

## Population Algorithms for optimal control of BLDC motor drive

<sup>1</sup>Manoj Kumar Merugumalla, <sup>2</sup>Prema Kumar Navuri

<sup>1,2</sup>Department of Electrical Engineering, Andhra University, Visakhapatnam, India

Email: [mkla2009@gmail.com](mailto:mkla2009@gmail.com), [Prema.ee@auvsp.edu.in](mailto:Prema.ee@auvsp.edu.in)

Received: 10<sup>th</sup> Feb 2018, Accepted: 18<sup>th</sup> March 2018, Published: 30<sup>th</sup> April 2018

### Abstract

This paper presents nature-inspired optimization algorithms such as particle swarm optimization (PSO) algorithm and bat algorithm (BA) for tuning PID controller parameters of BLDC motor drive. Both PSO algorithm BA are population based algorithms. Population based algorithms have number of advantages over classical methods for solving complex optimization problems. The position of BLDC rotor is determined by measuring the changes in the Back emf. Sensorless control method reduces the cost of motor as it does not need sensors for the detection of rotor position. The BLDC motor drive is modelled in Matlab/simulink. The simulation results reveals that proposed methods are effective in reducing the time domain parameters steady state error, rise time, settling time and peak overshoot.

### Keywords

Brushless direct current motor, particle swarm optimization, bat algorithm, position, velocity, PID controller

### Introduction

BLDC motor has wide range of applications because of its advantages such as high efficiency, flat speed-torque characteristics, and high speed range, smaller in size and lighter, longer life, low noise and good dynamic response when compared against brushed direct current motor. BLDC motor is a kind of permanent magnet synchronous motor (PMSM), trapezoidal type of PMSM is known as brushless direct current motor. In a brushless motor, stator contains windings and rotor incorporates permanent magnet. By moving permanent magnets to rotor and driving field coils with power electronic switch can eliminate brushes in dc motor. BLDC motors are often called as electronically commutated motors [1]-[3]. It requires some electronic control mechanism to determine the rotor position continuously. The rotor position can be determined either by measuring changes in back emf at each of the armature coils, which is known as sensorless control [4]-[7] or by using a Hall Effect sensors embedded into the stator on the non-driving end of the motor.

### Objective and Methodology

The objective in the optimal design of PID controller is to minimize the overshoots and settling time in system oscillations with minimum error. In this paper, particle swarm optimization algorithm and bat algorithm are

proposed for optimization of PID-parameters. Just like any optimization problem an objective function needs to be formulated for optimal PID design. The difference between reference speed and actual speed of BLDC motor drive is the objective function and it needs to be minimized. BLDC motor drive model is simulated in matlab/simulink and simulation results of PSO and BA are compared.

### BLDC motor drive Model

#### A. BLDC motor model

The schematic diagram of BLDC motor drive control is shown in figure 1. The motor drive system mainly consists of BLDC motor, three-phase voltage source inverter, hysteresis current controller, reference current generator and speed controller. Motor is supplied by the three phase voltage source inverter, the switching functions obtained from the hysteresis current controller serves as input to voltage source inverter. The reference currents are generated by reference current generator based on rotor position. The actual speed of the motor is compared with the reference speed and error is applied to controller and controller output serves as the input to the reference current generator. The output voltages obtained from the three phase voltage source inverter can be expressed in the form of switching functions.

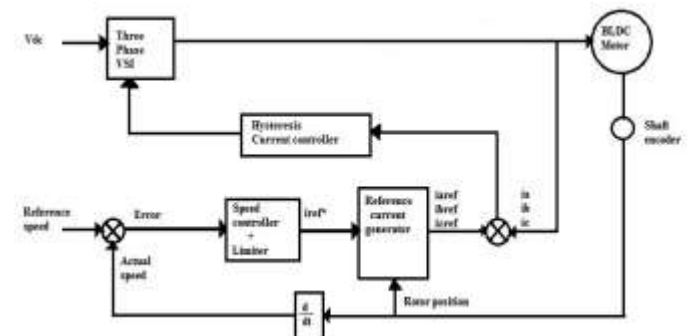


Fig. 1 BLDC motor drive control scheme

The stator winding voltages in terms of the winding parameters are expressed as:

$$V_{an} = R_a i_a + \frac{d}{dt} (L_{aa} i_a + L_{ba} i_b + L_{ca} i_c + e_a) \quad (1)$$

$$V_{bn} = R_b i_b + \frac{d}{dt} (L_{ab} i_a + L_{bb} i_b + L_{cb} i_c + e_b) \quad (2)$$

$$V_{cn} = R_c i_c + \frac{d}{dt}(L_{ac} i_a + L_{bc} i_b + L_{cc} i_c + e_c) \quad (3)$$

$$R_a = R_b = R_c = R \quad (4)$$

$$L_{aa} = L_{bb} = L_{cc} = L_s \quad (5)$$

$$L_{ab} = L_{ba} = L_{ca} = L_{ac} = L_{bc} = L_{cb} = M \quad (6)$$

and since  $i_a+i_b+i_c=0$  and  $L_s-M=L$

In matrix form

$$\begin{bmatrix} V_{an} \\ V_{bn} \\ V_{cn} \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & L \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \quad (7)$$

where

$R$  : Stator resistance per phase, assumed equal for all phases

$L_s$  : Stator inductance per phase, assumed equal for all phases.

$M$  : Mutual inductance between the phases.

$i_a, i_b, i_c$  : Stator current/phase.

The electromotive forces of three- phase windings are:

$$\begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} = E \begin{bmatrix} f_a(\theta_e) \\ f_b(\theta_e) \\ f_c(\theta_e) \end{bmatrix}, \quad \text{where, } (E=K_e \cdot \omega_m) \quad (8)$$

The Equation of motion is expressed as:

$$T_j + T_D + T_S + T_l = T_e \quad (9)$$

Torque due to Inertia,  $T_j = J \frac{d\omega_r}{dt}$

where,  $J$  – Moment of inertia,

Torque due to viscous friction,  $T_D = B \cdot \omega_r$

where,  $B$  – Friction coefficient,

The Electromagnetic torque of 3-phase motor is

$$T_e = K_e (f_a(\theta_e) \cdot i_a + f_b(\theta_e) \cdot i_b + f_c(\theta_e) \cdot i_c) \quad (10)$$

### B. PID Controller

PID controller is the most widely used controller, it was earlier used in pneumatic devices later on occupied major portion of the industrial applications. It has a simple structure easily understandable by the plant operator relatively easy to tune its controller parameters. The performance of PID controller can be adjusted by tuning the proportional-Integral-Derivative parameters using any method [10].  $K_p$  is proportional gain,  $K_i$  is integral gain and

$K_d$  is derivative gain which are chosen to meet the criteria specified in terms of rise time, settling time, peak time, overshoot and steady state error. Settling time and rise times are to measure the response of the system and peak value, peak time and overshoot measures the quality of the response.

### PARTICLE SWARM OPTIMIZATION ALGORITHM

Particle Swarm Optimization algorithm is based on behavior of organized groups. It is based on the principle on which flock of birds behave while moving as an unit to one or the other direction [9]. In PSO, particles are distributed in parametric space and play the role of agents, on its movement they change their direction and speed of motion based on certain rules. A swarm in PSO consists of a number of particles. Each particle represents a potential solution to the optimization task. Each particle represents a candidate solution. Each particle moves to a new position according to the new velocity which includes its previous velocity and the moving vectors according to the past best solution and global best solution [8]. The best solution is then kept; each particle accelerates in the directions of not only the local best solution but also the global best position. If a particle discovers a new probable solution, other particles will move closer to it in order to explore the region. The optimization technique is applying on to the objective functions, which one is most efficient to calculate the individual particle's fitness value.

