Performance analysis of controllers in BLDC motor

S.Muralidharan[1], *Balasubramaniyam.T*[2], *Prethesh* Kumar.K.S[2], and *Aswin.C*[2]
[1]Professor, EEE Department, Mepco Schlenk Engineering College, India
[2]UG students, EEE Department, Mepco Schlenk Engineering College, India

Abstract. Due to their better electrical properties and mechanical features, the usage of BLDC motors is increasing and they have replaced conventional DC and asynchronous motors in many applications. This project provides a sensor-based permanent magnet brushless DC motor for commercial and electric vehicles. To improve the reliability of BLDC motor speed regulation, a PID controller is employed. The three-phase voltage inverter delivers a constant DC voltage and is synced with the BLDC motor. Closed-loop speed control is done utilizing the Ziegler-Nichols technique, Particle Swarm optimization and PID controller improved using Firefly algorithm. The recommended system is simulated using MATLAB simulation and the performance of the BLDC motor is tested by comparing attributes such as Rise time, Settling time and Peak overshoot.

1 Introduction

For BLDC, the rotor's DC field winding is replaced by a permanent magnet to produce variable airflow. Thanks to the magnet of the rotor, the electrical loss in the winding of the motor is reduced and the thermal performance and performance of the PM motor are improved. Without mechanical components such as brushes or slip rings, the motor is lighter and has a higher power-to-weight ratio, increasing efficiency and reliability. Permanent magnets have disadvantages such as the manufacturing process, high temperature demagnetization of the magnet, and expensive magnet material. PMDC motors and PMAC motors are two types of PM motors. Permanent magnets are used to switch field windings in PMDC motors, similar to a DC commutator. PMACs do not use slip rings, brushes and commutators as Permanent magnets installed on the rotor create the magnetic field. Therefore, PMAC is easier to use than BLDC. According to the type of back electromotive force (EMF), PMACs are divided into two types: sinusoidal and trapezoidal. Surface-mounted PMSMs and internal PMSMs are additional categories PMAC machines of the sinusoidal kind. The brushless DC motors (BLDC), often known as trapezoidal BLDC machines, It generates trapezoidal back EMF waves with the following properties:

- Magnet flux distribution in the air gap is rectangular.
- Concentrated stator windings and a rectangular current waveform

*Corresponding author:smurali@mepcoeng.ac.in

1.1 Literature Survey

Speed control of a BLDC motor is important to keep the motor running at the desired speed. The speed of the BLDC motor can be controlled by controlling the input DC voltage and current. The higher the voltage, the faster it will be. The speed control can be achieved by open loop and closed loop control. Open loop control are not much preferred due to their poor efficiency, whereas closed loop control provide smooth operation and higher efficiency than open loop control. Closed loop control provides a feedback path, and a controller is placed in that path. The paper [1] proposed the speed control of BLDC motors by PWM technique using FPGA; studies on microcontrollers and DSP controllers are done to control the speed of the motor [2, 3]. There are various studies and researches done to control the speed of BLDC motors using PI and PID controllers [4, 5],The majority of PI controllers used in industry today are traditional ones. Because of the simplicity and convenience of online tweaking, traditional PID controllers are also employed in industries. To provide optimized output, tuning must be done by various algorithms. The paper [6, 7] demonstrates the tuning of a controller by the Ziegler-Nichols method, Ziegler-Nichols (Z-N) method is a common tuning method for PID controllers [8, 9]. Using this strategy, the integral and derivative gains are set to zero before raising the proportional gain until the system becomes unstable, due to the high time consumption nature Z-N methods are not used now a days; instead, fuzzy logic controllers are being designed and tuned for PI,PID speed control applications [10, 11]. Firefly algorithms are commonly used algorithms nowadays. The various studies and researches [12, 13] describe the effectiveness of the firefly algorithm. To enhance the effectiveness and efficiency of the firefly algorithm, the continuous firefly algorithm and the Continuous firefly algorithm are used [14, 15]. The above extract explains the performance of BLDC motors with various algorithms.

