

PID Designs Using DE and PSO Algorithms for Damping Oscillations in a DC Motor Speed

Lailis Syafaah, Widiyanto, Ilham Pakaya, Diding Suhardi, M. Irfan

Department of Electrical Engineering
The University of Muhammadiyah Malang
Jl. Raya Tlogomas 246 Malang, Indonesia
e-mai: lailis_tsd@yahoo.com

Abstract—In this paper, a design and a simulation of a PID controller are presented to control a DC motor speed. The design is proposed to increase a dynamic stability of the motor speed. Changing the speed setpoint values may generate oscillations, thus the controller is required to maintain a system stability. The PID controller is tuned using Differential Evolutionary (DE) and Particle Swarm Optimization (PSO) algorithms. Performance of both algorithms is compared by Integral Square Error (ISE) and rise time values. Experiments are performed by changing the speed setpoint values. In simulation results, the ISE and the rise time values of the DE algorithm are slower than the PSO algorithm.

Keywords—DC Motor; Differential Evolutionary (DE); Particle Swarm Optimization (PSO); Integral Square Error (ISE)

I. INTRODUCTION

Using electric motors in industries are neglected requirements. These are caused by automation and easier in controlling. The electric motors are divided into two types based on input source types, that is Direct Current (DC) and Alternating Current (AC) sources. Both the electric motors have differences in applications caused by generated torques. The DC motor generates the lower torques compared than the AC motor [1]. However, the DC motor is easier than it in controlling [2]. Thus, controlling the DC motor is a research focus in this paper.

There are disadvantages in using the electric motors such as decreasing a speed by loading. In order to overcome the weakness, the industries used some controllers, one of these, is a PID controller. The controller is a control feedback mechanism. It is applied to correct an error between a measured variable process and a desired value by giving correctness. By integrating the PID controller to the DC motor, it may correct the error generated by the motor and control the speed or a desired position value.

Problems in using the PID controller are to determine a gain and parameters. Some methods have been proposed to determine PID parameters, beginning with conventionally tuning methods such as root-locus and Ziegler Nichols methods [3]. Determining the PID parameter methods conventionally are not necessary to produce optimal values. Some random tuning methods, such as Genetic Algorithm (GA) and Particle

Swarm Optimization (PSO) have been proposed to generate an efficiency by searching an optimum global solution in a search-space [4-7].

In controlling model of the DC motor speed, tuning the PID controller parameters is a complex. In this paper, the PID parameters are searched using an artificial intelligence, that is Differential Evolutionary (DE) which increases a system performance in changing the speed setpoint value.

A performance indicator of the controller is comparison between an Integral Square Error (ISE) and a rise time value. The minimum ISE value shows a condition of the most optimal parameter combination of the controller. It is a stable dynamic response in controlling the DC motor speed. Meanwhile, the rise time value shows a speed of the controller in responding to the change of the setpoint value.

II. SYSTEM DESCRIPTION

In this paper, controlling model of a DC motor speed consists of two big parts, that is mathematical models of the motor and a PID controller. These models will be discussed as follows.

A. Model of DC Motor

A system plant consists of the DC motor and an inertia load parts. The DC motor part is composed of a voltage control (armature) as a schematic diagram shown in Fig. 1. The proposed system model is the speed control.

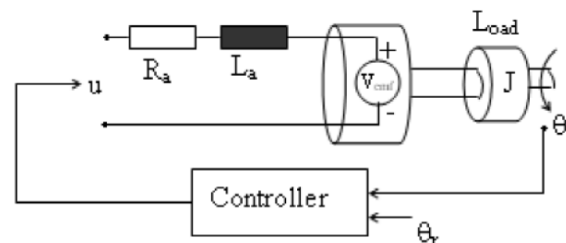


Fig. 1. Equivalent Circuit of DC Motor

A derivative equation of controlling an electric model can be expressed in Eq. 1,

$$V = iR + \frac{Ldi}{dt} + E \tag{1}$$

Eq. 1 can be expressed in Eq. 2,

$$\frac{di}{dt} = 1/L(-E - iR + V) \tag{2}$$

where,

- V = DC voltage (volt)
- L = armature inductance (Henry)
- R = armature resistance (ohms)
- E = back EMF (volt)

Relation between a torque and the speed can be denoted in Eq. 3,

$$T = \frac{Jd\omega}{dt} + B\omega + TL \tag{3}$$

where,

- T = Torque (Newton.meter)
- J = inertia moment(Kg.m²)
- TL = disturbance input
- B = frictional coefficient(Kg.ms)
- ω = angular displacement(rad/sec)

