

*Jurnal Teknologi*, 45(D) Dis. 2006: 1–17  
© Universiti Teknologi Malaysia

## PERFORMANCE COMPARISON BETWEEN PID AND FUZZY LOGIC CONTROLLER IN POSITION CONTROL SYSTEM OF DC SERVOMOTOR

MOHD FUA'AD RAHMAT<sup>1</sup> & MARIAM MD GHAZALY<sup>2</sup>

**Abstract.** The objective of this paper is to compare the time specification performance between conventional controller and artificial intelligence controller in position control system of a DC motor. This will include design and development of a GUI software using Microsoft Visual Basic 6.0 for position control system experiment. The scope of this research is to apply direct digital control technique in position control system. Two types of controller namely PID and fuzzy logic controller will be used to control the output response. An interactive software will be developed to visualize and analyze the system. This project consists of hardware equipment and software design. The hardware parts involve in interfacing MS150 Modular servo System and Data Acquisition System with a personal computer. The software part includes programming real-time software using Microsoft Visual Basic 6.0. Finally, the software will be integrated with hardware to produce a GUI position control system.

**Key words:** virtual instrumentation, PID, fuzzy logic, position control system, servomotor, direct digital control, graphic user interface

**Abstrak.** Objektif kertas kerja ini adalah untuk menyiasat dan membandingkan prestasi spesifikasi masa diantara pengawal konvensional dan pengawal suatu perisian interaktif komputer akan direka dengan menggunakan Microsoft Visual Basic 6.0 yang berupaya mengawal kedudukan shaft motor dalam sistem motor servo. Skop projek ini adalah untuk menggunakan teknik kawalan terus digital dalam mengawal kedudukan shaft motor servo. Pengawal PID dan logik kabur akan digunakan untuk mengawal isyarat keluaran motor servo. Projek ini terbahagi kepada dua bahagian iaitu perkakasan dan perisian. Bahagian perkakasan terdiri daripada membina hubungan antaramuka antara MS150 Modular servo sistem dengan kad perantaramukaan dan komputer peribadi. Bahagian perisian pula adalah lebih kepada mereka bentuk perisian untuk pengawal PID dan logik kabur dengan menggunakan Microsoft Visual Basic 6.0.

**Kata kunci:** virtual instrumentation, PID, logik kabur, kedudukan sistem kawalan, motor servo, kawalan terus digital

<sup>1</sup> Department of Control and Instrumentation Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor. Email: [drfuaad@fke.utm.my](mailto:drfuaad@fke.utm.my)

<sup>2</sup> Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka (UTEM), Locked Bag 1200, 75450 Ayer Keroh Melaka.

## 1.0 INTRODUCTION

The evolution of programming paradigms was marked by the transition from machine language to assembly languages, and from these to functional, procedural, and, eventually, object-oriented, high-level languages. During these past years, a totally new paradigm evolved along these lines, offering problem solving approaches in a new dimension graphical data flow programming in the form of so called virtual instruments. Virtual instrumentation is defined as a visualization and centralization of complex measurement system on a standard personal computer in the form of a virtual user interface. A user sees a uniform comprehensive single system, consisting of many individual measuring components in a virtual instrumentation(VI).

The main purpose of this research is to design a software based program with Graphical User Interface (GUI) in controlling the position of a servomechanism or also known as servomotor. It involves understanding on how to interface personal computer (PC) with position control system through Data Acquisition System (DAS). This research can be divided into two parts; namely hardware and software. The hardware part involves interfacing MS150 Modular servo System and Data Acquisition System with a personal computer whereas the software part includes programming real-time software using Microsoft Visual Basic 6.0.

## 2.0 DIRECT DIGITAL CONTROL SYSTEM (DDC)

Direct Digital Control (DDC) technology gives today's industrial user the flexibility to tailor near any system to specific needs using convenient software-based control. A simple concept that has blossomed into a complex array of components and systems, DDC is basically a microprocessor-based technology in which a controller performs closed-loop functions via sophisticated algorithms and strategies. A typical schematic of direct digital control is shown in Figure 1.

The controller function handles inputs and outputs electronically while the software provides the logic. Within the controller, the analogue signal enters the system where it is converted to a digital form so that the data can be acted upon according to a set of parameters or formulas. The result is the execution of commands directly to end device. Since the programs used by the controllers are in digital form, the operations are known as direct digital control [2,3].

DDC control brings speed, precision, flexibility to control functions at low cost. It can range from simple control of a single loop to the application of larger systems controlling a multitude of loops. Controllers may function as stand-alone units or be networked into systems using a PC as a host to provide additional functions. Most intelligent DDC devices residing on a network today are capable of sophisticated equipment control and remote communication.

