Super-Twisting Sliding Mode Control of Permanent Magnet DC Motor

Ali Kadhim Mohammed
Electrical Power Technology Engineering Department
Middle Technical University , Technical College of Electrical Engineering
Babylon, Iraq
alikmj87@gmail.com

Nabil Kadhim Al-Shamaa
Department of Control and Automation Techniques Engineering
Middle Technical University, Electrical Engineering Technical College
Baghdad, Iraq
nabil_alshamaa@mtu.edu.iq

Ayad Qasim Al-Dujaili
Department of Control and Automation Techniques Engineering
Middle Technical University, Electrical Engineering Technical College
Baghdad, Iraq
ayad.qasim@mtu.edu.iq

Abstract **- Permanent Magnet Direct Current (PMDC) motors have been broadly used in mechanical and electrical fields that integrate several applications. This type of motor is subjected to disturbances or sudden changes in loads, and motor speed control is necessary under these conditions. In this paper, we have implemented traditional controllers' Proportional Integral Derivative (PID) and advanced controllers Super-Twisting Sliding Mode Control (STSMC) and Sliding Mode Control (SMC) through tuning Particle Swarm Optimization (PSO) that used to control the speed of the PMDC motor. According to the results that appeared in the simulations in the MATLAB\Simulink software, STSMC has superiority compared to other controllers in improving the performance of the speed control for PMDC motors. An experimental setup for PMDC motor by proposed controllers to validate the simulated results in the real environment based on the Arduino UNO interface device. Experimental results confirmed the advantages of employing the STSMC technique over other controllers in improving the performance of speed control for PMDC motors to attain the required speed in the least amount of time and rapid rejection of disturbances in the load.**

Keywords: Speed control, PMDC motor, PID, SMC, STSMC, PSO.

I. INTRODUCTION

PMDC motors have been broadly used within mechanical and electrical fields that integrate with several applications like robotics, air conditioners, computer peripherals, air cleaners, and other industrial applications [1]. The main characteristics and features that make using PMDC motor so widely spread and much popular in modern applications are low cost, high efficiency, ease of speed control, ease in maintenance and the structure is very simple [2]. Practically, the PMDC motors are used with variable speed drive applications due to the higher power that is used to drive the nonlinear loads, these types of motors are subjected to disturbances or suddenly changes in loads, the speed control of these motors under these conditions represent the main difficult challenge. Therefore, the optimal design of the speed control system under variable speed drive conditions must consider the disturbances during the design process [3]. To control the speed of the PMDC motor, the armature voltage is changed and during the speed control operation of the DC motors, usually, the armature current is variable while the field winding current is assumed to be constant. These assumptions of the current conditions provide special speed control results for PMDC motors in a wide range of the desired speeds, the main purpose of these applications is to observe the speed by holding the performance speed at the correct pace and to reach the desired speed [2]. To control the speed of PMDC motors, a variety of controllers can be used such as PID, adaptive fuzzy algorithm and SMC [4]. The earliest controllers that can be used for PMDC motors speed control are the PID controllers. Because of their simplicity, it was used in a huge scope of applications. Many new methods can be used to tackle the tuning problem of many design parameters for controlling the PMDC motors', such as PSO, Genetic-Algorithm (GA) and Simulated-Annealing (SA) methods and other methods [1]. Traditional controllers are incapable of dealing with real-time systems that exhibit nonlinear behaviour. To ensure the good performance of the control system, an advanced control system is required. SMC is a useful method for controlling complex and dynamic systems because it has a low sensitivity to system parametric changes and disturbance effects. Furthermore, it can operate in the face of unpredictability, which is a typical occurrence in modern technology [5]. The SMC could provide numerous great equities; for example, rapid response and quick powerful reaction to parametric perturbations, external disturbances. These benefits of SMC have been utilized in velocity control of PMDC motors, the fundamental disadvantage of SMC is chattering, which is caused by discontinuous controllers. To avoid this constraint and to get more accurate results STSMC is used [6]. As a result, optimization approaches may be more appropriate as a tool for tuning parameters design, resulting in greater stability and performance. The PSO technique has been proposed as an optimal tuner in the optimization field. This research has made two major contributions, the first is to solve the problem associated with the chattering in control signals and the second is to tune parameters design to achieve optimal performance.

Therefore, the aim of this work can be stated as follows:

- Control law development for the PID, SMC and STSMC based on PSO optimization technique.

