Interval Type 2 Fuzzy Logic Tuning for PID Controller of DC Servo Motors

by

Phan Nguyen Quy Nhon

14159

Supervisor: Assoc. Prof. Dr. Irraivan Elamvazuthi

UNIVERSITI
TEKNOLOGI
PETRONAS

Dissertation Submitted in partial fulfillment of
the requirements for the
Bachelor of Engineering (Hons)
(Electrical and Electroncis Engineering)
December, 2013

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Interval Type 2 Fuzzy Logic Tuning for PID Controller of DC Servo Motors

by

Phan Nguyen Quy Nhon

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL AND ELECTRONCIS ENGINEERING)

Approved by,

____

(Main Supervisor: Assoc. Prof. Dr. Irraivan Elamvazuthi)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

December, 2013

CERTIFICATION OF ORGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Phan Nguyen Quy Nhon

Abstract

This report presents the design of Interval Type 2 Fuzzy Logic Tuning for PI Controller of DC servo motors project. DC servo motors have been in use extensively for many applications vary from industrial to electronics to consumers. However, its conventional PID controller still induces several problems such as unexpected response in non-linear systems, poor response when there is frequent disturbance. A new solution for PID controller of DC servo motor is proposed, that is to tune the PID controller automatically with Interval Type-2 Fuzzy Logic.

3 controllers including conventional PID, type-1 fuzzy tuned PID and interval type-2 fuzzy tuned PID are analyzed and compared extensively based on their response to step input in different working conditions of varying load and noise disturbance. Rise time, percent overshoot, integral absolute error and integral time-weighted absolute error are used to judge controllers' performance.

Matlab Simulink is the software tool to simulate and collect results. PID Tuner and Type-1 Fuzzy Toolbox are used for optimizing respective controllers. Comparative analysis work shows that fuzzy logic can be used for auto tuning PID controller and gives better result than conventional PID. Moreover, interval type-2 fuzzy tuned PID is the best solution for working conditions consist of uncertainties like varying load or noise disturbance.

Table of Contents

Lists of Figures

Lists of Tables

Abbreviations

DC: Direct Current

PID: Proportional, Integral and Derivative

T1FL: Type-1 Fuzzy Logic

T2FL: Type-2 Fuzzy Logic

IT2FL: Interval Type-2 Fuzzy Logic

MF: Membership Function

LMF: Lower Membership Function

UMF: Upper Membership Function

FOU: Footprint Of Uncertainty

IAE: Integral Absolute Error

ITAE: Integral Time-weighted Absolute Error

Chapter 1: Introduction

1.1 Background of study

Servo motors are used for precise positioning and speed control. The name "servo" means that they provide feedback signal for closed loop control scheme to improve control performance [1]. DC servo motors have been widely in use for decades in automatic steering, radar tracking, computer disk drives, and industrial manufacturing robots etc. The reason for DC servo motor's popularity is that it offers fast response, relative simple characteristic curve, and high efficiency [2].

As stated, servo motors provide feedback signal for controlling, and the most basic and prevailing continuous feedback controller is PID controller. PID controller regulates the corrective signal (manipulated variable) based on the error between output feedback signal (measured process variable) and expected output signal (set point). There are still several problems with PID control such as: low efficiency in non-linear system, degraded performance with disturbance, and requirement of manual tuning. Many researches were attempted to mitigate these defects utilizing genetic algorithm, neural network, sliding-mode control, and fuzzy logic.

Fuzzy logic concept was introduced by professor Zadeh [3] in 1965 to deal with uncertainties. One of the most popular uncertainties is intuitive rationality, in which human brain describes objectives without exact measurement, e.g. fast, slow, big, and small. Other sources of uncertainties are insufficient data or approximate estimation. The most objective of fuzzy logic is to provide electronic devices the ability of dealing with variables which have no strict boundaries of variation.

Recently in intelligent control, DC servo motor control using type 1 fuzzy logic (T1FL) is quite popular with many researches like [4], [5], [6], and [7]. However, the initial fuzzy logic concept did not completely solve imprecise problems since there are still uncertainties within itself [8]. These remaining uncertainties may come from the fuzzy sets that were defined arbitrarily (no guaranty of precise range) or from the noisy numerical data, inaccurate measurement, or disturbance [9].

For improvement of original fuzzy logic (type 1 fuzzy logic), in 1975, professor Zadeh introduced type 2 fuzzy logic (T2FL) that increases 1 more degree of freedom to handle uncertainties [10]. Hence T2FL offers the potential to better the performance of T1FL controller, specifically for DC servo motors.

1.2 Problems Statement

Although previous researchers suggested that controllers utilizing T1FL are capable of providing better performance, there is still room for further improvement with T2FL. This T2FL theoretically cures uncertainties more effectively but it is also more difficult to work on due to its complicated composition [8]. Recently, T2FL systems have been an attractive research area since lots of unexplored factors remain in control applications. Hence, more work is required for better understanding of T2FL performance in controlling, particularly for DC servo motors.

Moreover, the variation of membership functions, rules and different methods of fuzzification and defuzzification in T2FL provide different results and are still case-specific. More cases are being simulated by researchers to generalize the behavior of fuzzy sets. Doubt on whether PID controllers still have a better response than fuzzy sets in motor control needs to be proved.

Lastly, because type-2 fuzzy set is a three dimension concept, it brings difficulties to define, manipulate and express into fuzzy rules. Hence, a visual tool is necessary to allow users to interact with the type-2 fuzzy system more easily.

1.3 Objectives & Scope of Study

1.3.1 Objectives

Regarding T2FL application to DC servo motor control, this project aims to satisfy the following objectives:

- ✓ Model a typical DC servo motor operation with suitable parameters and simulate it with PID Controller.
- ✓ Explore & simulate Type-1 Fuzzy Logic Tuned PID Controller.
- ✓ Explore & simulate Interval Type-2 Fuzzy Logic Tuned PID Controller.
- ✓ Comparative analysis of conventional PID, T1FL and IT2FL tuned PID Controllers under different working conditions, e.g: varying load and noise.
- ✓ Develop user interface for Interval Type-2 Fuzzy Logic System.

1.3.2 Scope of Study

This project focuses on research, coding, simulation, analysis and providing visual aid for working with interval type-2 fuzzy logic, specifically applicable for DC servo motor. The research acquires DC servo motor modeling with PID Controller and fuzzy logic understanding. Coding involves developing Matlab functions to manipulate T1FL and IT2FL according to the requirements of the motor controller. Simulation includes construction of T1FL and IT2FL tuned PID virtual motor controller in Simulink environment, implementation and analysis of the system output, and comparison to conventional PID controller. Finally, graphical user interface is developed for the ease of interaction with type-2 fuzzy logic system.

Chapter 2: Literature Review

2.1 DC servo motor modeling

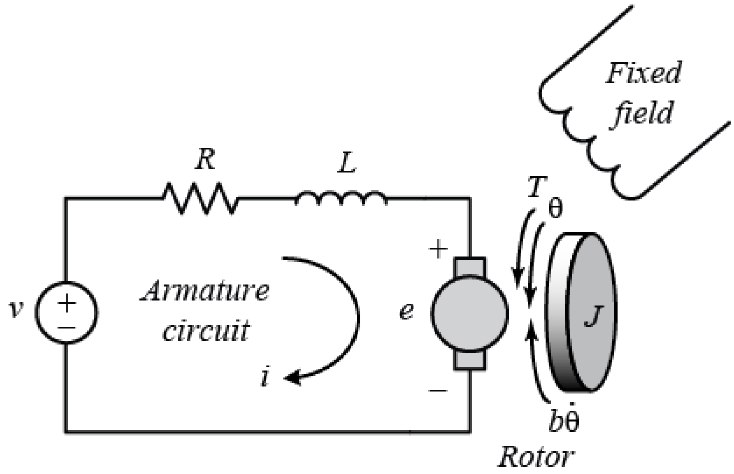


Figure 1. DC servo motor modeling free-body diagram

From [1], the modeling process starts with parameters assumption as follow:

V	input voltage, V
θ	output angular position, rad
ω	output angular velocity, rad/s
E	back emf voltage, V
J	moment of inertia of the rotor, kg.m^2
b	motor viscous friction constant, N.m.s
K_b	electromotive force constant, V/rad/s
K_t	motor torque constant, N.m/A
R	electric resistance, Ω
L	electric inductance, H

The torque generated by the DC motor is proportional to the armature current and the strength of the magnetic field. The magnetic field is assumed to be constant therefore the motor torque is proportional to only the armature current i with a constant K_t as shown in the equation below:

$$T = K_t * i$$

The back emf (electromotive force) voltage is proportional to the output shaft angular velocity by a constant factor K_b:

$$E = K_b * \omega$$

In which, the angular velocity is the derivative of angular position:

$$\omega = \frac{d\theta}{dt}$$

In SI units, the motor torque and back emf constants are equal: $K_t = K_b$; therefore, K is used to represent both of them. From the Figure 1 the following governing equations based on Newton's 2nd law and Kirchhoff's voltage law, are derived:

$$J\frac{d\omega}{dt} + b\frac{d\theta}{dt} = J\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} = Ki$$

$$L\frac{di}{dt} + Ri = V - E = V - K\frac{d\theta}{dt}$$

Applying Laplace transform, the above modeling equations can be expressed in terms of the Laplace variable s:

$$s(Js + b)\Theta(s) = KI(s)$$

$$(Ls + R)I(s) = V(s) - Ks\Theta(s)$$

The open-loop transfer function is derived by eliminating $I(s)$ in the two equations above, in which the rotational angular position $\Theta(s)$ is considered the output and the armature voltage $V(s)$ is considered the input:

$$P(s) = \frac{\Theta(s)}{V(s)} = \frac{K}{s\big((Js + b)(Ls + R) + K^2\big)}$$

Open-loop transfer function if the rotational angular velocity $\omega(s)$ is considered the output and the armature voltage $V(s)$ is considered the input:

$$G(s) = \frac{\omega(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2}$$

The block diagram of the DC servo motor system is shown below with K_b=K_t=K as mentioned:

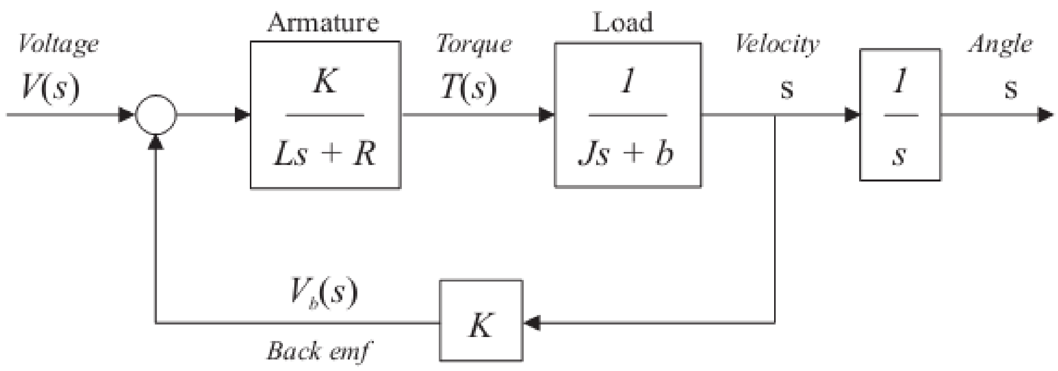


Figure 2. DC servo motor system block diagram

Another block diagram with load disturbance is Figure 3 with $K_b = K_m(K_t) = K$, K_f is b in above equations and T_d is the load disturbance.

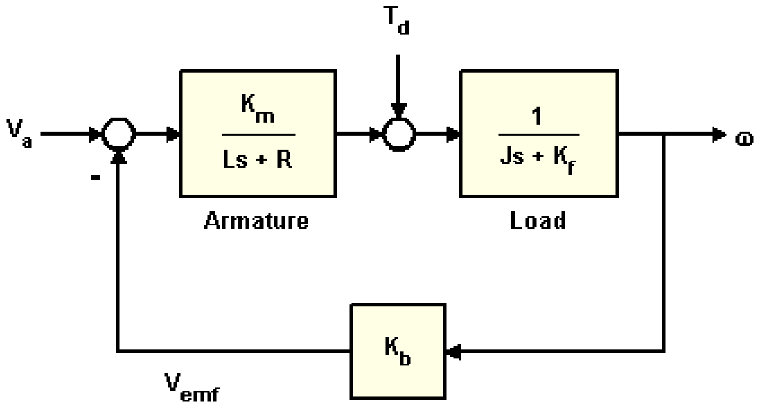


Figure 3. DC Servo motor system block diagram with load disturbance

There should be reminder that the back emf K_b loop is not feedback loop used for external control, it is the internal loop of the servo motor system.