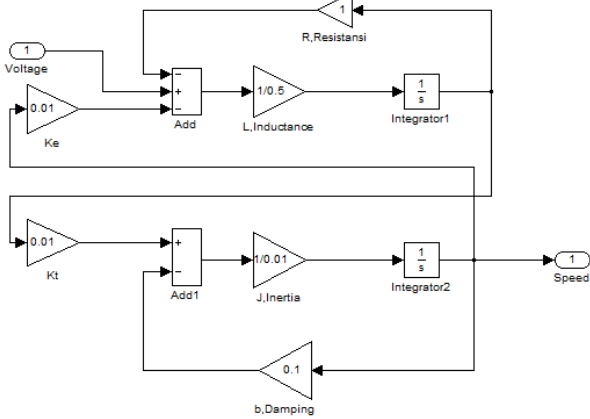


Fig. 2. Model of DC Motor

Figure 2 shows the mathematical model of the DC motor in the Matlab/Simulink program.

B. Design of PID Controller

In this paper, a system will be controlled is changing the DC motor speed, thus adding a controller is required in input voltage of the motor. The PID controller is a controller algorithm and used in a control system. In a close loop system, a feedback is most important part to know any condition as a system response. The controller is designed to correct error values occurring between the feedback value and a desired setpoint value. It consists of variable coefficients, proportional (P), integral (I), and derivative (D). A transfer function of the controller can be expressed in Eq. 4,

$$G_c(s) = K_p(1 + \frac{K_I}{s} + T_d s) \tag{4}$$

Where values of K_p , $\frac{K_I}{s}$ and $K_p T_d$ are a representation of proportional, integral, and derivative gains. Figure 3 shows a placement of the controller in the motor.

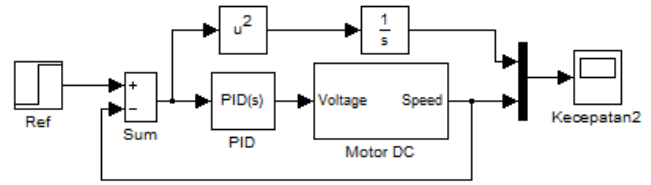


Fig. 3. Model of PID Controller

The controller has a simple structure control and an inexpensive in implementing on a device. However, the controller has a weakness in control parameter values with a long searching time. It is caused by examining a combination of the three parameters. There are some conventional methods in determining controller values, namely root locus and Ziegler-nichols methods. Both the methods are not optimal, thus it is required a special algorithm to optimize the variable values. By adding the controller, it is expected to maintain the speed rotation by the change of a reference value.

III. CONTROL ALGORITHM

A. Differential Evolution (DE)

A DE is a searching method relies on a population using a iteration cycle of a recombination and a selection to pick the population to a global optimum value [8-10]. The following is the DE step to search the optimum value:

Initialization

Evaluation

Iterate

Mutation

Recombination

Evaluation

Selection

Until the desired criteria is expected

Population Structure

The DE algorithm searches an optimal value using a pair of a vector population. Each the population consists of a vector of N_p with a parameter dimensional D . An initial vector P_x consists of a vector of $x_{i,g}$ determined as an initial point. The following is a definition of the DE initial population mathematically denoted in Eq. 5:

$$P_{x,g} = (x_{i,g}), \quad i=1, \dots, N_p, \quad g=1, \dots, g_{max} \tag{5}$$

$$x_{i,g} = (x_{j,i,g}), \quad j=1, \dots, D.$$

$x_{i,g}$ is i -th vector in g -th generation. $x_{j,i,g}$ is i -th vector value on j -th parameter, in g -th generation. A value of i is an integer

number from 0 to N_p , g is the integer number from 0 to g_{max} , and j is the integer number from 0 to $D-1$. The next population, $P_{v,g}$ is the population which contains of DE vectors, N_p , which mutate randomly of $v_{i,g}$. The following is the definition of $P_{v,g}$, mathematically denoted in Eq.6:

$$P_{v,g} = (v_{i,g}), \quad i=1, \dots, N_p, \quad g=1, \dots, g_{max} \quad (6)$$

$$v_{i,g} = (v_{j,i,g}), \quad j = 1, \dots, D - 1.$$

Then, each the vector in the initial population is recombined by the mutant vector to produce the trial population, $P_{u,g}$, with the N_p number of the trial vector, $u_{i,g}$.

$$P_{u,g} = (u_{i,g}), \quad i=1, \dots, N_p, \quad g=1, \dots, g_{max} \quad (7)$$

$$u_{i,g} = (u_{j,i,g}), \quad j = 1, \dots, D.$$

In the recombination process, the mutant population is replaced by the trial population, thus the derived pair population will be processed in the DE is the current population and the trial population.

