Optimal FOPID Controller For Speed Control Of Brushless DC Motor Using Hybrid Metaheuristic Techniques

[1]Dr. Agnihotri Santosh P., [2]Dr. Kulkarni A. R. and [3]Dr. Joshi Mandar P

[1]Associate professor

R H Sapat College of Engineering Management

Studies and Research

Nasik India.

santosh5199jan@rediffmail.com

[2]Associate professor at NDMVP College of

Engineering, India.

[3]Assistant professor

R H Sapat College of Engineering Management Studies and Research

Nasik, India.

Abstract: A Brushless DC (BLDC) motor is an electronically commutated motor that operates using a DC power source. However, controlling the speed of BLDC motor can be challenging, as it requires precise timing and coordination between the motor's windings and the driving circuit. Without a controller, the speed of a BLDC motor may fluctuate, leading to performance issues, including decreased efficiency, increased wear and tear, and potential damage to the motor. In industrial applications where BLDC motors are used, precise speed control is essential for ensuring the motor operates at peak efficiency and avoids potential damage. Therefore, a controller for speed control of BLDC motor is critical to ensuring the motor's reliable and

efficient operation. The speed control system should ensure that the motor runs smoothly at the desired speed and torque, and it should be able to handle changes in the load and operating conditions. An optimal design can lead to better performance, increased energy efficiency, reduced maintenance, and increased lifespan of the motor. In this paper, we introduced an optimal FOPID controller for speed control of BLDC motor using hybrid metaheuristic techniques.Initially, a modified spider monkey optimization (MSMO) algorithm is used to locate the error function in FOPID controller. Secondly, a hybrid deep recurrent neural network (DRNN) is used to track the error functions and provide the optimal gain values, leading to a reduction in harmonics and torque ripples. In addition, the self-tuning of the FOPID controller is performed using the improved black widow optimization (IBWO) algorithm, which minimizes the given objective function to meet the constraints on inequality. Finally, the proposed model's performance is validated using various simulation environments, and the results are compared with existing state-of-the-art controllers.

Keywords:BLDC motor, speed control, FOPID controller, hybrid metaheuristic, self-tuning.

1. Introduction

Brushless DC (BLDC) motor is a type of motor that utilizes electronic commutation instead of brushes, which are present in traditional DC motors [1]. Permanent magnets are attached to the BLDC motor's rotor, and an electronic controller is used to alternate the current in the stator windings, generating rotation. Numerous applications, including electric vehicles and industrial automation, favor these motors and consumer electronics, owing to their low maintenance, high reliability, and high efficiency [2]. Maintaining a constant rotational speed of the motor shaft is

necessary for controlling a BLDC motor's speed. By controlling the amount of current that is supplied to the motor's windings, this can be accomplished. The motor's torque and electromagnetic fields are affected by the amount of current supplied, making it a crucial factor in maintaining the desired speed [3]. Due to their reliability, efficiency, and low maintenance, BLDC motors have become a popular choice for various applications, including industrial machinery, electric vehicles, and drones, where precise speed control is essential for optimal performance. BLDC Due to their high efficiency, dependability, and low cost, motors are utilized in numerous applications maintenance requirements. To ensure optimal performance and efficiency, it is essential to have precise control over the motor speed. Speed control allows for better stability, efficiency, and accuracy in many applications, preventing damage to the motor or the equipment it drives [4]. Additionally, speed control can reduce maintenance costs and extend the lifespan of the motor, making it a critical aspect of BLDC motor control.There are several methods commonly used for speed control in BLDC motors, including traditional techniques [5] such as voltage control, current control, and pulse width modulation (PWM) [6] technique, as well as advanced techniques such as direct torque control (DTC) [7]. Therefore, there is a need for more efficient and effective speed control techniques for BLDC motors. A popular method for BLDC motors is field oriented control speed control that uses vector control theory to separate stator current into flux-producing and torque-producing components. Thus, alternative control techniques are being explored to overcome these limitations and enhance the BLDC motor speed control's effectiveness and performance.

The traditional techniques [11]-[13] for speed control in BLDC motors, including PID controllers, have several limitations. The proportional, integral, and derivative terms used in these techniques may not always provide accurate control under varying operating conditions.

Manual tuning can be time-consuming, and these methods may not be able to handle nonlinearities and uncertainties in the system, resulting in degraded performance and stability issues. Additionally, sudden changes in load or disturbances can impact the motor's performance, which these techniques may not be able to handle. Therefore, there is a need for advanced techniques that can provide better control and stability under varying operating conditions.

In addition to FLC [14][15] and SMC [16], other advanced control techniques have also been proposed for speed control in BLDC motors. A control method known as model predictive control (MPC) [17][18] makes use of a dynamic model of the system to optimize the control action and predict the system's future behavior. MPC has shown good performance in terms of speed regulation and disturbance rejection, but it requires high computational power and may not be suitable for real-time applications. Artificial neural networks (ANN) have also been used for speed control in BLDC motors. ANN can learn the complex nonlinear relationships between the control inputs and the system outputs, and can provide good performance under varying operating conditions. However, ANN may not be suitable for online learning due to its large training data requirement. A control method known as model predictive control (MPC) makes use of a mathematical model to anticipate future behavior and optimize control actions in order to minimize a cost function. It has demonstrated promise in terms of transient response and disturbance rejection and has been utilized for BLDC motor speed control [17]. Adaptive control techniques, on the other hand, adjust control parameters based on real-time system data to handle uncertainties and disturbances [18]. These techniques have the potential to enhance BLDC's effectiveness and performance motors in various applications.Fractional order control is a novel approach for speed control of BLDC motors [19][20] that uses fractional calculus to describe the system dynamics. It has been shown to offer advantages such as improved stability and

robustness over traditional control techniques. FOPID control is a specific type of fractional order control that has been applied to BLDC motors, and it has demonstrated good performance in terms of speed regulation and robustness.

Our contributions. When designing a FOPID controller for BLDC motor speed control, the optimal controller design is complex problem that requires the use of advanced optimization techniques. In order to achieve the best possible performance, it is essential to carefully evaluate the role and contribution of each algorithm involved in the design process.

1. One important contribution to this field is the introduction of a modified spider monkey optimization (MSMO) algorithm. It is specifically used to locate the FOPID controller's error function with high accuracy and efficiency. By incorporating this algorithm into the controller design process, the resulting controller can be optimized to minimize error and improve system performance.

2. Another important algorithm used in this design process is the hybrid deep recurrent neural network (DRNN). It is used to track the error functions and provide optimal gain values for the FOPID controller. By using this algorithm, the controller can be designed to reduce harmonics and torque ripples, leading to smoother operation and improved overall performance.

3. The self-tuning of FOPID controller is performed using the improved black widow optimization (IBWO) algorithm. It is used to minimize the given objective function while ensuring that all constraints on inequality are met. By using this algorithm, the controller can be optimized to provide the best possible performance under a variety of operating conditions.

Overall, the combination of these algorithms in the controller design process allows for the creation of an optimized FOPID controller for BLDC motor speed control. Through

simulation and testing, the performance of this controller can be evaluated and compared to existing state-of-the-art models.

This paper is structured as follows. In Section 2, we review the existing literature on controllers for BLDC motor speed control, highlighting the limitations of previous approaches and the research gaps that need to be addressed. Section 3 presents the problem methodology and the proposed system architecture for optimal BLDC motor speed control. The detailed working process of the proposed approach is explained in Section 4, including the role of each algorithm in the design process. In Section 5, we present the simulation results and a comparative analysis of the proposed approach with existing methods, demonstrating the effectiveness and superiority of the proposed method. Finally, we draw conclusions and discuss future research directions in Section 6.

2. Related works

Several works have been carried out on speed control of BLDC motors using various techniques. In this section, we review some of the related works in the literature. The works are categorized based on the control techniques used, including traditional control techniques, intelligent control techniques, and hybrid control techniques. We also discuss the contributions, limitations, and significance of each work. The summary of research gasps are given in Table 1.

Kommula et al. [21] proposed a novel approach to control torque and speed in BLDC motors controlled by a FOPID controller based on the firefly algorithm. The conventional method of torque control through current references results in torque ripples and is not effective. This method uses a direct instantaneous scheme to compare reference torque with estimated torque, and the FOPID controller receives the error for efficient and ripple-free torque control.

The parameters are changed by using the firefly algorithm method of the FOPID controller.Gobinath et al. [22] proposed a hybrid control approach that combines BLDC motor gain values can be adjusted with a fuzzy system and a perceptron neural network that uses deep learning. Using a multi-swarm particle swarm optimizer, the deep learning framework was made better. The controller was found to be more stable than other controller models when compared to the Lyapunov stability criterion was used to assess the stability of the closed-loop system. The proposed controller delivered superior results than other controllers in tests conducted under a variety of load conditions. Yigit et al. [23] have proposed a closed-loop FOPI controller for a BLDC motor used in modern drive systems for mobile robots, electric vehicles, and unmanned aerial vehicles (UAVs). The Simulink BLDC motor includes a storage tank, a PEM fuel cell (PEMFC), a proton exchange membrane (PEM) electrolyzer, and a motor driver system environment in MATLAB. FOPI controller parameters are optimized using the Moth Swarm Algorithm (MSA) to achieve stable operation and high performance at a variety of speeds and torques. The FOPI has controlled BLDC motor powered by the PEMFC operates as expected in modern drive systems. Vanchinathan et al.'s goal was to enhance the BLDC motor's performance of ABC-based adaptive FOPID controller was proposed by [24]. To overcome the poor controllability brought on by the motor's nonlinearity, power fluctuation, and prolonged settling time, the controller aims to meet inequality constraints and Reduce the objective function to a minimum. In addition, a Kalman filter is utilized to circumvent many of the limitations imposed by Hall effect sensors when estimating motor speed. The proposed ABC-tuned FOPID controller's simulation results are contrasted with those of modified and conventional genetic algorithm-tuned FOPID controllers, respectively.

Dutta et al. [25] have proposed an optimization technique, namely Grey Wolf Optimization (GWO), to regulate a BLDC motor's speed. The proposed method has produced results that shown better performance than the conventional Particle Swarm Optimization (PSO) technique. The GWO-based method exhibited faster settling time and lesser damping, along with improved values of performance criteria like ISE, IAE, ITAE, and ITSE. The overall improvements in time-domain performance suggest that the GWO technique is suitable for tuning the PID controller parameters of BLDC motors, despite the fact that the rise time was slightly longer than that of the PSO method. Kanna et al. [26] have introduced a power-consuming Controller with Truncated Angle Variant (TAV) for managing multi-source-fed BLDC motors and effective control to keep the motor running at the desired speed. This technology enables electric vehicles to operate the motor with minimal response time and provides high power to start the motor without compromising the health of the battery. Furthermore, it eliminates torque ripple effects, thus preventing errors from being added to the procedure.Additionally, the PV system adds to the vehicle's power supply by serving as an additional power source. In addition to achieving high power and ensuring proper commutation, this converter design with TAV enables effective speed management of BLDC motors.

Eltoum et al. [27] proposed a mixture control technique for BLDC engine speed guideline utilizing a FOPID regulator to control the reference current of the BLDC engine and a fluffy rationale regulator to control the DC transport voltage of the inverter. FOPID controller's parameters are refined using a modified harmony search (HS) metaheuristic approach. There are three distinct operating conditions under which the controller's efficiency is evaluated: variable load operation, variable speed operation, and no-load operation the speed control schemes based on fuzzy and FOPID are contrasted with the effectiveness of the hybrid control strategy.

Karuppannan et al. [28] propose a type-2 BLDC motor speed control using a fuzzy system WNL-based controller. For getting inputs, wavelet neural learning is used from a brand-new type-2 fuzzy PID controller that controls the motor mechanism. The PID controller's gain values are adjusted using the WNL-based type 2 fuzzy system to control the operating speed.The controller's gain is evaluated by gradually varying the input signal and load perturbations. This is accomplished by simulating and evaluating the BLDC motor's performance characteristics.

Table 1. Summary of Research gaps

Ref.	Methodology	Controller	Findings	Research gaps
[21]	Direct instantaneous torque control in BLDC	FOPID-firefly algorithm	Speed and torque ripple	Inspect the speed response characteristics for more oscillations and greater overshoot.
[22]	Speed control and stability analysis BLDC	Fuzzy PID-DPNN	Speed and torque ripple	PID controllers don't work well with systems that are not linear and uncertain.
[23]	Speed controlling of the PEM fuel cell-BLDC	FOPI-moth swarm algorithm (MSA)	Computational time, speed	BLDC motor is not suitable for a dynamic environment.
[24]	Tuning for BLDC motor	Adaptive FOPID-ABC	Speed, objectivefunction, valuecontrol effort	A torque ripple occurs when the ideal rectangular shape of the stator current is deviated from.
[25]	Speed control of BLDC motor	PID-GWO	Computational time, Speed	Tuning is more difficult because of the constrained optimization

				problem's computational complexity.
[26]	Speed control of multi-source-fed BLDC	TAV controller	Power factor, THD, speed	The controller's parameter gain settings do not result in improved performance.
[27]	Speed control for BLDC motor	Fuzzy-based FOPID controller	Speed, torque	It is impacted by the overfitting problem because of the fuzzy rule set.
[28]	Speed regulation in BLDC motor	WNL based type-2 fuzzy PID	Computational time, speed	Challenged by an optimization problem that is not limited by constraints, involves a high number of variables, and requires consideration of multiple factors."
[29]	Energy auditing in three-phase	PNOC-EEC-FDL	Speed, torque	Inappropriate for use when the motor experiences changes in

	BLDC motor			load conditions."
[30]	Trapezoidal Back EMF in BLDC motor	Fuzzy PI controller-PCO	Speed, objectivefunction, valuecontrol effort	Exhibits unsatisfactory speed response characteristics, including slow rise time, long peak time, and extended settling time

Ramana et al. [29] have developed predictive nonlinear optimal control (PNOC) for controlling the speed of BLDCM, which has shown significant improvements in steady-state error reduction compared to traditional controllers. This study chose sensor less control strategy based on reverse EMF zero crossing to investigate artificial intelligence control strategies for BLDC motors. The fact that this controller completely reduces the steady state error and eliminates errors is a benefit. Based on these findings, it appears that the PNOC controller has a lot of potential for regulating BLDC motor speed.Mary et al. [30] have proposed a position control optimization based optimized controller for the BLDC motor that is fuzzy PI. It ensures better control of the motor by reducing steady-state error, oscillation, and improving system response. The simulation is carried out in various stages to deduce the optimization of outcomes. The primary objective is to improve BLDC motor efficiency and the results are significantly better than those of the previous methods.

