Speed Control of DC Motor Using PID & Smart Controller

Purushotam Kumar, Prabhas Kumar Prabhat, Mithun Kumar, Dr. S.D. Choudhary

Abstract- The thesis describes about the concept of DC motor and Speed control separately excited DC motor. Motor speed is controlled 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 the result is discussed. This paper describes about the basic concepts of Fuzzy Set and Fuzzy Logic, the speed control with the help of Fuzzy controller. Fuzzy controller provides better control strategies than other controllers. Optimization of Fuzzy controller with Simulink model describes in this paper and a new way for faster response and smooth output. The comparison of these two controllers’ results is also showed. From the results it is proved that Fuzzy Controller is the best controller. Finally the MATLAB Simulation is discussed.

Index Terms: DC Motor, PID Controller, Existing tuning, Steady State Error, Fuzzy logic, Fuzzy controller, Simulink, MATLAB.

1. CONCEPTS OF DC MOTOR

For a motor convert electrical energy into mechanical energy. There are two types of motor: AC motor & 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. 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 affected. DC motor provides 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.

2. SPEED CONTROL OF DC MOTOR

The term speed control stand for intentional speed variation carried out manually or automatically DC motors are most suitable for wide range speed control and are there for many adjustable speed drives.

- Purushotam Kumar, Research Scholar, Department of Electrical Engineering, Asansol Engineering College, WB, India E-mail: purushotanikumar88@gmail.com
- Prabhakar Kumar Prabhat is currently associated with the department of Electrical Engineering, DAVIET, Daltonganj, Jharkhand, India, E-mail: pkp.nitp@gmail.com
- Mithun Kumar is currently associated with the department of Electrical Engineering, DAVIET, Daltonganj, Jharkhand, India, E-mail: mithunkism@gmail.com
- Dr. S.D. Choudhary is currently associated with the department of Electronics & Communication Engineering, DAVIET, Daltonganj, Jharkhand, India, E-mail: suryadeo.bit@gmail.com

Model of DC Motor

Where, $V_A$ is the armature voltage. (In volt)
$E_b$ is back emf the motor (In volt)
$I_a$ is the armature current (In amperes)
$R_a$ is the armature resistance (In ohm)
$L_a$ is the armature inductance (In Henry)
$T_m$ is the mechanical torque developed (In Nm)
$J_m$ is moment of inertia (In kg/m²)
$B_m$ is friction coefficient of the motor (In Nm/ (rad/sec))
$\omega$ is angular velocity (In rad/sec)

We know that

$$\omega = \frac{V_A - IR}{K_a}$$

Where, $\phi$ = Field flux per pole
$K_a$ = Armature constant = $PZ / 2\pi$

Where, P = No. of pole

Specification of the dc motor:

Armature resistance ($R_a$) = 0.5Ω
Armature inductance ($L_a$) = 0.02 H
Armature voltage ($V_a$) = 200 V
Mechanical inertia ($J$) = 0.1 Kg .m
Friction coefficient ($B_m$) = 0.008 N.m/rad/sec
Back emf constant ($k$) = 1.25 V/rad/sec
Rated speed = 1500 r.p.m
Motor torque constant=N.m/A

1044
Calculation:

Speed at full load when $\omega = 157.07 \text{ rad/sec}$

$E_b = 1.25 \times 157.07 = 196.3 \text{ V}$

$V_A = E_b + I R$

$200 = 196.3 + I R$

$I_A = 7.325 \text{ Amps}$

3. SPEED CONTROL WITH 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.

The PID measurement depends 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 off 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 overshoots. The P term take the output to proportional of error value. Its response can be adjusted by multiplying the error by a constant $K_p$ 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 makes a small control action.

4. 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:

- Run the controller by taking only P value.
- Increase P value of the system until it self-oscillating with constant amplitude.
- Then take controller gain time period
Different values of $K_P$, $K_I$ & $K_d$ of PID controller for position from graph analysis:

