

PSO Based Optimal Design of Fractional Order Controller for Industrial Application

Rohit Gupta, Ruchika

Abstract—In this paper, a PSO based fractional order PID (FOPID) controller is proposed for concentration control of an isothermal Continuous Stirred Tank Reactor (CSTR) problem. CSTR is used to carry out chemical reactions in industries, which possesses complex nonlinear dynamic characteristics. Particle Swarm Optimization algorithm technique, which is an evolutionary optimization technique based on the movement and intelligence of swarm is proposed for tuning of the controller for this system. Comparisons of proposed controller with conventional and fuzzy based controller illustrate the superiority of proposed PSO-FOPID controller.

Keywords—CSTR, Fractional Order PID Controller, Particle Swarm Optimization.

I. INTRODUCTION

THERE is an increasing interest in dynamic systems of noninteger orders which are employed in extending the order of derivatives and integrals from integer to noninteger order & has a firm and long standing theoretical foundation.

In 1974, K. B. Oldham and J. Spanier presented theoretical and practical aspects of computing methods for mathematical modeling of nonlinear systems by considering number of computing techniques such as methods of operator approximation, operator interpolation techniques including a non-Lagrange interpolation [1]. The possible advantages of fractional order control in modeling and control design have motivated renewed interest in various applications of fractional order control [2], [3]. For the fractional order dynamic system modeling, control and filtering purpose, MATLAB is used [4]. A fractional order PID controller designing method is proposed by minimizing the integral of the squared errors [5]. Some examples of the fractional order PID control are seen in [6], [7]. In [8], a fractional controller was proposed to ensure that the closed-loop system is robust to gain variations and the step response exhibit an iso-damping property. A comparative introduction of four fractional order controllers can be found in [9]. Many researchers proposed various methods of tuning of fractional order controller like pole distribution of the characteristic equation in the complex plane [10], state space approach based on feedback pole placement [11], method using differential evolution (DE) technique [12] and using optimization method [13]. Some researchers also proposed optimal tuning of classical PID controller with various fitness functions but classical PID controller cannot achieve a high

quality solution that effectively improve the transient response of the system, Genetic algorithm and fuzzy logic is also used to optimize the parameters of controller [14]. Some group of researchers also proposed nature inspired optimization techniques to tune the controller, which are Bee Colony algorithms, Ant Colony algorithms and Particle Swarm Optimization technique [15]-[18]. In fractional order controllers, integral and derivative operations are usually of fractional order, therefore besides tuning the proportional (k_p), derivative (k_d) and integral (k_i) constants we have two more parameters: the power of s in integral and derivative actions λ and μ respectively. This adds flexibility and makes the system more robust, thus, enhancing its dynamic performance compared to its integer counterpart. Finding an optimal set of values for k_p , k_d , k_i , λ and μ to meet the specifications of user for a given process plant calls for real parameter optimization in five-dimensional hyperspace [19].

This paper is organized as follows. In Section II, the problem formulation is done, i.e. mathematical modeling of CSTR. Section III discusses a brief overview of fractional calculus especially fractional order PID controller. In Section IV, brief introduction to particle swarm optimization algorithm and proposed fitness function is discussed. In Section V, simulation results are presented and Section VI deals with the conclusion.

II. PROBLEM FORMULATION

Chemical reactors are the most important unit of chemical plants in the industries used for unit operations. Basically, a chemical reactor is a device in which chemical reaction takes place.

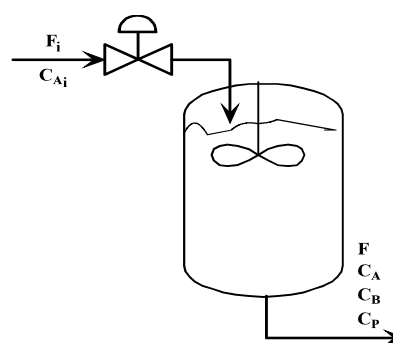


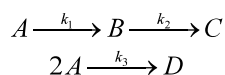
Fig. 1 Isothermal Stirred Tank Reactor

Rohit Gupta and Ruchika are with the Department of Electrical and Instrumentation Engineering at Thapar University, Patiala, Punjab, India. (e-mail: rohit.gupta@thapar.edu, ruchika.mehta@thapar.edu).

While designing a chemical reactor following factors have to be considered, (i) Overall size of reactor, (ii) Products emerging from reactor, (iii) Temperature inside the reactor (iv) Pressure inside the reactor (v) Rate of reaction (vi) Activity and mode of catalyst (vii) Stability and controllability of reactor.