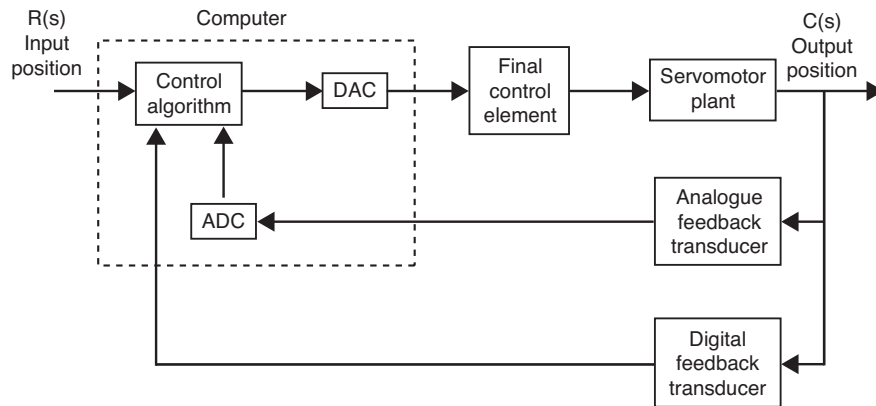


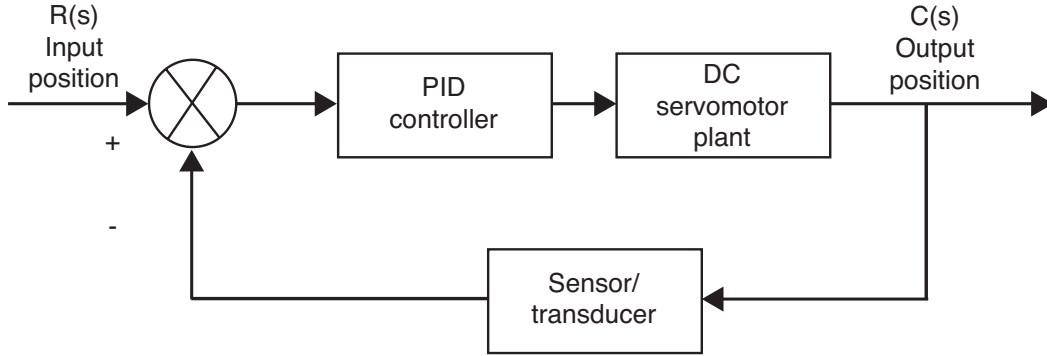
Figure 1 Direct digital control schematic

## 2.1 PID Controller

PID stands for Proportional-Integral-Derivative. This is a type of feedback controller whose output, a control variable (CV), is generally based on the error between some user-defined set point (SP) and some measured process variable (PV). Each element of the PID controller refers to a particular action taken on the error.

There are many situations that require some type of servo-control system. This section reviews the fundamental of PID controllers and presents detailed simulations or design for development of digital servomotor controller. PID controllers are commonly used to regulate the time-domain behaviour of many different types of dynamic plants. These controllers are extremely popular because they can usually provide good closed loop response characteristics, can be tuned using relatively simple rules and are easy to construct using either analogue or digital components. Consider the feedback system shown in Figure 2 where we assume that the plan is a DC motor whose shaft position must be accurately regulated.

The PID controller  $K(s)$  is placed in the forward path, so that its output becomes the voltage applied to the motor's armature. The feedback signal is either an angular shaft position or velocity, measured by a potentiometer or a tachometer, respectively. In the block diagram, these transducer dynamics are in the feedback path. The output position signal  $y(t)$  (or velocity signal) is summed with a reference signal  $r(t)$ , or command signal, to form the error signal  $e(t)$ . Finally, the error signal is the input to the PID controller.



**Figure 2** General feedback architecture

## 2.2 Digital PID Controller

The PID controller is defined by the following relationship between the controller input  $e(t)$  and the controller output  $v(t)$  that is applied to the motor armature:

$$v(t) = K_p E(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (1)$$

To discretise the controller, we need to approximate the integral and the derivative terms to form suitable for computation by a computer. By using z-transform, we may discretise the analogue PID controller to obtain the pulse transfer function for the digital PID controller as the following equation [4].

Positional Form of the PID control scheme is given as;

$$G(z) = \frac{C(z)}{R(z)} = K_p + \frac{K_I}{1 - z^{-1}} + K_D(1 - z^{-1}) \quad (2)$$

where

$$K_p = \text{proportional gain} = K - \frac{KT}{2T_i} = K - \frac{K_I}{2}$$

$$K_I = \text{Integral gain} = \frac{KT}{T_i}$$

$$K_D = \text{Derivative gain} = \frac{KT_d}{T}$$

Equation (2) can be written in a sequence operation as in Equation (3). Equation  $u(k)$  will be used to develop the coding program for PID controller which corresponds with sequence operation [5].

$$u(k) = K_p e(k) + K_i T_s \sum_{n=0}^k e(n) + K_d T_s^{-1} [e(k-1)] \quad (3)$$

where  $T_s = \text{time sampling} = 1\text{ms}$

$e(k) = \text{error between desired value and actual value}$

$$\sum_{i=0}^t e(k) = \text{total error}$$

$e(k) - e(k-1) = \text{deviation error}$

The values of  $K_p$ ,  $K_i$  and  $K_d$  are calculated using the root locus technique. The tuning method used is the Ziegler Nichols. The PID controller has been simulated using MATLAB to ensure the manual calculation is correct. The values of  $K_p$ ,  $K_i$  and  $K_d$  are as follow:

$$K_p = 1.266$$

$$K_i = 0.1261$$

$$\text{thus } T_i = \frac{K_d}{K_i} = 10.0444 \approx 10$$

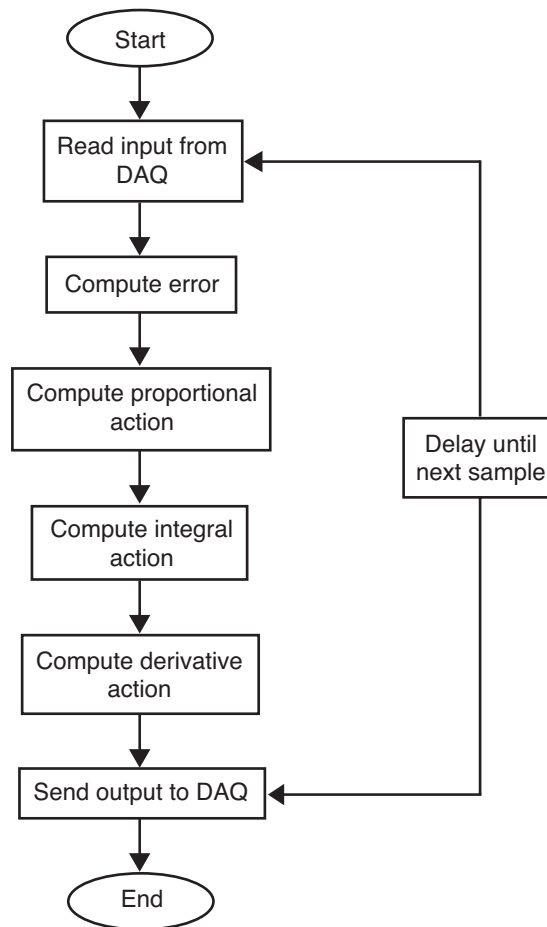
$$K_d = 0.0563$$

$$\text{thus } T_d = \frac{K_d}{K_p} = 0.044 \approx 1.0$$

After calculating these values, the coding for the PID controller can be written. The flow chart for the system can be virtualized as in Figure 3.

### 2.3 Fuzzy Logic Controller (FLC)

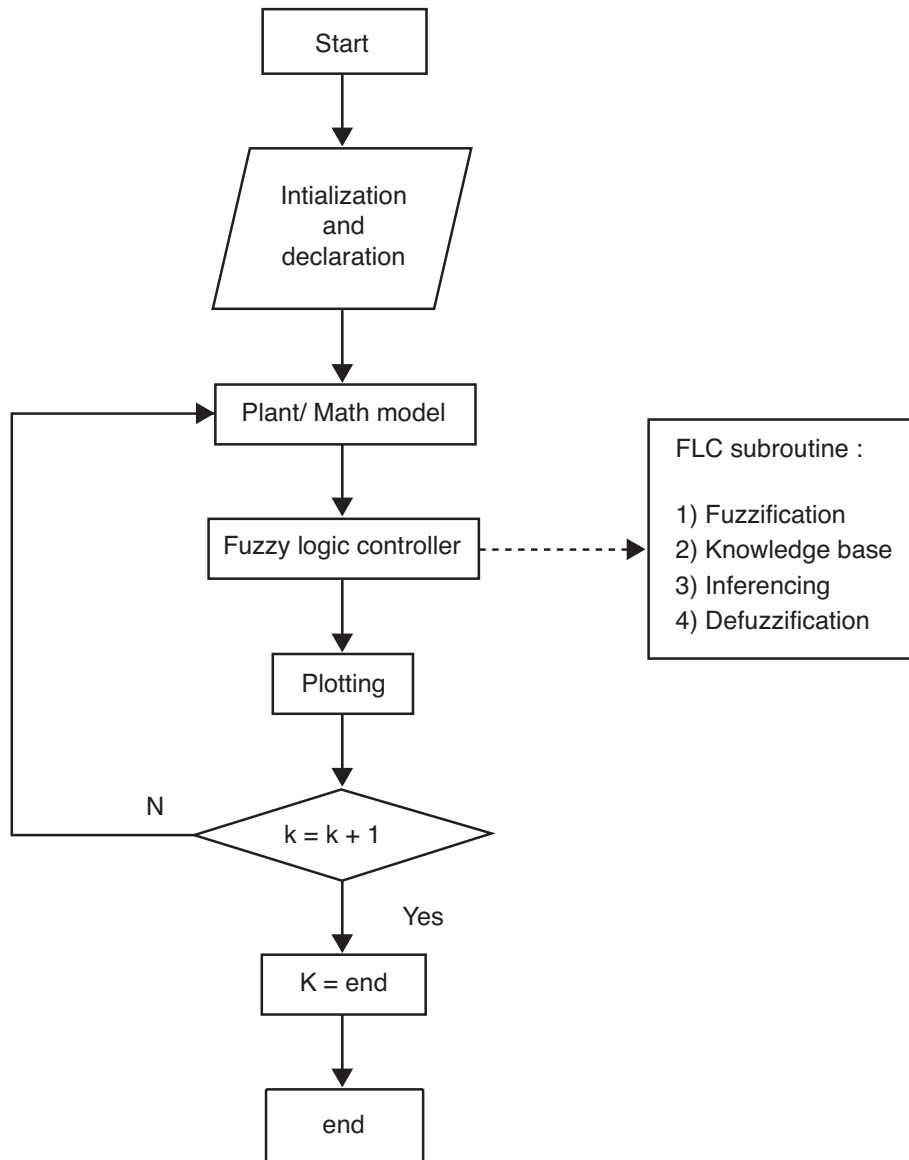
Figure 4 is the flow chart in designing the FLC. In developing the design, we need to initialize and declare what the inputs and outputs variables are. Then, define the mathematic model so that we could carry out simulation using the previous value and the changes in the output. All parameters are then used by FLC to be simulated. FLC is a subroutine of the main simulator program designed in this project. The FLC design methodology is shown in Figure 5.