1.2 Brushless DC motor

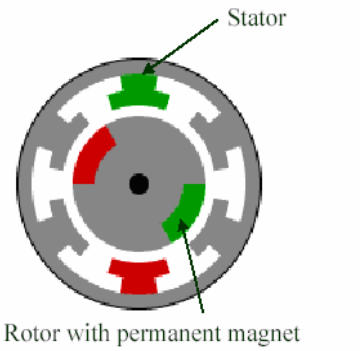

Fig 1. BLDC motor's cross-sectional view

A BLDC motor driver uses a generator to convert alternating current to power the permanent magnet synchronous motor. It does not seem possible to develop controllers for these machines using industrial simulation tools, possibly due to the high cost of software development. In low-cost applications for example, Numerical Control machine tools and robot drivers, the rotor magnet can be demagnetized during commissioning or modification. However, special drive topologies for high-end applications may require reconfiguration of motors and inverters, resulting in higher drive costs. The cost of using new magnetic materials is superior . The invention has the potential to increase the market for BLDC

motors to tens of kilowatt applications where debugging mistakes are costly. Modelling is essential for achieving cost reductions.

1.3 BLDC drives operation with inverter

As it is an essential part of the generator, the front end should have a three-phase inverter as shown in Fig 2. The inverter acts as an electric commutator in autonomous mode and

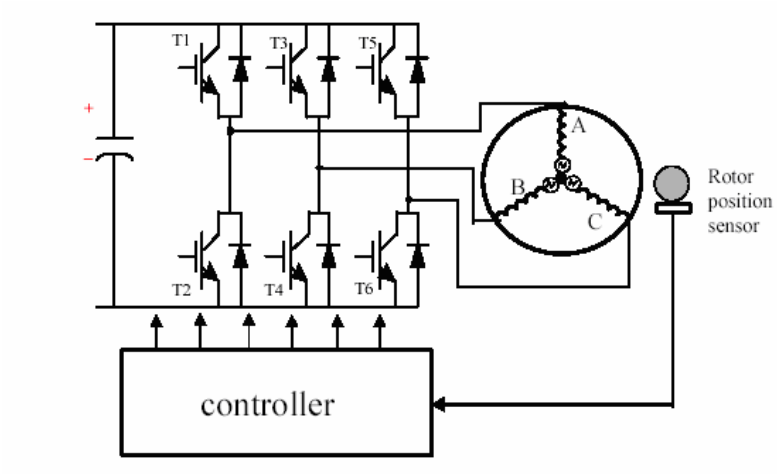

Fig 2. BLDC Motor drive system with Hall sensor

receives change logic pulses from the correct position sensors.

The inverter may essentially function in two modes.

- 120-degree switch-on mode
- 180-degree switch-on mode
- PWM voltage and current control mode

Table 1. Motor ratings

Parameters	Symbol	Ratings
Torque constant	K_t	0.06675
Back EMF	K_b	48
Moment of Inertia	J	$0.142e^{-3}$
Friction coefficient	B	0.04
Electrical Inductance	L	$0.389e^{-3}$
Electrical Resistance	R	0.323
Speed	RPM	3000

1.4 Transfer function of motor

A common sort of mathematical model is the transfer function. First-order and second-order transfer functions with voltage input and motor speed output are often used in BLDC motor models. Given that there is uncertainty and nonlinear processes in the real-time environment, experiments have been both simulated and carried out in order to compare their results.

The BLDC motor's Transfer function is given by:

$$G(S) = \frac{K_t}{(Js+b)(Ls+R)+K_tK_b} \quad (1)$$

Where,

- *s: Laplace variable*
- *K_t: Torque constant*
- *K_b: Back EMF*
- *J: Moment of Inertia*
- *b: Friction Coefficient*
- *L: Electrical Inductance*
- *R: Electrical Resistance*

From this the BLDC motor's Transfer function is given by:

$$G(S) = \frac{0.06675}{0.1369e^{-9}s^2+0.079s+4.496} \quad (2)$$

1.5 Without controller

A controller is necessary to manage the speed and torque of a BLDC (Brushless DC) motor. This is as a result of the motor's intricate electronic commutation system, which regulates the current flow to the motor coils in accordance with the rotor position. The motor cannot operate effectively or smoothly without a controller.

