Comparative Analysis of Different Fractional PID Tuning Methods for the First Order System

Sandip Mehta
Assistant Professor
Instrumentation and Control Engineering Section
Electrical Engineering Department, Institute of Technology
Nirma University, Ahmedabad, Gujarat, India
sandip.mehta@nirmauni.ac.in

Megha Jain
Research Scholar
Instrumentation and Control Engineering Section
Electrical Engineering Department, Institute of technology
Nirma University, Ahmedabad, Gujarat, India
13micc07@nirmauni.ac.in

***Abstract* – This paper deals with the use of fractional order PID (FOPID) controller for controlling the output response of first order systems. In this paper, the PID controller and FOPID controller has been tuned using various optimization techniques: fminsearch, fmincon, genetic and fuzzy. The simulation result shows that the FOPID controller performs better than PID controller in most cases.**

Keywords* – *Fractional order PID controller; Fuzzy logic; Genetic Algorithm

I. INTRODUCTION

The dynamics of many systems are represented by a simple model containing only one energy storing element; these systems are first order systems. Like DC motor, where we can consider speed as output and voltage as input. It is everywhere around us, it has numerous operations in industry as well as in commercial applications like robots, elevators etc. The speed control of DC motor is challenging task usually it is controlled by PI or PID controllers. Other examples include liquid level in tank, pressure control in gas tank. In this paper DC motor is controlled by unconventional way that is by fractional order PID (FOPID) controller. FOPID controller has developed during past some years and has various applications such as car suspension control, temperature control etc. The use of fractional calculus has been described in other works [1, 2, 3] also and its superiority in comparison to other controllers as well.

This paper is divided into following sections: Section 2 explains about fractional calculus, fractional order systems and fractional order controllers briefly. Section 3 shows various optimization techniques to tune fractional order PID controller and comparison of results. Section 4 concludes the paper.

II. FRACTIONAL CALCULUS

Fractional calculus is branch of mathematical analysis that takes the real number as power of differential operator and integration operator. There are many ways in which fractional integrator and differentiator can be defined; the most generalized differ-integrator can be shown as in equation 1.

$$ {}_aD_t^q f(t) = \frac{d^q f(t)}{[d(t-a)]^q} \qquad (1)$$

Where 'q' represents real order of differ-integral, 't' is parameter for which differ-integral is taken and 'a' is taken as lower limit. If lower limit is not stated, it will be left out to 0. The fractional order system takes the form of s^n or $1/s^n$; $n \in R^+$

The most commonly used definition for fractional order integral is given by Riemann-Liouville and is stated as anti-derivative of function f (t), shown in equation 2.

$$D^{-n} f(t) = \frac{1}{(n-1)!}\int_0^t f(y)(t-y)^{n-1}dy \quad ; n \in R^+ \qquad (2)$$

Similar to fractional integral, the fractional derivative is given by Grunwald-Letnikov, which is stated as in equation 3.

$$D^n f(t) = \frac{d^n f(t)}{dt^n} = \lim_{h\to 0}\frac{1}{h^n}\sum_{k=0}^{n}(-1)^n\binom{n}{k} f(t-kh) \ ; \qquad (3)$$
$n \in R^+$

Fractional Order PID Controller:

The FOPID controllers are expansion of conventional PID controller to non-integer form. It has five parameters, adding more flexibility and robustness to the system. The fractional PID controller is written as shown in equation 4.

$$K_p + K_i s^{-\lambda} + K_d s^{\mu} \quad ; (0 < \lambda, \mu < 1) \qquad (4)$$

Selection of λ and μ gives classical controller, λ=0 gives PD controller, μ=0 gives PI controller, λ=μ=1 gives PID controller.

III. TUNING METHODS OF FOPID CONTROLLER

Different tuning methods have been used to tune fractional order PID controller [4-9] using different systems. The system

978-1-4799-8433-6/15/$/31.00©2015IEEE

used in this paper is DC motor with first order transfer function. The block diagram for the whole process is shown as below in fig 1. Fractional PID Controller ($PI^{\lambda}D^{\mu}$) is tuned using different available optimization techniques, through which output of system is controlled.

