Controlling the Angle of Attack of an Aircraft Using Genetic Algorithm Based Flight Controller

S. Swain, P. S Khuntia

***Abstract*—**In this paper, the unstable angle of attack of a FOXTROT aircraft is controlled by using Genetic Algorithm based flight controller and the result is compared with the conventional techniques like Tyreus-Luyben (TL), Ziegler-Nichols (ZN) and Interpolation Rule (IR) for tuning the PID controller. Also the performance indices like Mean Square Error (MSE), Integral Square Error (ISE), and Integral Absolute Time Error (IATE) etc are improved by using Genetic Algorithm. It was established that the error by using GA is very less as compared to the conventional techniques thereby improving the performance indices of the dynamic system.

***Keywords*—** Angle of Attack, Genetic Algorithm, Performance Indices, PID Controller.

I. Introduction

GENERALLY an aircraft flies with a three dimensional plane by controlling its control surfaces aileron, rudder and elevator. These control surfaces control and change the motions of the aircraft about the roll, pitch and yaw axes. Elevators in an aircraft control the orientation of the aircraft by changing the pitch and angle of attack. Therefore it is required to control the angle of attack for better performance. Genetic Algorithms are applied in many fields for their outstanding performance in improving the response of a complicated dynamic system in last few decades. Among the various soft computing techniques, the most important techniques are the Genetic Algorithm which is most frequently used for their accuracy and easy of tuning. Chang, P.H. and Jung, J.H [1] suggested a systematic method for gain selection of a robust PI control for non linear plants of second order controller form. Gracey, W.[2] pointed out the summery of methods for measuring the angle of attack of an aircraft. Grimholt, C.[3] verified and improved the SIMC method for PI control. Haugen, F.[4] compared the PI tuning methods in a real benchmark temperature control system. Henrik, M.[5] extended the SIMC tuning rules to oscillatory and unstable processes. Jafarov, E.M., Parlakci, M.N.A. and Istefanopulos[6] suggested a new variable structure PID controller design for robot manipulators. Shamsuzzoha, M. and Skogestad, S.[7] analysed a simple and fast method for closed loop PID tuning. Wu, L., Wang, Y., Zhou, S. and Tan, W. [8] designed a a PID controller with incomplete derivation bsed on differential evolution algorithm. Yordanova, S. and Haralanova, E. [9] designed and implemented a robust multivariable PI like fuzzy logic controller for aerodynamic plant. H. Bevrani, S. Shokoohi [10] designed a robust stabilizer feedback loop for a radio-frequency amplifier. S. Skogestad [11] analysed a simple rule for model reduction and PID controller tuning. David Di Ruscio [12] suggested the tuning of PI controllers for integrating plus time delay systems. Ali, A., Majhi, S. [13] proposed a PI/PID controller design based on IMC and percentage overshoot specification to controller set point change. Neath, M.J.; Swain, A.K.; Madawala, U.K.; Thrimawithana, D.J.[14] proposed an optimal PID controller for a bidirectional inductive power transfer system using multiobjective Genetic Algorithm. Devaraj, D.; Selvabala, B.[15] suggested a real coded Genetic Algorithm for real time tuning of PID controller in automatic voltage regulator system. Whidborne, J.F.; Istepanian, R.S.H.[16] proposed a Genetic Algorithm Approach for designing a finite precision controller structures.

In this paper, Genetic Algorithm is applied for controlling the angle of attack and improving the performance indices of the system and finally the results are compared with the conventional techniques like ZN, TL and IR. It was established that GA gives excellent results and improves the performance indices as compared to the conventional methods like Tyreus-Luyben (TL), Ziegler-Nichols (ZN) and Interpolation Rule (IR) for tuning the PID controller.

II. Block Diagram of Angle of Attack

The block diagram of the system with disturbance shown in the Fig.1 below describes the feedback control system of angle of attack of an aircraft.

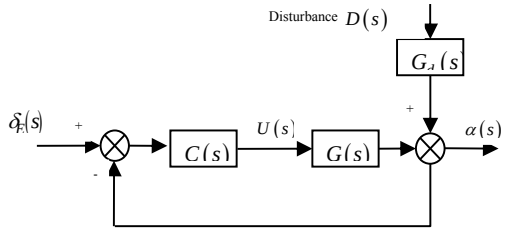

Fig. 1 Block diagram of angle of attack control system

Mr. S. Swain is with the Electrical Engineering of Synergy Institute of Technology, Bhubaneswar, Odisha, INDIA (phone: 0671-2356150; fax: 0671-2356150; e-mail: swainsrinibash@ gmail.com).

Dr. P. S. khuntia is with GMRIT, Rajam, Andhra Pradesh, INDIA. He is now with the Department of Electronics and Telecommunication, parthsarathi_K@yahoo.com).

