



Development of DC Motor Speed Control Using PID Based on Arduino and Matlab For Laboratory Trainer

Heru Supriyono¹, Fedrik Fajar Alanro¹, Agus Supardi¹

¹ Department of Electrical Engineering Universitas Muhammadiyah Surakarta, Surakarta, 57102, Indonesia

ARTICLE INFORMATION

Received: October 31, 2023
 Revised: March 23, 2024
 Accepted: March 31, 2024
 Available online: March 31, 2024

KEYWORDS

DC motor speed, DC motor control, PID, microcontroller, Arduino

CORRESPONDENCE

Phone: +62(0271)717417
 E-mail: Heru.Supriyono@ums.ac.id

A B S T R A C T

DC motors are widely used as propulsions, including in electric bicycles. The problem faced by students in the DC motor control laboratory working using software simulation is that they do not have practical learning experience using digital instruments. This article aims to develop a DC motor speed control that can be used to learn practical Proportional Integral Derivative (PID) control in the laboratory. The DC motor speed control was developed using a combination of Arduino UNO microcontroller and Matlab software. The PID method was used because it is still broadly studied and applied in industries. The test results showed that the developed trainer can work well with PID variable values that can be entered via the keypad, and DC motor transient responses can be displayed in Matlab. From the experimental results, it was found that the optimal PID variable values were $K_p=0.04$, $K_i=0.05$, and $K_d=0.004$, where the controller produced a low overshoot value, i.e., 0.73% of its set point and a settling time of 10.66 seconds. The test results of using the developed trainer in the Fundamental of Control Engineering laboratory work showed that the developed trainer gave students practical learning experience in designing PID control for DC motor speed by using digital equipment, i.e., microcontroller and actual DC motor as well as analyzing its corresponding transient response in Matlab software environment.

INTRODUCTION

A DC motor is a machine component that converts electrical energy into kinetic energy. The use of DC motors is widespread in many applications, mainly as an actuator, for example, in a mobile robot [1], an underwater vehicle [2], a fuel mixing process [3], a refrigerated cabinet [4], a solar tracker [5] and so on. Many studies were conducted on many aspects, such as control relating to its energy consumption [6] and its angular position control [7]. Relating to its speed, the two most controlled variables of a DC motor involve its speed and rotation direction, as reported by Glowinski et al. [8]. Speed controlling is very important in the application of DC motors, so that has attracted researchers to work in it, for example, by Tapia-Olvera et al. [9], Alejandro-Sanjines et al. [10] and other published works. Several studies have been carried out in terms of controlling the DC motor, for example, using fuzzy logic, Proportional Integral Derivative (PID) and Internal Model Control (IMC) methods [11], and so on.

Proportional Integral Derivative (PID) control has three control variables, namely proportional constant (K_p), integral constant (K_i), and derivative constant (K_d). In time-domain transient response, proportional control action affects the rise time, integral control action affects steady-state error, and differential control action affects overshoot/undershoot. PID control was studied widely for controlling a ball beam plant [12], a brushed DC motor

[13], and other plants. Because of its extensive use, PID control was also frequently used as a basic comparison of developing new control algorithms, such as reported by Velasco-Muñoz et al. [14]. A PID control has advantages such as minimizing or nullifying steady-state error and reaching reference set point [15]. Choosing the proper three PID constant combinations is called the tuning process. One of the most widely used tuning methods was the Ziegler-Nichols method, which was used by Lemus et al. [16]. Ziegler-Nichols method was also reported to provide PID constants with fast rise time [17]. Ziegler-Nichols method offers more straightforward computation than analytical tuning models, such as an auto-regression model [18] or a fuzzy-based tuning algorithm [19]. However, the transient response from PID control chosen using Ziegler-Nichols has the disadvantage that the maximum overshoot is still considerably significant, more than 20%.

Controlling a DC motor for practical learning or laboratory work could be performed in a complete simulation environment using software such as Sulityawati et al. [20] and Munadi et al. [21]. However, studying DC motor control using simulation only has disadvantages, such as students will not have experience in practical control by using digital instruments. Learning DC motor control in the laboratory using hardware systems has the advantage that students would gain experience in control design

by using digital instruments compared to simulation software, as also studied by Sozanski [22].

