

JOURNAL OF SCIENCE



SAKARYA UNIVERSITY

Sakarya University Journal of Science

ISSN 1301-4048 | e-ISSN 2147-835X | Period Bimonthly | Founded: 1997 | Publisher Sakarya University |
<http://www.saujs.sakarya.edu.tr/>

Title: Comparison Performances of PSO and GA to Tuning PID Controller for the DC Motor

Authors: Harun Yazgan, Furkan Yener, Semih Soysal, Ahmet Gür

Received: 2018-01-09 00:00:00

Revised: 2018-05-26 00:00:00

Accepted: 2018-10-10 00:00:00

Article Type: Research Article

Volume: 23

Issue: 2

Month: April

Year: 2019

Pages: 162-174

How to cite

Harun Yazgan, Furkan Yener, Semih Soysal, Ahmet Gür; (2019), Comparison Performances of PSO and GA to Tuning PID Controller for the DC Motor. Sakarya University Journal of Science, 23(2), 162-174, DOI: 10.16984/saufenbilder.376464

Access link

<http://www.saujs.sakarya.edu.tr/issue/39539/376464>

New submission to SAUJS

<http://dergipark.gov.tr/journal/1115/submission/start>

Comparison Performances of PSO and GA to Tuning PID Controller for the DC Motor

Harun Resit Yazgan¹, Furkan Yener^{1*}, Semih Soysal¹, Ahmet Enis Gür¹

Abstract

A DC motor widely uses for sensitive speed and position in industry. Stability and productivity of a system are important for controlling of a DC motor speed. Stable of speed which affected from load fluctuation and environmental factors. Therefore, it is important for the speed value which is required as constant and to keep it as its value. In this study, it is aimed that the speed value which is achieved as required value and keeping it as constant using Proportional, Integral and Derivative (PID) controller for tuning parameters. Firstly, Ziegler-Nichols (ZN) is one of a traditional method used. PID parameters are determined with responses of open-loop under running system. Later, parameters of the PID are estimated using two metaheuristic algorithms such as Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). As a result, three algorithms' results are compared based on five criteria. The PSO algorithm produces better results than Genetic Algorithm for each criteria.

Keywords: DC motor, PID, PSO, GA, Tuning

1. INTRODUCTION

The main aim of the Control theory is, analysing and designing of controller/monitor which provides running of the system as required property. Different structure and characteristic of controllers have been developed. One of them is a basic Proportional-Integral-Derivative (PID) controller which is mostly used for industrial purposes[1]. Because of robustness and basic structure, the PID controller is mostly used for controlling speed and position of a DC motor in industry[2]. In addition this, the PID become more popular because of integrating easily both in terms of hardware and also software of the system.

Ziegler-Nichols is the first method for tuning of parameters in the PID controller[3]. Except that this, there are new metaheuristic algorithms such as Genetic Algorithm (GA)[2], [4]–[7], Particle Swarm Optimization (PSO)[8]–[10], Ant Colony Optimization[9],[10], Artificial Neural Network[11],[13], Differential Evaluation Algorithm[14], Bacterial Foraging Optimization Algorithm[15], Gravitational Search Algorithm[16], Simulated Annealing[17], Artificial Immune System[18], Artificial Bee Colony Optimization[19], African Buffalo Optimization[20] and Bat Algorithm[21] in the PID control problem.

GA was improved by Holland from Michigan University in 1975. He explained fundamentals of the algorithm in his book which is called

¹ Sakarya University, Department of Industrial Engineering, Sakarya, Turkey

* Corresponding Author, e-mail: fyener@sakarya.edu.tr

“Adaptation in natural and artificial systems”. He is the first researcher who is used evaluation rules to solve optimization problems[22].

PSO was developed by Kennedy and Eberhart who observed social behaviours that inspired movement of organism of a bird flock or fish school in 1995. This algorithm is applied on many different problems. At the same time, It is also used to solve the nonlinear problem that consists of more than one variables are multiples[23].

In this study, ZN, PSO and GA algorithms are used for tuning PID controller on DC motor and the solutions are compared. For this comparison, values of overshoot, rise time, settling time and steady state error from response of a unit step are used. Different criterias are used when the fitness functions of PSO and GA are determined.

Rest of the study is designed as follow. A DC motor model, fundamentals of a PID and a mathematical model are explained in section 2. Tuning methods such as ZN, PSO and GA are explained in section 3. At section 4, an application part of the study is given. Fitness criterias are discussed at section 5. At last, results and discussions are presented in section 6.

2. MODELS

In this part, General information related with both PID controller and DC motor is given in this section in addition to be discussed mathematical equations of PID controller on frequency domain.

2.1. DC Motor Model

The DC motor system is a type of motor is generally preferred to control speed and tuning position which is an armature control of DC motor. An Armature controlled DC motor is controlled with arranging armature for keeping constant of static field current. Equivalent circuit of the armature controlled on the DC motor is illustrated in Figure 1[1].

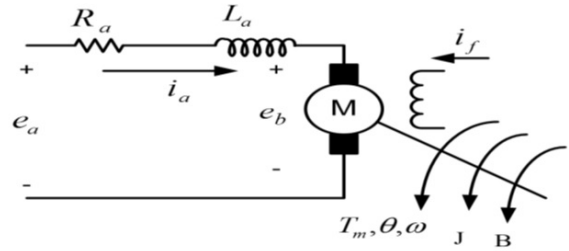


Figure 1 Equivalent circuit of an armature controlled on the DC motor

Electrical and mechanical equations based on time domain represents:

$$e_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e_b(t) \quad (1)$$

$$T_e(t) = K_i i_a(t) \quad (2)$$

$$T_e(t) = K_i i_a(t) \quad (3)$$

$$T_m(t) = J \frac{dw(t)}{dt} + Bw(t) + T_y \quad (4)$$

$$e_b(t) = K_b w(t) \quad (5)$$

$$T_e(t) = T_m(t) \quad (6)$$

R_a : Armature resistance (Ω)

i_a : Armature current (A)

