Swarm Optimized Pitch Angle Controller for an Aircraft

[1]C.S. MOHANTY, [2]P.S. KHUNTIA, [3] D. MITRA
[1,2,3]Department of Electronics
[1,2]B.P.U.T, [3]IIT, Dhanbad
[1,2]Post Box No.21, Techno Park, Jatni
Khurda. Odisha, PIN-752050.
[3]Dhanbad-826004, Jharkhand
[1, 2, 3] INDIA
[1]E-mail:csmkist@gmail.com
[2]E-mail:parthsarathi_k@yahoo.com
[3]E-mail:debjani7@yahoo.com

Abstract: - The flight dynamics are non linear, time invariant and uncertain, hence the flight dynamics are linearized at some flight conditions and thus flight control systems are designed by using these linearized mathematical models. These linearized mathematical models can be controlled by using nonlinear controllers. The principal objective of this paper is to design a Swarm Optimized Proportional Integral Derivative (PID) controller for a non linear pitch control system to obtain the desired pitch angle as commanded by the pilot while manoeuvring a Delta Aircraft (four engine very large cargo jet aircraft). Here the Bacterial Foraging Optimization is applied as offline to optimize the PID controller. A fine tuned PID controller (particle Swarm Optimization based) i.e. PSOPID and a neural controller are designed to compare and establish the superiority of our proposed system. It is further established that BFOPID controller provides better performance in comparison with Radial Basis function Neural Controller (RBFNC) and PSOPID controller in terms of early settling time and overshoot.

Key-Words: PID Controller, BFOPID, PSOPID, RBFNC, Pitch Control System, Non linear controller.

1 Introduction

In general an aircraft flies in a three dimensional plane by controlling its control surfaces such as aileron, rudder and elevator. These control surfaces control and change the motions of the aircraft about the roll, pitch and yaw axes [1]. Elevators are usually at the rarest end of an aircraft which controls the orientation of an aircraft by changing the pitch and angle of attack of the aircraft [2-3].Due to nonlinear, time varying and uncertain flight dynamics the flight control systems are very complicated to design. Thus the aircraft dynamics are linearized at some flight conditions and flight control systems are designed by using this linearized mathematical model of the aircraft. These linearized mathematical models can be controlled by using non linear controllers [4-9].As the flight dynamics are non linear and uncertain, the use of classical control systems may not provide the desired stability and performance characteristics. Thus instead of using these classical controllers other controllers like PID controller [2], neural controller [10], fuzzy controller [11], H∞ controller [12], etc are used. As the aircraft is a fast acting dynamic system the pitch angle plays an important role.Various techniques have been adopted for controlling the pitch angle of an aircraft such as non linear inversion/sliding technique [8], neural controller with learning mechanism [10], convex combination of transfer matrices of the system and controller[13], Self-tuning fuzzy PID controller [11] etc. This research still remains as an open issue in present and future works. Now a day's various optimization techniques are developed using swarming and social foraging behaviour for different controllers in order to develop an optimized controller to solve variety of problems. In the year 1991-1992, M Dorigo and colleagues developed an ant based optimization known as Ant Colony Optimization for the solution of hard combinatorial optimization problem. Eberhart and Kennedy developed Particle Swarm Optimization based on swarming strategy of bird and fish [14]. Farooq et al developed a bee inspired algorithm for routing in a telecommunication network. A relatively newer evolutionary computational algorithm based on a swarming strategy of E coli bacteria was developed and used to develop adaptive controllers [15] and cooperative control strategies for autonomous vehicles [16-17].Some hybrid BFO algorithms are also designed for better optimizations of the PID controller. These hybrid algorithms such as BFO hybrid with genetic algorithm is used for efficient control of an automatic voltage regulator [18] and with Particle Swarm optimization to get the better optimized values of PID control parameters [19].

In this paper non linear control of an aircraft pitch control system using PID controller is presented. Bacterial foraging optimization is applied to offline optimized PID controller. Several performances of the proposed system are compared with a fine tuned PID controller (PSO-PID) and with a RBFN controller. The simulation result shows the superiority of the proposed system to the other systems presented in this article.

The paper is organized as follows: In section 2 we present a Problem formulation for an aircraft pitch control System. Section 3 gives a brief review of design of RBFNC .Section 4 and 5 deal with PSO and BFO algorithms for optimizing PID controller parameters respectively. Section 6 deals with design of BFO and PSO based PID controller. Section 7 deals with simulation and results. Finally conclusion and analysis are drawn in section 8.

2 Problem Formulation

The pitch angle of the aircraft is generally described by a coordinate system that is fixed to the aircraft [1]. The pitch angle and other forces acting in case of an aircraft are shown in figure 1.

