The proportional integral derivative control system on the mini conveyor with message queuing telemetry transport protocol based on the internet of things

P G Winoto[1], AWidodo[2] dan R R H P A Tjahyaningtyas[3]

Electrical Engineering Department, Engineering Faculty, Universitas Negeri Surabaya, Indonesia

[a]puthutwinoto@mhs.unesa.ac.id, [b]arifwidodo@unesa.ac.id, [c]hapsaripeni@gmail.com

Abstract. Motor speed controlling on the conveyor is necessary considered in order to control the speed of the motor at the desired setpoint. The most commonly used controller is a PID controller. The PID controller requires Kp, Ki, Kd and setpoint parameters to work. These parameters need to be tuned to work optimally on the system. The well-tuned parameters can be considered with the motor speed over time graph. Nowadays, parameters tuning and system response monitoring is done via cable with limited distance. The objective of this research is to make Proportional Integral Derivative control system on mini conveyor with Message Queuing Telemetry Transport protocol based on Internet Of Things so that the parameters tuning and system response monitoring can be done by Android smartphone from long distance and also know the system performance by calculating the Bit Error Rate (BER) of data transmitted from microcontroller and data received on Android smartphone. From 8 experiments performed, it found that the BER was 0 for each experiment. The 8 experiments were 10 seconds, 30 seconds, 60 seconds, 90 seconds, 120 seconds, 150 seconds, 180 seconds, and 210 seconds.

1. Introduction

The conveyor is a mechanical system that serves to move goods from one place to another place. The conveyor is widely used in the industry to move goods that are very large and continuously. The principle of conveyor work is to move goods above the belt. The belt is driven by a drive motor through the pulley. The drive Motor is given voltage to run at the desired speed (setpoint) [1].

To maintain the conveyor velocity steady according to the setpoint although working with varying loads, it requires a controller that can maintain the speed of the conveyor drive motor. The most commonly used controller is the PID controller (Proportional Integral Derivative). PID controllers require set point, Kp, Ki and Kd parameters to work. To assessing the PID controller parameter setting, monitoring and analysis of the PID controller output against time (system response) is required. In order to be easily understood, the system response is displayed in graphical form.

Tuning of PID parameters and monitoring of PID controller system response can be done in two ways: through the keypad and/or PC/computer connected to the controller via cable. This causes the tuning and monitoring distance to be limited to the cable used.

To facilitate the setup and monitoring of PID control system to be done anytime required an internet connection. The use of the internet for easy tuning and monitoring is the concept of the

Internet of Things (IoT). IoT is a concept that aims to expand the benefits of continuously connected internet connectivity [2].

There are several protocol options for implementing IoT, one of the options is Message Queuing Telemetry Transport (MQTT). The “SCADA protocol” and the “MQ Integrator SCADA Device Protocol” (MQIsdp) are both old names for what is now known as the MQ Telemetry Transport (MQTT). The protocol has also been known as “WebSphere MQTT” (WMQTT), though that name is also no longer used. The protocol specification has been openly published with a royalty-free license for many years, and companies such as Eurotech (formerly known as Arcom) have implemented the protocol in their products. MQTT is simple and lightweight, designed for limited devices and networks with low bandwidth, high latency or unreliable. So MQTT is suitable for "machine to machine" (M2M) communications and mobile applications with low bandwidth and battery power [3].

Bit error rate, BER is a key parameter that is used in assessing systems that transmit digital data from one location to another location. In digital transmission, the bit error rate is defined as the rate at which errors occur in a transmission system. This can be directly translated into the number of errors that occur in a string of a stated number of bits. The definition of bit error rate can be translated into a simple Equation (1) [4]:

$$BER = \frac{Number\ of\ bit\ error}{Total\ number\ of\ transferred\ bit} \tag{1}$$

2. Theory

2.1. STM32F103C8

The STM32F103xx medium-density performance line family from ST microelectronics incorporates high-performance ARM Cortex M3 32-bit RISC core operating at 72 MHz frequency, high speed embedded memories, and an extensive range of enhanced I/Os and peripherals connected to two APB buses. The devices operate from a 2.0 to 3.6 V power supply. They are available in both the -40 to +85°C temperature range and the -40 to +105°C extended temperature range. A comprehensive set of power saving mode allows the design of low power applications [5].