<table>
<thead>
<tr>
<th>No</th>
<th>$K_P$</th>
<th>$K_I$</th>
<th>$K_d$</th>
<th>Equation</th>
<th>Steady state error (e)</th>
<th>Setting time (t)</th>
<th>Rise time (r)</th>
<th>System</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>1000</td>
<td>45</td>
<td>$1.000+1.000+10.000$</td>
<td>0.022</td>
<td>0.002</td>
<td>0.004</td>
<td>Blue</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>95</td>
<td>950</td>
<td>45</td>
<td>$1.95+1.10+9.45$</td>
<td>0.021</td>
<td>0.002</td>
<td>0.004</td>
<td>Green</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>97</td>
<td>970</td>
<td>45</td>
<td>$1.97+1.11+9.47$</td>
<td>0.022</td>
<td>0.002</td>
<td>0.004</td>
<td>Red</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>85</td>
<td>900</td>
<td>45</td>
<td>$1.85+1.05+8.95$</td>
<td>0.025</td>
<td>0.003</td>
<td>0.004</td>
<td>Cyan</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>150</td>
<td>1000</td>
<td>68</td>
<td>$1.10+0.87+10.68$</td>
<td>0.018</td>
<td>0.005</td>
<td>0.004</td>
<td>Blue</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>140</td>
<td>1200</td>
<td>80</td>
<td>$1.14+0.89+12.80$</td>
<td>0.018</td>
<td>0.005</td>
<td>0.004</td>
<td>Green</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>140</td>
<td>1500</td>
<td>80</td>
<td>$1.14+0.89+15.80$</td>
<td>0.018</td>
<td>0.005</td>
<td>0.004</td>
<td>Yellow</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>2000</td>
<td>80</td>
<td>$1.10+1.20+20.10$</td>
<td>0.018</td>
<td>0.005</td>
<td>0.004</td>
<td>Black</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>100</td>
<td>1500</td>
<td>80</td>
<td>$1.10+1.20+15.10$</td>
<td>0.018</td>
<td>0.005</td>
<td>0.004</td>
<td>Blue</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>250</td>
<td>2500</td>
<td>125</td>
<td>$1.25+1.37+25.125$</td>
<td>0.018</td>
<td>0.005</td>
<td>0.004</td>
<td>Red</td>
<td></td>
</tr>
</tbody>
</table>

Different values of $K_P$, $K_I$ & $K_d$ of PID controller for angle from graph analysis:

<table>
<thead>
<tr>
<th>No</th>
<th>$K_P$</th>
<th>$K_I$</th>
<th>$K_d$</th>
<th>Equation</th>
<th>Steady state error (e)</th>
<th>Setting time (t)</th>
<th>Rise time (r)</th>
<th>System</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-45</td>
<td>-200</td>
<td>-25</td>
<td>$-45.200+25.225$</td>
<td>0.035</td>
<td>0.002</td>
<td>0.004</td>
<td>Blue</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-25</td>
<td>-200</td>
<td>-25</td>
<td>$-25.200+25.225$</td>
<td>0.035</td>
<td>0.002</td>
<td>0.004</td>
<td>Green</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-72</td>
<td>-222</td>
<td>-15</td>
<td>$-72.222+15.1722$</td>
<td>0.035</td>
<td>0.002</td>
<td>0.004</td>
<td>Red</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-20</td>
<td>-300</td>
<td>-15.9</td>
<td>$-20.300+15.999$</td>
<td>0.035</td>
<td>0.002</td>
<td>0.004</td>
<td>Cyan</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-20</td>
<td>-400</td>
<td>-15</td>
<td>$-20.400+15.400$</td>
<td>0.035</td>
<td>0.002</td>
<td>0.004</td>
<td>Blue</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-50</td>
<td>-300</td>
<td>-15</td>
<td>$-50.300+15.300$</td>
<td>0.035</td>
<td>0.002</td>
<td>0.004</td>
<td>Green</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-15</td>
<td>-200</td>
<td>-10</td>
<td>$-15.200+10.200$</td>
<td>0.035</td>
<td>0.002</td>
<td>0.004</td>
<td>Red</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-10</td>
<td>-200</td>
<td>-10</td>
<td>$-10.200+10.200$</td>
<td>0.035</td>
<td>0.002</td>
<td>0.004</td>
<td>Yellow</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>-10</td>
<td>-300</td>
<td>-10</td>
<td>$-10.300+10.300$</td>
<td>0.035</td>
<td>0.002</td>
<td>0.004</td>
<td>Black</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-67</td>
<td>-55</td>
<td>-10</td>
<td>$-67.55+10.55$</td>
<td>0.035</td>
<td>0.002</td>
<td>0.004</td>
<td>Red</td>
<td></td>
</tr>
</tbody>
</table>

CONCLUSION:

Applying PID controller both system become marginally stable. But settling time is 10 sec for angle of pendulum which should be decreased. Too much damping is occurs for both angle and position. So we can’t say that PID is good for inverted pendulum.

