International Journal of Electrical and Electronics Engineering Research (IJEEER)
ISSN(P): 2250-155X; ISSN(E): 2278-943X
Vol. 4, Issue 4, Aug 2014, 113-126

MODELING OF BRUSHLESS DC DRIVE USING GENETIC ALGORITHM BASED TUNING OF PID CONTROLLER

S. DINESH KUMAR

Research Scholar, Department of EEE, St. Peter's University, Avadi, Chennai, Tamil Nadu, India

ABSTRACT

The main core of this paper is to design a speed control system to control the Brushless DC motor speed at desired speed through the technique of soft computing based self tuning of PID which is designed in the platform of Lab view program, The application of soft computing technique to the PID controller imparts the ability of tuning itself automatically. The BLDC motors are preferred today because of the cost, afford-ability and variety they offer in choice of application involving the fractional horsepower which can be increased up to 3 Horse Power in most of the DC motors. In practice, controlled systems usually have some features, such as nonlinearity, time–variability, and time delay, which make controller parameter tuning more complex. Moreover, in some cases, system parameters and even system structure can vary with time and environment. As a result, the traditional PID parameter tuning methods are not suitable for these difficult calculations. Using genetic algorithms to perform the tuning of the controller will result in the optimum controller being evaluated for the system every time. A comparison analysis has been made to Show the BLDC drive Control with PID controller conventional tuning Genetic Algorithm based tuning. The Algorithm functions on three basic genetic operators of selection, crossover and mutation. Based on the types of these operators GA has many variants like Real coded GA, Binary coded GA, These parameters have a great influence on the stability and performance of the control system. This Paper focuses the Binary coded GA & finds the value of crossover, mutation of PID controller and the mathematical analysis has been carried out and the transfer function has been analyzed in the system with MATLAB.

KEYWORDS: BLDC Motor, PID Parameter Tuning Methods, Genetic Algorithm

INTRODUCTION

The BLDC motor is widely used in applications including appliances, automotive, aerospace, consumer, medical, automated industrial equipment and instrumentation. The BLDC motor is electrically commutated by power switches instead of brushes. Compared with a brushed DC motor or an induction motor, the BLDC motor has many advantages are Higher efficiency, reliability, Lower acoustic noise, Smaller and lighter, Greater dynamic response, Better speed versus torque characteristics, Higher speed range, Longer life.

The motor part of a brushless motor is often a permanent magnet synchronous motor, but can also be a switched reluctance motor, or induction motor. Brushless motors may be described as stepper motors; however, the term stepper motor tends to be used for motors that are designed specifically to be operated in a mode where they are frequently stopped with the rotor in a defined angular position. This paper describes more general brushless motor principles, though there is overlap. A BLDC motor is similar to DC shunt motor in which stationary field winding is replaced by permanent magnet. DC Motor plays a crucial role in research, laboratory experiments and Electric traction, high speed tools applications in an industry because of their simplicity and low cost. Therefore, the speed of DC motor can be controlled by varying flux/pole, Armature resistance and applied voltage. Here, DC motor speed is adjusted by varying the terminal voltage. Real - time Speed control of the DC motor is realized by Genetic Algorithm Based Tuning of PID Controller.

Due to its excellent speed control characteristics, the BLDC motor has been widely used in industry even though its maintenance costs are higher than the induction motor. As a result, Speed control of BLDC motor has attracted considerable research and several methods have evolved. Proportional-Integral Derivative (PID) controllers have been widely used for speed and position control of BLDC motor. This paper proposes a new method to design a speed controller of a BLDC motor by selection of PID parameters using GA To show the efficiency of GA. the results of this method are compared with simple PID Controller. Using genetic algorithms to perform the tuning of the controller results in the optimum controller being evaluated for the system every time. The objective of this paper is to show the Binary coded GA & find the value of crossover, mutation of PID controller

MODELING OF THE BLDC MOTOR

Below figure describes the basic building blocks of the BLDC motor and inverter that results in a system producing a linear speed-torque characteristic similar to the conventional DC motor. BLDC motor has three phase windings on the stator similar to three phase squirrel cage induction motor and magnets are placed on the rotor to provide air gap flux resulting in brushless rotor construction. When the motor is operated at a certain speed, trapezoidal emfs are induced in stator phase windings. The quasi-square wave AC current is fed to stator phase windings through electronic commutator using current controlled voltage source inverter and rotor position sensor resulting in constant torque development by the motor.

