DC MOTOR SPEED CONTROL VIA FUZZY / POLE PLACEMENT / PI CONTROLLER

ALI JUNAID ASHRAF
JUN 2010
<table>
<thead>
<tr>
<th>Programme</th>
<th>Reg. Number</th>
<th>Extent</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASTER OF SCIENCE IN ELECTRICAL ENGINEERING</td>
<td>E3939E</td>
<td>15 ECTS</td>
</tr>
</tbody>
</table>

**Name of student**

ALI JUNAID ASHRAF

**Year-Month-Day**

2010-06-23

**Supervisor**

Mr. KABIR KHAN

**Examiner**

PROF. BJÖRN SOHLBERG

**Company/Department**

DEPARTMENT OF ELECTRICAL ENGINEERING

**Supervisor at the Company/Department**

PROF. BJÖRN SOHLBERG

**Title**

DC MOTOR SPEED CONTROL VIA FUZZY / POLE PLACEMENT / PI CONTROLLER

**Keywords**

DC motor, Fuzzy Controller, Pole Placement Controller, PID controller, GUI
Abstract

This report presents a new way of control engineering. Dc motor speed controlled by three controllers PID, pole placement and Fuzzy controller and discusses the advantages and disadvantages of each controller for different conditions under loaded and unloaded scenarios using software Matlab. The brushless series wound Dc motor is very popular in industrial application and control systems because of the high torque density, high efficiency and small size. First suitable equations are developed for DC motor. PID controller is developed and tuned in order to get faster step response. The simulation results of PID controller provide very good results and the controller is further tuned in order to decrease its overshoot error which is common in PID controllers. Further it is purposed that in industrial environment these controllers are better than others controllers as PID controllers are easy to tuned and cheap. Pole placement controller is the best example of control engineering. An addition of integrator reduced the noise disturbances in pole placement controller and this makes it a good choice for industrial applications. The fuzzy controller is introduce with a DC chopper to make the DC motor speed control smooth and almost no steady state error is observed. Another advantage is achieved in fuzzy controller that the simulations of three different controllers are compared and concluded from the results that Fuzzy controller outperforms to PID controller in terms of steady state error and smooth step response. While Pole placement controller have no comparison in terms of controls because designer can change the step response according to nature of control systems, so this controller provide wide range of control over a system. Poles location change the step response in a sense that if poles are near to origin then step response of motor is fast. Finally a GUI of these three controllers are developed which allow the user to select any controller and change its parameters according to the situation.
Acknowledgements

First, I owe my immense gratitude to Almighty Allah, the most Merciful, the most Gracious and the most Beneficent whose countless favours made it possible for me to accomplish this task.

I would like to take this opportunity to acknowledge the guidance and teaching I received at Högskolan Dalarna during my periods of studies. I also extend my sincerest thanks to Dr Björn Sohlberg who inspired me through their lectures on control systems which leads me to this moment that I am able to make a project in control engineering.

Finally I am very thankful to Swedish government policies for humanity which leads me to come to Sweden and offer free education of engineering which was a dream for me.
Dedication

I would like to dedicate this thesis to my father who encourages me for higher studies and to my mother for endless prayers is the sole reason for my success and achievements in life.
I love you Ami and Abu.
Introduction

Dc motor is commonly used in robotic application and industrial machinery. The beauty of this motor is it provide high torque load sustaining properties. Chapter one describes about the general Dc motor and about its motion. For this thesis DC series wound motor is selected after comparing with others wounding techniques.

Second chapter discuss about the background work in this field. The remarkable work is presented in this section of some previous researchers on DC motor speed control.

Chapter third discuss the suitable equations of DC motor and electrical equation and mechanical equation are developed to check the system using Matlab software. State space representation and transfer function is obtained in this section.

Chapter four discuss the Motor speed control with PID controller and first system is checked without controller on loaded and unloaded condition then add PID controller and system is tuned using its existing tuning methods. After it system is further tuned in order to get desired value with less steady state error. And then discuss the results.

Chapter five discuss the speed control of motor by Pole Placement Controller. This controller is developed by using transfer function of DC motor system and check the system on load disturbance and found that results are not satisfactory. In order to reduce noise factor integrator is added with reduced the effect of disturbance.

Chapter six discuss the speed control with the help of Fuzzy controller. Fuzzy controller provide better control strategies than other controllers. Optimization of Fuzzy controller with simulink model describes in this chapter and a new way for faster response and smooth output Dc chopper is added in the model and results are better than the previous controllers.

Chapter seven discuss the comparison of these three controllers’ results. From the results it proved that Fuzzy Controller is the best controller.
In chapter eight GUI model is developed to from these three controllers which allow the user to change inputs and use any controller according to nature of system.
Contents

1. Introduction ...........................................................................................................................14
   1.1 DC Motor .........................................................................................................................14
   1.2 DC Motion .......................................................................................................................19
   1.3 Why need to control motor .........................................................................................21
   1.4 Speed controllers .........................................................................................................21
   1.5 Why these controllers .................................................................................................22
   1.6 Objective of thesis work ..............................................................................................2
   1.7 Limitation ......................................................................................................................2

2. Background work ..............................................................................................................2

3 Modeling of DC motor ....................................................................................................30
   3.1 Electrical Equations ......................................................................................................30
   3.2 Mechanical Equations ..................................................................................................32
   3.3 State Space Representations .......................................................................................33
   3.4 Transfer Function Block Diagram ................................................................................34

4. Fuzzy logic controller ......................................................................................................39
   4.1 Fuzzy Logic Controller ...............................................................................................39
   4.2 Mamdani Fuzzy Inference ...........................................................................................39
   4.3 Sugeno Fuzzy Inference ..............................................................................................39
   4.4 Developing Fuzzy Expert system ................................................................................42
   4.5 DC Chopper ..................................................................................................................51
   4.6 Simulink Model of Dc Motor using Fuzzy controller ..................................................53
5. Pole placement controller .................................................................................................................. 58

5.1 Pole Placement Controller ........................................................................................................ 58
5.2 The Diophantine Equation ........................................................................................................ 60
5.3 Pole Placement Controller with Disturbances ........................................................................ 64
5.4 Integrator Design ...................................................................................................................... 68
5.5 Simulink Model of Pole placement Controller with Integrator .............................................. 69

6. PID controller .................................................................................................................................. 74

6.1 PID Controller ............................................................................................................................ 74
6.2 PID tuning methods ..................................................................................................................... 76
6.3 Tuning of PID controller by Ziegler-Nichols method .............................................................. 77

7. Comparision of controllers ............................................................................................................. 84

7.1 Comparision between PID and Fuzzy controller ................................................................. 84
7.2 Pole Placement Controller Vs.Fuzzy PID controller .......................................................... 85

8. Graphical User Interface ............................................................................................................... 88

9. Conclusion ...................................................................................................................................... 94

10. Bibliography ................................................................................................................................. 118
Chapter 1.

Introduction

1.1 DC Motor.

A motor convert electrical energy into mechanical energy. There are two types of motor.

1. AC motor.
2. DC motor.

A simple Dc motor use electricity and magnetic field for producing torque which rotate the motor. PMDC permanent magnet DC motor outperforms to AC motor because it provide better speed control on high torque loads and use in wide industrial application. Dc motors are more usable as it designed to use with batteries and solar cells energy sources, which provide portability where we required it and thus provide cost effective solution because it is not possible to have AC power supply in every place, DC motor show its response at both voltage and current. The applied voltage describes the speed of motor while current in the armature windings shows the torque. If applied load increased in the shaft of motor then in order to sustain its speed motor draw more current from supply and if supply is not able to provide enough current then motor speed will be effected. Generally it can be said that applied voltage effect speed while torque is controlled by current. DC motor provide more effective results if chopping circuit is used. Low power DC motors usually use in lifting and transportation purposes as low power AC motors don’t have good torque capability. DC motor used in railway engines, electric cars, elevators, robotic applications, car windows and wide verity of small appliances and complex industrial mixing process where torque cannot be compromised. There are several types of DC motor but most common are brushed DC motor, brushless DC motor, stepper motor, servo motor. These motors are discussed latter also DC motors have three winding techniques such as shunt DC motor, series DC motor, and compound DC motor.
Types of DC motor

**Brushed DC motor.**

The brushed DC motor has permanent magnets and its rotor has electrical magnets which move and generate torque according to applied DC power. This motor works on principle of Lorentz force, according to that if a current carrying conductor placed in magnetic field then it experiences a force or torque. This motor has low initial costs, high reliability and it is simple in speed control which considered its advantages while high maintenance, and low life time is its disadvantages. Cleaning and replacing of commutator as well as brushes(carry current) and springs(attached to one end of brushes so it attached to commutator) involves in maintenance.[1] This motor use different wound techniques which are as follows
**Shunt-Wound.**

This motor has shunt field coils in parallel with the armature. So current is independent of one another between armature and field coils as shown in Fig 1.2. This motor provides excellent speed control. These types of motor use where requirement of power is more.[8]

![Shunt wound DC motor](Fig 1.2)

**Series-Wound.**

This motor has shunt field coils in series with the armature as shown in Fig 1.3. So current is dependent of one another, this motor provide high torque applications because current increased in armature and stator when high load is applied to it. This motor does not provide excellent speed control.[8]

![Series wound DC motor](Fig 1.3)
**Compound-Wound.**

This type of motor is a combination of a shunt wound and series wound motors as shown in Fig1.4. This motor provide the properties of both motors, it provide high torque as well as well speed control as compared to series-wound and shunt wound.