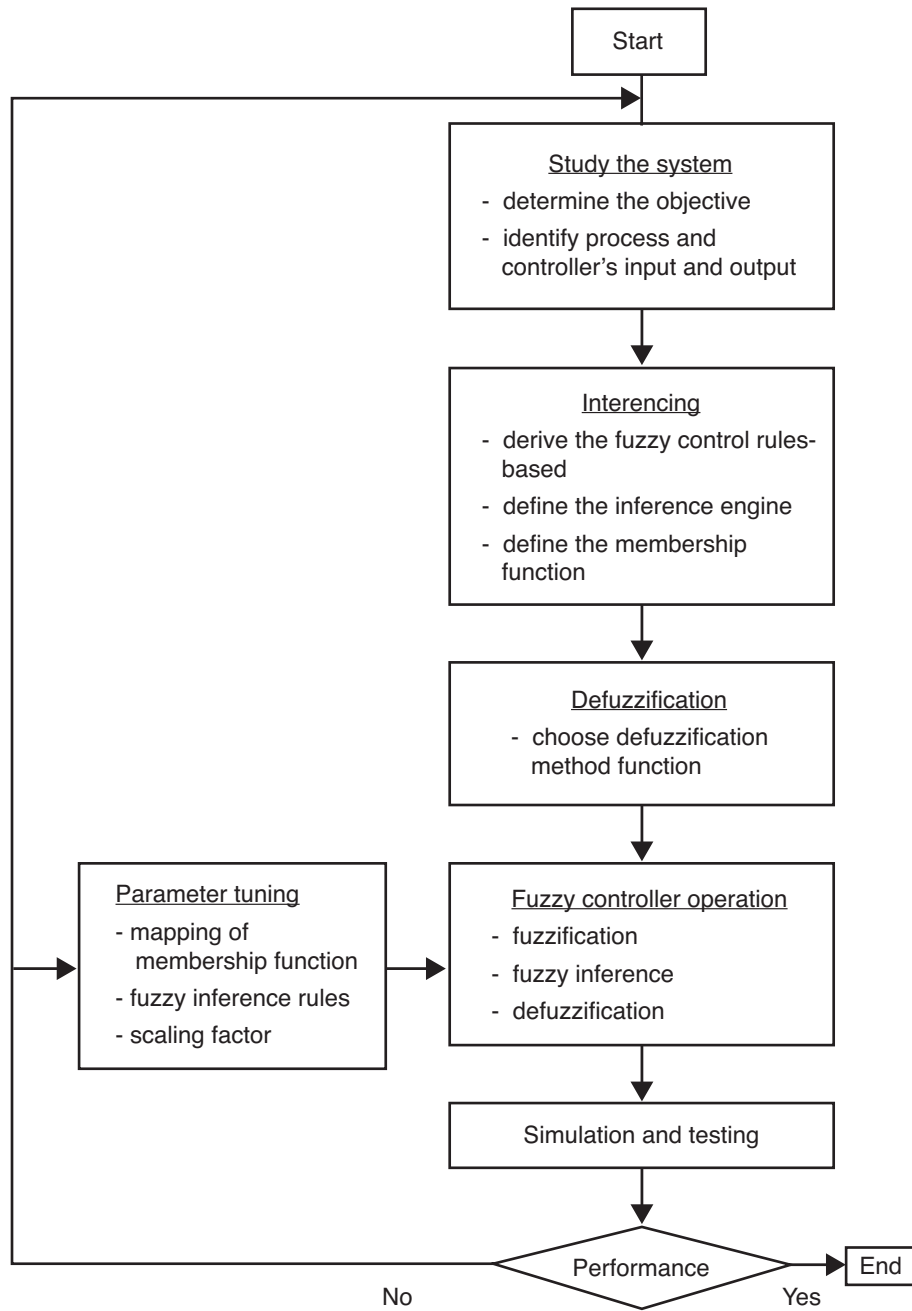


**Figure 3** Flow chart for the PID controller system

## 2.4 Fuzzification

The position control of the servo system configuration was designed based on 2 inputs and 1 output. Inputs for this controller are error and rate (de), and the motor output. Fuzzification is where we define the quantization and membership function for input variable, error and rate (de) and output variable, motor output in universe of discourse. It involves the conversion of the input and output signals into a number of fuzzy represented values (fuzzy sets). Figure 6 shows the input and output variables that are used in the system.