L_a : Armature inductance (H)

i_f : Field current (A)

e_a : Input voltage (V)

e_b : Back electromotive force (EMF) (V)

K_b : EMF constant ($\frac{Vs}{rad}$)

T_m : Motor torque (Nm)

K_i : Moment constant ($\frac{Nm}{A}$)

w : Angular velocity of rotor ($\frac{rad}{sec}$)

B : Friction constant (Nms)

J : Moment of inertia constant (kgm^2)

Laplace transformations of equations based on a time domain are completed as follows;

$$E_a(s) = R_a I_a(s) + sL_a I_a(s) + E_b(s) \quad (7)$$

$$T_e(s) = K_i I_a(s) \quad (8)$$

$$T_m(s) = sJ\Omega(s) + B\Omega(s) \quad (9)$$

$$T_e(s) = T_m(s) \quad (10)$$

$$E_b(s) = K_b(s)\Omega(s) \quad (11)$$

The block diagram using and determining a Laplace transformation based on equations from 7 to 11 on the armature controlled DC motor system is given in Figure 2.

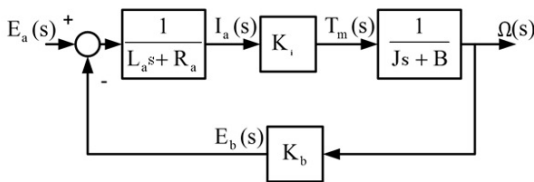


Figure 2 DC motor with the armature control diagram

As a result, the DC motor closed loop transfer function of an acceleration based on input voltage is given in equation 12.

$$G(s) = \frac{\Omega(s)}{E_a(s)} = \frac{K_i}{(sL_a + R_a)(Js + B) + K_b K_i} \quad (12)$$

2.2. PID Controller

Proportional, Integral and Derivative (PID) controller used extensively in the industry means that a control signal $u(t)$ is determined to consider with a power applied to a system with taking into consideration of an input error signal (differences of a reference signal and a feedback) and $e(t)$ signal [17], [22]. The PID controller block diagram is given in Figure 3.

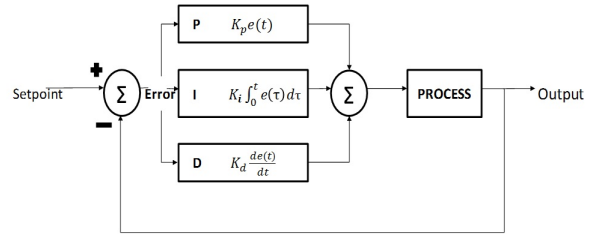


Figure 3 The PID controller block diagram

The PID means that Proportional is “P” keeping proportional reference level, Integral parameter is “I” summation of previous errors, Derivative is “D” trend of error. The PID control belong with the structure that continuous and discrete of the mathematical equations are given as follows [19], [24].

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau + K_p T_d \frac{d}{dt} e(t) \quad (13)$$

$$K_i = \frac{K_p}{T_i} \quad \text{ve} \quad K_d = K_p T_d \quad (14)$$

Having used equations of 13 and 14, equations of 15 and 16 are found.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (15)$$

$$u[n] = K_p \left(e[n] + \frac{T_s}{T_i} \sum_{i=0}^n e[i] + \frac{T_d}{T_s} (e[n] - e[n-1]) \right) \quad (16)$$

Transfer function of the PID controller on frequency domain is as follow.

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (17)$$

Having used equations 16 and 17, equation 18 is determined.

$$G_c(s) = K_p + \frac{K_i}{s} + K_d \quad (18)$$

K_p : Proportional gain

K_i : Integral gain

K_d : Differential gain

T_i : Integral time

T_d : Derivative time

T_s : Sampling time

$e(t)$: Error signal

In others way, the PID controller is formed as a unique controller with collecting three fundamental controllers of P, I and D. A system response time is improved with a derivative structure for minimizing continuous steady-state error that affect from an integrator in the system.

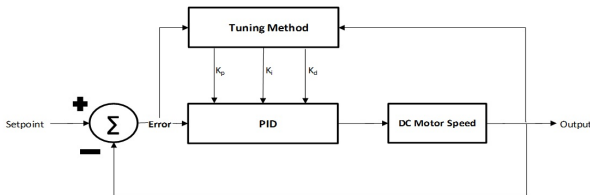


Figure 4 Structure of Tuning PID controller

3. SOLUTION METHODS

In this section, a traditional ZN and two metaheuristic algorithms PSO and GA that are used for tuning the PID controller parameters, are explained shortly.

3.1. Ziegler-Nichols Method

There are different methods which can be used for tuning in a starting position with being considered with open loop response of the system. Although all methods have the same aim, having used model parameters are the response of the system and appropriate the values because of the model parameters are different.

Although their objectives and input model parameters (K , L , T) are same, obtaining PID parameters' approaches are different[24]. Ziegler-Nichols is a one of traditional method which defines as the first time delay order system. It consists of two parameters of lead time (L) and time constant (T)[1]. Formulation of the method is given in equation 19.

$$G_p(s) = \frac{Ke^{-sL}}{Ts + 1} \tag{19}$$

Being controlled open loop transfer function $G_p(s)$ for the system is illustrated how determined the way of an open loop response in Figure 5.

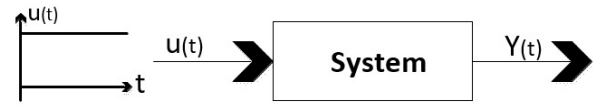


Figure 5 System response unit step diagram

The example of the response curve for the ZN is presented in Figure 6.