![Compound-wound DC motor](image)

**[8]Fig 1.4 Compound-wound DC motor**

**Brushless.**

Brushless motor is a type of synchronous electric motor run on DC current with electronic communication system. This motor not have commentator or brushes like mechanical parts. Current to torque and voltage to rpm shown linear relationship. This motor have rotating permanent magnet in rotor and its stator have no moving permanent magnates. The DC current given to motor is converted into AC current by a controller. This simple design removes the difficulties of transferring power to the rotor. Long life time, little maintenance and high efficiency are its advantages while high initial costs and more complicated controller are its disadvantages. This motor is similar to AC motor and required a external communication to generate torque. This motor have two subtype motors,

1. Stepper motor.
2. Reluctance motor.

Stepper motor is also its type which is normally run by controllers and use in highly precise CNC machines and it have many poles on stator.
Reluctance motor do not have permanent magnates. Its pole are pulled to alignment by moving stator drive.[5]
Key characteristics of BLDC motor.

- In this motor heat is produced in the stator and it is very easy to remove.
- It is more lighter as its rotor have permanent magnetic vs. coil so it is easy to start or stop.
- Linear torque and current relationship is smooth.
- For a specific speed this motor is simple to design at low cost.
- It is clean, fast, and efficient.

[3] Fig 1.5 Brushless DC motor

Fig 1.5 shown that BLDC motor have permanent magnet rotor which is fixed while its stator moves, in this case it is not needed to replace brushes so this motor is cost effective and low maintenance costs. This motor not operate on DC power, instead its rotor have permanent magnets, its stator have winding and communication that works electronically.[5]
1.2 DC Motion.

The motor moving process depend upon the electromagnetism. A current carrying coil have magnetic field so when electric current pass from a coil in the presence of magnetic field, then the magnetic force produced a torque which is directly proportional to the applied current in the conductor. This in turns produced rotational motion.[4]

Magnetic force $F=ILB$ acts as perpendicular to both wire and magnetic field.

$F = \text{Force}$

$L = \text{Length}$

$B = \text{Magnetic field}$

$I = \text{current}$
If the field are uniformly vertical then the angle between $B$ and $I$ is $90^\circ$ then according to right hand rule direction of force is shown in figure 1.7 we found direction of force. These two forces are equal and opposite so these forces produced torque.

![Diagram of DC motor](image)

Fig 1.7  DC motor

The function of commutator is to reverse the current in every half revolution so torque produced in the coil in the same direction. The direction of magnetic field is from north to south and the turning effect is proportional to the magnetic field.[4]

The motor is designed to produced rotational motion using magnetic interaction between coil and external magnetic field. Magnetic force are from north to south in this case and current direction is out from the coil then force will be in left direction and it tend to move the armature of motor. A motor have many poles so this effect continues and rotational movement achieved at the shaft of motor connected by armature. The speed is due to applied voltage while the torque is due to the current applied to the motor.
1.3 Why need to control motor.

The purpose to control the speed of motor is because there are many application in which user needs to change the speed of motor to get certain tasks. In robotics speed control is very important because only robots can work properly if motors speed is controlled in precise way. One example is CNC machine in which 1mm of error cannot be compromised so dc motor in such case provides exact speed control. The speed control normally done by Feedback speed controllers or closed loop speed controllers. So it is oblivious that without speed controllers we cannot got our goal in robotic and industrial application.

1.4 Speed controllers.

The speed of dc motor is proportional to supply voltage, if voltage is reduced it reduced its speed, but in the case of a battery whose voltage is constant then how to control speed of motor. The answer is speed controller. These controller works by sending varying voltages provide to motor. The speed controller on and off voltage supply very quickly. The switching frequency is so fast that it produce a average voltage given to motor. The process is called pulse width modulation. There are many types of controller, each one have its own advantages and disadvantages, so in this project three controllers are used which are.

1. **PID controller.**  
2. **Pole placement controller.**  
3. **Fuzzy logic controllers.**
1.5 Why these three controllers.

PID controllers used in large quantities for the choice when controller needed a closed loop. PID provide the designer a huge options if designer wants to change the dynamics of system. Almost in any industry 90% controllers are PI controllers while the rest 10% are PID and other controllers. If everything goes in right direction then designer can get following advantages. P part reduces step time to reach to desired position. This part removes the steady state error. Integral part remove the overshot got by P.D part used to reach at steady state error but if system is noisy then D part not used.

Pole placement controller provides excellent control as designer can adjust the time to reach on the reference point by adjusting poles. This controller use most straight forward design. It starts with assumption that what controller wants to do. From the assumption symbolic characteristic equation is formed, and poles determined which leads to design overshooting rise time. In this controller mostly equation is 2nd order and most equation have more that 2 poles. By this way equation can be formed and model is formulated as discrete transfer function. Disturbance model is not used when designer use pole placement design. Integrator design is used to drive steady state error to 0. In pole placement poles are placed in closed loop transfer function in reasonable position.

[9] Fig 1.8 Feedback control system
**Fuzzy controller** is a innovative technology that modifies the design of systems with engineering expertise. Fuzzy logic use human knowledge to implement a system. It is more effective than PID controller as its reach to its reference level in less time. It is mostly use in that system in which there is no mathematical equations for handling system. Common sense, human thinking and judgment are fuzzy rules. It help engineers to solve non linear control problems. It mathematically emulates human knowledge for intelligent control system and complex application.

### 1.6 Objective of thesis work

The objective is to control the speed of DC series wound motor which usually use in electric trains, self starting of engines, elevators and complex industrial mixing process. The selection of series wound motor is because it provide excellent torque load properties to the system which other motor of this class cannot provide. Also if devices are portable then the only option is to use DC motor as most portable devices DC power source is available and it make less expensive to the device if DC to AC convertor not used.

### 1.7 Limitation.

In this thesis DC motor used is series wound Dc motor as it provide good torque load characterizes, and usually this motor use in railway engines and self starting of engines. This motor usually operates at 10V and its speed is limited to 196 rad/sec. This speed is enough to start any engine and normally and it also be used in any robotic application. This control technique can be applied to any motor. But in this case reference speed cannot be exceeded than 196.
Chapter 2

Background work.

For fuzzy controller and its comparison with PID controller

Seda Aydemir & serkan Sezen work.

Summary.

Dc motors are used in so many applications like electric trains and robotic manipulators. In order to realize these systems Dc motors should be controlled. In this paper Dc motor speed controlled by fuzzy controller using Matlab. DC motor provides high performance because the speed of the motor can be adjusted in wide range. Fuzzy logic is best to use where system is non linear and difficult to calculate mathematical equations. As known fuzzy logic control based on human experience and linguistic definitions of the system. In this project dc motor speed was controlled by PID controller and fuzzy controller and then a comparison is made between the controllers. [10]

Fuzzy logic controller uses the error value and changing of this error value for our task in here. And the goal with using FLC is to minimize the error. The error here is the difference between reference speed and available speed of the motor. The output of fuzzy controller is created by rules which are composed using these two inputs with linguistic definitions of the system. The general rule is that if error value is negative and change of error value is negative then output of controller is positive. Or if error value is positive and change of error value is positive then output of controller is negative. But in this article the membership functions of inputs and output are not only positive and negative. Two input are error (e) and change error(ce) have trimf
membership function negativelarge, negativemedium, negativesmall, Zero, positivesmall, positivemedium and positivelarge in order to get desired value with high accuracy. As mentioned before fuzzy logic control is based on linguistic definition of the system. These membership functions convert the crisp values such as error value and changing of error value into linguistic forms. So two input each with seven membership function will make 7 multiply 7 = 49 rules in order to get accurate value and efficiency. The gain coefficient K1e, K2ce, K3ca were used in order to scale the desired inputs and outputs within the range of membership function’s range. According to the specification provided in fuzzy controller as membership function range is between -1 to 1. Membership functions are from -1 to 1 range so values of K calculated according to that range.[10]

Electrical and mechanical equations were gotten by Kirchhoff’s voltage law and torque law for Simulink model. Then to make its fuzzy controller we have to design a chopper because chopper can drive the motor by reducing overshoot and fluctuation of output speed. In other words chopper gives the average of our output so small undesired impulses removed by using pulse width modulation. At 2KHZ this chopper is designed for reference speed 200 radians per second. Experiment results were very good because the angular velocity of the Dc motor was the same as the reference velocity value. After a while some torque was applied to the motor and the angular velocity has decreased immediately and it settled back to the reference velocity value instantly. This shows that FLC is working in a good way. One illustration can be made think that you are going with electric train and suddenly a hill (like torque) appeared on your way. If the train is controlled with a good FL Controller you will cross the hill with same speed if not your speed will continue to decrease along the hill. Then in compare with PI controller we evaluated the results of the authors that fuzzy controller is more effective and has less simulation time than PI controller. Fuzzy controller is more sensitive against disturbances and has high response time as compared to PI controller. Also steady state error of the Fuzzy controlled system is small compare to PID.[10]
Yodyium and Mo-Yuen work.

Fuzzy logic microcontroller implementation for DC motor speed control.[11]

Summery.

