JEEE-U
Journal of Electrical and Electronic Engineering-UMSIDA
ISSN 2460-9250 (print), ISSN 2540-8658 (online)
Vol. 8, No. 2, October 2024
10.21070/jeeeu.v8i2.1701

DC Motor Speed Control using Particle Swarm Optimization based on Labview

Rifqi Firmansyah[1]*, Muhammad Badruddin A. M.[2], Mochamad Masnur K.[3], Prayuda A. G.[4], Muhammad Rafli F.[5], Pressa P. S. Saputra[6], Ahmad Althobiti[7]
[1,2,3,4,5]Department of Electrical Engineering, Universitas Negeri Surabaya, Surabaya, Indonesia
[6]Department of Electrical Engineering, Universitas Muhammadiyah Gresik, Gresik, Indonesia
[7]Technical and Vocational Training Corporation, Jeddah, Saudi Arabia
*rifqifirmansyah@unesa.ac.id

Abstract_ *In Industrial applications, DC motors are commonly applied because of high reliability, ease of control and ability to provide accurate speed. However, to get accurate speed control under several operation conditions such as disturbances and changes in the load is significant challenge. This research explores the implementation of particle swarm optimization (PSO) to tune parameters of proportional-integral (PI) controller. PSO that is a population-based optimization technique, is inspired by the social behavior of swarms. It is a population-based optimization technique. By automation process in the algorithm. Using the tuning process of PSO, it can effectively obtain the parameters of PI controller. experimental hardware using DIGIAC 1750 is used to assess the performance of the proposed method. The parameters of* K_p *and* T_i *are 0.7492 and 0.2007, respectively. The results show that the performance of the DC motor using PSO tuned by PI for* t_d, t_r, *and* t_s *are 0.3687 s, 0.5106 s, and 0.6051 s, respectively. Furthermore, when the system is given a disturbance, the response can come back again following the setpoint and when the setpoint is changed, the response can follow the setpoint quickly as well. The proposed method can address the challenges associated with DC motor speed control.*

Keywords: *PI control; DC motor; Speed; Particle Swarm Optimization*

I. INTRODUCTION

In various industries, DC motors that efficiently convert electrical energy into mechanical motion are necessary components because of their ability to provide accurate speed [1][2]. Moreover, they have a straightforward structure, relatively low cost, and flexible control capabilities. They play a vital role in driving modern technologies such as robotic systems, conveyor belts, industrial automation, consumer electronics and electric vehicles. Therefore, to ensure the great performance and energy efficiency of DC motors, accurate speed is needed especially under varying conditions. However, to get the accurate speed and stability of DC motor is not an easy task and challenging to solve [3][4]. Accurate speed control of DC motors enables systems to keep desired operating conditions, operate effectively under varying conditions and fast response in the load changes.

In the literature, several researchers have proposed some techniques to control DC motor [5][6]. In [7], Sayyad et al. have investigated DC motor speed control using proportional-integral (PI) controller. The results reveal that the performance of the proposed method depends on the tuning of their gains. This method is straightforward [8]. However, the method often fails to provide optimal performance, particularly in nonlinear systems like motor DC [9]. In addition, when the load and operating conditions of DC motors vary frequently, the classical PI controller may struggle to provide the desired speed [10].

To tackle this problem, this research proposes the application of particle swarm optimization (PSO) for tuning the parameters of PI controller. In [11], the researchers proposed PSO-PID to control a high speed MPPT. The PID is created to combine with PSO to observe the maximum voltage. The results reveal that the method is useful in improving the PSO algorithm and solar performance. In [12], PSO is used to tune parameter PID for regulating DC motor. The results show that the proposed method can control the speed in the desired performance. PSO is inspired by the social behavior of swarms. It is a population-based optimization technique. By automation process in the algorithm, PSO can provide the precise parameters of PI controller and improve the control performance of the proposed method.

II. SYSTEM COMPONENTS

To apply the proposed method, this research has used several components for experimental hardware such as module sensors and transducers DIGIAC 1750, DC motor, Tachogenerator, and NI Elvis II+. DIGIAC 1750 Trainer is a transducer, instrumentation and control tool used by students for practicum on sensor input, actuator output, signal conditioning circuitry, and display devices as shown in Fig. 1. The trainer also provides detailed curriculum manuals that provide background theory for practical activities and questions for students.