5. SPEED CONTROL WITH FUZZY CONTROLLER

Controllers based on the fuzzy logic give the linguistic strategies control conversion from expert knowledge in automatic control strategies. The first fuzzy logic based controllers application was done by Assilian and Mandani [2]. The recent fuzzy logic controller application [3] in water quality control, train operation automatic system, elevators control, nuclear reactor control and fuzzy computers shows an efficient way for using the fuzzy control in complex process which can be controlled by a skilful human being without
knowing its dynamic. The fuzzy logic controller consists of a linguistic propositions and rules set, which defines individual control actions. A fuzzy logic controller designed on the basis of the fuzzy logic is an approximate reasoning-based controller, which does not require exactly analytical models and is much closer in spirit to human thinking and natural language than the traditional logic system. Essentially, the control strategies in the FLC are based on expert experience, so the fuzzy logic controller can be regarded as the simulation of a humanoid control model. When the designing a FLC, the control strategies have to be regarded as the simulation of a humanoid control rules pre-constructed by control result fails to meet the system requirement due to a change in the outside environment. The possible solution to this problem is that we can adjust either the membership function of the fuzzy sets or the control rules to achieve the control objective.

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 LotfiZadeh in 1965. Fuzzy logic has currently used in control theory, artificial intelligence systems specially to control complex aircraft engines and control surfaces, helicopter control, missle guidance, automatic transmission, wheel slip control, auto focus cameras and washing machines, railway engines for smooth drive and fuel consumption and many industrial processes. Fuzzy logic provide better results if we compared it with PID controller. Fuzzy logic control is a control algorithm based on a linguistic control strategy, which is derived from expert knowledge into an automatic control strategy. The operation of a FLC is based on qualitative knowledge about the system being controlled .It doesn’t need any difficult mathematical calculation like the others control system. While the others control system use difficult mathematical calculation to provide a model of the controlled plant, it only uses simple mathematical calculation to simulate the expert knowledge. Fuzzy set 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 models 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 allows decision making with incomplete information. Human reasoning can also be known as multi valued imprecise. \[21\] The requirement for the application of a FLC arises mainly in situations where:

6. The description of the technological process is available only in word form, not in analytical form.
7. It is not possible to identify the parameters of the process with precision.
8. The description of the process is too complex and it is more reasonable to express its description in plain language words.
9. The controlled technological process has a “fuzzy” character.
10. It is not possible to precisely define these conditions.

A fuzzy logic controller has four main components as shown in Figure:

1. Fuzzification
2. Inference engine
3. Rule base
4. Defuzzification

**Fuzzification**

The first step in designing a fuzzy controller is to decide which state variables represent the system dynamic performance must be taken as the input signal to the controller. Fuzzy logic uses linguistic variables instead of numerical variables. The process of converting a numerical variable (real number or crisp variables) into a linguistic variable (fuzzy number) is called fuzzification. This is achieved with the different types of fuzzifiers. There are generally three types of fuzzifiers, which are used for the fuzzification process; they are:
Singleton fuzzifier
Gaussian fuzzifier
Trapezoidal or triangular fuzzifier

Rule Base

A decision making logic which is, simulating a human decision process, inters fuzzy control action from the knowledge of the control rules and linguistic variable definitions [9]. The rules are in “If Then” format and formally the If side is called the conditions and then side is called the conclusion. The computer is able to execute the rules and compute a control signal depending on the measured inputs error (e) and change in error (de). In a rule based controller the control strategy is stored in a more or less natural language. A rule base controller is easy to understand and easy to maintain for a non-specialist end user and an equivalent controller could be implemented using conventional techniques [14].

Inference engine

Inference engine is defined as the Software code which processes the rules, cases, objects or other type of knowledge and expertise based on the facts of a given situation. When there is a problem to be solved that involves logic rather than fencing skills, we take a series of inference steps that may include deduction, association, recognition, and decision making. An inference engine is an information processing system (such as a computer program) that systematically employs inference steps similar to that of a human brain.

