Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2014, 1(2):55-63**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Speed control of DC motors using PID-controller tuned by bacterial foraging optimization technique

WISAM NAJM AL-DIN ABED

Electronic Department, Engineering College, University of Diyala, Iraq

Abstract The aim of this work is to design proportional-integral-derivative (PID) controller based on bacterial foraging optimization (BFO) technique for speed control of separately excited dc motor (SEDM). The social foraging behavior of Escherichia (E. Coli) bacteria has been used to optimize the controller performance by adjusting it's parameters (Kp, Ki and Kd).The SEDM mathematical model is used because it's more reality to the actual plant rather than linear transfer function model in the control design and studies and give more accurate results. The SEDM model is simulated using MATLAB R2013a simulink toolbox. The SEDM is loading for different loads ranging from no-load to full-load to test the controller behavior and it's robustness for wide range of loadings variations. The results are compared with controller tuned by Ziegler-Nichols (ZN) method. The results show the superiority of BFO versus ZN method for SEDM speed control, which leads to improve the transient and steady state of speed responses of SEDM for different loads. The proposed method is very efficient and could easily be extended for other global optimization problems.

Keywords Bacterial Foraging Optimization (BFO), Escherichia (E.Coli) Bacteria, Separately Excited DC Motor (SEDM), Proportional-Integral-Derivative (PID), Ziegler-Nichols (ZN).

1. Introduction

Direct - current (DC) motors are one of the most widely used prime movers in the industry today. Years ago, the majority of the small servomotors used for control purposes were ac. In reality, ac motors are more difficult to control, especially for position control, and their characteristics are quite nonlinear, which makes the analytical task more difficult. DC motors, on the other hand, are more expensive, because of their brushes and commutators, and variable-flux dc motors are suitable only for certain types of control applications [1]. DC motors have been widely used in many industrial applications such as electric vehicles, steel rolling mills, electric cranes, and robotic manipulators due to precise, wide, simple, and continuous control characteristics [2]. DC machines are characterized by their versatility. By means of various combinations of shunt-, series-, and separately-excited field windings they can be designed to display a wide variety of volt-ampere or speed-torque characteristics for both dynamic and steady-state operation. Because of the ease with which they can be controlled systems of DC machines have been frequently used in many applications requiring a wide range of motor speeds and a precise output motor control [3-4].

The desired torque-speed characteristics could be achieved by the use of conventional proportional-integral-derivative (PID) controllers. As PID controllers require exact mathematical modeling, the performance of the system is questionable if there is parameter variation. However the PID (proportional _integral _derivative) controller is still extensively used in the industry this is due to its simplicity and the ability to apply in a wide range of situations On the other hand a PID controller is rather difficult and can be a time consuming process. The speed of DC motor can be adjusted to a great extent so as to provide easy control and high performance [2].

Several methods have been proposed for the tuning of PID controllers. Among the conventional PID tuning methods, the Ziegler–Nichols method may be the most well known technique. For a wide range of practical processes, this tuning approach works quite well. However, sometimes it does not provide good tuning and tends to produce a big overshoot. Therefore, this method usually needs retuning before applied to control industrial processes. To enhance the capabilities of traditional PID parameter tuning techniques, several intelligent approaches have been suggested to improve the PID tuning [5].

There are several conventional and numeric controller types intended for controlling the DC motor speed at its executing various tasks. There are several optimization algorithms which can be used for searching the optimal gain parameter a very basic one is the random search. In recent year, many intelligence algorithms are proposed to tuning the PID parameters by the optimal algorithms such as the simulated Annealing (SA), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) algorithm [2].

In recent years, chemotaxis (i.e. the bacterial foraging behavior) as a rich source of potential engineering applications and computational model has attracted more and more attention. A few models have been developed to mimic bacterial foraging behavior and have been applied for solving some practical problems. Among them, bacterial foraging optimization is a population-based numerical optimization algorithm presented by Passino. BFO is a simple but powerful optimization tool that mimics the foraging behavior of *E. coli* bacteria. Until now, BFO has been applied successfully to some engineering problems, such as optimal control, harmonic estimation, transmission loss reduction, and machine learning [6].

