LEBANESE UNIVERSITY
Faculty of Sciences II

PROJECT REPORT

Conventional PID Controller versus Fuzzy Logic Tuned PID Controller for DC Motor

By

Elissa Bou Karam

Presented to Dr. Georges MAANI

July 5, 2021
Spring Semester

I. Abstract

The development of *Proportional-Integral-Derivative (PID)* and *Self-Tuning Fuzzy PID* Controllers is to be applied to control a direct current (DC) motor. The nonlinearities and uncertainties of the system are overcome by the simulation study of both controllers for the armature voltage-controlled DC motor. To obtain fundamentally robust systems, the *Fuzzy Logic Controller (FLC)* is designed according to 49 fuzzy rules for self-tuning every parameter of the Fuzzy PID controller. The parameters of the proportional, integral and derivative (K_P, K_I, K_D) gains of the PID controller are tuned using fuzzy logic. The two inputs of the FLC are the motor speed error between the reference and the actual speed, and the rate of speed change error. The output of the FLC is used as the parameter of the PID controller to control the speed of the DC motor. The best performance of the Fuzzy PID is compared with the conventional PID. To find the best Fuzzy PID performance, triangular, trapezoidal and Gaussian *membership functions* defined in the FIS editor in MATLAB are assessed. The step response, load disturbances and noise disturbances are the three simulated scenarios. To tune the Fuzzy PID, trial and error is used by taking into consideration the *Rise time, Settling time, Present Overshoot and Integral Absolute error* as performance parameters.

Keywords: *Proportional-Integral-Derivative (PID), Self-Tuning Fuzzy PID, Fuzzy Logic Controller (FLC), membership functions, Rise time, Settling time, Present Overshoot and Integral Absolute error*.

Table of Contents

I. Abstract 1

List of Figures 3

List of Tables 3

II. Introduction 4

III. PID 4

IV. Fuzzy Logic 5

V. PID Control of a DC Motor 6

A. DC Motor Control Theory 6

B. Block Diagram Representation 6

VI. Tuning K_P, K_I, K_D 7

A. Stability 7

B. Optimum behavior 7

C. Tuning Methods 7

D. Manual Tuning 8

E. Ziegler-Nichols Method 8

F. Cohen-Coon Method 8

VII. Using Fuzzy Logic to tune K_P, K_I, K_D 9

A. Structure of a Fuzzy Controller 9

B. Structure of Self-Tuning Fuzzy PID Controller 10

VIII. Fuzzy Logic for Tuning K_P, K_I, K_D for a DC Motor Example 10

IX. Simulink Model from MATLAB 11

A. Conventional PID 11

B. Fuzzy-PID 14

X. Using Lookup Tables for Tuning K_P, K_I, and K_D in the Fuzzy Logic Controller 19

XI. Neuro Fuzzy DC Motor 19

XII. Scope for Further Work 19

References 20

List of Figures

Figure 1: PID controller block diagram 5
Figure 2: circuit diagram of a DC motor 6
Figure 3: block diagram of a DC motor 6
Figure 4: block diagram of a fuzzy logic controller 9
Figure 5: structure of self-tuning fuzzy PID controller 10
Figure 6: Simulink model of the conventional PID 11
Figure 7: DC Motor block diagram 11
Figure 8: K_P, K_I, K_D Chosen Values for the Conventional PID 12
Figure 9: Block Response and Tuned Response as a Function of Time 13
Figure 10: Output of the Conventional PID as a function of time 13
Figure 11: Simulink model of the Fuzzy-PID controller 14
Figure 12: Simulink model of the MCC Block 15
Figure 13: Simulink model of the Fuzzy logic block 15
Figure 14: Fuzzy Inference System (FIS) 16
Figure 15: the membership function used in the FIS 16
Figure 16: Fuzzy Rules used in the Fuzzy-PID System 17
Figure 17: Fuzzy Inference System rules 17
Figure 18: Fuzzy Inference System surface 18
Figure 19: Output of the Fuzzy-PID System as a function of time 18

List of Tables

Table 1: Advantages and Disadvantages of Different Tuning Methods 7
Table 2: Effects of Increasing a Parameter Independently 8
Table 3: Setting the Gains for Different Control Types using the Ziegler-Nichols Method 8
Table 4: Cohen-Coon Method Settings for Different Control Types 9
Table 5: Definition of the Variables R, N, and L used in the Cohen-Coon Method 9
Table 6: Membership Function Rules 10
Table 7: Summary Table of the Fuzzy rules 17

II. Introduction

High performance motor drives are widely used in various industrial and daily applications. The development of a high-performance motor must ensure good dynamics, speed, command tracking, and load regulating response to perform tasks. Home appliances, robotics, and industrial applications require speed and position control; hence, DC drives have a crucial role due to their simplicity, ease of application, high reliability, flexibility and convenient cost. A single power conversion from AC to DC is used rendering the DC drives less complex than AC motors, which also have much inferior speed torque characteristics. In addition, DC motors whose drives are usually less expensive for most horsepower ratings provide an excellent control of acceleration and deceleration speeds. As a result, DC motors have long been used as adjustable speed machines with countless uses where precision is crucial for the desired performance. Speed controllers can be: proportional integral (PI), proportional integral derivative (PID) Fuzzy Logic Controller (FLC) or the combination between them: Fuzzy-Neural Networks, Fuzzy- Genetics Algorithm, Fuzzy-Antz Colony, Fuzzy-Swarm. Most control systems are operated by the proportional – integral – derivatives (PID) controller due to its simplicity, clear functionality, applicability, and ease of use. If the PID parameters are tuned properly, a very robust and reliable performance is obtained from the PID controller.

However, speed controllers show nonlinearity problems in DC motors when applying the conventional algorithm (PI, PD, and PID). Finding an accurate nonlinear model of a DC motor is difficult; furthermore, parameters obtained from the system are considered as approximation values. On the other hand, Fuzzy Logic Control (FLC) is becoming an alternative or a complementary to the conventional control strategies. Fuzzy controllers are nonlinear and capable of executing diverse complex nonlinear control action. Contrasting conventional control, one can design an FLC without specific knowledge of the system model including the zeroes and poles of the transfer function governing the system. Two of the critical inputs of the fuzzy control system designing are imitating the way humans learn, the tracking error, and the rate change of error.

III. PID

Throughout the decades, PID has been the most common form of control algorithm used in industrial control; this is due to its robust performance and functional simplicity. The three basic coefficients of PID are proportional, integral, and derivative; the variation of those three coefficients gives the optimal results needed in the specific application. [1] Since PID has a loop feedback mechanism, the error can be calculated as the difference between the measured output and the desired output. The controller then minimizes the error by tuning the three PID coefficients, which are the inputs for the process control. Despite the design being generic, the parameters used in calculation are tuned according to the specific system.