Defuzzification

The reverse of Fuzzification is called Defuzzification. The use of Fuzzy Logic Controller (FLC) produces required output in a linguistic variable (fuzzy number). According to real world requirements, the linguistic variables have to be transformed to crisp output. There are many defuzzification methods but the most common methods are as follows [11]:

- Center of gravity (COG)
- Bisector of area (BOA)
- Mean of maximum (MOM)

Center of gravity (COG)

For discrete sets COG is called center of gravity for singletons (COGS) where the crisp control value is the abscissa of the center of gravity of the fuzzy set is calculated as follows:

\[ u_{COGS} = \frac{\sum_i \mu_c(x_i) x_i}{\sum_i \mu_c(x_i)} \]

Where \( x_i \) is a point in the universe of the conclusion \((i=1, 2, 3,...)\) and \( \mu_c(x_i) \) is the membership value of the resulting conclusion set. For continuous sets summations are replaced by integrals.

![Fig: Centre of Gravity](image)

Bisector of area (BOA)

The bisector of area (BOA) defuzzification method calculates the abscissa of the vertical line that divides the area of the resulting membership function into two equal areas. For discrete sets, is the abscissa \( x_j \) that minimizes.

\[ \sum_{i=1}^j \mu_c(x_i) - \sum_{i=j+1}^{l_{max}} \mu_c(x_i) = \min \]

Here \( l_{max} \) is the index of the largest abscissa \( x_{l_{max}} \). BOA is a computationally complex method.

Mean of maximum (MOM)

In this method the crisp value is to choose the point with the highest membership. There may be several points in the overall implied fuzzy set which have maximum membership value. Therefore it is a common practice to calculate the mean value of these points. This method is called mean of maximum (MOM) and the crisp value is calculated as follows:

\[ u_{MOM} = \frac{\sum_{i \in I} x_i}{|I|}, \ I = \{ i \mid \mu_c(x_i) = \mu_{max} \} \]

Here \( I \) is the (crisp) set of indices \( i \) where \( \mu_c(x_i) \) reaches its maximum \( \mu_{max} \) cardinality (the number of members), and \( |I| \) is its Implementation of an FLC requires the choice of four key factors:
Number of fuzzy sets that constitute linguistic variables.
- Mapping of the measurements onto the support sets.
- Control protocol that determines the controller behavior.
- Shape of membership functions.

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 converts input data to degree of membership functions. In this process data is matched with condition of rules 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.

1. Centre of gravity (COG)
2. Centre of gravity method for singletons (COGS)
3. Bisector of area (BOA)
4. Mean of Maxima (MOM)
5. Left most maximum (LM), and right most maximum (RM)

Sugeno Fuzzy Inference

Mamdani style is not computationally efficient as it find the centroid of two dimensional shapes 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]

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 and nine possible regions are selected 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 minimized. The bigger speed error causes the bigger controller input. In addition changing of the error plays an important role to define controller input.
In region one according to Fig. 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 but this result gives us crisp value for fuzzy control this crisp value should be converted into linguistic form.

- 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

The output of controller plotted against the rules describes. 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.

The surface in the next Fig. 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.

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

\[ \text{Error (ek)} = \text{Wref} - \text{wm} \]

\[ \text{Change in error (ce)} = e(k) - e(k-1) \]

Back emf provides 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. 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.

**Design of Fuzzy Rules**

Rule bases for tuning $K_p$

<table>
<thead>
<tr>
<th>de/e</th>
<th>NL</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL</td>
<td>PVS</td>
<td>PMS</td>
<td>PM</td>
<td>PL</td>
<td>PVL</td>
</tr>
<tr>
<td>NS</td>
<td>PMS</td>
<td>PML</td>
<td>PL</td>
<td>PVL</td>
<td>PVL</td>
</tr>
<tr>
<td>ZE</td>
<td>PM</td>
<td>PML</td>
<td>PL</td>
<td>PVL</td>
<td>PVL</td>
</tr>
<tr>
<td>PS</td>
<td>PML</td>
<td>PVL</td>
<td>PVL</td>
<td>PVL</td>
<td>PVL</td>
</tr>
</tbody>
</table>

