Tuning PID controller parameters of the DC motor with PSO algorithm

Şahin Yıldırım¹*, Mehmet Safa Bingöl¹,² and Sertac Savas¹

¹ Mechatronics Engineering, Engineering Faculty, Erciyes University, Kayseri, Turkey
² Mechatronics Engineering, Engineering Faculty, Nigde Omer Halisdemir University, Nigde, Turkey

Received: June 23, 2023  •  Accepted: December 13, 2023

ABSTRACT

Direct current (DC) motors have superior features such as operating at different speeds, being affordable and easily controllable. Therefore, DC motors have many uses, such as machine tools and robotic systems in many factories up to the textile industry. The PID controller is one of the most common methods used to control DC motors. PID is a feedback controller with the terms Proportional, Integral, and Derivative. The proper selection of P, I, and D parameters is critical for achieving the desired control in the PID controller. In this study, the transfer function of a DC motor is first obtained, and the speed of the DC motor is controlled by the PID controller using this transfer function. Then, Particle Swarm Optimization (PSO), an optimization method based on swarm intelligence, is used to adjust the P, I, and D parameters. By using the obtained P, I, and D coefficients, the speed of the DC motor is tried to be controlled, and the effect of the filter coefficient on the system output is examined. The performance of the proposed PSO-PID controller with successful results is given in tables and graphics. Control and optimization studies are carried out with MATLAB Simulink.

KEYWORDS

DC motor, particle swarm optimization, PID, filter coefficient

1. INTRODUCTION

At the present time, direct current (DC) motors have a wide range of uses, from machine tools to robotic systems. In some unique and sensitive systems, it is critical that the motor provides the desired speed output. However, obtaining the desired speed output in open loop systems is difficult, especially due to external disturbances. As a result, a controller is needed for DC motor speed control. There are two types of control systems: open loop and closed loop. The Proportional Integral Derivative (PID), a closed-loop controller, is one of the approaches used to control systems. PID calculates the difference between the desired output and the actual output, called the error, and a control signal is generated according to this error value. It is essential to adjust the PID parameters to obtain the desired output. For this purpose, various optimization algorithms are used. One of them is particle swarm optimization (PSO) [1, 2].

PID is used in many applications such as control of bus suspension systems [3, 4], tracking problems of a mobile robot [5], upper limb rehabilitation robot [6], hybrid energy storage system for electric vehicles [7], trajectory tracking control of robot manipulator [8]. Baidya et al. used firefly-PID controller to reduce system error between the reference and output signals in electric vehicles [9]. The results obtained from the proposed controller are compared with classical PID control, firefly-PID controller, teaching learning-based optimization-PID controller, and PSO-PID controller. Ibrahim et al. used bacterial foraging and PSO technique to find the best PID controller parameters for controlling the speed of a brushless DC motor that is modeled in Simulink [10]. Lins and Krishnakumar presented tuning of PID controller for a PV-fed BLDC motor using PSO and teaching-learning-based optimization (TLBO) algorithm in their paper [11]. Gökce et al. analyzed PID control of brushed DC motor in their paper [12]. PID controller parameters are tuned with PSO and
Ziegler-Nichols (ZN). Results show that PSO approach has clearly higher performance. In this study, firstly, a DC motor transfer function is obtained. Afterwards, this transfer function is controlled with PID, and PID parameters are optimized with PSO algorithm. The effect on the output has been examined by changing the N (filter coefficient) value.

In addition, the optimization performance of the PSO algorithm has been demonstrated in articles in which experimental studies have been carried out. Vishal et al. studied the determination of control parameters of an experimental DC motor using different optimization algorithms by minimizing Integral Time Absolute Error (ITAE) [13]. Payakkawan et al. used AI-based heuristic optimization techniques based on particle swarm optimization (PSO) in their studies. On-line tuning has also been performed on a real DC motor [14]. Rahayu et al. used PSO to optimize their studies. On-line tuning has also been performed on techniques based on particle swarm optimization (PSO) in [13]. Payakkawan et al. used AI-based heuristic optimization algorithms by minimizing Integral Time Absolute Error (ITAE) experimental DC motor using different optimization algo-

2. MATERIALS AND METHOD

2.1. DC motor

DC motors widely used today generate rotational motion directly. This movement is used by transmitting with elements such as gears or belts. The electrical equivalent circuit of the armature is given in Fig. 1. This section [17] explains how to extract the DC motor transfer function used in this study.

