

# Optimal Gain Scheduling of PID Controller for the Speed Control of PMSM Drive Using Bio-Inspired Optimization Algorithms

Satish Kumar Injeti and M. Divyavathi

Department of Electrical and Electronics Engineering  
GMR Institute of Technology, Rajam, A.P., India-532127

*Abstract:* PMSM is widely used in servo-drive applications because of its advantages such as high efficiency, high power density and torque/inertia ratio and maintenance free. In a permanent magnet synchronous motor, the dc field winding of the rotor is replaced by a permanent magnet. The advantages are elimination of field copper loss, higher power density, lower rotor inertia and more robust construction of the rotor. The PMSM has higher efficiency than an induction motor, but generally its cost is higher, which makes the life cycle cost of the drive somewhat lower. In this paper, the speed control (tracking) of PMSM is achieved through PID controller. On-line tuning of PID controller parameters is done with the help of popular bio inspired optimization techniques such as PSO, Bat and Firefly algorithms based on most effective objective function Integral Time Absolute Error (ITAE). Obtained results are presented in tabular as well as graphical form for better understanding.

*Keywords:* PMSM, PID controller, Performance indexes (ISE, IAE, ITSE, ITAE), Bio-inspired optimization techniques

## 1. Introduction

Permanent magnet synchronous motors are widely used in high performance drives such as industrial robots and machine tools. In recent years, the magnetic and thermal capabilities of the Permanent Magnet Synchronous Motors have been considerably increased by employing the high-coercive permanent magnet material [1]. The speed control of synchronous motor depends upon two factors visualization number of poles,  $P$  and supply frequency,  $f$ . as in case of shipping propulsion, the speed of the motor can be changed by changing the speed of the alternator – the speed of the motor changes exactly in the same proportion as that of the alternator supplying power to it. It is to be noted here that the voltage and frequency are directly proportional to the speed at which alternator is driven [2-3]. Unlike a DC motors Permanent magnet synchronous motors (PMSM) are very popular in a wide range of applications, the PMSM does not have a Commutator, which makes it more reliable than a DC motor. The PMSM also has advantages when compared to an AC induction motor. The PMSM generates the rotor magnetic flux with rotor magnets, achieving higher efficiency. Therefore, the PMSM is used in applications that require high reliability and efficiency [4-5]. Every electric motor has to have some sort of controller. The motor controller will have different features and complexity depending on the task that the motor will be performing. An electrical adjustable-speed drive consists of an electric motor and a speed controller plus auxiliary devices and equipment. In common usage, the term “drive” is often applied to just the controller.

The controllers used are P controllers, PI controllers, PID controllers to control the speed of the PMSM drive. Most of the industrial processes are uses conventional PID controllers due to their simple and robust design, affordable price, and effectiveness for linear systems [6], but conventional PID controllers are usually not effective if the processes involved are higher order and time delay systems [7]. But in some applications it may be useful to employ more general controllers, which makes it easier to reach the system specifications and improve their performance, though they can be also more difficult to tune manually. Although the number of parameters to adjust in a PID is very small, there are many tuning rules [8]. It has been experimentally checked that more than 30% of controllers are operating in manual mode and

65% of the loops operating in automatic mode are poorly tuned because of the inappropriate parameters [9]. Currently, most of the current-speed closed-loop control in the PMSM servo system adopts PID controller [10]. Nevertheless, the PID controller has poor performance in PMSM control due to the inappropriate parameters.

Over the years, many methods have been proposed for the tuning of PID controller, both in the deterministic or in the stochastic frameworks [11, 12]. The pole placement and minimum variance techniques [13], Ziegler-Nichols and Cohen-Coon methods [6], are the most commonly used conventional methods for tuning PID controllers. However, there will be large overshoot and shaking by these methods [14]. There are also other advanced methods to design the PID controller, such as  $H_\infty$  method [15] or the linear matrix inequality technique [16]. Tuning of the PID controller is not a straightforward problem especially when the plants to be controlled are nonlinear and unstable. It can be considered as a parameter optimization process to achieve a good system response, such as a minimum rise time, overshoot, and regulating time. Thus, the tuning process of the controller has multiple objectives to be achieved, and they are conflicting with one another in most cases.

During the past decades, great attention has been paid to the stochastic approach, which has potential to solve this problem [17, 18]. GA, for instance, is a powerful search algorithm used by research to optimize PID controllers. The optimization process of GA does not require any gradient information and inherent parallelism in searching the design space. Latest research is focused on optimization methods based on intelligent algorithms, which results to solve the difficult optimization problems in very effective manner. The algorithms are inspired by the Bio-inspired organisms in nature. Among all Bio-inspired algorithms we have chosen Particle Swarm Optimization (PSO), Fire Fly Algorithm (FFA) and BAT Algorithms as optimization methods. These three algorithms are the main focus on the ITAE minimization and it will be useful for optimal tuning of PID controller parameters. In this paper, the four most common integral performance indexes are proposed as objective functions, these performance indexes will be used to obtain the parameters of the PID controllers. ITAE is proved to be more efficient than the proposed indexes ISE, IAE and ITSE based on settling time, peak over shoot and minimum convergence criterion [19, 20]. Based on ITAE, three bio inspired optimization techniques are adapted to optimize the parameters of PID controller for speed control of a PMSM drive. The structure of this paper as follows, Section 2 represents the different controllers and the objective functions. Section 3 represents the Transfer function models of the PMSM drive, Speed and Current controller Transfer functions and feedback (current, speed) paths transfer functions, Inverter model. In section 4 presents discussion of different optimization techniques and their performance. In section V, conclusion obtained from the comparison of the simulation results.

## 2. Back ground of PID controller

### A. Proportional (P) Control

The block diagram for proportional controller is shown in Figure. 1. The current error value may changes by using the proportional (P) controller. The error value is adjusted by multiplying a constant  $K_p$ , i.e., called as the proportional gain. The transfer function of a proportional controller is  $K_p$ , it is simply the gain value. If error signal,  $e(t)$  is the input to the controller then the output is  $u(t) = K_p e(t)$  or in a Laplace transform domain  $U(s) = K_p E(s)$ . As  $K_p$  value increases the unit-step response may become faster and eventually the feedback system may become unstable. For different  $K_p$  values the steady-state plant outputs are different with same unit-step reference input. Tuning theory and industrial practice indicate that the proportional term should contribute the bulk of the output change.

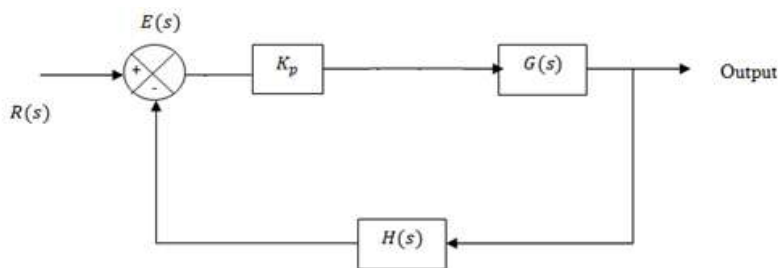


Figure 1. Block Diagram of proportional controller

**B. Proportional Plus Integral (PI) Control**

The block diagram for proportional plus integral controller is shown in Figure 2. Integral control action itself is not sufficient, as it introduces hunting in the system. Therefore a combination of Proportional and integral control action is introduced to improve the system performance. In this type of system, the actuating signal consists of proportional error signal added with the integral of the error signal.