Rule bases for tuning $K_i$

<table>
<thead>
<tr>
<th>de/e</th>
<th>NL</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL</td>
<td>PVL</td>
<td>PVL</td>
<td>PVL</td>
<td>PVL</td>
<td>PVL</td>
</tr>
<tr>
<td>NS</td>
<td>PML</td>
<td>PML</td>
<td>PML</td>
<td>PL</td>
<td>PVL</td>
</tr>
<tr>
<td>ZE</td>
<td>PVS</td>
<td>PVS</td>
<td>PS</td>
<td>PMS</td>
<td>PMS</td>
</tr>
<tr>
<td>PS</td>
<td>PML</td>
<td>PML</td>
<td>PML</td>
<td>PL</td>
<td>PVL</td>
</tr>
</tbody>
</table>

Rule bases for tuning $K_d$

<table>
<thead>
<tr>
<th>de/e</th>
<th>NL</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL</td>
<td>FM</td>
<td>PM</td>
<td>PM</td>
<td>PM</td>
<td>PM</td>
</tr>
<tr>
<td>NS</td>
<td>PMS</td>
<td>PMS</td>
<td>PMS</td>
<td>PMS</td>
<td>PMS</td>
</tr>
<tr>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
<td>PVS</td>
<td>PS</td>
<td>PS</td>
</tr>
<tr>
<td>PS</td>
<td>PMS</td>
<td>PMS</td>
<td>PMS</td>
<td>PMS</td>
<td>PMS</td>
</tr>
<tr>
<td>PL</td>
<td>FM</td>
<td>PM</td>
<td>PM</td>
<td>PM</td>
<td>PM</td>
</tr>
</tbody>
</table>

**Comparison between fuzzy PID and conventional PID controller:**

- Self-tuned tuning PID controller is less compared to conventional PID controller.
- The three parameters "$K_p$, "$K_i$, "$K_d$" of conventional PID control need to be constantly adjusted online in order to achieve better control performance.
- Fuzzy self-tuning PID parameters controller can automatically adjust PID parameters in accordance with the speed error and the rate of speed error-change, so it has better self-adaptive capacity fuzzy PID parameter controller has smaller overshoot and less rising and settling time than conventional PID controller and has better dynamic response properties and steady-state properties.
- Steady state error in case of self tuned fuzzy PID is less compared to conventional PID controller.

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>$K_d$</th>
<th>$K_p$</th>
<th>$K_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.75</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3.5</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>0.02</td>
<td>-3</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>2.5</td>
<td>0.8</td>
</tr>
<tr>
<td>5</td>
<td>55</td>
<td>4</td>
<td>-10.25</td>
</tr>
</tbody>
</table>

11. RESPONSE OF FUZZY LOGIC CONTROLLER

Output curve of Transfer Function applying the FLC in MATLAB for $K_p=0.75$, $K_i=15$, $K_d=1$
12. SIMULATION RESULTS

Output curve of Transfer Function applying the FLC in MATLAB\(k_1=3.5\), \(k_2=5k_3=20\)

Comparison of PID and Fuzzy PI with respect to Step Response

Output curve of Transfer Function applying the FLC in MATLAB\(k_1=0.02\), \(k_2=3k_3=15\)

Comparison of PID and Fuzzy PI with respect to Sine Response

Output curve of Transfer Function applying the FLC in MATLAB\(k_1=10\), \(k_2=5k_3=0.8\)

Result of Step Response of PID and Fuzzy
13. DISCUSSION AND CONCLUSION OF THE RESULT

In this project we have studied about different method for speed control of DC motor. The steady state operation and its various torque-speeds, torque-current characteristics of DC motor are studied. We have also studied basic definition and terminology of fuzzy logic and fuzzy set. This project introduces a design method of two inputs and three outputs self-tuning fuzzy PID controller and make use of MATLAB fuzzy toolbox to design fuzzy controller. fuzzy controller adjusted the proportional, integral and derivate (Kp, Ki, Kd) gains of the PID controller according to speed error and change in speed error. From the simulation results it is concluded that compared with the conventional PID controller, self-tuning PID controller has a better performance in both transient and steady state response. The FLC has better dynamic response curve, shorter response time, small overshoot, small steady state error (SSE), high steady precision compared to the conventional PID controller.

14. FUTURE SCOPE

MATLAB simulation for speed control of DC motor has been done which can be implemented in hardware to observe actual feasibility of the approach applied in this thesis. This technique can be extended to other types of motors. The parameters of PID controller can also be tuned by using genetic algorithm (GA).

REFERENCES

[18] MATLAB and SIMULINK Version 2010a, the MathworksInc, USA.

Books