# Performance Optimization of BLDC Motor Drives Using an Adaptive Elite Particle Swarm Optimization-Tuned Sliding Mode Controller

Chun-Yao Lee[1] and Edu Daryl C. Maceren[2*]
[1]Department of Electrical Engineering
National Taiwan University of Science and Technology
Taipei City, 106 Taiwan
*edcmaceren@addu.edu.ph*

[2]Electrical Engineering Department
Ateneo de Davao University
Davao City, Davao del Sur, 8000 Philippines

Date received: November 13, 2025
Revision accepted: April 29, 2026

## Abstract

*A Brushless DC (BLDC) motor is a synchronous motor with a trapezoidal back electromotive force (EMF) waveform, which is typically operated with nonlinear control systems. This device depends on its speed controller to operate robustly. In most control applications, BLDC motor drives employ a proportional-integral-derivative (PID) controller to regulate speed; however, adjusting the controller's parameters is very challenging. Also, the PID controller may not be robust enough to maintain optimal motor performance when a disturbance occurs. Thus, a sliding mode controller (SMC) is proposed in this study, as it has been proven to achieve robust performance in the presence of external interference. This study presents a tuned SMC using an elite-based particle swarm optimization (EPSO) for the BLDC motor's control system. The aim is to determine the SMC's parameters optimally, obtain a fast speed response without overshooting during its operation, and minimize the torque ripple. In this study, three performance indicators – integral time absolute error (ITAE), integral time squared error (ITSE), and integral squared error (ISE) are used to measure the effectiveness of optimizing the SMC. The results show that the lowest torque ripple and the best speed response curve are obtained when ITSE is used as the performance indicator. Finally, the proposed control system demonstrates superior performance considering external disturbances.*

***Keywords****:* brushless DC motor, elite-based particle swarm optimization, performance indicator, proportional integral derivative, proportional integral derivative

## 1. Introduction

The BLDC motor has been widely used in various industrial applications due to its high efficiency, long lifespan, and simple structure (Shi *et al*., 2022; Xia *et al*., 2020). Recently, many studies have focused on controlling the speed of a BLDC motor. However, much of the research is currently focused on the BLDC motor's speed regulation and torque ripple minimization, using an advanced control strategy to improve its reliability in regulating speed with a fast transient response and to expand its application range.

It is challenging to achieve high-precision control of a BLDC motor using a traditional PID control algorithm under high-speed conditions with a sudden change in load torque. Therefore, many researchers are finding ways to improve the PID's precision range. Many studies have been done about the fuzzy logic-PID controller (Chao *et al.*, 2019; Kim & Han, 2020; Pan & Lin, 2021; Xu & Wang, 2017); however, experimental data are required to design the controller. Also, fuzzy control systems require memory (Ma'arif & Çakan, 2021). Nonetheless, the effectiveness of the PID control algorithm depends primarily on the model's accuracy, which is susceptible to parameter changes and external disturbances during its operation (Wang *et al.*, 2020). The SMC has become an emerging research topic because it has low requirements for model accuracy and has proven robust performance in dealing with external and internal disturbances when applied to a BLDC motor for speed regulation (Wang *et al.*, 2020).

Over the past few years, various PSO algorithms have been proposed for tuning PID controllers. In the study conducted by Tang *et al*. (2025), an improved PSO algorithm was proposed to determine the optimal PID controller coefficients to achieve high trajectory-tracking accuracy. Hong *et al*. developed a novel adaptive elite-based PSO to minimize the active power loss in power systems and provide an optimal voltage profile (Hong *et al.*, 2014). Shima and Bashir (2021) employ PSO to search for optimal parameter values of the integral sliding mode controller to improve settling time and reduce overshoot when controlling the cart-inverted pendulum system (Hong *et al.*, 2014; Shima & Bashir, 2021). A particle swarm algorithm-tuned fuzzy logic controller is presented in the study of Housny *et al*. (2019). As the controller has been applied to the closed-loop system of a quadrotor, the results show conclusive effectiveness in improving the speed control of an unmanned aerial vehicle. Malla *et al*. (2022) have proven that PSO-tuned SMC improves BLDC for a water pumping station.

This article employs an SMC with optimally tuned parameters to enhance the dynamic performance of BLDC motor speed regulation and minimize torque ripple. An elite-based PSO approach is introduced as a reliable method for determining the optimal SMC parameters. When these tuned parameters are applied to the closed-loop control system of a BLDC motor, the scheme's effectiveness is verified under load disturbance conditions.

### *1.1 Mathematical modeling*

#### 1.1.1 BLDC motor

The derivation of the phase winding voltage equations is based on the circuit diagram shown in Figure 1, where *V* is set as the input DC voltage, *I* as the input current, E as the induced winding counter-electromotive force (EMF), R as the winding resistance, L as the inductance, ω as the angular speed, and J as the motor's shaft inertia.

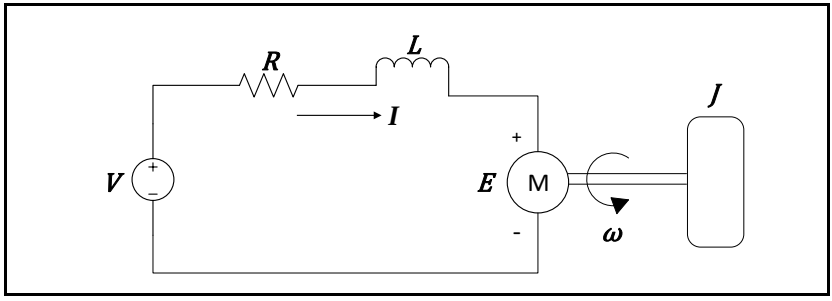


Figure 1. DC Motor diagram

Using Kirchhoff's voltage law, the winding voltage is obtained as Equation 1:

$$RI + L\frac{dI}{dt} + K_e\omega = V \tag{1}$$

Where:

$V$ = applied phase voltage (V)
$K_e$ = back-electromotive force constant
$R$ = stator resistance (Ω)
$L$ = phase inductance (H)
$I$ = phase current (A)
$\omega$ = angular speed (radians per second)

The mechanical equation for the motor's rotation is obtained as Equation 2:

$$J\dot{\omega} + K_f\omega = K_t I \tag{2}$$

Where:

$K_f$ = friction constant
$K_t$ = torque constant
$J$ = rotor moment of inertia (kg·m²)
$\dot{\omega}$ = angular acceleration (radians per second per second)

The state variables, the input, and the output can be defined as Equations 3, 4, and 5:

$$x_1 = \omega \tag{3}$$

$$x_2 = I \tag{4}$$

$$u = V. \tag{5}$$

Thus, the state space equations of the BLDC motor using Equation 1 and 2 can be expressed as Equations 6 and 7:

$$\dot{x}_1 = \dot{\omega} = \frac{d\theta}{dt} = -\frac{K_f}{J}\omega + \frac{K_t}{J}I \tag{6}$$

$$\dot{x}_2 = \dot{I} = \frac{dI}{dt} = -\frac{K_e}{L}\omega + \frac{R}{L}I + \frac{1}{L}V. \tag{7}$$

The matrix form of the space equations is expressed as Equation 8:

$$\dot{x} = \begin{bmatrix}\dot{x_1}\\ \dot{x_2}\end{bmatrix}; A = \begin{bmatrix}-\frac{K_f}{J} & \frac{K_t}{J}\\ -\frac{K_e}{L} & -\frac{R}{L}\end{bmatrix}; B = \begin{bmatrix}0\\ \frac{1}{L}\end{bmatrix}; C = [1 \quad 0] \tag{8}$$

