Iraqi Journal of Information and Communication Technology(IJICT)
Vol. 5, Issue 1, April 2022

www.ijict.edu.iq

ISSN:2222-758X
e-ISSN: 2789-7362

AN OPTIMIZED FUZZY LOGIC CONTROLLER FOR WIRELESS NETWORK CONTROL SYSTEM USING PSO

Ibtihal A. Hasan [1], **Osama A. Awad** [2]
[1,2] College of Information Engineering, Al-Nahrain University, Baghdad, Iraq
engibtihal70@gmail.com [1], usamaawa@coie-nahrain.edu.iq [2]
Received:7/5/2021, Accepted:23/9/2021
DOI:10.31987/ijict.5.1.180

***Abstract*- Wireless Networked Control System (WNCS), has the advantage of control signal efficiency, reduced costs, robustness, and more flexibility over the wired Networked Control System (NCS). However, one of the most important challenges of WNCS is the variable and random time delay associated with the network that affects the whole system stability. In this paper, a fuzzy PID controller is proposed to overcome this issue by controlling the DC motor over the Wi-Fi network. PSO algorithm is used to tune the controller, and TrueTime simulator is used to simulate the Wi-Fi network. The results showed that the tuned Fuzzy PID controller is efficiently reduced the effect of the time delay. The results show that the network can handle up to 7500 and 6000 nodes when the bandwidth is 0.4 and 0.9 respectively without a sensible degradation in the system performance of the tackled system.**

keywords: **WNCS, FPID controller, PSO, Time delay, TrueTime simulator.**

I. INTRODUCTION

The fast expansion of communication, control, and computer technologies has an important impact on the structure of the control system. Wireless networked control systems (WNCS) are composed of controllers, actuators, and sensors that communicate and are managed over wireless networks rather than old conventional point-to-point wired connections [1] as seen in Fig. 1. Recently, the use of wireless networks has increased according to a lot of features such as simplicity in ease of maintenance, flexibility, and installation. Wireless networked control systems (WNCS) came to eliminate Networked Control Systems (NCS) issues like limitations in flexibility and mobility because of wires connections of the network, likewise, packet loss and time delay which affect system performance and instability [2]. Despite the benefit of wireless technology but the main issues are the time delay in the feedback loop between the controller and sensors/actuators nodes as soon as the packet loss and limitation in the available number of nodes, which decrease the system performance than lead to system instability [3]. The researchers tried to reduce the time delay in the system using NCS and WNCS architectures with different plants, such as the work in [4] proposed a new Fuzzy-PID like Gain Scheduling controller over Ethernet 802.3 and studied the problem of time delay that affect the stability of NCS. Jalal Rostami, et al. in [5] examined the efficiency of the fuzzy logic-based programmable logic controller (PLC) to control the DC-motor speed in the MATLAB simulator, and the results show the effectiveness of the controller and enhanced the speed of the motor. Ye Zhang, et al. in [6] studied the stability of NCS under different packet loss and jitter using PID controller by converting continuous sampling system to a discrete system. Saeed, et al. in [7] proposed the Fractional Order Sliding Mode controller (FO-SMC) and experimentally tested the performance of the PMDC motor. Russel, Abdullah, Rasha, et al. in [3] [8] [9] sequentially, proposed Fuzzy Logic Controller based PID (FLC-PID) and compare it with the traditional PID controller to control the speed of the DC motor, both works in [3] [9] controlled the stepper motor's speed over Wi-Fi and ZigBee networks Sequentially and used Particle Swarm Optimization (PSO) algorithm in tunning the controller. Abdullah in [8] used the genetic algorithm (GA)

algorithm to tune the FLC-PID controller and improved the speed tracking and sudden load changes disturbance rejection for the motor. This paper focused on reducing the time delay of the DC motor over a Wi-Fi network using a fuzzy controller tuned by the Particle Swarm optimization technique. The paper is structured as: Section II presents the fuzzy controller design, Section III introduces the True-Time toolbox, Section IV presents the methodology of the work, Section V presents the simulation of WNCS. Section VI shows the experimental results and finally, Section VII summarizes the conclusion of the paper.

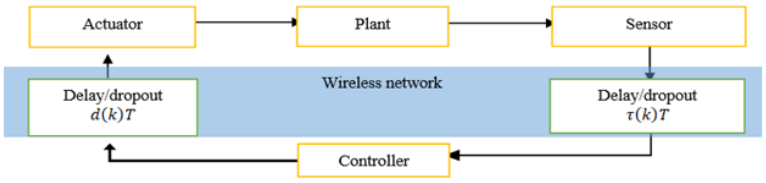

Figure 1: Block diagram of a WNCS

II. THE CONTROLLER DESIGN

A. *Fuzzy PID Controller*

Generally, Fuzzy PID controllers, it is an intelligent control that proved their efficiency and robustness to work with the DC-motors compared to classical PID controllers [10] [11] [12]. Fuzzy PID controller shows better control that backs to its characteristics on systems have produced robustness, rising time, also a good dynamic response [13]. 1965, Lotfi A. Zadeh presented Fuzzy logic (FL) as a type of soft computing [4]. Fuzzy logic control (FLC) is a successful means of expressing human thought activities. It represents the crisp values and deals with the issues that have vagueness. FLC uses membership functions (MF) with values varying between 0 and 1. The fuzzy inference system (FIS) comprises three sequential steps as in Fig. 2. The first step is fuzzification of input data that turns the data of input which is crisp values to fuzzy values using the MF. The second is fuzzy inference system (FIS). In this step, the control will implement a rule represented by several of (IF-THEN). Each fuzzy rule has a variable called a linguistic variable (e), also a value named linguistic value as shown in Table I whereas NB, NM, NS are the negative (big, medium, small) sequentially. Likewise, PB, PM, PS are positive (big, medium, small) and Z is zero. Fig. 3 shows the membership function type that used in this work. Two models of fuzzy inference can be present:

1) Mamdani model: in this fuzzy model, both premise and consequent values are fuzzy.
2) Takagi-Sugeno model: the premise is fuzzy, and consequent is a linear function of the inputs or crisply fixed. The last step in the FIS is Defuzzification, it turns the output of fuzzy values into non-fuzzy values. There are several types of defuzzification methods: Mean of maximum, Center of gravity, First maximum, and Last maximum [4] [14].

B. *Particle Swarm Optimization (PSO)*

Kennedy and Eberhart developed Particle Swarm Optimization (PSO) in 1995 as an effective way of optimization issues. The PSO technique aims to use animal behavior to looking for the closest positions to a global minimum or maximum

ISSN:2222-758X
e-ISSN: 2789-7362

solution. It simulates social behavior by using a population named a swarm. Every swarm has an individual named a particle. Each particle represents a position or a solution to the problem. In each iteration, particles evaluate the location to discover the best solution [15]. The benefits of the PSO technique are simplicity in calculations, and it considers as an auto-tuning method, so there is no mathematical calculation process need. The particles evaluate their positions based on the formulas below [3]:

$$V_i(t+1) = wV_i(t) + C_1 r_1(P_i(t) - X_i(t)) + C_2 r_2(G - X_i(t)) \quad (1)$$

$$V_i(t+1) = X_i(t) + V_i(t+1) \quad (2)$$

where C_1 is a cognitive learning factor, C_2 is social learning factor, r_1 and r_2 are random numbers ranged between [0,1] , P_i is the best position of an i_{th} particle, and w is the inertia weight.

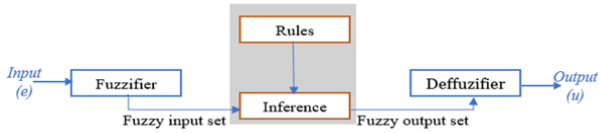

Figure 2: General fuzzy logic controller structure

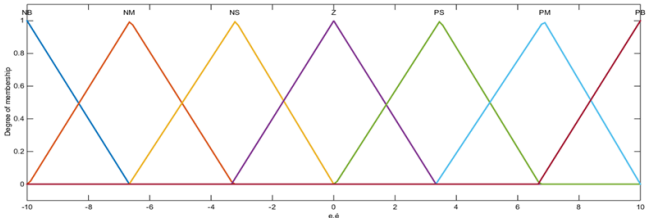

Figure 3: The membership functions of FLC

TABLE I
The Table of Fuzzy Rules

e / ė	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	Z
NM	NB	NB	NB	NM	NS	Z	PS
NS	NB	NB	NM	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PM	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PB	Z	PS	PM	PB	PB	PB	PB

ISSN:2222-758X
e-ISSN: 2789-7362

III. TRUE TIME TOOLBOX

In 1999, a MATLAB simulation toolbox was developed at the department of automatic control, Lund University. TrueTime is considered a freeware toolbox for simulating networked and embedded real-time control systems. One of the major advantages of the TrueTime simulator was having the ability to co-simulation the interaction between the computer architecture and real-world dynamics communication systems. Fig. 4 shows the fundamental blocks of the TrueTime toolbox library that consist of Kernel block, Send, Receive, Battery, Wired and Wireless Network, and Ultrasound Network. The kernel block is event-driven and executes code written using m-file or C++ language, for example, I/O tasks, control algorithms, and network interfaces. TrueTime also has two types of network blocks: wired and wireless networks. The network blocks are event-driven and execute messages between computer nodes in the network [3] [16] [17] [18].

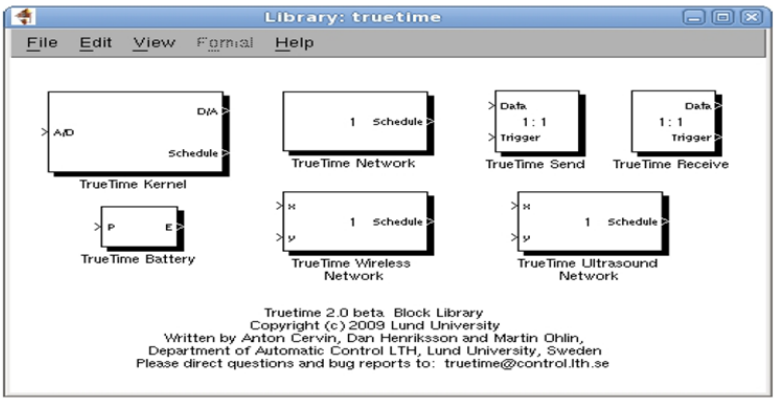

Figure 4: The TrueTime block library

IV. METHODOLOGY OF WORK

Based on fuzzy scaling factors in [19], when these parameters were implemented in the TrueTime network, the system response showed to be unstable. Due to this insatiability, this paper proposed a PSO technique to find the model of feed-forward delay (D_1) and feedback delay D_2 depending on the response that was taken from the simulated network. To improve the response of the DC motor with a gearbox system, the PSO technique with the delay model is used in this work to find the best values for the controller parameters K_1, K_2, K_3 and K_4 . PSO parameters in Table II are considered as best parameters that have been used after trial-and-error technique results with Root Mean square error is used for evaluating the fitness function equation as follow [19]

$$F = \sum_{i=1}^{N} \sqrt{\theta_a^2(i) - \theta_m^2(i)} \times 0.95 + (sys - overshoot \times 0.005) \times T_S \tag{3}$$

