2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Soft Computing Techniques based Optimal Tuning of Virtual Feedback PID Controller for Chemical Tank Reactor

Geetha M, Manikandan P, Jovitha Jerome,
Department of Instrumentation and Control Systems Engineering
PSG College of Technology,
Coimbatore, India
vanajapandi@gmail.com

***Abstract* - CSTR plays a vital role in almost all the chemical reactions and is a highly nonlinear system exhibiting stable as well as unstable steady states. The variables which characterize the quality of the final product in CSTR are often difficult to measure in real-time and cannot be directly measured using the feedback configuration [1]. So, a virtual feedback control is implemented to control the state variables using Extended Kalman Filter (EKF) in the feedback path. Since it is hard to determine the optimal or near optimal PID parameters using classical tuning techniques like Ziegler Nichols method, a highly skilled optimization algorithm like Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) are used. This work is based on the optimal tuning of virtual feedback PID control for a CSTR system using soft computing algorithm for minimum Integral Square Error (ISE) condition.**

***Keywords* – CSTR, PSO, PID, EKF, ISE.**

I. INTRODUCTION

During the past decades, the process control industries have made great advances. Numerous control methods such as Adaptive Control, Neural Network and Fuzzy Control have been studied. Among them, the best known is the Proportional-Integral-Derivative (PID) controller which has been widely used in the industry because of its simple structure and robust performance under wide range of operating conditions. Unfortunately, it has been quite difficult to tune properly the gains of PID controller because many industrial plants are often burdened with problems such as higher order of the system, time delay and nonlinearities associated with the system. Also, it is hard to determine the optimal or near optimal PID parameters using classical tuning methods. For these reasons, it is highly desirable to increase the capabilities of PID controllers by adding many new features. Many Artificial Intelligence (AI) techniques have been employed to improve the controller performance for a wide range of plants while retaining their basic characteristics. Artificial Intelligence techniques such as Neural Network, Fuzzy Logic have been widely applied to proper tuning of PID controller parameters.

Particle Swarm Optimization (PSO) [9], first introduced by Kennedy and Eberhart, is one of the modern heuristic algorithms. It can generate high quality solution within short calculation time and show stable convergence characteristics than other stochastic methods.

Ant colony optimization (ACO) system is a competing meta-heuristic for large-scale and difficult combinatorial optimization problems. It is based on the ant system – first defined by Colorni and Maniezzo (1991), Dorigo (1992) and Dorigo, Maniezzo, and Colorni (1996) in early 1990s – that imitates the foraging behavior of the real ants. The main idea of ant colony optimization is that a population of artificial ants repeatedly builds and improves solutions to a given instance of a combinatorial optimization problem. The main idea behind ant colony optimization is that when the ants search for food, they initially explore the area surrounding their nest randomly. When one finds a food source, it evaluates it, take some food and goes back to the nest. As they move back, they deposit on the ground a chemical substance called pheromone, which is detectable by other ants. The amount of pheromone that is deposited varies depending on the quantity and quality of the food, and leads other ants to that food source. By the use of this property, the ants can find the shortest path between their nest and the source (Socha & Dorigo, 2009).

In this paper, PSO and ACO are an excellent optimization methodology and is a promising approach for optimal tuning of PID controller parameters. Here, the soft computing techniques are done for optimal tuning of PID controller parameters for Continuous Stirred Tank Reactor (CSTR) system.

II. MATHEMATICAL MODEL OF CSTR

A perfectly mixed Continuously Stirred Tank Reactor (CSTR) is shown in Figure 1. It is a single, first order exothermic irreversible reaction A→B in which a fluid stream is continuously fed to the reactor [2]. Since the fluid is perfectly mixed, the exit stream has the same concentration and temperature as the reactor fluid. The jacket surrounding the reactor also has feed and exit streams. The jacket is assumed to have a uniform lower temperature than the reactor [3]. Energy then passes through the reactor walls from the reactor into the jacket removing the heat generated by reaction.

There are many examples of this type of reactors used in industries. The industrial reactors typically have more complicated kinetics, but the characteristic behaviour is similar [4].