Isothermal CSTR is a type of CSTR which is operating at a constant temperature. Here a constant volume, series-parallel reaction is assumed.

Consider an isothermal CSTR shown in Fig. 1. The following reaction scheme consists of the irreversible reactions, which is called Van de Vusse reaction:



The feed stream contains only component A which is cyclopentanol, the desired component is B which is cyclopentanol, the intermediate components are cyclopentanediol and dicyclopentadiene denoted by C and D respectively. For the above reaction the values of rate constant are

$$k_1 = 50h^{-1} = 0.83 \text{ min}^{-1}$$

$$k_2 = 100h^{-1} = 1.66 \text{ min}^{-1}$$

$$k_3 = 10 \text{ mol}^{-1} h^{-1} = 0.166 \text{ mol}^{-1} \text{ min}^{-1}$$

Steady state feed concentration is $C_{Afs} = 10 \text{ gmol}^{-1}$. Overall material balance is given by (1)

$$\frac{d(V\rho)}{dt} = F_i \rho_i - F\rho \quad (1)$$

Here assumption is made that the liquid phase density ρ is not a function of concentration $\rho = \rho_i$. So,

$$\frac{dV}{dt} = F_i - F \quad (2)$$

Let C_A and C_B represent the molar concentration A and B in moles/volume, component material balance can be written as (3):

$$\frac{d(V C_A)}{dt} = F(C_{Afs} - C_A) - V k_1 C_A - V k_3 C_A^2 \quad (3)$$

Simplifying (3) we obtain (4)

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{Afs} - C_A) - r_A \quad (4)$$

Similarly, for B, C and D

$$\frac{dC_B}{dt} = -\frac{F}{V} C_B + r_B \quad (5)$$

$$\frac{dC_C}{dt} = -\frac{F}{V} C_C + r_C \quad (6)$$

$$\frac{dC_D}{dt} = -\frac{F}{V} C_D + r_D \quad (7)$$

For each component the molar rate of formation (per unit volume) is given as:

$$r_A = -k_1 C_A - k_3 C_A^2 \quad (8)$$

$$r_B = k_1 C_A - k_2 C_B \quad (9)$$

$$r_C = k_2 C_B \quad (10)$$

$$r_D = \frac{1}{2} k_3 C_A^2 \quad (11)$$

It is noticed that (4) and (5) do not depend on the concentration of components C or D, since for this particular problem component B is our concern hence only (4) and (5) are considered hence the two dynamic functional equations can be represented as:

$$\frac{dC_A}{dt} = f_1\left(C_A, C_B, \frac{F}{V}\right) = \frac{F}{V}(C_{Afs} - C_A) - k_1 C_A - k_3 C_A^2$$

$$\frac{dC_B}{dt} = f_2\left(C_A, C_B, \frac{F}{V}\right) = -\frac{F}{V} C_B + k_1 C_A - k_2 C_B$$

Using (4) and (5) at steady state, here subscript s is used to indicate the steady state value represented by (12) and (13):

$$-k_3 C_{As}^2 + \left(-k_1 - \frac{F_s}{V}\right) C_{As} + \frac{F_s}{V} C_{Afs} = 0 \quad (12)$$

$$-\frac{F_s}{V} C_{Bs} + k_1 C_{As} - k_2 C_{Bs} = 0 \quad (13)$$

Steady state concentration of A and B is defined as (14) and (15) respectively using (12) and (13), only positive roots are considered as there cannot be negative concentrations.

$$C_{As} = \frac{-\left(k_1 + \frac{F_s}{V}\right) + \sqrt{\left(k_1 + \frac{F_s}{V}\right)^2 + 4k_3 \frac{F_s}{V} C_{Afs}}}{2k_3} \quad (14)$$

$$C_{Bs} = \frac{k_1 C_{As}}{\frac{F_s}{V} + k_2} \quad (15)$$

The linear state space model is represented as

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

$$y = Cx + Du$$

The state variable is represented as

$$x = \begin{bmatrix} C_A - C_{As} \\ C_B - C_{Bs} \end{bmatrix}$$

The output variable is represented as

$$y = [C_B - C_{Bs}]$$

The input variable is represented as

$$u = \begin{bmatrix} \frac{F}{V} - \frac{F_s}{V} \\ C_{Af} - C_{Afs} \end{bmatrix}$$

The elements of state space A matrix are found by

$$A_{ij} = \left. \frac{\partial f_i}{\partial x_j} \right|_{x_s, u_s}$$

The elements of state space B matrix are found by

$$B_{ij} = \left. \frac{\partial f_i}{\partial u_j} \right|_{x_s, u_s}$$

The state space model is represented as [20].

