2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS) | 978-1-6654-0343-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/I2CACIS52118.2021.9495918

Real-time Design and Implementation of Nonlinear Speed Controller for Permanent Magnet DC Motor Based on PSO Tuner

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

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

Nabil Kadhim Al-Shamaa
Electrical Engineering Department
Middle Technical University ,
Technical College of Electrical Engineering Baghdad, Iraq
nabilshamaa59@gmail.com

***Abstract* – Speed control of PMDC motors finds applications in various industries. For certain structures, the controller of Proportional, Integral and Derivative (PID) is typically the first choice because of its ease of execution and fast tuning. So, all conventional techniques and optimization stochastic for PID controller tuning provide preliminary feasible parameters for Kp, Ki, and Kd. This paper uses traditional PID and nonlinear PID to control the speed of the PMDC motor, which is tuned by Particle Swarm Optimization (PSO). The methodology is demonstrated by performing simulations using the MATLAB tool. The simulation results exhibit that the role of the nonlinear PID based scheme is more robust than the traditional PID controller, as well as the speed, tracked the desired reference rabidly.**

Keywords: DC motor; tuning controller PSO; NPID controller; PID controller.

I. INTRODUCTION

There are wide applications that use DC motor in industry [1]. This is due to the ease of control, low costs, especially in the form of brushless DC motors and its ruggedness across a wide range of applications. Machine tools that use DC motors include paper mills, electrical traction, the garment industry and robots are all examples of industrial applications. The ability to monitor armature winding and field winding independently is credited with the simplicity of DC motor controller configuration [2-3]. In most DC motor speed control implementations, the field winding current is kept constant while the armature winding current varies or vice versa, resulting in excellent speed control performance over a wide range of desired values .The aim goal of the plant control is to achieve the target speed or position in the shortest time possible with the least amount of overshoots and settling times possible [4]. There are numerous kinds of controllers, such as lead, lag, linear quadratic regulator (LQR), PID and sliding-mode control that could be integrated into control applications [1-5-6].Among the few types of controllers listed, PID controllers are one of the earliest and best understood controllers because of their efficiency and ease of execution ,so it is implemented into almost every industrial control application [4-7]. While there are several traditional techniques for the configuration and tuning of parameters of the PID controller (K_p , K_i and K_d). One of those is commonly known, fine tuning of parameters by the trial and error. Metaheuristic strategies, though, could be a reasonable option due to their diverse existence. Over the years, several metaheuristic optimization techniques have been developed that are being introduced in any life discipline [8-9]. These strategies are influenced by nature based on the swarm intelligence developmental or foraging actions of distinct animals.these strategies are influenced by nature. Some of these techniques are Genetic algorithm (GA), particle swarm optimization (PSO) and simulated annealing (SA). These metaheuristic algorithms have also been successfully implemented in different fields of control systems and their superiority over classical techniques has been shown by the findings obtained by these techniques [2]. A PID and NPID controller configuration for DC motor speed regulation is introduced in this article. by PSO optimization algorithm used to find the best possible PID and NPID controller parameters that improve the performance. Kennedy and Eberhart suggested a hybrid metaheuristic technique in 1995, which is a type of evolutionary algorithm focused on a population of individuals and driven by the simulation of social activity rather than the individual's survival. It's an evolutionary algorithm that's dependent on population. PSO is started with a population of random solutions, much like the other population-based evolutionary algorithms. Unlike other evolutionary algorithm solutions, PSO uses a randomized velocity and particles to represent possible solutions that are flown through the problem space. The main distinction between PSO and other evolutionary algorithms is that PSO prefers collaboration to competition. Algorithms, on the other hand, often employ some sort of decimation, such as survival of the fittest. The PSO population, on the other hand, is stable, with no individuals being produced or killed. Individuals are affected by their neighbors' best results. Individuals finally converge on the problem domain's optimum points [9]. These metaheuristic algorithms have also been efficiently implemented in a variety of control system fields, with the findings demonstrating their superiority over traditional techniques [2–8]. The contribution of the present work can be embodied by pursuing the following steps of objectives.