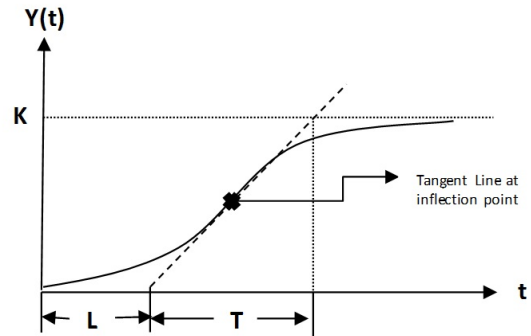


Figure 6 System unit step response curve $Y(t)$

Controller parameters are found with using response curve $Y(t)$, L and T .

Table 1 K_p , T_i , T_d and PID parameters according a transient response method

| Controller | K_p | T_i | T_d |
|--|-----------|-------|--------|
| Proportional (P) | T/KL | - | - |
| Proportional-Integral (PI) | $0.9T/KL$ | $3L$ | - |
| Proportional-Integral-Derivative (PID) | $1.2T/KL$ | $2L$ | $0.5L$ |

3.2. Particle Swarm Optimization Algorithm

Firstly, Particle Swarm algorithm are proposed by Kennedy and Eberhart (1995), is an evolutionary metaheuristic algorithm which bases on swarm base for using an observation of social behaviour of moving organisms in a bird flock or fish school.

The algorithm is related with the computational method that optimizes the problem which bird flocks aim of finding food behaviours is used

iterative steps starting with a candidate solution to reach best solution.

This means that, candidate solutions are particles and moving these particles in the search-space based on a formulation over the particle's position. Each particle's movement is affected by its local value, and its objective is to reach best known positions in the search-space with updating better positions found by other particles.

The algorithm is begun with establishing starting position and velocity vectors. And each iteration is tried to find best value by evaluating position and velocity vectors. Each particle has a dimension whose variables are our problems. If the problem consists of five different variables, the particles' dimension should be chosen as five.

Each particle's best value is called as local best value and recorded into the Pbest matrix. The best value of each particle is updated after each iteration if a new best value is found for controlling each particle current and previous positions. In addition, the position is recorded as global best (Gbest) after controlling best values of the matrix in each iteration. So the iteration is repeated till the specified number of iteration, and last updated best position value is assumed as an optimum value. The algorithm is fundamentally based on finding each particle own position of local best value and a swarm' general best position in each iteration. Each particle's velocity and position are updated according to equation of 20 and 21 in each iteration[25].

$$V_{i,j}(t+1) = V_{i,j}(t) + c_1 r_1 * [P_{best_{i,j}}(t) - X_{i,j}(t)] + c_2 r_2 * [g_{best}(t) - X_{i,j}(t)] \quad (20)$$

$$X_{i,j}(t+1) = X_{i,j}(t) + V_{i,j}(t+1) \quad i = 1, 2, \dots, M ; j = 1, 2, \dots, N \quad (21)$$

c_1 and c_2 parameters are selected between [0-2] and affect on the positions of particles inside swarm. The particles can lay on the large area (search space) if c_1 and c_2 are chosen in a big value and also equal. In our problem, their values are chosen as 2. r_1 and r_2 are random coefficient and normal distributed between [0-1].

N : number of search space

M : number of particle

i : particle index

j : dimension number

X : position matrix

V : velocity matrix

Pbest and Gbest matrix are illustrated as follows:

$$X_{i,j} = \begin{bmatrix} X_{11} & X_{12} & X_{13} & \dots & X_{1N} \\ X_{21} & X_{22} & X_{23} & \dots & X_{2N} \\ \dots & \dots & \dots & \dots & \dots \\ X_{M1} & X_{M2} & X_{M3} & \dots & X_{MN} \end{bmatrix} \quad (22)$$

$$V_{i,j} = \begin{bmatrix} V_{11} & V_{12} & V_{13} & \dots & V_{1N} \\ V_{21} & V_{22} & V_{23} & \dots & V_{2N} \\ \dots & \dots & \dots & \dots & \dots \\ V_{M1} & V_{M2} & V_{M3} & \dots & V_{MN} \end{bmatrix} \quad (23)$$

$$P_{best_{i,j}} = \begin{bmatrix} P_{best_{11}} & P_{best_{12}} & P_{best_{13}} & \dots & P_{best_{1N}} \\ P_{best_{21}} & P_{best_{22}} & P_{best_{23}} & \dots & P_{best_{2N}} \\ \dots & \dots & \dots & \dots & \dots \\ P_{best_{M1}} & P_{best_{M2}} & P_{best_{M3}} & \dots & P_{best_{MN}} \end{bmatrix} \quad (24)$$

$$G_{best} = [g_{best_1} \ g_{best_2} \ \dots \ g_{best_N}] \quad (25)$$

The particles in the PSO, position values are updated until the number of iteration is reached to specified value. The particles' changing positions are illustrated in Figure 7[9].

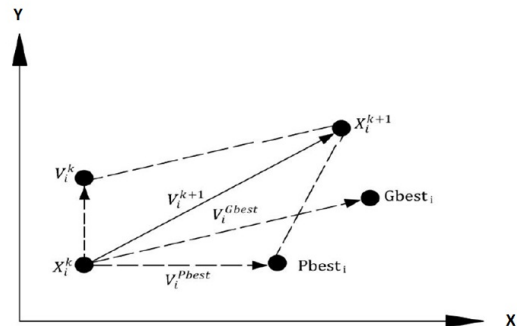


Figure 7 PSO vectorel notation

- X_i^k : Particle's instant position
- X_i^{k+1} : Particle's next position
- V_i^k : Particle's instant velocity
- V_i^{k+1} : Particle's next instant velocity
- V^{Pbest} : Local best in Particle's instant velocity values
- V^{Gbest} : Global best in all of Particle's instant velocity values

Some unsuitable positions can be possible if particles' velocity is not restricted. Therefore, each value of velocity should be satisfied between Vmin and Vmax. Vmax should generally be between 10% and 20% of each dimension range (i.e. positional dimensional range)[8]. "a" and "b" variable values are starting and finishing values of position constraint.

