Nature Inspired Based Pitch Controller Design of an UAV

Chakravarthi Jada[a], Kiran Paul[b], S.N. Omkar[a] and Santosh Kumar Tutika[c]
[a]Department of Aerospace Engineering, Indian Institute of Science, Bangalore-560012, INDIA.
[b]Department of Computer Engineering, National Institute of Technology, Mangalore-575025, INDIA.
[c]Department of Civil engineering, Indian Institute of Science, Bangalore-560012, INDIA.

Abstract
In this paper a longitudinal control system has been designed for a 2 Kg payload Unmanned Aerial Vehicle using classical *PID* controller. To overcome the difficulty in tuning the *PID* gains to the required values, different nature inspired techniques have been used, for the Integral Squared Error (*ISE*) parameter optimization on UAV dynamic data at a specified trim condition. For this purpose, the penalty function that has been chosen includes time domain specifications as constraints. Finally all the results are compared for optimum *PID* parameters.

Keywords: UAV; Longitudinal Flight Dynamics; Root locus; GSO; GA; PSO.

1. INTRODUCTION
With the advent of autonomous highly maneuverable Unmanned Aerial Vehicles various applications like mapping, aerial reconnaissance, disaster management in hazardous environments; surveillance etc. have been decreasing the unbearable workload for pilots. The main purpose of autonomous nature is to maintain a specified heading, attitude and speed. The autopilot will do this task by automatically adjusting the control surfaces, which is done by continuously measuring aircraft state to the desired state. Thus for an autopilot design of an UAV it is necessary to do dynamic modeling. In literature there is quite a lot of work done in modeling of aircraft. Peddle [1] has designed and tested the autonomous flight for a small model UAV with inner and outer loops. Jodeh et al [2] developed an aerodynamic model 6-DOF simulation for SIG Rascal 110 RC aircraft. Navabalachandran et al [3] has done the modeling of flying wing UAV and designed the controller for pitch and roll attitudes by using an optimization technique in tuning the *PID* gains. Jin et al [4] designed controller for a 60 cm sized mini UAV for longitudinal and lateral motions and real time testing was done. The computer aerodynamic software called Athena Vortex Lattice (AVL) [5] is used to generate the control and stability derivative data. Cook [6] and Nagarath et al [7] discussed the control system design with classical *PID* controller for an aircraft. Myint et al [8] in their paper improved the stability of the aircraft for pitch attitude controller by using the *PID* controller for the specified time domain specifications such as rise time, overshoot and settling time.

Ziegler and Nicholes [9, 10] both proposed some empirical methods for finding the initial guess values for *PID* parameters. Chaudhuri et al [11] and Branimir Stojilikori et al [12] designed the pitch attitude controller by taking the time domain specifications as constraints. Turkolu [13] designed *PID* pitch controller by optimizing the Integral Squared Error (ISE) using KKT conditions. For optimizing the actual functions there are many methods are available. Nature inspired techniques are used frequently for optimization. Daniel [14] designed a method to use genetic algorithm to tune the *PID* controller. Eberhart and Kennedy [15] developed the Particle Swarn Optimiztion algorithm for function optimization. Mohammed El-Said El-Telbany [16] proposed a method to use PSO for *PID* tuning. Ghose and Krishnanad [17] developed an algorithm called Glowworm Swarm Optimization (GSO) for finding the global as well as multiple optimal for multimodal functions.

2. LONGITUDINAL DYNAMIC MODEL OF UAV

A 2 kg Payload Unmanned Aerial Vehicle was designed and developed at the Indian Institute of Science. Dynamic model of the aircraft is necessary for autopilot design purpose. Thus the longitudinal dynamic equation of motion after linearization and decoupling for an aircraft has been taken from [18]. The specifications of 2 Kg Payload UAV is given in table-1.

Equations of motion are in linearized about a trim condition. The trim conditions are shown in table-2.

For generating the stability and control derivatives the AVL software has been used for the above mentioned trim condition. Figure 1 shows the photo graph of the UAV and figure 2 shows the 3-D mesh view of UAV extracted from AVL. Table 3 shows the longitudinal dimensional derivatives calculated as per [18, 19].

Table 1. Specifications of 2 Kg Payload UAV

Mass	–	5.3 Kg
Wing span	–	1.84 m
Wing airfoil	–	Clark-Y
I_{xx}	–	0.069015 Kg-m^2
I_{yy}	–	1.33459 Kg-m^2
I_{zz}	–	1.43147 Kg-m^2
I_{xz}	–	0.03573 Kg-m^2
Wing chord	–	0.29 m

Table 2. Trim condition parameters

Mach	–	0.05779
α	–	9.22933°
β	–	0°
Air velocity	–	15 m/sec
p	–	0 °/sec
q	–	0 °/sec
r	–	0 °/sec
Aileron	–	0°
Elevator	–	8.3195 °/sec
Rudder	–	0°

Figure 1. Photograph of UAV.