- Comparative study conducted between traditional controllers and advanced controllers in terms of robustness and dynamic performance.
- MATLAB based experimental verification results of the simulated interface with Arduino UNO are obtained.

The article is organized as follows: The model of the PMDC motor is presented in the second section. The third section is a description of the controllers (PID, SMC, STSMC), and the utilization of the PSO algorithm to find the optimal design parameters of the suggested controller. A simulation study and comparison are conducted to ensure that the suggested strategy is effective as mentioned in the fourth section. In the fifth section, real-time results have been conducted to validate the results of a simulation. The conclusion of this work appears in the sixth section.

II. Mathematical Model of PMDC Motor

PMDC motors are the most widely used machine drives, the motor consists of two main parts, the electrical part and the mechanical part. A rotational speed can be obtained by combining these parts together. The equivalent circuit of PMDC motor as shown in Fig.1.The goal of modelling the PMDC motor is to find the differential equation that describes the relationship between an input voltage and output speed [7, 8].

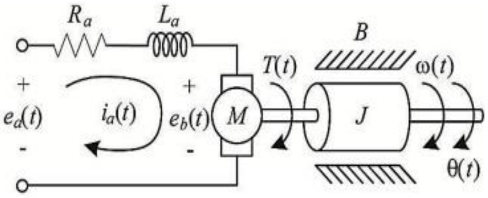


Fig.1. Equivalent circuit of PMDC motor [9].

Since the field winding of the PMDC motor is a permanent magnet, air gap flux density is constant. The electrical equations of the PMDC motor can be expressed as:

$$V_a (t) = e_b (t) + R_a\, i_a(t) + L_a \frac{d}{dt} i_a (t) \tag{1}$$

$$e_b (t) = k_b\, \omega_m(t) \tag{2}$$

where:

$k_b = BAN$

B: Flux density (Wb/m^2).

A: Cross-section area of the rotor (m^2).

N: Number of conductors of the rotor.

$$T_m (t) = J_m \frac{d}{dt} \omega_m(t) + B_m \omega_m(t) + T_L \tag{3}$$

where:

$$T_m (t) = K_t\, i_a(t) \tag{4}$$

where:

$k_t = k_b$

The motor is considered to operate in one direction within the stated speed range, hence zero crossing non-linearity is ignored. The following Laplacian transfer function can be used to characterize the PMDC model in this scenario.

$$\frac{\omega_m(s)}{V_a (s)} = \frac{k_t}{J_m L_a S^2 + (J_m R_a + B_m L_a) S + B_m R_a + k_t k_b} \tag{5}$$

III. CONTROL ANALYSIS FOR PMDC MOTOR

A. *PID Controller*

The PID controller is one of the most common controllers available. The PID controller is used to improve dynamic response while also reducing or eliminating steady-state error [9]. PID controllers, on the other hand, account for over 90% of all systems control utilized in industrial environments. The PID algorithm has the benefit of having a little number of design parameters and being able to readily tie controller parameters to performance measures. In this controller the error signal is send to the PID controller. The proportional, integral and derivative of this error signal are computed by the controller. It transforms the error signal $e(t)$ into the control signal $u(t)$. Equation (6) represents the general equation of the PID controller[10]:

$$u(t) = K_p\, e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{6}$$

where:

K_p : The proportional gain, K_i : The integral gain, K_d : The derivative gain.

B. *Sliding Mode Control*

There may be problems with the use of classical control techniques. One of the factors leading to this is inconsistencies between the actual plant and the mathematical model established for controller design. This mismatch may prevent the required level of performance from being achieved. To eliminate any disparity, a set of reliable control procedures was devised. One approach to robust control design is the SMC methodology [11]. The most prominent property of SMC is that it is fully immune to parametric uncertainty and external disturbances [12]. SMC is divided into two parts. Fig. 2 depicts the reaching and sliding modes, as well as their phase portraits. The SMC method tried to pull the system states from the sliding surface to the origin. Movement in sliding the surface refers to SM which is a stage from the starting point to SM is called the reaching mode. The purpose of SMC is for an output to be traced the reference and for a control signal u(t) to be produced, which lowers error tracking [5]. Signal $u(t)$ can be demonstrated as in equation (7) which are consisted of two parts. The first part represents an equivalent control signal $u_{eq}(t)$, whereas the second part represents a switching control signal $u_{sw}(t)$.

$$u(t) = u_{eq}(t) + u_{sw}(t) \tag{7}$$

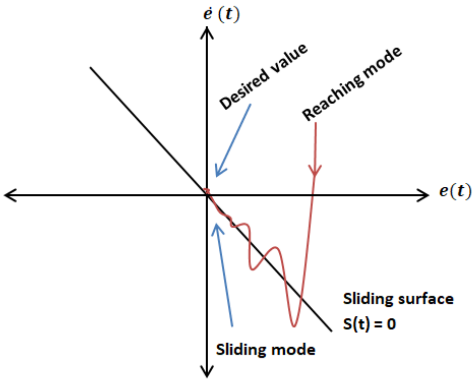


Fig.2. SMC phase portrait [5].