Mathematically,

$$u(t) = e(t) + K \int_0^t e(t)dt \tag{1}$$

Where  $e(t)$  = error signal;

And  $\int_0^t e(t)dt$  = integral of error signal (2)

Or  $U(S) = E(s) [1 + \frac{K}{S}]$  (3)

Proportional plus Integral control increases the order and type of the system by one, respectively. Therefore, it improves steady state performance. The effect of proportional and integral control improves system steady state response with in less time and rise time also increases.

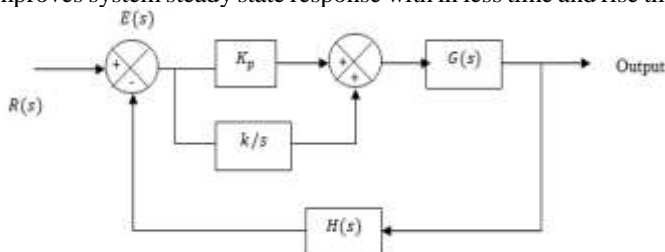


Figure 2. Block Diagram of Proportional-Integral control

**C. Proportional plus Integral plus Derivative (PID) Control**

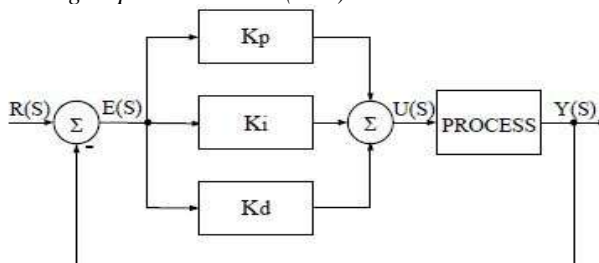


Figure 3. Block Diagram of PID Controller and Plant

The block diagram for proportional plus integral plus derivative controller is shown in Figure. 3. The output of a PID controller, equal to the control input to the plant, in the time-domain is as follows:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de}{dt} \tag{4}$$

The error signal ( $e$ ) will be sent to the PID controller, and the controller computes both the derivative and the integral of this error signal. The control signal ( $u$ ) to the plant is equal to the proportional gain ( $K_p$ ) times the magnitude of the error plus the integral gain ( $K_i$ ) times the integral of the error plus the derivative gain ( $K_d$ ) times the derivative of the error. This control signal ( $u$ ) is sent to the plant, and the new output ( $y$ ) is obtained. The new output ( $y$ ) is then fed back and compared to the reference to find the new error signal( $e$ ) . The controller takes this new error signal and computes its derivative and it's integral again, ad infinitum.

The transfer function of a PID controller is found by taking the Laplace transform of Eq. 4

$$K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} \quad (5)$$

$K_p$  = Proportional gain  $K_i$  = Integral gain  $K_D$  = Derivative gain

### 3. Mathematical model of PMSM drive

#### A. Speed-Controller Design

The design of the speed-controller is important from the point of view of imparting desired transient and steady-state characteristics to the speed-controlled PMSM drive system. A proportional-plus-integral controller is sufficient for many industrial applications; hence, it is considered in this section. Selection of the gain and time constants of such a controller by using the symmetric-optimum principle is straightforward if the d axis stator current is assumed to be zero. In the presence of a d axis stator current, the  $d$  and  $q$  current channels are cross-coupled, and the model is nonlinear, as a result of the torque term. Under the assumption, that  $i_{ds}^r = 0$  , the system becomes linear and resembles that of a separately-excited dc motor with constant excitation. From then on, the block-diagram derivation, current-loop approximation, speed-loop approximation, and derivation of the speed-controller by using symmetric optimum are identical to those for a dc or vector-controlled induction-motor-drive speed-controller design.

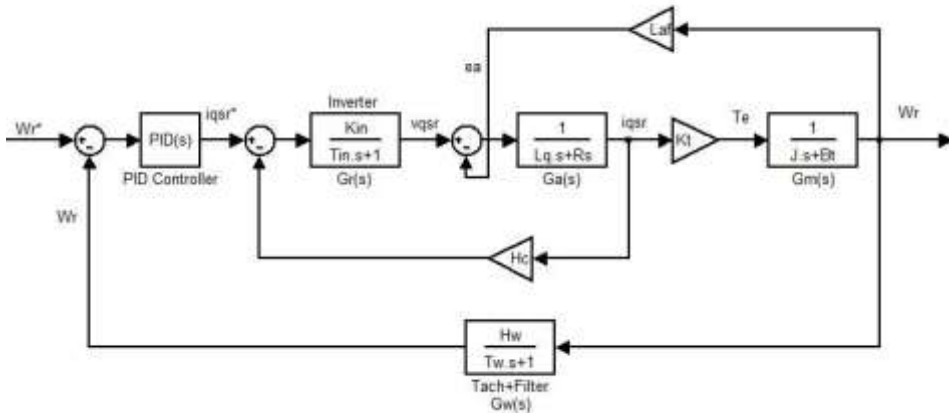


Figure 4. Block Diagram of the Speed Controlled PMSM Drive

#### B. PMSM transfer function

The motor q axis voltage equation with the d axis current being zero becomes

$$v_{qs}^r = (R_s + L_q P) i_{qs}^r + \omega_r \lambda_{af} \quad (6)$$

And the electromechanical equation is

$$\frac{P}{2} (T_e - T_l) = j P \omega_r + B_1 \omega_r \quad (7)$$

Where the electromagnetic torque is given by

$$T_e = \frac{3}{2} \cdot \frac{P}{2} \lambda_{af} i_{qs}^r \quad (8)$$

and, if the load is assumed to be frictional, then

$$T_l = B_l \omega_m \quad (9)$$

Which, upon substitution, gives the electromechanical equation as:

$$(JP + B_t)\omega_r = \left\{ \frac{3}{2} \left( \frac{P}{2} \right)^2 \cdot \lambda_{af} \right\} i_{qs}^r = K_t \cdot i_{qs}^r \quad (10)$$

Where,

$$B_t = \frac{P}{2} B_l + B_1 \quad (11)$$

$$K_t = \frac{3}{2} \left( \frac{P}{2} \right)^2 \cdot \lambda_{af} \quad (12)$$

The Eq. 6 and Eq. 10, when combined into a block diagram with the Current - and speed-feedback loops added, are shown in Figure. 4.

### C. Inverter transfer function

The inverter is modeled as a gain with a time lag by

$$G_r(s) = \frac{K_{in}}{1+sT_{in}} \quad (13)$$

Where

$$K_{in} = 0.65 \frac{V_{dc}}{V_{cm}} \quad (14)$$

$$T_{in} = \frac{1}{2f_c} \quad (15)$$

Where  $V_{dc}$  is dc-link voltage input to the inverter,  $V_{cm}$  is maximum control voltage, and  $f_c$  is switching (carrier) frequency of the inverter.