Where

$\delta_E(s)$ = The deflection of elevator as commanded by the pilot

α = The actual angle of attack of the aircraft

$G(s)$ = The open loop transfer function between the deflection of the elevator δ_E and the angle of attack α

$C(s)$ = The PI controller to be designed (tuned)

$G_d(s)$ = The transfer function of the disturbance which occurs after some interval of the given input δ_E

Again, $G_d(s) = G(s)$

III. Relation between Angle of Attack and Elevator Deflection

Angle of attack [17] of an aircraft specifies the angle between the chord line of the wing of a fixed-wing aircraft and the vector representing the relative motion between the aircraft and the atmosphere. The angle of attack is controlled by the deflection in control surface (elevator). The description of angle of attack is shown in Fig. 2 below.

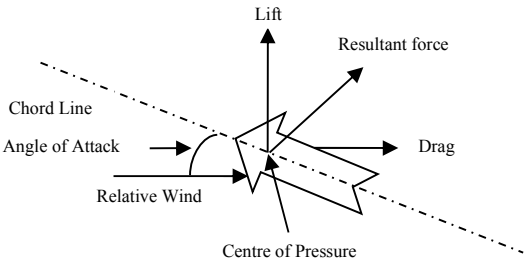

Fig. 2 Description of angle of attack

The short period approximation [17] consists of assuming that any variations in speed of the aircraft (*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 equation of motion involving '*u*' are negligible. In other words, the approximation assumes that short period transients are of sufficiently short duration that speed of the aircraft U_0 remain essentially constant, i.e., *u* = 0. Thus, the equations of longitudinal motion in terms of stability may now be written as:

$$\begin{aligned}\dot{w} &= Z_w w + U_0 q + Z_{\delta_E}\delta_E \\ q &= M_{\dot{w}} w + M_w \dot{w} + M_q q + M_{\delta_E}\delta_E \\ &= \left(M_{\dot{w}} + M_w Z_w\right) w + \left(M_q + U_0 M_w\right) q \\ &\quad + \left(M_{\delta_E} + Z_{\delta_E} M_{\dot{w}}\right)\delta_E\end{aligned} \tag{2}$$

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

$$x \square \begin{bmatrix} w \\ q \end{bmatrix}$$

and the control vector '*u*' is taken as the elevator deflection δ_E, then equations (1) and (2) may be written as a state equation:

$$\dot{x} = Ax + Bu$$

In equation (3), the values of *A* and *B* are

$$A = \begin{bmatrix} Z_w & U_0 \\ \left(M_w + M_{\dot{w}} Z_w\right) & \left(M_q + U_0 M_{\dot{w}}\right) \end{bmatrix}$$

$$B = \begin{bmatrix} Z_{\delta_E} \\ M_{\delta_E} + Z_{\delta_E} M_{\dot{w}} \end{bmatrix}$$

$$\therefore [sI - A] = s\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} Z_w & U_0 \\ \left(M_w + M_{\dot{w}} Z_w\right) & \left(M_q + U_0 M_{\dot{w}}\right) \end{bmatrix}$$

$$= \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} Z_w & U_0 \\ \left(M_w + M_{\dot{w}} Z_w\right) & \left(M_q + U_0 M_{\dot{w}}\right) \end{bmatrix}$$

$$= \begin{bmatrix} s - Z_w & -U_0 \\ -\left(M_w + M_{\dot{w}} Z_w\right) & \left[s - \left(M_q + U_0 M_{\dot{w}}\right)\right] \end{bmatrix}$$

$$\begin{aligned}\Delta_{sp}(s) &= \det[sI - A] \\ &= s^2 + \left[-\left(Z_w + M_q + U_0 M_{\dot{w}}\right)\right] s + \left[Z_w M_q - U_0 M_{\dot{w}}\right] \\ &= s^2 + 2\zeta_{sp}\omega_{sp} s + \omega_{sp}^2\end{aligned}$$

In equation (4)

$$\begin{aligned}2\zeta_{sp}\omega_{sp} s &= -\left(Z_w + M_q + U_0 M_{\dot{w}}\right) \\ \omega_{sp} &= \left[Z_w M_q - U_0 M_{\dot{w}}\right]^{1/2}\end{aligned} \tag{5}$$

$$\begin{aligned}\frac{w(s)}{\delta_E(s)} &= \frac{\left(U_0 M_{\delta_E} + M_q Z_{\delta_E}\right)\left\{1 + \dfrac{Z_{\delta_E}}{U_0 M_{\delta_E} - M_q Z_{\delta_E}} s\right\}}{\Delta_{sp}(s)} \\ &= \frac{K_w(1 + sT_1)}{\Delta_{sp}(s)}\end{aligned}$$