<u>2.2 Conventional PID Controler for DC Servo motor</u>

Figure 4 shows the block diagram of a basic PID controller for DC servo motor with unity feedback in Simulink simulation:

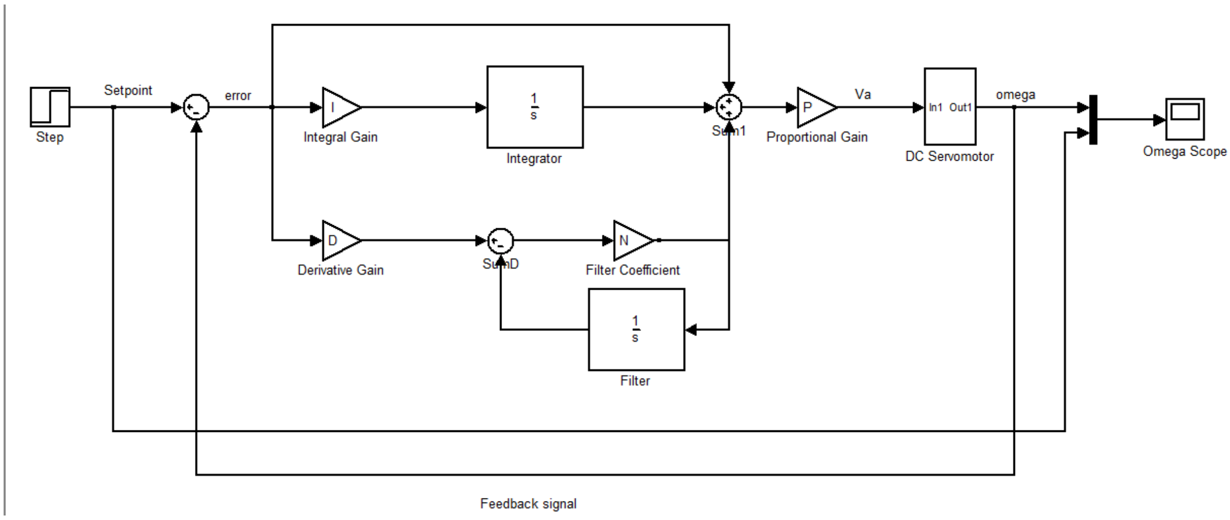


Figure 4. PID controller block diagram

The transfer function of PID Controller can be written as:

$$C(s) = K_P + \frac{K_I}{s} + K_D s$$

Differentiation is always sensitive to noise. In a practical, controller with derivative action needs to limit the high frequency gain of the derivative term. This can be done by implementing the derivative term as in transfer function of the PID controller:

$$P + I\frac{1}{s} + D\frac{N}{1 + N\frac{1}{s}}$$

The "DC servo motor system" block contains the transfer function derived above, that is:

$$G(s) = \frac{C(s)}{U(s)} = \frac{K}{(Js + b)(Ls + R) + K^2}$$

if $C(s)$ is the angular velocity $C(s)=\omega(s)$

Thus, the overall transfer function of PID controller for DC servo motor will be:

$$T(s) = \frac{G(s)C(s)}{1 + G(s)c(s)}$$

2.3 Fuzzy Logic Tuned PID Controller for DC Servo motor

2.3.1 Fuzzy Logic Concept

For a conventional set, an element is either a member or not a member of that set; hence, there is a clear distinction of the set boundary. Conventional sets are not flexible that allow the case of any element to be both member and non-member of the sets. For a fuzzy set, an element is partially a member of the set, and its membership measurement is displayed by its *membership grade* that lies between 0 and 1. This definition generalizes the case of crisp sets where membership value is either 0 or 1. Figure 4 [12] shows the membership grade of variable *age* in the fuzzy set of *old people*.

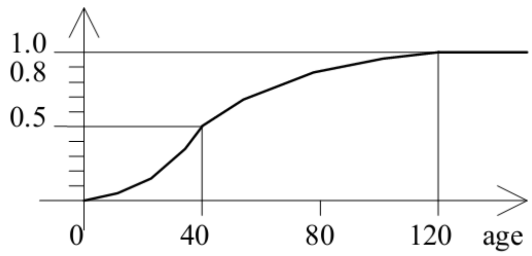


Figure 5. Example of Membership Function of Fuzzy Set of "Old"

A fuzzy set may have several subsets, and each of these subsets is represented by a membership function. The most common types of simple membership functions are triangular, trapezoidal and Gaussian functions as shown [12]:

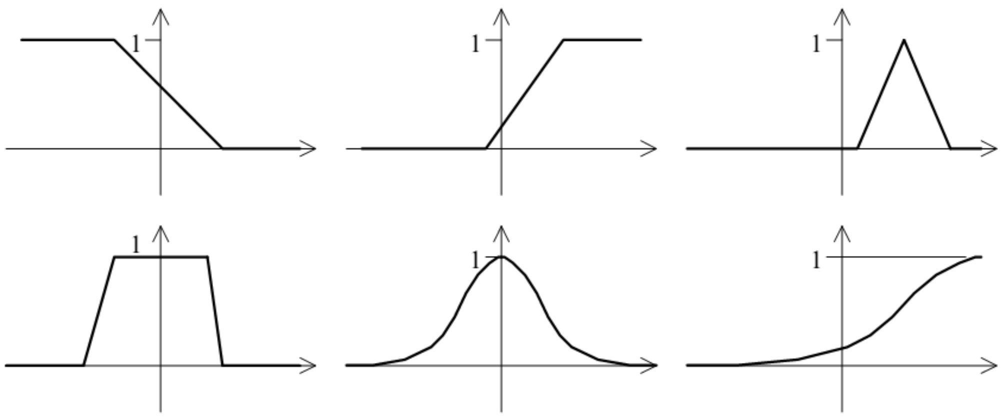


Figure 6. Common Types of Membership Function

There are 3 available operations between 2 fuzzy sets that are: union, intersection and implementation ([13], [14] and [12]). Union is interpreted to be *max* operator between membership values of fuzzy sets $\tilde{A}$ and $\tilde{B}$, while intersection is *min* operator and implementation is $1\text{-}\tilde{A}$. The graph [13] below illustrates these 3 operations:

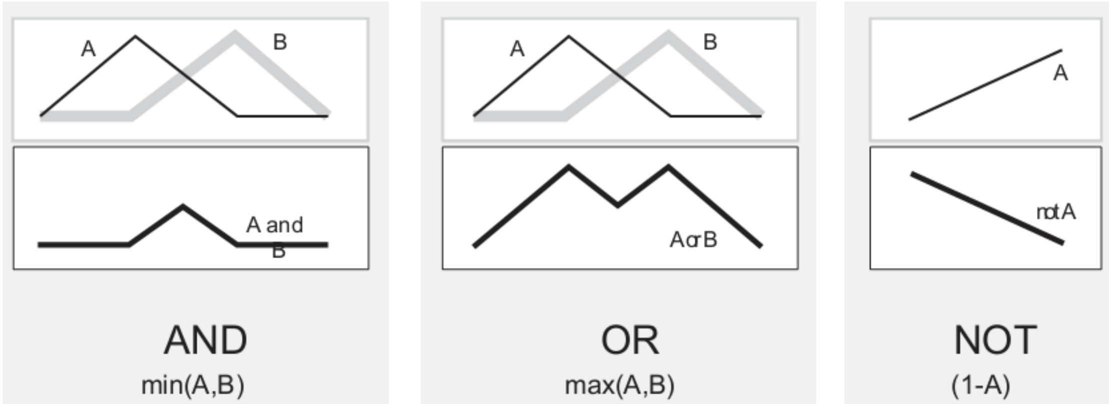


Figure 7. Operations on Fuzzy Sets

<u>2.3.2 Fuzzy Inference Systems and Fuzzy Work Flow</u>

Inference system, or in details, rules-based inference system, is considered the "processing unit" of a fuzzy logic system. This inference system consists of user-defined rules in the form of IF-THEN rules to manipulate fuzzy input(s) and produce desired fuzzy output(s), with the operators AND, OR and NOT connect different input(s). Figure 7 gives outline for tasks that an inference system performs, including:

- Fuzzification: transform crisp input(s) into fuzzy sets with membership functions.
- Combination: combine input(s) with corresponding operators AND, OR and NOT.
- Rules processing: produce desired consequence(s) from combined input(s) based on rules that were defined in rule base.
- Consequence(s) aggregation: combine consequence(s) to get a united output.
- Defuzzification: transform the single output fuzzy set back crisp value.

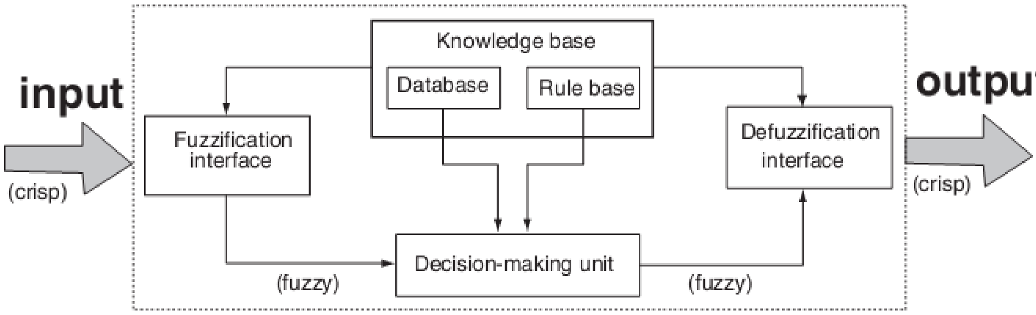


Figure 8. Inference System Outline

[14] and [12] introduce 2 types of available inference systems: Mamdani's fuzzy inference method and Takagi–Sugeno–Kang method. Mamdani method is the most common method and easier to understand.

2.3.3 Type-1 Fuzzy Logic Tuned PID Controller

The application of type-1 fuzzy logic into PID Controller is implemented based on the idea that fuzzy logic would help to improve the corrective signal according to the feedback error and rate of error change [15]. Figure 9 shows the block diagram of a discrete-time type-1 fuzzy logic PID controller. $e_c[k]$ is the error between set point and output, $e_r[k]$ is the rate of error change, K_p and K_i are obtained from PID controller and $u_{pi}[k]$ is the improved corrective signal compared to normal PID controller.

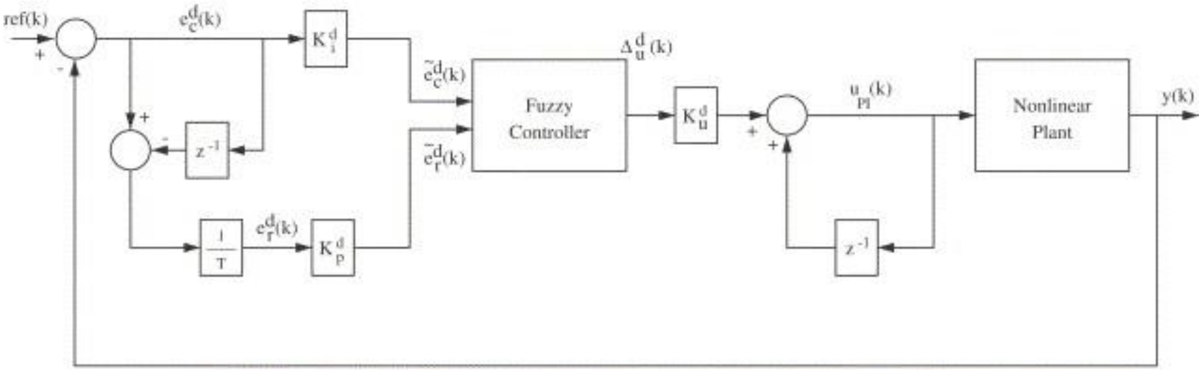


Figure 9. Discrete Fuzzy Logic PID Controller

The conventional correct action of PID controller in Laplace transform equation is:

$$U_{\mathrm{PI}}(s) = \left(K_{\mathrm{p}}^{\mathrm{c}} + \frac{K_{\mathrm{i}}^{\mathrm{c}}}{s} \right) E(s)$$

[15] showed that the equation can be converted into the discrete z -domain by applying bilinear transformation given by:

$$s = \left(\frac{2}{T}\right)\left(\frac{z-1}{z+1}\right) = \left(\frac{2}{T}\right)\left(\frac{1-z^{-1}}{1+z^{-1}}\right)$$

After several derivations and using inverse z-transform, the final summary is:

PI controller update: $u_{\mathrm{PI}}(nT) = u_{\mathrm{PI}}(nT-T) + K_{\mathrm{uPI}}\Delta u_{\mathrm{PI}}(nT)$

Incremental control output: $\Delta u_{\mathrm{PI}}(nT) = K_{\mathrm{p}}e_{\mathrm{v}}(nT) + K_{\mathrm{i}}e_{\mathrm{p}}(nT)$

Error signal: $e_{\mathrm{p}}(nT) = e(nT)$

Error signal rate of change: $e_{\mathrm{v}}(nT) = \dfrac{e(nT) - e(nT-T)}{T}$

Among the above parameters, the incremental control output (corrective signal) Δu_{pi} is the output of the fuzzy logic inference system. As observed, it is determined by error e_p and error change rate e_r according to fuzzy rules as follow:

(R1) IF $\tilde{e}_{\mathrm{p}} = \tilde{e}_{\mathrm{p}} \bullet n$ AND $\tilde{e}_{\mathrm{v}} = \tilde{e}_{\mathrm{v}} \bullet n$, THEN $\Delta u_{\mathrm{PI}}(nT) = o \bullet n$

(R2) IF $\tilde{e}_{\mathrm{p}} = \tilde{e}_{\mathrm{p}} \bullet n$ AND $\tilde{e}_{\mathrm{v}} = \tilde{e}_{\mathrm{v}} \bullet p$, THEN $\Delta u_{\mathrm{PI}}(nT) = o \bullet z$

(R3) IF $\tilde{e}_{\mathrm{p}} = \tilde{e}_{\mathrm{p}} \bullet p$ AND $\tilde{e}_{\mathrm{v}} = \tilde{e}_{\mathrm{v}} \bullet n$, THEN $\Delta u_{\mathrm{PI}}(nT) = o \bullet z$

(R4) IF $\tilde{e}_{\mathrm{p}} = \tilde{e}_{\mathrm{p}} \bullet p$ AND $\tilde{e}_{\mathrm{v}} = \tilde{e}_{\mathrm{v}} \bullet p$, THEN $\Delta u_{\mathrm{PI}}(nT) = o \bullet p$

Where *n* means negative, *z* means zero and *p* means positive.

2.3.4 Type-2 Fuzzy Logic Tuned PID Controller

2.3.4.1 Type 2 Fuzzy Set Theory and Work Flow

Type 2 fuzzy sets are the extended version of original (type 1) fuzzy set. In type 2 fuzzy sets, the membership function itself is a fuzzy concept. As shown in Figure 11, type 1 membership function is represented by a sharp and rigorous line or curve. For type 2, it is not a single line or curve anymore but a region with the shape varies depend on user definition.