The induced e.m.f due to rotor flux linkages,  $e_a$ , is expressed as

$$e_a = \lambda_{af} \omega_r(V) \quad (16)$$

### D. Speed controller transfer function

A proportional-plus-integral-plus-derivative (PID) controller is used to process the speed error between the speed-reference and filtered speed-feedback signals. The transfer function of the speed controller is given as

$$G_s(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) = K_p + \frac{K_i}{s} + K_d s \quad (17)$$

Where,  $K_p$  is the proportional gain,  $T_i$  is the integral time and  $T_d$  is the derivative time.

### E. Feedback transfer functions

The feedback signals are current and speed, which are processed through first-order filters. They are given in the following.

1. Current feedback transfer function: Very little filtering is common in the current feedback signal; the signal gain is denoted by

$$G_c(s) = H_c \quad (18)$$

2. Speed-feedback transfer function: The speed-feedback signal is processed through a first-order filter is denoted by

$$G_\omega(s) = \frac{H_\omega}{1+sT_\omega} \quad (19)$$

Where  $H_\omega$  is the gain and  $T_\omega$  is the time constant of the speed filter.

The speed filter accepts the speed signal as input and produces a modified speed signal for comparison to the speed-reference signal,  $\omega_r^*$ .

## 4. Objectives and optimization

### A. Objective function

The optimization technique is applying on to the objective functions, which one is most efficient to evaluate the fitness of each particle. The objective functions used here are Integral of

the squared error (ISE), Integral of absolute magnitude of the error (IAE), Integral of time multiplied by squared error (ITSE), Integral of time multiplied by absolute error (ITAE). In this paper we evaluate all the above performance indices and compare all the performances ITAE is settles faster and minimize error signal, so ITAE is the most suitable one [20].

The performance indices are defined as follows:

- Integral of the squared error (ISE) =  $\int_0^{\tau} e(t)^2 dt$
- Integral of time multiplied by square error (ITSE) =  $\int_0^{\tau} t_{sim} * e(t)^2 dt$
- Integral of absolute magnitude of the error (IAE) =  $\int_0^{\tau} |e(t)| dt$
- Integral of time multiplied by absolute error (ITAE) =  $\int_0^{\tau} t_{sim} * |e(t)| dt$

Where,  $e(t)$  is the time domain error signal.  $t_{sim}$  is the time range of simulation. The problem constraints are the PID controller parameter bounds. Therefore, the design problem can be formulated as the optimization problem and the objective function is expressed as

$$OF = \text{Minimize } (f) \text{ where, } f \in ISE \text{ or ITSE or IAE or ITAE} \quad (20)$$

Subjected to constraints

$$K_{Pmin} \leq K_P \leq K_{Pmax} \quad (21)$$

$$K_{Imin} \leq K_I \leq K_{Imax} \quad (22)$$

$$K_{Dmin} \leq K_D \leq K_{Dmax} \quad (23)$$

### B. Optimization algorithms

To search the highly multimodal space, most popular bio inspired optimization algorithms like PSO, Bat and Firefly are employed in this paper.

#### B.1. Particle Swam Optimization (PSO)

Particle Swarm Optimization (PSO) is a population-based continuous optimization technique proposed by Ebberhert and Kennedy [21]. The algorithm simulates a simplified social milieu in a swarm of potential solutions (called “particles”), which means that a single particle bases its search not only on its own experience but also on the information given by its neighbors in the swarm. This paradigm leads to successful results and contributes to the popularity of PSO.

- Steps Followed in PSO Algorithm

PSO algorithm implementation steps are as follows:

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

$$X_{i,j} = (x_{1,1}, x_{1,2}, x_{1,3}, \dots \dots \dots x_{pop,n}), i = 1 \text{ to } pop \text{ and } j = 1 \text{ to } n \quad (24)$$

$$V_{i,j} = (v_{1,1}, v_{1,2}, v_{1,3}, \dots \dots \dots v_{pop,n}), i = 1 \text{ to } pop \text{ and } j = 1 \text{ to } n \quad (25)$$

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

Step 2: Calculation of fitness value of the objective function using Eq. 20.

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:

$$pbest_m^{k+1} = \begin{cases} pbest_m^k & \text{if } f_m^{k+1} \geq f_m^k \\ x_m^{k+1} & \text{if } f_m^{k+1} \leq f_m^k \end{cases} \quad (26)$$

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*.

$$gbest_m^{k+1} = \begin{cases} gbest_m^k & \text{if } f_m^{k+1} \geq f_m^k \\ pbest_m^{k+1} & \text{if } f_m^{k+1} \leq f_m^k \end{cases} \quad (27)$$

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:

$$V_m^{k+1} = \omega V_m^k + C_1 \text{rand}(\text{pbest}_m^k - X_m^k) + C_2 \text{rand}(\text{gbest}^k - X_m^k) \quad (28)$$

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

$$\omega = \omega_{\max} - \frac{(\omega_{\max} - \omega_{\min})}{\text{iter}_{\max}} * \text{iter} \quad (29)$$

$C_1, C_2$  are the acceleration coefficients usually in range [1, 2]. A large inertia weight ( $w$ ) 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,

$$\text{If } V_{id} > V_{\max}, \text{ then } V_{id} = V_{\max} \text{ (or) If } V_{id} < -V_{\max} \text{ then } V_{id} = -V_{\max} \quad (30)$$

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

$$X_j^{k+1} = X_j^k + V_j^{k+1} \quad (31)$$

Step 8: If the number of iterations reaches the maximum i.e.  $\text{iter} = \text{iter}_{\max}$ , then go to step 9. Otherwise, go to step 2.

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

## B.2. Bat Algorithm (BA)

The majority of heuristic and meta-heuristic algorithms have been derived from the behavior of biological systems and/or physical systems in nature. The Bat Algorithm (BA) is based on the echolocation behavior of bats, proposed by Xin-She-Xang for engineering optimization in [22, 24]. If we idealize some of the echolocation characteristics of micro bats, we can develop various bat-inspired algorithms or bat algorithms.

### - Steps for implementation of Bat algorithm

In this section, BAT algorithm is described for solving the optimal placement of capacitors in radial distribution systems.

Step 1: Initialization of problem and algorithm parameters

In the first step, the algorithm parameters such as population size (Pop), dimension of the problem and maximum number of iterations (Itermax), limits of  $f$ ,  $\beta$  and  $A$  are to be initialized. And initialize dimension of the problem.

Step 2: Random generation of PID gains

$$X = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_{d-1}^1 & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_{d-1}^2 & x_d^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{\text{pop}-1} & x_2^{\text{pop}-1} & \dots & x_{d-1}^{\text{pop}-1} & x_d^{\text{pop}-1} \\ x_1^{\text{pop}} & x_2^{\text{pop}} & \dots & x_{d-1}^{\text{pop}} & x_d^{\text{pop}} \end{bmatrix} \quad (32)$$

$$x_i^j = x_{\min,i} + (x_{\max,i} - x_{\min,i}) * \text{rand}() \quad (33)$$

Where,  $d$  is the number of decision variables,  $x_i^j$  represents PID gains, i.e.,  $j^{\text{th}}$  population of  $i^{\text{th}}$  parameter, which is generated randomly in between the limits as  $x_{\max,i}$  and  $x_{\min,i}$  are the  $i^{\text{th}}$  parameter limits and  $\text{rand}()$  is a random number in between 0 and 1.

$$\text{Soln} = [X] \quad (34)$$

In Bat algorithm, Soln. represents a group of Bats, where Bat is one position in search space. Bat is a solution that contains capacitor locations and sizes.

