Automatic landing System Design using PSO method for Charlie Aircraft

Gaber El-Saady, El-Nobi A.Ibrahim, Abdelhamid A.Raafat

***Abstract*—This paper presents an aircraft automatic landing controller that uses Particle Swarm Optimization technique to improve the performance of automatic landing system and guide the aircraft to a safe landing, and this paper demonstrate in detail how to employ the PSO to search efficiently the optimal PID controller parameters of an automatic landing system.**

***Index Terms*—Aircraft, flight control, ALS, PSO and PID control.**

I. Introduction

Most commercial aircraft currently have available an optional automatic landing system. The first ALS was made in England on june 10,1965 using a trident aircraft operated by the British airline BEA [1]. Since then, most aircraft have had this system installed. They are most often activated in clear, calm weather but can also be used in fog or rain. When they are used in place of a pilot it is usually for two reasons. First, automatic system give a reliably smother landing, leading to increased the passenger comfort and to reduction of wear on tires and landing gear. Second, their use in calm weather provides an important training function, accustoming pilots to a system that they must understand and trust if it to be used in adverse conditions in the future. In recently years, several of control techniques are applied to ALS to improve it, such as GNSS Integrity Beacons, Global Positioning System, Microwave Landing System, and Autoland Position Sensor [2,3,4,5]. By using improvement calculation methods and high accuracy instruments, these systems provide more accurate flight data to the ALS and can help to make landing smoothers. Recently, some researches have applied some intelligent concepts such as neural networks, fuzzy systems, genetic
algorithm, and hybrid systems to flight control increase the flight controller's adaptively to different environments [6,7,8,1,9].

In this paper, we proposed a heuristic method to select PID controllers for Charlie aircraft using Particle Swarm Optimization algorithm. The paper is organized as follows, section II provides PSO Algorithm, section III describes aircraft equation of motion while section IV details the Charlie aircraft model used in this work. Section V describes simulation and results. Finally section VI presents the conclusions of this work.

Manuscript received October 09, 2014.

Gaber El-Saady, PhD, Electrical Engineer Department, Faculty of Engineering, Assiut University, Assiut.,Egypt.

EL-Nobi A.Ibrahim, PhD, Electrical Engineer Department, Faculty of Engineering, Assiut University, Assiut.,Egypt.

Abdulhamid A.Raafat Navigation aids engineer, National Air Navigation Services Company-Ministry of Civil Aviation, Assiut.,Egypt

II. PSO Algorithm

The PSO has been successfully applied in function optimization problems. In this paper, we have attempted to obtain PID controllers by converting an ALS design problem to a parameter optimization problem.

The PSO is one of the evolutionary computation techniques introduced by Kennedy and Eberhart in 1995 [10]. The PSO algorithm is similar to evolutionary computation in producing a random population initially and generating the next population based on current cost, but it does not need reproduction or mutation to produce the next generation. Thus, PSO is faster in finding solutions compared to any other evolutionary computation technique. In PSO algorithm, each particle is moving, and hence has a velocity. Also, each particle remembers the position it was in and where it had its best result so far. Moreover, the particles in the swarm cooperate exchanging information about what they have discovered in the search region they have visited [11]. The basic PSO can be summarized as follows.

A. Basic PSO Algorithm

In a PSO system, a swarm of individuals (called particles or intelligent agents) fly through the search space. Each particle represents a candidate solution to the optimization problem. The position of a particle is influenced by the best position visited by itself (i.e. its own experience) and the position of the best particle in its entire population. The best position obtained is referred to as the global best particle. The performance of each particle (i.e. how close the particle is from the global optimum) is measured using a fitness function that varies depending on the optimization problem. Each particle traverses the XY coordinate within a two-dimensional search space. Its velocity is expressed by vx and vy (the velocity along the X-axis and Y-axis, respectively). Modification of the particles position is realized by the position and velocity information [12]. Each agent knows its best value obtained so far in the search (pbest) and its XY position. This information is an analogy of the personal experiences of each agent. Individual particles also have knowledge about the best value achieved by the group (gbest) among pbest. Each agent uses information relating to: its current position (x,y), its current velocities (vx,vy), distance between its current position and its pbest and the distance between its current position and the groups gbest to modify its position.