**Figure 4** Flowchart of simulator



**Figure 5** FLC design methodology



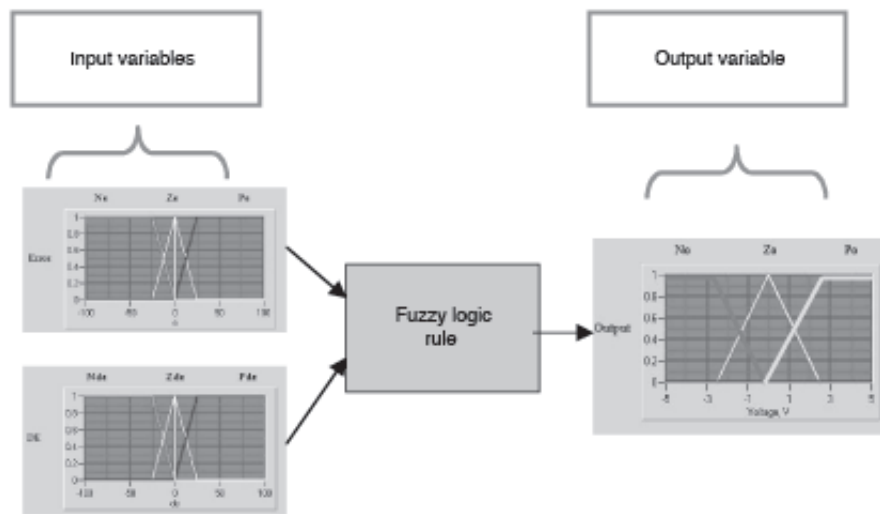


Figure 6 The input and output variable

Quantization of input and output variables:

(i). Input variables :

• **Error( $e$ )**

Quantized into 3 fuzzy sets: negative error (Ne), zero error (Ze) and positive error (Pe). Figure 7 shows the error input variable.

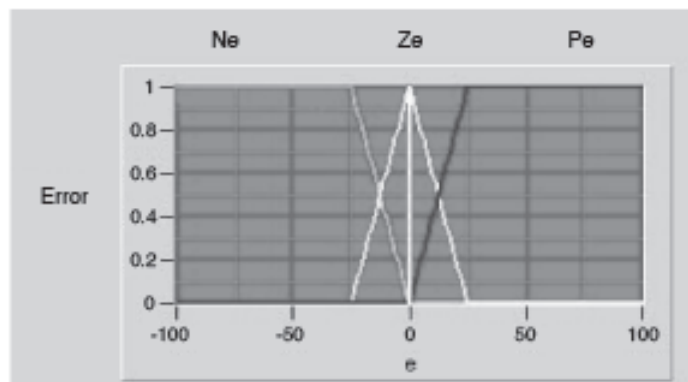
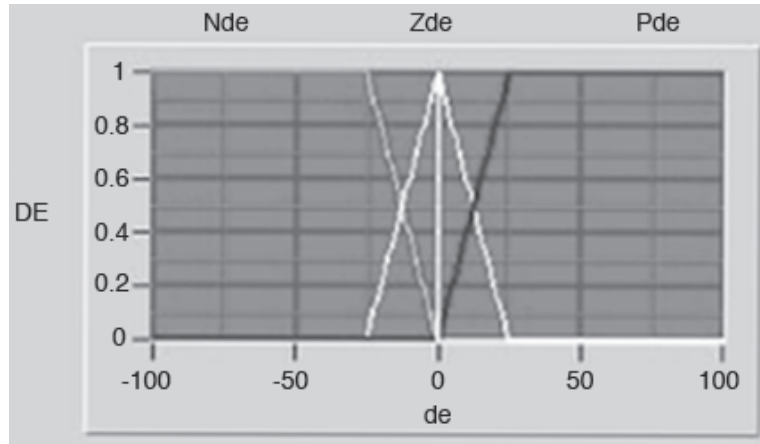


Figure 7 The error input variable

- ***Rate(de)***

Quantized into 3 fuzzy sets: negative rate (Nde), zero rate (Zde) and positive rate (Pde). Figure 8 shows the rate input variable.

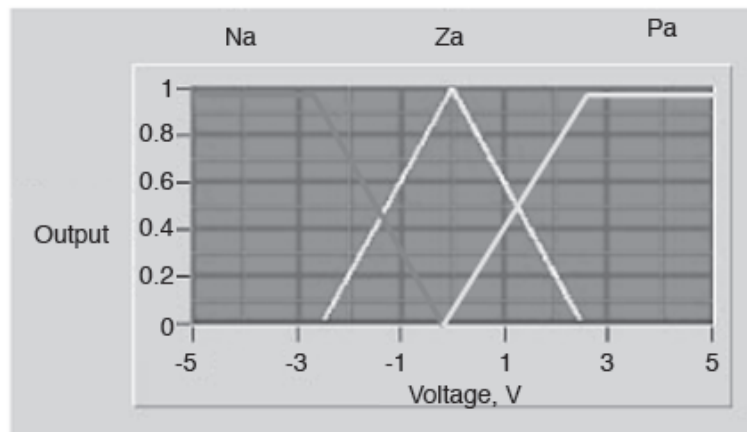
(ii). Output variable:



**Figure 8** The rate input variable

- ***Motor output***

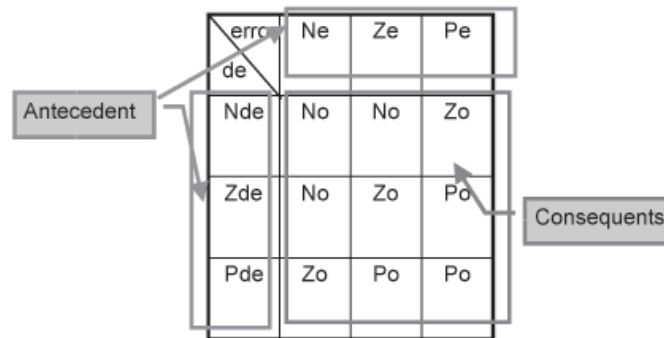
Quantized into 3 fuzzy sets: negative output (No), zero output (Zo) and positive output (Po). Figure 9 shows the motor output variable.



