Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-1 (2016) 830–835

Novel two phase algorithm to design three loop autopilot using parameter plane technique and particle swarm optimization

Siddhardha, Kedarisetty

Indian Institute of Technology Madras, Chennai, India
(e-mail: siddhardhak2010@ gmail.com).

Abstract: Three loop autopilot is a popular autopilot structure used in aerial vehicles. Design of autopilot is based on user requirements that are given in terms of time domain specifications (like rise time, maximum overshoot, etc.) and frequency domain specifications (like gain and phase margins). In literature, it has been shown that three loop lateral acceleration autopilot for a rigid tactical aerospace vehicle (TAV) can be designed using time domain techniques such as pole placement or frequency domain techniques such as parameter plane technique. However an autopilot designed using time domain techniques may not satisfy frequency domain specifications and vice versa. This paper proposes a two phase algorithm for three loop autopilot design for a TAV such that both the time domain and the frequency domain requirements are met. In the first phase, the space of the control gains of the three loop autopilot is reduced to a subspace from which if control gains are chosen, the desired stability margins are guaranteed. This phase uses parameter plane methodology. The subspace so obtained is used in the second phase as a domain on which particle swarm optimization is used to obtain the optimal set of control gains that minimizes the deviation of the system from required time domain specifications. The efficacy of the proposed algorithm is illustrated through three loop lateral acceleration autopilot design example for a rigid body TAV.

Keywords: Lateral acceleration autopilot, Parameter plane technique, Particle swarm optimization

1. INTRODUCTION

A popular architecture for control of a tactical aerospace vehicle (TAV) consists of three layers -- the guidance system, the autopilot system, and the actuator system [1], [2], [4] & [5]. In this paper, we deal with the autopilot system design for a TAV. The dynamics of a TAV are non-linear, time varying, and are subject to uncertainties. In practice, a few assumptions are made to simplify the system dynamics, and controllers are designed for a simplified linear time invariant (LTI) model of the vehicle [3] & [5]. The LTI model assumption is valid for a small region of flight envelope (altitude versus Mach number) about a given operating point. These controllers typically involve feedback loops with some control gains, and therefore, the autopilot design, in these cases, boils down to finding appropriate values for these gains [5], [11] & [13]. Thus, a common practice in the design of autopilot for a TAV is to linearize the dynamics of the vehicle about a number of operating points, and design controllers for each of these operating points. For the purpose of this paper, we are interested in three loop autopilot design by tuning of the control gains for a given operating point.

Autopilot, like any other controller, is required to meet certain design specifications/requirements. Some of the these (stability requirements) are in the frequency domain -- like, gain and phase margins -- while others (performance requirements) are in the time domain -- rise time, overshoot, etc. Since the autopilot design is equivalent to selecting a set of control gains, in principle, it is possible to perform a search over the space of control gains, and using a suitable cost function that incorporates the design requirements, find gains that optimize the chosen cost function. However, because of the search domain size and the nature of cost function (non-monotonous with multiple local minima and maxima), such a brute force optimization approach will turn out to be tedious. Nonetheless, there have been efforts in the literature to design autopilot following this approach. Metaheuristic optimization techniques like particle swarm optimization (PSO) and genetic algorithm (GA) have been used on autopilot designs [15]& [16]. However, these works typically do not consider stability requirements and the autopilot structure is different from three loop autopilot [5] & [8]. An approach in designing autopilot that will guarantee stability, in terms of required gain and phase margins, is parameter plane technique. The references [9] & [12] discuss autopilot design using parameter plane technique for rigid and flexible TAVs respectively. However, autopilots designed thus may not satisfy performance requirements. All the above design techniques are single phase design techniques, that is, they either search the entire domain space of control gains to find solution that optimizes a 'not so meaningful' cost function, or find a region of the space (set of solution points) that satisfies only part of the requirements.

In this paper, we propose a two phase method for three loop autopilot design wherein, in the first phase, the parameter plane technique is used to find a subdomain of the control gains space such that any set of control gains from this sub-domain is guaranteed to satisfy the stability requirements. The first phase also ensures that the sub-domain is considerably small in size compared to the original domain. In the second phase, PSO is used to find an optimal set of control gains from this sub-domain that will optimize a cost function of the time domain requirements.

