# A Novel Hybrid ABF-PSO algorithm Based Tuning of Optimal FOPI Speed Controller for PMSM Drive

Anguluri Rajasekhar, Ravi Kumar Jatoth
National Institute of Technology-Warangal,
Andhra Pradesh-506021, INDIA.
rajasekhar.anguluri@ieee.org, ravikumar.nitw.ac.in

Ajith Abraham, Vaclav Snasel
VSB - Technical University of Ostrava,
Ostrava – Poruba, Czech Republic
ajith.abraham@ieee.org, vaclav.snasel@vsb.cz

***Abstract*—Bacterial Foraging Optimization algorithm (BFOA) has recently emerged as a very powerful technique for real parameter optimization. One of the major driving forces of BFOA is the chemotactic movement of a virtual bacterium that models a trial solution of optimization process. In the classical BFOA proposed by Passino, during the process of chemotaxis, optimization depends on a random search direction which may lead to delay in reaching global solution. To accelerate the convergence speed of group of bacteria near global optima the chemotactic step has been made adaptive and the resultant is Adaptive Bacterial Foraging Optimization (ABFO). In order to overcome the delay in optimization and to further enhance the performance of ABFO, this paper proposed a new hybrid algorithm combining the features of Adaptive Bacterial Foraging (ABF) and Particle Swarm Optimization (PSO) for tuning a Fractional order Proportional Integral speed controller in a vector controlled Permanent Magnet Synchronous Motor Drive. Our tuning method focuses on minimizing the Integral Time Absolute Error (ITAE) criterion. Computer simulations illustrate the effectiveness of the proposed approach compared to that of classical methods and state of art optimization techniques like PSO and ABFO.**

***Keywords-Permanent Magnet Synchronous Motor; Fractional order systems; global optimization; chemotaxis; adaptiveness; ITAE;***

## I. INTRODUCTION

According to recent studies, with the advancement of control theories, power electronics, microelectronics in connection with new motor design and magnetic materials since 1980's electrical (A.C) drives are making tremendous impact in the area of variable speed control systems [1-2]. Among A.C drives newly developed Permanent Magnet Synchronous Motors with high energy permanent magnet materials like "Neodymium Iron Boron" ("Nd-Fe-B") provide fast dynamics and computability with the applications if controlled properly. Electrically excited field windings are replaced by Permanent Magnets (PM) because of their advantages which include elimination of brushes, slip-rings, rotor copper losses; which yields high efficiency [3].

PMSMs are commonly used for applications like actuators, machine tools and robotics. This is due to some of its advantages such as high power-density, efficiency, reduced volume and weight, low noise and robustness [3]. Now-a-days vector control technique has made it possible to apply the PMSMs in high-performance industrial applications where only D.C motor drives were previously available. In order to make the most of a motor performance a very effective control system is needed. Although many possible solutions are available eg., non-linear, adaptive, intelligent control [4-5] the market of electrical drives doesn't justify the expense needed to implement such sophisticated solution in industrial drives and Proportional-Integral (PI) based control system scheme still remains the more widely adopted solution. Such a propensity is supported by a fact that, although simple, a PI based control allows achieving of very high performances when optimally designed [6]. PI controllers have been widely used for decades in industries for process control applications. The reason for their wide popularity lies in the simplicity of design and performances including low percent overshoot, less maintenance cost [6].

An elegant way of enhancing the performance of PI controllers is to use *fractional-order controllers.* Dynamic systems based on fractional order calculus have been a subject of extensive research in recent years since the proposition of concept of the *fractional-order controllers* and the demonstration of their effectiveness in actuating desired fractional order system responses by Podlubny [7]. Fractional Order Proportional Integral (FOPI) controller is one of a convenient fractional order structure that has been employed for control purposes. In an FOPI controller (in general $PI^{\alpha}$ ) I-operations are usually fractional order; therefore besides setting $K_P, K_I$ we have another parameter i.e., order of fractional integration $\alpha$. If $\alpha = 1$ it is an integer PI controller and if $\alpha = 0$ it is proportional gain. Finding appropriate set of values for these three parameters to achieve optimum performance of PMSM drive in three dimensional hyper-space calls for real parameter optimization. Our tuning method focuses on minimizing the Integral Time Absolute Error (ITAE) criterion. This is done by using Nature inspired heuristic methods for optimal design of the controller.