**Figure 9** The motor output variable

## 2.5 Knowledge Based

The basic function of the rule based is to represent the expert knowledge in a form of if-then rule structure. The fuzzy logic can then be derived into a  $3 \times 3$  rule which consist of 9 rules. Figure 10 shows the fuzzy logic rules.



**Figure 10** The fuzzy logic rules

The fuzzy logic rules-based are:

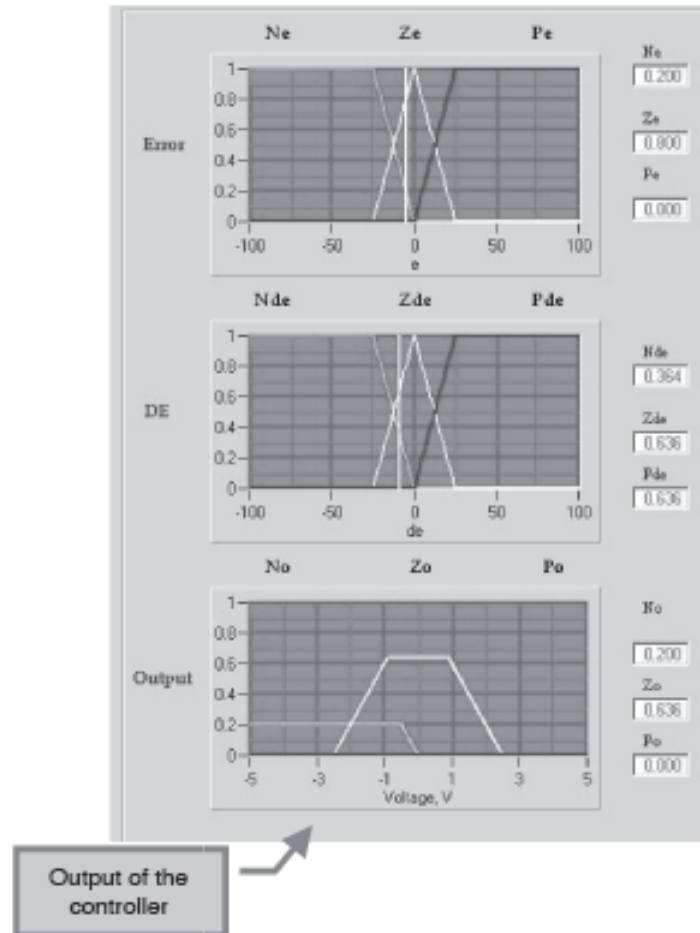
- 1) If error is Ne AND de is Nde THEN output is No
- 2) If error is Ne AND de is Zde THEN output is No
- 3) If error is Ne AND de is Pde THEN output is Zo
- 4) If error is Ze AND de is Nde THEN output is No
- 5) If error is Ze AND de is Zde THEN output is Zo
- 6) If error is Ze AND de is Pde THEN output is Po
- 7) If error is Pe AND de is Nde THEN output is Zo
- 8) If error is Pe AND de is Zde THEN output is Po
- 9) If error is Pe AND de is Pde THEN output is Po

## 2.6 Inferencing

The inference mechanism provides the mechanism for invoking or referring to the rule base such that the appropriate rules are fired. There are several methodologies to derive the Inferencing method. Two most common methods used in fuzzy logic control are the max-min composition and the max-(algebraic) product composition. The inference or firing with this fuzzy relation is performed via the operations between the fuzzified crisp input and the fuzzy relation

representing the meaning of the overall set of rules. As a result of the composition, one obtains the fuzzy set describing the fuzzy value of the overall control output.

For this system, max-min composition is used for the inferencing. Figure 11 shows the firing strength for the rule base using max-min inference.