$$\text{Objective function} = \omega(t)_{ref} - \omega(t)_{act} \quad (11)$$

where  $\omega(t)_{ref}$  is reference speed and  $\omega(t)_{act}$  is actual speed. Therefore, the design problem can be formulated as the optimization problem and the objective function is expressed as:

$$OF = \text{Minimize } (f) \text{ where, } f \in \text{error} \quad (12)$$

Subjected to constraints

$$K_{Pmin} \leq K_P \leq K_{Pmax}, K_{Imin} \leq K_I \leq K_{Imax}$$

$$K_{Dmin} \leq K_D \leq K_{Dmax}$$

PSO algorithm implementation steps are as follows:

Step 1: Read the data and generate the initial solution randomly.

$$X_{i,j} = (x_{1,1}, x_{1,2}, x_{1,3}, \dots \dots \dots x_{pop,n}), \quad (13)$$

$i = 1 \text{ to } pop \text{ and } j = 1 \text{ to } n$

$$V_{i,j} = (v_{1,1}, v_{1,2}, v_{1,3}, \dots \dots \dots v_{pop,n}), \quad (14)$$

$i = 1 \text{ to } pop \text{ and } j = 1 \text{ to } n$

Where, pop is population size and n is dimension of the problem

Step 2: Calculation of fitness value of the objective function using Equation 12.

Step 3: Calculate *pbest* i.e. objective function value of each particle in the population of the current iteration is compared with its previous iteration and the position of the particle having a lower objective function value as *pbest* for the current iteration is recorded:

$$pbest_m^{k+1} = \begin{cases} pbest_m^k & \text{if } f_m^{k+1} \geq f_m^k \\ x_m^{k+1} & \text{if } f_m^{k+1} \leq f_m^k \end{cases} \quad (15)$$

Where, k is the number of iterations, and f is objective function evaluated for the particle.

Step 4: Calculation of *gbest* i.e. the best objective function associated with the *pbest* among all particles in the current iteration is compared with that in the previous iteration and the lower value is selected as the current overall *gbest*.

$$gbest_m^{k+1} = \begin{cases} gbest_m^k & \text{if } f_m^{k+1} \geq f_m^k \\ pbest_m^{k+1} & \text{if } f_m^{k+1} \leq f_m^k \end{cases} \quad (16)$$

Step 5: Velocity updating, after calculation of the *pbest* and *gbest* the velocity of particles for the next iteration should be modified by using equation:

$$V_m^{k+1} = \omega V_m^k + C_1 \text{rand}(pbest_m^k - X_m^k) + C_2 \text{rand}(gbest^k - X_m^k) \quad (17)$$

Where, the parameters of the above equation should be determined in advance and  $\omega$  is the inertia weight factor, defined as follows:

$$\omega = \omega_{\max} - \frac{(\omega_{\max} - \omega_{\min})}{\text{iter}_{\max}} * \text{iter} \quad (18)$$

C1, C2 are the acceleration coefficients usually in range [1, 2]. A large inertia weight ( $\omega$ ) facilitates a global search while a small inertia weight facilitates a local search.

Step 6: Check the velocity components constraints occurring in the limits from the following conditions,

$$\text{If } Vid > Vmax, \text{ then } Vid = Vmax \text{ (or) If } Vid < -Vmax \text{ then } Vid = -Vmax \quad (19)$$

Step 7: Position updating, the position of each particle at the next iteration (k+1) is modified as follows:

$$X_j^{k+1} = X_j^k + V_j^{k+1} \quad (20)$$

Step 8: If the number of iterations reaches the maximum then go to step 9 for *gbest* otherwise, go to step 2.

Step 9: The individual that generates the latest *gbest* is the optimal PID parameters at minimum objective function.

The flow chart for Implementation of PSO algorithm is shown in figure 2.