2.2. ESP8266 Wifi Module

ESP8266 is SoC (system on chip) based produced by Espressif Systems and it is a highly integrated chip designed to provide full internet connectivity in a small package[6]. It is also a serial Wifi wireless transceiver module for IoT. ESP-12-E is the latest generation of ESP8266 series module. The device size is small and operates on 3.3 V DC. It supports 802.11 b/g/n, built-in TCP/IP, TR switch, power amplifier and matching network, voltage regulator, and low power 32-bit CPU [7].

2.3. Rotary Encoder Sensor

A rotary encoder is an electro-mechanical device that converts an angular position or motion into a measurable electrical unit. The incremental/decremental encoder is chiefly used to measure a change of speed, distance, and position [8]. It consists of an optocoupler and binary disk so that the output of the sensor is square wave signal.

2.4. Proportional Integral Derivative (PID)

The PID controller is a three term controller and one of the earlier control strategies, starting from the beginning of the last century [9]. The PID controller has been the standard controller in industrial settings. The time constant formula of the PID controller is given as in Equation (2) [10].

$$G_C = K_P(1 + \frac{1}{T_i s} + T_d s) \tag{2}$$

Where K_i is Proportional gain which is used to increase the system response speed and reduce steady-state error [11], and K_i is an integral gain which used to eliminate the steady state error at all but it produces unwanted increase in the response overshoot, while K_d derivative gain used to reduce the system response overshoot [12]. Figure 1 shows the control system block diagram for the DC motor.

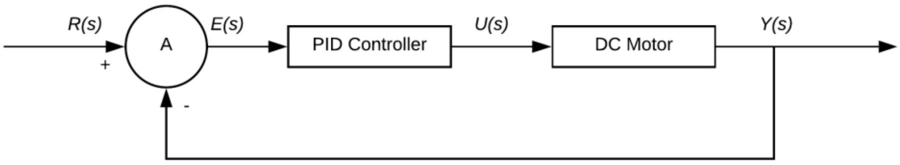

Figure 1. Block diagram of DC motor control system.

2.5. Message Queuing Telemetry Transport

MQTT is the proposed protocol in this paper. MQTT works with four major components. The foremost component of MQTT protocol is a broker. The broker is the same as a server, but here broker will automatically allow connecting the different devices with the use of three different Quality of Service (QoS) [13]. The second component of MQTT is the topic. The topic generally works as an identity of the information provided by the sensor as well as the devices which may get the information based on the same topics. Now to send and receive the data, MQTT protocol has another two terms named Publish and Subscribe. Publish means, it sends the data to the broker and Subscribe means, it will get the data from the broker. In addition on that, publishing information on the topic will send the information to the broker and the subscriber side if the subscribing topic is same as the publishing topic at the subscribing side, the information will be reached [14].

3. Method

The block diagram of the system is shown in Figure 2.

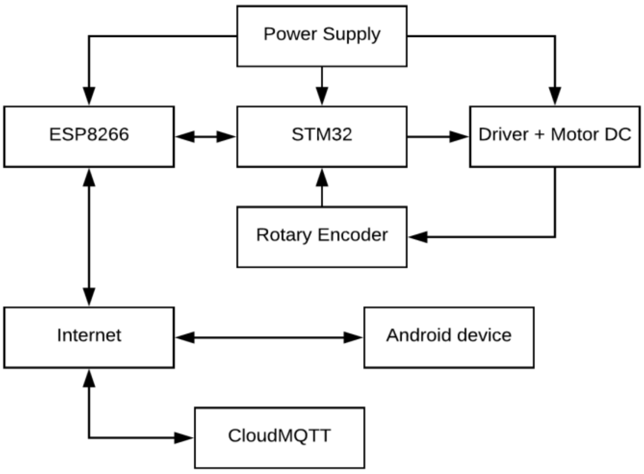

Figure 2. System diagram.