**Figure 11** The firing strength for the rule base using max-min inference

## 2.7 Defuzzification

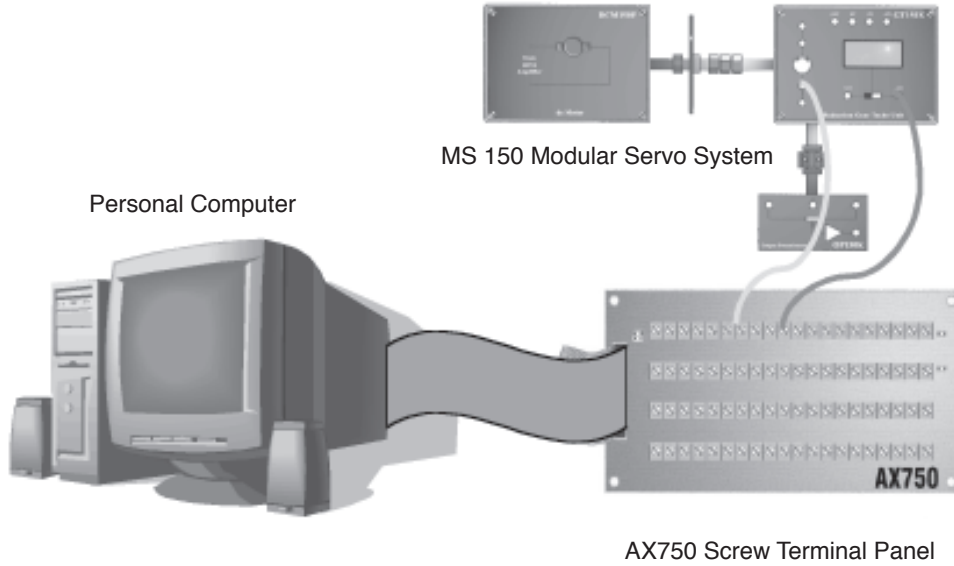
For this system, centroid method is used for defuzzification. It is given by the algebraic expression:

$$z^* = \frac{\sum u_z(z).z}{\sum u_z(z)} \quad (4)$$

## 2.8 Hardware Connection

Hardware connection is done by interfacing between Personal Computer and MS150 Modular Servo System through AX5412H Data acquisition card. The connection is shown in Figure 12. There are 8 differential inputs (or 16 single input) and 2 outputs from the AX5412 data acquisition card. Only 2 differential inputs are used for measurement and both outputs are used to control the direction and speed of the motor. The connection between MS150 Modular Servo System and AX5412H data acquisition card will be shown in the experiment procedure respectively.

The driver used in this project is the Axiom Device Manager version 1.1 where AX5412H is set at based address 300 with differential input and output range set at 10v bipolar.



**Figure 12** Connection between PC and MS150 through AX750 Screw Terminal Panel

### 3.0 PID RESULTS

The waveform obtained from the direct digital control experiment is then analyzed. The terms defined above are used for each combination of PID controller. Figure 13 shows the result which is captured within the GUI. Table 1 shows the analysis of the best output result which is captured within the GUI. All the system characteristics terms are analyzed using Yokogawa DL1546 Digital Oscilloscope. The unit used for rise time,  $T_r$ , settling time,  $T_s$ , peak time,  $T_p$  are all in milliseconds (ms). %OS is in percentage (%),  $\zeta$  is dimensionless,  $\omega_n$  is rad/s and unit for steady state error,  $e_{ss}$  is in Volt.

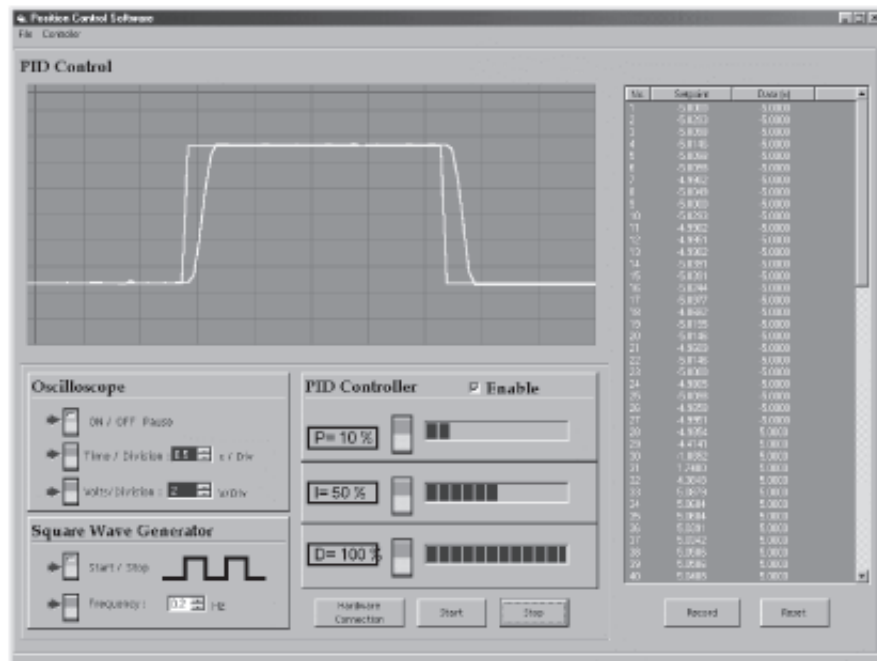
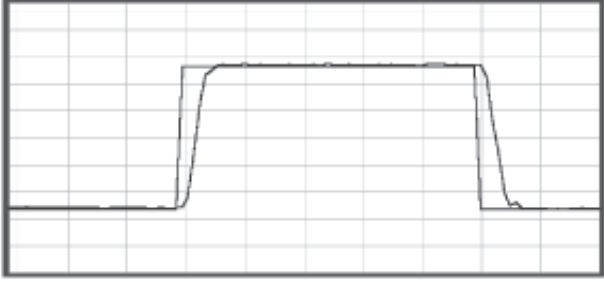