Otherwise, a PWM circuit may be used to regulate a BLDC motor's speed even without a controller. A PWM circuit is an electrical device that alternately turns on and off the voltage at a high frequency, usually between tens and hundreds of kilohertz. You may modify the PWM signal's duty cycle to alter the average voltage delivered to the motor. However, it is very difficult to reach the desired speed.

1.6 PID controller

This PID speed control system is used because of its advantages such as 0% steady state error, no oscillation, better reaction, and greater stability. This controller is used to reduce oscillation and overshoot in system output. The PID controller's output is the total of the proportional, integral, and derivative gains.
Mathematically:

$$\text{Output} = K_p e(t) + K_i \int e(t)dt + K_d \frac{d}{dt} e(t) \quad (3)$$

- *K_p: Proportional gain constant*
- *K_i: Integral gain constant*
- *K_d: Derivative gain constant*

The PID (Proportional-Integral-Derivative) controller calculates the system output using three constants: K_p, K_i, and K_d .The disparity between the predicted and actual values used to determine the output is referred to as signal error.

2 Algorithms

Algorithms automate and enhance the tuning process, they are essential for tuning PID controllers for BLDC motors. Both model-based and model-free techniques are available. Model-based algorithms employ the system's mathematical models to identify the best PID

gains, whereas model-free algorithms change the gains depending on the system's reaction through trial and error.

2.1 Ziegler - Nichols method

The Ziegler-Nichols technique is a popular PID controller tuning method. The integral and derivative gains are set to zero in this technique before increasing the proportional gain until the system becomes unstable.

To implement this method in MATLAB by the following steps:
Step 1: Define your system and obtain its transfer function. For instance, if you have a system that has a transfer function,

$$G(S) = \frac{1}{s^2+2s+1} \qquad (4)$$

Step 2: To determine the Ultimate gain and period, use the system's step response. To do this, you can use the 'stepinfo' function in MATLAB, It returns a structure containing different information on a system's step response.
Step 3: Calculate the PID parameters using the Ziegler-Nichols tuning method. There are two variants of the Ziegler-Nichols method: the classic method and the modified method. Below are the formulas for both methods:
Traditional Method:

$$K_p = 0.6 * K_u, K_i = 1.2 * {K_p}/{T_u} \text{ and } K_d = 0.075 * K_p * T_u$$

Modified Method:

$$K_p = 0.2 * K_u, K_i = 0.4 * {K_p}/{T_u} \text{ and } K_d = 0.05 * K_p * T_u$$

Here,

- *Ku: Constant of Ultimate Gain*
- *Tu: Constant of Ultimate Time*

Step 4: Display the results

2.2 Particle swarm optimization

Particle swarm optimization, or PSO, is a well-liked optimization method that is widely applied in many disciplines, including physics, computer science, engineering, and economics. The PSO algorithm takes its cues from social behaviours like fish scooling or bird flocking, where each individual moves in accordance with its neighbours to discover the best solution to a particular issue.
Procedure for PSO:
Step 1: The purpose of PSO is to define the problem that must be addressed. This includes identifying the target function to be optimized as well as any search space constraints.
Step 2: In PSO, start the swarm, and a swarm of particles will explore the search space. Each particle considered will be responsible for the solution. By dispersing particles at random over the search space, the swarm is created.
Step 3: Assess fitness: Using the objective function, the fitness of each particle is assessed. In terms of locating the ideal answer, this establishes how effective or ineffective the particle is.

Step 4: Personal best is an updated record of each particle's best location (or solution) to date. The personal best is updated if the fitness of the current position is higher than the fitness of the previous position.
Step 5: Update global best: Each particle also maintains track of the best location (or solution) discovered by the whole swarm, known as the global best, in addition to their own personal best. If the fitness of the current position is greater than the fitness of the global best, the global best is updated.
Step 6: Each particle's velocity is updated in accordance with its current location, personal best, and world best. The velocity of the particle controls its migration over the search space, as well as its direction and speed.
Step 7: location update: Taking into account the new velocity, each particle's location is revised. This establishes the particle's new position within the search space.
Step 8: Repeat: The steps from 3 to 7 are repeated a predetermined number of times. (or until convergence is reached). Until the ideal solution is identified, the swarm advances around the search area while updating individual and collective best locations and velocities.
Step 9: Return of the solution: After the PSO algorithm converges, the best location obtained globally by the swarm is the ideal solution.