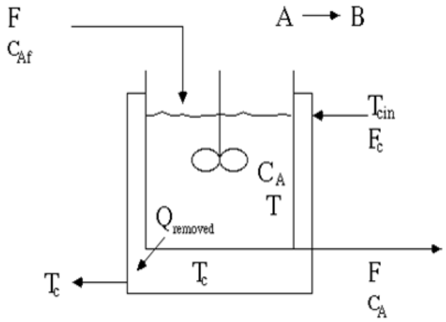

Figure 1. Continuous Stirred Tank Reactor with cooling Jacket

A. *Steady State Solution*

By applying material balance and energy equations, the resulting mathematical model may be obtained. The steady-state solution is obtained when $dC_A/dt = 0$ and $dT/dt = 0$, that is

$$f_1(C_A, T) = 0 = \frac{q}{V}\left(C_{Af} - C_A\right) - k_o exp\left(\frac{-\Delta E}{RT}\right) C_A \tag{1}$$

$$f_2(C_A, T) = 0 = \frac{q}{V}\left(T_f - T\right) + \left(\frac{-\Delta H}{\rho C_p}\right) k_o exp\left(\frac{-\Delta E}{RT}\right) C_A - \frac{\rho_c C_{pc}}{\rho C_p V} q_c \left\{1 - \exp\left(\frac{-hA}{q_c \rho C_p}\right)\right\}\left(T_{cf} - T\right) \tag{2}$$

The linear model of the system is obtained by taking the Jacobian form of equations (1) and (2)

$$\dot{X} = \begin{bmatrix} \frac{-q}{v} - k_s & -\acute{k}_s C_A \\ \frac{-\Delta H}{\rho C_p} k_s & \frac{-q}{v} + \frac{(-\Delta H)}{\rho C_p} C_A \acute{k}_s - \frac{\rho_c C_{pc}}{\rho C_p V}\left\{1 - \exp\left(\frac{-hA}{q_c \rho C_p}\right)\right\} \end{bmatrix} \begin{bmatrix} C_A \\ T \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ \rho_c C_{pc} \Big/ \rho C_p V \left\{1 + 2\exp\left(\frac{hA}{q_c \rho C_p}\right)^2\right\} \end{bmatrix} [q_c]$$

$$\dot{Y} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} C_A \\ T \end{bmatrix}$$

where, $k_s = k_0 exp\left(\frac{-E}{RT}\right)$ and $k_s' = \partial k_s \big/ \partial T = k_s \left[E \big/ RT^2\right]$

B. *Steady state operating data for CSTR*

The specifications of CSTR taken in this work [2] are given below in table 1.

Table 1 Steady state operating data

Process variable	**Normal operating condition**
Measured Product Concentration (C_A)	0.08235 mol/lit
Reactor Temperature (T)	441.81 K
Volumetric Flow rate (q)	100 L/min
Reactor Volume (V)	100 L
Feed Concentration (C_{Af})	1 mol/lit
Feed Temperature (T_f)	350 K
Coolant Temperature (T_{cf})	350 K
Coolant Flow rate(q_c)	100 L/min
Heat of Reaction (ΔH)	2e5 cal/mol
Reaction rate constant(k_0)	7.2e10 min^{-1}
Activation energy term(E/R)	9980 K
Heat transfer term (hA)	7e5 cal/(min.K)
Liquid Density(ρ, ρ_c)	1000 g/L
Specific Heat capacity (C_p, C_{pc})	1 cal/(g.K)

III. TUNING ALGORITHMS

Tuning of a controller refers to the methods of determining the parameters of a PID controller for the given system. The design methods differ with respect to the knowledge of the process dynamics that is required [5]. A PID controller is described by three parameters; K_p, τ_i and τ_d; there are many different methods to find the suitable parameters of the controller. The methods differ in complexity, flexibility and the amount of process knowledge used. Three tuning algorithms have been discussed below are used in this work.

A. *Ziegler Nichols Tuning*

In 1942, Ziegler and Nichols [6], described simple mathematical procedures, for tuning the PID controllers. Both the techniques make a priori assumption on the system model, but do not require the system model to be specifically known. Ziegler-Nichols formulae for specifying the controllers are based on the plant step response.

1) Open Loop Response

The open-loop method is typical for a first-order system with transportation delay. The response is characterized by 2 parameters, L the time-delay and T the time-constant. These are found by drawing a tangent to the step response at its point of inflection and noting its intersections with the time axis and steady-state value.