Where $K_w = \left(U_0 M_{\delta_E} + M_q Z_{\delta_E}\right)$

$$T_1 = \frac{Z_{\delta_E}}{K_w} \tag{1}$$

Again,

$$\dot{\alpha} = \frac{\dot{w}}{U_0}$$

$$\Rightarrow \alpha(s) = \frac{w(s)}{U_0}$$

$$\Rightarrow w(s) = U_0 \alpha(s)$$

TABLE I
STABILITY DERIVATIVES OF LONGITUDINAL DYNAMICS OF FOXTROT AIRCRAFT FOR VARIOUS FLIGHT CONDITIONS

Stability Derivatives	Flight Conditions (FC)		
	FC-1	FC-2	FC-3
$U_0\left(ms^{-1}\right)$	70	265	350
X_u	-0.012	-0.009	-0.0135
X_w	0.14	0.016	0.006
Z_u	-0.117	-0.088	0.0125
Z_w	-0.452	-0.547	-0.727
Z_q	-0.76	-0.88	-1.25
M_u	0.0024	-0.008	0.009
M_w	-0.006	-0.03	-0.08
$M_{\dot{w}}$	-0.002	-0.001	-0.001
M_q	-0.317	-0.487	-0.745
X_{δ_E}	1.83	0.69	0.77
Z_{δ_E}	-2.03	-15.12	-27.55
M_{δ_E}	-1.46	-11.14	-20.07

$$\therefore \frac{\alpha(s)}{\delta_E(s)} = \frac{\left(U_0 M_{\delta_E} + M_q Z_{\delta_E}\right)\left\{1 + \frac{Z_{\delta_E}}{U_0 M_{\delta_E} - M_q Z_{\delta_E}} s\right\}}{U_0 \Delta_{sp}(s)}$$

$$= \frac{K_w\left(1 + sT_1\right)}{U_0 \Delta_{sp}(s)}$$

$$\Rightarrow \frac{\alpha(s)}{\delta_E(s)} = \frac{K_w\left(1 + sT_1\right)}{U_0 \Delta_{sp}(s)}$$

A. Stability Derivatives of Longitudinal Dynamics of FOXTROT Aircraft

The stability derivatives of longitudinal dynamics of FOXTROT aircraft is shown in Table-I below. By using stability derivatives the transfer function for a particular flight condition can be determined.

B. Transfer Functions for Flight Condition-1

Using the stability derivatives of longitudinal dynamics of FOXTROT as shown in Table-1 above and substituting the values in equation (6), the transfer function G(s) between δ_E and α for flight condition-1 is given by

$$G(s) = \frac{2.0302s + 102.8}{s^2 + 0.901s + 0.5633}$$

$$= \frac{3.604s + 182.5}{1.7752s^2 + 1.5798s + 1}$$

IV. CONVENTIONAL METHODS FOR PID CONTROLLER TUNING

A. Zeigler-Nichols (ZN) Method

There are two versions of ZN method. Out of which one version depends on the reaction curve and the other version depends on ultimate gain K_u and the ultimate period P_u. Here, the 2nd version is considered. The values of K_p, T_i and T_d for 2nd version is shown in Table-II below.

TABLE II
VALUES OF K_p, T_i AND T_d FOR 2ND VERSION OF ZN TECHNIQUES

PID Type	K_p	T_i	T_d
PI	$0.45k_u$	$\frac{P_u}{1.2}$	0
PID	$0.6k_u$	$\frac{P_u}{2}$	$\frac{P_u}{8}$

B. Tyreus-Luyben (TL) Method

A ZN classical tuning constants are aggressive and oscillate. This results in a controller not very robust to model imprecise. Therefore, TL tuning constants are used for minor oscillations and robustness. The values of K_p, T_i and T_d for TL is shown in Table-III below.

TABLE III
VALUES OF K_p, T_i AND T_d FOR 2ND VERSION OF ZN TECHNIQUES

PID Type	K_p	T_i	T_d
PI	$\frac{k_u}{3.2}$	$2.2P_u$	0
PID	$\frac{k_u}{2.2}$	$2.2P_u$	$\frac{P_u}{6.3}$

C. Interpolation Rule

Let the controller be of the form

$$C(s) = K_C + \frac{K_I}{s} + K_D s$$

Where K_C, K_I and K_D are proportional, integral and derivative constants. The tuning parameter for an under damped second order process is as follows:

$K_C = \max\{A, X\}$, where $X = B$ for $\zeta \geq 1$

and $X = \zeta B' + (1-\zeta)C$ for $\zeta < 1$

$K_I = \max\{A, X\}$, where $X = B$ for $\zeta \geq 1$

and $X = \zeta B' + (1-\zeta)C$ for $\zeta < 1$

K_D = Either A, B or C

Where $A, B, B^{'}$ and C for proportional, integral and derivative controllers are given in Table-IV below:

TABLE IV
VALUES OF A , B, $B^{'}$ AND C FOR CALCULATION OF K_C , K_I AND K_D

	Calculation of K_C	Calculation of K_I	Calculation of K_D
A	$\frac{2\zeta}{k^{"}(\tau_c+\theta)\tau_0}$	$\frac{1}{k^{"}(\tau_c+\theta)\tau_0^2}$	$\frac{1}{k^{"}(\tau_c+\theta)}$
B	$\frac{1+4(\tau_c+\theta)+\frac{\left(\zeta+\sqrt{\zeta^2-1}\right)}{\tau_0}}{k^{"}(\tau_c+\theta)^2}$	$\frac{\left(\zeta+\sqrt{\zeta^2-1}\right)}{k^{"}(\tau_c+\theta)^2\tau_0}$	$\frac{1}{k^{"}(\tau_c+\theta)}$
$B^{'}$	$\frac{1+4(\tau_c+\theta)+\frac{\zeta}{\tau_0}}{k^{"}(\tau_c+\theta)^2}$	$\frac{\zeta}{k^{"}(\tau_c+\theta)\tau_0}$	$\frac{1}{k^{"}(\tau_c+\theta)}$
C	$\frac{1}{2k^{"}(\tau_c+\theta)^2}$	$\frac{1}{16k^{"}(\tau_c+\theta)^3}$	$\frac{1}{k^{"}(\tau_c+\theta)}$

$k^{"} = \frac{k}{\tau_0^2}$, where k is the gain

$\tau_0 = \frac{1}{\omega_n}$, where ω_n = Natural frequency of oscillation

ζ = Damping ratio

τ_c = The controller tuning parameter

$\theta = \tau_0(1.5+0.5\zeta)(0.6)^a$ = The delay angle

$a = \tau_0^2$

$B^{'}$ is obtained by setting $\sqrt{\zeta-1}=0$ in B.

D. Simulation Result

Time domain simulations of controller output 'u' and process output 'y' for ZN, TL and IR Techniques for the processes result from Flight Condition-1 of FOXTROT aircraft with setpoint and disturbances are shown below:

For Flight Condition-1

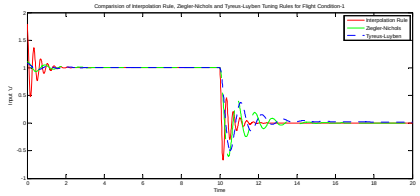

Fig. 3Time-domain simulations of controller output 'u' for the process g(s) = $\frac{k(3.604s+182.5)}{\tau_0^2 s^2+2\tau_0\zeta s+1}$ with conditions (k, τ_0, θ, ζ) = (1, 1.3324, 0.9683, 0.5996). Setpoint at t=0 and disturbance at t=10 time units.

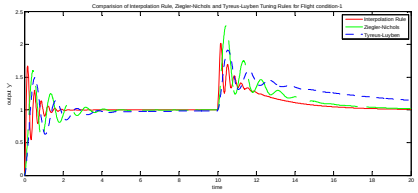

Fig. 4 Time-domain simulations of process output 'y' for the process g(s) = $\dfrac{k(3.604s+182.5)}{\tau_0^2 s^2 + 2\tau_0\zeta s + 1}$ with conditions (k, τ_0, θ, ζ) = (1, 1.3324, 0.9683, 0.5996). Setpoint at t=0 and disturbance at t=10 time units.

V. Genetic Algorithm (GA)

The genetic algorithm (GA) is a computer simulation that incorporates ideas from Darwin's theory on natural selection, and Mendel's work in genetics on inheritance, and it tries to simulate natural evolution of biological systems [18]. Darwin pioneered the idea that biological organisms develop and adapt over long periods of time via "descent with modification." The genetic algorithm is an optimization technique that evaluates more than one area of the search space and can discover more than one solution to a problem. It is also called stochastic direct search method because it provides a stochastic optimization method where, if it "gets stuck" at a local optimum, it tries, via multiple search points, to simultaneously find other parts of the search space and "jump out" of the local optimum to a global one that represents the highest fitness individuals. GA is a stochastic global adaptive search optimization technique based on the mechanisms of natural selection. GA starts with an initial population containing a number of chromosomes where each one represents a solution of the problem which performance is evaluated by a fitness function.

