

# Tuning PID Controller Bases on Random Search Algorithms

Dung Vuong Quoc\*  
Center of Information Technology  
Hanoi University of Industry  
Hanoi, Vietnam

Thuy Tran Thi  
Center of Information Technology  
Hanoi university of Industry  
Hanoi, Vietnam

Loi Nguyen Tien  
Center of Information Technology  
Hanoi university of Industry  
Hanoi, Vietnam

Nguyet Do Thi Minh  
Center of Information Technology  
Hanoi University of Industry  
Hanoi, Vietnam

Minh Nguyen Quang  
Center of Information Technology  
Hanoi university of Industry  
Hanoi, Vietnam

Binh Nguyen Thi Thanh  
Center of Information Technology  
Hanoi university of Industry  
Hanoi, Vietnam

**Abstract**—In this paper, we present two well known random search algorithms which are Genetic algorithm (GA) and Particle Swarm optimization (PSO) to find optimal parameters for PID controller for a model of DC motor is used as a plant. To test performance of GA and PSO algorithms, we compare them with the traditional Ziegler-Nichols method in term of performance indices. The simulation results show that Proportional Integral and Derivative controller (PID) designed by GA and PSO algorithms yields better results than the traditional method in terms of the performance index.

**Keywords**—Tuning PID; GA algorithm; PSO algorithm; Ziegler-Nichols method; performance index; optimization.

## I. INTRODUCTION

Proportional Integral and Derivative (PID) controllers have been used in industrial control applications for a long time. This is due to its simplicity, low-cost design and robust performance in a wide range of operating conditions. According to a survey in 1998, nearly 90% of the controllers used in industries are PID controllers [1]. By adding a derivative term into Proportional Integral (PI) controller, it improves the stability of control loop. The combined effect of Proportional and Derivative (D) controllers introduces a predictive capability in the controller since D term is able to react to foreseeable future control errors. The proportional term works as a fast-acting correction which will produce a change in the output as quickly as the error arises. The integral action takes a period of finite time to act but has the capability to make the steady zero state error. Unfortunately, tuning the PID controllers is tedious and it might be difficult to tune the PID gains properly due to the nonlinearity and the high complexity of the system [2]. Therefore, beside traditional tuning method we need heuristic methods for tuning parameters of controller. Recently, random search algorithms are widely chosen to solve nonlinear optimization problems. The categories of these algorithms do not require differentiable and continuous objective functions. Some random search algorithms are used for tuning PID controller such as GA [3-4], PSO [2], CRO [8].

In this paper, we use GA and PSO algorithm to find global optimization for parameters of PID controller and compare result with traditional Ziegler-Nichols tuning method with respect to integral performance indices.

This paper is organized as follows: In the next section, PID controller is presented. Section 3 discusses GA and PSO algorithms. Next section presents the tuning PID controller. Section 5 demonstrates the performance of the various methods by experiment. Final section presents the conclusion and future studies.

## II. PID CONTROLLER

Proportional Integral Derivative (PID) controller is widely used in industry and practice. PID is often used in close loop control system presented in figure 1.

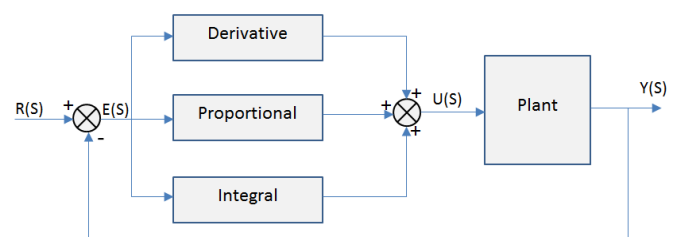


Fig. 1. Block diagram of close loop control system with PID controller.

PID controller combines three features of P, I and D controllers. P element instantaneously adjusts proportional control variable to the error signal. This is quite logical and it may be compared to the natural reaction of a process operator. The larger the error signal is, the larger the proportional control variable should be adjusted. One drawback of P term is any change in the operating conditions which will result into an offset, which is a steady state deviation of controlled variable from the set point. To overcome this drawback, the I term is added to form PI controller. Meanwhile, D term is added to PI controller to form PID because this element considers change rate of the error. Therefore, using the derivative element in

controller allows the control variable to anticipate and counteract the cause of the error signal before it can take significant values. In PID controller, the control variable is related to the error signal by equation (1).

$$u(t) = K_P \times e(t) + K_I \times \int_0^t e(\tau) \times d\tau + K_D \times \frac{de(t)}{dt} \quad (1)$$

This expression often writes in other form.

$$u(t) = K_P \times \left( e(t) + \frac{K_I}{K_P} \times \int_0^t e(\tau) \times d\tau + \frac{K_D}{K_P} \times \frac{de(t)}{dt} \right) = K_P \times \left( e(t) + \frac{1}{T_I} \times \int_0^t e(\tau) \times d\tau + T_D \times \frac{de(t)}{dt} \right) \quad (2)$$

Where  $T_I = \frac{K_P}{K_I}$  is referred as the integral time constant or reset time of controller.