Natural selection tends to eliminate animals with poor foraging strategies and favor the propagation of genes of those animals that have successful foraging strategies since they are more likely to enjoy reproductive success. After many generations poor foraging strategies are either eliminated or shaped into good ones. Based on the researches on the foraging behavior of *E-coli* bacteria K.M. Passino proposed a new Evolutionary computation technique known as Bacterial Foraging Optimization

Algorithm (BFOA) [8], briefly explained in the following sections. Until date BFOA has found its successful implementation real world problems such as PI/PID controller design, stock market prediction, and power systems. However, during the process of chemotaxis, the BFOA depends on random search directions which may lead to delay in reaching global solution.

In order to speed the convergence of Bacterial Foraging Optimization W. Karoni had proposed an improved BFOA namely BF-PSO [9]. The BF-PSO algorithm borrowed the ideas of velocity updating from particle swarm optimization (PSO), the search directions specified by the tumble of bacteria are oriented by the individual best location and global best locations concurrently. The proposed method by W.Korani had shown remarkable results in PID controller tuning. To reduce the time of optimization and to accelerate the convergence speed of group of bacteria near global optima for this BFO-PSO we propose a new hybrid algorithm "ABF-PSO" in which the chemotactic step had been made adaptive. The proposed method is applied in parameter optimization of FOPI speed controller for a PMSM drive. We also compared the obtained results with that of traditional methods and state of art optimization techniques.

The rest of paper is organized as follows. Section II deals with mathematical model of PMSM, followed by problem formulation in Section III. Section IV describes the brief overview of the proposed method. To demonstrate the speed response of the controller simulation study is carried in Section V. Finally we end up with some conclusions in Section VI

## II. PERMANENT MAGNET SYNCHRONOUS MOTOR DRIVE

PMSM drives with approximately sinusoidal back electromotive forces (EMF) can be broadly categorized in to two types 1) Interior (buried) Permanent Magnet Motors 2) Surface Mounted Permanent Magnet Motors (SPM). In this paper we considered Surface mounted Permanent Magnet Motor and the cross sectional view is shown in Figure 1.

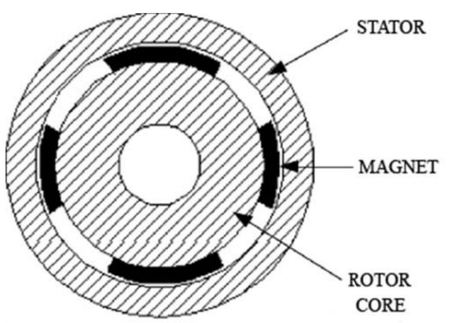


**Figure 1. Cross sectional view of PMSM**

In this surface mounted motor, magnets are mounted on the surface. Because the incremental permeability of magnets is 1.02-1.20 relative to external fields, the magnets have high reluctance and SPM can be considered to have large and effective uniform air gap. This property makes saliency negligible. Thus quadrature axis synchronous inductance of motor is equal to direct axis inductance i.e., $L_q = L_d$ . As a result magnetic torque only can be produced by the motor which arises from the interaction of Magnetic flux and quadrature axis current [10].

The following [11] assumptions are taken in to account for deriving the mathematical equations.

- Saturation is neglected
- Back Emf is sinusoidal
- Eddy currents, Hysteresis losses are negligible.