In Universitas Muhammadiyah Surakarta, DC motor speed control is learned in fundamental control engineering subjects, as well as both theory and laboratory works. In the laboratory work, students used to learn to control the DC motor speed by using Matlab simulation only. A low-cost, practical trainer based on microcontroller and Matlab software simulation is needed to increase student learning experience. This article aims to develop a low-cost DC motor speed control using a combination of a microcontroller-based system and Matlab software environment, which can be used as a trainer for studying DC speed motor control in the laboratory and to give practical learning experience by using digital instruments. The controlled DC motor speed output can be displayed using Matlab software for further analysis. PID control is consistently implemented in the closed-loop control architecture in the DC motor control, as depicted in Figure 1.

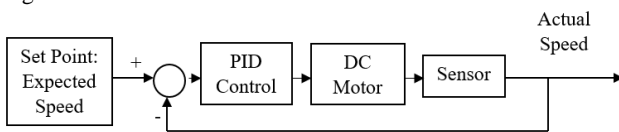


Figure 1. Block Diagram Representing Closed Loop Control of the DC Motor Speed

METHODS

General Architecture

The general architecture of the developed DC motor speed control for a laboratory trainer with Arduino-based systems can be seen in the block diagram presented in Figure 2.

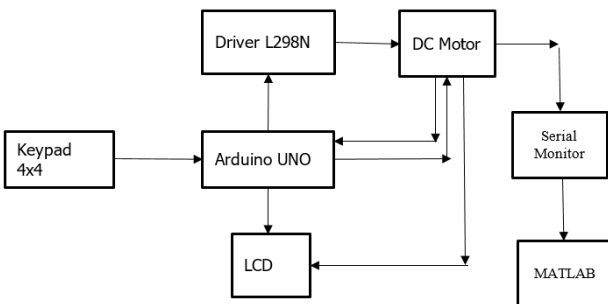


Figure 2. Block Diagram of the Developed DC Motor Speed Control

The block diagram shows that the developed system combines microcontroller-based and Matlab software. This combination was also used by previous researchers such as Syllignakis et al. [23]. Another model combination of microcontroller and Matlab for controlling a DC motor was reported by Latif et al. [24] and Rex and Praba [25], where Matlab was used for inputting PID variable constants and giving control commands to the microcontroller. The same concept was used, but LabVIEW software was used in the research reported by Saad et al. [26]. An Arduino UNO microcontroller was used in the developed system. This microcontroller board was developed based on the ATmega328P microcontroller used by researchers such as Adejumo and Azeez [27]. Arduino UNO is a popular microcontroller used in experiments such as for motor control

[28], controlling home appliances [29], and ultrasonic sensor-based parking assistant [30].

As a practical trainer, the set point value and the three PID parameters should be inputted and changeable all the time for trial and error learning. The keypad inputs K_p , K_i , K_d , and set point values. The Arduino UNO processes the inputs and then displays them on the LCD. The Arduino UNO then gives control commands based on these PID variable values to the DC motor via the driver, and the speed value is displayed on the LCD. In the developed system, the L298N motor driver differs from the L293D driver used by Divakar et al. [31]. A closed-loop control system managed the motor speed to remain stable as close as the set point. In addition to the LCD, the speed of the DC motor was also sent to the serial monitor, which will later be used to input the data into Matlab software.

Hardware Configuration

The hardware configuration of the proposed system used an Arduino UNO microcontroller, a DC motor, an L298N driver, an LCD, and keypad instruments, as seen in Figure 3.

