Tuning PID Controller for DC Motor: An Artificial Bees Optimization Approach

Mohammed E. El-Telbany
Computers and Systems Department
Electronics Research Institute
Egypt

ABSTRACT

Due to its excellent speed control characteristics, the DC motor has been widely used in industry even though its maintenance costs are higher than the induction motor. As a result, speed control of DC motor has attracted considerable research and several methods have evolved. PID controllers have been widely used for speed control of DC motor. In this paper the using Artificial Bees Colony (ABC) optimization algorithm for regulation parameters of PID controller of DC motors. The aim of the proposed method is to improve tracking performance of DC motor. At the end of the study, ABC algorithm showed better performance than the other population based optimization algorithm.

Keywords

Artificial Colony Optimization, Particle Swarm Optimization, Genetic Algorithms, DC-Motor, PID-Controller.

1. INTRODUCTION

Due to its excellent speed control characteristics, the DC motor has been widely used in industry even though its maintenance costs are higher than the induction motor. As a result, speed control of DC motor has attracted considerable research and several methods have evolved. To reduce the loading effect and minimize time delay, a Proportional-Integral-Derivative (PID) controller is added. The PID controller is by far the most dominating form of feedback in use today. Due to its functional simplicity and performance robustness, the proportional-integral-derivative controller has been widely used in the process industries [10]. The standard PID control configuration is as shown in Figure 1. This is a type of feedback controller whose output, a control variable, is generally based on the error between some user-defined set point and some measured process variable. A PID controller attempts to correct the error between a measured process variable and a desired set point by calculating and then outputting a corrective action that can adjust the process accordingly. So by integrating the PID controller to the DC motor were able to correct the error made by the DC motor and control the speed or the position of the motor to the desired point or speed. However, PID controllers cannot be tuned in such way that the optimum step response is achieved for different inertia, load and speed reference, to achieve the desired step response of the system has minimal rise time and without overshoot. For design and tuning of PID controller parameters K_p, K_i, K_d, an optimization methods are used. There are a variety of PID controllers tuning methods have been developed such as Ziegler-Nichols rules, Cohen-Coon rule and so on [1,2,8]. These methods are applied directly since they provide simple tuning rules to determine the PID parameters. However, since they rely on a minimum amount of dynamic information by making a certain assumption about nature of the controlled process, such as linearity, weak interactions within the process, absence of noise, etc.

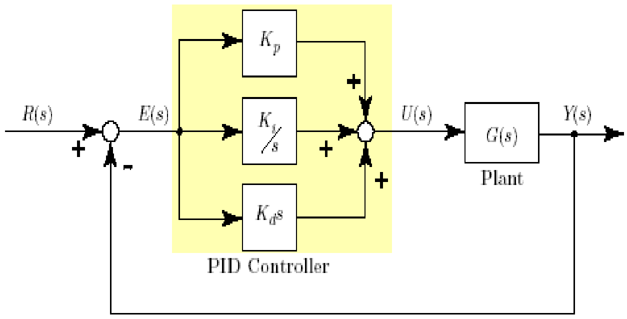

Fig 1: PID Controlled System

The realized closed loop response is less than optimum since the real world processes are non-linear and very complex. Since the PID controllers are usually poorly tuned for the systems with above said features, a higher degree of experience and technology are required for tuning. Recently, the inspiration from natural developed a host of natural-inspired algorithms in evolution (genetic algorithms, etc.), neurology (artificial neural networks), immunology (artificial immune systems), social networks (ant colony optimization, particle swarm optimization, bees algorithm, etc), and more [3] as opened paths to a new generation of advanced process control. Among the techniques found out, stochastic search techniques, particle swarm optimization and genetic algorithms optimization techniques have found themselves a place in tuning of the parameters [5,11,12,13,14]. As intelligent algorithms, genetic algorithm and particle swarm optimization have great superiority in tuning the parameters of PID controllers.

One of the new developed nature inspired algorithms is the artificial bees colony optimization algorithm has been particularly appealing various scientific circles. In the proposed work we compare the time domain specifications, the values of the performance measures the integral of square time error (ISTE) obtained by using the conventional technique i,e the Ziegler-Nichols method and our proposed method using bees colony optimization to prove that our method is better than the conventional methods. In the section that follows we have given the explanation of our set up, a view of the conventional methods used, our proposed algorithm, the values obtained, the results and graphs and finally the conclusion.

2. Model of DC Motor

DC motors are widely used in industrial and domestic equipment. The control of the position of a motor with high accuracy is required. The electric circuit of the armature and the free body diagram of the rotor are shown in Fig. 1.