### A. Mathematical Model of PMSM

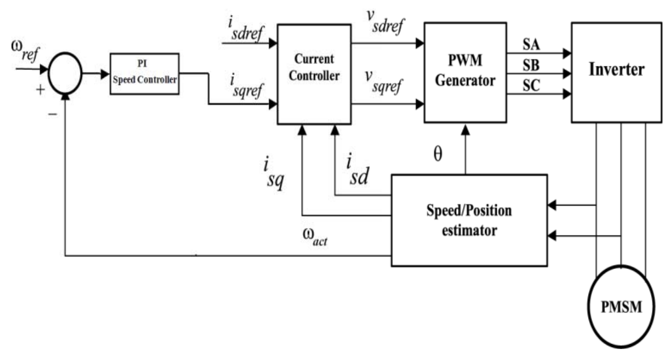


**Figure 2. Block Diagram of PMSM Drive**

Figure 2 shows the block diagram of Permanent Magnet Synchronous Motor Drive. The PMSM drive model consists of a Pulse width Modulation (PWM) inverter, a PWM generator, a current controller followed by speed controller and it is also embedded with speed/position estimator. The schematic representations of these components are shown in Fig. 1. The PMSM drive receives power from three-phase AC supply and runs mechanical load at desired speed. The developed model of the drive system is used for design in current and speed controllers.

The mathematical model of PMSM in *d-q* synchronously rotating frame of reference can be obtained from synchronous machine model. The PMSM can be represented by the set of following nonlinear [10, 11] (differential) equations

$$v_d = r_s i_d + p\lambda_d - \omega_e \lambda_q \quad (1)$$

$$v_q = r_s i_q + p\lambda_q + \omega_e \lambda_d \quad (2)$$

where $v_q$, $v_d$, $i_q$, $i_d$ are *d-q* axis voltages and currents respectively; while $\omega_e, r_s$ are electrical speed of motor and stator resistance respectively; $p$ is the differential operator. The linkage fluxes $\lambda_d$ and $\lambda_q$ can be expressed in the terms of stator currents, inductances and the constant flux linkage $\lambda_m$ due to rotor permanent magnet as;

$$\lambda_d = L_d i_d + \lambda_m \quad (3)$$

$$\lambda_q = L_q i_q \quad (4)$$

Here $L_d, L_q$ represents *d-q* axis inductances; by substituting Eqn (3), (4) into Eqn (1), (2) the stator voltage equations are modified as

$$v_d = r_s i_d + L_d p i_d - \omega_e L_q i_q \quad (5)$$

$$v_q = r_s i_q + L_q p i_q + \omega_e L_d i_d + \omega_e \lambda_m \quad (6)$$

The Electromagnetic torque is

$$T_e = \frac{3}{2}\frac{P}{2}[\lambda_m i_q + (L_d - L_q) i_d i_q] \quad (7)$$

$$T_e = J\frac{2}{P}\frac{dw_e}{dt} + B\frac{2}{P}\omega_e + T_l \quad (8)$$

$T_e$ is the electromagnetic torque; $T_l$ is the load torque; $P$ represents number of poles; $p$ is the differential operator; $B$ is the damping coefficient; $\theta_r$ is the rotor position; $\omega_r$ is the rotor speed; and $J$ is the moment of inertia.

The transformation of PMSM into equivalent separately excited DC motor has been made possible with the introduction of Field oriented or Vector control technique [10]. The main idea of the Vector oriented control is to force the stator current vector on the *q-axis* and rotor flux on the *d-axis.* This implies that $i_{d(ref)} = 0$. The electric torque is modified as $T_e = \frac{3}{2}\frac{P}{2}[\lambda_m i_q] = K_t i_q$ where $K_t$ is the motor torque constant. Hence in state space form, the equations are modeled as follows

$$p i_d = (v_d - r_s i_d + \omega_e L_q i_q)/L_d \quad (9)$$

$$p i_q = (v_q - r_s i_q - \omega_e L_d i_d - \omega_e \lambda_m)/L_q \quad (10)$$

$$\frac{dw_e}{dt} = \frac{1}{J}[\frac{P}{2}(T_e - T_l) - B\omega_e] \quad (11)$$

$$\omega_e = P\omega_r / 2 \quad (12)$$

$$p\theta_r = \omega_r \quad (13)$$

By using above equations and transformations, vector controlled model of PMSM in *d-q* frame of reference is modeled using MATLAB/SIMULINK.

