

Fuzzy Logic Controller Implementation for Motor DC Control Position With Real-Time Operating System

Martin¹, Dodi Budiman Margana², Endang Habinuddin³
^{1,2,3}Politeknik Negeri Bandung, Jawa Barat, Indonesia
Email: martin@polban.ac.id¹, dodi.budiman@polban.ac.id²,
endang.h@polban.ac.id³

Abstract

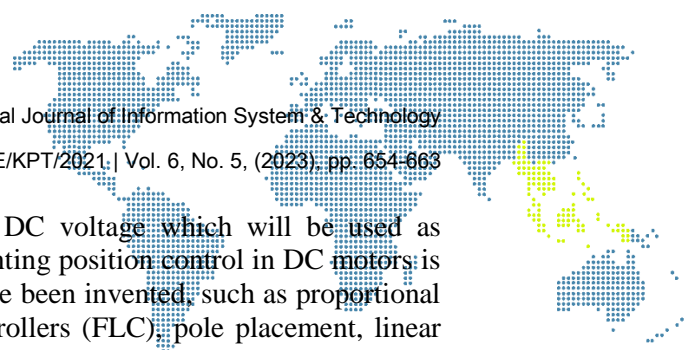
A fuzzy Logic Controller (FLC) is a control system that expresses in linguistics, there are several ways to defuzzification such as Sugeno. This paper proposes FLC to control the Motor DC Module position within -180 to 180 degrees using Real-Time Operating System (RTOS) running in the microcontroller. The FLC design based on input errors and changes of errors get from the feedback module, and defuzzification uses Sugeno rules. The FLC design simulates in the simulator program with the motor dc ideal model to show the output response transient. Experimental is done by using step input and variable inputs, step input is used to calculate the output response transient, and variable input to show the reliability of the control scheme. The output response shows that the motor DC control position has 0% of overshoot, 0.9 seconds of settling time, and 2.7769 rise time from position -80 to 0. The FLC success controls the position motor, and RTOS can be implemented in a microcontroller based on ATmega 2560 series.

Keywords: Fuzzy Logic Controller, Motor DC, RTOS, Control Position

1. Introduction

A fuzzy logic controller (FLC) is a control system based on logic, which is close to the human way of thinking expressed in linguistics [1], [2]. It provides an effective way to capture approximate, exact properties of the real world. Seen in this perspective, an important part of a fuzzy logic controller (FLC) is a set of linguistic control rules related to the concept of dual implication membership function and inference composition rules. FLC produces linguistic algorithms derived from expert knowledge and based on experiments on an automated control system. In particular, the FLC methodology is useful when the processes of a plant are too complex to be analyzed by conventional quantitative techniques or when available sources of information are interpreted qualitatively, which is imprecise. The feedback module used for the position control plant system in the trials conducted has several drawbacks including high Deadband motors and position sensors that are not linear and have Deadband [3]. The problems above can be solved by using FLC.

FLC is commonly used to control uncertainty areas that are hard to control by a traditional control method, such as inverted pendulum [4], non-linear control, etc. However, FLC offers methods that can solve uncertain areas or grey areas in certain areas, but it takes a lot of computational processes for microcontrollers can take. A real-time operating system (RTOS) can optimize the process of computational process in a microcontroller which can't be done by a traditional loop program [5]. DC motors are important components that are widely used in industry, automotive, or everyday life. The use of these components such as the car remote control, and conveyor belt drive, can be used as a generator. DC motors are one of the most frequently used electric drives and are also being developed because DC motors are easy to control using linear controls [6]. One of the many developments that have been carried out is to regulate or control the position of the DC motor movement. The position of the DC motor movement can be known and controlled by placing sensors to detect changes that occur. The encoder sensor is one of the components to read the movement of the speed or position of the DC motor rotation.



The sensor produces an output in the form of a DC voltage which will be used as feedback from the device to be controlled. Implementing position control in DC motors is a challenge with the many types of controls that have been invented, such as proportional integral derivative (PID) controls, fuzzy logic controllers (FLC), pole placement, linear squared regulators (LQR), and linear squared. gaussian (LQG) [7], [8].

