Design and Implementation of Wireless Networked PID Control Algorithm Using MQTT Device Simulator for IoT Systems

SAMBA AIME HERVE, HOUWE ALPHONSE, YEREMOU TAMTSIA AURELIEN

National Higher Polytechnic School, University of Douala, Douala, Cameroon
Faculty of Science, The University of Maroua, Maroua, Cameroon
University of Maroua
Po. Box 1872, Douala
CAMEROON

Abstract: In this work, the implementation, simulation and analysis of PID algorithms have been carried out; applied in a DC motor with brushes by means of a control system in a wireless network, where the network is part of the closed control loop, using the protocol Message Queuing Telemetry Transport (MQTT) for communication between the broker and the plant. The performance of each of the control algorithm was evaluated through simulations in the network, which allowed determining the dynamic response parameters of the system. With the data obtained, the algorithms were implemented using a development tool based on the flow for visual programming, with the purpose of controlling and monitoring wirelessly or even through a mobile device remotely in real time, thus becoming an IoT application. Optimum performance is corroborated through communication and control tests when the plant is subjected to disturbances.

Key-Words: Protocol Message Queuing Telemetry Transport (MQTT), PID control algorithm, IoT application, Wireless

Received: June 15, 2021. Revised: March 21, 2022. Accepted: April 24, 2022. Published: June 9, 2022

1 Introduction

Industry 4.0 contains numerous technologies and associated paradigms, such as Internet of things (IoT) control through the cloud, artificial intelligence as described in [1, 2]. The objective of Industry 4.0 is to achieve a higher level of operational efficiency and productivity, a higher level of automation, for which, optimal control algorithms are needed, which allow greater adaptation and human–machine interaction (HMI) and automatic data communication. These characteristics are not only highly correlated with Internet of Things technologies and advanced algorithms, in this work, it is proposed to implement PID controller over a network, with the help of MATLAB software and the True Time library, which allows simulation of the behavior of the dynamic system in the network.

For the implementation of WNCS, the MQTT protocol was used, which is defined as a lightweight messaging protocol ideal for M2M scenarios [3], as well as being an alternative for the output signal to continue without delay despite variations in the signal input, such as: motor speed variation and data exchange in real time, since it allows the implementation of different qualities of service that identify the loss of messages or duplication in them. In Section.4, the results of the design of the system and the application of the algorithms are presented, and finally, in Section. 5, the results are analyzed to determine the conclusions

2 Background

In this work, the MQTT protocol was used, which uses an editor-subscriber model, which has a low overhead in the network, which allows its implementation in lowpower devices, such as microcontrollers that could be used in remote IoT sensors. In addition to using the Mosquitto broker that is designed to be used in all situations, where there is a need for lightweight messaging, particularly on devices with limited resources [4]. It is complemented together with Node-RED, which is a flow-based open-source development tool for the combination of IoT hardware devices and online services developed by IBM Emerging Technology.

In addition to being free and JavaScript-based, it is built on the Node.js platform, which provides a browser-based visual flow editor [5]. For the design of the system, the broker that will be hosted on the Raspberry Pi is used. It is a small computer, the size of a credit card at a very low cost for any type of project, and it can be used practically as a PC. The software used in this card is open source, and it occupies an operating system adapted from Debian, called Raspbian, although the card can also work in other operating systems, including version 10 of Windows [6]. According to the aforementioned, it is summarized in wireless network control systems, based on the study of control systems through wireless networks, dynamic control systems require communication between sensors and actuators; therefore, establish a closed loop control system that uses a wireless network for the transmission of information [7]. It should be taken into account that due to the different failures that wireless networks present, the application of feedback controllers is discussed, that is, WNCS in the industrial field [8].

For PID control, there are several types of considerations. Due to its simplicity, it is one of the most used options at the moment in control applications, and due to its three-parameter structure, it is considered to use the PID up or PID in parallel in this research, since each action of the components that make up the controller are separated in the equation to later be added [9].

Understanding the alteration of each of the variables that make up the interior of the control process in a system, reinforces control and solves the difficulties of classical control theory [10], where its main foundation is the analysis of input and output for dynamic systems with constant parameters. The state space facilitates the understanding of the behavior of one or more variables with a minimum amount of information in a given time, which allows forecasting future instants in the behavior of the variables [11].