B. Particle Swarm Optimization (PSO)

An optimization technique relies on a population. The PSO population is denoted as a swarm. In the PSO system, some candidates are initialized and evaluated simultaneously. Each the candidate which produces a solution is called a particle. A characteristic of the particle will move on a problem space to search an optimal value. The particle will move to learn from its particle along moving a time. Each the particle will organize a position based on its around particle experience. When the particle finds the best position to generate the optimal value, the other particles will move to guide to the position's. The particles will move with a different velocity to guide the best position found by the known particles. The velocity of the particle for moving can be calculated using information from:

- Current velocity,
- Space between the first position to the found best position.

This model will be simulated in a search-space of a specific dimensional by iteration numbers, thus in each the iteration, a particle position will guide to an expected target. It is done until the iteration maximum is expected or also using other stopping criteria. Although some modifications on the original swarm algorithm have been done to increase the performance and adapted by a specific problem, a previous series was implemented successfully. Based on a concept or an analogy of the PSO algorithm, thus it is denoted mathematically as follows:

- Each the individual particle i has characteristics as : the current position in a search-space, x_{id} , with the movement velocity, v_{id} , and a best individual position in the search-space, p_{id}
- The best individual position, p_{id} , updated by the position in the search-space in which the particle, i , presents an error which is determined by an objective function, f , to minimize the error.

- The best global position is indicated by presence of the particle position which produces the small error within all the particles p_{gd} .

During iterations, each the particle in swarm is updated using Eqs. 8 and 9,

$$v_{id}(t+1) = w \cdot v_{id}(t) + c_1 r_1 (p_{id} - x_{id}(t)) + c_2 r_2 (p_{gd} - x_{id}(t)) \quad (8)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t) \quad (9)$$

Where $v_{id}(t+1)$ and $v_{id}(t)$ are a newest velocity and a current velocity, respectively. $x_{id}(t+1)$ and $x_{id}(t)$ are a newest position and a current position, respectively. c_1 and c_2 are positive constants. r_1 and r_2 are random numbers in a matrix of $[0,1]$. w is a weight momentum.

Generally, the PSO process is started by an initialization of the position and the particle velocity. The velocity of each the particle is determined until the searched position is found. This process is repeated until the best position is expected. The searching process of the optimal value will be stopped until the maximum iteration is expected, before it is expected, thus a new searching and velocity development processes will occur.

C. Optimization Algorithm

Determining the PID controller parameters of the DC motor optimally is done by a search using DE and PSO algorithms. Pseudocode of the DE algorithm to determine PID parameters as follows,

Begin

Initialization parameters of DE

Initialization parameters of Kp, Ki, and Kd

Do

Mutation

Recombination

Evaluation

Selection

While (value of ISE < criteria)

End

As an objective function used to examine Integral Square Error (ISE). The ISE value is calculated by a formula in Eq. 10.

$$ISE = \sum_{t=0}^{t_{max}} (e_t^2 dt) \quad (10)$$

Parameter data of the used DE algorithm are explained in Table 1.

While some parameter data of the used PSO algorithm are same with the data of the DE algorithm, the same data are population number dimensional, maximum iteration, and the others data. It is caused by a parameter type of a different algorithm equation. The data of the PSO algorithm are explained in Table 2.

TABLE I. DE PARAMETERS FOR CONTROL DESIGN

Parameters	Values
Dimension	3
Crossover	0.8
Number of population	50
Weight	0.7
Maximum iteration	100

TABLE II. PSO PARAMETERS FOR CONTROL DESIGN

Parameters	Values
Dimensional	3
Learning factor (1,2)	0.12 and 1.2
Number of population	50
Momentum	0.9
Maximum iteration	100

A pseudocode of the PSO algorithm to search the PID controller parameters is shown as below

Begin

Initialization parameters of PSO

Initialization parameters of K_p , K_i , and K_d

Do

Update Velocity

Update Position

Evaluation

Selection Global Best Position

While (*value of ISE < criteria*)

IV. RESULTS

Examination of system performances is derived by comparing a behavior to the PID controller. First, the PID controller is tuned without using the optimization algorithm. Second, the controller is tuned using the PSO algorithm. Third, it is tuned using the DE algorithm. Examination results are shown in a table and a graphic of a DC motor speed response. Results of tuning the PID parameter are shown in Table 3.

TABLE III. COMPARISON RESULTS OF CONTROL PARAMETERS

	Controller Parameters			ISE Values	Rise time (s)
	K_p	K_i	K_d		
Conventional PID	1.606	8.583	-1.586	0.84	0.8694
PID with PSO	37.195	84.164	0.6315	0.1098	0.4875
PID with DE	47.940	97.144	0.9730	0.09059	0.3561

Table 3 shows some combination values of controller parameters with each controller PID behaviour. Each the PID controller behaviour derives different ISE values. It is caused by each the controller PID design generating different

combination values of the controller PID parameters. The ISE values of a conventional PID controller generate higher value compared with the controller PID design of an optimization algorithm. It means that the generated value of the optimization algorithm is more satisfied compared than parameter results of the PID controller tuned conventionally. While the ISE values of the controller PID design using the PSO algorithm are higher compared to the generated value by the DE algorithm. The generated parameter values of the DE optimization algorithm are more satisfied than the generated results of the PSO algorithm.