Research on DC motor control and its application has been carried out a lot. Researchers [9] researched modeling and positioning DC motors. The researcher made a comparison between PID and LQR controls in controlling the position of the DC motor that has been modeled. Research [10] displays speed control of a brushless DC (BLDC) motor using a hall-effect position sensor as a feedback to determine the speed of a BLDC motor. The results of this study indicate that the position sensor can be used as a speed detector for BLDC motors. Fuzzy logic is also used in research [11]–[15] to adjust the position of DC motors, although there are differences in the fuzzy logic used in the three research titles. Research [11] uses a fast fuzzy controller to control the position of a DC motor that is implemented in a field-programmable gate array (FPGA). Then research [12] uses an adaptive fuzzy controller to control DC motors by optimizing the control of various types of loads on the system. Furthermore, researchers [13] conducted research to control the position of DC motors using fuzzy logic combined with proportional-integral (PI) control. Fuzzy logic is used to determine the best PI parameters that will be implemented in the DC motor position control. As for the application of DC motor position control, one of them is for the solar tracking system. The sliding mode controller is used to adjust the position of the DC motor so that it can follow the direction of sunlight it can be used optimally on solar panels [16]. The LQG controller in combination with a Kalman filter can also be used as a DC motor position control. Optimization of the feedback gain value was carried out with a Kalman filter as in research [17]. Another controller that can be used is the pole placement or what can be called a state feedback controller. Strengthening is given to system feedback and system input where reinforcement is obtained based on mathematical calculations obtained after knowing the system model to be controlled. Finally, studying [18] and [19] designs state feedback controls by applying control to different types of DC motors. This research is a continuation of previous research, where the control of dc motors uses a state feedback controller with an RTOS [20].

2. Research Methodology

2.1. Real-Time Operating System

Most dynamic system controls will have problems at program execution time. In control, various orders must be received at the right time to ensure operations run smoothly and are fulfilled real-time [21]. A real-time operating system (RTOS) is a multitasking operating system intended for real-time applications.

RTOS can make the system work in real-time, but the results obtained cannot be guaranteed to work in real-time. This is influenced by several factors and development carried out by the developer well. The basic character of an RTOS is a system that has several consequences that can affect system performance if the deadline for executing tasks is not met. Based on the characteristics itself, RTOS is divided into two parts, namely soft and hard RTOS. Soft RTOS is a system that in carrying out the system has a real-time percentage completion rate of less than 100%, while a hard RTOS is a system that can complete tasks in a predetermined time with a success rate of 100%. The RTOS explanation will be divided into three main points, including tasks, kernels, and mutual exclusion or processing at the same time.

A task is an object/program that is executed and executed as if it had its own CPU. The division of work in applications into several tasks is one of the design processes of an RTOS. Each task is a circle that will continue to repeat if the application is running. During the process, the task will experience three states as shown in Figure

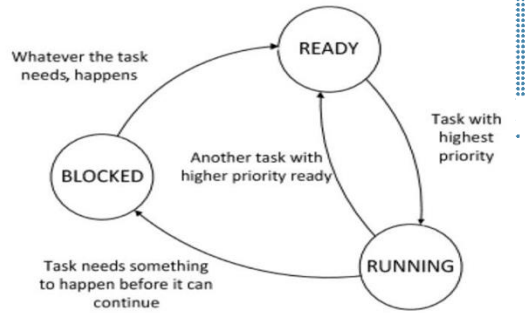


Figure 1. The state cycle on an RTOS [21]

The kernel is one part of a multitasking system that has a function as the management of all tasks, manages the communication of each task, and most importantly, manages the timing for the central processing unit (CPU) so that no CPU crashes occur. The kernel itself consists of two types, namely, non-preemptive and preemptive. The system used will use a preemptive RTOS.

Preemptive kernels are widely used to create applications with RTOS. This is because preemptive kernels have a better response than non-preemptive kernels. So it can be concluded that the preemptive kernel always prioritizes the task with the highest priority that is ready to be executed. With a preemptive kernel, system response can be optimal and the time to run the task with the highest priority can be determined, in contrast to the non-preemptive kernel which cannot be determined. Figure 2 displays the working principle of the preemptive kernel.