Step 3: Fitness evaluation

Calculate the fitness value for each initial solution using Eq. 20 and record the best solution.

Step 4: Start evolution procedure of BAT algorithm. Assign frequency for each Bat randomly

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \quad (35)$$

Where  $\beta \in [0, 1]$  is a random vector drawn from a uniform distribution

Initially each bat is randomly assigned a frequency which is drawn uniformly from  $[f_{\min}, f_{\max}]$ .

Step 5: Random generation of Bat positions (PID parameters)

$$XV_i^t = V_i^{t-1} + (X_i^t - \text{best}X_*)f_i \quad (36)$$

$$X_{\text{new}_i}^t = X_i^{t-1} + XV_i^t \quad (37)$$

Step 6: Fitness evaluation (Objective function)

Calculate the fitness value for each initial solution using Eq. 20

Step 7: Selection

Compare each new bat solution with corresponding initial bat solution and replace better solution new bats to initial bat & find best bat, best solution among initial bats.

Step 8: Stopping criterion

If the maximum number of iterations is reached, computation is terminated. Otherwise, Step 4 to Step 7 is repeated.

### B.3. Firefly Algorithm (FFA)

The idealized Flashing characteristics of fireflies are used to develop firefly-inspired algorithm. Firefly Algorithm (FFA) [23, 24] developed by Xin-She Yang at Cambridge University, use the following three idealized rules:

- All the fireflies are unisex so it means that one firefly is attracted to other fireflies irrespective of their sex.
- Attractiveness and brightness are proportional to each other, so for any two flashing fireflies, the less bright one will move towards the one which is brighter.
- Attractiveness and brightness both decrease as their distance increases. If there is no one brighter than other firefly, it will move randomly.

The brightness of a firefly is determined by the view of the objective function. For a maximization problem, the brightness is simply proportional to the value of the objective function. Other forms of the brightness could be defined in an identical way to the fitness function in genetic algorithms.

The distance between any two fireflies  $i$  and  $j$  at  $x_i$  and  $x_j$ , is expressed as

$$r_{ij} = \sqrt{(x_i - x_j)^2 - (y_i - y_j)^2} \quad (38)$$

The movement of the  $i$ th firefly is attracted to another more attractive (brighter) firefly  $j$ th is expressed as

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \epsilon_i \quad (39)$$

The problem specific implementation flow chart of FFA has been given in Figure 5



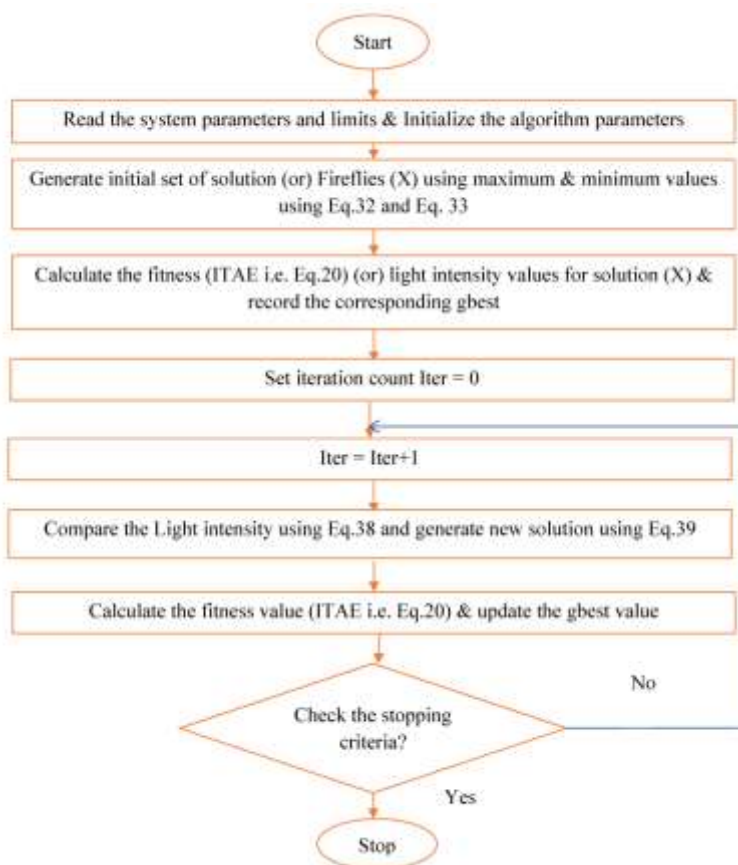


Figure 5. Implementation flow chart of Firefly Algorithm

**5. Simulation results and discussions**

Table 1. Parameters used in the simulation of PMSM drive

S. No.	Name of the parameter	Value
1	Stator Resistance	1.4 Ohm
2	q-axis self-inductance	0.009 H
3	d-axis self-inductance	0.0056 H
4	Mutual flux linkage due to rotor magnets	0.1546 Wb-Turn
5	Moment of Inertia	0.006 Kg-m <sup>2</sup>
6	Friction coefficient	0.01 N-m/(rad/s)
7	Number of poles	6
8	Constant frequency	2 KHz
9	Maximum control voltage	10 V
10	DC link voltage	285 V
11	Gain of current transducer	0.8 V/A
12	Gain of the speed filter	0.05 V/V

Form the mathematical modelling; a transfer function model of PMSM drive has been simulated using MATLAB/SIMULINK as shown in Figure 4. A PID controller is added to the PMSM drive to establish the closed loop speed. The parameters of PMSM are furnished in Table 1. For efficient automatic speed tracking of PMSM drive the gains of the PID controller should be optimized. Online gain scheduling of PID controller for PMSM drive has been done by bio

inspired optimization algorithms such as PSO, BAT and FFA. In any optimization algorithm, the optimal output obtained is always dependent on design of objective function which is to be minimized or maximized. So, an objective function plays a major role in the optimization process. In the present paper four various objective functions such as ISE, ITSE, IAE and ITAE are tested initially for speed control of PMSM drive using PSO algorithm. Later, most efficient objective function has been selected and implemented for the speed control of PMSM drive using Bat algorithm and Fire fly algorithm. Assigned values for optimization algorithm parameters are selected by trial and error method and are given in Table 2. All codes for optimization are self-developed MATLAB scripts using Intel Core i3 2.7 GHz Processor with 4 GB RAM.