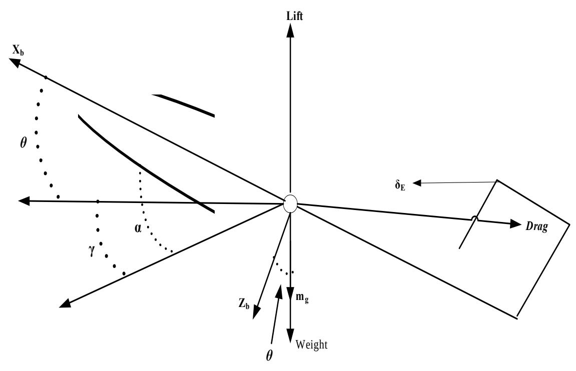

Fig.1 Description of pitch control system

Where δ_E =Deflection of the elevator, θ =Pitch angle, α = angle of attack and γ = flight path angle X_b, Y_b and Z_b represents the aerodynamics forces.

First, the aircraft is assumed to be in steady state cruising at constant altitude and velocity, thus the thrust and drag are canceled out, the lift and weight balance out each other. Second, the change in pitch angle does not change the speed of an aircraft under any circumstances. Also, the atmosphere in which the plane flies is assumed undisturbed, thus forces and moment due to atmospheric disturbance is zero. Hence longitudinal dynamics of an aircraft can be described by the following equations [20].

$$\dot{u} = X_u u + X_w + W_0 q - g cos\varphi_0 \theta \quad (1)$$
$$\dot{w} = Z_u u + Z_w w + U_0 q - g sin\varphi_0 \theta + Z_{\delta_E}\delta_E \quad (2)$$
$$\dot{q} = M_u u + M_w w + M_{\dot{w}}\dot{w} + M_q q + M_{\delta_E}\delta_E \quad (3)$$
$$\dot{\theta} = q \quad (4)$$

where X_u,X_w,W_0,Z_u, Z_w,U_0,M_u, M_w, $M_{\dot{w}}$, M_q and M_{δ_E} are called stability derivatives. u, w, q are called change in speed, change in angle of attack and change in pitch angle respectively.

The transfer function of the change in pitch angle to the change in elevator angle can be obtained from the change in pitch rates to the change in elevation angle in the following way

$$q = \dot{\theta} \quad (5)$$

Taking Laplace transform of equation 5

$$q(s) = s\theta(s) \quad (6)$$

$$\frac{\theta(s)}{\delta_E(s)} = \frac{1}{s}\frac{q(s)}{\delta_E(s)} \quad (7)$$

$$\frac{q(s)}{\delta_E(s)} = \frac{k_q(1+sT_2)}{\Delta_{sp}(s)} \quad (8)$$

where $\frac{q(s)}{\delta_E(s)}$ is the change in the pitch rate to the change in elevator deflection angle.

Further,

$$\Delta_{sp}(s) = s^2 + 2\vartheta_{sp}w_{sp}s + {w_{sp}}^2 \quad (9)$$

$$w_{sp} = \left[Z_w M_q - U_0 M_w\right]^{\frac{1}{2}} \quad (10)$$

$$k_q = Z_{\delta_E}M_w - M_{\delta_E}Z_w \quad (11)$$

$$T_2 = \frac{M_{\delta_E} + Z_{\delta_E}M_{\dot{w}}}{k_q} \quad (12)$$

The values of stability derivatives for flight condition -3 [20] of a delta aircraft is given below

$Z_w = -0.925, U_0 = 253ms^{-1}, Z_{\delta_E} = -9.51$

$Z_{\delta_{th}} = 0.05 \times 10^{-5}$

$M_w = -0.0011, M_q = -1.02, M_{\delta_E} = -1.51$

Putting the values of stability derivatives in equations (7) and (8), we get

$$\frac{q(s)}{\delta_E(s)} = \frac{1.386+1.5s}{(s^2+2.198s+1.222)} \quad (13)$$

$$\frac{\theta(s)}{\delta_E(s)} = \frac{1.386+1.5s}{s(s^2+2.198s+1.222)} \quad (14)$$

Thus equation 13 can be rewritten as

$$\dddot{\theta}(t) + \left(\frac{1}{\tau_1}+\frac{1}{\tau_2}\right)\ddot{\theta}(t) + \left(\frac{1}{\tau_1\tau_2}\right)\dot{\theta}(t) = \frac{k}{\tau_1\tau_2}\left[\tau_3\dot{\delta}_E(t) + \delta_E(t)\right] \quad (15)$$

From equations (14) and (15) we obtained the values of τ_1,τ_2,τ_3 and K are 0.8995 + 0.0968i , 0.8995 - 0.0968i, 1.0824 and -1.1346 respectively.

For larger elevation angle the model can be approximated as [15]

$$\dddot{\theta}(t) + \left(\frac{1}{\tau_1}+\frac{1}{\tau_2}\right)\ddot{\theta}(t) + \left(\frac{1}{\tau_1\tau_2}\right)H\left(\dot{\theta}(t)\right) = \frac{k}{\tau_1\tau_2}\left[\tau_3\dot{\delta}_E(t) + \delta_E(t)\right] \quad (16)$$

where $H(\dot{\theta})$ is a non linear function of $\dot{\theta}(t)$. The function $H(\dot{\theta})$ can be found from the relationship between δ and $\dot{\theta}$,which for a steady state becomes $\ddot{\theta} = \dot{\delta} = 0$. An experiment known as "spiral test" has shown that the $H(\dot{\theta})$ can be approximated by [21]

$$H\left(\dot{\theta}(t)\right) = \bar{a}\dot{\theta}^3 + \bar{b}\dot{\theta} \quad (17)$$

The values of $\bar{a}$ and $\bar{b}$ are chosen to be 1 for our simulation and the maximum deviation of elevation angle is $\pm$ 15 degrees.