In this paper a another method is provide to implement Dc motor speed control by using fuzzy logic microcontroller. The design of this motor is same to other motor and results are shown both on loaded and unloaded condition. This controller is cheap as it require only small amount of components and easily improved to adaptive fuzzy controller. This controller provide good performance and compact size and low cost.[11]

In this paper the error range of controller is between 1000 to -1000 rad/sec. The change of error is taken +/- 5.5 rad/sec and these values are taken by PI controller. The terms small and big is used to quantize inputs and outputs. Seven fuzzy sets are obtained by applying seven linguistic terms and one crisp value moved to partially in many fuzzy sets. The shapes of membership functions are rectangular and trapezoidal as these shapes exactly describe designer shape. Then these ideas are converted into Motorola microcontroller, using PWM input condition circuit is constructed running at +/-15 voltages.[11]

The software of program is composed of two parts. For input the reference speed keypad is used, the program calculates the speed from A/D unit and computes E and CE to apply fuzzy rules. The second part is composed of two timer interrupts which performs counter.[11]

FLC tested on different speed on both loaded and unloaded conditions. FLC calculate speed in no load condition by changing duty cycle of PWM signal. Then this motor rum in loaded condition and speed remain constant as controller increase duty cycle of voltage.[11]
Guoshing huang, shoucheng Lee work.

PC based PID speed control in DC motor[12]

Summary.

This paper gives a good view of speed control of DC motor using LabView-aided PID controller and for analysis software VisSim is used. This paper provides a very good idea about the modeling of equations of DC motors and conversion to transfer function. The design to drive the dc motor and its feedback signal done by 8051 chip modules which produce rotational signal it it shows from SEG7 display system. LabView-aided PID controller, parameters are adjusted to show output. The results are quiet similar to the theory of Pid controller so it provide excellent way to control of motor for the engineers. By using Pid controller and setting the values of Kp,Ki,Kd provide the best response to satisfied the system, also under different condition the system is checked. So finally actual control parameters are taken into VisSim, and results are quiet similar to actual results.[12]

The beauty of this system is that it provide output speed in real time in order to obtain system responses of PID controller. There is a card used DAQ has the capabilities of data scratch and transmission and it matches LabView virtual instrument characteristic and analysis. This is cost effective system for checking the response of DC motors. In short this system provide the indoor office control of a industry of outdoor systems. [12]

Intelligent PI controller for speed control of DC motor.[13]

**Summary.**

In this paper speed control of separately excited Dc motor is done by using artificial neural networks. Designer actually enhance the tracking performance of Dc motor as compare to conventional PI controller. ANN not only control the speed but also control the current through controller. This make it suitable for large number of industrial use. [13]

To make PI controller a intelligent controller, NARMA L-2 controller is used to enhance the performance of DC motor. The basic idea is this project is to convert the system non linearties into linear dynamics by canceling the non linearties.[13]

ANN use 2 two controllers one for speed and one for current. Also designer use one controller to control both current and speed. And the analysis show that in order to make system perfect current control performs excellent role. Current controller reduced the steady state error and selting time. Simulation done on different situation and different reference speed to check the controller, also a comparison is done with use of a single ANN controller for current control and then use another controller for speed and concluded that by using speed control and current control at the same time results are better than one controller or conventional PI controller. By using ANN controller following advantage can achieved.[13]

If the Torque load is constant then the controller shows the ability of the drive to instantaneously reject the perturbation.

If use cascade structure then the design of controller is simple for independent control of torque and flux. So best results and simplicity in the system can make ANN controller best for industrial work.[13]
Chapter 3

Mathematical model of PMDC motor.

The goal is to develop the mathematical model in sense that it related the applied voltage to the armature to the velocity of motor. By considering electrical and mechanical characteristics of the system two balance equations are developed.

3.1 Electrical Equation.

The electrical circuit of motor is shown in fig 3.1 represent a series wound dc motor. The selection on series wound motor is better because this type of motor provide better performance on heavy torque loads. Fig 3.1 shows a voltage source(Va) across the coil of armature and the electrical equivalent of the armature and inductance(La) and resistance(Ra) are in series with induced voltage(Vc) whose oppose the voltage source. The induced voltage is generated due to the electrical coils rotating in the fixed flux lines of the permanent magnets. This voltage is often called back emf or electromotive force.

[15]Fig 3.1 electrical representation of dc motor
Back emf usually refers the voltage generated by the spinning of rotor to estimate the speed of motor rotation. In other words the voltage produced in the conductor that tends to neutralize the present voltage. For example if magnetic field lines cut by conductor, then a voltage is generated in that conductor which causes the current to flow in one direction. At the same time another voltage is appeared which try to remove this effect and it forces the electron to move in opposite direction. This emf is proportional to the velocity of magnetic field which in other words is proportional to the relative motion between them.[14]

By using Kirchoff's voltage law a differential equation can be found from the circuit. This law states that the sum of all the voltages in a loop must equal to zero.

\[ \text{..............................1.1} \]

According to ohm's law voltage across the resistor given as

\[ \text{where } I_a \text{ is the armature current. The voltage across the inductor can be written as} \]

\[ \text{This voltage is proportional to the change of current in the coil with respect to time. Where } L_a \text{ is the inductance of armature coil and back emf expressed as} \]

\[ K_v \text{ is the velocity constant and it determined the flux density of permanent magnets and } W_a \text{ is the rotational velocity of the armature. It is the number of turns of armature winding which determines the speed of motor.}[15] \]

Substituting these values in equation 1.1 gives the differential electrical equation

\[ \text{..........................1.2} \]
3.2 Mechanical equation.

The turning effect of a force called torque or a force which tries to rotate about its own axis is called torque. Here different torques are effecting the motors and in order to get mechanical equation take all the torques equal to zero.[15]

\[ \text{............................. 1.3} \]

\[ \begin{align*}
    \text{Te} &= \text{Electromagnetic torque} \\
    &= \text{Torque due to rotational acceleration of the rotor} \\
    \text{Tw} &= \text{Torque due to velocity of rotor} \\
    &= \text{Torque due to mechanical load}
\end{align*} \]

The current which passes through the armature windings is proportional to the electromagnetic torque which can be written as

\[ \text{Like velocity constant } K_t \text{ is torque constant and it depend upon the flux density of fix magnets, reluctance of iron core and number of turns in the armature winding. It can be written as} \]

\[ \text{J = inertia of rotor and mechanical rotor} \]

The torque which is associated with velocity written as

\[ \text{B = damping coefficient associated with the mechanical rotational system} \]
Substituting these values in equation 1.3 gives the differential mechanical equation.

\[ ..................1.4 \]

3.3 State Space Representation.

By using equation 1.2 and 1.4 armature current and angular velocity can be measured which describe the dc motor system.

\[ ..................1.5 \]

\[ ..................1.6 \]

Converting these equations into state space form

\[ \]

\[ = \]

\[ = \]

From state space equation it shows that the output of state space equation is angular speed and current while at input Voltage and Torque load has been given.
3.4 Transfer Function Block Diagram.

The transfer function can be developed by equation 1.5 and 1.6. By taking laplace transform following equation are formed.

\[ = - (s) - - \] ..................................1.7

\[ - - - - (s) \] ..................................1.8

The final equations can be expressed as

\[ \] ..................................1.9

\[ \] ..................................1.10

The equation 1.9 and 1.10 now can be use in order to obtain block diagram

[15] Fig.3.2 Block diagram representation
The block diagram can be simplified by considering torque load equal to zero as in case of motor some time torque is constant and in some cases torque is changing so it is up to user whether is put changing or torque or not. In this project simulation has done on both torque and change of torque. If change of TL is zero then there is no need to show in block diagram. The transfer function is obtained by considering TL is constant which reduced the algebra and it is possible that in many cases TL is constant and it is not changing. But in order to make this project more generic change in TL is introduced.

\[ G(s) = \frac{G_1(s)}{1 + G_1(s)H(s)} \]

[15]Fig.3.3 Overall transfer function for the dc motor.
By using equation 1.5 and 1.6 we can make a simulink model of PMDC motor.

![Simulink diagram of PMDC motor](image)

Fig 3.4 show the block diagram representation of equations of PMDC motor

In order to perform DC motor speed control by PID controller first we have to run the system in order to get the response of system. Any equation can be used to measure the speed or current on motor at specific values whether the above equations 1.5,1.6 or state space representation or transfer function representation. The output will be the same in all cases. In order to perform transfer function representation the simulink diagram is shown in Fig 4.4.

![Simulink diagram of transfer function](image)

Fig 3.5 Transfer function representation of PMDC motor.

Initially we assume torque load(TL) is zero so the output graph of speed and current is shown in Fig 4.5
Fig 3.6 DC motor put at no load.

From the Fig 4.5 it is shown that motor maximum speed is 190 rad/sec and maximum current initially it take that is 18 amperes after that working current is 0.4 amps. Motor reach to maximum speed in almost 0.3 seconds. If we convert the speed into RPM then according to formula

\[
\text{RPM} = \frac{196 \times 60}{2\pi}
\]

\[
\text{RPM} = 1872
\]

The values for this motor is shown in appendix. The same condition by applying TL the output of motor is shown in Fig 4.6
Fig 3.7 DC motor put at torque load.