Modeling of a BLDC motor can be developed in the similar manner as a three phase synchronous machine. Since its rotor is mounted with a permanent magnet, some dynamic characteristics are different. Flux linkage from the rotor is dependent upon the magnet. Therefore, saturation of magnetic flux linkage is typical for this kind of motors. As any typical three phase motors, one structure of the BLDC motor is fed by a three phase voltage source as shown in below figure. The source is not necessary to be sinusoidal. Square wave or other wave- shape can be applied as long as the peak voltage is not exceeded the maximum voltage limit of the motor. Similarly, the model of the armature winding for the BLDC motor is expressed as follows.

$$\mathbf{v_a} = \mathbf{R i_a} + \mathbf{L}\frac{\mathbf{d i_a}}{\mathbf{dt}} + \mathbf{e_a}$$
$$\mathbf{v_b} = \mathbf{R i_b} + \mathbf{L}\frac{\mathbf{d i_b}}{\mathbf{dt}} + \mathbf{e_b}$$
$$\mathbf{v_c} = \mathbf{R i_c} + \mathbf{L}\frac{\mathbf{d i_c}}{\mathbf{dt}} + \mathbf{e_c}$$

Or in the compact matrix form as follows.

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R+pL & 0 & 0 \\ 0 & R+pL & 0 \\ 0 & 0 & R+pL \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix}$$

Where: $L_a = L_b = L_c = L = L_s = M$

$L_a = L_b = L_c = L = L_s$ = Armature of self inductance

M= Mutual Inductance

$R_a = R_b = R_c = R$ = Armature resistance in ohm

v_a, v_b, v_c = Terminal phase voltage in volts

i_a, i_b, i_c = Motor input current in amperes

e_a, e_b, e_c = Motor back emf in volts

p is the matrix represents d/dt

Due to the permanent magnet mounted on the rotor, its back emf is trapezoidal as below shown in figure

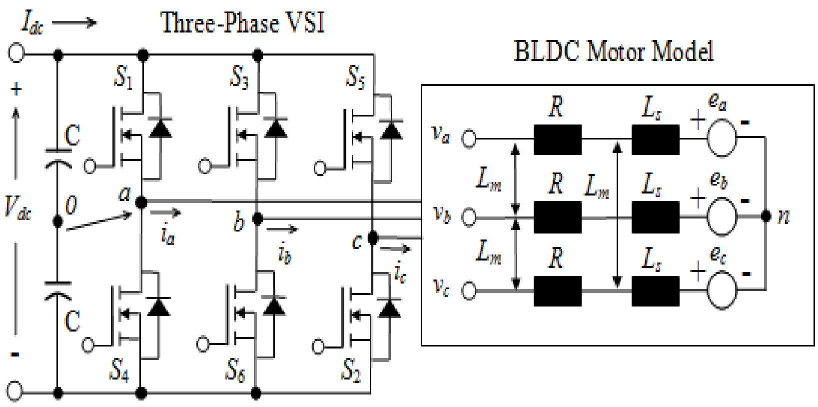

Figure 1: Configuration of the BLDC Motor and Inverter System

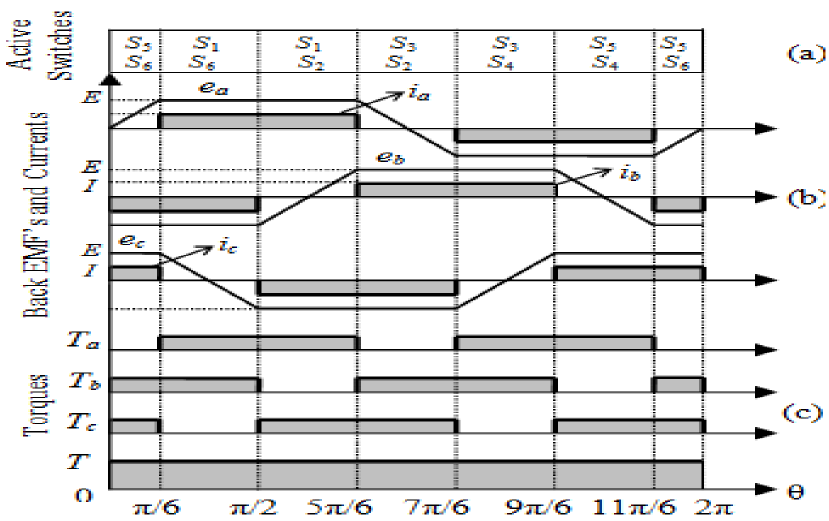

Figure 2: a) Active Switches, (b) Back EMF and Phase Current Waveforms, and (c) Three Phase Torques of the BLDC Motor Drive System

