JAREE-Journal on Advanced Research in Electrical Engineering
Volume 1, Number 2, October 2017

Implementation of Mamdani Fuzzy Logic Control System on DC Motor Speed Controller

Basilio Mendonça Freitas, Mochammad Rameli, Rusdhianto EAK
Department of Electrical Engineering
Sepuluh Nopember Institute of Technology
Surabaya, Indonesia
jnascerfreitas@gmail.com, rameli@ee.its.ac.id, ditto@ee.its.ac.id

***Abstract*— Drive system is an important role in industrial processes, especially in electrical control. Maintaining a DC motor speed is a control system task that requires several method. Generally, set-point is defined as point of demand, and its comparation against process value (current speed) resulted in error. Both error and delta-error are two parameters required to the control system to determine system behavior in correction action. Such system of controller is 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 Uno microcontroller to meet control specification on laboratory scale, implementing control application and Fuzzy Control System as control system algorithm.**

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, Mamdani Fuzzy Logic Control System is used.

***Keywords*— Mamdani Inference Engine, Fuzzy Logic, Arduino Uno, Rotation Speed Control, DC Motor**

I. Introduction

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. 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. The controller is also a useful component to suppress the error signal so that the desired performance can be obtained. In order to work in ideal manner, the controller must have a combination of rapid but smooth response while continuously perform the task. This component has its own system called the control system, because of the importance of the controlling component.

DC motors are widely use with a variety of applications and have many approaches controlling the speed that has been proposed. Such as feeding machines, hose machines and industrial conveyor. However due to complex calculation many control approaches are not applicable in practical experiment [2].

A 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, i.e. the azimuth and elevation angles, based on the target position. The results show fuzzified-PID control method outshines the fuzzy-logic control method in terms of steady-state error performance and settling-time performance in general [3].

In this research, a DC motor rotation control system will be designed using Arduino Uno microcontroller to mimic prototype control specification and Mamdani Fuzzy Logic Control System as the algorithm. The use of Arduino is due to its easy-to-develop and in accordance with the needs of prototyping. While the Mamdani Fuzzy Logic 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. In contrast to AC motors behavior, DC motors torque, hence the rotational speed, can be easily adjusted by varying their input voltage. Higher input voltage generates higher torque, which in turn deliver more rapid rotation output under constant load.

Based on the above description, a problem can be formulated as: How to design a varied DC motor rotation speed control system using Arduino-based Mamdani Fuzzy Logic Control System.

The purpose of this research is to design a control system of varied DC motor rotation using Arduino-based Mamdani Fuzzy Logic 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. *Closed-loop Control*

There are two common classes of control systems, open loop control systems, and closed loop control systems.

In an open loop control system, the control action from the controller is independent of the "process output". A good example of this is a central heating boiler controlled only by a timer, so that heat is applied for a constant time, regardless of the temperature of the building. (The control action is the switching on/off of the boiler. The process output is the building temperature).

In a closed loop control system, the control action from the controller is dependent on the desired and actual process output values. In the case of the boiler analogy this would utilise a thermostat to monitor the building temperature, and thereby feedback a signal to ensure the controller output maintains the building temperature to that set on the thermostat.

A closed loop controller therefore has a feedback loop which ensures the controller exerts a control action to give a process output the same as the "Reference input" or "set point". For this reason, closed loop controllers are also called feedback controllers.

C. *Control System Phenomena*

In signal processing, control theory, electronics, and mathematics, overshoot is the occurrence of a signal or function exceeding its target. It arises especially in the step response of bandlimited systems such as low-pass filters. It is often followed by ringing, and at times conflated with the latter. 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.

In electronics, signal processing, and video, ringing is oscillation of a signal, particularly in the step response (the response to a sudden change in input). Often ringing is undesirable, but not always, as in the case of resonant inductive coupling. It is also known as hunting. [1] It is closely related to overshoot, generally occurring following overshoot, and thus the terms are at times conflated. It is also known as ripple, particularly in electricity or in frequency domain response.

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.

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.

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].