An equivalent control signal can be found by determine the parameters and transfer functions of the PMDC motors. Switching control sign signals can be found through the sliding surface. The equation mostly used for the sliding surface is as follows

$$s(t) = C\,X_1 + X_2 = C\,e(t) + \dot{e}(t) \tag{8}$$

$$\dot{s}(t) = C\,\dot{X_1} + \dot{X_2} = C\,\dot{e}(t) + \ddot{e}(t) \tag{9}$$

$$e\,(t) = \omega_{ref} - \omega_{act} \tag{10}$$

where:

C: Performance parameter, ω_{ref}: Reference speed.

ω_{act}: Actual speed from dc motor.

It is shown that sliding surface is action, which is related to tracking error for any difference between reference here output. In here $C > 0$ is a performance parameter that guaranteed the stability of the system on the sliding surface.

Both a sliding surface and its derivative a sliding surface is equal to zero as $s(t) = 0$ and $\dot{s}(t) = 0$. To ensure sliding mode motion on the surface. After found the sliding surface it can be written the equation of the switching control signal as follows:

$$u_{sw}(t) = K\,sgn(s) \tag{11}$$

$$Sgn\,\big(s(t)\big) = \begin{cases} +1\,, if\ s\,(t) > 0 \\ \ 0\,, if\ s\,(t)\ =\ 0 \\ -1, if\ s\,(t) < 0 \end{cases} \tag{12}$$

where:

K: Positive design parameters.

$sgn(s)$: Presents signum function.

After simplifying the equation (5), the design of SMC for PMDC motor.

$$\frac{\omega_m(s)}{V_a\,(s)} = \frac{\frac{k_t}{J_m\,L_a}}{S^2 + \left(\frac{R_a}{L_a} + \frac{B_m}{J_m}\right)S + \left(\frac{B_m R_a}{L_a\,J_m} + \frac{k_t\,k_b}{L_a\,J_m}\right)} \tag{13}$$

Let

$$A_1 = \frac{k_t}{J_m\,L_a},\quad A_2 = \frac{R_a}{L_a},\quad A_3 = \frac{B_m}{J_m},\quad A_4 = \frac{k_t\,k_b}{L_a\,J_m}$$

$$\frac{\omega_m(s)}{V_a\,(s)} = \frac{A_1}{S^2 + (\,A_2 + A_3\,)\,S + (A_2\,.\,A_3 + A_4)} \tag{14}$$

In time-domain equation (14) can be written as:

$$\ddot{\omega}_m\,(t) + (A_2 + A_3)\dot{\omega}_m\,(t) + (A_2A_3 + A_4)\omega_m(t) = A_1\,V_a(t) \tag{15}$$

The state-space depiction of the model is

$$x_1 = \omega\,(t) \tag{16}$$

$$x_2 = \dot{\omega}\,(t) \tag{17}$$

$$\dot{x}_1 = \dot{\omega}\,(t) = X_2 \tag{18}$$

$$\dot{x}_2 = \ddot{\omega}\,(t) \tag{19}$$

$$y = \omega\,(t) \tag{20}$$

$$\dot{x}_2 = \ddot{\omega}_m\,(t) = -(A_2 + A_3)\dot{\omega}_m\,(t) - (A_2A_3 + A_4)\omega_m\,(t) + A_1\,v_a(t) \tag{21}$$

Equivalent control can be found by substituting equation (18) and equation (21) into equation (9).

$$\dot{s}(t) = C\,X_2 - (A_2 + A_3)X_2 - (A_2A_3 + A_4)X_1 + A_1\,v_a(t) \tag{22}$$

To assure SMC motion on the surface, $\dot{s}$ should be equal to zero, and the corresponding control will be achieved by employing the aforementioned condition.

