DESIGN OF DC MOTOR SPEED CONTROL SYSTEM USING ARDUINO-BASED FUZZY LOGIC CONTROLLER

Ismail; Muslim, M. Aziz; Siswojo, Bambang
Jurusan Teknik Elektro
Universitas Brawijaya Malang, Indonesia
ismailaloha@gmail.com; muh_aziz@ub.ac.id; bsiswojo@ub.ac.id

Abstract— **One of the electrical machinery components that play an important role in industrial processes is drive system, in which DC motor is one of the most widely used component. In DC motor drive system, it is necessary to control the required motion required as well as maintaining the rotation speed as much as possible in accordance with the given reference (set point). In addition, the controller is also a useful component to suppress the error signal so that the desired performance can be obtained. This research designs system of DC motor rotation speed control using Arduino Mega 2560 microcontroller to fulfill control specification on laboratory scale control application and Fuzzy Control System as control system algorithm.**

Fuzzy Control System with Mamdani Inference Engine is used because of its ability to be easily modeled using human intuitive, adaptive, does not require complex mathematical equations, not limited to linear or constant systems, and easily adapted to human input.

Keywords: Fuzzy Control System, Arduino, DC Motor, Speed Control

I. INTRODUCTION

One of the electrical machinery components that play an important role in industrial processes is the drive system, where the DC motor is one of the most widely used types of drivers. In the DC motor drive system it is necessary to control the required motion as well as to maintain the rotation speed as much as possible in accordance with the given reference. In addition, the controller is also a useful component to suppress the error signal so that the desired performance can be obtained. In order to work optimally, the controller must have a fast response and continuously perform the task. Because of the importance of the controlling component, this component has its own system called the control system.

The need for variable rotational speed of DC motors is found in applications such as washing machines, food processor machines, and factory conveyors. In this case, demand of the rotational speed of DC motor varies according to the requirement of the programmed process in the angular speed sequence program and, moreover, the speed transition must occur within a short period of time.

Azadi declared that Azadi Controller has the ability to stabilize DC motor rotation speed by minimizing overshoot and undershoot. Azadi’s research, however, did not show overshoot value generated by system [1].

A research about neuro-fuzzy controller using particle swarm optimization conducted by Farid, et. al. shows that the system can be made as a flexible controller. There is no quantitative result obtained from the research. A qualitative result states that overshoot, undershoot, as well as ringing can be managed and minimized by the method [2].

Another research shows comparison analysis between fuzzy and fuzzified-PID methods on gun-barrel motion control, which was conducted by Muslim, et. al. The control is aimed to direct the gun barrel toward the desired direction, in terms of both the azimuth and elevation angles based on the target position. The results show fuzzified-PID control method excels the fuzzy-logic control method in terms of steady-state error performance and settling-time performance in general [3].

In this research, we will design a DC motor rotation control system using Arduino Mega 2560 microcontroller to fulfill control specification on laboratory scale application (prototype) and Fuzzy Control System as the algorithm. The use of Arduino is due to its easy-to-grasp and in accordance with the needs of prototyping. While the Fuzzy Control System is used because of its ability to be easily modeled using human intuitive, adaptive, does not require complex mathematical equations, not limited to linear or constant systems, and easily adapted to human input.

The DC motors used in this research will be assigned a varying set points (requested rotational speed from low to high and vice versa) and varying loads for later tested for performance by observing the rotation speed response, i.e. the speed error.

Based on the above description of the background, it can then formulate a problem as follows: How to design a varied and loaded DC motor rotation speed control system using Arduino-based Fuzzy Control System.

The purpose of this research is to design a control system of varied and loaded DC motor rotation using Arduino-based Fuzzy Control System in order to get accurate rotation speed, minimal overshoot or undershoot, and minimum ringing according to requirement.

II. THEORETICAL BASIS

A. Control System

A control system manages, commands, directs or regulates the behavior of other devices or systems. It can range from a home heating controller using a thermostat controlling a domestic boiler to large Industrial control systems which are used for controlling processes or machines.

In the most common form, the feedback control system it is desired to control a process, called the plant, so its output follows a control signal, which may be a fixed or changing value. The control system compares the output of the plant to the control signal, and applies the difference as an error signal to bring the output of the plant closer to the control signal [4].

B. Overshoot and Undershoot

Overshoot is a condition in which the signal or function value goes beyond the steady-state (set-point – desired value). The opposite, undershoot, occurs when the value of a signal or function is smaller than its reference. Both overshoot and undershoot occur mostly followed by a swing (ringing or oscillation) caused by an excessive counter-response performed by a control system.