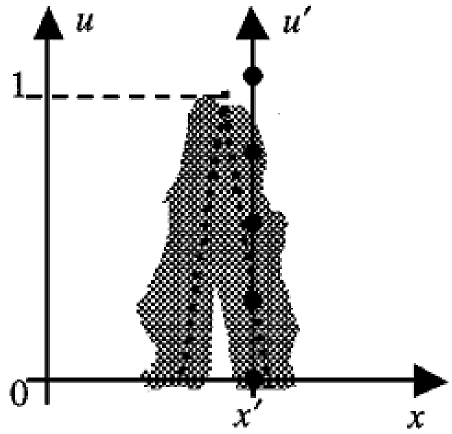


Figure 10. Type 2 Fuzzy Membership Function

Each element in the shade (type 2 fuzzy) region has its own membership grade in the fuzzy set. Consider $\tilde{A}$ is a type 2 fuzzy set where variable *x* has its *primary membership* grade is *u*. We can see from Figure 3 that each *x (*in this case *x=x'*) may have several corresponding *u (u=u')* which lie on the vertical line in the type 2 fuzzy region (shade). Each of these *u'* has its own secondary membership value named $\mu_{\tilde{A}}(x,u)$. So, the type 2 fuzzy set is fully represented only with a 3 dimensional diagram:

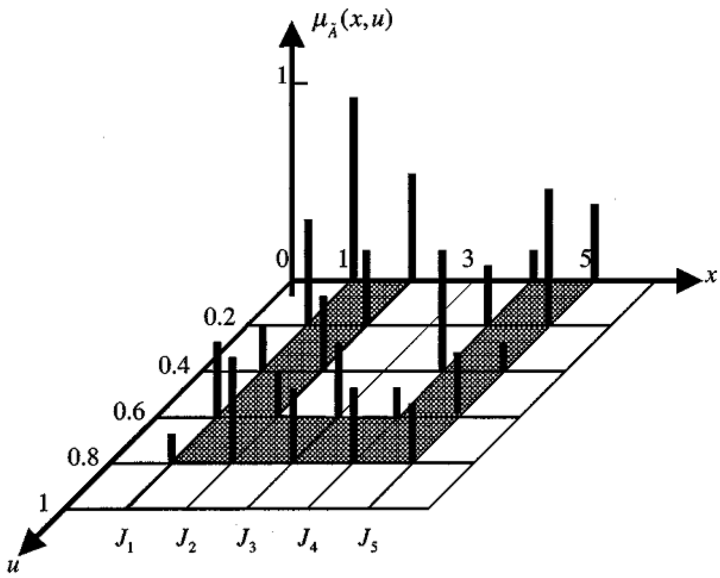


Figure 11. 3D Representation of Type 2 Fuzzy Set

The shaded region created by fuzzy variable *x* and its *primary membership* value *u* is called *footprint of uncertainty (FOU).* The FOU can be represented by 2 normal membership functions which are called *upper membership function (UMF)* and *lower membership function (LMF).*

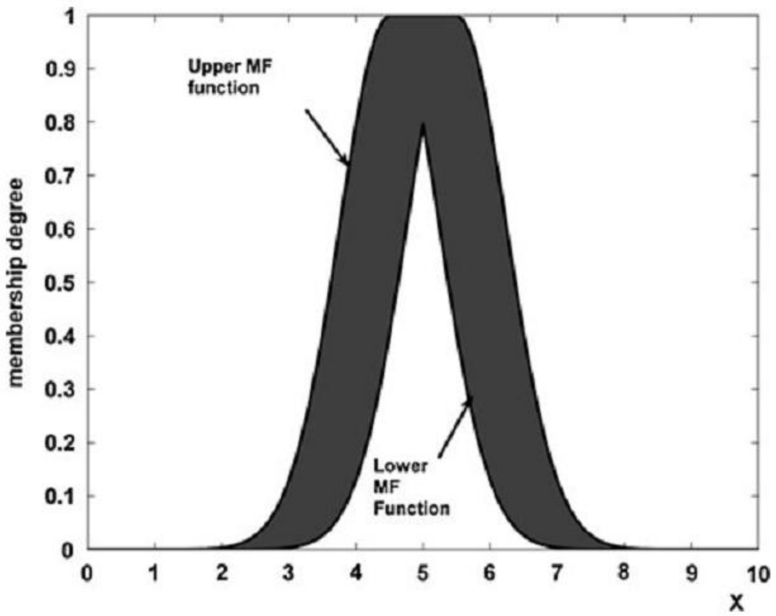


Figure 12. UMF & LMF that construct FOU

[8] gives more detailed explanation and definition of type 2 fuzzy concepts and operations. Because of unknown distribution of *secondary membership grade,* all of these grades are considered equal to 1. This type 2 fuzzy set with unity secondary membership grade is named *interval type 2 fuzzy sets*. According to [8], interval type 2 fuzzy sets prevail thanks to its simplicity and the incapability of proving any better candidate.

Due to the complicated nature of type 2 fuzzy sets, the work flow for type 2 fuzzy inference system is upgraded. The rule base is the same, but how rules manipulate fuzzy sets is different, not as simple as *intersection*, *union* or *implementation;* in type 2 fuzzy logic, these operations are *meet*, *join* and *negation.* The corresponding inference system for T2FL is:

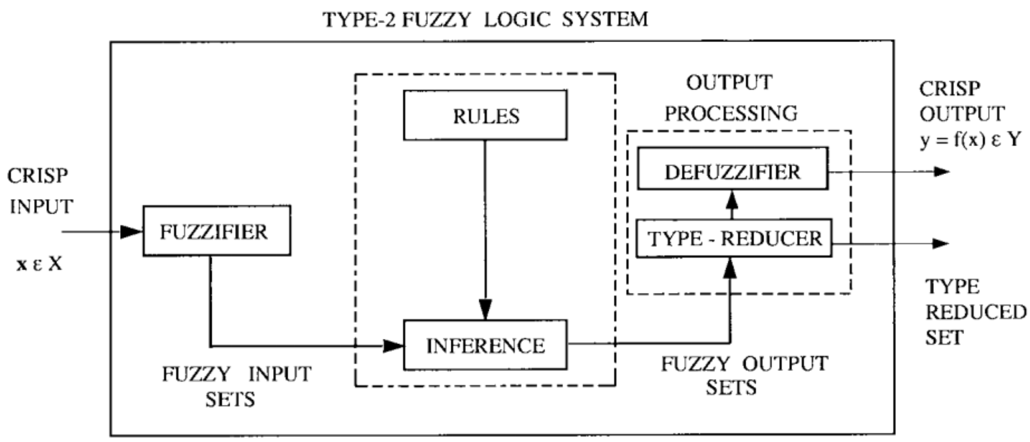


Figure 13. Type 2 Fuzzy Logic Inference System

Compared to type 1 inference system, type 2 has 1 more block that is "type reducer". The role of this block is to convert a type 2 fuzzy set to a simple type 1 set so that defuzzifier can transform it to a crisp output for further application. Therefore, not only complex operations on IF-THEN rules, to gain successful function of type 2 fuzzy system, there must be an effective method for type reduction and defuzzification. [16] presented some methods of type reduction that are: centroid, center-of-sums, height and center-of-sets. [17] suggested enhancement for work [16].

Continue the work of type reducer is defuzzifier. Besides Karnik-Mendel Iterative Procedure (KMIP) defuzzification method introduced in [17], recently there is Sampling Method [18] that claimed to have faster processing with negligible error.

2.3.4.2 Interval Type-2 Fuzzy Logic Tuned PID Controller

[19] and [20] show that IT2FL implementation in improving PID controller is similar to T1FL in terms of inputs and output as follow:

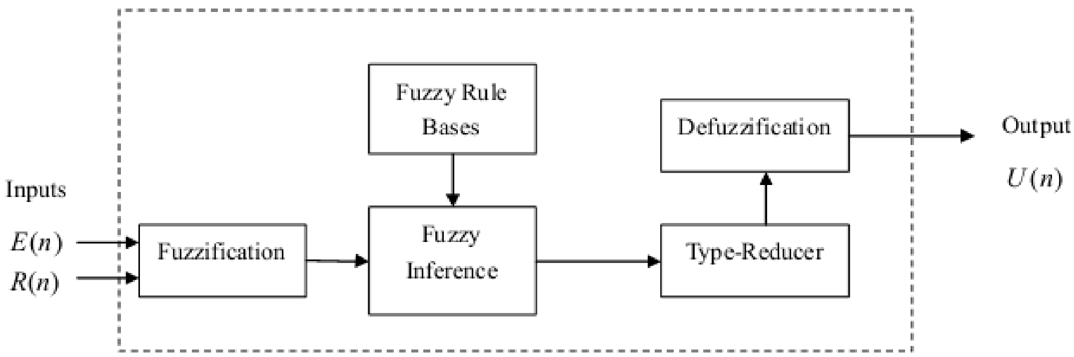


Figure 14. Fuzzy Logic Controller Structure

So for T2FL, the proposed system also has 2 inputs as error and error change rate, with 1 output as supplementary manipulated variable. The governing rules are also identical. The difference here is composition of input signals which are now type 2 fuzzy sets and how they are processed in inference system. Theoretically, T2FL can help to improve performance in case of inaccurate measurement of error or imprecise estimation of error change rate, or frequent disturbance that makes error vary.

2.4 Related Work of Auto Tuned PID Controller

Besides application of fuzzy logic PID controller, there have been several different methods that aim for the same purpose. Table 1 shows some of them with basic information:

Year	Author	Method	Limitation
2012	Bindu R. & Mini K. Namboothiripad	Genetic Algorithm [1]	- Simulation only - No noise in feedback
2012	Al-Ubaidi S. M. Z. & Algreer M. M. F.	Type-1 Fuzzy PD Controller [21]	- No integral leads to offset
2008	Wahyunggoro, O. Saad, N.	Type-1 Fuzzy Self Tuning [22]	- Slow settling time, high overshoot during load change
1992	Yanagawa, S. Miki, I.	Neural networks with single neuron [23]	- Speed drop in step response in simulation.
2007	Susperregui, A. Tapia, G. Tapia, A.	First-order sliding-mode control [24]	- Significant chattering, excite high-frequency unmodeled dynamics

2004	Bottura, C. P. Serra, G. Ld O.	Adaptive control scheme [25]	- Slow and noisy step response
2011	Rios-Gutierrez, F. Makableh, Y. F.	Intelligent Neural Network [26]	- Small offset.

Table 1. Related works summary

Chapter 3: Project Planning & Methodology

3.1 Project Planning

3.1.1 Project Plan

This project encompasses 4 phases: the first phase is to model DC servo motor and simulate it with conventional PID Controller in Simulink, phase 2 includes simulation and evaluation of T1FL tuned PID controller, phase 3 is for IT2FL tuned PID controller coding and simulation in Matlab and Simulink, phase 4 is comparison between type 1 and type 2 fuzzy controllers. Lastly, a graphic user interface (GUI) is developed to offer visual observation and ease of defining and optimizing type-2 fuzzy system output.

For the first phase, research on DC servo motor model and simulation of the model is attempted in Simulink. The next step is to simulate and tune the conventional PID controller for the DC servo motor. Several parameters must be recorded for analysis later such as settling time, rise time, and overshoot. The PID controller should be tuned by Matlab GUI tuning tool to select the best performance so that it can be used later for evaluate fuzzy logic controller.

In third phase, the original PID controller is supplemented with type-1 fuzzy PID controller. This phase requires the use of Fuzzy Logic Tool Box that is embedded in Matlab. The inputs, output(s) and rules should be optimally defined to give best performance. The inference system would be Mamdani instead of Takagi-Sugeno-Kang. The T1FL PID controller performance is also recorded for comparative analysis later on.

The fourth phase demands intensive research to comprehend main concepts, definitions and operations of type-2 fuzzy logic and prepare for next step which is programming. After this step there should be strong foundation of knowledge about type 2 fuzzy sets definition, type 2 fuzzy sets operations, type 2 fuzzification methods, type 2 fuzzy inference system, type reduction and defuzzification methods. This knowledge is then translated into Matlab codes so that it can be used later in controller simulation. Remaining parts are like in third phase which is simulation and record the performance for comparison between typical PID, T1FL and IT2FL tuned PID controllers. There should be some variation in type 2 fuzzy model such as changing number of membership functions or types of membership functions or method of defuzzification so that the best result is not missed.

Finally, interval type-2 fuzzy inference system is integrated to a user interface for better observation and easier manipulation. This aims to provide an efficient yet user friendly tool for interaction with the simulated controller and prove that the proposed model has the potential to be generally applicable, not just limited for DC servo motor.

3.1.2 Tools & Softwares

- DC Brushless Servo motor specifications
- Matlab & Simulink
- Type 1 Fuzzy Logic Toolbox (embedded in Matlab)

- Microsoft Office 2010 for data analysis and illustration

3.1.3 Flow Chart

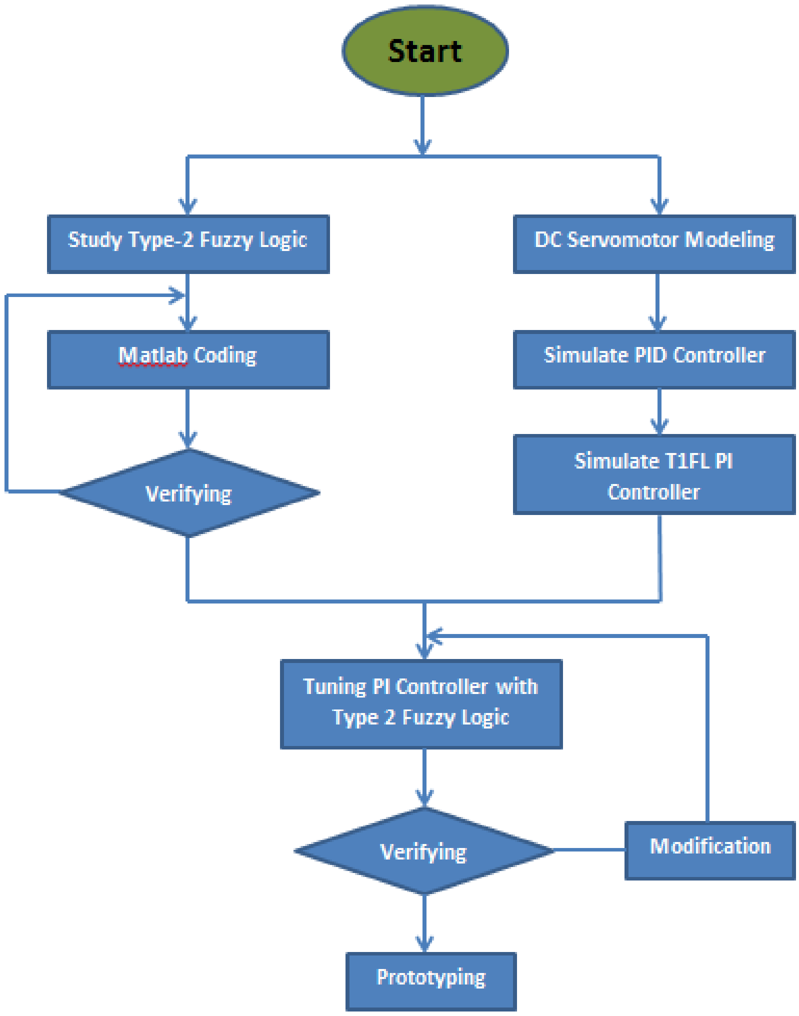


Figure 15. Work Flow Chart

3.1.4 Gantt Chart

First Semester Gantt Chart (FYP 1):