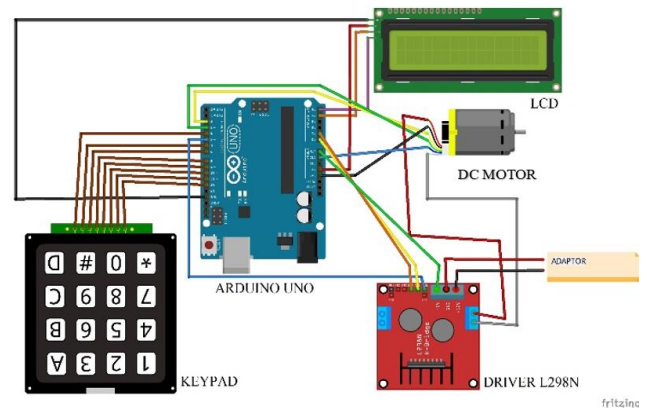


Figure 3. Hardware Configuration

This hardware configuration connects The keypad to pins 4, 6, 7, 8, 9, 10, 11, and 12 of the Arduino UNO microcontroller. The 5 DCV adapter used as the power source was connected to the LCD VCC and DC motor VCC pins, while the Arduino UNO ground was connected to the DC motor ground lead, LCD ground lead, and driver ground lead. Both SCL and SDA leads of the LCD were connected to pins A5 and A4, respectively. The IN 1 and 2 leads of the L298N driver were connected to pins A0 and A1 of the Arduino UNO microcontroller, respectively, and then the enabler section was connected to pin 5. The VCC driver section is connected to the Arduino Vin pin. In the DC motor, encoders A and B are connected to pin 2 and pin 3 of the driver, respectively, while out 1 and 2 leads of the DC motor are connected to the L298N driver output.

Software Design

The program written in the Arduino UNO generally has computation steps, as seen in the flow chart in Figure 4. Initially, the program determined which Arduino UNO microcontroller pins to use, and then it determined the library and all variables. After that, the microcontroller receives the input using the keypad. The input contains K_p , K_d , K_i , and set point values. The microcontroller then computes the PID control and gives the control command to the DC motor. The DC motor speed in

Revolutions per Minute (RPM) is then displayed on the LCD and sent to the serial connection monitor to be displayed in Matlab software. The developed system could be turned off by pressing the off button; otherwise, the system will wait for another input.

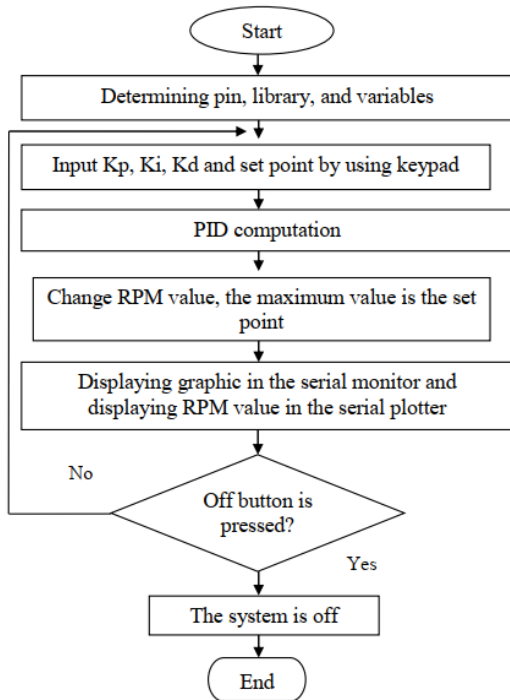


Figure 4. Flowchart of the Program

RESULTS AND DISCUSSION

Obtained System

The obtained DC motor speed control trainer can be seen in Figure 5 and 6. All hardware circuits were placed in a plastic container box to protect them from external environments such as dirt, water spray, dent, or vibration, which could damage the circuit. The controlled DC motor is placed outside the container box.

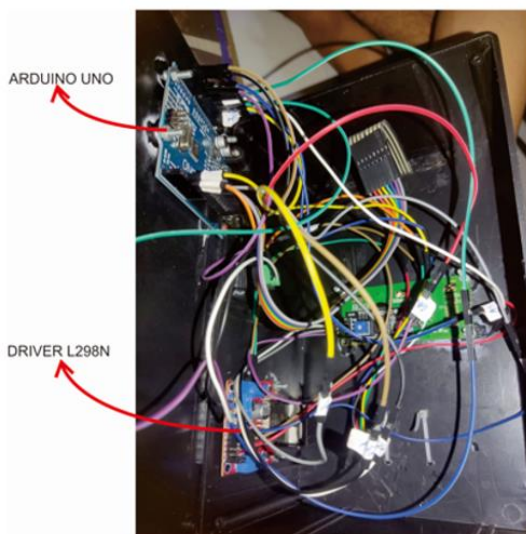


Figure 5. Obtained Hardware Circuit Placed in The Box



Figure 6. The Hardware Circuit of the Obtained System

System Testing

The obtained system was tested by using the following procedure. Firstly, the initial PID variables K_p , K_i , and K_d were manually calculated by using the Ziegler-Nichols method. Secondly, the K_p , K_i , and K_d found were inputted into the system using a keypad, and the obtained DC motor speed was displayed to the MATLAB software to analyze its transient response. After that, the PID control must be fine-tuned to get the best transient output using trial and error. The display of the LCD when inputting PID constants and the DC motor speed setpoint can be seen in Figure 7.

