International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 24 (2017) pp. 15999-16007
© Research India Publications. http://www.ripublication.com

Comparative Study of Conventional and Optimal PID Tunned Methods for PMDCM Speed Control

Ali A. Hassan [1], **Nabil K. Al-Shamaa** [2]
[1,2] Department of Computer Engineering, Electrical Engineering Technical College,
Middle Technical University, Baghdad, Iraq.

[1]Orcid: 0000-0003-2093-5383, [2]Orcid: 0000-0002-5134-7464,

Kasim K. Abdalla[3]
[3]Department of Electrical Engineering, College of Engineering, University of Babylon, Babylon, Iraq.

[3]Orcid: 0000-0002-9456-1479

Abstract

The purpose of this work is to design and implementation of a speed controller for a Permanent Magnet Direct Current (PMDC) motor by setting of the suitable PID controller parameters using Ziegler-Nichols conventional technique (ZN) and Particle Swarm Optimization (PSO). The controlling process of PMDC motor is difficult and mathematically tedious because of non-linearity property. The conventional gain tuning of PID controller (such as Ziegler-Nichols method) usually produces a big overshoot, to overcome this difficulty a PSO is employed to improve the capability of conventional techniques. The PMDC motor is classified as a second order system for armature voltage control technique of speed control. In this study, the parameters of the targeted plant have been derived practically as well as obtained the transfer function and built the mathematical model which the tuning algorithms have been applied to. The simulation and practical results show that the PSO technique has more advantages over ZN tuning method, which they simulated by using MATLAB/Simulink and experimented by using Atmega328 controller, PMDC motor (YA-070) and IBT-2 driver.

Keywords: Ziegler-Nichols, Particle Swarm Optimization and PID controller tuning

INTRODUCTION

In control systems, the most well-known and highly efficient is the PID controller. The structure of PID-controller is simple and easy to handle by plant operators, which be relatively easy to tune and have effectively improved. PID control strategy has been used for many several decades since 1940. Most of the advantages that made PID-controller which is very popular academically, industrially and productivity processes are a wide-range of a control application, simplicity and near-optimal performance. PID controller could be used as a single control unit or part of a control system. Tuning procedure should be made for the controller after the implementation of the controller to ensure achieving the best performance to a controlled plant. Industrial process control has experienced numerous advances in the recent decades for the designing of controller and its method of implementation [1]. Despite of these advances, conventional PID controller is undoubtedly the most common controller in the industrial processes because of its simplicity in structure and robust performance in various operating conditions [2].

PID CONTROLLER

If the constant values of the controller system are not accurate, the performance of the control system is poor in properties and/or they become unstable [3]. Therefore, it is important to choose the proper tuning constants for tuning the controller to perform a good control and optimal performance. The major purpose of the controller is to use a certain algorithm to keep the output in the desired range. PID controller includes the proportional (P), integral (I) and derivative (D) parameters which they are set by using a certain tuning algorithm [4, 5], where the (P) parameter is responsible for the desired set-point, (I) parameter is responsible for accumulating the recent errors and (D) parameter is responsible for determining the rate of change of error of the plant, [6]. Figure 1 shows the PID Controller model structure.

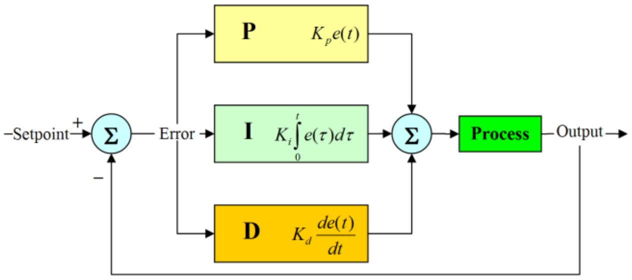


Figure 1: PID Controller model structure

The PID controller transfer function is presented in Eq. (1).

$$U(t) = Kp\, e(t) + Ki \int e(t)\, d(t) + Kd \frac{de(t)}{dt} \quad (1)$$

PID controller performance could be effectively obtained with suitable tuning gains for its parameters. J. G. Zeigler and N. B. Nichols submitted the first simplified tuning approach for PID controller [6, 7]. Zeigler-Nichols and Particle Swarm Optimization PID controller-tuning methods have been studied and discussed in this paper.

Dc motor

The electrical motors are electro-mechanical energy conversion devices. DC motor (direct current motor) is the targeted plant. It converts electricity and magnetic field to producing motoring torque which acts to rotate the armature around the shaft center. DC motors are very suitable for the application that required adjustable, frequent starting, good speed regulation, braking and reversing, such as in mills equipment's, paper factories, electro-mechanic machine-tools, load traction, press printing house, textile factories, elevators, electric driven railway [8, 9]. PMDC motors are commonly used with batteries or solar cells energy sources, which provide portability and thus provide cost effective solution, during the lack of presence AC power supply in every place such as electric hybrid cars, elevators, car windows, robotic-arms, etc. Fractional horsepower is mostly used for positioning purpose and tracking systems as closed loop servo motor. DC motor responses related to both source voltage and armature current. The industrial, robotics and electro-mechanical applications have major need for speed control task. In order to maintain constant speed for the required task different types of controllers could be used [10].