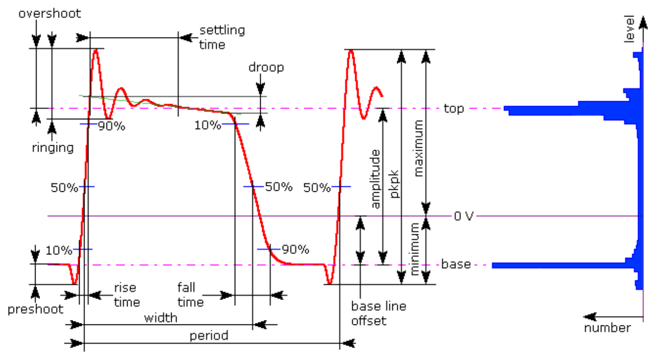

Fig. 1. Various control system phenomena.

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].

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.

D. *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.

E. *Arduino Microcontroller*

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.

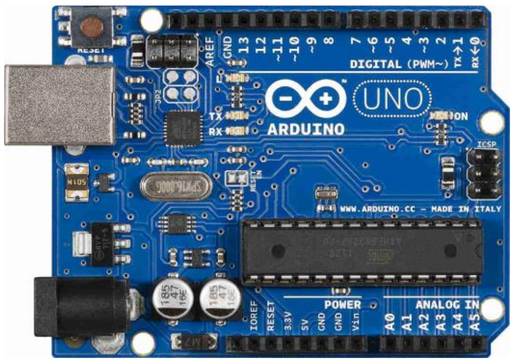

Fig. 2. Arduino Uno used in this research.

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 Arduino of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014 [21]. This research uses Arduino Uno.

F. *DC Motor*

An electronic machine that converts DC electric current into mechanical power is DC motor. A typical type of DC motor uses a magnetic field to rotate 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).

If its torque (and rotational speed) controlled by Arduino, DC motor needs a driver to convert small voltage and current from Arduino (mostly operates on 5 volts) into a specific voltage and current requirement. 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.

G. *Tachometer*

Tachometer is a device to detect object rotational speed. This research uses tachometer to gather information about rotational speed of controlled DC motor. The tachometer consists of an infrared transmitter, a receiver, and an optical chopper. The rotational speed is determined by rapidity of optical chopper variance between hole and barrier.

III. Research Methods

As shown in Figure 3, this research is done by several stages. First step is to design the hardware required, followed by the development of a specific hardware. The control software is then developed using Mamdani Fuzzy Logic Control (FLC) System.

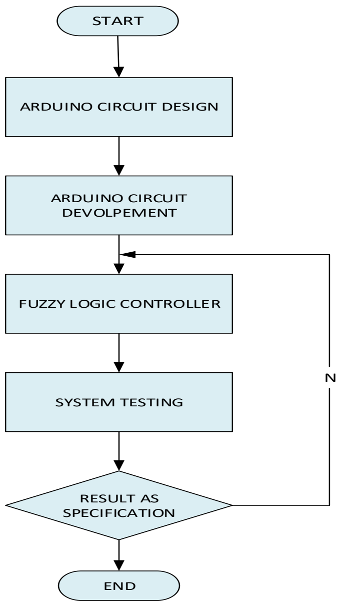

Fig. 3. Diagram of research methods.

The next step is testing the system to obtain system performance 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. *Controller Specification*

Some specifications expected from this control system are as follows:

- Maximum overshoot is 30% of set-point difference.
- Maximum overshoot is 50% of set-point from load changing.
- Maximum 15 times of correction period for rising and falling time.
- Error band is defined for 8% of speed range.

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 sequencer, keypad, navigation buttons, and display. Feedback from the tachometer is also a part of the input. The second part is a process, consisting of a Mamdani Fuzzy Logic Controller based software inside the Arduino Uno microcontroller. The third part is the output to the driver of the DC motor which will generate voltage variance to control the DC motor torque. Tachometer are added to provide feedback to the process, and is considered as an input device.