Task/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Project Conceptualization														
Literature Review														
Extended Proposal						*								
DC Servomotor Modeling														
PID Controller for DC motor simulation														
Proposal Defence									*					
T1FL PI Controller Simulation														
Matlab coding for T2FL sets operation														
Interim Report														

Legend

Completed Work | Future Work | Deadline *

Second Semester Gantt Chart (FYP 2):

Task/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Matlab coding for T2FL interference system														
T2FL PI Controller Simulation														
Progress Report						*								
Comparative Analysis of Results														
Controller Optimization														
Interface to Digital Board														
Prototyping														
Submission of Technical Paper												*		
Prototype Modification														
Submission of Dissertation													*	
Viva														*

Legend

Completed Work | Future Work | Deadline *

3.2 Methodology

3.2.1 DC servo motor System Simulation

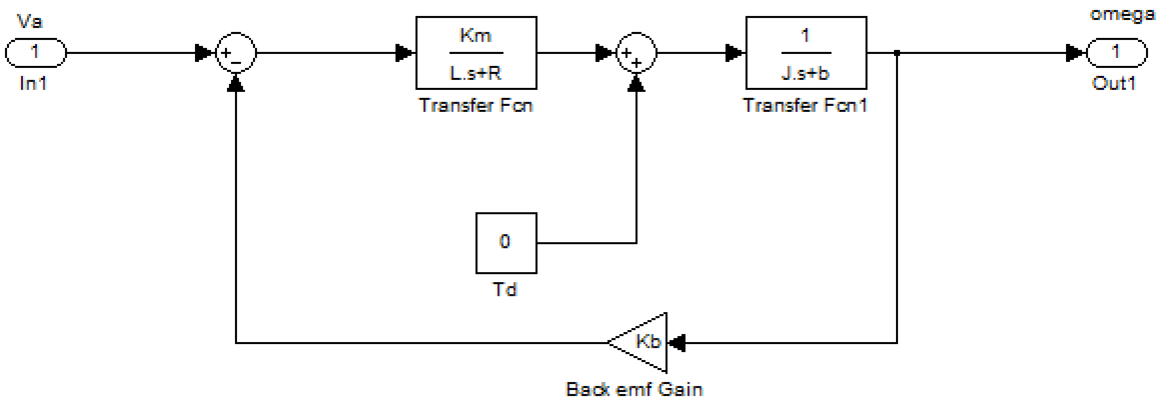


Figure 16. DC Servo motor System Simulation

Figure 16 shows the simulated DC servo motor system that will be used throughout the project with V_a as the input voltage and ω (omega) is the output angular velocity, T_d is the load disturbance which is either zero or varied values based on the simulation condition, in Figure 16 T_d=0. The parameters used are:

J = 0.01 kg.m^2
b = 0.1 N.m.s
K_b = 0.01 V/rad/s
K_m = 0.01 N.m/A
R = 1.0 Ω
L = 0.5 H

3.2.2 PID Controller for DC Servo motor Simulation

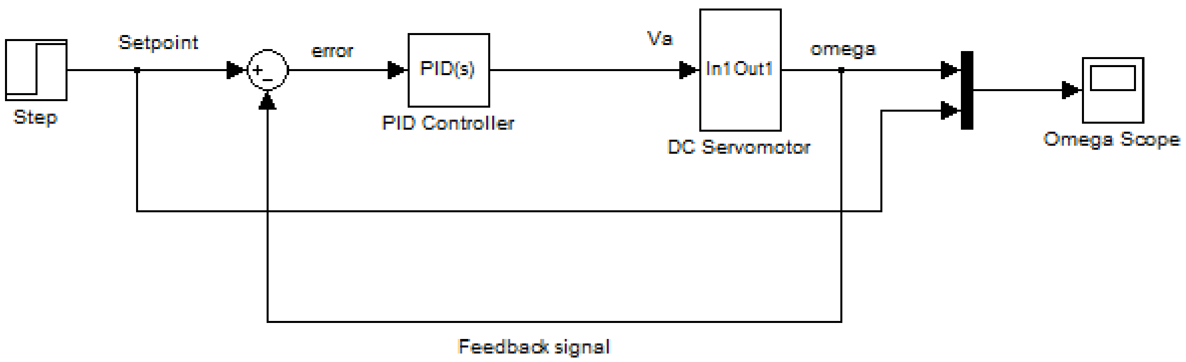


Figure 17. PID Controller for DC Servo motor

Figure 17 displays the simulated PID controller with the subsystem name 'DC Servo motor', this subsystem consists all block in Figure 16. PID Controller block is taken from Simulink library. Input of PID Controller is error between set point and output; and the output of PID controller is V_a which regulates the angular velocity ω. All the working parameters for PID controller is set and optimized by PID tuning tool available in Simulink. Set point change (step change) and output ω are recorded in ‘Omega Scope’ block to analyze the performance.

The PID controller is tuned with Matlab GUI tuning tool under unit step change condition as shown below:

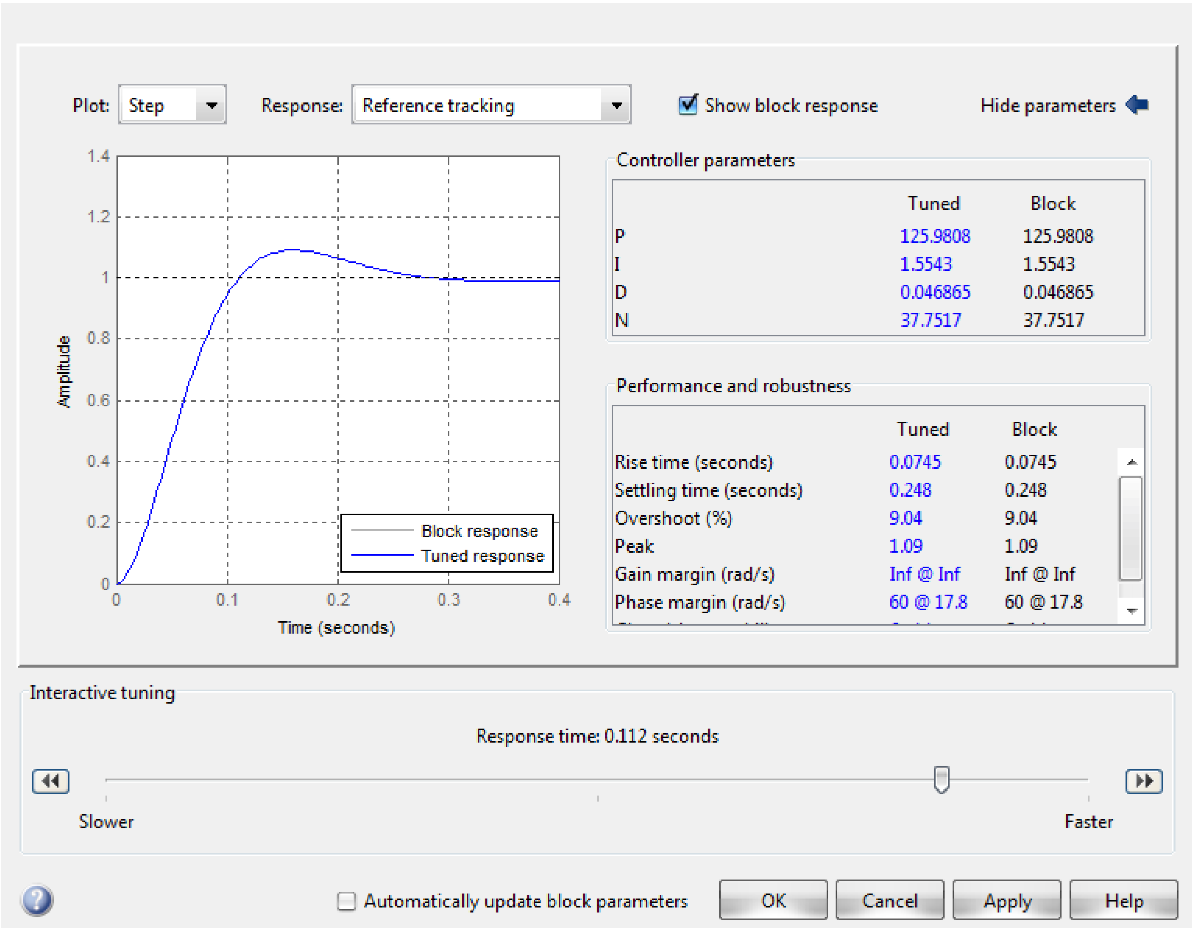


Figure 18. PID Tuning

Figure 18 shows the best response from PID tuning with rise time: 0.0745(s), settling time: 0.248(s), overshoot: 9.04%, K_p= 125.980844549916, K_i = 1.55425557500709 and K_d = 0.0468651166018972.

3.2.3 T1FL PID Controller for DC Servo motor Simulation

Based on discrete model of T1FL PID Controller described in [15] and shown in Figure 9, a continuous time model is constructed in Simulink as follow:

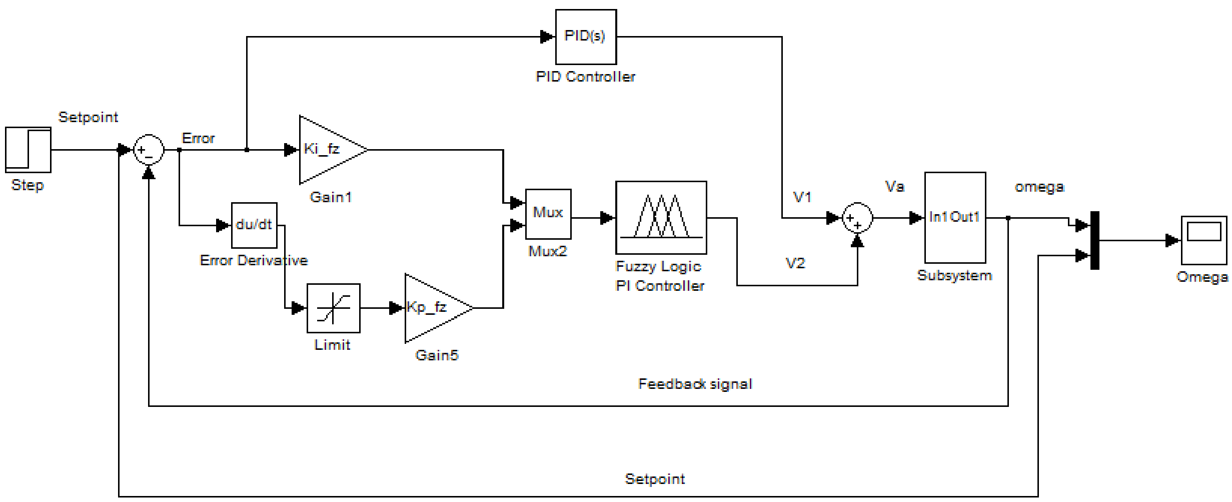


Figure 19. T1FL Tuned PID Controller

In comparison with conventional PID controller in Figure 17, Figure 19 has additional blocks:

- 'Gain1' and 'Gain5' are the gains derived from K_p and K_i respectively. They are the reason of the name "T1FL PID Controller".

- 'Error Derivative' block is used to take the change rate of the error, this rate may be very large, hence it needs to be followed by a limit block to restrict the value.

- 'Fuzzy Logic PID Controller' is Fuzzy Logic Toolbox provided by Matlab. It processes error and derivative of error to produce supplemental signal V2 for corrective action, as in Va = V1 + V2, where V1 is the output of conventional PID controller.

T1FL is configured to apply 3 MFs to each of input variable and these MFs are triangular and trapezoidal functions. The rules of fuzzy logic inference system to determine supplementing corrective signal are:

DE \ E	n	o	p
n	**n**	**n**	**o**
o	**n**	**o**	**p**
p	**o**	**p**	**p**

Table 2. T1FL Rules for PID Controller

Where:

E is the error, *DE* is the change rate of error (derivative of error), *n* is negative, *o* is zero and *p* is positive.