Table 2. Parameter description for PSO, BA and FFA algorithms

Type	Parameter	Description	Assigned value
PSO	$P$	Population size	20
	$n_{par}$	Number of particles	6
	$W_{max}$	Maximum inertia weight factor	0.9
	$W_{min}$	Minimum inertia weight factor	0.4
	$C_1, C_2$	Cognitive parameters	2, 2
	$Iter_{max}$	Maximum no of iterations	50
BA	$pop$	Population of Bats	20
	$n$	Dimensional search space of a bats	6
	$A$	Loudness	0.50
	$r$	Pulse rate	0.50
	$f_{min}$	Minimum frequency	0.00
	$f_{max}$	Maximum frequency	2.00
	$Iter_{max}$	Maximum no of iterations	50
FFA	$N$	Population of fireflies	20
	$n$	Search space dimension of a firefly	6
	$\beta_0$	Initial attractiveness	1
	$A$	Randomness	0.25
	$\Gamma$	Absorption	1
	$Iter_{max}$	Maximum no of iterations	50

#### A. Selection of objective function for speed control of PMSM drive using PSO based PID controller

In order to select efficient objective function for the optimization process, four different objective functions have been tested using PSO algorithm for the speed control of PMSM drive. To analyze the performance of each objective function, PID control parameters are tuned at different reference speeds of PMSM drive i.e. lowest speed 100 rpm, medium speed 600 rpm and higher speed 1200 rpm. Performance Indices (ISE, IAE, ISTE, and ITAE) For PSO Tuned PID controller are presented in Table 3. Form Table 3, it is observed that ITAE has been performed well in achieving the desired target than other objective functions. Because the settling time for ITAE is around 0.6 s, but for other objective functions it is greater than 1 s. The settling time near to zero shows the efficiency of objective function for tuning of PID control parameters for speed control of PMSM drive. Form the results it is also evident that ITAE has been successful in achieving the quick speed control by tuning the PID controller parameters at different reference speeds.

## Optimal Gain Scheduling of PID Controller for the Speed Control

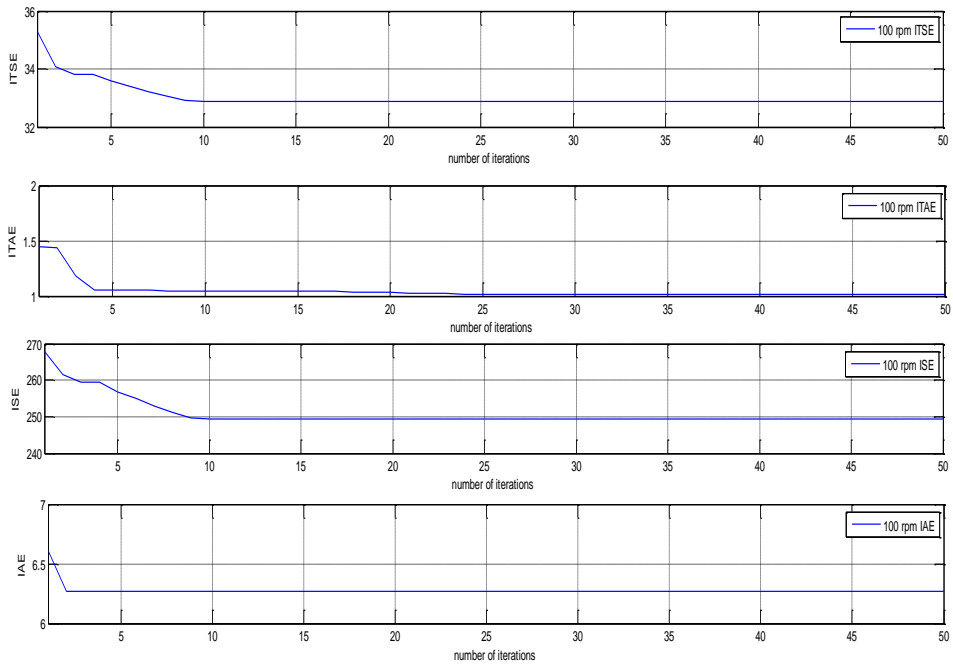


Figure 6. Performance comparison of objective functions using PSO algorithm at 100 rpm

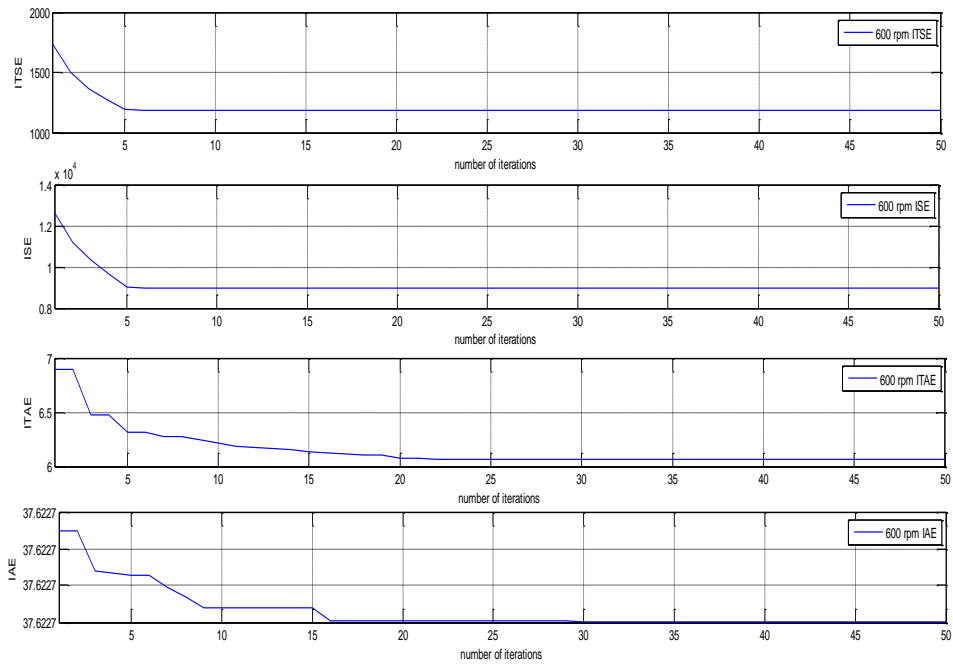


Figure 7. Performance comparison of objective functions using PSO algorithm at 600 rpm

Table 3. Performance Indices (ISE, IAE, ITSE, and ITAE) For PSO Tuned PID Controller

Objective Function	Reference Speed (rpm)	$K_p$	$K_i$	$K_d$	Settling Time (s)	Function Value
ISE	100	1.0000	4.0000	0.0090	1.1520	249.2672
	600	0.0010	0.0040	0.0000	1.1389	8.9736
	1200	0.0001	0.0004	0.0000	1.3940	3.5894
IAE	100	1.0000	4.0000	0.0046	1.01	6.2704
	600	0.8775	4.0000	0.0019	1.061	37.6227
	1200	0.9058	4.0000	0.0029	1.208	75.2454
ITSE	100	1.0000	4.0000	0.0090	0.983	32.8769
	600	0.0010	0.0040	0.0000	1.383	1.1836
	1200	0.0010	0.0040	0.0000	1.108	4.7343
ITAE	100	0.8050	4.0000	0.0009	0.6459	1.0108
	600	0.8050	4.0000	0.0009	0.6306	6.0647
	1200	0.8048	4.0000	0.0009	0.5606	12.1294