$$u_{eq} = \frac{1}{A_1}\left[(A_2 + A_3 - C\,)X_2 + (A_2A_3 + A_4)\,X_1\,\right] \tag{23}$$

$u(t)$ control signal will produce as

$$u(t) = \frac{1}{A_1}\left[(A_2 + A_3 - C\,)X_2 + (A_2A_3 + A_4)X_1\,\right] + ksgn(s) \tag{24}$$

A change in signum function as shown in (25) where δ is a small positive design constant also called as tuning parameter used is done to reduce processing load and relieve chattering.

$$u(t) = \frac{1}{A_1}\left[(A_2 + A_3 - C\,)X_2 + (A_2A_3 + A_4)\,X_1\,\right] + K\left[\frac{S}{|S| + \delta}\right] \tag{25}$$

where: $0 < \delta < 1.$

C. Super Twisting SMC

Super-twisting is a new hypothesis in SMC design that has been shown to be effective for electromechanical systems [12]. SMC is a technology that has been widely employed in real-world applications, particularly because of its complicated and simple control structure, which has a wide range of applications in mechanical systems. However, in terms of chattering minimization and enhanced performance, the fundamental disadvantage of SMC is the chattering signal. One method to get around this constraint is to use the SMC with a higher-order control structure. The STSMC is a second-order control structure. It is possible to exhibit signal $u(t)$ as (26) which are made up of two sections. The first part represents an equivalent control signal $u_{eq}(t)$, it is similar to SMC and it can be found by parameters and transfer functions of the PMDC motor. Whereas the second part represents a switching control signal $u_{sw}(t)$ [13].

$$u(t) = u_{eq}(t) + u_{sw}(t) \tag{26}$$

Equation switching control signal is

$$u_{sw}(t) = k\,\sqrt{|s|}\,sgn(s) + w\,\int sgn(s)\,dt \tag{27}$$

where:

k, w: Positive design parameters, s: Sliding surface,

$sgn(s)$: The signum function.

$$S(t) = C\,X_1 + X_2 = C\,e(t) + \dot{e}(t) \tag{28}$$

D. Particle Swarm Optimization

In 1995, Kennedy and Eberhart developed the PSO technique [14]. It was improved by introducing a new parameter called inertia weight to boost its performance. The PSO has a strong global search, a faster convergence rate, a few configurable parameters, algorithm simplicity, and ease of implementation as compared to other optimization methods. The PSO algorithm is based on swarm intelligence approaches for observing the social behavior of moving organisms like fish or birds [15]. As it moves through the problem space, each particle has a different velocity. The main premise of the PSO technique is to use a random weighted acceleration to accelerate each particle to its pbest and gbest positions at each time step. Each PSO particle has a specific velocity in the search space, which is dynamically adjusted based on its own flight experience and information gathered from its peers' velocities. The particles adjust their velocity and location every iteration depending on their previous best position pbest and the best position of all other particles in the swarm gbest. Variables and dimensions exist in every particle, and these variables are problems that must be solved. If there are multiple variables in the problem, the particle dimension must be chosen to be equal to these variables. Each iteration updates the position and speed using equations [16, 17]:

$$V_{i,j}(t+1) = W.V_{i,j}\,(t) + r_1c_1\big[Pbest_{i,j}\,(t) - X_{i,j}\,(t)\big] + r_2c_2\big[Gbest_{i,j}\,(t) - X_{i,j}\,(t)\big] \tag{29}$$

$$X_{i,j}(t+1) = V_{i,j}(t+1) + X_{i,j}(t) \tag{30}$$

where:

i: Particle index, j: Number dimension, t: Iteration, W: The values of Weight inertia range [0 - 1], $V_{i,j}(t)$: The instant velocity of the particle, $V_{i,j}(t+1)$: The next instantaneous velocity of the particle, $X_{i,j}(t+1)$: The next position of the particle, $X_{i,j}(t)$: The instant position of the particle, r_1, r_2: Random coefficient they're between values [0 - 1], c_1, c_2: The values of acceleration coefficients range [0 – 2].

However, the objective function employed in these algorithms to mini the speed error is defined as:

$$ITAE = \int_0^T t\,|e(t)|\,dt \tag{31}$$

$$e(t) = (\omega_{ref} - \omega_{act}) \tag{32}$$

where T is the simulation time, ω_{act} is the actual speed, and ω_{ref} is the reference speed of the PMDC motor.