3.2.4 IT2FL PID Controller for DC Servo motor Simulation

The block diagram of IT2FL PID controller is almost identical with T1FL one, only the fuzzy logic block is now type 2 fuzzy inference system:

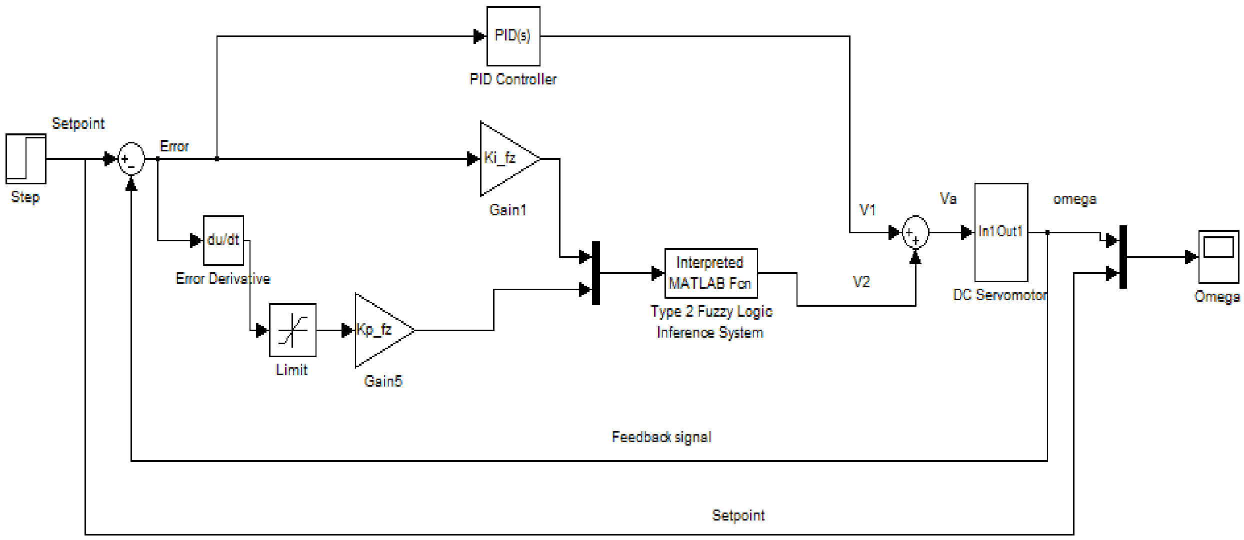
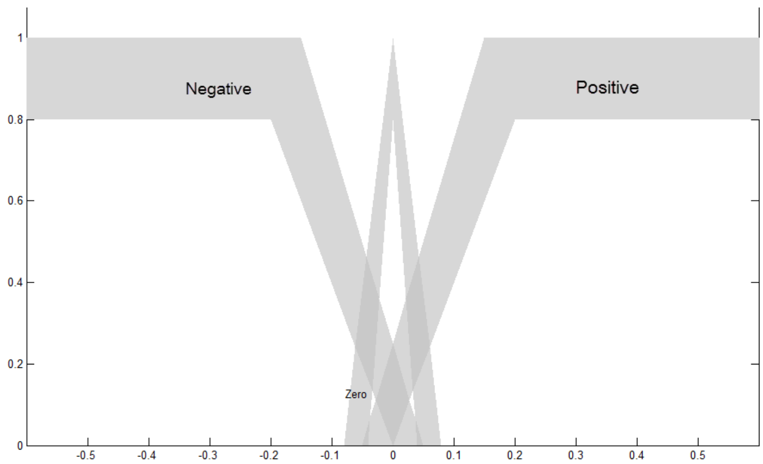
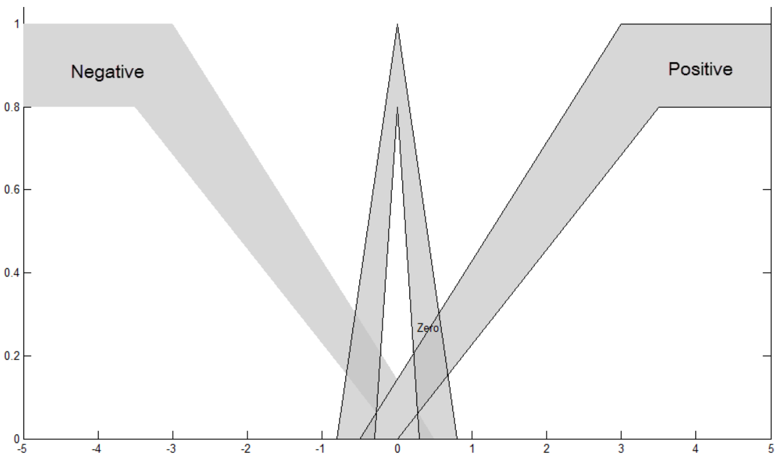


Figure 20. Interval T2FL Tuned PID Controller

Compared to type 1 fuzzy logic, input fuzzy sets (error and error derivative) are different as they are interval type-2 fuzzy sets with secondary membership function, and are presented with footprint of uncertainty (FOU) as follow:



a)



b)

Figure 21. Type 2 fuzzy set representation of a) Error; b) Derivative of Error

Figure 21 shows the membership functions that were used for this particular research, however, the Matlab coding and simulation is flexible to handle any membership function that is defined by 9 points as shown below:

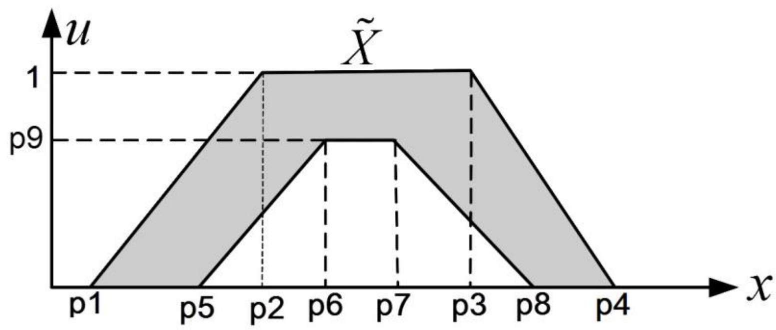


Figure 22. Membership Function defined by 9 points

By changing the values of these points, the interval type-2 fuzzy inference system can be optimized easily for expected response. The inputs of membership function are expected to be presented in form of a user interface for friendly interaction.

As refer to literature review, the method for type reduction in this project is chosen as *center-of-set* method. This method replaces the rule consequent which normally is a type 2 fuzzy set with a singleton at its centroid, this results a type 1 fuzzy set comprised of these singletons. This type 1 fuzzy centroid is represented as an interval of output value. Then find the weighted average of these type 1 fuzzy sets in which the weight of the n^{th} consequent is the firing degree of the n^{th} rule [27].

It can be presented as:

$$Y_{cos}(\mathbf{x}') = \bigcup_{\substack{f^n \in F^n(\mathbf{x}') \\ y^n \in Y^n}} \frac{\sum_{n=1}^{N} f^n y^n}{\sum_{n=1}^{N} f^n} = [y_l,\ y_r]$$

where:

$Y_{cos}(\mathbf{x}')$ is the consequent type 1 fuzzy set.

N is the total number of rules.

n is the n^{th} rule.

y^n is the consequence of n^{th} rule.

f^n is the firing degree of n^{th} rule.

Finding y_l and y_r is a complicated task, where:

$$y_l = \min_{\substack{\forall x_i \in [\underline{x}_i, \overline{x}_i] \\ \forall w_i \in [\underline{w}_i, \overline{w}_i]}} \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i}$$

$$y_r = \max_{\substack{\forall x_i \in [\underline{x}_i, \overline{x}_i] \\ \forall w_i \in [\underline{w}_i, \overline{w}_i]}} \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i}$$

There are several different algorithms to complete this task, and among them *enhanced iterative algorithm with stop condition (EIASC)* is an outstanding solution as it reduces number iterative computations and saves more processing time [17]. Details of implementation of the algorithm can be referred to the source.

The rule table for interval type 2 fuzzy inference system of this particular project is:

DE \ E	n	o	p
n	**Y^1=[-300, -200]**	**Y^2=[-100, 0]**	**Y^3=[-100, 100]**
o	**Y^4=[-100, 0]**	**Y^5=[-50, 50]**	**Y^6=[0, 100]**
p	**Y^7=[-100, 100]**	**Y^8=[0, 100]**	**Y^9=[200, 300]**

Table 3. Type 2 fuzzy logic rules table

3.2.5 Simulation Testing Conditions & Criteria

The simulated test is conducted with unit step change at time t=0.2 and lasts for 10 second, under 2 variations: with and without load disturbance; with and without noise. Refer to Figure 3 for DC servo motor block diagram with load disturbance. The load disturbance T_d is simulated with a Signal Builder block as follow:

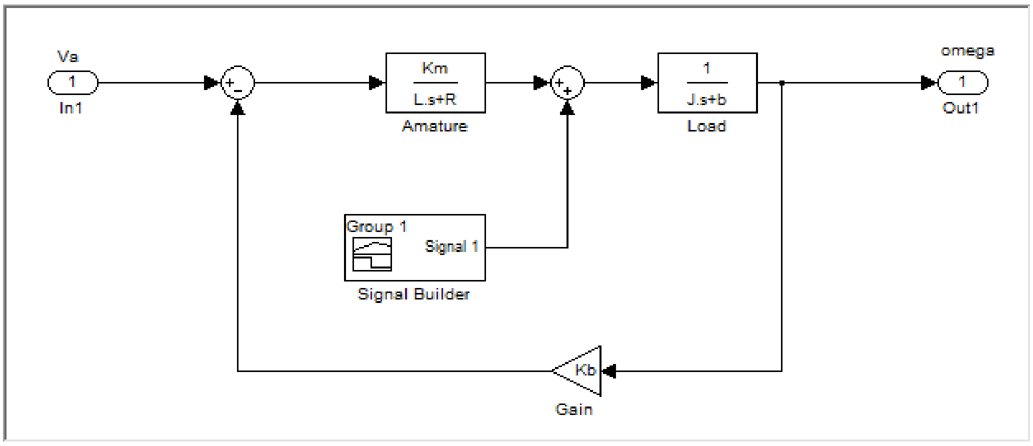


Figure 23. DC Servo motor subsystem with load disturbance

The load disturbance takes change at time t_1=0, t_2=3, t_3=6 and the values are -0.02, 0.04 and -0.04. The waveform of load disturbance is shown below:

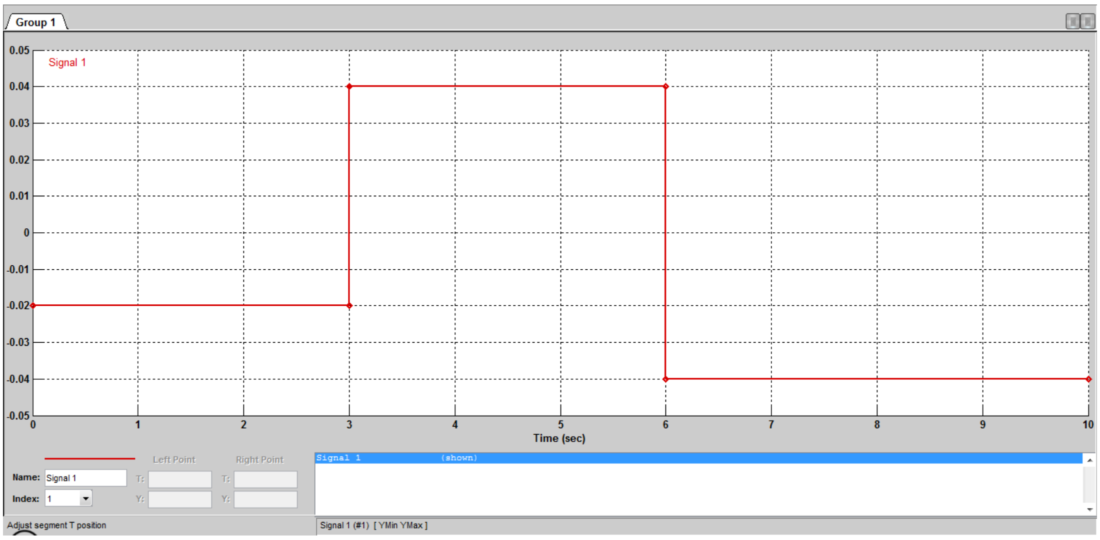


Figure 24. Load disturbance waveform

The scenario for choosing such load disturbance is to imitate the movement of a robot arm: it starts at idle position, then it picks 1 object up, hence the load increases, finally it drops the object, making the load decreases.

The noise for testing is uniformly distributed at different magnitude limit as 0.1, 0.15 and 0.2. That means the maximum noise is 20% of testing step input. The noise is added to the error input signal (feedback signal) of the system. Noise is generated by Uniform Noise Generator block in Simulink.

All 3 systems: conventional PID controller, type 1 fuzzy tuned controller and interval type 2 fuzzy tuned controller will be tested under the same load and noise signals and the responses are recorded in same graph for analysis. It is supposed that interval type 2 fuzzy logic would handle noise better that the other two.

The main criteria for assessment are: IAE (integral absolute error), ITAE (integral time-weighted absolute error), overshoot and rise time. These values will be tabulated for easiness of analysis.

Chapter 4: Result & Discussion

For all the results in graph, the responses' colors are as follow:

- PID controller
- Type 1 fuzzy tuned PID
- Interval type 2 fuzzy tuned PID

The test is divided into 2 subtests: first is no varying load to DC servo motor, and second is varying load with description as in Methodology. Under each subtest, it is detailed further to 3 different magnitudes of noise in error signal, those are: 0.1, 0.15 and 0.2.

4.1 No varying load to the DC servo motor

Without any change in motor load, the noise absolute magnitude to be added in error signal is consequently 0.1; 0.15 and 0.2. That means the maximum error noise is 20% of step change.