Mathematical Model of Separately Excited D.C. Motor

The system contains a separately excited D.C. motor (SEDM), a model based on the motor specifications needs to be obtained. As shown in Figure (1). In a separately excited dc motor, the field coil is supplied from a different voltage source than that of the armature coil**.** The field circuit normally incorporates a rheostat through which the field current, and thus the motor ○s characteristics, can be externally controlled. This motor is mainly suitable for two types of loads; those that require constant torque for speed variations up to full-load speed, and those whose power requirements are constant for speed variations above nominal speed. The field current is constant, and then the flux must be constant. The electrical armature and field circuit can model the motor. In this simple model R_a and L_a indicate the equivalent armature coil resistance and inductance respectively and R_f and L_f indicate the equivalent field resistance and inductance respectively, v_a is the voltage supplied by the power source. The basic motor equations are:

$$T_d = K_f\, i_f\, i_a = K_m\, i_a \tag{1}$$

$$e_g = K_f\, i_f\, \omega_m = K_m\, \omega_m \tag{2}$$

$$V_a = e_g + R_a\, i_a + L_a \frac{di_a}{dt} \tag{3}$$

$$\frac{d\omega_m}{dt} = \frac{1}{J}\left(K_m\, i_a - T_L - B\, \omega_m\right) \tag{4}$$

Where $K_m = K_f$ if, is a constant, e_g is the back electromotor force, T_d is the torque of the motor, T_L is the torque of the mechanical load; J is the inertia of the rotor and B is the damping coefficient associated with the mechanical rotational system of the motor [3].

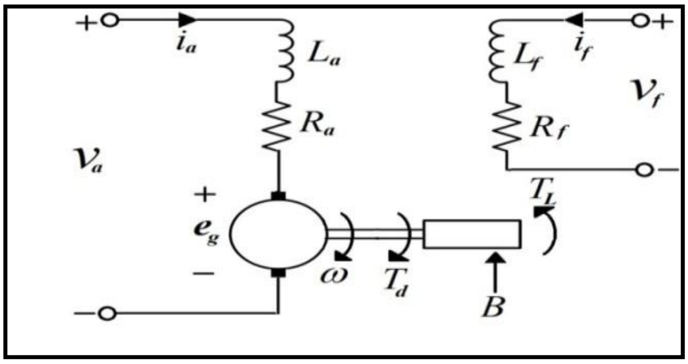


Figure 1: Equivalent circuit of separately excited DC motor

Proportional–Integral–Derivative Controller (PID)

PID is a generic control loop feedback mechanism (controller) widely used in industrial control systems – a PID is the most commonly used feedback controller. A PID controller calculates an "error" value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process control inputs. The PID controller calculation (algorithm) involves three separate constant parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted K_p, K_i, and K_d. Heuristically, these values can be interpreted in terms of time: K_p depends on the present error, K_i on the accumulation of past errors, and K_d is a prediction of future errors, based on current rate of change [7].

Tuning of PID Controller using Z-N Method

The first method of Z-N tuning is based on the open-loop step response of the system. The open loop system's S shaped response is characterized by the parameters, namely the process time constant T and L. These parameters are used to determine the controller's tuning parameters. The second method of Z-N tuning is closed-loop tuning method that requires the determination of the ultimate gain and ultimate period. The method can be interpreted as a technique of positioning one point on the Nyquist curve. This can be achieved by adjusting the controller gain (K_u) till the system undergoes sustained oscillations (at the ultimate gain or critical gain), whilst maintaining the integral time constant (T_i) at infinity and the derivative time constant (T_d) at zero. This paper uses the second method as shown in Table 1 [7].

Table 1: Ziegler-Nichols open-loop tuning rule [8]

Controller	K_P	T_i	T_d
P	T /L	∞	0
PI	0.9(T /L)	L / 0.3	0
PID	1.2(T/L)	2L	0.5L

The controller output is computed in continuous time as follows:

$$U(t) = K_p.\left(e(t) + \frac{1}{T_i}\int_0^t e(t).dt + T_d\frac{de(t)}{dt} \right) \quad (5)$$

Where K_p is the proportional gain, T_i and T_d is reset time and derivative time [9].