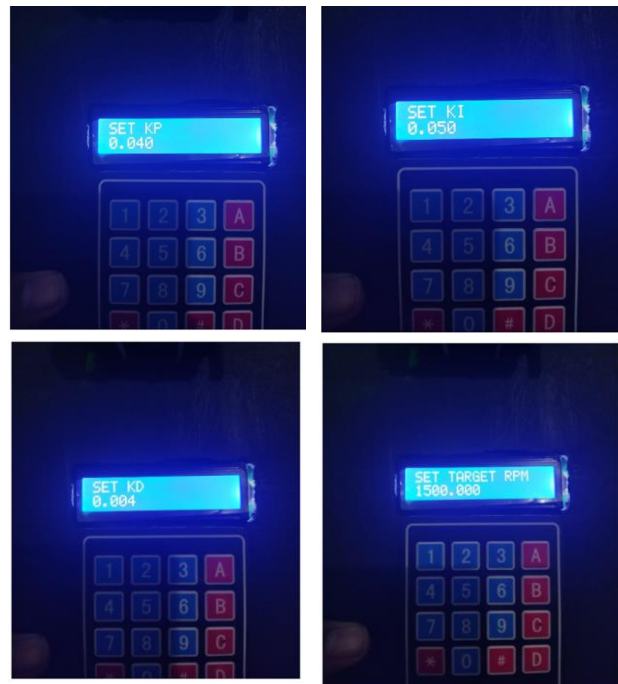


Figure 7. Example of Setting the PID Constants and Setpoint Values by Using Keypad.

After inputting the values of K_p , K_i , K_d , and set points and running the program, the obtained DC motor speed in rpm value can be seen on the serial monitor in the Arduino application; the DC motor speed data on the serial monitor needed to be copied to the spreadsheet software MS Excel, which was used in the

experiment. The spreadsheet contains three columns of data: column 1 contains the numbering, column 2 contains the set point value (in the experiment was set equal to 1500 rpm), and column 3 contains the actual DC motor speed value in rpm. The data taken is free, but it should not be too small. After that, the data was then inputted into the Matlab software by copying and pasting the data from the spreadsheet software into Matlab's workspace. The Matlab software displayed the transient response graphic, including its time-domain parameter overshoot, settling time, rise, and time values.

Figure 8 – 13 shows the transient response in several experiments, in which the blue line represents set points, and the red line represents RPM. In the first experiment, the K_p constant was set equal to 1, while the K_i and K_d values were 0. The graph (Figure 8) showed that the motor speed could not stably reach the set point value. There was still continuous oscillation in the range of $\pm 30\%$ below its set point value at 1500 rpm.

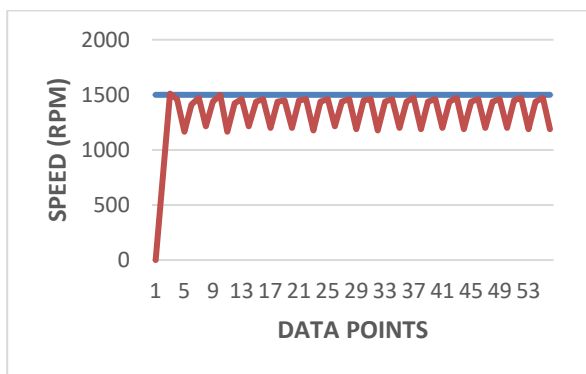


Figure 8. Transient Response of DC Motor Speed when $K_p=1$, $K_i=0$, and $K_d=0$

In experiment 2, as seen in Figure 9, the PID variables were changed to $K_p=1$, $K_i=0.5$, and $K_d=0$. The DC motor speed depicted in Figure 9 showed that the output reached the set point at 1500 rpm but had a relatively high overshoot, i.e., around 13.33 %, and there was still oscillation $\pm 12\%$ around the set point.

