# Speed Control of Brushless DC Motor Using PID and Fuzzy Controller

Abhishek D. Gandhi[1], Suryaprakash Singh[2], Bhautik M. Daxini[3]
[1]Student, Instrumentation & Control Department, AITS, Rajkot, India
[2]Assistant Professor, Electrical Department, AITS, Rajkot, India
[3]Assistant Professor, Instrumentation & Control Department, AITS, Rajkot, India

***Abstract*—Brushless DC (BLDC) Motors are widely used in industries because of their high efficiency and high torque. This paper proposed PID and Fuzzy controller to control the speed of BLDC Motor. This paper provides an overview of PID controller and Fuzzy controller. PID controllers are insufficient to control the speed of BLDC motor as it gives high overshoot in the response. So for the better performance, intelligent controller such as Fuzzy is used. Fuzzy has the ability to satisfied control characteristics and it is easy for computing. The experimental results verify that Fuzzy has better control performance than the PID controller. The modeling, control and effects are studied through computer simulation using MATLAB/SIMULINK toolbox.**

***Index Terms*—BLDC, FLC, GGM, PID Controller**

## I. INTRODUCTION

Conventional DC motors[8] are highly efficient and their characteristics make them suitable for use as servomotors. However, their only drawback is that they need a commutator and brushes which are subject to wear and require maintenance. When the functions of commutator and brushes were implemented by solid-state switches, maintenance-free motors were realized. These motors are now known as brushless dc motors.

Design of the BLDCM drive involves a complex process such as modeling, control scheme selection, simulation and parameters tuning etc. To achieve desired level of performance the motor requires suitable speed controllers.

Speed control of BLDC Motors is generally done using proportional integral Derivative (PID) controller. Conventional PID controllers are widely used in the industry as it has simple control structure and easy to implement but these controllers are insufficient to control the speed of BLDC motor as it gives high overshoot in the response.

Fuzzy logic control offers an improvement in the quality of the speed response. Most of these controllers use mathematical models and are sensitive to parametric variations. These controllers are inherently robust to load disturbances. Besides, fuzzy logic controllers can be easily implemented.

## II. MATHEMATICAL MODELING OF BLDC MOTOR

### A. Transfer Function

Transfer Function derived by the mathematical modeling[4],

$$G(s) = \frac{1/k_e}{\tau_m \tau_e s^2 + \tau_m s + 1}$$

Where,

$\tau_m$ = Mechanical time constant = $\frac{3RJ}{k_e k_t}$

$\tau_e$ = Electrical time constant = $\frac{L}{3R}$

R = Terminal resistance phase to phase
L = Terminal inductance phase to phase
J = Rotor inertia
$k_e$ = back emf constant (V-sec/rad)
$k_t$ = torque constant (Nm/A)

### B. Specifications of BLDC Motor[12]

| Parameters | Values |
|---|---|
| Nominal Voltage | 9 V |
| Terminal Resistance phase to phase | 1.25 Ω |
| Terminal Inductance phase to phase | 0.32 mH |
| Torque Constant | 10.4 mNm/A |
| Mechanical Time Constant | 60.5 ms |
| Rotor Inertia | 52.3 gcm$^2$ |

### C. Model of BLDC Motor

Final Transfer Function,

$$G(s) = \frac{32.0816}{5.1624 * 10^{-6}\, s^2 + 0.0605\, s + 1}$$

## III. CONTROL SCHEMES

### A. *PID Controller*

Proportional-Integral-Derivative (PID)[7] control is the most common control algorithm used in industry and has been universally accepted in industrial control. As the name suggests, PID controller algorithm involves three separate constant parameters and is accordingly sometimes called three-term control: proportional, integral and derivative.

The main aim of the PID controller is to sense the sensor signal and evaluate the desired output by calculating the proportional, integral and derivative responses and summing those three components to generate the output. Most of the time, system is affected not only by the actuator output but also by the external factors which are called as the disturbances. PID controller is usually designed to eliminate the effects of the disturbances. Figure 1 shows the typical block diagram of the PID controller.