From Fig 3.7 It is observed that at torque load speed of motor decreased and this is not desired in industrial and robotic application. In order to overcome this fault speed controllers are introduced which will control the speed of motor by increasing the voltages. Three controllers are introduced in this thesis. So next task is to develop the controllers.
Chapter 4

4.1 Fuzzy Logic Controller.

Fuzzy logic is a type of multi valued logic. It deals with approximate reasoning rather than precise. Fuzzy logic derived from fuzzy set theory. Fuzzy logic was first proposed by Lotfi Zadeh in 1965. Fuzzy logic has currently used in control theory, artificial intelligence systems specially to control complex aircraft engines and control surfaces, helicopter control, missile guidance, automatic transmission, wheel slip control, auto focus cameras and washing machines, railway engines for smoother drive and fuel consumption and many industrial processes. Fuzzy logic provide better results if we compared it with PID controller.

Fuzzy set of theory represent the human reasoning with knowledge that is almost impossible to represent in quantitative measures or for that control plants that are hard to control or ill defined. Fuzzy inference system model the system using if-then rules. Fuzzy set theory proposed the membership function at range of numbers $[0, 1]$ or False or true membership function. This theory provides the mathematical strength to check the uncertainties connected with human thinking or reasoning. Fuzzy logic is suitable for a model that is hard to control or non linear models. This system also provides control over MIMO systems and also allow decision making with incomplete information. Human reasoning can also be known as multi valued ‘imprecise’. [21]

Advantages of fuzzy control.

- To control any system mathematical model is not required in fuzzy controller.
- Human knowledge and experience can be implemented using linguistic rules.
- Non linear plants can be controlled
- It can also control fast processes.
**Disadvantages of fuzzy control.**

- Human knowledge is often incomplete and episodic as compared to systematic way.
- If the model is not known then it is impossible to achieve the stability of the controller system.
- Sometime rules are mismatched and non coherent.
- In complex operation fuzzification and defuzzification take long time.

**Fuzzy expert systems.**

There are two famous type of system currently used in fuzzy logic

- Mamdani fuzzy inference
- Sugeno fuzzy inference

**4.2 Mamdani fuzzy inference.**

The most common method is used currently is fuzzy inference system. In 1975, Professor Ebrahim Mamdani of London University introduced first time fuzzy systems to control a steam engine and boiler combination. He applied a set of fuzzy rules experienced human operators. The mamdani system usually done in four steps.[21]

- Fuzzification of the inputs
- Rule evaluation.
- Aggregation of the rules.
- Defuzzification.

Fuzzification convert input data to degree of membership functions. In this process data is matched with condition of rules and and determined how well data is matched with rule at particular instance. Thus a degree of membership function is developed.
Then in Rule-base block rules are written according to system requirement. Fuzzy controller work on both MIMO and SISO. In case of Dc motor there are two input variables Error and Change in error are selected. This system is limited to single loop control. Usually rules are in if, and, then form. In inference engine aggregation is done in which degree of fulfillment is calculated of the condition specifies by a rule. In activation min of two aggregated value is selected and only thickened part of singleton are activated. Its multiplication result in slighter smooth control. Then all activated conclusions are accumulated using max operation.

Defuzzification block converted resulting fuzzy set into a number that is sent to the system and this number is actually the control signal. There are seven defuzzification methods.

- Centre of gravity(COG)
- Centre of gravity method for singletons(COGS)
- Bisector of area(BOA)
- Mean of Maxima(MOM)
- Left most maximum(LM),and right most maximum(RM)
4.3 Sugeno Fuzzy Inference.

Mamdani style is not computationally efficient as it find the centroid of two dimensional shape by integration of carrying function. Michio Sugeno proposed a new method to use single spike, a singleton, as a membership function inputs. Its mean fuzzy set is at unity point at one particular point on the universe of discourse and zero at remaining area. This system is almost same of Mamdani method but with the exception of consequent change and instead of fuzzy set it use a mathematical function as input variable.[21]

**Mamdani is better or Sugeno.**

In this project Mamdani method is selected because this method is widely accepted for capturing expert knowledge. Mamdani allow designer to describe his thinking or knowledge in more effective way. But this system sometime slows due to computational burden in defuzzification process.

Sugeno works well with optimization and adaptive techniques, that’s why it is computationally effective which make it attractive in control problems but in this case mathematical inputs are not available so Mamdani style is used to control the speed of motor.

4.4 Developing Fuzzy expert system.

Define linguistic variables and also specify the problem statement is the first step to make a mamdani controller.

There are three main linguistic variables:

1. Error
2. Change in error
3. output
Table 4.1 shows input variable Error

<table>
<thead>
<tr>
<th>Linguistic value</th>
<th>Notation</th>
<th>Numerical Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative large</td>
<td>NL</td>
<td>[-450 -300 -150]</td>
</tr>
<tr>
<td>Negative Zero</td>
<td>NZ</td>
<td>[-301.5 -151.5 -1.567]</td>
</tr>
<tr>
<td>Zero</td>
<td>ZZ</td>
<td>[-150 0 150]</td>
</tr>
<tr>
<td>Positive Zero</td>
<td>PZ</td>
<td>[0 150 300]</td>
</tr>
<tr>
<td>Positive Large</td>
<td>PL</td>
<td>[150 300 450]</td>
</tr>
</tbody>
</table>

Table 1.1 shows the ranges of Linguistic variable Error and its member functions. Five membership function are used and ranges them from between -300 to 300 according to Fig 4.5 which shows that the maximum speed of motor is 196 rad/sec so this range is suitable for DC motor control speed. Five membership functions provide best results for motor and trimf shape.

Fig 4.2 shows input variable error

is useful for measuring speed ranges.
Linguistic variable Change in Error

<table>
<thead>
<tr>
<th>Linguistic value</th>
<th>Notation</th>
<th>Numerical Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative large</td>
<td>NL</td>
<td>[-0.45, -0.3, -0.15]</td>
</tr>
<tr>
<td>Negative Zero</td>
<td>NZ</td>
<td>[-0.3, -0.15, 0]</td>
</tr>
<tr>
<td>Zero</td>
<td>ZZ</td>
<td>[-0.15, 0, 0.15]</td>
</tr>
<tr>
<td>Positive Zero</td>
<td>PZ</td>
<td>[0, 0.15, 0.3]</td>
</tr>
<tr>
<td>Positive Large</td>
<td>PL</td>
<td>[0.1485, 0.2985, 0.4485]</td>
</tr>
</tbody>
</table>

Table 4.2 shows input variable Change in Error

Table 6.2 show the range of membership function of linguistic variable change in error and its range is kept from 0.3 to -0.3

c_e = 300/1000 = 0.3

The remaining membership functions and notations are same as other input error.

Fig 4.3 shows input variable change in error
Figure 6.3 shows the membership functions and their ranges. As two inputs are selected so now to decide the output of controller output.

<table>
<thead>
<tr>
<th>Linguistic value</th>
<th>Notation</th>
<th>Numerical Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative large</td>
<td>NL</td>
<td>[-450 -300 -150]</td>
</tr>
<tr>
<td>Negative Zero</td>
<td>NZ</td>
<td>[-301.5 -151.5 -1.567]</td>
</tr>
<tr>
<td>Zero</td>
<td>ZZ</td>
<td>[-150 0 150]</td>
</tr>
<tr>
<td>Positive Zero</td>
<td>PZ</td>
<td>[0 150 300]</td>
</tr>
<tr>
<td>Positive Large</td>
<td>PL</td>
<td>[150 300 450]</td>
</tr>
</tbody>
</table>

Table 4.3 shows output variable output speed

The speed variable is between -300 to 300 and trimf shape is used to represent five membership function. Membership function ranges are shown in table 6.3.

Fig 4.4 shows output variable speed
Rules.

The general rules for Dc motor speed control is that if motor speed is less than desired speed then speed up the motor and if motor speed is more than reference speed then slows it speed. There are nine possible conditions which motor can be seen in table 6.4 and nine possible reign is selected from Fig 6.5. from which 25 possible rules in fuzzy controller are written. In the process of producing necessary output voltage with Fuzzy Logic Controller the speed error should be minimised. The bigger speed error causes the bigger controller input. In addition changing of the error plays an important role to define controller input.

In reign one according to Fig 6.5 error is in positive reign while error direction is going to negative reign, so output of controller must be positive. In this case speed still not touch to the reference point so increase in speed is required which is directly proportional to the voltage of controller so increase in voltage help the motor to got reference speed.
In third reign, if error value is negative large and change of error value is negative large than output will be negative large. This condition is corresponding to the reign 3 interval in Fig 6.5 but this result gives us crisp value for fuzzy control this crisp value should be converted into linguistic form.

<table>
<thead>
<tr>
<th>Reign</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>+</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>0</td>
</tr>
<tr>
<td>Change in error</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>du(output)</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.4 shows dynamic signal analysis

So the 25 corresponding rules are shown in table 6.5 and these rules are derived from the decision of output in table 6.4

<table>
<thead>
<tr>
<th>Change in error ‘de’</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘du’(output)</td>
</tr>
<tr>
<td>NL</td>
</tr>
<tr>
<td>NZ</td>
</tr>
<tr>
<td>ZZ</td>
</tr>
<tr>
<td>PZ</td>
</tr>
<tr>
<td>PL</td>
</tr>
</tbody>
</table>

Table 4.5 Rules for DC motor speed control

- If error is NL and change in error is NL then output is NL
- If error is NL and change in error is NZ then output is NL
- If error is NL and change in error is ZZ then output is NZ

Fig 6.6 shows the output of controller plotted against the rules described in table 6.5.
Rules behavior can be checked by changing of error or change in error point. From the surface an idea can be built that in a certain case what will be the output of controller.