Vmin and Vmax are found with using equations 26-28 as follows:

$$V = \frac{\{b_j - a_j\}}{k} \quad (26)$$

$$V_{min} = -V_{max} \quad j = 1, 2, \dots, N \quad 10 \leq k \leq 20 \quad (27)$$

$$V_{i,j}(t+1) = \begin{cases} V_{max}, & \text{if } V_{i,j}(t+1) > V_{max} \\ V_{min}, & \text{if } V_{i,j}(t+1) < V_{min} \end{cases} \quad (28)$$

At the beginning; each particle is assigned to the position value that based on specified constraints. After the evolution of each particles, if a particle position is over than constraints, the particle is omitted and a new particle is created within position constraints otherwise this particle position can be adapted by arranging suitable particles position. Different range can be used as a position constraint[15].

3.3. Genetic Algorithm

The GA is a metaheuristic algorithm which based on a random search technique with using a parametric coded. The GA is proposed firstly by John Holland (1975) whose student Goldberd applied first time controlling gasoline pipeline problem[26].

The GA can be preferred to solve the control problem if the problem is not easy to solve with using an analytical approach because the search space is huge. The algorithm bases on a principle of living organism which is the best of them will survive, but others will disappear. A child character is a mixed characters of parents. It can survive when it achieves adaptation of new conditions. A new child may carry a better or a worst character from parents. In that case, a child will not survive if its characters are worse than its parents. In the problem solving strategy, a new solution is generated with using parents (previous solutions). A new child (a new solution) is found used operators such as mutation and crossover. One of the advantages of the GA is that a new solution is continuously searched among previous better solutions. Finally, the best solution is selected among better solutions. The performance of GA is affected from parameters of mutation and crossover rates. Flow diagram of the GA is illustrated in Figure 8.

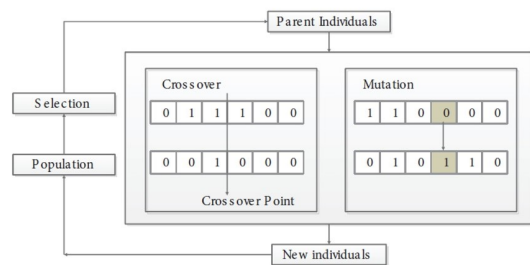


Figure 8 The flow diagram of GA[27]

4. AN APPLICATION

In this section, an example is given to illustrate effectiveness of these algorithms. Two DC motors are chosen for a comparison. Parameters of DC motors are given in Table 2. Testing of the system is done with being used the SIMULINK and all computers codes are generated on the MATLAB platform[28].

Table 2 Parameters of DC Motor 1 and 2

| Parameters | Motor 1 | Motor 2 |
|--|---------|---------|
| Armature Resistance (Ω) | 1 | 2 |
| Armature Inductance (H) | 0.5 | 0.7 |
| Moment of inertia constant (Kgm ²) | 0.01 | 0.05 |
| Friction factor (Nms) | 0.1 | 0.2 |
| Moment constant(Nm / A) | 0.01 | 0.02 |
| EMF constant (Vs / rad) | 0.01 | 0.02 |

4.1. Ziegler-Nichols Method

Starting parameter values, application steps of Ziegler-Nichols method and first step response of closed loop for trialling are shown in Table 3.

Table 3. Pseudocode of Ziegler-Nichols method

```

Begin
  Run a motor as open loop
  Draw that top maximum gradient point tangent on
  an open loop answer
  Set L (Lead time) and T (Time constant)
  Calculate beginning PID parameters ( $K_p$ ,  $K_i$ ,  $K_d$ )
  Set PID Controller transfer function
  Compound a DC Motor and a PID Controller
  transfer function
  Get Closed loop unit step response
  Change that parameters in y axis
    Workable  $K_p$ , for postpone Closed loop
    unit step response
    Workable  $K_i$ , for an adjust oscillation
    Workable  $K_d$ , for an adjust system
    response speed
    
```

4.2. Particle Swarm Optimization Algorithm

The PID controller parameters are found using PSO[29]. Closed loop DC motor step response, are found using previously founded parameters in MATLAB. The pseudocode of the PSO is given in Table 4.

Table 4. Pseudocode of PSO Algorithm

```

Begin
  For each particle
    Initialize particle
  end
Do
  For each particle
    Calculate fitness value
    If the fitness value is better than the best fitness
    value (pBest) in history
    Set current value as the new pBest
  end
  Choose the particle with the best fitness value of all the
  particles as the gBest
  For each particle
    Calculate particle velocity equation
    Update particle position equation
  end
  While maximum iterations or optimum fitness value is
  attained
    
```

4.3. Genetic Algorithm

The PID controller parameters which are closed loop DC motor step response, are found using previously founded parameters in MATLAB. The pseudocode of the GA is given in Table 5.

The GA parameters are chosen as follows:

- Many articles are examined in terms of how to determined GA parameters in the literature.
- Total 45 different results are examined for five different error functions with combinations of three different mutation and crossover operators. As a result, the best crossover rate and mutation operators are chosen as 0,5 and 0,03 respectively.