Rest of the paper is organized as follows: In Section 2, the vehicle dynamics and the autopilot structure is discussed. Section 3 describes the application of the

10.1016/j.ifacol.2016.03.161

parameter plane technique to TAV autopilot design. Bounds for PSO are obtained in section 4. The use of PSO to meet time domain specifications is discussed in Section 5. Simulation results and their discussion are provided in Section 6 followed by conclusions.

2. VEHICLE DYNAMICS

In this paper, we consider a skid to turn, tail-controlled TAV. Therefore, we restrict ourselves to the dynamics of TAV in the plane of turn where the angular rate (q) and the lateral acceleration (a_z) form the states. The transfer function that relates angular rate to the tail deflection (δ) is given as [8]

$$G_q(s) = \frac{q(s)}{\delta(s)} = \frac{K_3(1+T_\alpha s)}{\left(\frac{s^2}{\omega_d{}^2} + \frac{2\varepsilon_d}{\omega_d}s + 1\right)} \quad (1)$$

and the transfer function between lateral acceleration and tail deflection is [8]

$$G_z(s) = \frac{a_z(s)}{\delta(s)} = \frac{K_1(1+A_{11}s+A_{12}s^2)}{\left(\frac{s^2}{\omega_d{}^2} + \frac{2\varepsilon_d}{\omega_d}s + 1\right)} \quad (2)$$

The constants in these transfer functions depend on the airframe. The values of these constants used for the simulations presented in this paper is provided in Table 1. These values are taken from [8]

ω_d	ε_d	K_1	K_3	T_α	A_{11}	A_{12}
22.4	0.052	-1116.5	0.6477	0.676	0.001054	-0.00081

Table 1 Values of constants used in transfer function

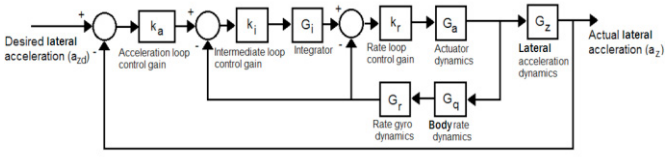

Figure 1 Generalized three loop latax autopilot block diagram

Simple but effective autopilot architecture has a 3-loop structure as shown in Figure 1 where the angular rate and lateral acceleration are fed back [5]. Such a 3-loop autopilot architecture is widely used in literature as well as in practice [8]. As seen in the figure, the autopilot commands are executed by an actuator which is assumed to have a second order dynamics as follows.

$$G_a(s) = \frac{\omega_a^2}{(s^2+2\xi_a\omega_a s+\omega_a^2)} \quad (3)$$

The rate gyro are also assumed to have a second order dynamics as shown in 4

$$G_r(s) = \frac{\omega_r^2}{(s^2+2\xi_r\omega_r s+\omega_r^2)} \quad (4)$$

The values of constants in the above equations, as used for producing simulation results presented in this paper, are given in Table 2.

ξ_a	ω_a	ξ_r	ω_r
0.7	250	0.65	500

Table 2 Damping and natural frequency values of actuator and rate gyro

Note from Figure 1 that the autopilot has controller gains -- acceleration loop gain (k_a), intermediate loop gain (k_i), and rate loop gain (k_r). The design of autopilot is equivalent to choosing these gains so as to attain the desired stability and performance requirements.

Now, we will discuss the selection of these control gains using the parameter plane technique so as to ensure desired stability margins.

3. PARAMETER PLANE TECHNIQUE

In the first phase of the autopilot design approach that we propose, we find the set of triplets (k_a, k_i, k_r) that satisfy the desired stability requirements. The stability requirements are usually specified in form of gain and phase margins that the inner, intermediate, and the outer loops of the autopilot (refer Figure 1) should possess [8]. We propose a solution to this problem using the parameter plane technique. In the literature, parameter plane technique has been used for the design and analysis of control systems in general [9], and autopilots for TAVs in particular [12] & [13]. For systems with two parameters, the technique gives a way to partition the parameter space into stable (desirable) and unstable (undesirable) regions.