The velocity and position of each agent is modified according (1) and (2) respectively [13]:

$$V_i^{k+1} = V_i^{k} + c_1 rand_1 \times (pbest_i - s_i^{k}) + c_2 rand_2 \times (gbest - s_i^{k}) \quad (1)$$

$$s_i^{k+1} = s_i^{k} + V_i^{k+1} \quad (2)$$

With regards to equation 1 and 2, s^{k} and s^{k+1} denote the current and modified search point, respectively; V^k and V^{k+1} respectively represent the current and modified velocity; V_{pbest} and V_{gbest} represents the velocity based upon pbest and gbest, respectively.

To control swarm convergence involves a system of "constricted coefficients" applied to various terms of the conventional swarm velocity algorithm. This so called constriction factor approach controls the swarm convergence so that:

- The swarm does not diverge in a real value region.
- The swarm converges and searches region more efficiently.

The modified velocity update equation is given in (3) [14]:

$$V_i^{k+1} = X[V_i^{k} + c_1 rand_1 \times (pbest_i - s_i^{k}) + c_2 rand_2 \times (gbest - s_i^{k})] \quad (3)$$

With regards to (3): *X* represents the constriction factor and is defined in (4)

$$x = \frac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}|} \quad (4)$$

B. Steps in implementing the PSO method

Figure 2.1 illustrates the general flowchart for the PSO technique. The sequence can be described as follows:

Step 1: Generation of initial conditions of each agent.

Initial searching points (s_i^o) and the velocities (v_i^o) of each agent are usually generated randomly within the allowable range. The current searching point is set to pbest for each agent. The best-evaluated value of pbest is set to *gbest* and the agent number with the best value is stored.

Step 2: Evaluation of searching point of each agent.

The objective function is calculated for each agent. If the value is better than the current *pbest* value of the agent, then *pbest* is replaced by the current value. If the best value of *pbest* is better than the current *gbest*, the *gbest* value is replaced by the best value and the agent number with the best value is stored.

Step 3: Modification of each searching point.

The current searching point of each agent is changed, using equations (2), (3) and (4) is used for the constriction factor method.

Step 4: Checking to exit condition.

The terminating criterion is checked to determine whether it has been achieved. If the terminating criterion is not met then the process is repeated from Step 1, otherwise the algorithm is stopped.