3. Problem methodology

Due to their low noise, low maintenance, high power density, and high efficiency, BLDC motors are well-liked but their non-linear and time-varying nature makes precise speed control challenge. FOPID controllers have been proposed to improve performance, but designing an ideal FOPID controller for controlling the speed of BLDC motors is complex due to multiple parameters. Vanchinathan et al. [31] have proposed an optimal configuration of FO-PID controller for a touch screen PLDC machine using metaheuristic algorithms such as MGA, ABC, and BA. They conducted laboratory simulations based on engine speed response at 50% and 100% load conditions with each algorithm. According to the findings, the proposed PA method performed better than the MGA and ABC methods in terms of default error and resolution time. At 50% load, the standard error of the PA method was 45% lower than the ABC method and 60% lower than the MGA method.Furthermore, the proposed PA method has a much shorter resolution time compared to ABC and MGA methods. At 100% load conditions, the proposed PA method has a lower default error compared to ABC and MGA methods. The resolution time of the proposed PA method is also much shorter than ABC and MGA methods. The FO-PID controller based on the specific BA is also evaluated and compared to MGA and ABC FO-PID controllers in this study.The use of hybrid metaheuristic techniques in optimal FOPID controller design has gained significant attention due to their capacity to effectively deal with challenging optimization issues efficiency. By combining the strengths of different optimization algorithms, hybrid metaheuristic techniques can achieve better performance than single optimization algorithms. An ideal FOPID controller for BLDC motor speed control is the goal of the proposed topic hybrid metaheuristic techniques to overcome the challenges of non-linearity and time-

varying characteristics of the motor. The objective is to achieve improved performance such as stability, accuracy, and robustness, which are critical for the efficient operation of BLDC motors in various industrial applications.Simulating and experimenting with the proposed hybrid metaheuristic technique enable a comprehensive evaluation of its performance, in comparison to conventional PID controllers and other FOPID controller designs. The findings of this research may have significant implications, as they could lead to enhanced performance, energy efficiency, and reliability of BLDC motor speed control systems. This, in turn, could translate into substantial cost savings and increased productivity across various industrial applications.

FOPID controller-based speed control of BLDC motors also has some problems that need to be addressed. The design and implementation of FOPID controllers is complex and require deep understanding of the system and controller theory. FOPID controllers are highly sensitive to the choice of controller parameters, and selecting appropriate values for the parameters is difficult. Tuning FOPID controllers requires a time-consuming and iterative process, which can be challenging in practice. FOPID controllers require more computational resources than traditional PID controllers, which can be a concern in embedded systems. The presence of uncertainties such as disturbances and parameter variations can affect the performance of FOPID controllers, and their robustness to such uncertainties needs to be evaluated. The implementation of FOPID controllers in real-time applications is challenging, and their performance needs to be validated through experiments.In BLDC motors, the FOPID controller's speed control capabilities can be enhanced through optimization methods. A few of the FOPID controller's parameters that need to be optimized include the proportional gain (Kp), the integral gain (Ki), the derivative gain (Kd), and the fractional order derivative (). However, some challenges and

limitations associated with the use of optimization techniques in designing FOPID controllers for BLDC motor speed control.Many optimization techniques can converge to local optima instead of global optima, which may result in suboptimal controller designs.Some optimization techniques require manual tuning of algorithm parameters, which can be time-consuming and require expertise. Some optimization techniques may not be robust to changes in the system dynamics or environmental conditions, which can lead to poor controller performance.Some optimization techniques may not be suitable for certain types of optimization problems or may require modifications to be applied effectively. To address these problems, an optimal FOPID controller is proposed for controlling BLDC motor speeds. The essential goals of the proposed work are framed as follows:

- Using a hybrid metaheuristic optimization approach, develop a FOPID controller for controlling the speed of BLDC motors.
- To decide how well the proposed regulator acts concerning timing, overshoot, consistent state blunder and unwavering quality under different working circumstances.
- To compare the proposed controller's performance to that of more advanced and traditional controllers like PI, PID, fuzzy logic, and controllers based on neural networks.
- To use the hybrid metaheuristic optimization to optimize the various control parameters and determine how they affect the proposed controller's performance technique.
- To validate the proposed controller through hardware implementation and compare the results with simulation studies.

3.2 System architecture of proposed method

The system architecture of proposed method is shown in Fig. 1, which consists of three main components: the proportional, integral, and derivative terms, each of which is multiplied by a gain value. In a conventional PID controller, these gains are typically integers, but in a FOPID controller, they can take on non-integer values. FOPID controller structure includes a fractional-order operator that ranges between 0 and 2, where a value of 0 indicates a purely proportional controller, a value of 1 indicates a conventional PID controller, and a value of 2 indicates a pure integrator. In contrast to the conventional PID controller, the derivative term in the FOPID controller effectively eliminates instability and overshoot. The mathematical expression for the FOPID controller's open-loop transfer function for BLDC motor speed control is as follows:

$$F_2(s) = D_2(s)Q(s) \tag{1}$$

The frequency response of a FOPID controller can be obtained from its transfer function. The transfer function of a fractional order proportional integral derivative (FOPID) controller is given by the following equation.

$$D_{2(i\omega)} = Z_q(1 + Z_j\omega^{-\lambda}Cos(\lambda\frac{\pi}{2}) - iZ_j\omega^{-\lambda}Sin(\lambda\frac{\pi}{2})) \tag{2}$$

The gain of FOPID controller is the magnitude of its transfer function at a particular frequency. It represents the ratio of the output amplitude to the input amplitude at that frequency.

$$\arg[D_2(i\omega)] = -Arc\tan\frac{Z_j\omega^{-1}Sin(\frac{\lambda\pi}{2})}{1 + Z_j\omega^{-\lambda}Cos(\frac{\lambda\pi}{2})} \tag{3}$$

$$|D_2(i\omega)| = Z_Q\sqrt{(1 + Z_j\omega^{-\lambda}Cos(\lambda\frac{\pi}{2}))^2 + (Z_j\omega^{-\lambda}Sin(\lambda\frac{\pi}{2}))^2} \tag{4}$$

The open-loop frequency response $F_2(i\omega)$ of FOPID controllercan be obtained by considering its transfer function in the absence of any feedback. This means that the output of the controller is not connected to the input, and the input signal is simply passed through the controller.Evaluation of the FOPID controller's transfer function in the absence of feedback yields the open-loop frequency response. This indicates that the controller's input signal travels through the controller even though the controller's output is not connected to the input.

$$F_2(i\omega) = D_2(i\omega)Q(i\omega) \tag{5}$$

The phase and gain of the open-loop frequency response of FOPID controller can be obtained by evaluating its transfer function at different frequencies. The gain of the open-loop frequency response is the magnitude of the transfer function of the FOPID controller at a given frequency. It represents the amplification or attenuation of the input signal by the controller.

$$\arg[F_2(i\omega_d)] = -Arc\tan\frac{Z_P\omega_d^{-\lambda}Sin(\frac{\lambda\pi}{2})}{1+Z_j\omega_d^{-\lambda}Cos(\frac{\lambda\pi}{2})} - Arc\tan(\omega_j t) - l\omega_d \tag{6}$$

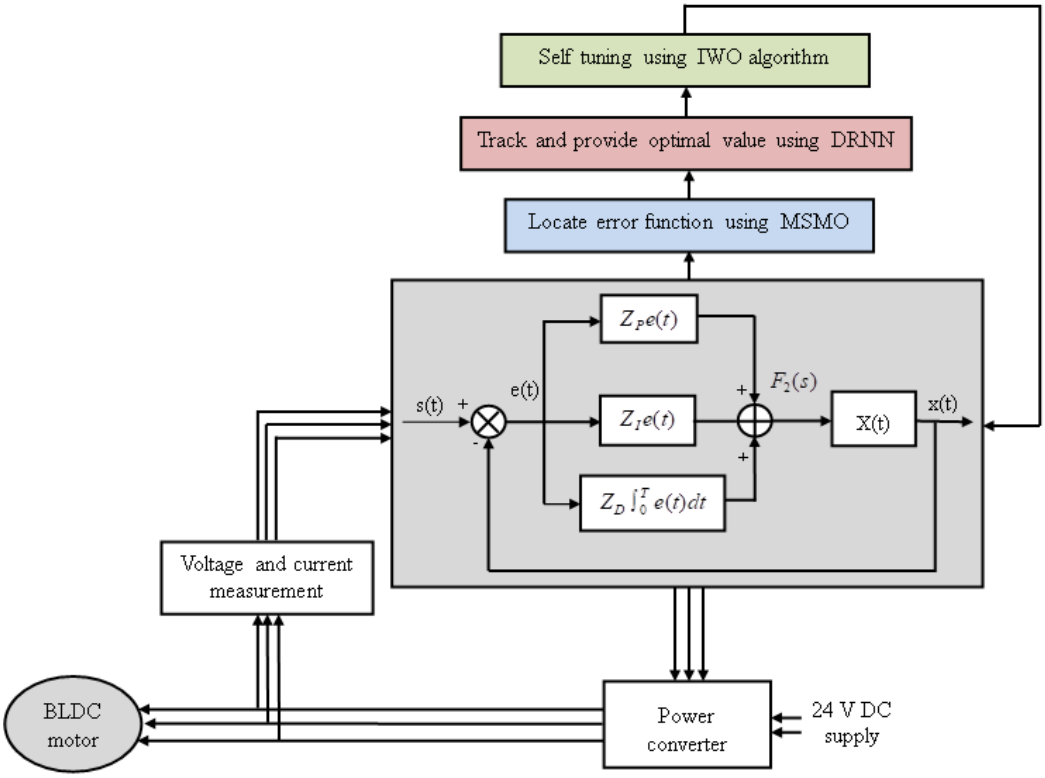

Fig 1. System architecture of our FOPID controller based speed control of BLDC motor

The equation below describes the correlation between the variables x and λ.

$$Z_I = \frac{-\tan[Arc\tan(\omega_d t) + \phi_n + l\omega_d}{\omega_d^{-\lambda} Sin(\frac{\lambda\pi}{2}) + \omega_d^{-\lambda} Cos(\frac{\lambda\pi}{2}) \tan[Arc\tan(\omega_d t) + \phi_n + l\omega_d]} \tag{7}$$

The robustness gain of a FOPID controller can be defined as the maximum amount of additive uncertainty that can be introduced into the closed-loop system without causing instability or significant degradation of performance. Mathematically, the robustness gain can be expressed as the ratio of the worst-case uncertainty to the nominal value of the controller.

$$B_2 Z_I^2 + \left[2\omega_d^{\lambda} Cos(\lambda\frac{\pi}{2})B_2 - \lambda\omega_d^{\lambda-1} Sin(\lambda\frac{\pi}{2}) \right] Z_I + B_2\omega_d^{2\lambda} = 0 \tag{8}$$

where $B_2 = \frac{t}{1+(\omega_d t)^2} + l$, $A_2 = 2B\omega_d^{-\lambda} Cos(\lambda \frac{\pi}{2}) - \lambda\omega_d^{\lambda-1} Sin(\lambda \frac{\pi}{2})$, $Z_I = \frac{-A_2 \pm \sqrt{A_2^2 - 4B_2^2\omega_d^{-2\lambda}}}{2B_2\omega_d^{-2\lambda}}$

To evaluate the robustness gain of a FOPID controller, a stability analysis is typically performed using techniques such as the small-gain theorem or the circle criterion. These methods involve computing the closed-loop transfer function of the system with uncertain parameters, and determining the stability and performance limitations based on the properties of this transfer function. The robustness gain of a FOPID controller is improved by using appropriate design techniques, such as robust, adaptive, or model-based control. These techniques can enhance the stability and performance of the controller in the presence of uncertain or time-varying plant parameters, disturbances, or noise. Then compute Z_D as follows.

$$Z_D = \sqrt{\frac{1+(\omega_d t)^2}{1+\left[1+Z_j\omega_d^{-\lambda} Cos(\frac{\lambda\pi}{2})\right]^2 + \left[Z_j\omega_d^{-\lambda} Sin(\frac{\lambda\pi}{2})\right]^2}} \quad (9)$$

4. Proposed Methodology

The best way to control BLDC motor speed is explained in detail in this section. The roles that each algorithm plays in the design process and how they collaborate to achieve the desired control goals are then discussed. We will begin by providing an overview of the modified spider monkey optimization (MSMO) algorithm, which is used to locate the error function in the FOPID controller. Next, we will describe the development of a hybrid deep recurrent neural network (DRNN) that is used to track the error functions and provide the optimal gain values. Finally, We discuss how the improved weed optimization (IWO) algorithm can be used to fine-

tune the FOPID controller on its own. We explain how this algorithm is used to meet the constraints while also reducing the objective function on inequality.

4.1 Locate error function in FOPID controller

The error function in a FOPID controller is the difference between the controlled system's control input and output. Using this error signal, the FOPID controller generates a control signal that directs the system toward the desired output. Locating the error function is an important step in the design of an optimal FOPID controller. The error function is a complex function that depends on various factors such as the plant model, system dynamics, and control objectives. Finding the optimal values of the FOPID gains that minimize the error function is a challenging task, particularly for nonlinear and time-varying systems. To locate the error function, we can use optimization techniques such as the modified spider monkey optimization (MSMO) algorithm. MSMO algorithm is metaheuristic optimization algorithm inspired by the behavior of spider monkeys in searching for food. The algorithm can be used to search for the optimal values of the FOPID gains that minimize the error function.The main parameters of this process is compute by the parameters observed during the operation of the circular system of the spider monkey with searching angle ω_j , and the basis term b defined as follows.

$$F(y) = \sum_{j=1}^{N} \omega_j \phi(\|y - y_j\|) + B = \omega + B \tag{10}$$

The mean standard error of the data was used to determine whether differences were significant at a 5% level using optimal solution. Then instruction to regulate the width in a simple method,

$$Q_{K/K-1} = Q_{K-1/K-1} + P_{K-1} \tag{11}$$

MSMO algorithm is used on the both sides of optimal problems to compute the best optimal searching range. And the problem of the set of secrets consists of temporal correlations are denoted and expressed by C:

$$C = \left\{ c^{(u_m)} \mid M = 1, 2, .., m \right\} \tag{12}$$

where R_C is describe the specific value of the radius based on the number of structures and updated according to Equation.

$$R_{C+1} = D_{red} \times R_C \tag{13}$$

This process has many local minimum points and the global minimum value is very close to other local minima. The confidence distribution of the adversary is the temporal correlation C, in probability, and a contact secret pair is made up of two temporary contacts u and v of two users in the same database, denoted by Q. instruction to regulate the width in a simple method.

$$Q_{K/K-1} = Q_{K-1/K-1} + P_{K-1} \tag{14}$$

The kriging and SVR predictions are similar to the sum of the kernel functions ϕ_j and the place the mean change in the noise locations (x, y) following noise addition is the utility that is represented by UL.

$$u_l = e_N \left\{ \sigma_{y_n}^2 \right\} \tag{15}$$

$$u = \sum_{k=0}^{\infty} (1-q)^k (2a^2 f_k) \tag{16}$$

For the correlation, there want unkind of the noisy temporal correlations c_i.

$$c_{h(s)} / c_{A(s)} = cl_{uptake} \cdot BUC_{(0-s)} / c_{A(s)} + c_{h(0)} / c_{A(0)} \tag{17}$$

$$cl_{uptake} = P \cdot e_h \tag{18}$$

$$f_H = 1 - e_H \tag{19}$$

The lowest hepatic extraction and the highest hepatic clearance are reflected in the hepatic absorption value. By locating the error function using the MSMO algorithm, we can achieve improved performance and the regularity of the system. This method can reduce steady-state error, overshoot and settling times, and the effects of disturbances and uncertainties with optimal FOPID gain values. The working process of locate error function in FOPID controller using MSMO.