IV. SIMULATION RESULTS

Based on the optimization of PSO tune design parameters, the performance of the PID, SMC, and STSMC controllers was compared in this study to improve the dynamic performance of the PMDC motor. In table (1), it can be found the values adopted for the PMDC motor. Simulations were run to demonstrate the performance of the PMDC motor with the controllers proposed. Table (2) shows the parameters of the controllers proposed based on the PSO. The parameters of the PSO that were employed in this study are listed in the table (3).

TABLE 1. THE PERMANENT MAGNET DC MOTOR PARAMETER

The PMDC motor Parameter		
Description	***Symbols***	***value***
Armature-winding resistance	*Ra*	11.27 Ω
Armature-winding inductance	*La*	0.0082 H
Torque constant	K_t	0.00556 Nm / A
Back EMF constant	K_b	0.00556 V – sec/rad
Viscous fractional coefficient	B_m	6.14 e-4 Nms/rad
Rotor inertia of the motor	J_m	1.23e-3 kg.m^2
Applied armature voltage	*Va*	24 v
Load torque	T_l	0.01Nm
No load speed of motor	ω_m	3800 rpm

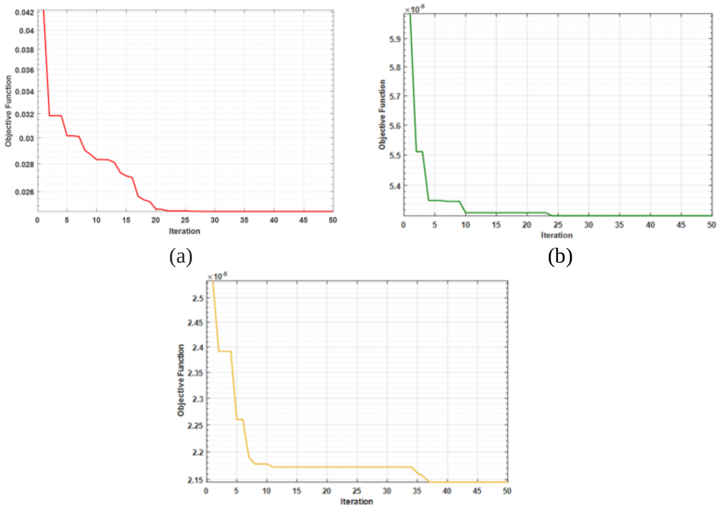
TABLE 2. OPTIMAL DESIGN PARAMETERS OF CONTROLLERS.

PID	K_p	K_i	K_d
	14.01098	6.73	0.1
SMC	K	C	δ
	150. 45237	80.765	0.9
STSMC	k	C	w
	203.67	98.456	15.023

TABLE 3. PSO PARAMETERS.

Parameters PSO	Value PID	Value SMC	Value STSMC
Number dimension (*D*)	3	3	3
Weighted inertia (*W*)	0.854	0.85	0.85
Coefficients of acceleration (c_1)	2	1.8	1.8
Coefficients of acceleration (c_2)	2	1.5	1.5
Iteration	50	50	50
Swarm size	20	20	20

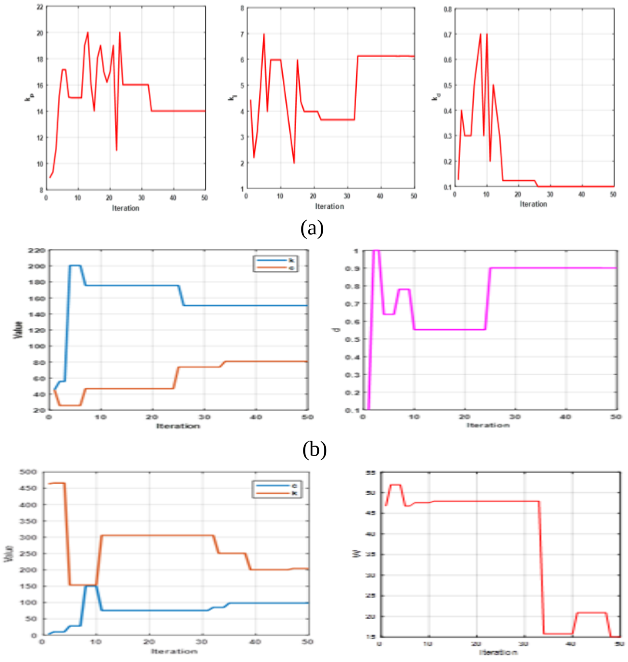
The cost function of the PSO that was employed in the controllers proposed is shown in fig.3. Fig.4 represents the best values parameters of the controller proposed.