MATHEMATICAL MODELING

The DC motor mathematical model provided below where:

R: The Armature Resistance,

L: The Armature Inductance,

i: The Armature Current,

Ea: The Input Voltage,

I: The Field Current,

e: The Back Electromotive Force (EMF),

T: The Motor Torque,

v: Angular Velocity of Rotor,

J: Rotating Inertial Measurement of Motor Bearing,

B: A Damping Coefficient

Because the back EMF e_b is proportional to speed ω directly, then

$$e_b(t) = K_b\frac{d\theta(t)}{dt} = K_b\omega(t) \tag{1}$$

Making use of the KCL voltage law can get

$$e_a(t) = R_a i_a(t) + L_a\frac{di_a(t)}{dt} + e_b(t) \tag{2}$$

From Newton law, the motor torque can obtain

$$T_m(t) = J\frac{d^2\theta(t)}{dt^2} + B\frac{d\theta}{dt} = K_T i_a(t) \tag{3}$$

Take (1), (2), and (3) into Laplace transform respectively, the equations can be formulated as follows:

$$E_a(s) = (R_a + L_a s)I_a(s) + E_b(s)$$

$$E_a(s) = K_b\Omega(s)$$

$$T_m(s) = B\Omega(s) + Js\Omega(s) = K_T I_a(s) \tag{4}$$

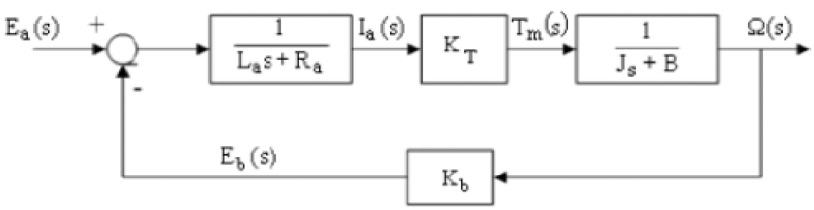

Figure 3: Function Block Diagram of DC Motor

The transfer function of DC motor speed with respect to the input voltage can be written as follows:

$$G(s) = \frac{\Omega(s)}{E_a(s)} = \frac{K_T}{(L_a s + R_a)(Js + B) + K_b K_T} \tag{5}$$

From the above Equation the armature inductance is very small in practices, hence, the transfer function of DC motor speed to the input voltage can be simplified as follows,

$$\frac{\Omega(s)}{E_a(s)} = \frac{K_m}{\tau s + 1} \tag{6}$$

$$K_m = \frac{K_T}{R_a B + K_b K_T}, \quad \tau = \frac{R_a J}{R_a B + K_a K_T}$$

Where: **Km**= motor gain, Ʈ=motor time constant.

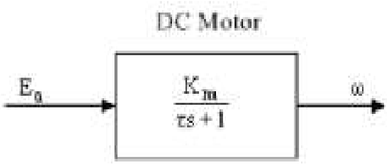

Figure 4

DESIGN AND IMPLEMENTATION OF PID CONTROLLER

PID controllers are the most widely-used type of controller for industrial applications. They are structurally simple and exhibit robust performance over a wide range of operating conditions. In the absence of the complete knowledge of the process these types of controllers are the most efficient of choices. The three main parameters involved are Proportional (P), Integral (I) and Derivative (D). The proportional part is responsible for following the desired set-point, while the integral and derivative part account for the accumulation of past errors and the rate of change of error in the process respectively.

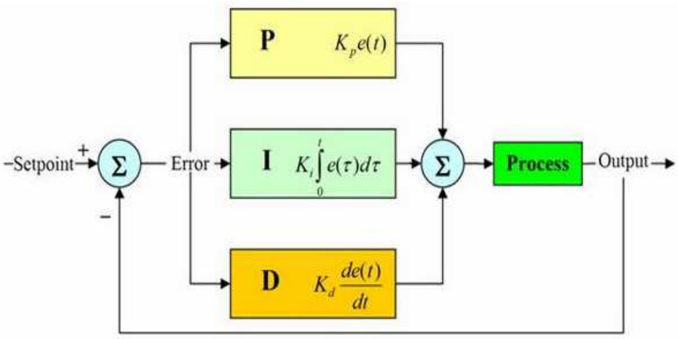

Figure 5: The Schematic Diagram of PID Controller

The relationship between the input e (t) and output u(t) can be formulated in the following

$$u(t) = K_p e(t) + K_I . \int_0^t e(t)\, dt + K_D \frac{de(t)}{dt} \tag{7}$$

The above equation can be expressed as follows.