2.1 Design of a digital controller for the aircraft pitch control system

Let $a = \left(\frac{1}{\tau_1}+\frac{1}{\tau_2}\right)$, b = $\left(\frac{1}{\tau_1\tau_2}\right)$, $c = \frac{k\tau_3}{\tau_1\tau_2}$ and $d = \frac{k}{\tau_1\tau_2}$

The model should be in the form given below

$$\dot{x}(t) = f\left(x(t), \delta_E(t)\right)$$

$$y(t) = g\left(x(t), \delta_E(t)\right)$$

where $\boldsymbol{x(t)} = [\boldsymbol{x_1(t)}, \boldsymbol{x_2(t)}, \boldsymbol{x_3(t)}]^T$

and $\boldsymbol{f} = [\boldsymbol{f_1}, \boldsymbol{f_2}, \boldsymbol{f_3}]^T$

Putting the above values in equation (2), we get

$$\dddot{\theta}(t) = -a\ddot{\theta}(t) - bH\left(\dot{\theta}(t)\right) + \left[c\dot{\delta}_E(t) + d\delta_E(t)\right]$$
(18)

The $\dot{x}_i$ are chosen in such a way that they depend only on x_i and δ.

Choose $\dot{x}_3(t) = \dddot{\theta}(t) - c\dot{\delta}_E(t)$

so $x_3(t) = \ddot{\theta}(t) - c\delta_E(t)$

Choose $\dot{x}_2(t) = \ddot{\theta}(t)$ thus $x_2(t) = \dot{\theta}(t)$.

Finally choose $x_1(t) = \theta$

The above assumption gives the following equations

$$\dot{x}_1(t) = x_2(t) = f_1\left(x(t), \delta_E(t)\right)$$

$$\dot{x}_2(t) = x_3(t) + c\delta_E(t) = f_2\left(x(t), \delta_E(t)\right)$$

$$\dot{x}_3(t) = -a\ddot{\theta}(t) - bH\left(\dot{\theta}(t)\right) + d\delta_E(t)$$

But $\ddot{\theta}(t) = \dot{x}_3(t) + c\dot{\delta}_E(t), x_2(t) = \dot{\theta}(t)$ and $H(x_2) = \bar{a}\dot{\theta}^3 + \bar{b}\dot{\theta}$
$= \bar{a}x_2^3(t) + \bar{b}x_2(t)$

Thus $\dot{x}_3(t) = -a[x_3(t) + c\delta_E(t)] - b[x_2^3(t) + x_2(t)] + d\delta_E(t) = f_3\left(x(t), \delta_E(t)\right) \quad (19)$

Also we have $\theta = g\left(x, \theta_{ref}\right) = x_1$

In this paper, for simulation purpose the above controller is used for all pitch control systems. The values of τ_1,τ_2,τ_3 and K are 0.8995 + 0.0968i , 0.8995 - 0.0968i, 1.0824 and -1.1346 respectively for a Delta aircraft (flight condition 3)[20].

2.2 Design of a discrete time PID controller for aircraft pitch control system

The Proportional-Integral-Derivative (PID) controller is used for its simplicity. After installation of PID controller the user has to adjust the three parameters i.e. K_p, K_i *and* K_d. Here our proposed BFO and PSO based PID controllers adjust the control parameters (K_p, K_i *and* K_d) to obtain desired pitch angle. Backward difference approximation method is used to discretize the PID controller.

A PID controller in time domain can be defined as [22].

$$u(t) = K_p + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{d}{dt} e(t)$$
(20)

The above equation can be converted to S domain by taking the Laplace transform.

Taking Laplace transform of equation (20) we get

$$G(s) = K_p + \frac{K_i}{s} + sK_d$$
(21)

Again to implement the PID controller in digital domain the above controller should be converted to discrete with the help of Z transform.

Taking Z Transform of equation (21) we get

$$U(z) = \left[K_p + \frac{K_i}{1-Z^{-1}} + K_d(1 - Z^{-1})\right] E(Z)$$
(22)

Thus equation (22) with rearrangement becomes,

$$U(z) = \left[\frac{(K_p+K_i+K_d)+(-K_p-2K_d)Z^{-1}+K_d Z^{-2}}{1-Z^{-1}}\right] E(Z)$$
(23)

Let $K_1 = (K_p + K_i + K_d)$
$K_2 = -K_p - 2K_d$
$K_3 = K_d$
So equation (23) becomes

$$U(z) = \left[\frac{K_1+K_2 Z^{-1}+K_3 Z^{-2}}{1-Z^{-1}}\right] E(Z) \quad (24)$$

$$U(z) - Z^{-1}U(Z) = [K_1 + K_2 Z^{-1} + K_3 Z^{-2}]E(Z)$$
(25)

The equation (25) is again converted back to a difference equation as

$$u[k] = u[k-1] + k_1 e(k) + k_2 e(k-1) + k_3 e(k-2)$$
(26)

The above equation is the mathematical representation of PID controller in discrete domain which is used in the present work.

