International Journal for Modern Trends in Science and Technology
Volume 10, Issue 04, pages 222-226.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol10issue04.html
DOI: https://doi.org/10.46501/IJMTST1004033

Survey on the quality of automatic parameter detection of PID controller for DC motor using Genetic Algorithm and Particle Swarm Optimization

Nhat Quang Dao

Control, Automation in Production and Improvement of Technology Institute (CAPITI), Academy of Military Science and Technology (AMST), Hanoi, Vietnam.
Corresponding author Email Id: quanglabotech@gmail.com

To Cite this Article
Nhat Quang Dao, Survey on the quality of automatic parameter detection of PID controller for DC motor using Genetic Algorithm and Particle Swarm Optimization, International Journal for Modern Trends in Science and Technology, 2024, 10(04), pages. 222-226.https://doi.org/10.46501/IJMTST1004033

Article Info
Received: 22 March 2024; Accepted: 05 April 2024; Published: 14 April 2024.

ABSTRACT

This article presents the results of a study on selecting optimal PID parameters tuned by Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) used for a DC motor. The simulating controller response results show that the PID - GA and PID - PSO combination algorithms are superior to traditional methods. The result also allows for the selection of the optimal algorithm - combining the PSO - PID to design a controller that has smaller settling error but larger overshoot and settling time compared to GA-PID method. The simulation was taken in Matlab environments.

*KEYWORDS— **Particle Swarm Optimazation, Genetic Algorithms, DC motor, Optimal.***

1. INTRODUCTION

Nowadays, PID controller (Proportional - Integral - Derivative controller) is very commonly used in industrial systems, due to its effective control ability, simplicity in design and wide range of applications. There are many methods to calibrate PID controller parameters, the most popular is the Ziegler-Nichols method. However, for some systems, calibrating the PID controller using this method requires a rather time-consuming experimental process due to the influence of noise and errors of the devices on the measurement signal, leading to misalignment. Adjusting the parameters of the PID controller is difficult to achieve good values. In this case, PID tuning methods combined with neural networks, genetic algorithms (GA- PID) [1] or particle swarm optimization algorithms (PSO-PID) [2] are better tuning method.

In this article, the authors use PSO and GA to tune PID controller parameters in control the speed of a DC motor. Simulation results show that the GA-PID controller [2] has fast response, low overshoot and smaller settling time than the PSO-PID controller [4]. But PSO-PID controller has smaller error after settling.

2 METHODOLOGY

2.1 Dynamic model of a DC motor

Dynamic model of a DC motor is mentioned in many literatures [1]:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -a_1 x_1 - a_2 x_2 + f(t) - bu \end{cases} \quad (1)$$

Where a_1, a_2, b are positive parameters; x_1, x_2 are the velocity and acceleration errors of the motor rotation angle, respectively, and *f(t)* is the expressions of load placed on the motor shaft. Assume that *f(t)* is internal limitless load:

$$|f(t)| \le M \quad (2)$$

Where M is a known positive value.

2.2 Tuning PID controller by classic Ziegler-Nichols method

The Ziegler–Nichols tuning method (ZN) is a heuristic method of tuning a PID controller. It was developed by John G. Ziegler and Nathaniel B. Nichols. It is performed by setting the I (integral) and D (derivative) gains to zero. The "P" (proportional) gainK_p, is then increased (from zero) until it reaches the ultimate gain K_u, at which the output of the control loop has stable and consistent oscillations. $K_u$and the oscillation period $T_u$are then used to set the P, I, and D gains depending on the type of controller used and behaviour desired as in Table 1.