Using the motor's parameter values in Table 1, the state space equations from Equation 8, and a Matlab function (*ALFIAN*), the transfer function of the BLDC motor is obtained as Equation 9:

$$\omega(s) = \frac{1150}{s^2+339.5s+1294} \times V(s) \tag{9}$$

Table 1. Implementation parameters of BLDC motor controller design

| Parameter | Value |
|---|---|
| Rated voltage (DC) | 500 V |
| Rated Power | 100 W |
| Rated Speed | 3000 RPM |
| Stator phase resistance | 2.875 Ω |
| Stator phase inductance | 8.5 mH |
| Flux linkage | 0.175 |
| Back EMF flat area | 120° |
| Inertia | 0.0008 kg-m$^2$ |
| Pole pairs | 4 |
| Friction coefficient | 0.001 N-m-s |

The control diagram of the conventional PID controller is shown in Figure 2. The controller has the capability to process the difference between the desired and actual angular speed of the motor; the control error equation of the PID is expressed as Equation 10:

$$PID = K_p + K_i \int_0^t e(t)dt + K_d * \frac{de(t)}{dt} \quad (10)$$

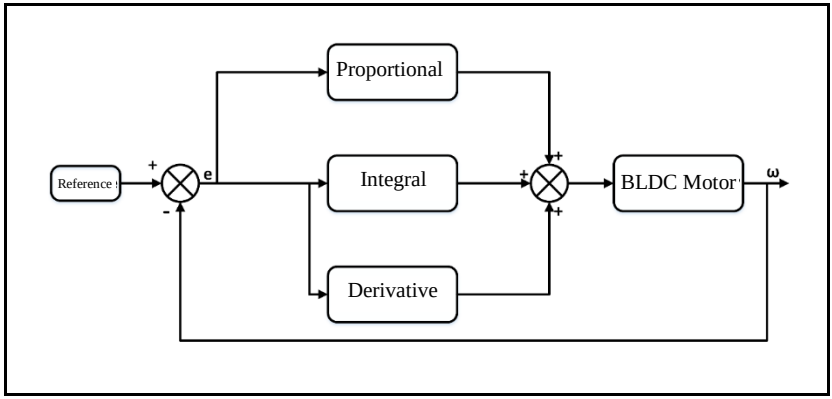


Figure 2. Conventional PID controller structure

Where:

$K_p$ = PID proportional constant
$K_i$ = PID integral constant
$K_d$ = PID derivative constant

Initially, the PID controller was designed using the Ziegler-Nichols method, which tunes the controller to maintain stable motor control. Using the Matlab toolbox for tuning a PID controller, the values of $K_p$, $K_i$, and $K_d$ are 0.592356,

6.632370, and -0.003010, respectively. Since it is practically impossible to attain an ideal PID controller (Malla *et al*., 2022), especially when a disturbance occurs in the system, an advanced controller and optimization method can be employed to achieve more robust and reliable performance.

1.2.2 Sliding mode controller design

The sliding mode controller's (SMC) design is based on the derived transfer function in Equation 9. According to Ma'arif and Çakan (2021), the transfer function model of the BLDC can be written as Equation 11:

$$\ddot{\omega} + 339.5\dot{\omega} + 1294\omega = 1150 \times V \tag{11}$$

and expressed as Equation 12:

$$\ddot{\omega} = -339.5\dot{\omega} - 1294\omega + 1150 \times V. \tag{12}$$

The typical sliding-mode surface is expressed as Equation 13

$$s = K_1 e + \dot{e} \tag{13}$$

where $e$ is the tracking error and the variable $K_1$ relates the asymptotic convergence rate and stability of the SMC based on the Hurwitz condition ($K_1 > 0$). Also, the tracking error and its derivatives are expressed as Equations 14, 15, and 16:

$$e = \omega r - \omega \tag{14}$$

$$\dot{e} = \omega \dot{r} - \dot{\omega} \tag{15}$$

$$\ddot{e} = \omega \ddot{r} - \ddot{\omega} \tag{16}$$

where $\omega r$ is the reference angular speed, and $\omega$ is the actual angular speed of the BLDC motor.

The stability condition of the SMC can be analyzed based on the Lyapunov function, which is defined as Equation 17:

$$P = \frac{1}{2} s^2 \tag{17}$$

To attain the stability of the controller, the first derivative of Equation 12 must be negative ($\dot{P} < 0$), and can be expressed as Equation 18:

$$s\dot{s} < 0. \tag{18}$$

Taking the first derivative of Equation 11, we have Equation 19:

$$\dot{s} = K_1\dot{e} + \ddot{e} \tag{19}$$

Substituting Equation 13 and Equation 15 in Equation 19 gives Equation 20:

$$\dot{s} = K_1(\dot{\omega}_r - \dot{\omega}) + \ddot{\omega}_r - \ddot{\omega} \tag{20}$$

Then substitute Equation 11 in Equation 20, which gives Equation 21:

$$\dot{s} = K_1\dot{e} + \ddot{\omega}_r + 339.5\dot{\omega} + 1294\omega - 1150 \times V. \tag{21}$$

Based on the literature from Ma'arif and Çakan (2021), to attain the stability condition in (18), the expression for the sliding mode control from Equation 21 can be designed as Equations 22 and 23:

$$V = \frac{1}{1150}(K_1(\dot{\omega}_r - \dot{\omega}) + \ddot{\omega}_r + 339.5\dot{\omega} + 1294\omega + K_2 sign(s)). \tag{22}$$

$$sign(s) = \begin{cases} 1, & s > 0 \\ 0, & s = 0 \\ -1, & s < 0 \end{cases} \tag{23}$$

Since the term with a constant coefficient $K_2$ ($K_2 > 0$) and the signum function can cause a considerable chattering effect; thus, the signum function can be expressed as Equation 24:

$$sign(s) = \frac{s}{s+K_3} \tag{24}$$

Finally, the control function is written as Equation 25:

$$V = \frac{1}{1150}\left(K_1(\dot{\omega}_r - \dot{\omega}) + \ddot{\omega}_r + 339.5\dot{\omega} + 1294\omega + K_2\frac{s}{s+K_3}\right). \tag{25}$$

where $K_1 > 0$, $K_2 > 0$, and $0 < K_3 < 1$ is the system state with the control diagram shown in Figure 3. The parameter values of $K_1$, $K_2$, and $K_3$ will be

determined using an advanced optimization scheme to implement a more robust control scheme that provides a faster response with minimal steady-state error, as shown in Figure 4.