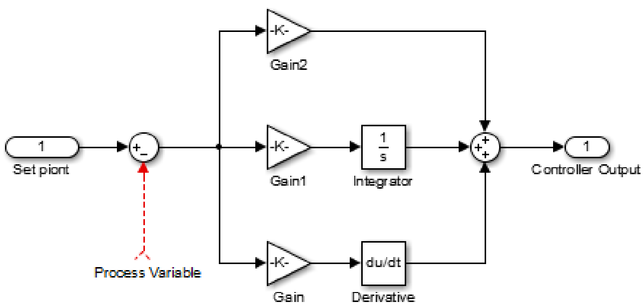


Fig. 1 : PID Block Diagram

Closed Loop Response :

| Parameters | Rise Time | Overshoot | Settling Time | S-S Error |
|---|---|---|---|---|
| **$K_P$** | Decrease | Increase | Small Change | Decrease |
| **$K_I$** | Decrease | Increase | Increase | Eliminate |
| **$K_D$** | Small Change | Decrease | Decrease | No Change |

PID Tuning :

There are various method for PID tuning, here two method are used to tune the PID for and the performance of system is compared

1. Trial and Error Method
2. Good Gain Method

#### 1) *Trial and Error Method*

It is a trial and error method[4] but a computational stability rule is needed to set a mark for its effect. This is done by using the R-H stability rule.

R-H Criteria :

Keeping $K_P$ part, set $T_i$ and $T_d$ to infinite and zero respectively, controller gain $K_C$ could be obtained that would sustain the oscillation output which is called the ultimate gain, $K_{CU}$. For proper oscillations, $K_C$ is set to be less than $K_{CU}$.

$$1 + K_{CU} * G(s) = 0$$

$$G(s) = \frac{32.0816}{5.1624 * 10^{-6}\, s^2 + 0.0605\, s + 1}$$

$$1 + K_{CU} * \frac{32.0816}{5.1624 * 10^{-6}\, s^2 + 0.0605\, s + 1} = 0$$

$$5.1624 * 10^{-6}\, s^2 + 0.0605\, s + 1 + 32.0816 * K_{CU} = 0$$

Routh's array,

| | | |
|---|---|---|
| $S^2$ | $5.1624 * 10^{-6}$ | $1 + 32.0816 * K_{CU}$ |
| $S^1$ | 0.0605 | 0 |
| $S^0$ | $1 + 32.0816 * K_{CU}$ | |

For no change in first column,

$$1 + 32.0816 * K_{CU} > 0$$

$$32.0816 * K_{CU} > -1$$

$$K_{CU} > -0.03117$$

The above result shows that the main values of $K_{CU}$ are greater than zero. With a trial and error tuning, the value of $K_P$ can be set to num of the system transfer function i.e. 32.0816. The value of $K_I$ is the inverse value of 0.03117 and the value of $K_D$ is equal to the 0.03117.

So, from trial and error method the obtained value of PID parameters are $K_P$= 32.0816, $K_I$ = 32.0821, $K_D$ = 0.03117.

#### 2) *Good Gain Method*

The Good Gain Method[10] is a simple method based on the experiments similarly to a trial and error method. It can be implemented on a simulation set up or on a real system. For tuning the parameters of PID controller, follow the steps given below.

1. Initially, set $T_i = \infty$ and $T_D = 0$ and keep increasing the value of $K_P$ until satisfactorily stable response is obtained i.e. slight overshoot and undershoot.

2. Find the value of $T_{OU}$ i.e. the time difference between first overshoot and first undershoot of the response.

3. Calculate the value of $T_i$

$$T_i = 1.5 * T_{OU}$$

4. Calculate the value of $T_D$

$$T_D = \frac{T_i}{4}$$

For the given system, the values of the parameters of PID controller are $K_P = 12$, $K_I = 1350.074$, $K_D = 1.851 * 10^{-4}$

The Simulink model for the closed loop PID controller is shown in figure 2. Here, Trial and Error method and Good Gain methods have been implemented for the tuning of PID parameters.

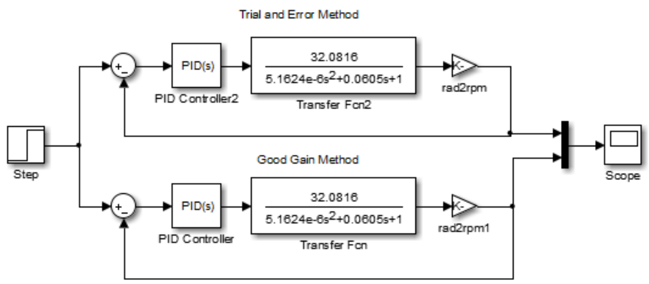


Fig. 2 : Simulink model for the closed loop PID controller

The comparison of the responses of both trial and error method and good gain method is shown in figure 3.