Mathematical Modeling Construction

The DC motor has consolidated mathematical models presented by scientific community [11-13], generally, the models are composed by two parts: electrical and mechanical parts. The electrical equations the main parameters are armature resistance (R_a) and inductance (L_a). The main parameters in mechanical equations are inertia moment (J_m) and the friction coefficient (B_m).

The equations below represent the characteristics of the PMDC:

$$V_a = R_a I_a(t) + L_a \frac{d\,Ia}{dt} + E_b\,(t) \quad (2)$$

$$E_b = K_b\, \omega(t) \quad (3)$$

$$T_m = K_t\, I_a(t) \quad (4)$$

$$T_m\,(t) - T_l\,(t) = J_m \frac{d\,\omega}{dt} + B_m\, \omega\,(t) \quad (5)$$

where, V_a, R_a, La, I_a, represent the voltage, resistance, inductance and current of the armature respectively, E_b represents Back emf, ω is the angular speed, T_m is the motor torque, T_l is the load torque, J_m is the rotor inertia, B_m is the viscous friction coefficient, K_t is the torque constant, K_b is the Back emf constant.

By substituting (3) in (2), (4) in (5) and T_l= 0, the resultant equations are represented in Eq. (6) and Eq. (7).

$$V_a = R_a I_a(t) + L_a \frac{d\,Ia}{dt} + K_b\, \omega(t) \quad (6)$$

$$K_t\, I_a(t) = J_m \frac{d\,\omega}{dt} + B_m\, \omega\,(t) \quad (7)$$

The speed transfer function of the model is written in Eq. (8), [14].

$$\frac{\omega(S)}{Va(S)} = \frac{K_t}{L_a J_m S^2 + (R_a J_m + L_a B_m)S + (R_a B_m + K_t K_b)} \quad (8)$$

By integrating Eq. (8), the angular position is derived and written in Eq. (9). [15]

$$\frac{\theta(S)}{Va(S)} = \frac{Kt}{La\, Jm\, S^3 + (Ra\, Jm + La\, Bm)S^2 + (Ra\, Bm + Kt\, Kb)S} \quad (9)$$

where θ is the angular position of rotor shaft.

By using the equations (8) and (9) the mathematical model of PMDC motor can be built by Matlab Simulink [16] as shown in Figure2

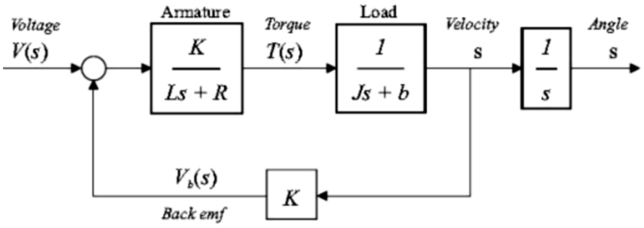


Figure 2: Dc motor mathematical model structure

Ziegler and Nichols tuning method

The Ziegler-Nichols closed-loop tuning method provides using the ultimate gain value (*Ku*), and the ultimate period of oscillation (*Pu*). Even though this tuning method was devised in 1940 [17], it is still one of the most widely used methods of tuning a PID controller due to its applicability to almost all the processes irrespective of its order.

K_u and P_u values for the system have been calculated by using Matlab Simulink. The measured values should be applied in Table 1 to get the controller parameters.

Table 1: Ziegler-Nichols Tuning Rule Based

Controller type	K_p	T_i	T_d
P	$K_u/2$		
PI	$0.45K_u$	$P_u/1.2$	
PID	$0.6\ K_u$	$P_u/2$	$P_u/8$

Particle swarm optimization (PSO)

Particle Swarm Optimization (PSO) is an evolutionary-type global optimization technique, which has been invented by Kennedy and Eberhart [18]. It is inspired by the social behavior of flocking of birds and schooling of fish. Various engineering problems have been considered it as a superior technique based on its high computational efficiency [19- 27]. PSO approach featured with a capability of the effective search for various optimized solutions of engineering problems, with quickest solutions, higher convergence degree of stability and the initiation of parameters which is assigned is less compared with other technique of optimization which inspired by population, like genetic-algorithm (GA) and ant-colony optimization (ACO). It can be considered as a powerful optimization approach in system parameter for identifying the PID controller gains [28]. This technique starting with an artificial swarm group such as a bird, it begins by initiation of random positions X_i and velocities V_i, and randomly dispersed inside D dimensional search space. Based on Objective Function (OF) supervision, their own flying experience, and flying experience of their companions, every swarm's particle in the group adjusts its flight position and its velocity dynamically. Every particle keep remembering its best position represented as (p_{best}—($P_{i,n}^t$)) and also it obtains the global best position datum which is achieved by any particle in the group of population as (g_{best}—($G_{i,D}^t$)) during the optimization searching process. In the (t) iteration, every particle ( ) which has personal position defined as below:

$X^t_{i,n}= [X_{i,1}, X_{i,2}, . . . , X_{i,D}]$

and it's velocity which is defined as:

$V^t_{i,n}=[V_{i,1}, V_{i,2}, . . . , V_{i,D}]$ in D dimensional searching space.