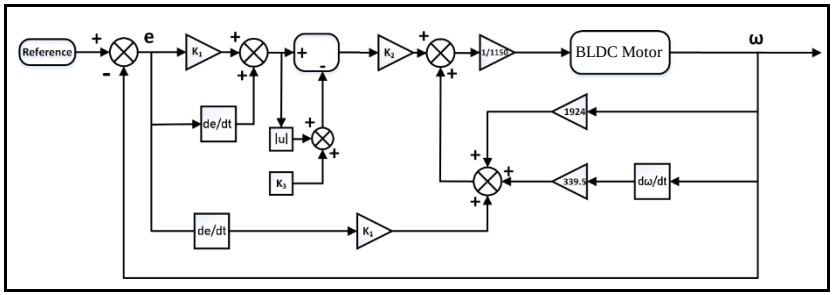


Figure 3. Control block diagram of the speed-regulation system

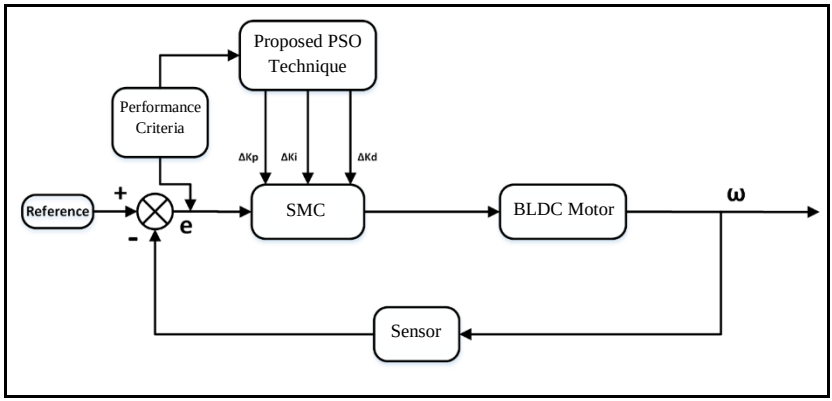


Figure 4. EPSO-SMC structure

## 2. Methodology

### *2.1 Traditional Particle Swarm Optimization*

The Traditional Particle Swarm Optimization (TPSO) Algorithm is a heuristic method for obtaining the best solution to a particular objective function. The TPSO contains a population of particles, with each particle representing a possible solution, which are iteratively updated by continuously moving within a bounded search space (Hong *et al.*, 2014). The position and velocity of each particle p are updated based on Equations 26 and 27:

$$v^{t+1} = w^t \times v_p^t + C_1 \times r_1^t \times \left(P_{best}^t - X_p^t\right) + C_2 \times r_2^t (G_{best}^t - X_p^t) \quad (26)$$

$$X_p^{t+1} = X_p^t + v_p^{t+1} \quad (27)$$

Where:

$w^t$ = inertia weight at iteration $t$
$v_p^t$ = the particle velocity during iteration $t$
$v^{t+1}$ = is the updated velocity of particle
$w^t$ = is the inertia weight at iteration $t$
$G_{best}^t$ = global best position among all particles
$P_{best}^t$ = personal best position of particle
$C_1$ = cognitive coefficient
$C_2$ = social acceleration coefficient
$r_1^t, r_2^t, r_3^t$ = uniformly distributed random numbers in the range from 0 to 1

In this paper, the notation $t$ is the iteration index. The vectors $X_p^t$ and $v_p^t$ are the position and velocity of each particle $p$ in a population within a bounded search space. The inertia $w$ is a key factor in the algorithm, controlling convergence behavior by copying previously updated features to the next iteration (Hong *et al.*, 2014; Maurya *et al.*, 2019). The constants $C_1$ and $C_2$ are called the acceleration coefficients (Zhan *et al.*, 2009), $r_1^t$ and $r_2^t$ are random numbers (from 0 to 1). The $p_{best}^t$and $g_{best}^t$ They are called the best particle and the global best, respectively. Also, the $p_{best}^t$ denotes the position of the particle with the best solution for the fitness function, while the $g_{best}^t$ is the best position in each search space. Furthermore, the second and third terms of Equation 25 are designated for global search exploration and local search exploitation, respectively, in each search space.

However, the value of the inertia weight may decrease as the number of iterations increases; it also affects the behavior of the particles' changes in their current positions and velocities. Since a large value for *w* is assigned for global search, while a small *w* value can be used for local search. Thus, a large *w* value is used during the initial stage, then a smaller *w* value during the latter part of the process (Hong *et al.*, 2014).

### *2.2 Elite-based Particle Swarm Optimization*

This study employs an adaptive elite-based PSO (EPSO) to tune the sliding mode controller. The proposed adaptive-elite PSO performs the following

main tasks: inertia weight calculation, particle mean search, pruning, and cloning of particles (Hong *et al.*, 2014). The expected outcome of these tasks has been proven to ensure the stability of the iterative process.

Initially, since there is a need to determine the optimum value of *w* instead of setting a constant random number, it would be beneficial to use a sigmoid mapping equation (Zhan *et al.*, 2009). Thus, the equation of the inertia weight in the first term in Equation 25 can be modified as Equations 28 and 29:

$$w^t = \frac{1}{1+1.5e^{-1.5f}}\left(\frac{t}{T}\right) \in [0.9, 1.2] \; \forall f \in [0,1] \tag{28}$$

$$f = \frac{G_{best}^t - X_p^t}{P_{best}^t - X_p^t} \tag{29}$$

where $t$ is the iteration index, and $T$ is the declared maximum number of iterations. However, in the study conducted by Zhan *et al*. (2009), the range of inertia weight can be set from 0.4 to 0.9 since it has been proven that these values can provide excellent results. The adaptive mechanism of adaptive PSO can be further improved by using a set of strategies to manipulate the values of $C_1$ and $C_2$ in Equation 26.

Secondly, the mean of the particles and the standard deviation between any two particles during the iterative process are then computed, and the expression for the particle's velocity using EPSO is modified as Equation 30:

$$v^{t+1} = w^t \times v_p^t + C_1^t \times r_1^t \times \left(P_{best}^t - X_p^t\right) + C_2^t \times r_2^t\left(G_{best}^t - X_p^t\right) + C_3^t r_3^t (G_{best}^t - X_m^t) \tag{30}$$

where, $C_3$ is a new learning factor that should not be greater than four and is defined by Caponetto *et al.* (2003) as Equation 31 and 32:

$$C_3^t = 4 - C_1^t - C_2^t \tag{31}$$

$$C_1^t = C_2^t = 1 + \frac{1}{1+1.5e^{1.5\alpha}} \tag{32}$$

where $\alpha = \frac{G_{best}^1}{G_{best}^t}$. The convergence of the iterative process can be guaranteed by using Equations 30 and 31 since $C_3$ will gradually reduce to zero. Thus, Equation 29 will be reduced to Equation 25. However, the global and local searches can be coordinated using the last term in Equation 30.

Lastly, pruning can be used to eliminate distant particles within the local search. The standard deviation, $\sigma^t$ of all distances between pairs of all particles are evaluated to determine the range of the particle's position during the pruning process. Then, assign the number of particles that will be pruned using Equation 33:

$$n^t = 3 - 2C_3^t. \tag{33}$$

The values of $C_3^t$t will decrease approximately from 1.48 to 0, while the values of $n^t$ will increase from 0.04 to 3 (Hong *et al.*, 2014). Then, the pruning process can be executed by eliminating the particles with a distance outside the range equation, $X_p^{t+1} \pm n^t\sigma^t$. Finally, the pruned particles will be replaced by cloning the global best, that is, Equation 34.

$$X_p^{t+1} = G_{best}^t + V_p^{t+1}. \tag{34}$$

The steps of an adaptive elite-based PSO are as follows:

Step 1. Declare the input parameters, namely, particle swarm population size, number of variables, fitness function, dimensions, and factors values of $w$, $C_1$, and $C_2$.
Step 2. Initiate the iteration with the random position of the particles.
Step 3. Calculate the fitness values for all particles using the preferred objective function by iteratively updating the position and speed of the particle.
Step 4. Stop the process if the required solution is attained.
Step 5. Determine the $P_{best}^t$ and $G_{best}$ among all particles based on the objective values of all particles.
Step 6. Compute $C_1^t$, $C_2^t$, and $C_3^t$ using Equation 31 and 30, respectively.
Step 7. Determine $X_m^t$, $\sigma^t$, and $n^t$.
Step 8. Prune a particle if its distance is greater than the range equation, $X_p^{t+1} \pm n^t\sigma^t$. Then, clone the pruned particle using $X_p^t = X_p'^t = G_{best}^t$.
Step 9. Update the particles' velocities and positions using Equation 30 and 27, respectively.
Step 10. Let $t = t + 1$ and proceed to Step 3.