3 Simulation and Implementation Methodology

In this identification and experimental modeling section, you focus on the prototype shown in Figure (1). It is built and instrumented with: encoder sensor for speed measurement, the actuator, which is a brushed motor and the control unit. Regarding the location of the sensor, it must be in a place where the luminosity has no influence so that the data acquisition is obtained with the least amount of oscillations possible and the control does not become complex. Methodologically, assuming the constant variation of PWM to reach the maximum speed of the motor; the transfer function of the plant is obtained; simulated using MATLAB software to later discuss the performance of the PID control in the DC motor through the implemented IoT tools. The case of this system is to automatically control and compare the performance of each applied algorithm.

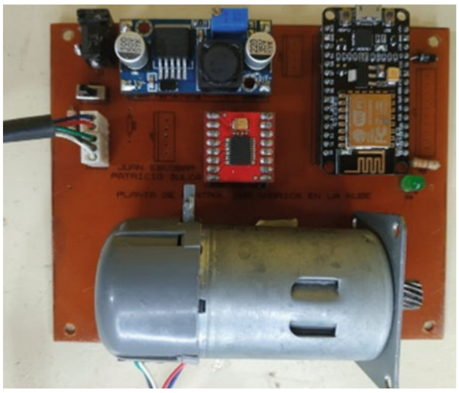


Figure 1: Speed control system implemented.

3.1 Operation

To manage the process in the experimental prototype, the ESP-8266 is used, which must be able to read the message from the MQTT broker that is hosted on the Raspberry Pi and then apply the PID controller that must stabilize the plant speed on a local server. The response time to any disturbance in the closedcircuit must be minimal to stabilize the DC motor, since, according to the result, the MQTT protocol used in this research uses the Mosquitto corridor and the QoS level 0, understood as the way of managing robustness to send messages to the client in case they are not published and subscribed as a messaging service mechanism [13]. It is an event-based architecture, so each message is sent to the recipients who have subscribed to a specific publication, the broker is in charge of distributing the messages to the recipient, in this case to the engine. To install MQTT on a Raspberry Pi, you must use a server widely known as Mosquitto [14] (Figure (2)).

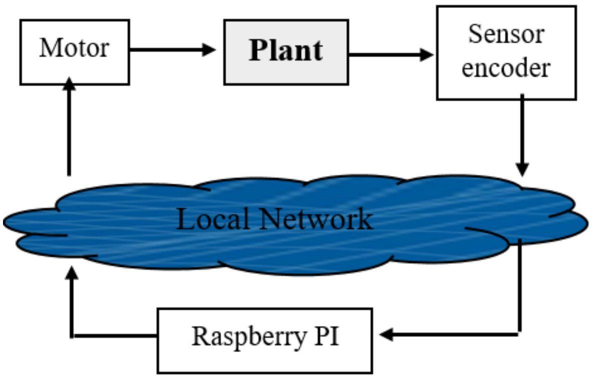


Figure 2: WNCS block diagram.

3.2 Implementation

With the help of the Arduino software, a sample of PWM versus speed data is obtained, which will later be entered in the Ident tool of MATLAB to find the transfer function of the DC motor, as can be seen in the Equation.1.

$$G(s) = 0.4745s^2 + 1.071s + 0.2813 \qquad (1)$$

Once the transfer function has been obtained with the assistance of the Node-RED tool, the PID control algorithm is carried out on our local server. For the method of tuning the controller parameters k_P, k_I, and k_D , the PID Tuner tool was used of the MATLAB software that automatically calculates the controller gains in order to offer a fast and stable response, obtaining values as shown in the Table.1.

Table 1: PID controller gains.

Gains	Values
k_P	0.24259
k_I	0.093188
k_D	0.15788

For the communication of the plant, the ESP8266 is programmed in Arduino, which is responsible for sending the data obtained from the encoder sensor and receiving the data to stabilize the speed at the set point established in the PID controller. For verification purposes, the simulation of the PID control loop was carried out with the Simulink tool as shown in Figure.3

Once the physical connections of the plant have been made, the local server is created. For this, the Mosquitto Broker must be installed on the Raspberry Pi single board computer, where the PID algorithm will be applied.

As can be seen in Figure (4), using the Node-RED software, we proceed to carry out the programming of the PID algorithm that will be implemented in the broker hosted on the raspberry Pi.

3.3 Results and Discussion

Once the controls have been simulated and implemented in the local network, the following results are presented: Fig. 50.6 shows the stabilization of the simulated PID control signal in a local wireless network, and Fig. 50.4 corrects the error in a time of 2.55 s. In Fig. 50.7, the performance of the PID control implemented in a local wireless network is shown, where by subjecting the system to a disturbance, it tends to correct the error in a time of 4 s. The delay that occurs in the PID control implemented in a practical way is considerable compared to the one carried out in the MATLAB Simulink simulation tool.