$T_D = \frac{K_D}{K_P}$  is derivative time constant.

The transfer function of PID controller can be written as flowing form:

$$C(s) = \frac{U(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D \times s = K_P \times \left( 1 + \frac{1}{T_I \times s} + T_D \times s \right) = K_P \times \frac{1 + T_I \times s + T_I \times T_D \times s^2}{T_I \times s} \quad (3)$$

Where  $K_P$ ,  $K_I$ ,  $K_D$  are proportional gain, integral gain and derivative gain respectively.

By tuning three parameters in the PID controller algorithm, the controller can provide control action designed for specific process requirements. There are some traditional methods to determine  $K_P$ ,  $K_I$ ,  $K_D$  such as Ziegler - Nichols, CHR modification of the Ziegler - Nichols settings, Cohen-Coon controller settings, Lambda Tuning Technique, Haalman Controller Settings. In addition, there are some optimization methods widely used for tuning controller. In these methods, the controller parameters are determined with a view to the minimization of a given integral performance criteria such as integral of the absolute error (IAE), integral of the square error [5] are listed below:

$$IAE = \int_0^{+\infty} |e(t)| dt \quad (4)$$

$$ISE = \int_0^{+\infty} e(t)^2 dt \quad (5)$$

$$ITAE = \int_0^{+\infty} t |e(t)| dt \quad (6)$$

$$ITSE = \int_0^{+\infty} t e(t)^2 dt \quad (7)$$

In case of PID controller, error signal  $e(t)$  is function of  $K_P$ ,  $K_I$ ,  $K_D$ . We can choose one of integrals above as objective function for modern heuristic random search algorithms.

### III. GA AND PSO ALGORITHMS

#### A. Genetic Algorithm

The genetic algorithm is formed based on the concept that natural evolution is the most perfect, the most logical and optimal process in itself. This concept is considered to be a true and unprovable axiom, but it is consistent with objective reality and was discovered by the scientist Darwin in his theory of evolution. The evolutionary process is optimally shown that the next generation is always more developed and perfect than the previous generation. Natural evolution is maintained by two natural processes of reproduction and selection. During natural evolution new individuals are born to supplement the old one. Natural selection will retain more developed and environmentally adaptable individuals, and discard those less adapted to the environment. Thus, genetic algorithms in

particular and evolutionary algorithms in general simulate natural evolution through four basic operations: crossover, mutation, reproduction and natural selection.

Genetic algorithms are often used to solve optimization problems and then the solution of the problem is represented by individuals, also known as chromosomes, sets of individuals is called a population. A chromosome is made up of genes, the value of genes can be decimal numbers, binary numbers, symbols or letters depending on the problem to be solved. To evaluate the individuals in a population, the genetic algorithm uses the fitness function, this value is to measure the suitability of each solution to the problem. Some individuals in the population are selected for crossover to create new offspring and these offspring will inherit part of the father's genes and part of the mother's genes. Some individuals are selected to mutate, through mutation that changes their genes in one or more specific locations. The number of individuals to be crossover and mutation was controlled by parameters of the crossover probability ( $pc$ ) and the mutation probability ( $pm$ ). Individuals selected for the new population follow the Darwin theory of evolution, those individuals with a larger fitness function value will have more chances of being selected into the new population. After some evolution we choose the individual with the largest fitness function value and it will be the best solution to the given problem to be solved.

Thus, to solve the problem with genetic algorithms, we usually proceed through the following steps:

- Step 1: Specify the parameters of algorithm which includes: population size, max iteration (generation), crossover probability and mutation probability.
- Step 2: Create initial population.
- Step 3: Repeat step 4 to 7 until stop criterion is satisfied.
- Step 4: Calculate fitness value for each individual in population base on fitness function.
- Step 5: Perform natural selection and crossover.
- Step 6: Perform mutation.
- Step 7: Create a new population.
- Step 8: Select best individual which is the solution for given problem.