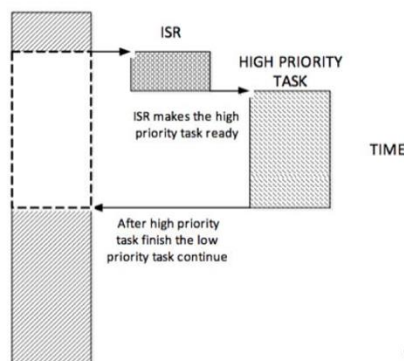


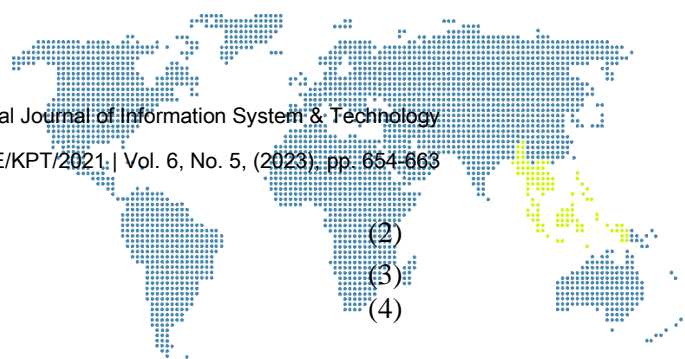
Figure 2. The working principle of the preemptive kernel [21]

Mutual exclusion is a condition where each resource is given at exactly one time. There are several methods to support mutual exclusion, including semaphores and message queues. The message queue is used to send one or more messages to the message queue used to prevent errors due to shared data.

2.2. Motor DC Feedback Model

DC motors are commonly used in various industrial applications, used as drives in production machines or lifting equipment. It is also used in portable equipment such as printers and others. As input for a dc motor is electric power, namely voltage, and electric current, then the output of the motor is mechanical power, namely torque and rotor speed. In controlling dc motors, it is necessary to know the characteristics of these variables as a function of time in the transient and steady state conditions, so that a model can then be built that can be used as material for further analysis such as designing a controller or filter. Mechanical equations and Electrical equation :

$$T_e - T_l = J \frac{d\omega_m}{dt} + B\omega_m, \tag{1}$$



$$Vm = Ra \cdot Ia + La \frac{dI_a}{dt} + V_{b\phi} \quad (2)$$

$$T_\phi = K_t \cdot I_a \quad (3)$$

$$V_{b\phi} = K_e \cdot \omega_m \quad (4)$$

$T_i = 0$ (At no-load condition)

In Laplace domain, equations (1) and (2) can be written as follows :

$$K_t \cdot I_a = (Js + B)\omega_m \quad (5)$$

$$Vm = (Ra + SL_a)I_a + K_e \cdot \omega_m \quad (6)$$

Equation of speed to input voltage

$$\frac{\omega_m(s)}{Vm(s)} = \frac{Kt}{JmLms^2 + (JmRm + DmLm)s + RmDm + KmKt} \quad (7)$$

In general, Lm is small enough compared to Rm , so the equation can be simplified as follows:

$$\frac{\omega_m(s)}{Vm(s)} = \frac{K}{\tau s + 1} \quad (8)$$

Equation of position concerning input voltage

$$\frac{\theta_m(s)}{Vm(s)} = \frac{\omega_m(s)}{Vm(s)} \times \frac{1}{s} = \frac{K}{\tau s^2 + 1} \quad (9)$$

To find out the value of K and τ then identify the DC motor plant , by giving a step input to the input plant and the output is acquired we can get a 1st order value using the toolbox identification in Matlab as show in Figure 3:

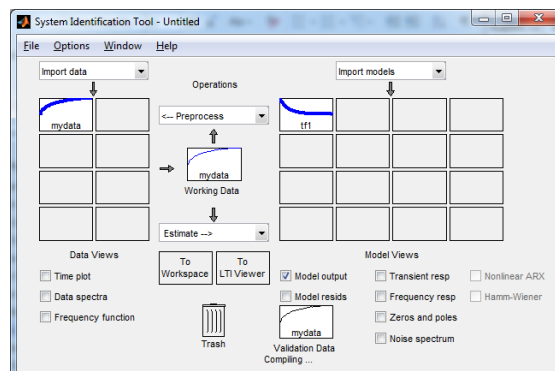


Figure 3. Identification Motor DC using APLICATION

Meanwhile, the transfer function of the system can be seen in Figure 4 and 5.