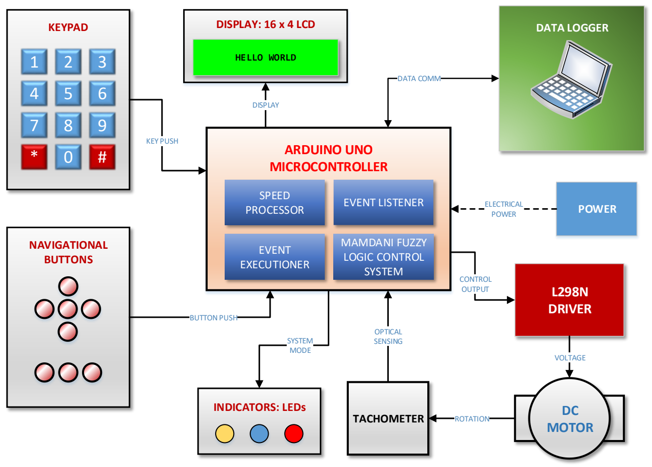

Fig. 4. Schematic system diagram.

C. *Sequencer*

As a user interface which allow user to determine set-points in time domain, a sequencer is built for this research using commands as follow:

- START SET-POINT, is defined as set-point (speed in RPM) definition with a time label marks the start point of a new speed demand.
- START, is a command to run the sequencer. The sequencer can only be started from beginning. It does not support pausing and resuming activity.
- STOP, is a command to terminate the process and quit immediately.

The sequencer must be defined at least for one command-line before the system can perform the test run.

D. *Navigation*

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

The second 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.

E. *DC Motor and Driver*

A driver is a series of current amplifiers from the Arduino Uno output to a larger current and voltage adjusted to DC motor specifications to power the DC motor. Unlike an AC motor that using a constant voltage but different frequency to control its speed, a DC motor uses voltage difference. A DC motor will be controlled by the driver, not directly from Arduino Uno. It can be assigned to different set-points, and like any other DC motor, it is controlled by a voltage variance. Technically speaking, the motor produces higher torque as input voltage increases.

F. *Tachometer*

This part 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 logged to get the DC motor speed readings to be compared with the program on the Sequencer.

IV. Results and Discussions

A. *System Modelling*

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

- Testing microcontroller process resolution.
- Defining speed range.
- Designing control mapping function between pin value and output voltage (from driver).
- Defining maximum speed.
- Designing fuzzy controller.
- Designing rule table
- Designing defuzzification process.

B. *Microcontroller Process Resolution*

Microcontroller speed is needed for determining process resolution. A slight difference of process rate occurs from datasheet caused by several factors, mimicking a real world asymmetric calculation load. Arduino Uno has 8µs process resolution according to its datasheet. However, the value needs to be verified.

Experimental results suggest that the maximum real-world process resolution is 12.6 µs. This result is approximately 57% slower than the Arduino Uno 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 50 times slower can be achieved, i.e. about 600 µs. A slow and locking process will only be executed every 50000 µs therefore it does not make significant impact to the overall system performance.

C. Rotational Speed Constraint According the Process Resolution

An optical chopper with 10 holes is used as the disk for sensor. The sensor works by calculating the interrupted time between each holes and barriers in the chopper which lets and blocks light from one side to the receiver side, respectively.

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

$$\omega = \frac{1000000}{2 \cdot r \cdot C_H} = \frac{1000000}{2 \cdot 600 \cdot 10} = 83.3Hz = 5000RPM \quad (1)$$

Where ω is the frequency of rotation, r is the resolution of the process, and CH is the number of holes in the chopper. At a speed higher than 5000 RPM, then when the sensor sends the switch information between the holes and the barriers there will be several undetected information as it goes beyond the process resolution, thus reduces speed reading accuracy.

While performing other relatively slow and locking processes performed every 50000 µs (50 ms), 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).

$$\omega = \frac{1000000}{2 \cdot r \cdot C_H} = \frac{1000000}{2 \cdot 50000 \cdot 10} = 1Hz = 60RPM \quad (2)$$

Equation (2) shows that the minimum rotational speed the process can calculate is 60 RPM. So the range of rotational speeds that can be read by the process is between 60 RPM to 5000 RPM.