Fig 4.6 rules surface
Fig 4.7 shows the surface of controller

The surface in fig 6.7 shows that motor touch the reference speed in smooth way. According to this design motor can reach to the maximum speed of 300 rad/sec.
DC chopper.

A chopper is a device that convert fixed and unregulated DC voltage into variable regulated, controllable DC voltage to its desired level. In DC motor speed control DC chopper used as a driver circuit which change the average value of load voltage by switching the power switch as BJT. Chopper use PWM in which a switch connects source to the load and disconnects the load from the source at a fast speed. The switching frequency is so fast that motor receive a average voltage. Function of chopper is same like a transformer but it actually work on DC voltage and steps ups or down it. Choppers provide smooth control, high efficiency, fast response and regeneration. As from the name in chopper output voltage is chopped up.[23][24]

![Diagram of chopper](image)

[24] Fig 4.8 operating principle of chopper

The average output voltage can be calculated from the formula

\[
V_{avg} = \frac{V}{2} \cdot \frac{t_{on}}{T}
\]

For DC motor one quadrant driver is used and load voltage and current is always positive. A diode is connected across the load to prevent the current go to negative reign.
Basically dc chopper compare the signal or voltage coming from the fuzzy controller with its own voltage and then make a average voltage signal to motor. The duty cycle of chopper is determined by fuzzy controller. The average output voltage is obtained by the integration of chopped input voltage over the chopping period until the desired set point is obtained. The average output voltage is vary from 0 to 1. [23]
Simulink model of DC motor using fuzzy controller.

Fig 4.11 Simulink model of DC motor using fuzzy controller

Fig 6.11 shows the simulink model of DC motor speed control through fuzzy controller.

In fuzzy controller error and change in error is measure by following formulas

\[
\text{Error (ek)} = W_{\text{ref}} - w_m \\
\text{Change in error (ce)} = e(k) - e(k-1)
\]

Back emf provide the error while change in error measured that error goes in negative direction or positive direction. In simulink model reference speed is selected 150 rad/sec. In fig 6.12 shows the output of DC motor.
In Fig 6.12 desired speed is 150 rad/sec. Motor achieve this speed in less than 0.2 second and torque load is applied at 0.5 second but it did not have any effect on the speed and it remain constant. This is the advantage of fuzzy controller that in pole placement and PID controller speed reduced for minor time and then goes to reference point but in this case speed not reduced.
Also in PID and pole placement controller in the start a undesired overshoot in speed is measured where in fuzzy controller this overshoot not observed. So it can be say that fuzzy controller is the best controller. Some more plots are shown in Fig 6.13

Fig 4.13 chopper voltage, and fuzzy controller plots

Fig 6.13 shows that at time0.5 second when load is applied to fuzzy controller then chopper provide more volts to the motor in order to maintain its speed. Fuzzy controller output is also very smooth and shows a fast step response.
Fig 4.14 error, and change in error plots

Fig 6.14 shows that error start reducing from value of 150 and goes to zero. This whole process take less than 0.1 second. As error value touches the zero reference speed, its mean reference speed achieved. Change in error shows that when change in error is positive and when it is negative.
Chapter 5

5.1 Pole Placement Controller.

Pole placement is one of the best and useful control techniques which is currently use solve many control problems for different engineering fields. Pole placement controller is depend upon a model of considered processes in which model is executed as a discrete transfer function \( H(z) \). In order to design controller a assumption is made the controller work in order to control a plant and from this assumption a equation is formed. Then desired closed loop poles selected. The final equation will have more than two poles and equation is 2nd order equation. To determine the coefficients algebra is used to get desired close loop control. Then a integrator is added to remove steady-state error. Addition of integrator and one more pole in the characteristic equation. In pole placement controller the reference and the process output effect the control signal individually [18][19]

\[
U(z) = \text{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots5.1}
\]

The above equation can also be written as:

\[
C(z)U(z) = T(z)R(z) - D(z)Y(z)\ldots\ldots5.2
\]

Where \( C, T, D \) are the polynomials in backward shift transform
P(z) = A(z)C(z) + B(z)D(z)........5.4

The unknown polynomials C(z) and D(z) are calculated from variables B(z) and A(z) and P(z). This problem was first solved by Diophantus. He found the order of polynomials as,

\[ \begin{align*}
\frac{1}{z} &= -1 \\
\frac{1}{z} &= -1 \\
\frac{1}{z} &= + -1 \\
\frac{1}{z} &= 1 + 1 -1 + \cdots + - \\
\frac{1}{z} &= 0 + 1 -1 + \cdots + - \\
\end{align*} \]

The gain factor Kr

\[ = (1) \quad (1) \]
5.2 The Diophantine Equation.

Previous section discussion give an idea into the design problem. It is observe that the Diophantine equation played a Key role and now analyze this equation. The fundamental mathematical problem is to study the properties of the polynomial equation.

\[ A + B = C \]

Where \( A, B, \) and \( C \) are known polynomials \( X \) and \( Y \) are unknown polynomials.

This is a famous problem in elementary algebra. Diophantine equation is named after Diophantus (~ A.D. 300). He was one of the original developer of algebra. It has also many other names in literature, the *Aryabhatta's identity* or the *Bezout identity*.

Now calculating the transfer function shown in Fig 5.3 and appendix 1.1

\[
\text{>> ali = tf(num,den)}
\]

Transfer function:

1.333e005

-------------------
\[ s^2 + 333.7 \, s + 6800 \]

\[
\text{>> [num,dend]=c2d(ali,0.1)}
\]

Transfer function:

17.22 \( z + 0.1665 \)

--------------------------
$$z^2 - 0.113\, z + 3.207\times10^{-15}$$

Sampling time: 0.1

The same result can be obtained by using state space equation and available in appendix 1.4

Fig 5.2 simulink model of pole placement controller

Instead on manual calculation for the parameters a file copp_new is used for calculation of pole placement controller parameters C,D and gain. So initially system is checked without disturbance and Fig 5.3 shows the output at poles 0.1 and reference point 50
Fig 5.3 pole placement controller at poles 0.1

Fig 5.3 shows the speed of dc motor motor gain at desired point at 100 seconds. Now speed is checked at poles 0.5
Fig 5.4 pole placement controller at poles 0.5

If poles are far from the origin then the response of system is slow while if poles are near to the origin then the system response is faster. In Fig 5.3 all poles are at 0.1 that’s why response of system is faster while in Fig 5.4 poles located at 0.5 so the response of the system become slow. For poles = 0 system becomes dead beat controller and as poles are located at origin so the system response is the fastest in such case. The sample interval is selected is 0.1 because this sample interval give the maximum response because signal is stable for each 0.1 second.
5.3 Pole placement controller with disturbance.

Disturbance or load disturbance is added in the simulink in order to check the controller on load disturbance or in noise environment. Fig 5.7 shows the controller in the environment of disturbances at poles 0.
Fig 5.7 Dead beat controller showing fastest response in disturbance

Fig 5.7 shows the effect of disturbance on the controller and these disturbance deviate the speed of controller which is not good for a smart engineering design. For all result that is seen that outputs fit to the value of reference signals after a while. This means that these results validate that calculations are well for pole placement controller. As this is dead beat controller so the affect of disturbances not shown well so if poles are placed at 0.2 then according to figure 5.9 it is hard to reach the reference point and motor cannot run properly in such case.
So load disturbance can cause a constant decrease in amplitude in the speed of motor. So in order to remove this problem another pole is introduced in the controller and this pole with integrator remove the effect load disturbances.
As discussed above that pole placement controller is not good for handling loads or load disturbance so in Fig 5.9 at 250 second load is applied to the system and it decrease the speed of motor. Integrator is added to handle this problem.
5.4 Integrator design.

Integrator is added by adding C polynomial to contain an integrator, and this integrator remove the error and take that error to zero. The design is same as pole placement controller but with the addition of integrator. The controller scheme is shown in fig 5.10 while the process of model changed is as under,

\[ = \ldots \ldots \ldots 5.4 \]

So the transfer function can be formed as

\[ \ldots \ldots \ldots \ldots 5.5 \]

The characteristic equation is formed as

\[ P(z) = A'(z)C'(z) + B(z)D(z) \ldots \ldots 5.6 \]

From equation 5.6 we can get two polynomial C(z) and D(z) which is calculated from polynomial B,A and P.

\[ '=' -1 \]
\[ = ' -1 \]
\[ = ' + -1 \]

Kr is formed from the transfer function

\[ Kr = \ldots \ldots \ldots \ldots \]
So by using discrete transfer function obtained in previous section so first we have to multiply $A(z)$ by $(1-\ldots)$ and the calculation are as under.

\[ = \quad = \]

\[ = \quad = \]

\[A'(z) = (1-\ldots)(\ldots)\]

5.5 Simulink model of pole placement controller with integrator.

Fig 5.10 simulink model of pole placement controller with integrator
The model in figure 5.10 is the simulink model of pole placement controller with integrator in the presence of disturbance. As discussed pervious that integrator helped the system to remove the effects of disturbance. Figure 5.11 shows the dead beat controller output where disturbance effect completely reduced.

