International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

*Research Article*

# Optimally Tuned PID Controller of A DC Motor System using Bacterial Foraging Algorithm

**Rashid Alzuabi**†*

†Electronics Engineering Technology Department, College of Technological Studies, PAAET, Kuwait

Received 13 Jan 2018, Accepted 21 March 2018, Available online 24 March 2018, **Vol.8, No.2 (March/April 2018)**

***Abstract***

*This paper presents an exercise in applying the bacterial foraging algorithm (BFA) optimisation method on a proportional—integral-derivative controller (PID) of a DC motor circuit. The paper presents the system description of the DC motor transfer function and the simulation of the close loop system using MATLAB. The BFA algorithm is described and discussed with the simulation results presented to illustrate the enhancement of the system response that in result enhances the operation of the DC motor system.*

***Keywords:*** *PID, Bacterial Foraging optimization, feedback control, DC motor*

## 1. Introduction

One of the most important characteristics of deploying a direct-current (DC) motor is to maintain a proper control to maximize the efficiency and performance. In addition, a proper control algorithm will elongate the lifetime of DC motor and reduce its wear and tear of usage. This paper illustrates an exercise of applying a bacterial foraging algorithm optimisation (BFA) method to tune a proportional-integral-derivative (PID) controller for a DC motor. The paper sections present the system mathematical model in section 1, Section 2 presents the BFA algorithm. Section 3 illustrates the simulation methodology and parameters. Lastly, the fourth section illustrates the results and discussions of simulations.

## 2. System description and modelling

The DC engine circuit is represented in figure 1. Table 1 exhibits the circuit parameters explanations and standards.

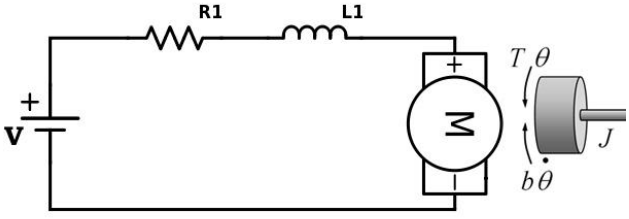


**Fig.1** DC motor circuit

**Table 1** DC motor parameters

| Parameter | Description | Value | Unit |
|---|---|---|---|
| V | Voltage source | 1 | Volts |
| R | Resistor | 1 | Ohm |
| L | Inductor | 0.5 | Henry |
| T | Motor torque | 0.01 | N.m/A |
| Ke | Electromotive force | 0.01 | V/rad/sec |
| J | Moment of inertia | 0.01 | Kg.m$^2$ |
| B | Motor viscous friction | 0.1 | N.m.s |

we assume that the magnetic field is consistent and the DC engine is armature controlled DC engine where the torque is relative to the armature current by a steady K as:

$$T = K_T i$$

The back electromotive force (EMF) is relative to the angular velocity of the motor shaft as shown in the equation:

$$e = K_e \dot{\theta}$$

Referring to the circuit schematic of Figure 1, these equations are derived:

$$J\ddot{\theta} + b\dot{\theta} = Ki \quad (1)$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta} \quad (2)$$

Laplace transformation is applied to equation 1 and 2 correspondingly to get:

*Corresponding author's ORCID ID: 0000-0001-9868-4735,
DOI: https://doi.org/10.14741/ijcet/v.8.2.16

$$s(Js + b)\theta(s) = KI(s) \quad (3)$$

$$(Ls + R)I(s) = V(s) - Ks\theta(s) \quad (4)$$

Thus, transfer function of the system can be written as:

$$G(s) = \frac{\theta(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2}$$

## 3. Bacterial Foraging optimization Algorithm (BFA)

BFA was highly presented by [Passino, 2000; Passino 2002]. The process imitates and resembles the way the bacteria lives and searches the proper place. E. Coli managed to use a group of movements to make it able to swim and search for food and be away from the toxin locations. E. Coli bacterium has the type of body that enable it to swim and take directions easily through tumbling.