Table 1. Ziegler -Nichols method

Control Type	K_p	T_i	T_d	K_i	K_d
P	$0.5K_u$	-	-	-	-
PI	$0.45K_u$	$0.83T_u$	-	$0.54K_u/T_u$	-
PD	$0.8K_u$	-	$0.125T_u$	-	$0.1K_u/T_u$
Classic PID	$0.6K_u$	$0.5T_u$	$0.125T_u$	$1.2K_u/T_u$	$0.075K_u/T_u$

2.3 Automatic parameter detection using genetic algorithm

Objective function: Is the function used to evaluate the solutions of the problem. Depending on each problem, the objective function is different. Because the desired requirement is to minimize the output error ($e(t)$), the objective function can be chosen as follows: [5]

$$fitness=\int_0^T e(t)^2 dt \quad (3)$$

In the GA algorithm, each element will contain 3 parameters K_p, K_i and K_d, from which we will have the algorithm flow chart of the PID-GA control system as follows:

<u>Step 1</u>: Create an initial random population including K_p, K_i and K_d.

<u>Step 2</u>: Set up PID and simulate the closed loop system to determine the error e(t).

<u>Step 3</u>: Estimate the objective function value.

<u>Step 4</u>: Check convergence

Step 4.1: If convergence occurs, save the K_p, K_i and K_d values. End the loop

Step 4.2: If not yet converged

Step 4.2.1: Select

Step 4.2.2: Breeding

Step 4.2.3: Mutation

<u>Step 5</u>: Give birth to a new generation.

<u>Step 6</u>: Repeat step 2 until convergence

2.4 Automatic parameter detection using PSO

In the PSO algorithm, each element will contain 3 parameters K_p, K_i and K_d, which means that the search space is the above 3 parameters, from which we will have the algorithm flow chart of the PSO-PID control system. as follows: [2], [5]

<u>Step 1</u>: Initialize each i-th individual in the population:

Step 1.1: Initialize the position value (X_i^k) for each individual in the population with a random position value.

Step 1.2: Initialize V_i^k velocity value.

<u>Step 2</u>: Run the model

Step 2.1: Run the control model with the given parameters

preset.

Step 2.2: Find parameters K_p, K_i and K_d of the PID.

Step 2.3: Find the objective function.

Step 2.4: Evaluate the position function X_i^k according to the objective function value (*fitness*).

<u>Step 3</u>: Update position and velocity values for each individual:

Step 3.1: Update velocity and position values according to formulas (4) and (5):

$$v_{i,m}^{(t+1)} = w.v_{i,m}^{(t)} + c_1.\text{rand}(1).(Pbest_{i,m} - x_{i,m}^{(t)}) + c_2.\text{rand}(1).(Gbest_{i,m} - x_{i,m}^{(t)}) \quad (4)$$

$$x_{i,m}^{(t+1)} = x_{i,m}^{(t)} + v_{i,m}^{(t+1)}; i = 1,2..,n; m = 1,2,..d \quad (5)$$

In there:

n: Number of herds; *d:* Population size (dimension);

t: Number of repetitions;

$v_{i,m}^{(t)}$:Velocity of the i-th element at the t-th iteration;

w: Inertial weight coefficient;

c1, *c2*: Acceleration coefficient;

rand(1): Is a random number in the range (0,1);

$x_{i,m}^{(t)}$: Position of the i-th element in the t-th iteration

Step 3.2: Evaluate the objective function (*fitness*)

Step 3.3: If *fitness* < P_{best}*_fitness* then

$P_{best}= X_i^k$, P_{best}*_fitness* = *fitness*.

Step 3.4: Update the G_{best} value for each individual corresponding to the current smallest position of the objective function in the population.

Step 4: Find the new element value

If the value of the new element is better than the best value of the previous element in the swarm, then replace the previous best value with the current new value.

Step 5: Repeat step 2 until you have enough repetitions.

The goals of the PID tuning method using the PSO algorithm are: Minimize the objective function.

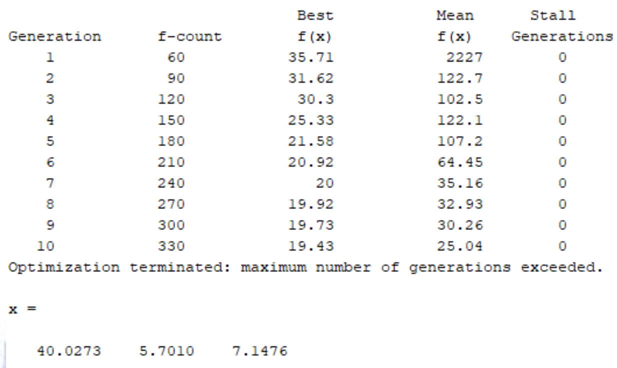
Find the response step of the system and reduce errors

Repeat the steps until you have enough repetitions.