T refers to motor torque; i refers to armature current. T is obtained by the product of K_0 and i. The armature voltage e is proportional to the rotational speed and is shown by Equation (2).

\[ T = K_0 i \]  

\[ e = K_0 \dot{\theta} \]  

Using Kirchoff’s Law and Newton’s Law, we get Equation (3) and Equation (4).

\[ J \ddot{\theta} + b \dot{\theta} = Ki \]  

\[ L \frac{di}{dt} + Ri = V - K \dot{\theta} \]  

Laplace transforms of Equation (3) and Equation (4) are made to obtain the transfer function of the DC motor.

\[ s(Js + b)\theta(s) = Ki(s) \]  

\[ (Ls + R)I(s) = V - K\dot{\theta}(s) \]  

As a result, the transfer function of the DC motor is found and given in Equation (7).

\[ P(s) = \frac{\dot{\theta}(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R)K^2} \]

The DC motor parameters used in this study are given in Table 1 [18].

The DC motor parameters given in Table 1 are used while creating the simulink model. J, L, R, b, and K represent motor inertia, motor inductance, motor resistance, viscous friction, torque constant, respectively.

2.2. PID control

One of the automatic control techniques is PID control technique. The PID control technique uses three parameters, proportional, integral, and derivative, to reduce the error. The control signal u(t) is calculated as Equation (8) [19].

\[ u(t) = K_p e(t) + K_i \int_0^t e(t)dt(t) + K_d \frac{d}{dt}e(t) \]  

When designing PID control systems, the main purpose is to set PID parameters and apply them to the system so that the desired output can be obtained.

2.3. Particle swarm optimization

The widespread use of PID controllers in the industry has increased the interest in tuning their parameters. Particle swarm optimization algorithm is proposed by Kennedy and Eberhart in 1995 [22]. PSO is developed by drawing on

---

**Table 1. DC motor parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor inertia</td>
<td>J</td>
<td>0.01 (kg m² s⁻²)</td>
</tr>
<tr>
<td>Motor inductance</td>
<td>L</td>
<td>0.5 (H)</td>
</tr>
<tr>
<td>Motor resistance</td>
<td>R</td>
<td>1 (ohm)</td>
</tr>
<tr>
<td>Viscous friction</td>
<td>b</td>
<td>0.1 (N m s)</td>
</tr>
<tr>
<td>Torque constant</td>
<td>K</td>
<td>0.01 (N m/Amp)</td>
</tr>
</tbody>
</table>

---

**Fig. 1.** The electrical equivalent circuit of the armature

**Fig. 2.** Block diagram representation of the PID control scheme
the foraging behavior of bird flocks. PSO is a calculation method that progresses the solution recursively, starting from a candidate solution to reach the best solution while optimizing the problem [23].

In PSO, particles represent the possible solutions, and the particle community refers to the flock. The particle symbolizes any bird in the search space. While birds in nature act based on their own experiences while searching for food, they also show the behavior of following the bird closest to food. Thus, the value of each particle reaches a better point with the effect of the flock memory [24]. Flowchart representation of the PSO algorithm is given in Fig. 3. A long distance of movement works in the first iterations, while this distance should be shortened in the last iterations. Thus, it will be easier to find the optimum value. Therefore, different PSO methods have come to the fore recently [25–27].

In PSO algorithm, the initial positions of the particles are randomly generated in the relevant search space. Generally, the initial velocities of the particles are taken as zero. The fitness values of randomly generated positions are calculated using the objective function of the problem. According to the fitness values, the best position of each particle so far (xbest,i) and the position of the best of the population (gbest) are determined. The velocities of the particles are updated using these positions [28].