In control theory, overshoot refers to an output exceeding its final, steady-state value [5]. For a step input, the percentage overshoot (PO) is the maximum value minus the step value divided by the step value. In the case of the unit step, the overshoot is just the maximum value of the step response minus one. Also see the definition of overshoot in an electronics context.

C. Ringing

Ringing is a signal swing which occurs particularly after a sudden set-point change. In some cases, ringing is an undesirable phenomenon, though not always. The ringing phenomenon is closely related to the overshoot, which in this case the ringing occurs following the overshoot

In electrical circuits, ringing is an unwanted oscillation of a voltage or current. It happens when an electrical pulse causes the parasitic capacitances and inductances in the circuit (i.e. those that are not part of the design, but just by-products of the materials used to construct the circuit) to resonate at their characteristic frequency [6].

D. Steady State

A system or process is said to be in a steady state or stable if the variable that determines the behavior of the system or process is unchanged over time.

E. Steady State Error (Error Band)

Steady-state error is defined as the difference between the input (set-point) and the output of a system at a certain extent when the time-scale leads to infinity (i.e. when the response has reached steady-state conditions). Steady state error will depend on input type and also on system type. Steady-state error is also called as error band.

F. Rise Time

In electronics, when describing a voltage or current step function, rise time is the time taken by a signal to change from a specified low value to a specified high value [7]. These values may be expressed as ratios [8] or, equivalently, as percentages [9] with respect to a given reference value. In analog electronics or digital electronics, these percentages are commonly the 10% and 90% (or equivalently 0.1 and 0.9) of the output step height: [10] however, other values are commonly used [11]. For applications in control theory, according to Levine (1996), rise time is defined as "the time required for the response to rise from x% to y% of its final value", with 0% to 100% rise time common for underdamped second order systems, 5% to 95% for critically damped and 10% to 90% for over damped ones [12].

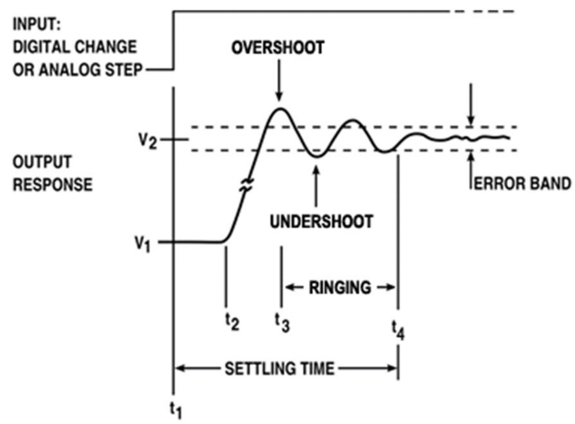

Figure 1. An illustration of overshoot phenomena followed by ringing and settling time.

According to Orwiler (1969), the term "rise time" applies to either positive or negative step response, even if a displayed negative excursion is popularly termed fall time [13].

G. Settling Time

Settling time of an output device is the time taken from the reference input until the existing signal enters the steady-state stage (or already in a tolerable steady state error state). Settling time depends on system response capability and time constant. Settling time on a controlled system must be kept as small as possible by the control system.

H. Fuzzy Logic

Fuzzy logic is a form of many-valued logic in which the truth values of variables may be any real number between 0 and 1. It is employed to handle the concept of partial truth, where the truth value may range between completely true and completely false [14]. By contrast, in Boolean logic, the truth values of variables may only be the integer values 0 or 1. Furthermore, when linguistic variables are used, these degrees may be managed by specific (membership) functions [15].

The term fuzzy logic was introduced with the 1965 proposal of fuzzy set theory by Lotfi Zadeh [16][17]. Fuzzy logic had however been studied since the 1920s, as infinite-valued logic—notably by Łukasiewicz and Tarski [18].

Fuzzy logic has been applied to many fields, from control theory to artificial intelligence.

I. Arduino

Arduino is an open source, computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL) [19], permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

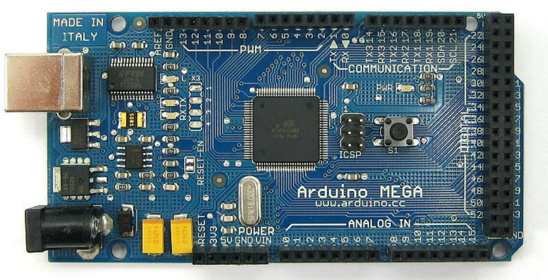

Figure 2. Arduino Mega 2560 used in this research.

The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy [20], aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

The name Arduino comes from a bar in Ivrea, Italy, where some of the founders of the project used to meet. The bar was named after Arduin of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014 [21]. This research uses Arduino Mega 2560.