Bacterial Foraging Optimization

The Bacterial Foraging Optimization (Passino 2002) is based on foraging strategy of *E. coli* bacteria. The foraging theory is based on the assumption that animals obtain maximum energy nutrients 'E' in a suppose to be a small time 'T'. The basic Bacterial Foraging Optimization consists of three principal mechanisms; namely chemotaxis, reproduction and elimination-dispersal. The brief descriptions of these steps involved in Bacterial Foraging are presented below [10]. To define our optimization model of E. coli bacterial foraging, we need to define a population (set) of bacteria, and then model how they execute chemotaxis, swarming, reproduction, and elimination/dispersal. After doing this, we will highlight the limitations (inaccuracies) in our model [11].

Chemotaxis

In the classical BFO, a unit walk with random direction represents a "tumble" and a unit walk with the same direction in the last step indicates a "run". Suppose $\theta^i(j, k, \ell)$ represents the bacterium at j^{th} chemotactic, k^{th} reproductive, and ℓ^{th} elimination-dispersal step. C(i), namely, the run-length unit parameter, is the chemotactic step size during each run or tumble. Then, in each computational chemotactic step, the movement of the i^{th} bacterium can be represented as:

$$\theta^i(j+1,k,\ell) = \theta^i(j,k,\ell) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (6)$$

where Δ(i) is the direction vector of the j^{th} chemotactic step. When the bacterial movement is run, Δ(i) is the same with the last chemotactic step; otherwise, Δ(i) is a random vector whose elements lie in [−1, 1]. With the activity of run or tumble taken at each step of the chemotaxis process, a step fitness, denoted as $J(i,j,k,\ell)$, will be evaluated [6].

Swarming

During the movements, cells release attractants and repellents to signal other cells so that they should swarm together, provided that they get nutrient-rich environment or avoided the noxious environment. The cell-to cell attraction and repelling effects are denoted as:

$$J_{cc}(\theta,P(j,k,\ell)) = \sum_{i=1}^{S} J_{cc}^{i}(\theta,\theta^{i}(j,k,\ell)) = \sum_{i=1}^{S}\left[-d_{attract}\ exp\left(-w_{attract}\ \sum_{m=1}^{p}\left(\theta_m-\theta_m^i\right)^2\right)\right]$$
$$+$$
$$\sum_{i=1}^{S}\left[-h_{repellant}\ exp\left(-w_{repellant}\ \sum_{m=1}^{p}\left(\theta_m-\theta_m^i\right)^2\right)\right] \qquad (7)$$

where $J_{cc}(\theta,P(j,k,\ell))$ is the objective function value to be added to the actual objective function to present time varying objective function, S is the total number of bacteria, P is the number of variables involved in the search space, $\theta = [\theta_1, \theta_2, \ldots, \theta_P]^T$ is a point on the optimization domain, and θ_m^i is the m[th] components of the i[th] bacterium position θ^i.$d_{attract}$, $w_{attract}$, $h_{repellant}$, and $w_{repellant}$ are different coefficients used for signaling [12].

Reproduction &Elimination/Dispersal

After N_c chemotactic steps, a reproduction step is taken. Let N_{re} be the number of reproduction steps to be taken. For convenience, we assume that S is a positive even integer. Let

$$Sr = S / 2 \qquad (8)$$

be the number of population members who have had sufficient nutrients so that they will reproduce (split in two) with no mutations. For reproduction, the population is sorted in order of ascending accumulated cost (higher accumulated cost represents that it did not get as many nutrients during its lifetime of foraging and hence, is not as "healthy" and thus unlikely to reproduce); then the S_r least healthy bacteria die and the other S_r healthiest bacteria each split into two bacteria, which are placed at the same location. Other fractions or approaches could be used in place of Equation (8) this method rewards bacteria that have encountered a lot of nutrients, and allows us to keep a constant population size, which is convenient in coding the algorithm. Let N_{ed} be the number of elimination-dispersal events, and for each such event event, each bacterium in the population is subjected to elimination-dispersal with probability p_{ed}. We assume that the frequency of chemotactic steps is greater than the frequency of reproduction steps, which is in turn greater in frequency than elimination-dispersal events (e.g., a bacterium will take many chemotactic steps before reproduction, and several generations may take place before an elimination dispersal event) [11].

Simulation and Results

The simulation is doing using MATLAB tool box**.** This work is based on tuning PID controller for speed control of SEDM based BFO technique. SEDM is loading with different loads to see the performance of the designing controller, and then comparing the results with controller tuned by Z-N method to show the superiority of the PID controlle based on BFO.