In this chapter, a PID controller is evolved by using a fitness function that quantifies closed-loop performance and is evaluated by repeated simulations. Each chromosome represents a point in the search space of the genetic algorithm. The term "phenotype" from biology to refer to the whole structure of the controller that is to be evolved. Hence, in this case the phenotype is given by

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

Where $e = \delta_E - \alpha$ = the error input to the PID controller

δ_E = The reference input

α = The output of the plant

A. The population of Individuals

A population is a set of candidate solutions (chromosomes). A notation is used for representing a whole set of individuals i.e. a population. Let k denote the generation number and $\theta_i^j(k)$ be a single parameter at time k. The superscript j refers to the j^{th} chromosome. A trait in a biological system is most often thought of as a property of the phenotype, like hair or eye color, whiles in our systems i.e. in the PID controller the genes, which are digits of parameters, are "expressed" as traits of the phenotype. Hence, the strings of genes are expressed as traits in the phenotype. Again the subscript i on $\theta_i^j(k)$ refers to the i^{th} trait on the j^{th} chromosome. Suppose the chromosome j is composed of p of these parameters (traits).

Let $\theta^j(k) = \left[\theta_1^j(k), \theta_2^j(k), ..., \theta_p^j(k)\right]^T$ be the j^{th} chromosome. The population of individuals at time *k* is given by $P(k) = \left\{\theta^j(k) : j = 1, 2, ..., S\right\}$. Where S= the number of individuals in the population.

B. Genetic Algorithm operators

Basically, GA consists of three main operators. These are Selection, Crossover and Mutation. The application of these three basic operators allows the creation of new individuals which may be better than their parents. This algorithm is repeated for many generations and finally stops when reaching individuals that represent the optimum solution to the problem.

1. Selection

The selection operator selects chromosomes from the current generation to be parents for the next generation. In this method, a few good chromosomes are used for creating new offspring in every iteration. Then some bad chromosomes are removed and the new offspring is placed in their places. The rest of population migrates to the next generation without going through selection process.

There are many ways to perform selection, but by far the most common one used in practice is fitness-proportionate selection.

1.1 Fitness Proportionate Selection

In this case, an individual is selected (the *ith* chromosome) for mating by letting each $m^j(k)$ be equal to $\theta^i(k) \in P(k)$ with probability

$$p_i = \frac{\bar{J}\left(\theta^i(k)\right)}{\sum_{j=1}^{S}\left(\theta^j(k)\right)} \tag{5.1}$$

Here the analogy of spinning a unit circumference roulette wheel is used where the wheel is divided like a pie into *S* regions where the i^{th} region is associated with the i^{th} individual of *P* (*k*). Each pie-shaped region has a portion of the circumference that is given by p_i in Equation (5.1). You spin the wheel, and if the pointer points at region *i* when the wheel stops, then you place θ_i into the mating pool *M*(*k*). You spin the wheel *S* times so that *S* elements end up in the mating pool and the population size stays constant.

Clearly, individuals who are more fit will end up with more copies in the mating pool. Hence, chromosomes with larger-than-average fitness will embody a greater portion of the next generation. At the same time, due to the probabilistic nature of the selection process, it is possible that some

relatively unfit individuals may end up in the mating pool M (k).

2. Reproduction Phase, Crossover

The crossover/reproduction operator computes two offspring for each parent pair given from the selection operator. The crossover operator is used to create new solutions from the existing solutions available in the mating pool after applying selection operator. This operator exchanges the gene information between the solutions in the mating pool. The most popular crossover selects any two solutions strings randomly from the mating pool and some portion of the strings is exchanged between the strings. The selection point is selected randomly. A probability of crossover is also introduced in order to give freedom to an individual solution string to determine whether the solution would go for crossover or not.

Crossover is the process of combining (mixing) chromosomes. The crossover operation operates on the mating pool M (k) by "mating" different individuals. First, the crossover probability" p_c (usually chosen to be near 1 since, when mating occurs in biological systems, genetic material is certainly swapped between the parents) is specified. There are many types of crossover (i.e., ways to swap genetic material on chromosomes), but the simplest one is "single-point" crossover.

2.1 Single-Point Crossover

The procedural steps for single-point crossover are:

Step-1

Randomly pair off the individuals in the mating pool M (k). There are many ways to do this. For instance, pick up each individual from the mating pool and then randomly select a different individual for it to mate with. Or, mate all the individuals with the ones that are right next to each other. In this approach, if there are an odd number of individuals in M (k), then you could simply take the last individual and pair it off with another individual who has already been paired off or you could pair it off with the individual with the highest fitness.

Step-2

Consider the chromosome pair θ^j, θ^i that was formed in Step-1.Generate a random number $r \in [0, 1]$.

(a) If $r < p_c$, then cross over θ^j and θ^i. To cross over these chromosomes, select a "cross site" at random and exchange all the digits to the right of the cross site of one string with those of the other. This process is shown in Fig.5.1below. In this example, the cross site is position 5 on the string and hence, swap the last eight digits between the two strings. Clearly, the cross site is a random number that is greater than or equal to 1, and less than or equal to the number of digits in the string minus 1.