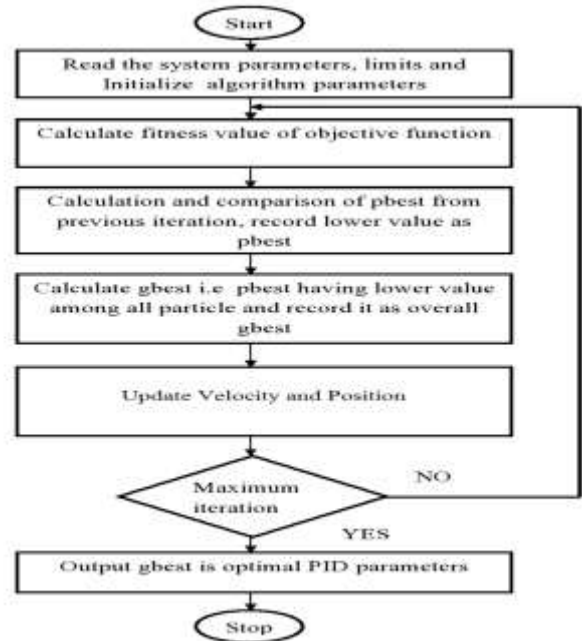


Fig.2 Implementation flowchart of PSO

**Bat algorithm**

Bat algorithm is another population based algorithm proposed by Xin-She-Xang, for solving global optimization problems [11]. This algorithm is inspired by the social behavior of bats. Bats are the second largest order of mammals, they migrate to hundreds of kilometres to find their prey from different types of insects even in complete darkness based on property known as echolocation [12]. Bat uses sonar type echolocation to distinguish the difference prey and avoid obstruction. Optimal parameter setting is important for measuring algorithm performance. Bat algorithm is bad at exploitation and exploration. Bat emits a loud sound pulse and waits for echo that bounces back from surrounding object. Each sound pulse includes loudness and frequency. The loudness decreases as bat moves towards prey, the sound pulse emitted by bat lasts only for a very small duration. The frequency range is 25 KHz to 100 KHz for most of the bat species and for some it is up to 150 KHz, normally bats emits 10-20 sound burst per second but during hunt for prey they emits 200 sound bursts per second.

The wave length of the sound burst with constant frequency f is given by

$$\lambda = v/f \quad (21)$$

Where ‘v’ is the velocity of light and wavelength ranges between 2mm-14mm for the frequency range of 25 KHz-150 KHz.

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \tag{27}$$

Xin-She-Xiang set out rules with some assumptions for BA implementation:

- Bats are sensible to distance as well as difference between prey and barriers using echolocation.
- Bats fly randomly with the velocity  $v_i$  at a position  $x_i$  with a fixed frequency  $f_i$  and varying wavelength  $\lambda$  and loudness  $A_0$  to search for prey. The wavelength of emitted pulse is automatically adjusted by bats and depending on the proximity of target, they also adjust the rate of pulse emission  $r \in [0,1]$  where 0 refers to no pulses and 1 refers to maximum rate of pulse emission. The loudness  $A_0$  varies from a large positive value to a minimum constant value.

**A. Position and velocity**

The bat position and velocity in a d-dimensional search space at a time ‘t’ can be updated using the following equations :

The new solutions and velocities are given by

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{22}$$

$$v_i^t = v_i^{t-1} + (x_i^t - x^*)f_i \tag{23}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{24}$$

where  $\beta \in [0,1]$  is random vector drawn from uniform distribution and is the current global best solution after comparing all solutions among all bats, from each bat is assigned a random value. Once a solution is selected among current best solutions, a new solution is generated locally using

$$X_{new} = X_{old} + \epsilon A^t \tag{25}$$

where  $\epsilon \in [-1,1]$  is a random number,

while  $A_{new} = A_{old} + \epsilon A^t$  is the average loudness of all bats

at time ‘t’.