Table 2. PSO initial parameters

Number of Particles	20
Maximum Generation	50
Inertia Weight	0.7
Cognitive Weight	1.5
Social Weight	1.5
Number of Iterations	1000

2.3 Firefly algorithm

The flashing activity of fireflies served as the inspiration for the development of the Firefly algorithm, a unique metaheuristic technique for optimization problems Randomly produced solutions are represented as fireflies in this manner, and their brightness is governed by how well they perform on the objective function. Yang devised this swarm-based metaheuristic algorithm that mimics how fireflies communicate by flashing their lights.
Procedure for Firefly algorithm:
Step 1: Set the algorithm parameters, such as the number of fireflies (n), the maximum number of generations (MaxGen), and the attraction coefficient ().
Step 2: Define the goal function f(x), where x represents a d-dimensional vector (**x1,..., xd**)T.
Step 3: Create the initial firefly population, X = [x1, x2,..., xn].
Step 4: Using f(xi), calculate the light intensity of each firefly i at xi.
Step 5: While the halting requirement (e.g., t MaxGen) is not met:
For each firefly i ranging from 1 to n:
For each individual firefly j = 1 to n:
If the light intensity of firefly j is greater than that of firefly i, move firefly i closer to firefly j.
j using the following formula:
xi = xi + * exp(-r2) * (xj - xi) + * (rand - 0.5)

Attractiveness varies with r through Exp[-r2].
Examine new ideas and adjust light intensity
end of j
end of i

Step 6: Sort the firefly by light intensity to discover the best answer right now.
End while

Step 7: Visualisation of post-process outcomes

Step 8: Conclude the operation.

Table 3. Firefly algorithm initial parameters

Number of Fireflies	20
Maximum Generation	50
Alpha	0.5
Gamma	0.2
Number of Iterations	1000

+2.4 Objective function

The objective function is a real-valued function used to either minimize or maximize its value while adhering to the specified constraints. It is a crucial component in solving optimization problems.

ISE stands for Integral Square Error, which is a common objective function used in control systems engineering. The ISE objective function is used to measure control system performance by evaluating the error between the system's actual output and the intended setpoint over time.

The mathematical equation for ISE can be expressed as:

$$ISE = \int_0^T e(t)^2 dt \qquad (6)$$

Where,

- *e(t) : error between the setpoint and the actual output at time t*
- *T : final time of the interval.*

3 Simulation and results

The Mathematical mode BLDC motor with PID was implemented in MATLAB. In MATLAB the simulation was done by Simulink

model.