$$C(s) = \frac{U(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D s \tag{8}$$

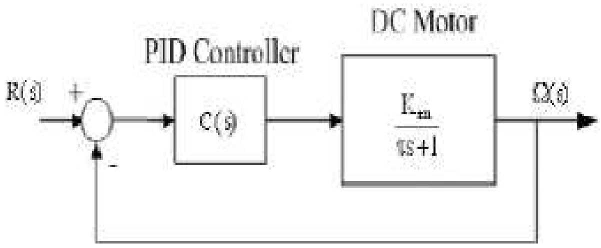

Figure 6: The Block Diagram Representation for PID Based BLDC

Closed loop transfer function of the DC motor speed control system.

$$G(s) = \frac{\Omega(s)}{R(s)} = \frac{(K_P + \frac{K_I}{s} K_D s)\frac{K_m}{1+\tau s}}{1 + (K_P + \frac{K_I}{s} + K_D s)\frac{K_m}{1+\tau s}} = \frac{(K_D s^2 + K_P s + K_I)K_m}{(K_D K_m + \tau)s^2 + (1 + K_P K_m)s + K_I K_m} \tag{9}$$

Tuning method for PID controller is very important for the process industries.

Here we are using BLDC motor with PID Controller without G.A. By the use of MATLAB programming PID controller is tuned & the result is given below

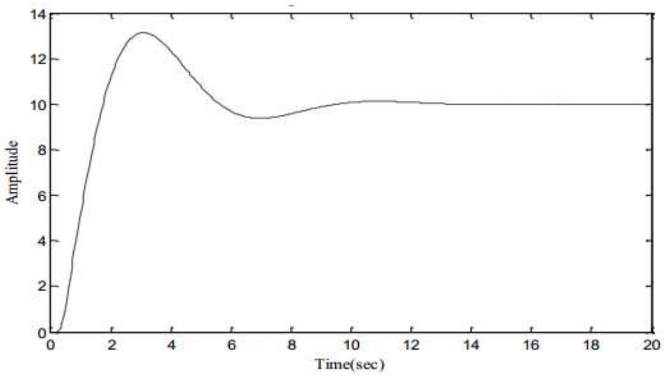

Figure 7

From the above response, we can analyze the following parameter:

- Rise Time, t_r
- Maximum Overshoot, M_p
- Settling time, t_s

The rise time, t_r is the time taken to reach 10 to 90 % of the final value is about 0.5 sec. The Maximum Overshoot, M_p of the system is approximately 1.01. Finally the Settling time, t_s is about 0.25sec. From the analysis above, the system has not been tuned to its optimum. So in order to achieve the following parameters we have to go for genetic algorithm approach. Our system requirements are given below:

Table 1

Specification	M_p	t_r	t_s
	<1.01	< 0.5	<0.25sec

GENETIC ALGORITHM

Introduction

GA is a stochastic global adaptive search optimization technique based on the mechanisms of natural selection. Recently, GA has been recognized as an effective and efficient technique to solve optimization problems and Compared with other optimization techniques. GA starts with an initial population containing a number of chromosomes where each one represents a solution of the problem which performance is evaluated by a fitness function. Basically, GA consists of three main stages: Selection, Crossover and Mutation. The application of these three basic operations allows the creation of new individuals which may be better than their parents. This algorithm is repeated for many generations and finally stops when reaching individuals that represent the optimum solution to the problem. The GA architecture is below shown in Figure

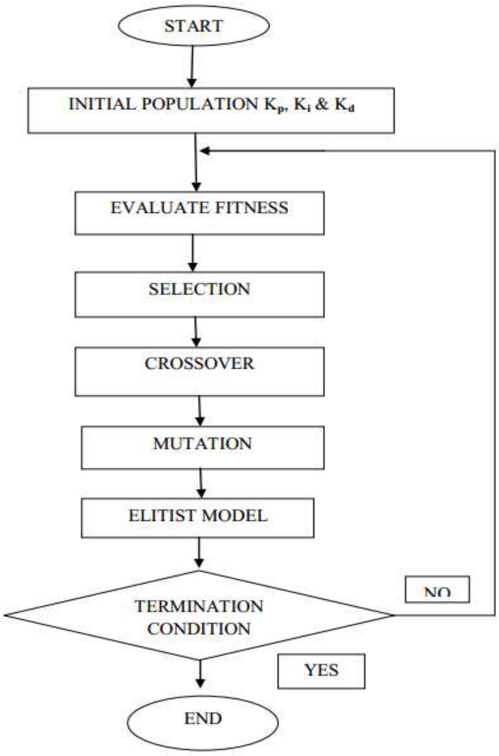

Figure 8

Objective Function of the Genetic Algorithm

The most challenging part of creating a genetic algorithm is writing the objective functions. In this project, the objective function is required to evaluate the best PID controller for the system. An objective function could be created to find a PID controller that gives the smallest overshoot, fastest rise time or quickest settling time. However in order to combine all of these objectives an objective function is designed to minimize the performance indices of the controlled system instead.