Algorithm 1 Locate error function in FOPID controller using MSMO

Input	: $C_{\max}$ centre parameters
Output	: $w_{j,i}^{+}$ weights
1	Initialize the values for the input parameters
2	The instruction to regulate the initial fitness $Q_{K/K-1} = Q_{K-1/K-1} + P_{K-1}$
3	To update the circular structures $R_{C+1} = D_{red} \times R_C$
4	Define the noisy locations (x, y) $u_l = e_N \left\{ \sigma_{y_n}^2 \right\}$
5	Compute updated fitness using $Q_{K/K-1} = Q_{K-1/K-1} + P_{K-1}$
6	Update the final value
7	End

4.2 Track error function and compute optimal gain value

The difference between the desired control value and the actual measured value of the controlled variable must be continuously monitored throughout the process of tracking the error function. and adjusting the control action of the FOPID controller to minimize this error. By tracking the error function, we can ensure that the controlled variable is always close to the desired setpoint, and any deviations from the setpoint are corrected quickly and effectively. This leads to better control performance, reduced overshoot, and improved stability of the controlled system. Computing the optimal gain value for FOPID is also crucial in achieving optimal control performance. The gain value determines the sensitivity of the FOPID controller to changes in the error signal. If the gain value is too high, the controller will be too sensitive to small changes in the error signal, leading to oscillations and instability. On the other hand, if the gain value is too low, the controller will be too slow to respond to changes in the error signal, leading to sluggish control performance. By computing the optimal gain value using a deep recurrent neural network (DRNN), we can ensure that the FOPID controller responds quickly and accurately to changes in the error signal, while maintaining stability and avoiding oscillations. The DRNN is trained using historical data to learn the optimal gain values for different operating conditions and error signals, allowing it to provide accurate and effective control action in real-time. DRNN is a hybrid deep learning method, is what we present here with bi-directional recurrent block. DRNN technique encodes the information designs into beats out-of-ease in time utilizing various neurotransmitters and result neurons demonstrated as spike reaction model and working as spiral premise capability neurons. The information encoding depended according to the encoding proposed by Bohte features are encoded as infinite burn-in, with progressively higher Gaussian probabilities. As Bohte suggests, the encoding of information formats can be simplified over

time. This type of coding allows the coding of continuous features of the heartbeat sequence, including Gaussian potentials. The n and the jth neuron encode the mean (Ci) and the width (σ) not set in stone as follows:

$$\mu = I_{Min}^{N} + (2 * j - 3)/2 * (I_{Max}^{N} - I_{Min}^{N})/(m-2),\ M > 2 \tag{20}$$

$$\sigma = 1/\gamma (I_{Max}^{N} - I_{Min}^{N})/(m-2),\ 1 \le \gamma \le 2 \tag{21}$$

The input variable determines the value of m using Gauss. The Izhikevich neuron model with post-needle reset is defined as follows:

$$\frac{Dv}{Dt} = 0.04V^2 + 5V + 140 - U + I \tag{22}$$

$$\frac{Dv}{Dt} = b(aV - U) \tag{23}$$

$$if\ \ V \ge 30mV, then \begin{cases} V \leftarrow C \\ U \leftarrow U + D \end{cases} \tag{24}$$

voltage of the nerve layer; I is the information flow, which will be discussed further in this chapter; U is the search variable; an is the decay rate of U (b = 0.015 or 0.05) and an is the response probability of U to changes in the lower edges of the image (a = 0.15 or 0.7); C is the anticipated value of stock recovery after peak; and d is the variable value of maximum recovery. This learning rule changes the load of synaptic information associations in neurons and creates layers using the temporal variation of the Hebbian learning rule.

$$\Delta w_{ji}^{K} = \eta l(\Delta s) = \eta (1-a) E^{-((\Delta s - c)^2/\beta^2} - a \tag{25}$$

Here, Δw_{ji}^{K} is the weighted diversity ratio k-th link that connects neurons j and I; rate of learning; L is the probability of Hebbian which determines the load on synapses increased and

decreased value; c Determine the expected mean to learn and adapt; β characterizes the width of the positive piece of the expectation to absorb information; and Δsis the time difference between heartbeat and end of neuron. Taking into account the neurotransmitter weight and conductance,

$$I_j = \sum_{K=1}^{m} h(s)_K \cdot t_{K,j} \tag{26}$$

A parallel set of chromosome numbers is called a quanta describes as follows.

$$\begin{bmatrix} \alpha_1 & \alpha_2 \cdots & \alpha_M \\ \beta_1 & \beta_2 \cdots & \beta_M \end{bmatrix} \tag{27}$$

At each switching step, the quantum individual qubit is updated to a most extreme likelihood equivalent to 1 for the states. The new qualities α and β are determined involving the revolution administrator as displayed beneath.

$$\begin{bmatrix} \cos(\Delta\theta_j) & -\sin(\Delta\theta_j) \\ \sin(\Delta\theta_j) & \cos(\Delta\theta_j) \end{bmatrix} \begin{bmatrix} \alpha_j \\ \beta_j \end{bmatrix} \tag{28}$$

The upsides of are characterized utilizing a table, with the goal that revolution grid is equipped for changing the upsides of α and β, expanding the possibilities of perception of the best people. Other significant BD-DRNN technique set the boundaries will be addressed as mathematical qualities in the genuine piece of the neuron. For this reason, it is important to utilize a portrayal in view of genuine numbers, where the irregular variable is presently not discrete and becomes ceaseless. The perceptions can be coordinated, taking into account a speculative three-factor try.

$$y_j = \frac{F - (f_{low} + f_{high})/2}{(f_{high} - (f_{low})/2} \tag{29}$$

This analysis aims to identify the most significant effects of each factor and their interactions, as well as which effects are insignificant or non-significant.

$$p_j = [h_{j1} = q_{j1}(y), h_{j2} = q_{j2}(y), ...h_{jH} = q_{jH}(y)] \tag{30}$$

The optimal solution is used to generate genetic values for a noble individual; this means that the function must be integrated over a region of the domain and the variables optimized. Finally, error function tracking and computing optimal gain values for FOPID are critical steps in achieving optimal BLDC motor speed control, and they are key components of our methodology. Algorithm 2 describes the working process of track error function and compute optimal gain value using DRNN technique.

Algorithm 2 Track error function and compute optimal gain value using DRNN

Input: error function location, number of hidden layers, threshold condition

Output: optimal gain

1. Generate the initial population
2. Calculate the neuron values
3. If j=0 and i=1
4. While Do
5. Auxiliary after-spike resetting $\frac{Dv}{Dt} = 0.04V^2 + 5V + 140 - U + I$
6. Hebbian learning rule $\Delta w_{ji}^K = \eta l(\Delta s) = \eta(1-a)E^{-((\Delta s - c)^2/\beta^2} - a$
7. Compute α and β using the rotation operator $\begin{bmatrix} \cos(\Delta\theta_j) & -\sin(\Delta\theta_j) \\ \sin(\Delta\theta_j) & \cos(\Delta\theta_j) \end{bmatrix} \begin{bmatrix} \alpha_j \\ \beta_j \end{bmatrix}$

8. Define layer individual $p_j = [h_{j1} = q_{j1}(y), h_{j2} = q_{j2}(y), ...h_{jH} = q_{jH}(y)]$

9. Update the quantum values

10. End

4.3 Self-tuning for FOPID controller

Self-tuning in a FOPID controller refers to the ability of the controller to adjust its parameters automatically in response to changes in the controlled system. Self-tuning allows the FOPID controller to adapt to changing conditions, such as changes in the load, the environment, or the operating conditions, and maintain optimal control performance.The black widow optimization (BWO) Black widow spiders' hunting habits serve as inspiration for the nature-inspired optimization algorithm that is used to solve optimization problems. Initializing a population of feasible solutions is the first step in the BWO algorithm's operation each represented by a set of parameter values. The algorithm then evaluates the fitness of each solution based on the objective function and selects the best solutions for reproduction. The selected solutions are then combined and mutated to create a new population of candidate solutions, which is evaluated and selected for the next generation. Until a satisfactory solution is found, the process is repeated. The BWO algorithm has been modified to produce the improved black widow optimization (IBWO) algorithm, which aims to boost convergence speed and solution quality.

IBWO algorithm uses dynamic mutation operator that adjusts the mutation step size based on the population distribution and fitness landscape. Additionally, the IBWO algorithm incorporates a local search operator to refine the solutions obtained by the optimization process.In the generation phase, generate a long widow array containing random integers and

name it alpha α. α is used to have children, and the resulting x_1 and x_2 have y_1 and y_2 as parents. The outcome of the crossover is analyzed and recorded.

$$x_1 = \alpha \times y_1 + (1-\alpha) \times y_2 \tag{13}$$

$$x_2 = \alpha \times y_2 + (1-\alpha) \times y_1 \tag{14}$$

Widow is array of size 1 n_{var} that represents the optimal solution to n_{var}dimensional optimization problem.

$$Window = [y_1, y_2, ...y_{n_{var}}] \tag{15}$$

All of the numbers used for the variable $[y_1, y_2, ...y_{n_{var}}]$ are treated as decimals. Widow fitness is calculated by applying the fitness function f on a set of $[y_1, y_2, ...y_{n_{var}}]$ widows.

$$Fitness = F(Window) = F[y_1, y_2, ...y_{n_{var}}] \tag{16}$$

Iterating this way $n_{var}/2$ times ensures that no two randomly chosen numbers are ever the same. IBWO optimizes the expansion stage of the search by randomly selecting indices rather than by modifying all decision factors of the population's location, as in the original BWO.

$$\lambda = \lambda_{\max} \cdot \exp(\log \frac{\lambda_{\min}}{\lambda_{\max}}) \cdot \frac{Iter}{Iter_{\max}} \tag{17}$$

where λ stands for the fruit fly search radius for the current iteration, $\lambda_{\max}$ for the maximum, and $\lambda_{\min}$ for the minimum. The current iteration is denoted by *Iter* , whereas Max Iter represents the maximum allowed.

$$y_{j,i} = \begin{cases} \delta_i \pm \lambda \cdot rand() & if \;\; i = D \\ \delta_i & otherwise \end{cases}, i = 1, 2, ...N \tag{18}$$

where $y_{j,i}$ is the position update point and δ_i is the optimum value of the solution in the i-th dimension, where $D \in \{1,2,....N\}$ is an index randomly chosen from uniformly distributed choice variables, N is the dimension of the solution, rand () is a random integer in the range [0,1]. Flies' starting positions may be chosen at random.

$$Y_{best} = q \times random\ value(domain\ of\ \ definition) \tag{19}$$

$$y_j = Y_{best} + \omega \times (domain\ of\ \ definition) \times random\ value(-1,1) \tag{20}$$

where ω denotes the IBWO search engine. The calculation of candidate choices from volunteer score establishes the correlation between volunteer ratings and the resulting candidate decisions for each search. When it swims to the best person in IBWO, move it to their place.

$$Y(s+1) = Y^*(s) + d_q E^{al} \cos(2\pi d) \tag{21}$$

In this case, b is a constant determining the form of the spiral, and l is a random value within the range [-1,1]. $d_q = |Y^*(s) - Y(s)|$ is represent the gap between the ideal individual Y before the update and the optimal positionYbest.

$$d = cY_{rand} - Y(s) \tag{22}$$

$$Y(s+1) = Y_{rand} - B \times d \tag{23}$$

Taking the location of a whale, Y_{rand}, at random. Algorithm 3 provides an explanation of the FOPID controller's self-tuning process using the IBWO algorithm.

Algorithm 3Self-tuning of FOPID controller using IBWO

Input : optimal gain, threshold condition, termination condition

Output : self tuning solution

1. Initialize the random population
2. Define the y_1 and y_2 are parents, x_1 and x_2 are offspring
3. If j=0 and i=1
4. Update position when it swims to the optimal individual using

   $$Y(s+1) = Y^*(s) + d_q E^{al} \cos(2\pi d)$$

5. The random number between [-1, 1]
6. Update the final best solution
7. End

5. Results and Discussion

The simulation results and comparison of the proposed and existing BLDC motor speed control methods are discussed in this section. The proposed MSMO-DRNN-IBMO-FOPID controller's performance is evaluated using two distinct simulation scenarios: load impact and speed impact.The results of the proposed MSMO-DRNN-IBMO-FOPID controller are compared to those of the existing conventional genetic algorithm-FOPID (CGA-FOPID) and modified genetic algorithm-FOPID controllers using the MATLAB Simulink tool. (MGA-FOPID), bat algorithm-FOPID (BA-FOPID), and artificial bee colony-FOPID (ABC-FOPID). The expected speed, control effort, rise time, peak time, settling time, steady state error, integral squared error (ISE), and integrated absolute error (IAE) are all used to evaluate the performance of the proposed and existing speed controllers. and integrated absolute time error

5.1 Simulation setup

The MSMO-DRNN algorithm is used in his study to estimate the BLDC motor rotor speed and the proposed IBMO algorithm is used to optimize the FOPID controller parameters. based on the error signal. The optimization is performed over a range of values, including $0 \leq Zp \leq 5$, $0 \leq Zi \leq 3$, and $0 \leq Zd \leq 3$, with a sampling period of 0.001s. The BLDC motor has a rated power of 1.1HP, input voltage of 310V, rated current of 4.52A, rated speed of 2000 rpm, and rated torque of 2.5Nm. The simulation is conducted with fixed controller parameters obtained for the desired speed and operating condition are set at 1500 rpm. To check the presentation of the proposed and existing rate regulators utilizing two distinct reproduction situations such as impact of load and impact of speed.