Therefore, the velocity and the next iteration particle's position could be found as:

$$V^{t+1}_{i,n}= W*V\ ^t_{i,D}+ C1*\ R1*\ (P^t_{i,D}-X^t_{i,D})\ +\ C2*\ R2*\ (G^t_{i,n}-X^t_{i,D}) \quad (10)$$

$$X^{t+1}_{i,n}= X^t_{i,D}+ V^{t+1}_{i,D} \quad (11)$$

where i = 1, 2, . …. , N, $C1$: acceleration coefficient which mentioned as a cognitive coefficient, it is responsible for pulling every particle in the direction of local best position, $C2$: a coefficient called social parameter which is responsible for pulling the particle in the direction of the global best position. $R1$ and $R2$ are randomly generated numbers with 0-1 range. W: the inertia of weight which is considered as a significant factor responsible for the convergence of PSO algorithm and controlling the influence of prior velocities on the instant velocity at the current time step [29]. Figure 3 represents the parameters searching for PSO algorithm.

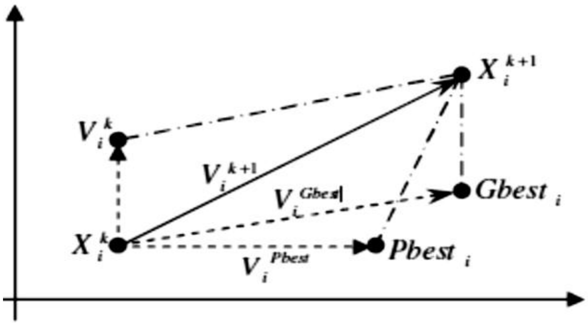


Figure 3: Parameter Searching by PSO algorithm.

where X^k : representing the current position

X^{k+1}: representing the new (modified) position

V^k: representing the current velocity

V^{k+1}: representing the new (modified) velocity

V^{Pbest}: representing the velocity based on P_{best}

V^{Gbest}: representing the velocity based on G_{best}

The PSO algorithm has to meet the following consequent steps:

A. Swarm initialized with N size population, the particles are generated between the minimum and maximum limits of parameter values randomly.
B. Objective function values for particles have been evaluated using the performance criteria for algorithm convergence.
C. The objective values which obtained above for the initial particles of the swarm are set to be the initial p_{best} values of particles, while the best value among all the p_{best} values will be identified as a g_{best}.
D. The new modified velocity for each particle is computed by using Eq. (10).
E. The position of particle is updated according to Eq. (11). However the objective function values are calculated for updated positions of particles, if the new value is better than the previous p_{best}, the new value is set to be p_{best}. Similarly, g_{best} value is also updated as the best p_{best}.
F. If the stopping criteria are met, the positions of particles represented by p_{best} are considered as the optimal values. Otherwise, the above procedure is repeated from Step D until the specified iteration is completed as shown in

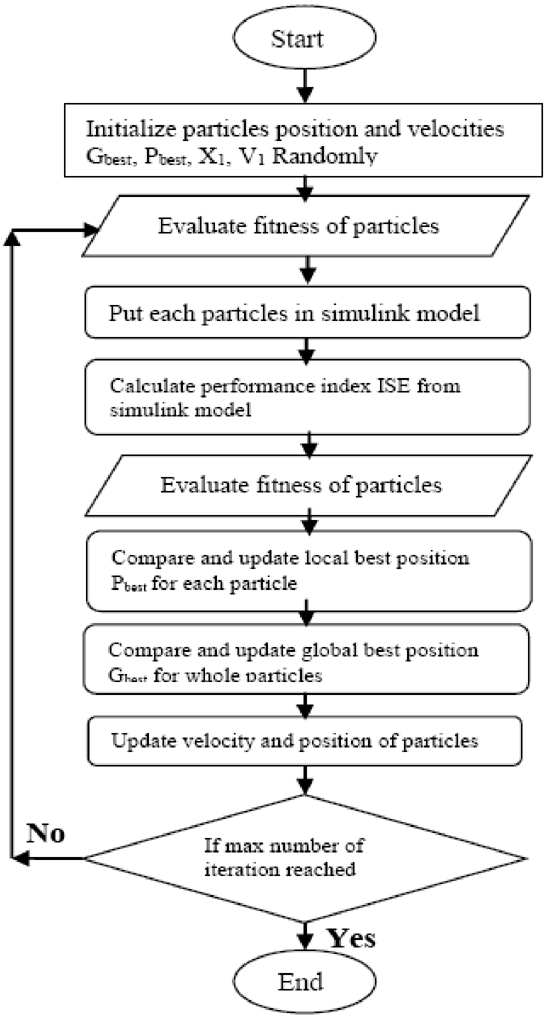


Figure 4: Flow chart process of Algorithm of PSO

The performance, efficiency and accuracy of the process to be controlled by PSO essentially are depending on Objective Function (OF), which is the function that monitors the optimization searching process. The objective function is initiated on assumption that at least there is one set for the optimal solution parameters exists in a universe "U" which is ensure that all the constraints will be satisfied Eq. (12).

$$\mathrm{OF} = \min_{D\in U} J(D) \qquad (12)$$

The controller design problems are mostly considered as a multi objective (J). Multi objective optimization always provides developed results compared with a single-objective function as presented in Eq.(13).