Table 3 shows rise time values in each the PID controller design. It shows system response according to the setpoint value. The rise time values generated by the PID controller using the DE algorithm are slower compared to the other PID designs.

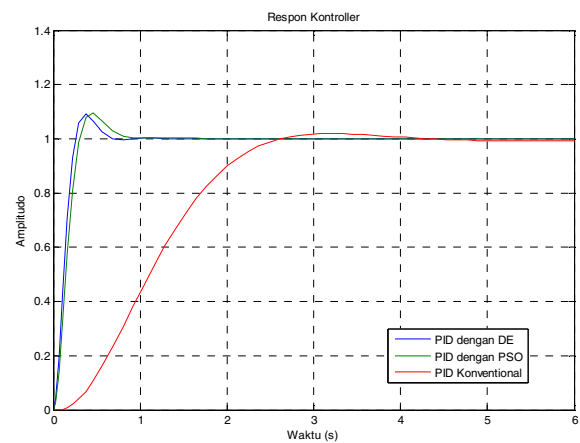


Fig. 4. Response comparison of controller PID design

Figure 4 shows response results of the DC motor speed in giving responses to changing a reference value. A system of the DC motor is given by an input reference, that is a step value on 0th second to 6th second. By changing the reference value, the motor may give the speed change response. It shows that using the conventional PID controller will generate the slower response compared to the PID controller using the optimization algorithm. In Fig. 4, the used time to take on a steady state condition, the conventional PID controller need a longer time, that is 5th second, while the response of PID controller using the optimization is more faster to take the steady state, that is 1st second.

The given responses of the PID controller using the DE algorithm may generate the speed change responses of the motor more faster than using the PSO algorithm. It means that the DE algorithm generates the controller parameters more optimally compared to it. Thus, it is called that the performances of the DE algorithm are more robust compared to the PSO algorithm in designing the controller parameters of the DC motor speed.

V. CONCLUSION

In this paper, a new method is proposed to increasing responses of a DC motor speed change. The proposed method is using a PID controller tuned by a DE algorithm. A system installed by the controller using an optimization algorithm and without it is examined using a simulation by giving a change of a setpoint value. Simulation results show that the system tuned by the DE algorithm may give the speed change responses are more faster and robust compared to using a PSO algorithm and a conventional PID controller method, these are shown by rise time values.

Overshoot occurring in the system tuned by the DE algorithm is slower than the others system. An ISE value of the system using the PID controller with the optimization algorithm is most significant compared without using the algorithm.

ACKNOWLEDGMENT

This work was supported by Ministry of Research, Technology and Higher Education of the Republic of Indonesia under grant PUPT No. 025/E3/2017.

REFERENCES

- [1] F. Giri, "AC Electric Motor Control", John Wiley & Sons Ltd, 2013.
- [2] A. Hughes, "Electric Motor and Drives: Fundamentals, Types and Application", Elsevier Ltd, 2006.
- [3] S. Sheel and O. Gupta, "New Techniques of PID Controller Tuning of a DC Motor – Development of Toolbox", MIT International Journal of Electrical and Instrument Engineering, Vol. 2, pp. 65-69, 2012.
- [4] M.G. Dozein, A. Gholami, M. Kalantar, "Speed Control of DC Motor Using Different Optimization Techniques Based PID Controller", Journal of Basic and Applied Research, Vol. 2, pp. 6488-6494, 2012.
- [5] M. Kushwash and A. Patra, "Tuning PID Controller for Speed Control of DC Motor Using Soft Computing Technique-A Review", Journal of Advance in Electronic and Electric Engineering, Vol. 4, pp. 141-148, 2014
- [6] A. Ayman Aly, "PID Parameters Optimization using Genetic Algorithm Technique for Electrohydraulic Servo Control System", Journal of Intelligence and Automation, Vol.2, pp. 69-76, 2011.
- [7] Q. Bai, "Analysis of Particle Swarm Optimization Algorithm", Journal of Computer and Information Science, Vol.3, 2010
- [8] J.H. Van Sickel, K.Y. Lee, J.S. Heo, "Differential Evolution and its Applications to Power Plant Control", International Conference on Intelligent System Applications to Power Syatems, pp. 560-565, 2007
- [9] C. Liu and M. Yiu, "Modified Differential Evolution Algorithm and Its Application in Thermal Process Model Identification", International Conference on Intelligent Systems and Knowledge Engineering, pp. 450-453, 2010
- [10] M. Ali, M. Pant and A. Abraham, "A Modified Differential Evolution and Its Application to Engineering Problem", International Conference of Soft Computing and Pattern Recognition, pp. 196-201, 2009