

Improving the performance indices of a dynamic system using adaptive learning controllers

Srinibash Swain^{1,*}, Partha Sarathi Khuntia²

¹Faculty of Electrical Engineering, Bijupattnaik University of Technology, Bhubaneswar, India

²Faculty of Electronics and Telecommunications Engineering, Bijupattnaik University of Technology, Bhubaneswar, India

ARTICLE INFO

Article history:

Received 13 February 2017

Received in revised form

12 July 2017

Accepted 17 July 2017

Keywords:

Angle of attack performance indices

Adaptive controller

Learning controller

Radial basis function

ABSTRACT

In this paper, the angle of attack of an aircraft is controlled by using soft computing techniques like Genetic Algorithm (GA), Fuzzy Model Reference Learning Controller (FMRLC) and Radial Basis Function Neural Controller (RBFNC) and the performance indices like Mean Square Error (MSE), Integral Square Error (ISE), and Integral Absolute Time Error (IATE) etc. of the dynamic system is improved. The result is compared with the conventional techniques like Tyreus-Luyben (TL), Ziegler-Nichols (ZN) and Interpolation Rule (IR) for tuning the PID controller. It was established that the errors by using soft computing techniques are very less as compared to the conventional techniques thereby improving the performance indices of the dynamic system.

© 2017 The Authors. Published by IASE. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

An aircraft flies with a three dimensional plane by controlling its control surfaces such as aileron, rudder and elevator. These control surfaces control and change the motions of the aircraft about the roll, pitch and yaw axes. Elevators in an aircraft control the orientation of the aircraft by changing the pitch angle and angle of attack. Therefore it is required to control the angle of attack for better performance of the system. Soft computing techniques like Genetic Algorithm, Fuzzy Model Reference Learning Controller and Radial Basis Function Neural Controller etc. are applied in many fields for their outstanding performance in improving the response of a complicated dynamic system in last few decades. These soft computing techniques are used most frequently for their accuracy and easy of tuning a complicated system.

Chang and Jung (2009) proposed systematic methods for gain selection of robust PI control for nonlinear plants. Gracey (1985) highlighted the summary of methods of measuring angle of attack of an aircraft. Grimholt (2010) verified and made an improvement of SIMC method for PI control. Haugen (2010) compared the PI tuning methods in a real benchmark temperature control system.

Shamsuzzoha and Skogestad (2010) discussed the set point overshoot method for closed loop PID tuning. Skogestad (2010) pointed out the tuning for smooth PID control with acceptable disturbance rejection. Yordanova and Haralanova (2011) implemented the designed and implemented a robust multivariable PI like fuzzy logic controller for aerodynamic plant. Di Ruscio (2010) discussed on tuning PI controllers for integrating and time delay systems. Ali and Majhi (2009) designed a PID controller based on IMC and percentage overshoot specification to controller set point change. Xia and Wang (2004) proposed a recurrent neural network for nonlinear convex optimization. Neath et al. (2014) discussed about an optimal PID controller for a bidirectional inductive power transfer system using multiobjective Genetic Algorithm. Devaraj and Selvabala (2009) proposed the real coded genetic algorithm and fuzzy logic approach for real time tuning of PID controller. Seng et al. (1999) tuned a Neuro-fuzzy controller by Genetic Algorithm. Whidborne and Istefanian (2001) approached the Genetic algorithm to designing finite precision controller structures. Alfaro-Cid et al. (2006) proposed a GA optimized PID and pole placement real and simulated performance of a supply ship. Kwong et al. (1995) supervised expert fuzzy learning systems for fault tolerant air craft control. Dimeas and Aspragathos (2014) discussed a Fuzzy learning variable admittance control for human-robot cooperation. Baruch and Hernandez (2011) proposed a decentralized direct I-term Fuzzy – neural control of an anaerobic digestion bioprocess

* Corresponding Author.

Email Address: swainsrinibash@gmail.com (S. Swain)

<https://doi.org/10.21833/ijaas.2017.08.020>

2313-626X/© 2017 The Authors. Published by IASE.

This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

plant. Lin et al. (2009) discussed about the Recurrent Functional link based Fuzzy Neural Network Controller with Improved Particle Swarm Optimization for a Linear Synchronous Motor Drive.

Lian (2014) discussed an adaptive self-organizing fuzzy sliding mode Radial Basis Function Neural network controller for robotic system. In this paper, the above mentioned soft computing techniques are applied for controlling the angle of attack of a FOXTROT aircraft and the performance indices of the system is improved and finally the results are compared with the conventional techniques like ZN, TL and IR techniques.

It was established that the soft computing techniques 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.

2. Block diagram of angle of attack

The block diagram of the angle of attack control system with disturbance is shown in the Fig. 1.

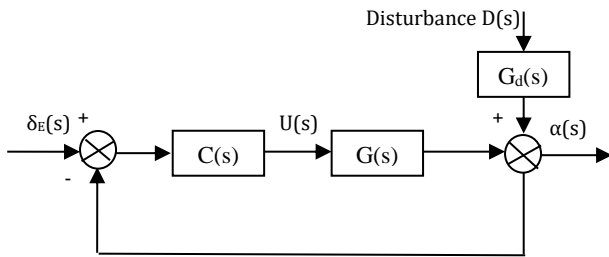


Fig. 1: Block diagram of angle of attack control system

Where,

δ_E = The deflection of elevator as commanded by the pilot;

α = The actual angle of attack of the aircraft;

$G(s)$ = The open loop transfer functions between the deflection of the elevator and the angle of attack;

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

$G_d(s)$ = Transfer functions of the disturbance which occurs after some interval of the given input;

$G_d(s) = G(s)$

3. Relation between angle of attack and elevator deflection

Angle of attack 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.