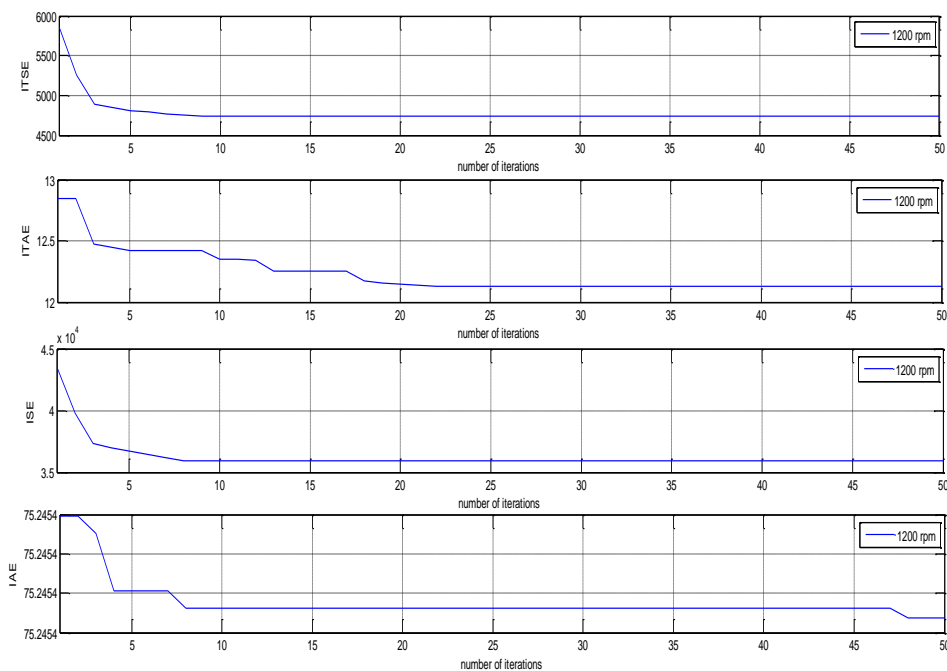


Figure 8. Performance comparison of objective functions using PSO algorithm at 1200 rpm

Convergence characteristics of PSO algorithm for speed control of PMSM drive based on different objective functions has been depicted in Figure 6, Figure 7 and Figure 8 respectively. From Figure 6, Figure 7 and Figure 8 it is observed that almost all objective functions has shown better convergence property at different reference speeds. And best optimal solution has been achieved before 30<sup>th</sup> iteration except IAE at 1200 rpm reference speed. The performance of objective functions in the context of speed control has been shown in Figure 9, Figure 10 and Figure 11 respectively. From Figure 9, Figure 10 and Figure 11 it is clear that ITAE has been successful in achieving the quick speed control of PMSM drive at all reference speeds among ISE, ITSE, IAE and ITAE. From Table 3 it is observed that the settling time for ITAE is quite smaller than that of other objective functions ISE, IAE and ITSE at all different reference speeds. Hence ITAE has been chosen as a best objective function for optimal tuning of PID controller for speed control of PMSM drive.

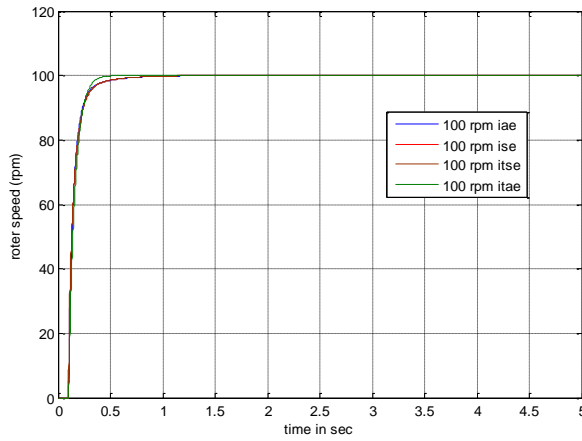


Figure 9. PMSM rotor speed response characteristics with ISE, IAE, ITSE and ITAE at 100 rpm reference speed

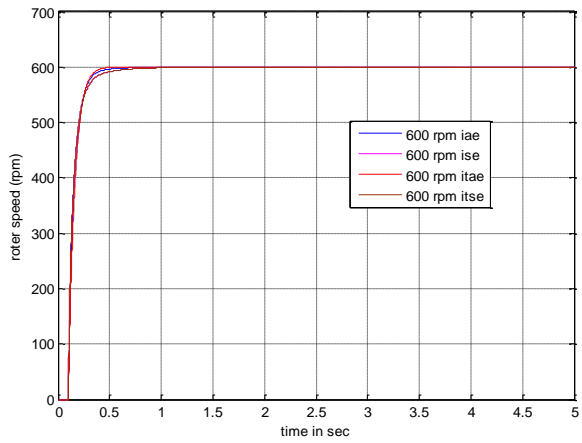


Figure 10. PMSM rotor speed response characteristics with ISE, IAE, ITSE and ITAE at 600 rpm reference speed

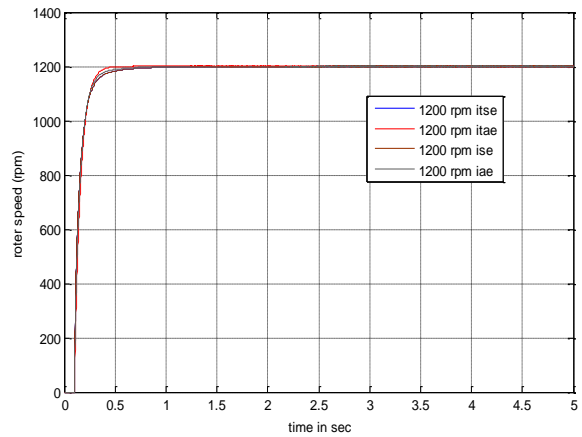


Figure 10. PMSM rotor speed response characteristics with ISE, IAE, ITSE and ITAE at 1200 rpm reference speed

*B. Simulation results of BAT and Firefly algorithms for speed control of PMSM drive using ITAE based PID controller*

This section describes the implementation and investigation of BAT and Firefly algorithms for optimal gain scheduling of PID controller for speed control of PMSM drive using ITAE as an objective function. And the obtained results have been compared with results of PSO. Comparison of numerical results has been presented in Table 4. From Table 4 it is observed that the settling time for PSO+PID and FFA+PID are in close agreement. The settling time for BAT+PID is larger than other methods. Among PSO+PID and FFA+PID, PSO+PID have been succeeded in attaining the reference speed at various levels. That means PSO algorithm has a great potential for tuning of PID controller for speed control of PMSM drive. From Table 4 it is evident that PSO and FFA completed the task (online tuning of PID controller for speed control of PMSM drive) in less than one second. PSO, BAT and FFA all are similar bio inspired meta-heuristic algorithms but with different evolution strategy. Among PSO, BAT and FFA, PSO has very simple evolution procedure. Hence PSO performed well in achieving the target with good convergence. The performance characteristics of PSO, BAT and FFA based PID controller for speed control of PMSM drive at different reference speeds has been shown in Figure 11, Figure 12 and Figure 13 respectively.