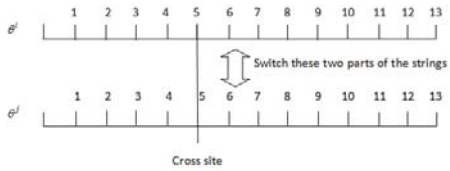

Fig. 6 Crossover operation

(b) If $r > p_c$ then not cross over and hence do not modify the strings and go to the mutation operation.

Step-3

Repeat Step-2 for each pair of strings in the mating pool M (k).

3. Reproduction Phase, Mutation

Mutations are global searches. A probability of mutation is again predetermined before the algorithm is started which is applied to each individual bit of each offspring chromosome to determine if it is to be inverted. Mutation changes the structure of the string by changing the value of a bit chosen at random. Mutation is the occasional introduction of new features in to the solution strings of the population pool to maintain diversity in the population. Mutation operator changes a 1 to 0 or vice-versa, with a mutation probability. The mutation probability is generally kept low for steady convergence.

Like crossover, mutation modifies the mating pool. The operation of mutation is normally performed on the elements in the mating pool after crossover has been performed. The biological analog of our mutation operation is the random mutation of genetic material. Again, there are many ways to perform mutations. The most common methods of mutation are described below.

3.1 Gene Mutations

To perform mutation in the computer, first choose a mutation probability p_m. With probability p_m change (mutate) each gene location on each chromosome randomly to a member of the number system being used.

For example, in a base-2 genetic algorithm, we can mutate

1010111
to
1011111

Where the fourth bit was mutated to one. For a base-10 number system, you would simply pick a number at random to replace a digit, if you are going to mutate a digit location. Usually, the mutation probability is chosen to be quite small (e.g., less than 0.01), since this will help guarantee that all the individuals in the mating pool are not mutated, so that any search progress that was made is lost i.e. it is kept relatively low to avoid degradation to exhaustive search via a random walk in the search space).

C. Computational Procedure of GA

The computational procedure for GA is stated below:

Step-1

Define the GA parameters (e.g. crossover and mutation probability, population size, termination parameters).

Step-2

Define the initial population *P* (0).

Step-3

For *k* = 1 to *Nga* (main loop for producing generations).Compute the fitness function for each individual.

Selection: From *P* (*k*), form *M* (*k*), the mating pool at iteration *k*, using fitness-proportionate selection.

Reproduction: For each individual in *M* (*k*), select another individual in *M* (*k*), mate the two via crossover with probability p_c, and mutate each gene position with probability p_m. Take the *S* children produced by this process, and put the children in *P* (*k* + 1), the next generation.

Step-4

Next *k* (i.e. return to step 3).

Step-5

Provide results.

D. Flow Chart of GA

The flow chart of Genetic Algorithm is shown in Fig.5.1 below.

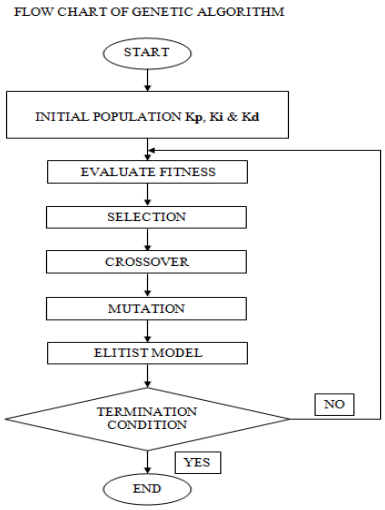

Fig. 7 Flow Chart of Genetic Algorithm

E. Block diagram of GA Based PID Controller

In this work, a genetic PID controller is designed to control the angle of attack of FOXTROT aircraft. The tuning of PID controller is governed by system nonlinearities and continuous parameter variations. The PID controller parameters K_p, K_i and K_d can be tuned with automatic tuning rules rather than conventional tuning rules such as Ziegler-Nichols, Cohen-Coon, Tyerus-Luyben etc. for better performances in many applications. Considerable revival of PID controller in the last decade is due to the introduction of automatic tuning and new approaches to automatically optimizing the performance of a dynamic system. A control problem is likely to have a unique optimum solution for the problem if it is specified by strict constraints. On the other hand, if some system specifications are left free to vary in a certain range several solutions might be possible for a control problem.

The use of GA tuning method as a powerful tool in the controller parameter design is implemented here as single objective function optimization. Most realistic optimization problems, particularly those in design require the simultaneous optimization of more than one objective function. Fig.5.2 below shows the block diagram of a GA based PID controller for controlling the angle of attack of an aircraft. Here the controlling parameters are K_p, K_i and K_d .