Design Requirements

Since the most basic requirements of a motor are that it should rotate at the desired speed, the steady-state error e_{ss} of the motor speed should be less than 2%, the settling time T_s for 2% criterion should be less than 1sec, percent overshoot less than 50%.

Simulation of SEDM Using Matlab/Simulink

The proposed mathematical model is developed from the mechanical and electrical dynamic equations of the SEDM, equations (1), (2), (3) & (4). The simulink of the SEDM mathematical model is shown in Figure 2.

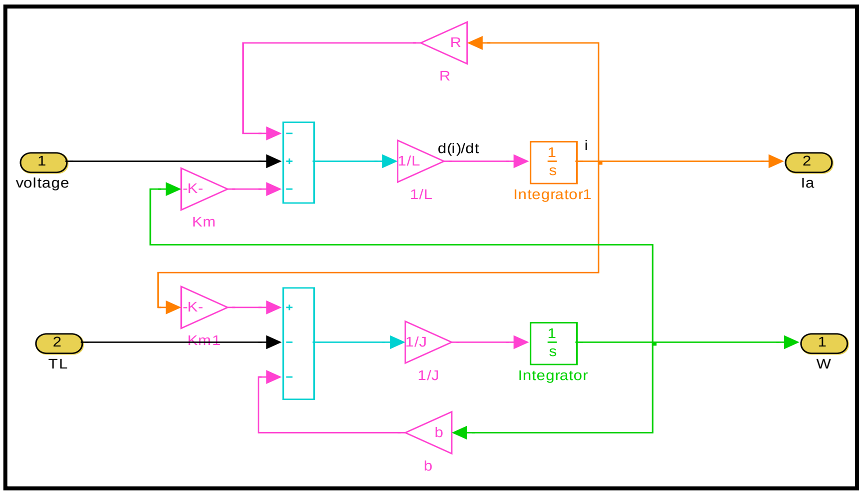


Figure 2: SEDM simulation using MATLAB/SIMULINK

SEDM Rating & Parameters

The parameters values of SEDM used in the simulation is taken from MATLAB/Toolbox and shown in Table 2.

Table 2: 10 hp, 500V, 1750 R.P.M. supply SEDM parameters

Motor ratings and parameters	Values
Armature resistance (R_a)	4.712Ω
Armature inductance (L_a)	0.05277 H
K_m	2.242
Inertia of the rotor (J)	0.04251 Kg.m^2
damping coefficient (B)	0.003406 N.m.s

SEDM Loads

The SEDM are loaded for four different loads (assumed). These loads are: (no-load, (0.3 of full-load) as a light load, (0.5 of full-load) as a half full load, and finally (full-load). Figure 3 shows the complete simulink model of closed loop control system for SEDM.