3 SİMULATİON AND DİSCUSSİON

To simulate the two methods the common parameters must be defined. When the number of particles in PSO is similar to population size in GA, they will be denoted as N. The number of iterations in PSO is also similar to number of generations in GA, they will be denoted as r.

The article chooses to simulate two cases: when keeping r and changing N and keeping N changing r to compare the two methods. But firstly, compare the two of them together with the classic ZN method. The result was shown in Fig.3 where the PSO and GA method are better in many indicators.

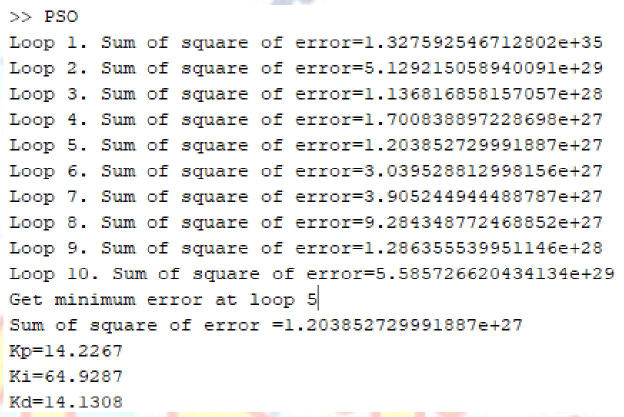


```
                                 Best           Mean      Stall
Generation      f-count          f(x)           f(x)    Generations
    1             60             35.71          2227        0
    2             90             31.62         122.7        0
    3            120              30.3         102.5        0
    4            150             25.33         122.1        0
    5            180             21.58         107.2        0
    6            210             20.92         64.45        0
    7            240                20         35.16        0
    8            270             19.92         32.93        0
    9            300             19.73         30.26        0
   10            330             19.43         25.04        0
Optimization terminated: maximum number of generations exceeded.

x =

   40.0273    5.7010    7.1476
```

Fig. 1.GA method running in Matlab



```
>> PSO
Loop 1. Sum of square of error=1.327592546712802e+35
Loop 2. Sum of square of error=5.129215058940091e+29
Loop 3. Sum of square of error=1.136816858157057e+28
Loop 4. Sum of square of error=1.700838897228698e+27
Loop 5. Sum of square of error=1.203852729991887e+27
Loop 6. Sum of square of error=3.039528812998156e+27
Loop 7. Sum of square of error=3.905244944488787e+27
Loop 8. Sum of square of error=9.284348772468852e+27
Loop 9. Sum of square of error=1.286355539951146e+28
Loop 10. Sum of square of error=5.585726620434134e+29
Get minimum error at loop 5
Sum of square of error =1.203852729991887e+27
Kp=14.2267
Ki=64.9287
Kd=14.1308
```