$$A = \begin{bmatrix} -\frac{F_s}{V} - k_1 - 2k_3 C_{As} & 0 \\ k_1 & -\frac{F_s}{V} - k_2 \end{bmatrix}$$

$$B = \begin{bmatrix} C_{Afs} - C_{As} & \frac{F_s}{V} \\ -C_{Bs} & 0 \end{bmatrix}$$

$$C = [0 \quad 1]$$

$$D = [0 \quad 0]$$

Based on steady state operating point

$$C_{As} = 3 \text{ gmol}^{-1}, C_{Bs} = 1.117 \text{ gmol}^{-1},$$

$$\frac{F_s}{V} = 0.5714 \text{ min}^{-1}$$

$$A = \begin{bmatrix} -2.4 & 0 \\ 0.83 & -2.23 \end{bmatrix}$$

$$B = \begin{bmatrix} 7 & 0.57 \\ -1.117 & 0 \end{bmatrix}$$

$$C = [0 \quad 1]$$

$$D = [0 \quad 0]$$

Converting the state space model to transfer function

$$G(s) = C(sI - A)^{-1} B$$

$$g_p(s) = \frac{-1.117s + 3.1472}{s^2 + 4.6429s + 5.3821} \quad (16)$$

Equation (16) represents the process transfer function which is a non-minimum phase type system as there is a right half zero which act as a delay in the system.

Fig. 2 shows the block diagram of CSTR with controller, the primary objective of the control mechanism developed for the isothermal CSTR is that the product concentration should be controlled.

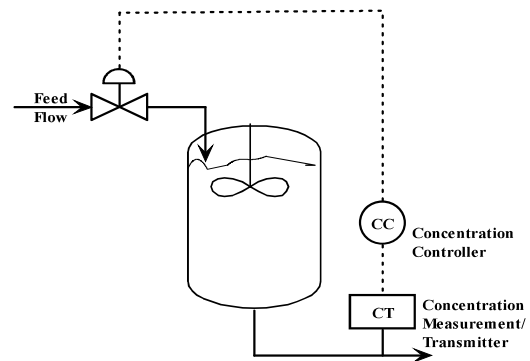


Fig. 2 Feedback Control Strategy for Isothermal CSTR

III. FRACTIONAL ORDER CONTROLLER

In the past decade, fractional order calculus attracted many researchers and also the use of fractional calculus in control system widely increased as well as accepted. For a control loop perspective, four situations may arise:

1. Integer order plant with integer order controller.
2. Integer order plant with fractional order controller.
3. Fractional order plant with integer order controller.
4. Fractional order plant with fractional order controller.

One of the primary controllers is PID controller, which is widely used. Fractional controller is denoted by $PI^\lambda D^\mu$ was proposed by Igor Podlubny in 1997 [21], here λ and μ have non-integer values. Fig. 3 shows the block diagram of fractional order PID controller.

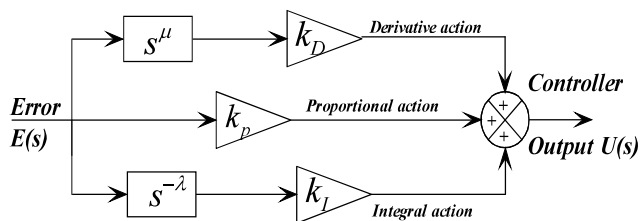


Fig. 3 Fractional order PID controller

Equation (17) represents the transfer function for conventional PID controller:

$$G_{PID}(s) = \frac{U(s)}{E(s)} = \left(K_p + \frac{1}{T_i s} + T_d s \right) \quad (17)$$

Transfer function for fractional order PID controller can be written as (18):

$$G_{FOPID}(s) = \frac{U(s)}{E(s)} = \left(K_p + \frac{1}{T_i s^\lambda} + T_d s^\mu \right) \quad (18)$$

where λ and μ are arbitrary real numbers, K_p is amplification (gain), T_i is integration constant and T_d is differentiation constant. Taking $\lambda=1$ and $\mu=1$, a classical PID controller is obtained.

The $PI^\lambda D^\mu$ controller is more flexible and gives an opportunity to better adjustment of the dynamics of control system. It is compact and simple but the analog realization of fractional order system is very difficult [22].

IV. PARTICAL SWARM OPTIMIZATION (PSO)

In this paper, Particle Swarm Optimization (PSO) technique is used. PSO algorithm adopts a strategy based on particle swarm and parallel global random search. PSO algorithm determines search path according to the velocity and current position of particle without more complicated evolution operation. It has better performance than early intelligent algorithms on calculation speed and memory occupation, and has less parameters and is easier to realize [23].