Under 0.1 noise magnitude without varying load, the responses are:

	PID	Type 1 fuzzy tuned PID	Interval type 2 fuzzy tuned PID
IAE	**0.1277**	**0.1414**	**0.107**
ITAE	**0.0006467**	**0.000715**	**0.0005432**
Overshoot	**11%**	**4%**	**2.5%**
Rising time	**0.109**	**0.105**	**0.103**

Table 4. Result with no varying load, noise is 0.1

Under 0.15 noise magnitude, the responses are:

	PID	Type 1 fuzzy tuned PID	Interval type 2 fuzzy tuned PID
IAE	**0.1601**	**0.1794**	**0.1298**
ITAE	**0.0008084**	**0.000905**	**0.0006572**
Overshoot	**12%**	**5.7%**	**3.7%**

Rising time	**0.109**	**0.101**	**0.102**

Table 5. Result with no varying load, noise is 0.15

Under 0.2 noise magnitude, the responses are:

	PID	Type 1 fuzzy tuned PID	Interval type 2 fuzzy tuned PID
IAE	**0.1927**	**0.2044**	**0.1411**
ITAE	**0.0009712**	**0.00103**	**0.0007136**
Overshoot	**12.8%**	**4.3%**	**2.2%**
Rising time (s)	**0.108**	**0.109**	**0.106**

Table 6. Result with no varying load, noise is 0.2

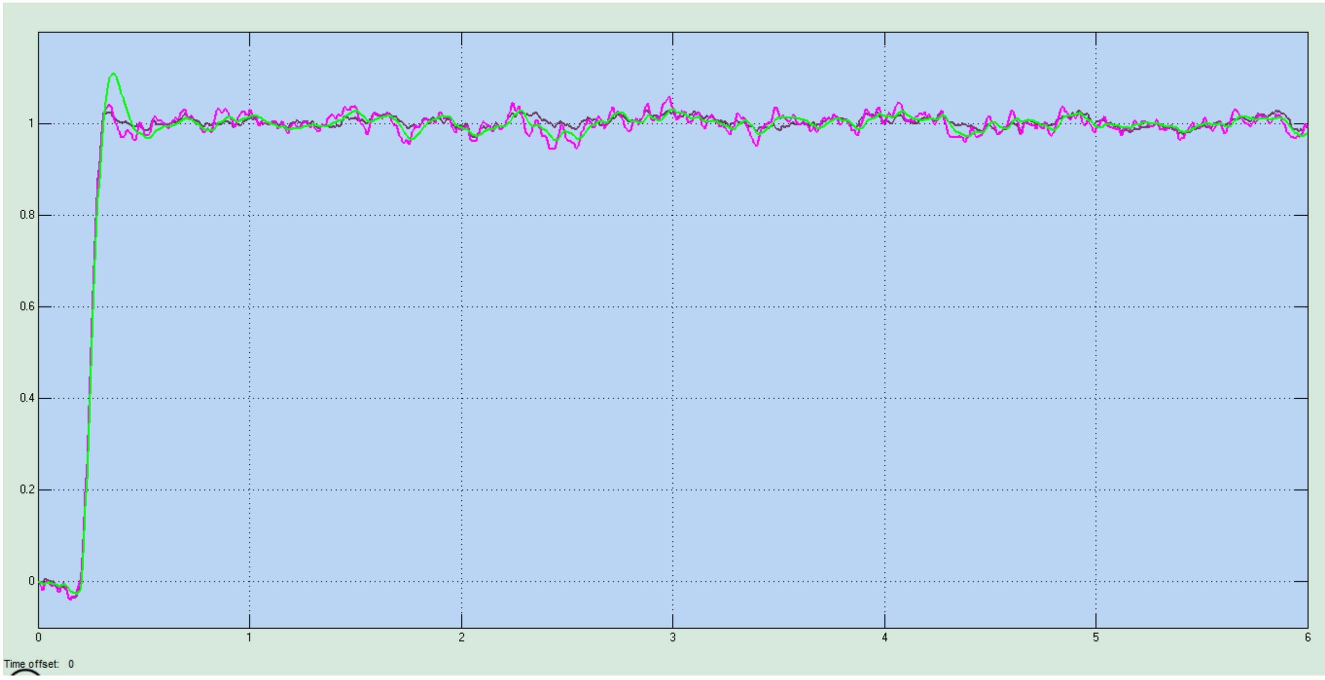


Figure 25. Step responses without varying load, noise is 0.1

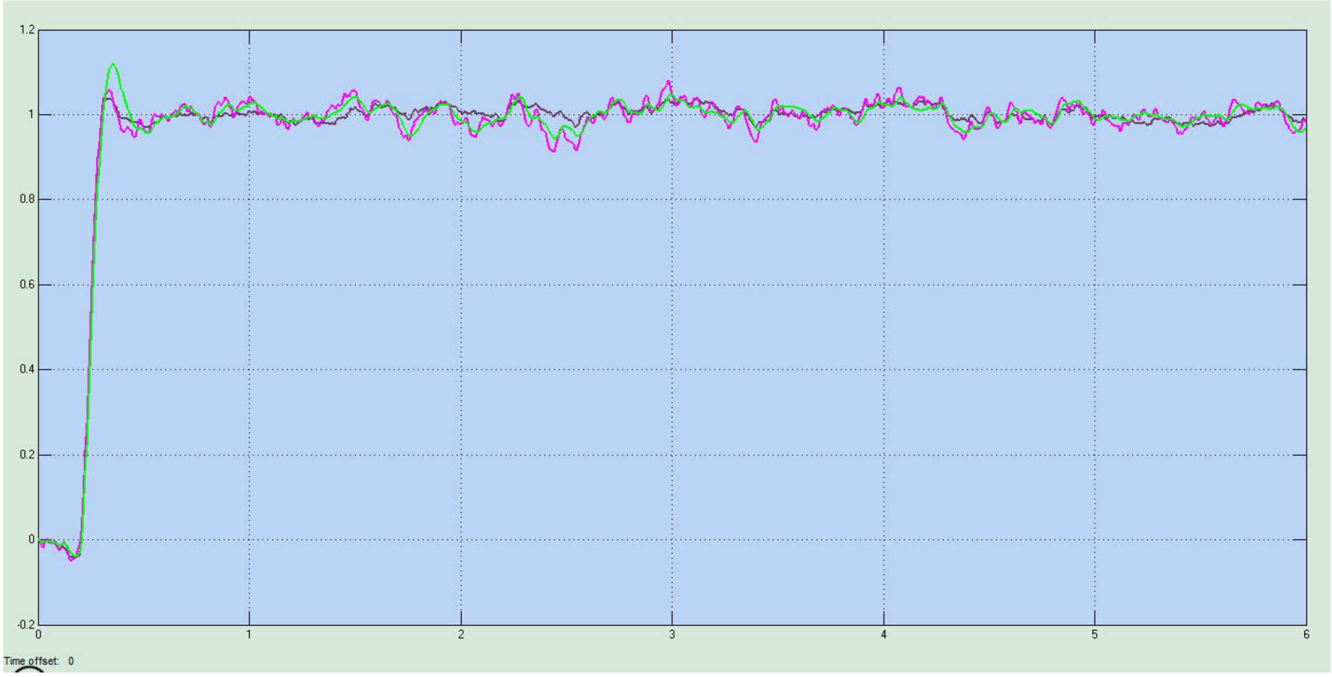


Figure 26. Step responses without varying load, noise is 0.15

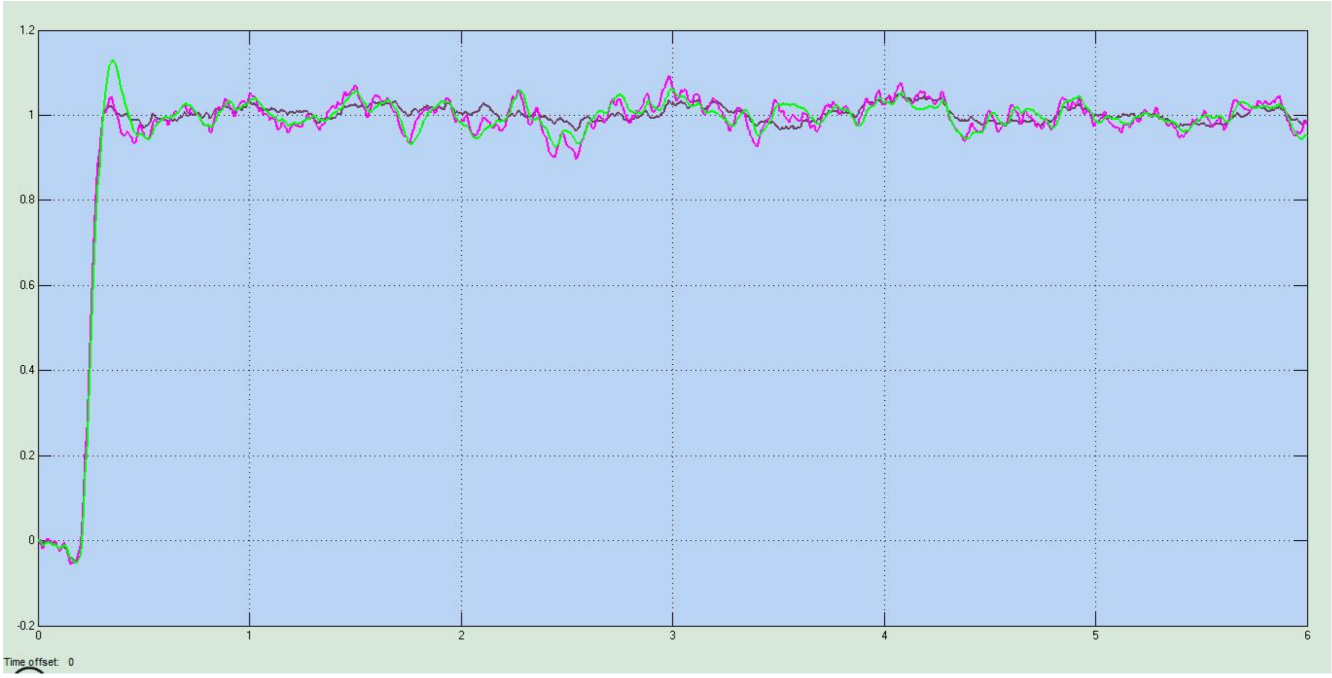


Figure 27. Step responses without varying load, noise is 0.20

From Table 4, Table 5 and Table 6, the chart of average overshoot with different noise magnitude under no varying load is derived:

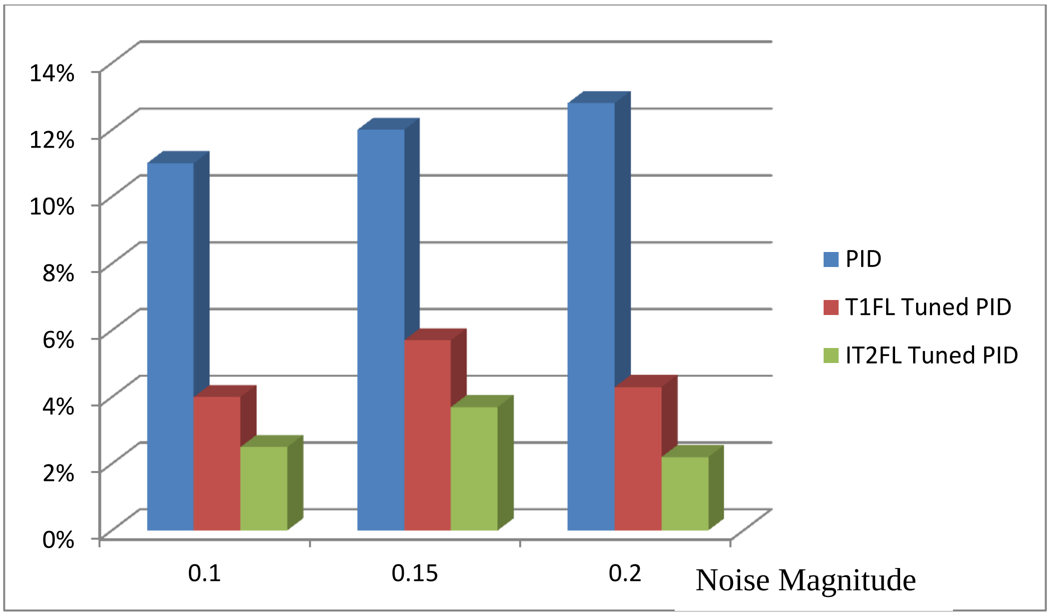


Figure 28. Average Overshoot without Varying Load

From the chart in Figure 28, it is observed that fuzzy logic tuned PID controllers provide significant lower overshoot compared to conventional PID controller. And it is clear that IT2FL tuned PID controller gives smallest overshoot in both conditions of noise magnitude. Therefore, the application of fuzzy tuned PID controller is very promising for highly precise or delicate final elements which are sensitive to overshoot value.

From Table 5 and Table 6, it is observed that in terms of IAE, ITAE and rise time, conventional PID and type-1 fuzzy tuned PID swap their position of better solution. In other words, several times conventional PID gets better (smaller) values, and the rest type-1 fuzzy tuned PID is better. So, besides overshoot value, type-1 fuzzy tuned PID controller is not absolutely justified to be better than conventional PID controller.

However, from comparative observation of Table 4, 5 and 6, it can be concluded that interval type 2 fuzzy tuned PID offers the smallest IAE and IATE, smallest overshoot and also smallest rise time in all cases. This implies that the IT2FL tuned PID controller is overwhelming the other two controllers. Thus, the conclusion for testing without varying load under different noise disturbances is that IT2FL tuned PID controller is the best controller.

4.2 With varying load to the DC servo motor

The simulation is carried out for 10 seconds with DC servo motor initial load is different than zero and load changes at t_1=3s and t_2=6s. The noise to error signal is the same with previous test as consequent magnitudes are 0.1; 0.15 and 0.2.