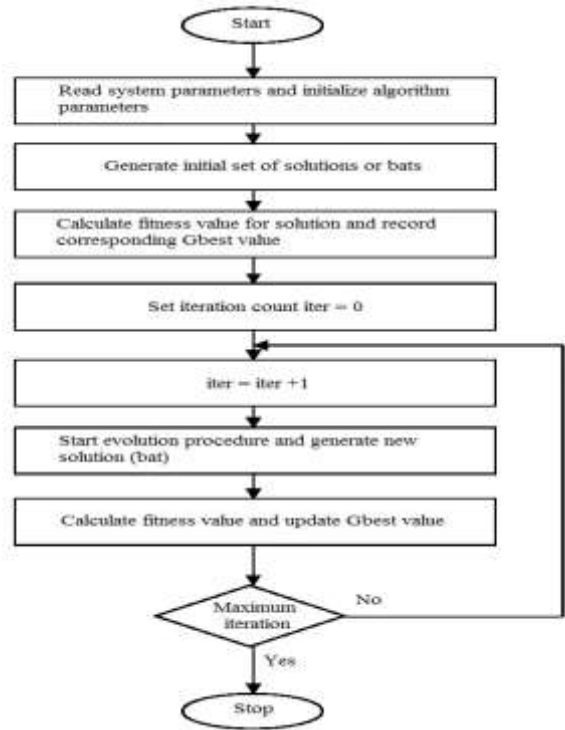
**B. Loudness and rate of pulse emission**

The loudness  $A_i$  and rate of pulse emission  $r$  also to be updated accordingly with position and velocity as iteration continues. The rate of pulse emission increases and loudness decreases once bat finds its prey. Loudness can be given any value as per convenience, here  $A_0 = 0.25$  and  $A_{min} = 0$ , which means bat found its prey and temporarily stopped emission. If the new solutions are developed, then loudness and rate of pulse emission will be updated as:

$$A_i^{t+1} = \alpha A_i^t \tag{26}$$

where  $\alpha$  and  $\gamma$  are constants, for any  $0 < \alpha < 1$  and  $\gamma > 0$ ,

Soln = [X], Soln represents a group of bats in bat algorithm, where bat is one position in search space. The problem specific implementation flow chart of bat algorithm has been given in figure 3.



**Fig. 3** Implementation Flowchart of BA

**Results and Discussion**

The model of BLDC motor drive is simulated in matlab/simulink using the specifications presented at the end of this section. The simulation results with PSO are shown in figures 5-9. The speed response is shown in figure 4, no oscillations can be seen, rise time is more but settles to the final value in less time with zero steady-state error. The electromagnetic torque response is shown in figure 5 in which steady value of torque can be observed. The three-phase current and trapezoidal back-emf are shown in figure 6 and figure 7 respectively. The reference speed tracking performance with PSO is shown in figure 8.

