Journal of Power Technologies 100 (2) (2020) 161–170

Closed Loop speed control of BLDC Motor Drive by using classical controllers with Genetic Algorithm

Upama Das [*], Pabitra Kumar Biswas [†]

Abstract

Permanent magnet brushless DC motors (PMBLDC) find broad applications in industries due to their huge power density, efficiency, low maintenance, low cost, quiet operation, compact form and ease of control. The motor needs suitable speed controllers to conduct the required level of interpretation. As with PI controller, PID controller, fuzzy logic, genetic algorithms, neural networks, PWM control, and sensorless control, there are several methods for managing the BLDC motor. Generally, speed control is provided by a proportional-integral (PI) controller if permanent magnet motors are involved. Although standard PI controllers are extensively used in industry owing to their simple control structure and execution, these controllers have a few control complexities such as nonlinearity, load disruption, and parametric variations. Besides, PI controllers need more precise linear mathematical models. This statement reflects the use of Classic Controller and Genetic Algorithm Based PI, PID Controller with the BLDC motor drive. The technique is used to regulate velocity, direct the BLDC motor drive system's improved dynamic behavior, resolve the immune load problem and handle changes in parameters. Classical control & GA-based control provides qualitative velocity reaction enhancement. This article focuses on exploring and estimating the efficiency of a continuous brushless DC motor (PMBLDC) drive, regulated as a current controller by various combinations of Classical Controllers such as PI, GA-based PI, PID Controller. The controllers are simulated using MATLAB software for the BLDC motor drive.

Keywords: BLDC motor, closed-loop control, conventional controllers, Genetic Algorithm.

1 Introduction

The BLDC motor is commonly used in applications such as appliances, automotive, aerospace, consumer, medical, automated industrial equipment and instrumentation due to its high efficiency, low volume, high strength, and easy system design. BLDC motors also find application in a light motor cycle powered by fuel cell energy for mass production [1]. Instead of brushes, the BLDC motor is electrically switched by power switches. The BLDC motor has many benefits compared to a brushed DC motor or an induction motor: Greater effectiveness, reliability, lower acoustic noise, smaller and lighter, higher dynamic response, better speed versus torque characteristics, higher speed range, longer life. A brushless motor's motor portion is often a synchronous motor with a permanent magnet, but it can also be an induction motor or a switched reluctance motor. Brushless motors can be described as stepper motors, but the term stepper motor tends to be used for motors explicitly designed to work in a mode where they are often stopped at a defined angular position with the rotor. The BLDC motor is comparable to the DC shunt motor, where static field winding is substituted by a permanent magnet; alternatively, we can say that it is an AC synchronous motor with permanent magnets on the rotor (moving portion) and stator windings (fix portion). The rotor flux is created by permanent magnets and the windings of the energized stator generate electromagnetic poles. The stage of the energized stator attracts the rotor (equal to a bar magnet). A rotating field on the stator is developed and preserved by using a suitable sequence to supply the stator stages. This action of the rotor is the basic action used in synchronous permanent magnet motors-chasing after the electromagnet poles on the stator. In order to produce torque, the lead between the rotor and the rotor field must be controlled and this synchronization implies knowledge of the position of the rotor [2]. Normally, three Hall sensors are used to identify the position of the rotor and the switching is done based on the Hall sensor inputs.

Due to its simplicity and low price, DC Motor plays an important role in studies, laboratory experiments, electrical traction and high-speed tool apps in the sector. The BLDC motor has a resemblance to the DC motor and the DC motor speed can be regulated by variable flux/pole, armature resistance and applied voltage. The BLDC motor has been commonly used in the sector due to its outstanding velocity control features, although the maintenance costs are greater than for the induction motor. As a consequence, BLDC motor speed control has attracted

[*] NIT Mizoram e-mail

[†] NIT Mizoram e-mail

intensive study and several techniques have been developed. In practice, the BLDCM drive design includes a complicated process like modeling, choice of control schemes, simulation and tuning of parameters, etc. [3]; [4]. To achieve optimum efficiency, an expert understanding of the scheme is needed to tune the controller parameters of the BLDC Drive scheme. Several contemporary control solutions have recently been suggested to optimize BLDC motor control design [5]; [6]. These techniques, however, are complicated in nature and involve excessive computation. Proportional-Integral-Derivative (PID) controllers were commonly used to regulate BLDC engine velocity and position. On the other hand, PID control action offers an easy and efficient option for exerting control [7]. Though this control action provides a simple interface, the difficulty arises at the time of achieving the ultimate PID benefits. Continuous advances in computational system performance make Genetic Algorithms (GA) ideal for seeking a globally optimal system control solution such as looking for optimal PID controller parameters [8]; [9]. This article proposes a fresh technique for designing a BLDC engine speed controller by selecting PID parameters using GA to demonstrate GA's efficiency and the outcomes of this method. The optimum controller is assessed for the system using GAs to achieve the controller tuning outcomes.