According to [Passino, 2002], the bacterium often swims in the middle locations seeking for the nutrients and in case of the area they search in is noxious, bacteria can release the chemical substance and fall to escape. For areas that are neutral which have no nutrients or noxious elements, the bacteria manage to be in continuous movement to change the direction to find the desired area. In case the area is nutrient-rich, the bacteria swim directly to the area and call the other bacteria in the medium through releasing attractant. So, what motivates researchers to model bacteria's optimization method is the exponential growth rate. High-fitted bacteria can reproduce and other bacteria eliminate to achieve the high-supplement mediums as soon and accurate as possible.

They are considered models done through a group of the continuous processes called dispersal events, chemotaxis, elimination, swarming and the reproduction.

The applied methods for penalizing the function are utilized to change constrained problem into the unconstrained problem using the addition of the function of penalty P(x) to target work when the requirement is damaged. That can be described in:

$$J(x) = \begin{cases} x \in feasible\ region \\ J(x) + P(x) \quad x \in feasible\ region \end{cases}$$

In the cost function, P(x) refer to penalty function and (x) refer to cost function. Through addition of the penalty to cost function, the outcome of cost is high. That can be

$$P(x) = \begin{cases} 0 \in feasible\ region \\ 10.J(x) \ \ x \in feasible\ region \end{cases}$$

**Table 2** BFA optimisation parameters

| Parameter | Description |
|---|---|
| *p* | The dimension of the search space |
| *S* | Total number of the bacteria in the population. S need to be even. |
| *Nc* | Number of chemotactic phases of the bacterium lifetime among reproduction steps |
| *Ns* | Number of the swims of the bacterium in the similar direction |
| *Nre* | Number of reproduction steps |
| $p_{ed}$ | Probability of bacterium to spread |
| *i=1,2,3,...S* | Index of bacterium |
| *j=1,2,3,...Nc* | Index of chemotaxis |
| *k=1,2,3,...Nre* | Index of reproduction steps |
| *l=1,2,3,...Ned* | Index of removal and dispersal events |
| $m_s$*=1,2,3,...Ns* | Index of the number of swims |
| *J* | The rate of function value |
| *C* | Step size of the tumble of the bacterium |

## 4. Classical BFA optimisation algorithm:

The BFA optimisation pseudo code as described by [Passino, 2002] is as follows:

1) E limination and dispersal loop: for $l = 1,2,3,...,N_{ed}$ do $l = l + 1$
2) Reproduction loop: for $k = 1,2,3,...,N_{re}$ do $k = k + 1$
3) Chemotaxis loop: for $j = 1,2,3,...,N_c$ do $j = j + 1$
   a) For $i = 1,2,3,...,S$, take a chemotactic step for bacterium i as follows
   b) Compute the nutrient media (cost function) value $J(i,j,k,l)$ calculate $J_{(i,j,k,l)} = J(i,j,k,l) + J_{cc}(q^i(j,k,l), P(j,k,l))$ (i.e., add on the cell-to-cell attractant effect to the nutrient concentration). If there is no swarming effect then $J_{cc}(q^i(j,k,l), P(j,k,l)) = 0$
   c) Put $J_{last} = J(i,j,k,l)$ to save this value since a better cost via a run may be found.
   d) Tumble: generate a random vector $\Delta(i) \in \Re^r$ with each element $\Delta_{m_r}(i), m_r = 1,2,3,...,r$ a random number on the range [-1,1]
   e) Move: Let $q^i(j+1,k,l) = q^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$ be the result in a step of size C(i) in the direction of the tumble of bacterium i
   f) Compute the nutrient media (cost function) value $J(i,j+1,k,l)$, and calculate $J_{(i,j+1,k,l)} = J(i,j+1,k,l) + J_{cc}(q^i(j+1,k,l), P(j+1,k,l))$. If there is no swarming effect then $J_{cc}(q^i(j+1,k,l), P(j+1,k,l)) = 0$
   g) Swim (note that we use an approximation since we decide behavior of each cell as if the bacteria numbered {1,2,...,*i*} have moved and {*i*+1,*i*+2, *i*+3, ...,S} have not; this is much simpler to simulate than simultaneous decisions about swimming and tumbling by all bacteria at the same time):
      - Let $m_s = 0$ (counter for swim length)
      - While $m_s < N_s$ ( if have not climbed down too long)