- design the conventional PID controller and nonlinear PID controller , based PSO algorithm technique to Controller for the speed PMDC motor .
- To compare the performance of the designed controllers to control speed PMDC motor .
- To implement the designed controllers experimental to control PMDC motor through MATLAB interface with Arduino UNO.

Section two represents mathematical model of PMDC. Section three introduces the principle of PSO. PID and NPID are explained in section four and five respectively. The results and discussion are given in section six. Practical implementation of PID and NPID using PSO are given in section seven. and finally section the Conclusion.

II. Mathematical Model of PMDC Motor

The following set of relations [1] describes the dynamic action of the PMDC motor, and its block diagram is shown in Fig.1. To promote the use of metaheuristic techniques, a generalized linear model is given that ignores nonlinearities such as backlash and dead zones. Newton's law and Kirchhoff's law are included in the PMDC motor equations. This can be written as:

$$J_m \frac{d}{dt}\,\omega_m(t) + B_m\omega_m(t) = K_t i_a\,(t) - T_L \tag{1}$$

$$L_a \frac{d}{dt}\,i_a\,(t) + R_a\,i_a\,(t) = v_a\,(t) - k_b\,\omega_r(t) \tag{2}$$

where

V_a is armature applied voltage.

e_b the back- emf, is related to the rotational velocity by:

$e_b = k_b\,\omega_r(t)$.

T_m is engine torque produced, $T_m = K_t i_a\,(t)$ For the separately excited PMDC motor.

J_m the moment of inertia.

B_m damping ratio of the mechanical system.

T_L is torque delivered to load.

R_a the electrical resistance of the armature circuit.

L_a a the electrical inductance of the armature circuit.

I_a is current of armature.

k_t (armature constant) is equal to k_b (motor constant). In the state-space form, the equations above can be expressed by choosing the rotational speed and electric current as the state variables and the voltage as an input. The output is chosen to be the rotational speed.

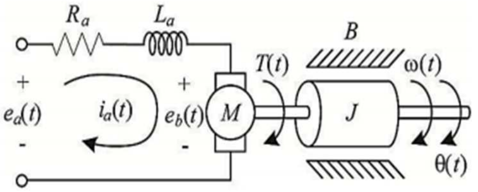


Fig.1. DC motor circuit The Equivalent Circuit of PMDC Motor using the Armature Voltage Control.

$$\dot{X} = Ax + Bu \tag{3}$$

$$Y = Cx \tag{4}$$

$$\begin{bmatrix} \dot{\omega_r} \\ \dot{i_a} \end{bmatrix} = \begin{bmatrix} -\frac{B_m}{J_m} & \frac{K_t}{J_m} \\ -\frac{K_b}{L_a} & -\frac{R_a}{L_a} \end{bmatrix} \begin{bmatrix} \omega_r \\ i_a \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L_a} \end{bmatrix} V_a \tag{5}$$

$$\omega_r = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_r \\ i_a \end{bmatrix} \tag{6}$$

III . PARTICLE SWARM OPTIMIZATION (PSO)

Kennedy and Eberhart introduced the particle swarm optimization algorithm in 1995 . Particle swarm optimization (PSO) is an evolutionary computation optimization technique (a natural system-based search method). A population of random selective solutions is initially present in the method. Each possible solution is referred to as a particle. Each particle has a different velocity as it travels through the problem space. The particles have memory, and each one remembers its previous best position (referred to as the pbest) and health. There are many pbest for each particle in the swarm, with the global best (gbest) of the swarm being the particle with the best fitness. The PSO technique's basic principle is to accelerate each particle towards its pbest and gbest positions at each time stage with a random weighted acceleration. Three measures make up the simple PSO algorithm (generating particles positions and velocities, velocity update, and position update):

$$V_i^{k+1} = W.V_i^k + rand\,.c_1\left[p_{best} - X_i^k\right] + rand.c_2 \left[g_{best} - X_i^k\right] \tag{7}$$

$$X_i^{k+1} = V_i^{k+1} + X_i^k \tag{8}$$

where the weights w, C1, and C2 are, respectively, the inertia, the self-confidence, and the swarm confidence. The suitable value range (C_1 and C_2) is between (1–2), The function "rand" generates random numbers with zero . where represent the current and updated values, respectively, containing the PID and NPID design parameters which are required to be tuned. The PSO-based PID and NPID scheme for the PMDC motor system is depicted in Fig.2 [9,11,12].

