Controlling of D.C. Motor using Fuzzy Logic Controller

Husain Ahmed[1]

Assistant Professor, Department of Electrical Engineering

Dehradun Institute of Technology, Dehradun (Uttarakhand)

husain_ahmed@rediffmail.com

Dr. Gagan Singh[2]

Professor & Head, Department of Electrical Engineering

Dehradun Institute of Technology, Dehradun (Uttarakhand)

gaganus@gmail.com

Vikash Bhardwaj[3] **, Saket Saurav**[4]**, Shubham Agarwal**[5]

Department of Electrical Engineering

Dehradun Institute of Technology, Dehradun (Uttarakhand)

vikas.dsb@gmail.com

Abstract: Most of the industrial controllers in use today utilize PID controllers. In this paper four methods for tuning of a PID controller are compared. A mathematical model of the most commonly used dynamic system i.e., a d.c. motor is derived and a PID controller is used in conjunction to it. Ziegler-Nichols rule based on the value of K_p is used to tune the PID controller and the response is studied. Further, fuzzy controller is used with PID i.e. Fuzzy PID Controller. The results obtained are compared and a conclusion is drawn that Fuzzy controlled PID responds better in case of change in load and other system disturbances.

1.Introduction:

Because of their high reliabilities, flexibilities and low costs, DC motors are widely used in industrial applications, robot manipulators and home appliances where speed and position control of motor are required. PID controllers are commonly used for motor control applications because of their simple structures and intuitionally comprehensible control algorithms. Controller parameters are generally tuned using hand-tuning or Ziegler-Nichols frequency response method. Both of these methods have successful results but long time and effort are required to obtain a satisfactory system response. Two main problems encountered in motor control are the time-varying nature of motor parameters under operating conditions and existence of noise in system loop.

Analysis and control of complex, nonlinear and/or time-varying systems is a challenging task using conventional methods because of uncertainties. Fuzzy set theory (Zadeh , 1965) which led to a new control method called *Fuzzy Control[1]* which is able to cope with system uncertainties. DC motor control is generally realized by adjusting the terminal voltage applied to the armature but other methods such as adjusting the field resistance, inserting a resistor in series with the armature circuit are also available.

Ziegler-Nichols [2] frequency response method is usually used to adjust the parameters of the PID controllers. However, It is needed to get the system into the oscillation mode to realize the tuning procedure. The proposed approach uses both fuzzy controllers and response optimization method to obtain the approximate values of the controller parameters. Then the parameters may be slightly varied to obtain the user-defined performance of the real-time control system. Then the parameters may be slightly varied to obtain the user-defined performance of the real-time control system. Thus, it's an actual problem to design adaptive PID controllers without getting the system into the oscillation mode. Here the mathematical model of a dc motor is used to obtain a transfer function between shaft position and applied armature voltage. This model is then built in MATLAB Simulink. Then design and tuning of proportional-integral-derivative (PID) controllers are reviewed in Simulink with the proposed design procedure.

2. DC motor model

In armature control of separately excited DC motors, the voltage applied to the armature of the motor is adjusted without changing the voltage applied to the field. Figure shows a separately excited DC motor equivalent model.

$$V_a(t) = R_a.I_a(t) + L_a.\frac{dI_a(t)}{dt} + e_b(t)..(1)$$

$$e_b(t) = K_b.w(t)\ldots\ldots\ldots\ldots\ldots\ .(2)$$

$$T_m(t) = K_t.I_a(t)\ \ldots\ldots\ldots\ldots\ldots\ .(3)$$

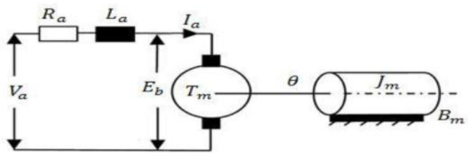


Figure: 1 *D.C. Motor Model*

$$T_m(t) = J_m.\frac{dw(t)}{dt} + B_m.w(t)\ \ldots\ldots(4)$$

Where:
V_a = *armature voltage*(V)
R_a = armature resistance (Ω)
L_a = armature inductance (H)
I_a = armature current (A)
e_b = back emf (V)
w = angular speed (rad/sec)
T_m = motor torque (Nm)
θ = angular position of rotor shaft (rad)
J_m = rotor inertia (kgm^2)
B_m = viscous friction coefficient (Nms/rad)
K_t = torque constant (Nm/A)
K_b = back emf constant (Vs/rad)

