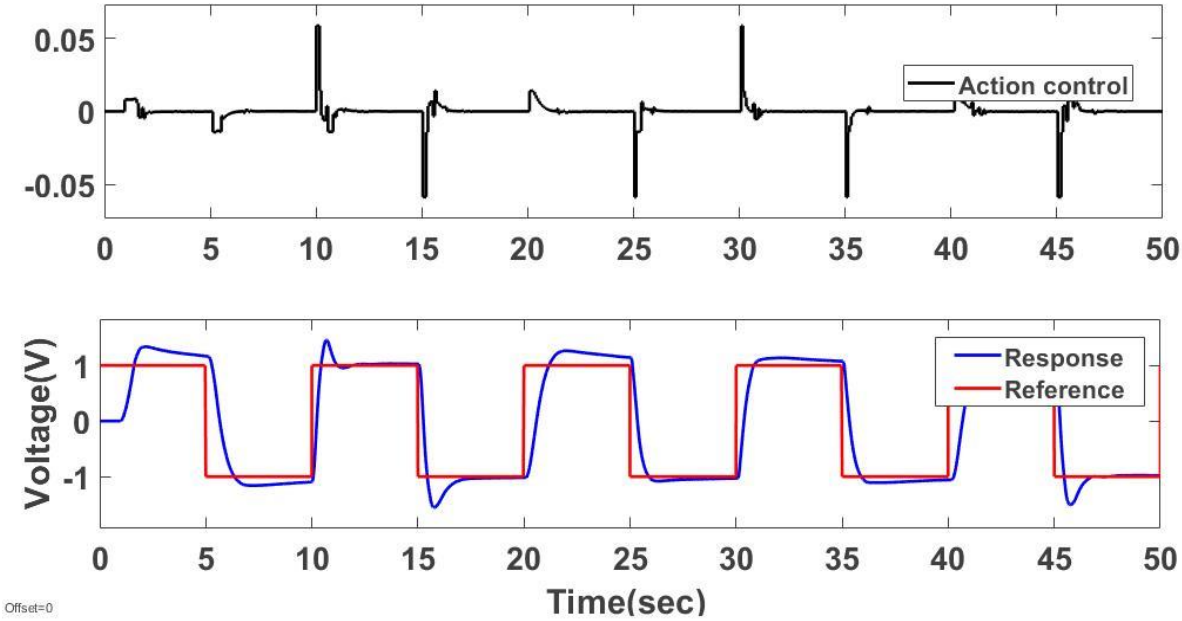


Fig 4.7 Stepper motor response and action control of WNCS with PID controller in medium load at BWshare =0.4, number of nodes=100

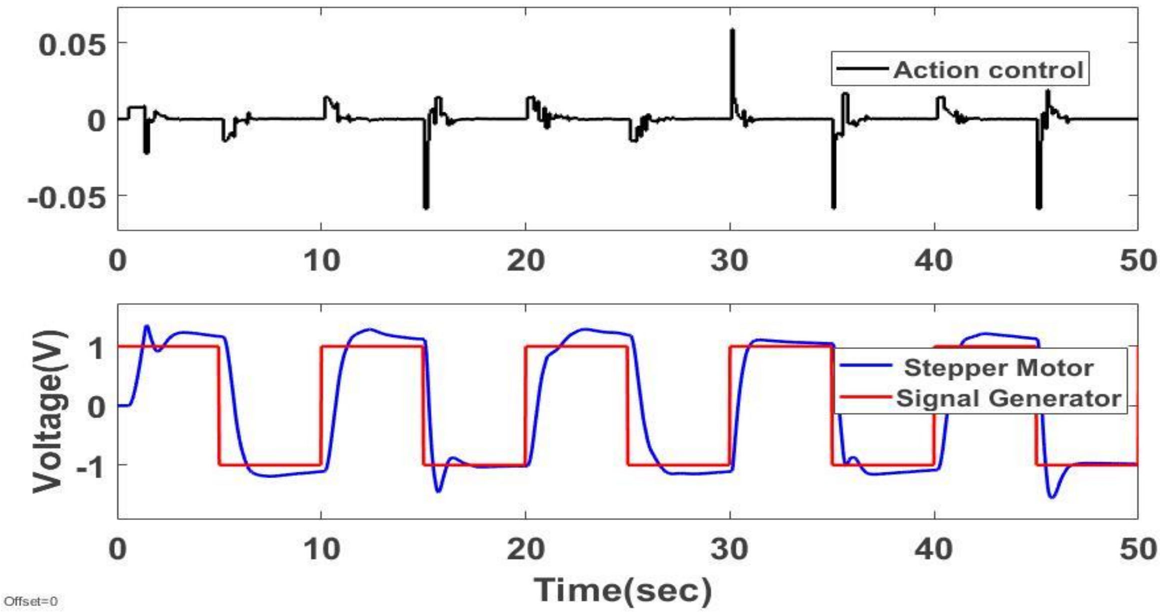


Fig 4.8: Stepper motor response and action control of WNCS with PID controller in high load at BWshare =0.9, Nodes=75

Table-4.2: Performance of the PID controller

Criteria / Case	TR	TS	MP%	ISE	Stability
No Load	0.5173	9.0599	8.7723%	0.3107	Stable
Medium Load	0.4979	13.2525	14.6948%	0.7891	Unstable
High load	0.4497	14.7903	46.4870%	1.6137	Unstable

The PID controller is simple and easy to implement and does not require storage memory when implemented. The PID controller is somewhat stable in the case of no load on the ZigBee wireless network and fails at the case of the medium load when the number of nodes is 100 and the high load at the number of nodes 75. Therefore, the PID controller not suitable to operate with WNCS and also the difficulty of determining the parameters of gain automatically.

4.4.2 Fuzzy PID Controller with WNCS Model

The Fuzzy Logic controller adjusts the gain parameters ($Kp, Ki,$ and Kd) of the PID controller, these parameters values will be variable when the system is running. The FL does not require mathematical operations but it depends on the smart operator of the system control to handle and monitor errors and change errors. The FL takes E and CE as input and gives Kp, Ki, and Kd gains to the PID controller to obtain an adaptive fuzzy PID controller. The sampling time for the stepper motor is 0.08sec. WNCS is simulated in MATLAB R2018b and using True Time tools. Fuzzy PID controller is used with a ZigBee network with two interference nodes as shown in Fig 4.9.

In the case of no-load, the response output of the Fuzzy PID controller at BWshare =0 and the number of nodes =0 is shown in Fig 4.10. Figure 4.11 shows the stepper motor response of the Fuzzy PID controller with medium load at BWshare 0.4 and the number of nodes is 150, in Fig 4.12 shows the stepper response with high load at BWshare 0.9 and number of node is 100. The performance of the Fuzzy PID controller shows in table 4.4.

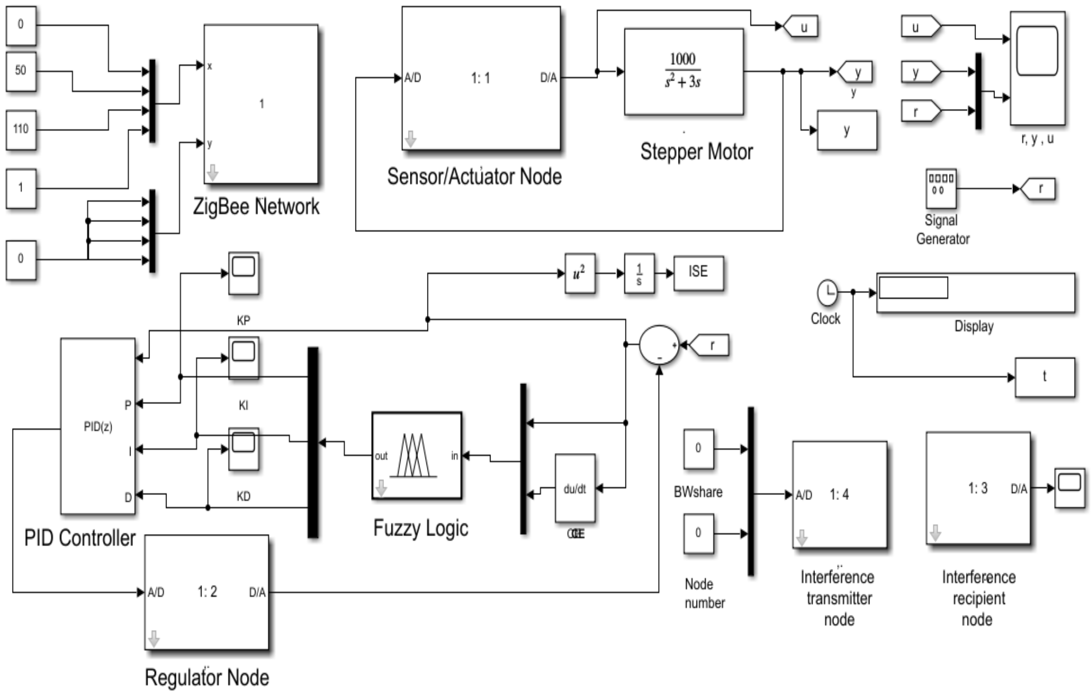


Fig 4.9: Fuzzy PID controller with a ZigBee NCS

In this work, the fuzzy logic rules are based on trial and error. Mamdani is used into the inference mechanism with the two input variables are E and CE using three triangular Membership functions with linguistic variables are N (Negative), Z (Zero), P (Positive) and five triangular Membership functions with a linguistic variable for each are NB (Negative Big), NS (Negative Small), Z (Zero), PS (Positive Small), and PB (Positive Big). The rules of fuzzy logic are being shown in table 4.3.

Table-4.3: Rules for the Fuzzy controller (Kp, Ki, Kd)

Kp

E CE	N	Z	P
N	NB	PB	PS
Z	NS	Z	NB
P	Z	PB	PS

Ki

E CE	N	Z	P
N	NB	PS	NB
Z	PB	NS	NS
P	PS	Z	PS

Kd

E CE	N	Z	P
N	NB	PB	PS
Z	NS	PS	Z
P	PB	Z	PB

Table-4.4: Fuzzy PID controller Performance

Criteria / Case	TR	TS	MP%	ISE	Stability
No load	0.3014	1.7458	2.2778%	0.2280	Stable
Medium load	0.4648	2.36194	16.3275%	1.2606	Unstable
High load	0.4591	9.7201	19.625%	1.5342	Unstable

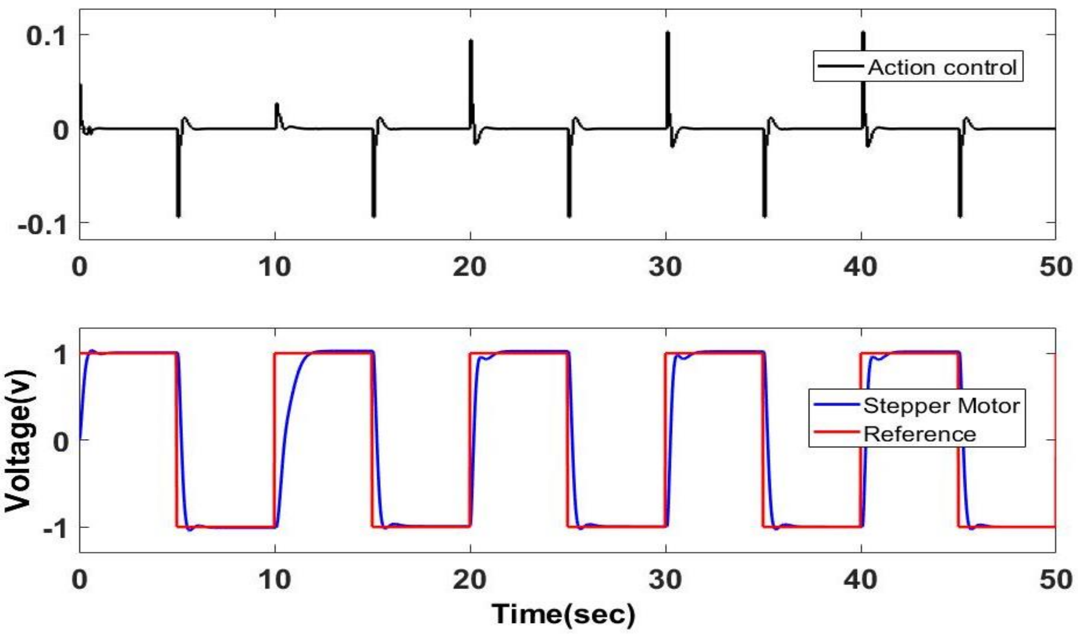


Fig 4.10: Stepper motor response and action control of WNCS with Fuzzy PID controller at BWshare =0, Nodes=0

Fuzzy PID Controller with stepper motor and ZigBee Network is to run for 50 seconds. The fuzzy logic is work to tuning the parameter of the PID controller when the system is running. The variable value of the parameter PID controller is shown in Fig 4.13. The performance of the fuzzy PID controller is shown in table 4.4.

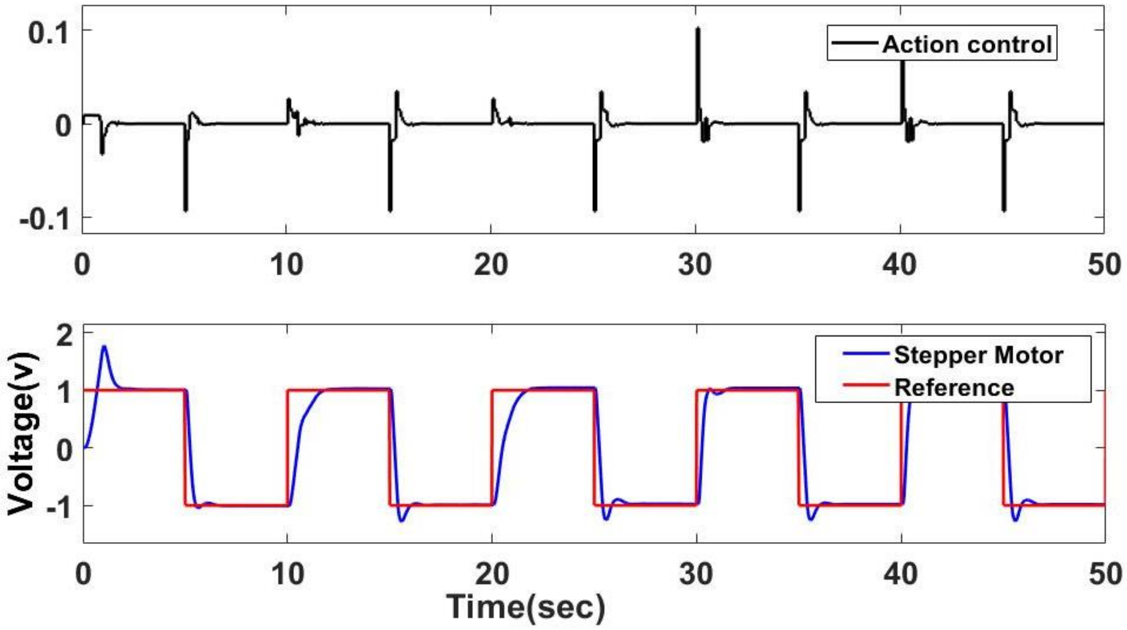


Fig 4.11: Stepper motor response and action control of WNCS with Fuzzy PID controller in medium load at BWshare =0.4, Nodes=150