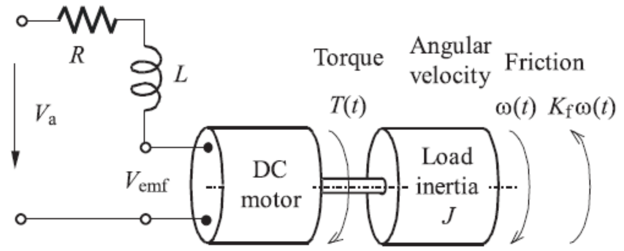

Fig 2: The structure of a DC motor

The applied voltage V_a controls the angular velocity $\omega(t)$. The relations for the armature controlled DC motor are shown schematically in Fig. 2.

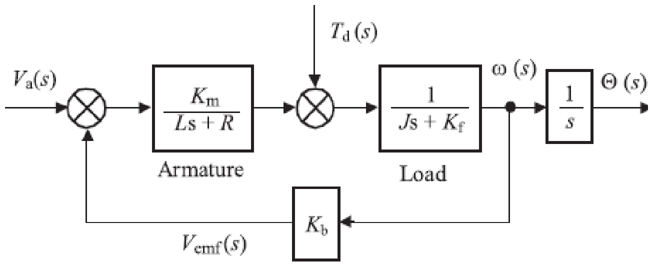

Fig 3: Mathematical model of DC motor

The transfer function (with $T_d(s)$) has the form [4]:

$$G(s) = \frac{\theta(s)}{V_a(s)} = \frac{Km}{s\left[(Ls+R)(Js+K_f)+K_bK_m\right]} \tag{1}$$

It is of interest to note that $K_b = K_m$. For our DC motor the physical constants are:

$R = 1\Omega$, $L = 0.5H$ $K_m = K_b = 0.01,$ $K_f = 0.1Nms$ and $J = 0.02kgm^2/s^2$. For these motor constants the transfer function (2) of the DC motor has the form:

$$G(s) = \frac{0.01}{0.005S^2 + 0.06S + 0.1} \tag{2}$$

3. Artificial Bees Colony Optimization

The artificial bee colony (ABC) algorithm, which simulates the foraging behavior of honey bee colonies, was recently proposed by Karaboga [9]. It is a very simple, robust and population based stochastic optimization algorithm. In ABC algorithm, possible solution of the problem is represented by the food source. Quality of solution is indicating by the amount of nectar amount of a particular food source. The employed bees, onlooker bees, and scout bees are tasked with finding optimum food sources, and first the food source positions are generated randomly [9]. The detailed pseudo-code of the ABC algorithm is given below:

Table 1: The ABC Optimization Algorithm

```
function ABC(Population, Fitness-FN) returns a solution
inputs: Population, a set of solutions by random search strategy
        Fitness-Fn, a the solution response according Eq. (7).
Evaluate(Population, Fitness-Fn)
while (stopping criterion is not met) do
   Loop for i from 1 to Size (Population) do
        Generate_Solution(Population, Local-Strategy)
        Evaluate(Population, Fitness-Fn)
        Generate_Solution(Population, Selection-Strategy)
        Evaluate(Population, Fitness-Fn)
        if (abandoned solutions exist)
           Generate_Solution(Population, Random-Strategy)
           Evaluate(Population, Fitness-Fn)
        end if
   end loop
   Memorize_Best_Solution(Population)
end
return the best solution
```

Firstly, half of the colony consists of the employed bees and the second half consists the onlookers, the number of employed bees is equal to the number of food sources which is also equal to the number of onlooker bees. The relation between employed bee and the food source is one-to-one, that means that there is only one employed bee per each food source. If a food source becomes abandoned, mapped employed bee to that food source becomes a scout, and as soon as it finds a new food source, it again becomes employed. ABC algorithm, as an iterative algorithm, starts by associating each employed bee with randomly generated food source (solution). In each iteration, each employed bee discovers a food source in its neighborhood and evaluates its nectar amount (fitness) using Equation (3), and computes the nectar amount of this new food source:

$$v_{ij} = x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right) \tag{3}$$

Where ϕ_{ij} is a random number between $[-1,1]$.If fitness of new food source is better than the fitness of the old one, employed bee moves to the new source, otherwise it retains the old one. After all employed bees complete the search process, they share the information about their food sources with onlooker bees. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability p_i that is proportional to the fitness of the food sources, using Equation (4).

$$p_i = \frac{fit_i}{\sum_{i=1}^{m} fit_i} \tag{4}$$

where fit_i is the fitness value of the solution i which is proportional to the nectar amount of the food source in the position i and m is the number of food sources which is equal to the number of employed bees. The employed bee becomes a scout bee when the food source which is exhausted by the employed and onlooker bees is assigned as abandoned. In that position, scout generates randomly a new solution by Equation (5).

$$x_i^j = x_{\min}^j + \mathrm{rand}(0,1)\left(x_{\max}^j - x_{\min}^j\right) \tag{5}$$

Assume that the abandoned source is x_i and $j \in \{1,2,\ldots,D\}$ then the scout discovers a new food source to be replaced with x_i.