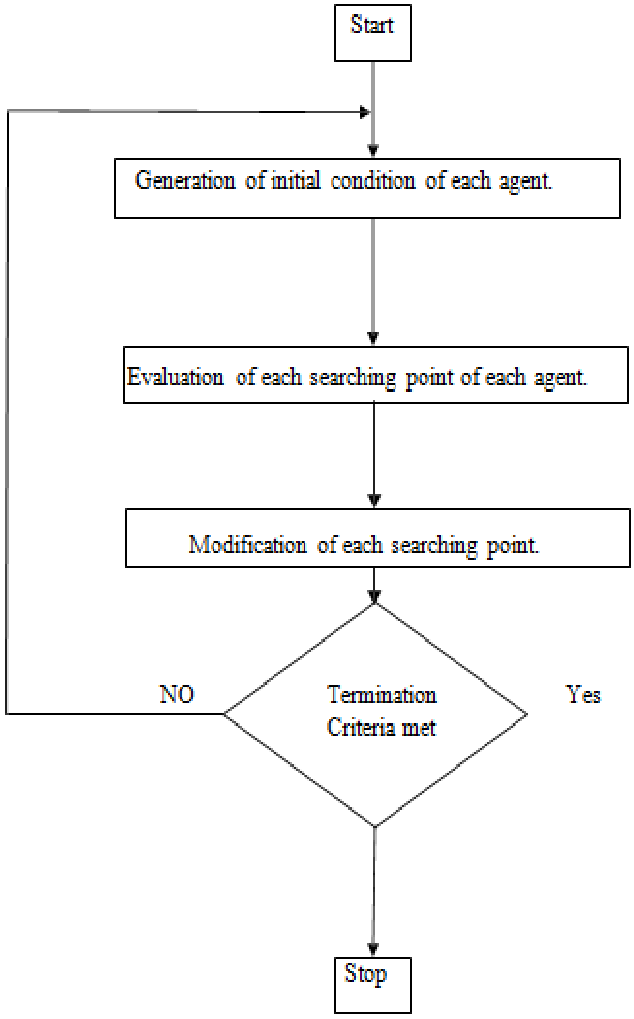

Figure 2.1: Steps in PSO

C. Performance Index

The PSO algorithm has been developed that uses a numerical optimization method. In general, parameters to optimize are evaluated by performance index in the numerical optimization method. Therefore, the way of defining the performance index effect on the optimization results considerably. In general, the PID controller design method using the integrated absolute error (IAE), or the integral of squared error (ISE),or the integrated of time weighted squared error (ITSE), or the integral of time multiplied by absolute error (ITAE) is often employed in control system because it can be evaluated analytically in frequency domain [15,16,17,18].

The four integral performance criteria have their own advantages and disadvantages. In this paper we are use

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

Systems based on this index penalize the control error. System's designed using this criterion has small overshoots and well damped oscillations [19].

A summary of the performance indices and their properties is shown in table 2.1

Table 2.1: Summary of performance indices

Performance Index	Equation	Properties
ISE	$ISE = \int_0^{\infty} e^2(t)\,dt$	Penalizes large control errors. Settling time longer than ITSE. Suitable for highly damped systems.
ITSE	$ITSE = \int_0^{\infty} te^2(t)\,dt$	Penalizes long settling time and large control errors. Suitable for highly damped systems.
IAE	$IAE = \int_0^{\infty} \lvert e(t)\rvert\,dt$	Penalizes control errors.
ITAE	$ITAE = \int_0^{\infty} t\lvert e(t)\rvert\,dt$	Penalizes long settling time and control errors.

III. Aircraft Equation of Motion

A. Aircraft Equations of Longitudinal Motion

In order to obtain the transfer function of the aircraft, it is first necessary to obtain the equations of motion for the aircraft. The equations of motion are derived by applying Newton's Laws of motion which relate to the summation of the external forces and moments to the linear and angular accelerations of the system or body. Certain assumptions must be made to do this application. By the way, the application is done according to [20].

Furthermore in longitudinal dynamics in order to get the linearized and Laplace transformed equations of motion, stability derivatives have to be also calculated. Then the related force term and moment term are handled, the longitudinal equations of motion for the aircraft are written as;

$$\dot{u} = X_u u + X_w w - g\cos\gamma_0\,\theta$$
$$\dot{w} = Z_u u + Z_w w + U_o q - g\sin\gamma_0\,\theta + Z_{\delta E}\delta_E$$
$$\dot{q} = M_u u + M_w w + M_{\dot{w}}\dot{w} + M_q q + M_{\delta E}\delta_E$$
$$\dot{\theta} = q \qquad (5)$$

B. Transfer functions obtained from Short Period Approximation

The short period approximation consists of assuming that any variations, u, which arise in airspeed as a result of control surface deflection, atmospheric turbulence, or just aircraft motion, are so small that any terms in the equations of motion involving u are negligible. In other words, the approximation assumes that short period transients are of sufficiently short duration that Uo remain essentially constant, i.e. u = 0. Thus, the equations of longitudinal motion may now be written as:

$$\dot{w} = Z_w w + U_o q + Z_{\delta E}\delta_E$$
$$\dot{q} = M_w w + M_{\dot{w}}\dot{w} + M_q q + M_{\delta E}\delta_E = (M_w + M_{\dot{w}} Z_w)w + (M_q q + U_o M_{\dot{w}})q + (M_{\delta E} + Z_{\delta E} M_{\dot{w}})\delta_E \qquad (6)$$