J. DC Motor

DC motor is an electronic machine that converts DC electric current into mechanical power. A typical type of DC motor uses a magnetic field to move it. Almost all types of DC motors have internal structures, electromechanical or electronic mechanisms, which periodically change the direction of current on the part inside the DC motor. Most types of DC motors produce rotary motion, the other type is a linear DC motor that produces force and movement in a straight line (usually called an actuator).

K. Driver Motor DC

The DC motor driver is a tool that controls the DC motor torque. A DC motor driver is a small current amplifier (which becomes a control current) into a larger current to supply power to a DC motor. DC motor drivers are usually self-made circuits adapted to the control, voltage, and current inputs required by controlled DC motors. DC motor drivers do the controlling by variating input voltage to DC motor as its output torque needed.

L. Speed Sensor

Speed sensor is a device to detect object's speed. The object's speed can be linear, rotational, or any other types. This research uses rotational speed sensor to gather information about controlled DC motor speed. The speed sensor type is tachometer, which detects rotational speed of a disk in RPM (revolution per minute).

III. Research Methods

The diagram in Figure 3 describes several stages done in this research. First of all, designs the hardware required, followed by the development of hardware as the specifications. The control software is then developed, which in this case using fuzzy control system.

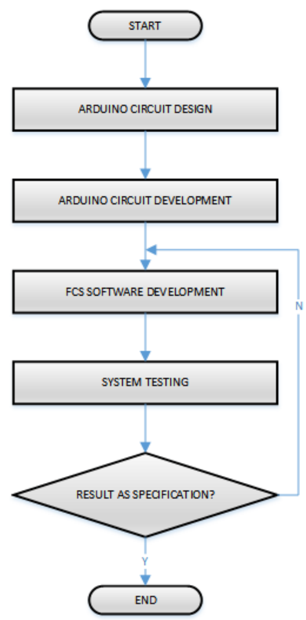

Figure 3. Diagram of research methods.

The next step is testing the system to know the performance of the system according to specifications. After testing, it is known whether the performance is in optimal state or not. When it does not meet optimal standard, it will retest the system. But when it is already in optimal state, then the research process is considered completed.

A. Specification of Controller

Some specifications expected from this control system are as follows:

- Suppress overshoot/undershoot for maximum 20% of set-point difference.
- Suppress overshoot/undershoot for maximum 30% of set-point from load application or releasing.
- Suppress rising time for maximum 5 times of correction period.
- Determine the error band for no more than 5% of maximum rotational speed.

B. System Diagram

In accordance with the existing system diagram as shown in Figure 4, it can be shown that there are three parts to this built-in system. The first part is the input, which consists of angular speed sequence program, program start trigger, and emergency stop trigger. In addition there is feedback from the speed sensor as part of the input. The second part is a process, consisting of a fuzzy logic based program within the Arduino Mega 2560 microcontroller. The third part is the output to the driver of the DC motor which will generate voltage to drive the DC motor. Speed sensors are added to provide feedback to the process, and are considered as input devices.

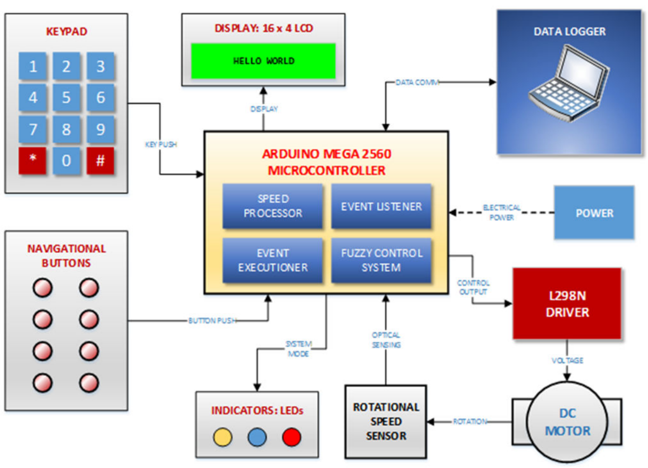

Figure 4. Schematic system diagram.

C. Angular Speed Sequence Program

This part is a user interface which allow user to program set-points in respect of time (in second). Programming for this part using commands as follow:

- START, is a command to reset time and begin to run the program.
- STOP, is a command to mark the end of sequence.
- T:hh:mm:ss.zzz V:x.yyy, is a command that shows time coordinate (in second) relative from START and command to alter the set-point in that time coordinate in RPM.

The angular speed sequence program must be defined at least for one command-line before the system can perform the test run.