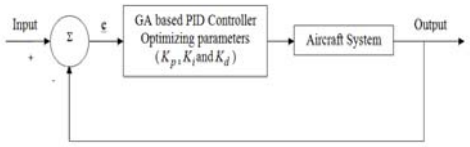

Fig.8 Block Diagram of GA Optimized PID Controller for Aircraft Control System

F. Simulation Results

The Matlab7.1 is used for simulation work. In this simulation the value of K_P, K_I, and K_D is calculated for 100 generations and the values for IAE are shown in Fig. 9 below.

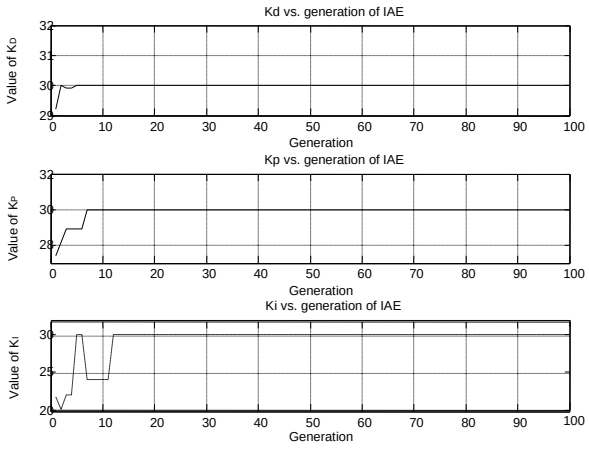

Fig. 9 The value of K_P, K_I, and K_D for IAE

Similarly, the values of K_P, K_I, and K_D for MSE and for IATE are calculated 100 generations and shown in Fig.10 and Fig. 11 below.

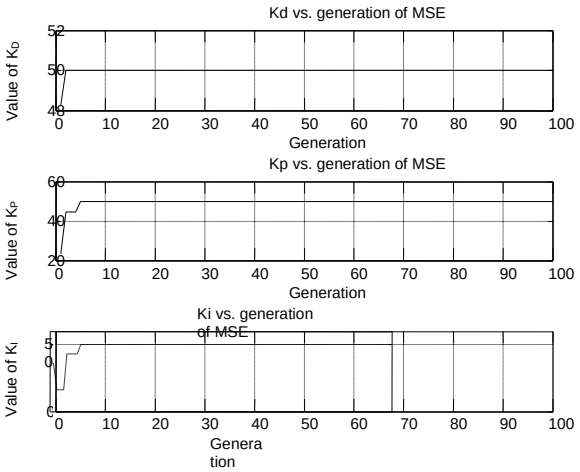

Fig. 10 The value of K_P, K_I, and K_D for MSE

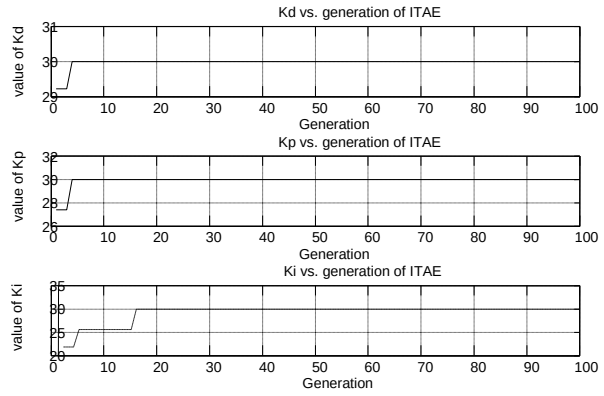

Fig. 11 The value of K_P, K_I, and K_D for MSE

IV. Comparison of (performance indices) soft computing technique with convensional techniques

The various types of performance indices like Integral Absolute Error (IAE), Mean Square Error (MSE) and Integral Time Absolute Error (IATE) are defined as

$$IAE = \int_0^\infty |e(t)|dt, MSE = \frac{1}{T}\int_0^\infty e^2(t)dt, IATE = \int_0^\infty t|e(t)|dt$$

Where the control error, $e = \alpha - \delta_E$.

The performance indices of various soft computing techniques like GA, FMRLC and RBFNC are compared with conventional techniques like ZN, TL and IR and the results are given in Table-V below.

TABLE V
Performance indices of a Dynamic System

Techniques	MSE	IAE	IATE
ZN	0.1311	57.3971	63.0637
TL	0.1256	53.4471	30.6712
IR	0.0570	27.3914	1.9471
GA	0.0015	0.1585	1.9471

It is clear from the above Table-V that the performance indices for using GA is improved i.e.it is very less as compared to the conversional methods like ZN, TL and IR

References

[1] Chang, P.H. and Jung, J.H.,"A systematic method for gain selection of robust PI control for nonlinear plants of second-order controller canonical form", IEEE Transactions on Control Systems Technology, Vol. 17, March, No. 2, 2009, pp.473-483.

[2] Gracey, W.,"Summary of methods of measuring angle of attack on aircraft", NACA Technical Note (NASA Technical Reports) (NACA-TN-4351), 1958, pp.1-30.