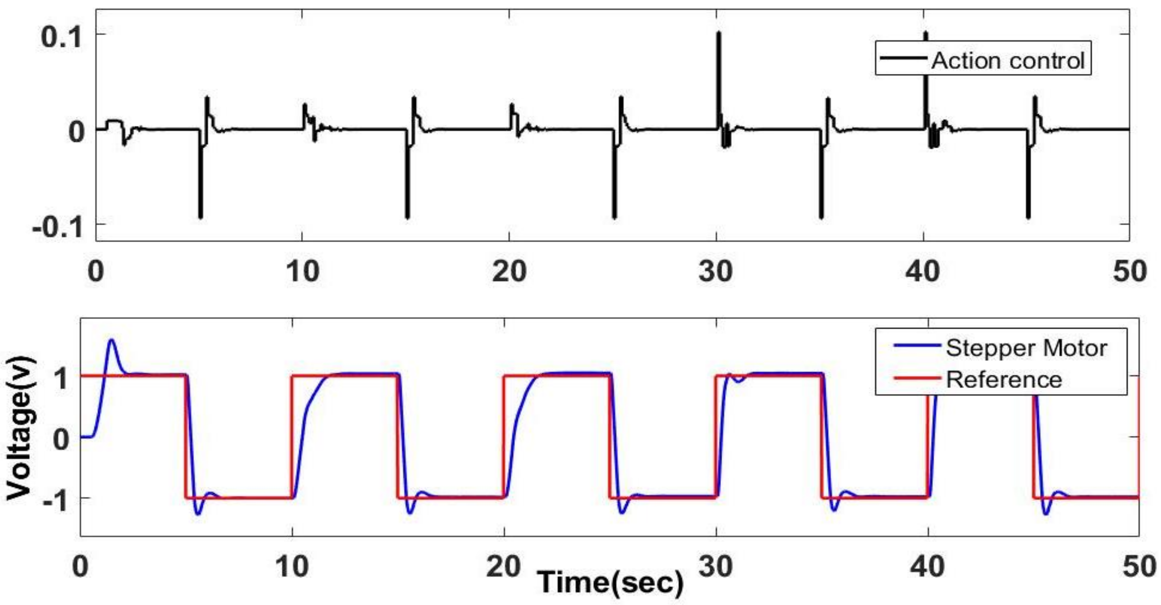
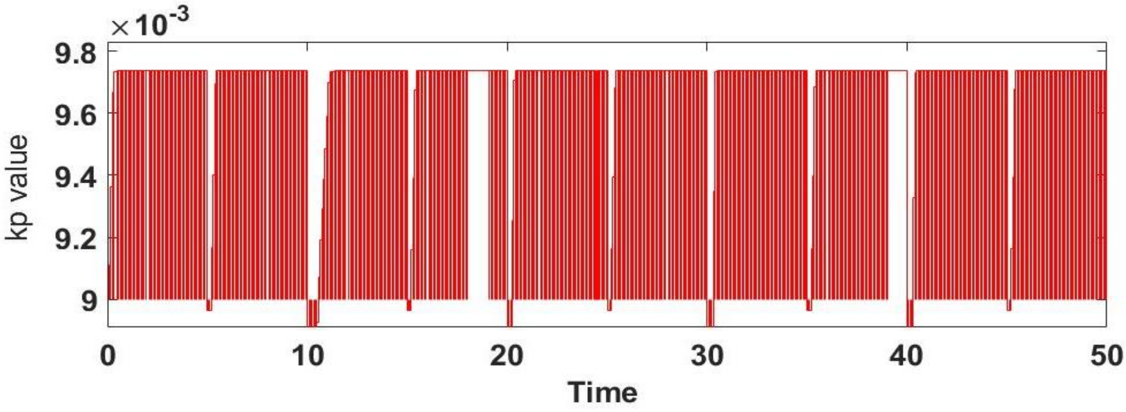
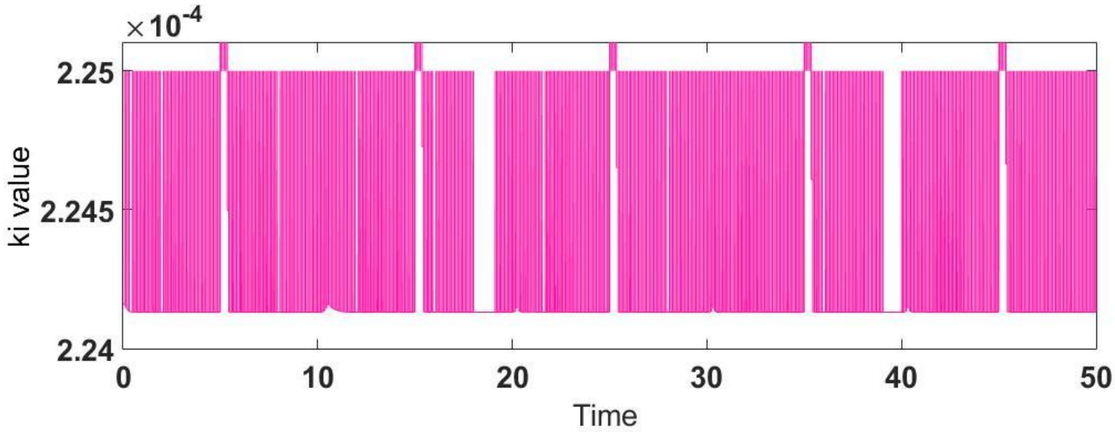


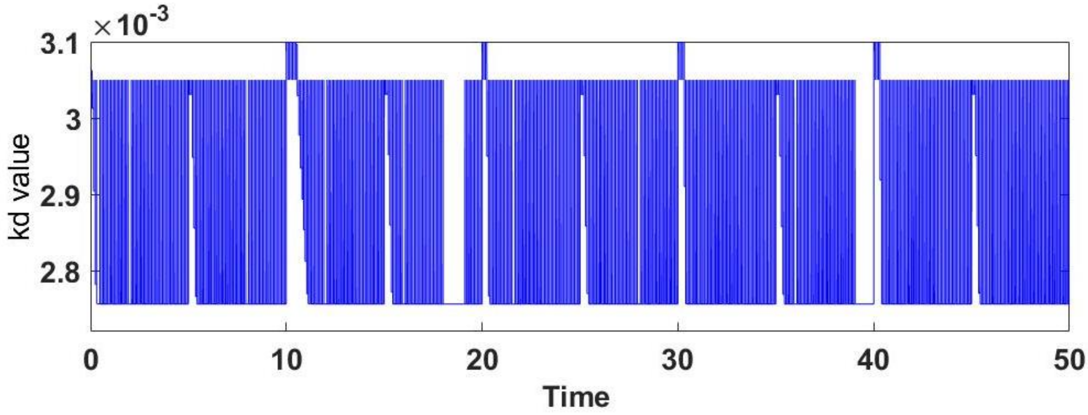
Fig 4.12: Stepper motor response and action control of WNCS with Fuzzy PID controller in high load at BWshare =0.9, Nodes=100



(a)



(b)



(c)

Fig 4.13: Variable value of the gain parameter of Fuzzy PID controller

$(a. kp, b. ki, c. kd)$

Although fuzzy controller has work advantages and does not require mathematical equations, the Fuzzy PID controller is stable in the no-load in the ZigBee wireless network and fails in the case of a medium load when the number of nodes is 150 and also fails in the case of high load in the number of nodes is100. Therefore, the Fuzzy PID controller is not suitable for working with WNCS and because it needs more experience to write fuzzy logic rules, thus the particle swarm optimization algorithm (PSO) is used to find the optimal rules that suitable for the proposed work WNCS.

4.4.3 PSO Fuzzy PID without Network

As mentioned previously in chapter three. The Fuzzy logic proposed in this thesis has two input variables which are E and CE and three output variables which are $Kp, Ki, and\ Kd$. The Fuzzy logic has five membership functions for input and output as mentioned before.

The number of a rule is the number of memberships raised to the exponent of inputs (5^2), so our work needs twenty-five rules, that Particle Swarm Optimization (PSO) finds optimal rules. The proposed PSO uses to obtained optimal rules for the Fuzzy PID controller with fifty swarms, seventy-five of particles, and one hundred iterations. The PSO fitness characteristic is shown in Fig 4.14. The PSO is work to obtain optimal rules as in table 4.5. Table 4.6 shows the performance of the PSO fuzzy PID without a network.

Table-4.6: Performance of the PSO Fuzzy PID controller without a network

TR	TS	MP	ISE	Fitness function
0.5682	1.1607	0.4519 %	0.1562	675.0256

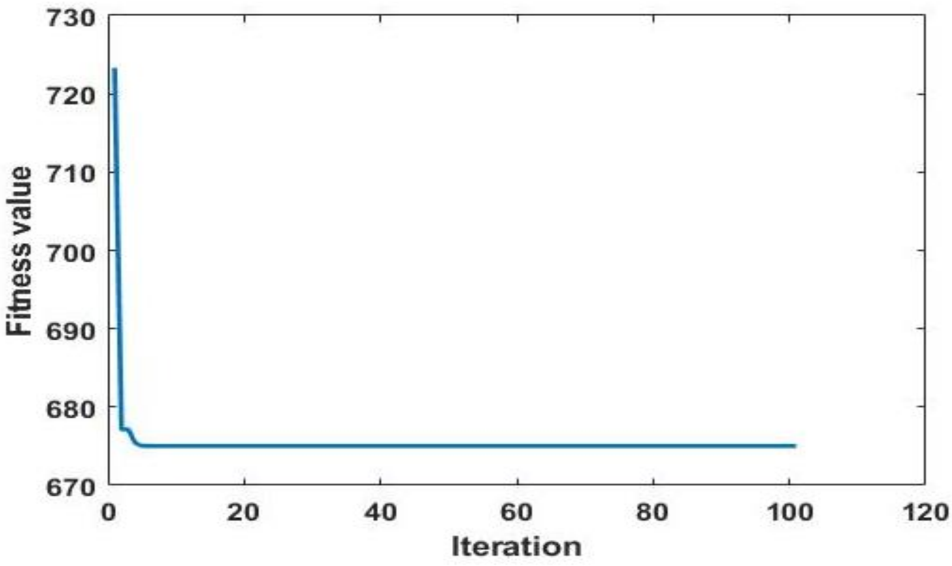


Fig 4.14: The PSO fitness characteristic

Figure 4.15 illustrates the design of PSO fuzzy PID with executing time for 50 seconds. The stepper motor response and action control of the PSO Fuzzy PID controller without a network are shown in Fig 4.16. The Fuzzy adaptive PID controller to give gain parameter in Fig 4.17 is being shown the variable value of the parameter for the PID controller when the system is running.

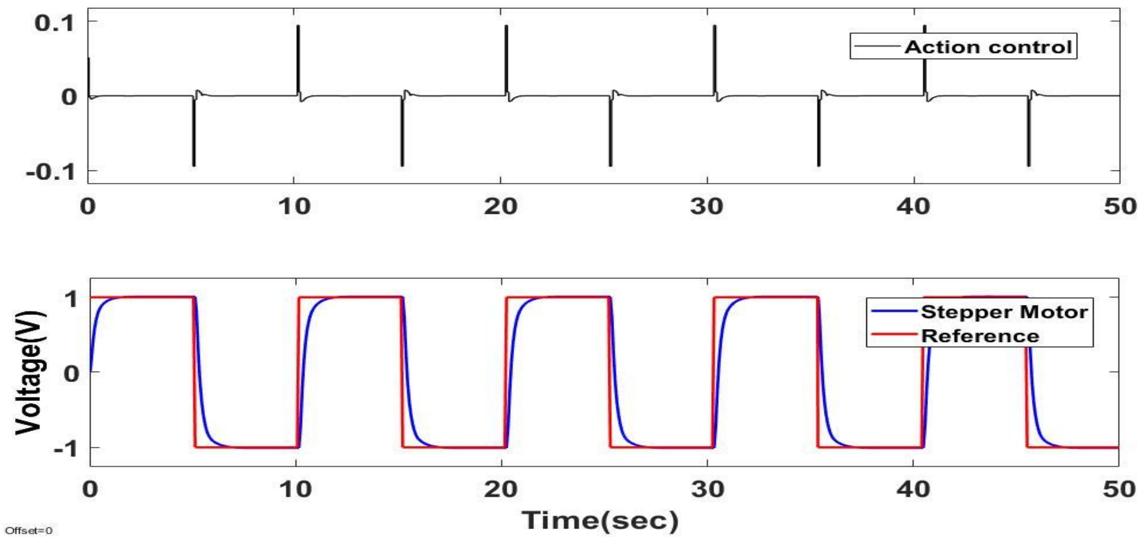


Fig 4.16: Stepper motor response and action control of PSO Fuzzy PID controller (without network)

Table-4.5: Results of optimized fuzzy rules using PSO

1. IF E==NB & CE==NB THEN KP=PB, KI=NM, KD=NM
2. IF E==NB & CE==NM THEN KP=NM, KI=NB, KD=Z
3. IF E==NB & CE==Z THEN KP=NB, KI=PB, KD=Z
4. IF E==NB & CE==PM THEN KP=PB, KI=Z, KD=PM
5. IF E==NB & CE==PB THEN KP=PM, KI=PB, KD=NM
6. IF E==NM & CE==NB THEN KP=PB, KI=Z, KD=PM
7. IF E==NM & CE==NM THEN KP=PB, KI=PM, KD=PB
8. IF E==NM & CE==Z THEN KP=NM, KI=PM, KD=NB
9. IF E==NM & CE==PM THEN KP=Z, KI=PB, KD=PB
10. IF E==NM & CE==PB THEN KP=PM, KI=PB, KD=PB
11. IF E==Z & CE==NB THEN KP=PB, KI=PB, KD=Z
12. IF E==Z & CE==NM THEN KP=NB, KI=PM, KD=NB
13. IF E==Z & CE==Z THEN KP=PB, KI=NB, KD=NB
14. IF E==Z & CE==PM THEN KP=PB, KI=PB, KD=PM
15. IF E==Z & CE==PB THEN KP=PB, KI=PM, KD=PM
16. IF E==PM & CE==NB THEN KP=NM, KI=Z, KD=Z
17. IF E==PM & CE==NM THEN KP=PB, KI=NM, KD=PM
18. IF E==PM & CE==Z THEN KP=PM, KI=NB, KD=PM
19. IF E==PM & CE==PM THEN KP=NB, KI=NM, KD=Z
20. IF E==PM & CE==PB THEN KP=Z, KI=NM, KD=PB
21. IF E==PB & CE==NB THEN KP=Z, KI=PB, KD=PM
22. IF E==PB & CE==NM THEN KP=PB, KI=NB, KD=Z
23. IF E==PB & CE==Z THEN KP=NB, KI=Z, KD=Z
24. IF E==PB & CE==PM THEN KP=PM, KI=Z, KD=PB
25. IF E==PB & CE==PB THEN KP=Z, KI=NB, KD=Z

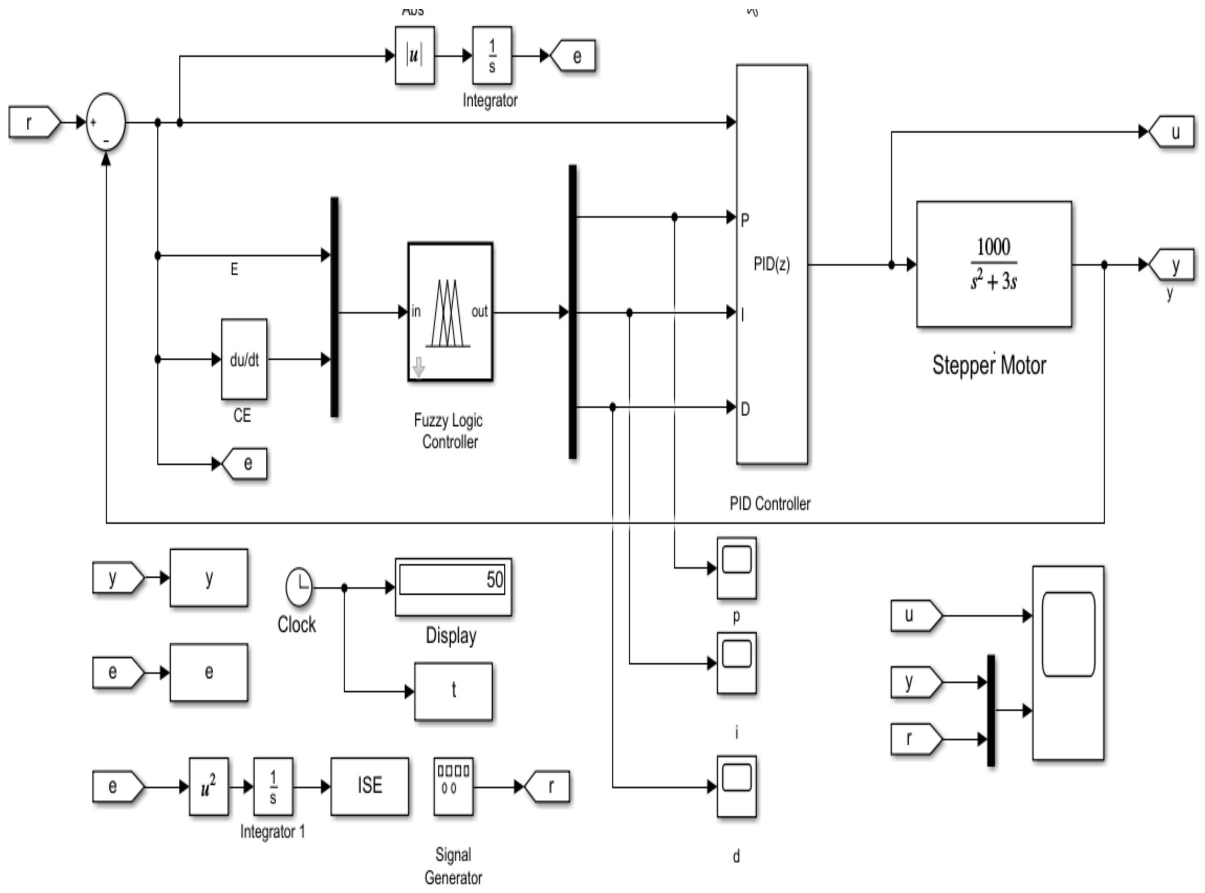
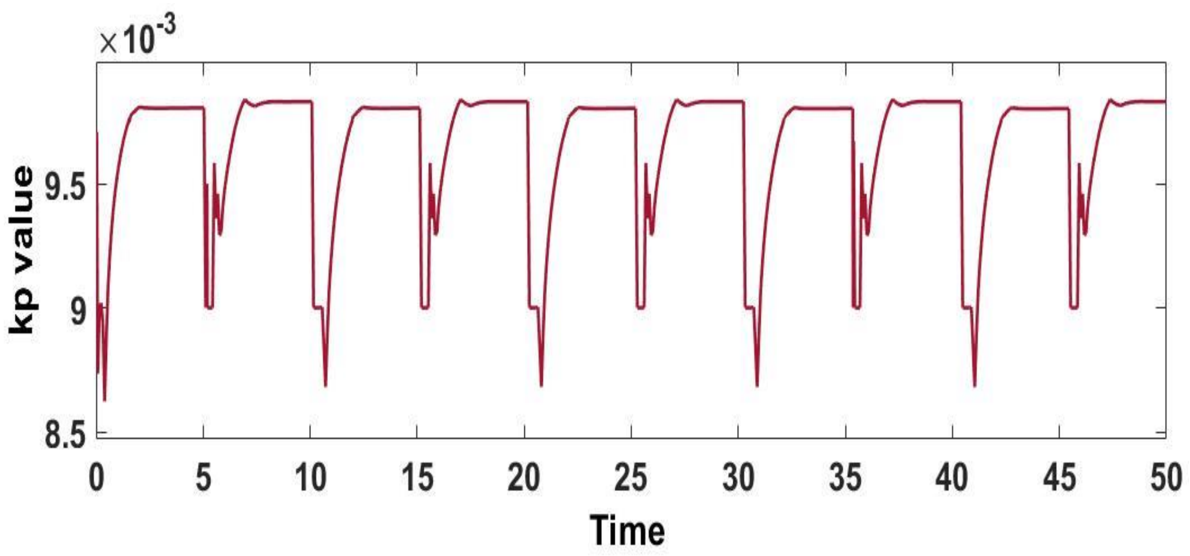
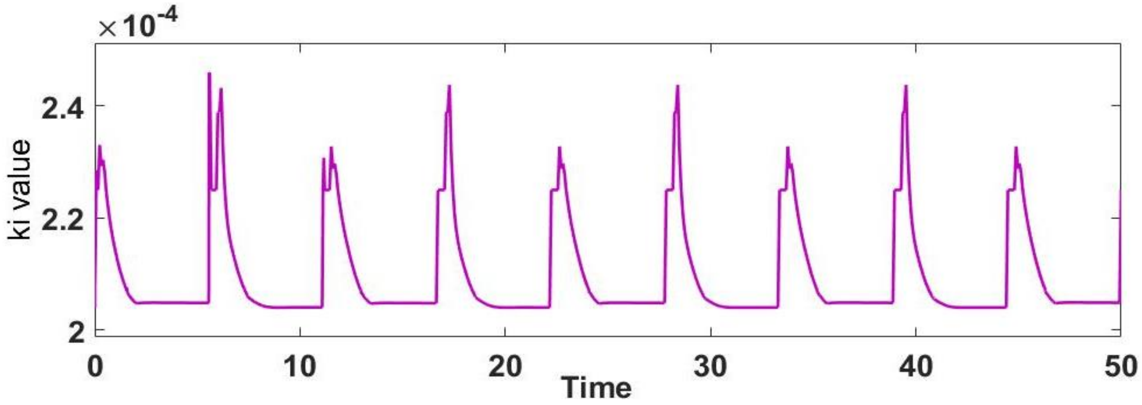