D. Program Start Trigger

It is a push-button used to start the program. It's labeled as "START" and once it is pressed, the program starts to run.

E. Emergency Stop Trigger

This part is a push-button to stop the program running in case of emergency. It's labeled as "STOP" and once it is pressed, the program will stop immediately.

F. DC Motor Driver

This section is a series of current amplifiers from the Arduino Mega 2560 output to a larger current and voltage adjusted to DC motor specifications to power the DC motor.

G. DC Motor

This section is a DC motor that will be controlled. These DC motors can be assigned to different set-points and loads. This DC motor has a voltage input from the DC Motor Driver circuit.

H. Speed Sensor

This section is a sensor to detect the speed of DC motor rotation. This sensor is a tachometer, and the result of its speed measurement has an RPM unit. The results of these measurements are given as feedback on the system, and also one result is given to the logger to get the DC motor speed readings to be compared with the program on the Angular Speed Sequence Program.

I. Fuzzy Control System in Arduino Microcontroller

This section is the main part that contains the program with fuzzy logic algorithm to control the loaded DC motor and simultaneously get input from the angular speed sequence program, start trigger program, emergency stop trigger, and feedback from the speed sensor.

IV. Results and Discussions

A. Modeling of Fuzzy Logic Control System

The steps taken to get the modeling of FCS (fuzzy control system) in this case are:

- Testing to get the processing speed of the programming performance in Arduino.
- Determine the lowest speed (other than zero) and the highest speed possible to get process service according to processing speed performance.
- Obtain a control mapping function (an integer value according to the Arduino specification, complete with separation between values) to the output voltage leading to the DC motor (after the driver).
- Determine the unlimited maximum speed specifications and ideal load speeds expected from DC motors.
- Designing a fuzzy controller.
- Designing defuzzification process.

B. Arduino Processing Speed

Each Arduino hardware has a theoretical process speed specification. The specifications must be different from the real conditions due to several factors, among which are the calculation loads in each process cycle varying. Based on the existing datasheet, the Arduino Mega 2560 has a maximum process resolution of 4µs.

From the experimental results it was found that the maximum real process resolution is 5.72 µs. This result is 43% slower than the Arduino Mega 2560 datasheet. Under loaded process conditions using various calculations and not taking into slow and locking processes account (such as displaying data on display), it is expected that a resolution of 100 times slower can be achieved, i.e. about 600 µs. A slow and locking process will only be executed every 50000 µs once so it does not really affect the overall system work.

C. Minimum and Maximum Rotation Speed

The rotation speed is obtained from the LM 393 light sensor device and combined with 20-hole disks in a loop. The sensor works by calculating the interrupted time between the holes that emit the light from one side and received on the other side with a non-hole that block the light.

In accordance with 600 µs loaded process resolution, the maximum rotational speed (with period of one hole with one non-hole) as Equation (1) is obtained.

$$\omega = \frac{1000000}{2 \cdot r \cdot C_H} = \frac{1000000}{2 \cdot 600 \cdot 20} = 41.67Hz = 2500RPM \quad (1)$$

Where ω is the frequency of rotation, r is the resolution of the process, and C_H is the number of holes in a single disc. In other words, at a speed higher than 2500 RPM, then when the sensor sends the switch information between the holes with the non-holes there will be several undetected information as it goes beyond the process resolution, so that a speed reading will be inaccurate.

While performing other relatively slow and locking processes performed every 50000 µs, the speed reading results must be obtained to perform the action according to the data received. This can be used to find the minimum speed that can be calculated as well as in Equation (2).

$$\omega = \frac{1000000}{2 \cdot r \cdot C_H} = \frac{1000000}{2 \cdot 50000 \cdot 20} = 0.5Hz = 30RPM \quad (2)$$

In other words, Equation (2) shows that the minimum rotational speed the process can calculate is 30 RPM. So the range of rotational speeds that can be read by the process is 30 RPM to 2500 RPM.

D. Mapping Function

The mapping function is required to map the range of controllers to the output voltage generated by the driver to the DC motor. Each output control pin owned by Arduino Mega 2560 has two types of data:

- Digital pins with possible either LOW or HIGH values.
- Pin analog PWM (pulse width modulation) with a range of values from 0 through 255 in the form of integers. So there are 256 possible values used. The PWM voltage coming out of the pin (without using the driver) is 0V through 5V.

In this case an analog pin 2 is used to perform the control. As for the output voltage of the DC motor driver is 0V through 20V. The output voltage can have positive and negative polarization. The polarization controller is in combination of digital pins 3 and 4.