2) Closed Loop Response

The closed-loop method targets plant that can be rendered unstable under proportional control. The technique is designed to result in a closed loop system with 25% overshoot.

B. *Genetic Algorithm [GA]*

Genetic Algorithm[7] is a search technique to determine approximate solutions to optimization and search problems. The problem consists in finding out the solution that fits the best from all the possible solutions. GA handles a population of possible solutions. Each solution is represented through a chromosome, which is just an abstract representation. A set of reproduction operators has to be determined. Reproduction operators are applied directly on the chromosomes, and are used to perform mutations and recombinations over solutions of the problem. It can be extremely difficult to find a representation, which respects the structure of the search space and reproduction operators, which are coherent and relevant according to the properties of the problems. Selection is supposed to be able to compare each individual in the population.

Selection is done by using a fitness function [8]. Each chromosome has an associated value corresponding to the fitness of the solution it represents. The fitness should correspond to an evaluation of how good the candidate solution is. The optimal solution is the one, which maximizes the fitness function. Genetic Algorithm deals with the problems that maximize the fitness function. But, if the problem consists in minimizing a cost function, the adaptation is quite easy. Either the cost function can be transformed into a fitness function, for example by inverting it; or the selection can be adapted in such way that they consider individuals with low evaluation functions as better. Once the reproduction and the fitness function have been properly defined, a Genetic Algorithm is evolved according to the same basic structure. It starts by generating an initial population of chromosomes [7]. Generally, the initial population is generated randomly.

C. *Particle Swarm Optimization*

Particle swarm optimization (PSO) is a population based stochastic optimization [9] technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions called particles, fly through the problem space by following the current optimum particles.

In the past several years, PSO [10] has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods. Another reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. Particle swarm optimization has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement.

1) Operations in PSO

In D-dimensional search space [5], the position of the i^{th} particle can be represented by a D-dimensional vector $x_i = (x_{i1}, \dots, x_{id}, \dots, x_{iD})$. The velocity of the particle can be represented by another D-dimensional vector $v_i = (v_{i1}, \dots, v_{id}, \dots, v_{iD})$. The best position previously visited by the i^{th} particle is denoted as $p_i = (p_{i1}, \dots, p_{id}, \dots, p_{iD})$ and p_g as the index of the particle visited the previous position in the swarm, then p_g becomes the best solution found so far, and the velocity of the particle and its new position will be determined according to the following two equations with inertia weight w added to it.

$$v_{id} = wv_{id} + c_1 r(p_{id} - x_{id}) + c_2 R(p_{gd} - x_{id}) \tag{3}$$

$$x_{id} = x_{id} + v_{id} \tag{4}$$

where c_1 and c_2 are positive constants, and r and R are two random functions in the range [0,1]. $x_i = (x_{i1}, \dots, x_{id}, \dots, x_{iD})$ represents the location of i^{th} particle and $p_i = (p_{i1}, \dots, p_{id}, \dots, p_{iD})$ represents the previous best position (the position giving the best fitness value) of the i^{th} particle. The symbol g represents the index of the best particle among all the particles in the population. $v_i = (v_{i1}, \dots, v_{id}, \dots, v_{iD})$ represents the rate of change of position (velocity) for the i^{th} particle [10]. The parameter w in the equation (12) is the inertia weight that increases the overall performance of PSO. Larger value of w can increase the ability for global search while lower value of w implies higher ability for local search.

2) Optimal Tuning of PID controllers using PSO

The value of fitness function defined by the optimization algorithm would be minimal. Performance characteristic of evaluation function would include over-shoot, rise-time and settling-time. Evaluation function computes the evaluation value of each particle in the swarm according to the control objective. The steps involved in designing PSO algorithm is given below. The block diagram representation of tuning the PID controllers of the CSTR system using Particle Swarm Optimization is shown in the Figure 2. The soft sensor used here is Extended Kalman Filter (EKF) which aids in the virtual feedback control.