The parameter space for a 3-loop autopilot is 3-dimensional -- one dimension for each of the parameters (control gains) k_a, k_i, and k_r. However, the parameter plane technique in its original form can be applied only to a 2D parameter space [9]. Therefore, we eliminate one of the parameters, k_r, by fixing it as follows: The value of k_r is chosen such that the inner loop bandwidth is approximately 1/3rd of the actuator bandwidth [5]& [12]. Thus k_r is fixed as

$$k_r = \left(\frac{1}{3}\right)\left(\frac{Bandwidth(G_a)}{Bandwidth(G_aG_qG_r)}\right) = 0.2903 \quad (5)$$

Having fixed k_r, we are left with a 2D parameter plane k_a-k_i, and our interest is in finding a region of this plane from which if we pick controller gains, the resultant autopilot will satisfy the stability requirements. The open loop transfer functions for the inner, intermediate, and the outer loops can be easily obtained from the autopilot structure shown in Figure 1. For example, the open loop transfer function of the inner loop is

$$G(j\omega) = \frac{N(j\omega)}{D(j\omega)} = Be^{j\varphi} \quad (6)$$

Where B and φ are amplitude or gain of the system and phase lag of the system respectively. Let A be the gain margin of the system and θ be the phase margin of the system then

$$G(j\omega) = \frac{N(j\omega)}{D(j\omega)} = -\frac{1}{A}e^{j\theta} \quad (7)$$

$$D(j\omega) + Ae^{-j\theta}N(j\omega) = 0 \quad (8)$$

$$M(j\omega) = D(j\omega) + Ae^{-j\theta}N(j\omega) \quad (9)$$

Where, $M(j\omega)$ is known as gain-phase margin tester function [12] & [13].

The equation (9) is equivalent to two equations i.e. real part and the imaginary part. Since the gain-phase margin tester function is also a function of k_a and k_i (apart from ω), for a given value of ω, equation (9) provides a solution (k_a,k_i). For different values of ω, the set of solutions (k_a,k_i) obtained will form a curve in the k_a-k_i space that partitions the parameter space into two [12]. Thus, for given values of A and θ, the equation $M(j\omega) = 0$ partitions the controller gain space into a region where the required gain and phase margins (A and θ) are satisfied and a region where they are not [13].

Step1: Fix the value of inner loop gain (k_3) such that the inner loop bandwidth is approximately 1/3rd the bandwidth of actuator. Also choose various values of A and θ for desired phase and gain margins.

Step 2. Calculate all the three loops open loop transfer functions G_{y1}, G_{y2}, G_{y3} and disintegrate G_a, G_i, G_q, G_z, and G_r into their corresponding numerators and denominators as shown below.

$$G_{y1} = \frac{N_{y1}(j\omega)}{D_{y1}(j\omega)} = \frac{k_a k_i G_i(s) k_r G_a(s) G_z(s)}{1 + k_r G_a(s) G_r(s) G_q(s) + k_r G_a(s) G_q(s) G_r(s) k_i G_i(s)} \quad (10)$$

$$G_{y2} = \frac{N_{y2}(j\omega)}{D_{y2}(j\omega)} = \frac{k_a k_i G_i(s) k_r G_a(s) G_z(s) + k_r G_a(s) G_q(s) G_r(s) k_i G_i(s)}{1 + k_r G_a(s) G_r(s) G_q(s)} \quad (11)$$

$$G_{y3} = \frac{N_{y3}(j\omega)}{D_{y3}(j\omega)} = k_a k_i G_i(s) k_r G_a(s) G_z(s) + k_r G_a(s) G_q(s) G_r(s) k_i G_i(s) + k_r G_a(s) G_r(s) G_q(s) \quad (12)$$

The values of in N_i (jω), D_i (jω), N_a (jω), D_a (jω), N_r (jω), D_r (jω), N_q (jω), D_q (jω), N_z (jω), D_z (jω) for various frequencies are calculated in MATLAB.