- If pin 3 is LOW and pin 4 is HIGH, then the polarization of the output voltage is positive, where the DC motor will rotate forward.
- If pin 3 is HIGH and pin 4 is LOW, then the polarization of the output voltage is negative, where the DC motor will rotate in reverse.
- If pin 3 and 4 are LOW respectively, or respectively HIGH, then the output voltage is 0V, although the analog pin 2 is not 0, where the DC motor does not rotate.

E. Maximum and Minimum Speed of DC Motor

According to the DC motor specifications used in this research, the maximum unloaded speed is 3500 RPM. While the maximum speed of the expected DC motor under ideal load condition is 2500 RPM.

In accordance with the experimental results, the minimum speed at which the DC motor starts to move without load is about 500 RPM.

By knowing the minimum speed of a DC motor (unloaded and start spinning) and the maximum speed of a DC motor (with ideal load), it can be deduced that the expected rotational speed range is 500 RPM to 2500 RPM, with forward and reverse rotation.

F. Designing The Fuzzy Controller

The final stage is to design the fuzzy set, fuzzy rules, and defuzzification to get output in the form of DC motor voltage in accordance with the input of demand speed of rotation (set-point) and feedback from the rotational speed sensor. In this case, in the fuzzy set design, there are two types of inputs to determine fuzzy rules, i.e. error (from DC motor rotation speed) and the derivative of the error.

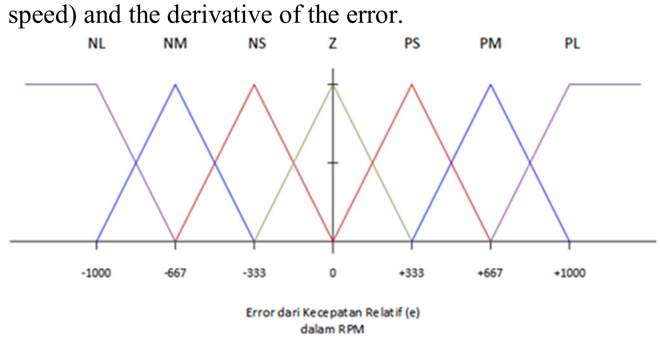

Figure 5. Fuzzy membership function based on error of set-point.

The first part is the fuzzy membership degree function based on the input error of the desired rotational speed, presented in figure 5.

In the membership degree function, there are seven memberships, of which are

- NL, which is negative large, or a large negative value.

- NM, i.e. a negative medium, or an intermediate negative value.
- NS, which is negative small, or a small negative value.
- ZE, i.e. zero, or a zero value.
- PS, which is positive small, or a small positive value.
- PM, the positive medium, or a medium positive value.
- PL, which is positive large, or a large positive value.

The seven types of membership are also used in the fuzzy membership degree function based on the derivative of the error of the rotational speed

$$NL(e) = \begin{cases} 1; & e < -1000 \\ 1 - \frac{e+1000}{-667+1000}; & -1000 \le e \le -667 \\ 0; & e > -667 \end{cases} \quad (3)$$

$$NM(e) = \begin{cases} \frac{e+1000}{-667+1000}; & -1000 < e < -667 \\ 1 - \frac{e+667}{-333+667}; & -667 \le e < -333 \\ 0; & -1000 \ge e \ge -333 \end{cases} \quad (4)$$

$$NS(e) = \begin{cases} \frac{e+667}{-333+667}; & -667 < e < -333 \\ 1 - \frac{e+333}{333}; & -333 \le e < 0 \\ 0; & -667 \ge e \ge 0 \end{cases} \quad (5)$$

$$ZE(e) = \begin{cases} \frac{e+333}{333}; & -333 < e < 0 \\ 1 - \frac{e}{333}; & 0 \le e < 333 \\ 0; & -333 \ge e \ge 333 \end{cases} \quad (6)$$

$$PS(e) = \begin{cases} \frac{e}{333}; & 0 < e < 333 \\ 1 - \frac{e-333}{667-333}; & 333 \le e < 667 \\ 0; & 0 \ge e \ge 667 \end{cases} \quad (7)$$

$$PM(e) = \begin{cases} \frac{e-333}{667-333}; & 333 < e < 667 \\ 1 - \frac{e-667}{1000-667}; & 667 \le e < 1000 \\ 0; & 333 \ge e \ge 1000 \end{cases} \quad (8)$$

$$PL(e) = \begin{cases} 1; & e > 1000 \\ \frac{e-667}{1000-667}; & 667 \le e \le 1000 \\ 0; & e < 667 \end{cases} \quad (9)$$

Equation (3) to Equation (9) presented the functions of fuzzy membership degree based on the input error of the desired rotational speed.