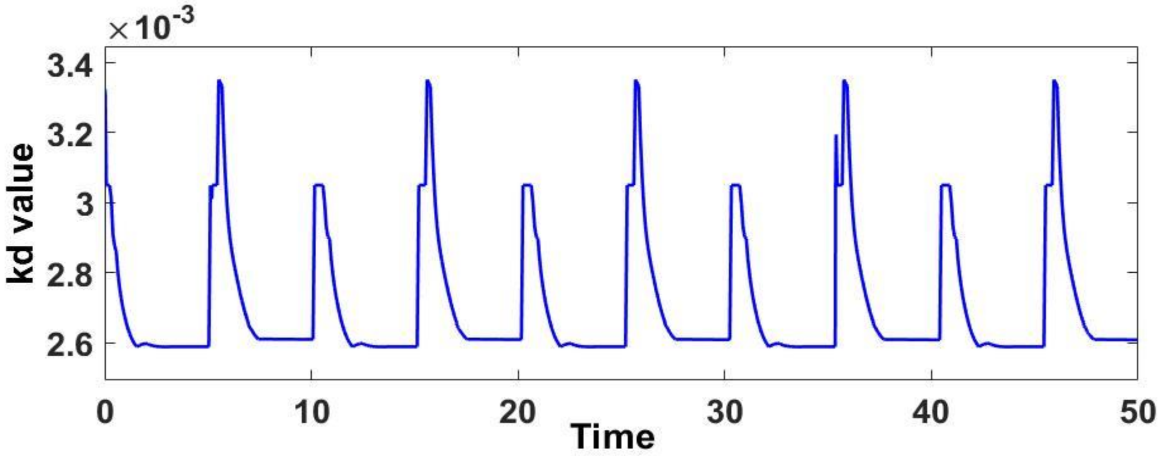
Fig 4.15: The PSO Fuzzy PID controller (without Network)



a: kp



$b: ki$



$c: kd$

Fig 4.17: Variable value of the parameter PID (without network)
($\boldsymbol{a}: \boldsymbol{kp}, \boldsymbol{b}: \boldsymbol{ki}, \boldsymbol{c}: \boldsymbol{kd}$)

4.4.4 PSO Fuzzy PID with Network

The WNCS model proposed with the PSO Fuzzy PID controller is designed as shown in Fig 4.18. PSO algorithm works to find the optimal rules of the Fuzzy PID controller to control in a stepper motor. The sampling time for the stepper motor is 0.08sec, while the sampling time for interference will be studied in three stages 0.035, 0.5, and 1.

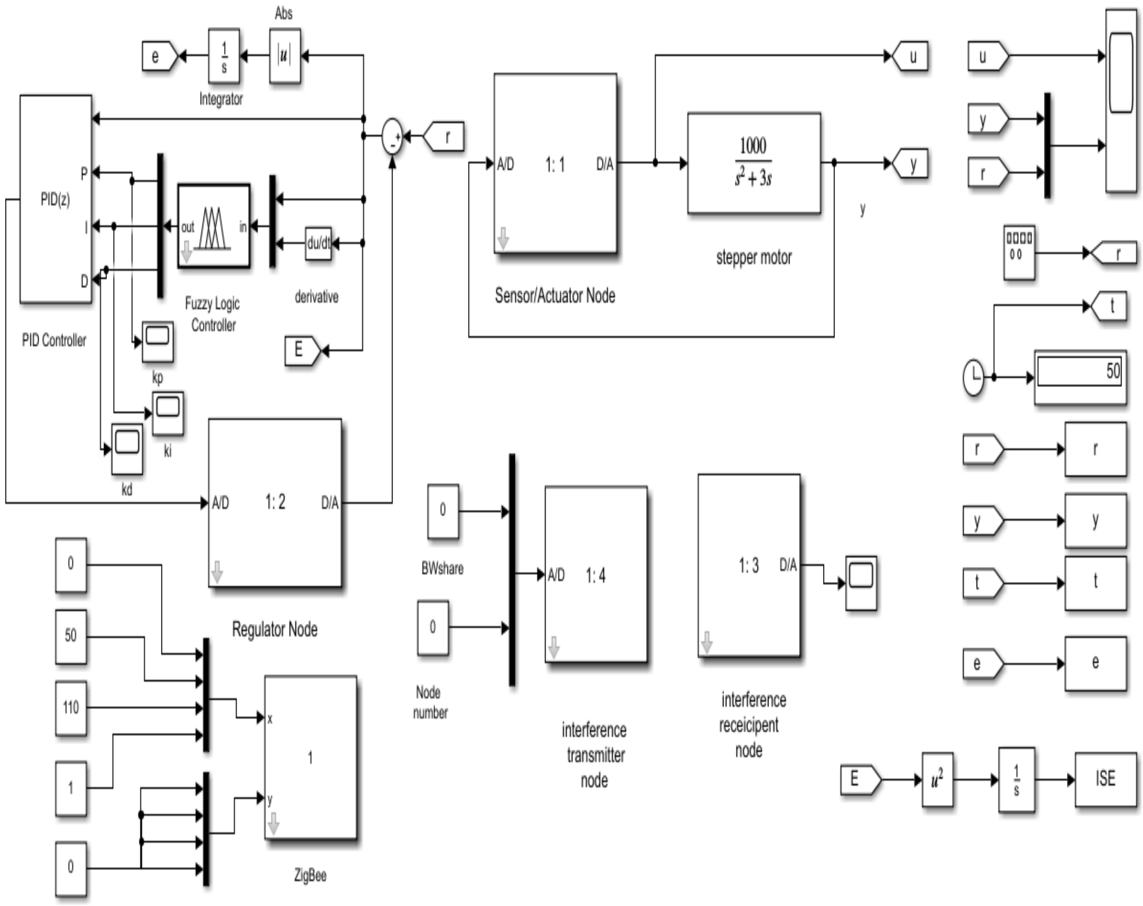


Fig 4.18: The proposed WNCS model with the PSO Fuzzy PID controller

4.4.4.1 Simulation of WNCS Model (with No – Load)

Figure 4.19 shows the response of stepper motor when BWshare is zero and the number of nodes is zero, with fifty-second simulation time. The total number of packets is 590. The RTT histogram and RTT values are shown in Figs 4.20 and 4.21 respectively. The average RTT is 0.0208 seconds that is calculated when a packet is sent from the sensor until it is received in the actuator, where this time is only for sending the control packet and sensor packet. The performance of the PSO Fuzzy PID controller is shown in table 4.7. The response of the PSO Fuzzy PID is stable in WNCS because the Fuzzy adaptive gain of the PID controller means the fuzzy logic is tuning gain of PID automatically, these variable values of the parameter PID shows in Fig 4.22.

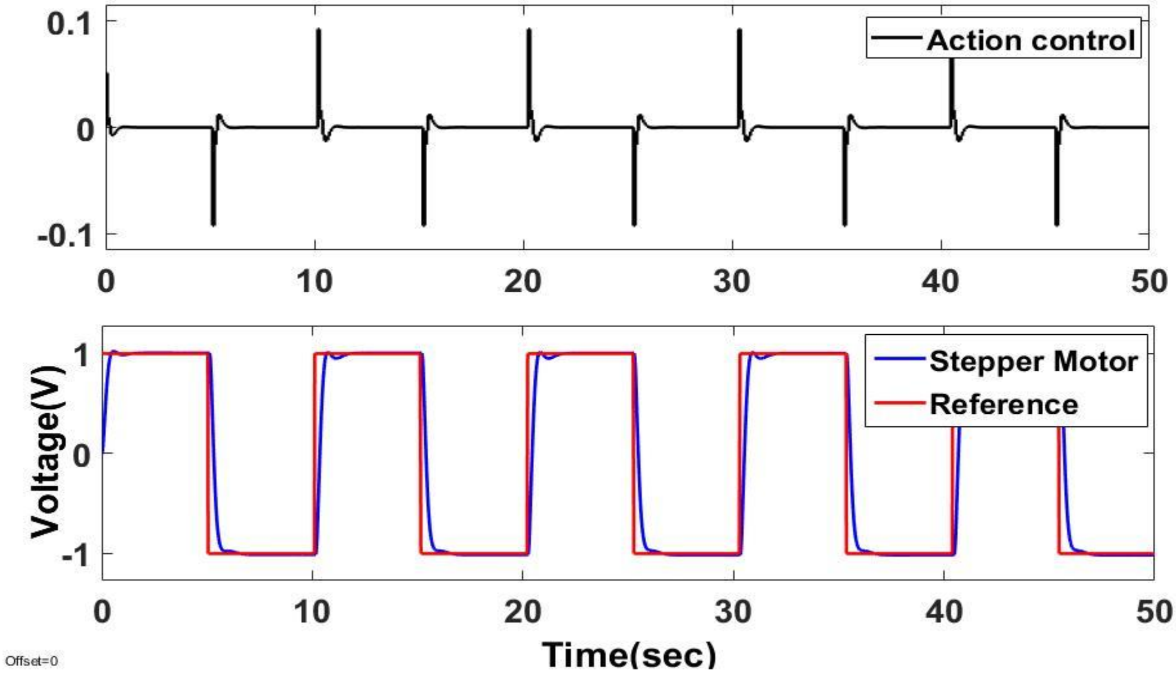


Fig 4.19: Stepper motor response of WNCS with PSO Fuzzy PID controller, at BWshare=0, and Node = 0 (in case of No-load).

Table-4.7: Performance of the PSO Fuzzy PID controller with network

TR	TS	MP	ISE	Fitness Function
0.278	0.606	2.28%	0.1803	1.0566

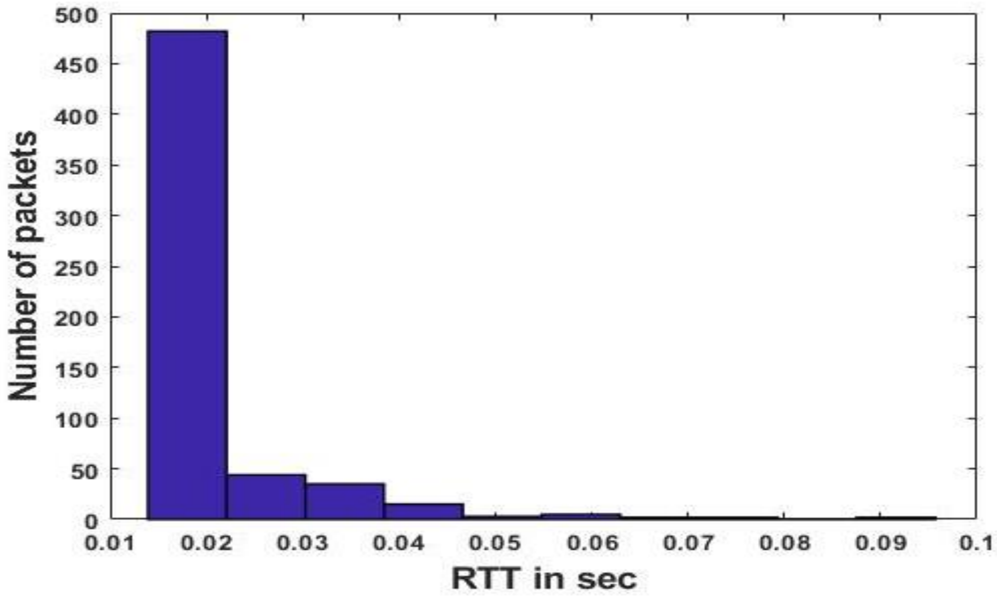


Fig 4.20: RTT histogram for the WNCS with PSO Fuzzy PID controller at BWshare=0 and Node=0 (in case of No-load)

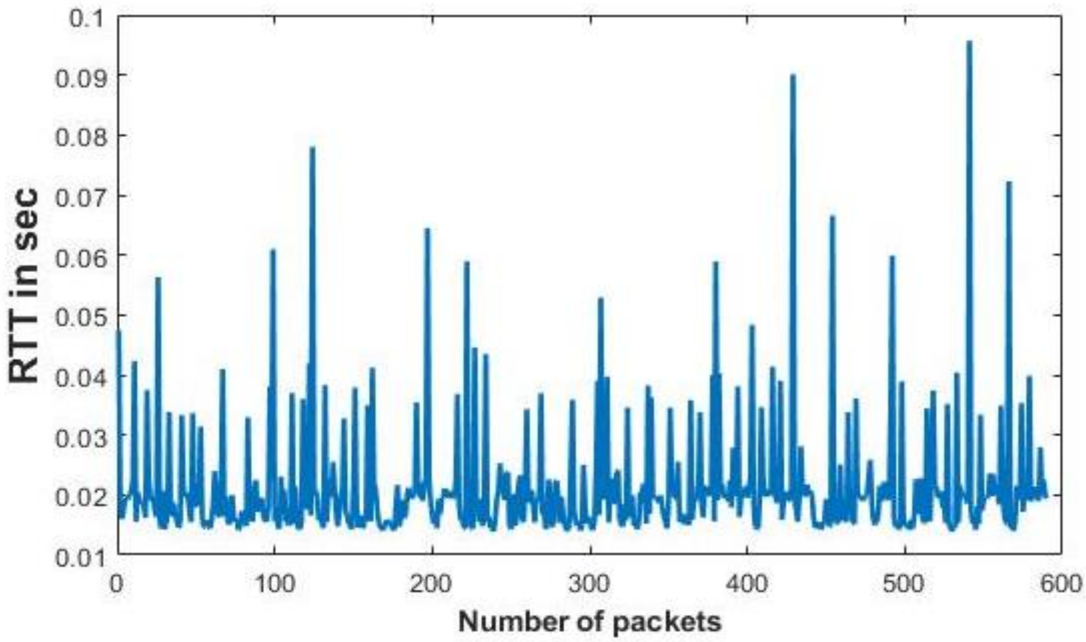
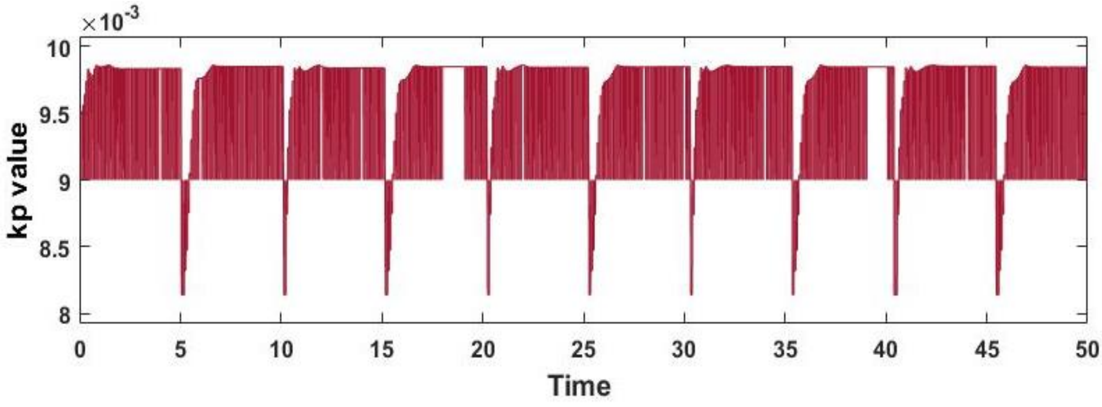
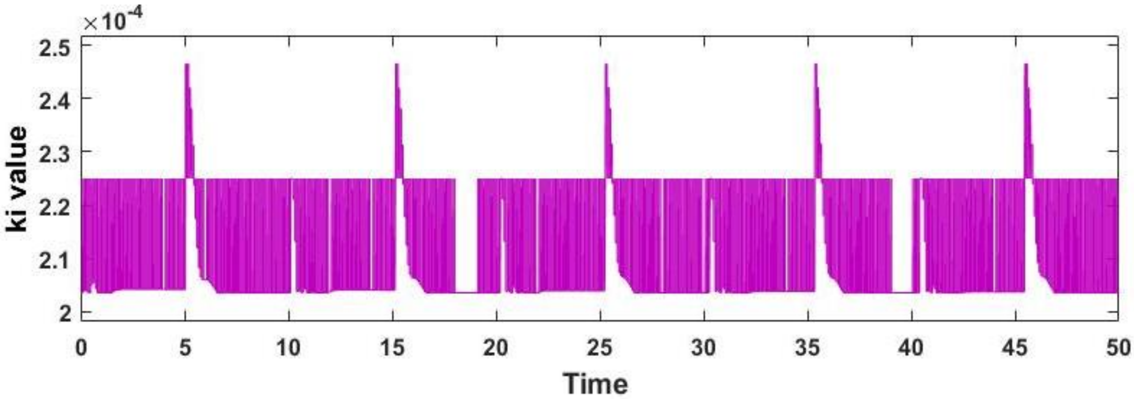