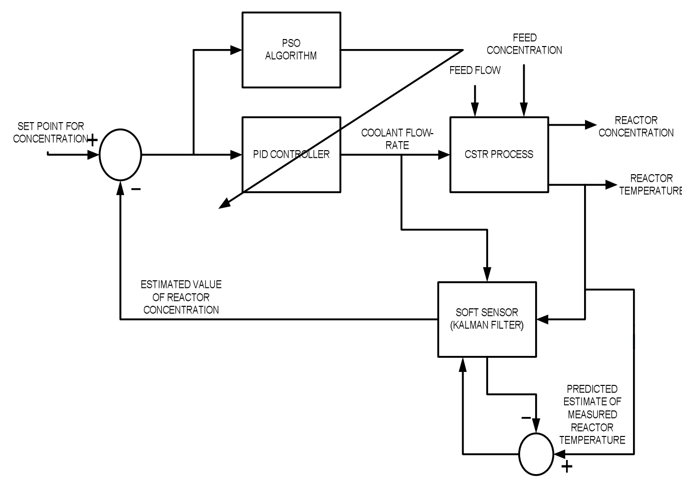

Figure 2. Block Diagram of PID parameters using PSO

The sequence of steps to study the PSO for the CSTR system is given below:

STEP 1: Specify the lower and upper bounds of the three K_p, K_i and K_d. Initialize randomly the particles of the swarm including swarm size, iteration, acceleration constant, inertia weight factor, the position matrix x_i and the velocity matrix v_i and so on.

STEP 2: Calculate the evaluation value of each particle using the evaluation function given.

STEP 3: Compare each particle's new fitness value with its personal best position's fitness value, and update the personal best position p_{best}.

STEP 4: Search for the best position among all particles' personal best position, and denote the best position as g_{best}.

STEP 5: Update the velocity v_i of each particle according to equation (3), and update the particle position matrix according to equation (4).
STEP 6: Update control parameter.
STEP 7: If the number of iterations reaches the maximum, then stop. The latest g_{best} is regarded as the optimal PID controller parameter. Otherwise, go to step 2.

D. *Ant Colony Optimization*

For a CO (constrained optimization) the designed model P=(S, F, Ω) consists of

- An objective function to be minimized
- A solution space S defined over a limited set of discrete decision variables and a set Ω of constraints between the variables;

The solution space S is defined in this fashion:
Specified is a set of n discrete variables X_i using values v_i^j □D_i ={ v_i^1,……. $v_i^{|D_i|}$ }, i=1,2,…….,n. Areasonable solution is a wide-ranging assignment with each decision variable assigned with domain value to gratify set of constraints. P shall be termed as unconstrained optimization problem. When Ω is empty thus allowing each decision variable taking any value from its domain individualistically of the value of the other decision variables. For a CO problem, the solution is s* □ *S**. The framework for ACO agreeing to Dorigo and Blum (2005) shall be formulated as follows
Initialize pheromone values (τ): In the beginning of the algorithm the pheromone values are all initialized to a constant value *c*>0.
Constant solution (τ): The elementary ingredient of any ACO algorithm is a constructive heuristic for the probabilistically building solutions. A constructive heuristic assembles solutions as orders of elements from the finite set of solution components □. A solutions construction starts with an vacant partial solutions ^{p}=<>.the process of constructing solutions can be considered as a walk on the purported construction graph g_c=(□ ,£), which is a fully linked graph whose vertices are the solution components in □ and whose edges are the elements of £. In most ACO algorithms the probabilities for taking the next solution component likewise called the transition probabilities are defined as

$$P\left(c_i^j \middle| S^p\right) = \frac{[\tau_i^j].[\square(c_i^j)]^{\beta}}{\sum_{c_k^l}^{N} [\tau_i^l].[\square(c_i^l)]^{\beta}} \; for \text{ all } c_i^j \qquad (5)$$

Pseudo code for basic framework of ACO algorithm

1. **Input:**An occurrence P of an unconstrained optimization problem model P=(S, f)
2. **Intiliaze:**Pheromone Values (τ)
3. S_{bs}←NULL
4. **While** Termination conditons not met **do**
   a. $Є_{iter}$← Ø
   b. **For** j=1,2,….,n_a**do**
      1. S←Constant solution (τ)
      2. **If** s is a valid solution **then**
      3. S ← Local Search(s) {optional}
      4. **If**(f(s)<f(s_{bs})) or (s_{bs}=NULL), **then** s_{bs}← s
      5. $Є_{iter}$←$Є_{iter}$□ {s}
      6. **End if**