Where θ_a is the angular position of the actual system,T_S is the sampling time, θ_m and is the angular position of the modelled system. These parameters are used for finding both delay models and fuzzy scaling factors. Fig. 5 shows the methodology flowchart.

ISSN:2222-758X
e-ISSN: 2789-7362

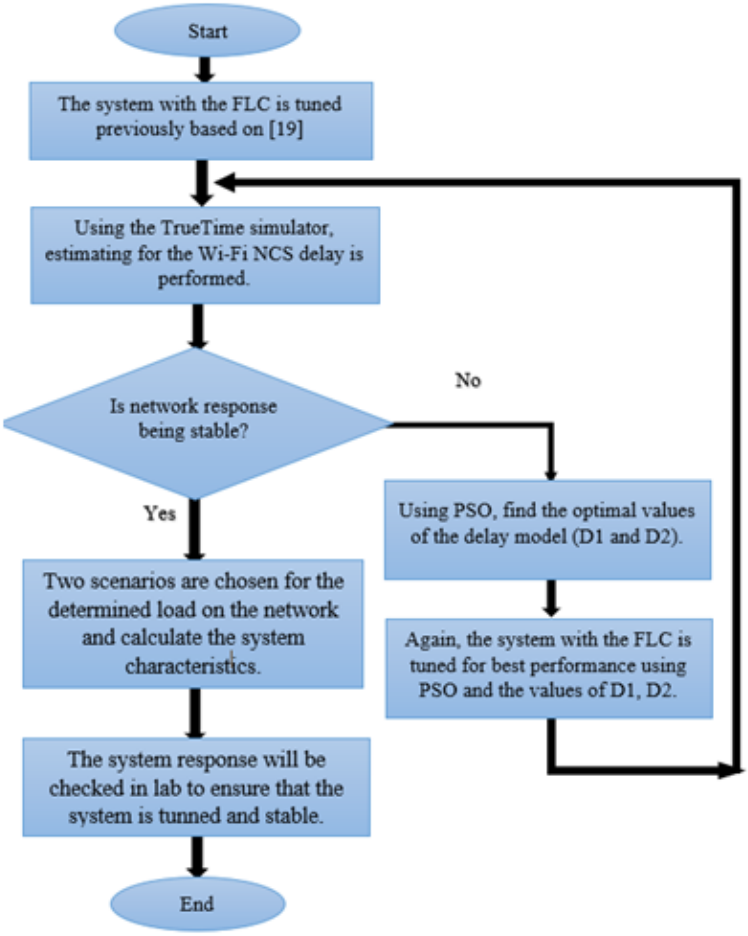

Figure 5: Flowchart of the methodology

TABLE II
PSO Parameters for Controller Parameters Calculation

Parameters	Value
Number of birds	35
Birds steps	85
The dimension of the problem	4
C_1	2.5
C_2	2.5
W	0.6180

ISSN:2222-758X
e-ISSN: 2789-7362

After the simulation of the programs, feedforward and feedback delay were found to be $D_1 = 0.0705$, $D_2 = 0.0041$. Fig. 6 shows the simulated result of optimized FLC scaling factors using PSO. the optimized Fuzzy parameters are $k_1 = 0.03524$, $k_2 = 0.0080$, $k_3 = 0.5762$, and $k_4 = 27.9820$, with best fitness function equal to 11.5572. Fig. 7 represents the fuzzy scaling factors positions.

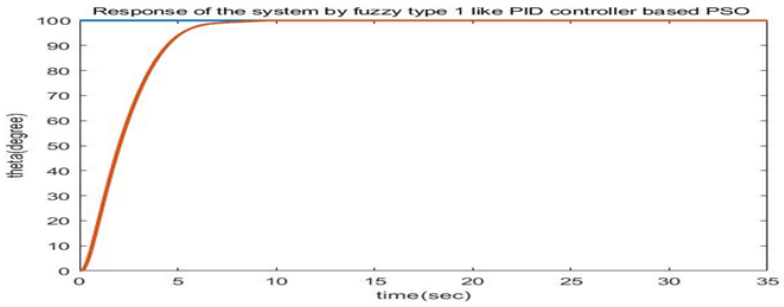

Figure 6: Angular position response of the system using fuzzy PID controller tuned by PSO with step input