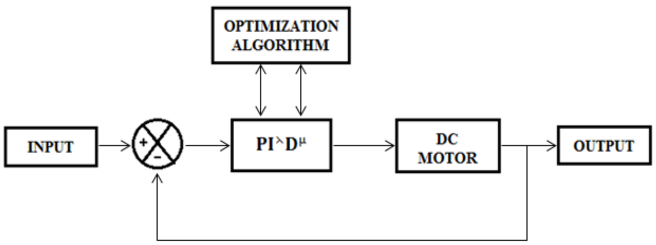

Fig 1: Block Diagram of System

DC MOTOR:

The speed of dc motor has to be controlled by applied input voltage. A constant applied voltage produces a constant torque and hence motor runs at a constant speed. The applied voltage is plant input. The first order transfer function of the dc motor used is:

$$G(s) = \frac{0.8333}{0.267s+1} \qquad (5)$$

OPTIMIZATION:

A. Fminsearch

The local search based optimization fminsearch, unconstrained nonlinear minimization, uses Nelder Mead's algorithm to find the minima of a given problem. It does not require the user to give gradients to find the solution. The user only needs to give the starting point to start the search process. ITAE criterion is used to reduce the error. The PID step response tuned using fminsearch with start point of [1, 1] is shown in fig 2.

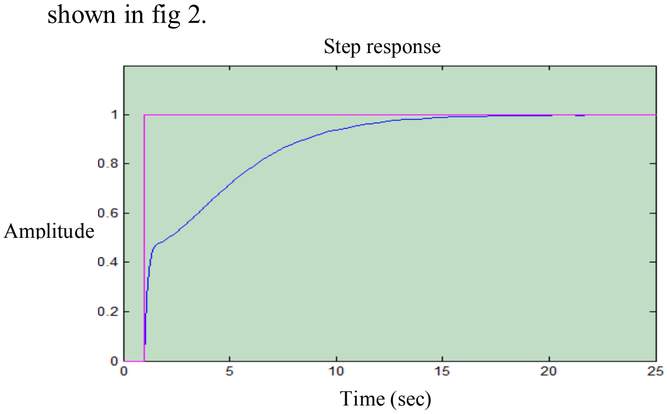

Fig 2: Step Response of PID, tuned using Fminsearch

The specifications obtained were:
Rise Time: 7.4496
Settling Time: 13.2420
Overshoot: 0.0193

The step response for FOPID tuned using fminsearch with start point of [0, 1] is shown in fig 3.

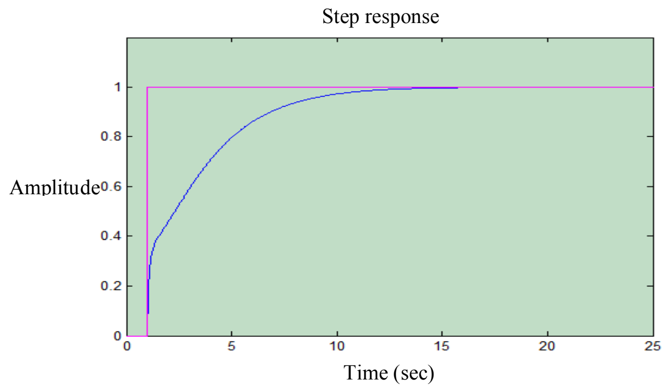

Fig 3: Step Response of FOPID, tuned using Fminsearch

The specifications obtained are:
Rise Time: 5.8068
Settling Time: 10.6561
Overshoot: 0.0031

B. Fmincon

Fminsearch is unconstrained nonlinear minimization, so there can't be bounds on the values of Kp, Ki, Kd, λ and μ. Whereas, Fmincon is constrained based nonlinear minimization in which there can be constraints of any type like equality constraint, bounds, inequality constraint. So by keeping bound on the values of Kp, Ki, Kd, λ and μ and giving the starting point to start the search process.