The algorithm PSO tuning method for PID and NPID controller is used in order to reduce the error of speed of PMDC motor. There might be many possible objective functions such as integral time-absolute-error (ITAE), integral absolute error (IAE), integral time-squared-error (ITSE). but the objective function used in these algorithms for the minimization the error of speed is defined as [13]:

$$ITAE = J_{min} = \int_0^T t\,|e(t)|\,dt \tag{9}$$

$$e(t) = (\omega_r - \omega_a) \tag{10}$$

where T is the final simulation time, ω_r is the reference speed and ω_a is the $actual$ speed from DC motor.

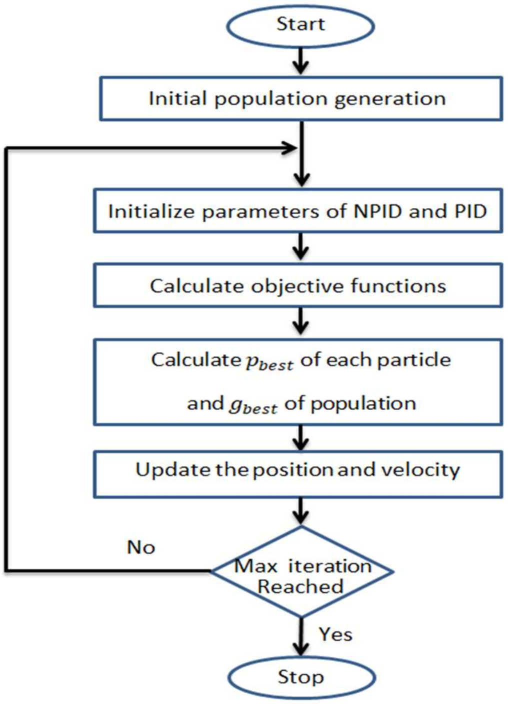


Fig.2. Flowchart of NPID and PID - PSO algorithm

IV . TRADITIONAL PID (PID).

Fundamentally, PID controllers are composed of three basic control actions,

$$u(t) = K_p\,e(t) + K_i\int_0^t e(t)dt + K_d\frac{de(t)}{dt} \tag{11}$$

Where e (t) represents the system error (the difference between the reference input and the system output), u(t) represents the control variable, K_p represents the proportional gain, K_i represents the integral gain, and K_d represents the derivative gain [9] . PID controllers are one of the most widely used in a variety of industries. The most critical aspect of using these controls is to fine-tune their parameters in order to achieve the desired result. For the determination of these control parameters (K_p , K_i and K_d), an accessible system with high precision and speed must be used. The control architecture used for PID controller is shown in fig.3 implementation of Particle Swarm Optimization (PSO) tuned PID controller for speed control of Permanent Magnet DC motor [12].

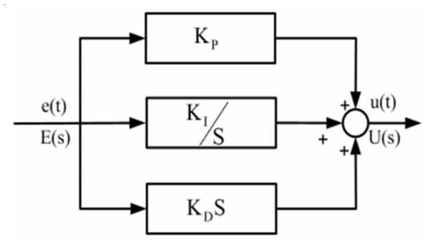


Fig. 3. Block diagram of PID controller.

V. NONLINEAR PID (NPID).

Several modified proportional integral derivative (PID) controller structures have been presented in industrial control applications over the last two decades. The nonlinear PID (NPID), which was introduced by HAN, is one of these controllers. The key idea was to use a nonlinear gain to replace the gain scheduling. As previously mentioned, the NPID controller's key algorithm is based on a nonlinear function that is an intrinsic part of the controller. The main aim is to achieve a desired response in the plant's production when traditional PID failed to do so.