The PID control system can be expressed in the following equation:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

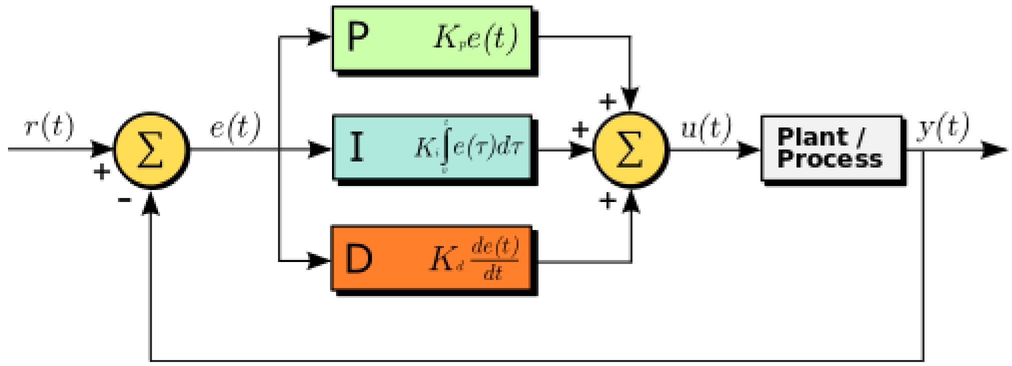

Figure 1: PID controller block diagram

The PID controller calculation is called three-term since it involves the three parameters proportional (P), integral (I), and derivative (D). The proportional component is only related to the current error; the integral component sums the recent errors, and the derivative component is proportional to the rate of change of the error. The weighted sum of the three components is used to change the process or plant. To interpret these values in terms of time, it could be said that the P depends on the error right now, the I depends on the accumulation of past errors, and the D depends on the current change rate of errors to predict the future errors.

Tuning the three parameters allows the controller to provide control action needed for the specific process requirements. The controller's response can be evaluated as its response to an error, how much it overshoots the set point, and how much the system oscillates.

Not all applications require all three modes to provide the appropriate system control; one or two modes can be used by setting a null gain to the undesired control. A PID controller can thus be a PI, PD, P, or I controller.

IV. Fuzzy Logic

Fuzzy logic is the concept of modeling values that are vague in the concept of being true or false. Instead of being either true or false, the concept of degree of truth is introduced. In 1965, Lotfi Zadeh introduced fuzzy logic theory where a fuzzy set assigns a degree of membership to an element of the universe. The standard set of truth degrees is [0,1], where 0 represents "totally false", 1 represents "totally true", and the other numbers refer to partial truth, i.e., intermediate degrees of truth. [2]

Fuzzy logic has advantages by introducing a concept of human intelligence to the controller, having a simple design, reliability, fast response, high precision, cost effectivity, and ease of handling the non-linearity of the system. In systems where the process parameters cannot be identified and described precisely, the use of fuzzy logic controllers comes as an advantage. [3]

However, the use of fuzzy logic poses some disadvantages and difficulties in some systems. For large systems, developing the fuzzy rules by hand is quite difficult. Selecting the appropriate membership function shapes and fine-tuning the fuzzy solutions for specific accuracy imposes a hassle due to the need of trial and error to improve expert knowledge needed for accuracy and robustness.

[4]

V. PID Control of a DC Motor

A. DC Motor Control Theory

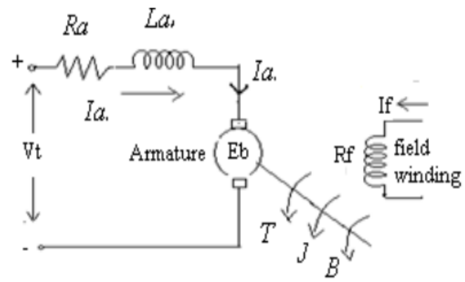

Figure 2: circuit diagram of a DC motor

Figure 2 shows the electric circuit of the motor. The motor's speed can be controlled by controlling the voltage using a PID controller.

B. Block Diagram Representation

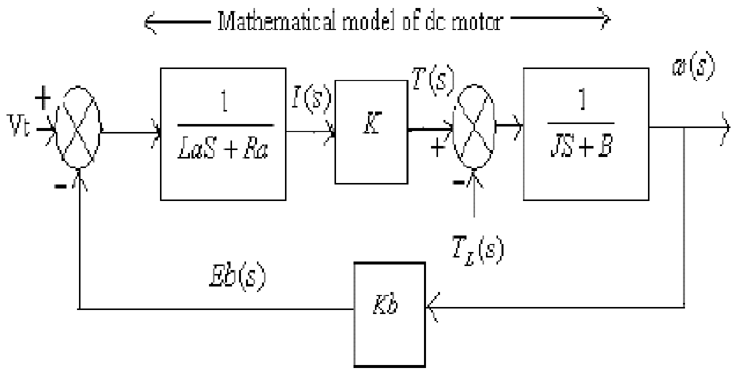

Figure 3: block diagram of a DC motor

The transfer function of the DC motor's speed as a function of the input voltage is given by:

$$G(s) = \frac{\omega(s)}{V(s)} = \frac{K_T}{(R_a + sL_a)(Js + B) + K_b K_T}$$

Practically the armature inductance is relatively small and can be neglected; thus, the transfer function can be simplified to the following:

$$G(s) = \frac{K_m}{\tau s + 1}$$

$$where \quad K_m = \frac{K_T}{R_a B + K_b K_T} \quad is\ the\ motor\ gain$$

$$and \quad \tau = \frac{R_a J}{R_a B + K_b K_T} \quad is\ the\ motor\ time\ constant$$

[4]

VI. Tuning K_P, K_I, K_D

Tuning the parameters is the adjustment made to obtain the optimal control response in terms of stability and other requirements based on the application. To tune the K_P, K_I, and K_D parameters, different methods can be used depending on the system.

A. Stability

The stability of the system response is crucial and oscillations should be minimal. Oscillations occur when the gains of the proportional, integral, and derivative terms are chosen incorrectly. This renders the system unstable meaning that the output diverges from the desired output and cannot be limited except by saturation or mechanical breakage. Excess gain causes this instability specifically in the presence of lag. Usually, oscillations should not occur for any combination of process conditions and set points. However, some systems allow marginal stability.

B. Optimum behavior

The application of the system dictates the optimum behavior on a process change or set point change. Regulation and command tracking are two requirements that refer to how well the controlled variable tracks the desired value. Rise time and settling time are some of the specific command tracking criteria. In some applications, it would be unsafe to have an overshoot of the process variable beyond the set point. Other processes should keep the energy expended in reaching a new set point minimal. [5]

C. Tuning Methods

Multiple methods for tuning the PID loop exist. Choosing the method is affected by the possibility of taking the system "offline" to perform the tuning and the response time of the system.

Table 1: Advantages and Disadvantages of Different Tuning Methods [6]

Method	**Advantages**	**Disadvantages**
Manual Tuning	No math required Online method	Requires experienced personnel
Ziegler-Nichols	Proven method Online method	Process up set Some trial-and-error Very aggressive tuning
Software Tools	Consistent tuning Online or offline method May include valve and sensor analysis Allow simulation before downloading	Some cost and training involved
Cohen-Coon	Good process models	Some math Offline method Only good for 1st order for processes

D. Manual Tuning

As previously discussed in Table 1, this method is used when the system must remain online. First, the K_I and K_D are set to zero. Then, the K_P is increased to reach the ultimate gain K_U at which output oscillates; next, the K_P must be set to almost half of K_U at which the response is now a "quarter amplitude decay". After that, K_I is increased until the offset is correct in enough time for the process at hand but not too much to avoid instability. Lastly, K_D is increased for the loop to reach the reference quickly after a load disturbance if needed. Note that increasing K_D greatly causes an overshoot due to excessive response. Tuning a PID loop to be fast usually results in a slight overshoot to reach the set point faster. This might not be acceptable in some systems where an overshoot might be dangerous; in such systems, the K_P should be set to much less than half of K_U; this is called an overdamped closed-loop system.