3 Radial Basis Function Neural Controller

Radial Basic Function Neural Network is an artificial neural network that uses radial basis functions i.e. Gaussian functions. Radial basis neural networks are feed forward networks containing hidden nodes as radial basis functions. In these types of networks first the weight from input to hidden nodes are determined and then the weights from hidden to output nodes are determined. The training /learning of these networks are very fast and are very good at interpolation [23]. Radial Basis Function Neural Network with single neuron output y is presented in figure-2 which consists of three-layers. The output can be represented as $\boldsymbol{y} = F_{rbf}(\boldsymbol{x})$ where $x = [x_1, x_2, x_3 \ldots\ldots x_n]^T$ is the input and $R_i(x)$ is the output of the i^{th} receptive field with strength denoted by b_i.

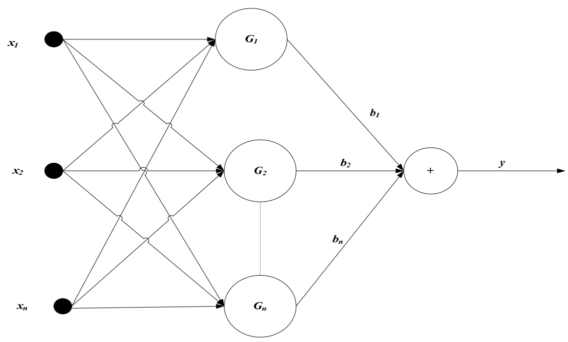

Fig. 2 Radial Basic Function Neural Network

Assuming n_R receptive fields present in the 2RBFNC, the output y can be written as

$$\boldsymbol{y} = F_{rbf}(\boldsymbol{x}, \boldsymbol{\varphi}) = \sum_{i=1}^{n_R} b_i \boldsymbol{R_i}(\boldsymbol{x}) \quad (27)$$

Where φ holds the parameters of the receptive field units which consist of the parameters b_i and possibly the parameters of the $R_i(x)$. The Gaussian-shaped functions are preferred for analytical convenience i.e.

$$\boldsymbol{R}_i(\boldsymbol{x}) = exp\left[\frac{|\boldsymbol{x}-\boldsymbol{C}^i|^2}{\boldsymbol{\sigma}^{i^2}}\right] \quad (28)$$

where $\boldsymbol{C_i} = [C_1^i, C_2^i \ldots\ldots\ldots\ldots C_n^i]^T$ parameterize the locations and σ decides the spreading of the receptive fields in the input space. The weighted average output of the RBF neural network can be written as

$$\boldsymbol{y} = F_{rbf}\,(x, \varphi) = \frac{\sum_{i=1}^{n_R} b_i \boldsymbol{R_i(x)}}{\sum_{i=1}^{n_R} \boldsymbol{R_i(x)}} \quad (29)$$

3.1 Design of Radial Basis Function Neural Controller for Aircraft Pitch control System

In this section we modify the Pitch control system which is designed baseing on RBFNC. Here the Pitch control system utilizes a RBFN controller. The RBFNC for aircraft pitch control system is shown in figure 3.

The error $e[k]$ and derivative of that error ($\dot{e}[k]$) are the inputs to the RBFNC expressed as

$$e[k] = \theta_{ref}[k] - \theta[k] \quad (30)$$

and

$$\dot{e}[k] = \dot{\theta}_{ref}[k] - \dot{\theta}[k] \quad (31)$$

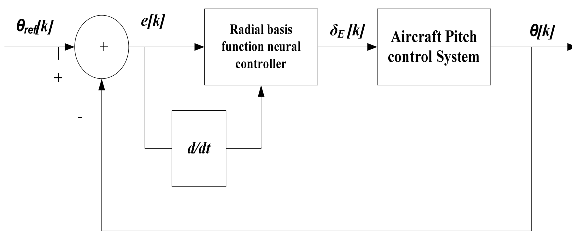

Fig.3 RBFNC for an aircraft pitch control system

Using a backward difference approximation

$$\dot{e} \approx \frac{e(kT) - e(kT-T)}{T} = c(kT) \quad (32)$$

where

k = time step size.

T=Sampling period (controller).