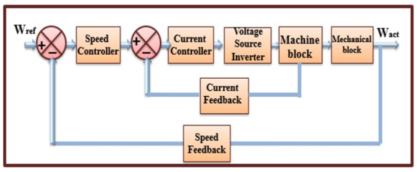

Figure 1: Schematic Block Diagram of the closed-loop system of BLDC Motor

As shown in Fig. 1, for the corresponding velocity and present controllers the classical controllers are used, which provide signals to inverter switches that supply the electrical and mechanical blocks to the BLDC machine, as shown in the block diagram. For comparison with the reference speed and current, feedback is provided so that the controllers act and settle the output at the desired value for any deviation in reference values. Inherently unstable, the BLDC system is highly non-linear in nature. While a classical controller can be built based on a linear model, choice and design of such a controller is always a difficult task, as both the stability and efficiency of such critical structures need to be assured. A schematic diagram of the generally closed-loop system is shown in Fig. 1. As can be seen, the system uses two control loops, an internal current loop and an external position loop comparable to a typical servo position scheme. Here, the current loop reaction is assumed to be quicker than the outer loop position, as the electrical time limit of the inner loop is much lower than the mechanical moment constant of the outer loop. Control loop design starts from the innermost (fastest) loop and proceeds to the slowest loop, which is the outer loop in this case. The reason to proceed in the design process from the inner to the outer loop is that it is necessary to solve the gain and time constants of only one controller at a time, instead of simultaneously solving all controllers for the gain and time constants. This section considers the design of present and position controllers. The controllers are intended for automatic tuning and loop evaluation based on a good insight into PID tuning, which is also helpful in the development of other schemes. The guidelines for tuning the PID controller from Ziegler–Nichols were very influential. However, the rules have serious disadvantages, they use inadequate process data, and the design criterion provides bad robustness to closed-loop systems [2]. Two techniques, a step response method, and a frequency response method were provided by Ziegler and Nichols.

2 Classical Controller

The automatic controller determines the value of the controlled variable, compares the real value with the required value, determines the deviation, and generates a control signal that reduces the deviation to zero or to a minimal value. Controllers are categorized depending on the type of control action used. There are primarily three kinds of control actions and a mixture of control actions is also possible. Two types of compensators were also used and various combinations are common for managing systems other than through these control actions.

Classical control actions include proportional, integral and derivative actions. In a controller with proportional control action, there is a continuous linear relationship between the output of the controller manipulated variable (M)and actuating error signal (E) and, relating these two signals, we obtain the proportional gain known as Kp, as shown in equation 1.

$$K_p = \frac{M(s)}{E(s)} \tag{1}$$

Proportional Controller reduces the steady-state error, but never eliminates it. This controller can be used to speed up the slow response of an overdamped system. This means rise time decreases. Some offset occurs due to the presence of this controller. It also increases the maximum overshoot of the system. To minimize the overshoot, a combination of proportional and integral control action might be preferred. The transfer function of this control action is shown in equation 2. This combination gives exact control output, meaning the steady-state error is zero and damping also improves. But it has some drawbacks like an increase in rising time and settling time, which means it takes longer to achieve stability.

$$\frac{M(s)}{E(s)} = K_p(1 + \frac{1}{sT}) \tag{2}$$

As the PI controller uses only two controller actions, this type of control takes longer to stabilize output and it is better to combine derivative action. This type of combined controller is called a PID controller or 3-action controller. Here, all three gains, i.e. proportional, integral and derivative action, can be varied to control rise time, offset and maximum overshoot and settling time respectively. The transfer function of the following control action is given below in equation 3. This type of classical controller gives almost optimum output if the gain parameters are tuned properly. But sometimes it may show some disadvantages, as PID Controllers are linear, and in particular symmetric. Hence, it has variable performance in the non-linear system (particularly HVAC). The derivative term amplifies higher frequency measurement/process noise, which can cause considerable changes in output. Table 1 shows acomparison between controller properties.