Let us combine the upper equations together:

$$V_a(t) = R_a.I_a(t) + L_a.\frac{dI_a(t)}{dt} + K_b.w(t)..(5)$$

$$K_T.I_a(t) = J_m.\frac{dw(t)}{dt} + B_m.w(t)\ \ldots\ldots(6)$$

Taking Laplace Transform of (5) & (6)…..

$$V_a(s) = R_a.I_a(s) + L_a.I_a(s) + K_b.W(s)\ldots(7)$$

$$K_T.I_a(s) = J_m.W(s) + B_m.W(s)\ \ldots\ldots(8)$$

If current is obtained from (8) and substituted in (7) we have…

$$V_a(s) = W(s).\frac{1}{K_T}[L_a.J_m.s^2 + (R_a.J_m + L_a.B_m(s) + (R_a.B_m + K_b.K_T)]\ \ldots\ldots\ldots(9)$$

Then the relation between rotor shaft speed and applied armature voltage is represented by transfer function:

$$\frac{W(s)}{V_a(s)} = \frac{K_T}{[L_a.J_m.s^2 + (R_a.J_m + L_a.B_m)s + (R_a.B_m + K_b.K_T)]}\ \ldots.(10)$$

The relation between position and speed is:

$$\theta(s) = \frac{1}{s}.W(s)\ \ldots\ldots\ldots\ldots\ldots\ldots(11)$$

Then the transfers function between shaft position and armature voltage at no-load is:

$$\frac{\theta(s)}{V_a(s)} = \frac{K_T}{[L_a.J_m.s^3 + (R_a.J_m + L_a.B_m)s^2 + (R_a.B_m + K_b.K_T)s]}\ \ldots.(12)$$

Below figure shows the DC motor model built in Simulink. Motor model was converted to a 2-in 2-out subsystem. Input ports are armature voltage (V_a) and load torque (T_{load}) and the output ports are angular speed in (w) and position (θ).

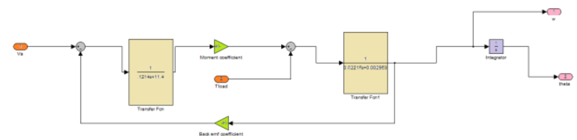


Figure 2: Mathematical SIMULINK Model of D.C. Motor

A 3.70 kW, 240V, 1750 rpm DC motor with the below parameters was used:

R_a = 11.4 Ω

L_a = 0.1214 H

J_m = 0.02215 kgm^2

B_m = 0.002953 Nms/rad

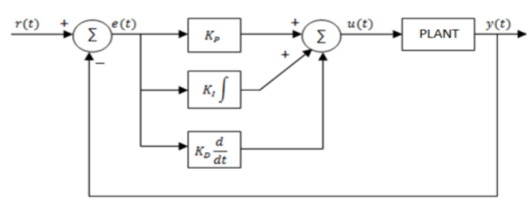
K_I = 1.28 Nm/A

K_b = 0.0045 Vs/rad

3. Proportional-integral-derivative (PID) Controller:

PID controllers are widely used in industrial control applications due to their simple structures, comprehensible control algorithms and low costs. Below figure shows the schematic model of a control system with a PID controller.

Control signal u(t) is a linear combination of error e(t), its integral and derivative.



PID control system

Figure 3: PID Control System

$$u(t) = K_p.e(t) + K_I\int e(t)dt + K_D\frac{de(t)}{dt}\ \ldots.(13)$$

$$u(t) = K_p(e(t) + \frac{1}{T_I}\int e(t)dt + T_D\frac{de(t)}{dt}\ \ldots.(14)$$

where ;

K_p = proportional gain

K_I = integral gain

K_D = derivative gain

T_I = integral time

$T_D =$ derivative time

If the controller is digital, then the derivative term may be replaced with a backward difference and the integral term may be replaced with a sum. For a small constant sampling time(T_S) , (14) can be approximated as:

$$u(n) = K_P.\left(e(n) + \frac{1}{T_I}\sum_{j=1}^{n} e(j)T_s + T_D.\frac{e(n)-e(n-1)}{T_s} \right) \dots\dots (15)$$

4. Tuning PID parameters : [2]

PID controllers are usually tuned using Hand Tuning or Ziegler –Nicholas methods and soft tuning. Hand tuning or Ziegler Nichols is generally used by experienced control engineers based on the rules shown in the Table. But these rules are not always valid. For example if an integrator exists in the plant, then increasing K_P results in a more stable control.