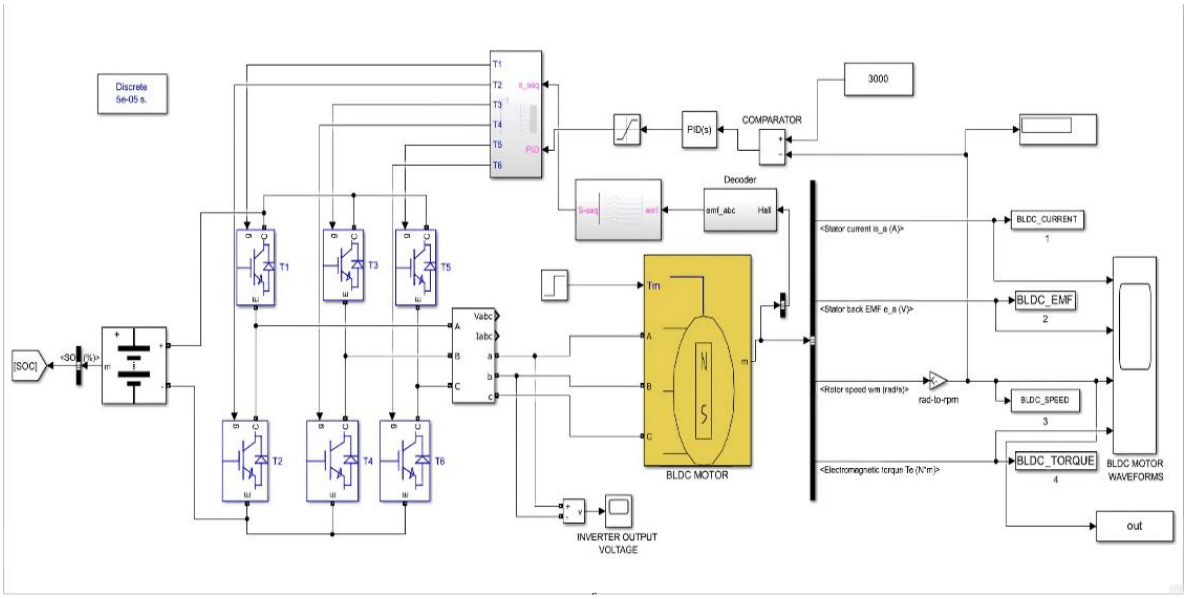

Fig 3. Simulation model in MATLAB

The following PID parameters are generated by Ziegler-Nichols method, Particle Swarm Optimization and Firefly Algorithm.

Table 4. PID parameters

Controller	PID with ZN	PID with PSO	PID with FA
K_p	0.008898	0.9048	1.3039
K_i	0.097949	1.4775	2.9315
K_d	7.2749e-5	0.0750	0.0790

The generated PID parameters (K_p, K_i and K_d) are substituted into the PID controller, which is available in the Simulink model. Then, the Simulink model is simulated for 10 seconds to obtain the speed waveform using the scope.

By using the To Workspace block in Simulink, the waveform characteristics are saved to the workspace of MATLAB. Using the 'stepinfo' command, the parameters of that speed waveform can be determined.

The control parameters are compared and tabulated below:

Table 5. Waveform parameters

Controller	Parameters	
Without Controller	Rise Time	0.0554
	Settling Time	9.9942
	Peak Overshoot	6.5394
PID with ZN	Rise Time	0.3205
	Settling Time	0.5760
	Peak Overshoot	0.7631
PID with PSO	Rise Time	0.1375
	Settling Time	0.7234
	Peak Overshoot	0.3388

PID with FA	Rise Time	0.0975
	Settling Time	0.6836
	Peak Overshoot	0.0975

The waveforms displayed below are obtained from simulation with PID controller tuned with Ziegler-Nichols method, Particle Swarm Optimization and Firefly Algorithm. And the speed waveform also obtained without PID controller.

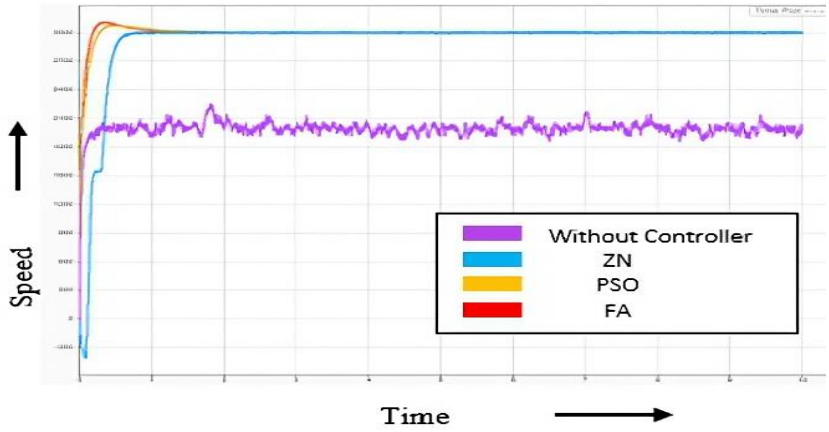

Fig 4. Speed waveform of BLDC motor with PID controller tuned with FA, PSO, ZN and without controller