[Figure 1 about here.]

DC motor presented in Fig. 2 requires direct voltage supply in the field coil to be converted into energy mechanic. The field coil on the motor is called the stator, and the anchor coil is called a rotor. DC motors consist of various types, one of which is a permanent magnet DC motors. In this study, the motor used is a permanent magnet DC motor with a maximum voltage of 12 V.

[Figure 2 about here.]

Figure 3 is a tachometer used for speed sensors. It can generate DC voltage which can directly produce velocity information, the sensitivity of tachometer DC generators is quite good especially in high-speed areas.

[Figure 3 about here.]

The National Instrument Educational Laboratories Virtual Instrumentation Suite (NI ELVIS) presented in Fig. 4 can be described as a prototyping module that is LabVIEW or computer based instrumental. NI ELVIS consists of prototyping boards, multi-functional data acquisition (DAQ) devices and virtual instruments based on LabVIEW.

[Figure 4 about here.]

Drawing from the mathematical model of a system, one can determine the order of the system from the rank of the variable s in the Laplace transformation. A system is described as first-order when it’s transfer function has the s term with the rank of one as the highest.

$$\frac{C(s)}{R(s)} = \frac{K}{\tau s+1} \tag{1}$$

Where K is Overall Gain and τ is Time Constants. Determining the System Parameter K (Overall Gain) of a Linear System Y_{ss} can be evaluated as provided.

$$K = \frac{Y_{ss}}{X_{ss}} \tag{2}$$

To determine the time constant τ can be calculated through the system output response when it reaches 63.2% of the steady state, $C(\tau)$ can be calculated as follows:

$$C(\tau) = 0{,}632Y_{ss} \tag{3}$$

Where $C(\tau)$ is system output 63.2% of the steady state. The first-order response curve for unit step signal input is shown in Figure 5.

[Figure 5 about here.]

The most prominent form of the PI controller few folks argue is the ‘ideal’ PI controller which is given by

$$G_c(s) = K_c\left(1 + \frac{1}{T_i s}\right) \tag{4}$$

Where K_c states proportional reinforcement and T_i represents Integral time. Both K_c and T_i can be set. Among the elements of devices equipped with the P controller, the speed domain steady state error is the one which is ideally approached by including device with the PI controller. Nevertheless, as far as response time and the system's general stability are concerned, it has negative effects. PSO Algorithm was first introduced by Eberhart and Kennedy in 1995. The origin of the PSO inspired the behavior of a bird flock or a school of fish while searching for prey. The flowchart of PSO algorithm is depicted in Fig. 6. The initial step undertaken in this domain is referred to as initialization parameters, which encompasses specifying the number of iterations. The next step is to generate the population. The population size (n), inertia weight (w), and the constants for cognitive learning and social learning (c1 and c2). The next step is to create the population matrix to display a random values matrix of dimensions, ranging within [0,1]. The next step is initialization of speed and position where both the velocity and position of the particle is set to zero before error is computed (ref-out) [11]. Then, the system will calculate the fitness value or a function immanent to be maximized. The process will continue until the maximum iteration is reached. Then, the last step will examine whether the output of calculating has already achieved convergence.

[Figure 6 about here.]

III. MOTOR DC SPEED CONTROL DESIGN

This dc motor speed control system uses three supporting devices in this research are Laptop (Personal Computer), NI ELVIS, and DIGIAC 1750 Trainer. PI controller, disturbance, and setpoint are included in the LabVIEW software that has been installed on the laptop (PC). NI ELVIS functions as a digital converter to analog or analog to digital to connect the laptop with the DIGIAC 1750 Trainer. While for the Power Amplifier, DC Motor and Tachogenerator and the process value are found in the DIGIAC 1750 Trainer.

[Figure 7 about here.]

The speed control system of dc motor as shown in Figure 7 uses a closed loop system consisting of Setpoint, Controller, Actuator, Plant, Sensor, Disturbance and Process Value. The setpoint input is the desired speed value. A block diagram of PI controller program with PSO is created using LabVIEW software on a laptop. Where value of K_p and T_i will be found with MATLAB code using PSO. The actuator in this system is a power amplifier to change the signal from the controller so that it can be read by a dc motor so that the dc motor can rotate. Plants in this system are dc motors. Tachogenerator as a sensor to read the speed of a dc motor. Disturbance in system is disturbance input through LabVIEW software. For the output or process value of this system is the actual speed.