3. SIMULATION AND RESULTS

In this study, the DC motor, whose transfer function is obtained, is tried to be controlled with a speed of 50 rad s⁻¹. First of all, the DC motor is operated with open loop. The graphic obtained is given in Fig. 4. As can be seen from the graph, the DC motor cannot provide the speed output as desired. The DC motor is controlled with PID control to achieve the desired output. The effect of PID parameters on the max overshoot, rise time, settling time, and steady-state error is enormous. Therefore, the PID parameters are optimized by the PSO method. For this optimization process, 50 particles (n = 50) are first used, then the same optimization process is carried out for 100 particles (n = 100). The PSO algorithm is run at 1000 iterations for both n values. The system is controlled using the PID parameters obtained for both. PID parameters obtained as a result of PSO are given in Table 2.

The MATLAB Simulink model used in this study is given in Fig. 5.

<table>
<thead>
<tr>
<th>Table 2. Optimized PID parameters with PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of particles</td>
</tr>
<tr>
<td>PSO-PID1 (n = 50)</td>
</tr>
<tr>
<td>PSO-PID2 (n = 100)</td>
</tr>
</tbody>
</table>

Fig. 4. Velocity of DC motor for open loop
Fig. 5. Simulink model of PID control

Table 3. DC motor parameters

<table>
<thead>
<tr>
<th>N (filter coefficient)</th>
<th>Max overshoot (%)</th>
<th>Rise time (s)</th>
<th>Settling time (s)</th>
<th>Steady-state error (rad s⁻¹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open loop</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>45.01</td>
</tr>
<tr>
<td>PSO-PID1</td>
<td>1</td>
<td>39.98</td>
<td>0.0935</td>
<td>0.3975</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>37.98</td>
<td>0.0756</td>
<td>0.3001</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0</td>
<td>0.0668</td>
<td>0.0668</td>
</tr>
<tr>
<td>PSO-PID2</td>
<td>1</td>
<td>43.58</td>
<td>0.0744</td>
<td>0.3316</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>37.66</td>
<td>0.0600</td>
<td>0.2585</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1.86</td>
<td>0.0450</td>
<td>0.0450</td>
</tr>
</tbody>
</table>

Fig. 6. PSO-PID1 control of DC motor velocity

Fig. 7. PSO-PID2 control of DC motor velocity
The model in Fig. 5 is run using the PSO parameters in Table 2. In addition, different N (filter coefficient) values are used. Three different values are used for the N value as 1, 10, and 100. These N values have been tested for both PSO-PID1 and PSO-PID2. The results obtained for max overshoot, rise time, settling time, and steady-state error are given in Table 3. The graphic obtained for PSO-PID1 is given in Fig. 6, and the graphic obtained for PSO-PID2 is given in Fig. 7. In Fig. 8, all results are given in a single graphic.

As can be seen in Fig. 6, the maximum overshoot gets its best value when N is at its lowest value. For N = 1, the maximum overshoot is 39.98%, while for N = 10, the maximum overshoot decreased to 37.98%. When N = 100, no maximum overshoot is observed. The rise time is 0.0935 for N = 1 and 0.0668 for N = 100. Similarly, settling time decreased from 0.3975 to 0.0668. No steady state error occurred for any N value.

As can be seen in Fig. 7, the maximum overshoot got its best value when N was at its lowest value. While the maximum overshoot is 43.58% for N = 1, the maximum overshoot decreased to 1.86% for N = 100. The rise time is 0.0744 for N = 1 and 0.045 for N = 100. Similarly, settling time decreased from 0.3316 to 0.045. No steady state error occurred for any N value.

4. CONCLUSIONS

This paper has presented a proposed controller approach for DC motor velocity control. The PSO-PID controller results have better performance than the standard PID controller. The increase in the number of n particles in the PSO algorithm has not significantly contributed to finding the optimum PID parameters while increasing the computational load. Increasing the N (filter coefficient) value at a specific rate is crucial for the DC motor output to give the desired output. Although the same PID parameters are used, increasing the N value has resulted in a serious decrease, especially in the max overshoot. Also, rise time and settling time have decreased. This proposed controller structure will be employed in real-time applications of such systems.

REFERENCES