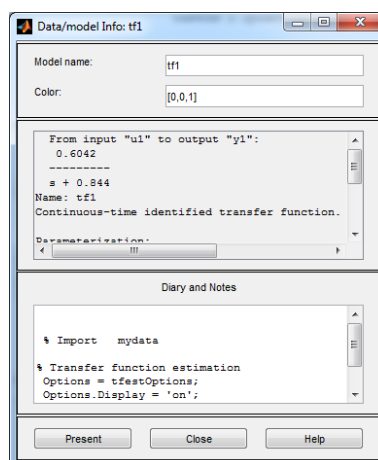


Figure 4. Transfer function of system identification

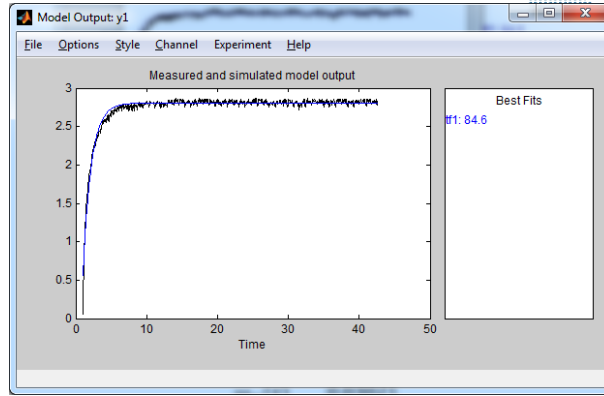


Figure 5. Plant validation

The value of the transfer function is obtained from the plant used

$$\frac{\omega_m(s)}{V_m(s)} = \frac{0.6042}{s+0.844} \tag{10}$$

Equation of position concerning input voltage

$$\frac{\theta_m(s)}{V_m(s)} = \frac{0.6042}{s+0.844} \times \frac{1}{s} = \frac{0.6042}{s^2+0.844s} \tag{11}$$

$$G_{h1} = \frac{H(s)}{X^*(s)} = (1 - e^{-Ts})^2 \frac{Ts+1}{Ts^2} = \left(\frac{1-e^{-Ts}}{s}\right)^2 \frac{Ts+1}{T} \tag{12}$$

System switch function continuous

$$\frac{\theta_m(s)}{V_m(s)} = \frac{0.6042}{s^2+0.844s} \tag{13}$$

With using Zero Order Hold, obtained function over system discrete

$$\frac{0.0007447z+0.0007342}{z^2-1.959z+0.9587} \tag{14}$$

2.3. Fuzzy Logic Controller Design

Figure 6 is a block of FCL that was built, in the first block in the form of fuzzification where the input crips value is in the form of errors and delta-errors obtained from the plant's feedback signal. In the second block is a rule-base containing implication rules between the two inputs, 25 rule-base rules are built. The last block is defuzzification where the method used in this block is sugeno.

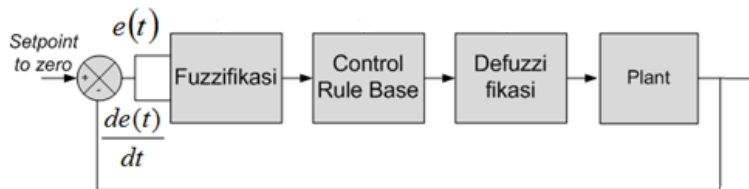


Figure 6. Diagram of the FLC system control

The Fuzzy controller has 2 inputs: input errors and error changes (d_error). The motor position error input is selected with an angle range between -180° to 180° . The membership input error function is formed from 5 memberships, namely, NB (Negative Big), NS (Negative Small), Z (Zero), PS (Positive Small) and PB (Positive Big). The membership error function can be seen in Figure 7.

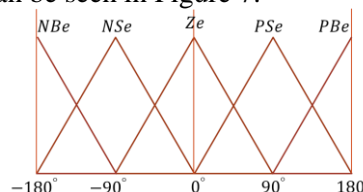
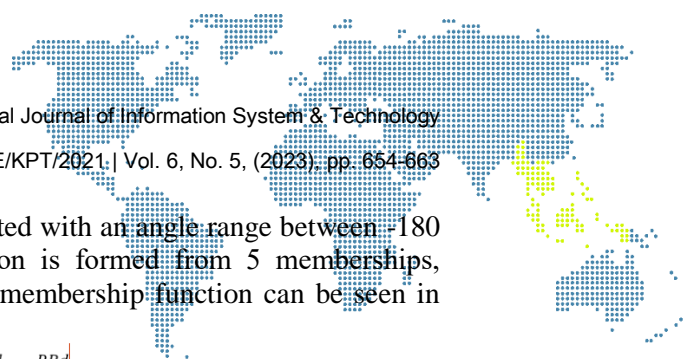


Figure 7. Membership function error



As with input errors, input delta errors are also selected with an angle range between -180° to 180° . The input d_error membership function is formed from 5 memberships, namely, NB, NS, Z, PS, and PB. The d_error membership function can be seen in Figure 8.