According to Storn & Price (1997), the convergence of an adaptive elite-based PSO can be guaranteed. Therefore, the proposed method can be a reliable process for searching the optimal parameter values of the SMC. The flowchart

of utilizing EPSO to determine the optimal values of the SMC is shown in Figure 5.

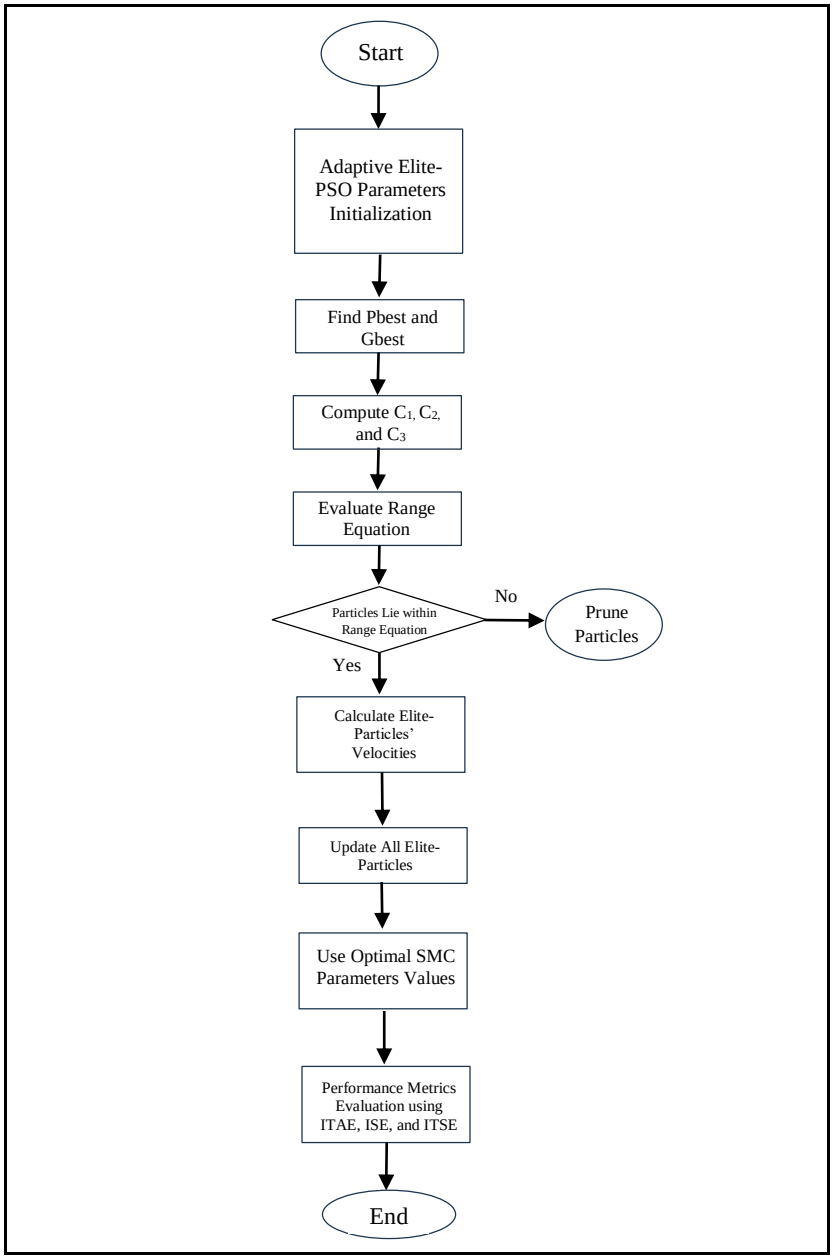


Figure 5. EPSO flowchart for SMC

## 3. Results and Discussion

### *3.1 BLDC Motor Modeling Based on SMC*

The simulation of the BLDC with a sliding mode controller model is implemented using the Matlab/Simulink environment. The control system mainly consists of a three-phase inverter, a BLDC motor, a Hall sensor, a PWM module, and a programmed control module, like the SMC.

Initially, the motor is operated with a 2-Nm mechanical load at 1500 RPM for 0.5 sec; then, the motor's speed is increased to 3000 RPM from 0.5 to 1 sec., as well as the load from 2 Nm to 6 Nm.

The speed error will be evaluated as the difference between the references and measured motor speeds. The SMC will process the speed error and provide the appropriate voltage magnitude to the DC supply for the required speed and load torque. In the commutation mechanism, the decoder block processes the hall sensor signals and then provides the switching signal to the gate pulse block, as shown in Figure 6.

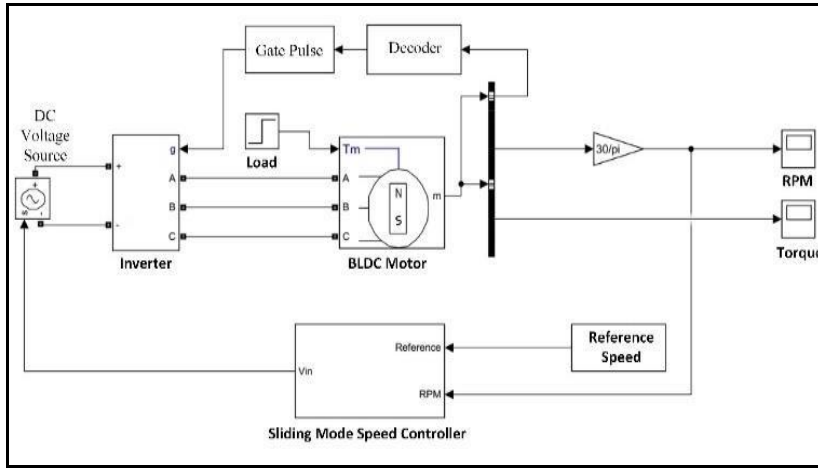


Figure 6. Block diagram of BLDC motor speed-regulation system

### *3.2 Performance Indicators Selection (Step-response Analysis)*

The sliding mode controller that controls the BLDC motor's rotational speed will be tuned with its parameters, namely $K_1$, $K_2$, and $K_3$ using the proposed

Elite-based PSO algorithm (EPSO) method. The fitness function performance indicators used to optimize the system's performance are ITAE, ISE, and ITSE. Also, the three metrics are selected to verify the proposed elite-based PSO's feasibility (Equations 35, 36, 37).

$$ITAE = \int_0^\infty t \cdot |e(t)| dt \tag{35}$$

$$ITSE = \int_0^\infty t \cdot e(t)^2 dt \tag{36}$$

$$ISE = \int_0^\infty e(t)^2 dt \tag{37}$$

Where:

$e(t)$ = error signal at time $t$

The ITAE, ITSE, and ISE are performance indicators widely used in control system optimization for their sensitivity to different characteristics of the system response. The ISE emphasizes reducing large errors but can result in longer settling times, as it heavily penalizes magnitude regardless of timing. The ITAE places more weight on errors that persist longer, encouraging faster settling, but may allow higher initial overshoot. The ITSE penalizes large errors that occur later in the response more strongly than either ISE or ITAE, making it particularly effective for improving both transient and steady-state performance.

In this study, three performance indicators were used to tune the SMC via EPSO over 50 iterations. The SMC optimization curves and the respective fitness function values are obtained. The system's step response is then evaluated to determine the best performance indicators; the resulting step response time is shown in Table 2. The optimization of the fitness function using ISE, ITAE, and ITSE is shown in Figures 7-10, respectively. Also, the step-response curves for the three performance indicators are shown in Figure 11.