- Count $m_s = m_s + 1$
- If $J(i, j+1, k, l) < J_{last}$ ( if doing better), then $J_{last} = J(i, j+1, k, l)$ and calculate $q^i(j+1, k, l) = q^i(j, k, l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$ this results in a step of size C(i) in the direction of the tumble for bacterium *i.* Use this $q^i(j+1, k, l)$ to compute the new $J(i, j+1, k, l)$ as in step f above.
  - Else let $m = N_s$ (the end of the while statement)

h) Go to the next bacterium $(i+1)$ if $i \neq S$ (i.e. go to step b above) to process the next bacterium.

4) If $j < N_c$ go to step 3.

5) Reproduction:

a) For the given $k$ and $l$, for each $i = 1, 2, 3, \ldots, S$, let $J^i_{health} = \sum_{j=1}^{N_c+1} J(i, j, k, l)$ be the health of bacterium $i$ ( a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters C(i) in an ascending order since that higher cost means a lower health.

b) The $S_r$ bacteria with the highest $J_{health}$ values die and the other $S_r$ bacteria with the best value splits ( and the copies that were made are replaced at the same location as their parent)

6) If $k < N_{re}$ go to step 2.

7) Elimination-dispersal: for $i = 1, 2, 3, \ldots, S$, eliminate and disperse each bacterium which has probability value less than $p_{ed}$. If one bacterium is eliminated then it is dispersed to random location of nutrient media. This mechanism makes computation simple and keeps the number of bacteria in the population constant

For m=1:S

If $p_{ed} > rand$ (generate random number for each bacterium and if the generated number is smaller than $p_{ed}$ then eliminate/disperse the bacterium)

Generate new random positions for bacteria

Else

Bacteria keep their current position (not dispersed)

End

end

8) If $l < N_{ed}$ then go to step 1; otherwise end

## 6. Simulation method

With the parameters of the system presented in Table 1 and using the transfer function of equation 5, we can simulate the system using Matlab Simulink block diagram approach. The block diagram is presented in Figure 2 below.

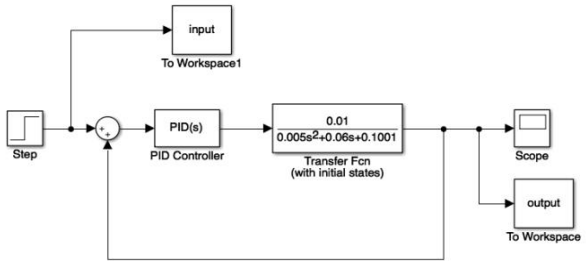


**Fig.2** Simulation of the system in Simulink

The heuristically tuned PID controller was connected to the closed loop system of Figure 2 and the simulation result of the step response of the system illustrated in Figure 3.

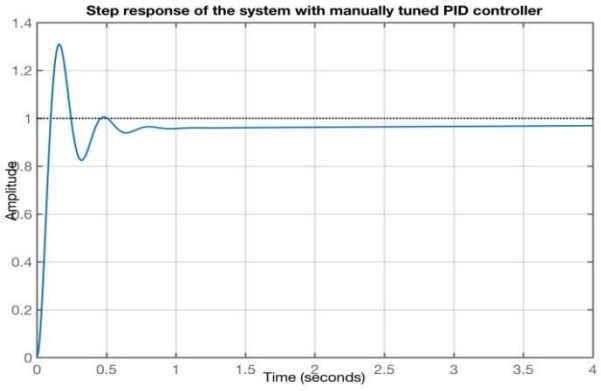


**Fig.3** Step response of the system with manually tuned PID controller

The step response resulted a poorly controlled system due to the high overshoot and oscillations of the settling time as well as the steady state error. This is due to the heuristically tuned PID controller parameters that needs fine tuning to achieve a better response.