## III. Problem Formulation

### A. FOPI Speed Controller

Fractional order calculus is used in control areas by more and more researchers in recent years [7, 12]. If $\omega_{ref}$ is the reference speed of motor, $\omega_{act}$ is the output of the drive, then in time domain the fractional $PI^{\alpha}$ is represented in following equation.

$$u(t) = K_P e(t) + K_i D_t^{-\alpha} e(t) \quad (14)$$

The input *e(t)* of controller is given as

$$e(t) = \omega_{ref} - \omega_{act} \quad (15)$$

Where $D_t^{\alpha}$ is the fractional differointegral operator. The following definition is used for the fractional derivative of order $\alpha$ of function $f(t)$.

$$\frac{d^{\alpha}}{dt^{\alpha}} f(t) \dot{=} \begin{cases} f^{(n)}(t), & if \quad a = n \in N, \\ \frac{t^{n-\alpha-1}}{\Gamma(n-\alpha)} * f^{n}(t) & if \quad n-1 < \alpha < n, \end{cases}$$

Where represents the time convolution between two functions. Expressing in frequency domain the transfer function of FOPI controller can be obtained as follows

$$K_P + \frac{K_i}{s^{\alpha}} \quad (16)$$

Where are $K_P$ and $K_i$ the proportional integral gain values of *fractional controller* and is the non-integer order of the fractional integrator. The design requirements like Rise time $(t_r)$, Settling time $(t_s)$, percent peak overshoot $(po\%)$ and Steady state error $(e_{ss})$ depends on these parameters so, to get the good transient response controller is to be tuned properly.

### B. Optimization of FO-PI parametric gains using BFO, PSO, ABFO techniques.

BFO, PSO and ABF-PSO techniques are utilized to get the controller parameters $(K_P, K_i, \alpha)$ based on speed error. The performance of PMSM varies according to FOPI parameters and it is judged by value of ITAE (Integral Time Absolute Error). ITAE is chosen as objective function because it has an advantage of producing smaller overshoots and oscillations.

$$\text{ITAE} = \int_0^{\infty} (t^* | e(t) |) dt \quad (17)$$

The main objective of stochastic algorithm is to minimize the objective function. Here as we can't integrate up to infinity the time is set as per the machine design requirement. In addition "Gradient Descent Search Method" is also used for tuning controller.

## IV. Classical Bacterial Foraging Optimization Algorithm

Bacterial foraging optimization is a new method based on foraging behavior of "Escherichia coli" (*E-coli)* bacterial present in the human intestine, and been already implemented to real world problems. In this foraging theory the objective of the animal is to search for and obtain nutrients in such a fashion that energy intake per unit time (E/T) is minimized. A group of bacteria move in search of food and away from noxious elements known as Foraging. BFO algorithm draws its inspiration from this foraging behavior. Bacteria have a tendency to gather to the nutrient-rich areas by activity called Chemotaxis. Its movement and behavior is characterized by spinning flagella which acts as a Biological motor and helps bacteria to swim. The control system of these bacteria that dictates how foraging should be precede is subdivided into four sections namely Chemotaxis, Swarming, Reproduction, and Elimination and Dispersal [8].

*A. BF-PSO Algorithm:-*

BF-PSO algorithm combines both BFO and PSO. The aim is to make PSO ability to exchange social information and BF ability in finding new solution by elimination and dispersal, a unit length direction of tumble behavior is randomly generated. Random direction may lead to delay in reaching the global solution. In "BF-PSO" algorithm the unit length random direction of tumble behavior can be decided by the global best position and the best position of each bacterium. During the chemotaxis loop tumble direction is updated by:

$$\phi(j+1) = w*\phi(j) + C1*rand*(pbest - pcurrent) + C2*rand*(gbest - pcurrent) \quad (18)$$

Where *pbest* is the best position of each bacterium and *gbest* is the global best bacterial. The brief pseudo-code of BF-PSO has been provided below.