From the figure 3 it is clear that the output of RBFN Controller for Pitch control system can be defined as [14]

$$\delta[k] = F_{rbf}(e[k], c[k]) \quad (33)$$

where $c[k] = \frac{d}{dk} e[k]$

4 Particle Swarm Optimization

Kennedy and Eberhert in 1995 introduced a new technique based on social-psychological theory of fish schooling and bird flocking [14]. Here the intelligent swarms (particles) are moved in a two dimension search space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position (*Pbest)* and also guided towards the global best known positions (*gbest)*, which are updated as better positions found by other particles. The PSO concept consists of changing the velocity of each particle toward its *pbest* and *gbest* locations. The PSO has a good computational efficiency with stable convergence characteristic. The position of each particle in space is calculated using simple mathematical formulae shown below

$$v_{id}^{t+1} = w.v_{id}^{t} + c_1.\vartheta_1\left(P_{id}^{t} - x_{id}^{t}\right) + c_2.\vartheta_2.\left(P_{gd}^{t} - x_{id}^{t}\right) \quad (34)$$

Where $x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1}$ (35)

v_i^{t+1}: Component in dimension d of the i^{th} particle velocity in iteration *t*

x_{id}^{t}: Component in dimension d of the i^{th} particle position in iteration *t*

c_1, c_2 : Constant weight factors.

ϑ_1, ϑ_2: Random factors in the [0, 1] interval.

P_i: Best position achieved so long by particle i

P_g: Best position found by the neighbors of particle i

w: Inertia weight

The PSO algorithm requires tuning of some parameters such as individual, sociality weights c_1, c_2 and the inertia factor (w).

4.1 Algorithm

Step [1] Initialize the particles

Step [2] Calculate the fitness value of each particle.

Step [3] Compare the current fitness with the best previous fitness.

- If the current fitness better than previous best fitness then current fitness= *pbest*
- Else previous best fitness = *pbest*.

Step [4] Calculate the best of *pbest* and denote it as g*best*.

Step [5] Calculate the velocity of each particle using equation (20).

Step [6] Then the position of each particle can be updated by using equation (21).

Step [7] Maximum iteration reach yes the process is stop otherwise go to step 2.

5 Bacterial Foraging Optimization

The E-coli bacteria consists of a plasma membrane, cell wall and a capsule. The cell is about $1\mu m$ diameter and 2 μm in length and the weight is about 1 picogram. The E-coli bacterium has a control system that enables it to search for food and avoid noxious environments [16]. The control mechanism of E-coli bacteria is described by following 4 steps.

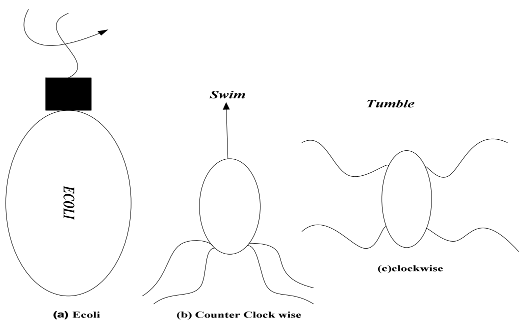

Fig. 4 E coli bacteria while it swimming and tumbling

5.1 Swimming and Tumbling

The E-coli bacterium has set of rigid flagella that enable it to swim. It can move in two different ways: swim and tumble. The flagellum can rotate clock wise and counter clock wise as shown in figure 4. If the flagellum rotates clockwise then the bacteria tumble and if it is anticlockwise then the bacteria swim. The Bacteria can swim up to a maximum no of N_s steps.

5.2 Chemotaxis

The chemotaxis step is the combination of swimming and tumbling for example if an E-coli bacterium is in some substance that is neutral then the flagella simultaneously alternates between counter clockwise and clockwise, thus the bacteria will alternately tumble and swim. The maximum no of swim with in a chemotaxis is N_s and when the swim steps are stop the tumble action takes place.

5.3 Reproduction

After N_c chemotaxis step a reproduction step is taken. Let N_{re} be the no of reproduction steps. If S is the total no of bacterium and S_r be the No of bacterium having sufficient nutrients so that they will reproduce with no mutation.

$$\text{Let } S_r = \frac{S}{2} \tag{23}$$

The S_r poor healthy bacteria will die and other S_r healthy bacteria will survive. Each healthy bacterium will then split into two.

5.4 Elimination and dispersal

Elimination and dispersal event arises if all the bacteria in a region are killed or some of the bacteria are dispersed into a new environment. Due to elimination and dispersion the chemotaxis process is destroyed but this process also assists the chemotaxis process like it placed the bacteria near good food source.

5.5 Algorithm

1. Initialized the parameter $p, S, N_c, N_s, N_{re}, N_{ed}, P_{ed}$.

Where

p = Dimension of search space

S = No. of Bacteria

N_c= No. of chemotactic steps taken by each bacteria

N_s=No of stepsize

N_{re}= No of reproduction steps

N_{ed}= No. of elimination and dispersal events

P_{ed} = Probability of elimination and dispersion

2. Elimination and dispersal loop *l=l+1*

2.1 Reproduction loop *k=k+1*

2.2 Chemotaxis loop *j=j+1*

5.5.1 Chemotaxis

I. For *i=1, 2, 3 …S* take a chemotaxis step for bacteria *i*

II. Compute $J(i,j,k,l)$ Let $J(i,j,k,l) = J(i,j,k,l) + J_{CC}\left(\varphi^i(j,k,l), P(j,k,l)\right)$ (add on the cell to cell attractant effect to the nutrient concentration).