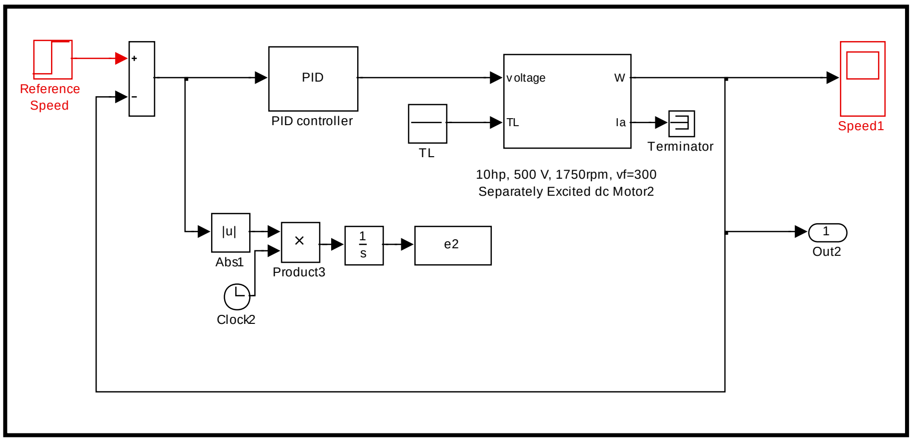


Figure 3: Closed loop speed control system of SEDM

PID Controller Tuned by BFO

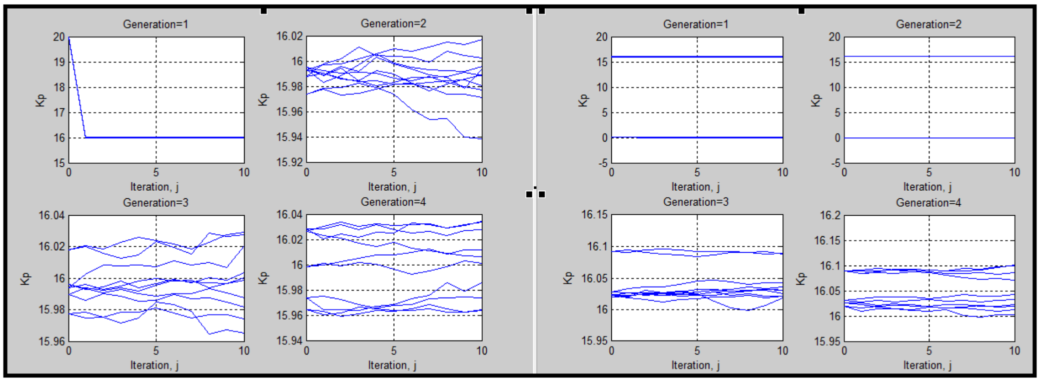
The parameters of BFO algorithm are listed in Table 3, while the obtained PID controller parameters are listed in Table 4.

Table 3: BFO parameters used in tuning PID controller

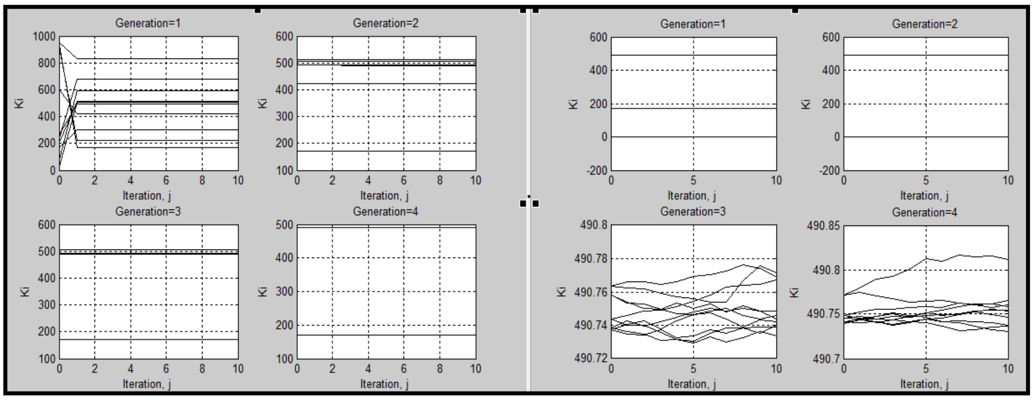
BFO Parameters	Values
Number of bacteria in the population (s)	10
The length of swim (N_s)	2
Number of reproduction steps (N_{re})	4
Number of chemotactic step (N_c)	10
Number of elimination/dispersal events (N_{ed})	2
Number of bacteria splits per generation (S_r)	s/2
Probability of dispersal occurrence (P_{ed})	0.3
Height of repellent effect ($h_{rep.}$)	0.1
Width of repellent effect ($w_{rep.}$)	10
Width of attractant effect ($w_{attr.}$)	0.2
Width of attractant effect ($d_{attr.}$)	0.1