5. Hand-tuning rules

Operation	Rise Time	Overshoot	Stability
$K_P \uparrow$	Faster	Increases	Decreases
$T_D \uparrow$	Slower	Decreases	Increases
$1/T_I \uparrow$	Faster	Increases	Decreases

Table 1: Hand Tuning Rules

A simple hand-tuning procedure is as follows:

1. Remove derivative and integral actions by setting $T_D = 0$ and $\frac{1}{T_I} = 0$.

2. Tune K_P such that it gives the desired response except the final offset value from the set point.

3. Increase K_P slightly and adjust T_D to dampen the overshoot.

4. Tune $1/T_I$ such that final offset is removed.

5. Repeat steps from 3 until K_P is as large as possible.

6. Simulink implementation:

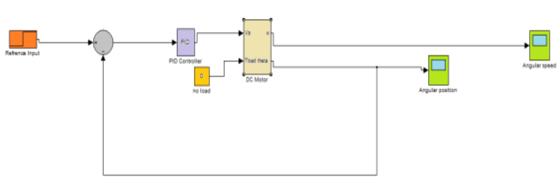


Figure 4: SIMULINK Model of PID Controlling of DC Motor

Here We first set $T_I = \infty$ (i.e. $K_I = 0$) and $T_d = 0$ (i.e. $K_d = 0$) .Using the proportional control action only, increase K_p from 0 to a critical value K_u , at which the output first exhibits sustained oscillations . The value of K_u so obtained is 2.9 .

The parameters thus obtained are:

K_p= 2.9, K_I= 0, K_d= 0

The sustained oscillation obtained is shown below:

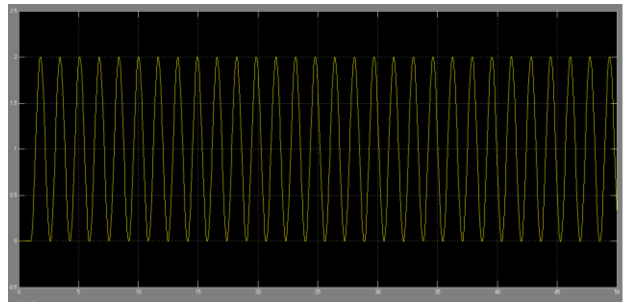


Figure 5: Angular Position Waveform

The disadvantage of this method is that it should take a long time to find the optimal values. Another method to tune PID parameters is Ziegler-Nichols frequency response method. The procedure is as follows:

1. Increase K_P until system response oscillates with a constant amplitude and record that gain value as K_u (ultimate gain).
2. Calculate the oscillation period and record it asT_u.
3. Tune parameters using the table given below:

7. Ziegler-Nichols rules

Controller	K_P	T_I	T_D
P	$0.5K_u$		
PI	$0.45K_u$	$T_u/1.2$	
PID	$0.6K_u$	$T_u/2$	$T_u/8$

Table 2: Ziegler Nichols Rules

The value of T_u is found to be 1.67.
The value of the parameters K_p, K_I and K_d based on Ziegler Nichols tuning method:
K_p= 1.74, K_I= 2.0963 , K_d= 0.363225

The response obtained is:

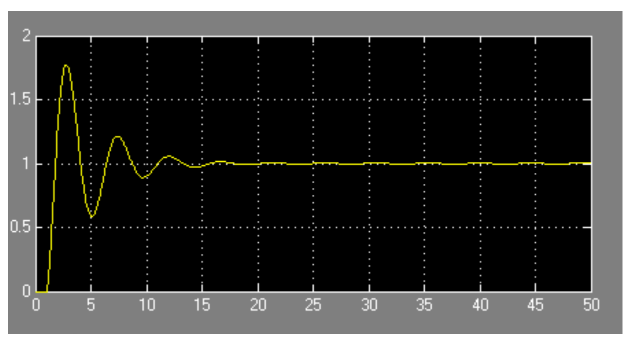


Figure 6: Response of the Model using Ziegler Nichols Tuning

Hand Tuning of the parameters is done to reduce the overshoot to a tolerable range. The values of the parameters which yielded the most suitable results are :

K_p= 1.9 , K_I= 1.5 , K_d= 0.7

The response thus obtained is shown below:

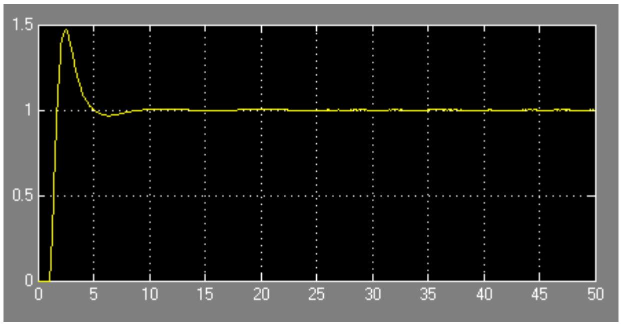


Figure 7: Response of the Model using Hand Tuning

8. Soft Tuning:

Soft Tuning of the parameters is done using a tool in MATLAB® /SIMULINK®

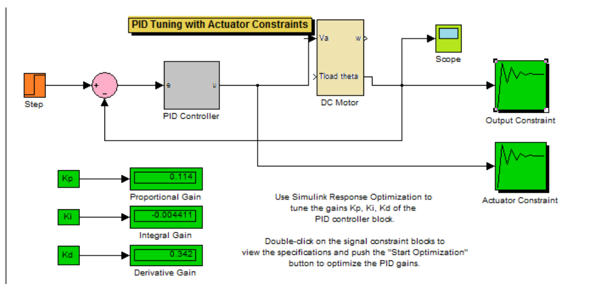


Figure 8: Functional Block Diagram SIMULINK Model of Soft Tuning

The parameters obtained via iteration are as follows:

K_p= 0.3420 , K_I= -0.0044 , K_d= 0.1027

The corresponding response thus obtained is shown below:

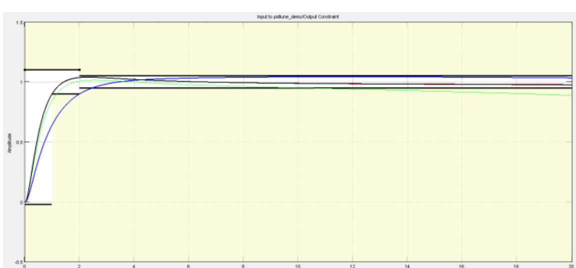


Figure 9: Response Waveform of Soft Tuning Iterations

9. Fuzzy Logic Controller:

Fuzzy logic is expressed by means of the human language [3].Based on fuzzy logic, a fuzzy controller converts a linguistic control strategy into an automatic control strategy, and fuzzy rules are constructed by expert experience or knowledge database.

First, set the error e(t) and the error variation de(t) of the angular velocity to be the input variables of the fuzzy logic controller. The control voltage u(t) is the output variable of the fuzzy logic controller.

The linguistic variables are defined as {NB, NS, Z, PS, PB}, where NB means negative big, NS means negative small, Z means zero, PS means positive small and PB means positive big.

The fuzzy rules are summarized in Table 3. The type of fuzzy inference engine is Mamdani. The fuzzy inference mechanism in this study follows as:

$u(t)$		$ce(t)$				
		NB	NS	Z	PS	PB
$e(t)$	NB	NB	NB	NS	NS	Z
	NS	NB	NS	NS	Z	PS
	Z	NS	NS	Z	PS	PS
	PS	NS	Z	PS	PS	PB
	PB	Z	PS	PS	PB	PB