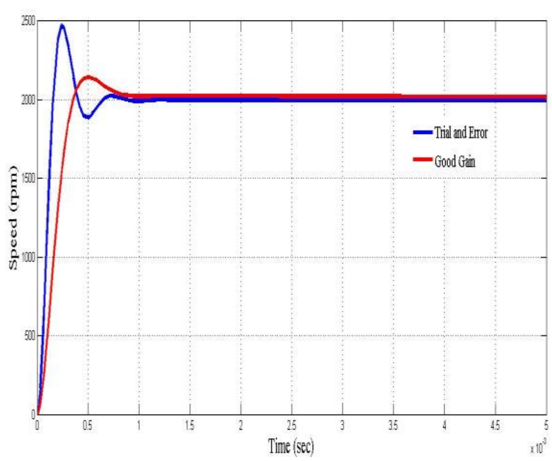


Fig. 3 : Responses of PID Controller

## B. *Fuzzy Logic Control*

The concept of Fuzzy Logic was conceived by Prof. Lotfi A. Zadeh at the University of California at Berkley in 1965. Fuzzy logic is basically a multivalued logic that allows intermediate values to be defined between conventional evaluations like yes/no, true/false, etc. Notions like rather warm or pretty cold can be formulated mathematically and algorithmically processed. In this way an attempt is made to apply a more humanlike way of thinking in the programming of computers ("soft" computing).

FIS contains three components [3]:

1. Fuzzifier : The fuzzifier takes input values and determines the degree to which they belong to each of the fuzzy sets via membership functions of fuzzy logic system.
2. Rule base : The rule base contains linguistic rules that are provided by experts. It is also possible to extract rules from numeric data. Once the rules have been established, the FIS can be viewed as a system that maps an input vector to an output vector.
3. Defuzzifier : The Defuzzifier takes the values from fuzzy sets via membership functions and gives the crisp output.

Simulink model for the Fuzzy Logic controller is shown in figure 4.

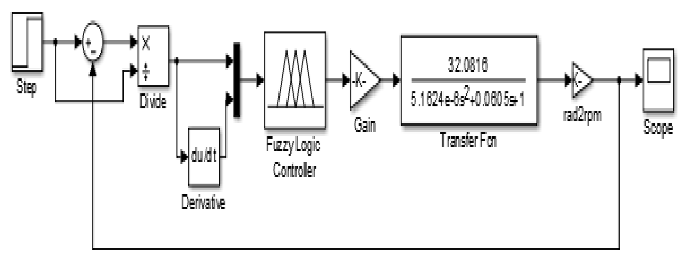


Fig. 4 : Simulink model for the Fuzzy Logic controller

The response of the Fuzzy Logic Control is shown in figure 5.

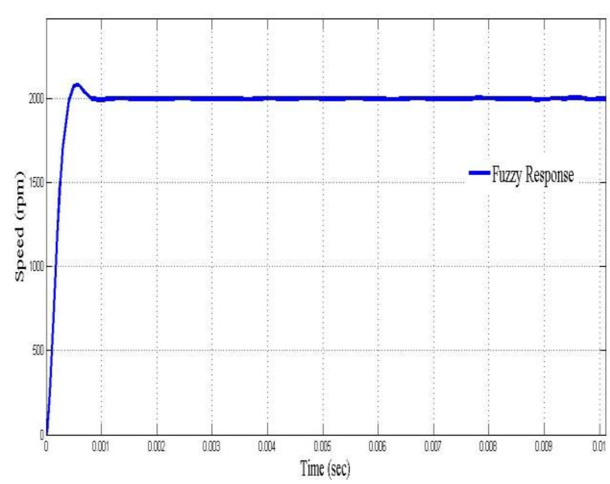


Fig. 5 : response of the Fuzzy Logic Control for speed 2000 rpm

The comparison of the responses of trial and error method, good gain method and Fuzzy Logic Control is shown in figure 6.