Table 4: PID controller parameters

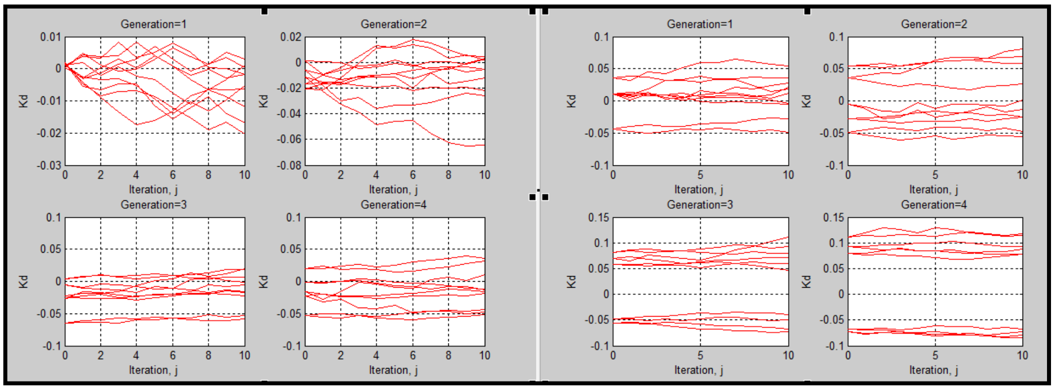
Controller parameters	Z-N	BFO technique
K_p	6.96	16.0093
K_i	348	490.8125
K_d	0.0348	0.1289



(a) first elimination/dispersal event (b)second elimination/dispersal event



(c)first elimination/dispersal event (d)second elimination/dispersal event

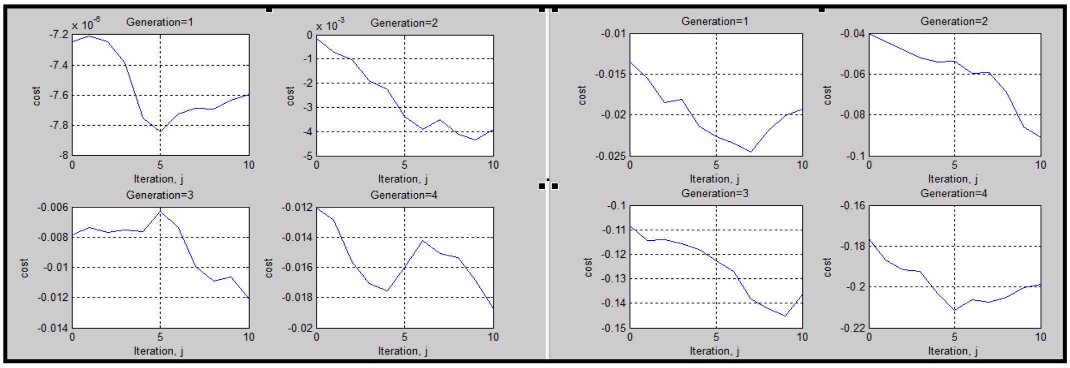


(e)first elimination/dispersal event (f)second elimination/dispersal event

Figures 4 (a, b, c, d, e, f): Bacteria trajectories

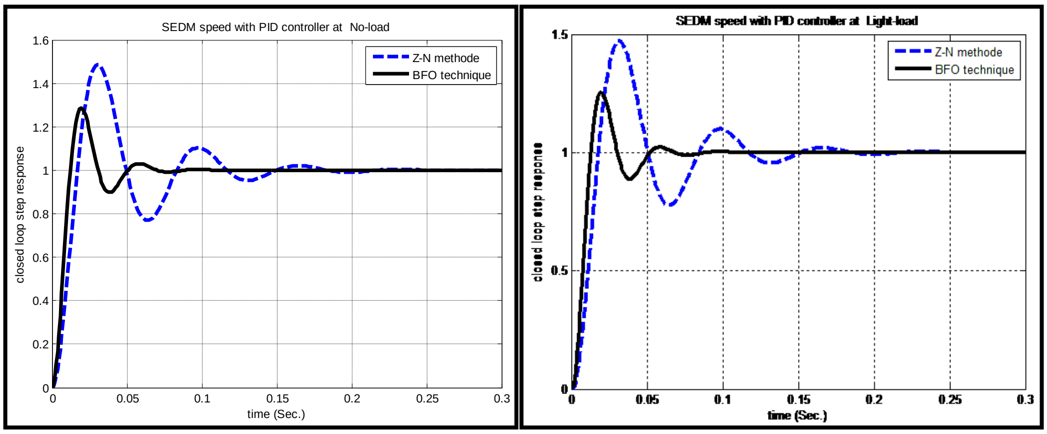
Figures 4 shows the bacteria (S=10) motility behavior or bacteria trajectories for tuning PID controller parameters. This motility behavior depends on bacteria average cost achieved during each iteration (chemotactic step N_c). The generation number represent reproduction step (N_{re}) while iteration j represent chemotactic steps (N_c). These bacteria motility behavior achieved for two elimination/dispersal events (Ned =2). For every generation at the end of all chemotactic steps, the PID parameters are obtained with best cost (or fitness) value which represents the best value of compensator parameters.