Figure 1 is a proportional integral control system design derivative on a mini conveyor with message queuing protocol telemetry transport based on the internet of things. Power Supply is used to supply voltage for the microcontroller, ESP8266 module, and DC motor + driver. STM32 as a PID controller algorithm implementation with setpoint, Kp, Ki, and Kd parameters which can be set from Android device. STM32 controls DC motor with PWM through DC motor driver. DC motor speed is measured with a rotary encoder sensor as feedback for a PID controller. The controller also sends the system response data with the MQTT protocol via the ESP8266 module. The ESP8266 module is connected to the internet in order to publish to the broker. The broker used is CloudMQTT, with a free broker service type. Then the data will be sent to the subscribed Android device. The data will be processed and displayed in graphical form.

3.1. Microcontroller Software Design

The design of the microcontroller software is shown in Figure 3 and 4.

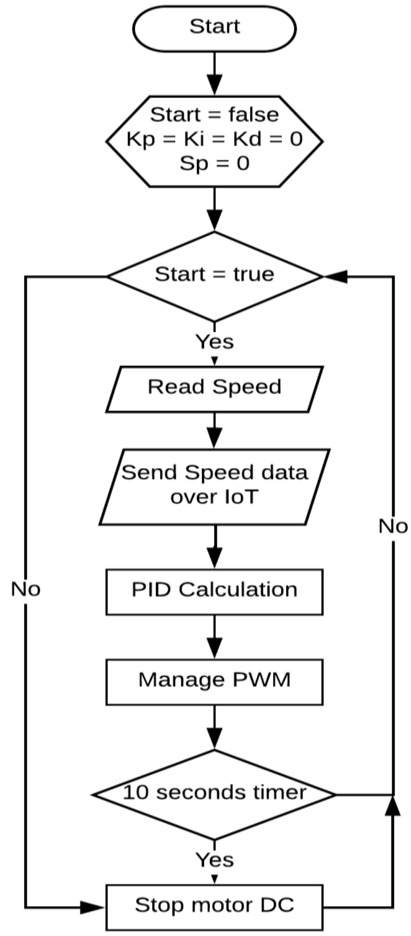

Figure 3. Main microcontroller software design.

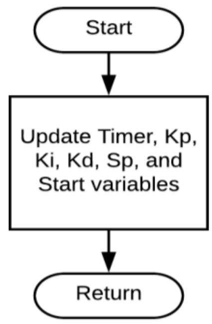

Figure 4. Interrupt microcontroller software design.

An overview of the microcontroller program is to check whether the start variable is true or not. If not, then the DC motor is conditioned in an off state. If yes, then do the PID calculation algorithm to determine the duty cycle PWM for DC motor rotates according to setpoint. Then the timer variable checked whether it is on or not and already 10 seconds or not. If it turns on and it has been 10 seconds, then the DC motor will be conditioned in an off state. Next, the program back to re-check the variable start and so on. To change the start variables and other PID parameters is done on the interrupt subroutine.