Table 5. Pseudocode of the proposed GA

```

Begin:
  Generate population randomly
  Compute error function (i) values and Total Error
  Function Value
Selection
  Compute fitness function (j) values
  Compute total fitness values
  Compute selection probability (j)
  Compute cumulative selection probability for each
  chromosomes
  Select n chromosomes according to a cumulative
  selection probability
Crossover
  Crossover operator = p_c
  Select chromosomes from selected 100
  chromosomes according to crossover operator and
  Cumulative selection probability and Group them
  double randomly
  Crossover these groups in themselves from random
  bits
  Change these crossed chromosomes with
  previous chromosomes
Mutation
  Mutation operator: p_m
  Generate as total bits as random numbers between
  0 to 1
  Save them in the p(i)
  For i = 1 to total_bits
    if (p(i) <= P_m)
      if (bit == 1)
        bit = 0
      else
        bit = 1
      end
    end
  end
end
end
    
```


Table 6. Parameters of the GA

| | |
|--------------------------|--------|
| Number Of Population | 100 |
| Crossover Operator | 0.5 |
| Mutation Operator | 0.03 |
| Length Of Chromosome | 67 Bit |
| Sensitivity Of Variables | 4 |
| Number Of Iteration | 50 |

Comparison of performances of the PSO and the GA for the DC motor 1 and 2 are given in Table 7 and 8. The graphs of results for DC motor 1 and 2 are given in appendix A.

5. FITNESS CRITERIA

Some fitness functions are determined to evaluate the PSO and GA performances[30], [31].

- Total absolute value of error (IAE)
- Total weighted absolute value error (ITAE)
- Total square of error (ISE)
- Total time weighted square of error (ITSE)

Fitness functions of IAE, ISE, ITAE, and ITSE as follows:

$$IAE = \int_0^{\infty} (|r(t) - y(t)|) dt = \int_0^{\infty} |e(t)| dt \quad (29)$$

$$ISE = \int_0^{\infty} (r(t) - y(t))^2 dt = \int_0^{\infty} (e(t))^2 dt \quad (30)$$

$$ITAE = \int_0^{\infty} (t \cdot |r(t) - y(t)|) dt = \int_0^{\infty} (t \cdot |e(t)|) dt \quad (31)$$

$$ITSE = \int_0^{\infty} (t \cdot (r(t) - y(t))^2) dt = \int_0^{\infty} (t \cdot (e(t))^2) dt \quad (32)$$

In this study, in addition, fitness function defined in equation between 29-32, a special fitness function based on a time domain is also used[6]. The fitness function consists of an overshoot a rise time, a settling time and a steady-state error. The fitness function is defined as follows:

$$\min_{K_{stabilize}} W(K) = (1 - e^{-\beta}) \cdot (M_p + E_{ss}) + e^{-\beta} \cdot (t_s - t_r) \quad (33)$$

M_p :Overshoot

E_{ss} :Steady-state error

t_s :Settling time

t_r :Rise time

Different results of W(K) are obtained while a weight factor β is changed. If β is greater than 0.7, an overshoot and continue to steady-state errors are reduced. Therefore, β is less than 0.7, rising time and saturation time are reduced[8]. In this study, β value is chosen as 0.7 to assign equal importance for performance of criteria.

Table 7. Comparison of performances of the PSO and the GA for the DC motor 1

| Criteria | K_p | K_i | K_d | Max overshoot (%) | Rise time | Settling time (%) | Steady-state error | Fitness function values | |
|------------|------------|-----------|----------|-------------------|-----------|-------------------|--------------------|-------------------------|------------|
| Origin P. | 261.0000 | 2273.0000 | 7.5000 | 60.9959 | 0.0845 | 0.4821 | -1.5231E-05 | | |
| Manual T. | 261.0000 | 285.0000 | 11.0000 | 5.6693 | 0.1046 | 1.0882 | 8.4895E-07 | | |
| PSO | IAE | 369.8041 | 673.7863 | 14.9988 | 10.8650 | 0.0947 | 1.1361 | 4.5794E-10 | 3.6998 |
| | ITAE | 271.6442 | 484.1309 | 14.8082 | 9.8230 | 0.1113 | 1.1854 | 4.1711E-10 | 0.1382 |
| | ISE | 516.3980 | 410.8292 | 15.0000 | 5.5817 | 0.0901 | 1.1533 | 1.6003E-05 | 1.9069 |
| | ITSE | 321.5744 | 469.3710 | 15.0000 | 8.0175 | 0.1040 | 1.2334 | 2.1601E-08 | 0.0210 |
| | Special F. | 196.9658 | 179.4634 | 13.5444 | 3.5154 | 0.1573 | 1.1819 | 4.5593E-06 | 0.0209 |
| | GA | IAE | 401.0081 | 725.0342 | 14.7727 | 11.2082 | 0.0915 | 1.1315 | 6.1758E-10 |
| ITAE | | 331.2391 | 608.8628 | 14.3449 | 10.6032 | 0.0975 | 1.1319 | 3.3576E-10 | 0.1644 |
| ISE | | 494.6147 | 509.2583 | 14.7927 | 7.1522 | 0.0897 | 1.2485 | 1.9692E-06 | 1.9325 |
| ITSE | | 360.3639 | 523.8116 | 14.8020 | 8.2178 | 0.0977 | 1.2167 | 2.5582E-08 | 0.0220 |
| Special F. | | 161.058 | 191.6586 | 12.1972 | 5.1515 | 0.1683 | 1.3384 | 2.7194E-07 | 0.0384 |