The short period approximation of a dynamic system 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.

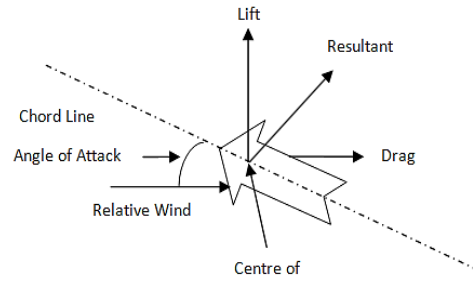


Fig. 2: Description of angle of attack

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 follows (Eqs. 1 and 2):

$$\dot{w} = Z_w w + U_0 q + Z_{\delta_E} \delta_E \quad (1)$$

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

If the state vector for short period motion is now defined as $x = \begin{bmatrix} w \\ q \end{bmatrix}$ and the control vector ' u ' is taken as the elevator deflection (δ_E), then Eqs. 1 and 2 may be written in the form of a state equation defined by (Eq. 3)

$$\dot{x} = Ax + Bu \quad (3)$$

in Eq. 3, the values of A and B are (Eq. 4)

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

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

now,

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

again,

$$\begin{aligned} \Delta_{SP}(s) &= \det[sI - A] \\ &= s^2 - [Z_w + M_q + U_0 M_{\dot{w}}]s + [Z_w M_q - U_0 M_{\dot{w}}] \\ &= s^2 + 2\zeta_{SP}\omega_{SP}s + \omega_{SP}^2 \end{aligned} \quad (4)$$

in Eq. 4,

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

On simplifying the above equations, the transfer function $\frac{w(s)}{\delta_E(s)}$ is given by (Eq. 6)

$$\begin{aligned} \frac{w(s)}{\delta_E(s)} &= \frac{(U_0 M_{\delta_E} + M_q Z_{\delta_E}) \left\{ 1 + \frac{s Z_{\delta_E}}{U_0 M_{\delta_E} - M_q Z_{\delta_E}} \right\}}{\Delta_{SP}(s)} \\ \frac{w(s)}{\delta(s)} &= \frac{K_w(1 + sT_1)}{\Delta_{SP}(s)} \end{aligned} \quad (6)$$

in Eq. 6,

$$T_1 = \frac{Z_{\delta_E}}{K_w}, K_w = U_0 M_{\delta_E} + M_q Z_{\delta_E}$$

again (Eq. 7),

$$\dot{\alpha} = \frac{\dot{w}}{U_0}, \alpha(s) = \frac{w(s)}{U_0} \quad (7)$$

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

Solving Eq. 6 and Eq. 7, the transfer functions between δ_E and α is given by (Eq. 8)

$$\frac{\alpha(s)}{\delta_E(s)} = \frac{K_w(1+sT_1)}{U_0 \Delta_{SP}(s)} \quad (8)$$

4. Stability derivatives of longitudinal dynamics of FOXTROT aircraft

The stability derivatives of longitudinal dynamics of FOXTROT aircraft is mentioned in Table 1 below. By using stability derivatives the transfer function for a particular flight condition can be determined.

Table1: Stability derivatives of longitudinal dynamics of FOXTROT aircraft for various flight conditions

Stability Derivatives	Flight Condition(FC)		
	FC-1	FC-2	FC-3
$U_0(ms^{-1})$	70	265	350
Z_w	-0.452	-0.547	-0.727
M_w	-0.006	-0.03	-0.08
$Z_{\dot{w}}$	-0.002	-0.001	-0.001
M_q	-0.317	-0.487	-0.745
Z_{δ_E}	-2.03	-15.12	-27.55
M_{δ_E}	-1.46	-11.14	-20.07

5. Transfer functions for FC-1 and FC-2

Using the stability derivatives of longitudinal dynamics of FOXTROT as mentioned in Table 1 above and substituting the values in Eq. 8, the transfer function $G_1(s)$ and $G_2(s)$ between δ_E and α for FC-1 and FC-2 is given by (Eqs. 9 and 10)

• For FC-1,

$$G_1(s) = \frac{2.0302s+102.8}{s^2+0.901s+0.5633} \quad (9)$$

$$= \frac{3.604s+182.5}{1.775s^2+1.598s+1}$$

• For FC-2,

$$G_2(s) = \frac{15.11s+0.003027}{s^2+1.2989s+8.216} \quad (10)$$

$$= \frac{1.84s+368.5}{0.1217s^2+0.1581s+1}$$

6. Conventional methods for PID controller tuning

6.1. 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 mentioned in Table 2.

Table 2: Values of K_p , T_i and T_d for 2nd version of ZN technique

PID Type	K_p	T_i	T_d
PI	$0.45K_u$	$P_u/1.2$	0
PID	$0.6K_u$	$P_u/2$	$P_u/8$

6.2. 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 mentioned in Table 3.