Step 3. Calculate margin tester function for open loop transfer functions using (9) choosing k_a (outer loop control gain) and k_i (middle loop control gain) as unknowns. For example outer loop open loop margin tester function is given in equation (13)

$$\begin{aligned} &D_i(j\omega)D_a(j\omega)D_z(j\omega)D_q(j\omega)D_r(j\omega) \\ &+ k_r N_a(j\omega)N_r(j\omega)N_q(j\omega)D_i(j\omega)D_z(j\omega) \\ &+ k_r N_a(j\omega)N_q(j\omega)N_r(j\omega)k_i N_i(j\omega)D_z(j\omega) \\ &+ Ae^{-j\theta}k_a k_i k_r N_i(j\omega)N_a(j\omega)N_z(j\omega)D_q(j\omega)D_r(j\omega) = 0 \end{aligned} \quad (13)$$

In the equation (13) k_a and k_i are unknowns. Thus writing it in a simplified format we obtain

$$P_o k_i + Q_o k_a k_i + R_o = 0 \quad (14)$$

Where,

$$P_o = k_r N_a(j\omega)N_q(j\omega)N_r(j\omega)N_i(j\omega)D_z(j\omega) \quad (15)$$

$$Q_o = Ae^{-j\theta}k_r N_i(j\omega)N_a(j\omega)N_z(j\omega)D_q(j\omega)D_r(j\omega) \quad (16)$$

$$R_o = D_i(j\omega)D_a(j\omega)D_z(j\omega)D_q(j\omega)D_r(j\omega) + k_r N_a(j\omega)N_r(j\omega)N_q(j\omega)D_i(j\omega)D_z(j\omega) \quad (17)$$

Separating real and imaginary parts yields two equations with two unknowns.

Step 4. Solving the two equations we obtain k_a and k_i for different values of ω. Thus plotting k_a vs k_i we obtain a curve on the parameter plane for desired stability margins.

Step 5. Steps 2 to 4 are repeated for middle and inner most loop to obtain region of control gains which satisfy desired stability margins.

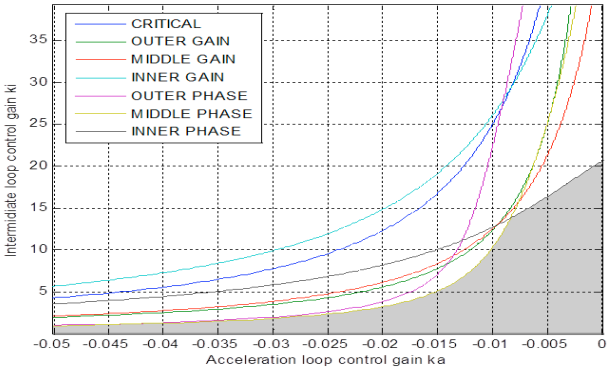

Figure 2 Parameter plane for three loop autopilot with rigid body dynamics

For the simulations presented in this paper, the desired gain and phase margins of all the loops are chosen as 6 dB and 40 degrees. The plot of how gain and phase margin requirements of the inner, intermediate, and the outer loops partition the controller gain space is given in Figure 2. Intersection of all the favourable partitions gives a region (shown as shaded in Figure 2) from where if a set of controller gains (k_a,k_i) are chosen, the resultant 3-loop autopilot is guaranteed to meet all the stability requirements.

This completes phase one of our autopilot design algorithm. In phase two, a set of controller gains is chosen from the feasible region of the controller gain space obtained in phase one such that a cost function that penalizes the deviations from desired time domain specifications is minimized. This is explained in next section.

4. BOUNDS FOR PARTICLE SWARM OPTIMIZATION

The performance requirements of a controller is usually specified in time domain in the form of requirements on the step response characteristics like rise time, overshoot, and settling time [10]& [15]. As seen from equation (2), TAV is a non-minimum phase system and therefore will also have

an undershoot as part of the step response. Given the desired time domain behaviour, a cost function that penalizes the deviation from desired response can be formulated as

$$Cst = w_1(us - usd)^2 + w_2(os - osd)^2 + w_3(ts - tsd)^2 + w_4(rs - rsd)^2 \quad (18)$$

Where,

usd,us are desired and actual undershoots respectively.
osd, os are desired and actual overshoots respectively.
tsd, ts are desired and actual settling times respectively.
rsd are desired and actual rise times respectively.
w_1, w_2, w_3, w_4 are weights.