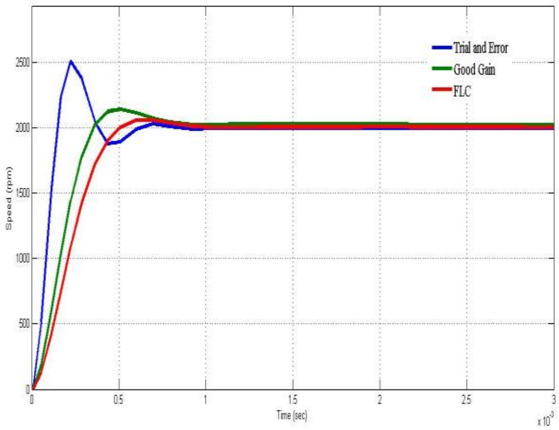


Fig. 6 : Responses of PID and FLC

## IV. RESULT

The transient parameter of the system for the different tuning parameters are obtained and shown in Table 1.

Table 1 : Transient Response for different tuning methods

| Tuning Methods | Transient Parameters | | |
|---|---|---|---|
| | Rise Time (ms) | Settling Time (ms) | Peak Overshoot (%) |
| Trial and error | 0.1253 | 0.7632 | 23.5 |
| Good Gain | 0.24 | 3.745 | 7.1 |

From Table 1, it is clearly visible that the rise time and settling time of the system for the trial and error method is much better than good gain method but good gain method gives much less peak overshoot as compared to trial and error method.

The transient parameter of the system for the different tuning parameters and Fuzzy Logic Control are obtained and shown in Table 2.

Table 2 : Transient Response for PID and FLC

| Control Schemes | Transient Parameters | | |
|---|---|---|---|
| | Rise Time (ms) | Settling Time (ms) | Peak Overshoot (%) |
| **Trial and error** | 0.1253 | 0.7632 | 23.5 |
| **Good Gain** | 0.24 | 3.745 | 7.1 |
| **FLC** | 0.339 | 0.6949 | 4.05 |

From Table 2, it is clearly visible that the settling time and peak overshoot of the system for the Fuzzy Logic Control is much better than that of good gain method but good gain method gives much less rise time as compared to Fuzzy Logic Control.

## V. CONCLUSION

With the help of simulation results and the data presented, it can be concluded that response of the system is better when PID controller parameter is tuned by GGM instead of trial and error method. The result also reveals that the application of intelligent control like FLC to the system can give a better performance than the conventional controller. Other intelligent control schemes can be used to control the system that may result in better performance of system.

## REFERENCES

[1] C. SheebaJoice, P. Nivedhitha, "Simulation of Speed Control of Brushless DC Motor with Fuzzy Logic Controller", International Journal of Electrical, Electronics and Data Communication, Volume-2, Issue-4, April-2014.
[2] Pooja Agarwal, Arpita Bose, "Brushless Dc Motor Speed Control Using Proportional- Integral and Fuzzy Controller", IOSR-JEEE, Volume 5, Issue 5 (May. - Jun. 2013), PP 68-78.
[3] Umesh Kumar Bansal and Rakesh Narvey, "Speed Control of DC Motor Using Fuzzy PID Controller", Advance in Electronic and Electric Engineering, Volume 3, Number 9 (2013).
[4] Oludayo John Oguntoyinbo, "PID Control of Brushless DC Mptpr and Robot Trajectory Planning and Simulation with MATLAB/SIMULINK," Degree Program of IT, Vaasa University of Applied Sciences, Finland, December 2009.
[5] Laurent Foulloy, Sylvie Galichet, "Fuzzy Control with Fuzzy Inputs", IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL. 11, NO. 4, AUGUST 2003.
[6] P.C. Sen, "Principles of Electric Machines and Power Electronics", John Wiley & Sons, 1997.
[7] Karl J. Astrom and Tore Hagglund, "PID Controllers : theory, design and tuning", second edition.
[8] Takashi Kenjo, Shigenobu Nagamori, "Permanent-magnet and brushless DC motors", volume 18, Oxford University Press,16 Jan 1985.
[9] Padmaraja Yedamale, ―Brushless DC (BLDC) Motor Fundamentals‖, Microchip Technology Inc. 2003.
[10] Fuzzy Logic Toolbox User's Guide © copyright 1995-1999 by The MathWorks, Inc.
[11] Finn Haugen, "The Good Gain Method for PI(D) Controller Tuning", 19 July, 2010.
[12] George W. Younkin, "Electric Servo Motor Equations and Time Constants".
[13] http://www.maxonmotor.in/maxon/view/product/motor/ecmotor/ecflat/ecflat45/200141.