Figures 5 shows the average cost plots for each generation for two elimination/dispersal events (Ned =2).

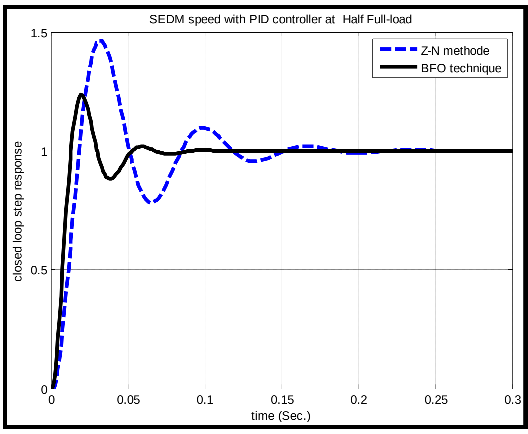


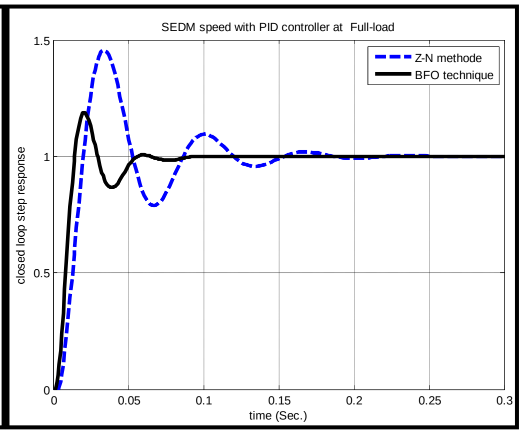
Figures 5: Average cost plot for bacteria trajectories

The speed step responses of SEDM with different loads for PID controller based on both designing methods are shown in Figures 6 (a, b, c, and d).



(a) SEDM at no-load (b) SEDM at light load

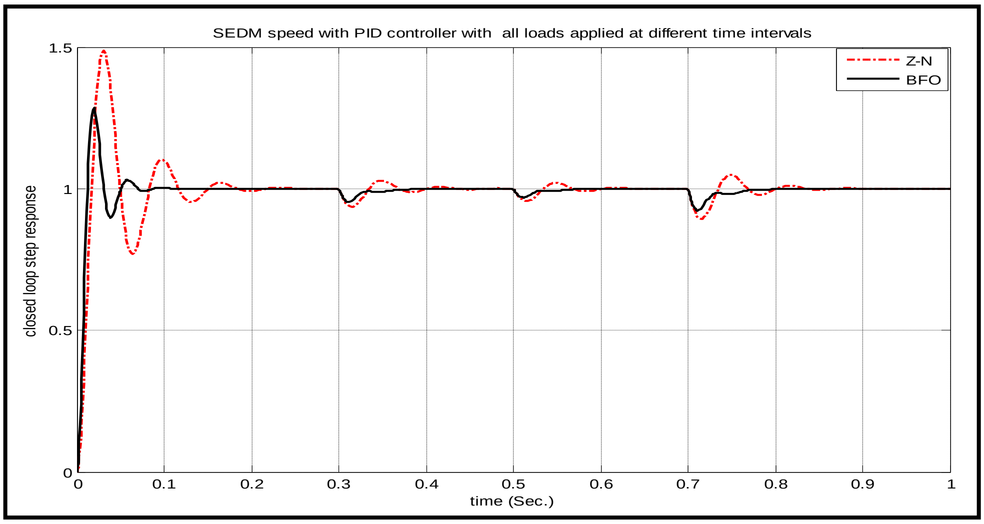




(c) SEDM at half full-load (d) SEDM at full-load

Figures 6 (a, b, c, d): SEDM speed responses with PID controller at different loads

Figures 7 shows the speed response of SEDM with load increased gradually at different time intervals, while).



Figures 7: Speed responses with PID controller at different loads

The transient response specifications of SEDM speed response are listed in Table 5 for PID controller with different loading conditions.