3.2. Microcontroller Software Design

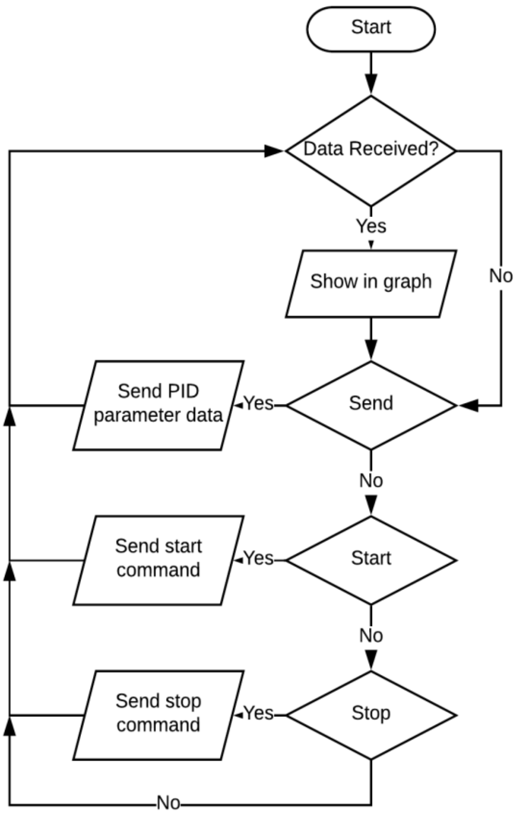

Figure 5. Android software design.

An overview of the program on an android device is to check whether the DC motor speed data is received or not. If yes, then the data is displayed in graphical form. Then check the Send button, Start whether pressed. If pressed then the Android app will send information to the microcontroller. If it meets the 10 second time it will send the stop command to the conveyor. Then back again to check if the DC motor speed data is received soon.

4. Results and Discussion

The DC motor speed transmitted from microcontroller to Android device over the internet. Transmitted data is monitored in PC by HTerm software so that the transmitted data can be compared to transmitted data in Android device. The example of transmitted data shown in Figure 6 and received data shown in Figure 7 and 8.

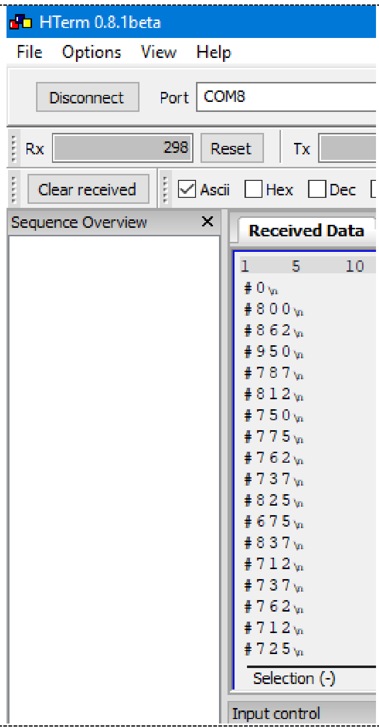

Figure 6. Monitored transmitted data.

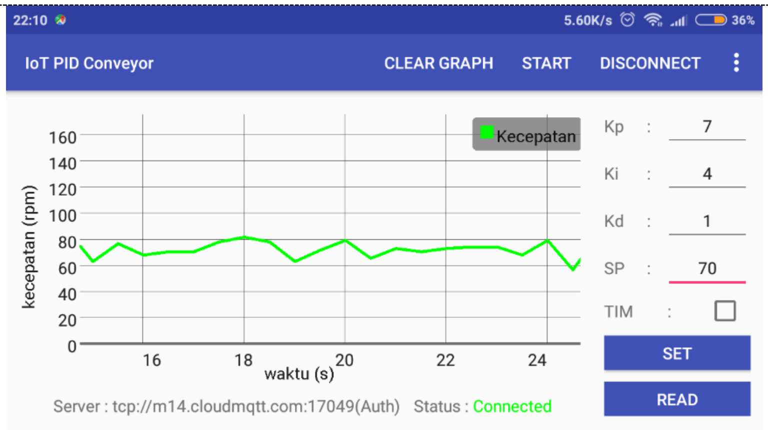

Figure 7. System response graphic.

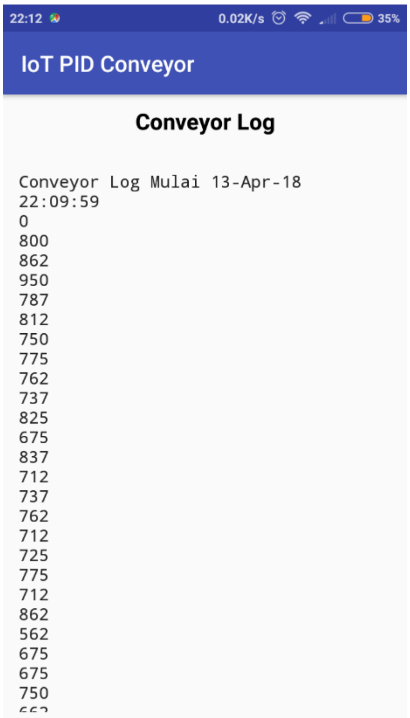

Figure 8. Logged received data.