The k_a and k_i values that minimize the cost function (18) will give an optimal autopilot which will have a performance that is close to the requirement. Thus, the second phase of the proposed autopilot design involves searching the controller gain space for an optimal pair (k_a,k_i) that will minimize the cost function (18). As this cost function is a multi modal function, we resort to numerical optimization for minimization. Although in phase I, we have eliminated a part of the search space, the resultant search domain is still unbounded. Such a search domain is not amenable for a numerical optimization technique. Reasonable bounds on the search domain can be given as follows.

$$k_{al} \le k_a \le k_{au}$$
$$k_{il} \le k_i \le k_{iu}$$

The lower bound for k_i and the upper bound for k_a are zeros. A lower bound for k_a can be obtained from the fact that outer loop should be three times faster than the intermediate loop and the intermediate loop must be three times faster than the inner loop. Thus, the lower bound for k_a is given as

$$k_{al} = \left(\frac{1}{27}\right)\left(\frac{Bandwidth(G_a)}{Bandwidth(G_z)}\right) \quad (19)$$

For the system that we are considering, this value turns out to be

$$k_{au} = 0,\ k_{il} = 0 \text{ and } k_{al} \approx -0.02$$

The upper bound for k_i is obtained from the computations performed in phase I. As can be seen from Figure 2, the upper bound on k_i can be given as a function of k_a. This information can be stored as a look up table or through a curve fitting.

$$k_{iu}(k_a) = \min(c(k_a), g_1(k_a), g_2(k_a), g_3(k_a), p_1(k_a), p_2(k_a), p_3(k_a))$$

Where, $c(k_a)$ is critical stability margin curve, $g_1(k_a)$, $g_2(k_a)$, $g_3(k_a)$ are outer loop, intermediate loop and inner loop gain margin curves respectively. Similarly $p_1(k_a)$, $p_2(k_a)$, $p_3(k_a)$ are outer loop, intermediate loop and inner loop phase margin curves respectively.

For the computations performed in this paper, we used an 8th degree polynomial to represent the boundary of the feasible region which also gives the upper bound for k_i as a function of k_a.

$$k_{iu}(k_a) = 0.0222z^8 - 0.0357z^7 - 0.576z^6 - 0.329z^5 + 2.26z^4 + 0.873z^3 - 3.318z^2 + 5.0694z + 14.861 \quad (20)$$

Where, $z = \frac{(k_a + 0.0069892)}{0.0048916}$

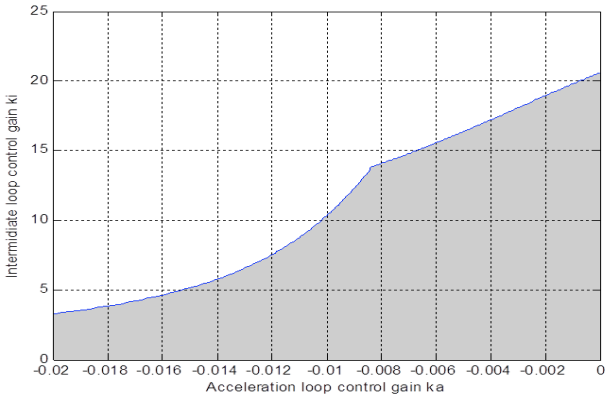

Figure 3 Seach region for PSO

Finally the bounds are

$$-0.02 \le k_a \le 0$$
$$0 \le k_i \le k_{iu}(k_a)$$

Now that there is a cost function that is to be minimized over a bounded domain, we can use some numerical optimization technique. Since the cost function in (18) is not 'smooth', any gradient based optimization technique cannot be used to find the optimal controller. Therefore, we propose to use the metaheuristic optimization technique called Particle Swarm Optimization (PSO) [6] & [7].