Table 4. Performance analysis of PSO, BAT and FFA based PID controller for speed control of PMSM drive

Controller	Reference Speed(rpm)	$K_p$	$K_i$	$K_d$	Settling Time	Function value
PSO+PID	100	0.8050	4.0000	0.0009	0.6459	1.0108
	600	0.8050	4.0000	0.0009	0.6306	6.0647
	1200	0.8048	4.0000	0.0009	0.5606	12.1294
BAT+PID	100	0.8006	3.5179	0.0090	2.2050	1.3960
	600	0.7898	3.8358	0.0036	1.8790	6.5992
	1200	0.7898	3.8358	0.0036	1.6420	13.1984
FFA+PID	100	0.6467	3.2281	0.0009	0.6932	1.3690
	600	0.7817	3.8873	0.0009	0.6315	6.3114
	1200	0.7288	3.6255	0.0009	0.5998	13.9221

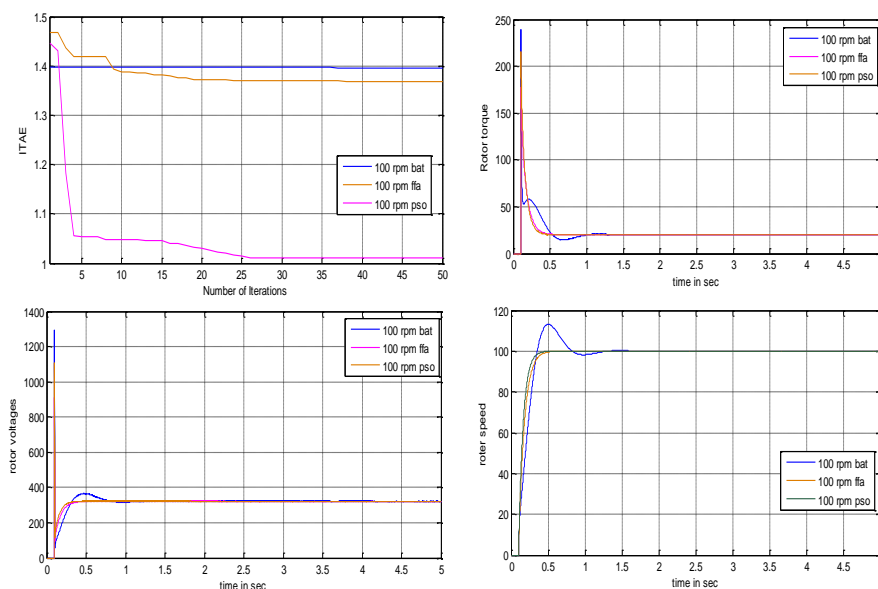


Figure 11. Performance comparison of PSO, BAT and FFA based PID controller at 100 rpm reference speed

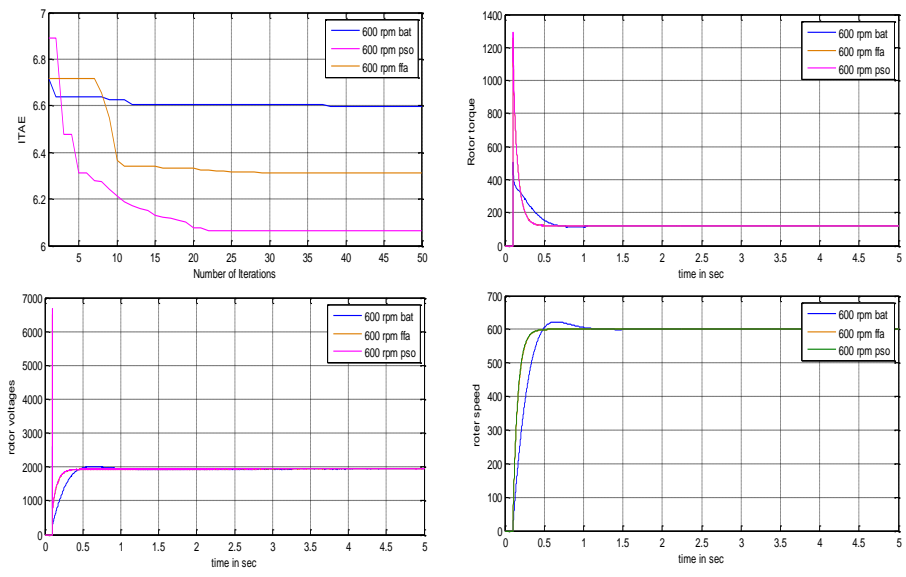


Figure 12. Performance comparison of PSO, BAT and FFA based PID controller at 600 rpm reference speed

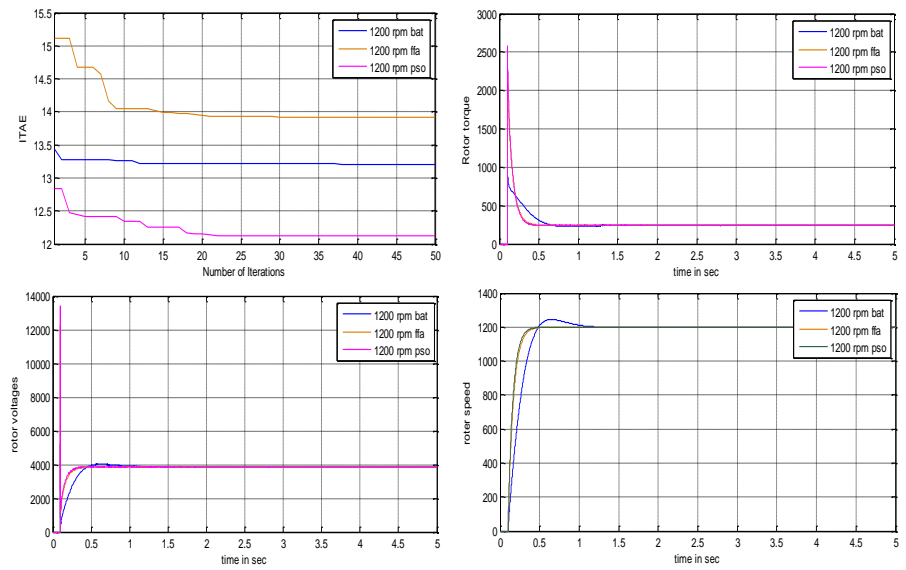


Figure 13. Performance comparison of PSO, BAT and FFA based PID controller at 1200 rpm reference speed

From Figure 11, Figure 12 and Figure 13 it is observed that PSO and FFA based PID controller has been performed well in the speed control of PMSM drive at various reference speeds. But comparatively PSO based PID controller performance is good. And over shoots are observed in rotor speed response plot with BAT based PID controller at all reference speeds. The reason behind this is evolution procedure of BAT algorithm is somewhat tedious and hence it takes comparatively larger time to settle down at reference speed. Figure 11, Figure 12 and Figure 13 also shows the rotor voltage response and rotor torque response at various reference speeds with PSO, BAT and FFA based PID controller for speed control of PMSM drive.

## 6. Conclusions

PID controller is a most popular controller in industry. The performance of PID controller depends on proper scheduling of gains which is a difficult task due to uncertainties in the industry. In this context present paper proposed an efficient methodology for automatic speed tracking of PMSM drive using bio inspired optimization based PID controller. PSO, BAT and FFA based online gain scheduling of PID controller for speed control of PMSM drive was developed and implemented with the help of various objective functions. From the obtained results it can be concluded that among ISE, IAE, ITSE and ITAE, ITAE is the best objective function for gain scheduling of PID controller for speed control of PMSM drive. Among PSO+PID, BAT+PID and FFA+PID, PSO+PID is proven to be robust as it was applied to control the speed of PMSM drive at different reference speeds. The developed control algorithm has been proven successfully in simulation and the next step is to be implemented in hardware using “dSPACE Module”.