Table 2. Step response information

| Parameter | ITAE | ITSE | ISE |
|---|---|---|---|
| Rise Time (sec) | 0.0237 | 0.0139 | 0.0158 |
| Settling Time (sec) | 0.0318 | 0.0250 | 0.0296 |

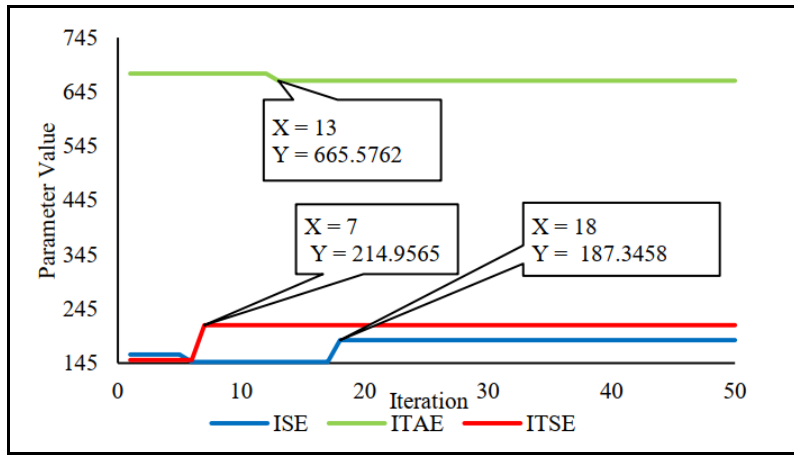


Figure 7. Optimization of $K_1$ parameter using different metrics

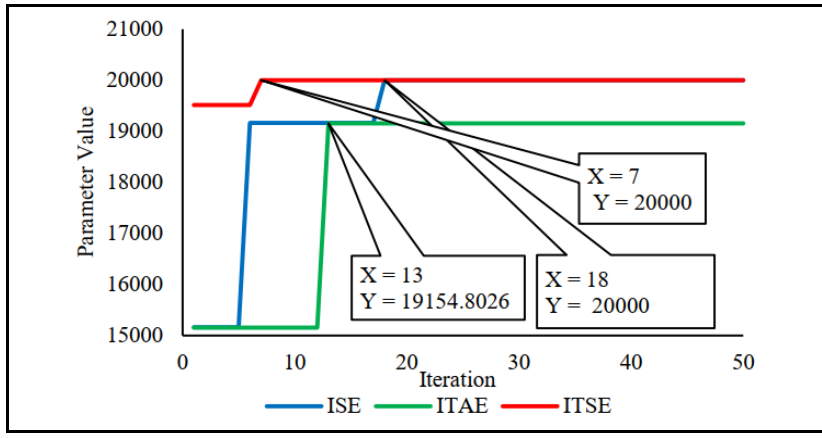


Figure 8. Optimization of $K_2$ parameter using different metrics

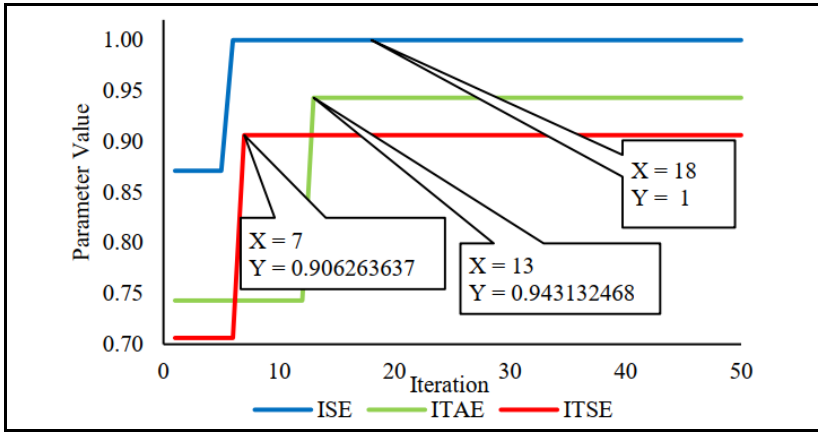


Figure 9. Optimization of $K_3$ parameter using different metrics

These performance indicators are commonly used in control system optimization because they capture different aspects of the transient response. In this study, each metric was tested separately with equal priority in the optimization process. The ITSE demonstrated the most favorable balance between rise time, settling time, and overshoot, as shown in Table 2 and Figure 11, and was therefore selected as the preferred performance indicator for tuning the SMC. The ITSE can effectively penalize large errors due to the squared-error term and emphasizes performance at later stages. These attributes make ITSE suitable for refining the controller's stability and smoothness over time.

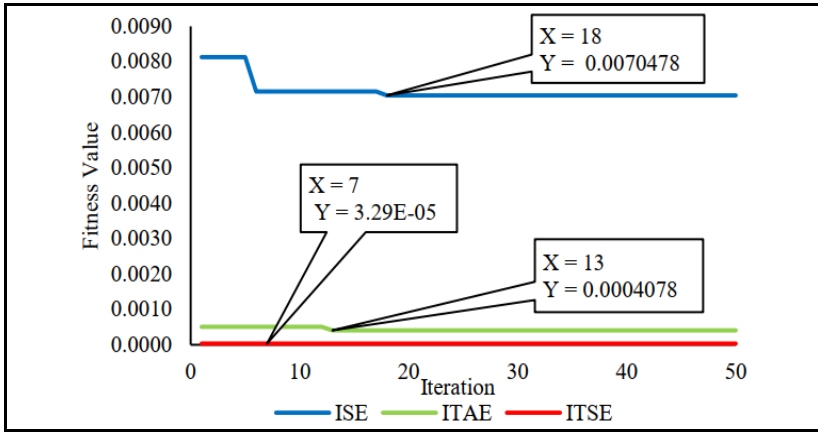


Figure 10. Error minimization curve using different metrics

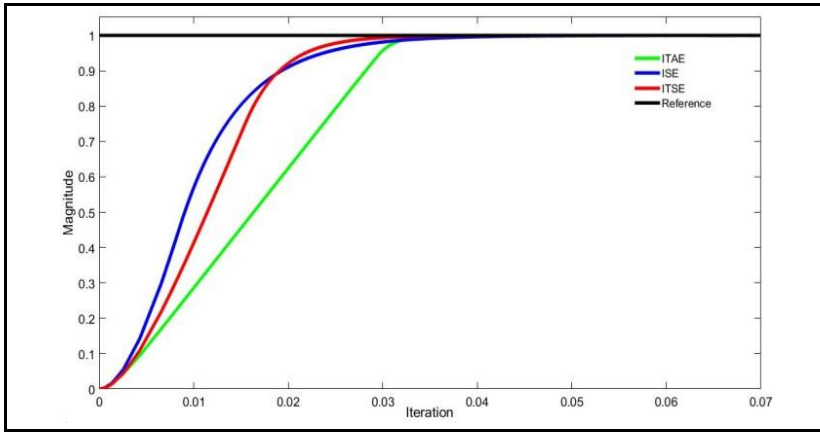


Figure 11. Step response of the speed-regulation system

The optimal SMC parameters and fitness values for EPSO, based on the three performance indicators, are obtained when the iteration convergence is at 18, 13, and 7 for ISE, ITAE, and ITSE, respectively. The ITSE attains the lowest fitness values among its counterparts. Also, it achieves the lowest step response time and fitness value, indicating it can provide the fastest optimization for the SMC. Thus, after obtaining the step response time of the SMC-driven BLDC motor, the ITSE performance indicator is selected and used as the fitness function for optimizing the SMC for the BLDC motor.