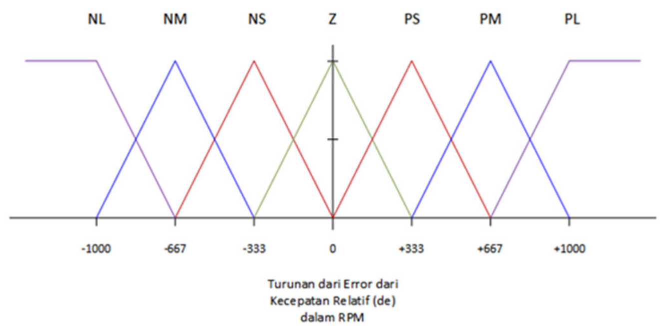

Figure 6. Fuzzy membership function based on derivative of error of set-point.

Figure 6 shows the fuzzy membership degree function based on the derivative of the desired speed error. The maximum unit for error and error derivative of the desired speed is maximum ± 1000 RPM, i.e. half of the maximum DC motor rotational speed range (2500 RPM), hence maximum transition requires at least two processes to smooth out the output and gives more chance of feedback. This method thus suppresses the level of overshoot or undershoot and ringing.

Table I. FUZZY RULES TABULATION.

e \ de	NL	NM	NS	ZE	PS	PM	PL
NL	APL	APL	APL	APL	APL	APM	APS
NM	APL	APM	APM	APM	APM	APS	APS
NS	APM	APS	APS	APS	APS	AZE	AZE
ZE	APS	APS	AZE	AZE	AZE	ANS	ANS
PS	AZE	AZE	ANS	ANS	ANS	ANS	ANM
PM	ANS	ANS	ANM	ANM	ANM	ANM	ANL
PL	ANS	ANM	ANL	ANL	ANL	ANL	ANL

Defuzzification process will be done by giving a firm value (crisp value) to be given as output. The output value is classified according to the seven categories as below:

- APL, which is Add Positive Large, or add a large positive number.
- APM, which is Add Positive Medium, or adds a moderate positive number.
- APS, which is Add Positive Small, or add a small positive number.
- AZE, Add Zero, or add nothing.
- ANS, which is Add Negative Small, or add a small negative number.
- ANM, which is Add Negative Medium, or add an intermediate negative number.
- ANL, which is Add Negative Large, or add a large negative number.

In the defuzzification process there is mapping from the output of Table I to the output value pin 2 analog Arduino Mega 2560. Therefore determined the category of firm values (crisp values) on defuzzification as follows and shown graphically in the Figure 7:

- APL, using the value of +080.
- APM, using the value of +020.
- APS, using the value of +005.

- AZE, using the value of 000.
- ANS, using the value of -005.
- ANM, using the value of -020.
- ANL, using the value of -080.

Based on the input of error and delta-error, a rule table is set to determine the output of the relative control value by weighing all possible membership of error and delta-error. In accordance with the nature of Mamdani inference, this table is an intuitive human form as a response to the error and delta-error. The table representation of fuzzy rules is presented in Table I.

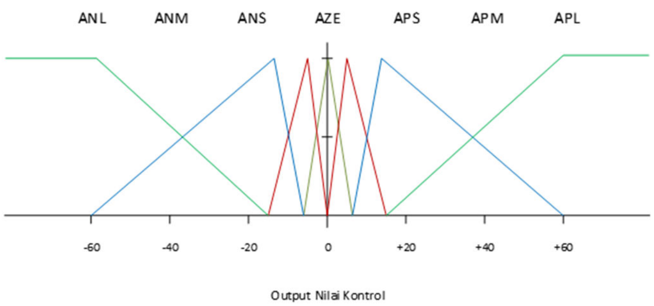

Figure 7. Fuzzy memberships of control actions.

The way how defuzzification is used in this method is center of gravity. This method performs each weights with the firm defuzzification value (as in Table I), then summed, and divided by the sum of all the weights.

$$u = \frac{\sum_{x=1}^{7} WR_x}{\sum_{y=1}^{49} W_y} \quad (10)$$

Equation 10 denotes the deffuzification that generates the value of the control value that is sent to correct the value at the DC motor driver output voltage, where W is the weight obtained from the minimum value between the error membership with delta-error, and R is the fuzzy set value of the control action.

G. *Fuzzy Control System Parameters for Testing*

The required sequence program is one that can test the parameters for some scenarios required on the DC motor later. Some scenarios include:

- Start the motor from the stop to the low, medium, and high rotation.
- Increase motor rotation speed from rotary rotation to medium and high rotation.
- Increase the motor rotation speed from the round to the high rotation.
- Reduce the motor rotation speed from high rotation to mid or low rotation.
- Reduce the motor rotation speed from mid-to-mid-rotation to low rotation.
- Apply the load at high, medium, or low rotation.
- Release the load at high, medium, or low rotation.