To determine the mathematical model of the system using first order. The first thing to do is run the system using LabVIEW Software as shown in Fig 8. The setpoint used is 3 volts.

[Figure 8 about here.]

[Figure 9 about here.]

[Figure 10 about here.]

After simulating the open loop diagram, we get a value of Y_{ss} is 2.542262 V (Response System) and X_{ss} is 3 V (Setpoint). Then, determine the value of K. $K = \frac{Y_{ss}}{X_{ss}} = \frac{2.54\,V}{3\,V} = 0.847\,V$.

Then determine the value of τ. $C_\tau = Y_{ss}.\,63.2\%$ and $C_\tau = 1.606\ s$. By using a linear interpolation formula, then the value of τ is obtained. τ = 0.183 s
using Order 1 approach,

$$\frac{C(s)}{R(s)} = \frac{K}{\tau s + 1}$$

The mathematical model of the DC motor is obtained

$$\frac{C(s)}{R(s)} = \frac{K}{\tau s + 1}$$

$$\frac{C(s)}{R(s)} = \frac{0.847}{0.183s + 1}$$

To get the tuning value of PI, we use the code that has been provided and applied to MATLAB, so we only enter the transfer function and other parameters, so that the process of finding the value with iteration that we want is done, until K_p and T_i values appear. This is called offline PI tuning methods.

[Figure 11 about here.]

[Figure 12 about here.]

From the result, we obtained $Kp = 0.1349$ and $Ti = 0.0813$

IV. RESULT AND DISCUSSION

After obtaining K_p and T_i values using Particle Swarm Optimization, we simulate the closed loop system in LabVIEW as shown in Figure 13. Where with the value of K_p and T_i are 0.7492 and 0.2007, respectively.

[Figure 13 about here.]

[Figure 14 about here.]

In the first experiment, a DC motor is given a set point of 3 V. Then the response results are obtained shown in Figure 15.

[Table 1 about here.]

Then in this session, the DC Motor is given a 3 V set point, after that the system is given disturbance and the PI Controller will set the motor to the set point. This disturbance uses Rheostat with a resistance of 15.6 Ω. The following response results are shown in Figure 16.

[Figure 16 about here.]

Then in the third experiment, the DC Motor is given a set point that changes from the setpoint are 2 V then 1 V continue to 3 v and the last is 2 V. The results obtained as shown in Figure 17. The response can follow the setpoint when the setpoint is changed.

[Figure 17 about here.]

V. CONCLUSION

This paper presents the design of a PI control tuned by PSO to control the speed of DC motor. By using the PSO method the value of K_p and T_i is 0.7492 and 0.2007, respectively. When the system is run using the K_p and T_i, the response performance can reach the setpoint with $\mathrm{t_d}$, $\mathrm{t_r}$, and $\mathrm{t_s}$ are 0.3687 s, 0.5106 s, and 0.6051 s, respectively. Moreover, when there is a disturbance in the system, the response can tackle the disturbances and back to the setpoint. Furthermore, when the setpoint is changed, the response can follow the setpoint quickly.

REFERENCES

[1] R. Firmansyah and R. Irmawanto, "Comparison Study of PI Controller Tuning Method to Regulate the DC Motor Speed," in *ICRACOS 2021 - 2021 3rd International Conference on Research and Academic Community Services: Sustainable Innovation in Research and Community Services for Better Quality of Life towards Society 5*, 2021. doi: 10.1109/ICRACOS53680.2021.9702007.

[2] Prashant Garg and Narottam Dutt Upadhyay, "Speed Control of DC Motor – A Review," *International Journal of Advanced Research in Science, Communication and Technology*, 2023, doi: 10.48175/ijarsct-14320.

[3] P. P. S. Saputra and R. Firmansyah, "Short Circuit Failure Detection in Induction Motor Using Wavelet Transform and Fuzzy C-Means," *SinkrOn*, vol. 8, no. 2, 2023, doi: 10.33395/sinkron.v8i2.12207.