Table 8. Comparison of performances of the PSO and the GA for the DC motor 2

| Criteria | K_p | K_i | K_d | Max overshoot (%) | Rise time | Settling time (%) | Steady-state error | Fitness function values | |
|-----------|------------|----------|----------|-------------------|-----------|-------------------|--------------------|-------------------------|---------|
| Origin P. | 161.0000 | 976.0000 | 6.6700 | 70.6344 | 0.1648 | 2.2561 | -5.9963E-10 | | |
| Manual T. | 37.98 | 91.63 | 3.803 | 10.2622 | 0.4410 | 1.6032 | 2.7491E-06 | | |
| PSO | IAE | 311.9932 | 506.5166 | 13.9869 | 25.4250 | 0.1671 | 1.4944 | 7.6949E-09 | 10.4522 |
| | ITAE | 124.9151 | 222.256 | 13.9332 | 12.9682 | 0.2467 | 1.4914 | 2.0157E-11 | 1.33337 |
| | ISE | 318.3887 | 275.9949 | 13.9492 | 15.1160 | 0.1712 | 1.4973 | 1.2184E-05 | 4.7610 |
| | ITSE | 265.6078 | 326.7835 | 13.9940 | 17.5732 | 0.1733 | 1.5386 | 3.8940E-07 | 0.2111 |
| | Special F. | 108.874 | 149.4915 | 11.6800 | 7.2930 | 0.2791 | 1.4479 | 3.2674E-08 | 0.0604 |
| GA | IAE | 290.1186 | 479.6816 | 13.7842 | 24.8108 | 0.1680 | 1.4656 | 4.9964E-09 | 10.6037 |
| | ITAE | 183.4106 | 353.3408 | 13.6922 | 20.2331 | 0.1846 | 1.3217 | 3.7012E-11 | 1.5289 |
| | ISE | 311.025 | 302.5226 | 13.7190 | 16.4301 | 0.1707 | 1.5623 | 4.9302E-06 | 4.8407 |
| | ITSE | 274.8557 | 345.1266 | 13.8170 | 18.5694 | 0.1719 | 1.5472 | 3.1524E-07 | 0.2174 |
| | Special F. | 82.7864 | 124.4792 | 9.7622 | 6.7907 | 0.3300 | 1.5044 | 4.1602E-09 | 0.1277 |

6. RESULT

In this study, a comparison of performances of the PSO and the GA for optimizing the PID controller being used for the DC motor speed control was done. An output signal was obtained for tuning the PID controller with parameters which based on some criterias and a unit step signals were applied as an input reference signal of the DC motor. An overshoot, a rise time, a settling time and a steady-state error were found by using MATLAB platform while being analysed output signals at the same time. So the manual tuning was done by being used MATLAB and SIMULINK.

First of all, origin parameters of DC motor 1 and 2 was shown in Table 7 and 8. At the beginning of the study, Manual tuning was applied to DC motors. Then, PSO and GA for optimizing the PID controller being used for both DC motors. The results of these algorithm for IAE, ITAE, ISE, ITSE indexes for the parameters, Max overshoot, Rise time, Settling time and Steady-state error, were decreased and shown in table 7 and 8. Max overshoot was 60.9959 with the Origin parameters, applying PSO was decreased to 25.4250 for IAE index, for ITAE index 12.9682 for DC motor 1. Settling time was 2.2561 with the Origin parameters, applying GA was decreased to 1.4656 for IAE index, for ITAE index 1.3217 for

DC motor 2. Finally, $W(K)$ special fitness function was obtained and fitness function values were decreased. PSO was produced the fitness function value 10.4522 for DC motor 2 with index IAE, with the special function $W(K)$ it was decreased to 0.0604. Comparison result of GA and PSO method based on all criteria are shown in appendix A.

Consequently, $W(K)$ special fitness function produced more satisfactory values than other fitness function in terms of overshoot, saturation time, rise time and produced a better optimal value in a case of manual tuning process. For stationary error value was less than "4", it was convenient to assume as "0". The PSO produced better results than GA in terms of fitness function value for each criteria.

Different equations can be used for the fitness function in the special function criteria[32], [33] for future study. However, a certain limit value of the error signal $e(t)$ was, namely the creation of a new control signal $u(t)$ can be provided for being over the threshold. The error value of the system under the certain threshold value will not be a difference in the control signal $u(t)$ when it is applied to the system[34]. All these extra cases are gathered and under the same function can be collected for all fitness criterias. Multi-criteria decision-making problems[12],[35], with new results can be reconsidered.

APPENDIX A

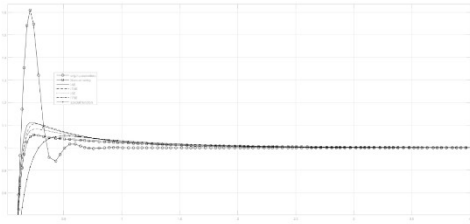


Figure 1. DC MOTOR 1 Comparison result of GA method based on all criteria

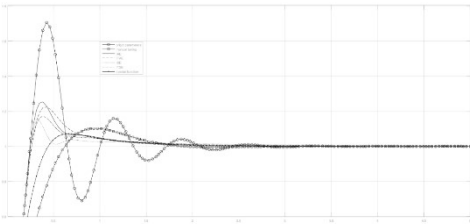


Figure 2. DC MOTOR 1 Comparison result of PSO method based on all criteria

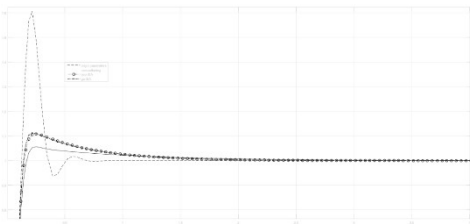


Figure 3. DC MOTOR 1 Comparison result of all methods based on IEA criteria

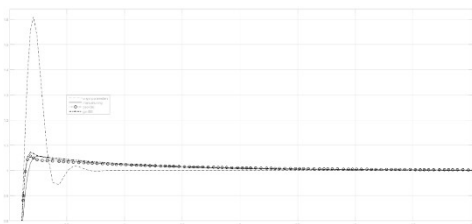


Figure 4. DC MOTOR 1 Comparison result of all methods based on ISE criteria

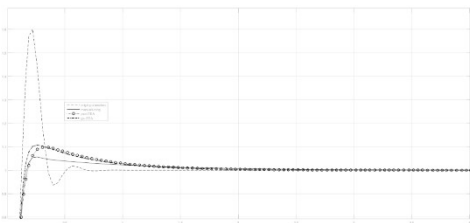


Figure 5. DC MOTOR 1 Comparison result of all methods based on ITEA criteria

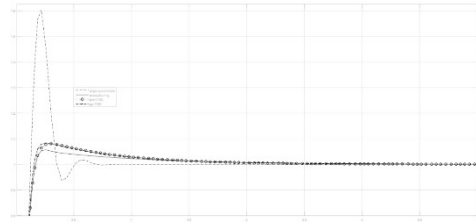


Figure 6. DC MOTOR 1 Comparison result of all methods based on ITSE criteria

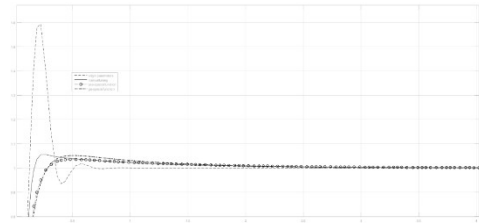


Figure 7. DC MOTOR 1 Comparison result of all methods based on special function criteria