Rich media available at

$$\frac{M(s)}{E(s)} = K_p(1 + \frac{1}{sT} + sT) \tag{3}$$

Table 1: Comparison between Controller Properties

Characteristic	PI	PD	PID
Rise time	Decreases	Decreases	Decreases
Overshoot	Increases	Decreases	Decreases
Bandwidth	Decreases	Increases	Increases
Response	Stable	Stable	Stable
Oscillations	No oscillation	Slight	No oscillation
Transient response	Decreases	Increases	Increases
Steady state error	Eliminates	Reduces	Eliminates

3 Genetic Algorithm as an optimization technique

GA is a technique for solving restricted and unconstrained issues of optimization based on natural selection, the process that drives biological development. A population of individual alternatives is constantly modified by the GA. The GA chooses people at random from the present population as parents at each step and utilizes them for the next generation to generate kids. The population "evolves" towards an ideal solution over consecutive generations. You can use the GA to address a range of optimization issues that are not well adapted to conventional optimization algorithms, including issues where the objective function is discontinuous, non-differentiable, stochastic, or extremely nonlinear. The GA can solve mixed-integer programming issues, where some parts are limited to being evaluated in their entirety. GAs differ significantly from the more traditional techniques of search and optimization. GAs search in parallel, not from a single point but from a population of points. It does not involve derivative data or other auxiliary knowledge; the direction of the search is influenced only by the objective function and associated fitness levels. The GA uses probabilistic transition rules and not deterministic rules. The genetic algorithm works on encoding a set of parameters, not the parameter set itself. GAs can provide a range of prospective alternatives to a particular issue and it is up to the user to choose the final. The exact operation of the GA involves the following steps. The figure below shows the hierarchy of the GA algorithm operation.

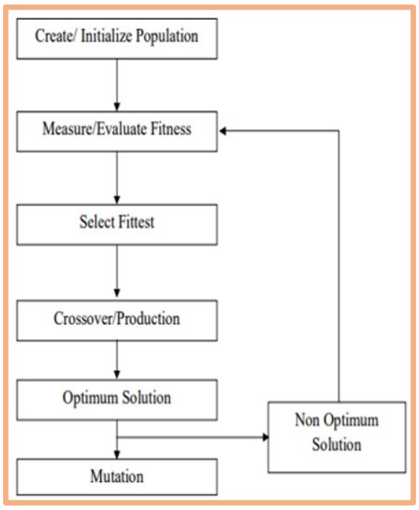

Figure 2: Process Step of Genetic Algorithm

3.1 Reproduction

Elected people from the current population can create a portion of the new people by copying without change. New populations also have the option of selecting alternatives that have already been created. Several other techniques of choice are accessible and it is up to an individual to select a suitable method for each method. The crossover proportion of reproduction is 0.8.

3.2 Crossover

New people are usually developed as two descendants (i.e. as a binary operator crossover). Any or more purported crossing points are picked (typically at arbitrarily) at the same place in every chromosome of each parent. Then parents interchange the components surrounded by the crossover points. The resulting people are the descendants. There are several forms of convergence outside one point and converge with different locations. The so-called arithmetic crossover produces a descendant in the latter stages of evolution as a wise linear component mixture of the parents. Maintaining people intact is more desirable, so using an adaptively evolving crossover rate is a good idea: Greater rates in early stages and reduced rates at the end of the GA.

3.3 Mutation

By making changes to a chosen person, a fresh person is developed. The improvements may comprise altering one or more representation standards or adding/deleting depiction parts. A mutation is a cause of variation in the GA, and an excessive rate of mutation outcomes in less effective evolution, except for particularly easy issues. This should, therefore, be used cautiously as it is a hit and miss investigate hand; otherwise, the algorithm will turn into slightly new explore with elevated mutation rates. In addition, separate mutation operators can be used at distinct phases. Mutation operators could be preferred at the start, leading to larger gaps in the rummage around the room. Further, when a mutation operator closes the solution, which results in slight changes, the search space could be preferred.

3.4 Fitness Function

A feature that takes as input a candidate solution to the issue and generates as output how “fit” or “good” the solution is in an account of the issue.