$$\min_{D\in U}(\phi 1(p), \phi 2(p), \ldots, \phi n(p)) \qquad (13)$$

The researchers are widely adopted the weighted sum, which is very helpful for converting the multi objective problems regarding minimizing the objectives into a scalar form [29] which is presented as in Eq. (14).

$$\min_{D\in U} \sum_{k=1}^{n} W_k J_k\,(D) \qquad (14)$$

where W_k is representing the weighting factor which is considered to be (between 0 to 1 or 0 to 100%). It is basically assigned according to the preference constraints that have to be minimized. In the control references, there are a number of weighted sum existing which they based on an objective functions [30]. Eq. (15) represent two-parameters depending on objective function with considering the error and overshoot (Mp) which they are very significant in time domain constraints.

$$J\,(\alpha) = w1\cdot \mathrm{ISE} + w2\cdot Mp, \qquad (15)$$

$$\alpha = [KpKiKd]$$

while α represents parameters that required to be optimized, Mp represents peak overshoot and weighting function

$w1 = w2 = 10$.

The variety of optimization problems suffers from the difficulty of satisfying all the required constraints, therefore negotiation have to be done between the preference constraints parameters without any compromising for the domain constraints [30]. Otherwise, the algorithm constantly varies the controller gain parameters till the objective function J minimized to $J_{\min}$ [31, 32].

Calculation of Target Plant Parameters

The parameters of the targeted plant (PMDC motor) will be identified practically; the following procedures have been performed on the selected motor to find the parameters of motor transfer function practically:

1) To find armature resistance (R_a), DC variable voltage supply is applied to the PMDC motor. The DC voltage increased gradually till the armature shaft about to rotate, then measure the current drawn by the armature using ampere meter, the armature resistance is given below:

$$R_a = \frac{\mathrm{V(supply)}}{I_a\mathrm{(armature\ current)}} \qquad (16)$$

2) To find armature inductance(L_a) AC variable voltage supply is connected to the motor terminal then increase the applied voltage till the shaft about to oscillate, the drawn current is measured, then the impedance Z can be calculated as:

$$\text{Z (impedance)} = \frac{\mathrm{V(supply)}}{I_a\mathrm{(armature\ current)}} \qquad (17)$$

Then inductance $Z = \sqrt{(R_a)^2 + (\omega L_a)^2}$ (18)

where, $\omega = 2\pi f$.

3) To find K_b (back EMF constant), the motor has been rotated as a generator using another shaft-coupled motor and record the generated voltage at different speeds and take the average then calculate: $K_b = \frac{E}{n}$ (19)

where, E=the generated voltage (volt) ,n=armature speed (rpm).

4) To find K_t we use the relations below:

$$K_t\left(\frac{lb-in}{amp}\right) = \frac{Kb}{0.011827}\left(\frac{volts}{rpm}\right) \tag{20}$$

$$K_b\left(\frac{volts}{rpm}\right) = \frac{E}{n} = 0.00684\frac{T}{I} \tag{21}$$

$$\text{where } \frac{T}{I} = K_t\left(\frac{lb-in}{amp}\right) \tag{22}$$

$$K_t = \frac{Ke\ (v-\sec/rad\ )}{0.00684\ x9.554} \tag{23}$$

$$K_t\left(\frac{lb-in}{amp}\right) = 15.3\ K_b\left(\frac{volt-sec}{rad}\right) \tag{24}$$

5) To find inertia (J_m)

$$J_m\,(Kgm^2) = \frac{acceleration\ torque}{acceleration} \tag{25}$$

where , Acceleration torque = max current * K_t (26)

Acceleration = the slope of the speed curve

The max current could be found using the Current sensor ACS712 the output Current show in Figure 5.

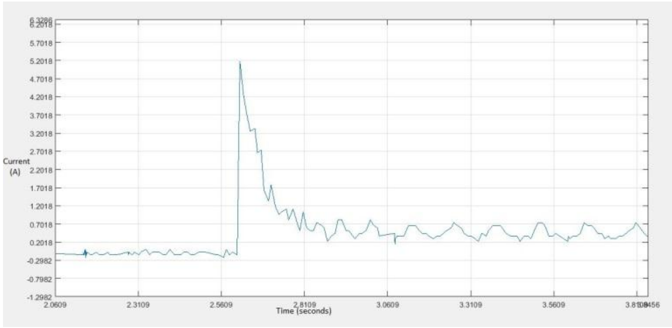


Figure 5: Max Current drawn by current sensor ACS712 for the PMDC motor

The acceleration curve could be found by using the tacho generator attached to motor shaft, as shown in Figure 6.

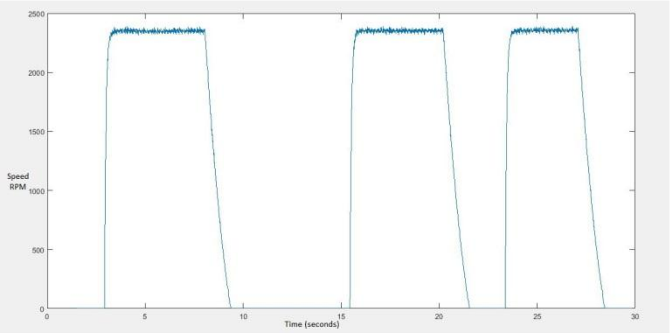


Figure 6: Acceleration curve for the PMDC motor

The final calculated parameters of the modelling PMDC motor are presented in Table 2.