Overview of Binary Coded G. A

GA has many variants like Real coded GA, Binary coded GA, Saw tooth GA, Micro GA, Improved GA, Differential Evolution GA. This paper is based on Binary coded G.A. The binary coded genetic algorithm is a probabilistic search algorithm that iteratively transforms a set (called a population) of mathematical objects (typically fixed-length binary character strings), each with an associated fitness value, into a new population of offspring objects using the Darwinian principle of natural selection and using operations that are patterned after naturally occurring genetic operations, such as crossover and mutation.

- **Encoding**: In genetic Algorithm, coding is expressing the individual by the binary strings of 0‟s & 1‟s. In the instance one every individual has there dimension and every dimension is expressed by a 10- bit string of 0‟s & 1‟s.
- **Selection:** The selection operator selects chromosomes from the current generation to be parents for the next generation. In this method, a few good chromosomes are used for creating new offspring in every iteration. Then some bad chromosomes are removed and the new offspring is placed in their places. The rest of population migrates to the next generation without going through selection process.
- **Crossover**: Crossover is the GA's primary local search routine. The crossover/reproduction operator computes two offspring for each parent pair given from the selection operator. The crossover operator is used to create new solutions from the existing solutions available in the mating pool after applying selection operator. This operator exchanges the gene information between the solutions in the mating pool. The most popular crossover selects any two solutions strings randomly from the mating pool and some portion of the strings is exchanged between the strings. The selection point is selected randomly. A probability of crossover is also introduced in order to give freedom to an individual solution string to determine whether the solution would go for crossover or not.
- **Mutation:** Mutations are global searches. A probability of mutation is again predetermined before the algorithm is started which is applied to each individual bit of each offspring chromosome to determine if it is to be inverted. Mutation changes the structure of the string by changing the value of a bit chosen at random. Mutation is the occasional introduction of new features in to the solution strings of the population pool to maintain diversity in the population. Though crossover has the main responsibility to search for the optimal solution, mutation is also used for this purpose. The mutation probability is generally kept low for steady convergence. A high value of mutation probability would search here and there like a random search technique.

TUNING THE PID CONTROLLER

Although you'll find many methods and theories on tuning a PID, here's a straight forward approach to get you up and soloing quickly.

- SET K_P. Starting with K_P=0, K_I=0 and K_D=0, increase K_P until the output starts overshooting and ringing significantly.
- SET K_D. Increase K_D until the overshoot is reduced to an acceptable level.
- SET K_I. Increase K_I until the final error is equal to zero.

Hands-On Design Run a simulation of the circuit files PID1.CIR. VSET generates a 10V step input voltage to the control system. You can adjust the PID terms at the EPID source that adds the P, I and D terms at V (3), V (4) and V (5). Initially, the PID multipliers are set to K_P=1, K_I=0 and K_D=0.

EPID 6 0 POLY (3) (3, 0) (4, 0) (5, 0) 0 1 0 0

SET K_P

Plot the system input V (1) and the sensor output (12V). Although the response looks smooth, what is the sensor voltage compared to the desired 10V? The output falls short by 5V and to reduce this error, increase K_P to 10 (Change EPID to look like... 10 0 0). The output now reaches 9V, reducing the error to 1V. But as you can see, the output is getting wild with overshoot and ringing. Push K_P up higher to 20 or 30. Yes, the error reduces, but the overshoot gets worse. Eventually, your system will become unstable and break out into song (oscillate). Back off K_P to 20 or so.

SET K_D

The derivative term can rescue the response by counteracting the K_P drive when the output is changing. Start with a small value like K_D=0.2 and rerun the simulation (Change EPID to look like. 20 0 0.2). Now you're wrestling control back into the system - the ringing and overshoot are reduced! Crank up K_D some more. Improvement should continue to a point where the system becomes less stable and overshoot increases again. Return K_D to around 0.5.