5. PARTICLE SWARM OPTIMIZATION

PSO involves an iterative process. It starts with a set of particles (solution candidates) scattered in the search domain. Each particle has a cost associated with it which is equal to the value of the cost function evaluated at that point in the search domain where the particle is. The PSO algorithm used in this paper is modified PSO [7].

PSO algorithm:

1) Initialization: Choose ten coordinates within the calculated bounds as shown in the figure 5. Where i ϵ [1, 10]

2) Cost calculation: Calculate the cost at every point of (ka, ki) pair cost_n(i) using the cost function (18). The weights chosen are w1 = 1, w2 = 1, w3 = 10, w4 = 10.

3) Calculation of minima: Find the global minimum point (gmin_k_a, gmin_k_i) in the present iteration and local minima (lmin_k_a(i), lmin_k_i(i)) for all the particles in all the previous iterations.

4) Velocity: Velocity of every particle based in the global and local minimum for every particle is given by

$$V_{ka}(i) = wt1 * V_{ka}(i) + c_1 * rand(0,1) * (lmin\_k_a(i) - k_a(i)) + c_2 * rand(0,1) * (gmin\_k_a - k_a(i)) \quad (21)$$

$$V_{ki}(i) = wt1 * V_{ki}(i) + c_1 * rand(0,1) * (lmin\_k_i(i) - k_i(i)) + c_2 * rand(0,1) * (gmin\_k_i - k_i(i)) \quad (22)$$

5) Gain updation:

$$k_a(i) = V_{ka}(i) + k_a(i) \quad (23)$$
$$k_i(i) = V_{ki}(i) + k_i(i) \quad (24)$$

6) Cost updation: The cost values now will be updated as the previous iteration cost values by

$$cost\_o = cost\_n \quad (25)$$

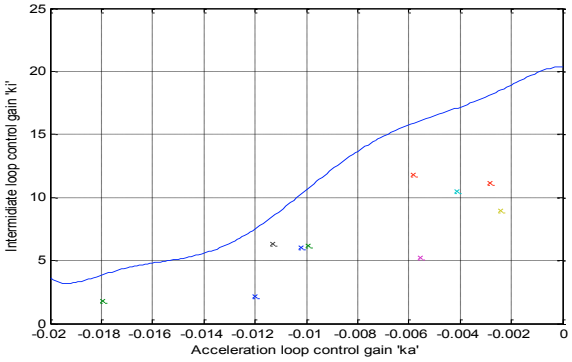

Figure 4 Initial positions of the particles in the search region

This process will iterate till the convergence is achieved. From figure 6 we can see that the convergence is achieved in less than 50 iterations (as the original search region k_a - k_i is reduced because of parameter plane analysis).

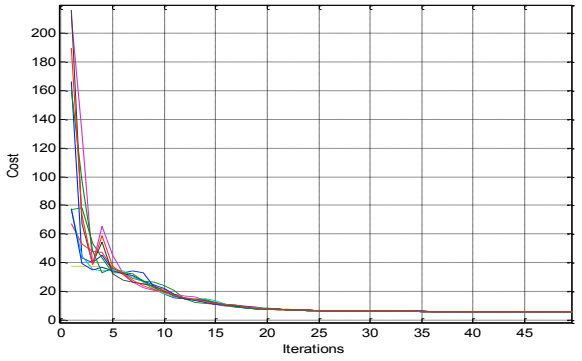

Figure 5 Convergence of PSO

To keep the computational requirements low, we employ only a few particles. Therefore, there is a tendency, as was observed during the simulations we conducted, for the particles to converge to a local minimum rather than a global one. One way to overcome this is by increasing the number of particles at the expense of computational cost. The other workaround, as adopted for simulation studies in this paper, will be to re-run the PSO algorithm by randomly placing of the particles once the convergence is achieved. In the end we consider the minimum cost achieved after every PSO iteration and consider it to be the optimal most result PSO could achieve. Such process is done five times which gives us a higher probability of achieving global minimum. The plot of cost vs. iterations after PSO iterations is shown in figure 7.