Fig. 2.PSO method running in Matlab

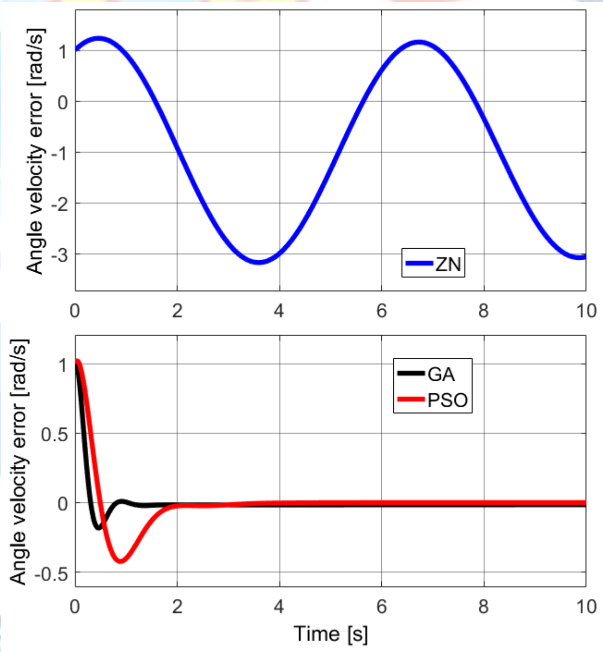


Fig. 3. Error of classic Ziegler-Nichols versus GA and PSO

Then first situation is shown in Fig.4 to Fig. 6 when r is keeping at one value (r=10) and N is being changed from 30 to 50. The results showed that the PSO method has the better settling value of error when GA has better settling time and smaller overshoot.

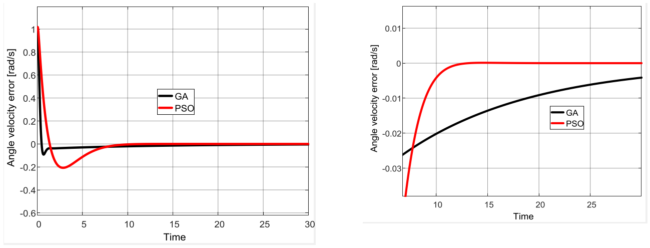


Fig. 4. Error of GA and PSO when number of particles/ population's size (N=30), number of iterations / generations (r=10)

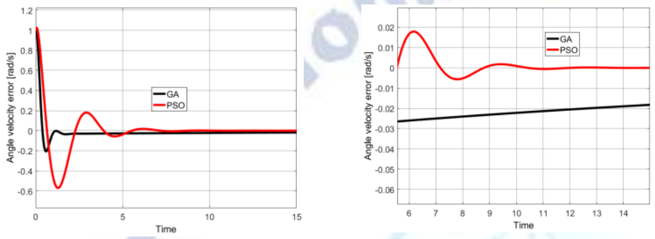


Fig. 5. Error of GA and PSO when number of particles/ population's size (N=40), number of iterations / generations (r=10)

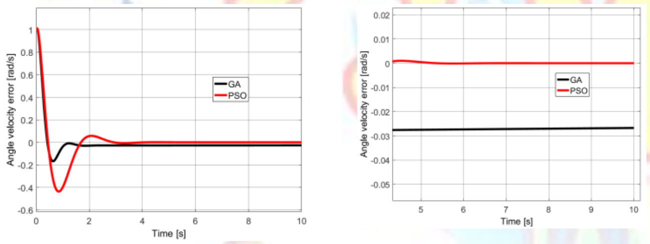


Fig. 6. Error of GA and PSO when number of particles/ population's size (N=50), number of iterations / generations (r=10)

The second situation is shown in Fig. 6 to Fig.9 when N is keeping at one value (N=50) and r is changing among 5, 10, 15 and 20. The results are the same when PSO has better settling value error and GA is better at settling time and overshoot value.

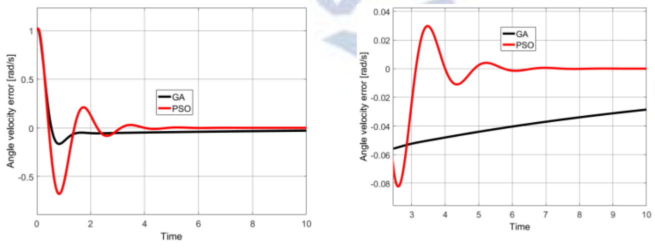


Fig. 7. Error of GA and PSO when number of particles/ population's size (N=50), number of iterations / generations (r=5)

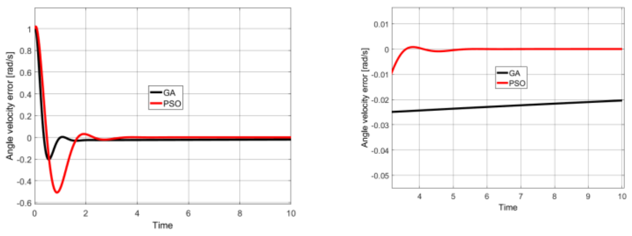


Fig. 8. Error of GA and PSO when number of particles/ population's size (N=50), number of iterations / generations (r=15)