The step response obtained for PID tuned using fmincon is shown in fig 4.

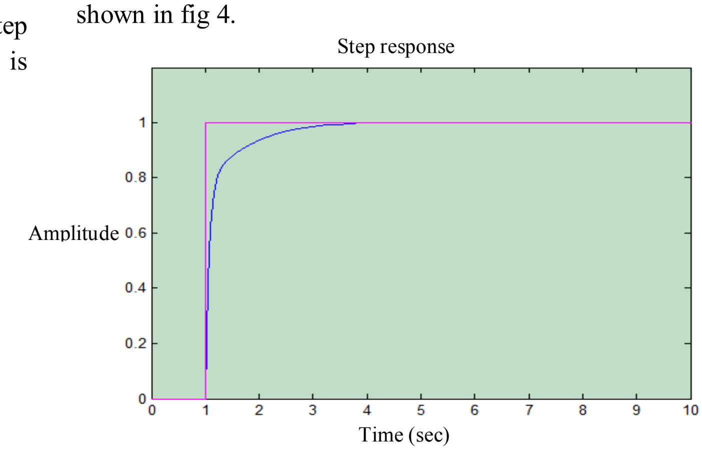

Fig 4: Step Response of PID tuned using Fmincon

The specifications obtained are:
RiseTime: 0.6376
SettlingTime: 2.7917
Overshoot: 0.1361

The step response for FOPID tuned using Fmincon is shown in fig 5.

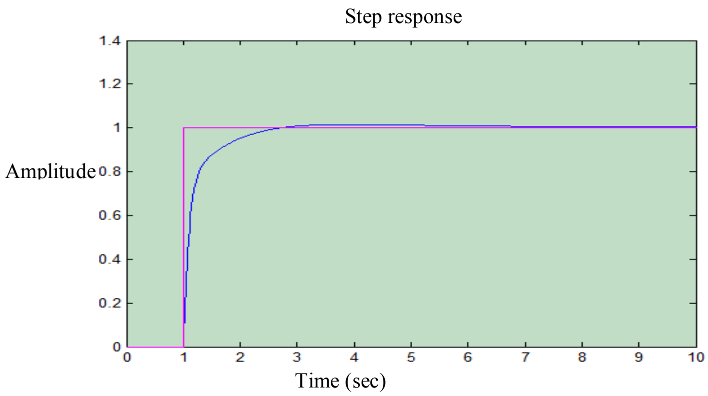

Fig 5: Step Response of FOPID, tuned using Fmincon

The specifications obtained are:
RiseTime: 0.6251
SettlingTime: 2.3768
Overshoot: 1.1538

The Fminsearch and Fmincon are local search methods, they will find local minima, for global minimization genetic algorithm or any other method should be used. The artificial intelligence method based Fuzzy logic is used here and after that genetic based tuning is also done.

C. Fuzzy Logic

Fuzzy control system employs human experience for the design of controller. Fuzzy set transform the input values in the range of 0 to 1 whereas crisp set gives value either 0 or 1. Fuzzy rules are made by human operators by using membership value in the range of 0 to 1. Fuzzy rules are in the form of “If A then B”.

The Fuzzy Inference System or Fuzzy Controller takes the crisp input, then fuzzification is done which converts crisp data into fuzzy data then based on rules and input, the output is obtained in fuzzy form, this is then converted to crisp output by defuzzification. The block diagram is as shown below:

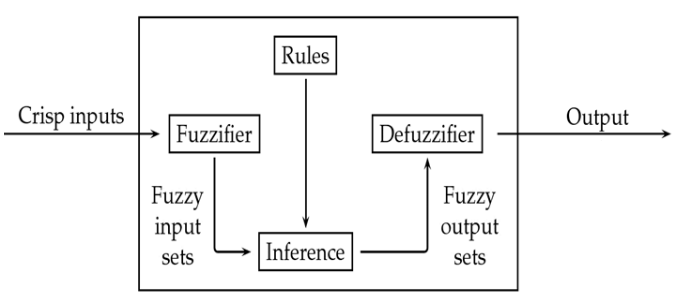

Fig 6: Block Diagram of Fuzzy Inference System

Fuzzy Based Tuning of FOPID:

Various works has been done using fuzzy in fractional order controller [2, 10, 11 and 12]. The system to be employed is as shown below, the five parameters to be tuned are obtained though fuzzy controller based on rules. As there are in total 7 variables, 2 inputs and 5outputs, 49 rules are made and based on that output is obtained. The inputs to fuzzy logic controller are error and derivative of error.

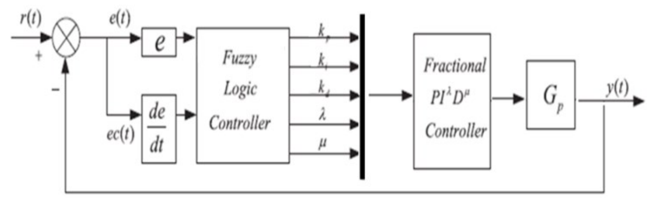

Fig 7: Block Diagram of Fuzzy System implemented

The membership function used for inputs can be shown as in fig.8 and fig.9

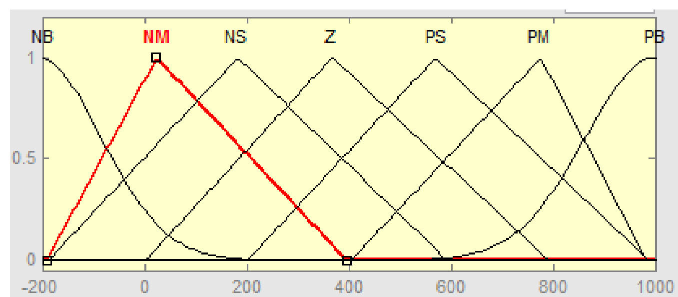

Fig 8: Input variable error (e)

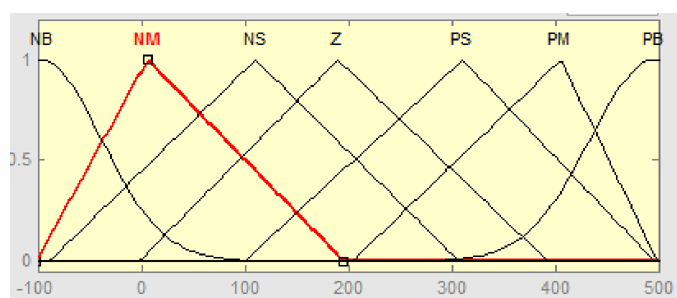

Fig 9: Input variable derivative of error (de/dt)

While defining membership function the range used for inputs and outputs are:
e: -200 to 1000
de/dt: -100 to 500
Kp: 0 to 200
Ki: 0 to 200
Kd: 0 to 20
⋋: 0 to 1
µ: 0 to 1

Here, ‘e’ represents error between input and output; ‘de/dt’ is derivative of error.

The step response for PID tuned using fuzzy logic is:

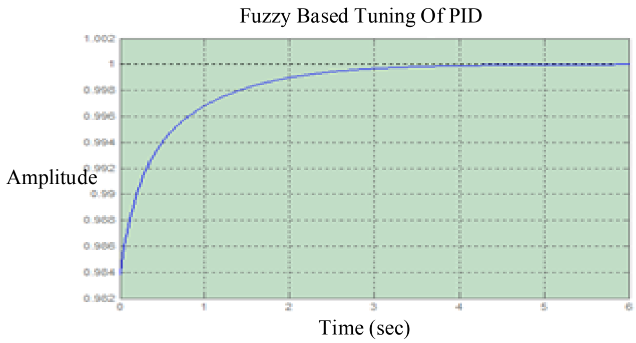

Fig 9: Step Response of PID, tuned by fuzzy controller

The specifications obtained are:
Rise Time: 1.56
SettlingTime: 3

The step response for FOPID tuned using fuzzy logic is:

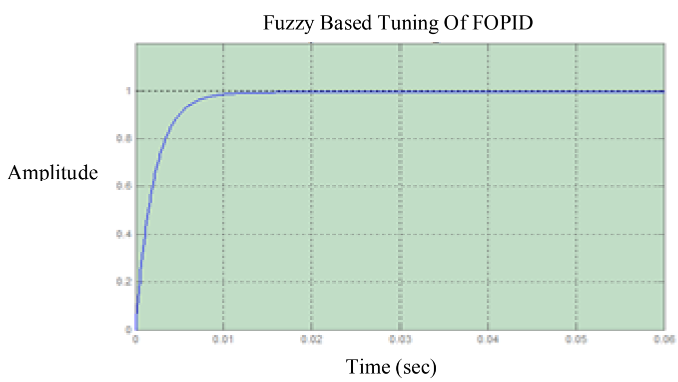

Fig 10: Step Response of FOPID, tuned using Fuzzy algorithm

The specifications obtained are:
Rise Time: 0.00471
Settling Time: 0.0089

D. Genetic Algorithm

Genetic Algorithm is used to solve global optimization problems. It is heuristic random search method inspired from Darwin’s evolution theory. It is based on the idea of naturally select, fitter survive

The benefit of using genetic algorithm is that it does not need know anything about the problem. It finds the solution of problem into chromosome and selects chromosome based on fitness, so that the good adaptability chromosome has more chances to reproduce. Genetic algorithm has been used to tune PID [1]. The genetic algorithm used in this work is based on Simple Genetic Algorithm (SGA), which was invented by Professor J.Hollan and it follows the steps shown in fig 11. The reason for using SGA is that while implanting genetic algorithm on hardware, the computation has to be reduced as ARM chip will be used which is not as powerful as computer’s CPU.

Terminology:

1. Chromosome: Also called as individuals, used to represent solutions.
2. Population: It is a group of individuals.
3. Fitness: It indicates the adaptive capacity of individual to the environment.
4. Elite: The best individual of each generation is known as elite.
5. Genetic Representation: It shows the operators to be used and the design of fitness evaluation function

The common methods used for encoding are: binary encoding method, floating encoding method and symbol encoding method. Usually, binary encoding is used as decoding operation is faster and crossover-mutation operations are easy to achieve.

Genetic Operators:

1. Selection: It selects the best individuals from current generation of population based on fitness of individuals.

2. Crossover: It randomly selects the points to splice the parent genes. The various crossover methods are:
   a) Single Point
   b) Two Point

The single point crossover can be shown as below.

Father 00000-01110001110001110
Mother 11111-00001110000011111

After Crossover

Offspring 00000-00001110000011111
Offspring 11111-01110001110001110

3. Mutation: It changes one or more gene values in an individual. It is an aiding method to generate new individuals and keeps the diversity of population.

Before Mutation: 000111 1 1011
After Mutation: 000111 0 1011

4. Terminal Condition: The genetic loop is repeated until terminal condition is reached. Generally the terminal conditions are:

- Reach the maximal generation
- Found the required individual

- Could not find required individual for many generations

Genetic Parameters:

a) Maximum Generation:

It is maximum number of generation, for example if maximum generation is 70, means genetic loop will be repeated for 70 times.

b) Population Size:

It is the number of individuals in the population.

c) Crossover Probability:

It generally takes a value from 0.6 to 0.95

d) Mutation Probability:

It can't be more than 0.5; else genetic algorithm will become a complete random search algorithm. The flow chart of genetic algorithm used is shown in fig 11.