### *3.3 Comparison of TPSO and EPSO using ITSE*

The performance of the proposed EPSO is verified through a comparison with TPSO. Since the ITSE is the chosen fitness function for tuning the SMC, the EPSO will be compared with the TPSO using the parameters in Table 3 over 100 iterations. The optimization results are shown in Figures 12, 13, 14, and 15. The results show that using TPSO, the value for SMC's parameters $K_1$, $K_2$, and $K_3$ are 182.903948, 17875.6744, and 0.684843372, respectively, with a fitness value of $2.42 \times 10^{-6}$. However, the values obtained using EPSO for $K_1$, $K_2$, and $K_3$ are 217.244, 2000, and 1, respectively, with a fitness value of $6.25 \times 10^{-6}$.

Table 3. Implementation Parameters of TPSO

| Parameter | Value |
|---|---|
| Swarm Size | 15 |
| Number of iterations | 100 |
| Dimension | 3 |
| $w$ | 0.7 |
| $C1$ | 0.9 |
| $C2$ | 0.4 |

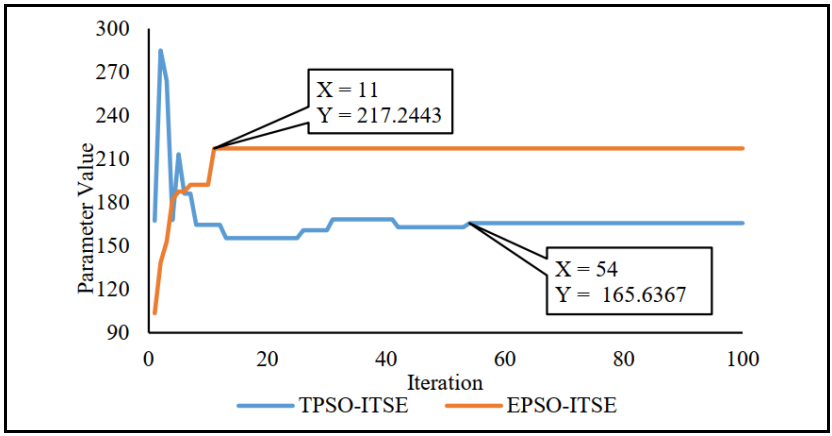


Figure 12. Optimization of $K_1$ using TPSO and EPSO

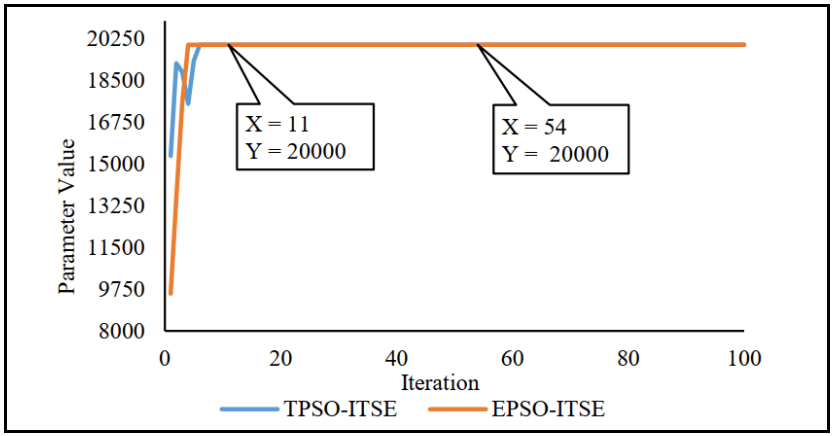


Figure 13. Optimization of $K_2$ parameter using TPSO and EPSO

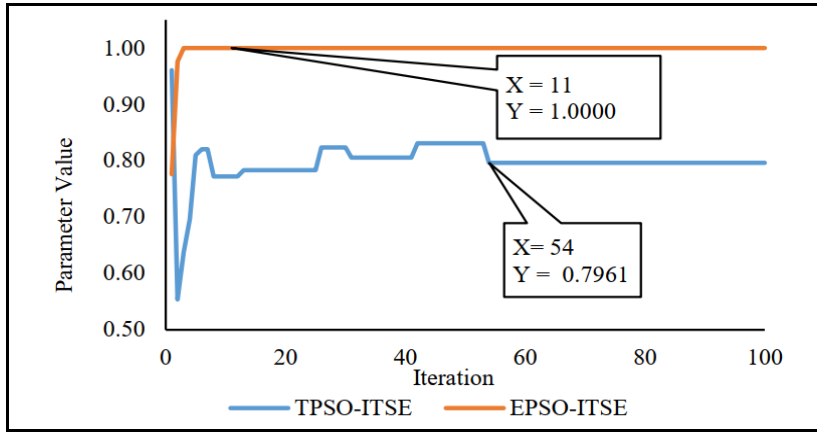


Figure 14. Optimization of $K_3$ parameter using TPSO and EPSO

The EPSO achieves the quickest convergence compared to the TPSO, since the proposed method has 11 iterations, whereas its counterpart has 54 iterations. Also, EPSO's optimization performance is more stable than TPSO's. Regarding computational complexity, the EPSO introduces additional steps such as particle mean search, pruning, and cloning operations, which slightly increase per-iteration overhead compared to traditional PSO. However, this added complexity can enable the EPSO to have fewer iterations required for convergence. On average, EPSO converged within 11 iterations, while TPSO required approximately 54 iterations to achieve comparable results. This result shows that, despite the slightly higher per-iteration computational load, EPSO achieves faster overall convergence and improved computational efficiency. Specifically, the EPSO converged within 11 iterations, while TPSO required 54 iterations to achieve comparable results. This difference translates to approximately a 79.6% reduction in the number of iterations required to converge. Efficiency gain demonstrates EPSO's ability to expedite convergence by avoiding redundant or ineffective searches through its pruning and cloning mechanisms.

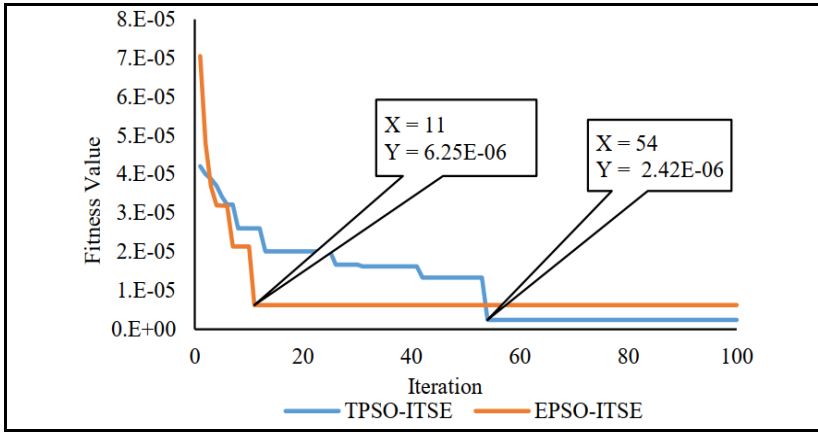


Figure 15. Error minimization using TPSO and EPSO

When using ITSE as the fitness function, the optimization results obtained from EPSO converge slowly at the start but quickly by the 11$^{th}$ iteration. Also, since the EPSO produces the convergence curves without fluctuations, thus, the proposed method proves that the two main tasks have been executed successfully: mean search and particle pruning/cloning. The mean search can ensure stable iteration, while particle pruning/cloning prevents unnecessary searches. The effect of the two main tasks has greatly improved the iteration process.

On the contrary, the TPSO yields more iterations and greater fluctuations during the optimization process; however, it achieves a better fitness value than EPSO. Although the TPSO has the lowest fitness value, it requires more iterations to find the optimal SMC parameter values. Reducing the number of iterations implies that EPSO is computationally efficient by avoiding ineffective searches. Lastly, the EPSO can offer a stable solution search to obtain solutions of almost equal quality to TPSO. The acquired value of $K_1$, $K_2$, and $K_3$will then be used to design a sliding-mode controller for a BLDC motor that provides robust performance.