Table 2: Effects of Increasing a Parameter Independently

Response	Rise Time	Overshoot	Settling Time	Steady-state Error	Stability
K_P	Decrease	Increase	Small Change	Decrease	Degrade
K_I	Decrease	Increase	Increase	Eliminate	Degrade
K_D	Small Change	Decrease	Decrease	No Change	Improve if K_D is small

E. Ziegler-Nichols Method

John G. Ziegler and Nathaniel B. Nichols introduced this tuning method that is also used when the system needs to stay online and starts by also setting the K_I and K_D to zero. Then, K_P is increased to reach the ultimate gain K_U. The following table shows how to set the gains using the ultimate gain K_U and the oscillation period T_U.

Table 3: Setting the Gains for Different Control Types using the Ziegler-Nichols Method [7]

Control Type	K_P	K_I	K_D
P	$0.50K_U$	-	-
PI	$0.45K_U$	$1.2K_P/T_U$	-
PID	$0.60K_U$	$2K_P/T_U$	$K_P T_U/8$

F. Cohen-Coon Method

As previously stated, the Cohen-Coon method can only be applied to first-order models with large process delays. This method is used to correct the slow steady-state response given by the Ziegler-Nichols method if the dead time is large relative to the open loop time constant. The Cohen-Coon method is used when the system can be taken offline for tuning; this means that a step change can be introduced to the input when it is at steady state. After that, the output is measured according to the time constant and the time delay; then, this response is used to evaluate the initial control parameters. A set of pre-determined settings is needed to get the minimum offset and standard decay ratio that refers to the response having decreasing oscillations where the second oscillation has ¼ the amplitude of the first oscillations. [8]

To apply the Cohen-Coon method, one must first wait for the process to reach its steady state. After that, a step change is introduced in the input. Then, according to the output, an approximate first order process is obtained with time constant τ delayed by τ_{dead} units from when the input was introduced.

Table 4: Cohen-Coon Method Settings for Different Control Types [9]

Control Type	**K_P**	**T_I**	**T_D**
P	$\frac{P}{NL}\left(1+\frac{R}{3}\right)$	-	-
PI	$\frac{P}{NL}\left(0.9+\frac{R}{12}\right)$	$L\left(\frac{30+3R}{9+20R}\right)$	-
PID	$\frac{P}{NL}\left(1.33+\frac{R}{4}\right)$	$L\left(\frac{30+3R}{9+20R}\right)$	$\frac{4L}{11+2R}$

Where T_I and T_D are the integral time and derivative time respectively, and the variables R, N, and L are defined as follows:

Table 5: Definition of the Variables R, N, and L used in the Cohen-Coon Method

P	Percentage change of input
N	Percentage change of output $/\tau$
L	τ_{dead}
R	τ_{dead}/τ

VII. Using Fuzzy Logic to tune K_P, K_I, K_D

A. Structure of a Fuzzy Controller

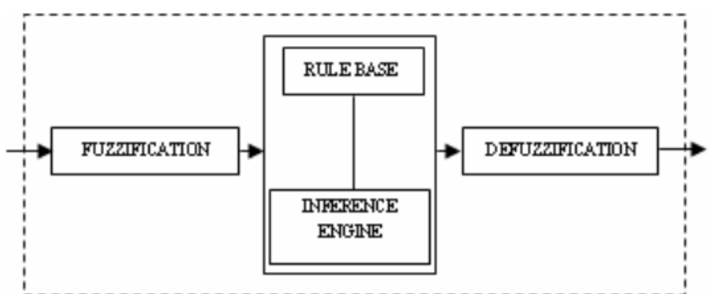

Figure 4: block diagram of a fuzzy logic controller

Figure 4 shows that a fuzzy controller consists of four main parts: fuzzification, rule base, inference engine, and defuzzification. [10]

B. Structure of Self-Tuning Fuzzy PID Controller

In a self-tuning fuzzy PID, the fuzzy tuner tunes the parameters K_P, K_I, K_D of the PID controller. In nonlinear plants with unpredictable parameter variations, it is necessary to automatically tune the PID parameters. Consider the structure of the self-tuning fuzzy PID controller where $e(t)$ is the error between the desired output and the actual output, and $de(t)/dt$ is the derivation error. To tune the PID parameters, fuzzy inference is used; this provides a nonlinear mapping from the error and the derivation of error to PID parameters. [10]

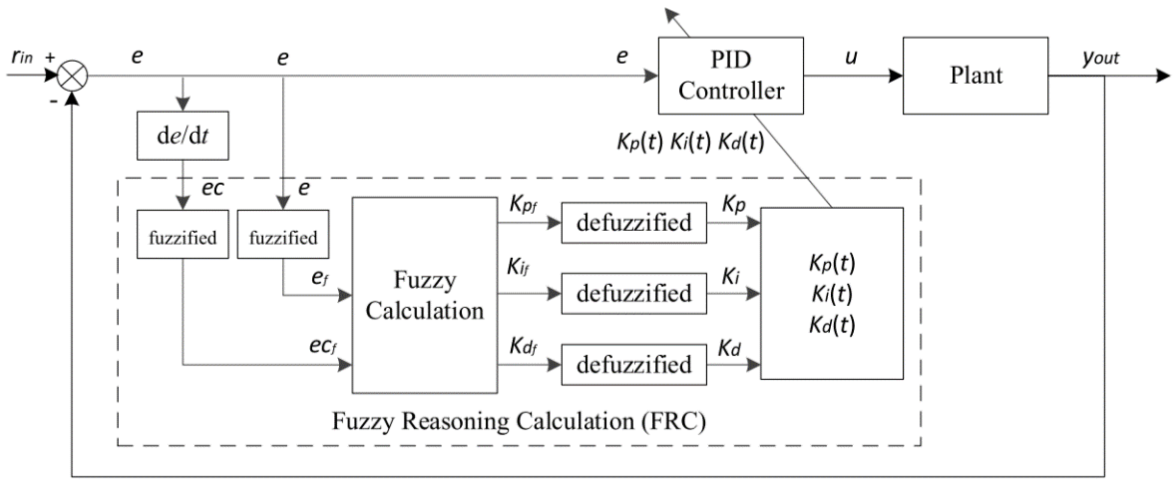

Figure 5: structure of self-tuning fuzzy PID controller [11]

VIII. Fuzzy Logic for Tuning K_P, K_I, K_D for a DC Motor Example

For the fuzzy system to operate, the membership function along with the fuzzy rules must be defined. All rules are defined using sentences. For example, If (e is NG) and (de is PG) then (du is EZ); this translates to the following: if the error e is negative and the derivative error de is positive, then the derivative output du is equal to zero. For this application, nine rules are used and are stated below.

Table 6: Membership Function Rules

1	If (e is NG) and (de is NG) then (du is NG)
2	If (e is NG) and (de is EZ) then (du is NG)
3	If (e is NG) and (de is PG) then (du is EZ)
4	If (e is EZ) and (de is NG) then (du is NG)
5	If (e is EZ) and (de is EZ) then (du is EZ)
6	If (e is EZ) and (de is PG) then (du is PG)
7	If (e is PG) and (de is NG) then (du is EZ)
8	If (e is PG) and (de is EZ) then (du is PG)
9	If (e is PG) and (de is PG) then (du is PG)

IX. Simulink Model from MATLAB

For this project, the conventional PID and fuzzy-PID controllers were simulated in MATLAB using Simulink.

A. Conventional PID

A simulation of the DC motor was conducted using the conventional PID model to control the speed of the motor. The PID block diagram consists of the DC motor block having two outputs and one input from the PID controller. The input of the PID controller is the difference between the desired speed and the output speed passing through a zero-order holder. The PID parameters were chosen to have the values shown in Figure 8. After tuning the parameters, the graph shown in Figure 10 is obtained which shows the output before and after tuning.