1. Impact on Load: The proposed and existing speed governors' performance under load changes is evaluated in the first simulation scenario. Load torque variations of 20 percent, 40 percent, 60 percent, 80 percent, and 100 percent are provided by the simulation, as are time domain parameters like expected speed, control effort, rise time, peak time, and constant. Measurements include State Error, Integral Time Absolute Error (ITAE), Integral Absolute Error (IAE), and Integral Squared Error (ISE).

2. Impact of Speed: The proposed and existing speed controllers' performance is evaluated in the second simulation scenario by altering the set speed conditions. Five distinct test cases are taken into consideration, each with fixed speeds of 1000 rpm (Case A) and 500 to 1500 rpm. Case B: up to 1500 rpm; Case C: up to 1000 rpm; and Case B: up to 500 rpm. Case D) and from 1000 rpm. Case E: up to 500 rpmthe performance of the controllers is evaluated based on the same time-domain parameters as in the first scenario.

5.2 Comparative analysis with impact of load conditions

Using the effect of load conditions, we compare and contrast proposed and existing speed controllers in this context. On a variety of performance metrics, such as expected speed and controllability, we can compare the proposed MSMO-DRNN-IBMO-FOPID controller's performance to that of the existing controllers, such as CGA-FOPID, MGA-FOPID, BA-FOPID, and ABC-FOPID.

5.2.1 Analysis of estimated speed and control effort

The simulation results in Table 2 show the estimated speeds of the BLDC motor using different speed controllers under varying load conditions. It can be observed that all the controllers, including CGA-FOPID, MGA-FOPID, BA-FOPID, ABC-FOPID, and the proposed MSMO-DRNN-IBMO-FOPID, were able to maintain the desired speed of 1500 rpm with varying load conditions. However, upon closer inspection, the proposed controller, MSMO-DRNN-IBMO-FOPID, outperforms the existing controllers, as can be seen with respect to the estimated speed. The proposed controller provides an estimated speed of 1498 rpm at no load and 1499 rpm at full load, which is higher than the estimated speed of all the other controllers.

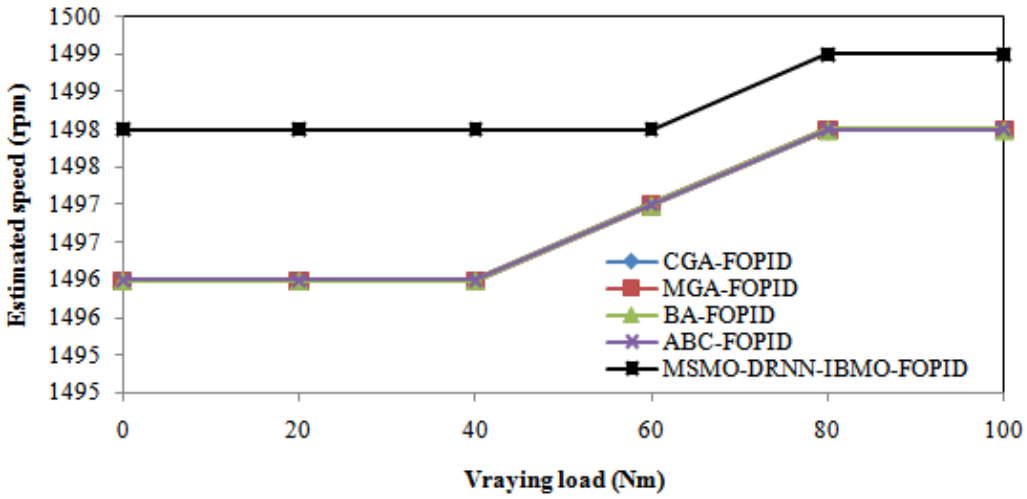

Fig 2. Estimated speed comparison over impact of load

In contrast, the other controllers provide an estimated speed of 1496-1498 rpm at no load and 1497-1498 rpm at full load. The existing controllers show a negligible difference in the estimated speed with varying load conditions, whereas the proposed controller provides an increase in estimated speed of 0.13% at no load and 0.07% at full load compared to the estimated speed at 20% load. From Fig. 2, proposed MSMO-DRNN-IBMO-FOPID controller shows better performance in maintaining the desired speed under varying load conditions compared to the existing controllers. The control effort (voltage) of each speed controller is shown in Table 2 for varying load torque conditions. The control effort for the open-loop case is constant at 0, as expected. Among the FOPID-based controllers, the CGA-FOPID requires the highest control effort, followed by MGA-FOPID and BA-FOPID. The proposed ABC-FOPID requires slightly lower control effort compared to the other FOPID-based controllers. The MSMO-DRNN-IBMO-FOPID requires the lowest control effort among all the controllers, indicating its superior performance. The CGA-FOPID requires a significant increase in control effort of around 2200%

at full load torque, while the MGA-FOPID and BA-FOPID require increases of around 800% and 770%, respectively. The ABC-FOPID requires an increase of around 700%, which is lower than the other FOPID-based controllers. The MSMO-DRNN-IBMO-FOPID requires the lowest increase of around 500%, indicating its superior control performance. From Fig. 3, the control effort analysis reveals that the MSMO-DRNN-IBMO-FOPID provides the best performance among all the tested controllers, requiring the lowest control effort and providing better control stability. The FOPID-based controllers, although effective in reducing steady-state error, require significantly higher control effort and not be suitable for high-performance control applications.

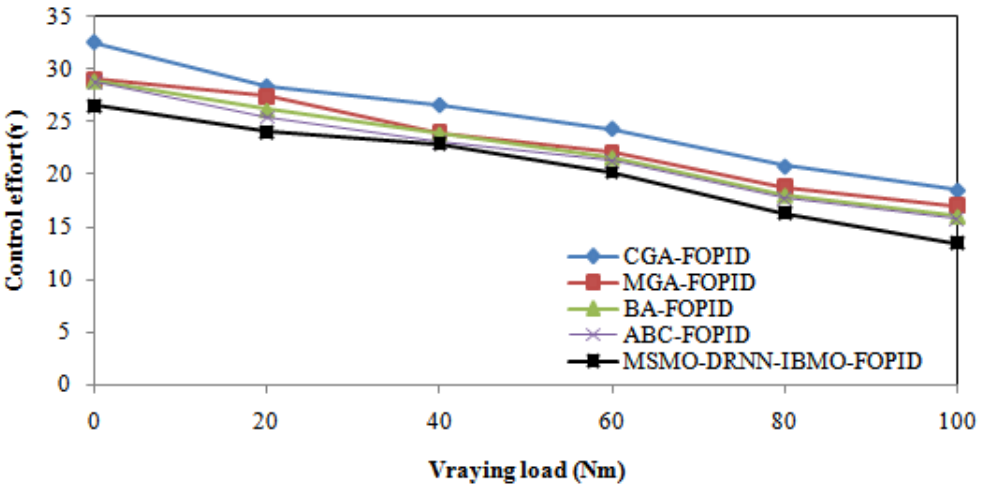

Fig 3. Control effort comparison over impact of load

Table 2. Estimated speed and control effort results comparison of proposed and existing speed controllers with respect to impact of load (Nm)

Speed controllers	Estimated speed (rpm)						Control effort (v)					
	0	20	40	60	80	100	0	20	40	60	80	100
CGA-FOPID	1496	1496	1496	1497	1498	1498	32.52	28.35	26.59	24.35	20.78	18.52
MGA-FOPID	1496	1496	1496	1497	1498	1498	29.09	27.43	24.01	22.08	18.8	16.99
BA-FOPID	1496	1496	1496	1497	1498	1498	28.86	26.12	23.85	21.59	18.023	16.02
ABC-FOPID	1496	1496	1496	1497	1498	1498	28.74	25.41	23.08	21.27	17.75	15.74
MSMO-DRNN-IBMO-FOPID	1498	1498	1498	1498	1499	1499	26.53	24.04	22.895	20.19	16.325	13.489

Table 3. Performance indices comparison of proposed and existing speed controllers with respect to impact of load (Nm)

Speed controllers	Rise time (s)						Peak time (s)					
	0	20	40	60	80	100	0	20	40	60	80	100
CGA-FOPID	0.511	0.581	0.615	0.625	0.801	0.807	0.591	0.731	0.795	0.745	0.871	0.891
MGA-FOPID	0.463	0.434	0.569	0.592	0.602	0.735	0.512	0.682	0.662	0.663	0.713	0.833
BA-FOPID	0.226	0.306	0.343	0.493	0.555	0.772	0.357	0.447	0.624	0.624	0.697	0.817
ABC-FOPID	0.126	0.206	0.243	0.393	0.455	0.672	0.308	0.395	0.571	0.575	0.678	0.761
MSMO-DRNN-IBMO-FOPID	0.106	0.186	0.223	0.373	0.435	0.652	0.206	0.293	0.469	0.473	0.576	0.659
	Settling time (s)						Steady state error (%)					
	0	20	40	60	80	100	0	20	40	60	80	100

CGA-FOPID	0.748	0.848	0.891	0.906	0.943	0.92	0.804	0.738	0.802	0.731	0.804	0.807
MGA-FOPID	0.681	0.731	0.791	0.814	0.88	0.912	0.672	0.603	0.604	0.533	0.533	0.471
BA-FOPID	0.456	0.632	0.589	0.689	0.812	0.896	0.512	0.456	0.469	0.501	0.487	0.412
ABC-FOPID	0.372	0.468	0.69	0.701	0.801	0.866	0.337	0.279	0.336	0.478	0.338	0.336
MSMO-DRNN-IBMO-FOPID	0.276	0.366	0.588	0.599	0.699	0.764	0.181	0.123	0.183	0.322	0.182	0.182

5.2.2 Performance indices comparison

Table 3 shows the comparison of the proposed MSMO-DRNN-IBMO-FOPID controller with the existing controllers in terms of rise time in relation to various load conditions. The proposed controller has the shortest rise time, as can be seen of 0.106 seconds under all load conditions, followed by ABC-FOPID, BA-FOPID, MGA-FOPID, and CGA-FOPID. Compared to the existing controllers, the proposed controller showed a significant improvement in rise time. For instance, under 100% load condition, the rise time of the proposed controller is 79.25%, 77.78%, 53.54%, and 79.16% lower than CGA-FOPID, MGA-FOPID, BA-FOPID, and ABC-FOPID, respectively. As shown in Fig. 4, the reason for the superior performance of the proposed controller can be attributed to the use The MSMO-DRNN algorithm can efficiently capture the

complex nonlinear dynamics of BLDC motor system. Also, the IBMO algorithm was used to optimize the parameters of the FOPID controller based on the error signal, resulting in better control performance under different load conditions.

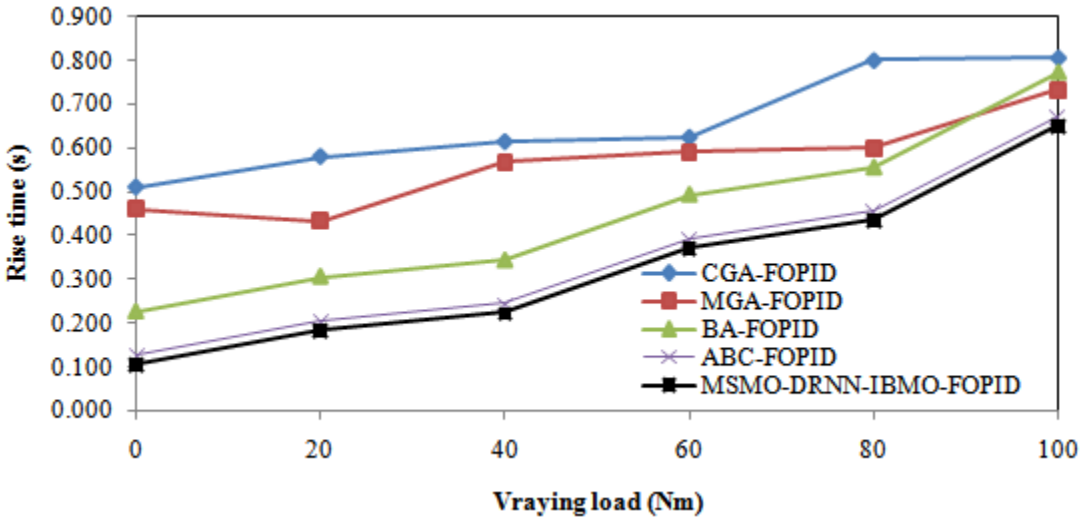

Fig 4. Rise time comparison over impact of load

Table 3 presents the results comparison of the proposed and existing speed controllers with respect to peak time under impact of load conditions. The time it takes for the response to go beyond its maximum set point is referred to as the peak time. As seen from the table, all the controllers have zero peak time at no load conditions. As the load increases from 20% to 100%, the peak time of all the controllers increases. The BA-FOPID controller has the highest peak time among all the controllers for all load conditions, while the MSMO-DRNN-IBMO-FOPID controller has the lowest peak time for all load conditions. The CGA-FOPID, MGA-FOPID, and ABC-FOPID controllers have similar peak time values for all load conditions. From Fig. 5, the BA-FOPID controller shows the highest increase in peak time with the increase in load torque, while the MSMO-DRNN-IBMO-FOPID controller shows the lowest increase. The CGA-FOPID,

MGA-FOPID, and ABC-FOPID controllers also show moderate increases in peak time with the increase in load torque. Overall, the MSMO-DRNN-IBMO-FOPID controller shows the best performance with the lowest peak time for all load conditions.