Table 3: Values of K_p , T_i and T_d for TL Technique

PID Type	K_p	T_i	T_d
PI	$K_u/3.2$	$2.2P_u$	0
PID	$K_u/2.2$	$2.2P_u$	$P_u/6.3$

6.3. Interpolation rule

Let the controller be of the form

$$C(s) = K_C + K_I/s + K_D s$$

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:

$$K_C = \max \{A, X\}, \text{ Where } X = B \text{ for } \zeta \geq 1 \text{ and } X = \zeta B + (1-\zeta) C \text{ for } \zeta < 1$$

$$K_I = \max \{A, X\}, \text{ Where } X = B \text{ for } \zeta \geq 1 \text{ and } X = \zeta B + (1-\zeta) C \text{ for } \zeta < 1$$

$$K_D = \text{Either } A, B \text{ or } C$$

where A , B , B' and C for proportional, integral and derivative controllers are given in Table 4 below:

Table 4: The values of A , B , B' and C for calculation of K_C , K_I and K_D

	K_C	K_I	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	$\frac{1+4(\tau_C+\theta)+\frac{\zeta+\sqrt{\zeta^2-1}}{\tau_0}}{k''(\tau_C+\theta)^2}$	$\frac{\zeta+\sqrt{\zeta^2-1}}{k''(\tau_C+\theta)^2\tau_0}$	$\frac{1}{k''(\tau_C+\theta)}$
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	$\frac{1}{2k''(\tau_C+\theta)^2}$	$\frac{1}{16k''(\tau_C+\theta)^3}$	$\frac{1}{k''(\tau_C+\theta)}$

In Table 4 above,

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

$$\tau_0 = \frac{1}{\omega_n},$$

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, $\sqrt{\zeta^2 - 1} = 0$ in B.

6.4. Simulation result

The time domain simulations of controller output 'u' and process output 'y' for ZN, TL and IR techniques are result from FC-1 and FC-2 of FOXTROT aircraft with setpoint and disturbances are mentioned in Fig. 3 and Fig. 4 and in Fig. 5 and Fig. 6 below:

• For FC-1

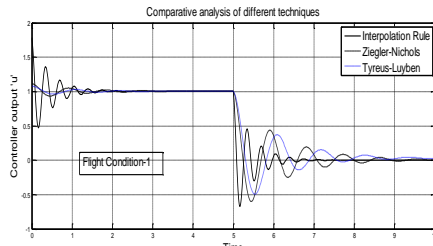


Fig. 3: Step response of controller output 'u' for FC-1 (Set point at $t = 0$ and disturbance at $t = 5$ sec.)

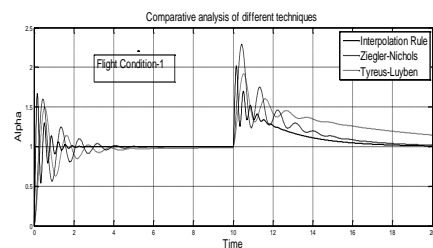


Fig. 4: Step response of angle of attack ' α ' for FC-1 (Set point at $t = 0$ and disturbance at $t = 10$ sec.)

• For FC-2

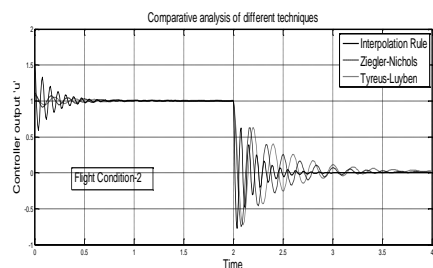


Fig. 5: Step response of controller output 'u' for FC-2 (Set point at $t = 0$ and disturbance at $t = 2$ sec.)

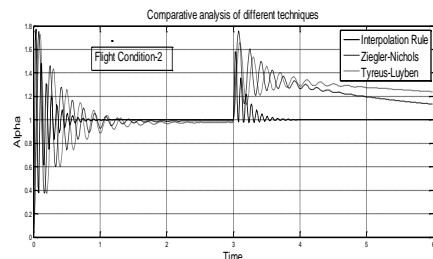


Fig. 6: Step response of angle of attack ' α ' for FC-2 (Set point at $t = 0$ and disturbance at $t = 3$ sec)

7. Soft computing techniques

The various soft computing techniques discussed here are Genetic Algorithm (GA), Fuzzy Model Reference Learning Controller (FMRLC) and Radial Basis Function Neural Controller (RBFNC).

7.1. 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.

In this chapter, a PID controller is evolved by using a fitness function that quantifies closed-loop performance and is evaluated by repeated simulations. 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

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

where

$e = \alpha - \delta_E$ = the error input to the PID controller
 δ_E = The reference input

7.1.1. The population of individuals

A population is a set of candidate solutions (chromosomes). 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 and 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 $\theta^j(k) = [\theta_1^k, \theta_2^k, \dots, \theta_p^k]^T$ be the j th chromosome. The population of individuals at time k is given by $P(k) = \{\theta^j(k) : j = 1, 2, \dots, S\}$. Where S = the number of individuals in the population.

7.1.2. 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.

7.1.2.1. 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. Here Fitness-Proportionate selection is used.

Fitness-Proportionate Selection

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

$$P_i = \frac{\bar{J}(\theta^i(k))}{\sum_{j=1}^S (\theta^i(k))}$$

7.1.2.2. Reproduction phase crossover

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. Here single point crossover is proposed.

Single-Point crossover

The Single-Point crossover operation is mentioned in Fig. 7.

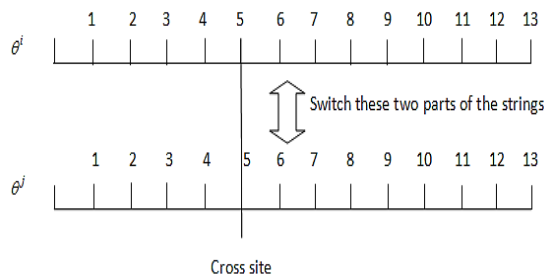


Fig. 7: Crossover operation

7.1.2.3. Reproduction phase, mutation

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 most common methods of mutation are described below.

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.