Table 2: Calculated parameters of PMDC motor

Armature resistance (R_a)	7	ohm
Armature inductance (L_a)	0.008436	Henry
Torque constant (K_t)	0.094	Nm/A
Back emf constant (K_b)	0.094	v/rad/sec
Rotor inertia (J_m)	2.2097e-04	Nm/rad/ sec²
Viscous friction constant(B_m)	1e-05	Nm/ rad/ sec

Matlab Model and Simulation Results

The DC motor and PID controller model using Matlab Simulink is illustrated in Figure 7.

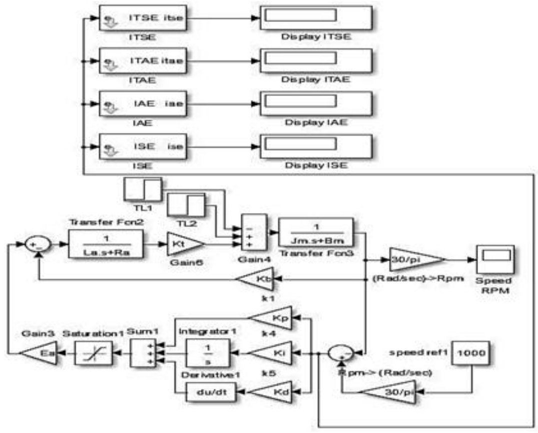


Figure 7: PID controller structure connected with PMDC Motor

In order to apply the Zeigler-Nichols tuning method it required that the PID controller gains K_i and K_d have to be set to zero and set K_p to an ultimate value k_u to sustain oscillation according to Eq. (9). K_u represents a suitable k_p which sustains oscillation at the output. P_u is the corresponding period for any symmetrically consequent points of the oscillation output as shown in Figure 8.

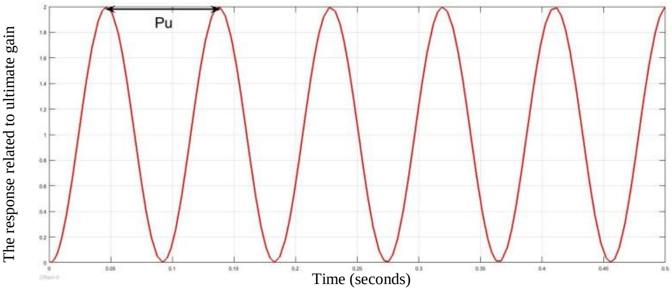


Figure 8: Critical gain K_u and corresponding ultimate period P_u

According to Figure 8 the ultimate gain K_u= 8.178, and ultimate period P_u= 0.0913, then applying these values to Table 1 to calculate the required PID controller gains.

The PSO tuning algorithm required to be connected according to Figure 9. The generic equation of the proposed fitness is based on minimizing the integral of square error (ISE) as in Eq. (16) which is presented as block e2 in Figure 9.

$$\text{ISE} = \int_0^t e^2(t)dt \tag{27}$$

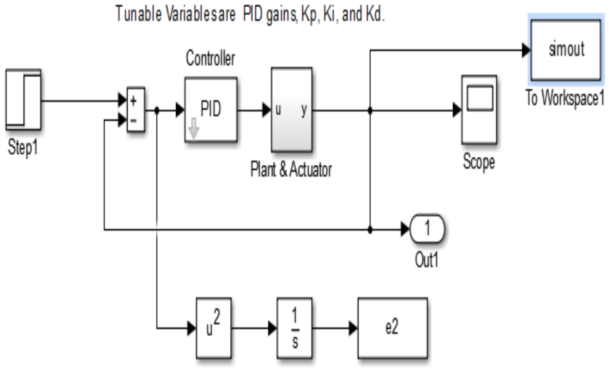


Figure 9: Tunable PID Controller using PSO algorithm

The fitness function has been considered based on time domain characteristics for adaptation. The number of adaptation iterations has been set based on expected parameters and time of computation as shown in Figure10.

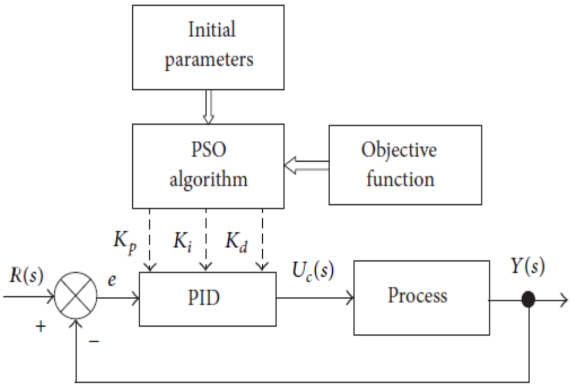


Figure 10: The illustration of PSO algorithm for PID Controller tuning Simulink model

By applying the initial parameters are set for PSO algorithm as below:

Swarm size "number of birds "= 50

Iterations "birds steps"=50

PSO coefficient c1=1.2

PSO coefficient c2 =0.12

Dimension of the problem= 3

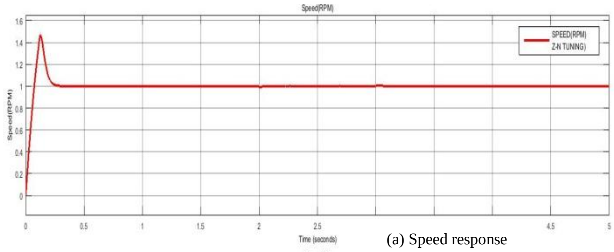
PSO momentum or inertia =0.9

Table 3 shows the controller tuning gains for ZN and PSO.