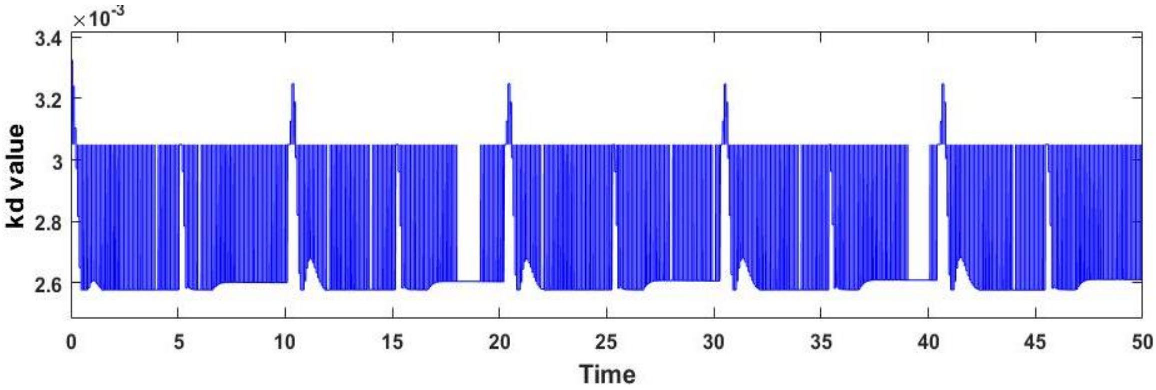
Fig 4.21: RTT values for WNCS with PSO Fuzzy PID controller at BWshare=0 and Node = 0 (in case of No-load)



(a)



(b)



(c)

Fig 4.22: Variable value of the parameter PID in the case with no load ($\boldsymbol{a}$: $\boldsymbol{kp}$, $\boldsymbol{b}$: $\boldsymbol{ki}$, $\boldsymbol{c}$: $\boldsymbol{kd}$)

4.4.4.2 Simulation of WNCS Model (with Load)

The proposed WNCS model will be tested with different sampling times for interference transmitter node which are 0.035, 0.5, and 1 second to test the control system viability stable while sampling time is 0.08sec.

On the other hand, the load on the ZigBee network will represent by BWshare which takes 0.4 or 0.9. At the interference transmitter node, the nodes number will simulate by equation 3.2 to test the stability of the response of the stepper motor, when the ZigBee network is loaded. The simulation of WNCS model with Load is divided into two cases as follows :

A- Medium Load: At BWshare is equal to 0.4 with sampling time in interference transmitter node which is [0.035, 0.5, and 1] second. Table 4.8 illustrates nodes number for loading ZigBee network and the performance of the WNCS with the PSO Fuzzy PID controller for the response of the stepper motor. When the sampling time of the interference transmitter is an increase, the nodes number will increase so the nodes number is proportional to the sampling time of the interference transmitter.

B- High Load: At BWshare is 0.9 with sampling time in interference transmitter node which is [0.035, 0.5, and 1] second. The performance of the stepper response motor for WNCS with the PSO Fuzzy PID controller is shown in table 4.9.

Table-4.8: WNCS model with a medium loaded network (at BWshare 0.4 and various numbers of nodes)

Sampling Time in Interference Transmitter Node	**Sampling time of system = 0.08 sec Medium Loaded network at BWshare 0.4**			
	Number of nodes	**MP%**	**TS**	**TR**
0.035	25	0.9001	1.5879	0.8478
0.5	250	0.9246	1.5943	0.9009
1	500	0.9257	1.5621	0.8987

Table-4.9: WNCS model with the high loaded network (BWshare 0.9 and various number of nodes)

Sampling Time in Interference Transmitter Node	**Sampling time of system = 0.08sec High loaded network at BWshare 0.9**			
	Number of nodes	**MP%**	**TS**	**TR**
0.035	10	1.006	1.647	0.833
0.5	100	1.2199	1.799	0.887
1	300	1.0360	1.8095	0.8971

In table 4.8, the nodes number is five hundred for 0.4 of BWshare while the nodes number is three-hundred for 0.9 of BWshare such as in table 4.9. The two cases (500 and 300 nodes), The largest number of nodes is taken, then the result of output response is shown, calculate the average and value of the RTT, and display the criteria of the system response performance remain stable :

1- In the case of a medium load (500 nodes) is tested. The WNCS contains regulator node, sensor/actuator nodes, and five hundred (500) emulated nodes that emulate in interference transmitter node and interference recipient node so the total nodes in WNCS are equal to 503 nodes. A simulated packet length describes in equation 3.2will be (503 ×0.4×133 byte× w) byte. In another word, the controller packet, sensor packet, and simulated packet will be sent via the wireless network. The system response stays stable with 503 nodes. The stepper motor response and action control of the PSO Fuzzy PID controller are shown in Fig 4.23. While Fig 4.24 shows the variable value of the gain parameters of the PID controller.

The total number of packets is 576. The RTT histogram and RTT values are shown in Fig 4.25 and 4.26 respectively. The average RTT is 0.0219 seconds that is calculated when a packet is sent from the sensor until it is received in the actuator, where this time is only for sending the control packet and sensor packet. In table 4.10 is shown the performance of the PSO Fuzzy PID controller.

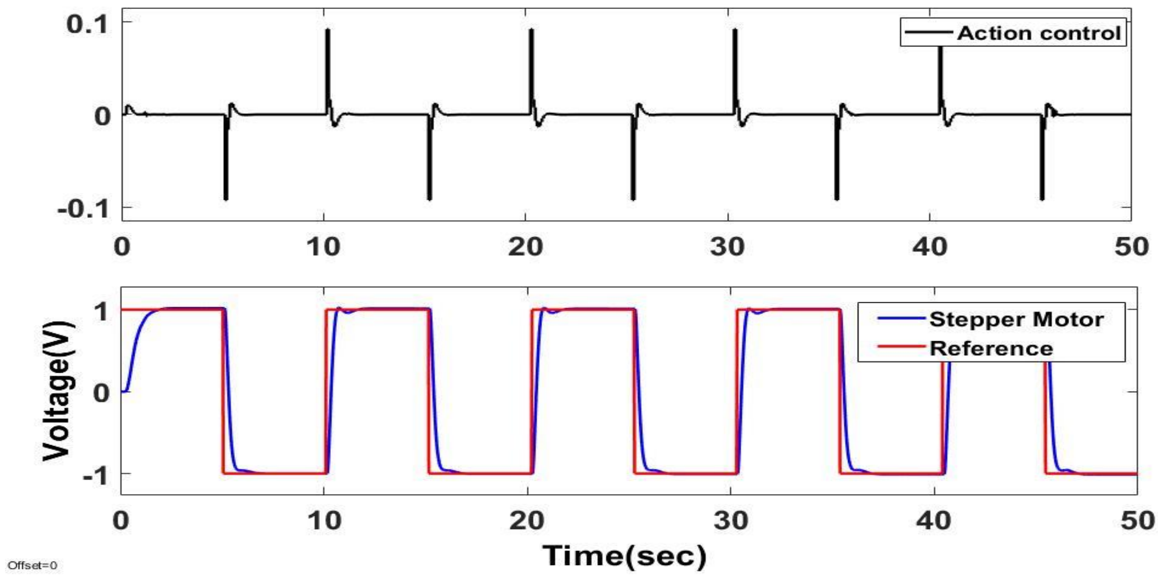
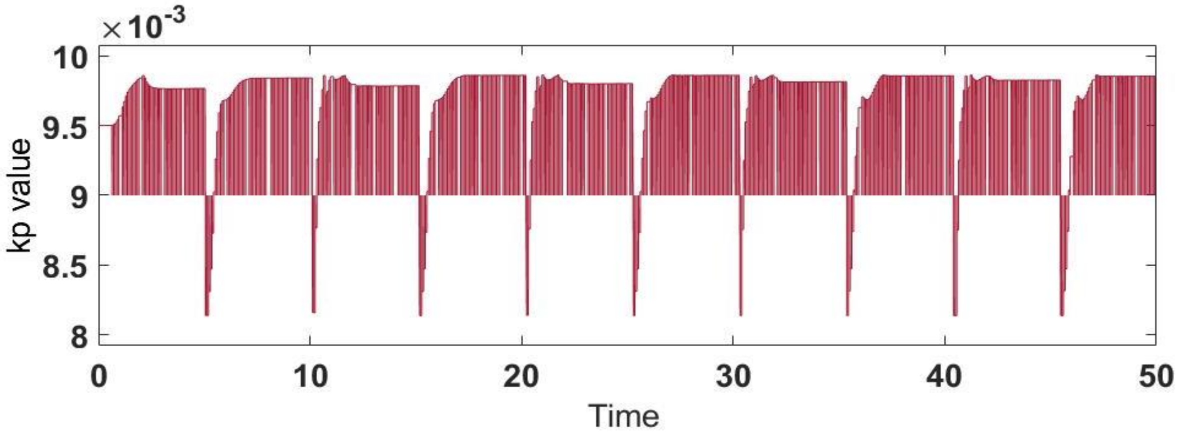
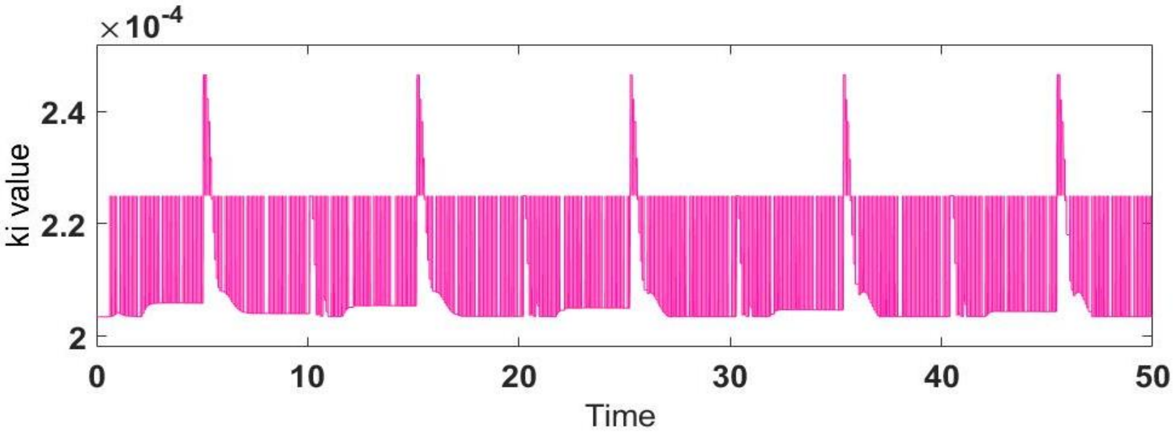


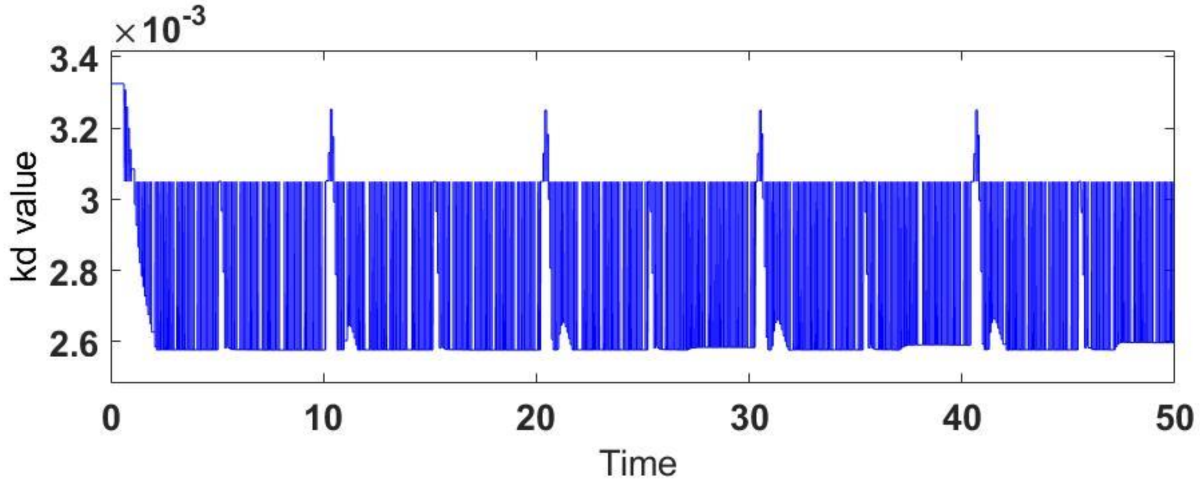
Fig 4.23: Stepper motor response and action control of WNCS with PSO Fuzzy PID controller, at BWshare=0.4, and N.of Node = 500 nodes.



a: kp



b: ki



c: kd

Fig 4.24: Variable value of the parameter PID (a: kp, b: ki, c: kd) in case medium load (500 nodes)

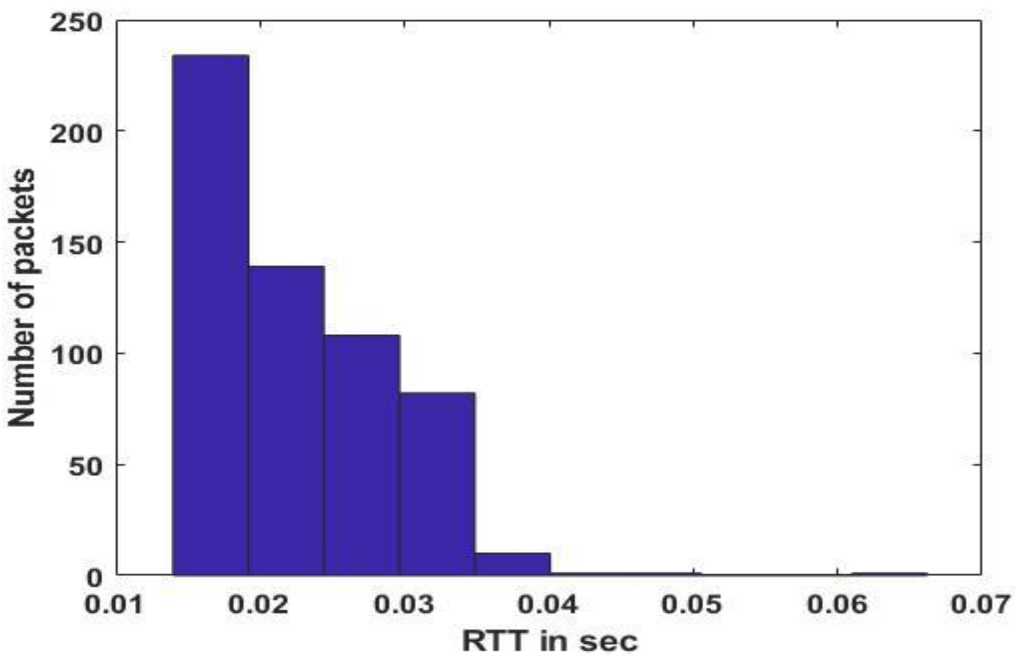


Fig 4.25: RTT histogram for the WNCS with PSO Fuzzy PID controller, at BWshare=0.4, and 500 nodes

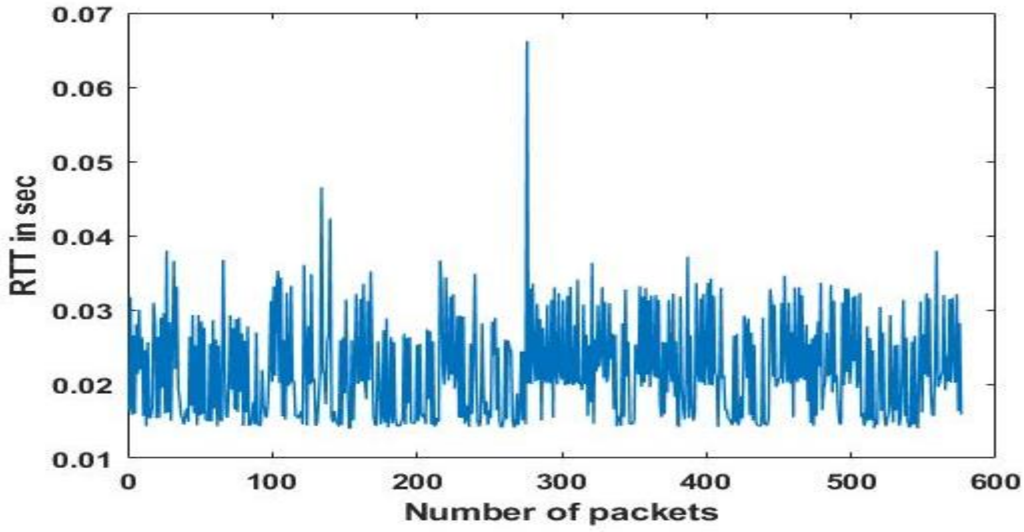


Fig 4.26: RTT values for the WNCS with PSO Fuzzy PID controller, at BWshare=0.4, and 500 nodes

Table-4.10: Performance of the PSO Fuzzy PID controller in case medium load

TR	TS	MP	ISE	Fitness function
0.8987	1.5621	0.9257%	0.8692	3.687

2- In the case of a high load (300 nodes) is tested. The total number of nodes calculated as mentioned in the first case, where total nodes in WNCS are equal to 303 nodes. A simulated packet length describes in equation 3.2 will be (303 ×0.4×133 × w) byte. The response output and action control of the PSO Fuzzy PID controller is shown in Fig 4.27. While Fig 4.28 shows of the variable value of the parameter PID ($a{:}\,kp, b{:}\,ki, c{:}\,kd$). The total number of packets is 563. The RTT histogram and RTT values are shown in Fig 4.29 and 4.30 respectively. The average RTT is 0.0216 seconds that is calculated when a packet is sent from the sensor until it is received in the actuator, where this time is only for sending the control packet and sensor packet. In table 4.11 is shown the performance of the PID controller.