Usually, the mutation probability is chosen to be quite small (e.g., less than 0.01)

7.1.3. Flow chart of GA

The flow chart of Genetic Algorithm is shown in Fig. 8.

7.1.4. Block diagram of GA based PID controller

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. 9 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 .

7.1.5. 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 MSE are shown in Fig. 10 below.

FLOW CHART OF GENETIC ALGORITHM

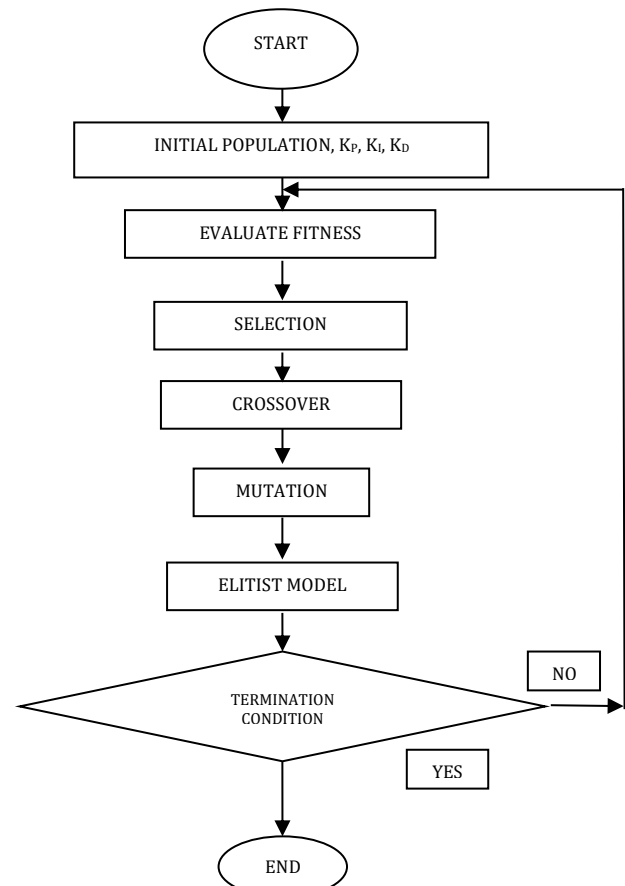


Fig. 8: Flow chart of genetic algorithm

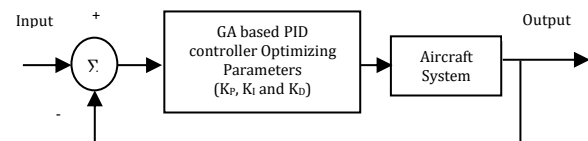


Fig. 9: Block diagram of GA optimized PID controller for aircraft control system

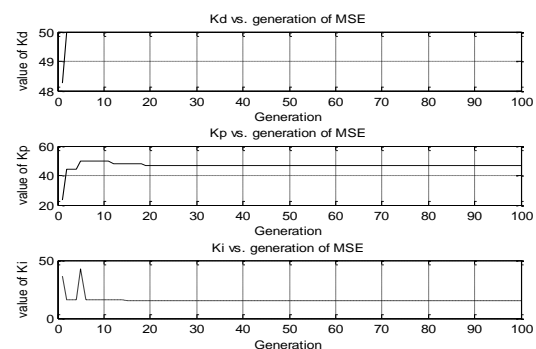


Fig. 10: The value of K_p , K_i , and K_d for MSE

Similarly, the values of K_P , K_I , and K_D for IAE and for ITAE are calculated 100 generations and depicted in Fig. 11 and Fig. 12.

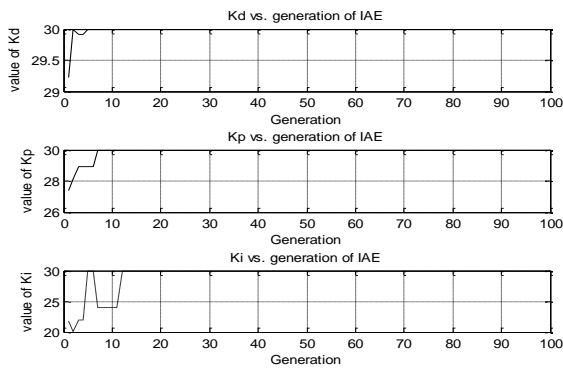


Fig. 11: The value of K_P , K_I , and K_D for IAE

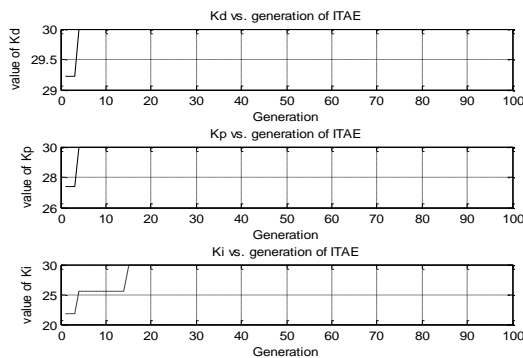


Fig. 12: The value of K_P , K_I , and K_D for ITAE

7.2. Fuzzy model reference learning controller (FMRLC)

Fuzzy Model Reference Learning Controller (FMRLC) utilizes a learning mechanism which observes the plant outputs and adjusts the rules in a direct fuzzy controller so that the overall system behaves like a reference model which characterizes the desired behavior. FMRLC improves the stability of a time-variant nonlinear system by tuning the fuzzy controller parameters. FMRLC for aircraft control system is mentioned in Fig. 13.