      End for
5. **Apply pheromone update** (τ, $Є_{iter}$,s_{bs}) as below

$$\tau_i^j \leftarrow (1-\partial).\tau_i^j + \frac{\partial}{Є_{upd}} \sum_{\{s \, \epsilon Є_{upd} | c_i^j \epsilon \, s\}} F(s)$$

Where, i=1,2,….n, j=1,……..,|D_i|, $Є_{iter}$←arg max {F(s)|s□$Є_{iter}$}. $Є_{iter}$is the set of solutions constructed in the current iteration and$s_{bs}$is the best so far solution. The evaporation rate
∂□ [0,1] has the function of uniformly decreasing all the pheromone values to avoid a too rapid convergence (forgetting) towards a sub optimal solution.

6. **End while**
7. **Output:** The best so far solution, s_{bs}→[K_P,K_I,K_D].

[K_P,K_I,K_D] correspond to best so far solution gains of the proportional, integral and derivative component of PID controller and Smith PID.

IV. SIMULATION RESULTS

A. *Mathematical Model of CSTR*

The mathematical model of CSTR is obtained by solving the differential equations. The response of the temperature and concentration of the reactant for the step change in coolant flow rate is obtained.

1) *Influence of Coolant Flow-rate*

Both the concentration and the temperature of the reactant in a CSTR are influenced by its coolant flow-rate around the CSTR. These are shown in the following figures below.

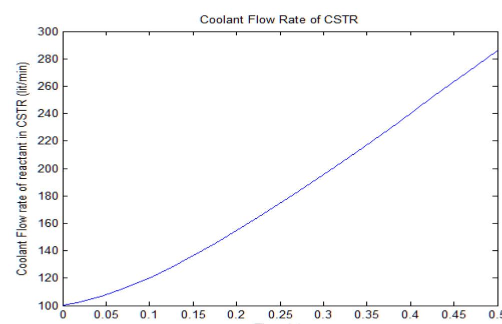

Figure 3. Variation in coolant flow rate

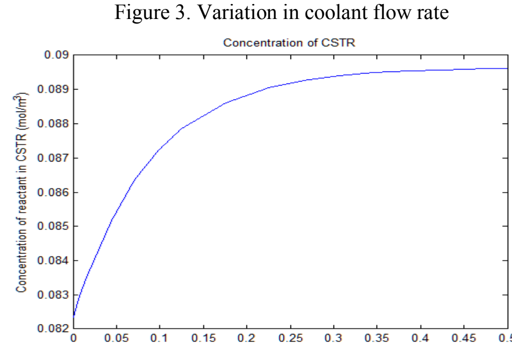

Figure 4. Open loop response of the concentration in CSTR

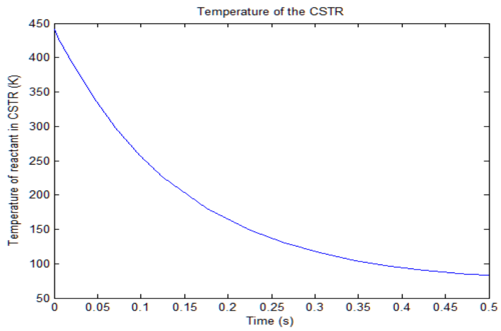

Figure 5. Open loop response of the temperature in CSTR

B. *Estimation using Kalman Filter as a Virtual Feedback Controller*

Simulation of CSTR and the estimation using Extended Kalman Filter as a soft-sensor [3] is carried out in open-loop condition.

Under normal operating conditions, the simulation is carried to show the true and estimated state responses of both concentration and temperature for a constant coolant flow-rate is shown below.