If the state vector for short period motion is now defined as:

$$X \triangleq \begin{vmatrix} w \\ q \end{vmatrix} \qquad (7)$$

And the control vector, u, is taken as the elevator deflection, δ_E, then eqs (6) may be written as a state equation:

X = Ax + Bu

Where:

$$A = \begin{vmatrix} Z_w & U_0 \\ (M_w + Z_{\dot{w}}) & (M_q + U_0 M_{\dot{w}}) \end{vmatrix}$$

$$B = \begin{vmatrix} Z_{\delta E} \\ (M_{\delta E} + Z_{\delta E} M_{\dot{w}}) \end{vmatrix}$$

$$[sI - A] = \begin{vmatrix} (s - Z_w) & -U_0 \\ (M_w + M_{\dot{w}} Z_w) & (s - [M_q + U_0 M_{\dot{w}}]) \end{vmatrix}$$

$$\Delta_{sp}(s) = \det[sI - A] = s^2 - [Z_w + M_q + M_{\dot{w}} U_o]s + [Z_w M_q - U_o M_w] \qquad (8)$$

It is easy to show that:

$$\frac{w(s)}{\delta_E(s)} = \frac{(U_0 M_{\delta E} - M_q Z_{\delta E})\left\{1 + \frac{Z_{\delta E}}{(U_0 M_{\delta E} - M_q Z_{\delta E})}s\right\}}{\Delta sp(s)} = \frac{K_w(1 + sT_1)}{\Delta sp(s)}$$

Where:

$$k_w = (U_0 M_{\delta E} - M_q Z_{\delta E})$$

$$T_1 = Z_{\delta E} / K_w$$

Also:

$$\frac{q(s)}{\delta_E(s)} = \frac{(Z_{\delta E} M_w - M_{\delta E} Z_w)\left\{1 + \frac{M_{\delta E} + Z_{\delta E} M_{\dot{w}}}{(Z_{\delta E} M_w - M_{\delta E} Z_w)}s\right\}}{\Delta sp(s)} = \frac{k_q(1 + sT_2)}{\Delta sp(s)}$$

Where:

$$K_q = (Z_{\delta E} M_w - M_{\delta E} Z_w)$$

$$T_2 = (M_{\delta E} + Z_{\delta E} M_{\dot{w}}) / K_q \qquad (9)$$

C. Flight Path Angle

There is a useful kinematic relationship which can be found by means of the short period approximation: to change the flight path angle, γ, of an aircraft it is customary to command a change in the pitch attitude, θ, of the aircraft. Since

$$\gamma = \theta - \alpha \qquad (10)$$

$$\frac{\gamma(s)}{\delta_E(s)} = 1 - \frac{\alpha(s)}{\delta_E(s)} \cdot \frac{\delta_E(s)}{\theta(s)} \qquad (11)$$

$$\frac{\gamma(s)}{\theta(s)} = \frac{-Z_w}{(s - Z_w)} \qquad (12)$$

IV. Aircraft Control System

In figure 4.1 a block diagram of typical aircraft system is shown [21]

The angle γ_0 represents the required glide path angle according to assigned landing path.

The error signal is ε is the difference between γ_0 and γ. δ_E is the command signal to the aircraft elevator.