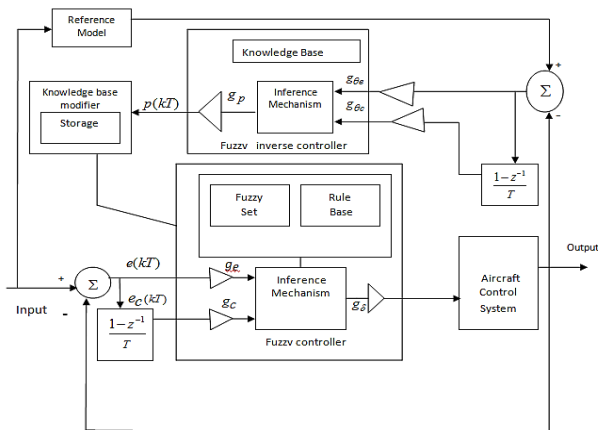


Fig. 13: FMRLC for aircraft control system

7.2.1. Simulation result of FMRLC

The simulation is done by using Matlab 7.1. The simulation is done by taking two cases into consideration. These are Case-I: Without Sensor Noise and Case-II: With Sensor Noise. The two cases are discussed below.

Case-I: Without sensor noise

In this case the reference signal is given for duration of 40sec. For first 25 second the the flight travels with flight condition-1 with a speed of 70m/sec and for next 15 second the flight travels with flight condition-2 with a speed of 265m/sec. Initially FMRLC has no adaptation but as the proceeds the controller gets adapted with changing the center of membership function.

Fig. 14a shows the angle of attack and desired angle of attack whereas Fig. 14b shows the elevator deflection input to the aircraft. Fuzzy inverse model output is shown in Fig. 14c. Fig. 14d and Fig. 14e shows the angle of attack error and the change in error respectively. Similarly Fig. 14f and Fig. 14g shows the angle of attack error and the change in error with respect to reference model.

Case -II: With sensor noise

In this case the reference model pulse duration is 40 second. Here a random noise is added uniformly with the Angle of attack by a random function

$$0.01 \frac{\pi}{180} (2 * rand - 1).$$

Now it is clear that the controller is noise adaptive. The response of the simulation with continuous noise is shown in Fig. 15. Here the adaptation takes place continuously due to continuous presence of random noise in sensor. It is clear from the following response that the controller is noise adaptive.

7.2.2. Control surface

The control surface of FMRLC without and with sensor noise is shown Fig. 16 and Fig. 17 respectively. It is clear from Fig. 16 and Fig. 17 that the control surface is nonlinear which is due to many factors like reference input, method of adaptation, dynamics of the system etc. the control surface nonlinearity changes with change in parameters of the system. It indicates the angle of attack error and change in error.

7.3. Radial basis function neural controller (RBFNC)

Artificial Neural Networks (ANN) during the last few decades has become popular for parameter identification, stability analysis; trajectory planning and motion control of robot. The idea of Radial Basis Function Neural Networks (RBFNN) derives from the theory of function approximation and takes a slightly different approach.

A locally tuned overlapping receptive field is found in parts of the cerebral cortex, in the visual cortex, and in other parts of the brain. The Radial Basis Function Neural Network model is based on these biological systems.

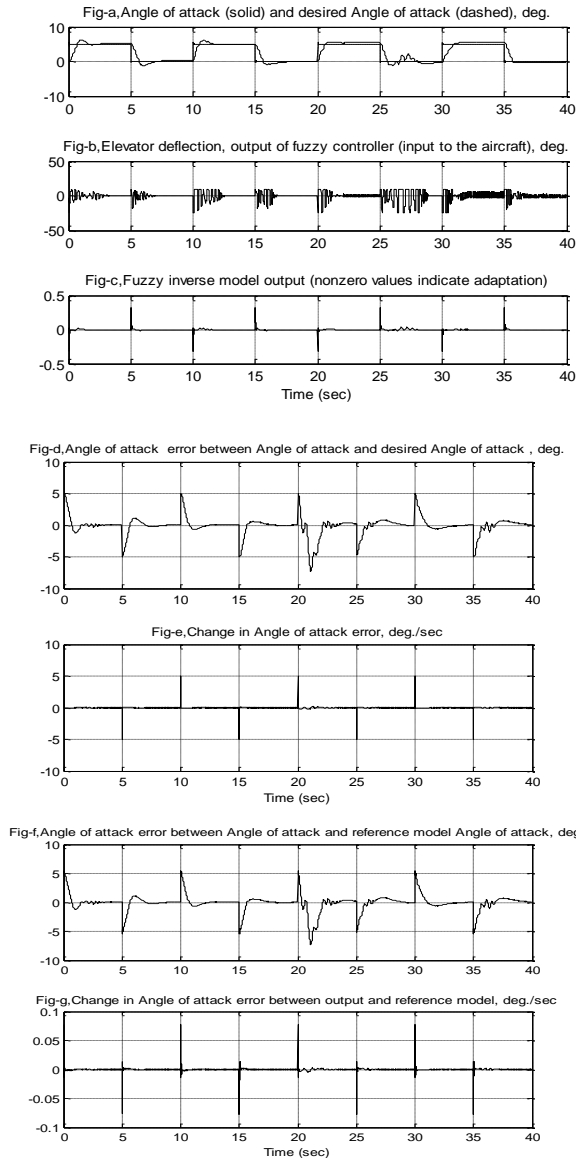


Fig. 14: Response without sensor noise

Here the inputs are $x_i, i = 1, 2, \dots, n$ and the output is $y = F_{rbf}(x)$ where F_{rbf} represents the processing by the entire radial basis function neural network. The input to the i th receptive field unit is x , and its output is denoted by $R_i(x)$ and the strength of the receptive field unit is b_i . Assuming b_{nr} receptive field units,

$$y = F_{rbf}(x, \theta) = \sum_{i=1}^{n_R} b_i R_i(x)$$

is the output of the radial basis function neural network, and θ holds the b_i parameters and the parameters of the receptive field units. Where

$$R_i(x) = \exp\left(-\sum_{j=1}^n \frac{(x_j - c_j^i)^2}{(\sigma_j^i)^2}\right)$$

where σ_j^i is the spread for the j th input for the i th receptive field unit.