[4] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A review of PID control, tuning methods and applications," 2020. doi: 10.1007/s40435-020-00665-4.

[5] M. Hilal, H. ALRikabi, and I. A. Aljazaery, "A Control System of DC Motor Speed: Systematic Review," *Wasit Journal of Computer and Mathematics Science*, vol. 2, no. 1, 2023.

[6] O. Ibrahim, N. Z. Yahaya, and N. Saad, "Comparative studies of PID controller tuning methods on a DC-DC boost converter," in *International Conference on Intelligent and Advanced Systems, ICIAS 2016*, Institute of Electrical and Electronics Engineers Inc., Jan. 2017. doi: 10.1109/ICIAS.2016.7824044.

[7] Shadab. N. Sayyad and Mr. W. Gavhane, "DC MOTOR SPEED CONTROL USING P.I CONTROLLER," *Open Access Repository*, vol. 9, no. 04, 2022.

[8] A. Intidam *et al.*, "Development and Experimental Implementation of Optimized PI-ANFIS Controller for Speed Control of a Brushless DC Motor in Fuel Cell Electric Vehicles," *Energies (Basel)*, vol. 16, no. 11, 2023, doi: 10.3390/en16114395.

[9] E. Çelik and M. Karayel, "Effective speed control of brushless DC motor using cascade 1PDf-PI controller tuned by snake optimizer," *Neural Comput Appl*, vol. 36, no. 13, 2024, doi: 10.1007/s00521-024-09470-y.

[10] J. U. Liceaga-Castro, I. I. Siller-Alcalá, J. D. G. S. Román, and R. A. Alcántara-Ramírez, "PI Speed Control with Reverse Motion of a Series DC Motor Based on the Noise Reduction Disturbance Observer," *Actuators*, vol. 11, no. 5, 2022, doi: 10.3390/act11050117.

[11] G. Al-Muthanna *et al.*, "A High Speed MPPT Control Utilizing a Hybrid PSO-PID Controller under Partially Shaded Photovoltaic Battery Chargers," *Sustainability (Switzerland)*, vol. 15, no. 4, Feb. 2023, doi: 10.3390/su15043578.

[12] E. S. Rahayu, A. Ma'arif, and A. Cakan, "Particle Swarm Optimization (PSO) Tuning of PID Control on DC Motor," *International Journal of Robotics and Control Systems*, vol. 2, no. 2, 2022, doi: 10.31763/ijrcs.v2i2.476.

*Corespondent e-mail address rifqifirmansyah@unesa.ac.id
Peer reviewed under reponsibility of Universitas Muhammadiyah Sidoarjo, Indonesia