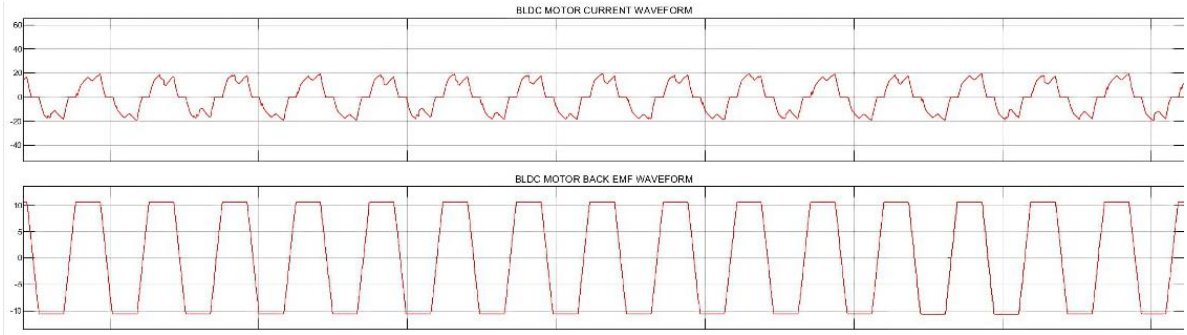

Fig 5. Stator current and Back EMF waveform

This simulation was also conducted at a different set speed. In Fig. 6, the set speed is changed from 2000 rpm to 3000 rpm. And for this waveform, the control parameters are calculated and tabulated in Table 6.

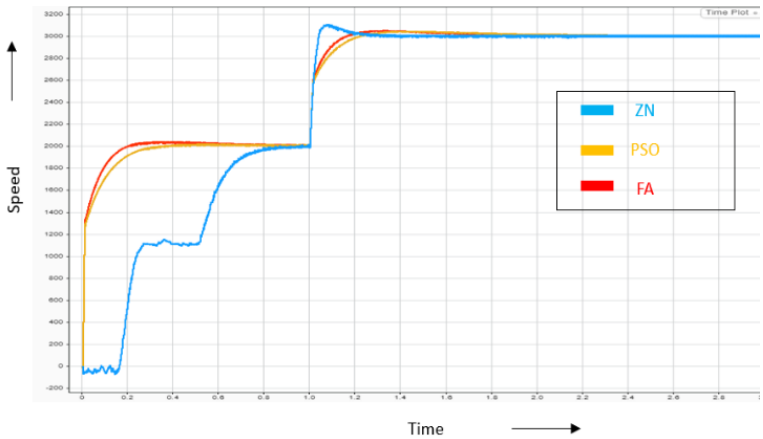

Fig 6. Speed waveform with variable set speed

For this speed waveform with a variable set speed, the parameters like rise time, settling time, and peak overshoot are tabulated below.

Table 6. Different set speed waveform parameters

Controller	Parameters	
PID with ZN	Rise Time	0.8382
	Settling Time	0.1661
	Peak Overshoot	1.3265
PID with PSO	Rise Time	0.0381
	Settling Time	0.1540
	Peak Overshoot	0.0238
PID with FA	Rise Time	0.0288
	Settling Time	0.1147
	Peak Overshoot	0.0258

The speed waveform was also monitored for different loaded conditions. The obtained speed waveform as shown in Fig 7.