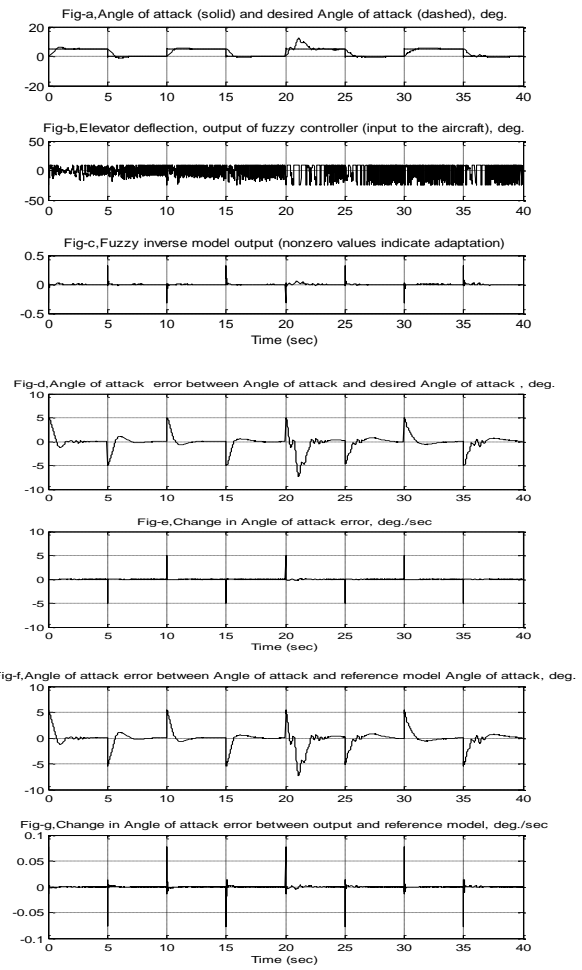


Fig. 15: Response with sensor noise

RBFNC is designed for controlling the angle of attack of a FOXTROT aircraft. Because of the quickened learning rate and avoidance of local minima, RBFNC is immensely fast growing topic of research interest and used in the fields of aircraft, robotics and other fields of control. RBFNC for aircraft control system is depicted in Fig. 18.

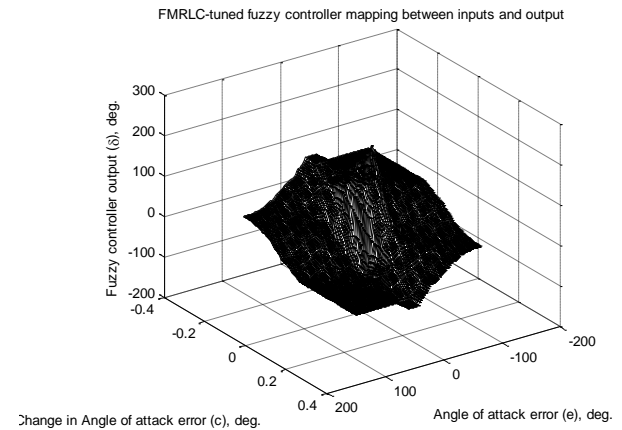


Fig. 16: Control surface without sensor noise

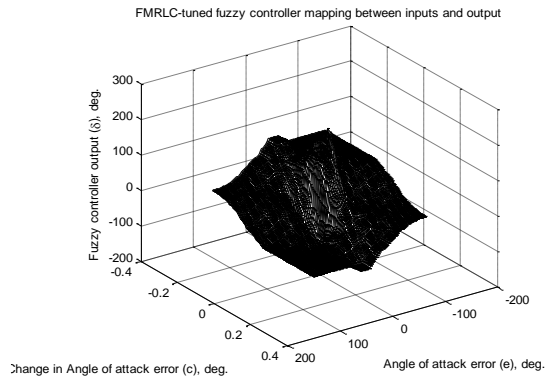


Fig. 17: Control surface with sensor noise

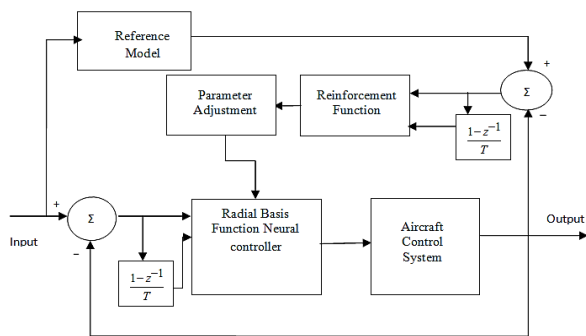


Fig. 18: RBFNC for aircraft control system

Therefore, it was established that the error and performance indices by using soft computing techniques are very less as compared to the conventional techniques like Tyreus-Luyben (TL), Ziegler-Nichols (ZN) and Interpolation Rule (IR) as discussed in 'Controlling the Angle of Attack of an Aircraft Using Extended SIMC and other Tuning Rules'.