Nonlinear PID Controller Properties

When comparing the proposed nonlinear PID (NPID) controller to the linear PID (LPID) controller, the following properties are mentioned:

- Property of disturbance attenuation: The use of a nonlinear function results in this property. The disturbance effects will be reduced if the extra design parameters (alpha) and (Delta) are chosen correctly. The NPID controller uses the same LPID control parameters as the LPID controller ($K_p\,, K_i\ and\ K_d$) as well as (alpha) and (Delta).
- Reduced control effort: In general, the linear function term will result in a significant amount of control. Value controllers ($K_p\,, K_i\ and\ K_d$) are responsible for this. The value controllers ($K_p\,, K_i\ and\ K_d$) will be reduced if the linear function term is replaced with a nonlinear function term.

The PID controller has been developed to get a more satisfactory response for the ,PMDC motor where it replaces each term of the PID controller with a nonlinear function which is a nonlinear combination of sign and exponential functions of the error signal as given below :

$$u(t) = K_p.Fun(e_p, \alpha_p, \delta_p) + K_i.Fun(e_i, \alpha_i, \delta_i) + K_d.Fun(e_d, \alpha_d, \delta_d) \tag{12}$$

Where Fun(e , α , δ) is the nonlinear function:

$$Fun(e\,, \alpha\,, \delta\,) = \begin{cases} |e|^{\alpha}.\,sign\,(x) & |x| > \delta \\ \delta^{\alpha-1}.\,x & |x| \le \delta \end{cases} \tag{13}$$

$K_p\,, K_i$ and K_d are the controller gains and they having the same meaning as the PID gains. The error expressions are:

This controller, obviously, has far more degrees of freedom, making it much more designable but also more difficult to tune.

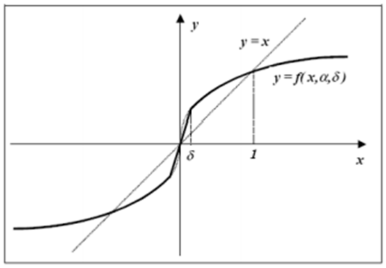


Fig. 4 . Illustration of NPI function.

where (Delta) is the error derivative threshold that distinguishes between the linear (below or equal) and nonlinear (above) regions. Above the value of, the parameter (alpha) controls the nonlinearity and complexity of the function Fun(.). As shown in Fig .4, the value of is usually chosen between (0-2) and the Fun (.) function has linear characteristics with =1. For the speed regulation of a PMDC motor, a Particle Swarm Optimization (PSO) tuned PID controller was used [14,15,16,17].

VI. SIMULATION RESULTS

In this paper the PID and NPID control are used to control the speed of the permanent magnet DC motor. The Parameters of PMDC motor are given in Table (I). The controllers was tuned to obtain the parameters of the (PID and NPID) using Particle Swarm Optimization (PSO) as shown in Table (II). Table (III) represents the parameters of the (PSO) that were used in the control units as shown in Fig.5, which represents the Cost function of the PSO that were used in the controllers of (PID and NPID).

TABLE I. THE PERMANENT MAGNET DC MOTOR PARAMETER.

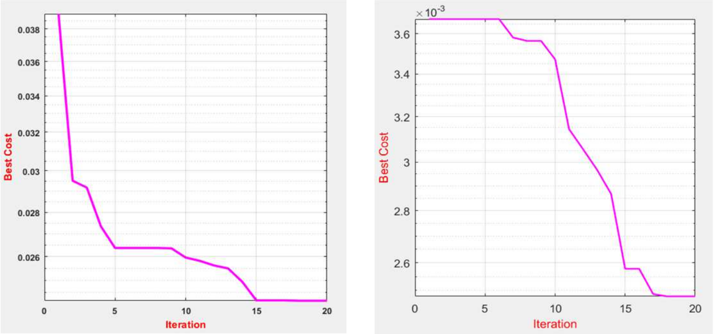
Description	*Symbols*	*Value*
Armature-winding resistance	R_a	11.27 Ω
Armature-winding inductance	L_a	0.0082 H
Torque constant	K_t	$0.00556\ \mathrm{N.m / A}$
Back EMF constant	K_B	$0.00556\ \mathrm{V.sec/rad}$
Friction constant	B_m	$6.14 \times 10^{-4}\ \mathrm{N.m.s/rad}$
Rotor inertia of the motor	J_m	$1.23\times10^{-3}\ \mathrm{kg.m^2}$
Applied armature voltage	Va	24 v
Load torque	T_l	0.01N.m
No load speed of motor	ω_m	3800 rpm