SET K_I

With K_P=20, K_I=0 and K_D=0.5 the response looks respectable, but the final error is a disappointing 0.5V Now, try the K_I term. This will integrate the remaining error into a drive signal big enough to reduce the error further. Start with K_I = 10 (EPID should look like... 20 10 0. 5). Check out the last half of the V (12) - the sensor output moves slowly toward 10V and you might want to put up a cursor on the plot to monitor the exact value of V (12). The bigger you make K_I, the faster it will move toward 10V. Like the other terms, a value is reached where the K_I does more harm than good as the system becomes less stable.

BLDC MOTOR CONTROL SYSTEM

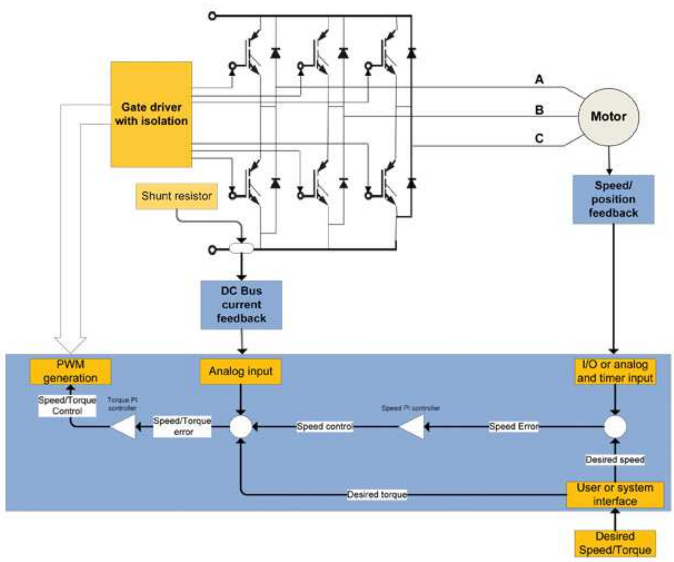

Figure 9: BLDC Control Block Diagram

Notice that this block-level representation works for both sensored and sensor less BLDC motor controllers. The main difference lies in the speed feedback and input sections. A typical sensored BLDC application will use three Hall Effect sensors as the speed/position feedback; the controller needs only input pins and a timer to detect the switching pattern and switching period. A sensor less algorithm, on the other hand, will measure the BEMF (back-electromagnetic force) voltage generated by the phase of the motor that is not being driven in a specific moment, to measure the rate of change of this voltage and deduce the motor speed based on that measurement. This requires an ADC or analog comparator and a timer, plus some more complex code.

Dead Time Insertion and Compensation

Many different control algorithms have been used to provide control of BLDC motors. Typically, the motor voltage is controlled using a power transistor operating as a linear voltage regulator. This is not practical when driving higher-power motors. High-power motors must use PWM control and require a microcontroller to provide starting and control functions.

The control algorithm must provide three things

- PWM voltage to control the motor speed
- Mechanism to commutate the motor
- Method to estimate the rotor position using the back-EMF or Hall Sensors

PROPOSED WORK RESULT AND ANALYSIS

Implementation of G.A Based PID Controller

In the proposed work a DC Motor model is called by a program which is coded in MATLAB for a fitness function i.e., cost function. In order to use GA to tune the PID controller for DC motor. Variables K_P, K_I, & K_D are coded to solve string structures. Binary coded string having 1"s & 0"s are mostly used. The length of string is usually determined according to the desired solution accuracy. Here 10 bits are used to code each variable. We can use 8 bit & 4 bit also. Thereafter select the random strings from the population to form the mating pool. In order to use roulette-wheel selection procedure, we calculate the average fitness of the population. Then the mating pool strings are used in the crossover operation. The next step is to perform mutation on strings in the intermediate population. The resulting population becomes the new population. The whole process is coded in MATLAB & after running the program we get the optimized values of K_P, K_I & K_D. The simulation modal for the entire system is given below and also the genetic algorithm parameters are chosen for the optimization.

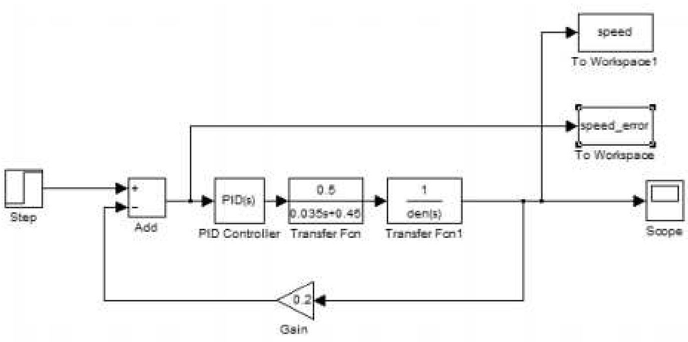

Figure 10: Simulation Modal of GA Based PID Controller

Table 2: G.A. Parameters

Parameter	Value
Population size	20
Iteration	05
Crossover Probability	> 0.8
Mutation Probability	< 0.05