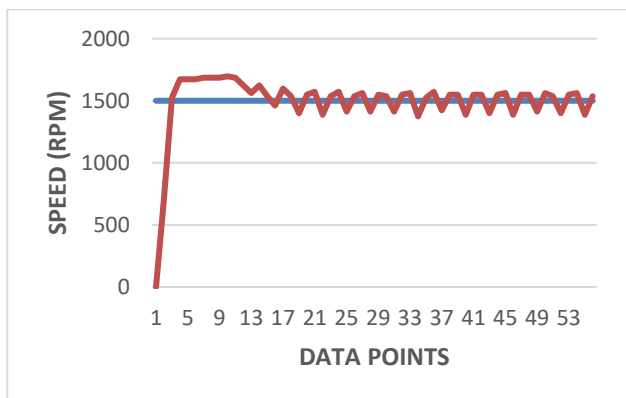


Figure 9. Transient Response of DC Motor Speed when $K_p=1$, $K_i=0.5$, and $K_d=0$

In experiment 3, the PID variables of $K_p=0.1$, $K_i=0.5$, and $K_d=0.7$ were used. The transient response depicted in Figure 10 showed that the overshoot is still around 13%, but the steady state condition has a minimal error with oscillation of less than $\pm 1\%$ around the set point 1500 rpm. It is shown in Figure 10 below.

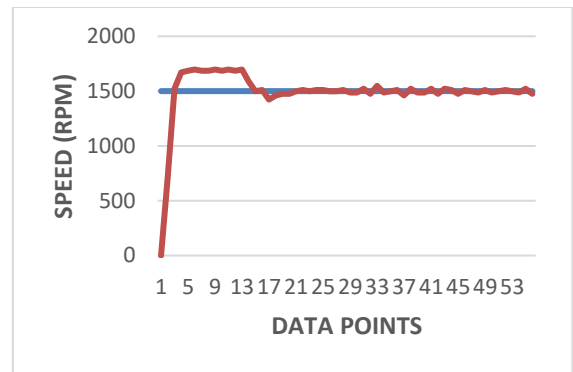


Figure 10. Transient Response of DC Motor Speed when $K_p=0.1$, $K_i=0.5$, and $K_d=0.7$

In experiment 4, shown in Figure 11, the PID variables were changed to $K_p=0.05$, $K_i=0.06$, and $K_d=0.005$. The transient response presented in Figure 11 showed that the overshoot was already minimal, less than 5%, and the steady state condition still has a ripple in considerably minimal magnitude, $\pm 0.5\%$, around the set point of 1500 rpm.

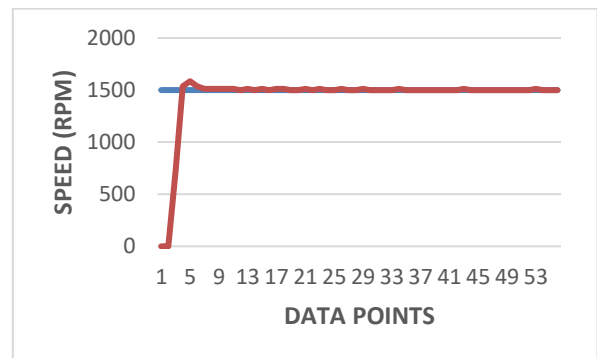


Figure 11. Transient Response of DC Motor Speed when $K_p=0.05$, $K_i=0.06$, and $K_d=0.005$

In experiment 5, the PID constants of $K_p = 0.03$, $K_i = 0.04$, and $K_d = 0.005$ were used. The obtained graphic depicted in Figure 12 was excellent; there was no overshoot, and the resulting speed was stable with a steady-state error lower than $\pm 0.2\%$. However, the transient response became slower compared to the results of experiment 4, as can be seen in Figure 12 below.