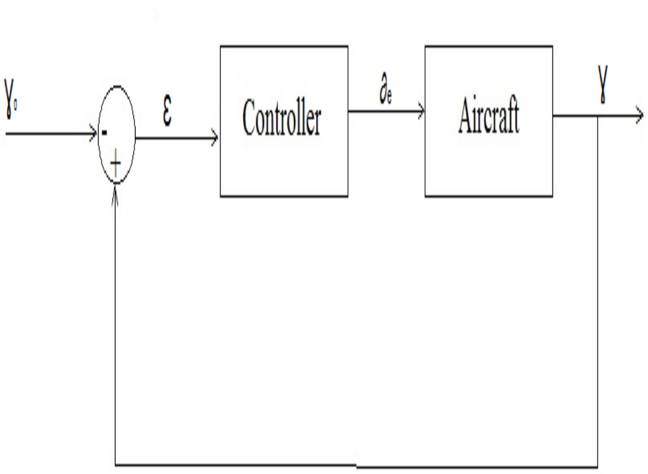

Figure 4.1 Block diagram of aircraft control system.

A. CHARLIE Aircraft Mathematical Model:

CHARLIE Aircraft: is a very large, four – engine passenger jet aircraft [21].

Table 4.1 Longitudinal Motion Stability derivative

Z_w	$Z_{\delta E}$	M_w	M_w	M_q	$M_{\delta E}$
-0.512	-1.96	-0.006	-0.0008	-0.357	-0.378

From eq. (9)

$K_q = -0.181776$

$T_2 = 2.07085$

$\Delta_{sp}(s) = S^2 + 0.909\,S + 0.484$

$$\frac{\theta(s)}{\delta_E(s)} = \frac{-0.376\,(S+0.484)}{S(S^2+0.909\,S+0.484)}$$

$$\frac{\gamma(s)}{\theta(s)} = \frac{0.512}{(s+0.512)}$$

$$\frac{\gamma(s)}{\delta_E(s)} = \frac{-0.193\,(S+0.484)}{S(s+0.512)(S^2+0.909\,S+0.484)}$$

$$\text{Actuator} = \frac{-7S^2-4.2\,S-0.35}{S^2+10\,S}$$

The open loop transfer function CHARLIE aircraft is founded as:

$$\left(\frac{-7S^2-4.2\,S-0.35}{S^2+10\,S}\right)\left(\frac{-0.193\,(S+0.484)}{S(s+0.512)(S^2+0.909\,S+0.484)}\right)$$

V. Simulation and Results

Controller gains, dynamic performances and performance index are summarized in table 5.1 and table 5.2. In table 5.1 controller gains in first row are designed by the classical approach (Z-N method). Controller gains in the last row are obtained by PSO method. The controller performance of the system is shown in figure 5.1. From figure 5.1 and table 5.2 ALS for Charlie Aircraft with PSO method shows better performance than classical PID.

Table 5.1 Controller gains

Tuning Method	PID Parameters		
	K_p	K_i	K_d
Z-N	18	11.25	7.2
PSO	20.39	0.868	45.03

Table 5.2 Controller gains, Dynamic Performance Specifications and Performance Index

Tuning Method	Dynamic Performance Specifications			Performance Index
Z-N	Rising Time (Tr)	Settling Time (Ts)	Percent Overshoot (% OS)	ITAE
	0.625	14.4	59.6	11.92
PSO	0.42	0.87	7.493	0.313395

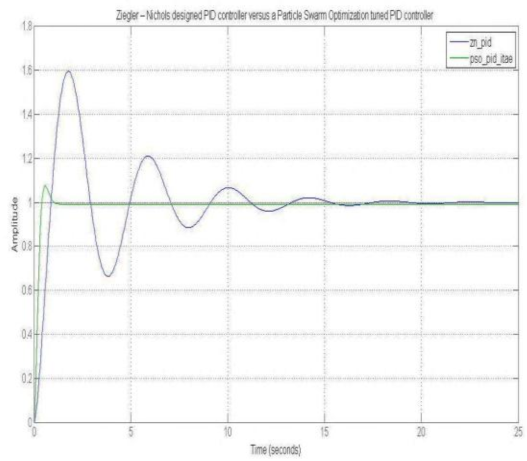

Figure 5.1 distinguishes the step response of the Ziegler – Nichols designed PID controller versus a Particle Swarm Optimization tuned PID controller using ITAE.