Fig 5.11 Pole placement integrator design at poles 0

When poles are at 0.2
From Fig 5.2 it is clear that integrator design remove the effect of disturbances or reduced it because in Fig 5.8 poles are located at 0.2 but speed output is not good it have huge oscillations while after adding integrator at same poles location disturbances considerably reduced and goal is achieved. If poles are near to origin then motor response is faster. So lets check this system on poles 0.4.
Fig 5.13 effect of disturbance at poles 0.4

System response is slow but speed reach to its reference point and disturbance effect is reduced after adding integrator. It is acceptable because before adding integrator speed have huge oscillation that the same poles location.
Chapter 6
6.1 PID controller.

A **proportional–integral–derivative controller** (PID controller) is widely used in industrial control systems. It is a generic control loop feedback mechanism and used as feedback controller. PID working principle is that it calculates an error value from the processed measured value and the desired reference point. The work of controller is to minimize the error by changing in the inputs of the system. If the system is not clearly known then applying PID controller provide the best results if it is tuned properly by keeping parameters of the system according to the nature of system.[16]

[16] Fig 6.1 Block diagram of PID controller

The PID measurement depend upon three parameters which is called the proportional, the integral and derivative part which is called P, I and D part.

P determine the reaction to current error,
I determine reaction to the sum of recently appeared errors,
D Determine reaction according to the rate of error changing.

The sum of all three parts contribute the control mechanism such as speed control of a motor in which P value depends upon current error, I on the accumulation of previous error and D predict future error based on the current rate of change.

As derivative action is sensitive to noise so mostly the controllers are PI controller rather than PID as it is not possible a system without disturbances. Integral part helps the system to reach onto its target value while P part increase overshoot.

The P term take the output to proportional of error value. Its response can be adjusted by multiplying the error by a constant Kp which is called proportional gain. If proportional gain is large then it creates a high overshoot which unstable the system, while a small output change make a small control action.[16]

![Fig 6.2 Three values of Kp and Ki are shown for PI controller behavior](image)

The integral term contribute error and duration of error proportionally. Error sum gives offset that corrected previously. The calculated error is multiplied by integral gain and then added to controller output. It finally reduced the steady state error.
6.2 PID tuning methods.

There are many methods of PID tuning. The most effective method is that in which a model is developed and select P,I and D values. Every methods have some advantages and disadvantages shown in table 3.1. which discussed latter.

<table>
<thead>
<tr>
<th>Method</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual Tuning</td>
<td>No math required. Online method.</td>
<td>Requires experienced personal.</td>
</tr>
<tr>
<td>Cohen-Coon</td>
<td>Good process models.</td>
<td></td>
</tr>
</tbody>
</table>

[16] Fig 6.3 showing tuning methods and their advantages and disadvantages

Fig 3.7 clearly show that when torque load is applied to motor it decrease its speed. This decrease of speed is not acceptable in industrial and robotic application. The decrease of speed is not good for the performance of any motor because if this motor is set in a railway engine then torque load decreased or increased often and this reduced speed is not in the favor of train. Also in paint industry this is very critical stage when it is required to mix of many chemicals in sequence then this speed decrease not provide the desired results. So in order to overcome this problem speed PID controller is introduced. PID controller needs tuning in order to work properly. Manual tuning is way of tuning in which controller is tuned by increasing its P value until output start oscillates by keeping I and D value to zero. Then start increasing I value but I value must be in optimum range as increase of I value to certain limit case instability. And then increase D value if required.
6.3 Tuning of PID controller by Ziegler–Nichols method.

As in this project the target is to control the speed so speed is send back for checking the system in closed loop and tuned PID controller. The method used for tuning is Ziegler–Nichols method. According to Ziegler–Nichols method

1. Run the controller by taking only P value.
2. Increase P value of the system until it self oscillating with constant amplitude.
3. Then take controller gain time period.

So at \( K_p = 6 \) motor output have constant amplitudes from graph time period of the graph can calculated which is \( K_i = 0.2 \)
Fig 6.5 shows tuning process for calculation of $\theta$ and $\phi$.

From Fig 4.8 following parameters are calculated:

$$= 0.43 - 0.41 = 0.02$$

### Ziegler–Nichols method

<table>
<thead>
<tr>
<th>Control Type</th>
<th>$K_p$</th>
<th>$K_i$</th>
<th>$K_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>$0.50K_u$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$PI$</td>
<td>$0.45K_u$ 1.2$K_p / P_u$</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>$PID$</td>
<td>$0.60K_u$ 2$K_p / P_u$ $K_pP_u / 8$</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

[16]Fig 6.6 Ziegler-Nichols method

So calculating the values of $K$ and $P$:

<table>
<thead>
<tr>
<th>Type of controller</th>
<th>$K$</th>
<th>$K_i$</th>
<th>$K_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$I$</td>
<td>2.7</td>
<td>0.0167</td>
<td>0</td>
</tr>
<tr>
<td>$D$</td>
<td>3.6</td>
<td>0.01</td>
<td>0.0025</td>
</tr>
</tbody>
</table>

Table 6.1 calculation of parameters of PID and PI controller
For PID controller.

\[ = 3.6[1 + \frac{+0.0025}{+0.009}] \]

So the values for PID controller are

P = 3.6
I = 360
D= 0.009

For PI controller.

In industries Instead of PID controller only PI controller is used mostly as D term increase noise in the system. As almost majority processes have noise in it so it is better to remove the D term and use only PI controller.

\[ = 2.7+2.7/0.167+0 \]

So the values for PI controller are

P = 2.7
I = 161
So if reference speed is 20 then by applying PID values output is shown in fig 4.9

Fig 6.7 output of Dc motor at no load

This result in fig 4.9 is acceptable but it is not good as speed have some overshoot which is not desired in actual motor so this system required more tuning to remove this over shoot. From past experience value of I reduced to fix this problem so after doing some manual tuning the following results are shown in Fig 4.10
Fig. 6.8 Tuned PID controller output of DC motor at no-load.

Fig. 4.10 shows some further manual tuning removes overshoot and the controller achieved desired speed without any undesired overshoot, and applying torque load will not affect the performance of the PID controller. Fig. 4.11 shows the PID controller working on torque load.

80
In Fig 4.11 torque load is applied at time 0.5 seconds and it minor effect the speed, reduces speed but speed again got its reference point back just in 0.1 second so PID controller solve the problem of DC motor that in any torque load its speed remains constant at the reference point.
Fig 6.10 Tuned PID controller output of at torque load

Tuned PID values

P = 3.6
I = 20
D = 0.009
Chapter 7

7.1 Comparison between PID and fuzzy controller.

PID controller is a simple controller and it provide easy and cheap solution in industrial environment. This controller tuning is very easy so currently majority industrial controllers are PID controllers. But the drawback of PID controller is that is only works on SISO systems while in industries in many control problems inputs are more than one so in such case more controller required in which it is obvious that system become costly. While fuzzy controller work on both SISO and MIMO systems and it provide better results than conventional controllers. Currently fuzzy controllers are used in satellites altitude control, space crafts and various NASA projects and military hardware’s.

In this thesis the theoretical hypothesis is proved by speed control of DC motor. And it is concluded that fuzzy controller output is more smooth, faster step response and no initial overshoot which was seen in PID controller. Fig 7.1 shows the comparison on two controllers.

![Fig 7.1 Comparison between PID and fuzzy controller](image)

In figure 7.1 speed output of both controllers are shown. Almost both controller reach to reference speed in 0.1 second but PID controller have overshoot initially which is undesired. This undesired overshoot leads to undesired results and negatively impact the DC motor. Also
this overshoot case the PID controller voltage to more peak than fuzzy controller voltage. If input voltages are fixed to a certain level then perhaps controller requirement of voltage is more that the certain level then this is not a good sign for PID controller. In case of fuzzy controller output speed graph is a smooth output and no overshoot in the initial step so this controller is more preferable in this case to use for motor.

Another advantage of fuzzy controller is achieved in this thesis is that in Fig 7.1 at time T=1 second a torque load is applied to both controllers. In PID controller speed is reduced for some time and then again restore but in fuzzy controller not even a minor reduction of speed is observed. That make fuzzy controller a best controller in controller categories that load tends to reduced the speed for some instance in other controllers and also this decrease of speed is observed but in fuzzy this change of torque load not effect even for a minor time. So if the control system required to run the motor at constant speed and torque load not affect the speed then in this case fuzzy controller is best to use instead of PID controller.

7.2 Pole placement controller vs. Fuzzy, PID controller.

Pole placement controller provides excellent control in control systems. In pole placement techniques designer can set sampling rate which provide small sampling rate for control hence output is more smooth and designer can set the step response so the processes where fast step response is needed it is achieved by making dead beat controller while where fast reaching to reference point is not necessary it is also possible. While in fuzzy controller or PID controller the step response is fixed. In pole placement controller step response variation is achieved by selecting the poles location. If poles are located close to origin then step response of controller is fast and if location of poles are far from origin then step response is slow. This gives pole placement controller a unique advantage over others controller.

Pole placement design can also be work on non-linear plants, but this is not easy to control because in such case position of zeros and poles are not known. So the system can be controlled by converting into linear by local linearization around a point and then apply pole placement strategies. Fig 5.11 shows the pole placement design using integrator. The system reached to reference point in almost 35 seconds which is slow as compared to Fuzzy and PID controller.
This is not good for any controller but this controller can be used in railway engines because rapid increase of speed is not required in such case.
GUI is designed that the user used any controller by changing the reference point and torque load. GUI results are shown in this chapter.