Particle Swarm has two primary operators: Velocity update and Position update. During each generation, each particle is accelerated toward the particles previous best position and the global best position. At each iteration a new velocity value for each particle is calculated based on its current velocity; the distance from its previous best position, and the distance from the global best position. The new velocity value is then used to calculate the next position of the particle in the search space. This process is then iterated a set number of times or until a minimum error is achieved [24].

Let in PSO, a swarm has N number of particles moving around in a D dimensional search space. The i th particle denoted as (19).

$$X_i = (x_{i1}, x_{i2}, \dots, x_{iD}) \quad (19)$$

whose previous best solution (p_{best}) is represented as (20):

$$P_i = (p_{i1}, p_{i2}, \dots, p_{iD}) \quad (20)$$

and the current velocity is described by (21):

$$V_i = (v_{i1}, v_{i2}, \dots, v_{iD}) \quad (21)$$

Finally, the best solution of whole swarm (g_{best}) also called global best is represented as (22):

$$P_g = (p_{g1}, p_{g2}, \dots, p_{gD}) \quad (22)$$

At each time step, each particle moves towards (P_{best}) and (g_{best}) locations. The fitness function evaluates the performance of particles to determine whether the best fitting solution is achieved [24]. The detailed operation of particle swarm optimization is given as

Step 1: Initialization The velocity and position of all particles are randomly set within pre-defined ranges.

Step 2: Fitness Function Evaluate the fitness of each particle of the swarm.

Step 3: Velocity Updating: At each iteration, the velocities of all particles are updated according to (23):

$$v_{id} = w * v_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * rand() * (p_{gd} - x_{id}) \quad (23)$$

where w is inertia factor which balances the global wide-range exploitation and the local nearby exploration abilities of the swarm given by (24), c_1 and c_2 are two positive constants, called cognitive learning rate and social learning rate respectively; $rand()$ is a random function in the range $[0,1]$.

$$w = (w_{max} - w_{min}) * \frac{(Iter_{max} - Iter_{now})}{Iter_{max}} + w_{min} \quad (24)$$

Step 4: Position Updating Assuming a unit time interval between successive iterations, the positions of all particles are updated according to (25):

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (25)$$

After updating, position should be checked and limited to the allowed range.

Step 5: Memory Updating Update p_{best} and g_{best} as per (26) and (27):

$$p_{best} = p_i \text{ if } f(p_i) < f(p_{best}) \quad (26)$$

$$g_{best} = p_g \text{ if } f(p_g) < f(g_{best}) \quad (27)$$

where $f(x)$ is the objective function subject to minimize.

Step 6:**Stop Criteria** The algorithm repeats from Steps 2 to 5 until certain stop conditions are met, such as a pre-defined number of iterations or a failure to make progress for a certain number of iterations. Once terminated, the algorithm reports the values of p_{best} and g_{best} as its solution [24].

The performance of the combination of FOPID parameters is determined by a fitness function. For CSTR, which is non-minimum phase type problem four components are considered to define fitness function. There are steady state error, peak overshoot, rise time and settling time. However, the contribution of these component functions towards the original fitness function is determined by a scale factor that depends upon the choice of the designer. For this particular problem the best point is the point where the fitness function has the minimum value. The chosen fitness function is given as (29):

$$F = 1 - e^{-\beta(M_p + e_{ss})} - e^{-\beta(T_s - T_r)} \quad (29)$$

where F : Fitness function; M_p : Peak Overshoot; T_s : Settling Time; T_r : Rise Time; e_{ss} : Steady State Error; β : Scaling Factor (Depends upon the choice of designer) For our case of design we have taken the scaling factor $\beta = 1$.

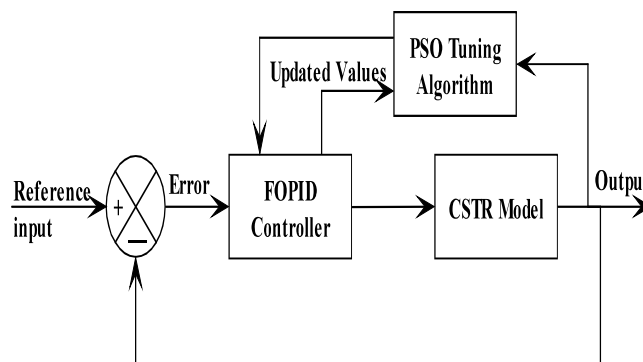


Fig. 4 PSO tuned FOPID controller

Fig. 4 shows the PSO tuned FOPID control structure for the controlling purpose of CSTR. The PSO tuning algorithm is using actual output as input, so it can be considered as a feedback structure.