#### B. Particle Swarm Optimization

Particle swarm optimization was developed in 1995 by the authors Kennedy and Eberhart, inspired by the behavior of social organisms in groups, such as bird and fish schooling or ant colonies. This algorithm emulates the interaction between members to share information. Particle swarm optimization has been applied to numerous areas in optimization and in combination with other existing algorithms. This method performs the search of the optimal solution through agents, referred to as particles, whose trajectories are adjusted by a stochastic and a deterministic component. Each particle is influenced by its 'best' achieved position and the group 'best' position, but tends to move randomly. A particle is defined by its position vector,  $P_i$ , and its velocity vector,  $V_i$ . Every iteration, each particle changes its position according to the new velocity. To seek the optimal solution, each particle moves in the direction to its previously best ( $P_{best}$ ) position and the global best ( $G_{best}$ ) position in the swarm [9].

$$P_{best}(i, t) = \min_{k=1,2,\dots,t} [f(P_i(k)), i \in \{1, 2, \dots, N_p\}]$$

$$G_{best}(t) = \min_{\substack{k=1,2,\dots,k \\ i=1,2,\dots,N_p}} [f(P_i(k))] \quad (8)$$

where  $i$  denotes particle index,  $N_p$  is total number of particles,  $t$  is the current iteration number,  $f$  is the fitness function,  $P$  is the position. Velocity  $V$  and position  $P$  of the particles are updated by the following equations:

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 [P_{best}(i, t) - P_i(t)] + c_2 r_2 [G_{best}(i, t) - P_i(t)] \quad (9)$$

$$P_i(t+1) = P_i(t) + V_i(t+1) \quad (10)$$

where  $\omega$  is the inertia weight used to balance the global exploration and local exploitation,  $r_1$  and  $r_2$  are uniformly distributed random variables within range  $[0, 1]$ , and  $c_1$  and  $c_2$  are positive constant parameters called "acceleration coefficients."

In summary, the PSO algorithm is illustrated with pseudocode as follows:

Step 1: Initialization

- Choose constant parameters  $c_1$ ,  $c_2$ , population size  $N_p$
- FOR EACH  $i^{\text{th}}$  particle in population DO
- ✓ Initialize the particle's position with a uniform random number in range of the lower bound (LB) and upper bound (UB) of the search space.  $P_i(0) \sim U(\text{LB}, \text{UB})$
- ✓ Initialize  $P_{best}(i, 0) = P_i(0)$
- ✓ Initialize  $V_i(0) \sim U(-|\text{UB}-\text{LB}|, |\text{UB}-\text{LB}|)$
- END\_FOR
- Find  $G_{best}(0) = \min[P_{best}(i, 0)]$ ,  $i=1, 2, \dots, N_p$

Step 2: Iteration until stop criteria is met.

- Create random number  $r_1, r_2 \sim U(0, 1)$
- Update the each particle's velocity according to the formular (5).
- Update the each particle's position according to the formular (6).
- IF  $f[P_i(t)] < f[P_{best}(i, t)]$  THEN  
 $P_{best}(i, t) = P_i(t)$   
 END\_IF
- IF  $f[P_i(t)] < f[G_{best}(t)]$  THEN  
 $G_{best}(t) = P_i(t)$   
 END\_IF
- $t = t + 1$

Step 3: OUTPUT  $G_{best}(t)$  is optimal solution.

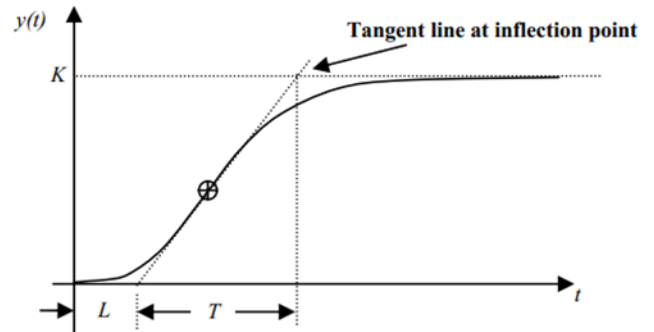
#### IV. TUNING PID CONTROLLER

In this article, we apply oldest traditional Ziegler Nichols (NN) method for tuning PID controller and our proposal method using GA and PSO algorithms for finding optimal parameters of PID controller so that minimum integral performance indices. NN method was proposed by John Ziegler and Nathaniel Nichols in 1942 and is still a simple and fairly

effective PID tuning method. Ziegler and Nathaniel Nichols proposed two methods for tuning PID controller. The first method applies for process model type of first order system with transport lag. Which has step response likes figure 2 [6].