**Step 1.** Initialize parameters $p, S, N_c, N_s N_{re}, N_{ed}, P_{ed}$, $C(i)\ (i = 1,2,3....,S),\ \theta^i$ where,

- $p$ : Dimension of the search space,
- $S$ : Total number of bacteria in the population,
- $N_c$ : Number of chemotactic steps,
- $N_s$ : Swimming length, $N_c > N_s$
- $N_{re}$ : Number of reproduction steps,
- $N_{ed}$ : the number of elimination-dispersal events,
- $P_{ed}$ : probability of elimination-dispersal,
- $\theta^i$ : Location of $i_{th}$ bacterium.
- $C(i)$ The size of step taken in random direction specified by the tumble. Generate a random vector $\phi(j)$ which elements lie in [-1, 1]
- $C1, C2, w$ : PSO parameters

**Step 2.** Elimination-dispersal loop: *l=l+1.*

**Step 3.** Reproduction loop: *k=k+1.*

**Step 4.** Chemotaxis loop :*j=j+1.*

For $i = 1,2,3,...,S$ take a chemotactic step for bacterium $i$ as follows

a) Compute cost function , $ITAE(i, j, k, l)$.

b) Let $ITAE_{last} = ITAE(i, j, k, l)$ to save this value since we may get better value via run.

c) Tumble: Let

$$\phi(j+1) = w*\phi(j) + C1*rand*(pbest - pcurrent) + C2*rand*(gbest - pcurrent)$$

d) Move: Let $\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\phi(j)$

e) Compute $ITAE(i, j, k, l)$

f) Swim:

(i) Let $m = 0$, counter for swim length

(ii) while $m < N_s$ (if have not climbed down to long)

- Let : $m = m + 1$
- If $ITAE(i, j+1, k, l) < ITAE_{last}$ (if doing better).
- Let $ITAE_{last} = ITAE(i, j+1, k, l)$ and $\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\phi(j)$ use this $\theta^i(j+1,k,l)$ to compute new cost function.
- Else, let $m = N_s$ this is the end of while loop

g) Go to next bacterium $(i+1)$ if $i \neq S$ [i.e., go to (b) to process the next bacterium]

**Step 5.** $j < N_c$ go to **[Step 4]** In this case continue Chemotaxis since life of the bacteria is not over.

**Step 6.** Reproduction:

(a) For given k, l and for each bacteria $i = 1,2,3,....S$ let

$$ITAE^i_{health} = \sum_{j=1}^{N_c+1} ITAE(i, j, k, l) \quad (19)$$

be the health of bacterium $i$ ( a measure of how many nutrients it got over its life time and how successful it was avoiding at noxious substances ). Sort bacterium in order of ascending cost $ITAE_{health}$ (higher cost means lower health)

(b) The $S_r$ bacteria with highest $ITAE_{health}$ values die and remaining $S_r$ bacteria with best values split and this is performed by the copies that are made are placed at same location as their parent.

**Step 7.** If $k < N_{re}$ go to **[Step 3].** Since in this case specified reproduction steps are not reached, start next generation of chemotactic loop

**Step 8.** Elimination-dispersal: -

For $i = 1,2,3....S$ with probability $P_{ed}$, eliminate and disperse each bacterium, which results in keeping number of bacteria in the population constant. To do this, if a bacterium is eliminated, simply disperse another one to a random location on the optimization domain. If $l < N_{ed}$ go to **[Step 2]**; otherwise end.

*B. The ABF-PSO algorithm for optimization of FOPI controller parameters*

Chemotaxis is a foraging strategy that implements a type of local optimization where the bacteria try to climb up the nutrient concentration, avoid noxious substance and search for ways out of neutral media. A chemotactic step size varying as the function of the current fitness value is expected to provide better convergence behavior as compared to a fixed step size. A simple adaption scheme for the step size for $i_{th}$ bacterium given in following equation is employed to get better optimal controller parameters for PMSM drive.

$$C(i) = \frac{|j^i(\theta)|}{|j^i(\theta) + \psi|} = \frac{1}{1 + \psi / |j^i(\theta)|} \quad (20)$$

Where $\psi$ is positive constant.

$j^i(\theta)$ = cost function of the $i_{th}$ bacterium.

$C(i)$ = variable run length unit of $i_{th}$ bacterium

If $j^i(\theta)$ tends to zero then $C(i) \rightarrow 0$ and when $j^i(\theta) \rightarrow$ large $C(i) \rightarrow 1$. This implies that the bacterium which is in the vicinity of noxious substance associates with higher cost function. Hence it takes larger steps to migrate to a place with higher nutrient concentration. Use of Eqn (20) in Eqn (18) is expected to give improved convergence performance compared to fixed step size due to the above phenomenon.