7.3.1. Simulation result of RBFNC

The simulation is done by using Matlab 7.1. The simulation is done by taking two cases into consideration. These are (i) Case-I: Without Sensor Noise and (ii) Case-II: With Sensor Noise. The two cases are discussed below.

Case-I: Without sensor noise

In this case the reference signal is given for duration of 40sec. For first 25 second the the flight travels with flight condition-1 with a speed of 70m/sec and for next 15 second the flight travels with flight condition-2 with a speed of 265m/sec. Initially the RBFNC controller has no adaptation but when the time elapsed the controller gets adapted.

Fig. 19a shows the angle of attack and the desired angle of attack whereas Fig. 19b shows the elevator angle which is input to the aircraft. The reinforcement signal is shown in Fig. 19c. The angle of attack error and change in error is illustrated in Fig. 19d and Fig. 19e respectively. Similarly Fig. 19f and Fig. 19g shows the angle of attack error and change in error with respect to the reference model.

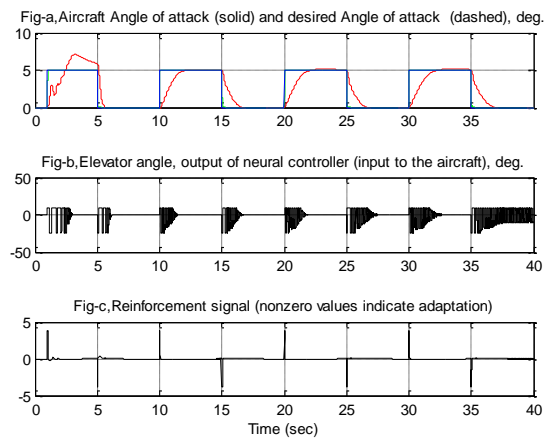


Fig. 19: Response of simulation with sensor noise

Case -II: With sensor noise

In this case the reference model pulse duration is 40 second. Here a random noise is added uniformly with the Angle of attack by a random function of $0.01 \frac{\pi}{180} (2 * rand - 1)$. Now it is clear that the controller is noise adaptive. The response of the simulation with continuous noise is shown in Fig. 20. It is clear from Fig. 20 that the controller is adaptive with the presence of continuous random noise.

7.3.2. Control surface of RBFNC

The control surface without sensor noise and with sensor noise is shown in Fig. 21 and Fig. 22 respectively. It is clear that the control surface changes with change in parameter of the system. The control surface with and without noise are different. Change in control surface is due to continuous adaptation of the RBFNC controller.

8. Comparison of (Performance Indices) Soft computing techniques with conventional 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 5.

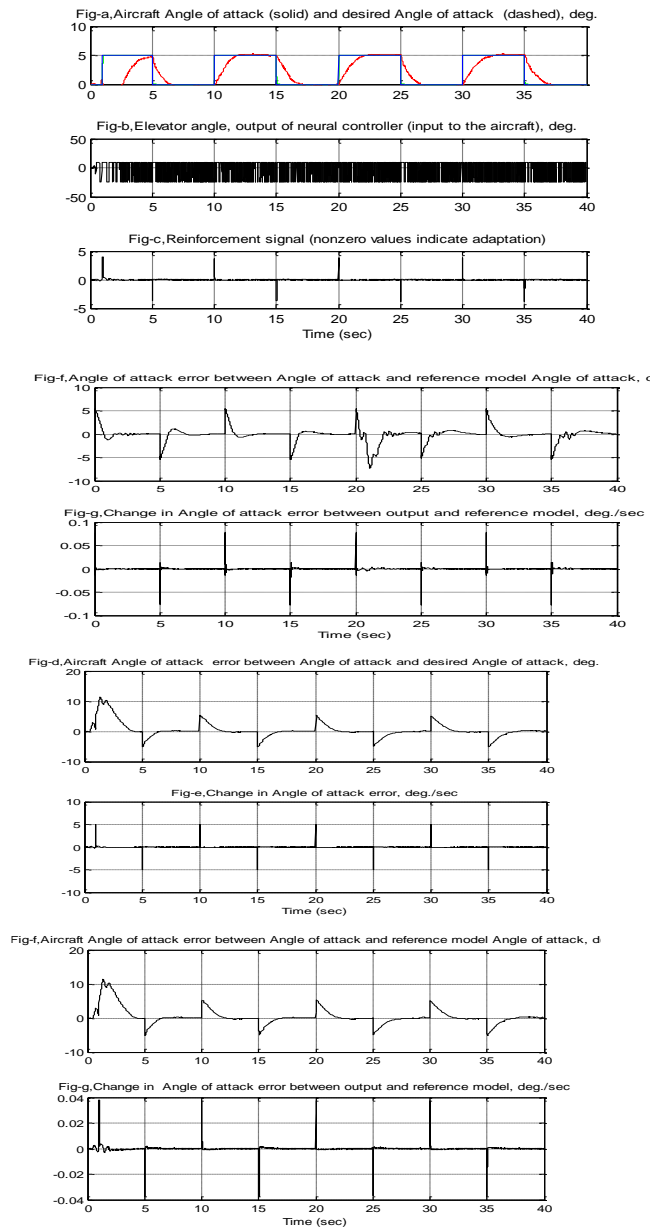


Fig. 20: Response of simulation with sensor noise

9. Conclusion

In this paper, various soft computing techniques like GA, FMRLC and RBFNC are applied for controlling the angle of attack of a FOXTROT aircraft and the performance indices like MSE, IAE and IATE of the system are compared with the conventional techniques like ZN, TL and IR techniques. It was

established that the soft computing techniques 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.