System performance test done as much as 8 experiments. Each of the experimentss are duration of 10 seconds, 30 seconds, 60 seconds, 90 seconds, 120 seconds, 150 seconds, 180 seconds, and 210 seconds. Each test was performed with PID parameters Kp = 7, Ki = 4, Kd = 1, and Setpoint = 70 RPM. Here's a table of each experiment:

Table 1. The result of the experiments.

Number of Experiment	Duration (second)	Amount bit	Bit Error Rate
1	10	464	0
2	30	1424	0
3	60	2896	0
4	90	4344	0
5	120	5776	0
6	150	7184	0
7	180	8936	0
8	210	10040	0

5. Conclusion

Proportional Integral Derivative control system has been created with Message Queuing Telemetry Transport protocol based on Internet of Things on mini conveyor plant. DC motor speed is detected by rotary encoder sensor, then the data is processed by a microcontroller into speed data with RPM unit, then data sent to Android device via the internet with Message Queuing Telemetry Transport (MQTT) protocol. DC motor speed itself is controlled by a microcontroller with PID algorithm. PID manipulates the PWM duty cycle which is used to control the DC motor speed.

Data transmission from the microcontroller to Android device is done using the MQTT protocol. The broker used is a free broker provided by CloudMQTT.

On sending DC motor speed data from microcontroller to Android device, the result of the BER calculation is 0 for each 8 times experiment. The 8 experiments are 10 seconds, 30 seconds, 60 seconds, 90 seconds, 120 seconds, 150 seconds, 180 seconds, and 210 seconds.

References

[1] Suluhito. 2012 *Anatomi Sistem Roller Conveyor* retrieved from http://suluhmania.wordpress.com/2012/04/04/anatomi-sistemroller-conveyor
[2] Kurniawan 2016 *Purwa Rupa IoT (Internet of Things) Kendali Lampu Gedung* (Lampung:: Universitas Lampung)
[3] Anonim *Frequently Asked Question* retrieved from http://mqtt.org/faq
[4] Poole I *BER Bit Error Rate Tutorial and Definition* retrieved from http://www.radio-electronics.com/info/rf-technology-design/ber/bit-error-rate-tutorial-definition.php
[5] Anonim *STM32F103C8* retrieved from https://www.st.com/en/microcontrollers/stm32f103c8.html
[6] Anonim *ESP8266 Wi-Fi Module* retrieved from http://www.esp-ruino.com/ESP8266
[7] Anonim *ESP12E Brief Spec* retrieved from http://www.seeedstudio.-com/wiki/-images/7/7d/ESP-12E_brief_spec.pdf

[8] Abdulameer A, Sulaiman M, Aras M S M and Saleem D 2016 *Indon. J. Electr. Eng. Computer Sci.* **3 (2)** 343
[9] Astrom K J and Hagglund T 1995 *PID Controllers: Theory, Design, and Tuning* (USA: Instrument Society of America)
[10] Visioli A 2006 *Advances in Industrial Control-Practical PID Control* (London: Springer-Verlag London Limited)
[11] Owen F *Control Systems Engineering: A Practical Approach* retrieved from http://www.aoengr.com/SampleBook.pdf
[12] Phillips C L and Harbor R D 2000 *Feedback Control Systems Fourth Edition* (New Jersey: PrenticeHall, Inc.)
[13] Anonim *MQTT Essentials Part 1 to 9* retrieved from http://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels
[14] Locke D *MQ Telemetry Transport (MQTT) V3.1 Protocol Specification* retrieved from http://www.ibm.com/-developerworks/library/ws-mqtt