TABLE II : OPTIMAL DESIGN PARAMETERS OF NON-LINEAR PID AND TRADITIONAL PID BASED ON PSO

Parameters of NPID		**Parameters of PID**	
Symbols	*Value*	*Symbols*	*Value*
K_p	30	K_p	13.28
K_i	3.7	K_i	6.12
K_d	0.0001	K_d	0.1
α_p	0.56		
α_i	1.65		
α_d	1.6		
δ_p	0.1		
δ_i	10		
δ_d	0.004		

TABLE III: REPRESENTS THE PARAMETERS OF THE PSO

Nonlinear PID (NPID)		**Traditional PID**	
PSO Parameters	***Value***	***PSO Parameters***	***Value***
Iteration	20	Iteration	20
Swarm size	20	Swarm size	20
No. dimension	9	No. dimension	3
Weighted inertia	0.4	Weighted inertia	0.4
C_1	2	C_1	2
C_2	2	C_2	2



(a) (b)

Fig. 5. Cost function of PSO (a) Traditional PID (b) Nonlinear PID (NPID)

Simulations were held to show the motor performance under the proposed controllers to show the difference as shown in Fig.6. According to the results in MATLAB/Simulink, the NPID controller is effectively stronger than the PID controller. It is seen that NPID give better results in terms of settling time (t_s), rise time (t_r), steady-state error (e_{ss}), overshoot (M_p) and rapid response to disturbance as shown in Table (IV). The NPID controller is the best controller that has delivered adequate target results and most stable. In Fig. 7. it is shown that control signal u (t) using NPID is less overshot from that control signal u (t) using PID . Error of Speed response using NPID in Fig. 8 Also better than error of Speed response using PID .

TABLE IV: THE PERMANENT MAGNET DC MOTOR OUTPUT RESPONSE IN MATLAB SIMULINK

Method	***Settling time, t_s***	***Rise time, t_r***	***Maximum percent overshoot, M_p***	***Steady state error, e_{ss}***
Traditional PID	0.831	0.4355	0.168 %	1×10^{-6}
Nonlinear PID	0.172	0.109	0.194 %	5×10^{-7}

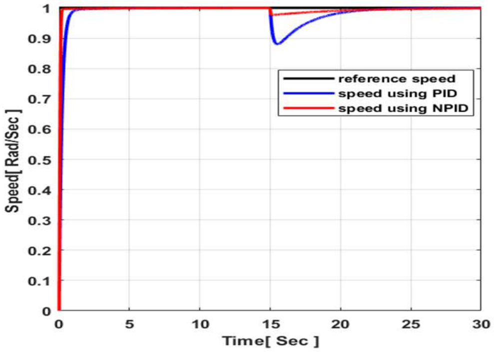


Fig. 6. Speed response of PMDC Motor under T_L=0.01Nm at 15 Sec. with Two controllers (Traditional PID and Nonlinear PID)

978-1-6654-0343-6/21/$31.00 ©2021 IEEE

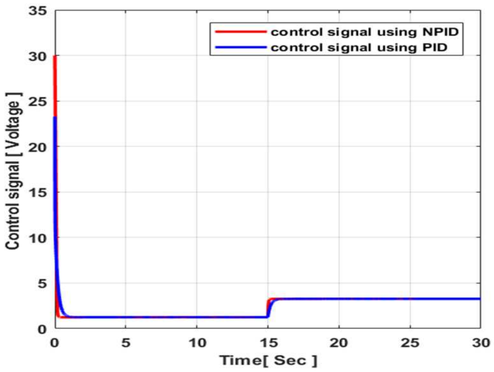


Fig.7. control signal u (t) with Two controllers (Traditional PID and Nonlinear PID)