From the above table all the parameter values are apply to G.A the PID controller with DC motor is optimized. The system response is given below

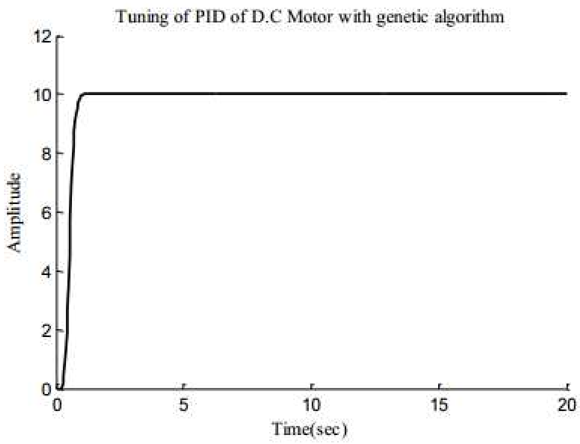

Figure 11: Result of G.A Based PID Controller

Table 3: G. A. Parameters

Parameter value	M_p	t_r	t_s
PID with G.A	0.004	0.303	0.943

Analysis of Result

In order to analysis the performance characteristics of BLDC drive the following Observations are discussed.

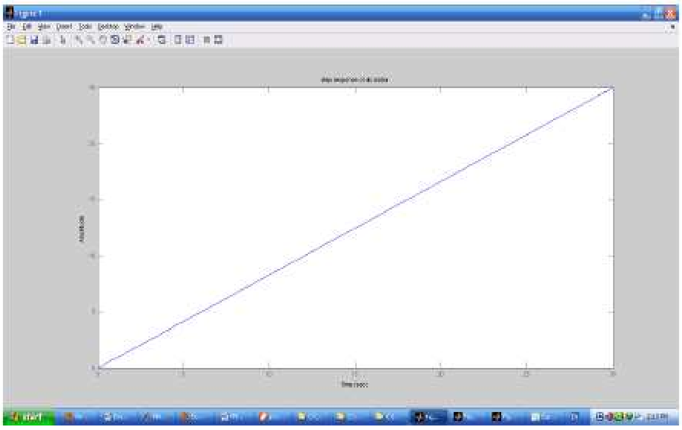

Figure 12: Step Response of the Motor

Motor Transfer Function is Given as

$1.2 / (0.00077S^3 + 0.0539\ S^2 + 1.441\ S)$

The time range is between 0 to 30 sec with a division of multiples of 5 Sec. When the given transfer function is applied with a step response, the observation is obtained as per the figure 13. BLDC drive performance with application PID Conventional tuning in MATLAB Tool.

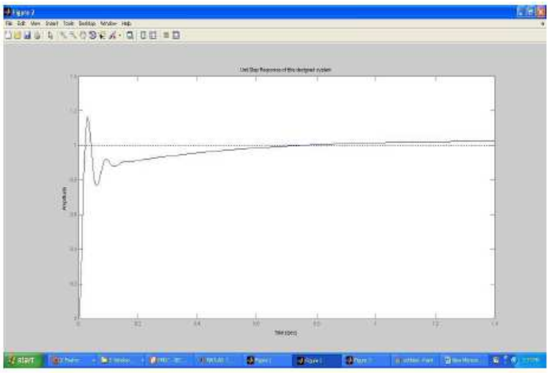

Figure 13: The Response of the Conventional Tuning Based PID Controller

As the mathematical model is solved with Mat lab with PID controller conventional tuning the dynamic system response is not approach to its steady state immediately but it will be having some damping oscillations initially. The main problem of the system is the in appropriate the steady state parameter. The gain constants of the PID controller are motioned below. Parameters obtained in PID based conventional tuning:

K_P=16, K_I=0.37, K_D= 0.017

- ***Rise Time (t_r) = 0.2Sec***
- ***Settling Time (t_s) = 0.25Sec***
- ***Steady State Error (e_{ss}) = 0.36***

Response of BLDC in Genetic Algorithm based Approach

When unstable system is applied with Genetic algorithm for its PID controller the response will be like above mentioned figure 14. As the problems of instability observed with conventional tuning based PID controller, those parameters are tuned in a better manner.