Table 5: Transient response specifications

	Rise time (sec)	Peak time (sec)	Percent Overshoot	Settling time (sec)
SEDM at no-load				
Z-N	0.0169	0.0309	48.6	0.168
BFO	0.0122	0.0196	28.65	0.0636
SEDM at light-load				
Z-N	0.0179	0.0306	46.97	0.1446
BFO	0.0128	0.0205	25.4	0.0614
SEDM at half full-load				
Z-N	0.0185	0.0313	46.26	0.1452
BFO	0.0132	0.0196	23.54	0.0504
SEDM at full-load				
Z-N	0.02	0.0334	45.44	0.1466
BFO	0.0143	0.0211	18.68	0.0525

From Table 5 it is clearly that, the transient specifications are improved of SEDM with PID controller tuned by BFO for different loads due to the search ability and fast convergence for BFO behavior.

Conclusion

In this work, BFO technique has been used to design PID controller for speed control of SEDM. BFO is used to find optimal controller parameters (K_p, K_i and K_d). The results are compared with PID controller tuned by Z-N method. The SEDM is simulated using mathematical model which is more reality and accurate for representation the actual plant. The SEDM is loading for different loads ranging from no-load to full-load for testing the controller robustness for load changing conditions. From simulation results the following tips can be concluded:

1. The BFO technique is robust and efficient for controllers tuning, and best than Z-N method for tuning PID controllers.
2. BFO required less execution time, due to the small numbers of bacterial foraging parameters and fast convergence ability.
3. BFO has fast convergence due to the bacteria social behavior for finding nutrient and it is efficient tool for optimization problems.
4. The proposed controllers are robust for wide range of loading conditions.
5. The proposed controller improved the time response specifications for speed control purpose of SEDM for different loads.
6. The proposed approach has potential to be useful for other practical optimization problems (e.g., engineering design, online distributed optimization in distributed computing, and cooperative control) as social foraging models work very well in such environments.

References

1. F. Golnaraghi, B. C. Kuo, " Automatic Control Systems", John Wiley & Sons, Inc.,9th Edition, 2010.
2. M. George, " Speed Control of Separately Excited DC Motor", American Journal of Applied Sciences 5 (3), 227-233, 2008.
3. A. J. Mohammed, " Speed Control for Separately Excited DC Motor Drive (SEDM) Based on Adaptive Neuro-Fuzzy Logic Controller", Eng. & Tech. Journal,Vol.31, No.2, 277-295,2013.
4. B. Allaoua, B. Gasbaoui And B. Mebarki, " Setting Up PID DC Motor Speed Control Alteration Parameters Using Particle Swarm Optimization Strategy", Leonardo Electronic Journal of Practices and Technologies, Issue 14, p. 19-32, January-June 2009.
5. B. K. Panigrahi, Y. Shi, and M. Lim, "Handbook of Swarm Intelligence", Springer-Verlag Berlin Heidelberg, 2011.
6. H. Chen, Y. Zhu, and K. Hu, "Adaptive Bacterial Foraging Optimization", Hindawi Publishing Corporation Abstract and Applied Analysis, Vol. 2011, Article, 27 pages, 2011.
7. A. M. Mahmood, "Design of ON-Line Tuned Idle Speed Controller for an Automotive Engine By Using NCD", IJCCCE, Vol.12, No.2, 2012.
8. K. Ogata, "Modern Control Engineering", 5th edition, 2010, Pearson Education, Inc., publishing as Prentice Hall, One Lake Street, Upper Saddle River, New Jersey.
9. T. Jain, V. Patel, and M. J. Nigam, " Implementation of PID Controlled SIMO Process on FPGA Using Bacterial Foraging for Optimal Performance", International Journal of Computer and Electrical Engineering, Vol. 1, No. 2, June 2009.
10. B. K. Panigrahi,Y. Shi, and M. Lim, "Handbook of Swarm Intelligence", Springer-Verlag Berlin Heidelberg, 2011.
11. V. Gazi and K. M. Passino, "Swarm Stability and Optimization", Springer Science + Business Media B.V. 2011.
12. R. Panda and M. K. Naik, "A Crossover Bacterial Foraging Optimization Algorithm", Hindawi Publishing Corporation Applied Computational Intelligence and Soft Computing Vol. 2012, 7 pages, 2012.