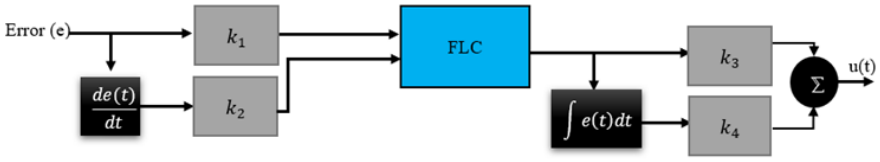

Figure 7: The position of fuzzy PID scaling factors

V. Simulation of WNCS

Based on the TrueTime simulator features discussed in the previous part, TrueTime has been chosen to simulate the WNCS. TrueTime Wireless Network block is used to model the wireless network where it received the sent packet from the input channel, perform the simulations then send it to the output channel to reach the corresponding receiving node. In this work, the WLAN 802.11b network type has been used with Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA). Fig. 8 shows the standard frame format for 802.11b where it consists of four nodes connected with the Wireless Network block (sensor/actuator node, a controller node, and two interference nodes for sending and receiving). Wireless Network block has two inputs (X, Y) to specify the location of each node. The location of the sensor/actuator node positioned at (5,0) m, controller node (X_c, Y_c) located at (80,0) m, for interference sender (XI_c, YI_c) location is positioned at (15,0) m and for Interference receiver (XI_r, YI_r) is located at (30,0) m. All the nodes are wirelessly connected through 802.11b technology at a data rate (11 Mbits/s) with a minimum frame size of 272 bits. The proposed Wi-Fi NCS model is shown in Fig. 9. The remote site contains an event-driven controller used to execute the designed controller algorithms with the reference input via A/D input port. The local side consists of a continuous-time plant, an event-driven actuator,

ISSN:2222-758X
e-ISSN: 2789-7362

and a time-based sensor with a Ts sampling period. Both sensor and actuator are assumed to have a shared local memory, also both of them are directly connected to the plant by A/D and D/A input/output ports.

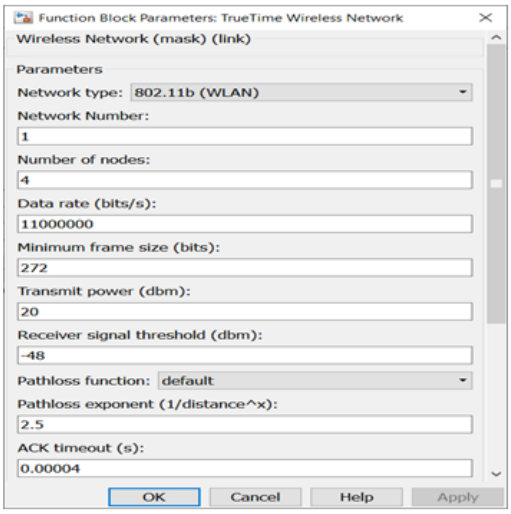

Figure 8: Wireless network block parameters

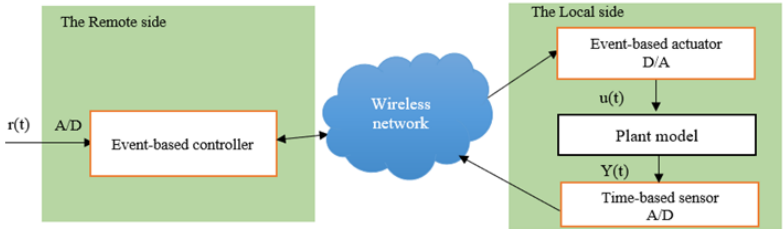

Figure 9: The proposed Wi-Fi NCS model

In this work, two interfering nodes are used in the simulation to generate the traffic. The first one is called an interfering sending node that is connected directly to MATLAB BWshare (Bandwidth) block by its A/D converter input port and used to send the packets over the network. Another one is the interfering receiver node. The main reason for designing interference receiver node is confirming that the network is loaded with several packets from the emulated nodes by the interference sender node, also to discuss the effect of this loaded network on the performance of the system and showing the number of nodes that can be handled by the network. Based on [20] four parameters clarify the operation of the interference sender node:

1) P_{max}: It's a constant value to consider the maximum size of the packet. For Wi-Fi network is 2304.
2) N: The number of nodes executing by interference sender node block.
3) w: is a random number between 0 and 1 specified the weight and used to generate different packet sizes to obtained different delays in the network.

ISSN:2222-758X
e-ISSN: 2789-7362

4) BW_{share}: a fraction of the occupied bandwidth by this node, where if it's high then the network will high be loaded too and vice versa.

To determine the total packet length, the following equation will be used:

$$P_l = (N \times BW_{share} \times P_{max} \times w) \tag{4}$$

where P_l is the total packet length in bytes, N is the number of nodes and is the maximum size of the network packet. For the Sensor/Actuator node block, there are two tasks to be implemented. The first one is called the sensor task, other is the actuator task. The defined kernel for the sensor and actuator task is chosen as prioDM. The sensor task is operating in a time-based mode, that calculates the time delay with period Ts(s). It will sense the output of the plant directly from the A/D input port, then send the sensed data across the wireless network to the remote controller node. The actuator task is an event-based task, that is initialized when the control packet is received from the controller side. The actuator task reads the controller packet (control signal) and stops the timer then saves time in an array called RTT, also it gets data from the D/A port to actuate the plant. The TrueTime Kernel block (controller node) consider as a remote node and defines the operating system as 'prioFP'. The operation mode of the controller task is event-based. In the controller node, when the controller task is triggered, first, it receives a sensor signal from the network interface, then this signal will be processed by FPID (fuzzy PID) controller and sent as control signal to the sensor/actuator node. To control the DC motor with the gearbox, the controller node uses an FPID controller that is optimized by the PSO technique. The control signal is sent back to the sensor/actuator to actuate the motor. The motor model is modelled and developed experimentally in previous work [19], and the transfer function is shown in the formula below:

$$\frac{\theta_o(s)}{\theta_r(s)} = \frac{1.0278}{s^2 + 1.7994s + 0.02806} \tag{5}$$

where θ_o is the output angle, representing the Dc motor angular displacement, and θ_r is the required angle angular displacement? Battery blocks are connected to each controller node, sensor/actuator node, and both interference sender/receiver nodes. it used to manage channel transmission. By running the Power control tasks will send out ping messages to another node to test the channel transmission. When the reply is received from the node that's mean the channel is specified to be good and the transmission power will be decreased and vice versa. Fig. 10 shows the proposed WNCS model designed in MATLAB and TrueTime toolbox. Each of the blocks in Fig. 10 has a sub-block, these sub-blocks will be represented in Fig. 11, 12, and 13 respectively.

ISSN:2222-758X
e-ISSN: 2789-7362

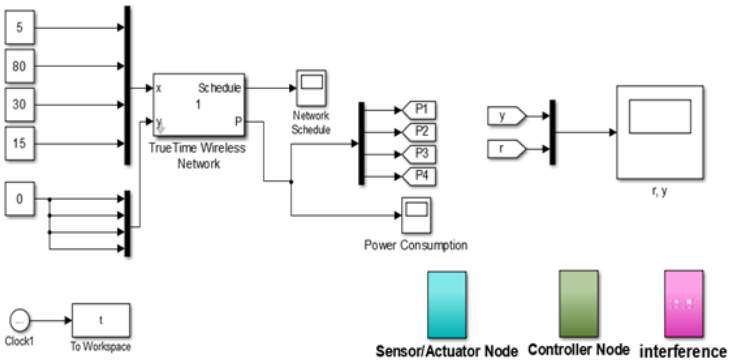

Figure 10: The proposed WNCS model design in MATLAB

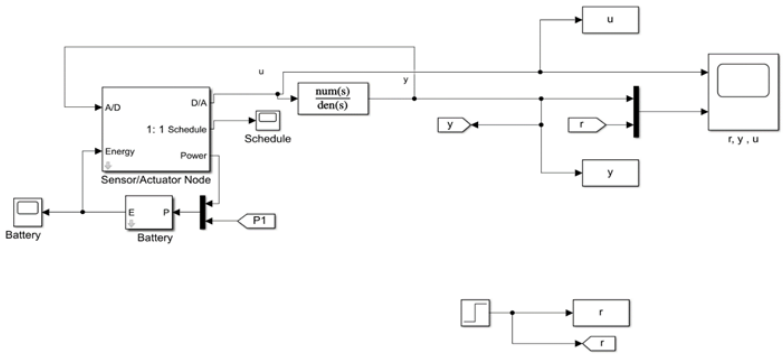

Figure 11: The sub-block of sensor/actuator node in MATLAB

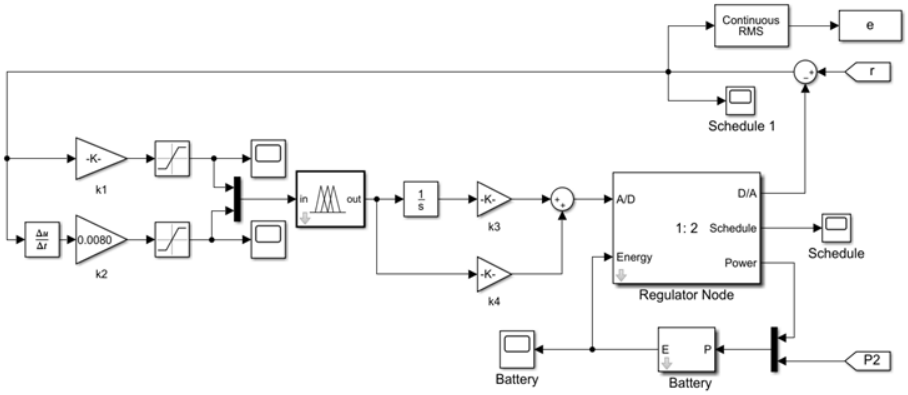

Figure 12: The sub-block of controller node in MATLAB

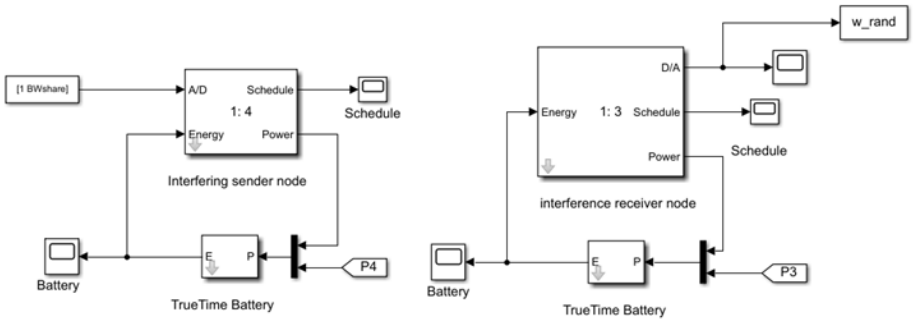

Figure 13: The sub-block of interference node in MATLAB

VI. RESULTS

Based on the proposed FPID controller parameters and Shannon's theory of samples, in this study, WNCS simulated for testing three levels of sampling time in interference sender node with the system sample time is equal to 0.01 sec. These samples are 0.035, 0.5, 1 second. Two scenarios are chosen for the determined load on the network. First, when the network is medium loaded ($BW_{share} = 0.4$). Second, high loaded network ($BW_{share} = 0.9$). The number of nodes that emulated in the interference sender node block is set to 7500 and 6000 respectively for the scenarios, which affect the number of packets and average Round Trip Time (RTT) to the packets wirelessly send between the controller and sensor/actuator nodes. Average RTT and system characteristics (settling time, rise time, overshoot) are also measured.