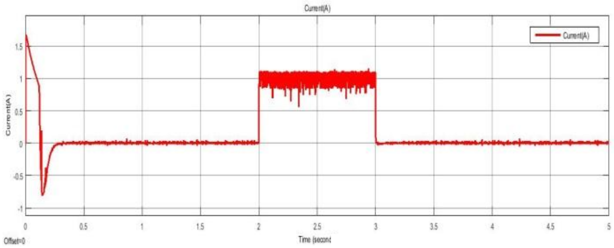
Table 3: Tuning Gains For ZN and PSO

Tuning method	PID parameters	Kp	Ki	Kd
Ziegler-Nichols Closed loop		4.9068	107.487	0.05599
PSO		2.80423878299188	0.766301702932533	0.0128807266480534

By applying these gains in PID controller for the system model Figure 7, the output system response for Zeigler-Nichols and Particle Swarm Optimization are plotted in Figure 10, Figure11 respectively.

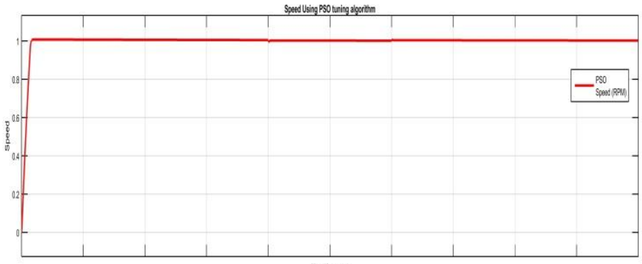


(a) Speed response

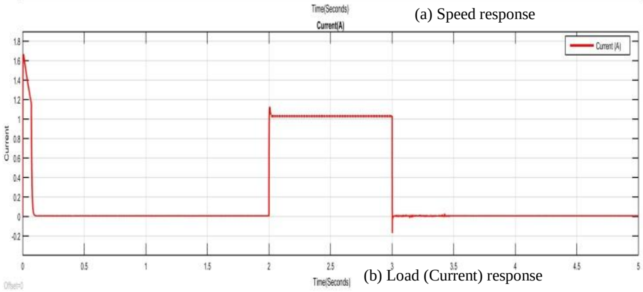


(b) Load (Current) response

Figure 10: Simulation responses of ZN method (a) Speed response (b) Load (Current) response



(a) Speed response



(b) Load (Current) response

Figure 11: Simulation responses of PSO method (a) Speed response (b) Load (Current) response

By using measurement and Bi-level instrument tools in the real time scope, the response analysis for different criteria has been found such that rise time "Tr", overshoot "Mp", settling time "Ts" and steady state error "Ess" as shown in Table 4.

Table 4: Response Analysis Comparison

Tuning method	Tr (s)	Ts (s)	Mp (%)	Ess (%)
Ziegler-Nichols Closed loop	0.0933	0.384	48.507	0.000833
PSO	0.058	0.082	0.505	0.002

The performance indices (integral squared error (ISE), integral absolute error (IAE), integral time squared error (ITSE) and integral time absolute error (ITAE)) for ZN and PSO are presented in Table 5.

Table 5: The performance indices of ZN and PSO tuning methods

Tuning method	ISE	IAE	ITAE	ITSE
ZN	372.5	7.524	1.064	20.44
PSO	252.6	5.054	3.427	5.308

Practical Implementation

The practical circuit implementation consists of microcontroller Atmega328, PMDC, IBT-2 motor driver that control the speed by using pulse width modulation (PWM) and tacho-generator which works as a feedback device that coupled to motor shaft for measuring the actual motor speed, which plot the corresponding system responses for each proposed tuning method as illustrated in Figure12. The gains in Table 3 have been applied to the practical circuit.

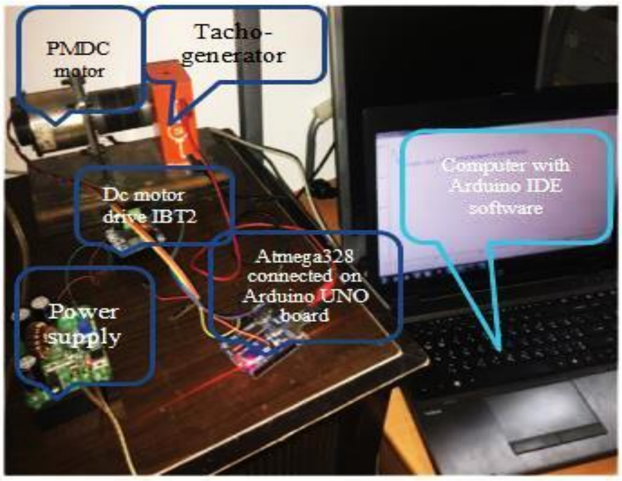


Figure 12: Practical implementation system

The speed response for the system using Zeigler-Nichols Closed loop tuning method gains which is presented in Table 3 and illustrated in Figure 13.