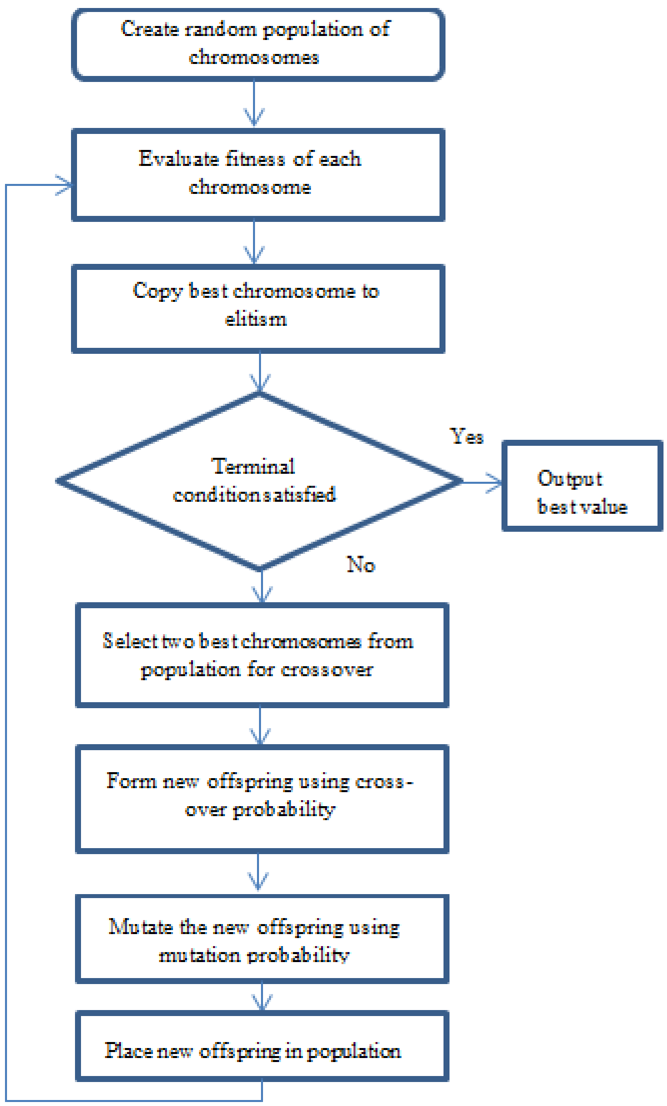

Fig 11: Flow chart of genetic algorithm

The best step response was achieved by taking following values in genetic programming.

Maximum Generation = 40

Population Size = 101

Perfect Fitness = 5500

Probability of Crossover = 0.8

Probability of Mutation = 0.2

Output obtained is: Kp, Ki, Kd, λ, μ = 5.3333, 9.3333, 3.7333, 1.000 and 0.3333

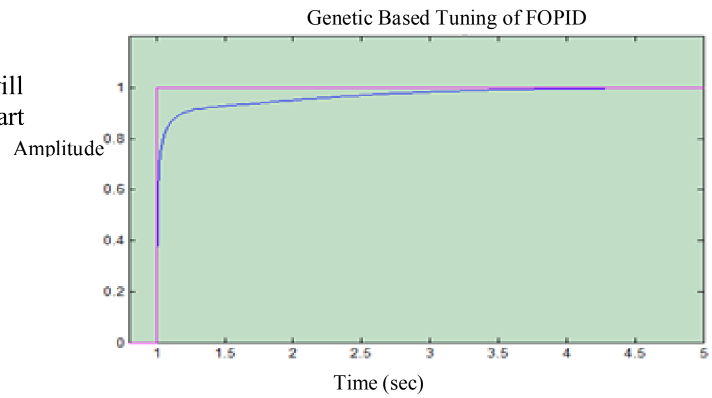

Fig 12: Step Response of FOPID tuned using genetic algorithm

The specifications obtained are:

Rise Time: 0.1857

Settling Time: 2.8277

Overshoot: 0.1263

COMPARISON OF RESULTS:

The various results obtained can be compared and summarized as shown in table 1.