**Step 1.** Same as that of BF-PSO based optimization

**Step 2-3.** Same as that of BF-PSO, but only difference is that while updating location in Eqn (18) (and also in swim) the adaptive run length unit, $C(i)$ defined in Eqn (20) is used instead of fixed run length unit.

**Step 5-8.** Same as that of BF-PSO based optimization technique.

## V. SIMULATIONS AND RESULTS

*A. Experimental Settings*

This section describes about the parametric set up of the PMSM drive and also the algorithmic parameters. Design Specifications of Drive are Peak overshoot ($po\%$) < 2, Rise time ($t_r$) < 0.02sec, Settling time ($t_s$) < 0.6 sec, Steady state error ($e_{ss}\%$) < 0.1

**Table 1 Parametric set up of PMSM Drive**

| Variable | Actual meaning | Value | Units |
|---|---|---|---|
| $r_s$ | Stator resistance | 2.0 | $\Omega$ |
| $L_d$ | *d-axis* inductance | 2.419 | $mH$ |
| $L_q$ | *q-axis* inductance | 2.419 | $mH$ |
| $J$ | Moment of Inertia | 0.0034468 | $kg - m^2$ |
| $\lambda_m$ | Magnet mutual flux | 0.27645 | $V / rad / \sec$ |
| $B$ | Damping coefficient | 0.0027715 | $Nm / rad / \sec$ |
| $P$ | No. of Poles | 8 | |

Simulation is done for a time T=1 sec under a load torque 5 Nm with a reference speed of 1300 rpm. The range for $K_P$ is taken from 0 to 1, for that of $K_i$ is 0 to 10 and $\alpha$ is taken between 0 and 1. Various parameters employed in the simulation study for BFO, PSO, ABF-PSO are given below

$S = 10$, $N_c = 5$, $N_s = 4$, $N_{re} = 2$, $N_{ed} = 2$, $P_{ed} = 0.25$, $C(i) = 0.075$, $\psi = 180$, $C1 = C2 = 2.05$, $w = 0.9$, No of particles fixed for PSO are 25 . As the optimization cannot be obtained in a single iteration, we used a total of 100 iterations for each algorithm for comparison to be fare enough

*B. Step response of FOPI controller*

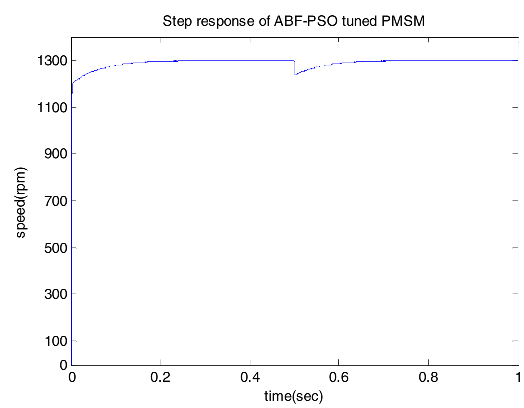


**Figure 3. Speed response of ABF-PSO tuned FOPI controller for PMSM Drive**

**Table 2 Comparison of Dynamic response of FOPI controller with ABF-PSP, BFO, PSO and traditional method**

| Method/parameter | Gradient | PSO | BFO | ABF-PSO |
|---|---|---|---|---|
| $K_p$ | 0.2541 | 0.3248 | 0.4375 | 0.5851 |
| $K_i$ | 4.5352 | 6.1264 | 7.2467 | 9.9531 |
| $\alpha$ | 0.5 | 0.5 | 0.7 | 0.9 |
| Rise time (sec) | 0.0042 | 0.0034 | 0.0025 | 0.0019 |
| Peak overshoot | 3.5069% | 1.2274% | 0.7036% | 0.0065% |
| Settling time (sec) | 0.6384 | 0.5704 | 0.5589 | 0.5575 |
| Steady state error | 1.2517 | 0.9087 | 0.8479 | 0.00412 |