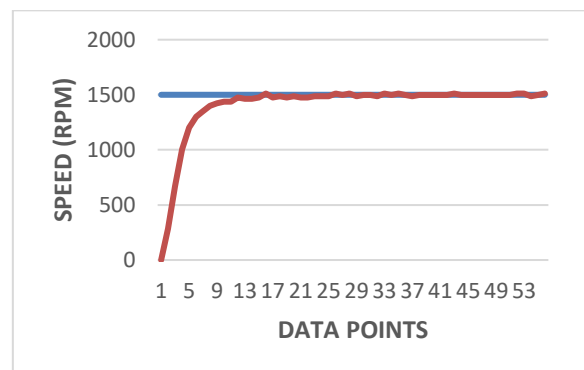


Figure 12. Transient Response of DC Motor Speed when $K_p=0.03$, $K_i=0.04$, and $K_d=0.005$

In experiment 6, PID variables were set to $K_p=0.04$, $K_i=0.05$, and $K_d=0.004$ were used. The obtained transient response depicted in Figure 13 showed that the result was outstanding because the

graph had no overshoot and was faster than the result of experiment 5. The steady-state error was tiny, less than $\pm 1\%$.

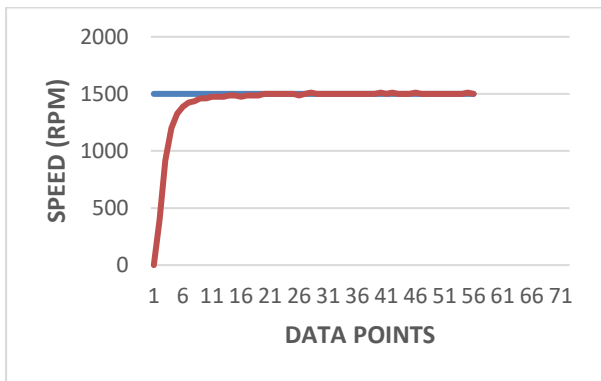


Figure 13. Transient Response of DC Motor Speed when $K_p=0.04$, $K_i=0.05$, and $K_d=0.004$

The results of experiments 1-6 are summarized in Table 1. It can be seen that the PID constants of experiment 6 produced the most optimal transient response, i.e., it has the smallest maximum overshoot and short settling time. However, it was still longer compared to experiment 4. The best PID output result, which produced the smallest overshoot value, was obtained at $K_p = 0.04$, $K_i = 0.05$, and $K_d = 0.004$, with an overshoot value of 0.73%. In contrast, the most significant overshoot was obtained with $K_p = 0.1$, $K_i = 0.5$, and $K_d = 0.7$, with an overshoot value of 13.13%. A small overshoot value close to 0 proves that the motor moves well and is stable.

Table 1. Summary of Experiment 1-6 Results

Exp.	K_p	K_i	K_d	Overshoot (%)	Settling Time (s)
1.	1	0	0	0,73	NaN
2.	1	0,5	0	13,13	NaN
3.	0,1	0,5	0,7	13,13	37,13
4.	0,05	0,06	0,005	5,6	6,24
5.	0,03	0,04	0,005	0,73	14,67
6.	0,04	0,05	0,004	0,73	10,66

Comparison to Previous Works

Using the obtained DC motor control system, students could learn practical designing PID control using digital instruments and Matlab software compared to control in [16] [17], which controlled the DC motor using Ziegler-Nichols in a whole simulation environment. Compared to the control system in [23] [27], the obtained DC motor control is more suitable for practical laboratory works because the developed system has a feature to record transient response data in a spreadsheet file, which can then be used for displaying it and further analysis especially its maximum overshoot and settling time. Besides, the article in [27] used a scale button to increase or decrease the DC motor speed without a keypad to input a particular set point value. Moreover, works in [23] [27] also did not have features to input the K_p , K_i , and K_d parameters for experimental purposes.

Testing of the Obtained System for Laboratory Works

The obtained system was tested for Fundamental Control Engineering laboratory works. This system replaced the full simulation laboratory works module used previously. The observation results suggested that using the proposed system, students had better learning experience in practicing designing PID control for DC motor speed by using digital equipment, i.e. microcontroller and actual DC motor, and analyzing its corresponding transient response in Matlab software environment.

CONCLUSIONS

A DC motor speed control using PID control based on Arduino UNO and Matlab, which can be used as a trainer in practical laboratory works, has been developed. The developed system has features involving a keypad for inputting set points and three PID parameter values for experimental purposes. The test results showed that based on the overshoot, settling time, and rise time values, the best PID controller parameters values, namely $K_p = 0.04$, $K_i = 0.05$, and $K_d = 0.004$, resulting an overshoot value of 0.73% and settling time of 10.66 seconds. Using this developed system would increase students' learning experience in practicing designing PID control for DC motor speed by using digital equipment, i.e. microcontroller and actual DC motor, and analyzing its corresponding transient response in a Matlab software environment.