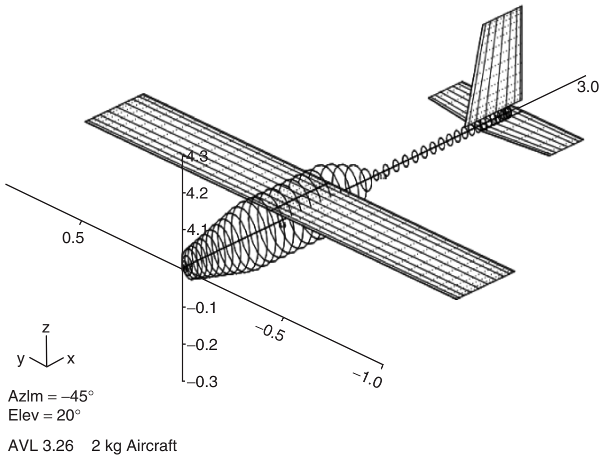

Figure 2. 3-D mesh view extracted from AVL software.

Table 3. Longitudinal dimensional derivatives

State variable	$X_{(\cdot)}$	$Z_{(\cdot)}$	$M_{(\cdot)}$
u	–0.7364	–3.5276	–0.1224
w	5.6481	–24.0948	–1.5532
q	0.0000	–5.8374	–3.9805
w	0.0000	–0.0021	–0.0024
δ_e	0.0429	–0.6795	–0.7017
δ_T	0.0000	0.0000	0.0000

The longitudinal matrices are calculated as per expressions in [18] shown below.

$$A_L = \begin{bmatrix} -0.1387 & 1.0637 & 0 & -9.8 \\ -0.6641 & -4.5358 & 13.8951 & 0 \\ -0.1169 & -1.4924 & -3.8836 & 0 \\ 0.000 & 0.000 & 1.000 & 0.000 \end{bmatrix}$$

$$B_L = \begin{bmatrix} 0.0081 & 0.000 \\ -0.1279 & 0.000 \\ -0.6787 & 0.000 \\ 0.000 & 0.000 \end{bmatrix}$$

$$C_L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D_L = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Using the data above for a given elevator input the pitch attitude transfer function is found as

$$\frac{\theta(s)}{\delta_e(s)} = \frac{(s+4.0843)(s+0.3103)}{\left(s^2+8.3588s+38.4434\right)\left(s^2+0.1994s+0.11744\right)} \tag{1}$$

The servo transfer function [20] is given as

$$\frac{\delta_e(s)}{u_{\delta_e}(s)} = \frac{6.7}{(s+9.5)} \tag{2}$$

3. PID CONTROL SYSTEM DESIGN

The role of the controllers in the automatic control system is to make the aircraft follow commanded motion without any pilot intervention. The basic approach for this requirement is to use feedback control which reduces the sensitivity of the system to modeling uncertainties and provides disturbance rejection to those quantities that detract from the desired aircraft motion. The basic control in aircraft is to control its attitude also called *displacement controller* or *displacement autopilot*. For example pitch displacement autopilot moves the aircraft to the desired pitch angle. The block diagram for pitch attitude control is shown in figure 3. In this paper the classical *PID* controller has been used. Analytical form of the *PID* is given by:

$$PID(s) = K_P + K_I/s + K_D s \tag{3}$$

Due to their simplicity in *PID* controllers are used in more than 95% of closed-loop industrial processes. The basic criteria in the *PID* controller design is to tune their gains such that the system reaches its desired performance specifications such as rise time, settling time, overshoot etc [7]. According to Nagarath [7] K_P can be used to decrease rise time, K_D to reduce the overshoot and settling time and K_I to eliminate the steady state error. But the question is what the starting values for the *PID* parameters are. For this requirement Ziegler and Nicholes [10] has proposed an empirical method explained in the next sections.

Before designing any control system for a given plant state space system it is necessary check whether the given system is controllable or not [7]. For the given state space system to be controllable, the rank of controllability matrix (C_t) must be equivalent to the order of system matrix A_L.

$$\text{Controllability matrix } C_t = [B\ AB\ A^2B\ A^3B] \tag{4}$$

$$\text{Rank } (C_t) = 4 = \text{order of matrix } A_L.$$

Therefore the UAV is controllable in longitudinal motion.

From the above controllability check analysis, it has been proved that the longitudinal UAV system is controllable, which enables the possibility to design a controller.

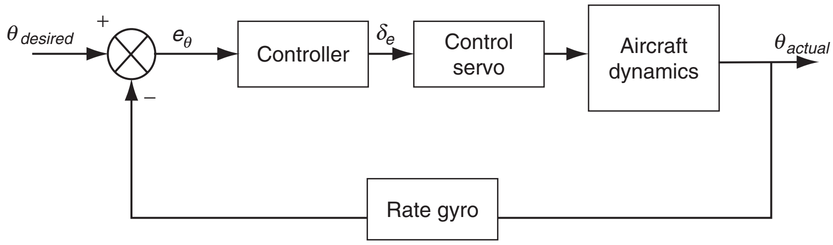

Figure 3. Block diagram for pitch attitude control.

3.1. Ziegler-nicholes tuning method

The Ziegler-Nichols method [9, 10] is based on the plant step response. This tuning method contains two methods, which depends upon the open loop step response. Without going into details the current plant dynamics comes under the second method of Ziegler-Nichols. The second method targets plants that can be rendered unstable under proportional control. The steps for tuning a *PID* controller are:

Step 1. Reduce the integrator and derivative gains to zero.