VI. Conclusion

This paper presents design of ALS for Charlie aircraft using PID controller parameters based on Particle Swarm Optimization method. From analysis of design results, we conclude that the proposed method is more efficient in ALS design problem. It is clear from results that the proposed PSO method is better than classical Z-N method and obtains higher quality solution with better computation efficiency.

References:

[1] Charles C. Jorgensen, C. Schley, A neural network baseline problem for control aircraft flare and touchdown, Neural Networks Control (1991), pp. 403-425.

[2] S. Asai, et al., Development of flight control system for automatic landing flight control experiment, Mitsubishi Heavy Ind. Tech. Rev. 34 (3) (1997).

[3] C. E. Cohen, et al., Automatic landing of a 737 using GNSS integrity beacons, Proc. ISPA (1995).
[4] DDC-I, Advanced auto landing system from Swiss Federal Aircraft factory, Real-Time J. Sprint (1995).
[5] D. N. Kaufmann, B. D. McNally, Flight test evaluation of the Stanford University and United Airlines differential GPS category III automatic landing system, NASA Technical Memorandum 110355, (1995).
[6] D. K. Chaturvedi, R. Chauhan, P. K. Kalra, Application of generalized neural network for aircraft landing control system, soft compute. 6 (2002), pp. 118-441.
[7] M. G. Cooper, Genetic design of rule – based fuzzy controllers, Ph.D. thesis, University of California, los Angeles, 1995.
[8] H. Izadi, M. Pakmehr, N. Sadati, Optimal neuro-controller in longitudinal auto landing of a commercial jet transport, in proceeding of IEEE International Conference on Control Applications, Istanbul, Turkey (2003).
[9] Y. Liguni, H. Akiyoshi, N. Adachi, An intelligent landing system based on human skill model, IEEE Trans. Aeros. Elctr. Syst. 34 (3) (1998), pp. 877-882.
[10] Kennedy, J., Eberhart. R., Particle swarm optimization. Proceedings of the IEEE International Conference on Neural network, Perth, Australia (1995).
[11] Byoung MunMin, et al., Autopilot design using hybrid PSO-SQP algorithm, Springer (2007), pp. 596-604.
[12] Kennedy J., Russell R. C, and Shi Y., "Swarm Intelligence" The morgan Kaufman series in Evolutionary Computation, 2001.
[13] Kennedy J. and Eberhart R., "Particle swarm optimization", Proc. IEEE Int. Conf. Neural Networks, Vol. 4, Perth, Australia, pp. 1942-1948, 1995
[14] Clerc M., "The Swarm and the queen: towards a deterministic and adaptive particle swarm optimization", Proceedings of the Conference on Evolutionary Computation, pp. 1951-1957, 1999
[15] R. A. Krohling and J. P. Rey, Design of optimal disturbance rejection PID controllers using genetic algorithm. IEEE Trans. Evol. Comput., Vol 5. (2001), pp. 78-88.
[16] Y. Mitsukura, T. Yamamoto, and M. Kaneda, A design of self-tuning PID controllers using a genetic algorithm, in Proc. Amer. Contr. Conf., San Diego, CA, June 1999, pp. 1361- 1365.
[17] T. Kawabe and T. Tagami, A real coded genetic algorithm for matrix in equality design approach of robust PID controller with two degree of freedom, in Proc. 12th IEEE Int. Contr., Istanbul, Turkey. July 1997, pp. 119-124.
[18] R. A. Krohling, H. Jaschek, and J. P. Rey, Designing of PI/PID controller for a motion control system based on genetic algorithm, in Proc. 12th IEEE Int. Contr., Istanbul, Turkey. July 1997, pp. 125-130.
[19] N. Pillay, A particle swarm optimization approach for tuning of SISO PID control loops, M. Sc. Thesis, University of Durban 2008.
[20] Donald McLean. Automatic flight control systems. Prentice hall international, 1990.
[21] M. S. Aly, Automatic landing system integrated with GPS, INS and MLS, M. Sc. Thesis, Military Technical College. 1998.