S.No	PSO iteration	Cost	(Ka, Ki)
1	1	6.2154	(-0.0066,6.2768)
2	2	0.0560	(-0.0060,8.6182)
3	3	4.2209	(-0.0037,13.6723)
4	4	0.0560	(-0.0060,8.6130)
5	5	0.0560	(-0.0060,8.6146)

Table 3 Cost and control gains after each scatter iteration

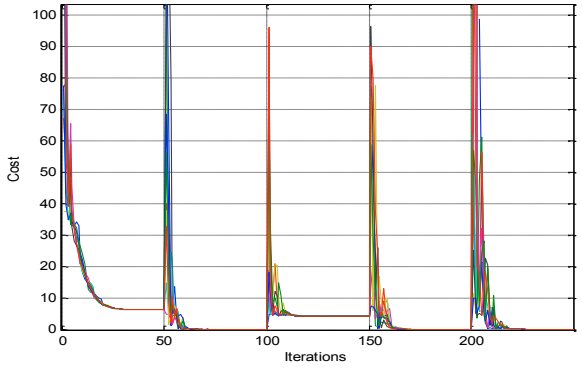

Figure 6 Convergence for each PSO iteration

The table 3 shows us the cost of the point to which PSO converges every time.

6. RESULTS

In this section, we will present the simulations results. The time domain specifications in terms of desired maximum percentage overshot, maximum percentage undershoot, rise and settling times are shown in table 4.

Overshoot	Undershoot	Rise time	Settling time
2%	10%	0.15 sec	0.3 sec

Table 4 Time domain specifications

The optimal (k_a, k_i) pair obtained from the PSO algorithm is

ka	-0.0060
ki	8.6182

Table 5 Optimal control gains obtained from PSO

The actual stability and performance characteristics of the system with the designed autopilot are given in Tables 6 and 7. The step response and bode plots of the TAV system augmented with the designed autopilot are shown in Figure 8 and Figure 9 respectively. From the figure and tables, it can be seen that all the frequency and time domain specifications are satisfied.

Overshoot	Undershoot	Rise time	Settling time
1.9946%	9.9984%	0.1449 sec	0.2492 sec

Table 6 Time domain characteristics of latax autopilot using optimal control gains

Loop	Gain margin (dB)	Phase margin (deg)	Gain crossover (rad/sec)	Phase crossover (rad/sec)
Inner	9.48	49.8	34	164
Middle	12.8	61.3	16	61.4
Outer	13.2	70.3	7.5	37.7

Table 7 Stability margins and crossover frequencies of all three loops of latax autopilot

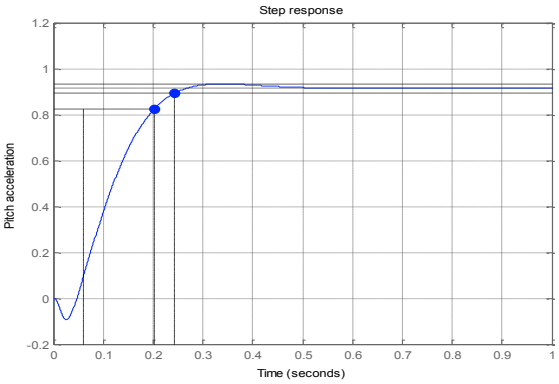

Figure 7 Step response of latax autopilot with optimal control gains

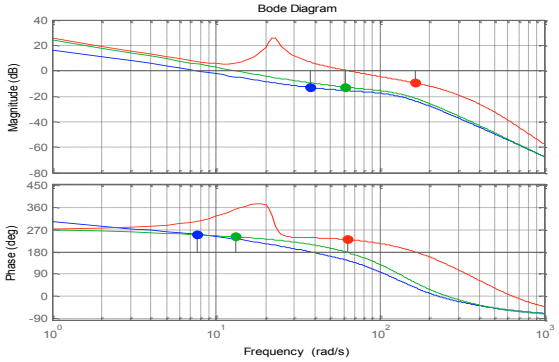

Figure 8 Bode plot of latax autopilot with optimal control gains

7. CONCLUSIONS

This paper dealt with the lateral acceleration autopilot design for a rigid body tactical aerospace vehicle. The proposed design procedure consists of two phases: In the phase I, parameter plane technique is used to find a set of autopilots that satisfy the stability requirements. In phase II an optimal autopilot is picked up from the set returned in phase I using particle swarm optimization technique. The resultant autopilot is optimal in the sense that it minimizes the deviation from desired time domain performance. The proposed design procedure is easy to implement and is computationally efficient. (Include any other salient features and advantages of the proposed algorithm). Future work will be to extend the proposed autopilot design procedure to be applied to a flexible body TAV and include filters in the auto pilot structure to shape frequency response of the system.