V. SIMULATION RESULTS

This section deals with the Simulink models used in this particular problem and simulation results obtained from those models. Figs. 5-7 show Simulink models of process with conventional PID controller, with fuzzy controller and with fractional order controller respectively. In all the Simulink models the disturbance is considered at the output. The transfer function of CSTR is given in (16).

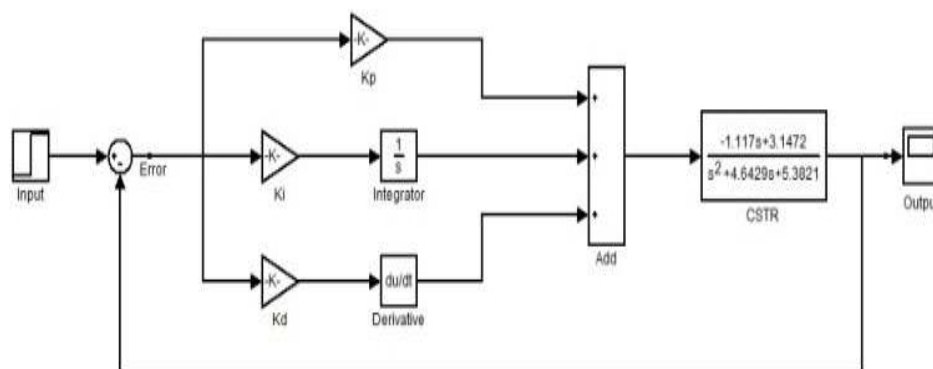


Fig. 5 Simulink model with conventional PID controller

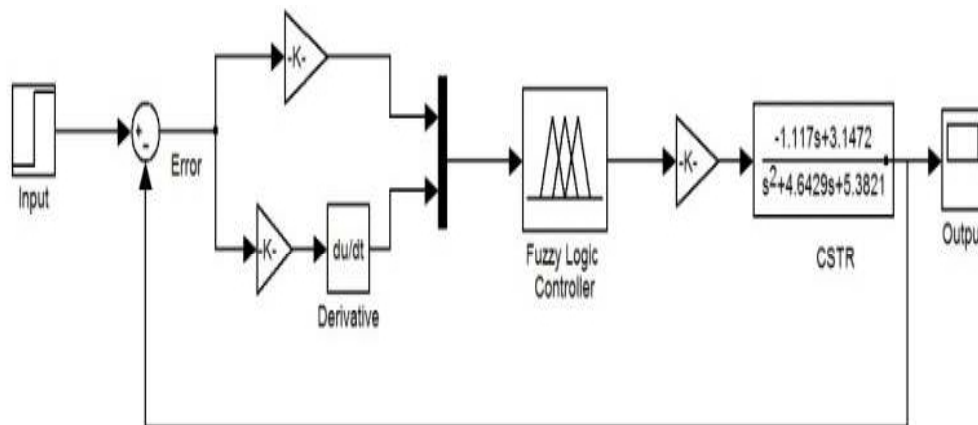


Fig. 6 Simulink model with Fuzzy controller

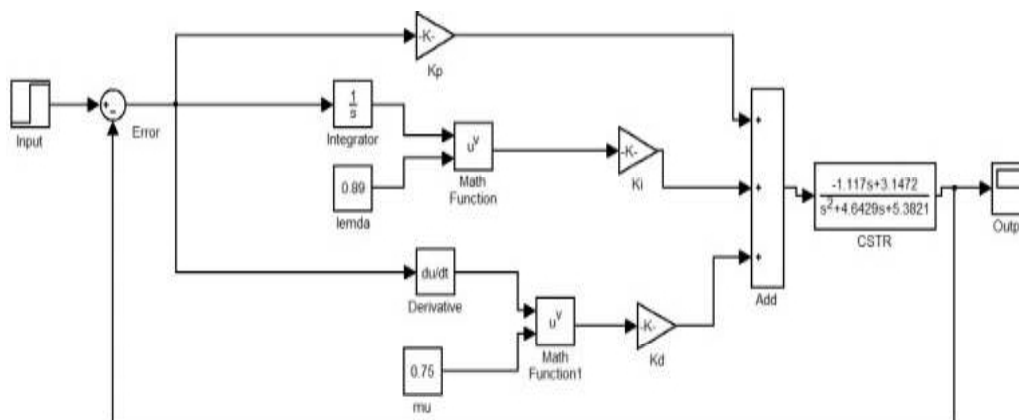


Fig. 7 Simulink model with FOPID controller

For tuning of conventional PID controller, Ziegler Nicholas method is used. The values of K_p , K_d and K_i obtained from it are 0.56, 0.102 and 3.02 respectively. The fuzzy controller is implemented using FIS toolbox in MATLAB with two inputs and one output variable having seven membership functions each. PSO algorithm is used to tune the FOPID controller.