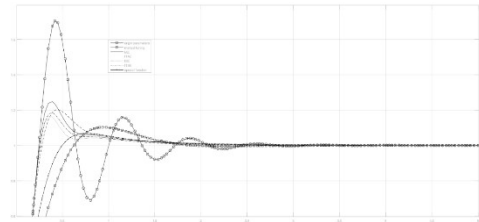


Figure 8. DC MOTOR 2 Comparison result of GA method based on all criteria

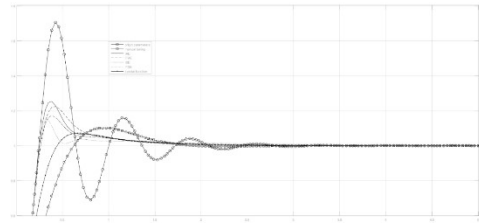


Figure 9. DC MOTOR 2 Comparison result of PSO method based on all criteria

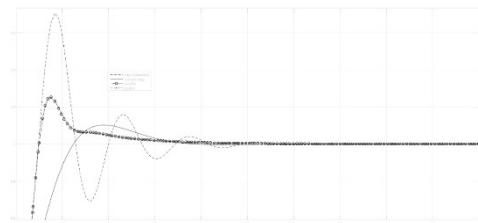


Figure 10. DC MOTOR 2 Comparison result of all methods based on IEA criteria

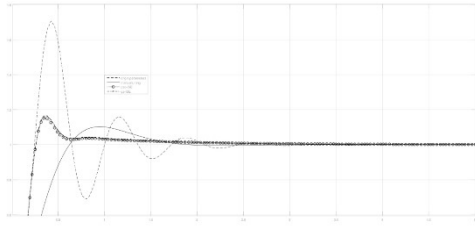


Figure 2. DC MOTOR 2 Comparison result of all methods based on ISE criteria

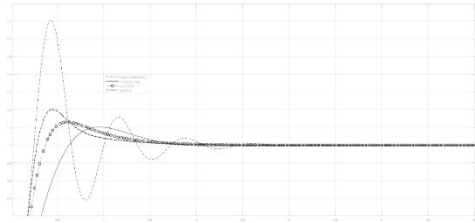


Figure 3. DC MOTOR 2 Comparison result of all methods based on ITEA criteria

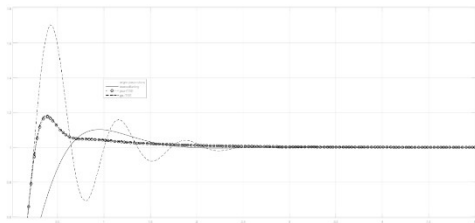


Figure 4. DC MOTOR 2 Comparison result of all methods based on ITSE criteria

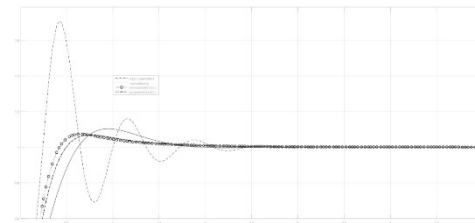


Figure 5. DC MOTOR 2 Comparison result of all methods based on special function criteria

References

- [1] R. G. Kanojiya and P. M. Meshram, "Optimal tuning of PI controller for speed control of DC motor drive using particle swarm optimization," in *Advances in Power Conversion and Energy Technologies (APCET), 2012 International Conference on*, 2012, pp. 1–6.
- [2] N. Thomas and D. P. Poongodi, "Position control of DC motor using genetic algorithm based PID controller," in *Proceedings of the World Congress on Engineering*, 2009, vol. 2, pp. 1–3.
- [3] J. C. Basilio and S. R. Matos, "Design of PI and PID controllers with transient performance specification," *IEEE Trans. Educ.*, vol. 45, no. 4, pp. 364–370, 2002.
- [4] S. G. Kumar, R. Jain, N. Anantharaman, V. Dharmalingam, and K. Begum, "Genetic algorithm based PID controller tuning for a model bioreactor," *Indian Chem. Eng.*, vol. 50, no. 3, pp. 214–226, 2008.
- [5] M. K. Tan, Y. K. Chin, H. J. Tham, and K. T. K. Teo, "Genetic algorithm based PID optimization in batch process control," in *Computer Applications and Industrial Electronics (ICCAIE), 2011 IEEE International Conference on*, 2011, pp. 162–167.
- [6] A. D. Lidbe, A. M. Hainen, and S. L. Jones, "Comparative study of simulated annealing, tabu search, and the genetic algorithm for calibration of the microsimulation model," *Simulation*, vol. 93, no. 1, pp. 21–33, 2017.
- [7] S. M. H. Mousakazemi, N. Ayoobian, and G. R. Ansarifar, "Control of the reactor core power in PWR using optimized PID controller with the real-coded GA," *Ann. Nucl. Energy*, vol. 118, pp. 107–121, 2018.
- [8] Z.-L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE Trans. Energy Convers.*, vol. 19, no. 2, pp. 384–391, 2004.
- [9] B. Allaoua, B. Gasbaoui, and B. Mebarki, "Setting up PID DC motor speed control alteration parameters using particle swarm optimization strategy," *Leonardo Electron. J. Pract. Technol.*, vol. 14, pp. 19–32, 2009.