## Appendix A

Nominal parameters of the transfer function model investigated as follows :

$$\begin{array}{llllll} K_{in} & 18.525 ; & R_s & 1.4 ; & B_t & 0.01 ; & H_\omega & 0.05 ; \\ T_{in} & 2.5 \times 10^{-4} ; & K_t & 2.087 ; & L_q & 0.009 ; & T_\omega & 0.002 ; \\ J & 0.006 ; & H_c & 0.8 ; & & & & \end{array}$$

## 7. References

- [1]. I. Takahashi and T. Noguchi, “A New Quick-Response and High-Efficiency Control Strategy of an Induction Motor,” *Ind. Appl. IEEE Trans.*, vol. IA-22, no. 5, pp. 820–827, 1986.
- [2]. A. Parviainen, M. Niemelä, and J. Pyrhönen, “Modeling of axial flux permanent-magnet machines,” *IEEE Trans. Ind. Appl.*, vol. 40, no. 5, pp. 1333–1340, 2004.
- [3]. A. Parviainen, M. Niemelä, and J. Pyrhönen, “Modeling of Axial-flux PM Machines”. In *Proceedings of IEEE International Electric Machines and Drives Conference, IEMDC’03*, Madison, United States, 1-4 June 2003, pp. 1955-1962.
- [4]. P. Pragasen, K. Ramu, “Application Characteristics of Permanent Magnet Synchronous and Brush less dc Motors for Servo Drives, *IEEE Transaction on Industry Applications*, Vol. 21, No. 5, 1991, pp. 986-996.
- [5]. T. M. Jahns, “Torque Production in Permanent-Magnet Synchronous Motor Drives with Rectangular Current Excitation,” *IEEE Trans. Ind. Appl.*, vol. IA-20, no. 4, pp. 803–813, 1984..
- [6]. S. Z. Zhao, M. W. Iruthayarajan, S. Baskar, and P. N. Suganthan, “Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization,” *Inf. Sci. (Ny)*, vol. 181, no. 16, pp. 3323–3335, 2011.
- [7]. Z. Yusheng and X. Qun, “Compared with PID, Fuzzy and PID Fuzzy Controller,” *Paper.Edu.Cn*, no. 210098, pp. 1–5, 1942.
- [8]. M. Azizur Rahman and P. Zhou, “Analysis of brushless permanent magnet synchronous motors,” *IEEE Transactions on Industrial Electronics*, vol. 43, no. 2, pp. 256–267, 1996.
- [9]. A. Herreros, E. Baeyens, and J. R. Per’an, “Design of PID type controllers using multiobjective genetic algorithms,” *ISA Transactions*, vol. 41, no. 4, pp. 457–472, 2002.
- [10]. M. Marufuzzaman, M. B. I. Reaz, L. F. Rahman, and T. G. Chang, “High-speed current dq PI controller for vector controlled PMSM drive,” *The Scientific World Journal*, vol. 2014, Article ID 709635, 9 pages, 2014.
- [11]. M. R. Rani, H. Selamat, H. Zamzuri, and Z. Ibrahim, “Multiobjective optimization for PID controller tuning using the global ranking genetic algorithm,” *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 1, pp. 269–284, 2012.



- [12]. M. A. Sahib, B. S. Ahmed, and M. Y. Potrus, "Application of combinatorial interaction design for DC servomotor PID controller tuning," *Journal of Control Science and Engineering*, vol. 2014, Article ID 576868, 7 pages, 2014.
- [13]. I. Pan and S. Das, "Chaotic multi-objective optimization based design of fractional order PI $\lambda$ DI $\mu$  controller in AVR system," *International Journal of Electrical Power & Energy Systems*, vol. 43, no. 1, pp. 393–407, 2012.
- [14]. Q.-G. Chen, N. Wang, and S.-F. Huang, "Distribution population- based genetic algorithm for parameter optimization PID controller," *Acta Automatica Sinica*, vol. 31, no. 4, pp. 646–650, 2005.
- [15]. H. Zhang, Y. Shi, and A. S. Mehr, "Robust  $H \infty$  PID control for multivariable networked control systems with disturbance/noise attenuation," *International Journal of Robust and Nonlinear Control*, vol. 22, no. 2, pp. 183–204, 2012.
- [16]. H. Zhang, Y. Shi, and A. S. Mehr, "Robust static output feedback control and remote PID design for networked motor systems," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 12, pp. 5396–5405, 2011.
- [17]. L. Huang, N. Wang, and J.-H. Zhao, "Multiobjective optimization for controller design," *Acta Automatica Sinica*, vol. 34, no. 4, pp. 472–477, 2008.
- [18]. W. Wojsznis, A. Mehta, P. Wojsznis, D. Thiele, and T. Blevins, "Multi-objective optimization for model predictive control," *ISA Transactions*, vol. 46, no. 3, pp. 351–361, 2007.
- [19]. I. Juniku and P. Marango, "PID design with bio-inspired intelligent algorithms for high order systems," *International Journal of Mathematics and Computers in Simulation*, vol. 9, pp. 44–52, 2015.
- [20]. R. K. Sahu, S. Panda, and S. Padhan, "A hybrid firefly algorithm and pattern search technique for automatic generation control of multi area power systems," *Int. J. Electr. Power Energy Syst.*, vol. 64, pp. 9–23, 2015.
- [21]. J. Kennedy and R. C. Eberhart, "Swarm Intelligence", San Francisco, CA: Morgan Kaufmann Publishers, 2001.
- [22]. X. S. Yang, "A new metaheuristic bat-inspired algorithm". In: J. Gonzalez, R., et al. (Eds.), *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, 2010, *SCI 284*, pp. 65–74.
- [23]. X. S. Yang, "Chaos-Enhanced Firefly Algorithm with Automatic Parameter Tuning", *International Journal of Swarm Intelligence Research*, vol. 2, no. 4, pp. 1-11, 2011.
- [24]. S. K. Injeti, V. K. Thunuguntla, and M. Shareef, "Optimal allocation of capacitor banks in radial distribution systems for minimization of real power loss and maximization of network savings using bio-inspired optimization algorithms," *Int. J. Electr. Power Energy Syst.*, vol. 69, pp. 441–455, 2015.



**Satish Kumar Injeti** is presently working as Assistant Professor in EE department of National Institute of Technology Warangal (An Institute of National Importance) located in Telangana State, India. He obtained his B.Tech. Specialization in Electrical and Electronics Engineering from the J.N.T.University, Hyderabad in 2002, Masters Degree from J.N.T. University, Anantapur in 2005 and Ph.D. in Electrical and Electronics Engineering specialized in Power Systems from J.N.T.University, Kakinada. His areas of research include Distributed Generation planning, Power System Optimization, optimal capacitor placement, economic dispatch, power system stability, LFC, AGC, Small signal stability of Smart Grids and Micro Grids, control of drives, Hybrid power systems and Digital Image Processing. He may be contacted at [satishinjeti@gmail.com](mailto:satishinjeti@gmail.com)



**M. Divyavathi**, She received B.Tech. Specialization in Electrical and Electronics Engineering in 2013 from Jawaharlal Nehru Technological University Kakinada and M.Tech (Power and Industrial Drives) in 2016 from Jawaharlal Nehru Technological University Kakinada, Andhra Pradesh State, India.