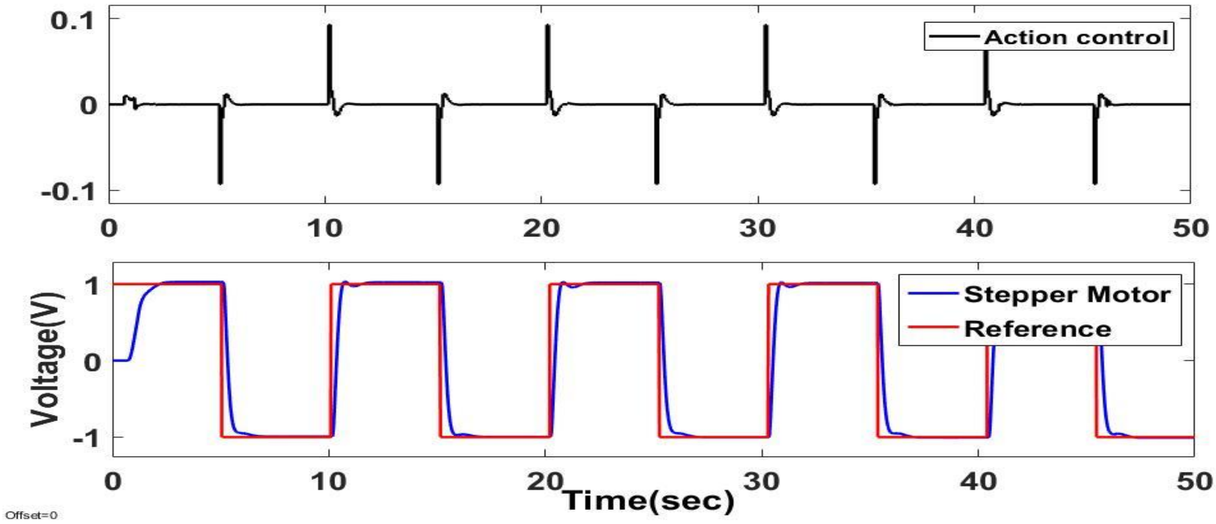
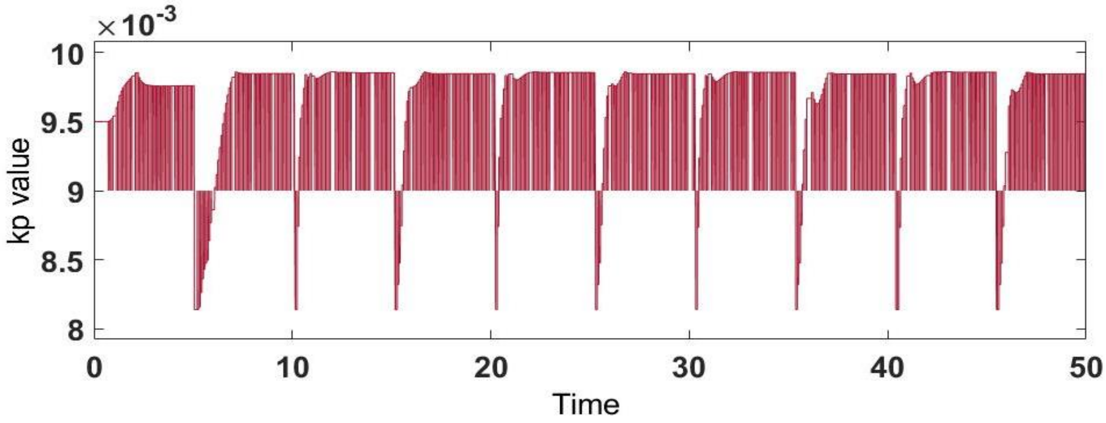
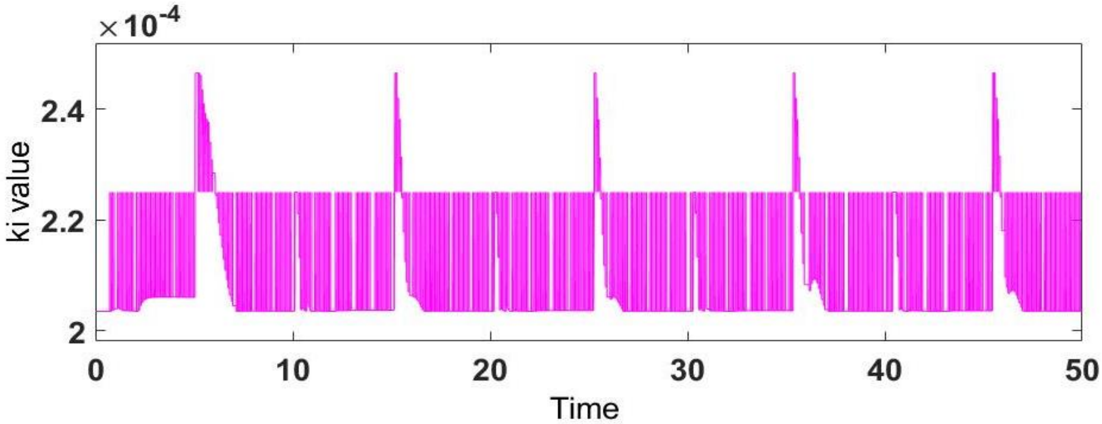


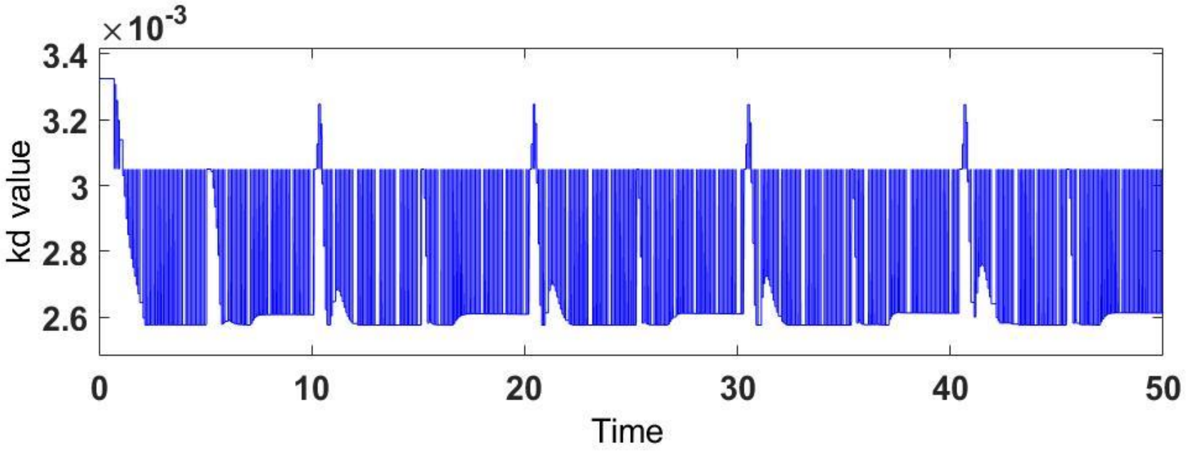
Fig 4.27: The response output and action control of WNCS with PSO Fuzzy PID controller, at BWshare=0.9, and N.of Node = 300 nodes



a: kp



b: ki



$c: kd$

Fig 4.28: Variable value of the parameter PID $(a: kp, b: ki, c: kd)$ in case high load (300 nodes)

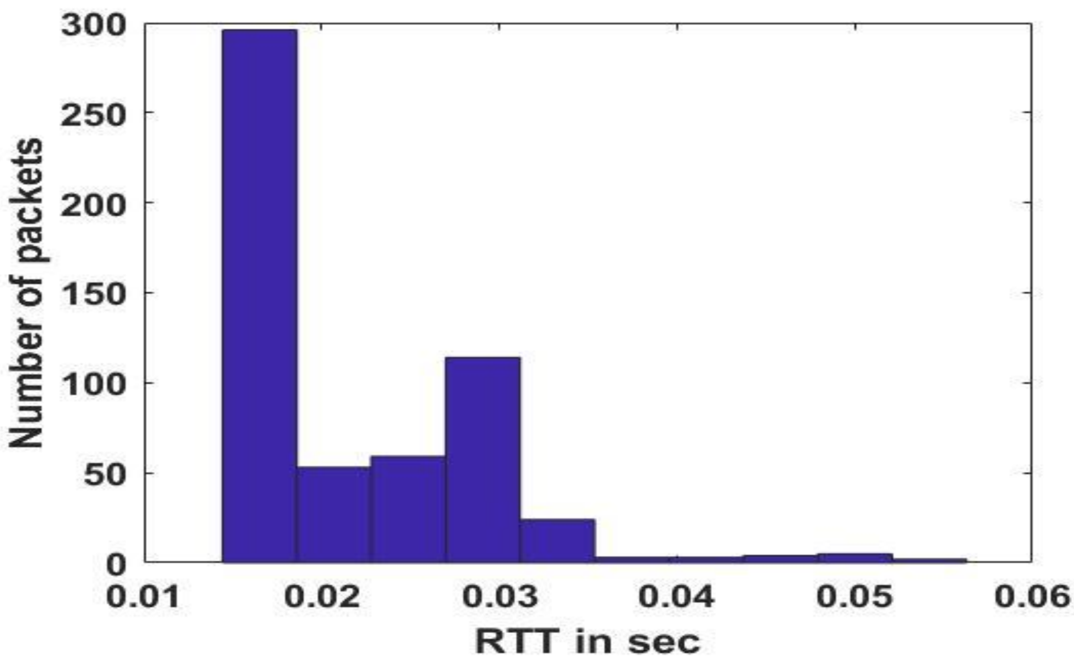


Fig 4.29: RTT histogram for WNCS with PSO Fuzzy PID controller at 300 nodes

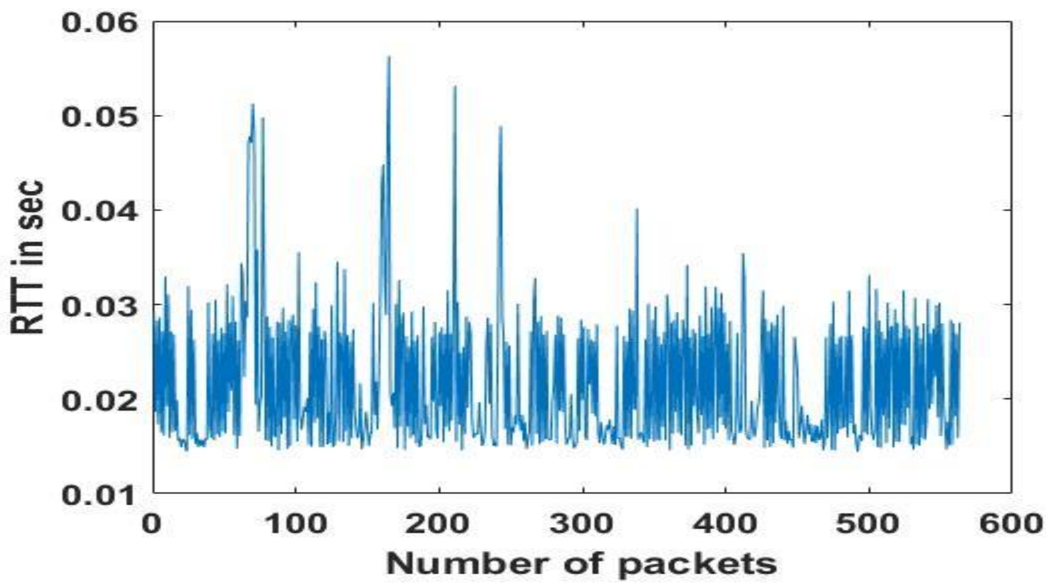


Fig 4.30: RTT values for WNCS with PSO Fuzzy PID controller at 300 nodes

Table- 4.11: The Performance of the PSO Fuzzy PID controller at 300 nodes

TR	TS	MP	ISE	Fitness function
0.8971	1.8095	1.0360%	1.109	4.2625

4.5 Packet Loss Test

The medium load case is tested when the sample time is 0.08 sec and the number of nodes is 500 at the value of BWshare = 0.4, thus random values from the probability of packet loss are taken to test the system performance.

1- **10% packet loss**
At the fifty-second simulation time, the response output and action control of WNCS with the PSO Fuzzy PID controller is shown in Fig 4.31. The average RTT is 0.0202, and the number of the packet is 571. The RTT histogram and RTT values are shown in Fig 4.32 and 4.33 respectively.