ACKNOWLEDGMENT

The authors would like to thank the DRTPM of the Ministry of Education, Culture, Research and Technology Republic Indonesia for funding this research through the Fundamental Research scheme for 2023.

REFERENCES

- [1] H. Supriyono, P. A. N. Rochman, and M. O. Tokhi, O. (2021). "IoT technology involving wheeled line follower robot for restaurant services automation," *ELKOMIKA*, vol. 9, no. 1, pp. 100-113, 2021.
- [2] T. Sands, "Control of DC motors to guide unmanned underwater vehicles," *Applied Science* 2021, vol. 11, pp. 1-13, 2021.
- [3] A. A. Mahadi, B. Santoso, Ubaidillah, and B. W. Lenggana, "Redesign of water-diesel emulsion fuel mixer," *Applied Research and Smart Technology*, vol. 3, no. 1, pp. 18-28, 2021.
- [4] K. H. Chao, L. Y. Chang, and C. Y. Hung, "Design and control of brushless DC motor drives for refrigerated cabinets," *Energies* 2022, vol. 15, pp. 1-28, 2022.
- [5] M. Engin, "Controller design for parallel mechanism solar tracker," *Machines* 2023, vol. 11, pp. 1-17, 2023.
- [6] P. Zur, "Combination of a DC motor controller and telemetry system to optimize energy consumption," *Sensors* 2023, vol. 23, pp. 1-12, 2023.
- [7] L. S. Mezher, "Position control for dynamic DC motor with robust PID controller using Matlab," *International Journal of*