Table I gives a comparative analysis for different controllers designed to control the product concentration of CSTR. It can be seen that the FOPID controller shows minimum overshoot and minimum negative peak; the fuzzy based controller shows infinite settling time as its output oscillates and never get settled.

TABLE I
TIME DOMAIN PARAMETERS FOR DIFFERENT CONTROLLERS

Parameters/ Controller	Rise Time (Sec.)	Peak Time (Sec.)	Settling Time (Sec.)	Peak Overshoot (%)	-ve Peak (%)
Conventional PID	2.44	3.202	18.74	68	11
Fuzzy Controller	3.94	5.41	Inf	17	18
FOPID Controller	7.92	7.92	7.92	00	3

Fig. 8 shows the comparative step response for all three types of controllers, which are conventional PID, fuzzy based and FOPID. The oscillations and highest negative peak in output of fuzzy based controller is clearly visible. The negative peak in the step response is because of RHP zero of the system which makes it a non-minimum phase type system.

Table II compares the various performance indices for the different type of controllers. Among all the performance indices ITAE is the main concerned performance index, as it tell about the speed and accuracy of the response. In terms of performance indices, FOPID controller performs fairly well.

TABLE II
PERFORMANCE INDICES FOR DIFFERENT CONTROLLERS

Error/Controller	IE	IAE	ISE	ITAE
Conventional PID	0.566	3.682	1.931	19.590
Fuzzy Controller	1.324	2.657	1.280	25.300
FOPID Controller	2.348	1.136	1.010	1.0460

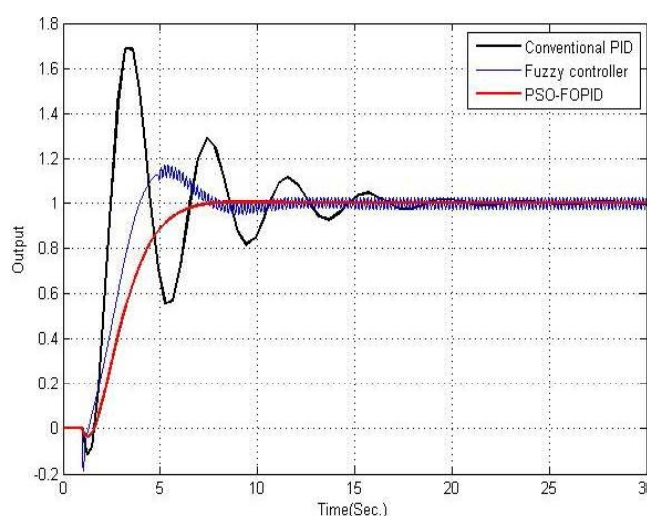


Fig. 8 Step responses of various controllers

Table III shows the parameters of fractional order PID controller with respect to number of iterations taken by the PSO algorithm which is used to optimize the parameters of the controller. The time taken by the PSO algorithm is also mentioned in seconds with respect to number of iterations. This time might vary with control problem for which it is used and the PSO parameters explained in section IV. It is on higher side for his CSTR problem due to the non-minimum phase system and use of fractional order controller.

TABLE III
PARAMETERS OF FOPID CONTROLLER FOR DIFFERENT NO. OF ITERATIONS

Parameters/ No. Iterations	K_p	K_d	K_i	λ	M	Run time in sec.
20	0.0500	0.0502	0.980	0.3000	0.8600	18.39
40	0.2510	0.0243	0.499	0.5968	0.0706	34.59
60	0.0027	0.3088	0.980	0.4006	0.6101	53.37
80	1.1930	0.0545	0.334	0.6046	0.4710	60.62
100	0.2000	0.0600	0.800	0.8900	0.7500	70.92

Fig. 9 shows the value of fitness function defined by (29) with each iteration. As seen from Fig. 9, there is no change in the value of fitness function after 100 iterations. So 100

iterations is considered as maximum iteration for this problem and the best solution is given by the set of variables for which the fitness function will be minimum.