Fig 8.1 GUI without controller
Fig 8.2 GUI output using PID controller

Fig 8.3 GUI output using fuzzy controller
Fig 8.4 GUI output using pole placement controller
Fig 8.5 GUI output using PID controller
Fig 8.5 GUI output using Fuzzy controller
Conclusion.

A remarkable and attractive tool for teaching controllers optimistic design and tuning steps has been presented in this thesis. A DC motor is mathematically modeled and then simulated in Matlab software. This is not an attractive way for electrical engineers for controllers designing but also in the field of artificial intelligence or for those students who want to know about Fuzzy logic control concepts, designing of fuzzy systems and rule base. The control of DC motor with PID, fuzzy logic controller and pole placement controller results Fuzzy controller is best for control applications. Chopper is also introduced in this thesis to get more smooth output and less steady state error. Pole placement controller provide good results in noisy systems and another fact is observed that fuzzy controller is better to sustain torque load condition as its speed remain same whatever torque load change or not. The whole thesis presented in the form of GUI so the user can select any controller according to the nature of control system and change inputs.
Appendix.

1.1 Parameters for PMDC motor

clear all, close all

% MOTOR CHARACTERISTICS (usually given with spec sheet or measured)

% Fuzzy Logic Controller with Simulink Toolbox for Pmdc Machines %
% File Name: set_pmdc.m %
% Initialization fuzzy rules and parameters %
% File prepared by Ali Junaid Ashraf (2010) %
% Values taken from [17] %
% http://www.ecircuitcenter.com/Circuits/dc_motor_model/

% DCmotor_model.htm %

% %%%%%%%%%%%%%%%%%%%%%%%  PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %
% The following are nominal values for a 10 V motor. %
% %
% Kb = kv,kt motor constant (N-m/A) %
% Ra = electrical resistance in armature circuit (ohms) %
% La = electrical inductance in armature circuit (henry) %
% J  = mechanical inertance in motor/gear reducer (kg-m^2) %
% B1 = effective linear mechanical resistance %
% Vn = terminal voltage %
% TL = load moment %
% %
% %%%%%%%%%%%%%%%%%%%%%%%  PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %

La = 0.0015;
Ra = 0.5;
Kv = 0.05;
kt = 0.05;
J  = 0.00025;
B  = 0.0001;
V  = 10;

num = kt/(La*J);
den = [1 (Ra*J + La*B)/(La*J) (Ra*B + kt*kv)/(La*J)];
ali = tf(num,den);

1.2 Plots of Dc motor with out PID controller.

sim('without_pid');
subplot(3,1,1);
plot(time,speed);
legend('speed',0);
grid on;
xlabel('Time sec');
ylabel('Speed rad/sec');
title('Speed and current output at Torque load ');

subplot(3,1,2);
plot(time,current);
xlabel('Time sec');
ylabel('Current AMPS');
legend('current',0);
grid on;

subplot(3,1,3);
plot(time,tl);
xlabel('Time sec');
ylabel('Torque load N.m');
legend('Torque load',0);
grid on;
1.3 Plots of Dc motor with PID controller.

```matlab
sim('with_pid');

subplot(3,1,1);
plot(time,speed);
legend('speed');
grid on;
xlabel('Time sec');
ylabel('Speed rad/sec');
title('Tunned PID controller output at Torque load');

subplot(3,1,2);
plot(time,current);
xlabel('Time sec');
ylabel('Current AMPS');
legend('current',0);
grid on;

subplot(3,1,3);
plot(time,tl);
xlabel('Time sec');
ylabel('Torque load N.m');
legend('Torque load',0);
grid on;
```
1.4 Calculation of transfer function.

```matlab
>> ali = tf(num,den)

Transfer function:
1.333e005
-------------------
s^2 + 333.7 s + 6800

>> [num,dend]=c2d(ali,0.1)

Transfer function:
17.22 z + 0.1665
--------------------------
z^2 - 0.113 z + 3.207e-015

Sampling time: 0.1

dend =

% From state space equation same transfer function can be obtained

A=[-Ra/La -kv/La; kt/J -B/J]

A =

-333.3333  -33.3333
200.0000   -0.4000

>> B=[1/La 0; 0 -1/J]
```
B =

1.0e+003 *

0.6667 0
0 -4.0000

>> C=[1 0; 0 1]

C =

1 0
0 1

>> D=[0 0; 0 0]

D =

0 0
0 0

sys_dc = ss(A,B,C,D)

a =

x1  x2
x1 -333.3 -33.33
x2  200   -0.4

b =

u1  u2
x1 666.7   0
x2 0  -4000
\[ c = \]
\[ \begin{array}{cccc}
  & x1 & x2 & \\
y1 & 1 & 0 & \\
y2 & 0 & 1 & \\
\end{array} \]

\[ d = \]
\[ \begin{array}{cccc}
  & u1 & u2 & \\
y1 & 0 & 0 & \\
y2 & 0 & 0 & \\
\end{array} \]

Continuous-time model.

\[ >> \text{sys\_tf} = \text{tf(sys\_dc)} \]

Transfer function from input 1 to output...

\[ 666.7 \, s + 266.7 \]
\#1: \---------------------------
\[ s^2 + 333.7 \, s + 6800 \]

\[ 1.333e005 \]
\#2: \---------------------------
\[ s^2 + 333.7 \, s + 6800 \]

Transfer function from input 2 to output...

\[ 1.333e005 \]
\#1: \---------------------------
\[ s^2 + 333.7 \, s + 6800 \]

\[ -4000 \, s - 1.333e006 \]
\#2: \---------------------------
\[ s^2 + 333.7 \, s + 6800 \]
sys_tf = tf(sys_dc)

Transfer function from input 1 to output...
  666.7 s + 266.7
#1:  -------------------
  s^2 + 333.7 s + 6800

1.333e005
#2:  -------------------
  s^2 + 333.7 s + 6800

Transfer function from input 2 to output...
  1.333e005
#1:  -------------------
  s^2 + 333.7 s + 6800

-4000 s - 1.333e006
#2:  -------------------
  s^2 + 333.7 s + 6800

>> [numd,dend]=c2d(sys_tf,0.1)

Transfer function from input 1 to output...
  0.2942 z - 0.2594
#1:  -----------------------------
  z^2 - 0.113 z + 3.207e-015

  17.22 z + 0.1665
#2:  -----------------------------
  z^2 - 0.113 z + 3.207e-015
Transfer function from input 2 to output...

\[ 17.22 z + 0.1665 \]

#1: --------------------------

\[ z^2 - 0.113 z + 3.207 \times 10^{-15} \]

-173.8 \[ z + 0.1069 \]

#2: --------------------------

\[ z^2 - 0.113 z + 3.205 \times 10^{-15} \]

Sampling time: 0.1

### 1.5 Calculation Of Pole-placement Parameters.[20]

% COPP Calculation Of Pole-placement Parameters.
% [Kr,C,D]=COPP(B,A,POL), calculate the parameters
% for a controller according to pole-placement method
% and returns the gain(Kr), C-parameters [c0,c1,...,cNc]
% and D-parameters [d0,d1,...,dND].

```
---------------------------------------------
|                         |
|                         |
| R(z) ____ + _____ | Y(z) |
|---| kr |---|1/C(z)|---|H(z)=B(z)/A(z)|---|
|   |    |    |_____| |___________|    |
|   |    |    |_____| |___________|    |
|   |    |    |_____| |________________|   |
|   |    |    |________|                |
|   |    |    |CONTROLER         |
---------------------------------------------
```

Vector B must contain the coefficients of the numerator in transfer function \( H(z) \), \( B=[b0*z^0 \ b1*z^-1 \ ... \ bN_b*z^-N_b] \).
Vector A must contain the coefficients of the
Na=length(A)-1;
Nb=length(B)-1;
Ndt=Nb-Na;
M=(2*Na-1)+Ndt;
Nc=Nb-1;
if (Na~=Nb)
    error('The B and A matrices must have the same number of columns !!!!');
end;

matrix_a=zeros(M);

for column=1 : Nc
    for row=1: Na+1
        matrix_a((row+column)-1,column)=A(row);
    end;
end;

for column_d=Nc+1:M
    for row_d=1:Nb
        matrix_a((row_d+column_d)-Nb,column_d)=B(row_d+1);
    end;
end;
el = length(A);
padd = 0;

while (A(el) == 0)
padd = padd + 1;
el = el - 1;
end

if (padd ~= 0)
    POL = [POL zeros(1, padd)];
end

e2 = length(B);
padd2 = 0;

while (B(e2) == 0)
padd2 = padd2 + 1;
e2 = e2 - 1;
end

if (padd2 ~= 0)
    POL = [POL zeros(1, padd2)];
end

P = poly(POL);
if (length(P) ~= M + 1)
    error('The number of poles don't match !!!');
end;
Kr = sum(P) / sum(B);

for count = 2:Na + 1
    P(count) = P(count) - A(count);
end;
matrix = P';
matrix_b = matrix(2:M + 1,:);
inv_matrix_a=inv(matrix_a);
result=inv_matrix_a * matrix_b;

C(1,1)=1;

for i=1:Nc
    C(i+1,1)=result(i,1);
end;

D=result(Nc+1:M,1);

1.6 Pole placement controller plots.

clear all;clc;close all;
% Pole placement controller

% Reference
    ref=50;
%disturbances
    e=0.009;

A=[1 -0.113 3.207e-015];
B=[0 17.22 0.1665];

q=0.2;
q1=q;
q2=q;
q3=q;

POL=[q1 q2 q3];
[Kr,C,D]=copp_new(B,A,POL);
c1=C(2,1);
d0=D(1,1);
d1=D(2,1);
sim('ali_pp',[0,500]);

subplot(2,1,1);
plot(time,y);hold on;
stairs(ref,'--r');
legend('speed',0);
grid on;
xlabel('Time sec');
ylabel('Speed rad/sec');
title('Speed and controller output with disturbance ');
legend('outputs','reference',0);

subplot(2,1,2);
plot(time,u);
legend('control signal');
grid on;
xlabel('Time sec');
ylabel('Voltage');
1.7 Plots for fuzzy controller.

clear; help set_pmdc

TL=0.1;

La=0.0015;
Ra = 0.5;
Kb=0.05;
J =0.00025;
B1=0.0001;
Vn=10;
winName = bdroot(gcs);
fprintf('Initializing ''fismatrix'' in %s...
', winName);
fismatrix_for_pmdc = readfis('pmdc.fis');
fprintf('Done with initialization.