As well as the control specification requirement, some expectations of this control system are as follows:

- Suppress overshoot/undershoot for maximum 20% of set-point difference.
- Suppress overshoot/undershoot for maximum 30% of set-point from load application or releasing.
- Suppress rising time for maximum 5 times of correction period. The correction period is 500 ms.
- Determine the error band for no more than 5% of maximum rotational speed. The error band is 125 RPM.

Adjustment parameters to be made include:

- The membership degree diagram of the error (Figure 4), since it is adjusted to the speed range of DC motor rotation.
- The degree of membership diagram of delta-error (Figure 6), because it is adjusted to the range of changes in rotation speed of DC motor.
- Tabulation of fuzzy rules (Table I), since they are based on intuition that involves adding or subtracting control values based on errors and delta-errors.

Table II. PROGRAM SEQUENCE FOR FCS TESTING.

S / T	0	5	10	15	20	25	30	35	40	45
S-01	500	1500	2500	1500	500	500	500	500	500	STOP
S-02	1500	2500	500	1000	2000	2500	1500	1500	1500	STOP
S-03	2500	500	2500	500	2500	2500	2500	2500	2500	STOP

While the parameters to be set according to the speed chart observation are:

- The membership diagram of the output value control or fuzzy set of control actions (Figure 7).

H. *Program Sequence for Testing Scenario*

Table II presents the sequence program for 3 types of testing. The first test (S-01) is a start test to the 500 RPM, then converts round 1500 RPM and 2500 RPM, then down again and tests the load at 500 RPM.

The second test (S-02) is the start to 1500 RPM then increased to 2500 RPM, then decreased to 500 RPM and step up gradually and test the load at 1500 RPM. The third test (S-03) is a start to 2500 RPM then decreased drastically to 500 RPM, rises again to 2500 RPM, goes down to 500 RPM, and finally tests loads at 2500 RPM. The red shaded part is the load testing section.

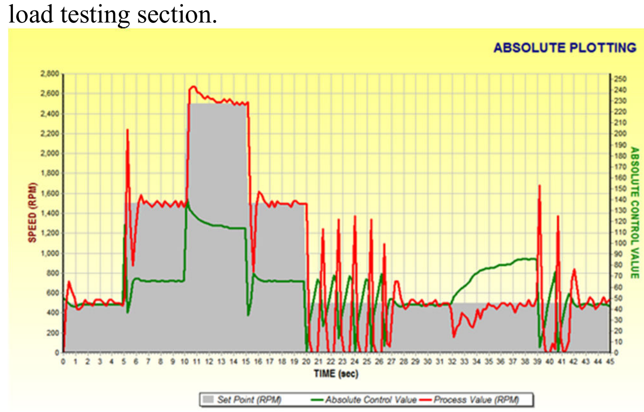

Figure 8. First testing result.

The first test uses a sequence of fuzzy sets of control actions ANL -080, ANM -020, ANS-005, AZE 000, APS +005, APM +020, and APL +080, and obtained the results as in Figure 8.

While the second test using the fuzzy set sequence of control actions is ANL -015, ANM-010, ANS-005, AZE 000, APS +005, APM +010, and APL +015, and the results are shown in Figure 9.

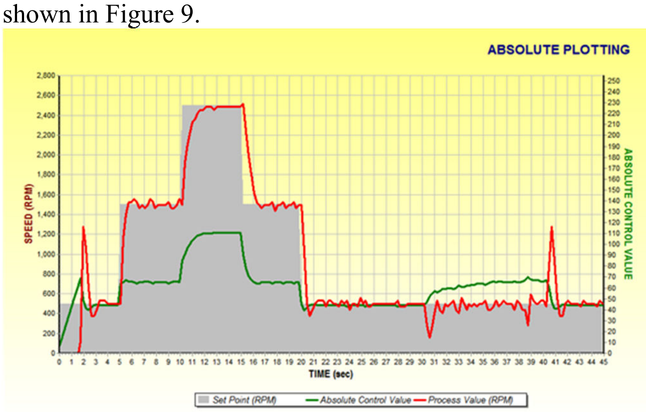

Figure 9. Second testing result.

I. Testing Result

From the test results obtained as follows:

- To obtain a minimum rise time, the order of the fuzzy set of control actions is ANL -080, ANM -020, ANS -005, AZE 000, APS +005, APM +020, and APL +080. But at low and medium rounds there will be overshoot and ringing beyond the specification limits.
- To obtain smooth results with overshoot and ringing included in the specification, the order of the fuzzy set of control actions is ANL -015, ANM-010, ANS -005, AZE 000, APS +005, APM +010, And APL +015. In this case rise time increases up to 2.5 seconds.