4. PID Controller for DC Motor Design

The aim of the control design is to provide required static and dynamic behavior of the controlled process. Usually, this behavior is represented in terms of the well-known concepts referred in the literature: maximum overshoot, settling time, decay rate, steady state error or various integral performance indices [4].

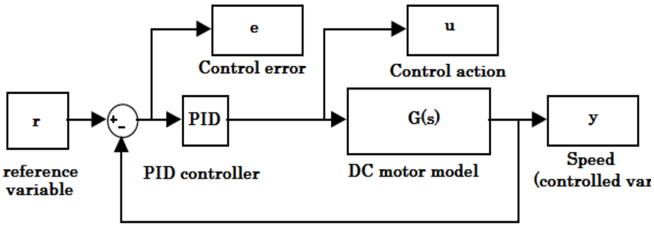

Fig 4: closed loop control for DC motor

Without loss of generality let us consider a feedback control loop as shown in Figure.4, where angular velocity y is the regulated variable, input voltage u is the manipulated variable, r is the reference variable of angular velocity and e is the control error $e=(r-y)$. The controller design principle is actually an optimization task - search for such controller parameters from the defined parameter space, which minimize the performance index. The performance index is a mapping $R^n \to R$, where n is the number of designed controller parameters. The performance index can to represent sum of function f in control errors as the following forms:

$$J=\sum_{i=1}^{N} f(r_i - y_i)=\sum_{i=1}^{N} f(e_i) \qquad (6)$$

where r_i is reference variable, y_i is controlled output, e_i is control error and N is number of samples. Fitness is represented by the performance index or in the case of control, by the modified performance index, which can be penalized for example by derivation of process output y or by overshoot of process output or by derivation of control action u. Therefore, the objective function is defined using the performance index of the integral of the square of the error (ISTE) [7].The fitness function is reciprocal of the performance criterion, in the other words, the fitness of the food source is defined as:

$$fitness=\frac{1}{I_{ISE}} \qquad (7)$$

The tuning of ABC-based PID controller' results will be compared to those obtained from the PSO, GA and traditional techniques. The ABC, PSO and GA tuning algorithms are conducted several pre-experiments to determine the parameters setting per algorithm that yields the best performance. For ABC, all bees start at a random position in the range [0.0, 500.0] for each dimension. The colony size is 50, control parameter in order to abandon the food source is 100. For PSO, all swarm particles start at a random position in the range [0.0, 500.0] for each dimension. The velocity of each particle is randomized to a small value to provide initial random impetus to the swarm. The swarm size was limited to 50 particles. The must important factor is *maximum velocity* parameter which affect the convergence speed of the algorithm is set to 100.0. The c_1 and c_2 are 2.0 and 2.0 respectively. For GA, the GAOT is used [9] with population size of 50, the search range [0.0, 500.0] and with other default parameters. The three algorithms are runs of 100 iterations. The tuned parameters of the self-tuning PID controller through each algorithm are represented in Table 2 and the step response curves which are obtained from the output of these controllers are shown in Figures 5 and 6.