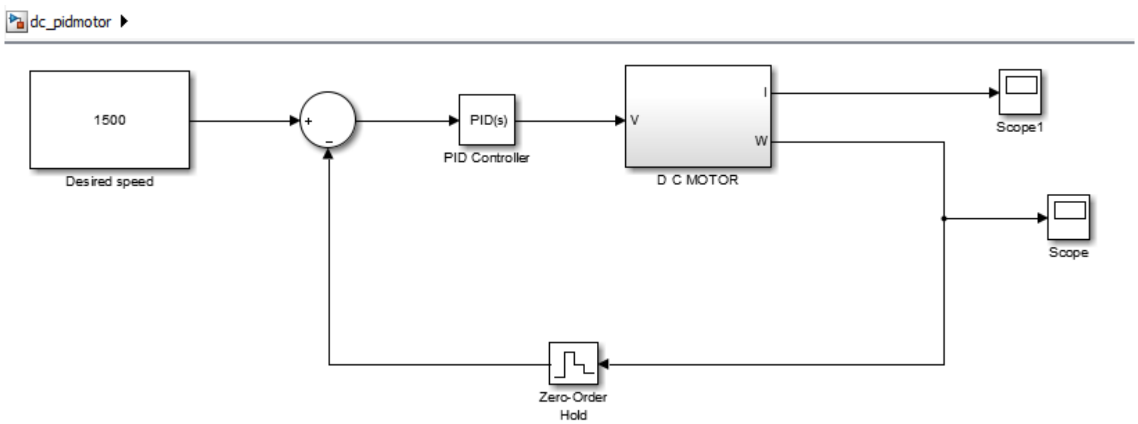

Figure 6: Simulink model of the conventional PID

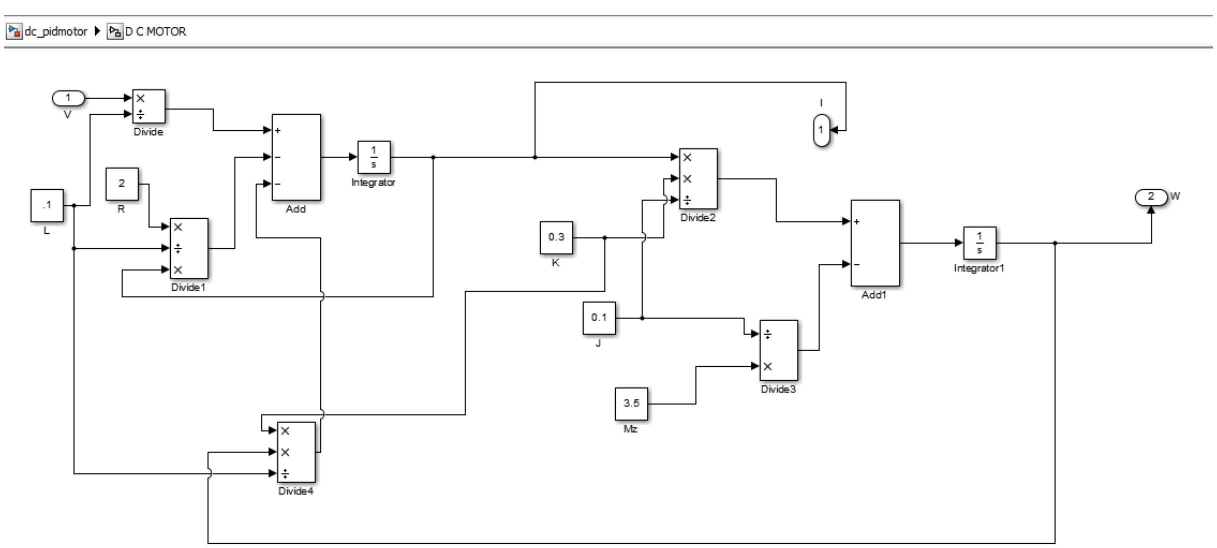

Figure 7: DC Motor block diagram

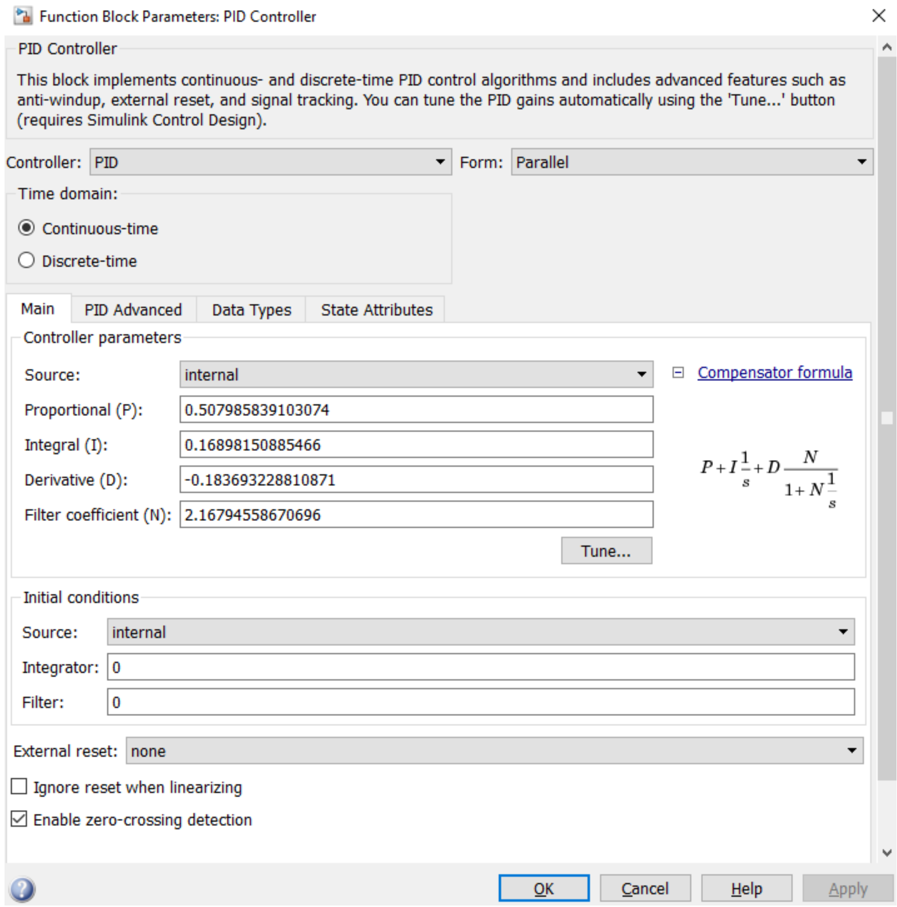

Figure 8: K_P, K_I, K_D Chosen Values for the Conventional PID

After conducting a simulation of the conventional PID Controller, the following graph of the output speed as a function of time can be seen from the scope block. A spike can be realized before the output reaches stability; in some systems, such an output might be dangerous. An oscillation also occurred before reaching the steady state. The output speed oscillates between 1553.1 rad/s at 2.85 s and 1377.95 rad/s at 5.19 s. The speed of the motor reached a steady state at approximately 25 seconds.