V. CONCLUSIONS

The conclusions that can be drawn from this research are:

- The defuzzification parameter [-080, -020, -005, 000, +005, +020, +080] causes out-of-threshold overshoot and ringing but the rise time meets the specification.
- The defuzzification parameter [-020, -010, -005, 000, +005, +010, +020] minimizes overshoot and ringing into permissible value but a bad rise time.
- Overshoot and ringing are the criteria which analog with system smooth respond, while the rise time specifies system responsiveness.
- As high as 89% of system accuracy to the given specification has been acquired from the testing.

While this research has demonstrated the accuracy and ease of Mamdani Inference Engine, several opportunities for extending its scope remains, including:

- The load application on DC motor can be modified quantitatively with a specific load generator.
- The application of scheduling and programmable load as of set-points in this research.

VI. REFERENCES

[1] Azadi, Sassan, & Mosa, Nouri. 2012. Utilizing Azadi Controller to Stabilize the Speed of a DC Motor. Proceedings of the 2012 International Conference on Advanced Mechatronic Systems, Tokyo, Japan. p. 269-274.

[2] Farid, Ali Moltajaei, & Barakati, S. Masoud. 2014. DC Motor Neuro-Fuzzy Controller Using PSO Identification. The 22nd Iranian Conference on Electrical Engineering. p. 1162-1167.

[3] Muslim, M. Aziz, Minggu, Desyderius, Saputra, Jeki, Hasanah, Rini Nur. 2015. Comparison Analysis Between Fuzzy and Fuzzified-PID Methods on Gun-Barrel Motion Control. ARPN Journal of Engineering and Applied Sciences. Asian Research Publishing Network.

[4] https://en.wikipedia.org/wiki/Control_system Retrieved 2016-11-03.

[5] Kuo, Benjamin C & Golnaraghi M F. 2003. Automatic Control Systems (Eighth ed.). NY: Wiley. p. §7.3 p. 236–237.

[6] Johnson, H. and Graham, 1993. M. High-Speed Digital Design: A Handbook of Black Magic. pp. 88–90.

[7] Cherry, E. M.; Hooper, D. E. 1968. Amplifying Devices and Low-pass Amplifier Design, New York–London–Sidney: John Wiley & Sons, pp. xxxii+1036.

[8] Elmore, William C. 1948. The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers, Journal of Applied Physics, pp. 55–63.

[9] Levine, William S. 1996. The Control Handbook, Boca Raton, FL: CRC Press, pp. xvi+1548.

[10] Levine, William S. 2011., The Control Handbook: Control Systems Fundamentals (2nd ed.), Boca Raton, FL: CRC Press, pp. xx+766.

[11] Millman, Jacob; Taub, Herbert. 1965. Pulse, Digital and Switching Waveforms. New York–St. Louis–San Francisco–Toronto–London–Sydney: McGraw-Hill, pp. xiv+958.

[12] National Communication Systems, Technology and Standards Division. 1997. Federal Standard 1037C. Telecommunications: Glossary of Telecommunications Terms, FSC TELE, FED–STD–1037, Washington: General Service Administration Information Technology Service, p. 488.

[13] Nise, Norman S. 2011. Control Systems Engineering (6th ed.), New York: John Wiley & Sons, pp. xviii+928.

[14] Novák, V., Perfilieva, I. and Močkoř, J. 1999. Mathematical principles of fuzzy logic Dodrecht: Kluwer Academic.

[15] Ahlawat, Nishant, Ashu Gautam, and Nidhi Sharma. 2014. Use of Logic Gates to Make Edge Avoider Robot. International Journal of Information & Computation Technology. pp. 630

[16] Fuzzy Logic. Stanford Encyclopedia of Philosophy. Bryant University. 2006-07-23. Retrieved 2008-09-30.

[17] Zadeh, L.A. 1965. Fuzzy Sets. Information and Control. pp. 338–353.

[18] Pelletier, Francis Jeffry. 2000. Review of Metamathematics of Fuzzy Logics. The Bulletin of Symbolic Logic. pp. 342–346.

[19] Arduino - Introduction. http://www.arduino.cc Retrieved 2016-08-12.

[20] David Kushner. 2011. The Making of Arduino. IEEE Spectrum.

[21] Justin Lahart. 2009. Taking an Open-Source Approach to Hardware. The Wall Street Journal. Retrieved 2014-09-07.