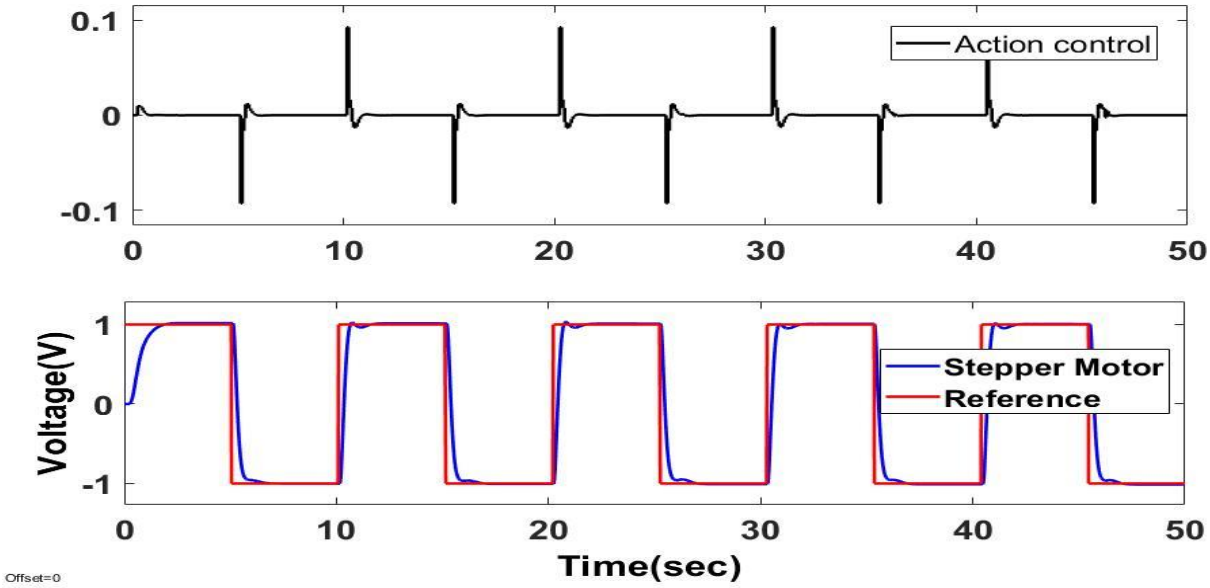


Fig 4.31: The response of WNCS with PSO Fuzzy PID at 10% packet loss

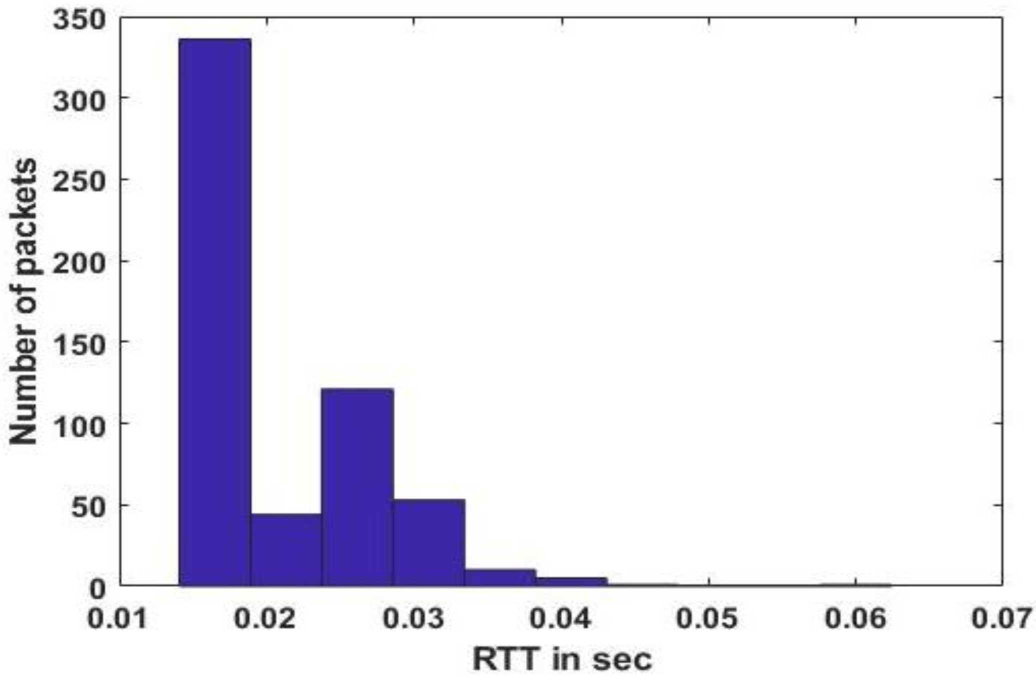


Fig 4.32: RTT histogram forWNCS the PSO Fuzzy PID at 10% packet loss

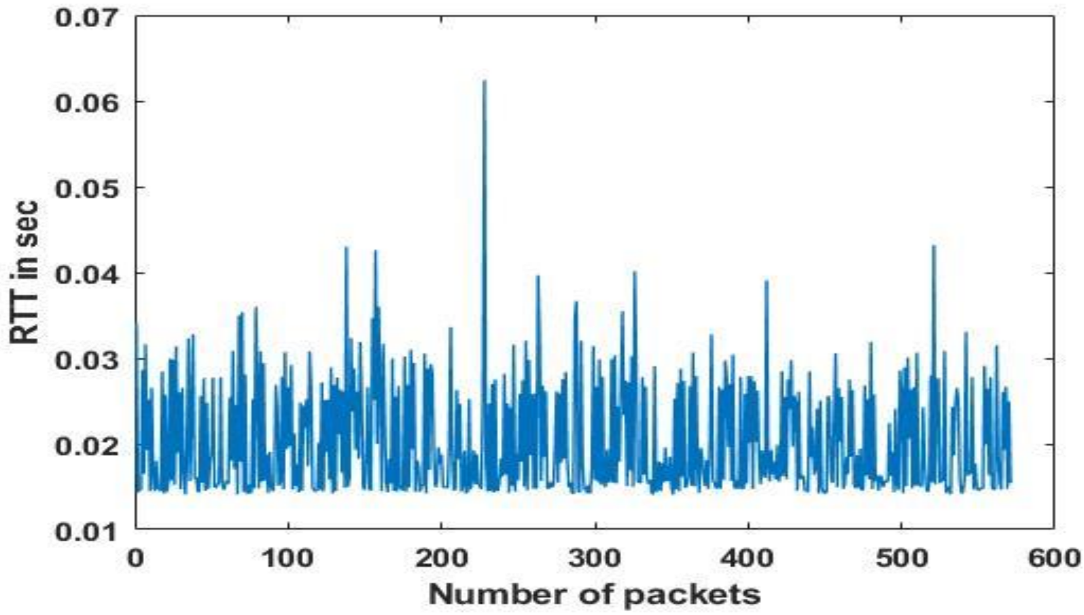


Fig 4.33: RTT values for WNCS with PSO Fuzzy PID at 10% packet loss

2- **20% packet loss**

The response output and action control of the PSO Fuzzy PID controller is shown in Fig 4.34 with 50-second simulation time. The average RTT is 0.0216, and the number of the packet is 561. The RTT histogram and RTT values are shown in Fig 4.35 and 4.36 respectively.

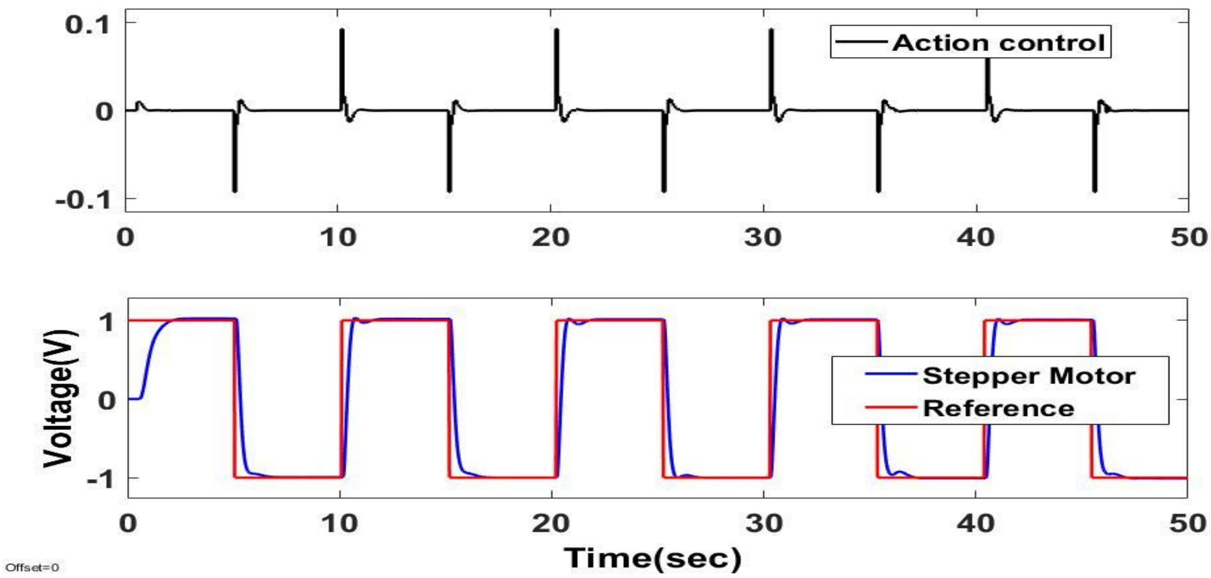


Fig 4.34: The response of WNCS with PSO Fuzzy PID at 20% packet loss

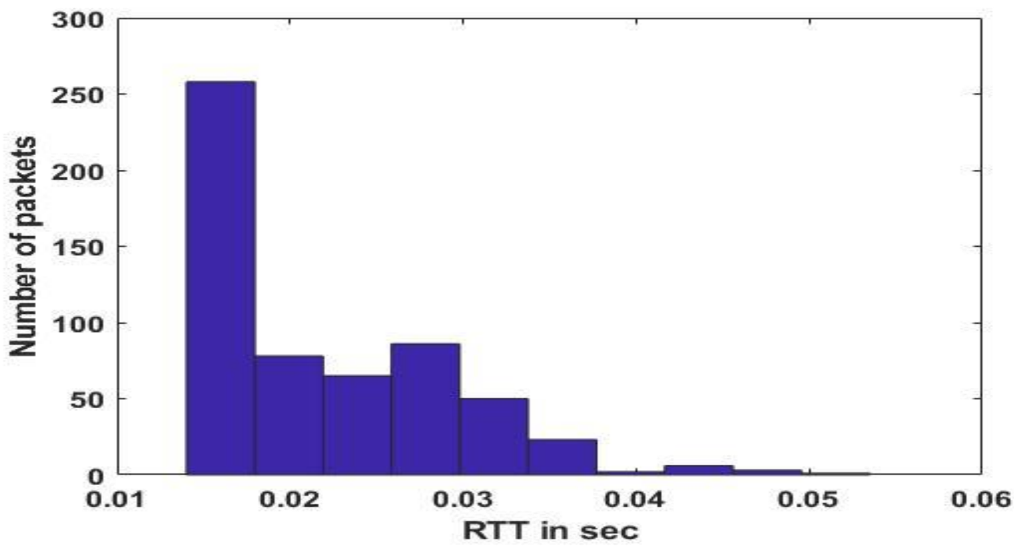


Fig 4.35: RTT histogram for WNCS with PSO Fuzzy PID at 20% packet loss.

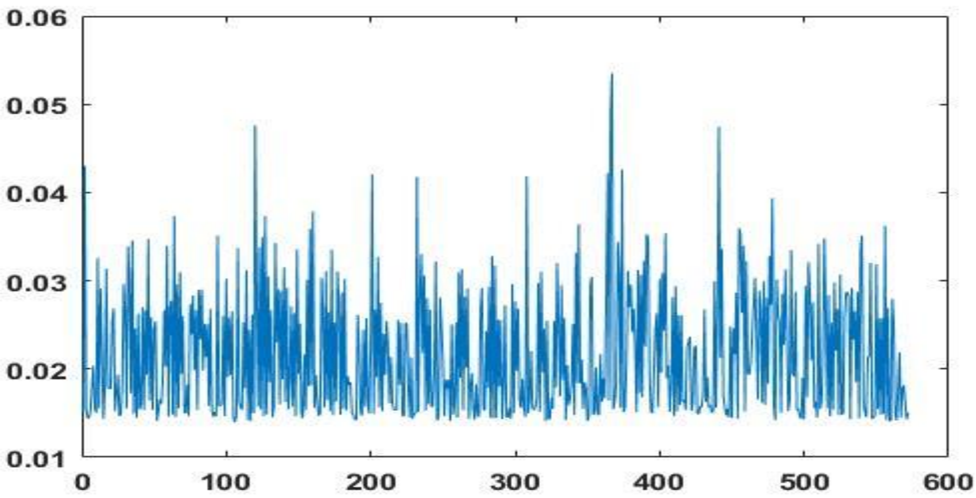


Fig 4.36: RTT values for WNCS with PSO Fuzzy PID at 20% packet loss

3- 40% packet loss

The response output and action control of the PSO Fuzzy PID controller is shown in Fig 4.37 with 50-second simulation time. The average RTT is 0.0246, and the number of the packet is 530. The RTT histogram and RTT values are shown in Fig 4.38 and 4.39 respectively. In table 4.12 shows the performance of the PSO Fuzzy PID controller with 10% and 20% and 40% packet loss.

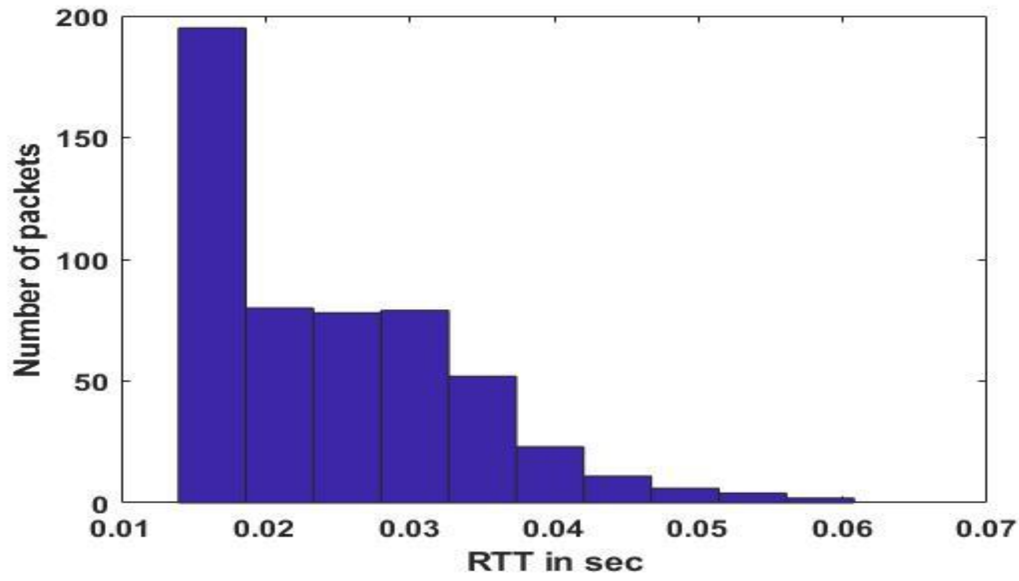


Fig 4.38: RTT histogram for WNCS with PSO Fuzzy PID at 40% packet loss

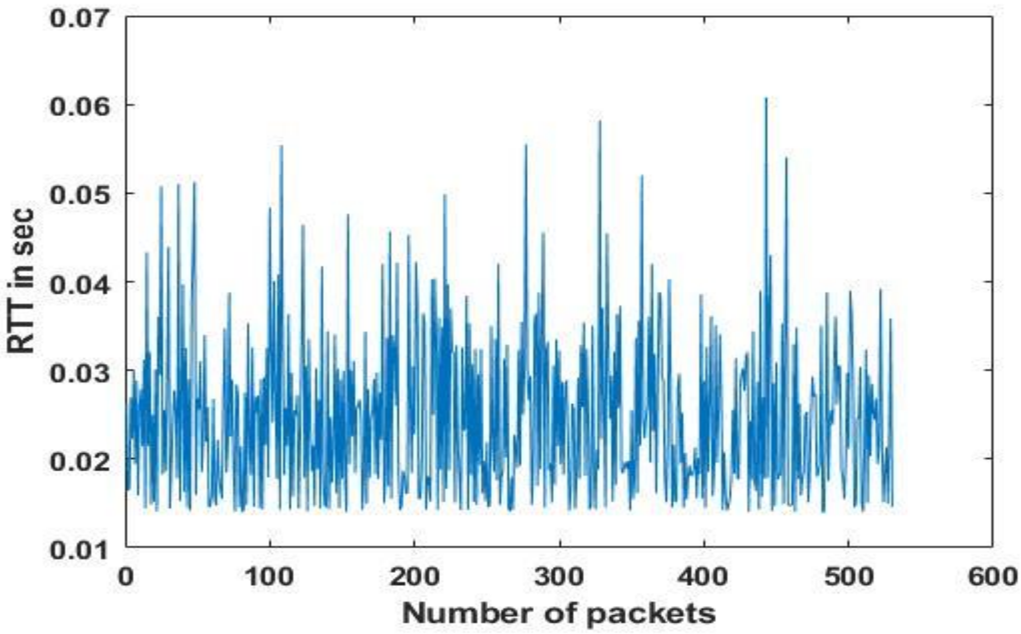


Fig 4.39: RTT values for WNCS with PSO Fuzzy PID at 40% packet loss

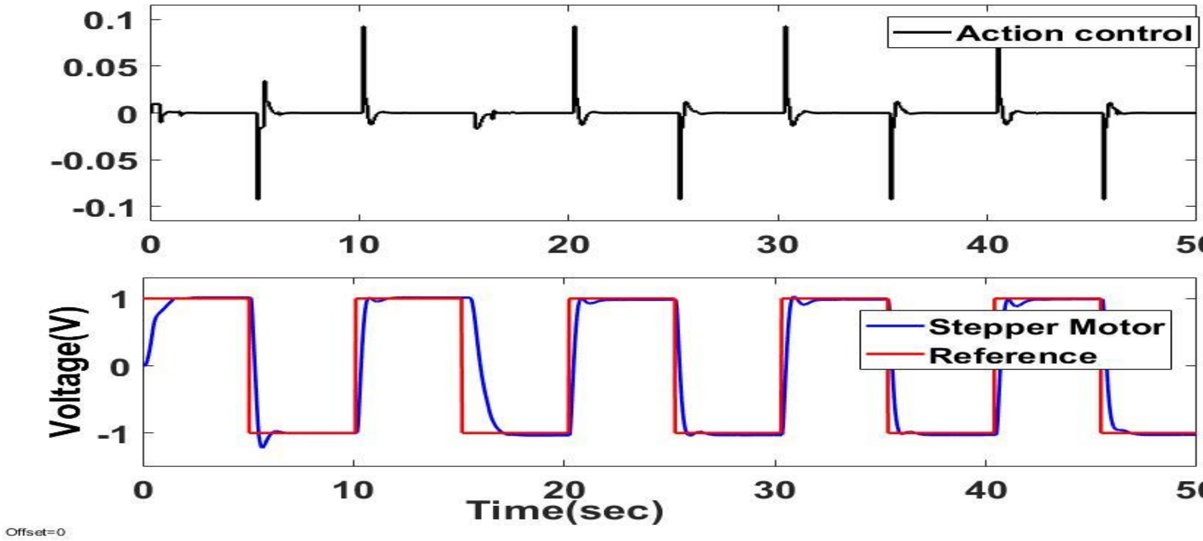


Fig 4.37: The response of WNCS with the PSO Fuzzy PID at 40% packet loss

Table-4.12: The Performance of WNCS with various packet loss

Criteria / packet loss	TR	TS	MP	ISE	Fitness function
10% packet loss	0.8551	1.7405	1.1319%	0.6313	3.110
20% packet loss	0.8589	1.7407	1.1422%	0.62424	3.0994
40% packet loss	0.8001	2.0076	1.5395%	0.9506	2.358