D. Physical Speed Range 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.

Since the DC motor type is the brushed one, which has a large start speed inconsistency due to uneven brush wear, 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 presumed that the expected rotational speed range is between 500 RPM to 2500 RPM, both have forward and reverse rotation direction.

E. The Mamdani Fuzzy Logic 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.

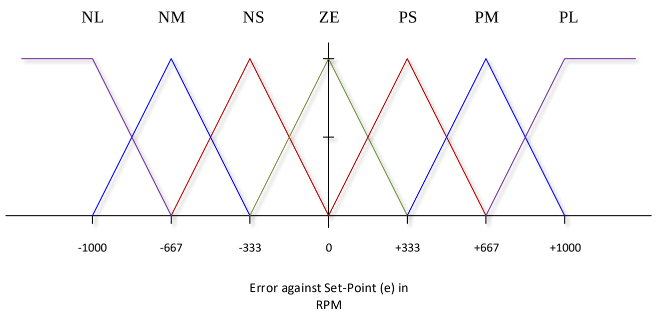

Fig. 5. Fuzzy membership function based on error of set-point.

The first part is the fuzzy membership degree function based on the input error of set-point, presented in Figure 5.

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

- NL, is a large negative value.
- NM, is an intermediate negative value.
- NS, is a small negative value.
- ZE, is a zero value.
- PS, is a small positive value.
- PM, is a medium positive value.
- PL, is a large positive value.

The seven types of membership are also used in the fuzzy membership degree function based on the derivative of error toward set-point:

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

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

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

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

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

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

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

Equation (3) to Equation (9) presented the functions of fuzzy membership degree based on input error of set-point.

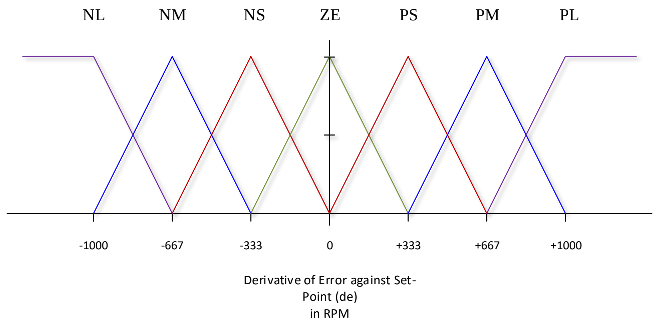

Fig. 6. Fuzzy membership function based on derivative of error of set-point.

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 process value toward set-point is ± 1000 RPM, i.e. half of the maximum DC motor rotational speed range constraint (2000 RPM), hereafter maximum transition requires at least two processes to smooth out the output and gives more chance of feedback. This method will suppress the level of overshoot 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, is adding a large positive number.
- APM, adding a moderate positive number.
- APS, is adding a small positive number.
- AZE, is adding nothing.
- ANS, is adding a small negative number.
- ANM, is adding an intermediate negative number.
- ANL, is adding 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 Uno. Therefore, it will determine the category of firm values (crisp values) on defuzzification as follows and shown graphically in the Figure 7:

- APL, with +030 value.
- APM, with +010 value.
- APS, with +005 value.
- AZE, with 000 value.
- ANS, with -005 value.
- ANM, with -010 value.
- ANL, with -030 value.

In order to simplify the expression, the parameter of defuzzification above can then be expressed as "**{±030-±010-±005-000}**" notation.

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.

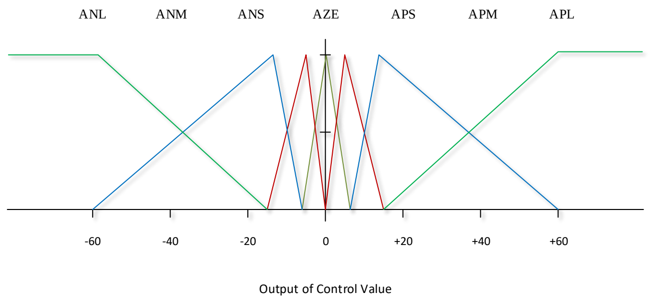

Fig. 7. Fuzzy memberships of control actions.

In this research, center of gravity method is used for defuzzification. 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.