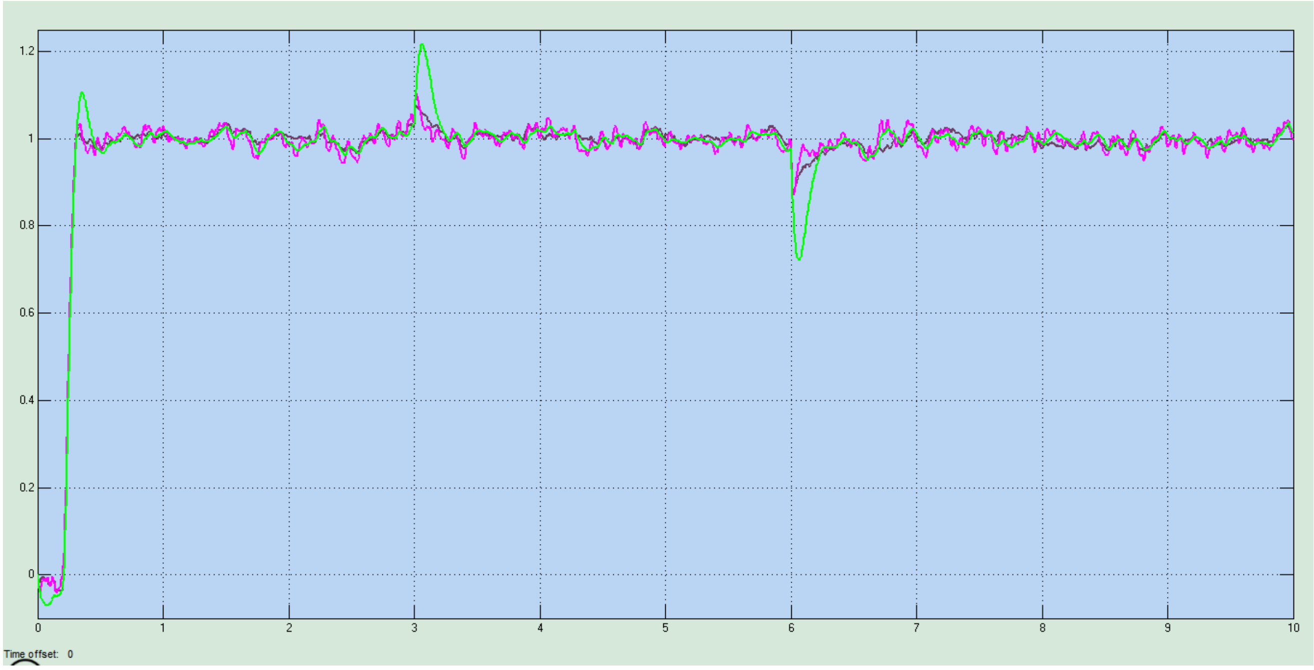


Figure 30. Step responses with varying load, noise is 0.1

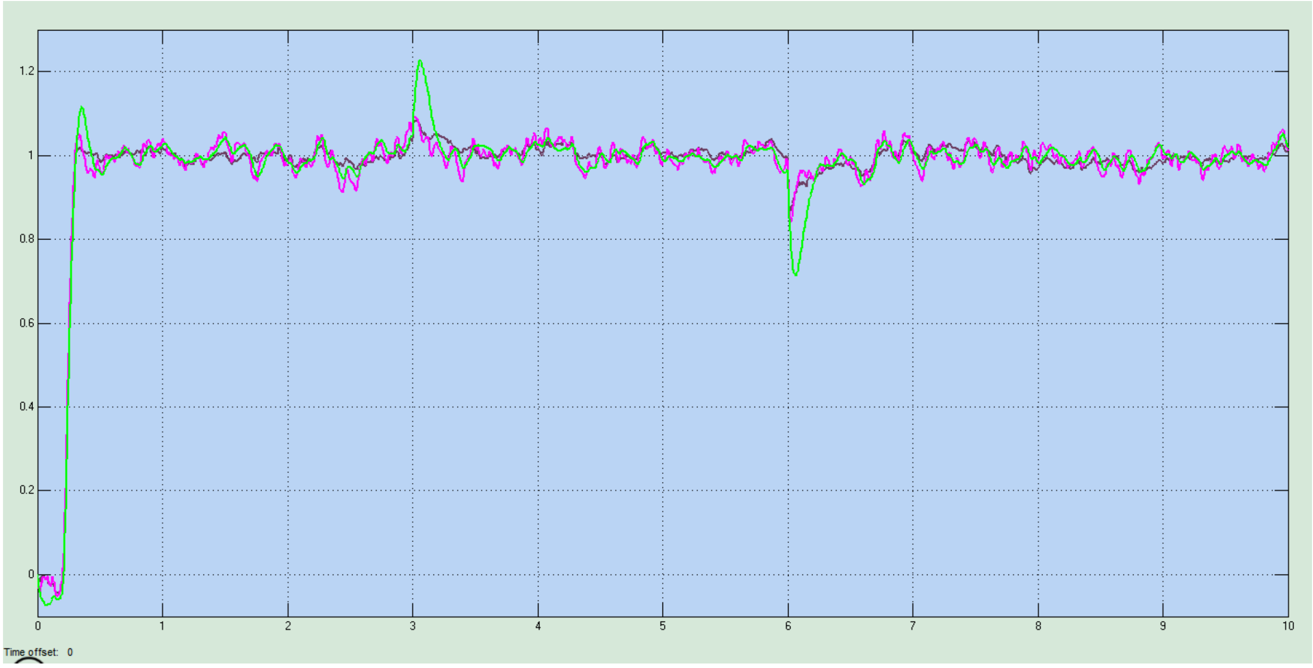


Figure 29. Step responses with varying load, noise is 0.15

Figure 31. Step responses with varying load, noise is 0.2

Figure 29, Figure 30 and Figure 31 show the response signals under 0.1, 0.15 and 0.2 noise magnitude with varying load. It is observed that at t_1=3s and t_2=6s, when the load changes, there are overshoots that are significant and can be used at an additional factor to assess performance of controllers. Thus, in these following tables, new criteria as Overshoot 2 and Overshoot 3 is added to compare controllers' performance:

Under 0.1 noise magnitude with varying load, the responses are:

	PID	Type 1 fuzzy tuned PID	Interval type 2 fuzzy tuned PID
IAE	**0.2437**	**0.2172**	**0.1806**
ITAE	**0.001226**	**0.001094**	**0.0009109**
Overshoot 1	**10.7%**	**3.6%**	**1.3%**
Overshoot 2	**22%**	**10.2%**	**7.5%**
Overshoot 3	**28%**	**12.9%**	**12.2%**
1^{st} Rising time (s)	**0.101**	**0.1073**	**0.1085**

Table 7. Result with varying load, noise is 0.1

Under 0.15 noise magnitude with varying load, the responses are:

	PID	Type 1 fuzzy tuned PID	Interval type 2 fuzzy tuned PID
IAE	**0.297**	**0.2785**	**0.2137**
ITAE	**0.001493**	**0.0014**	**0.001076**
Overshoot 1	**11.7%**	**5%**	**1.8%**
Overshoot 2	**22.8%**	**9.25%**	**8.35%**
Overshoot 3	**28.6%**	**16%**	**13.2%**
1st Rising time (s)	**0.109**	**0.103**	**0.106**

Table 8. Result with varying load, noise is 0.15

Under 0.2 noise magnitude with varying load, the responses are:

	PID	Type 1 fuzzy tuned PID	Interval type 2 fuzzy tuned PID
IAE	**0.3508**	**0.3207**	**0.2312**
ITAE	**0.001762**	**0.001611**	**0.001164**
Overshoot 1	**12.8%**	**4%**	**2.5%**
Overshoot 2	**23.8%**	**11%**	**9.7%**
Overshoot 3	**29.3%**	**16.5%**	**14.5%**
1st Rising time (s)	**0.108**	**0.104**	**0.105**

Table 9. Result with varying load, noise is 0.2

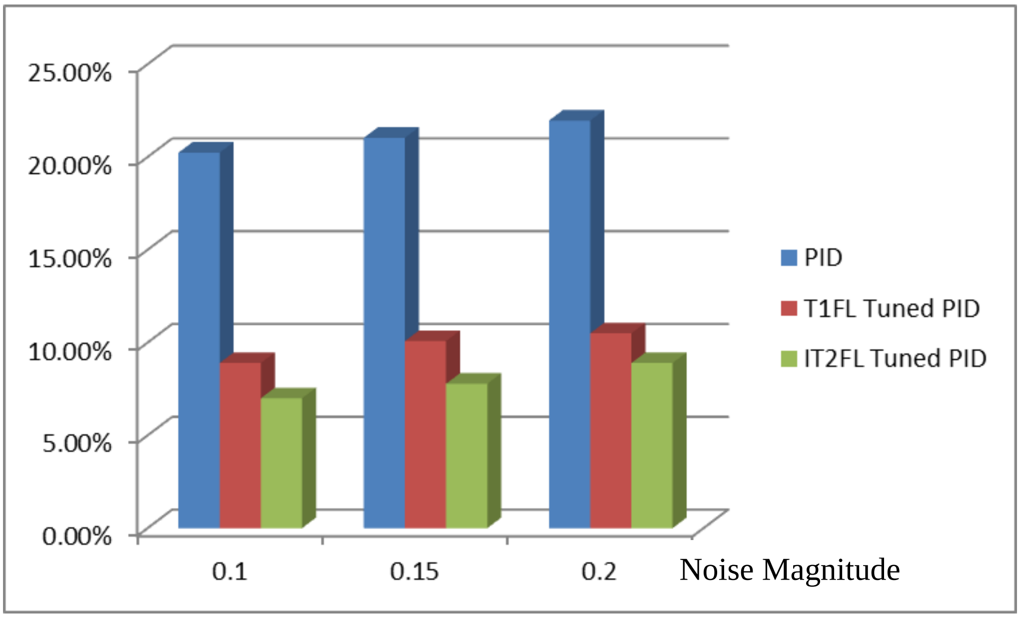


Figure 32. Average Overshoot in Varying Load Condition

Again, when testing with new condition of changing load, chart for average overshoot under 0.1 and 0.15 noise magnitudes consolidates the finding of previous section that fuzzy tuned PID controllers offer considerable smaller overshoot. And among these controllers, IT2FL tuned PID again shows the smallest overshoot in different conditions.

Now that both varying load and no varying load testing were performed, it is possible to compare the performance of controllers under these conditions. Choose arbitrarily the average IAE as a criterion to assess, the result is:

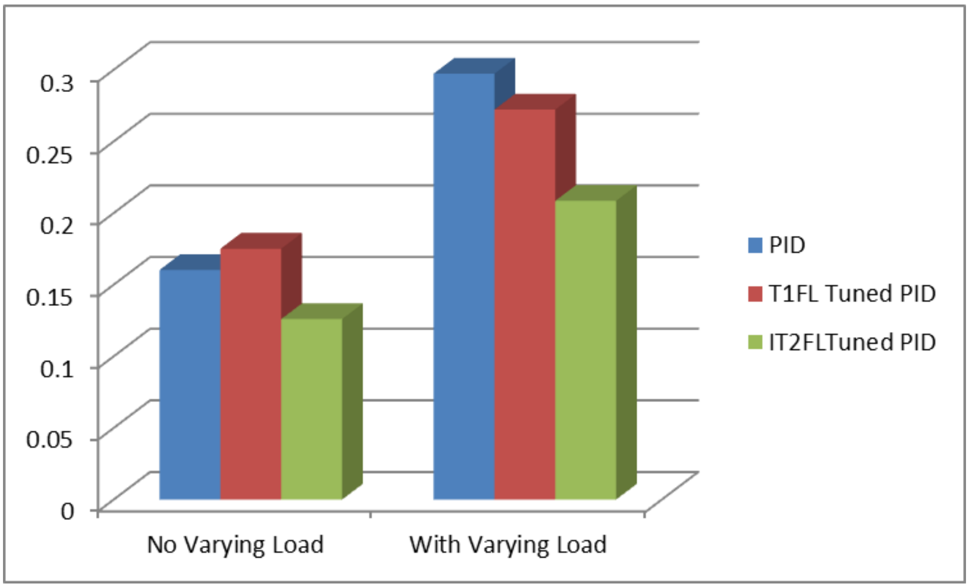


Figure 33. Average Integral Absolute Error w/wo Varying Load

In Figure 33, it is clear that under no varying load, IAE of T1FL tuned PID is higher than IAE of conventional PID while in condition of varying load, it is vice versa. However, in both cases, IT2FL tuned PID always offers the smallest IAE, thus it claims the merit to be the best controller.

4.3 Graphical User Interface

For the purpose of providing ease in interaction and manipulation of IT2FL inference system, a graphical user interface was built. This user interface is divided into 2 sub-interfaces. The first one is the Setting Window, where instruction and tools for input values and visual observation are provided as follow:

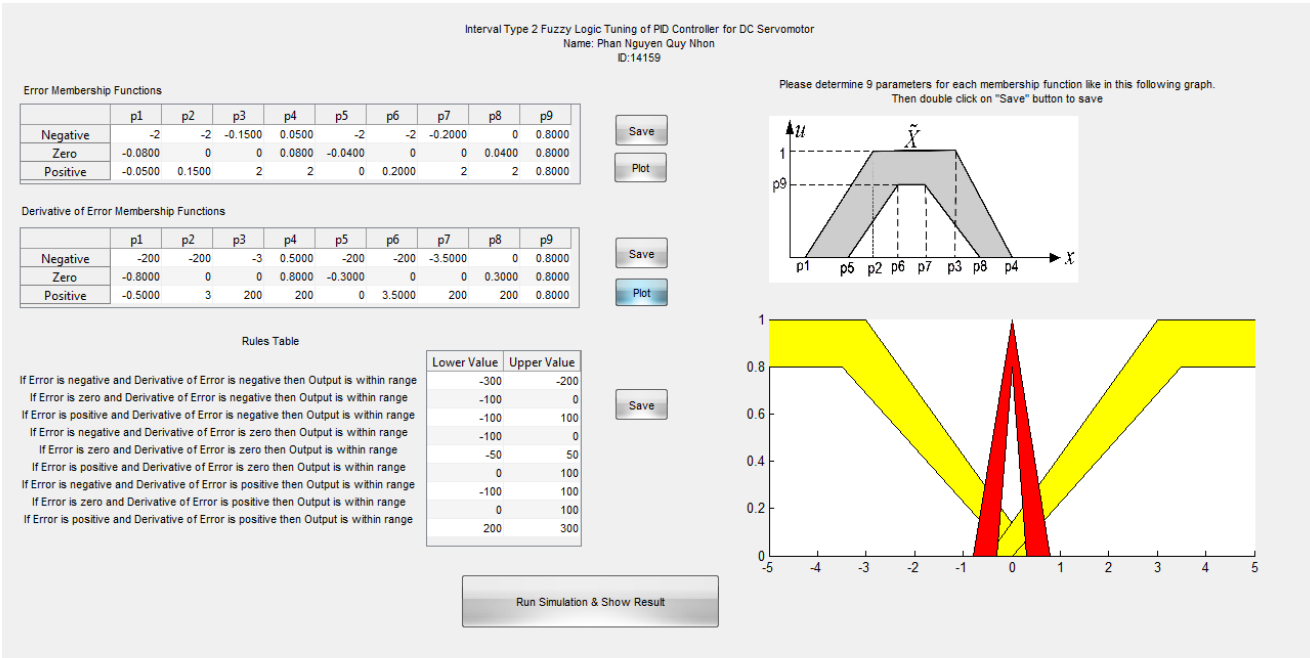
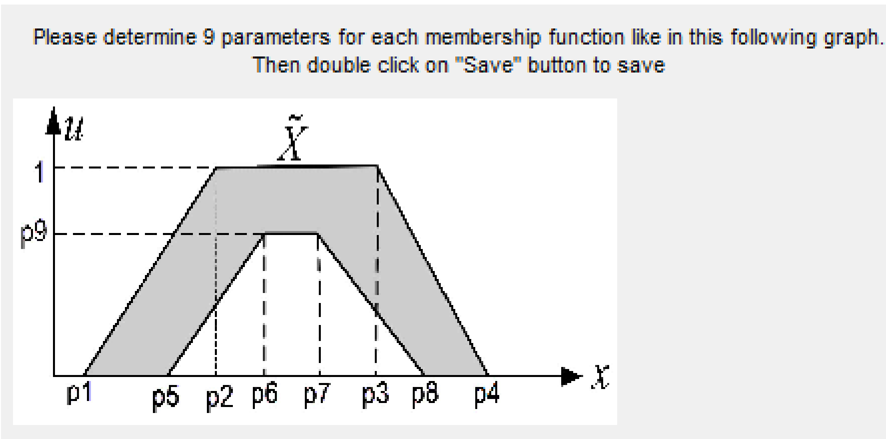