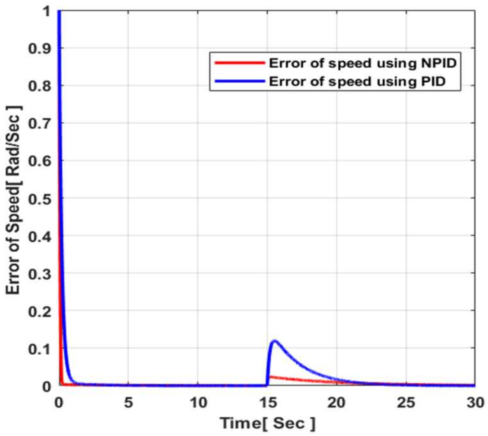


Fig.8. Error of Speed response of PMDC Motor with Two controllers (Traditional PID and Nonlinear PID)

VII . REAL-TIME IMPLEMENTATION OF TRADITIONAL PID AND NON-LINEAR PID CONTROL FOR PERMANENT MAGNET DC MOTOR BASED ON PSO TUNER

The block diagram of the hardware system is shown in Fig.9 The overall hardware setup includes the following components: a permanent magnet DC motor. An Arduino UNO interfacing device, a personal computer (PC), drive circuits, and a power supply. A permanent magnet DC motor has the parameters listed in Table (I). The control operation is achieved through software MATLAB. The MATLAB has a facility to interface with Arduino UNO. Signals generated by MATLAB-based Traditional PID and Nonlinear PID controllers are exported to drive circuits via digital (PWM) ports of Arduino UNO. The information of feedback signals are acquired via input analogue port of Arduino UNO. Fig. 11, and Fig 12 show the experimental speed behavior using Traditional PID and Nonlinear PID. According to the results as shown in Table (V), we notice that, the response speed based on Nonlinear PID Better than the response speed based on Traditional PID of terms settling time (t_s), Rise time (t_r), overshoot (M_p) and rapid response to disturbance. The hardware setup for the proposed Traditional PID and Nonlinear PID based on Particle Swarm Optimization (PSO) tuner is shown in Fig.10.

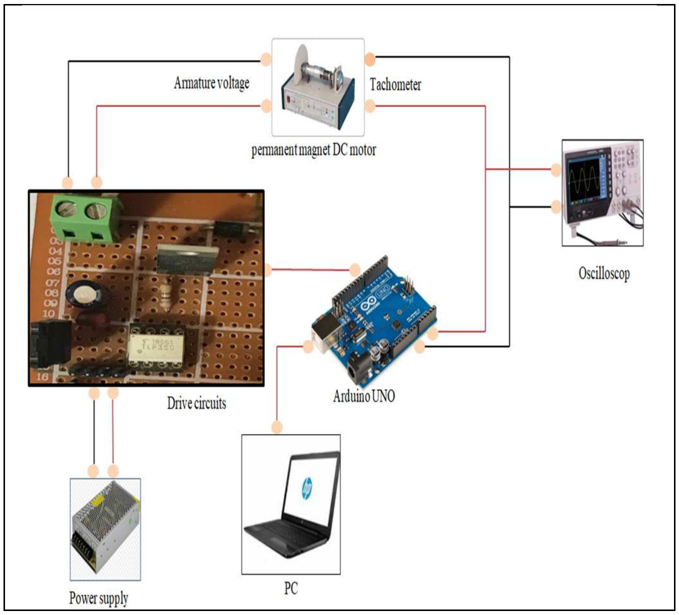


Fig.9. The block diagram of The hardware system

TABLE V: THE PERMANENT MAGNET DC MOTOR OUTPUT RESPONSE THE EXPERIMENTAL

Method	*Settling time, t_s*	*Rise time, t_r*	*Maximum percent overshoot, M_p*
Traditional PID	10	3.25	8.33 %
Non-linear PID	7	4	1.6 %