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

Equation 10 denotes the defuzzification 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.

F. Testing Fuzzy Parameter

For testing purpose, there are several categories to determine how sequencer is programmed. Some categories in its scenario includes:

- Start the motor 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 30% of set-point difference.
- Suppress overshoot/undershoot for maximum 50% of set-point from load application or releasing.
- Suppress rising time for maximum 15 times of correction period. The correction period is 500 ms.
- Determine the error band for no more than 8% 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.

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 as shown in Figure 7.

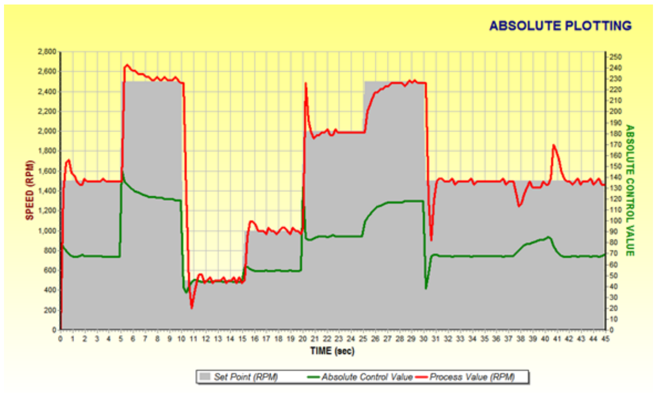

Fig. 8. Result of the first testing using **{±080-±030-±010-000}**.

Notation of sequencer is expressed as sequence of time and set-point with the "**[tttt:ssss-tttt:ssss-…-tttt:STOP]**" notation.

G. Testing Process

For a testing purpose, a sequencer is then determined with a sequence as follow: **[0000:1500-0005:2500-0010:0500-0015:1000-0020:2000-0025:2500-0030:1500-0045: STOP].** The defuzzification parameter is set first with **{±080-±030-±010-000}** to mimic responsive system, and then the second set to **{±030-±010-±005-000}** for increasing the system smoothness.

The first test is conducted as shown in Figure 8. The gray area is defined as set-point, while the red line refers to process value (speed), and the green line shows output control value (throttle). The second test is conducted as shown in Figure 9, with the defuzzification parameter reduced to **{±030-±010-±005-000}**.

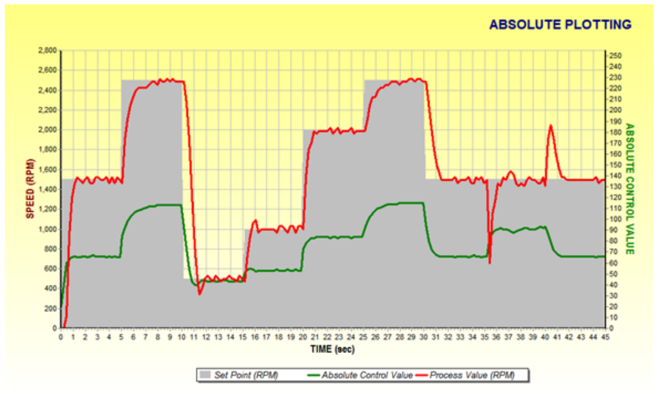

Fig. 9. Result of the second testing using **{±030-±010-±005-000}**.

H. Testing Result Analysis

From the test results obtained as follows:

- To obtain a minimum rise time, the order of the fuzzy set of control actions is **{±080-±030-±010-000}**. 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 **{±030-±010-±005-000}**. 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-±030-±010-000}** causes out-of-threshold overshoot and ringing but the rise time meets the specification.
- The defuzzification parameter **{±030-±010-±005-000}** minimizes overshoot and ringing into permissible value but a bad rise time.
- Overshoot and ringing are the criteria which analog with system smoothness, while the rise time specifies system responsiveness.
- The second tes at time 35th showed the controller was able to maintain the stability of motor rotation at the absolute control value of 85, at the change the speed of the motor from 140 rpm to 60 rpm.

While this research has demonstrated the accuracy and ease of Mamdani Fuzzy Logic Control System, 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.

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