Fig. 2. Step response of plant for first ZN method.

From this response we can estimate delay time  $L$ , constant time  $T$  and state value  $K$ . Then  $K_P$ ,  $K_I$  and  $K_D$  of PID controller can be calculated by equations (11) - (16) following:



$$K_P = 1.2 \times \frac{T}{L} \quad (11)$$

$$K_I = 0.5 \times \frac{K_P}{L} = 0.6 \times \frac{T}{L^2} \quad (12)$$

$$K_D = 0.5 \times L \times K_P = 0.6 \times T \quad (13)$$

The second of NN method for tuning PID controller as flowing steps:

Decrease integral and derivative gain to zero (disable integral and derivative effect)

Increase proportional gain from zero to ultimate value  $K_u$  at which oscillations occur with period  $T_u$ .

From estimated  $K_u$  and  $T_u$ , we can calculate parameters of PID controller as:

$$K_P = 0.6 \times K_u \quad (14)$$

$$K_I = 2 \times \frac{K_P}{T_u} = 1.2 \times \frac{K_u}{T_u} \quad (15)$$

$$K_D = \frac{K_P T_u}{8} = 0.6 \times \frac{K_u T_u}{8} \quad (16)$$

To apply PSO and GA for finding optimization of PID controller, we can choose one of integral performance index listed in (4) – (7) as objective function. In our case we choose integral of time absolute error (ITAE) as objective function. We encode  $K_P$ ,  $K_I$  and  $K_D$  as chromosome in GA and position of particle (solution) in PSO and then we run PSO and GA algorithm to search global min of ITAE. When stopping criterion is satisfied, we obtain optimal parameters of PID controller.

#### V. EXPERIMENTAL RESULTS

To test our proposal method, we consider model of DC motor is described by figure 3:

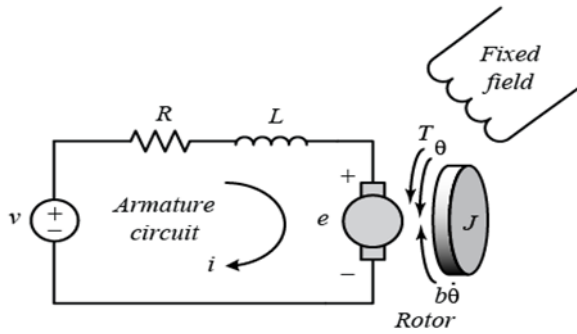


Fig. 3. Diagram of the DC motor.

And model can be expressed by system equation (17) below:

$$\begin{cases} J \times \frac{d^2\theta(t)}{dt^2} + b \times \frac{d\theta(t)}{dt} = K \times i(t) \\ L \times \frac{di(t)}{dt} + R \times i(t) = V - K \times \frac{d\theta(t)}{dt} \end{cases} \quad (17)$$

Substitute  $\omega(t) = \frac{d\theta(t)}{dt}$  is angular velocity of the motor to system equations (17) we obtain:

$$\begin{cases} J \times \frac{d\omega(t)}{dt} + b \times \omega(t) = K \times i(t) \\ L \times \frac{di(t)}{dt} + R \times i(t) = V - K \times \omega(t) \end{cases} \quad (18)$$

where:

$J$  is the moment of inertia of the rotor: 0.01 kg.N2  
 $b$  is the motor viscous friction constant: 0.1 N.m.s  
 $Ke$  is the electromotive force constant: 0.01 V/rad/s  
 $Kt$  is the motor torque constant: 0.01 N.m/A  
 $R$  is the electric resistance: 1  $\Omega$   
 $L$  is the electric inductance: 0.5 H  
 $K = Ke = Kt$

We perform Laplace transform two sides of system equation (19). Finally, we obtain transfer function of system as:

$$P(s) = \frac{\omega(s)}{V(s)} = \frac{K}{(J.s+b)(L.s+R)+K^2} \quad (19)$$

We apply PID controller to the system so that get the best step response presented on figure 4.

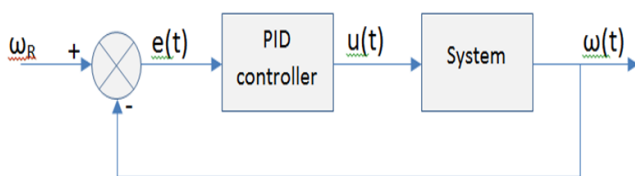


Fig. 4. Schema of the system with a PID controller.