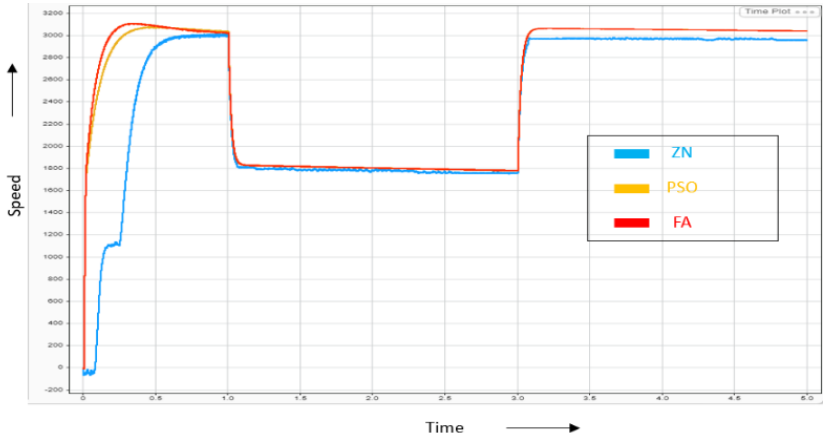

Fig 7. Speed waveform different loaded condition

5 Conclusion

This study aimed to develop and construct a PID controller in MATLAB utilizing the Ziegler-Nichols (ZN), Particle Swarm Optimization (PSO), and Firefly Algorithm (FA) tuning techniques for BLDC motor speed control. The primary goal of this initiative was to assess the performance of PID controllers adjusted using various optimization strategies and to choose the most effective one for BLDC motor speed control.

According to the simulation results, all three tuning techniques were successful in rising the PID controller's performance over that of the original guess tuning. But in terms of rising time, settling time, and overshoot, the performance of the controller tuned by the FA approach was discovered to be superior to that of the other two methods. In comparison to the ZN and PSO-tuned controllers, the FA-tuned PID controller had a quicker rising time, a lower settling time, and less overshoot.

Overall, the study was successful in demonstrating the PID controller's usefulness for controlling the speed of BLDC motors when used with various tuning techniques. Electric cars, drones, and robots are just a few examples of applications that might benefit from using the suggested ways to precisely regulate the speed of BLDC motors. The FA-tuned PID controller may be implemented in a physical system, and its performance can be assessed in real-world circumstances as part of future development.

References

1. R. M. Pindoriya , S. Rajendran , P. J. Chauhan *International J of Advance Eng and Research Development (IJAERD)*, 1-6, (2014).
2. Juhi Nishat Ansari , Sapna L *International J of Eng Research & Technology (IJERT)*, **3** ,1666-1671, (2014).
3. G.MadhusudhanaRao, B.V.SankkerRam, B.Smapath Kumar, K.Vijay Kumar, *International J of Eng Science and Technology***,2**,143-147 (2010).
4. Y.Narendra Kumar, P.Eswara Rao, P. Vijay Varma, V. V. Ram Vikas, P. Kasi Naidu, *Int. J of Eng Research and Applications* ,**4**, (2014).
5. C K Karthika and Peter Abraham , *AIP Conf Proceedings,*1-7,(2020).
6. Vishakha Vijay Patel, *Resonance***,25**, 1385-1397, (2020).
7. Mohd Sazli Saad, Hishamuddin Jamaluddin and Intan Zaurah Mat Darus*, ICIC International*, **8**,7761-7779, (2012).
8. Mohammed Abdelbar Shamseldin, Adel A. EL-Samahy *15th International Workshop on Research and Education in Mechatronics (REM)*,1-6, 9-11 September 2014 , Elgouna, Egypt (2014).
9. Zemmour N, Ishak Boushaki S, Bendjeghaba O,Firefly *4th International Conf on Power Eng, Energy and Electrical Drives*,1293-1296,13-17 , (2013).
10. Neethu U. and Jisha V. R. *2012 IEEE International Conference on Power Electronics, Drives and Energy Systems* ,1-5,16-19 , (2012).
11. Christina Mathew, Jebin Francis, Prathibha PK*, IJSRD - International Journal for Scientific Research & Development*, **5**,636-639 (2017).
12. D. Kumanan & B. Nagaraj, *An Open Access Journal*,52-56 (2013).
13. Pikaso Pal , Rajeeb Dey , Raj Kumar Biswas , Shubhashish Bhakta, *International Journal of Soft Computing, Mathematics and Control (IJSCMC)*, **4**,39-48, (2015).
14. Arman Hadi Azaha , Adam Samsudin , Yep Kow Wai , Amar Faiz Zainal Abidin , Rozi Rifin , Mohammad Haniff Harun , Mohd. Safirin Karis , Ezzatul Farhain Azmi , and Ili Najaa Aimi Mohd. Nordin, *ARPN J of Eng and Applied Sciences*,**16**, 757-761, (2021).
15. Omar Bendjeghaba, *J of Electrical Eng*, **65**, 44-49 (2014).