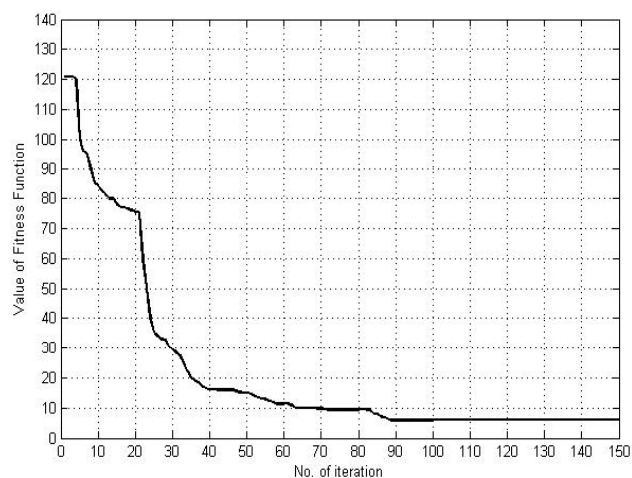


Fig. 9 Value of fitness function w.r.to no. of iteration

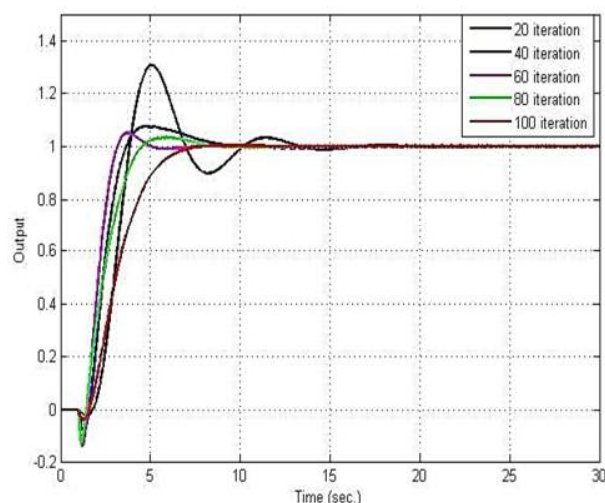


Fig. 10 Step responses of FOPID for different no. of iterations

The time domain parameters with respect to number of iterations are tabulated in Table IV, through which it can be concluded that for 100th iteration the overall performance of the system improves in terms of settling time, peak overshoot, and negative peak, which also reflects in Fig. 10.

TABLE IV
TIME DOMAIN PARAMETERS FOR DIFFERENT NO. OF ITERATIONS

Parameters/ No. Iterations	Rise Time(Sec.)	Peak Time (Sec.)	Settling Time (Sec.)	Peak Overshoot (%)	-ve Peak (%)
20	3.85	5.07	15	50	2.0
40	3.65	4.76	14	7.0	8.0
60	3.16	3.81	9	5.0	13
80	4.54	5.71	9.76	3.0	12
100	7.92	7.92	7.92	00	3.0

Fig. 10 shows the step response of fractional order PID controller for different number of iterations. It shows that as the

number of iterations increases, the overshoot, negative peak and oscillations in output decreases.

Table V compares the various performance indices of the step response with respect to number of iterations. The best results are obtained for 100 iterations which stands in support with Figs. 8 and 9.

TABLE V
PERFORMANCE INDICES FOR DIFFERENT NO. OF ITERATIONS

Errors/ No. Iterations	IE	IAE	ISE	ITAE
20	1.211	3.359	1.937	15.06
40	1.247	1.787	1.217	5.902
60	1.487	1.726	1.156	5.740
80	1.167	1.289	1.114	2.528
100	2.348	1.137	1.010	1.046

VI. CONCLUSION

This paper dealt with the analysis of CSTR, which is a typical non-minimum phase type nonlinear system with lumped parameters, having swarm intelligent based optimized fractional order PID controller. A comparative study is made with conventional PID controller and fuzzy controller. The comparison of controllers is done on the basis of time domain parameters and performance indices of the step response of CSTR model.

From the results explained in previous section, it is seen that the fractional order controller perform well in comparison to other controllers. The optimized PSO based tuning method is found most suitable for fractional order controller.