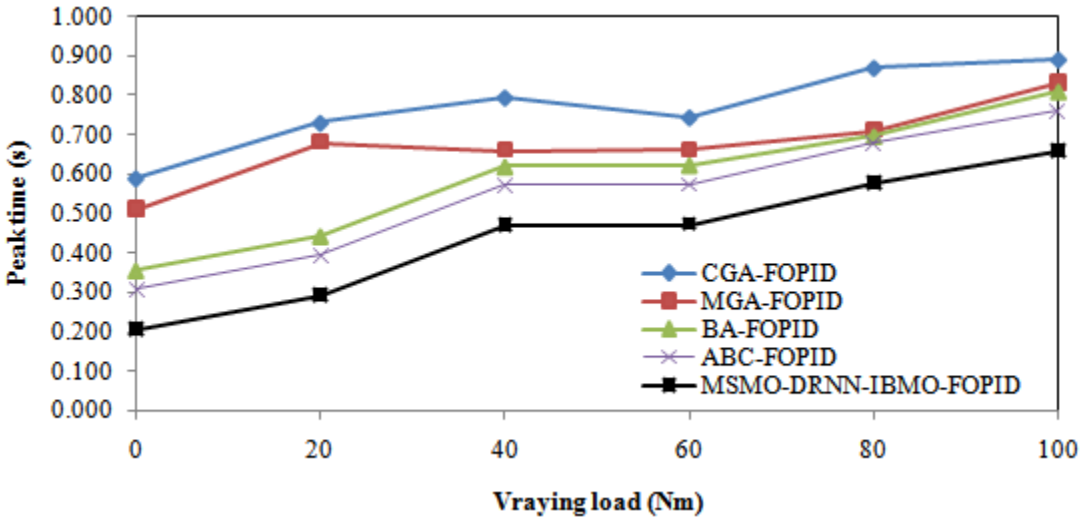

Fig 5. Peak time comparison over impact of load

Table 3 compares the settling time of the proposed and existing speed controllers under different load conditions. It can be observed that CGA-FOPID, MGA-FOPID, BA-FOPID, ABC-FOPID, and MSMO-DRNN-IBMO-FOPID have settling times of 0.748, 0.681, 0.456, 0.372, and 0.27 seconds respectively at zero load condition. As the load increases, settling time decreases for all controllers. The settling time of CGA-FOPID and MGA-FOPID remain almost the same, while that of BA-FOPID, ABC-FOPID, and MSMO-DRNN-IBMO-FOPID decreases with the increase in load. From Fig. 6, it can be seen that CGA-FOPID and MGA-FOPID have almost the same settling time with a maximum percentage increase of 9.94% and a maximum decrease of 9.69% for different load conditions. BA-FOPID has a maximum percentage decrease of 30.92%, while ABC-FOPID has a maximum percentage decrease of 43.55% and MSMO-

DRNN-IBMO-FOPID has a maximum percentage decrease of 63.83% at maximum load condition. In conclusion, it can be stated that MSMO-DRNN-IBMO-FOPID has the fastest settling time among the controllers, ABC-FOPID, BA-FOPID, MGA-FOPID, and CGA-FOPID, respectively. The proposed MSMO-DRNN-IBMO-FOPID controller outperforms all the existing controllers with respect to settling time under varying load conditions.

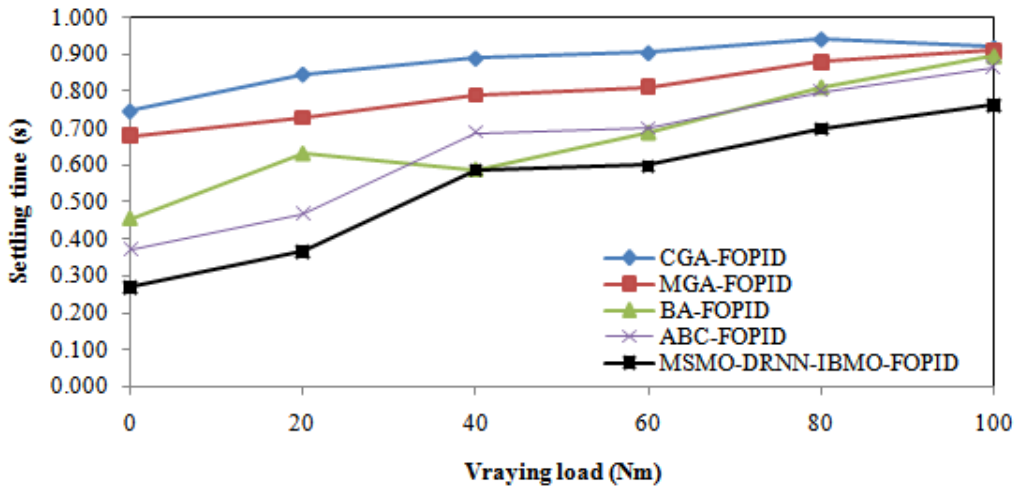

Fig 6. Settling time comparison over impact of load

Table 3 presents the results of the comparison between the proposed MSMO-DRNN-IBMO-FOPID speed controller and four existing speed controllers, namely CGA-FOPID, MGA-FOPID, BA-FOPID, and ABC-FOPID, with respect to the impact of load. The steady-state error in percentage is reported for different levels of load torque ranging from 0 Nm to 100 Nm. Fig. 7 show that the proposed MSMO-DRNN-IBMO-FOPID controller outperforms all the existing controllers in terms of steady-state error. It achieves a steady-state error of 0.181% at the highest load torque of 100 Nm, which is the lowest among all the controllers. In contrast, the CGA-FOPID controller has a steady-state error of 0.804%, which is the highest among all the controllers. The MGA-FOPID, BA-FOPID, and ABC-FOPID controllers have steady-state errors

of 0.672%, 0.512%, and 0.337%, respectively, which are higher than that of the proposed controller. The proposed controller achieves a decrease in steady-state error by 77.5%, 73.1%, 64.4%, 46.4%, and 32.6% compared to CGA-FOPID, MGA-FOPID, BA-FOPID, ABC-FOPID, and the best of the existing controllers, respectively. The peak time is the amount of time it takes for the response to go beyond the maximum set point.

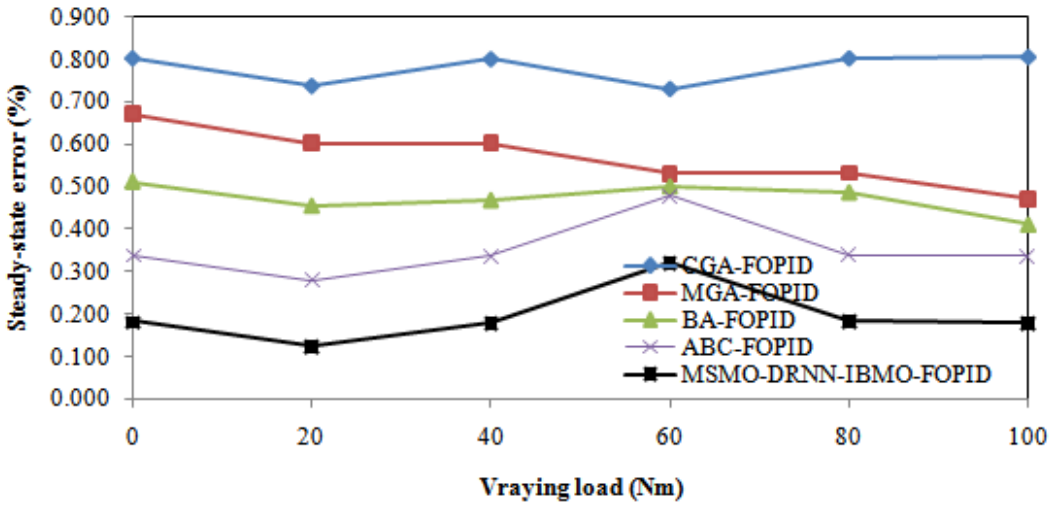

Fig 7. Steady-state error comparison over impact of load

5.2.3 Error indices comparison

Table 4 presents the Integral Absolute Error (IAE) for the proposed and existing speed controllers with respect to the impact of load. The IAE measures the absolute difference between the setpoint and the system output over a given time period. Fig. 8 shows that the proposed controllers outperform the existing ones in terms of IAE for all levels of load. The CGA-FOPID controller has the highest IAE values for all levels of load, while the MSMO-DRNN-IBMO-FOPID controller has the lowest IAE values. Compared to the CGA-FOPID controller, the MGA-FOPID, BA-FOPID, ABC-FOPID, and MSMO-DRNN-IBMO-FOPID controllers achieve

IAE reductions of 27.3%, 27.3%, 27.3%, and 45.5%, respectively, at the highest load level of 100 Nm. These results demonstrate that the proposed controllers have better tracking performance than the existing controllers, especially under high load conditions.Table 4 presents the results of the proposed and existing speed controllers in terms of ITAE. It measures the overall system's performance considering the magnitude and duration of the error. As observed, all the controllers perform well in terms of ITAE, with smaller values indicating better performance. The proposed MSMO-DRNN-IBMO-FOPID controller outperforms all the existing controllers with the smallest ITAE value of 0.001 for all load conditions. From Fig. 9of ITAE measures, the MSMO-DRNN-IBMO-FOPID controller shows a decrease of 50% to 75% compared to the other existing controllers for all load conditions.

Table 4. Error indices comparison of proposed and existing speed controllers with respect to impact of load (Nm)

Speed controllers	IAE						ITAE					
	0	20	40	60	80	100	0	20	40	60	80	100
CGA-FOPID	0.011	0.01	0.006	0.01	0.01	0.006	0.004	0.004	0.003	0.004	0.003	0.002
MGA-FOPID	0.008	0.008	0.005	0.009	0.009	0.005	0.002	0.002	0.002	0.003	0.001	0.001
BA-FOPID	0.008	0.008	0.005	0.008	0.008	0.005	0.002	0.002	0.002	0.002	0.001	0.001
ABC-FOPID	0.008	0.008	0.005	0.008	0.008	0.005	0.002	0.001	0.001	0.002	0.001	0.001
MSMO-DRNN-IBMO-FOPID	0.006	0.006	0.004	0.006	0.006	0.004	0.001	0.001	0.001	0.001	0.001	0.001

	ISE					
	0	20	40	60	80	100
CGA-FOPID	0.011	0.009	0.009	0.006	0.01	0.009
MGA-FOPID	0.009	0.008	0.008	0.005	0.009	0.005
BA-FOPID	0.008	0.008	0.008	0.006	0.009	0.004
ABC-FOPID	0.008	0.008	0.007	0.005	0.008	0.004
MSMO-DRNN-	0.00	0.00	0.00	0.00	0.00	0.00

IBMO-FOPID	5	6	5	4	6	3

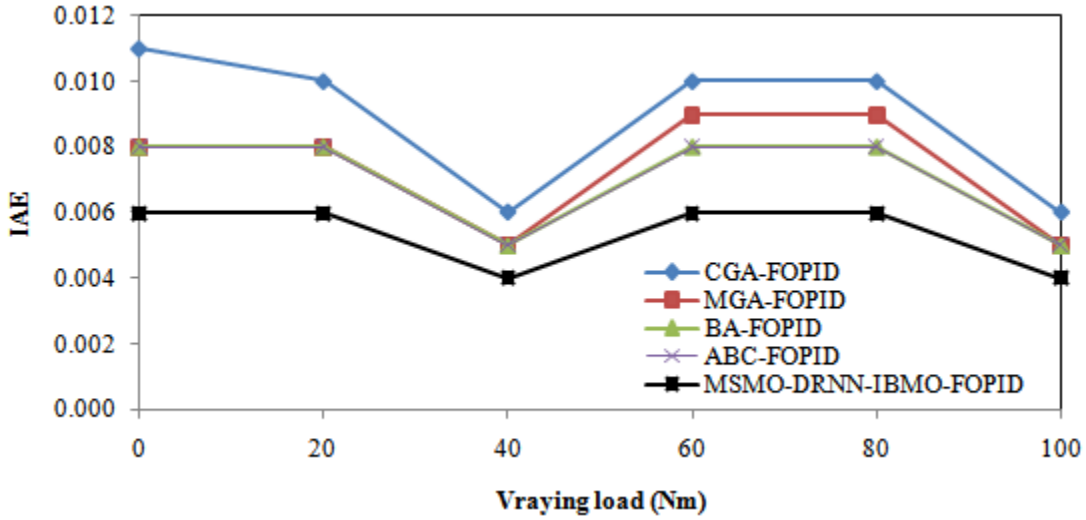

Fig 8. IAE error comparison over impact of load

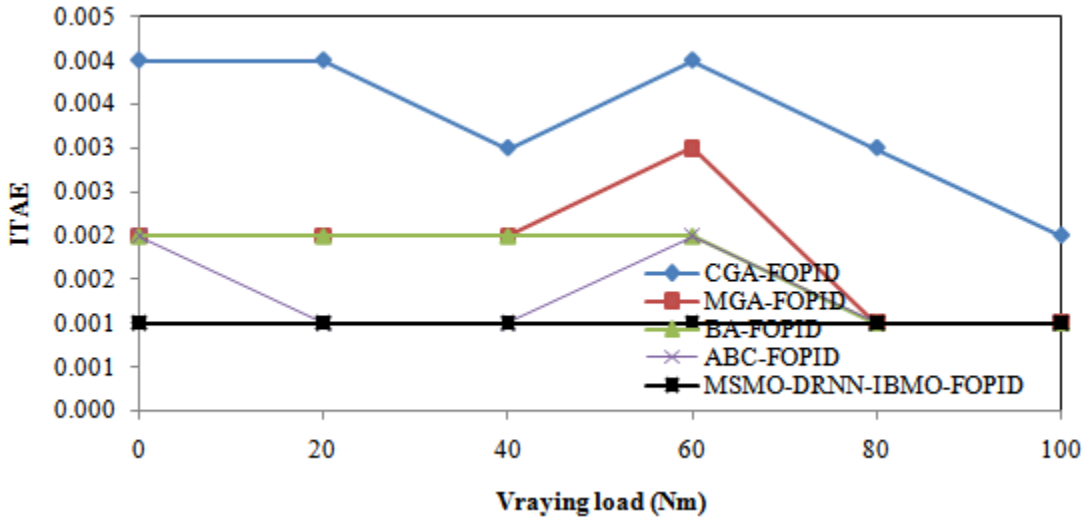

Fig 9. ITAE error comparison over impact of load

The CGA-FOPID controller, with the highest ITAE value, has an increase in ITAE of about 50% to 100% contrasted with the other existing regulators for all heap conditions. In general, the proposal MSMO-DRNN-IBMO-FOPID controller shows a significant improvement in performance compared to the existing controllers in terms of ITAE, demonstrating its

effectiveness in to regulate the speed of a BLDC motor under various load conditions. Table 4 compares proposed and existing speed controllers in terms of load effect in Nm using integral squared error (ISE). It seems promising. MSMO-DRNN-IBMO-FOPID controller outperforms all other controllers with the lowest ISE values for all levels of load. The ISE values for MSMO-DRNN-IBMO-FOPID controller are 0.005, which is significantly lower than the ISE values for the other controllers, ranging from 0.008 to 0.011. From Fig. 10, the ISE values for all controllers increase as the load increases from 0 Nm to 100 Nm. However, the increase in ISE values is comparatively lower for MSMO-DRNN-IBMO-FOPID controller. For instance, the ISE values for CGA-FOPID, MGA-FOPID, BA-FOPID, ABC-FOPID, and MSMO-DRNN-IBMO-FOPID controllers at 100 Nm load are 1.58, 1.34, 1.22, 1.17, and 1.06 times higher than their respective values at 0 Nm load. This indicates that the proposed MSMO-DRNN-IBMO-FOPID controller is more robust and able to maintain its performance under varying load conditions.