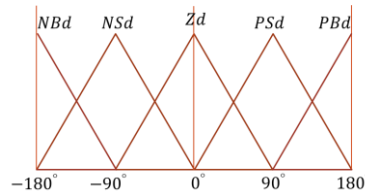


Figure 8. Membership function delta error

The fuzzy rule base uses If-Then rules. This fuzzy base contains implication rules between the two inputs, 25 rule-base rules are built which can be shown in the table below. The fuzzy rules of the if-then method determine the output, which consists of 4 PB conditions, 5 PS conditions, 7 Z conditions, 5 NS conditions, and 4 NB conditions. So that a table of fuzzy rules can be formed as shown in table 1.

Table 1. Database of Fuzzy If-Then Rules

	NB	NS	Z	PS	PB
NB	PB	PB	PS	PS	Z
NS	PB	PS	PS	Z	NS
Z	PS	PS	Z	NS	NS
PS	PS	Z	NS	NS	NB
PB	Z	NS	NS	NB	NB

The membership output function uses a pwm range between -255 up to 255 . Range The membership output function is formed from 5 membership functions namely, CCWB, CCWS, Z, CWS, and CWB. The membership output function can be seen in Figure 9.

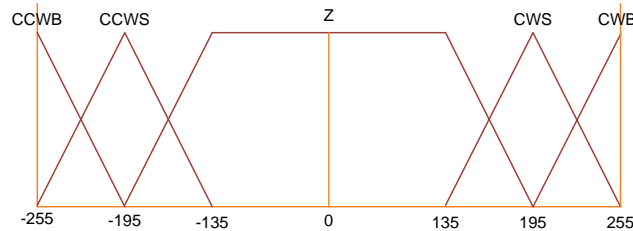


Figure 9. Membership output function

The defuzzification method used in this block is sugeno. The plot surface is shown in Figure 10.

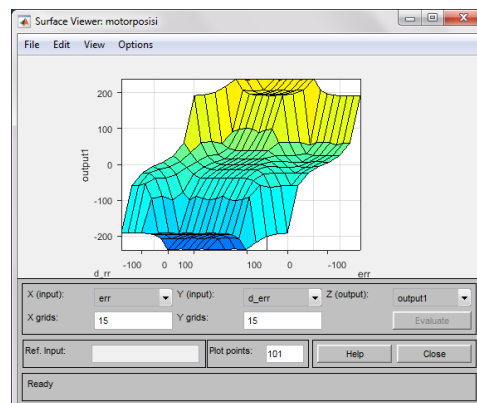


Figure 10. Surface viewer of fuzzy output



3. Result And Discussion

3.1. Fuzzy Logic Controller Simulation

The simulation will test the FLC control design with 5 membership functions for input errors, 5 membership functions for error changes, 5 membership functions for output. The simulation uses an ideal motor plant so that there will be no noise either from the motor or other disturbances that can hinder the movement of the dc motor. the dc motor setpoint is set at 110 degrees. Simulation block diagram of motor DC control position as show in Figure 11 and output response in Figure 12.

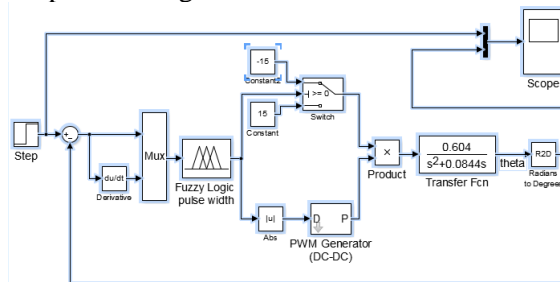


Figure 11. Simulation Block Diagram

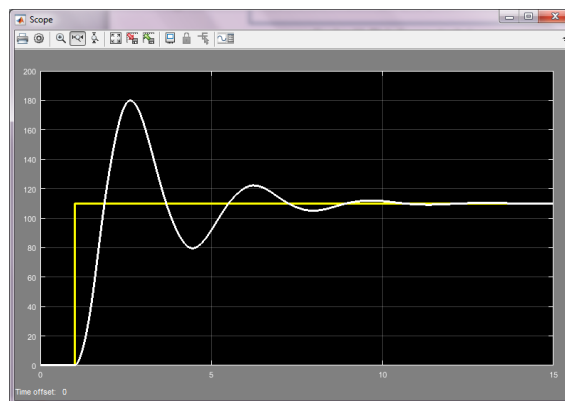


Figure 12. Output Response

In the simulation, the reference input is given in the form of a step input. The input step given is 110°. Within ±10 seconds the system is already steady at the reference condition. Meanwhile, the overshoot can reach 180°. The output graph obtained from the simulation results is shown in Figure 12. From the results of observing the output response to the input step, it can be seen that the FLC is functioning properly in setting the position of the DC motor.