- [10] R. Dong, "Differential evolution versus particle swarm optimization for PID controller design," in *2009 Fifth International Conference on Natural Computation*, 2009, vol. 3, pp. 236–240.
- [11] Y.-T. Hsiao, C.-L. Chuang, and C.-C. Chien, "Ant colony optimization for designing of PID controllers," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, 2004, pp. 321–326.
- [12] I. Chiha, N. Liouane, and P. Borne, "Tuning PID controller using multiobjective ant colony optimization," *Appl. Comput. Intell. Soft Comput.*, vol. 2012, p. 11, 2012.
- [13] K. Kinoshita, S. Ohno, and S. Wakitani, "Design of neural network PID controller based on E-FRIT," in *Society of Instrument and Control Engineers of Japan (SICE), 2017 56th Annual Conference of the*, 2017, pp. 1183–1186.
- [14] X. Dong and others, "The PID Controller Based on the Artificial Neural Network and the Differential Evolution Algorithm," 2012.
- [15] V. Rajinikanth and K. Latha, "I-PD controller tuning for unstable system using bacterial foraging algorithm: a study based on various error criterion," *Appl. Comput. Intell. Soft Comput.*, vol. 2012, p. 2, 2012.
- [16] S. Duman, D. Maden, and U. Güvenç, "Determination of the PID controller parameters for speed and position control of DC motor using gravitational search algorithm," in *Electrical and Electronics Engineering (ELECO), 2011 7th International Conference on*, 2011, pp. 1–225.
- [17] R. E. Haber, R. Haber-Haber, R. M. Del Toro, and J. R. Alique, "Using Simulated Annealing for Optimal Tuning of a PID Controller for Time-Delay Systems. An Application to a High-Performance Drilling Process," in *International Work-Conference on Artificial Neural Networks*, 2007, pp. 1155–1162.
- [18] Y. Peng, X. Luo, and W. Wei, "A New Control Method Based on Artificial Immune Adaptive Strategy," *Elektron. Ir Elektrotechnika*, vol. 19, no. 4, pp. 3–8, 2013.
- [19] P. Varma and B. A. Kumar, "Control of DC motor using artificial bee colony based PID controller," *Int J Digit. Appl Contemp Res*, vol. 2, pp. 1–9, 2013.
- [20] J. B. Odili, M. N. M. Kahar, and A. Noraziah, "Parameters-tuning of PID controller for automatic voltage regulators using the African buffalo optimization," *PloS One*, vol. 12, no. 4, p. e0175901, 2017.
- [21] L. Chaib, A. Choucha, and S. Arif, "Optimal design and tuning of novel fractional order PID power system stabilizer using a new metaheuristic Bat algorithm," *Ain Shams Eng. J.*, vol. 8, no. 2, pp. 113–125, 2017.
- [22] H. John, *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, 1992.
- [23] R. Ebenhart, "Kennedy. Particle swarm optimization," in *Proceeding IEEE Inter Conference on Neural Networks, Perth, Australia, Piscat-away*, 1995, vol. 4, pp. 1942–1948.
- [24] A. K. Mishra, V. K. Tiwari, R. Kumar, and T. Verma, "Speed control of DC motor using artificial bee colony optimization technique," in *Control, Automation, Robotics and Embedded Systems (CARE), 2013 International Conference on*, 2013, pp. 1–6.
- [25] G. Pereira, "Particle Swarm Optimization," *INESCID Inst. Super. Techno Porto Salvo Port.*, 2011.
- [26] D. E. Golberg, "Genetic algorithms in search, optimization, and machine learning," *Addion Wesley*, vol. 1989, p. 102, 1989.
- [27] D. H. Kim, W. P. Hong, and J. I. Park, "Auto-tuning of reference model based PID controller using immune algorithm," in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, 2002, vol. 1, pp. 483–488.
- [28] A. Schmidt, U. Durak, and T. Pawletta, "Model-based testing methodology using system entity structures for MATLAB/Simulink models," *Simulation*, vol. 92, no. 8, pp. 729–746, 2016.
- [29] S. P. Ghoshal, "Optimizations of PID gains by particle swarm optimizations in fuzzy based automatic generation control," *Electr.*

- Power Syst. Res.*, vol. 72, no. 3, pp. 203–212, 2004.
- [30] R. A. Krohling and J. P. Rey, “Design of optimal disturbance rejection PID controllers using genetic algorithms,” *IEEE Trans. Evol. Comput.*, vol. 5, no. 1, pp. 78–82, 2001.
- [31] Y. Mitsukura, T. Yamamoto, and M. Kaneda, “A design of self-tuning PID controllers using a genetic algorithm,” in *American Control Conference, 1999. Proceedings of the 1999*, 1999, vol. 2, pp. 1361–1365.
- [32] J. Zhang, J. Zhuang, H. Du, and others, “Self-organizing genetic algorithm based tuning of PID controllers,” *Inf. Sci.*, vol. 179, no. 7, pp. 1007–1018, 2009.
- [33] L. Fan and E. M. Joo, “Design for auto-tuning PID controller based on genetic algorithms,” in *2009 4th IEEE Conference on Industrial Electronics and Applications*, 2009, pp. 1924–1928.
- [34] K.-E. \AArzén, “A simple event-based PID controller,” in *14th IFAC world congress*, 1999.
- [35] A. A. El-Gammal and A. A. El-Samahy, “Adaptive tuning of a PID speed controller for DC motor drives using multi-objective particle swarm optimization,” in *Computer Modelling and Simulation, 2009. UKSIM'09. 11th International Conference on*, 2009, pp. 398–404.