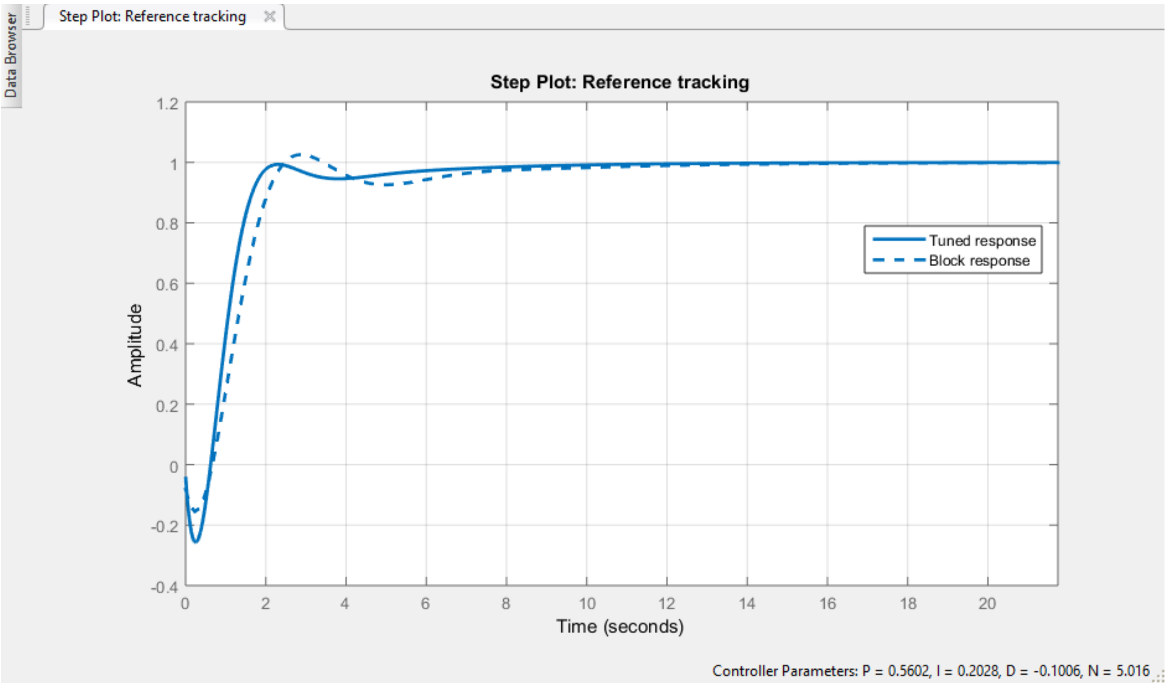

Figure 9: Block Response and Tuned Response as a Function of Time

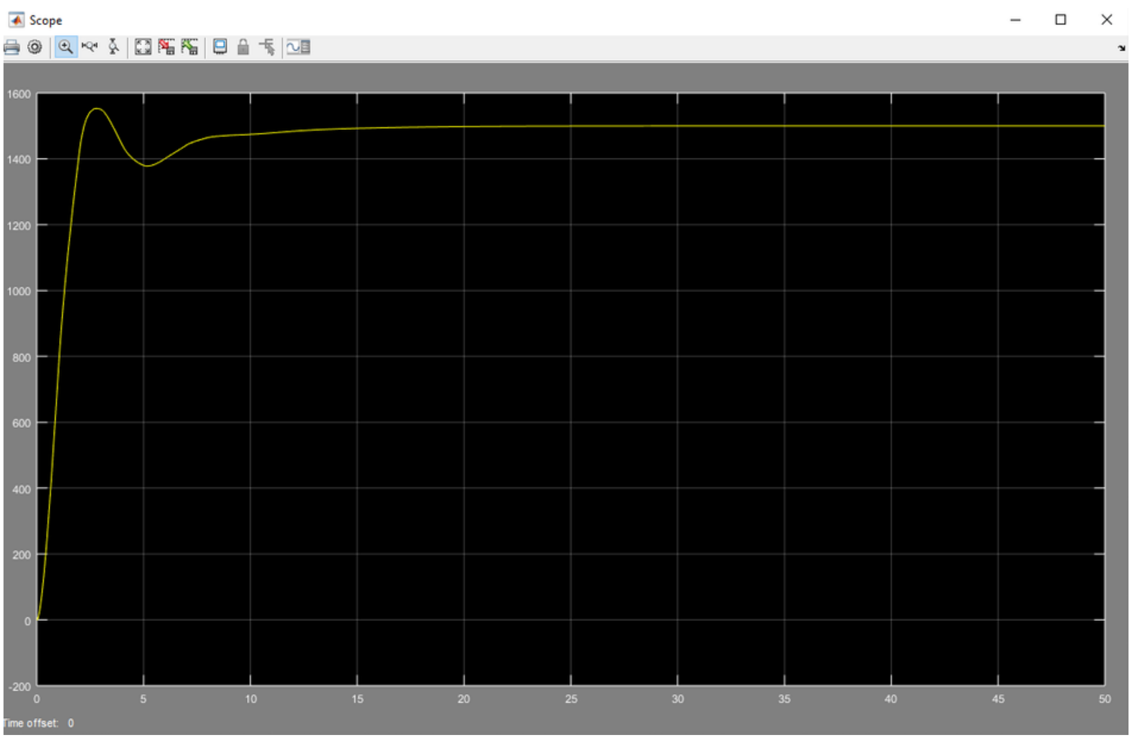

Figure 10: Output of the Conventional PID as a function of time

B. Fuzzy-PID

To enhance reaching the desired speed, developing a Fuzzy System to control the PID parameters was introduced and simulated on Simulink. The diagram below shows the block system that was designed to control the PID parameters using fuzzy logic.

To explain the block diagram, the path from the output to the input will be taken. The output speed seen on the oscilloscope is the output of the bus having as input the reference speed having a gain of 1000 along with the output of the MCC block having a gain of $40/\pi$. The MCC block has as input the step function and the fuzzy logic block with a gain of 100. The fuzzy logic block has as input a signal and its derivative with respect to time. This signal is the difference between the reference speed and the output of the MCC block with a gain of $1/\left(1000 \times \frac{\pi}{40}\right)$.