### *3.4 Controller Performance Testing and Analysis (Torque and Speed Responses)*

The performance of the BLDC motor with an EPSO-tuned SMC is compared with that of a TPSO-tuned SMC and a Ziegler-Nichols-tuned PID controller. The three control systems are simulated under different load conditions and at different speeds. Figures 16, 17, 18, and 19 show the speed response of different load conditions.

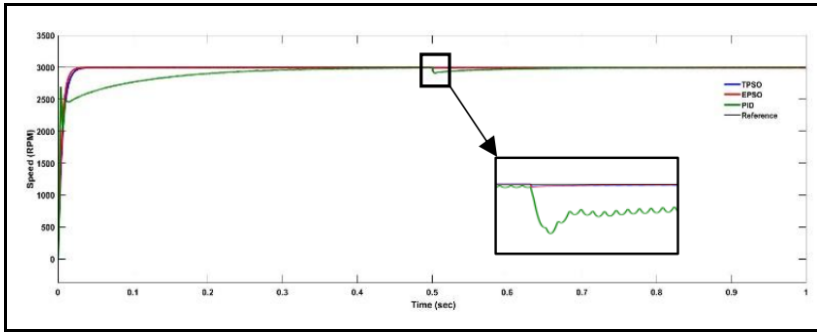


Figure 16. Speed response curves with sudden change of load under different speed conditions (0 – 3000 RPM)

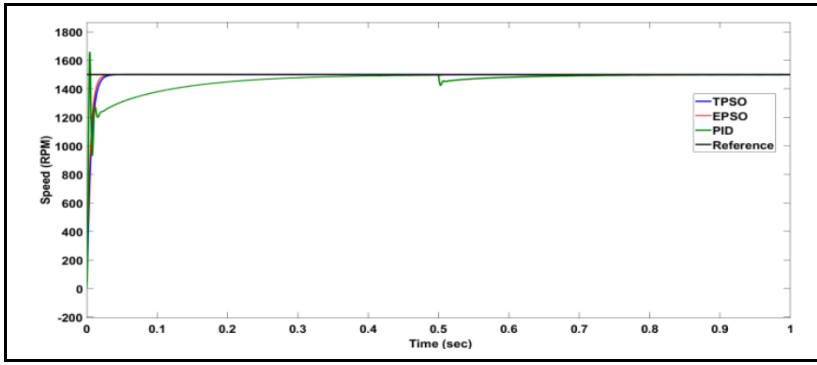


Figure 17. Speed response curves with sudden change of load under different speed conditions (0 – 1500 RPM)

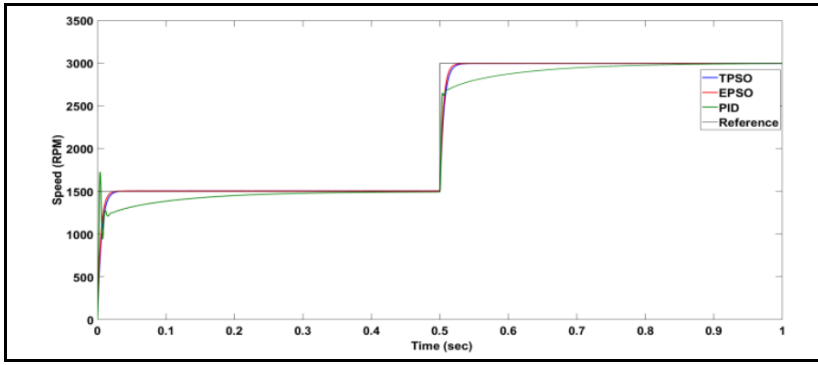


Figure 18. Speed response curve with sudden increase of speed from 1500 – 3000

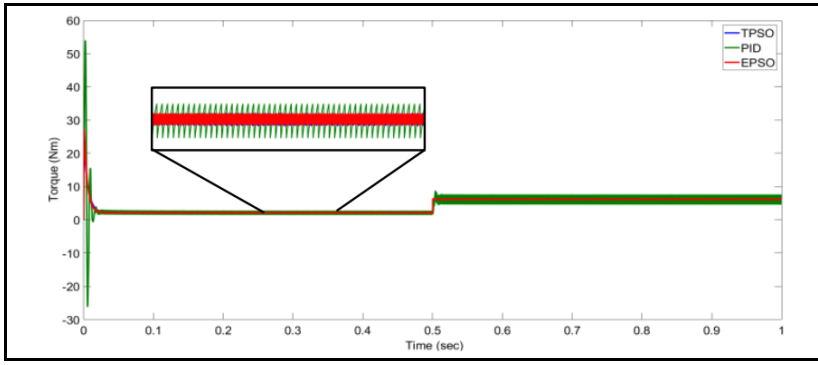


Figure 19. Torque response curve with sudden increase of load from 2 – 6 Nm

The torque ripple produced by the BLDC motor SMC (tuned using EPSO and TPSO) and a PID controller can be known using the torque ripple equation given (Equation 38):

$$TR(\%) = \frac{T_{max} - T_{min}}{T_{ave}} \times 100\% \quad (38)$$

The comparison of torque response under different load conditions is shown in Table 4, and the comparison of speed performance among different control schemes is shown in Table 5. The results show that the EPSO-tuned SMC reduces torque ripple by 15.32% and 24.26% at 3000 RPM and 1500 RPM, respectively, compared to the PID controller. Also, based on the speed response curve shown in Figure 17, the EPSO-tuned SMC for the BLDC motor has the best speed characteristics compared to the TPSO-tuned SMC and PID controller. Figures 16, 17, and 18 show that the PID controller

exhibits overshoot and undershoot. The PID control also has longer rise and settling times, unlike the tuned SMC using PSO and TPSO. Nevertheless, the tuned SMC using TPSO and EPSO can stabilize the speed without overshoot and undershoot, even though a sudden increase in speed occurs at 0.5 s. Thus, it can be concluded that the PSO-tuned SMC can outperform a conventional PID controller.

Table 4. Torque range under different load

| Scheme | Speed (RPM) | Load Torque (Nm) | Torque Ripple (%) |
|---|---|---|---|
| TPSO | 3000 | (0 – 1.5) | 85.52 |
| | 1500 | (2 – 6) | 54.52 |
| PID | 3000 | (0 – 1.5) | 91.44 |
| | 1500 | (2 – 6) | 71.72 |
| EPSO | 3000 | (0 – 1.5) | 77.43 |
| | 1500 | (2 – 6) | 54.32 |

Table 5. Speed performance of different schemes

| Scheme | Speed (RPM) | Overshoot (%) | Rise Time (ms) | Settling Time (ms) |
|---|---|---|---|---|
| TPSO | 3000 | 0.0205 | 8.7 | 17.6 |
| | 1500 | 0.0077 | 8.4 | 16.6 |
| PID | 3000 | 0.0037 | 48.9 | 203.3 |
| | 1500 | 12.022 | 1.3 | 206.7 |
| EPSO | 3000 | 0.007 | 7.0 | 14.6 |
| | 1500 | 0.005 | 6.7 | 13.4 |

The BLDC motor is tested at varying speeds to verify the tuning of the SMC using EPSO and TPSO with respect to torque response under different load conditions. Figure 18 shows that the tuned SMC using EPSO and TPSO has symmetrical speed responses to a sudden change in load torque and to an increase in speed. Also, the EPSO-tuned SMC consistently stabilized more quickly than the PID controller and the TPSO-tuned SMC, even during a sudden change in load torque.