Table 3: Fuzzy Rules

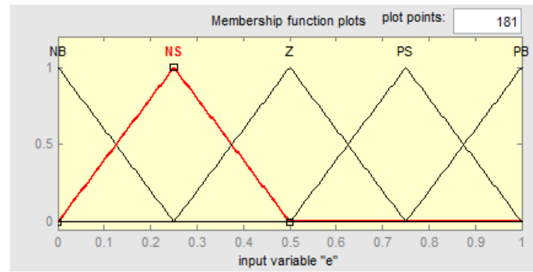


Figure 10: Membership functions for e normalised input

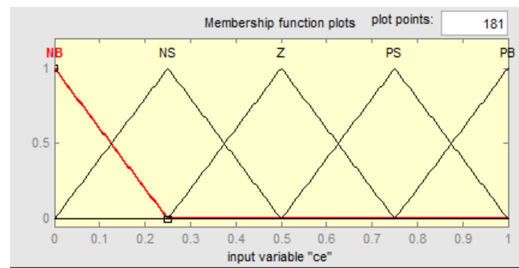


Figure 11: Membership functions for ce normalised input

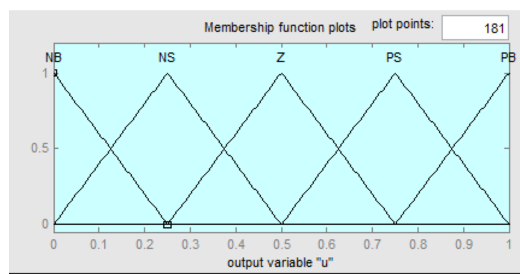


Figure 12: Membership functions for u normalised output

Here max-min type decomposition is used and the final output for system is calculated by using center of area gravity method.

$$\mu_B(u(t)) = \max[\mu_{A_1} j(e(t)), \mu_{A_2} j(ce(t)), \mu_B j(u(t))]$$

Where;

$$\mu_{A_1} j(e(t)) = \text{the membership function of e(t)}$$
$$\mu_{A_1} j(ce(t)) = \text{the membership function of ce(t)}$$
$$\mu_B j(u(t)) = \text{the membership function of u(t)}$$

j is an index of every membership function of fuzzy set, m is the number of rules and is the inference result. Fuzzy output u(t) can be calculated by the center of gravity defuzzification as:

$$u(t) = \frac{\sum_{i=1}^{m} \mu_B(u_i(t)).u_i}{\sum_{i=1}^{m} \mu_B(u_i(t))}$$

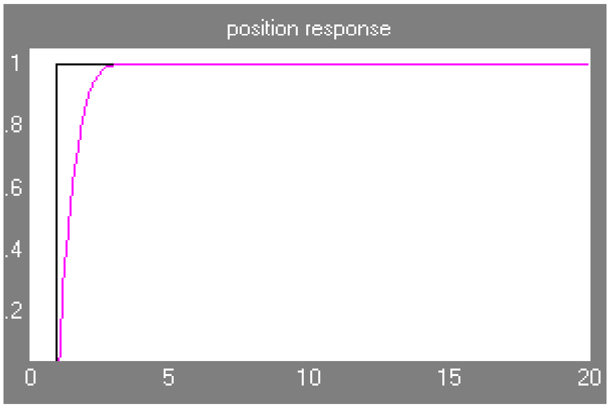


Figure 13: Response waveform of model using Fuzzy controller

10.Results:

The results obtained from different tuning methods are shown in the table given below.

	%Mp	Tr (sec)	Ts (sec)	ess
Z-N tuning	70	2	15	0
Hand tuning	50	1.5	10	0
MATLAB SIMULINK tuning	2	1	0	2
FLC	0	1	0	0

Table 4: Comparison of different performance indeces

11. Conclusion:

In this paper the position of D.C. motor is controlled by four methods of PID controlling techniques. The four methods used are: Ziegler-Nichols tuning, hand tuning, soft tuning in built in SIMULINK® and Fuzzy logic controller. According to the comparison of results of the simulations, it is found that the Fuzzy Logic Controller is better than other methods. The Fuzzy Logic Controller presents the following satisfactory performance indices:

1. Overshoot: Overshoot may be reduced by using Fuzzy Logic Controller.

2. Rise Time: 1 sec , which is minimum as compare to other methods.

3. Steady state error: 0

Hence it is concluded that the proposed Fuzzy Logic Controller provides better performance characteristics and improve the control of DC motor.

REFERENCES:

1. Zadeh, L. A., Fuzzy Sets. Information and Control, 8, 338-353, 1965.

2. Ogata, K., Modern Control Engineering, 3rd edition, NJ: Prentice Hall, 1997.

3. Foran J., 2002. Optimisation of a Fuzzy Logic Controller Using Genetic Algorithms, Doctorat, Texas University of America, USA.

4. Mathworks Inc., Simulink® Response Optimization™ Getting Started Guide, 3rd printing, 2008.

5. D. P. Kothari, I. J. Nagrath, “Electric Machines”, Tata McGraw – Hill Education Private Limited, ISBN-13: 978-0-07-058377-1, ISBN-10: 0-07-058377-3, 2004.

6. J. G. Ziegler and N. B. Nichols, “Optimum settings for automatic controllers,” *Trans. ASME*, vol. 64, pp. 759–768, 1942.

7. L. A. Zadeh, “Fuzzy sets,” Information and Control, vol. 8, pp. 338-353, 1965.