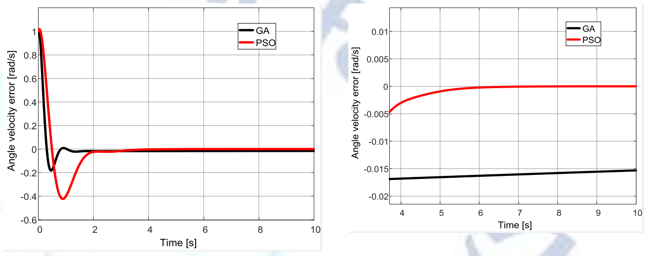


Fig. 9. Error of GA and PSO when number of particles/ population's size (N=50), number of iterations / generations (r=20)

Back in Fig.1 and Fig.2 where the listings of the two method on matlab environment were shown. The difference between them is that the GA method has better results after any iterations/ generations. Then wecan improve GA method by creating more generations while PSO has random results among different iterations.

Then in Fig. 10 we made a change when giving GA more generations with r=50 and N=20, when keeping N=20 for PSO and r at a very low value of 10. The result is not quite different form those cases when PSO is still better with settling value error and GA is better at overshoot value and settling time. Although the differences in settling value error is smaller between them from around 0.02 or larger to 0.008.

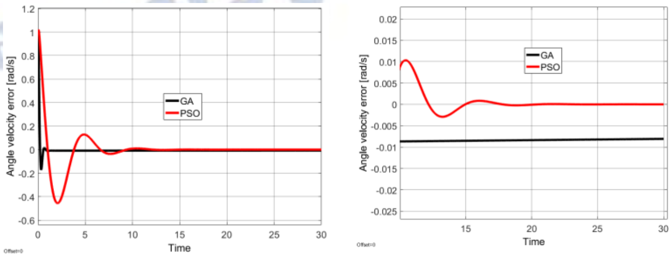


Fig. 10. Error of GA (N=20, r=50) and PSO (N=20, r=10)

4 CONCLUSİON

The article has presented a survey to compare GA and PSO to each other in auto tuning parameters of PID controler for a DC motor. The results showed that the PSO method has better settling value error when GA has better overshoot value and settling time. Although GA has better quality after every generation, the final settling error is still larger than PSO method. Research results are visualized by simulation in Matlab-Simulink software.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Tran Tan Khang, "Application of genetic algorithm (GA) to determine PID parameters in speed control of three-phase asynchronous motors", Ho Chi Minh City University of Technical Education, pages 67-72 , 2011.

[2] Huynh Duc Chan, "Application of swarm algorithm (PSO) to determine PID parameters in speed control of three-phase asynchronous motors" VCCA Conference- 2011.

[3] Johnson M.A. and M.H. MoradiM, "PID Control – NewIdentification and Design Methods" pp. 297-337. SpringerVerlag London Limited.ISBN-10: 1-85233-702-8, 2005.

[4] N. Pillay, "A Particle swarm optimization approach for tuningof SISO PID control loops", Durban university of technologydepartment of electronic engineering pp. 95-121, 2008

[5] Boumediene Allaoua Brahim Gasbaoui and Brahim Merbarki, "Setting Up PID DC Motor Speed ControlAlteration Parameters UsingParticleSwarmOptimization"Strategy, Bechar University Departementof Electrical Engineering B.P 417 BECHAR (08000) Algeria,pp. 19-32,2009.

[6] Chao Ou, Weixing Lin, "Comparison between PSO and GA for Parameters Optimization of PID Controller", The Faculty ofInformationScience and Technology University of NingBoUniversity of NingBo, pp. 2471-2475, 2006.

[7] Astrom, K.J. and T. Hagglund, "Automatic Tuning of PIDControllers. Instrument Society of America, Research TrianglePark" NC, 1988.