With the BFA algorithm linked to the Simulink diagram using a Matlab code file (M-File), the system can be simulated with a number of iterations until a satisfactory response is achieved and the PID control parameters can be extracted.

By applying BFA algorithm has optimized the PID gain values within 40 iterations. The convergence of the cost function of the BFA is presented in Figure 4.

The parameters of the simulation were selected as:

| | | | |
|---|---|---|---|
| P= 3 | S= 120 | Nc= 20 | Ns= 3 |
| Nre= 4 | Ped= 0.4 | Sr= 0.5S | Ned=2 |

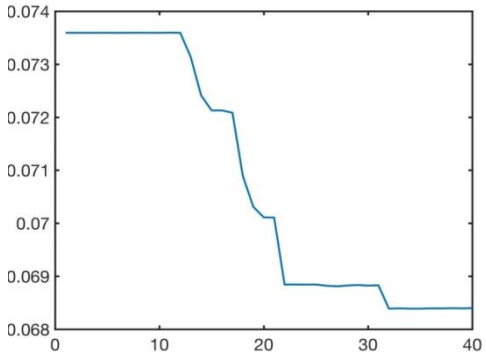


**Fig.4** Cost function convergence

The optimized controlled system performance plot is presented in Figure 5 below. It can be clearly noticed that the optimal gains that were found using the BFA algorithm have enhanced the system response by overriding the transient response into a smoother transition towards the set point and thus enhancing the operation of the DC motor.

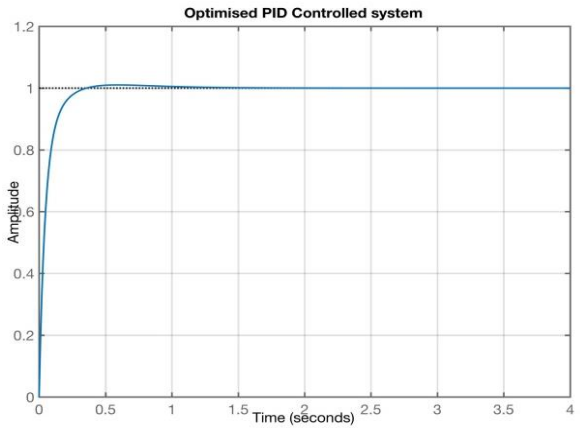


**Fig.4** Optimized closed-loop step response of the system

## Conclusion

The control of a DC motor using an optimized PID controller has been presented in this paper. BFA optimisation method has been utilised to achieve the properly tuned gains of the PID controller to enhance the operation and the response of the DC motor system. The system has been implemented and tested using MATLAB package and the BFA algorithm was linked to the system Simulink block diagram. The optimal gains of the controller were used to simulated the system and the results were smoother and more stable than the heuristically tuned PID controller. The results were illustrated and discussed to prove the feasibility of this exercise of application on DC motor systems.

## References

Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. IEEE control systems, 22(3), 52-67.

Liu, Y., & Passino, K. M. (2000). Swarm intelligence: Literature overview. Department of electrical engineering, the Ohio State University.

Thomas, N., & Poongodi, D. P. (2009, July). Position control of DC motor using genetic algorithm based PID controller. In Proceedings of the World Congress on Engineering (Vol. 2, pp. 1-3).

Passino, K. M. (2012). Bacterial foraging optimization. Innovations and Developments of Swarm Intelligence Applications, 219.

Morán, M. E. F., & Viera, N. A. P. (2017, October). Comparative study for DC motor position controllers. In Ecuador Technical Chapters Meeting (ETCM), 2017 IEEE (pp. 1-6). IEEE.