4 Closed-loop model for BLDC motor drive

Closed-Loop Model of BLDC Motor Drive As shown in 3 the closed-loop block diagram model consists of a two-loop structure, of which the outer loop is the speed control loop and the inner loop is the current control loop. As depicted in 3 the reference speed is compared with the actual speed then the error signal is fed to the speed controller, which converts the signal to reference current signal, which is compared with the actual current signal and the error is fed to the current controller, which can be any of the classical controllers, the signal received is fed to the inverter, which provides the voltage to the motor block as can be observed in 3 [10].

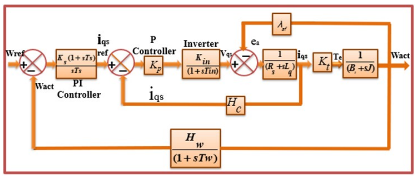

Figure 3: Block Diagram of the speed-controlled closed-loop model of BLDC drive

The development of the closed-loop speed controller is important from the point of view of providing the speed-controlled BLDC motor drive system with desirable transient and steady-state characteristics. In the presence of a d axis stator current, both the d and q axis currents are cross-coupled, and the system becomes non-linear [11].

For simplicity and linearization, the d-axis current is not considered and so the q-axis equation of the voltage of the machine becomes,

$$v_q s = r_s i_{qs} + L_q \frac{di_{qs}}{dt} + \omega_r \lambda_{af} \tag{4}$$

The electromechanical equation is,

$$\frac{P}{2}(T_e - T_l) = J\frac{d\omega_r}{dt} + B_l \omega_r \tag{5}$$

Electromagnetic torque,

$$T_e = J\frac{3P}{4}\lambda_{af} i_{qs} \tag{6}$$

Let the load be frictional in nature,

$$T_l = B_l \omega_m \tag{7}$$

Substituting equation 6 & 7 in equation 5,

$$(J\frac{d}{dt} + B_t)\omega_r = [\frac{3P^2}{8}\lambda_{af}]i_{qs} = K_t i_{qs} \tag{8}$$

where,

and

K$_t = \frac{3P^2}{8}\lambda_{af}$

The inverter modeled in 3 has the transfer function as,

$$G_r(s) = \frac{K_{in}}{1 + sT_{in}} \tag{9}$$

where,

$$K_{in} = 0.65\frac{V_{dc}}{V_{cm}}$$

$V_{dc} = DC\ link\ voltage\ and\ V_{cm} = Maximum\ control\ voltage$

$T_{in} = \frac{1}{2fc},\ f_c = Switching\ frequency\ of\ the\ inverter$

The q-axis current loop is crossed by the induced emf loop and it could be simplified by using a block diagram reduction method. By the block diagram reduction method, the current loop is given as

$$\frac{i_{qs}(s)}{i_{qsref}(s)} = \frac{K_{in}K_a(1 + sT_m)}{H_c K_a K_{in}(1_s T_m) + (1 + sT_{in})[K_a K_b + (1 + sT_a)(1 + sT_m)]} \tag{10}$$

$Where,\ K_a = \frac{1}{R_s},\ T_a = \frac{L_q}{R_s},\ K_m = \frac{1}{B_t},\ T_m = \frac{J}{B_t},\ K_b = K_t K_m \lambda_{af}$

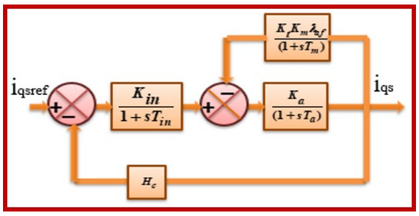
iqsref
+
−
$\frac{K_{in}}{1+sT_{in}}$
+
−
$\frac{K_a}{(1+sT_a)}$
$\frac{K_t K_m \lambda_{af}}{(1+sT_m)}$
iqs
H_c