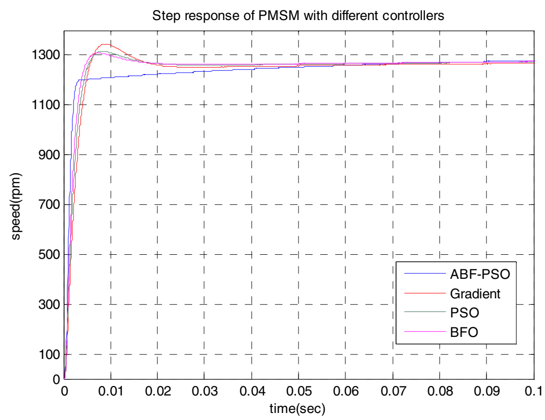


**Figure 4. Speed responses of PMSM using FOPI controller before load**

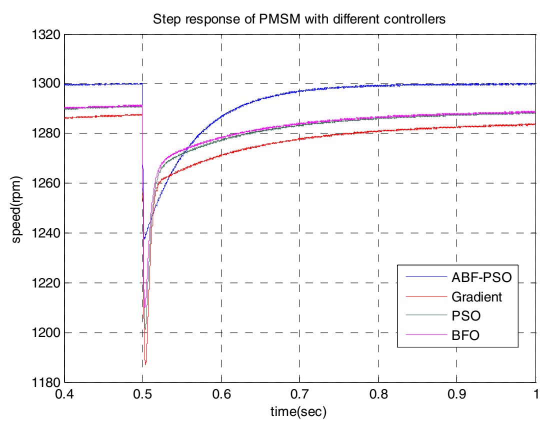


**Figure 5. Speed response of PMSM using FOPI controller after load**

## IV. CONCLUSIONS

In this paper a novel modified BF-PSO algorithm termed ABF-PSO is used in tuning a FOPI controller for PMSM drive. From the simulations and results table and also from the graphs obtained we can conclude that proposed method is performing better compared to that of PSO, BFO and gradient descent method. The design specifications like percent overshoot, steady state error are very much improved in ABF-PSO tuned FOPI controller.

Our further research will be focused on implementing FOPI controller to the sensor less PMSM drive.

## REFERENCES


[1] B.K. Bose, "Power electronics and motion control-technology status and recent trends", IEEE Trans. Ind. Applications, vol.29, pp.902-909, Sept./Oct., 1993.

[2] T.A. Lipo, "Recent progress in the development of solid state ac motor drives", IEEE Trans. Power Electronics.,vol. 3, pp. 105-117.

[3] Peter Vas, "Sensorless and Direct Torque Control, 1at Edition, Oxford University Press, 1988.

[4] J.J. Slotine and W. Li, *Applied Nonlinear Control.* Englewood Cliffs, NJ: Pearson Education, 1991.

[5] L.C. Jain, C.W. De Silva and L.C. Jain, "Intelligent Adaptive Control: Industrial Applications." Boca Raton, FL: CRC, Dec. 1998.

[6] K.J. Astrom and T. Hagglund, "The future of PID control", Control Eng. Pract., vol. 9, no. 11, pp. 11631175, Nov. 2001.

[7] I. Podluby, "Fractional-order systems and $PI^{\lambda}D$ controllers", IEEE Trans. On Automatic Control. Vol. 44, no.1, pp. 208-213 1999.

[8] K.M. Passion, "Bio mimicry of bacterial foraging for distributed optimization and control", IEEE Control Systems Magazine (2002) 52-67.

[9] W. Karoni Bacterial foraging oriented by particle swarm optimization strategy for PID tuning. In *GECCO 2008: Proceedings of the Genetic and Evolutionary computation conf,* pages 1823-1826, Atlanta, GA, USA, July 2008. ACM

[10] Song Chi, M.S.E.E "Position-Sensorless Control of Permanent Magnet Synchronous Machine Over Wide Range Speed", PH.D Thesis, Ohio State University 2001.

[11] P.Pillay, R. Krishnan: "Modeling, Simulation and Analysis of Permanent-Magnet Motor Drives, Part I; The Permanent-Magnet Synchronous Motor Drive", IEEE Trans. Ind. APP. Vol. 25, no.2, pp. 265-273, 1989.

[12] D. Valrio, "Fractional Robust System control" Ph. D thesis. Instituo Superior Tcninca, Universidadet Tcninca de Lisboa. 2005.