ACKNOWLEDGEMENT

The author would like to thank Dr. Joel George for his invaluable advices on paper write up and fruitful discussions on particle swarm optimization.

REFERENCES

[1] Garnell, P., East, D.J.(1977). “Guided weapon systems”. Pergamon press.

[2] Zarchan, P.” Tactical and Strategic Missile Guidance” American institute of Aeronotics and Astronomics Aug 1998.

[3] Siouris, G.M. (2004). “Missile guidance and control systems”, Springer-verlog, New York.

[4] J.H. Blakelock, “Automatic Control of Aircraft and Missile”, Vol. II, Amir Kabir University Publication, 2004.

[5] Kadam, N.V. (2009). “A practical design of flight control systems”, Allied publishers Pvt. Ltd.

[6] Kennedy, J.; Eberhart, R. (1995). *"Particle Swarm Optimization"*. Proceedings of IEEE International Conference on Neural Networks IV. pp. 1942–1948.

[7] Shi, Y.; Eberhart, R.C. (1998). "A modified particle swarm optimizer". Proceedings of IEEE International Conference on Evolutionary Computation. pp. 69–73.

[8] Nesline F, Jr., Nesline, L. Mark. (1985). “Phase vs gain stabilization of structural feedback oscillations in homing missile autopilots”. American control conference, Boston, 323-329.

[9] D. D. Siljak "Analysis and Synthesis of Feedback Control Systems in the Parameter Plane I-Linear Continuous Systems", *IEEE Trans. on Application and Industry*, vol. 83, pp.449 -473 1964.

[10] M. Zamani, N. Sadati and M. K. Ghartemani, 2009. Design of an H∞ PID Controller Using Particle Swarm Optimization, International Journal of Control, Automation, and Systems, vol. 7, pp. 273-280.

[11] Parijat Bhoumick and Dr. Gourhari Das, “Three loop Lateral Missile Autopilot Design in pitch plane using State feedback & Reduced Order Observer(DGO)”, International Journal of Engineering Research and Development (IJERD) , Volume 1, Issue 8(June 2012), PP 12-17.

[12] Sandip Ghosh, Kalyankumar Datta, Shyamal Kumar Goswami, Samar Bhattacharya. “A Parameter Plane Design Methodology for Three Loop Missile Autopilot”, Journal of Institute of Engineers(India) vol-86,2006 Page no. 214-219.

[13] Kedarisetty, Siddhardha; Narayanan, Vignesh; Halder, Pulak, “Autopilotot Design for Flexible Tactical Aerospace Vehicle Using Parameter Plane Technique”, ACODS, Volume # 3 | Part# 1, pages: 211-218, 2014.

[14] Shamma, J.S. ; Cloutier, James R., “A Linear Parameter Varying Approach to Gain Scheduled Missile Autopilot Design”, American Control Conference, 1992, Pages: 1317 – 1321.

[15] Mohammad Fiuzy, Javad Haddadnia, Seyed Kamaleddin Mousavi Mashhadi, “Designing an Optimal PID Controller for Control the Plan’s Height, Based on Control of Autopilot by using Evolutionary Algorithms”, Journal of mathematics and computer Science, Vol – 6, 2013, Pages: 260 – 271.

[16] Byoung-Mun Min, Hyeok Ryu, Daekyu Sang, Min-Jea Tahk, David Hyunchul Shim, “Autopilot Design Using Hybrid PSO-SQP Algorithm”, Communications in Computer and Information Science, Series vol- 2, 2007, pp 596-604.