Figure 4: Current Controller

The following assumptions are made which are considerable near the cross-over frequency:

$$(1 + sT_r) \cong 1 \tag{11}$$

$$(1 + sT_m) \cong sT_m \tag{12}$$

$$(1+sT_a)(1_s T_{in}) \cong 1+s(T_a+T_{in}) \cong 1+sT_{ar} \tag{13}$$

$$T_{ar} = T_a + T_{in} \tag{14}$$

With the above block diagram of 4 along with the assumptions made in equations 11, 12 & 13 we get,

$$\frac{i_{qs}(s)}{i_{qsref}(s)} \cong \frac{(K_{in}K_a T_m)s}{K_a K_b + (T_m + K_{in}K_a T_m H_c)s + (T_m T_{ar})s^2}$$

$$\cong (\frac{T_m K_{i_n}}{K_b})\frac{s}{(1 + sT_1)(1 + sT_2)} \tag{15}$$

It is observed that $T_1 < T_2 < T_m$ which gives scope for further approximation,

$$(1 + sT_2) \cong sT_2 \tag{16}$$

After incorporating all the assumptions, the current loop transfer function is given by,

$$\frac{i_{qs}(s)}{i_{qsref}(s)} \cong \frac{K_i}{(1 + sT_i)} \tag{17}$$

$Where,\ K_i = \frac{T_m K_{in}}{T_2 K_b},\ T_i = T_1$

The simplified current loop is substituted in the design of the speed controller and the block diagram becomes,

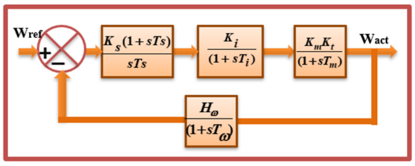

Figure 5: Simplify speed Control Loop

The simplified loop for speed along with a simplified current loop is depicted in 5. The following assumptions are valid near the cross-over frequency,

$$(1 + sT_m) \cong sT_m \tag{18}$$

$$(1 + sT_i)(1 + sT_w) \cong (1 + sT_{wi}) \tag{19}$$

$$(1 + sT_w) \cong 1 \tag{20}$$

Where

$$T_{wi} = T_w + T_i$$

The speed loop transfer function considering the above assumptions in equations 17, 18, 19 as shown below,

$$GH(s) = \frac{K_i K_m K_t H_w}{T_m} \frac{K_s}{T_s} \frac{(1 + sT_s)}{s^2(1 + sT_{wi})} \tag{21}$$

From these above equations, the closed-loop transfer function for speed can be obtained as,

$$\frac{\omega_{ract}(s)}{\omega_{rref}(s)} \cong \frac{1}{H_w}\left[\frac{K_g \frac{K_s}{T_s}(1 + sT_s)}{s^3 T_{wi} + s^2 + K_g \frac{K_s}{T_s}(1 + sT_s)}\right] \tag{22}$$

Where

$$K_g = \frac{K_i K_m K_t H_w}{T_m} \tag{23}$$

Considering the damping ratio is 0.707, the closed-loop transfer function can be written as given below,

$$\frac{\omega_{ract}(s)}{\omega_{rref}(s)} \cong \frac{1}{H_w} \frac{(1 + sT_s)}{1 + s(T_s) + (\frac{3}{8}T_s^2)s^2 + (\frac{1}{16}T_s^3)s^3} \tag{24}$$

Now, equating the co-efficient of equation 22 & 24 and solving further the time constants and gain constants of the speed controller,

$$T_s = 6T_{wi} \tag{25}$$

$$K_s = \frac{4}{9K_g T_{wi}} \tag{26}$$

The proportional gain and the integral gain for speed controller i.e. K_{ps} and K_{is}

respectively can be calculated as,

$$K_{ps} = K_s = \frac{4}{9K_g T_{wi}} \tag{27}$$

$$K_{is} = \frac{K_s}{T_s} = \frac{1}{27K_g T_{wi^2}} \tag{28}$$

$$K_{ds} = K_s T_s = \frac{24}{9K_g} \tag{29}$$

5 Simulation and result

Table 2 consists of the data of step response of different types of controller which are used for designing the closed-loop control systems for the BLDM. When the controllers PI and P are used as speed controller and current controller respectively, there is a high overshoot, rise time and much settling time. To minimise these response values, PI controller is used as both speed controller and current controller, and so, better response are obtained. But even now, overshoot, rise time and settling time are not considerably reduced. So, to get the desired response of the system, PID as speed and PI as current controller implemented. This may overcome the overshoot and rise time, but the settling time is very large. Therefore the system becomes more accurate than before, as the overshoot is Nil and peak amplitude and rise time is much better than the other controllers used before: PI & P. Since it is not the desired output, PID and PI are used as speed controller and current controller respectively. This may overcome the overshoot and rise time but the settling time is very large. Therefore the system becomes more accurate than before, as the overshoot is Nil and peak amplitude and rise time is much better than the other controllers used before.