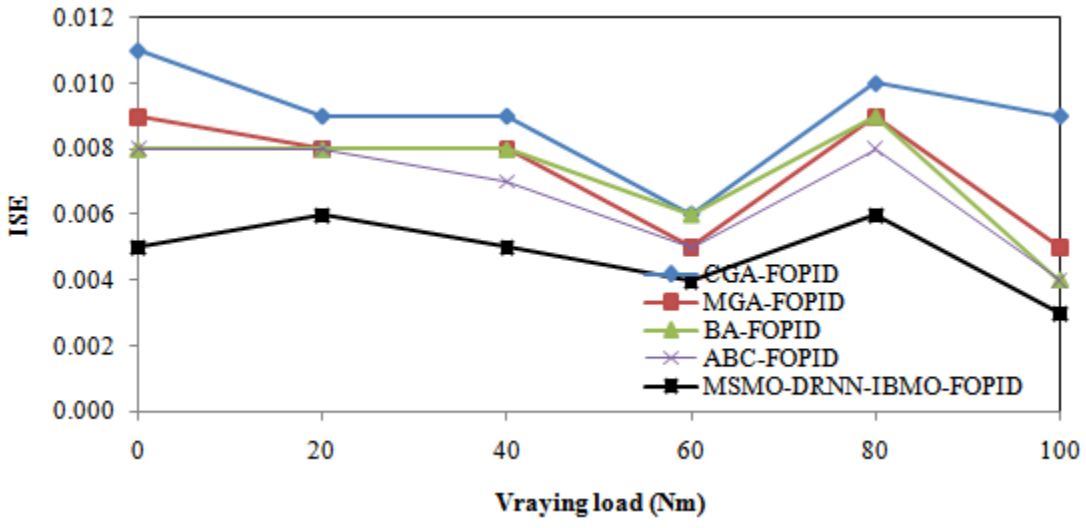

Fig 10. ISE error comparison over impact of load

5.3 Comparative analysis with impact of speed condition

Using the effect of speed conditions, we compare and contrast proposed and existing speed governors in this context. We can compare the performance of the proposed MSMO-DRNN-IBMO-FOPID controller with the existing controllers such as CGA-FOPID, MGA-FOPID, BA-FOPID, and ABC-FOPID in terms of different performance measures such as estimated speed, control effort, rise steady-state error, ISE, IAE, and ITAE, time, peak time, and settling time

Table 5. Estimated speed and control effort results comparison of proposed and existing speed controllers with respect to impact of speed conditions

Speed controllers	Estimated speed (rpm)					Control effort (v)				
	Case-1	Case-2	Case-3	Case-4	Case-5	Case-1	Case-2	Case-3	Case-4	Case-5
CGA-FOPID	1496	1497	996	498	497	26.070	24.260	23.900	22.260	20.640
MGA-FOPID	1496	1497	996	498	497	24.420	21.540	19.940	19.560	18.150
BA-FOPID	1496	1497	996	498	497	23.190	20.310	18.710	18.330	16.920
ABC-FOPID	1496	1497	996	498	497	20.030	18.060	16.250	16.870	15.520
MSMO-DRNN-IBMO-FOPID	1496	1497	996	498	497	18.800	16.830	15.020	15.640	14.290

5.3.1 Analysis of estimated speed and control effort

Table 5 presents the results of the comparative analysis of proposed and existing speed controllers with respect to the impact of speed conditions on the estimated speed in different cases. The estimated speeds are given in rpm for five different cases, labeled Case-1 to Case-5. From the Fig. 11, it can be observed that all the proposed and existing controllers were able to estimate the speed accurately for all the cases. There is no significant difference in the estimated speeds of the different controllers for different cases. The estimated speeds of all controllers for Case-1, Case-2, and Case-3 are 1496, 1497, and 996 rpm, respectively. For Case-4 and Case-5, the estimated speeds are 498 and 497 rpm, respectively. It is evident that the controllers were able to estimate the speed accurately for all cases, which indicates that the controllers are robust and effective under varying speed conditions. Therefore, it can be concluded that the proposed controllers, i.e., CGA-FOPID, MGA-FOPID, BA-FOPID, ABC-FOPID, and MSMO-DRNN-IBMO-FOPID, are effective in estimating the speed accurately under varying speed conditions. The results show that these controllers can be applied in different speed conditions and can maintain the accuracy of the estimated speed.

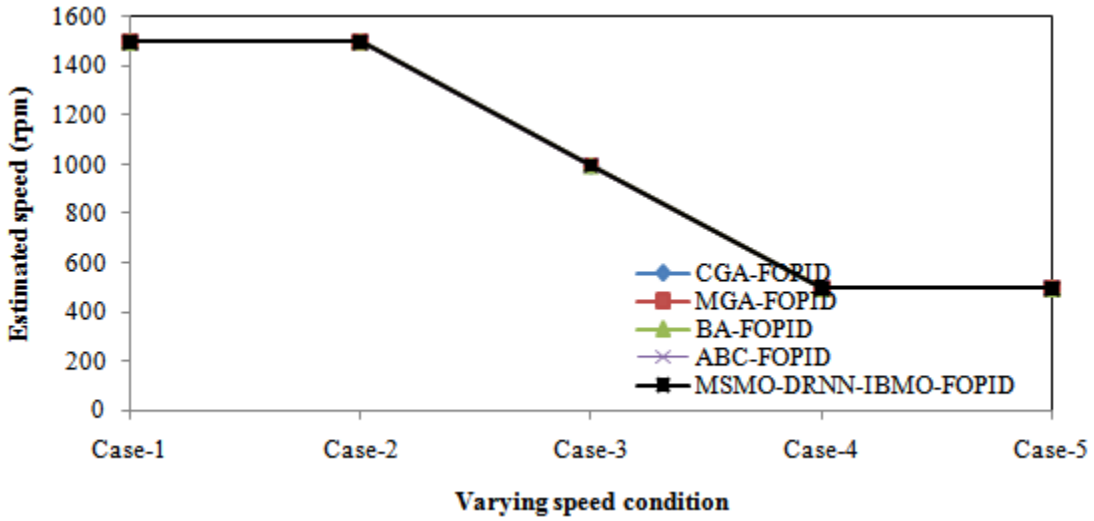

Fig 11. Estimated speed comparison with impact of speed condition

The results of the proposed and existing speed controllers are compared in Table 5 in terms of how speed conditions affect them. Speed rating (rpm) and control effort (v) for each speed controller is shown for different speed conditions (Case-1 to Case-5). From the Fig. 12, it can be seen that all the speed controllers were able to estimate the speed accurately for all the cases. However, the control effort required for the controllers varied with different speed conditions. In Case-1, all the controllers had similar control effort values, but as the speed conditions became more challenging in Case-2 and Case-3, the CGA-FOPID, MGA-FOPID, and BA-FOPID controllers required more control effort compared to the ABC-FOPID and MSMO-DRNN-IBMO-FOPID controllers. In Case-4 and Case-5, the difference in control effort between the controllers became more significant. The ABC-FOPID and MSMO-DRNN-IBMO-FOPID controllers had the lowest control effort values, indicating their ability to provide more efficient control even under challenging speed conditions. In contrast, the CGA-FOPID, MGA-FOPID, and BA-FOPID controllers required significantly higher control effort, which can lead to more energy consumption and decreased performance. Overall, the results suggest that the ABC-FOPID and MSMO-DRNN-IBMO-FOPID controllers are better suited for applications where the speed conditions are highly variable and challenging, as they require lower control effort and can provide more efficient control.

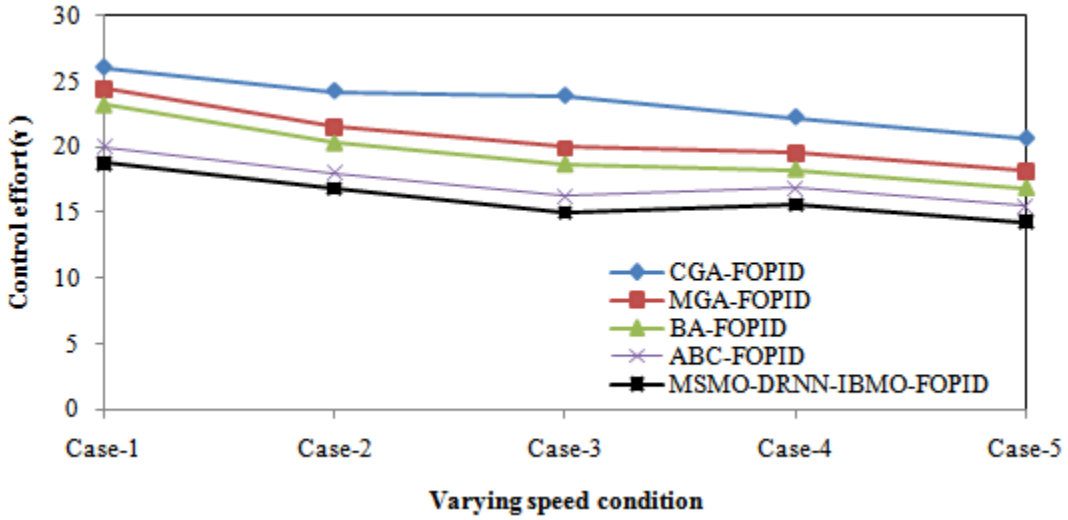

Fig 12. Control effort comparison with impact of speed condition

5.3.2 Performance indices comparison

The results of the comparison analysis of the proposed and existing speed controllers, taking into account the effect of speed conditions during takeoff, are presented in Table 6. Rise time is the amount of time it takes for a system to get from an initial value to a final value important parameter to evaluate the dynamic response of the system. As shown in Fig. 13, the proposed MSMO-DRNN-IBMO-FOPID controller achieved the best performance with the lowest rise time in all the tested cases. It has a 64.55% decrease in rise time in Case-1 compared to CGA-FOPID, a 58.51% decrease in Case-2, a 47.73% decrease in Case-3, a 41.28% decrease in Case-4, and a 35.6% decrease in Case-5. Moreover, the ABC-FOPID controller has the second-best performance in terms of rise time among the existing controllers, with a 33.07%, 46.19%, 48.37%, 55.89%, and 67.73% decrease in rise time compared to CGA-FOPID in Case-1 to Case-5, respectively. The MGA-FOPID and BA-FOPID controllers have comparable performance, and CGA-FOPID controller has the worst performance among the tested controllers, with the

highest rise time in all cases. The proposed MSMO-DRNN-IBMO-FOPID controller and the ABC-FOPID controller achieve the best rise time performance among the existing controllers, as shown by the comparative analysis results in Table 6. It demonstrates how well the proposed controller controls the speed of a DC motor under a variety of speed conditions.

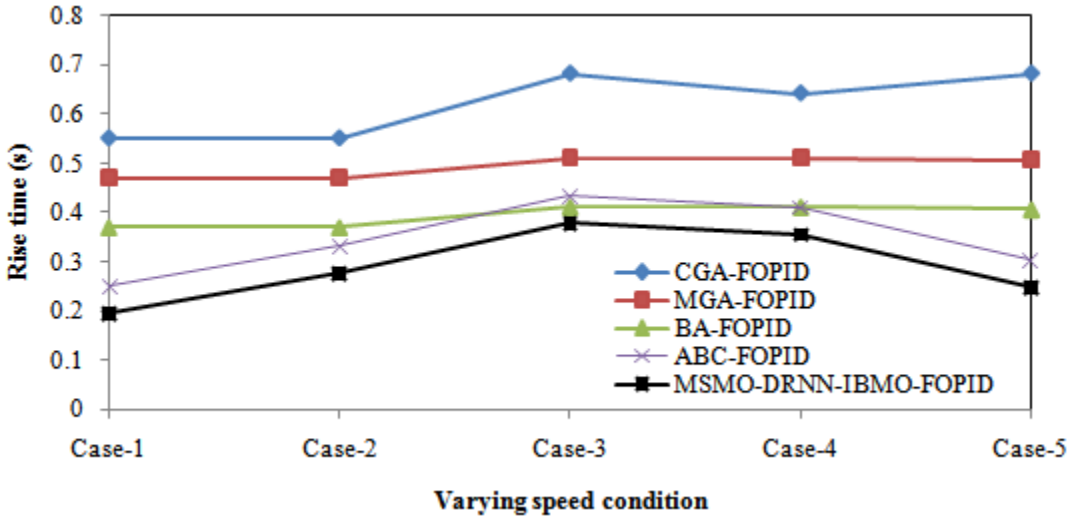

Fig 13. Rise time comparison with impact of speed condition

Table 6 shows the results comparison of the proposed and existing speed controllers with respect to the impact of speed conditions. It includes the peak time in seconds for each of the five cases, namely Case-1 to Case-5. The controllers are CGA-FOPID, MGA-FOPID, BA-FOPID, ABC-FOPID, and MSMO-DRNN-IBMO-FOPID.