Figure 34. Setting Window user interface

The instruction part gives information of 9-points membership function definition:



On the top left of Setting Window is 2 tables to define Error and Derivative of Error fuzzy sets. Each set comprises of 3 membership functions as Negative, Zero and Positive. The program is capable of handling higher number of membership functions; however the number is fixed to 3 membership functions for this project. User can key in value for 9 points of membership function as shown:

Error Membership Functions

	p1	p2	p3	p4	p5	p6	p7	p8	p9
Negative	-2	-2	-0.1500	0.0500	-2	-2	-0.2000	0	0.8000
Zero	-0.0800	0	0	0.0800	-0.0400	0	0	0.0400	0.8000
Positive	-0.0500	0.1500	2	2	0	0.2000	2	2	0.8000

Save

Plot

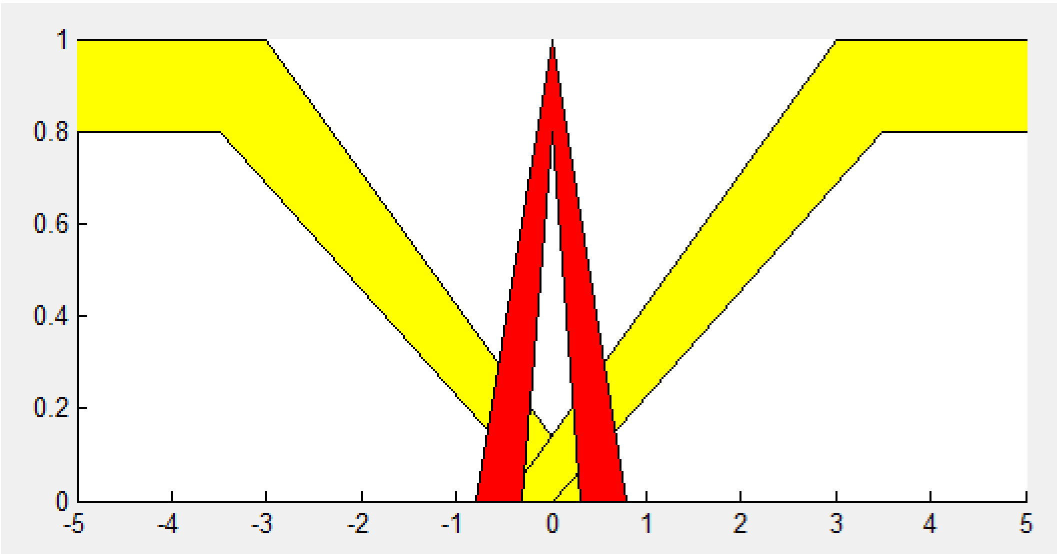
Derivative of Error Membership Functions

	p1	p2	p3	p4	p5	p6	p7	p8	p9
Negative	-200	-200	-3	0.5000	-200	-200	-3.5000	0	0.8000
Zero	-0.8000	0	0	0.8000	-0.3000	0	0	0.3000	0.8000
Positive	-0.5000	3	200	200	0	3.5000	200	200	0.8000

Save

Plot

After double click to save the values, user can check the shape of FOU (footprint of uncertainties) by clicking on Plot button. The FOU will be shown on graph at right bottom of Setting Window:



Finally, after defining the antecedents, user must define the consequence for each rule by key in the interval of the consequence. This interval is where the centroid of type reduction will lie within. In this project, each antecedent has 3 MFs, thus there should be 9 consequences as follow:

Rules Table

	Lower Value	Upper Value
If Error is negative and Derivative of Error is negative then Output is within range	-300	-200
If Error is zero and Derivative of Error is negative then Output is within range	-100	0
If Error is positive and Derivative of Error is negative then Output is within range	-100	100
If Error is negative and Derivative of Error is zero then Output is within range	-100	0
If Error is zero and Derivative of Error is zero then Output is within range	-50	50
If Error is positive and Derivative of Error is zero then Output is within range	0	100
If Error is negative and Derivative of Error is positive then Output is within range	-100	100
If Error is zero and Derivative of Error is positive then Output is within range	0	100
If Error is positive and Derivative of Error is positive then Output is within range	200	300

Save

Upon completion of defining IT2FL inference system, user can click on “Run Simulation & Show Result” button to run and show the response of 3 different controllers: PID, T1FL tuned PID and IT2FL tuned PID in the Result Window:

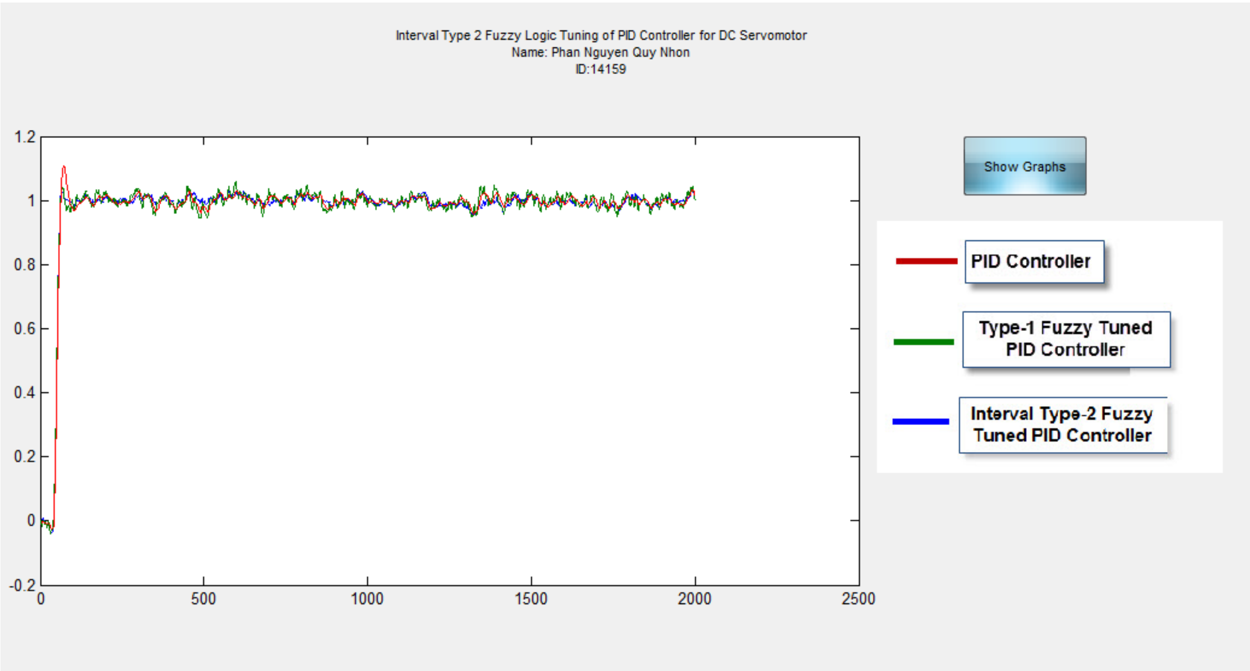


Figure 35. Result Window user interface

After clicking the Show Graphs button, the responses of 3 controllers will be displayed.

Chapter 5: Conclusion & Suggestion

5.1 Conclusion

From the results in chapter 4, it can be generally concluded that both type 1 and type 2 fuzzy logic tuned PID controllers provide improvement over the original PID controller without tuning. The most significantly improved characteristic is the overshoot which means that the DC servo motor would be better protected from abrupt torque change. Furthermore, the interval type 2 fuzzy tuned PID controller offers overshoot, rise time, and errors (IAE and ITAE) dominantly smaller than the other two controllers.

In conclusion, this paper has discussed the auto tuning PID Controller with Interval Type 2 Fuzzy Logic for DC servomotor where it was found that Interval Type-2 Fuzzy Logic Tuned PID controller is proven to be more efficient than conventional PID controller and Type-1 Fuzzy Logic Tuned PID controller in both varying no load and varying load conditions in terms of system performance parameters such as IAE, ITAE, overshoot and rising time. This is a strong foundation for the continuity of work in Interval Type 2 Fuzzy Logic application to tune conventional PID controller more efficiently. This also provides promising potentials in robotics and medical applications that require high precision and quick response such as medical diaphragm pump, infusion pump, pharmaceutical dispenser, minimally invasive surgery.

5.2 Suggestion & Recommendation

The work has only been implemented on 3 membership functions with only 2 types are triangle and trapezoidal. Therefore, the superiority of interval type 2 fuzzy over type 1 fuzzy and typical PID is yet guaranteed universally. Suggested work for further research is to implement comparison and assessment of interval type 2 fuzzy logic with other types of membership functions such as Gaussian and Bell functions.

Besides the simulation work, the efficiency of interval type 2 fuzzy in tuning PID controller requires working prototype of a typical DC servo motor. Thus, effort to transform from Matlab code to electronic devices interface for controlling is necessary.

Reference

[1] R. Bindu and K. N. Mini, "Tuning of PID Controller for DC Servo Motor using Genetic Algorithm " *International Journal of Emerging Technology and Advanced Engineering,* vol. 2, pp. 310-316, March 2012.

[2] R. Firoozian, *Servo Motors and Industrial Control Theory*: Springer London, Limited, 2009.

[3] L. A. Zadeh, "Fuzzy sets," *Information and Control,* vol. 8, pp. 338-353, 6// 1965.

[4] N. Khongkoom, A. Kanchanathep, S. Nopnakeepong, S. Tanuthong, S. Tunyasrirut, and R. Kagawa, "Control of the position DC servo motor by fuzzy logic," in *TENCON 2000. Proceedings*, 2000, pp. 354-357 vol.3.

[5] N. Sae Kyu, "Fuzzy time delay control for DC servo motor," in *Fuzzy Systems, 2001. The 10th IEEE International Conference on*, 2001, pp. 485-488.

[6] L. Ying-Chih, "The application of fuzzy logic control to speed control of a DC servo motor system," in *American Control Conference, 1994*, 1994, pp. 590-594 vol.1.

[7] K. Young-Ho and K. Lark-Kyo, "Design of neuro-fuzzy controller for the speed control of a DC servo motor," in *Electrical Machines and Systems, 2001. ICEMS 2001. Proceedings of the Fifth International Conference on*, 2001, pp. 731-734 vol.2.

[8] J. M. Mendel and R. I. B. John, "Type-2 fuzzy sets made simple," *Fuzzy Systems, IEEE Transactions on,* vol. 10, pp. 117-127, 2002.

[9] N. N. Karnik and J. M. Mendel, "Introduction to type-2 fuzzy logic systems," in *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, 1998, pp. 915-920.

[10] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I," *Information Sciences*, vol. 8, pp. 199-249, // 1975.

[11] S. Singh and A. K. Pandey, "Design of PI Controller to Minimize the Speed Error of DC Servo Motor " *International Journal of Scientific & Technology Research,* vol. 1, pp. 95-98, Nov 2012.

[12] G. Chen and T. T. Pham, *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*: Taylor & Francis, 2000.

[13] *Fuzzy Logic Toolbox for Use with MATLAB*: MathWorks, 2001.

[14] S. N. Sivanandam, S. Sumathi, and S. N. Deepa, *Introduction to Fuzzy Logic using MATLAB*: Springer-Verlag New York, Inc., 2006.

[15] M. A. Johnson, J. Crowe, and M. H. Moradi, *PID Control: New Identification and Design Methods*: Springer, 2006.

[16] N. N. Karnik, J. M. Mendel, and L. Qilian, "Type-2 fuzzy logic systems," *Fuzzy Systems, IEEE Transactions on,* vol. 7, pp. 643-658, 1999.

[17] W. Dongrui and J. M. Mendel, "Enhanced Karnik-Mendel Algorithms for Interval Type-2 Fuzzy Sets and Systems," in *Fuzzy Information Processing Society, 2007. NAFIPS '07. Annual Meeting of the North American*, 2007, pp. 184-189.

[18] S. Greenfield, F. Chiclana, R. John, and S. Coupland, "The sampling method of defuzzification for type-2 fuzzy sets: Experimental evaluation," *Information Sciences*, vol. 189, pp. 77-92, 4/15/ 2012.

[19] D. Xinyu and Y. Hao, "Deriving analytical structure of a type-2 fuzzy PD/PI controller," in *Fuzzy Information Processing Society, 2008. NAFIPS 2008. Annual Meeting of the North American*, 2008, pp. 1-6.
[20] W. Dongrui and W. W. Tan, "Interval Type-2 Fuzzy PI Controllers: Why They are More Robust," in *Granular Computing (GrC), 2010 IEEE International Conference on*, 2010, pp. 802-807.
[21] S. M. Z. Al-Ubaidi and M. M. F. Algreer, "Real Time Implementation of PID and Fuzzy PD Controllers for DC-Servo Motor Based on Lab View Environment," *Tikrit Journal of Engineering Sciences*, vol. 19, pp. 71-81, June 2012.
[22] O. Wahyunggoro and N. B. Saad, "Development of fuzzy-logic-based self tuning PI controller for servo motor," in *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, 2008, pp. 1545-1550.
[23] S. Yanagawa and I. Miki, "PID auto-tuning controller using a single neuron for DC servo motor," in *Industrial Electronics, 1992., Proceedings of the IEEE International Symposium on*, 1992, pp. 277-280.
[24] A. Susperregui, G. Tapia, and A. Tapia, "Application of two alternative sliding-mode control approaches to DC servo motor position tracking," *Electric Power Applications, IET*, vol. 1, pp. 611-621, 2007.
[25] C. P. Bottura and G. L. O. Serra, "Neural gain scheduling multiobjective genetic fuzzy PI control," in *Intelligent Control, 2004. Proceedings of the 2004 IEEE International Symposium on*, 2004, pp. 483-488.
[26] F. Rios-Gutierrez and Y. F. Makableh, "Efficient position control of DC Servo motor using backpropagation Neural Network," in *Natural Computation (ICNC), 2011 Seventh International Conference on*, 2011, pp. 653-657.
[27] N. N. Karnik and J. M. Mendel, "Type-2 fuzzy logic systems: type-reduction," in *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, 1998, pp. 2046-2051 vol.2.