In turn, the network control may have a delay due to the occupation of the channel of the devices connected to it.

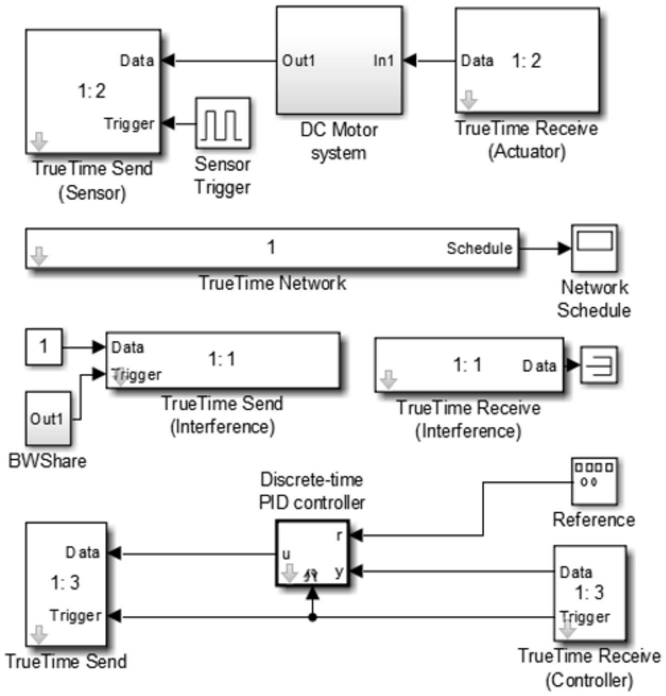


Figure 3: PID control simulation in WNCS.

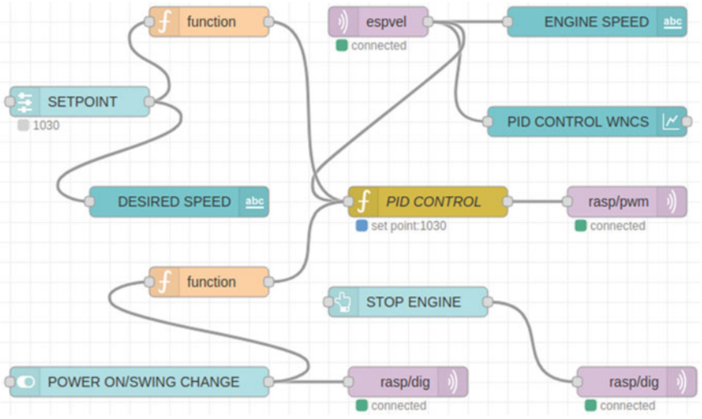


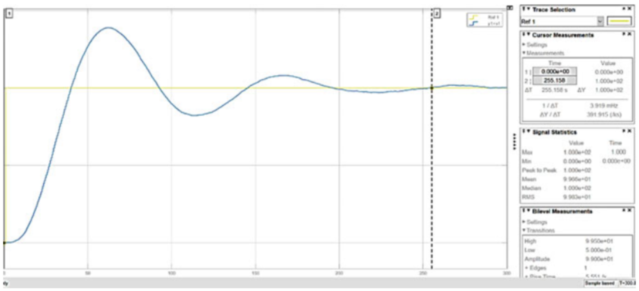
Figure 4: PID algorithm block code.

3.4 Conclusion

Through the design methodology used, it can be seen throughout this work that, thanks to the analysis of the technical aspects of the control systems, the effect of delay, channel occupancy and the programmed communication protocol were evidenced. Controller such as PID is applied for the manipulation of a DC motor, observing the LQG control a fast response to disturbances or to the change of the set point is obtained by 58.85

References:

[1] Y. Lu, Industry 4.0: A Survey on Technologies, Applications and Open Research Issues, vol. 6,



Trace Selection

Ref 1

Cursor Measurements

▸ Settings
▾ Measurements

	Time		Value
1	0.000e+00		0.000e+00
2	255.158		1.000e+02
ΔT	255.158 s	ΔY	1.000e+02
	1 / ΔT		3.919 mHz
	ΔY / ΔT		391.915 (/ks)