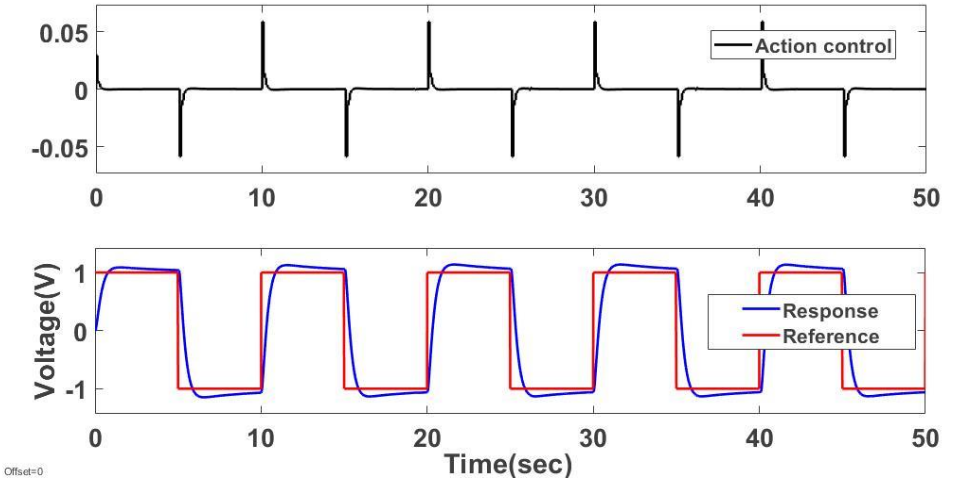
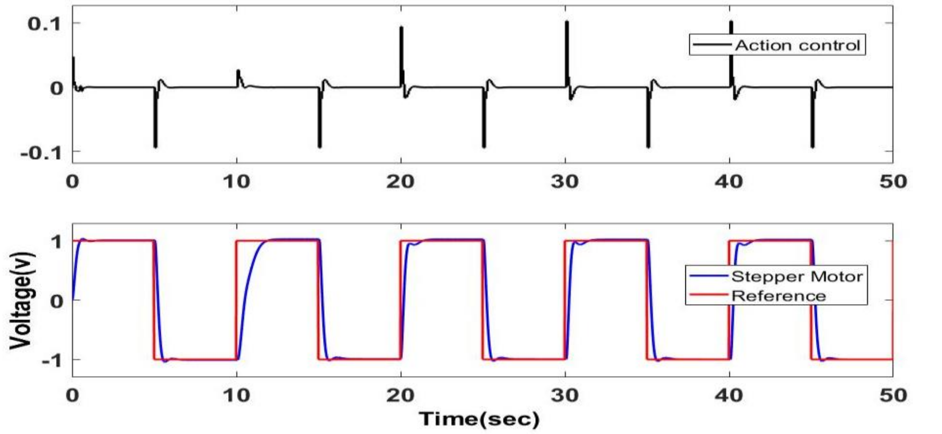
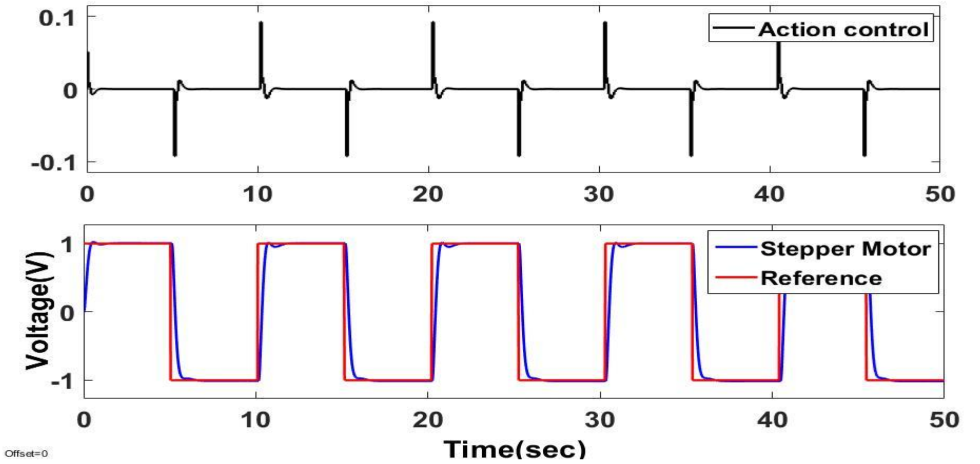
The proposed PSO Fuzzy PID controller is tested in the case of a medium load with three different packet loss values:

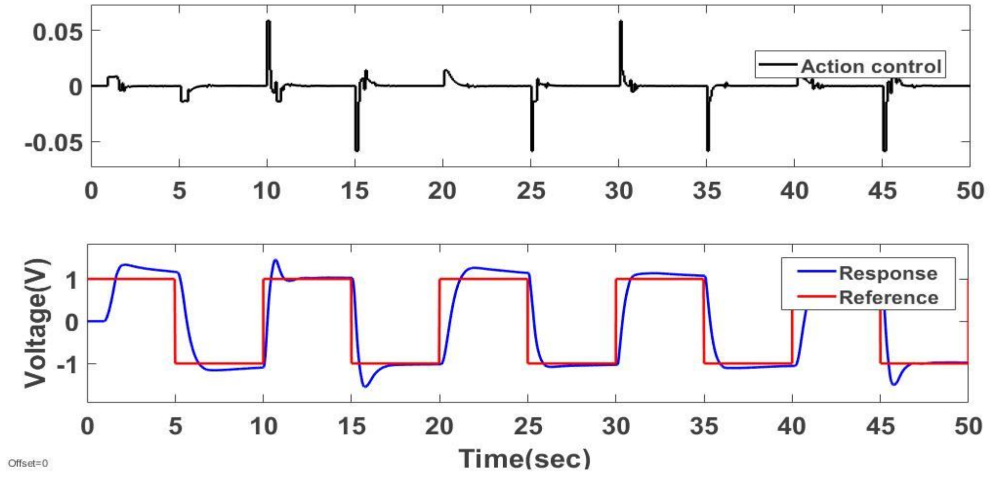
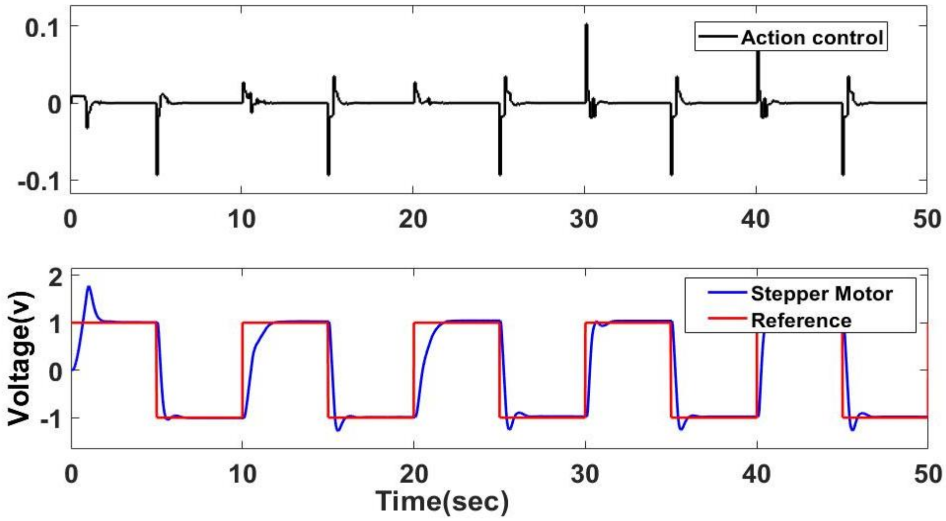
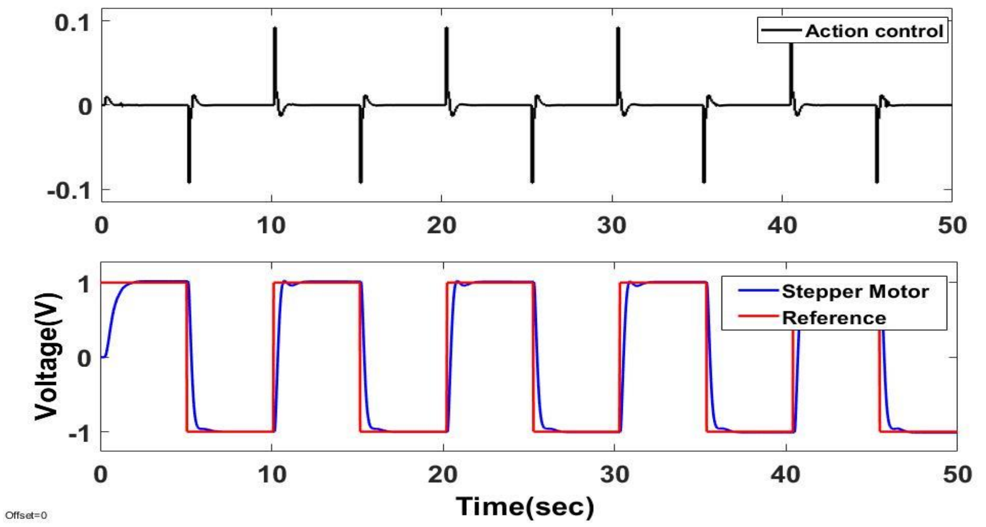
1- In the case of a ten percent and twenty percent packet loss, the system performance of the stepper motor control is not affected and remains stable.

2- In the case of forty percent, the performance of the system to control the stepper motor also remains stable and its response is slightly affected by 25%.

4.6 Comparison between three controllers with the WNCS model

Table-4.13: Comparison between output response of PID and Fuzzy PID and PSO Fuzzy PID controllers .

<table>
<tr><th></th><th>No-load (BWshare =0)</th></tr>
<tr><td>PID controller
No.of.Node=0</td><td>
</td></tr>
<tr><td>FuzzyPID controller
No.of.Node=0</td><td>
</td></tr>
<tr><td>PSO Fuzzy PID controller
No.of.Node=0</td><td>
</td></tr>
</table>

	Medium-load(BWshare=0.4)
PID controller No.of.Node=100	
FuzzyPID controller No.of.Node=150	
PSO Fuzzy PID controller No.of.Node=500	

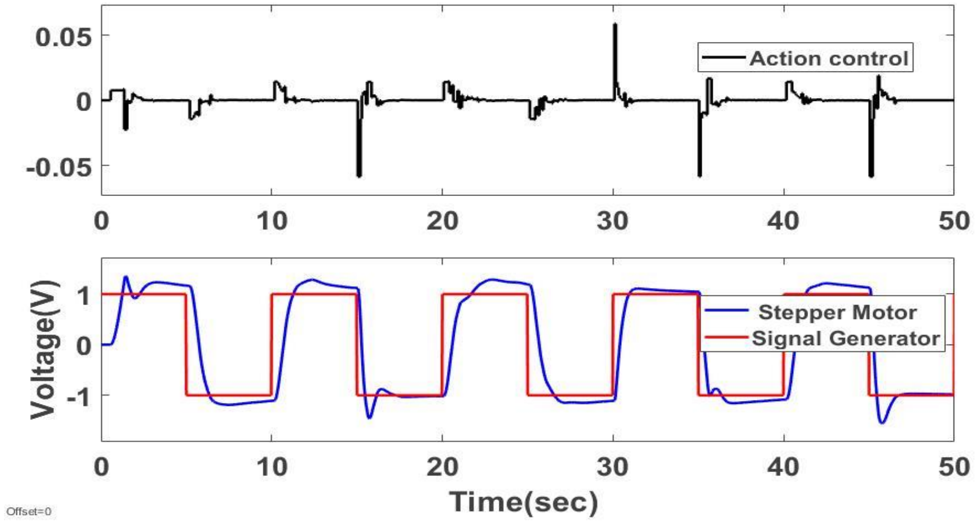
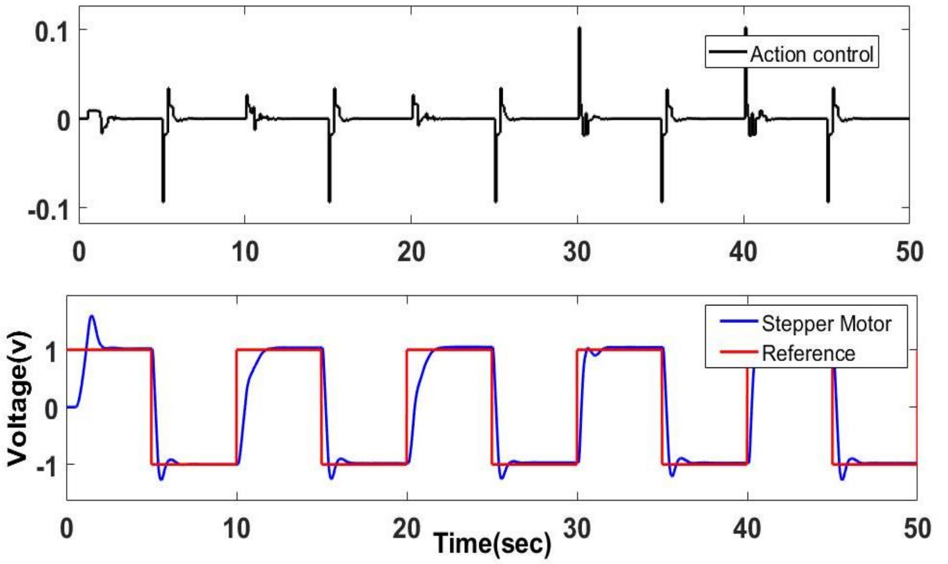
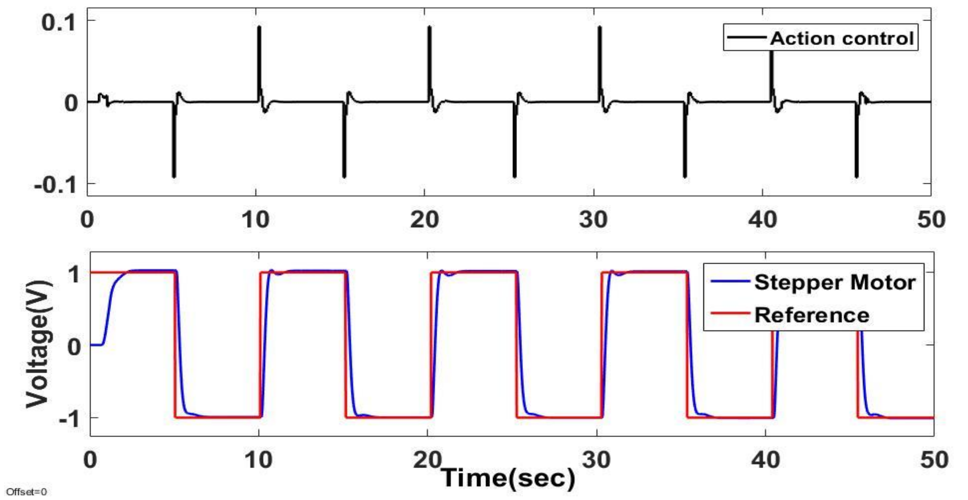
<table>
<tr><td></td><td>High-load(BWshare=0.9)</td></tr>
<tr><td>PID controller No.of.Node=75</td><td>
</td></tr>
<tr><td>FuzzyPID controller No.of.Node=100</td><td>
</td></tr>
<tr><td>PSO Fuzzy PID controller No.of.Node=300</td><td>
</td></tr>
</table>

Table-4.14: Performance of the PID controller and Fuzzy PID and PSO Fuzzy PID (in case No Load)

Criteria / Controller	TR	TS	MP%	ISE
PID	0.5173	9.0599	8.7723%	0.3107
Fuzzy PID	0.3014	1.7458	2.2778%	0.2280
PSO Fuzzy PID	0.278	0.606	2.28%	0.1803

Table-4.15: Performance of the PID controller and Fuzzy PID and PSO Fuzzy PID (in case Medium Load)

Criteria / Controller	TR	TS	MP%	ISE
PID	0.4979	13.2525	14.6948%	0.7891
Fuzzy PID	0.4648	2.36194	16.3275%	1.2606
PSO Fuzzy PID	0.8987	1.5621	0.9257%	0.8692

Table-4.16: Performance of the PID controller and Fuzzy PID and PSO Fuzzy PID (in case High load)

Criteria / Controller	TR	TS	MP%	ISE
PID	0.4497	14.7903	46.4870%	1.6137
Fuzzy PID	0.4591	9.7201	19.625%	1.5342
PSO Fuzzy PID	0.8971	1.8095	1.0360%	1.109

4.7 Comparison of the NCS Proposed Design

The proposed work and previous work for some research related to our work in the literature survey are compared in the following table 4.17.