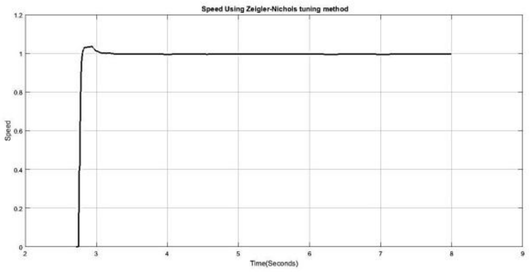


Figure 13: The response of speed for practical ZN method

By applying the real-time PSO optimization algorithm, the optimization results have been found and the results are presented in Table 6.

Table 6: PSO real-time Tuning Gains

Tuning method	PID parameters	Kp	Ki	Kd
PSO		2.0539	0.0331	4.6743

The output speed response is shown in Figure 14.

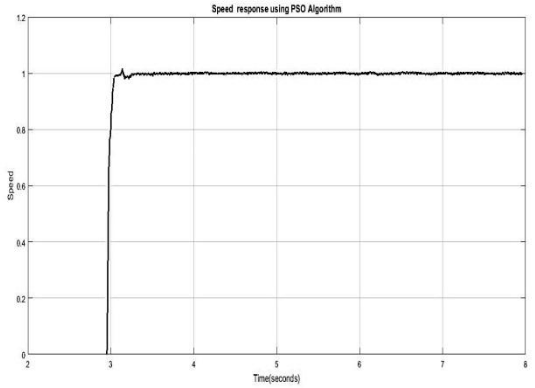


Figure 14: The practical result of PSO method for PID Controller tuning

COMPARATIVE STUDY FOR RESULTS

A comparative study is made for the designed PID controller with Zeigler-Nichols (ZN) and PSO method according to different criteria such as rise time (Tr), overshoot (Mp), settling time (Ts), steady state error (Ess), Table 4 shows that the PSO tuning method more suitable than the ZN tuning method, where the Tr, Ts, and Mp of PSO less than ZN, but Ess in PSO is more than in ZN. From Table 5 the performance indices of ISE, IAE, and ITSE of PSO are better than ZN, while ITAE of ZN is better than PSO. The practical responses of the PSO and ZN are approximately similar to each other, only more overshoot in ZN Therefore the results show that the PSO tuning method is better than the ZN for the PMDC motor.

CONCLUSIONS

A speed controller of a Permanent Magnet Direct Current (PMDC) motor by a setting of the suitable PID controller parameters using Ziegler-Nichols conventional technique (ZN) and Particle Swarm Optimization (PSO) have been simulated by Matlab and implemented practically by using Atmega328 controller, PMDC motor (YA-070) and IBT-2 driver. The simulation and practical comparison between these tuning methods has been made, the results show that the PSO tuning method more suitable than the ZN tuning method. The performance indices of ISE, IAE, and ITSE of PSO are better than ZN, while ITAE of ZN is better than PSO. The practical responses of the PSO and ZN are approximately similar to each other, only more overshoot in ZN.

ACKNOWLEDGMENT

I would like to present a greatest appreciation to Mr. Ali Shaaban (lecturer), DC machines Laboratory members (Department of Electrical Engineering, College of Engineering, University of Babylon), Mr. Ahmed Alwash and Mr. Haider hadi (Scientific Activity Center, Hilla, Babel, Iraq) for their tremendous support and help.

REFERENCES

[1] K. J. Astrom, Process Control-Past, Present and Future Control Systems Magazine, IEEE, Vol. 5, No. 3, pp. 3-10, August, 1985.

[2] K. J. Astrom and T. Hagglund, Revisiting the Zeigler-Nichols Step Response Method for PID Control, Journal of Process Control, Vol. 14, pp. 635-650, 2004.

[3] M. Zhuang and D.P. Atherton, Automatic tuning of optimum PID controllers II, lEE Proceedings-D, Vol. 140, No. 3, May 1993.

[4] Vishal Vishnoi, Subhransu Padhee, Gagandeep Kaur, Controller Performance Evaluation for Concentration Control of Isothermal Continuous Stirred Tank Reactor, International Journal of Scientific and Research Publications, Volume 2, Issue 6, June 20 12.,

[5] Khanduja, Neha, and Bharat Bhushan. Intelligent Control of CSTR using IMC-PID and PSO-PID controller. In Power Electronics, Intelligent Control and Energy Systems (ICPEICES), IEEE International Conference on, pp. 1-6. IEEE, 2016.

[6] J. G. Zeigler and N. B. Nichols, Optimum Settings for Automatic Controllers, Trans. ASME, Vol. 64, pp.759-768, 1942.

[7] Vishal, Vikrant, Vineet Kumar, K. P. S. Rana, and Puneet Mishra. Comparative study of some

optimization techniques applied to DC motor control. In Advance Computing Conference (IACC), 2014 IEEE International, pp. 1342-1347. IEEE, 2014.