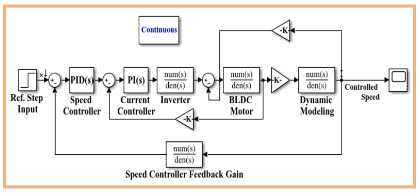

Figure 6: Simulink Block Diagram of a Closed Loop BLDC Motor Drive with PID speed and PI current controller

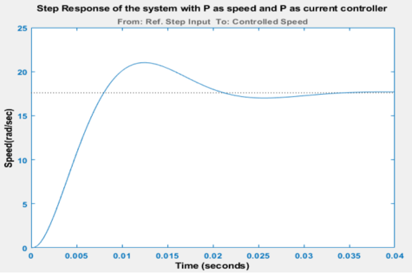

Figure 7: Speed response of the closed-loop system with P speed and P current controller.

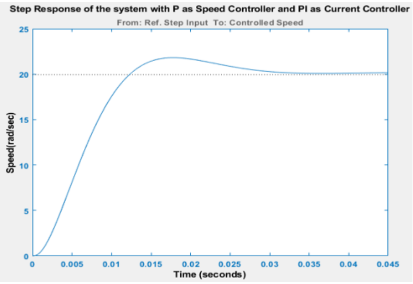

Figure 8: Fig.8. Speed response of the closed-loop system with P speed and PI current controller

Fig 6 shows the simulation model of the BLDC motor drive with a different combination of classical controllers. The speed responses of these different controller combinations are shown in 7 8 9 10 11 12 13 14 15 . 16 is the flow chart of the GA applied to get optimized proportional, integral, a derivative gain value which is later used in the simulation, and the new optimized responses are shown in 17 and 18. From 19 and 20 we can see how the fitness and current best values are changing and, finally, better response is achieved with the help of the GA. The table shows the different characteristics of various controllers, which shows a maximum 19.5% overshoot is achieved with the P & P combination as shown in 7 while there is a significant and unacceptable steady-state error with the P & PI combination as shown in figure 8. On the other hand, the PID as speed & current controller gives us an acceptable 1.23% max. overshoot with a very low percentage of error. At the same time, the GA based PID controller as the speed controller and PI as current controller yield the lowest percentage overshoot and is the best combination of controllers in terms of characteristic properties and zero steady-state error. So, we can say that the best combination of those presented here is the GA based PID as speed & PI as the current controller.

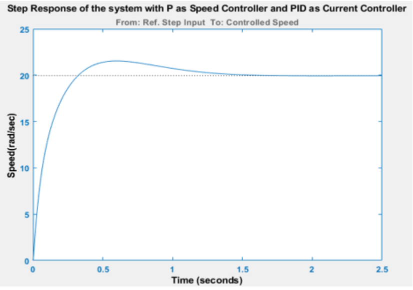

Figure 9: Speed response of the closed-loop system with P speed and PID current controller

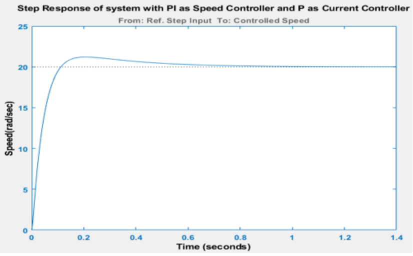

Figure 10: Speed response of the closed-loop system with PI speed and P current controller

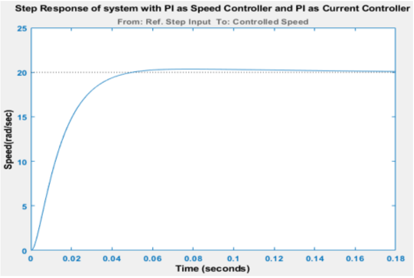

Figure 11: Speed response of the closed-loop system with PI speed and PI current controller

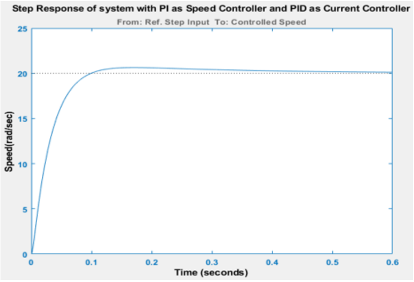

Figure 12: Speed response of the closed-loop system with PI speed and PID current controller