III. Let $J_{Last} = J(i,j,k,l)$ to save this value for finding better value via tumble.

5.5.2 Tumble

Generate a random vector ($\Delta(i) \epsilon R^p$) between [-1,1] with each element $\Delta_m(i)$,where *m=1,2,3………P*.

Move:

I. Let $\varphi^i(j1,k,l)\ \varphi^i(j,k,l)\ C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$

This results in a movement of stepsize $C(i)$ in the direction of tumble.

II. Compute $J(i,j+1,k,l)$ (New Cost function).

Swim

I. Let m=0 (Counter for swim length).

II. while $m < N_s$ (if have not climbed down too long).

III. Let *m=m+1*

IV. If $J_{Last} = J(i,j+1,k,l) < J_{Last}$ (if doing better), let $J_{Last} = J(i,j+1,k,l)$.

V. $\varphi^i(j+1,k,l) = \varphi^i(j,k,l)\ C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$ and use this $(j+1,k,l)$ to compute the new $J(i,j+1,k,l)$.

VI. Else let $m = N_s$. (End of while statement).

VII. Go to the next bacterium (*i+1*). if $i \neq S$ go to next bacterium.

VIII. if $J < N_c$ go to step 2.2.

5.5.3 Reproduction

I. For a given k and l and for each *i=1, 2, 3 …S*, let $J^i_{health} = \sum_{j=1}^{N_C+1} J(i,j,k,l)$ be the health of bacterium *i*.

II. The *S*, bacteria with the highest J_{health} values die and the other S_r Bacteria with the best values split.

III. If $R < N_{re}$, go to step 2.1.

5.5.4 Elimination and dispersal

I. For *i=1, 2, 3……S* with probability P_{ed} eliminate and disperse each bacterium keeping population constant.

II. If $l < N_{ed}$, then go to step 2; otherwise end.

6 Design of BFO /PSO Based PID Controller for Aircraft Pitch Control System

The closed loop PID controller with aircraft pitch control system is shown in figure 5.

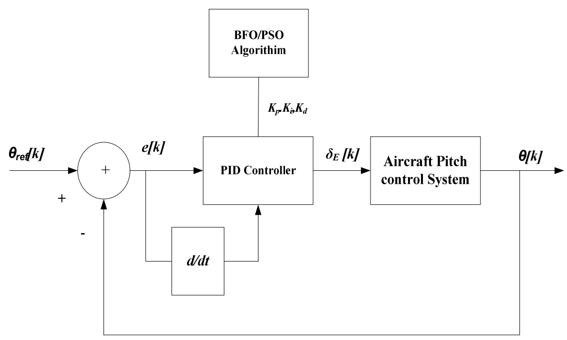

Fig.5 BFO/PSO based PID controller for Pitch control of an Aircraft.

The error e used as input to the PID controller is

$$e[k] = \theta_{ref}[k] - \theta[k] \quad (36)$$

$$\delta_E(k) = F_{swarm}\big[e[k]\big] \quad (37)$$

It is decided in the designing of pitch controller that elevator not to exceed more than ± 15 degrees in either upward or downward direction or the change of error not to be more than 0.01 radian/sec. It concludes range of error e as $e \epsilon \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$.The sampling time is 10 sec. so that a new plant input is calculated every 10 sec and applied to the elevator. The controller output is obtained after optimizing the mean square error using BFO and PSO techniques at every iteration of the simulation. Now the controller output becomes the input to the plant producing a particular value of pitch angle θ.This is repeated till the mean square error reduces to zero offering the desired pitch angle as commanded by the pilot.

7 Results and Analysis

The No. of Bacteria (S), dimension of search space (P), no. of chemotactic steps (N_c), no of reproduction steps (N_{re}), no. of elimination and dispersal events (N_{ed}) and probability of elimination and dispersion (P_{ed}) are taken as 30, 3,4,4,2 and 0.25 respectively for BFO simulation. The optimized PID parameters K_p, K_i and K_d are obtained here as 1.3285, 0.0084 and 0.585 respectively. The mean square error is found to be 0.7346. In case of PSO the value of parameters for no. of birds, bird steps, dimension of the problem, parameter C_1, Parameter C_2, and inertia weight *(w)* are taken as 50, 50, 3, 1.23, 3 and 0.9 respectively. The optimized PID parameters K_p, K_i and K_d are obtained as 1.3021, 0.9141, 4.554 and mean square error is found to be 1.0688 for PSO technique. In case of RBFNC the receptive field unit (n_R), no of inputs (n) are taken as 121 and 2 respectively. The mean square error is found here to be 2.938. The step responses of the closed loop pitch control system for all techniques are plotted and shown in the figure 9. The corresponding mean square error plots are plotted in figure 10.