Table-4.17: Comparison of the WNCS

Authors	Previous Work	The proposal work
Mais. Salman,et al.in 2017 [23]	Designing PID and SMC with: 1- WNCS(ZigBee) 2- Gain parameters are constant 3- Using one interference node 4- Not testing the number of nodes 5- Total time delay is 0.0928	Designing PSO Fuzzy PID with: 1- WNCS(ZigBee) 2- Gain parameters are variable with the system 3- Using two interference node 4- Testing the number of the node for system 5- Total time delay is 0.0219 for 500 nodes and is 0.0216 for 300 nodes
Isra. Laith, et al in 2019 [24]	Designing Fuzzy PID with: 1- NCS model (Ethernet network) 2- One interference node 3- The fuzzy rule is based on experts pervious 4- 250 nodes stable (medium load) 5- 300 critical (high load)	Designing PSO Fuzzy PID with: 1- Wireless NCS model (ZigBee network) 2- Two interference node 3- Fuzzy rules based on PSO optimization 4- 500 nodes stable (medium load) 5- 300 nodes stable (high load)

	6- Tested only with 10%packet loss is stable	6- Tested with: 10%packet loss and 20% packet loss the system be stable and in 40% the system slightly affected
Russul. Nasser, et al .in 2019 [25]	Designing PID and FOPID with: 1- WNCS (WI-FI) 2- Using PSO to found the PID and FOPID parameter with used filter coefficient N) 3- The gain parameters are boundary limited in PSO 4- Using step input as input reference 5- Testing packet loss with 40%.The system is highly affected	Designing PSO Fuzzy PID with: 1- WNCS(ZigBee) 2- Using PSO to found the fuzzy rules, and used fuzzy for adaptive tuning of PID gains without used filter coefficient N) 3- The gain parameters are online adjusting. 4- Using square signal as input reference 5- Testing packet loss with 40%. The system is slightly affected by 25%

Chapter Five

Conclusions and Suggestions for Future Works

5.1 Conclusions

The results of this thesis are summarized in the following points:

1- The PSO algorithm is used in the design of the proposed ZigBee network control system. The optimum rules for fuzzy logic with a PID controller are obtained. This PSO Fuzzy PID algorithm used to control the stepper motor wirelessly when any change occurs in the wireless network. The PID controller gain adaptive is tuned in proportion to these changes.

2- The average RTT has calculated through cooperation between the sensor and actuator tasks that it performs within one node, which each share the local clock, which leads to the release of the network from the additional packets that you need to synchronize and the average RTT must be less than sampling time Ts.

3- The proposed PSO Fuzzy PID controller is the most robust comparison with the PID and Fuzzy PID controllers in terms of stepper motor response and also the system performance in processing a larger number of nodes without loss of system stability. The PSO Fuzzy PID controller is represented with the sampling time of the system has 0.08 seconds, which the control system can use the PSO Fuzzy PID controller to process 500 and 300 nodes with BWshare 0.4 and 0.9, respectively while the PID controller system handles 100 and 75 nodes with BWshare 0.4 and 0.9 respectively. The Fuzzy PID controller system handles 150 and 100 with BWshare 0.4 and 0.9 respectively.

4- From this proposed work, the number of nodes is affected by two important parameters:

a- The sampling time at the interference transmitter node.

b- The value of BW, and the random value (w) used by chapter three according to the packet length equation 3.2.

5- The value of the parameters (kp, ki, kd) of the PID controller is variable online and according to the change that occurs to the system, and that the fuzzy logic with the PSO optimization algorithm will work to set these parameters automatically.

6- The proposed PSO Fuzzy PID controller is tested in the case of a medium load (500 nodes) with three different packet loss values (10%,20%,40%), and it appears more robustness against the noise and other disturbances.

5.2 Suggestions for Future Work

The following are some suggestions for future work that are summarized with the following points:

1. Use the other optimization algorithm in this work as the Ant colony optimization (ACO) algorithms or Chicken Swarm Optimization (CSO) algorithm to set the parameters controller online and with a shorter implementation time.
2. Implemented and design hardware system and tested practices to demonstrate the efficiency of using the hybrid PSO Fuzzy PID controller

References

[1] Guo, Lejiang, Yuanmin Tang, Zhou Liu, and Wei Xiong. "The theory and architecture of network control system." International Conference on Intelligent Computing and Cognitive Informatics, pp. 183-186. IEEE, 2010.

[2] Park, Pangun, Sinem Coleri Ergen, Carlo Fischione, Chenyang Lu, and Karl Henrik Johansson. "Wireless network design for control systems: A survey." IEEE Communications Surveys & Tutorials .Vol. 20, no. 2, pp.978-1013. 2017.

[3] Tang, Xiao-Ming, and Bao-Cang Ding. "Design of networked control systems with bounded arbitrary time delays." International Journal of Automation and Computing Vol.9, no. 2,pp: 182-190, 2012.

[4] Zhang, Dan, Peng Shi, Qing-Guo Wang, and Li Yu. "Analysis and synthesis of networked control systems: A survey of recent advances and challenges." ISA transactions .Vol .66 pp.376-392,2017.

[5] Gui, Li, Yu-Chu Tian, and Colin Fidge. "A conditional retransmission enabled transport protocol for real-time networked control systems." IEEE 36th Conference on Local Computer Networks, pp. 231-234. IEEE, 2011.

[6] Shi, Gaotao, and Keqiu Li. Signal interference in WiFi and ZigBee networks. Springer International Publishing, 2017.

[7] Mais M. Salman, “Sliding Mode Control for A Network Control System,”, M.Sc, In Networks Engineering And Internet Technologies, Iraq, 2017.

[8] Björkbom, Mikael. "Wireless control system simulation and network adaptive control." Aalto University, Department of Automation and Systems Technology 2010.

[9] Åström, Karl Johan, and Richard M. Murray. Feedback systems: an introduction for scientists and engineers. Princeton university press, 2010.

[10] Forsberg, Nils. "Improving a Hardware-based Real-time Kernel: Improving Embedded Systems Performance." 2014.

[11] Dantas, Amanda Danielle O. da S., et al. "PID control for electric vehicles subject to control and speed signal constraints." Journal of Control Science and Engineering 2018.

[12] Mehrpooya, Mehdi, and Sima Hejazi. "Design and implementation of optimized fuzzy logic controller for a nonlinear dynamic industrial plant using Hysys and Matlab simulation packages." Industrial & Engineering Chemistry Research .Vol 54, no. 44, pp11097-11105, 2015.

[13] Campos, Jefferson, Sebastián Jaramillo, Luis Morales, Oscar Camacho, Danilo Chávez, and David Pozo. "PSO Tuning for Fuzzy PD+ I Controller Applied to a Mobile Robot Trajectory Control." IEEE International Conference on Information Systems and Computer Science (INCISCOS), pp. 62-68, 2018.

[14] Peng, Chen, Yu-Chu Tian, and Dong Yue. "Output feedback control of discrete-time systems in networked environments." IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans .Vol.41, no. 1 , pp.185-190,2010.

[15] Cuenca, Ángel, Julián Salt, Antonio Sala, and Ricardo Pizá. "A delay-dependent dual-rate PID controller over an ethernet network." IEEE Transactions on Industrial Informatics .Vol.7, no. 1 pp. 18-29, 2010.

[16] Zhang, Hui, Yang Shi, and Aryan Saadat Mehr. "Robust static output feedback control and remote PID design for networked motor systems." IEEE Transactions on Industrial Electronics .Vol .58, no. 12, pp. 5396-5405, 2011.

[17] Urban, M., M. Blaho, J. Murgaš, and M. Foltin. "Simulation of Networked Control Systems via TrueTime." Technical Computing Prague 2011.

[18] Wang, Jing, Bingxin Shi, Jianjun Bai, Jinfeng Gao, and Xiaosheng Fang. "Stabilization of a wireless networked control system with packet loss and time delay: an ADS approach." Journal of Control Science and Engineering.Vol. 2012.

[19] Song, Zhiqiang, and Xianzhong Zhou. "Research and simulation of wireless sensor and actuator networked control system." 25th Chinese Control and Decision Conference (CCDC), pp. 3995-3998. IEEE, 2013.

[20] Peng, Daogang, Hao Zhang, Hui Li, and Fei Xia. "Research of networked control system based on fuzzy adaptive PID controller." Journal of Advances in Computer Networks .Vol.2, no. 1, pp.44-47, 2014.

[21] Shah, Dipesh, and A. J. Mehta. "Robust Controller Design for Networked Control system." International Conference on Computer, Communications, and Control Technology (I4CT), pp. 388-393. IEEE, 2014.

[22] Aung, N., Z. Naing, Mo Mo Myint Wai, and Hla Myo Tun. "Simulation of Wireless Networked Control System Using TrueTime And Matlab." International Journal of Scientific and Technology Research .Vol .6, no. 5, pp.31-35, 2015.

[23] Mais Salman, M. and Omnia Farouk Amin Awad. “Evaluating a ZigBee Network with SMC for Hard and Concurrent Parameter Variations.” Journal of Information Engineering and Applications. Vol .7, pp. 1-16,2017.

[24] Awad, Osama A., and Israa Laith."Gain Scheduling Fuzzy PID Controller for Distributed Control Systems." In International Conference on Applied Computing to Support Industry: Innovation and Technology, pp. 262-270. Springer, Cham, 2019.

[25] Russul N .Abdul-Hussain, and Osama A. Awad. "Studying The Effect of Sampling Time and Network Load on Wireless Networked Control Systems." Journal of Al-Qadisiyah for computer science and mathematics.Vol. 11, no. 3, pp.31,2019.

[26] Ge, Xiaohua, Fuwen Yang, and Qing-Long Han. "Distributed networked control systems: A brief overview." Information Sciences Vol.380, PP. 117-131, 2017.

[27] Buso, Simone, and Paolo Mattavelli. "Digital control in power electronics." Synthesis Lectures on Power Electronics Vol.5, no. 1 pp.1-229,2015.

[28] Zhao, Gang. "Wireless Sensor Networks for Industrial Process Monitoring and Control: A Survey." Network Protocols & Algorithms. Vol.3, no. 1, pp.46-63,2011.

[29] Nayyar, Anand, and Rajeshwar Singh. "A comprehensive review of simulation tools for wireless sensor networks (WSNs)." Journal of Wireless Networking and Communications .Vol.5, no. 1 ,pp. 19-47,2015.

[30] Delbari, Mohammad Mahdi, M. Taghi Hamidi Beheshti, Amin Ramezani, and Sadjaad Ozgoli. "Time delay sensitivity analysis in a Wireless Network Control System using LMI approach." 9th Asian Control Conference (ASCC), pp. 1-4. IEEE, 2013.

[31] Zhang, Xian-Ming, Qing-Long Han, and Xinghuo Yu. "Survey on recent advances in networked control systems." IEEE Transactions on industrial informatics .Vol.12, no. 5 ,pp. 1740-1752,2015.

[32] Brindha, M., and J. K. Mendiratta. "Networked control system-A survey." International Journal of Modern Education and Computer Science.Vol. 5, no. 6,pp. 42,2013.

[33] L Xu, Hao, Sarangapani Jagannathan, and Frank L. Lewis. "Stochastic optimal control of unknown linear networked control system in the presence of random delays and packet losses." Automatica ,Vol.48, no. 6 ,pp. 1017-1030,2012.

[34] Daniel .B .Irimia,and Ciprian. Lupu, “Process control over time delayed communication systems,” in Proceedings of the 9th WSEAS international conference on Telecommunications and informatics, pp. 19–25,2010.

[35] Mor, Kirti, Suresh Kumar, and Deepak Sharma. "Ad-Hoc Wireless Sensor Network Based on IEEE 802.15. 4: Theoretical Review." International Journal of Computer Sciences and Engineering” ISSN: 2347-2693, Vol. 6, pp. 220-225,2018.

[36] Purnama, Hendril Satrian, Tole Sutikno, Srinivasan Alavandar, and Arsyad Cahya Subrata. "Intelligent Control Strategies for Tuning PID of Speed

Control of DC Motor-A Review." IEEE Conference on Energy Conversion (CENCON), pp. 24-30. IEEE, 2019.

[37] Vilanova, Ramon, and Antonio Visioli. PID control in the third millennium. London: Springer, 2012.

[38] Kleiter, Gernot D. "Adams’p-validity in the research on human reasoning." J. Appl. Logics ,Vol.5 ,pp. 775-825,2018.

[39] Sreenivasappa, B. V., and R. Y. Udaykumar. "Analysis and implementation of discrete time PID controllers using FPGA." International journal of electrical and computer engineering .Vol,2, no. 1, pp. 71-82,2010.

[40] Czabanski, Robert, Michal Jezewski, and Jacek Leski. "Introduction to fuzzy systems." In Theory and Applications of Ordered Fuzzy Numbers, Springer, Cham, pp. 23-43, 2017.

[41] Ohlan, Anshu, and Ramphul Ohlan. "Fundamentals of Fuzzy Information Measures." In Generalizations of Fuzzy Information Measures, Springer, Cham, pp. 1-22. 2016.

[42] Shin, Yung C., and Chengying Xu. Intelligent systems: modeling, optimization, and control. CRC press, 2017.

[43] Mirjalili, Seyedali. "Evolutionary algorithms and neural networks." Studies in Computational Intelligence ,2019.

[44] Russul N.Abdul-Hussain,“An Optimized Qoc For A Wireless Network Controlled System”, M.Sc, In Networks Engineering And Internet Technologies, Iraq, 2019.

[45] TrueTime home page: http://www.control.lth.se/truetime , last accessed: 04.04.2017.

[46] Ohlin, Martin, Dan Henriksson, and Anton Cervin. TrueTime 1.5-reference manual. Department of Automatic Control, Lund Institute of Technology, Lund University, 2016.

[47] Tarnini, Mohamed Y. "Fast and cheap stepper motor drive." International Conference on Renewable Energy Research and Applications (ICRERA), . IEEE, pp. 689-693,2015.

[48] Laughton, Michael A., and Maurice George Say, eds. Electrical engineer's reference book. Elsevier, 2013.

[49] Wanga, Y., and L. Heb. "Analysis and Simulation of Networked Control Systems Delay Characteristics Based on Truetime." Computer Modelling and New Technologies .Vol.17, no. 4 ,pp.210-216,2013.

[50] Ogata, K. "Modern Control Engineering, 5th Prentice Hall." New York 2010.

المستخلص

توفر أنظمة التحكم في الشبكة اللاسلكية (WNCS) العديد من المزايا ، بما في ذلك المرونة وسهولة الحركة والصيانة. مما يؤدي إلى قيام الشبكة اللاسلكية المعتمدة بنقل البيانات بشكل صحيح بين وحدة التحكم والمشغل وعقد المستشعر.

في هذه الأطروحة، تم استخدام بروتوكول ZigBee اللاسلكي القياسي كوسيلة اتصال بين عقد NCS. تم تصميم ثلاث وحدات تحكم ومحاكاتها في هذه الأطروحة: وحدة تحكم PID النسبي والمشتق المتكامل ووحدة التحكم Fuzzy PID وخوارزمية تحسين سرب الجسيمات (PSO) مع وحدة التحكم Fuzzy PID (PSO Fuzzy PID) لتحديد استقرار محرك السائر السرعة لاسلكيًا عبر شبكة ZigBee. يتم اختبار وحدات التحكم مع حالات تحميل مختلفة لقياس تأثير التأخير الزمني على النظام عند زيادة عدد العقد.

يتم أخذ محرك السائر كحالة لاختبار أداء النظام. تتم مقارنة استجابة محرك السائر بوحدات التحكم الثلاثة المقترحة. معظم مشاكل WNCS هي التأخيرات الزمنية ومعالجة أكبر عدد من العقد وفقدان الحزمة التي ستختبر في مقترح العمل هذا.

يتم استخدام MATLAB 2018 مع أدوات محاكاة TrueTime في WNCS للتطبيق في مقترح العمل هذا. توضح نتائج المحاكاة أن وحدة التحكم PSO Fuzzy PID أكثر قوة مقارنة بوحدات التحكم الأخرى مع اختيار وقت أخذ العينات هو 0.08 ثانية ووقت أخذ العينات في عقدة مرسل التداخل هو ثانية واحدة.

يمكن لوحدة التحكم PSO Fuzzy PID التعامل مع 500 عقدة عند التحميل المتوسط (BWshare 0.4)، في حين أن وحدة التحكم PSO Fuzzy PID يمكنها التعامل مع ما يقرب من 300 عقدة عند التحميل العالي(BWshare 0.9). ومع ذلك ، يستمر النظام في الاستقرار. يتم استخدام خوارزمية تحسين PSO للعثور على أفضل قواعد التشويش. كان المنطق الضبابي يضبط معلمات الكسب التكيفية لوحدة التحكم PID. أيضًا ، يتم اختبار فقدان الحزمة لوحدة التحكم PSO Fuzzy PID. ويلاحظ أنه عندما تكون خسارة الحزم 10٪ و 20٪ في الشبكة ، يظل النظام مستقرًا. بينما تبلغ خسارة الحزمة 40٪ في الشبكة ، فإن استقرار النظام سيتأثر قليلاً بنسبة 25٪

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة الفرات الاوسط التقنية

الكلية التقنية الهندسية نجف

وحدة تحكم قائمة على جدولة الربح التكيفي لأنظمة التحكم في الشبكة

رسالة مقدمة الى

قسم هندسة تقنيات الاتصالات

كجزء من متطلبات نيل درجة ماجستير تقني في هندسة الاتصالات

تقدمة بها

رشا شامت سلمان

بكالوريوس في هندسة تقنيات الاتصالات

أشراف

الأستاذ المساعد الدكتور أحمد طه عبد الساده الجياشي

أيلول/2020