- Advanced Trends in Computer Science and Engineering, vol. 8, no. 3, pp. 936-942, 2019.
- [8] S. Glowinski, S. Pecolt, A. Błazejewski, and B. Młynski, "Control of brushless direct-current motors using bioelectric EMG signals," *Sensors* 2022, vol. 22, pp. 1-13, 2022.
- [9] R. Tapia-Olvera, F. Beltran-Carbajal, O. Aguilar-Mejia, and A. Valderrabano-Gonzalez, "An adaptive speed control approach for DC shunt motors," *Energies* 2016, vol. 9, pp. 1-16, 2016.
- [10] U. Alejandro-Sanjines, A. Maisincho-Jivaja, V. Asanza, L. L. Lorente-Leyva, and D. H. Peluffo-Ordóñez, "Adaptive PI controller based on a reinforcement learning algorithm for speed control of a DC motor," *Biomimetics* 2023, vol. 8, pp. 1-26, 2023.
- [11] Md. A. Ahmad, K. Kishor, and P. Rai, "Speed control of a DC motor using controllers," *Automation, Control and Intelligent Systems*, vol. 2, no. 6, pp. 1-9, 2014.
- [12] M. Aviles, J. Rodríguez-Reséndiz, J. Pérez-Ospina, and O. Lara-Mendoza, "A comprehensive methodology for the development of an open source experimental platform for control courses," *Technologies* 2023, vol. 11, pp. 1-25, 2023.
- [13] R. Bitriá, and J. Palacín, "Optimal PID control of a brushed DC motor with an embedded low-cost magnetic quadrature encoder for improved step overshoot and undershoot responses in a mobile robot application," *Sensors* 2022, vol. 22, 7817, 1-24, 2022.
- [14] H. Velasco-Muñoz, J. E. Candelo-Becerra, F. E. Hoyos, and A. Rincón, "Speed regulation of a permanent magnet DC motor with sliding mode control based on washout filter," *Symmetry* 2022, vol. 14, pp. 1-25, 2022.
- [15] A. Ma'arif, N. R. Setiawan, and E. S. Rahayu, "Embedded control system of DC motor using microcontroller arduino and PID algorithm," *IT Journal Research and Development (ITJRD)*, vol. 6, no. 1, pp. 30-42, 2021.
- [16] J. P. T. Lemus, G. C. V'elez, and N. J. C. Rodríguez, "PID controller design for DC motor," *Contemporary Engineering Sciences*, vol. 11, no. 9, pp. 4913 – 4920, 2018.
- [17] A. Abdulameer, M. Sulaiman, M. S. M. Aras, and D. Saleem, "Tuning methods of PID controller for DC motor speed control," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 3, no. 2, pp. 343-349, 2016.
- [18] J. A. Niembro-Ceceña, R. A. Gómez-Loenzo, J. Rodríguez-Reséndiz, O. Rodríguez-Abreo, and A. Odry, "Auto-regression model-based off-line PID controller tuning: an adaptive strategy for DC motor control," *Micromachines* 2022, vol. 13, pp. 1-21, 2022.
- [19] P. Kofinas, and A. I. Dounis, "Fuzzy Q-learning agent for online tuning of PID controller for DC motor speed control," *Algorithms* 2018, vol. 11, 1pp. 1-13, 2018.
- [20] N. Sulistyawati, Fahmizal, and I. Nathasya, "DC motor speed control with buck converter uses full state feedback-pole placement," *ELKOMIKA*, vol. 9, no. 2, pp. 415-429, 2021.
- [21] Munadi, M. A. Akbar, T. Naniwa, and Y. Taniai, "Model reference adaptive control for DC motor based on simulink," 2016 6th International Annual Engineering Seminar (InAES), Yogyakarta, Indonesia, pp. 1-6, 2016.
- [22] K. Soza'nski, "Low cost PID controller for student digital control laboratory based on arduino or STM32 modules," *Electronics* 2023, vol. 12, pp. 1-23, 2023.
- [23] J. Syllignakis, P. Panagiotakopoulos, and E. Karapidakis, "Automatic speed controller of a DC motor using arduino for laboratory applications," *Engineering and Industry Series Volume Power Systems, Energy Markets and Renewable Energy Sources in South-Eastern Europe 2016*, pp. 351-364, 2016.
- [24] A. Latif, A. Z. Arfianto, H. A. Widodo, R. Rahim, and E. T. Helmy, "Motor DC PID system regulator for mini conveyor drive based-on Matlab," *Journal of Robotics and Control (JRC)*, vol. 1, no. 6, pp. 185-190, 2020
- [25] S. R. Rex, and M. S. R. Praba, "Speed control of DC motor with PWM using microcontroller in hardware in loop," *International Journal of Engineering & Technology, International Journal of Engineering & Technology*, vol. 7, no. 3, pp. 116-119, 2018.
- [26] M. Saad, A. H. Ahmed, and M. Al Sharqawi, "Real time DC motor position control using PID controller in LabVIEW," *Journal of Robotics and Control (JRC)*, vol. 2, no. 5, pp. 342-348, 2021.
- [27] O. O. Adejumo, and W. A. Azeez, "Development of a microcontroller based DC motor Speed Control System," *World Journal of Engineering and Technology* 2019, vol. 7, pp. 634-639, 2019.
- [28] N. Shaharudin, M. Z. Hasan, and S. M. Noor, "Direct Current (DC) motor speed and direction controller," *Journal of Physics: Conference Series*, 2129 (2021) 012035, pp. 1-6, 2021.
- [29] R. N. Rohmah, F. N. Alwi, and Jatmiko, "Equipment for monitoring and controlling of household electricity consumption in a IoT-based smart house," *Emitor*, vol. 22, no. 01, pp. 34-38, 2022.
- [30] Supriyono and Marjuki, "Ultrasonic sensor parking assistant with arduino uno," *International Journal of Advanced Research in Engineering and Technology (IJARET)*, vol. 11, no. 5, pp. 26-33, 2020.
- [31] A. Divakar, J. Joseph, J. T. George, N. N. Prabhu, and A. Nabi, "A study on DC motor operations and speed control using microcontroller," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 5, no. 4, pp. 2460-2464, 2016.

AUTHORS BIOGRAPHY

Heru Supriyono

He is a full time lecturer at the Department of Electrical Engineering Universitas Muhammadiyah Surakarta. His main researches are in the areas of automation, robotics, control, and computer system.

Fedrik Fajar Alanro

He is full time student at the Department of Electrical Engineering Universitas Muhammadiyah Surakarta. His main interest is in power system and industrial automation

Agus Supardi

He is a full time lecturer at the Department of Electrical Engineering Universitas Muhammadiyah Surakarta. His main expertise falls in the area of electrical power systems, electrical machines (both generator and motor), and insulation systems.