3.2. Hardware Experimental Result

The overall test on the dc motor module with feedback to prove the results of the FLC control design that has been carried out, the test is carried out by providing a "step input" to calculate the transient response parameter value from the system output. after that, the system is given variable inputs to see the reliability and ability of the FLC control to follow changes in input. DC motor module and test block diagram can be seen in Figures 13 and 14. Motor DC module can move both clockwise and counter clockwise movement. The experimental will have area of work between -180 to 180 degrees.

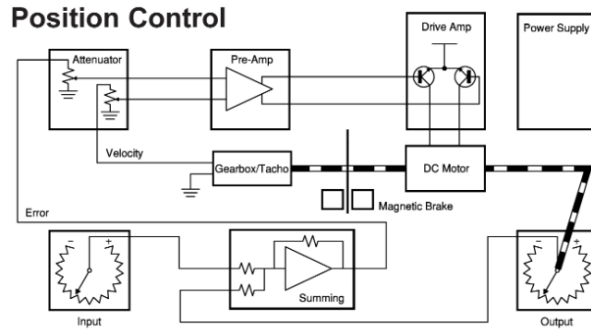


Figure 13. Motor DC Module Diagram

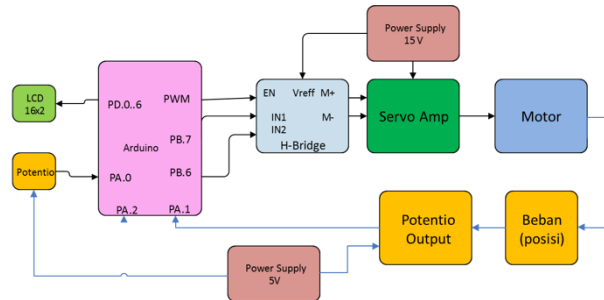


Figure 14. Block Diagram of Control Position

Performance tests were carried out on the controls built by analyzing changes in the output from the output to the setpoint and the results are shown in Figure 15. The performance test is shown in Table 2.

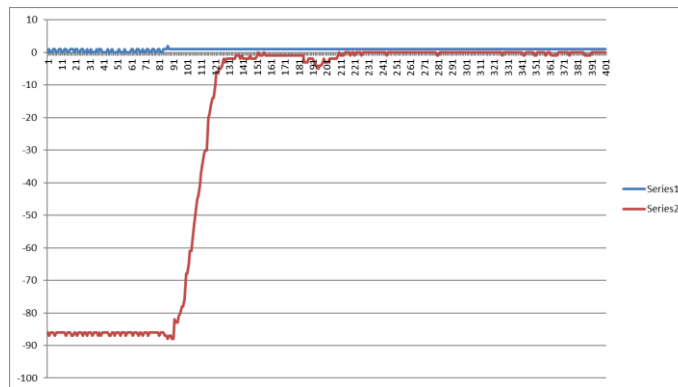


Figure 15. Step input response

Table 2. Output Parameter of Motor DC Control Position

Parameter	Score
Settling Time(s)	0.9 s
Delay Time(s)	0.930769 s
Rise Time(s)	2.7769 s
Peak time(s)	2.7769 s
Maximum overshoot	0