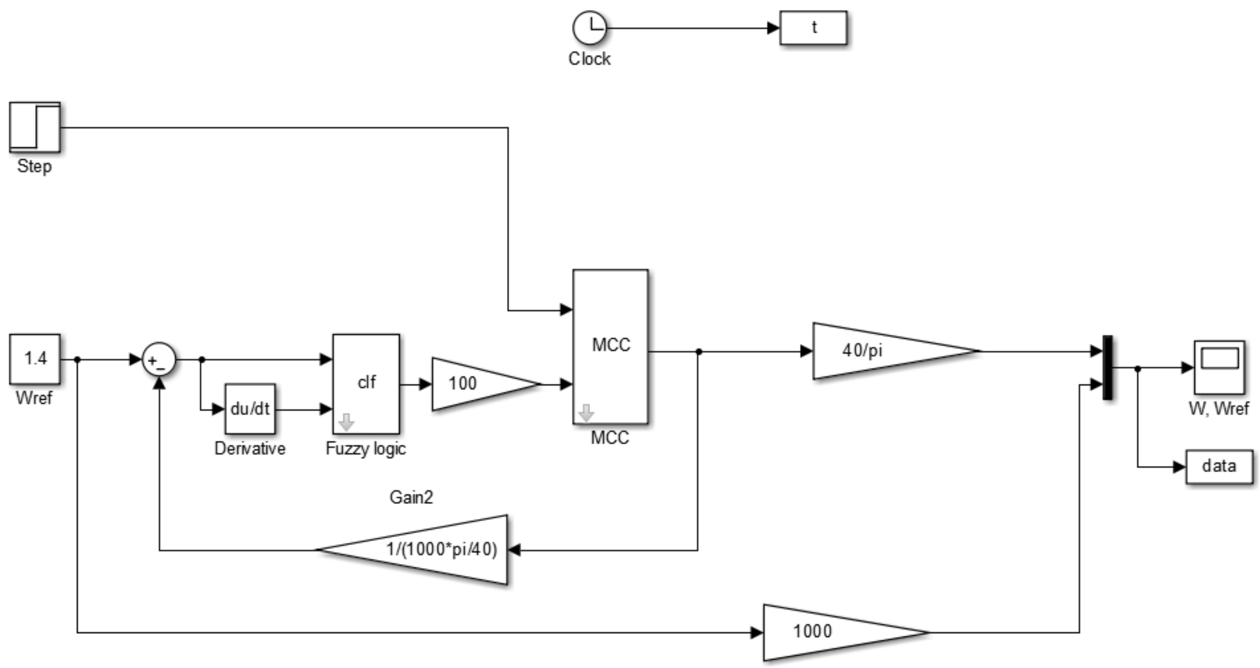

Figure 11: Simulink model of the Fuzzy-PID controller

The MCC block is a simulation of the DC motor block diagram discussed in section V.B. The path going from the output to the input will also be taken to explain the MCC block. The transfer function block called "Transfer Fcn1", which has $1/(Js + B)$ for equation, has as input the difference between a signal and the step function. The signal is the result of the transfer function, "Transfer Fcn", of equation $1/(L_m s + R_m)$ and having a gain of $Kv \times lf$. The transfer function of equation $1/(L_m s + R_m)$ has as input the difference between input number 2 and the output of "Transfer Fcn1" having a gain of $Kv \times lf$.

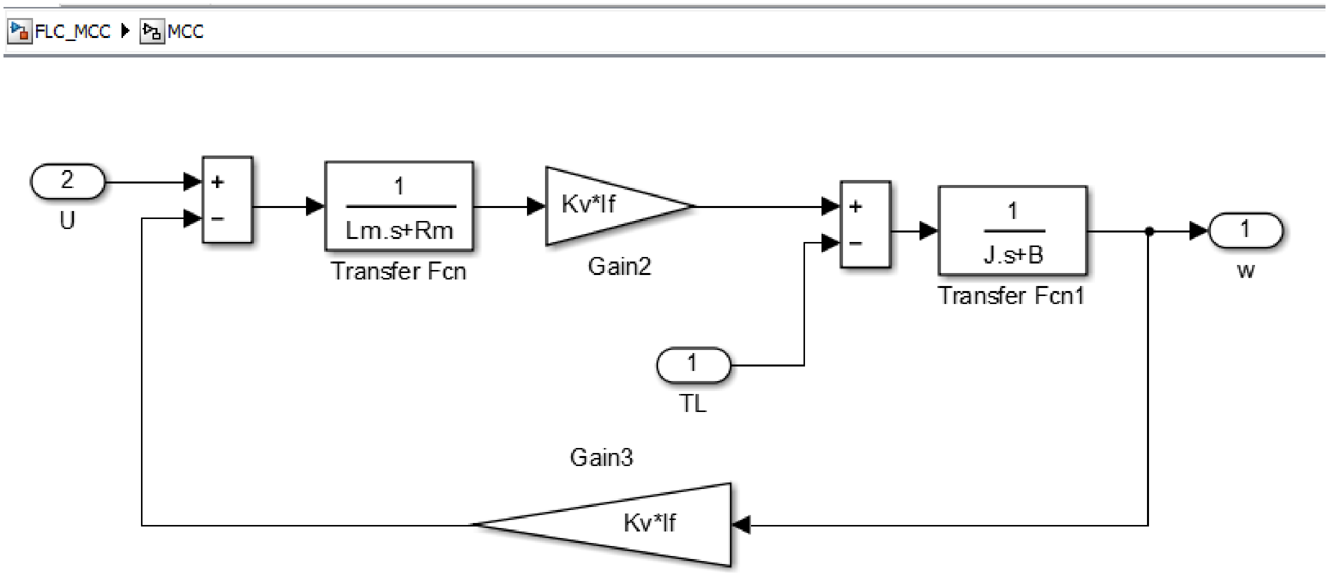

Figure 12: Simulink model of the MCC Block

The fuzzy logic block has 2 inputs as discussed: the error e with its gain G_e and the derivative error de with its gain G_{de}. The fuzzy logic controller is programmed using the fuzzy inference system that was created using the fuzzy rules discussed previously in section VIII. The output of the fuzzy logic controller goes then through a gain and an integration.

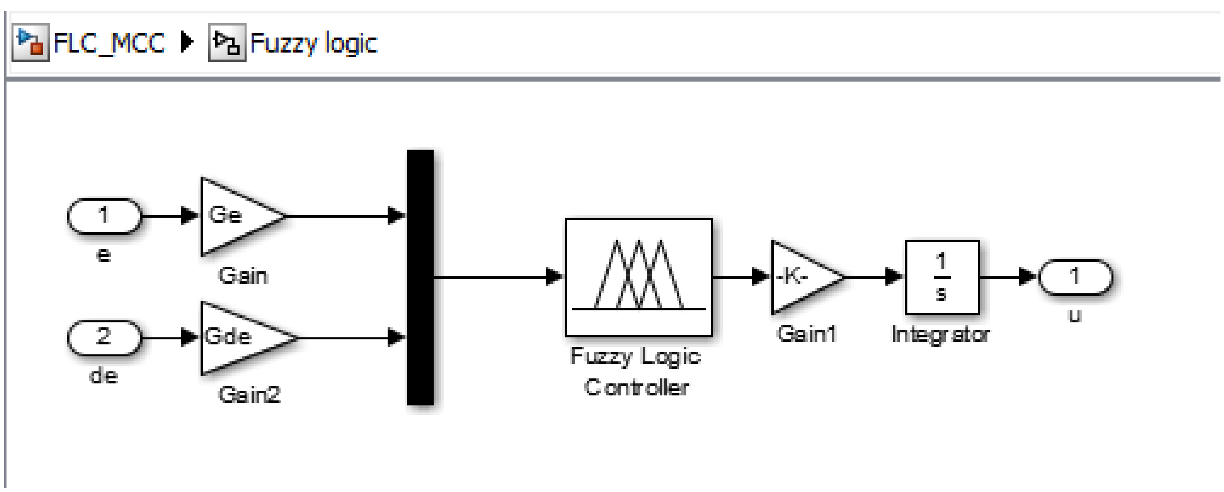

Figure 13: Simulink model of the Fuzzy logic block

The fuzzy inference system (FIS) is what controls the fuzzy logic controller based on fuzzy rules. Here, the FIS is of mandani type having a triangular form and defined by 9 rules. The FIS in use has the following properties: an AND method set to minimum, an OR method set to maximum, the implication set t minimum, the aggregation set to maximum, and the defuzzification method used is the centroid. The FIS has 2 inputs, the error e and the derivative error de, and 1 output du; each of these variables ranges between -1 and 1.