However, in Figure 16, the BLDC motor is operated at a constant speed of 3000 RPM under varying load conditions. When a sudden increase in load torque occurs at 0.5 s, both the tuned SMCs using TPSO and EPSO exhibit more stable performance than the PID controller. However, when the BLDC motor is operated at a rated speed of 0.5 s, a slight undershoot occurs when the load torque increases.

Consequently, SMCs employing different particle swarm algorithms outperform the conventionally tuned PID. The results show that the EPSO- and TPSO-tuned SMCs have significantly improved the settling time and rise time of the speed response, and the torque ripple decreases significantly with lower overshoot during initial operation.

## 4. Conclusion

An effective sliding mode controller is proposed for robust speed control and torque minimization. An adaptive elite-based PSO has been executed to tune the SMC. It has been shown that, compared with a traditional PID controller, the tuned SMC using EPSO and TPSO achieves superior speed regulation and torque ripple minimization. The superiority of the proposed EPSO-tuned SMC has been proven and compared with TPSO-tuned SMC using ITSE. Also, the EPSO demonstrates its advantage over the TPSO by achieving faster convergence in determining optimal SMC parameter values. Lastly, MATLAB/Simulink simulations confirm that the proposed control scheme produces a rapid transient response with robust disturbance-rejection capability under speed and load variations.

To further enhance the practical relevance and robustness of the proposed control scheme, several directions for future research are recommended. First, the proposed EPSO-tuned sliding mode controller should be implemented on real-time embedded platforms, such as field-programmable gate arrays (FPGAs), digital signal processors (DSPs), or microcontrollers, to validate its real-time performance and feasibility for industrial applications. In addition, an online adaptive version of the EPSO algorithm may be developed to dynamically tune the SMC parameters during operation, enabling the controller to adapt to varying load conditions and unexpected system disturbances. Furthermore, comparative investigations involving advanced intelligent control strategies, such as reinforcement learning and deep learning-based controllers, could provide valuable insights into further improving adaptability, learning capability, and control robustness.

Future studies may also extend the optimization framework by employing multi-objective optimization algorithms, such as NSGA-II, to simultaneously optimize speed regulation, torque ripple minimization, and energy efficiency. Moreover, the robustness of the proposed controller should be evaluated under

a wider range of operating conditions, including temperature variations, electromagnetic interference (EMI), and severe load disturbances, to further assess its suitability for industrial deployment. Finally, the generalizability of the proposed EPSO-SMC scheme should be investigated using BLDC motors with different power ratings, pole pairs, and back-EMF characteristics.

These future research directions are expected to bridge the gap between simulation-based validation and practical implementation while enhancing the adaptability, scalability, and industrial applicability of the proposed control strategy.

## 5. Acknowledgement

The authors would like to express their sincere appreciation to the Department of Electrical Engineering at National Taiwan University of Science and Technology and the Electrical Engineering Department at Ateneo de Davao University for their institutional support, which made this study possible.

## 6. References

Caponetto, R., Fortuna, L., Fazzino, S., & Xibilia, M.G. (2003). Chaotic sequences to improve the performance of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, *7*(3), 289–304. https://doi.org/10.1109/TEVC.2003.810069

Chao, C.-T., Sutarna, N., Chiou, J.-S., & Wang, C.-J. (2019). An optimal fuzzy PID controller design based on conventional PID control and nonlinear factors. *Applied Sciences*, *9*(6), 1224. https://doi.org/10.3390/app9061224

Hong, Y.-Y., Lin, F.-J., Chen, S.-Y., Lin, Y.-C., & Hsu, F.-Y. (2014). A novel adaptive elite-based particle swarm optimization applied to VAR optimization in electric power systems. *Mathematical Problems in Engineering*, 2014, 1–14. https://doi.org/10.1155/2014/761403

Housny, H., Chater, E. A., & Fadil, H.E. (2019). Fuzzy PID control tuning design using particle swarm optimization algorithm for a quadrotor. In 5*th International Conference on Optimization and Applications (ICOA)*, Kenitra, Morocco.

https://ui.adsabs.harvard.edu/link_gateway/2019icoa.conf...43H/doi:10.1109/ICOA.2019.8727702

Kim, T.-O., & Han, S.-S. (2020). Fuzzy PID control algorithm for improvement of BLDC motor speed response characteristics. *Journal of the Institute of Electronics and Information Engineers, 57*(6), 105–110. https://doi.org/10.5573/ieie.2020.57.6.105

Ma'arif, A., & Çakan, A. (2021). Simulation and Arduino hardware implementation of DC motor control using sliding mode controller. *Journal of Robotics and Control (JRC), 2*(6). https://doi.org/10.18196/jrc.26140

Malla, S.G., Malla, P., Malla, J.M.R., Singla, R., Choudekar, P., Koilada, R., & Sahu, M.K. (Varsh) (2022). Whale optimization algorithm for PV-based water pumping system driven by BLDC motor using sliding mode controller. *IEEE Journal of Emerging and Selected Topics in Power Electronics, 10*(4). https://doi.org/10.1109/JESTPE.2022.3150008

Maurya, H., Bohra, V.K., Pal, N.S., & Bhadoria, V.S. (2019). Effect of inertia weight of PSO on optimal placement of DG. In *IOP Conference Series: Materials Science and Engineering*, *594*, 012011. https://doi.org/10.1088/1757-899X/594/1/012011

Pan, T.-S., & Lin, S.-Y. (2021). Design and analysis of PID and fuzzy logic controller for simulation performance. *Journal of Internet Technology, 22*(1), 31–39. https://jit.ndhu.edu.tw/article/view/2457/2473

Shi, J., Mi, Q., Cao, W., & Zhou, L. (2022). Optimizing BLDC motor drive performance using particle swarm algorithm-tuned fuzzy logic controller. *SN Applied Sciences, 4*(11). https://doi.org/10.1007/s42452-022-05179-6

Shima, T.J., & Bashir, H.A. (2021). PSO-based integral sliding mode controller for optimal swing-up and stabilization of the cart-inverted pendulum system. *Nigerian Journal of Technological Development, 18*(2), 88–97. https://doi.org/10.63746/njtd.v18i2.610

Storn, R., & Price, K. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(4), 341–359. https://doi.org/10.1023/A:1008202821328

Tang, G., Lei, J., Du, H., Yao, B., Zhu, W., & Hu, X. (2025). Proportional–integral–derivative controller optimization by particle swarm optimization and back propagation neural network for a parallel stabilized platform in marine operations. *Journal of Ocean Engineering and Science, 10*(1). https://doi.org/10.1016/j.joes.2022.05.015

Wang, Y., Feng, Y., Zhang, X., & Liang, J. (2020). A new reaching law for antidisturbance sliding-mode control of PMSM speed regulation system. *IEEE Transactions on Power Electronics*, *35*(4), 4117–4126. https://doi.org/10.1109/TPEL.2019.2933613

Xia, K., Ye, Y., Ni, J., Wang, Y., & Xu, P. (2020). Model predictive control method of torque ripple reduction for BLDC motor. *IEEE Transactions on Magnetics*, *56*(1), 1–6. https://doi.org/10.1109/TMAG.2019.2950953

Xu, X., & Wang, Q. (2017). Speed control of hydraulic elevator using PID controller and self-tuning fuzzy PID controller. In *32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Hefei, China. https://doi.org/10.1109/YAC.2017.7967521

Zhan, Z.-H., Zhang, J., Li, Y., & Chung, H.S.-H. (2009). Adaptive particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics Part B*, *39*(6), 1362–1381. https://doi.org/10.1109/TSMCB.2009.2015956