(a) (b)

(c)

Fig.3. Behaviors of objective function based on of PSO a) PID, b) SMC, c) STSMC .



(a)

(b)

(c)

Fig.4. The best values of the controllers parameters based on of PSO a) PID, b) SMC, c) STSMC .

Fig (5 and 6) represents the simulation results for the speed of the PMDC motor under the load and multi-input reference speed respectively depending on the parameters extracted by PSO tuning for PID, SMC and STSMC controller gains. It is noted that the speed based on STSMC controller of PSO tuning is better than the speed based on PID and SMC controllers of PSO tuning in terms of settling time, rise time and rapid response to disturbance as shown in table (4).

Fig (7, 8, 9) show the control signal of the PMDC motor for PID, SMC and STSMC respectively. The figures show the control signal of the SMC controller has chattering, but the control signal of the STSMC controller has no chattering. This is why was using STSMC to avoid chattering.

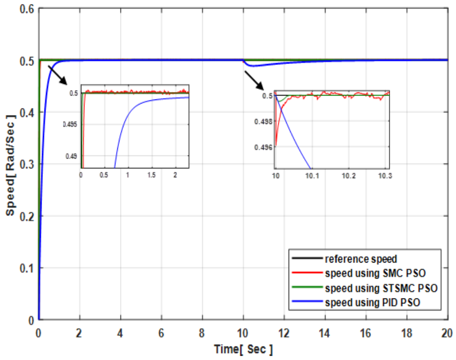


Fig. 5. Speed response of PMDC motor under $T_L = 0.01$ Nm at 10 sec using PID, SMC and STSMC controllers.

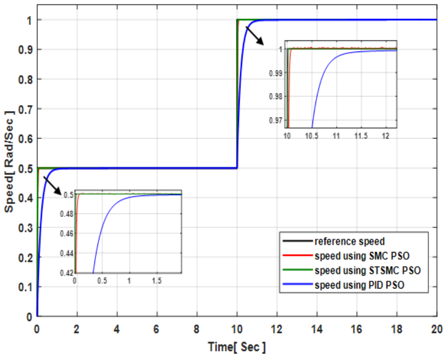
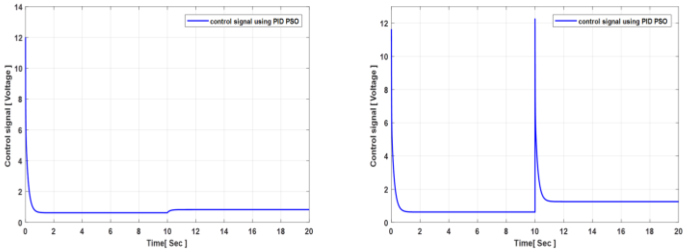


Fig. 6. Speed response of PMDC motor using PID, SMC and STSMC controllers under multi-input reference speed at no-load.

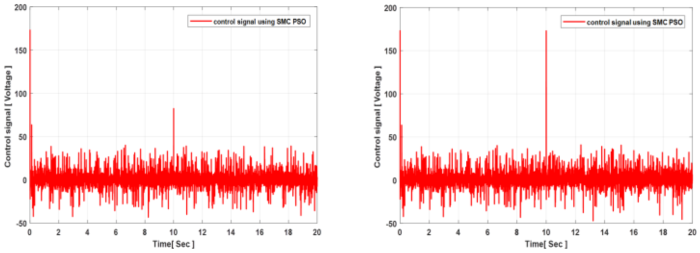
TABLE 4. THE OUTPUT RESPONSE OF A PMDC MOTOR IN MATLAB SIMULINK.

Method	***Settling time (t_s)***	***Rise time (t_r)***
PID	4.6	0.4011
SMC	0.18	0.0251
STSMC	0.04	0.0085



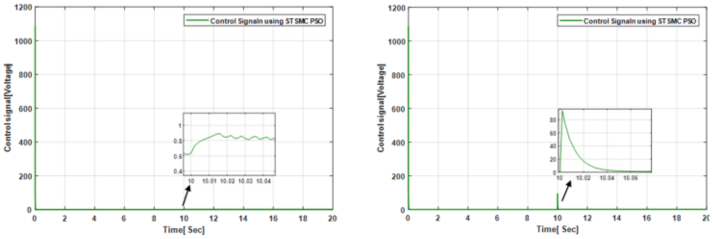
(a) (b)

Fig.7. Control signal, a) PID controller under $T_L = 0.01$ Nm at 10 sec, b) PID controller under multi-input reference speed at no-load.