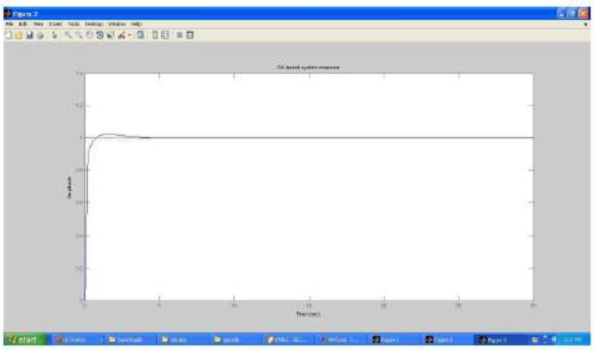

Figure 14: The Response of BLDC for Genetic Algorithm Based PID Controller Tuning

The observed values of gain parameters of PID and improved steady state response parameters are mentioned below.

K_P = 18.5; K_I =0.125; K_D= 0.53 values of the PID controller tuning under GA.

- ***Rise Time (t_r) =0.1Sec***
- ***Settling Time (t_s) = 0.1 Sec***
- ***Steady State Error (e_{ss}) =0.003***

CONCLUSIONS

In this paper a comparison analysis of BLDC drive performance based on conventional PID tuning and GA based PID tuning has presented. The system mathematical analysis has carried out and implemented with MATLAB. The behavior of BLDC drive response has verified with the Genetic algorithm based tuning of PID controller. The changes in terms of peak overshoot, rise time and settling time founded in considerable manner when compared to conventional tuning of PID controller. This paper provides the complete original binary coded G.A program in mat lab, which can be directly run through MAT LAB 7.10. G.A is applied to find optimal solution for the parameter of DC motor with PID controller & indicates that G.A is powerful global searching method. The final conclusion has drawn that GA based PID tuning can be implemented for many industrial applications. In the future Binary coded G.A. is implementing to other system of the plant for the speed control by using PID controller.

REFERENCES

1. J. Tang and R. Chasseing, “PID Controller Using theTMS320C31DSK for Real-Time DC Motor Control,” in Proceedings of the 1999 Texas Instruments DSPS Fest, Houston, Texas, August 1999.
2. Mukesh Kirar, Pankaj Swarnkar, Shailendra Jain, R. K. Nema, “Simulink based comparative study of PI, PD, PID Controllers for DC Servomotor”, 2nd National Conference on Power Electronics and Intelligent Control, 2009.
3. Neenu Thomas, Dr. P. Poongodi, 2009, “Position Control of DC Motor Using Genetic Algorithm Based PID Controller”, Proceedings of the World Congress on Engineering.
4. Huafen Yang, Dechun Dong, Youjun Ren, Jian Xu, siyuan yang, 2011, "Study on Fuzzy PID Control in Double Closed- Loop DC Speed Regulation System”, Third International Conference on Measuring Technology and Mechatronics Automation, p. p 465469.
5. Haidong Yu, 2011, "A Novel Control Strategy of Output Voltage Ripple Reduction in Buck Converters", Twenty-Sixth Annual IEEE Applied Power Electronics Conference and Exposition (APEC), p. p 1910-1914.
6. Neenu Thomas, Dr. P. Poongodi, "Position Control of DC Motor Using Genetic Algorithm Based PID Controller”, Proceedings of the World Congress on Engineering 2009 Vol. II, WCE 2009, July, 2009.
7. M. M. Kanai, J. N. Nderu, P. K. Hinga, “Adaptive PID Dc Motor Speed Controller with Parameters Optimized in Hybrid Optimization Strategy”, 2nd International conference on Advances in Engineering and Technology 2011.
8. Ayman A. Aly, “PID Parameters Optimization Using Genetic Algorithm Technique for Electro hydraulic Servo Control System”, Intelligent Control and Automation, 2011, 2, 69-76.

9. E. Cerruto, A. Consoli, A. Raciti, and A. Testa, “A robust adaptive controller for PM motor drives in robotic applications”. Power Electronics, IEEE Transactions on, 10(1): 62–71, 1995.

10. Kiran Raj A S et. al., “Lab view Based PID Speed Control and System Identification of a PMDC Motor,” in Proceeding of International Conference on Computing and Control Engineering, April, 2012.

11. P. Jagadeesh Chandra and C. R. Venugopal “Novel Design Solutions for Remote Access, Acquire and Control of Laboratory Experiments on DC Machines,” IEEE Journal on Measurement & Instrumentation Vol. 61, No. 2, Feb. 2012, pp. 349-357.