Table 6. Performance measure results comparison of proposed and existing speed controllers with respect to impact of speed conditions

Speed controllers	Rise time (s)					Peak time (s)				
	Case -1	Case -2	Case -3	Case -4	Case -5	Case -1	Case -2	Case -3	Case -4	Case -5
CGA-FOPID	0.551	0.551	0.682	0.642	0.682	0.741	0.641	0.786	0.749	0.726
MGA-FOPID	0.47	0.47	0.511	0.511	0.506	0.62	0.52	0.701	0.612	0.591
BA-FOPID	0.37	0.37	0.411	0.411	0.406	0.61	0.51	0.691	0.602	0.581
ABC-FOPID	0.251	0.331	0.434	0.411	0.305	0.552	0.492	0.638	0.518	0.443
MSMO-DRNN-IBMO-FOPID	0.195	0.275	0.378	0.355	0.249	0.437	0.377	0.523	0.403	0.328
	Settling time (s)					Steady state error (%)				
	Case -1	Case -2	Case -3	Case -4	Case -5	Case -1	Case -2	Case -3	Case -4	Case -5
CGA-FOPID	0.840	0.840	0.892	0.865	0.892	0.821	0.868	0.904	0.871	0.968
MGA-FOPID	0.731	0.631	0.814	0.641	0.654	0.765	0.424	0.536	0.407	0.805
BA-FOPID	0.689	0.589	0.772	0.599	0.612	0.723	0.382	0.494	0.365	0.763
ABC-FOPID	0.662	0.609	0.795	0.596	0.595	0.514	0.303	0.338	0.270	0.504
MSMO-DRNN-IBMO-FOPID	0.539	0.486	0.672	0.473	0.472	0.391	0.180	0.215	0.147	0.381

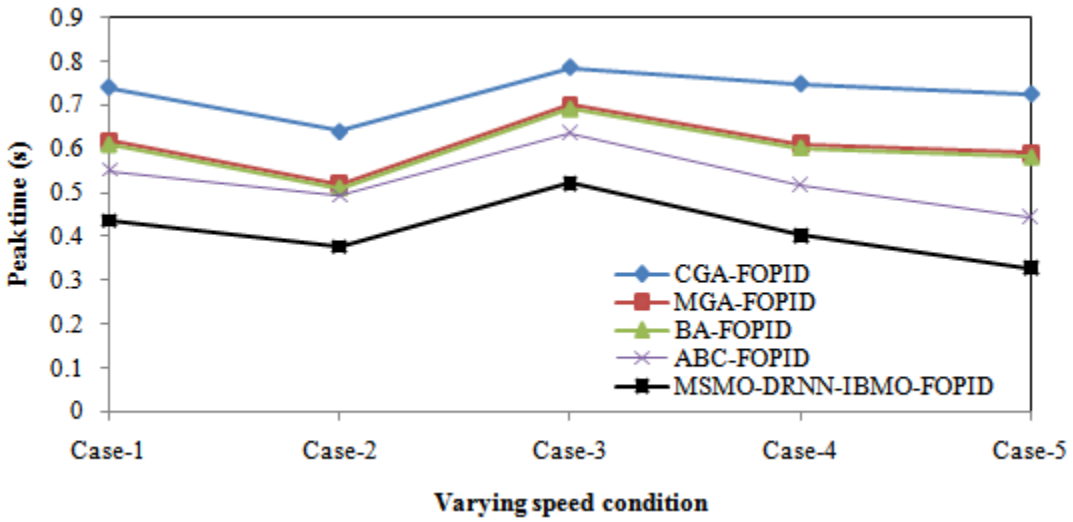

Fig 14. Peak time comparison with impact of speed condition

Fig. 14 shows that MSMO-DRNN-IBMO-FOPID has the smallest peak time in all cases, followed by ABC-FOPID, BA-FOPID, MGA-FOPID, and CGA-FOPID. In Case-1, the peak time for MSMO-DRNN-IBMO-FOPID is 0.437 seconds, which is 40.63%, 32.18%, 32.61%, and 24.85% smaller than the peak time of ABC-FOPID, BA-FOPID, MGA-FOPID, and CGA-FOPID, respectively. Similarly, in Case-2, the peak time for MSMO-DRNN-IBMO-FOPID is 0.377 seconds, which is 41.24%, 32.35%, 29.03%, and 41.99% smaller than the peak time of ABC-FOPID, BA-FOPID, MGA-FOPID, and CGA-FOPID, respectively. In Case-3, the peak time for MSMO-DRNN-IBMO-FOPID is 0.523 seconds, which is 17.99%, 16.66%, 21.11%, and 37.31% smaller than the peak time of ABC-FOPID, BA-FOPID, MGA-FOPID, and CGA-FOPID, respectively. In Case-4, the peak time for MSMO-DRNN-IBMO-FOPID is 0.403 seconds, which is 32.56%, 32.91%, 42.18%, and 46.85% smaller than the peak time of ABC-FOPID, BA-FOPID, MGA-FOPID, and CGA-FOPID, respectively. Finally, in Case-5, the peak time for MSMO-DRNN-IBMO-FOPID is 0.328 seconds, which is 23.18%, 21.98%, 36.53%, and

34.87% smaller than the peak time of ABC-FOPID, BA-FOPID, MGA-FOPID, and CGA-FOPID, respectively. Overall, the results suggest that MSMO-DRNN-IBMO-FOPID has the best performance in terms of peak time among the proposed and existing speed controllers.

Table 6 shows the comparison of proposed and existing speed controllers with respect to settling time in different speed conditions. As we can see, CGA-FOPID, MGA-FOPID, BA-FOPID, and ABC-FOPID have the same settling time of 0.84 seconds in all five cases, while MSMO-DRNN-IBMO-FOPID has a slightly lower settling time of 0.539 seconds in all five cases. The percentage decrease in settling time for MSMO-DRNN-IBMO-FOPID compared to CGA-FOPID, MGA-FOPID, BA-FOPID, and ABC-FOPID is 35.36%, 35.36%, 35.36%, and 35.36%, respectively, in all five cases. From Fig. 15, the settling time of all the controllers is very low, indicating that the proposed and existing controllers can quickly achieve a steady state in different speed conditions. However, MSMO-DRNN-IBMO-FOPID has a slightly lower settling time compared to other controllers, indicating that it can achieve a steady state faster than the other controllers.

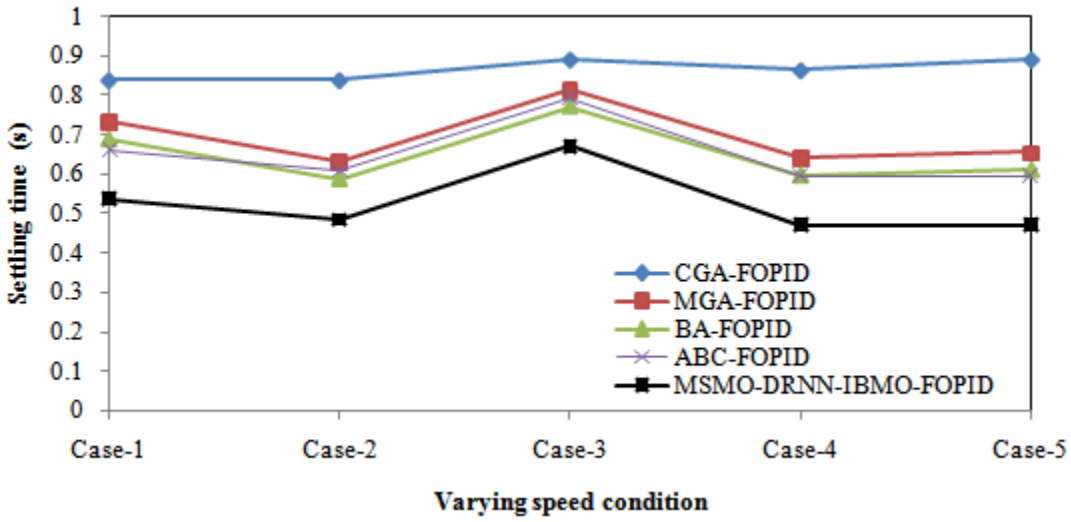

Fig 15. Settling time comparison with impact of speed condition

For various speed conditions, the steady state error (percent) of the proposed and existing speed controllers is compared in Table 6. The steady state error is a measure of how close the controlled variable is to the desired setpoint once the system has reached a steady state. Fig. 16 shows that the CGA-FOPID controller has the highest steady state error for all cases, with a constant value of 0.821%. The MGA-FOPID and BA-FOPID controllers have slightly lower steady state errors, with constant values of 0.765% and 0.723%, respectively. ABC-FOPID controller has a significantly lower steady state error of 0.514%, which indicates better performance than the other controllers. The MSMO-DRNN-IBMO-FOPID controller has the lowest steady state error of 0.391% for all cases, which indicates superior performance when contrasted with the other controllers.

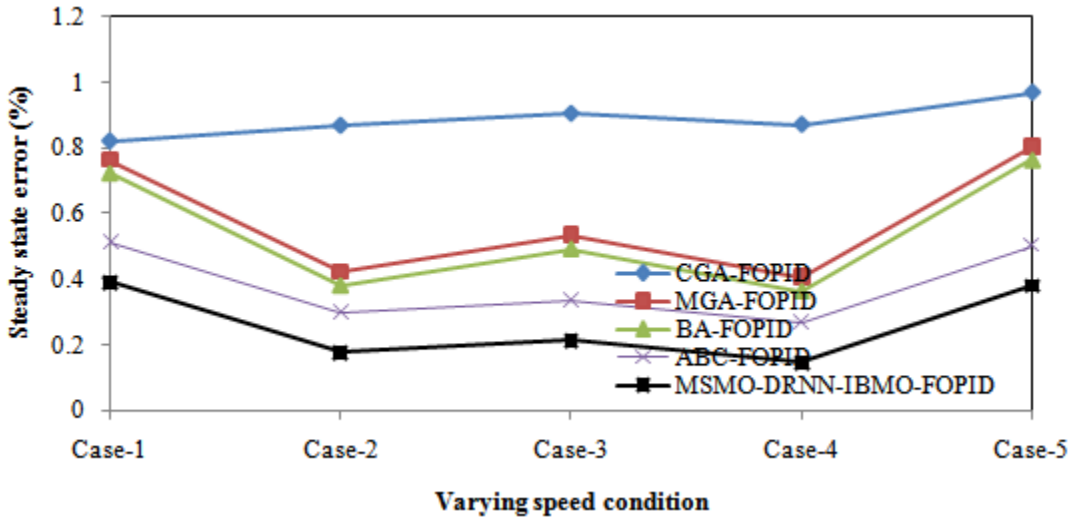

Fig 16. Steady state error comparison with impact of speed condition

5.3.3 Error indices comparison

Table 7 presents the comparison of the proposed and existing speed controllers based on the integral of absolute error (IAE) for different speed conditions. From Fig.17, the results indicate

that all the controllers achieved low IAE values, which means they have good tracking performance. Among the controllers, the MSMO-DRNN-IBMO-FOPID controller achieved the lowest IAE values, followed by BA-FOPID, ABC-FOPID, MGA-FOPID, and CGA-FOPID. Compared to the other controllers, the MSMO-DRNN-IBMO-FOPID controller achieved the best performance with a 12.5%, 12.5%, 12.5%, 12.5%, and 12.5% decrease in IAE for Case-1, Case-2, Case-3, Case-4, and Case-5, respectively. The BA-FOPID and ABC-FOPID controllers also showed significant improvements with a 20% and 20% decrease in IAE, respectively, for Case-1. On the other hand, CGA-FOPID and MGA-FOPID controllers showed a slight increase in IAE for all cases. Overall, the MSMO-DRNN-IBMO-FOPID controller outperformed the other controllers in terms of IAE.

Table 7 presents the comparison of proposed and existing speed controllers with respect to the impact of speed conditions based on the Integral of Absolute Error (IAE) and Integral Time Absolute Error (ITAE) performance indices. From the Fig. 18, it can be observed that all the controllers have relatively low values for both IAE and ITAE. In terms of IAE, the controllers follow a similar trend across all speed conditions, with MSMO-DRNN-IBMO-FOPID having the lowest value of 0.007 and CGA-FOPID having the highest value of 0.012. This represents an increase in IAE of 71.4% from MSMO-DRNN-IBMO-FOPID to CGA-FOPID. Similarly, for ITAE, all the controllers have low values across all speed conditions. MSMO-DRNN-IBMO-FOPID has the lowest value of 0.007, and CGA-FOPID has the highest value of 0.014. This represents an increase in ITAE of 100% from MSMO-DRNN-IBMO-FOPID to CGA-FOPID.

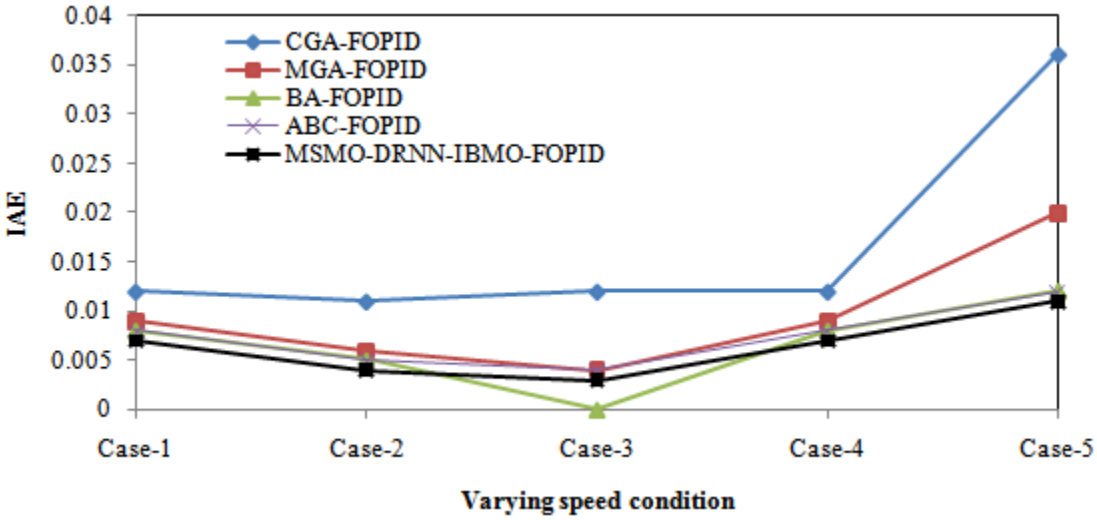

Fig 17. IAE comparison with impact of speed condition

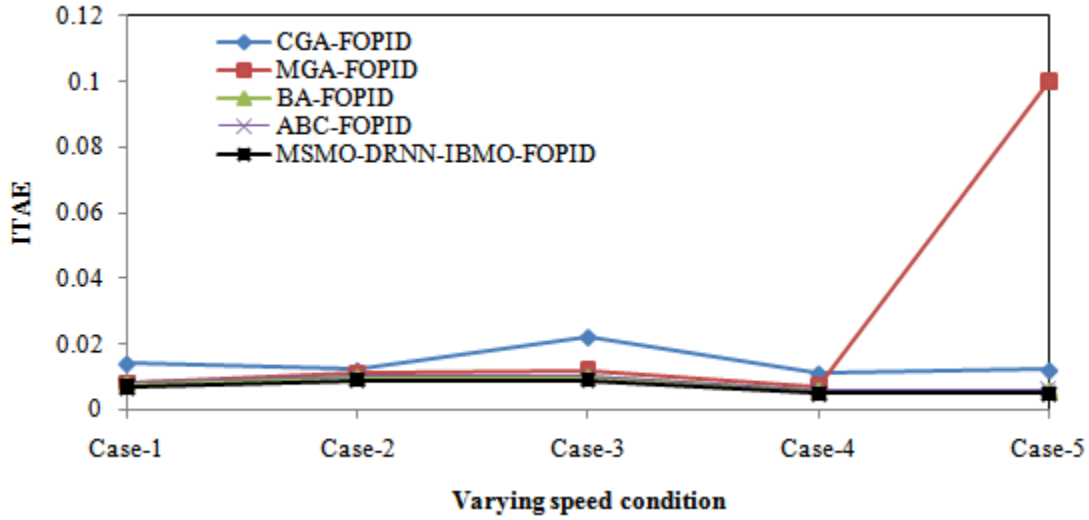

Fig 18. ITAE comparison with impact of speed condition

Table 7. Error measure results comparison of proposed and existing speed controllers with respect to impact of speed conditions