1) Scenario one: FPID controller with $BW_{share} = 0.4$.

   the total number of nodes in WNCS is 7503 nodes. these nodes are divided as 7500 nodes emulated by interference sender node, one node as Sensor/Actuator node, and the last one as controller node. the packet length is calculated using Eq. 4 as follows:

$$P_l = (7500 \times 0.4 \times 2304 \times w)byte \tag{6}$$

   for 35 sec. of simulation time, Table III shows the measured results of simulated WNCS until the steady-state is reached for (0.035, 0.5, and 1) seconds of sampling time. Fig. 14 shows the system response for the third case when $T_{S_i} = 1$. Fig. 15 shows the control action for the third case, also Fig. 16 shows the number of packets of FPID with RTT in seconds for the same case.

TABLE III
Medium Loaded WNCS at $BW_{share} = 0.4$

System sampling time T_s =0.01 sec.					
Sampling time in interference sender node Ts_i	NO. of nodes	Overshoot (%)	Settling time (Sec.)	Rise time (Sec.)	Average RTT
0.035	1000	8.7133	5.7529	1.9290	0.1775
0.5	6000	0.7225	4.6086	2.8582	0.3445
1	7500	0.6738	4.7525	3.0623	0.1636

ISSN:2222-758X
e-ISSN: 2789-7362

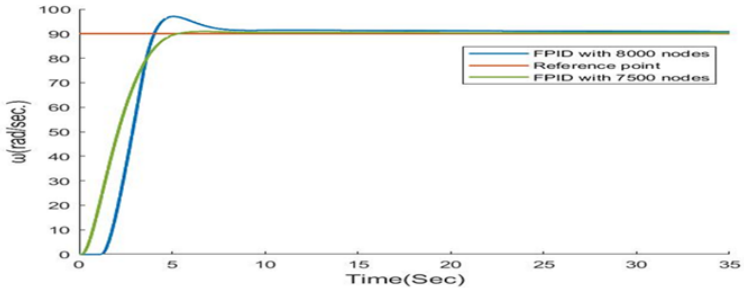

Figure 14: DC-motor angular displacement using fuzzy PID controller with $T_s = 0.01$, $BW_{share} = 0.4$, $T_{S_i} = 1$, simulation time = 35 seconds

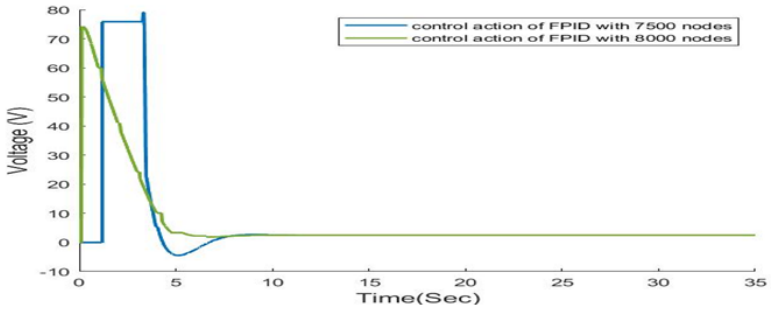

Figure 15: Fuzzy PID controller action with $T_s = 0.01$, $BW_{share} = 0.4$, $T_{S_i} = 1$, simulation time = 35 seconds

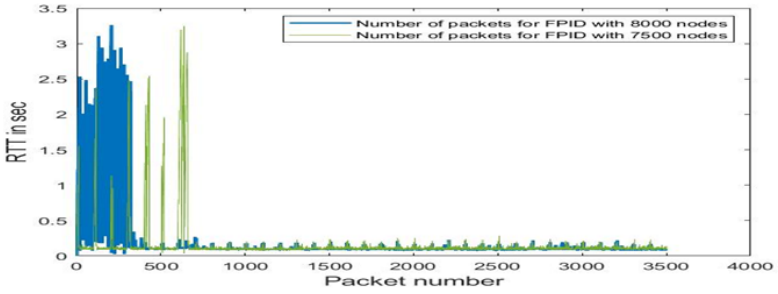

Figure 16: The number of packets of FPID and RTT values for Wi-Fi network with $T_s = 0.01$, $BW_{share} = 0.4$, $T_{S_i} = 1$, simulation time = 35 seconds

2) Scenario two: FPID controller with $BW_{share} = 0.9$

As scenario one, the total number of nodes in WNCS is 6003 nodes. the packet length is calculated as:

$$P_l = (6000 \times 0.9 \times 2304 \times w)byte \tag{7}$$

Table IV shows the measured results of simulated WNCS for 35 sec. of the simulation time until the stability reached (0.035, 0.5, 1) seconds of sampling time. Fig. 17 shows the system response for the third case when $T_{S_i} = 1$. Fig.