Step 2. Increase the K_P from zero to some critical value $K_P = K_{cr}$ at which sustained oscillations occur. If it doesn't occur then another method i.e first method [9] has to be applied.

Step 3. Note the value of K_{cr} and the corresponding period of sustained oscillation P_{cr}.

Now the controller gains are calculated by using the formulae given by:

$$K_P = 0.6\,K_{cr} \tag{5}$$

$$K_I = \frac{2K_P}{P_{cr}} \tag{6}$$

$$K_D = \left(\frac{K_P}{8}\right)P_{cr} \tag{7}$$

Now overall transfer function is

$$\frac{\theta(s)}{u_{\delta_e}(s)} = \frac{\theta(s)\delta_e(s)}{\delta_e(s)u_{\delta_e}(s)} = \frac{4.5473s^2 + 19.98s + 5.764}{s^5 + 18.058s^4 + 121.52s^3 + 390.8s^2 + 86.64s + 49.9134} \tag{8}$$

K_{cr} is found by using the root locus method. When root locus of transfer function in Eqn (8) is giving sustained oscillation it means, it crosses the imaginary axis at that gain value (K_{cr}).

From figure 4 it is seen that the K is between 123 and 312 when crossing the imaginary axis. Now by varying the values between 123 and 312 and using a proportional control feedback for the system it was found that sustained oscillations were obtained at K = 179.175 also called critical gain. It gives K_{cr} = 179.175 and P_{cr} = 0.774.

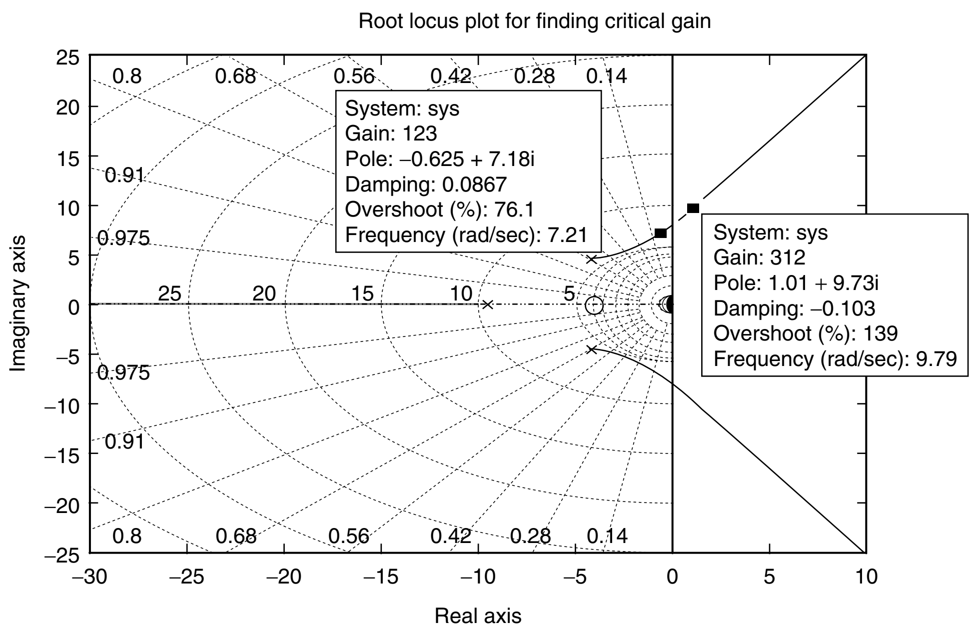

Figure 4. Root locus plot.

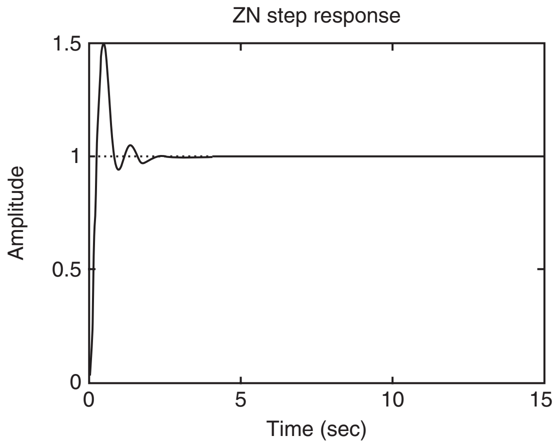

Figure 5. Step response of ZN method.

Table 4. Time specifications

Parameter	Value
Rise Time	0.1851
Settling Time	2.0215
Overshoot	**49.1098**

Now by using the equation (5), (6) and (7) :

$$K_P = 107.505$$

$$K_I = 277.790$$

$$K_D = 10.401$$

The step response is given in figure 5 and table 4 shows the time domain specifications.

From this it is clearly seen that the overshoot is exceeding the criteria required. Hence other methods for tuning the *PID* controller are explored, keeping the bounds of the constants in the *PID* controller in the range of values derived from the ZN method.

3.2. Error optimization criteria

We know that the original transfer function of pitch attitude from the Eqn (1). Dealing with transfer functions of higher order is not easy, hence the original transfer function of higher order has to be reduced into some lower order and it has to behave similar to that of the original system. This transfer function is also called as *Reduced system transfer function*. There are many methods available in literature for reducing the transfer functions. Here a method based on routh approximation [20] is used. After reduction the transfer function becomes

$$\frac{\theta(s)}{\delta_e(s)} = \frac{0.08342s + 0.02406}{s^2 + 0.2418s + 0.1263} \tag{9}$$

Figure 6 shows the step response of original as well as reduced order system. From the figure it is observed that the reduced order system is following the original order system approximately. Hence the reduced order system transfer function can be used for the controller design which ensures all its controller properties work similar to that of the original system.