Received: 2024-10-18
Accepted: 2024-10-29
Published: 2024-10-31

LIST OF TABLES

Table 1. Analysis of Response Using Pi Controller with PSO

Setpoint	τ(s)	Td(s)	Tr (s)	Ts (s)	
				5%	2%
3V	0.183	0.3687	0.5106	0.605	0.6558

LIST OF FIGURE

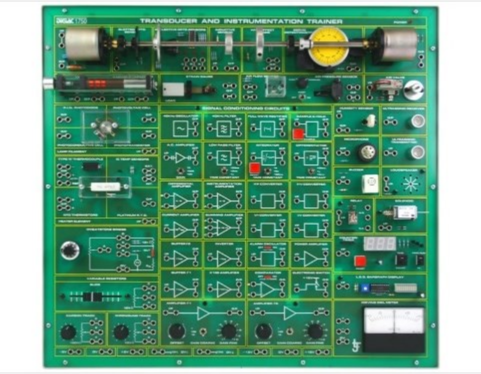

Figure 1. DIGIAC 1750

Figure 2. DC Motor

Figure 3. Tachogenerator

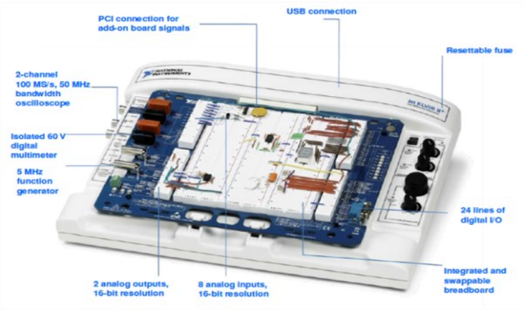

Figure 4. Ni Elvis Configuration

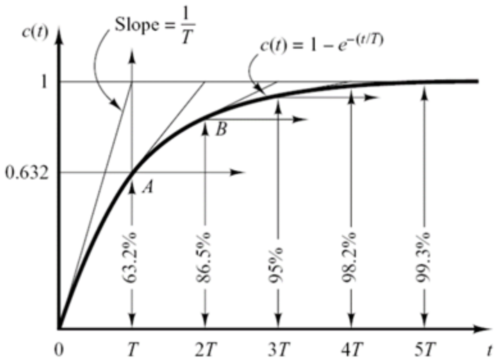

Figure 5. Graph of First Order Response to Enter Unit Step

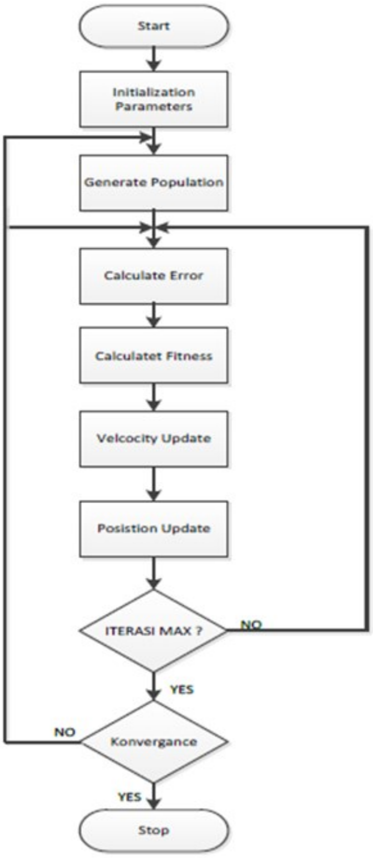

Figure 6. Flowchart of PSO Algorithm

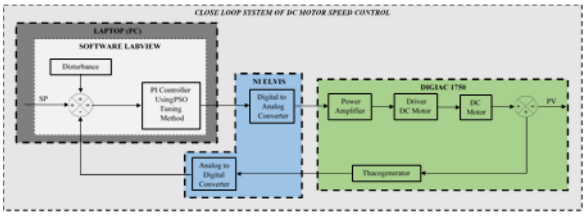

Figure 7. Block Diagram of System

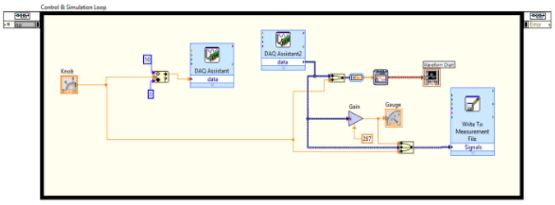

Figure 8. Design System Close Loop PI Controller