18 shows the control action, and Fig. 19 shows RTT values with the number of packets in sec. for the same case.

TABLE IV
High Loaded WNCS at $BW_{share} = 0.9$

System sampling time T_s =0.01 sec.					
Sampling time in interference sender node Ts_i	NO. of nodes	Overshoot (%)	Settling time (Sec.)	Rise time (Sec.)	Average RTT
0.035	400	2.0329	4.9178	2.2886	0.1627
0.5	4000	0.6865	4.6015	2.8546	0.2637
1	6000	1.3350	4.3624	2.9336	0.2709

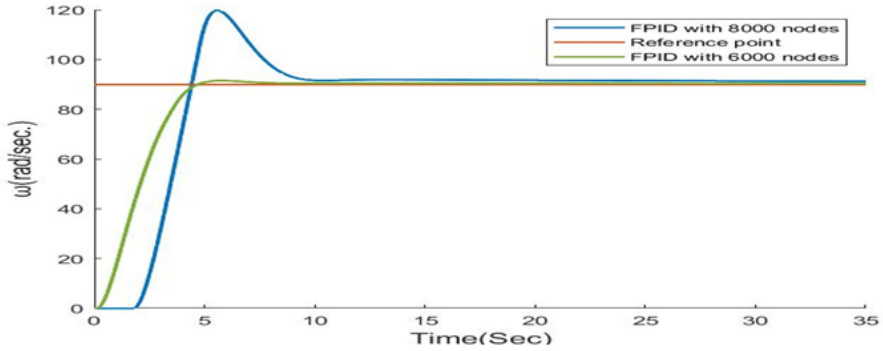

Figure 17: DC-motor angular displacement using fuzzy PID controller with $T_s = 0.01$, $BW_{share} = 0.9$, $T_{S_i} = 1$, simulation time = 35 seconds

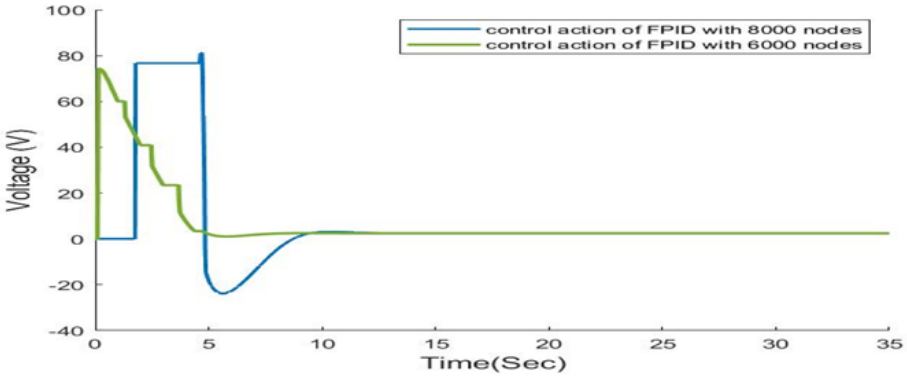

Figure 18: Fuzzy PID controller action with $T_s = 0.01$, $BW_{share} = 0.9$, $T_{S_i} = 1$, simulation time = 35 seconds

ISSN:2222-758X
e-ISSN: 2789-7362

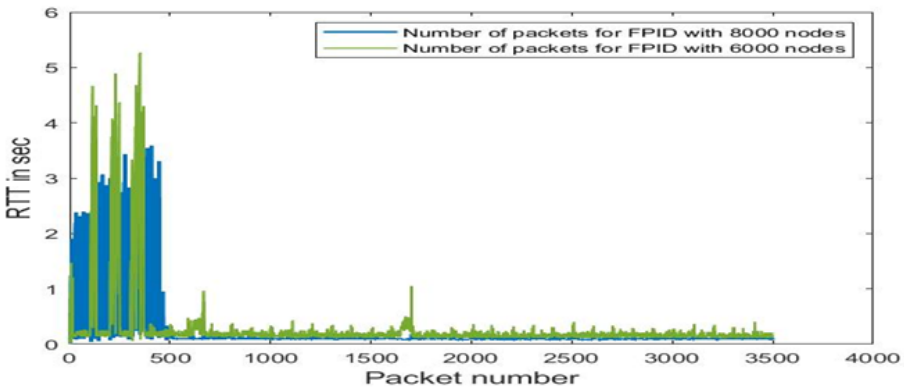

Figure 19: The number of packets for Wi-Fi network using a tuned FPID controller with $T_s = 0.01$, $BW_{share} = 0.9$, $T_{S_i} = 1$, simulation time = 35 seconds

From the results of the two scenarios, comparing with other cases, when $T_s = 0.01$ with medium loaded network and $T_{S_i=1}$, it gives the best system response where the network can handle 7500 nodes and remain stable. The T_{S_i} value had a critical effect on the system performance. A higher T_{S_i} value gave a better system response and could handle a greater number of nodes in the network. The interpretation for this can be back to the period where the load distribution has effectively appeared. It has a great negative effect if concentrated within a short period of the system sampling time T_s.