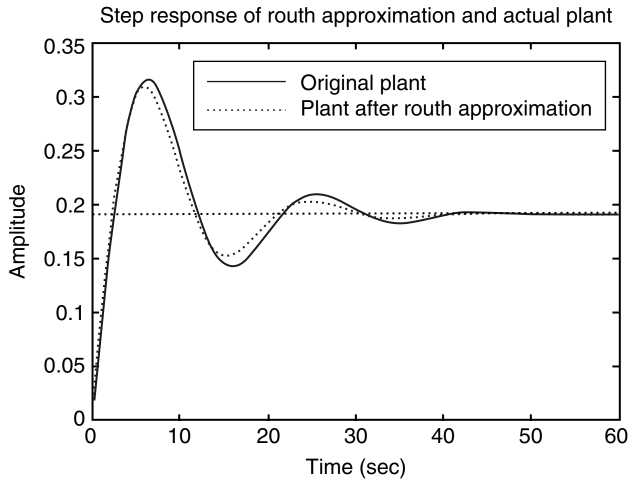

Figure 6. Plot for step response of original as well as reduced order T.F.

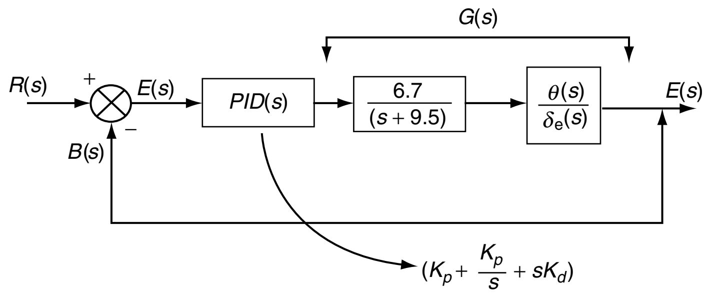

Figure 7. Closed loop control system structure.

Figure 7 shows the closed loop control system structure for the given system.
The T.F for the desired unit step is given as

$$\frac{E(s)}{R(s)} = \frac{1}{1 + G(s)PID(s)} \tag{10}$$

If the error is reduced to zero it will assure that the actual signal equals the desired signal. Any criterion used to measure the quality of the system response must take into account the variation of $E(s)$ over the whole time range. Basically there are four criteria which are in use - Integral of Absolute Error (IAE), Integral of Squared Error (*ISE*), Integral of Time multiplied by Absolute Error (ITAE) and Integral of Time multiplied by Squared Error (ITSE). Here the *ISE* criterion is chosen because the absolute value of an error function is not generally analytic in form. The ITAE and ITSE have an additional time multiplies of the error function, which emphasis long-duration errors, and therefore these criteria are most often applied in systems requiring a fast settling time. The *ISE* parameter optimization method i.e. Performance Index (*PI*) is given by

$$(PI)_{ISE} = \int_0^\infty \{e(t)\}^2 dt \tag{11}$$

The $E(s)$ expression is in Laplace domain but our *PI* is in time domain. If *PI* expressed in s-domain calculation required will be simplified greatly. In this course Parsevel [13] proposed a theorem in which the integral is defined as

$$(PI)_{ISE} = \frac{1}{2\pi i}\int_{-i\infty}^{+i\infty} E_1(s)E_2(s)dt \tag{12}$$

$$(PI)_{ISE} = \int_0^{\infty}\{e(t)\}^2 dt = \frac{1}{2\pi i}\int_{-i\infty}^{+i\infty} E_1(s)E_2(s)dt = \frac{1}{2\pi i}\int_{-i\infty}^{+i\infty} E(s)E(s)dt = \frac{1}{2\pi i}\int_{-i\infty}^{+i\infty} \frac{B(s)B(-s)}{A(s)A(-s)}dt \tag{13}$$

Where the numerator and denominators are written as

$$B(s) = b_{n-1}\, s^{n-1} + b_{n-1}s^{n-2} + \ldots + b_1 s + b_0 \tag{14}$$

$$A(s) = a_{n-1}\, s^{n-1} + a_{n-1}s^{n-2} + \ldots + a_1 s + a \tag{15}$$

Here the fourth order system is used, so the numerator and denominators are expressed in terms of their coefficients as

$$B(s) = b_0 + b_1 s + b_2 s^2 + b_3 s^3 \tag{16}$$

$$A(s) = a_0 + a_1 s + a_2 s^2 + a_3 s + a_4 s^4 \tag{17}$$

Finally for a fourth order system the Performance Index (*PI*) function is expressed in terms of their coefficients as per the Parseval's theorem as

$$(PI)_4 = \frac{b_3^2\left(-a_0^2 a_3 + a_0 a_1 a_2\right) + \left(b_2^2 - 2b_1 b_3\right)a_0 a_{14} + \left(b_1^2 - 2b_0 b_2\right)a_0 a_3 a_4 + b_0^2\left(-a_1 a_4^2 + a_2 a_3 a_4\right)}{2a_0 a_4\left(-a_0 a_3^2 - a_1^2 a_4 + a_1 a_2 a_3\right)} \tag{18}$$

The coefficients are shown in table 5.