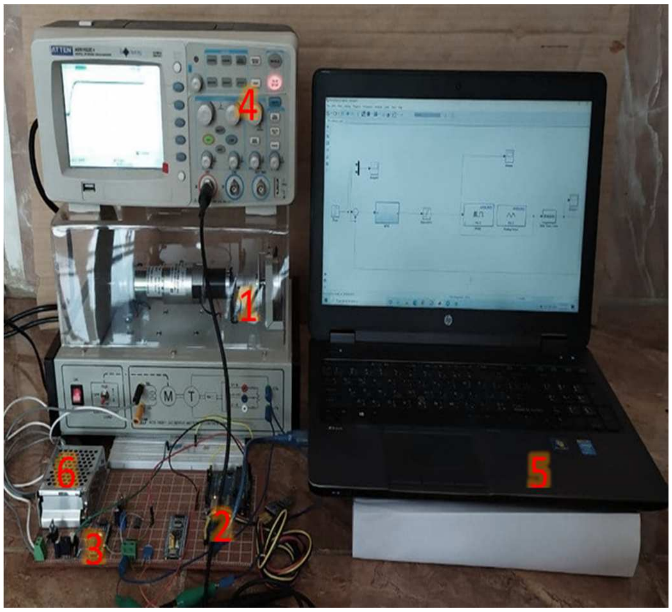


Fig.10. The hardware set-up permanent magnet DC motor Controlled by Traditional PID based on PSO and Nonlinear PID based on PSO (1) permanent magnet DC motor, (2) Arduino UNO interfacing device, (3) drive circuits, (4) digital storage oscilloscope and (5) PC, (6) Power supply.

978-1-6654-0343-6/21/$31.00 ©2021 IEEE

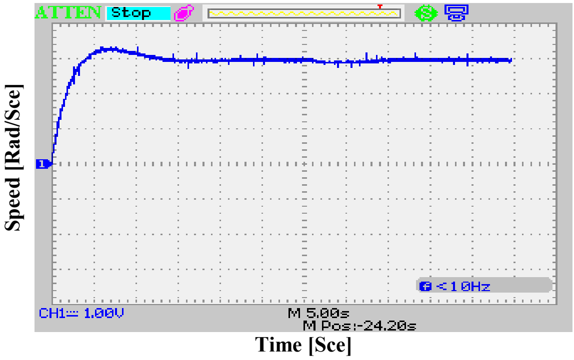


Fig. 11. Speed response of PMDC Motor under T_L=0.01Nm at 30 Sec with Traditional PID.

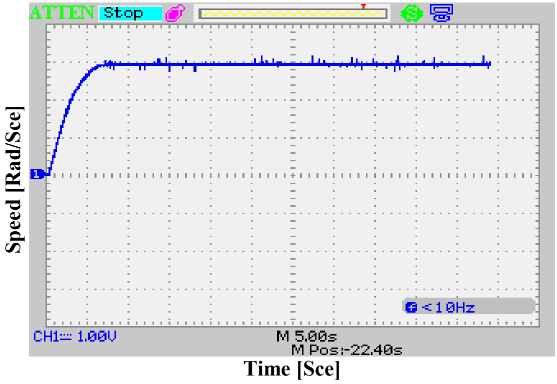


Fig. 12. Speed response of PMDC Motor under T_L=0.01Nm at 30 Sec with Nonlinear PID.

VIII . CONCLUSION

The parameters of the Traditional PID and Nonlinear PID were tuned using particle swarm optimization. to control the speed of the PMDC motor. In real condition, disturbances on PMDC motor can occur and it is difficult to be predicted. Disturbances on DC motor happened due to changes load This reduces system performance the traditional PID controller. it is needed eligible control method in order to robust tracking system against various disturbances .A Nonlinear PID controller is addressed in this study To overcome This problem for the PMDC motor speed control. according to the results of the computer simulation and The experimental. The Nonlinear PID controller is the best controller .The results prove the effectiveness of the Non-linear PID controller and it ability in forcing the motor speed to follow a desired speed . At the start of operation as well as when there is a disturbance due to changes load

.

REFERENCES

[1] Mirza Muhammad Sabir and Junaid Ali Khan " Optimal Design of PID Controller for the Speed Control of DC Motor by Using Metaheuristic Techniques" 2014 , Advances in Artificial Neural Systems , pp. 1–8, doi:10.1155/2014/126317.

[2] Rohit G. Kanojiya and P. M. Meshram, "Optimal tuning of PI controller for speed control of DC motor drive using particle swarm optimization" 2012 , International Conference on Advances in Power Conversion and Energy Technologies (APCET), Mylavaram, Andhra Pradesh, pp. 1-6, doi: 10.1109/APCET.2012.6302000.