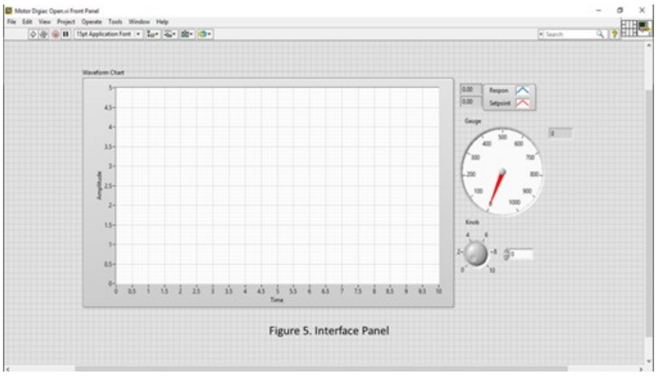

Figure 9. GUI Labview Software

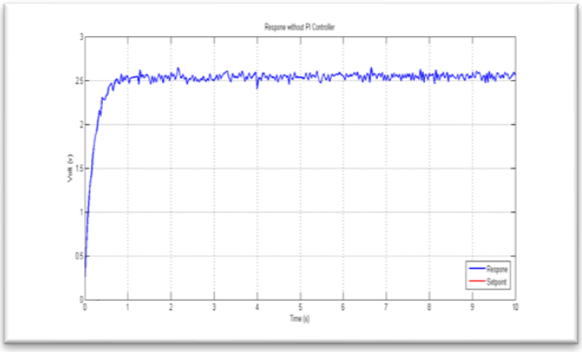

Figure 10. Graphic Response Open Loop System

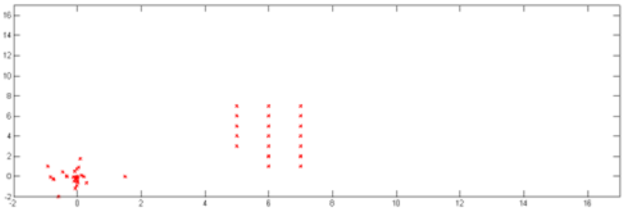

Figure 11. Result of Particel Swarm Optimization

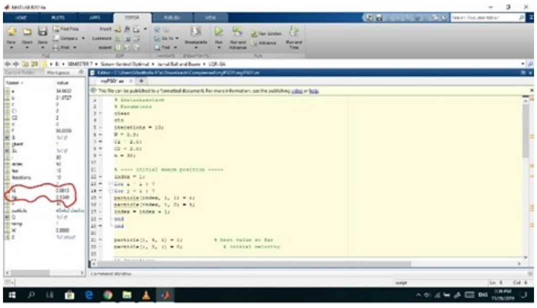

Figure 12. Result of K_p and T_i Using Particle Swarm Optimization Tuning Method

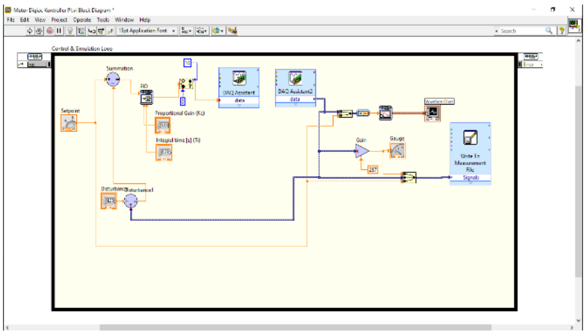

Figure 13. Design System Close Loop PI Controller

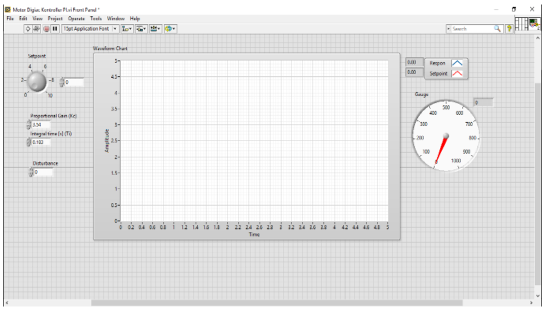

Figure 14. GUI LabVIEW Software

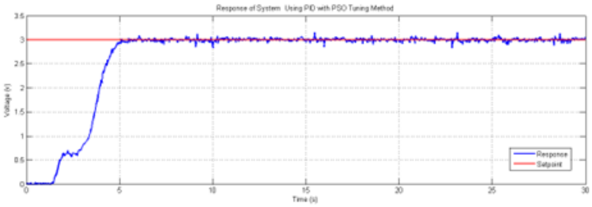

Figure 15. Result of Response with PSO

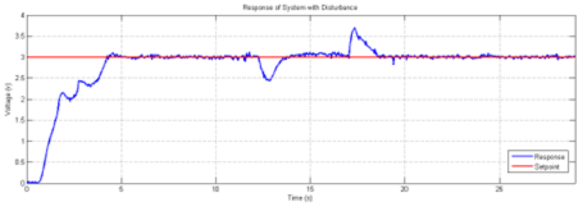

Figure 16. The Result of Response with Disturbance

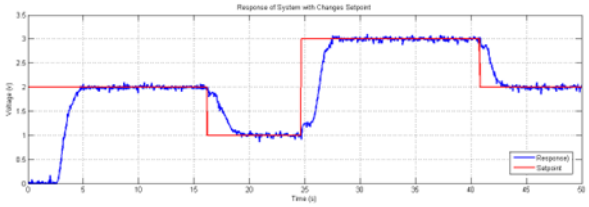

Figure 17. The Result of Response Setpoint Changeable