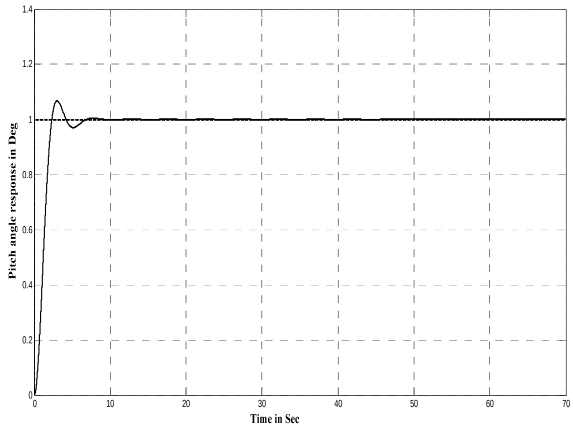

Fig.6 The step response between actual pitch angle and desired pitch angle using BFOPID

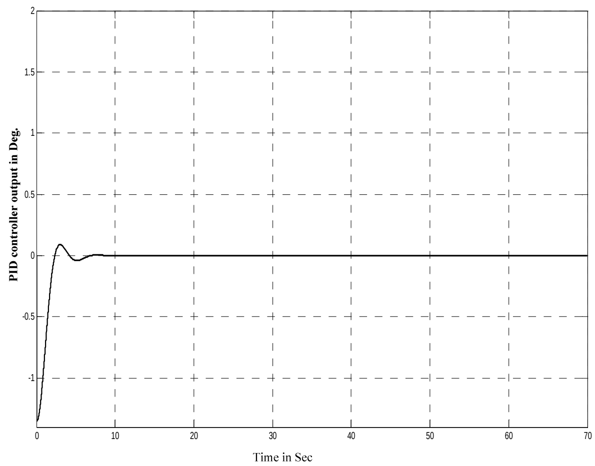

Fig.7 The output of PID controller (degree)

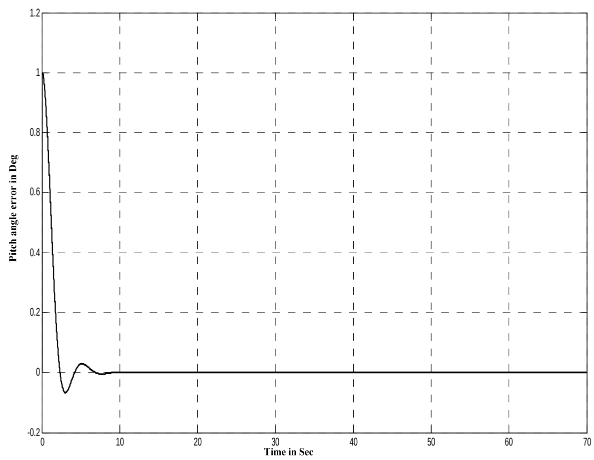

Fig.8 The pitch angle error of BFOPID Pitch control system

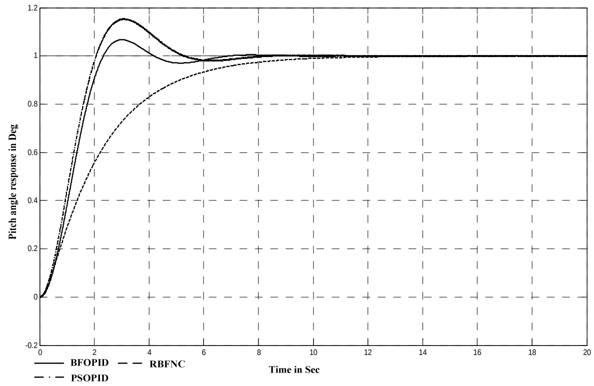

Fig.9 Comparison of pitch angle responses of BFOPID, PSOPID, and RBFNC

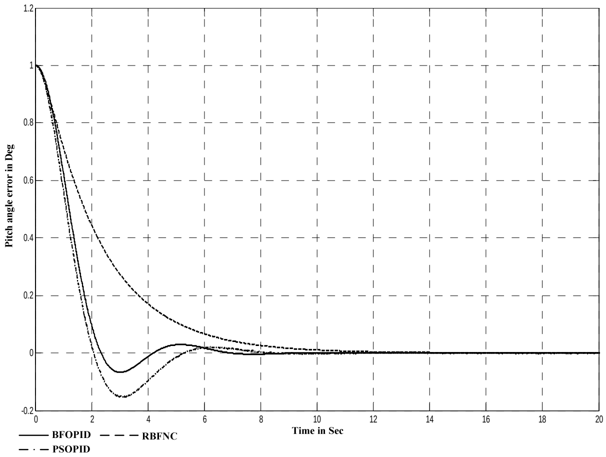

Fig.10 Comparison of error using BFOPID, PSOPID, and RBFNC

Table 1 Comparison of Time Domain Performance Specifications

Techniques	*MSE*	*Tr(sec)*	*Ts(sec)*	*OS (%)*
BFO	**0.7346**	**2.166**	**6.611**	**6.7**
PSO	**1.068**	**1.99**	**8.914**	**15.4**
RBFNC	**2.938**	**7.004**	**11.16**	**0**

8 Conclusion

The comparative analysis for all methods are discussed and shown in the Table 1. The overshoot is much less in case of BFOPID controller compared to PSOPID technique. In case of RBFNC the over shoot is zero but the settling time is very high. As aircraft is a fast acting dynamic system the settling time always plays a crucial role in case of takeoff and landing. BFOPID controller offers early settling time amongst all the techniques. In case of rise time BFOPID and PSOPID controllers exhibit better performance than RBFNC. The MSE is also less in BFOPID controller compared to all methods. For maneuvering the aircraft BFOPID pitch control system is a better system among the three systems offering the desired pitch angle as required by the pilot.