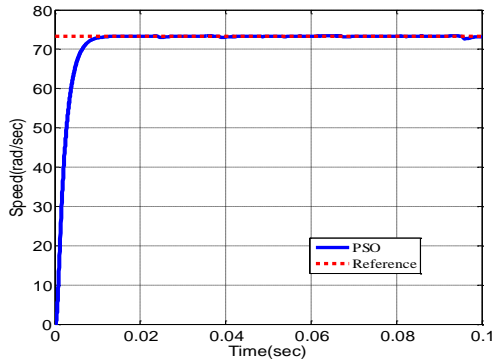


Fig. 4 Speed response of PSO

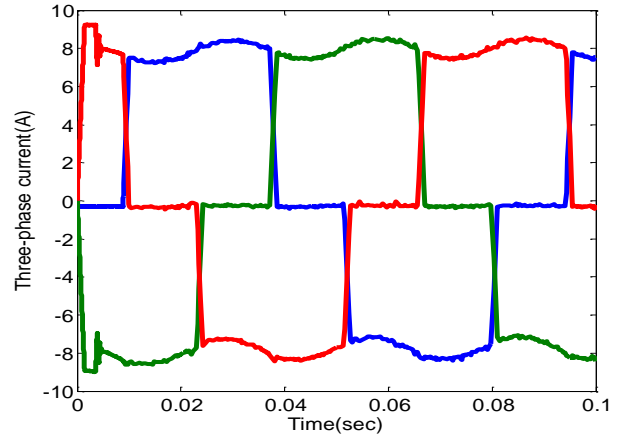


Fig. 12 Trapezoidal back-emf of BA

Fig. 9 Speed response of BA

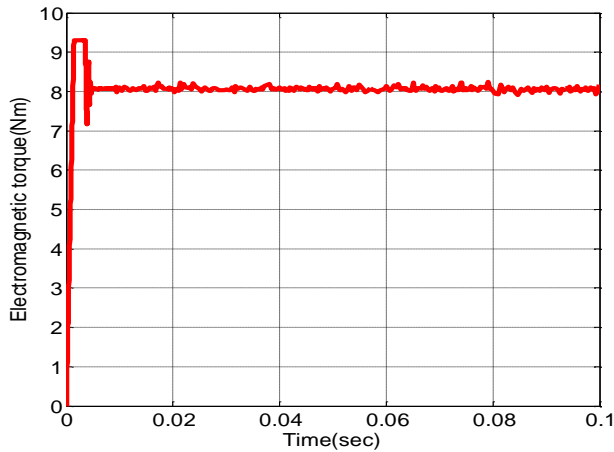


Fig. 10 Electromagnetic torque response of BA

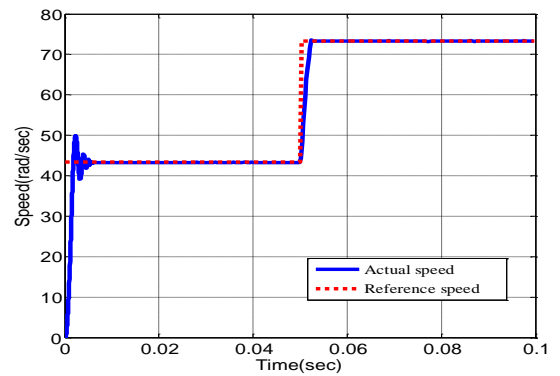


Fig. 13 Speed reference tracking of BA

The comparison of speed responses of PSO and BA is shown in figure 14 and time domain performance parameters such as rise time( $T_r$ ), settling time ( $T_s$ ), percentage peak-overshoot ( $\%M_p$ ) and steady-state error ( $E_{ss}$ ) along with the tuned PID controller parameters  $K_p$ ,  $K_i$ ,  $K_d$  are presented in table 1.

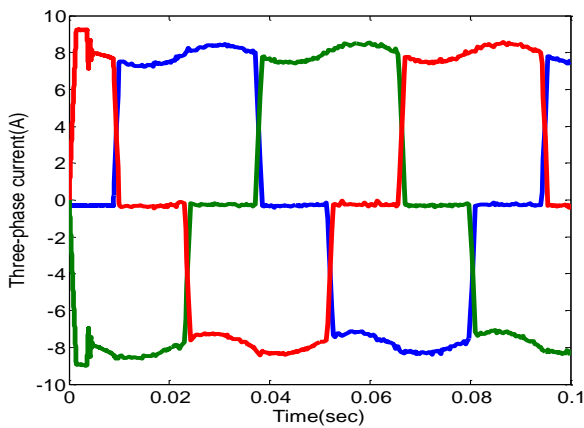


Fig. 11 Three-phase currents of BA

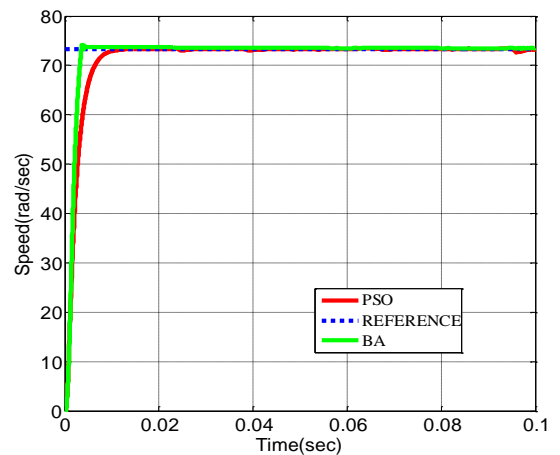


Fig. 14 Comparison of Speed responses with PSO and BA

TABLE 1 PERFORMANCE PARAMETERS OF PSO-PID AND BA-PID

Controller Parameter	PSO-PID	BA-PID
$K_p$	1.5869	1.0041
$K_i$	5.6103	10.00
$K_d$	0.0030	0
Rise time ( $T_r$ )	0.01	0.003
Settling time( $T_s$ )	0.02	0.08
Peak overshoot( $\%M_p$ )	0	0.5
Steady state Error( $E_{ss}$ )	0	0.03