We use ZN method 2 and PSO, GA methods to specify  $K_P$ ,  $K_I$  and  $K_D$  of PID controller for the model and we obtain results are described in table 1. To compare performance of tradition and two random search methods, we use unit step change of rotor speed. Figures 5, 6 illustrate the convergence of GA and PSO algorithms. Figures 7 to 9 show unit step response of system with PID controller tuned by traditional and heuristic search methods.

TABLE I. COMPARISON PERFORMANCE OF TRADITIONAL AND RANDOM SEARCH METHODS FOR TUNING PID CONTROLLER

Method	$K_P$	$K_I$	$K_D$	ITAE
Traditional tuning method	15.00	75.00	0.80	5.5759
GA tuning method	17.96	29.75	0.11	3.3687
PSO tuning method	16.53	29.32	0.03	3.4135

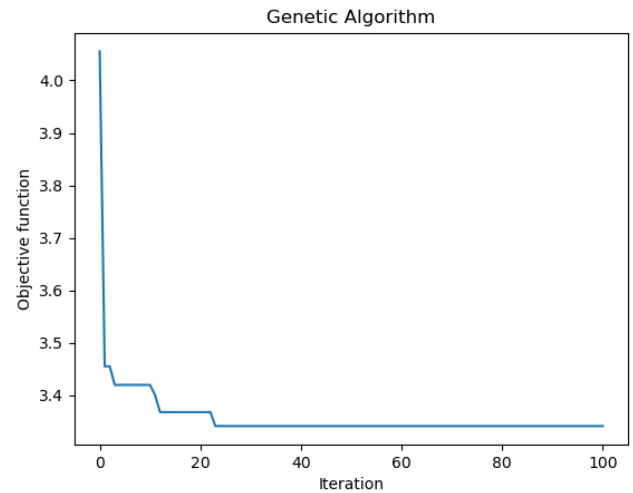


Fig. 5. Convergence of the GA algorithm.

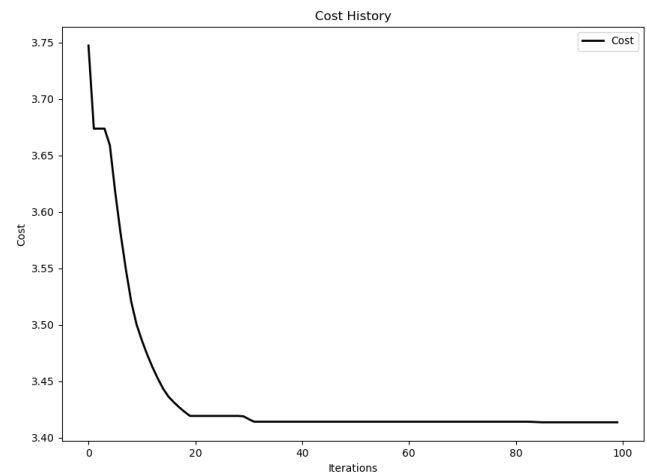


Fig. 6. Convergence of the PSO algorithm.

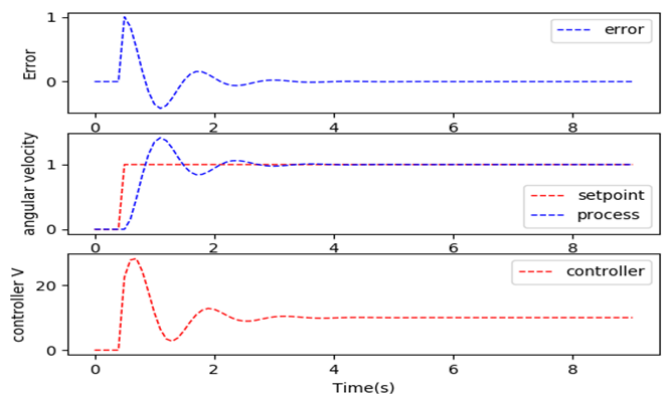


Fig. 7. Step response of the system with ZN tuning method.