Here the design specifications are decided based on from the available literature for the same type of vehicle specifications like same in wing span, weight etc. Here it was decided that the constraints are

Rise time < 0.7 sec

Settling time < 10 sec

Overshoot < 10 %

Then the optimization problem is formulated by adding the constraints to the *PI* which is also called the objective function with penalties as shown in below.

Table 5. Error function coefficients

Numerator	Denominator
$b_0 = 1.1998$	$a_0 = 0.1612K_i$
$b_1 = 2.4134$	$a_1 = (1.1998 + 0.558K_i + 0.1612K_p)$
$b_2 = 9.7418$	$a_2 = (2.4234 + 0.558K_p + 0.1612K_d)$
$b_3 = 1.0000$	$a_3 = (9.7418 + 0.558K_d)$
–	$a_4 = 1.000$

Minimize $(PI)_4$ + (Settling time-10) + (Rise time-0.7)+ (0vershoot-10) Subjected to

$$0 \le K_P \le 200$$

$$0 \le K_I \le 300$$

$$0 \le K_D \le 100$$

3.3. GA- *PID* controller

Genetic Algorithm (GA) is a stochastic global search method based on the mechanism of Darwin's survival of the fittest and natural genetics. They are iterative methods widely used in optimization problems in several branches of study. The algorithm starts by generating a set of random solutions called the population. Each solution is called a chromosome and it is usually encoded in the binary form. Each chromosome has a particular fitness value calculated by the fitness function which represents a measure of how good the solution is. In each iteration a new set of population is created. This is done by selecting two parents based on their fitness value and crossing them over i.e. some bits in the binary notation is taken from one parent and the remaining bits from the other parent. These may also be accompanied by some random mutation for preventing the solution to be stuck at local minima.

3.3.1. Simulation of results

Table 6 shows all the parameters used in the genetic algorithm.

Table 7 show the result generated after running the code.

Figure 8 shows the best function value vs. the iteration number

Figure 9 shows the step response for the result from GA.

Table 6. GA Parameters

Parameter	Value
K_d bound	0–100
K_p bound	0–200
K_i bound	0–300
Chromosomal representation	Binary
Population Size`	20
Chromosomes retained each iteration	2
No. of iterations	200
Mutation	2%

Table 7. GA Results

Parameter	Value
PI	0.05597
K_d	33.6920
K_p	76.0196
K_i	17.1679
Rise Time	0.1439
Settling Time	1.3083
Overshoot	1.1163

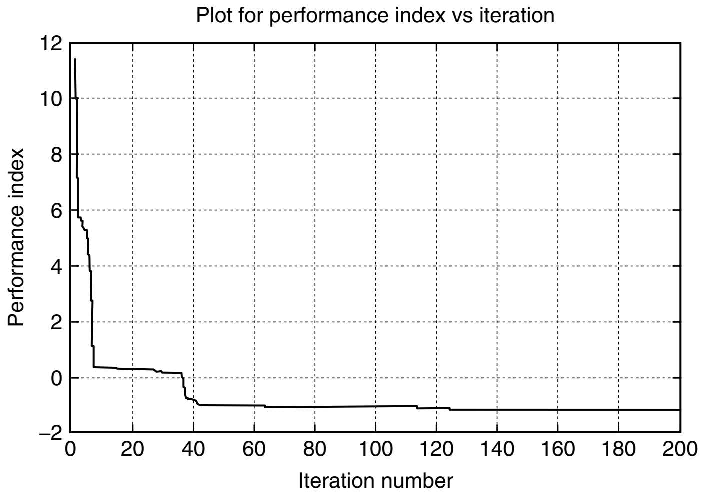

Figure 8. *PI* vs. iteration.

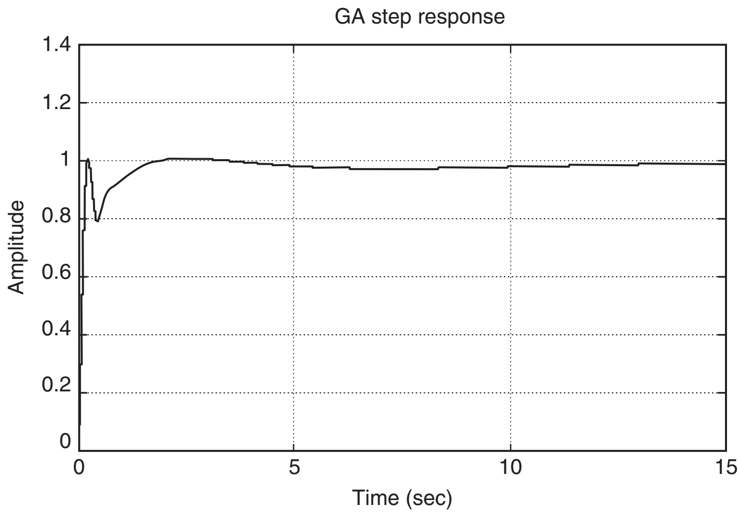

Figure 9. Step response with GA-*PID* controller.