### Conclusion

In this paper, the nature-inspired algorithms are proposed to search the PID controller parameters  $K_p$ ,  $K_i$ ,  $K_d$  for the speed control of BLDC motor drive. The motor drive system has been modeled and simulated in MATLAB/SIMULINK. The tracking performance is tested with change in speed reference speed and several time domain parameter performance measures such as rise time, peak time, settling time, peak overshoot and steady-state error of PSO algorithm and BA are compared. The simulation results of BA shows the less rise time compared to PSO algorithm but PSO results demonstrate the improved performance compared to BA in terms of settling time, peak-overshoot, steady-state error. Therefore, population algorithms are ideal to for optimal control of BLDC motor drive.

### References

- [1] Manoj Kumar Merugumalla and Dr. N. Prema Kumar "FFA Based speed control of BLDC motor drive", International journal of intelligent engineering informatics (inpress).
- [2] Manoj Kumar Merugumalla and Dr. N. Prema Kumar "Sensorless control of BLDC motor using bio-inspired optimization algorithm and classical methods of tuning PID controller", Journal on Instrumentation and Control Engineering, Vol.5.No.1, pp.16-23, 2017
- [3] Manoj Kumar Merugumalla and Dr. N. Prema Kumar "Optimized PID controller for BLDC motor using Nature-inspired Algorithms" International journal of Applied Engineering Research Vol.12, No.1, 2017.415-422, 2017.
- [4] Marcin Baszynski and Stanislaw Pirog, Member, IEEE, A Novel Speed Measurement Method for, a High-Speed BLDC Motor Based on the Signals From the Rotor Position Sensor, in :IEEE Transactions On Industrial Informatics, Vol. 10, No. 1, 2014
- [5] Alin, Stirban, Ion Boldea, Fellow, IEEE, and Gheorghe-Daniel Andreescu, Senior Member, IEEE, Motion-Sensorless Control of BLDC-PM Motor With Offline FEM-Information-Assisted Position and Speed Observer, in: IEEE Transactions On Industry Applications, Vol. 48, No. 6, 2012.
- [6] Chung-Wen Hung, Cheng-Tsung Lin, Chih-Wen Liu, Senior Member, IEEE, and Jia-Yush Yen, Member, IEEE, A Variable-Sampling Controller for Brushless DC Motor Drives With Low-Resolution Position Sensors, in : IEEE Transactions On Industrial Electronics, Vol. 54, No. 5, 2007.
- [7] Tae-Hyung Kim, Member, IEEE, and Mehrdad Ehsani, Fellow, IEEE, Sensorless Control of the BLDC Motors From Near-Zero to High Speeds, in : IEEE Transactions On Power Electronics, Vol. 19, No. 6, 2004.
- [8] Sri Latha Eti and N. Prema Kumar 'closed loop control of BLDC motor drive using adaptive fuzzy tuned PI controller', *journal of Engineering research and applications*, 2014.
- [9] Ker-Wei, Yu, Shang-Chang Hu, An application of AC servo motor by using particle swarm optimization base, sliding mode controller, in: Proceedings of IEEE Conference on Systems, Man and Cybernetics(5), 8-11 October 2006,
- [10] R. Jan, C. Tseng, and R. Liu, "Robust PID control design for permanent magnet synchronous motor : A genetic approach," vol. 78, pp. 1161–1168, 2008.
- [11] Yang, X-S. and He, X. 'Bat algorithm: literature review and applications', *Int. J. Bio-Inspired Computation*, Vol. 5, No. 3, pp.141–149, 2013.
- [12]
- [13] Xue, F., Cai, Y., Cao, Y., Cui, Z. and Li, F. 'Optimal parameter settings for bat algorithm', *Int. J. Bio-Inspired Computation*, Vol. 7, No. 2, pp.125–128, 2015.