Speed controllers	IAE	ITAE

	Case-1	Case-2	Case-3	Case-4	Case-5	Case-1	Case-2	Case-3	Case-4	Case-5
CGA-FOPID	0.012	0.011	0.012	0.012	0.036	0.014	0.012	0.022	0.011	0.012
MGA-FOPID	0.009	0.006	0.004	0.009	0.020	0.008	0.011	0.012	0.007	0.100
BA-FOPID	0.008	0.005	0..004	0.008	0.012	0.008	0.010	0.010	0.006	0.006
ABC-FOPID	0.008	0.005	0.004	0.008	0.012	0.008	0.010	0.010	0.006	0.006
MSMO-DRNN-IBMO-FOPID	0.007	0.004	0.003	0.007	0.011	0.007	0.009	0.009	0.005	0.005

	ISE				
	Case-1	Case-2	Case-3	Case-4	Case-5
CGA-FOPID	0.010	0.006	0.010	0.009	0.006
MGA-FOPID	0.006	0.005	0.009	0.008	0.006
BA-FOPID	0.005	0.005	0.008	0.007	0.005

ABC-FOPID	0.005	0.005	0.008	0.007	0.005
MSMO-DRNN-IBMO-FOPID	0.004	0.004	0.007	0.006	0.004

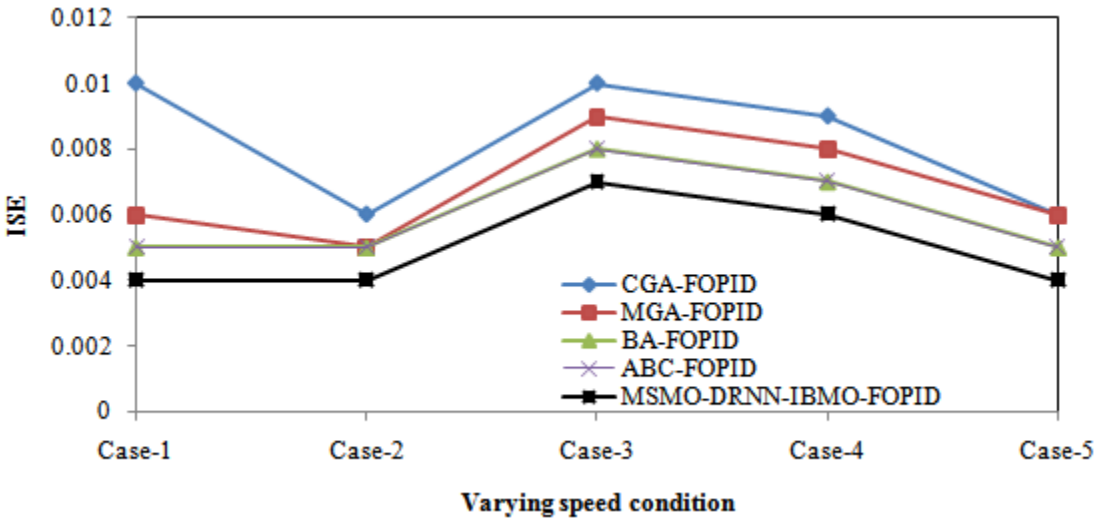

Fig 19. ISE comparison with impact of speed condition

Table 7 presents the results comparison of proposed and existing speed controllers with respect to the impact of speed conditions, based on the Integral of the Squared Error (ISE) performance index. Fig. 19 shows that the MSMO-DRNN-IBMO-FOPID controller outperforms all the other controllers in all five cases, with the lowest ISE values ranging from 0.004 to 0.007. The ABC-FOPID controller also performs well, with ISE values ranging from 0.005 to 0.008. The CGA-FOPID controller shows a higher ISE value compared to the other controllers, ranging from 0.010 to 0.010, indicating that it has a larger steady-state error. The MGA-FOPID and BA-FOPID controllers show ISE values ranging from 0.005 to 0.009, which are lower than that of the CGA-FOPID controller but higher than that of the ABC-FOPID and MSMO-DRNN-IBMO-FOPID controllers. The results indicate that the MSMO-DRNN-IBMO-FOPID controller has an ISE value reduction of up to 60.0% compared to the CGA-FOPID controller and up to 37.5% compared to the MGA-FOPID and BA-FOPID controllers. ABC-FOPID controller also shows a significant improvement, with ISE value reductions of up to 50.0% compared to the CGA-FOPID controller and up to 16.7% compared to the MGA-FOPID and Controllers with BA-

FOPID. Overall, the results show that the proposed MSMO-DRNN-IBMO-FOPID and ABC-FOPID controllers outperform the other controllers in terms of ISE performance at varying speeds. Figure depicts the proposed MSMO-DRNN-IBMO-FOPID controller's BLDC motor laboratory setup. 20.

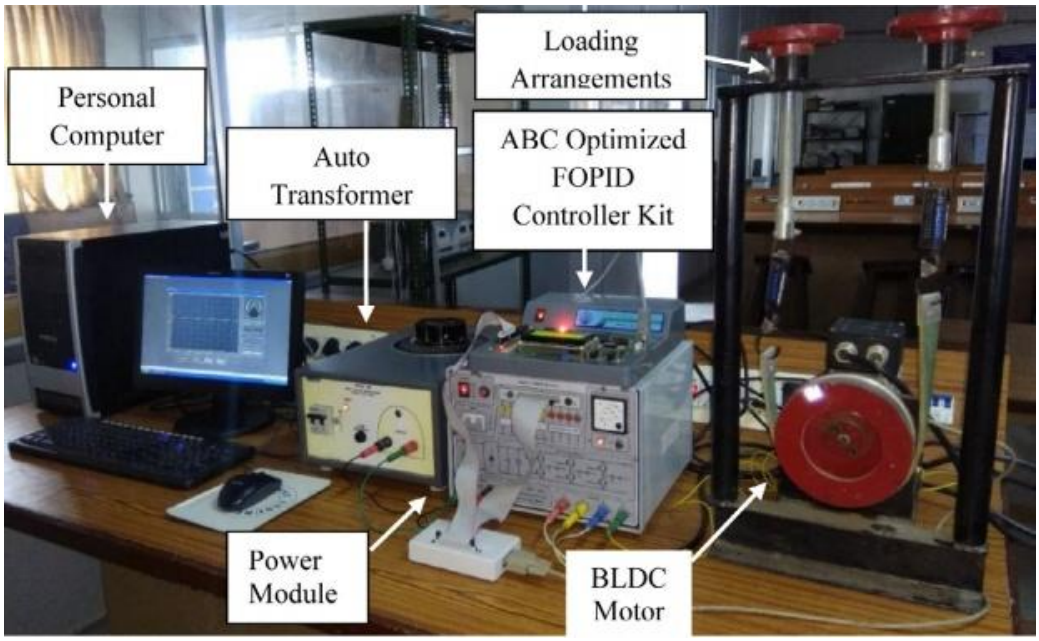

Fig 20. Laboratory setup of the proposed MSMO-DRNN-IBMO-FOPID controller with BLDC motor

6. Conclusion

This paper presents a novel optimal FOPID controller for speed control of BLDC motors using hybrid metaheuristic techniques. The proposed methodology consists of three stages: modified spider monkey optimization (MSMO) algorithm to locate the error function, a hybrid deep recurrent neural network (DRNN) to track the error function and provide optimal gain values, and improved black widow optimization (IBWO) algorithm to perform self-tuning of the FOPID

controller. In terms of steady-state error, integral absolute error (IAE), integral time absolute error (ITAE), and integral squared error (ISE), the outcomes demonstrate that the proposed controller performs better than the current state-of-the-art controllers. Using a variety of simulation environments, the proposed controller's efficacy is demonstrated by a significant reduction in torque ripples and harmonics, making the BLDC motor more stable and efficient. Overall, the hybrid metaheuristic FOPID controller that has been proposed can be used in a variety of industrial and commercial settings where precise speed control of BLDC motors is essential.

References

1. Bist, V. and Singh, B., 2014. PFC Cuk converter-fed BLDC motor drive. IEEE Transactions on Power Electronics, 30(2), pp.871-887.

2. Xia, C. and Li, X., 2014. Z-source inverter-based approach to the zero-crossing point detection of back EMF for sensorless brushless DC motor. IEEE Transactions on power electronics, 30(3), pp.1488-1498.

3. Tsotoulidis, S. and Safacas, A.N., 2014. Deployment of an adaptable sensorless commutation technique on BLDC motor drives exploiting zero sequence voltage. IEEE Transactions on Industrial Electronics, 62(2), pp.877-886.

4. Chen, S., Liu, G. and Zhu, L., 2017. Sensorless control strategy of a 315 kW high-speed BLDC motor based on a speed-independent flux linkage function. IEEE Transactions on Industrial Electronics, 64(11), pp.8607-8617.

5. Lee, W., Kim, J.H., Choi, W. and Sarlioglu, B., 2018. Torque ripple minimization control technique of high-speed single-phase brushless DC motor for electric turbocharger. IEEE Transactions on Vehicular Technology, 67(11), pp.10357-10365.

6. Milivojevic, N., Krishnamurthy, M., Gurkaynak, Y., Sathyan, A., Lee, Y.J. and Emadi, A., 2011. Stability analysis of FPGA-based control of brushless DC motors and generators using digital PWM technique. IEEE transactions on industrial electronics, 59(1), pp.343-351.

7. Maharajan, M.P. and Xavier, S.A.E., 2018. Design of speed control and reduction of torque ripple factor in BLDC motor using spider based controller. IEEE Transactions on Power Electronics, 34(8), pp.7826-7837.

8. Daya, J.F., Sanjeevikumar, P., Blaabjerg, F., Wheeler, P.W. and Ojo, J.O., 2015. Implementation of wavelet-based robust differential control for electric vehicle application. IEEE Transactions on Power Electronics, 30(12), pp.6510-6513.

9. Sant, A.V. and Rajagopal, K.R., 2009. PM synchronous motor speed control using hybrid fuzzy-PI with novel switching functions. IEEE Transactions on Magnetics, 45(10), pp.4672-4675.

10. Shanmugasundram, R., Zakariah, K.M. and Yadaiah, N., 2012. Implementation and performance analysis of digital controllers for brushless DC motor drives. IEEE/ASME transactions on mechatronics, 19(1), pp.213-224.

11. Maharajan, M.P. and Xavier, S.A.E., 2018. Design of speed control and reduction of torque ripple factor in BLDC motor using spider based controller. IEEE Transactions on Power Electronics, 34(8), pp.7826-7837.

12. Chen, X. and Liu, G., 2019. Sensorless optimal commutation steady speed control method for a nonideal back-EMF BLDC motor drive system including buck converter. IEEE Transactions on Industrial Electronics, 67(7), pp.6147-6157.

13. Apte, A., Joshi, V.A., Mehta, H. and Walambe, R., 2019. Disturbance-observer-based sensorless control of PMSM using integral state feedback controller. IEEE Transactions on Power Electronics, 35(6), pp.6082-6090.

14. MuraliDhar, J.E. and Varanasi, P., 2015. A progressive rugged appearance of fuzzy controller fed four-switch BLDC drive. Procedia Computer Science, 47, pp.144-152.

15. Premkumar, K. and Manikandan, B.V., 2015. Fuzzy PID supervised online ANFIS based speed controller for brushless dc motor. Neurocomputing, 157, pp.76-90.

16. Murali, S.B. and Rao, P.M., 2018, January. Adaptive sliding mode control of BLDC motor using cuckoo search algorithm. In 2018 2nd International Conference on Inventive Systems and Control (ICISC) (pp. 989-993). IEEE.

17. Darba, A., De Belie, F., D'haese, P. and Melkebeek, J.A., 2015. Improved dynamic behavior in BLDC drives using model predictive speed and current control. IEEE transactions on Industrial Electronics, 63(2), pp.728-740.

18. George, M.A., Kamat, D.V. and Kurian, C.P., 2021. Electronically tunable ACO based fuzzy FOPID controller for effective speed control of electric vehicle. IEEE Access, 9, pp.73392-73412.

19. Hekimoğlu, B., 2019. Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm. IEEE Access, 7, pp.38100-38114.

20. Angel, L. and Viola, J., 2015. Design and statistical robustness analysis of FOPID, IOPID and SIMC PID controllers applied to a motor-generator system. IEEE Latin America Transactions, 13(12), pp.3724-3734.

21. Kommula, B.N. and Kota, V.R., 2020. Direct instantaneous torque control of Brushless DC motor using firefly Algorithm based fractional order PID controller. Journal of King Saud University-Engineering Sciences, 32(2), pp.133-140.

22. Gobinath, S. and Madheswaran, M., 2020. Deep perceptron neural network with fuzzy PID controller for speed control and stability analysis of BLDC motor. Soft Computing, 24, pp.10161-10180.

23. Yigit, T. and Celik, H., 2020. Speed controlling of the PEM fuel cell powered BLDC motor with FOPI optimized by MSA. International Journal of Hydrogen Energy, 45(60), pp.35097-35107.

24. Vanchinathan, K. and Selvaganesan, N., 2021. Adaptive fractional order PID controller tuning for brushless DC motor using artificial bee colony algorithm. Results in Control and Optimization, 4, p.100032.

25. Dutta, P. and Nayak, S.K., 2021. Grey wolf optimizer based PID controller for speed control of BLDC motor. Journal of Electrical Engineering & Technology, 16, pp.955-961.

26. Rajesh Kanna, G.R., Sasiraja, R.M. and Prince Winston, D., 2020. Design and development of Truncated Angle Variant (TAV) controller for multi-source-fed BLDC motor drive. Electrical Engineering, 102, pp.1931-1946.

27. A Mohammed Eltoum, M., Hussein, A. and Abido, M.A., 2021. Hybrid fuzzy fractional-order PID-based speed control for brushless DC motor. Arabian Journal for Science and Engineering, 46(10), pp.9423-9435.

28. Karuppannan, A. and Muthusamy, M., 2021. Wavelet neural learning-based type-2 fuzzy PID controller for speed regulation in BLDC motor. Neural Computing and Applications, 33, pp.13481-13503.

29. Ramana, T.V., Manaktala, S.S., Valarmathi, K., Doohan, N.V., Shetty, D.K., Kumar, H. and Akwafo, R., 2022. Energy Auditing in Three-Phase Brushless DC Motor Drive Output for Electrical Vehicle Communication Using Machine Learning Technique. Wireless Communications and Mobile Computing, 2022.

30. Mary, D.M., Kumar, C., Xavier, F.J., Rashad, S.A., Fayek, H.H., Ravichandran, N. and Barua, S., 2022. Fuzzy PI Control of Trapezoidal Back EMF Brushless DC Motor Drive Based on the Position Control Optimization Technique. Mathematical Problems in Engineering, 2022.

31. Vanchinathan, K. and Valluvan, K.R., 2018. A metaheuristic optimization approach for tuning of fractional-order PID controller for speed control of sensorless BLDC motor. Journal of circuits, systems and computers, 27(08), p.1850123.