[8] K. Hameyer, R. J. M. Belmans, "Permanent Magnet Excited Brushed DC Motors", IEEE Tranactions on Industrial Electronics, vol.43, no.2, April 1996, pp. 247-255.

[9] Galijašević, Sanel, Šemsudin Mašić, Senad Smaka, Abdulah Akšamović, and Dinko Balić. "Parameter identification and digital control of speed of a permanent magnet DC motors." In Information, Communication and Automation Technologies (ICAT), 2011 International Symposium on, pp. 1-7. IEEE, 2011.

[10] W. G. M. Elnaimand S. F.Babiker ,Comparative Study on the Speed of DC Motor using PID and FLC, Conf. of Basic Sciences and Engineering Studies (SGCAC), Khartoum, Sudan, 2016

[11] _Kostenko M, Piotrovski L. Electrical Machines, vol. 1 and 2. Mir Publishers: Moscou, 1968.

[12] Smolensky, A. Ivanov. Electrical machines. Mir, 1983.

[13] Franklin, Gene F., J. David Powell, and Abbas Emami-Naeini. "Feedback control of dynamics systems." Pretince Hall Inc (1986).

[14] Jalilvand, A., A. Kimiyaghalam, A. Ashouri, and H. Kord. "Optimal tuning of PID controller parameters on a DC motor based on advanced particle swarm optimization algorithm." optimization 11 (2011), pp.12.

[15] K. K.Abd-AllaAlrobaie, *DC motor speed controller employing FPGA*. Master Thesis, University of technology, Baghdad, Iraq, Oct., 2004.

[16] Geerththanna, A., and Sarat Kumar Sahoo. "A comparative study of pv fed speed control of dc motor using soft computing techniques." (2017).

[17] Ziegler, G. and Nichols, N. B, 1942.Optimum settings for automatic controllers, Trans. ASME, 64,759-768.

[18] J. Kennedy and R. C. Eberhart, "Particle swarm optimization, "in Proceedings of the IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948, 1995.

[19] U. S. Banu and G. Uma, "Fuzzy gain scheduled continuous stirred tank reactor with particle swarm optimization based PID control minimizing integral square error," Instrumentation Science and Technology, vol. 36, no. 4, pp. 394–409, 2008.

[20] M. S. Arumugam and M. V. C. Rao, "On the performance of the particle swarm optimization algorithm with various inertia weight variants for computing optimal control of a class of hybrid systems," Discrete Dynamics in Nature and Society, vol. 2006, Article ID 79295, 17 pages, 2006.

[21] M. S. Arumugam and M. V. C. Rao, "On the optimal control of single-stage hybrid manufacturing systems via novel and different variants of particle swarm optimization algorithm," Discrete Dynamics in Nature and Society, vol. 2005, no. 3, pp.257–279, 2005.

[22] R. M. Chen and C. M. Wang, "Project scheduling heuristicsbased standard PSO for task-resource assignment in heterogeneous grid," Abstract and Applied Analysis, vol. 2011, Article ID 589862, 20 pages, 2011.

[23] R. F.Abdel-Kader, "Particle swarmoptimization for constrained instruction scheduling," VLSI Design, vol. 2008, Article ID 930610, 7 pages, 2008.

[24] C. H. Lin, J. L. Chen, and Z. L. Gaing, "Combining biometric fractal pattern and particle swarm optimization-based classifier for fingerprint recognition," Mathematical Problems in Engineering, vol. 2010, Article ID 328676, 14 pages, 2010.

[25] J. Zhang and K. W. Chau, "Multilayer ensemble pruning via novel multi-sub-swarm particle swarm optimization," Journal of Universal Computer Science, vol. 15, no. 4, pp. 840–858, 2009.

[26] K. W. Chau, "Application of a PSO-based neural network in analysis of outcomes of construction claims," Automation in Construction, vol. 16, no. 5, pp. 642–646, 2007.

[27] K. Chau, "Predicting construction litigation outcome using particle swarm optimization," in Innovations in Applied Artificial Intelligence, vol. 3533 of Lecture Notes in Computer Science, pp.571–578, Springer, New York, NY, USA, 2005.

[28] A. Alfi and H. Modares, "System identification and control using adaptive particle swarm optimization," Applied Mathematical Modelling, vol. 35, no. 3, pp. 1210–1221, 2011.

[29] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in Proceedings of the IEEE World Congress on Computational Intelligence, pp. 69–73, 1998.

[30] G. P. Liu, J. B. Yang, and J. F. Whidborne, Multiobjective Optimization and Control, Printice Hall,NewDelhi, India, 2008.

[31] Latha, K., V. Rajinikanth, and P. M. Surekha. "PSO-based PID controller design for a class of stable and unstable systems." ISRN Artificial Intelligence 2013 (2013).

[32] Anupam Aggrawal, Akhilesh Kumar Mishra, Abdul Zeeshan,"Speed Control of DC Motor Using Particle Swarm Optimization Technique by PSO Tunned PID and FOPID", International Journal of Engineering Trends and Technology (IJETT) – Vol 16 Number 2 – Oct 2014, pp. 72.