(a) (b)

Fig.8. Control signal, a) SMC controller under $T_L = 0.01$ Nm at 10 sec, b) SMC controller under multi-input reference speed at no-load.



(a) (b)

Fig.9. Control signal, a) STSMC controller under $T_L = 0.01$ Nm at 10 sec, b) STSMC controller under multi-input reference speed at no-load.

V. REAL TIME IMPLANTATIONS OF PID, SMC AND STSMC FOR PMDC MOTOR

The overall hardware platform in the experimental study was a PMDC motor with a PC, Arduino UNO device, power supply and drive circuits. The parameters of a PMDC motor are shown in table (1). MATLAB software is used to carry out the control action. A feature in MATLAB allows you to connect to an Arduino UNO. Signals created by the MATLAB-based controllers are sent from the Arduino UNO digital Pulse Width Modulation (PWM) ports to drive circuits which are used to control the armature voltage of the PMDC motor. The feedback signal data from the tachometer is received to the Arduino UNO input analogue port. The hardware system includes the all components as shown in fig.10. In this figure a real-time implementation of the PMDC motor performance under the PID, SMC, and STSMC controllers. As can be seen from the results of the experimental evaluation in fig. 11 by using PID, SMC, and STSMC controllers. In these results, it can be visually seen that the STSMC controller provides a better performance, in transient and also, steady-state response, and quick disturbance reaction. This indicates that a better response of performance can be obtained by the STSMC controller than that given by PID and SMC controller. To assess both controllers' performance and based on the results presented in table (5).

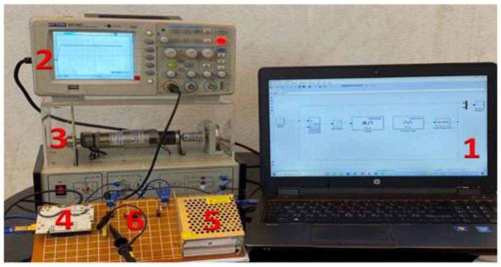
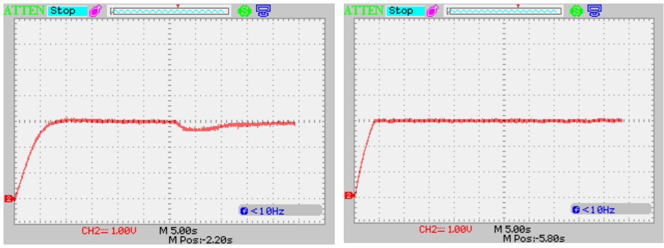
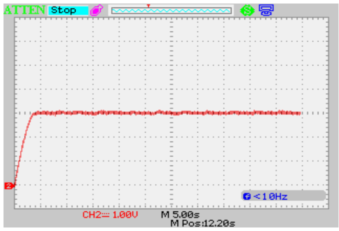


Fig.10. Overview of the experimental setup.



(a) (b)



(c)

Fig. 11. Experimental results using controllers proposed under $T_L = 0.01$Nm at 30 Sec. a) PID, b) SMC, c) STSMC .

TABLE. 5. THE OUTPUT RESPONSE OF A PMDC MOTOR IN MATLAB SIMULINK.

Method	*Settling time (t_s)*	*Rise time (t_r)*
PID	8	5
SMC	4	3.1
STSMC	3	2.3

VI. CONCLUSION

The current study was generally dedicated to control the PMDC motor speed. Advanced and traditional controllers are employed in this work to increase the speed responsiveness of a PMDC motor, which is a critical issue in control systems. The PSO algorithm has been proved to be a feasible alternative to classical trial-and-error approaches for optimal tuning of design parameters in PID, SMC and STSMC controllers. Moreover, it was confirmed that this optimization technique confirms that the STSMC controller outperforms the PID and SMC controller in general. According to computer simulation results, the STSMC is more effective than the controller PID and SMC which has been demonstrated in practice. In terms of settling time, rise time and quick disturbance reaction, STSMC has shown to be successful. The STSMC controller is the most stable system and the most effective controller in terms of achieving the desired goals. In addition, STSMC treats the chatter that appeared using SMC. STSMC treats the chatter that appeared using SMC. This study can be extended for future work in control direction. One may suggest other control schemes such adaptive backstepping control, block backstepping control, and model reference adaptive control to conduct comparison in performance to SMC for DC motor [18, 19].