From above *PID* tuning using Genetic Algorithm (GA) it can be observed from table 7 that the time domain specifications are in the limits of the design criteria. But the *PI* function is not a standard but actual. So it cannot be determined if the resultant optimum is correct. So it is better to check this with other well known algorithms like PSO which is explained in the next section.

3.4. PSO-*PID* controller

Particle swarm optimization (PSO) is a method for performing numerical optimization without explicit knowledge of the gradient of the problem to be optimized. PSO is originally attributed to Kennedy and Eberhart. In PSO the system is initialized with a set of random solution or particles. The potential solutions are called particles. They move through the problem space depending on the current optimum particle. Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called p_{best}. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called l_{best}. When a particle takes all the population as its topological neighbors, the best value is a global best and is called g_{best}. [16]. The particle swarm technique consists of each time changing the velocity and acceleration of the particle towards its l_{best} and p_{best} values depending on some random weights.

3.4.1. Simulation of results

Table 8 shows all the parameters used to run the PSO code for the problem at hand.

Table 9 show the result generated after running the code.

Figure 10 shows the fitness values vs. the iteration number for the g_{best}.

Figure 11 shows the response for the solution generated by PSO.

From the above simulation results it is observed that the PSO-*PID* is satisfying the time domain constraints and it gives the optimum *PI* as 0.05460 which nearly equivalent to that of GA-*PID*. It can also be checked with any other nature inspired techniques. Hence the novel nature inspired algorithm called Glowworm Swarm optimization (GSO) can be used and it is explained below.

Table 8. PSO Parameters

Parameter	Value
K_d bound	0–100
K_p bound	0–200
K_i bound	0–300
No of Agents	100
No of Iterations	50
$c1$	1
$c2$	1

Table 9. PSO results

Parameter	Value
PI	0.05460
K_d	35.8807
K_p	73.4228
K_i	17.1224
Rise Time	0.1384
Settling Time	1.3883
Overshoot	1.5485

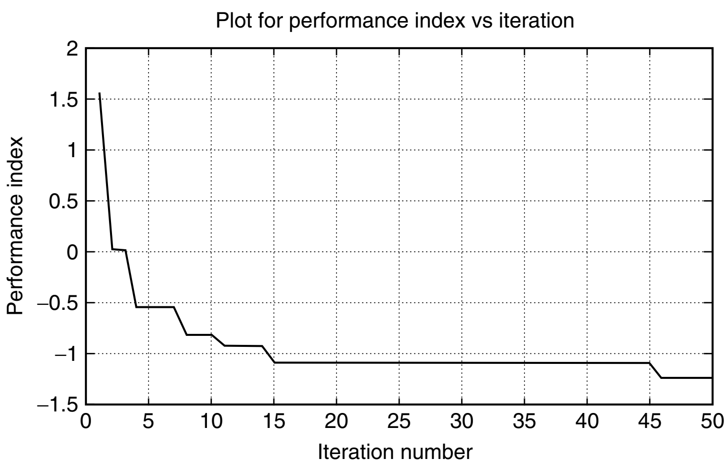

Figure 10. fitness values vs. the iteration number.

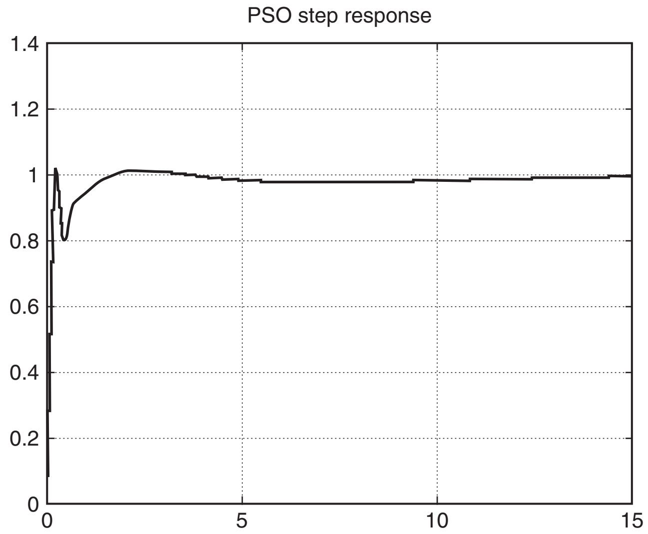

Figure 11. Step response with PSO-*PID* controller.

3.5. GSO-*PID* controller

The GSO algorithm solves continuous optimization problems. In the glowworm algorithm, the physical entities (agents) that are randomly distributed in the workspace are considered. The agents in the glowworm algorithm carry a luminescence quantity called *luciferin* along with them, which corresponds to the *pheromone* associated with the path between the nest and each region in ACO. Agents are thought of as glowworms that emit a light whose intensity is proportional to the associated luciferin and have a variable decision range. Each glowworm is attracted by the brighter glow of other neighboring glowworms. A glowworm identifies another glowworm as a neighbor when it is located within its current local-decision domain.