VII. CONCLUSION

The most important problem affecting the implementation of real-time systems via WNCS is the induced variable and random time delay associated with any network. This problem degrades the performance and stability of the system. The effect of the time delay can be eliminated by designing a robust controller that overcomes this issue. In this work, the Fuzzy PID controller was proposed and designed to control a DC motor wirelessly. The controller is tuned using the PSO algorithm to find its optimal scaling factors. FPID controller was tested with the sampling time of the system of 0.01s. Simulation results show that when the sampling time in the interfering sender node is one second, the system can handle 7500 nodes when the bandwidth is 0.4 and 6000 nodes when the bandwidth share is 0.9. The performance of the DC motor is still acceptable with this high load in the wireless network. These results will be implemented empirically on the real system, which was previously modelled by Eq. (5) and tested in this paper, using the true real-time simulator. Currently, the authors are setting a practical experiment, to demonstrate the applicability and effectiveness of the proposed methodology.

ISSN:2222-758X
e-ISSN: 2789-7362

REFERENCES

[1] P. Park, S. Coleri Ergen, C. Fischione, C. Lu and K. H. Johansson, "Wireless Network Design for Control Systems: A Survey" , in IEEE Communications Surveys & Tutorials, Vol. 20, No. 2, pp. 978-1013, Second quarter 2018, Doi: 10.1109/COMST.2017.2780114.
[2] M. M. Salman and O. A. Awad, "Evaluating a ZigBee Network with SMC for Hard and Concurrent Parameter Variations" , Journal of Information Engineering and Applications, Vol. 7, No. 5, pp. 1-16, 2017.
[3] R. N. Abdul-Hussain and O. A. Awad, "Studying the Effect of Sampling Time and Network Load on Wireless Networked Control Systems" , JQCM, Vol. 11, pp. 31-42, 2019.
[4] O. A. Awad and I. Laith, "Gain Scheduling Fuzzy PID Controller for Distributed Control Systems" , Applied Computing to Support Industry: Innovation and Technology, Springer, pp.262-270, 2019.
[5] J. R. Monfared, M. Fazeli, and Y. Lotfi, "Design and PLC Implementation for Speed Control of DC Motor Using Fuzzy Logic ARTICLE INFO" , Vol. 3, No. 2. pp. 71-75, 2015.
[6] Y. Zhang, Y. Yang, H. Q. Gu, and Y. J. Lu, "Stability Analysis of NCS under PID Control" , Proceedings of the 28th Chinese Control and Decision Conference, CCDC 2016, pp. 5718-5721, 2016.
[7] M. Akram Ahmad, "Speed Control of a DC Motor Using Controllers" , Automation, Control and Intelligent Systems, Vol. 2, No. 6. p. 1, 2014.
[8] A. Y. Al-Maliki and K. Iqbal, "FLC Based PID Controller Tuning for Sensorless Speed Control of DC Motor" , Proceedings of the IEEE International Conference on Industrial Technology, Vol. 2018, Febru. , pp. 169-174, 2018.
[9] R. S. Salman and A. T. Abdulsadda, "Network Scheduling by Using Expert Nonlinear Controller" , FJIECE, pp. 1708 - 1716, June, 2020.
[10] K. Sharma and D. K. Palwalia, "A Modified PID Control with Adaptive Fuzzy Controller Applied to DC Motor" , IEEE, an International Conference on Information, Communication, Instrumentation and Control, 2017.
[11] A. Muhyiddin Bin Yusof, "A Comparative Study of Conventional PID And Fuzzy PID for DC Motor Speed Control" , June, 2013.
[12] N. Sheng, X. Chen, H. Ge, and D. Li, "Optimal Interval Type 2 Fuzzy PID Controller and Its Application in Inverted Pendulum System" , IEEE 2nd International Conference on Automation, Electronics and Electrical Engineering, 2019.
[13] Q. Song and Y. Wu, "Study on The Robustness Based on PID Fuzzy Controller" , Proceedings, 2017 International Conference on Computing Intelligence and Information System, CIIS 2017, Vol. 2018, Janua. , pp. 142-145, 2018.
[14] T. Laufer, "Improvement Possibilities of The Maximum Defuzzification Methods" , Proceedings, INES 2019 IEEE 23rd International Conference on Intelligent Engineering Systems, April, 2019.
[15] D. Pal, P. Verma, D. Gautam, and P. Indait, "Improved Optimization Technique Using Hybrid ACO-PSO" , Proceedings on 2016 2nd International Conference on Next Generation Computing Technologies, NGCT 2016, pp. 277-282, 2017.
[16] A. Cardoso, S. Santos, A. Santos, and P. Gil, "Simulation Platform for Wireless Sensor Networks Based on The TrueTime Toolbox" , IECON Proceedings, Industrial Electronics Conference, pp. 2115-2120, 2009.
[17] D. Chotaliya and C. B. Bhatt, "Simulation of Wireless Network Using TrueTime Toolbox" , Baba Farid Conference on latest advancements in Science, Engineering and Research, 2011.
[18] A. Cervin, D. Henriksson, and M. Ohlin, "TRUETIME 2.0- Reference Manual" , 2016.
[19] I. Akram and O. A. Awad, "System Identification Based PSO Approach for Networked DC Servo Motor" , IOP Conf. Ser. Mater. Sci. Eng. , Vol. 1090, No. 1, p. 012059, 2021, doi: 10.1088/1757-899x/1090/1/012059.
[20] A. Cuenca, J. Salt, A. Sala, and R. Piza, "A Delay Dependent Dual-Rate PID Controller over An Ethernet Network" , IEEE Transactions on Industrial Informatics, Vol. 7, No. 1. pp. 18-29, 2011.