REFERENCES

[1] Mirza Muhammad Sabir and Junaid A. Khan “Optimal design of PID controller for the speed control of DC motor by using metaheuristic techniques” Hindawi Publishing Corporation Advances in Artificial Neural Systems, pp. 1-8, 2014.

[2] Ghazanfar Shahgholian, Pegah Shafaghi” State-space modelling and eigenvalue analysis of the permanent magnet DC motor drive system” IEEE International Conference on Electronic Computer Technology, pp. 63-67, 2010.

[3] Fredy E. Hoyos, Alejandro Rincón, John A. Taborda, Nicolás Toro, and Fabiola Angulo” Adaptive quasi-sliding mode control for permanent magnet DC motor” Hindawi Publishing Corporation Mathematical Problems in Engineering, pp. 1-12, 2013.

[4] Zhang, Panlong, and Zhongcheng Wang “Improvements of direct current motor control and motion trajectory algorithm development for automated guided vehicle” Advances in Mechanical Engineering, vol. 11, pp. 1-12, 2019.

[5] Emre H. Dursun and Akif Durdu “Speed control of a DC motor with variable load using sliding mode control” International Journal of Computer and Electrical Engineering, vol. 8, no. 3, pp. 219 -226, 2016.

[6] Ayad Q. Al-Dujaili, Alaq Falah, Amjad J. Humaidi, Daniel A. Pereira and Ibraheem K. Ibraheem “Optimal super-twisting sliding mode control design of robot manipulator: Design and comparison study” International Journal of Advanced Robotic Systems, pp. 1-17, 2020.

[7] Ahmed Alkamachi “Permanent magnet DC motor (PMDC) model identification and controller design” Journal of Electrical Engineering, vol. 70, no. 4, pp. 303-309, 2019.

[8] S. Z. Moussavi, M. Alasvandi and Sh. Javadi “Speed control of permanent magnet DC motor by using combination of adaptive controller and fuzzy controller” International Journal of Computer Applications, vol. 52, no. 20, pp. 11-15, 2012.

[9] Santosh Kumar Suman and Vinod Kumar Giri “Speed control of Dc motor using optimization techniques based PID controller” 2nd IEEE International Conference on Engineering and Technology (ICETECH), pp. 1-7, 2016.

[10] Kiam Ang, G. Chong and Yun Li “PID control system analysis, design, and technology” IEEE Transactions on Control Systems Technology, vol.13, no. 4, pp. 559-576, 2005.

[11] Akram Hameed, Ayad Al-Dujaili, Amjad Humaidi, Design of Terminal Sliding Position Control for Electronic Throttle Valve System: a Performance Comparative Study, International Review of Automatic Control, 12 (5), 2019.

[12] Alaq Falah, Amjad J. Humaidi, Ayad Al-Dujaili and Ibraheem Kasim Ibraheem “Robust super-twisting sliding control of PAM-actuated manipulator based on perturbation observer”, Cogent Engineering, vol. 7, pp. 1-30, 2021.

[13] Dil Kumar T R and Mija S. J" Design and performance evaluation of robust SMC schemes for speed control of DC motor" IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), pp. 88-92, 2014.

[14] James Kennedy and Russell Eberhart “Particle Swarm Optimization” IEEE Int Conf Neural Network, pp. 1942-1948, 1995.

[15] Harun Yazgan, et al "Comparison performances of PSO and GA to tuning PID controller for the DC motor" Sakarya University Journal of Science, vol. 23, pp. 162-174, 2019.

[16] Liu-Hsu Lin, Fu-Cheng Wang, et al “Robust PID controller design using particle swarm optimization” Proceedings of the 7th Asian Control Conference, Hong Kong China, pp. 1673-1678, 2014.

[17] Qinghai Bai “Analysis of particle swarm optimization algorithm” Computer and Information Science, vol. 3, no. 1, pp. 180-184, 2010.

[18] Al-Dujaili, A.; Amjad, H.; Pereira, A.; and Ibraheem Kasim, "Adaptive backstepping control design for ball and beam system," International Review of Applied Sciences and Engineering, 12 (3), pp. 211–221, 2021.

[19] Amjad, J.H., Hameed, M. R.,"Design and performance investigation of block-backstepping algorithms for ball and arc system," IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, ICPCSI 2017, 2018, pp. 325–332.