Description of algorithm:

Luciferin-update phase: The luciferin update depends on the function value at the glowworm position and so, even though all glowworms start with the same luciferin value during the initial iteration, these values change according to the function values at their current positions. During the luciferin update phase, each glowworm adds, to its previous luciferin level, a luciferin quantity proportional to the measured value of the sensed profile (temperature, radiation level) at that point. *Movement-phase:* During the movement-phase, each glowworm decides, using a probabilistic mechanism, to move towards a neighbor that has a luciferin value more than its own. That is, they are attracted to neighbors that glow brighter.

Local-decision range update rule: When the glowworms depend on only local information to decide their movements, it is expected that the number of peaks captured would be a strong function of the radial sensor range [17].

3.5.1. Simulation of results

Table 10 shows all the parameters used to run the GSO code for the problem at hand and table 11 shows the error, *PID* constants, and constraints got after running the code.

Figure 12 shows the emergence plot i.e. the path taken by each agent on the X and Y plane.

Figure 13 shows the fitness values vs. the iteration number for each agent.

Figure 14 shows the step response from the tuned *PID* parameters obtained from the GSO code

Here too the GSO-*PID* controller is satisfying the design considerations. Here the *PI* value is 0.050302 which is almost equivalent to that of GA-*PID* and GSO-*PID* controllers. In the next section a comparison of all three methods are done w.r.t their results.

3.6. Comparison of the three methods

Table 12 gives an overview of the results generated by each method.

Table 12 shows the results from different methods. Here it can be observed that *PI* is more in ZN method and the overshoot is more than the constraint specification. It has been overcome by using

Table 10. GSO Parameters

Parameter	Value
K_d bound	0–100
K_p bound	0–200
K_i bound	0–300
No of Agents	100
No of Iterations	1000
ρ	0.4
γ	0.6
β	0.08
Neighbor Threshold	2
S	1
l_o	5
r_s	1000
r_{d0}	20

Table 11. GSO Results

Parameter	Value
PI	0.050362
K_d	34.9303
K_p	80.2452
K_i	24.006
Rise Time	0.1388
Settling Time	1.1683
Overshoot	2.6844

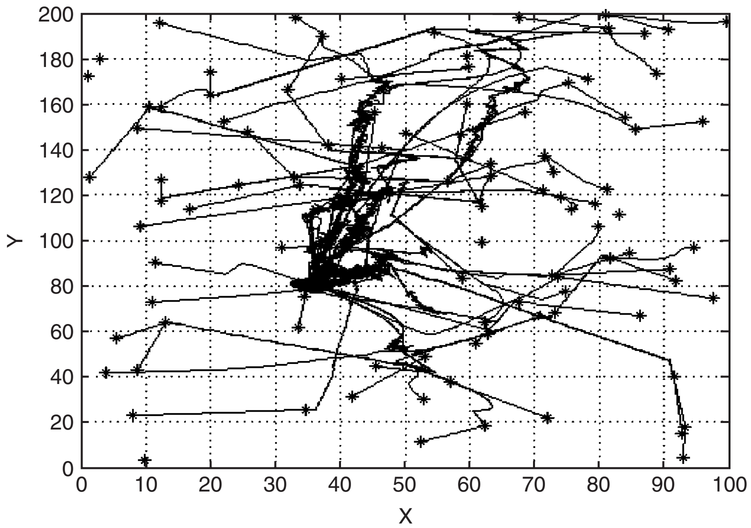

Figure 12. Emergence Plot of every agent over every iteration.

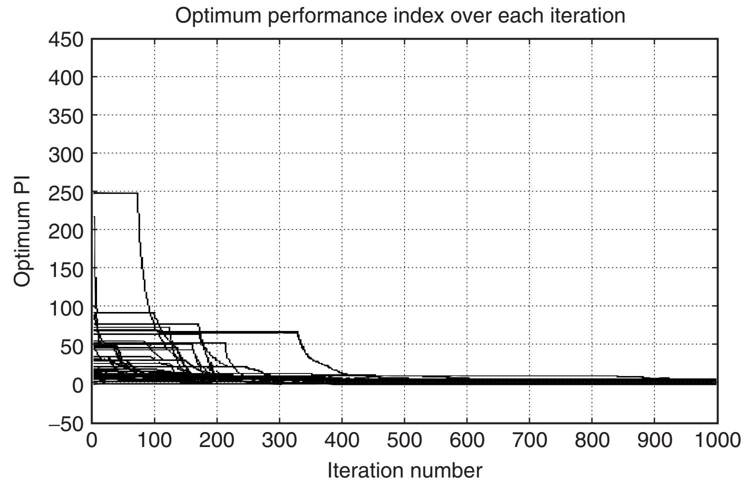

Figure 13. Fitness function vs. Iteration no.

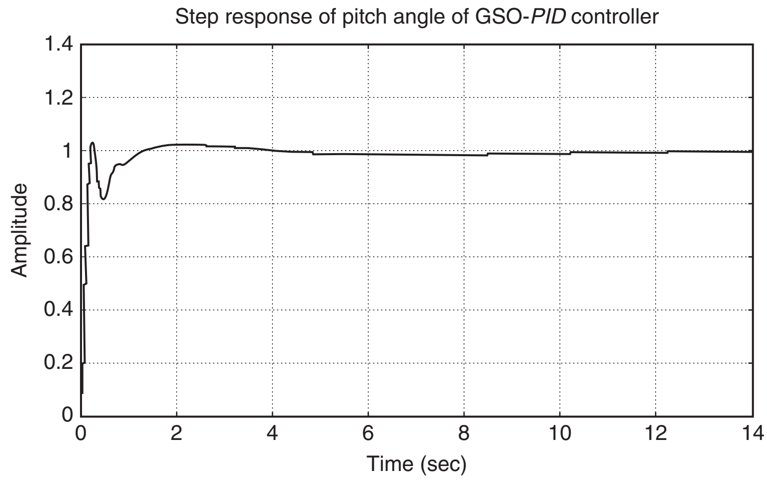

Figure 14. Step response plot with GSO-*PID* controller.