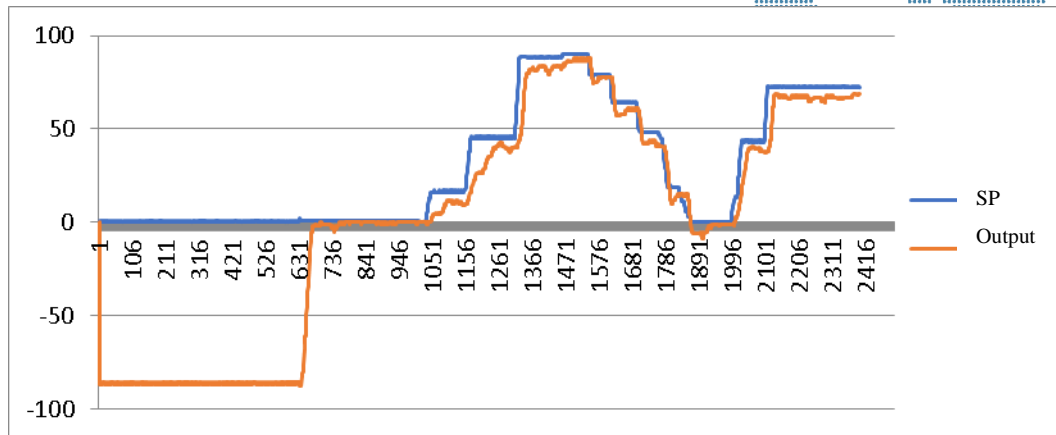
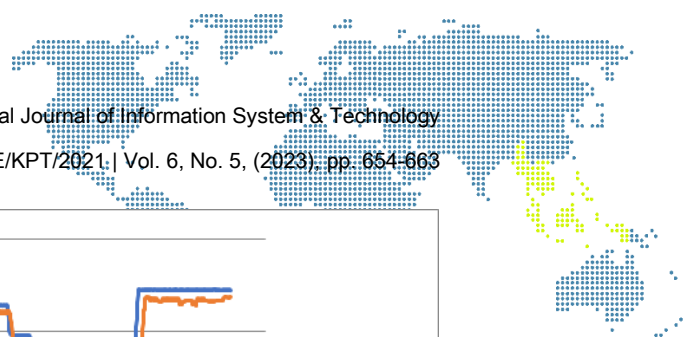


Figure 16. Response of Motor DC Control Position with different setpoint

From Figure 16 it can be seen how the position of the motor tries to reach the setpoint, there isn't an overshoot of the output response, so the design of the FLC is successful.

4. Conclusion

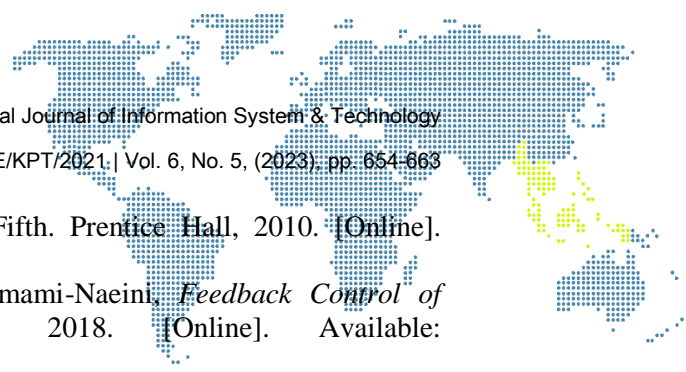
The Motor DC control position has successfully maintained the position given from the setpoint. FLC membership function input can be designed based on analysis of sensor reading data on the potentiometer, FLC membership function output can also be designed based on taking PWM value data to the tachometer output. The Control Position have 0.9 second of settling time, 2.7769 of rise time from position -80 to 0, and 0% of overshoot response output. RTOS can be implemented using a microcontroller based on ATmega 2560. The division of 3 tasks on the RTOS was successfully implemented to run FLC 5 membership function and 5 membership function. FLC can be used to control the position of the DC motor in the feedback module.

Acknowledgements

This research was supported by Laboratorium at Bandung Institute Technology, and Laboratorium Politeknik Negeri Bandung.

References

- [1] C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller—Part I," *IEEE Trans Syst Man Cybern*, vol. 20, no. 2, pp. 404–418, 1990, doi: 10.1109/21.52551.
- [2] C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part II," *IEEE Trans Syst Man Cybern*, vol. 20, no. 2, pp. 419–435, 1990, doi: 10.1109/21.52552.
- [3] J. M. Mendel, "Fuzzy Logic Systems for Engineering: A Tutorial," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 345–377, 1995, doi: 10.1109/5.364485.
- [4] X. J. Ma, Z. Q. Sun, and Y. Y. He, "Analysis and design of fuzzy controller and fuzzy observer," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 1, pp. 41–51, 1998, doi: 10.1109/91.660807.
- [5] S. L. Tan and T. N. B. Anh, "Real-time operating system (RTOS) for small (16-bit) microcontroller," *Digest of Technical Papers - IEEE International Conference on Consumer Electronics*, pp. 1007–1011, 2009, doi: 10.1109/ISCE.2009.5156833.
- [6] S. K. Das, N. Mondol, and N. A. Sultana, "Design And Implement Of A State Feedback Position Output Controller For A Maxon S-Dc Motor With Dspace," in *International Conference on Mechanical Engineering 2011*, 2011, vol. 2011, no. December, pp. 18–20.