Signal Statistics

	Value	Time
Max	1.000e+02	1.000
Min	0.000e+00	0.000e+00
Peak to Peak	1.000e+02	
Mean	9.906e+01	
Median	1.000e+02	
RMS	9.903e+01	

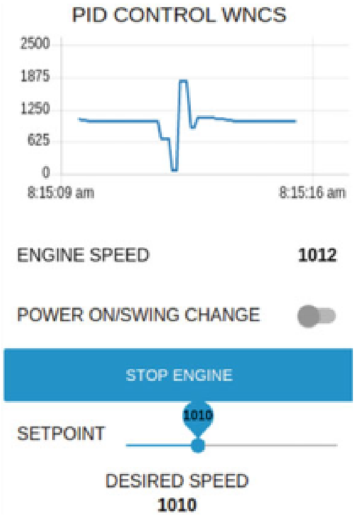
Bilevel Measurements

▸ Settings
▾ Transitions

High	9.950e+01
Low	5.000e-01
Amplitude	9.900e+01
+ Edges	1
+ Rise Time	5.551 /s

Sample based T=300.

Figure 5: Simulated PID control in WNCS.



ENGINE SPEED 1012

POWER ON/SWING CHANGE

SETPOINT 1010

DESIRED SPEED
1010

Figure 6: Real PID control implemented in WNCS.

pp. 1–10, 2017.

[2] L.S. Dalenogarea, G.B. Beniteza, N.F. Ayalab, A.G. Franka, The expected contribution of Industry 4.0 technologies for industrial performance. Sci. Direct 204, pp. 383–394, 2018.

[3] M. Kashyap, V. Sharma, N. Gupta, Taking MQTT and NodeMcu to Iot: Communication in Internet of Things. Sci. Direct, pp. 1611–1618, 2018.

[4] R.A. Light, Mosquitto: server and client implementation of the MQTT protocol. JOSS 265 (2017)

[5] M. Leki c, G. Gardasevi c, IoT Sensor Integration to Node-RED Platform IEEE, 2018.

[6] A,H.Samba , A.Yeremou Tamtsia , L.Nneme Nneme, I.J.Hermine Som ,and A.Houwe, "Design and Implementation of Wireless Iterative Learning Fault Estimation Algorithm with quantized measurement and Packet Losses"; IEEE Access, vol. 18, no. 1, Nov, 2021.

[7] C. Pillajo, R. Hincapié, Wireless Network Control Systems. Abya-Yala, Quito, 2018.

[8] B.D.S. Pozo, A.E.T. Bastidas, Sistema de control en red inalámbrico (wncs) mediante visión artificial y herramienta de mensajería Telegram, para sistemas de nivel. UPS, Quito, 2020.

[9] R. Vilanova, V.M. Alfaro, Control PID robusto: Una vision panorámica. Sci. Direct, vol. 8, pp.141–158, 2011.

[10] S. A. Herve, Y. T. Aurelien, and N. N. Leandre, "Optimal PD-type networked iterative learning algorithm based fault estimation for repetitive systems with delays, packet losses, sensor saturation and sensor failure," Int. J. Dyn. Control, pp. 1–13, Oct. 2021, doi: 10.1007/s40435-021-00871-8.

[11] C. Pillajo, R. Hincapié, Stochastic Control for a Wireless Network Control System, WNCS IEEE, 2020.

[12] S. A. Herve, "Computer Interface for Remote Teleoperated Mobile Robot Based on LabVIEW and Open Source Hardware," DESIGN, CONSTRUCTION, MAINTENANCE , vol. 2, pp. 115-124, 2022, doi: 10.37394/232022.2022.2.17.

[13] P.A.F. Agudelo, IoT como tendencia, retos y beneficios generados por la implementación de este tipo de soluciones en empresas colombianas del sector de industria y comercio , Bogotá, 2019.

[14] M.A.M. Sánchez, M.R.S. Herrera, A.M. Borrero, J.M.A. Márquez, Las ventajas del IoT en el diseño de laboratorios remotos: my GATEWAY Universidad de Huelva, Huelva, 2019.

[15] S. A. Herve, Y. T. Aurelien, H. S. I. Judith, and N. N. Leandre, "Networked iterative learning fault diagnosis algorithm for systems with sensor random packet losses, timevarying delays, limited communication and actuator failure: Application to the hydroturbine governor system," WSEAS Trans. Syst. Control, vol. 16, no. 20, pp. 244–252, May 2021, doi: 10.37394/23203.2021.16.20.