REFERENCES

- [1] K. B. Oldham and J. Spanier, *The Fractional Calculus*, New York and London: Academic Press, 1974.
- [2] J. A. Tenreiro Machado, "Analysis and design of fractional order digital control systems," *Journal of Systems Analysis Modeling Simulation*, vol. 27, pp. 107-122, 1997
- [3] A. Oustaloup, J. Sabatier and X. Moreau, "From fractal robustness to the CRONE approach," *ESAIM*, vol. 5, pp. 177-192, 1998.
- [4] Tepljakov, A. Petlenkov, E. Belikov, J., "A flexible MATLAB tool for optimal fractional-order PID controller design subject to specifications," *Control Conference (CCC), 2012, 31st Chinese*, 25-27 July 2012, pp.4698-4703.
- [5] Deepyaman Maiti, Ayan Acharya, Mithun Chakraborty, Amit Konar and Ramadoss Janarthanan, "Tuning PID and PI^λD^μ Controllers using the Integral Time Absolute Error Criterion," *4th IEEE International conference on Information and Automation for Sustainability (ICIAFS)*, Colombo, Sri Lanka 12-14 December 2008, pp.74-80.
- [6] Yeroglu, C. and Tan, N., "Note on fractional-order proportional-integral-differential controller design," *Control Theory & Applications, IET*, vol.5, no.17, pp.1978-1989, Nov. 17 2011.
- [7] Chunna Zhao, Dingy'u Xue and YangQuan Chen, "A Fractional Order PID Tuning Algorithm for A Class of Fractional Order Plants," in *Proc. IEEE International Conference on Mechatronics & Automation*, Niagara Falls, Canada, July 2005, pp. 216-221.
- [8] C. A. Monje, B. M. Vinagre, Y. Q. Chen, V. Feliu, P. Lanusse and J. Sabatier, "Proposals for fractional PI^λD^μ tuning," *The First IFAC Symposium on Fractional Differentiation and its Applications 2004*, Bordeaux, France, July 19-20, 2004.
- [9] Dingy'u Xue and Yang Quan Chen, "A comparative introduction of four fractional order controllers," *Proceedings of the 4th World Congress on Intelligent Control and Automation*, Shanghai, China, 2002, pp. 3228-3235.
- [10] Petras, I., "The fractional order controllers: methods for their synthesis and application," *Journal of Electrical Engineering*, 50(9-10), pp.284-288.

- [11] Dorcak, L., Petras, I., Kostial, I., and Terpak, J., "State-space controller design for the fractional-order regulated system," In *Proceedings of the ICCS'2001*, Krynica, Poland, May 22 - 25, 2001, pp. 15 - 20.
- [12] Biswas, A., Das, S., Abraham, A., and Dasgupta, S., "Design of fractional-order PID controllers with an improved differential evolution," *Engineering Applications of Artificial Intelligence*, vol. 22, pp. 343-350, 2009.
- [13] C. A. Monje, B. M. Vinagre, Y. Q. Chen, V. Feliu, P. Lanusse, and J. Sabatier, "Proposals for fractional $PI^{\lambda}D^{\mu}$ tuning," In *Proceedings of The First IFAC Symposium on Fractional Differentiation and its Applications (FDA04)*, Bordeaux, France, 2004.
- [14] Devaraj, D., and Selvabala, B., "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 & Distribution*, vol.3, no.7, pp.641-649, July 2009.
- [15] Wong, C. C., Li, S., and Wang, H. Y., "Optimal PID controller design for AVR system," *Tamkang Journal of Science and Engineering*, vol. 12, no. 3, pp. 259-270, 2009.
- [16] Dos Santos Coelho, L., "Tuning of PID controller for an automatic regulator voltage system using chaotic optimization approach," *Chaos, Solitons & Fractals*, vol. 39, issue 4, pp. 1504-1514, 2009.
- [17] Krohling, R. A., & Rey, J. P., "Design of optimal disturbance rejection PID controllers using genetic algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 5, pp. 78-82, 2001.
- [18] Gozde, H., & Taplamacioglu, M. C., "Application of artificial bee colony algorithm in an automatic voltage regulator (AVR) system," *International Journal on Technical and Physical Problems of Engineering*, vol.1, no. 3, pp. 88-92, 2010.
- [19] F. Gao and H Q Tong, "A novel optimal PID tuning and online tuning based on particle swarm intelligence," *Proc. Int. Conf. Sensing, Comput. And Automation*, 2006, pp. 182-186.
- [20] B. Wayne Bequette, "Process Control: Modeling, Design, and Simulation", Rensselaer Polytechnic Institute, Publication- PHI Learning Private Limited, New Delhi, 2010.
- [21] Podlubny I., Dorcak L., Kostial I., "On fractional derivatives, fractional-order dynamic system and $PI^{\lambda}D^{\mu}$ controllers," *Proc. 36th IEEE CDC*, San Diego, 1997, pp. 4985-4990.
- [22] Zhou F., "Design, implementation and application of distributed order PI control," *ISA Transactions*, 2013.
- [23] Daniel Bratton and James Kennedy, "Defining a Standard for Particle Swarm Optimization," in *Proceedings of IEEE Swarm Intelligence Symposium (SIS 2007)*, 2007.
- [24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks*, vol. IV, Perth, Australia, 1995, pp. 1942-1948.