- [7] K. Ogata, *Modern Control Engineering*, Fifth. Prentice Hall, 2010. [Online]. Available: www.pearsonhighered.com
- [8] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, Eight. Pearson, 2018. [Online]. Available: www.pearsonhighered.com
- [9] J. Chotai and K. Narwekar, "Modelling and Position Control of a DC Motor," in *IEEE Proceedings: Electric Power Applications*, 2017, pp. 1–8.
- [10] P. Mukherjee and M. Sengupta, "Closed loop speed control of a laboratory fabricated brush-less DC motor drive prototype using position sensor," in *2017 National Power Electronics Conference, NPEC 2017*, 2018, vol. 2018-Janua, no. 1, pp. 166–171. doi: 10.1109/NPEC.2017.8310453.
- [11] H. P. Wang, "Design of fast fuzzy controller and its application on position control of DC motor," in *2011 International Conference on Consumer Electronics, Communications and Networks, CECNet 2011 - Proceedings*, 2011, pp. 4902–4905. doi: 10.1109/CECNET.2011.5768497.
- [12] K. Sharma and D. K. Palwalia, "A modified PID control with adaptive fuzzy controller applied to DC motor," *IEEE International Conference on Information, Communication, Instrumentation and Control, ICICIC 2017*, vol. 2018-January, pp. 1–6, Feb. 2018, doi: 10.1109/ICOMICON.2017.8279151.
- [13] R. Manikandan and R. Arulmozhiyal, "Position control of DC servo drive using fuzzy logic controller," in *2014 International Conference on Advances in Electrical Engineering, ICAEE 2014*, 2014. doi: 10.1109/ICAEE.2014.6838474.
- [14] Aripin, Riyanto, Martin, D. I. H. Putri, and C. Machbub, "Maintaining Trajectory of CoM for Stable Locomotion of Humanoid Robot Using Kalman Filter and Fuzzy Logic Controller," in *2019 IEEE International Conference on Signals and Systems (ICSigSys)*, Jan. 2019, pp. 97–102. doi: 10.1109/ICSIGSYS.2019.8811090.
- [15] N. Farah *et al.*, "A Novel Self-Tuning Fuzzy Logic Controller Based Induction Motor Drive System: An Experimental Approach," *IEEE Access*, vol. 7, pp. 68172–68184, 2019, doi: 10.1109/ACCESS.2019.2916087.
- [16] K. Boudaraia, H. Mahmoudi, M. Abbou, and M. Hilal, "DC motor position control of a solar tracking system using second order sliding mode," in *International Conference on Multimedia Computing and Systems -Proceedings*, 2017, vol. 0, no. 3, pp. 594–598. doi: 10.1109/ICMCS.2016.7905590.
- [17] M. A. Aravind, N. Saikumar, and N. S. Dinesh, "Optimal position control of a DC motor using LQG with EKF," *2017 International Conference on Mechanical, System and Control Engineering, ICMSC 2017*, no. 2, pp. 149–154, 2017, doi: 10.1109/ICMSC.2017.7959461.
- [18] M. Mohamed and A. Mahmoud, "Design of State Feedback Gain Matrix for DC Motor Control Based on Damping Ratio and Natural Frequency," *International Journal of Engineering and Computer Science*, vol. 2, no. 7, pp. 2186–2188, 2013.
- [19] S. A. Kamilu, M. D. A. Hakeem, and L. Olatomiwa, "Design and Comparative Assessment of State Feedback Controllers for Position Control of 8692 DC Servomotor," *International Journal of Intelligent Systems and Applications*, vol. 7, no. 9, pp. 28–33, 2015, doi: 10.5815/ijisa.2015.09.04.
- [20] Martin, "Sistem kendali posisi motor DC menggunakan state feedback controller dan real-time operating system," *JITEL (Jurnal Ilmiah Telekomunikasi, Elektronika, dan Listrik Tenaga)*, vol. 1, no. 1, pp. 69–78, Mar. 2021, doi: 10.35313/JITEL.V1.I1.2021.69-78.
- [21] F. Rammig *et al.*, "Basic concepts of real time operating systems," *Hardware-dependent Software: Principles and Practice*. Springer, pp. 15–45, 2009. doi: 10.1007/978-1-4020-9436-1_2.