[3] Shahram Latifi "Information Technology - New Generations" 2018 , Advances in Intelligent Systems and Computing , doi:10.1007/978-3-319-54978-1.

[4] Kartik Sharma and Dheeraj Kumar Palwalia " A modified PID control with adaptive fuzzy controller applied to DC motor " 2017, International Conference on Information,Communication,Instrumentation and Control (ICICIC), pp. 1-6, doi:10.1109/icomicon.2017.8279151.

[5] Vishakha Vijay Patel " Ziegler-Nichols Tuning Method " 2020, Resonance , pp. 1-13 , doi:10.1007/s12045-020-1058-z.

[6] Ahmad .A. Masoud , Mohammad Abu-Ali and Ali Al-Shaikhi "Experimental Determination of an Extended DC Servo-Motor State Space Model An Undergraduate Experiment " 2019 , IEEE Access, pp. 1-16, doi: 10.1109/ACCESS.2019.2962612.

[7] M. S. Qureshi, P. Swarnkar and S. Gupta, "Assessment of DC servo motor with sliding mode control approach" 2016 , IEEE First International Conference on Control, Measurement and Instrumentation (CMI) , pp. 1-5, doi: 10.1109/CMI.2016.7413769.

[8] R. Rahmani, M. S. Mahmodian, S. Mekhilef and A. A. Shojaei "Fuzzy logic controller optimized by particle swarm optimization for DC motor speed control " 2012 , IEEE Student Conference on Research and Development , pp. 1-5, doi: 10.1109/Scored.2012.6518621.

[9] A. A. A. El-Gammal and A. A. El-Samahy, "A modified design of PID controller for DC motor drives using Particle Swarm Optimization PSO" 2009 , International Conference on Power Engineering, Energy and Electrical Drives, Lisbon , pp. 1-6 , doi: 10.1109/Powereng.2009.4915157.

[10] Yuhui Shi and Russell C. Eberhart "Parameter Selection in Particle Swarm Optimization" 1998 , Proc. Seventh Annual Conf. on Evolutionary Programming, pp. 1-10.

[11] 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 “ 2020 International Journal of Advanced Robotic Systems , pp. 1-17 , doi: 10.1177/1729881420981524.

[12] Ramya M.A , Sharad P. Jadhav and Sushant N. Pawar , "Design and Implementation of Particle Swarm Optimization (PSO) Tuned PID Controller for Speed Control of Permanent Magnet Brush Less DC (PMBLDC) Motor" 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-6, doi: 10.1109/INCET49848.2020.9154039.

[13] S.Durgadevi , K.Thirupura Sundari , Raaghavi.D and R S Akshaya "Comparative study of Controller Optimisation for CSTR using Particle Swarm Optimization Technique " ICEES 2019 Fifth International Conference on Electrical Energy Systems , pp.1-5, doi.org/10.1109/ICEES.2019.8719302

[14] Amjad J. Humaidi and Ahmed Ibraheem Abdul kareem, "Design of Augmented Nonlinear PD Controller of Delta/Par4-Like Robot" 2019 Journal of Control Science and Engineering, pp.1-11, doi:10.1155/2019/7689673 .

[15] Guy Zaidner, Sammy Korotkin, Eli Shteimberg, Amir Ellenbogen, Meir Arad and Yosef Cohen "Non Linear PID and its application in Process Control " 2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel, pp.1-4 , doi:10.1109/EEEI.2010.5662155.

[16] Dr. Shibly Ahmed Al-Samarraie , Bashar Fateh Midhat and Ivan Isho Gorial “Nonlinear integral control design for DC motor speed control with unknown and variable external torque ” 2016, Journal Engineering and sustainable Department , pp.1-15 , ISSN 2520-0917

[17] Aws Abdulsalam Najm and Ibraheem Kasim Ibraheem "Nonlinear PID controller design for a 6-DOF UAV quadrotor system ” 2019 , Engineering Science and Technology, an International Journal , pp.1-11, doi.org/10.1016/j.jestch.2019.02.005