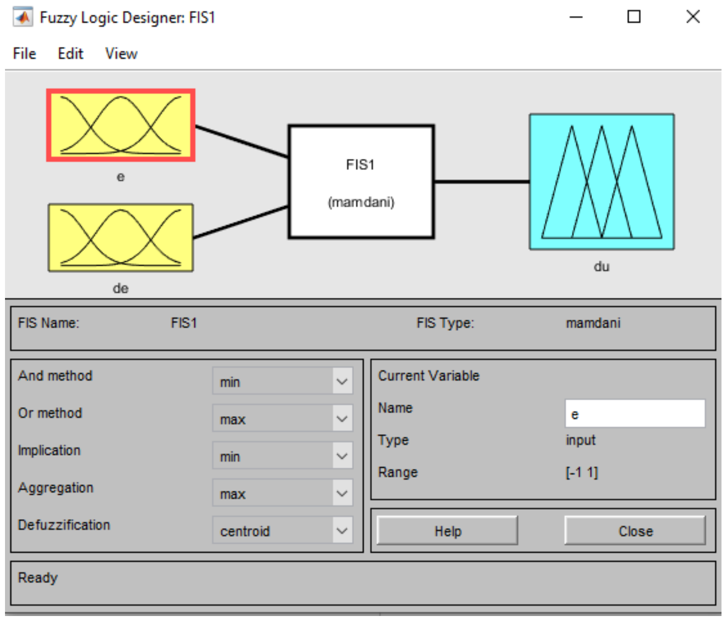

Figure 14: Fuzzy Inference System (FIS)

The membership function plots of the variables have a triangular membership function plot. "Negative" (NG) stays constant at 1 between -1000 and -0.5, then decreases to zero. "Equal to Zero" (EZ) increases from 0 to 1 between -0.5 and 0, then decreases back to 0 between 0 and 0.5. "Positive" (PG) increases from 0 to 1 between 0 and 0.5, then remains constant at 1 between 0.5 and 1000.

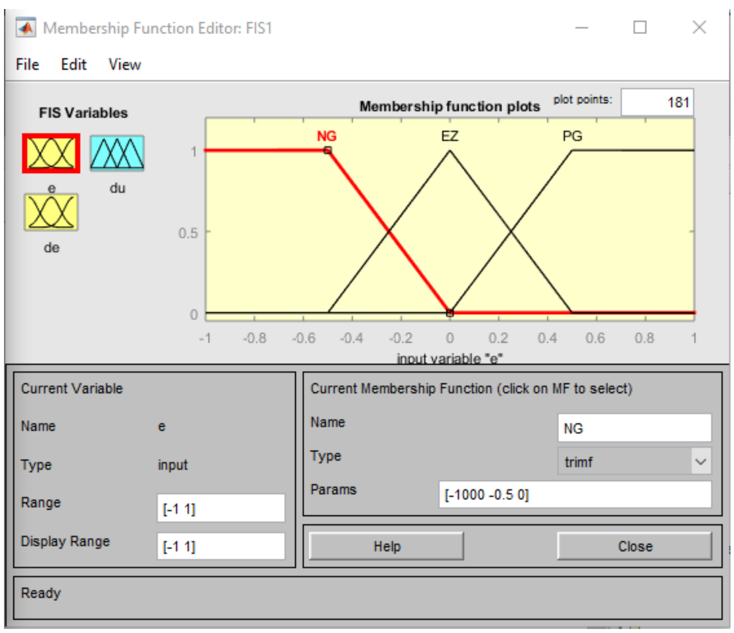

Figure 15: the membership function used in the FIS

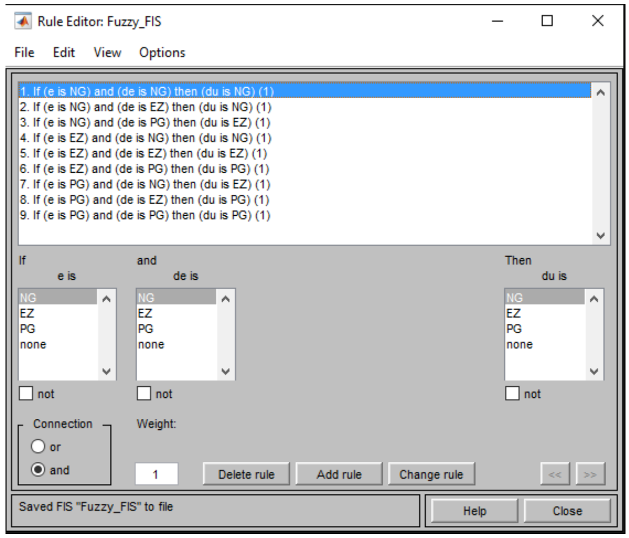

Figure 16: Fuzzy Rules used in the Fuzzy-PID System

The fuzzy rules can be summarized in the following table.

Table 7: Summary Table of the Fuzzy rules

		Error e		
		NG	**EZ**	**PG**
Derivative error de	**NG**	NG	NG	EZ
	EZ	NG	EZ	PG
	PG	EZ	PG	PG