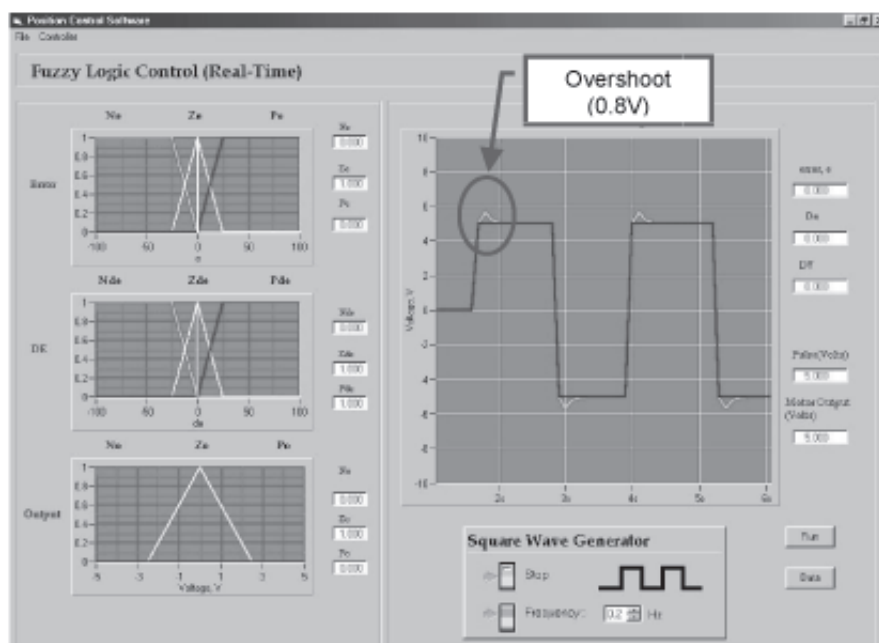
Figure 13 PID result which was capture within the GUI

### 3.1 Fuzzy Logic Results

The waveform obtained through the direct digital control experiment was analyzed. The terms defined above were used for the fuzzy logic controller. Figure 14 shows the result which is captured within the GUI. From the capture result, it can be concluded that the output response of the fuzzy logic controller has a percentage overshoot of 0.8V that is 16% overshoot.

**Table 1** The analysis of PID controller output result

Gain (%)	System characteristic	Output waveform
P=10 I=100 D=50	$T_r = 415.38$ $T_p = 476.92$ $T_s = 538.46$ $\%OS = 0$ $\zeta = 0.66$ $\omega_n = 11.27$ $e_{ss} = 0.10$	

**Figure 14** The fuzzy logic result which is captured within the GUI

#### 4.0 DISCUSSION

The output response of both the PID and Fuzzy logic were compared. Table 2 shows the comparison between the output response using the PID and Fuzzy logic. From the analysis and comparison of both controllers, the PID controller

performs better compared to Fuzzy logic in terms of percentage overshoot, with 0% overshoot compared to fuzzy logic with an overshoot of 16%. But, when comparing both of the rise time and settling time of both controllers, the fuzzy logic controller performs better, with a  $T_r$  of 400.00 ms and  $T_s$  of 100 ms.

**Table 2** Comparison output between PID and fuzzy logic controller

	PID (best response)	FLC
$T_r$	415.38ms	400.00ms
$T_s$	538.46ms	100ms
%OS	0%	16.0%

## 5.0 CONCLUSION

PID controllers are commonly used to regulate the time-domain behavior of many different types of dynamic plants. These controllers are extremely popular because they can usually provide good closed loop response characteristics, can be tuned using relatively simple rules and are easy to construct using either analogue or digital components. To discretise the PID controller, we need to approximate the integral and the derivative terms to forms suitable for computation by a computer. By using z-transform, we may discretise the analogue PID controller to obtain the pulse transfer function for the digital PID controller. Then, equation  $u(k)$  is used to develop the coding program for the PID controller which corresponds with the sequence operation.

To design the fuzzy logic controller, the position control of the servo system configuration is designed based on 2 inputs and 1 output. Inputs for this controller are error and rate (de), and the motor output. Fuzzification is where the quantization and membership functions for input variable, error and rate (de) and output variable, motor output in universe of discourse are defined. It involves the conversion of the input and output signals into a number of fuzzy represented values (fuzzy sets). The knowledge based of a fuzzy logic controller consists of a data based and a rule based. The basic function of the rule based is to represent the expert knowledge in the form of if-then rule structure. The fuzzy logic was derived into a  $3 \times 3$  rule which consist of 9 rules. For this system, max-min composition is used for the Inferencing to obtain the fuzzy set describing the fuzzy value of the overall control output. Defuzzification is a



mapping from a space of fuzzy control actions defined over an output universe of discourse into a space of non-fuzzy (crisp) control action. For this system, centroid method is used for defuzzification.

## REFERENCES

- [1] Axiom AX5412-H Data Acquisition Board User Manual
- [2] Auslander, D.M. 1990. *Real-time Software for Control*. Prentice Hall.
- [3] Perdikaris, G.A. 1991. *Computer Controlled Systems-Theory and Application*. Kluwer Academic Publisher.
- [4] Ogata, K. 1995. *Discrete Time Control Systems*. 2<sup>nd</sup> edition. Prentice Hall.
- [5] VanLandingHam, H.F. 1985. *Introduction to Digital Control System*. Macmillan Publishing Company.