TABLE I

COMPARISON OF RESULTS

Tuning Methods	Performance Specifications		
	Rise Time(s)	Settling Time(s)	Overshoot (%)
Fminsearch	5.8068	10.6561	0.31
Fmincon	0.6251	2.3768	11.538
Genetic	0.1857	2.8722	1.263
Fuzzy	0.0047	0.0089	0.00

As can be seen from experimental studies the genetic algorithm takes more time to settle to its steady state value, also Fmincon gives more overshoot as compared to others. In all, the fuzzy gives best result in all specifications.

IV. CONCLUSION

In this paper different methods have been used to tune PID controller as well as FOPID controller. The comparative analysis has been done for all the recently proposed tuning methods. The methods used include local minimization, global minimization as well as artificial intelligence method. The simulation results show that global minimization is better than local minimization and constrained minimization is better than unconstrained minimization. Also, fuzzy algorithm performs best in all the available methods for the first order system. Same analysis can be carried out for the second and higher order systems. Future work includes the real time implementation of the proposed methods.

REFERENCES

[1] Pengju Gong, "Automatic PID tuning based on Genetic Algorithm for Botnia Soccer Robots", Masters' Thesis, *Vaasan Ammattikorkeakoulu University of Applied Sciences*.

[2] Junran Jin, Hengshuo Huang, Junman Sun, Yongchao Pang, "Study on Fuzzy Self-Adaptive PID Control System of Biomass Boiler Drum Water", *Journal of Sustainable Bioenergy Systems*, vol.3, pp.93-98, 2013

[3] Vishal Mehra, Smriti Srivastava, and Pragya Varshney. "Fractional-Order PID Controller Design for Speed Control of DC Motor", *Third International Conference on Emerging Trends in Engineering and Technology*, 2010

[4] Concepcion Alicia Monje, YangQuan Chen, Blas Manuel Vinagre, Dingyu Xue, Vincente Feliu. *Fractional Order systems and Control – Fundamentals and Applications*, Springer, London, 2010

[5] Arijit Biswas, Swagatam Das, Ajith Abraham, Sambarta Dasgupta. "Design of fractional-order PID controllers with an improved differential evolution", *Engineering Applications of Artificial Intelligence,* vol. 22, pp. 343–350, 2009

[6] Saptarshi Das, Suman Saha, Shantanu Das, Amitava Gupta. "On the selection of tuning methodology of FOPID controllers for the control of higher order processes", ISA Transactions, vol.50, pp. 376–388, 2011

[7] Ying Luo, YangQuan Chen. "Fractional order [proportional derivative] controller for a class of fractional order systems", *Automatica*, vol.45, pp. 2446-2450, 2009

[8] Concepcion A. Monje, Blas M. Vinagre, Vicente Feliu, YangQuan Chen. "Tuning and auto-tuning of fractional order controllers for industry applications", *Control Engineering Practice*, vol. 16, pp. 798–812, 2008

[9] Duarte Valerio, Jose Sa da Costa. "Tuning of fractional PID controllers with ZieglerNichols-type rules", *Signal Processing,* vol. 86, pp. 2771–2784, 2006

[10] Jyh-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani, *Neuro-Fuzzy And Soft Computing, A computational approach to learning and machine intelligence,* Prentice-Hall Inc.,1997

[11] Saptarshi Das, Indranil Pan, Shantanu Das, Amitava Gupta. "A novel fractional order fuzzy PID controller and its optimal time domain tuning based on integral performance indices", Science Direct, *Engineering Applications of Artificial Intelligence,* vol. 25, pp. 430–442, 2012

[12] Lu Liu, Feng Pan, Dingyu Xue, "Variable-order fuzzy fractional PID controller", ISA Transactions, 2014

[13] Jyh-Shing Roger Jang, "ANFIS: Adaptive Network-Based Fuzzy Inference System", IEEE Transactions on Systems, Man, and Cybernetics, vol.23, No.3, May-June 1993