References:

[1] R.C. Nelson, *Flight Stability and Automatic Control*, McGraw Hill, Second Edition,1998

[2] M. Myint, H.K. Oo, Z.M. Naing and Y.M Myint, PID Controller for Stability of Piper Cherokee's Pitch Displacement using MATLAB, *International Conference on Sustainable Development: Issues and prospects for the GMS*, China, 2008.

[3] M. H. Sadraey *Aircraft Design: A Systems Engineering Approach*, Wiley, 2012.

[4] M. Asif Khan and Ping Lu. New technique for nonlinear control of aircraft, *Journal of Guidance Control and Dynamics*, Vol. 17, No. 5, pp. 1055-1060,1994

[5] L. R. Hunt, G. Meyer and R. Su, Nonlinear control of aircraft, *Lecture Notes in Control and Information Sciences Mathematical Theory of Networks and Systems*, Vol. 58, 1984, pp. 463-472.

[6] J. Wang and N. Sundararajan, A nonlinear flight controller design for aircraft, *Control Engineering Practice*, Vol. 3, No.6, pp. 813–825, 1995.

[7] X. Wang, R. Xie and Yan Li, Nonlinear Controller Design for a Super maneuverable Aircraft, *International Joint Conference on Computational Sciences and Optimization*, CSO 2009, Vol. 1, pp. 318-322, 24-26 April 2009.

[8] J. Karl Hedrick and S. Gopalswamy , Nonlinear flight control design via sliding methods,

Journal of Guidance, Control, and Dynamics, Vol. 13, No. 5 , pp. 850-858,1990.

[9] M. Zugaj and J.Narkiewicz, Autopilot Supported by Nonlinear Model Following Reconfigurable Flight Control System, *J. Aerosp. Eng*, Vol.23, No.4, pp.339–347, 2010.

[10] P.S Khuntia and D.Mitra, Radial Basic Function Neural Controller for Pitch Control of an Aircraft, *Georgian Electronic Scientific Journal: Computer Science and Telecommunications*, Vol.19, No.2, 2009.

[11] N. Wahid and N. Hassan, Self-Tuning Fuzzy PID Controller Design for Aircraft Pitch Control,*Third International Conference on Intelligent Systems, Modeling and Simulation (ISMS)*,pp. 19 – 24, 8-10 Feb,2012.

[12] R.A. Nichols, R.T. Reichert and W.J Rugh, Gain scheduling for H-infinity controllers: a flight control example, *IEEE Transactions on Control Systems Technology*, vol.-pp. 69-79, No.2, 1993.

[13] P.S Khuntia and D.Mitra, "Optimally Convex Controller and Model Reduction for Dynamic System", Electrotechnics, Electronics, Automatic Control, Informatic The Annals Of "Dunarea De Jos" University Of Galati Fascicle III, Vol.31, No.1,pp-35-42,2002.

[14] J. Kennedy and R. C. Eberhart, Particle Swarm Optimization, *IEEE Int. Conf. on Neural Networks*. Piscataway, NJ: IEEE Service Center, pp.1942–1948, 1995.

[15] K.M. Passino, *Biomimicry for Optimization, Control, and Automation*, London, UK: Springer-Verlag, 1990.

[16] K.M.Passino, Bacterial Foraging Optimiz-ation, *Int. J. Swarm Intelligence Research*, Vol.1, No. 1, pp. 1-16, 2010.

[17] K.M.Passino ,Biomimicry of Bacterial Foraging for Distributed Optimization and Control, *IEEE Control Systems Magazine*, Vol. 22, No. 3, pp. 52-67, 2002.

[18] D. H. Kim, A. Abraham, and J. H. Cho, A hybrid Genetic Algorithm and Bacterial Foraging Approach for Global Optimization, *Inf. Sci*, vol. 177, no. 18, pp. 3918–3937, 2007.

[19] W.M. Korani, H.T. Dorrah, and H.M. Emara, Bacterial Foraging Oriented by Particle Swarm Optimization Strategy for PID Tuning, *IEEE International Symposium on Computational Intelli-gence in Robotics and Automation (CIRA*), Daejeon, 2009, pp.445 – 450.

[20] D.McLean, *Automatic Flight Control System*, Frome: Prentice Hall, 1990.

[21] M.Bech and L.Smitt, Analogues simulation of ship maneuvers, *Technical report*, Hydro-og Aerod-ynmisk Laboratorium, Lyngby, Denmark,1969.

[22] K.Ogata, *Modern Control Systems*, 5th ed. USA: PHI, 2010.

[23] D. S. Broomhead and D. Lowe, Radial basis functions, *multi-variable functional interp-olation and adaptive networks (Technical report)*. RSRE. 4148,1988.