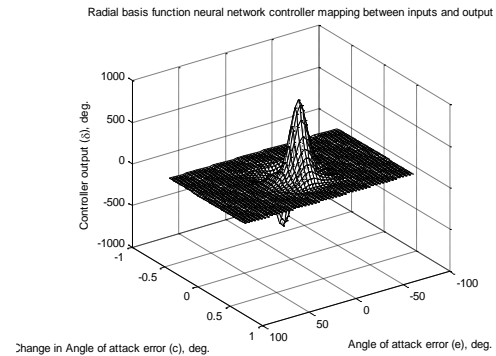


Fig. 21: Control Surface without sensor noise

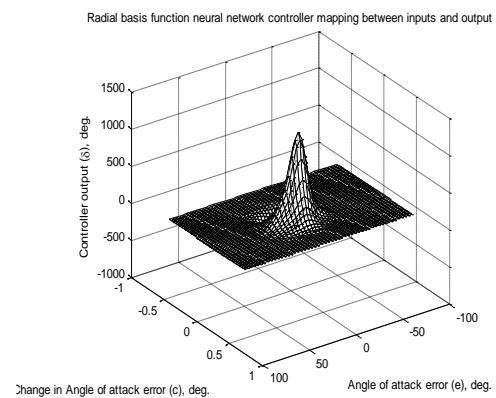


Fig. 22: Control Surface with a continuous sensor noise

Table 5: Simulation result of performance indices of various techniques

Techniques	Performance Indices		
	Mean Square Error (MSE)	Integral Absolute Error (IAE)	Integral Absolute Time Error (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	0.0190
FMRLC	0.0698	19.3787	27.1146
RBFNC	1.1387	16.3545	22.3596

References

- Alfaro-Cid E, McGookin EW, and Murray-Smith DJ (2006). GA-optimized PID and pole-placement real and simulated performance when controlling the dynamics of a supply ship. IEEE Proceedings on Control Theory and Applications, 153(2): 228-236.
- Ali A and Majhi S (2009). PI/PID controller design based on IMC and percentage overshoot specification to controller set point change. ISA Transactions, 48(1):10-15.
- Baruch IS and Hernandez S (2011). Decentralized direct I-Term fuzzy neural control of an anaerobic digestion bioprocess plant. Computational intelligence in control and automation. In the IEEE Conference on Computational Intelligence in Control and Automation, IEEE, Paris, France: 36-43. <https://doi.org/10.1109/CICA.2011.5945753>
- Chang PH and Jung JH (2009). 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*, 17(2):473-483.
- Devaraj D and Selvabala B (2009). Real coded genetic algorithm and fuzzy logic approach for real time tuning of proportional-integral-derivative controller in automatic voltage regulator system. *IET Generation, Transmission and Generation*, 3(7):641-649.
- Di Ruscio D (2010). On tuning PI controllers for integrating plus time delay systems. *Modeling, Identification and Control*, 31(4):145-164.
- Dimeas F and Aspragathos N (2014). Fuzzy learning variable admittance control for human-robot cooperation. In the *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Chicago, USA: 4770-4775. <https://doi.org/10.1109/IROS.2014.6943240>
- Gracey W (1985). Summary of methods of measuring angle of attack on aircraft. NACA Technical Note 4351. Available online at: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19930085167.pdf>
- Grimholt C (2010). Verification and improvement of SIMC method for PI control. Technical report 4550, Department of Chemical Engineering, Norwegian University of Science and Technology, Trondheim, Norway.
- Haugen F (2010). Comparing PI tuning methods in a real benchmark temperature control system. *Modeling, Identification and Control*, 31(3):79-91.
- Kwong WA, Passino KM, Laukonen EG, and Yurkovich S (1995). Expert supervision of fuzzy learning systems for fault tolerant aircraft control. *Proceedings of IEEE*, 83(3): 466-483.
- Lian RJ (2014). Adaptive self-organizing fuzzy sliding mode Radial basis function neural network controller for robotic systems. *IEEE Transactions on Industrial Electronics*, 61(3):1493-1503.
- Lin FJ, Chen SY, Teng LT, and Chu H (2009). Recurrent functional link based fuzzy neural network controller with improved particle swarm optimization for a linear synchronous motor drive. *IEEE Transactions on Magnetics*, 45(8): 3151-3165.
- Neath MJ, Swain AK, Madawala UK, and Thrimawithana DJ (2014). An optimal PID controller for a bidirectional inductive power transfer system using multi-objective genetic algorithm. *IEEE Transactions on Power Electronics*, 29(3):1523-1531.
- Seng TL, Khalid MB, and Yusof R (1999). Tuning of a neuro-fuzzy controller by genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(2): 226-236.
- Shamsuzzoha M and Skogestad S (2010). The Setpoint overshoot method: a simple and fast method for closed loop PID tuning. *Journal Process Control*, 20(10):1220-1234.
- Skogestad S (2010). Tuning for smooth PID control with acceptable disturbance rejection. *Industrial and Engineering Chemistry Research*, 45(23): 7817-7822.
- Whidborne JF and Istefanian RSH (2001). Genetic Algorithm approach to designing finite precision controller structures. *IEEE Proceedings Control Theory and Applications*, 148(5): 377-382.
- Xia Y and Wang J (2004). A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints. *IEEE Transactions on Circuits and Systems I*, 51(7):1385-1394.
- Yordanova S and Haralanova E (2011). Design and implementation of robust multivariable PI like fuzzy logic controller for aerodynamic plant. *International Journal of Advanced Intelligence Paradigms*, 3(3-4): 257-272.