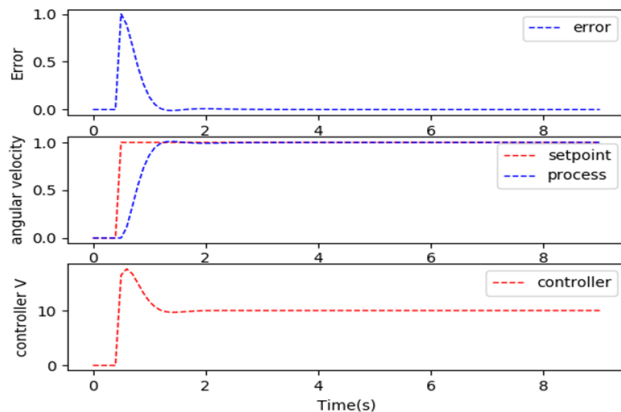


Fig. 8. Step response of the system with GA tuning method.

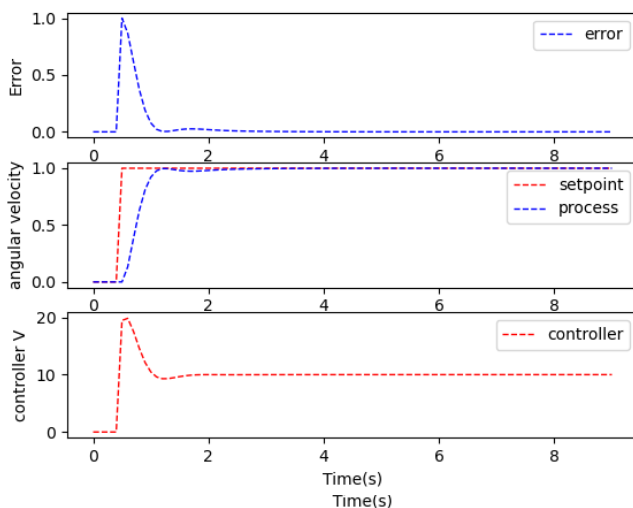


Fig. 9. Step response of the system with PSO tuning method.

## VI. CONCLUSION

In this paper, we present traditional rules and meta-heuristic random search algorithms PSO, GA for tuning PID controller. PSO and GA use the random search approach to find global optimal solution for given problem. To test performance of

applying PSO, GA for tuning PID controller, we perform simulation on DC motor model. The simulated results show that heuristic random search methods are better than traditional rules with respect to performance integral indices. In the future, we will study to solve multi-objective optimization problems using PSO and GA algorithm.

## ACKNOWLEDGMENT

This work is supported by the Hanoi University of Industry.

## REFERENCES

- [1] M. Araki, PID Control in Control systems, Robotics and Automation, vol II, edited by Heinz Unbehauen, Encyclopedia of Life Support Systems (EOLSS), Developed under the Auspices of the UNESCO, Eolss Publishers, Oxford, UK.
- [2] Qiao W, Venayagamoorthy GK, Harley RG (2006), Design of optimal PI controllers for doubly fed induction generators driven by wind turbines using particle swarm optimization. In: IEEE 2006 international joint conference on neural networks (IJCNN '06), Georgia Institute of Technology, Atlanta, pp. 1982–1987.
- [3] Ioan Constantin Tarca, Komal Khuwaja D/O Shoukat Ali Khuwaja, PID Controller Tuning Optimization with Genetic Algorithms for a Quadcopter, Recent Innovations in Mechatronics (RIIM) Vol. 5. (2018).
- [4] Ian Griffin, On-line PID Controller Tuning using Genetic Algorithms, Dublin City University, 2003.
- [5] Allaoua B, Gasbaoui B, Mebarki B, Setting up PID DC Motor speed control alteration parameters using particle swarm optimization strategy, Leonardo Electron J Pract Technol 4:19–32.
- [6] Brian R Copeland, The Design of PID Controllers using Ziegler Nichols Tuning, March 2008.
- [7] Fatiha Loucif, dc motor speed control using pid controller, ICCAS2005 June 2-5, KINTEX, Gyeonggi-Do, Korea.
- [8] Cuong Nguyen Cong, Ricardo Rodriguez-Jorge, Nghien Nguyen Ba, Chuong Trinh Trong, Nghia Nguyen Anh, Design of Optimal PI Controllers using the Chemical Reaction Optimization Algorithm for Indirect Power Control of a DFIG model with MPPT, Web, Artificial Intelligence and Network Applications. WAINA 2020. Advances in Intelligent Systems and Computing, vol 1150. Springer, Cham. [https://doi.org/10.1007/978-3-030-44038-1\\_114](https://doi.org/10.1007/978-3-030-44038-1_114).
- [9] Zhang, S. Balochian, P. Agarwal, V. Bhatnagar, and O. J. Housheya, Artificial intelligence and its applications, Mathematical Problems in Engineering, vol. 2014, Article ID 840491, 10 pages, 2014.