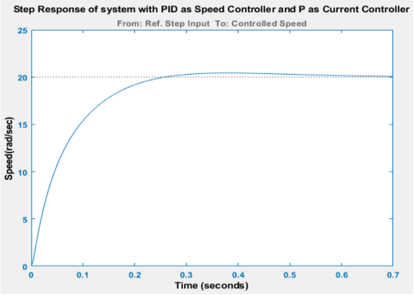

Figure 13: Speed response of the closed-loop system with PID speed and P current controller

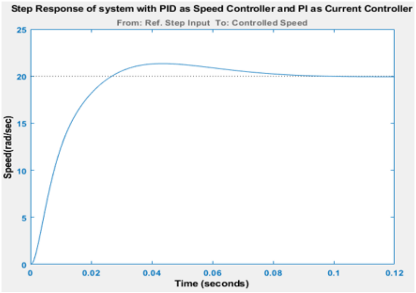

Figure 14: Speed response of the closed-loop system with PID speed and PI current controller

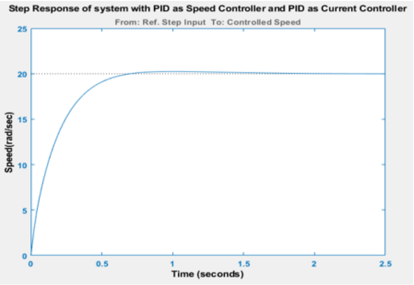

Figure 15: Speed response of the closed-loop system with PID speed and PID current controller

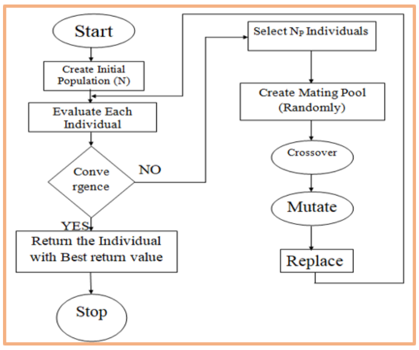
Start
Create Initial Population (N)
Evaluate Each Individual
Convergence
NO
YES
Return the Individual with Best return value
Stop
Select N_p Individuals
Create Mating Pool (Randomly)
Crossover
Mutate
Replace

Figure 16: Flowchart of Genetic Algorithm

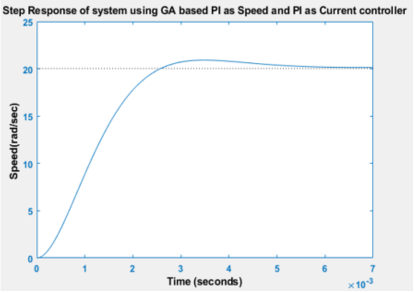

Figure 17: Speed response of the closed-loop system with GA based PI speed and PI current controller

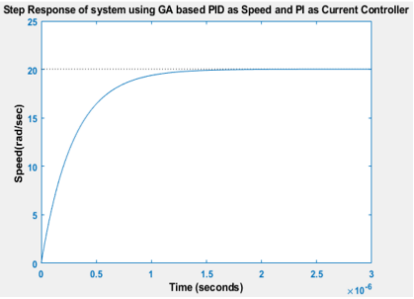

Figure 18: Speed response of the closed-loop system with GA based PID speed and PI current controller

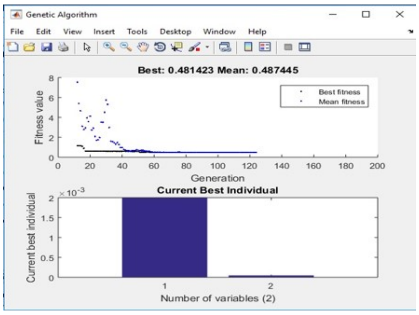

Figure 19: Fitness values and current best values achieved with GA based PI speed and PI current controller

Table 2: Observation Table with all combination of the conventional and genetic algorithm-based controller:

Speed Controller	Current Controller	Rise Time	Peak Time	Overshoot	Settling Time
P	P	.00536	.0125	19.5	.0298
P	PI	.00816	.0177	9.39	.0292
P	PID	.213	.598	7.95	1.2
PI	P	.0721	.205	6.07	.521
PI	PI	.0267	.0786	1.82	.0428
PI	PID	.058	.171	3.21	.307
PID	P	.147	.397	2.14	.425
PID	PI	.0171	.0436	6.71	.074
PID	PID	.337	1	1.23	.583
GA Based PI	PI	.00166	.0035	4.37	.00484
GA Based PID	PI	6.4e-07	3e-06	2.22e-06	1.14e-06

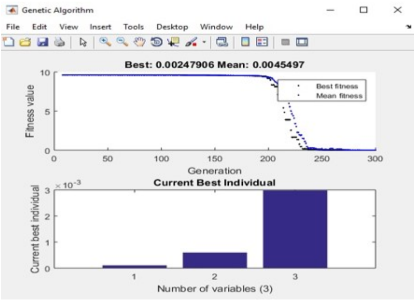

Figure 20: Fitness values and current best values achieved with GA based PID speed and PI current controller.

6 Conclusion

In this paper, a simulation study is analyzed and the genetic algorithm is found to be very effective in suppressing frequency oscillations and to improve system performance by effectively reducing the overshoot. The results show that by using time delay the dynamic response of the system will increase and the GA can be used to compensate effectively for the degradation in system performance. Simplicity of control is a major advantage of the proposed work: the GA can be easily created and modified, with increased robustness. The GA simulation, using MATLAB to control the speed of BLDC Motor, proves that the desired speed is attained with shorter response time compared to conventional controllers. The dynamic characteristics of the motor are obtained and analysis reveals that the GA is capable of controlling the motor drive over a wide speed range. The GA proved itself more efficient than the conventional controller. A prototype model will be developed to analyze characteristics and the hardware results will be compared with the results of conventional controllers.

7 Appendix

The considered parameters of the Brush-less DC motor are: Number of pole, $P = 6$, Stator resistance, R_s=1.4Ω, Direct axis inductance, L_d=0.0056H, Quadrature axis inductance, L_q=0.009H, Friction coefficient, B=0.01 N-m/rad/sec, Moment of inertia, J=0.006 kg-m^2 , Switching frequency of the inverter, f_c=2KHz , Maximum control voltage, V_{cm}=10V , H_c=0.8 V/A , DC link voltage,V_{dc}=285V.

References

1.Kim, T., and Yang, J. (2009) Control of a brushless DC motor/generator in a fuel cell hybrid electric vehicle. *2009 IEEE International Symposium on Industrial Electronics.*

2.Miller, T.J.E. (1989) *Brushless Permanent Magnet and Reluctance Motor Drives*, Clarendon Press, Oxford University Press.

3.Upama Das, P.K.B., and Debnath, S. (2017) Modeling and Simulation of Open Loop Model of Brush Less DC Motor by Using MATLAB Based Software. *International Journal of Electronics, Electrical and Computational System*, **6**.

4.Upama Das, S.D., Pabitra Kumar Biswas (2018) A Comparative Study between Load and No-Load condition of Brushless DC Motor Drives by Using MATLAB. *JOURNAL OF POWER TECHNOLOGIES*, **3**.

5.N. Hemati, 1. S.T., and Leu, M.C. (1990) Robust nonlinear control of Brushless dc motors for direct-drive robotic applications. *IEEE Trans. Ind. Electron.*, **37**.

6.Pelczewski, P.M., and Kunz, U.H. (1990) The optimal control of a constrained drive system with brushless DC motor. *IEEE Transactions on Industrial Electronics*, **37** (5), 342–348.

7.Ang, K.H., Chong, G., and Li, Y. (2005) PID control system analysis design, and technology. *IEEE Transactions on Control Systems Technology*, **13** (4), 559–576.

8.Tom O'Mahony, C.J.D.K.F. (2000) Genetic Algorithms for PID Parameter Optimisation: Minimising Error Criteria. *Conference: Process Control and InstrumentationAt: University of StrathclydeVolume: pp.148 153.*

9.Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc.75 Arlington Street, Suite 300 Boston, MAUnited States.

10.Pillay, P., and Krishnan, R. (1988) Modeling of permanent magnet motor drives. *IEEE Transactions on Industrial Electronics*, **35** (4), 537–541.

11.Krishnan, R. (2001) *Electric Motor Drives Modeling, Analysis, and Control*, Prentice-Hall International Inc New Jersey.