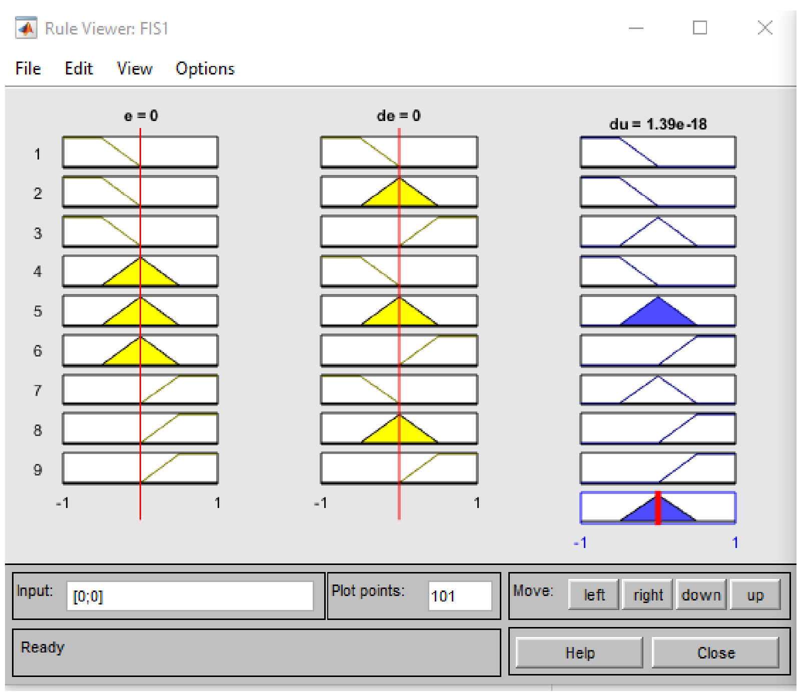

Figure 17: Fuzzy Inference System rules

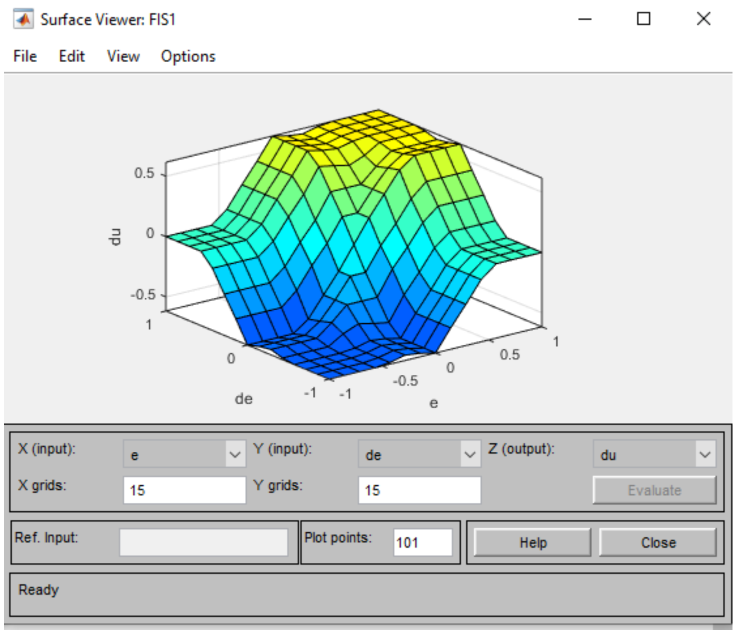

Figure 18: Fuzzy Inference System surface

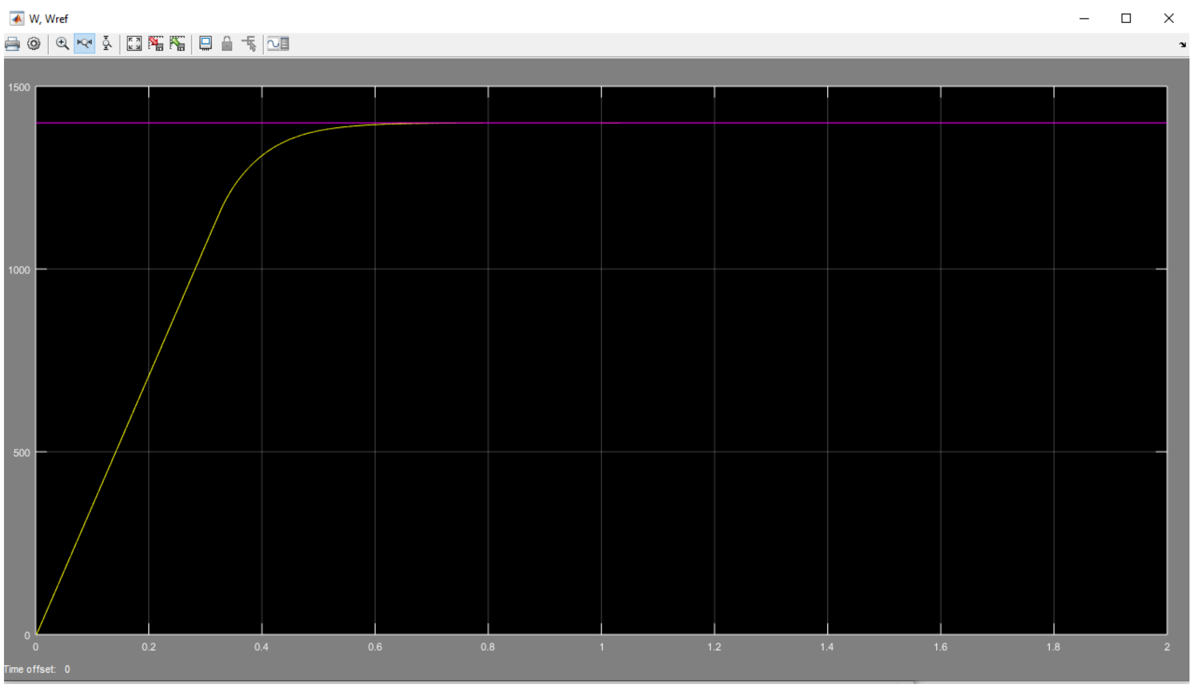

Figure 19: Output of the Fuzzy-PID System as a function of time

The output of the fuzzy-PID system shows a remarkable enhancement in the result. The target speed is now reached faster within 0.7 *s* only, without oscillations and without the spike that was occurring in the conventional PID. This goes to prove the hypothesis that using fuzzy logic to control the PID parameters renders the system faster and more stable.

X. Using Lookup Tables for Tuning K_P, K_I, and K_D in the Fuzzy Logic Controller

The fuzzy inference system surface viewer shows the output du as function of the input error e and the derivative error de in 3-D display. The graph shown in Figure 18 is referred to as the control surface graph. The designed FIS is then simulated using the fuzzy logic controller block in Simulink. The nonlinear surface can be accessed with a Lookup Table to simplify the generated code and improve execution speed. In other words, a set of Lookup Tables can replace a Fuzzy Logic Controller block in Simulink. One table is needed for each output. The data used in the Lookup Table can then be calculated using the commands like evalfis, which is provided by the Fuzzy Logic Toolbox. [12]

XI. Neuro Fuzzy DC Motor

The concept of an Adaptive Neuro-Fuzzy Inference System (ANFIS) combines fuzzy logic and artificial neural network principles, thus forming a hybrid intelligent system that has an enhanced ability to automatically learn and adapt. The main idea for the neuro-adaptive learning technique is to put forward a method to automatically compute the membership function parameters by learning information about a data set in the fuzzy modeling procedure; this allows the FIS to track the given input/output data. A combination of least squares estimation and back-propagation algorithm for membership-function parameter estimation is used to tune the membership function parameters. Through the learning process, the parameters associated with the membership function will change; this is similar to a learning process in a neural network. A gradient vector facilitates the adjustment of the membership function parameters; this gradient vector is a measure of how well the FIS is modeling the input/output data for a given set of parameters. After obtaining the gradient vector, an optimization routine is applied to adjust the parameters in order to reduce the error between the actual output and the desired output. This process lets the fuzzy system learn from the modeled data. This concept is advantageous over the usual fuzzy system since it eliminates the need for a human operator to tune the system by adjusting the bounds of the membership function. [13] Analyses using MATLAB/Simulink simulations of the conventional PID and the ANFIS-PID show a much better performance by the Adaptive Neuro-Fuzzy system, which can be expected due to the adaptive learning, and adjusting ability that enhances the ANFIS. [14]

XII. Scope for Further Work

In this project, the Fuzzy-PID and conventional PID controllers were studied. As the need for a fully automatic systems grows, solutions for smarter self-learning and self-tuning systems appear. To take this research to the next step would be to apply an Adaptive Neuro-Fuzzy Inference System to improve the performance of the DC motor in the aspects of learning and adaptability.

References

[1] PID Theory Explained, Updated Mar 17, 2020, by National Instruments

[2] Fuzzy Logic, First published Tue Nov 15, 2016; substantive revision Tue Jul 18, 2017, by Stanford Encyclopedia

[3] What Is Fuzzy Logic? By MathWorks

[4] Speed Control of DC Motor using Pid Controller Based on Matlab, by Aditya Pratap Singh, Udit Narayan, and Akash Verma

[5] Optimum PI/PID Controllers Tuning via an Evolutionary Algorithm, by Jorge Humberto Urrea-Quintero, Jesús-Antonio Hernández-Riveros, and Nicholas Munoz

[6] A Comparative Study of Conventional PID and Fuzzy-PID for DC Motor Speed Control, by Ahmad Muhyiddin Bin Yusof

[7] Automation Control Systems in Tokamak Plasmas: Current Status and Needs, b Christos Tsironis and K. Iordanis Giannopoulos

[8] Design and Implementation of Energy Efficiency in HVAC Systems Based on Robust PID Control for Industrial Applications, by Muharrem Imal

[9] PID Tuning via Classical Methods. (2021, March 5), Peter Woolf et al. Assistant Professor (Chemical Engineering) at University of Michigan

[10] Application of Self-Tuning Fuzzy PID Controller on Industrial Hydraulic Actuator using System Identification Approach, by Zulfatman and M. F. Rahmar

[11] Fuzzy Controllers with Neural Network Predictor for Second-Order Linear Systems with Time Delay, by Jingwei Liu, Xinyu Tao, Xueyang Ma, Kangmin Feng, and Jiaming Chen

[12] Comparative Study of Control Methods's Application for Pneumatic System in Simulation Environment, by E. Ntimeri, PhD Christos Drosos, and D. Tseles

[13] Neuro-Fuzzy DC Motor Speed Control Using Particle Swarm Optimization, by Boumediene Allaoua, Abdellah Laoufi, Brahim Gasbaoui, and Abdessalam Abderrahmani

[14] Speed Control of DC Motor Using Adaptive Neuro Fuzzy Inference System Based PID Controller, by Md. Asif Hasan, Raj Kumar Mishra, and Srishti Singh