Table 2: Comparisons of Steady State Responses

	ABC	GA	PSO	ZN
K_p	249.58	256.38	249.74	98.9
K_i	500	497.02	500	200
K_d	24.92	28.08	24.97	12.3

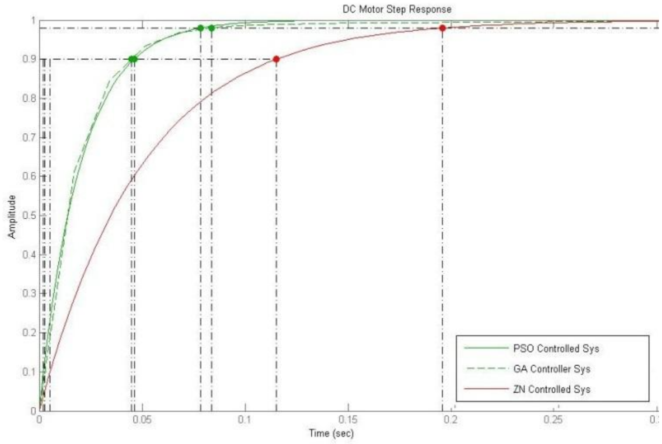

Fig 5: The step response of different tuning algorithms

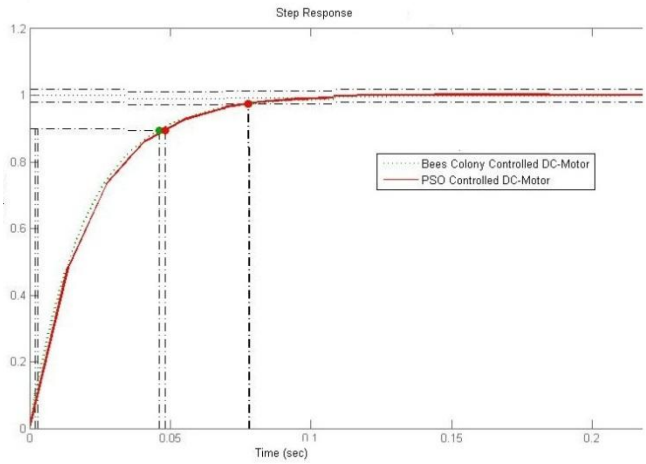

Fig.6: The step response of the best tuning algorithms

The settling times and the rise time of step response curves are measured and represented as Table 3, in order to determine the performances of the DC-Motor system.

Table 3: Comparisons of Steady State Responses

	ABC	PSO	GA	ZN
Rise Time	0.43	0.045	0.046	0.11
Settling Time	0.078	0.078	0.084	0.196

Both the ABC and PSO-tuned PID controllers outperform the GA-tuned and Ziegler-Nichols tuned controller in terms of settling time and rise time. However, the ABC showed better performance than PSO. Analyzing the transfer function of the DC-motor system is stable as it has three poles located on the left hand side of the s-plane and one critically stable pole located at the origin.

5. Conclusions and Future Work

One of the new developed algorithms is the artificial bees colony optimization algorithm has been particularly appealing various scientific circles, primarily because such algorithms allow autonomous adaptation/optimization of controller parameters without human intervention, are not easily trapped in locally optimal solutions, allow parallel exploitation of an optimization parameter space, and do not require gradient evaluation of the objective function. In the proposed work we compare the time domain performance of the PID controller for DC-motor system tuned by ABC algorithm. The ABC algorithm showed better performance than the other population based optimization algorithm.

6. References

[1] Astrom K., and T. Hagglund T., "The future of PID control", *Control Engineering Practice*, 9, pp-1163-1175, 2001.

[2] Astrom, K., and Hagglund, T., PID controllers: Theory, design, and tuning. 2nd Edition, Instrument Society of America, 1995.

[3] de Castro, L. (2007). Fundamentals of Natural Computing: An Overview, Physics of Life Reviews (4) 1–36.

[4] Dorf, R. C., and Bishop, R. H. : Modern Control Systems, Addison-Wesley, New York, 1990.

[5] El-Telbany M., Employing Particle Swarm Optimizer and Genetic Algorithms for Optimal Tuning of PID Controllers: A Comparative Study, In the International Journal on Automatic Control and System Engineering (ACSE), V7, Issue 2, pp. 49-54, ICGST, 2007.

[6] Griffin I., On-line PID Controller Tuning using Genetic Algorithms, Dublin City University, 2003.

[7] Halevi, Y., Palmor, Z.J., and Efrati, T., Automatic tuning of decentralized PID controllers for MIMO processes. *Journal of Process Control*, 7, 2, 119–128, 1997.

[8] Houck, C., Joines J., and Kay M., A Genetic Algorithm for Function Optimization: A MATLAB Implementation. ACM Transactions on Mathematical Software, 1996.

[9] Karaboga D, Basturk B. A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. J. Global. Optim. 39:459-471, 2007.

[10] Leva A., "PID autotuning algorithm based on relay feedback", IEE Porc., Control Theory Appl., vol. 140, 1993, pp. 328-337.

[11] Liu Y., Zhang J., and Wang S., "Optimization design based on PSO algorithm for PID controller", *Intelligent Control and Automation*, WCICA. pp. 2419 – 2422,. 2004.

[12] Ou C. and Lin W., "Comparison between PSO and GA for parameters optimization of PID Controller," Proc IEEE Int. Conf. on Mechatronics and Automation, pp. 2471-2475, 2006.

[13] Vlachos .C, Williams D., and Gomm J. "Solution to the Shell standard control problem using genetically tuned PID controllers," Control Engineering Practice, vol.10, no.2, pp.151-163, 2002.