');
sim('fuzzy_logic_controller_for_pmdc');
subplot(3,1,1);
plot(time,speed);
legend('speed',0);
grid on;
xlabel('Time sec');
ylabel('Speed rad/sec');
title('Speed and current output at Torque load ');

subplot(3,1,2);
plot(time,current);
xlabel('Time sec');
ylabel('Current AMPS');
legend('current',0);
grid on;

subplot(3,1,3);
plot(time,tl);
xlabel('Time sec');
ylabel('Torque load N.m');
legend('Torque load',0);
grid on;

legend('error',0);
grid on;
xlabel('Time sec');
ylabel('error');
title('error and change in error output');

subplot(2,1,2);
plot(time,ce);
xlabel('Time sec');
ylabel('change in error');
legend('change in error',0);
grid on;

1.8 Code of Gui

function varargout = DCmotorCONTROL(varargin)

% DCMOTORCONTROL M-file for DCmotorCONTROL.fig
% DCMOTORCONTROL, by itself, creates a new DCMOTORCONTROL or raises the
% existing
% singleton*.
%
% H = DCMOTORCONTROL returns the handle to a new DCMOTORCONTROL or the
% handle to
% the existing singleton*.
%
% DCMOTORCONTROL('CALLBACK',hObject,eventData,handles,...) calls the
% local
% function named CALLBACK in DCMOTORCONTROL.M with the given input
% arguments.
%
% DCMOTORCONTROL('Property','Value',...) creates a new DCMOTORCONTROL or
% raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before DCmotorCONTROL_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to DCmotorCONTROL_OpeningFcn via
% varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help DCmotorCONTROL
% Last Modified by GUIDE v2.5 18-May-2010 00:12:41

% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;

gui_State = struct('gui_Name', mfilename, ...
                    'gui_Singleton', gui_Singleton, ...
                    'gui_OpeningFcn', @DCmotorCONTROL_OpeningFcn, ...
                    'gui_OutputFcn', @DCmotorCONTROL_OutputFcn, ...
                    'gui_LayoutFcn', [], ...
                    'gui_Callback', []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% End initialization code - DO NOT EDIT

% --- Executes just before DCmotorCONTROL is made visible.

function DCmotorCONTROL_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to DCmotorCONTROL (see VARARGIN)

% Choose default command line output for DCmotorCONTROL
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
% UIWAIT makes DCmotorCONTROL wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
% function varargout = DCmotorCONTROL_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% function TL_Callback(hObject, eventdata, handles)
% hObject    handle to TL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TL as text
% str2double(get(hObject,'String')) returns contents of TL as a double

% --- Executes during object creation, after setting all properties.
% function TL_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function ref_Callback(hObject, eventdata, handles)
    % hObject    handle to ref (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    
    % Hints: get(hObject,'String') returns contents of ref as text
    % str2double(get(hObject,'String')) returns contents of ref as a double

    if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function control_CreateFcn(hObject, eventdata, handles)
    % hObject    handle to control (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    empty - handles not created until after all CreateFcns called
    
    % --- Executes during object creation, after setting all properties.
    function control_CreateFcn(hObject, eventdata, handles)
        % hObject    handle to control (see GCBO)
        % eventdata  reserved - to be defined in a future version of MATLAB
        % handles    empty - handles not created until after all CreateFcns called

        % --- Executes on button press in reset.
        function reset_Callback(hObject, eventdata, handles)
% hObject    handle to reset (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.TL,'string','0');   %set load torque value to 0
set(handles.ref,'string','0');  %set speed reference value to 0

cla(handles.axes1,'reset'); %clears the axes
cla(handles.axes2,'reset'); %clears the axes

% --- Executes on button press in start.
function start_Callback(hObject, eventdata, handles)
% hObject    handle to start (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

clc;
La=0.0015;
Ra =0.5;
kv=0.05;
kt =0.05;
J =0.00025;
B=0.0001;
V=10;
TL=str2double(get(handles.TL,'string'));    %take load torque value
ref=str2double(get(handles.ref,'string'));  %take speed reference value

h=get(handles.control,'SelectedObject'); %take to selecting for controllers
cont=get(h,'Tag');
global selecter;
switch cont
    case 'uncon'
        options=simset('SrcWorkspace','current');
        [t,x,y] = sim('without_control',[0 1],options);
        selecter=true;
case 'pid'
    options=simset('SrcWorkspace','current');
    [t,x,y] = sim('with_pid',[0 1],options);
    selecter=true;

case 'pp'
    A=[1 -0.113 3.207e-015];
    B=[0 17.22 0.1665];

    q=0.4; q1=q; q2=q; q3=q;
    POL=[q1 q2 q3];

    [Kr,C,D]=copp_new(B,A,POL);
    c1=C(2,1);  d0=D(1,1);  d1=D(2,1);
    options=simset( 'Solver','VariableStepDiscrete','Maxstep',0.1,...
                        'SrcWorkspace','current');
    [t,x,y] = sim('with_pp',[0 10],options);
    selecter=false;

case 'ppi'
    A=[1 -1.113 0.113 -3.2e-15 ];
    B=[0 17.22 0.1665 0];
    q=0.4;
    q1=q; q2=q; q3=q; q4=q;
    POL=[q1 q2 q3 q4];

    [Kr,C,D]=copp_new(B,A,POL);
    c1=C(2,1);
    c2=C(3,1);
    d0=D(1,1);
    d1=D(2,1);
    d2=D(3,1);

    options=simset( 'Solver','VariableStepDiscrete','Maxstep',0.1,...
                        'SrcWorkspace','current');
    [t,x,y] = sim('with_ppi',[0 10],options);
selecter=false;

case 'fuzzy'
    winName = bdroot(gcs);
    fprintf('Initializing ''fismatrix'' in %s...
', winName);
    fismatrix_for_pmdc = readfis('pmdc.fis');
    fprintf('Done with initialization.

');
    options=simset('SrcWorkspace','current');
    [t,x,y] = sim('with_fuzzy',[0 1],options);
    selecter=true;
end

% ***** setting a variable to results *******
ia=y(:,1);
w=y(:,2);
tr=y(:,3);
u=y(:,4);

% ****************** Plots *******************
axes(handles.axes1);
if selecter==true
    grid off;hold off;
    plot(t,w,'r');hold on;
    grid on;
    xlabel('time');
else
    grid off;hold off;
    stairs(t,w,'r');hold on;
    grid on;
    xlabel('Sample Index (h=0.1 sec)');
end
legend('mechanical speed',4);
title('Speed graph of DC Motor');

ylabel('Rad/sec')
zoom on;
axes(handles.axes2);

if selector==true
    grid off; hold off;
    plot(t,ia); hold on;
    stairs(t,tl,'--m'); grid on;
    xlabel('time');
else
    grid off; hold off;
    stairs(t,ia); hold on;
    stairs(t,tl,'--m'); grid on;
    xlabel('Sample Index (h=0.1 sec)');
end
legend('armature current','load torque',1);
title('Current graph of DC Motor');
ylabel('Amper');
zoom on;
Bibliography.

[4]. http://hyperphysics.phy-astr.gsu.edu/hbase/magnetic/motdc.html
[7]. http://www.animations.physics.unsw.edu.au/jw/electricmotors.html#DCmotors
[10]. Seda Sydemir, serkan sezen, H. Metin Kocaeli university, Turky
=04590052&isnumber=4589950&tag=1
BHU, Varanasi, UP, India
[14]. http://www.newton.dep.anl.gov/askasci/phy00/phy00351.htm
[19]. Applied model based control Bjorn Sohlberg 2009 Dalrna university
[20]. H. Ghods & R. M. Nikfarjam, EL Dept., Bachelor thesis
    Supervisor Dr Bjorn Sohlberg
    Univ. of Dalarna Sweden 18-4-96.
[22]. KOKSAL ERENTURK Department of Electronic Engineering, Faculty of Engineering,
Ataturk University, 25240 Erzurum, Turkey
[24]. Seda Aydemir, Serkan Sezen, H. Metin Ertunc Dept. of Electrical Engineering Kocaeli
University Izniit, Kocaeli, Turkey