[3] Grimholt, C.,"Verification and improvement of SIMC method for PI control', Technical report", Department of Chemical Engineering, Norwegian University of Science and Technology, 2010.

[4] Haugen, F.,"Comparing PI Tuning methods in a real benchmark temperature control system', Modeling, Identification and Control", Vol. 31, No. 3, 2010, pp.79–91.

[5] Henrik, M.,"Extensions of Skogestad's SIMC tuning rules to oscillatory and unstable processes", 19 December, 2005, pp.1–65.

[6] Jafarov, E.M., Parlakci, M.N.A. and Istefanopulos, Y.,"A new variable structure PIDcontroller design for robot manipulators", IEEE Transactions on Control Systems Technology, Vol. 13, January, No. 1, 2005, pp.122–130.

[7] Shamsuzzoha, M. and Skogestad, S.,"The setpoint overshoot method: a simple and fast method for closed loop PID tuning", J. Process Control, Vol. 20, No. 10, 2010, pp.1220-34.

[8] Wu, L., Wang, Y., Zhou, S. and Tan, W.,"Design of PID controller with incomplete derivation based on differential evolution algorithm", Journal of Systems Engineering and Electronics, June, Vol. 19, No. 3, 2008, pp.578-583.

[9] Yordanova, S. and Haralanova, E.,"Design and implementation of robust multivariable PI like fuzzy logic controller for aerodynamic plant", IJAIP, Vol. 3, Nos. 3/4, 2011, pp.257-272.

[10] H. Bevrani, S. Shokoohi, "Robust stabilizer feedback loop design for a radio-frequency amplifier", IEEE International Conference on Control Applications, pp. 2250-2255, Sept. 2010.

[11] S. Skogestad, "Simple analytic rules for model reduction and PID controller tuning, Journal of Process Control", Vol. 13, pp. 291-309, July 2003.

[12] David Di Ruscio, "On Tuning PI Controllers for Integrating plus Time Delay Systems, Modeling, Identification and Control", Vol. 31, issue: 4, pp: 145-164, 2010.

[13] Ali, A., Majhi, S., "PI/PID controller design based on IMC and percentage overshoot specification to controller set point change" ISA Transactions, Vol.48, Issue: 1, Jan. 2009, Pages 10-15.

[14] Neath, M.J.; Swain, A.K.; Madawala, U.K.; Thrimawithana, D.J.,"An Optimal PID Controller for a Bidirectional Inductive Power Transfer System Using Multiobjective Genetic Algorithm",Power Electronics, IEEE Transactions on, Year: 2014, Volume: 29, Issue: 3, pp: 1523 – 1531.

[15] Devaraj, D.; Selvabala, B.,"Real-coded genetic algorithm and fuzzy logic approach for real-time tuning of proportional-integral - derivative controller in automatic voltage regulator system",Generation, Transmission & Distribution, IET,Year: 2009, Volume: 3, Issue: 7, pp: 641 – 649.

[16] Whidborne, J.F.; Istepanian, R.S.H.,"Genetic algorithm approach to designing finite-precision controllerstructures",Control Theory and

Applications, IEE Proceedings,Year: 2001, Volume: 148, Issue: 5, pp: 377 – 382.
[17] McLean, D., "Automatic Flight Control System", Prentice Hall International Ltd., UK, 1990, pp.17–79.
[18] Kelvin M. Passino,"Biomimicry for optimization, control and automation", Department of Electrical and Computer Engineering, 416 Dreese Laboratories, The Ohio State University, 2015 Neil Ave., Columbus, OH 43210, USA.

Mr. S. Swain was born on 21st May 1970 in Odisha, India. He completed his ME degree in Power Systwm Engg. in 2001 from VSSUT, Burla, Odisha India. Now he is continuing his Ph.D degree in BPUT, Rourkela, Odisha, India. His major field of study is Control System analysis and design.

He is now working as Associate Professor in Synergy Institute of Technology, BBSR since last one year. He has published three papers in International Journals. His current research interest is Soft Computing Application in the field of Control system.

Mr. Swain is a life member in ISTE. He is also a senior member in Institution of Engineers, India.

Dr. P. S. Khuntia was born on 06th oct. 1969 in Odisha, India. He completed his ME degree in Automatic Control System and Robotics from MS University, Varodara in 1999. He has completed his Ph.D degree from ISM Dhanbad in 2011. India. His major field of study is Intelligent Control, Digital Signal Processing and Soft Computing.

He is now working as Professor in GMRIT, Rajam, Srikakulum, Andhra Pradeesh. He has published many papers in National and International Journals. His current research interest is Advanced Control Theory and its application.

Dr. Khuntia is a life member in ISTE. He is also a senior member in Institution of Engineers, India.