Table 12. Comparision of results of all the contrllers

Values	ZN	GSO	GA	PSO
PI	0.0978	0.050362	0.05597	0.05460
Rise Time	0.1851	0.1388	0.1439	0.1384
Settling Time	2.0215	1.1683	1.3083	1.3883
Overshoot	49.1098	2.6844	1.1163	1.5485
Iterations required	–	1000	200	50
Solutions evaluated at each iteration	–	100	20	100

nature inspired techniques. Among the nature inspired techniques GA performs well regarding the number of iterations as it results in lesser convergence time.

ACKNOWLEDGEMENTS
The authors wish to thank Prof V. Mani, Vinod Kumar A S and Kamlesh Kumar for useful discussions on this work.

REFERENCES

[1] Iain K. Peddle, *Autonomous flight of a model aircraft,* MSc Thesis, University of Stellenbosch, 2005.

[2] *Nidal M. Jodeh, Paul A. Blue, Athon A. Waldon, Development of small unmanned aerial vehicle research platform: Modeling and simulation with flight test validation, AIAA, 2006.*

[3] Navabalachandran J, Low J. H, Gerard Leng, *Reverse engineering and aerodynamic analysis of a flying wing UAV,* Proc. Of RSAF Aerospace Technology Seminar, 2005.

[4] Jin Fujinaga, Hirishi Tokutake, Shigeru Sunada, *Flight controller design and autonomous flight of 60 cm sized UAV,* Proc. of 3rd US-European Competition and Workshop on Micro Aerial Vehicle Systems and European Micro Aerial Vehicle Conference and Flight Competition, Toulouse, France, September, 2007, 1–11.

[5] http://www.//web.mit.edu/drelaPublic/web/avl/

[6] M.V Cook, *Flight Dynamics principles,* Elesevier Buttrworth Heinemann, 1997.

[7] I.J Nagarath, M. Gopal, *Control System Engineering,* 5th ed New Delhi, New Age International Publishers, 2007.

[8] Myo Myint, Htin Kyaw Oo, Zaw Min Naing, Yin Mon Myint, *PID controller for stability of Piper Cherokes pitch displacement using MATLAB,* Proc. of GMSARN International Conference on sustainable Development, Issues and Prospects for the GMS, November, 2008, 1–5.

[9] http://www.eng.uwi.tt/depts/elec/staff/copeland/ee27B/Ziegler_Nichols.pdf

[10] Ziegler J, Nichols B, *Optimum settings for automatic controllers,* Trans. ASME, Vol 64, 1942, 759–766.

[11] As Chaudhuri, P S Khuntia, S Sadhu, *Design of optimally convex controller for pitch control of an aircraft,* IE (I) Journal-As, November, Vol 88, 2007, 20–22.

[12] Branimir Stojiljkovic, Ljubisavasov, Caslav Mitrovic Dragan Cretkovic, *The application of the root locus method for the design of pitch controller of an F-104A aircraft,* Journal of Mechanical Engineering,Vol 55, 2009, 1–6.

[13] KamranTurkoglu,Ugur Ozdemir, Melike Nikvay, Elbrous M. Jafarov, *PID Parameter optimization of UAV longitudinal flight control system,* World Academy of Science, Engineering and technology 45, 2008, 341–345.

[14] Daniel Czarkowski, *Identification and Optimiztion of PID parameters using MATLAB,* Department of computer science, BSc, Thesis, Cork Institute of Technology.

[15] http://www.swarmintelligence.org/index.php

[16] Mohammed El-Said El-Telbany, *Employing Particle Swarm Optimizer and Genetic Algorithms for Optimal of PID Controllers: A Comparative Study,* Computers & Systems Department, Electronics Research Institute El-Tahrir St. Dokki.

[17] K. N. Krishnanand, D. Ghose, *Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions,* Technical Report GCDSL 2006/04, Department of Aerospace Engineering, Indian Institute of Science, July.(http://guidance.aero.iisc.ernet.in/gso.html).

[18] Bernard Etkin, Lloyd Duff Reid, *Dynamics of Flight, Stability and Control,* 3rd ed. John Wiley Sons.

[19] Chakravarthi Jada, *Modeling and controller design of an Unmanned Aerial Vehicle,* ME Thesis, Indian Institute of Science, Bangalore, India, 2010.

[20] Shashi Prakash Singh, *Autonomous landing of Unmanned Aerial Vehicles,* MSc Thesis, Indian Institute of Science, Bangalore, India, 2009.

[21] *V. Krishnamurthy, V. Seshadri, A simple and direct method for reducing order of systems using routh approximation in frequency domain,* IEEE Transactions on Automatic Control, vol. 22, no. 1, October, 1976, 797–799.