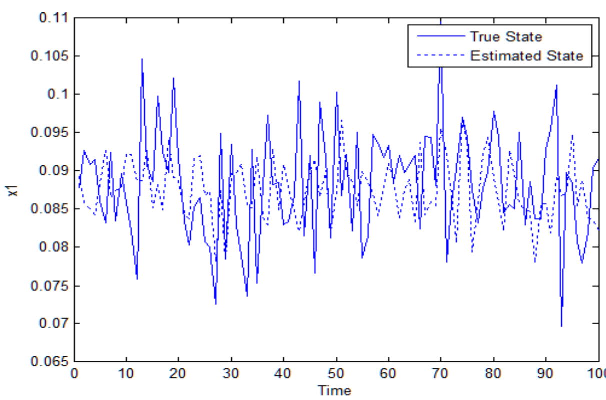

Figure 6. Evolution of true and estimated reactor concentration with constant coolant flow-rate

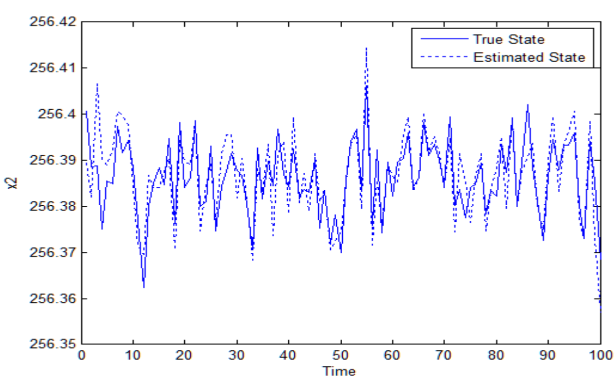

Figure 7. Evolution of true and estimated reactor temperature with constant coolant flow-rate

C. *Results of PSO Algorithm*

The optimized PID tuning parameters of the CSTR system is obtained using Ziegler – Nichols Tuning, Genetic Algorithm and Particle Swarm Optimization. The results from all the 3 tuning techniques are compared against each other.

1) Standard Operating Conditions

The standard operating conditions were considered for all the iterations as they have offered improved repeatability. The parameters of a PSO algorithm are given below. It is to be noted that for obtaining optimum PID parameters, the swarm iteration alone is varied.

- Weight / Inertia of the system – 0.5.
- Acceleration constants, c_1 and c_2 – 1.5.
- Swarm population – 100.
- Dimension of the search-space – 3 (K_p, K_i, K_d).

2) Robustness of PSO Algorithm

Since varying the swarm iteration is considered as the only tuning factor, we compare the results of PID controller for various iterations in PSO and conclude which among these gives the best fitness function. We have considered 50, 100, 150 and 200 iterations.

From the table 2, it is seen that the tuning parameters obtained for 200 iterations shows far better results than the others. The optimized PID parameters are listed in the Table 2 below.

Table 2. Optimized PID tuning parameters of PSO for different iterations

No of Iterations	K_p	K_i	K_d
50	0.2896	0.01177	0.6638
100	0.3260	0.02488	0.7497
150	0.3297	0.02548	0.7936
200	0.4140	0.02714	0.8487

3) Calculation of fitness function

A particular point in the search-space is the best point for which the fitness function attains an optimum value. In this case, four components are taken to define the fitness function. The fitness function is a function of 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 scaling factor. Scaling factor (β) is chosen as 1 in this application.

The chosen fitness function is expressed as

$$F = (1 - \exp(\beta))(M_P + E_{SS}) + (\exp(-\beta))(T_s - T_r) \qquad (6)$$

where,

- F – Fitness Function
- M_P – Peak Overshoot
- E_{SS} – Steady State Error
- T_s – Settling Time
- T_r – Rise Time
- β - Scaling Factor

The fitness function calculated for different values of iteration are shown below in table 3.

Table 3. Fitness Function calculated for different iterations

No of Iteration	Fitness Function
50	7.1221
100	5.7467
150	5.2415
200	4.9916

4) Performance Index

Performance Index [11] is a quantitative measure of the performance of the system. A system is considered as an optimal system when its parameters are adjusted so that the index reaches an extreme value, commonly a minimum value. A suitable performance index is the Integral Square Error (ISE), which is defined as

$$\mathrm{ISE} = \int_0^T e(t)^2 \, dt \tag{7}$$

ISE is more suitable to minimize large amount of errors. The squared error is mathematically more convenient for analytical and computational purposes.

Other performance criteria include evaluation of rise-time, settling-time and peak overshoot. Rise time is the time taken for the response to rise from 0 to 100% for the first time. Settling time is defined as the time taken by the response to reach and stay within specified error limit. Peak Overshoot is the ratio of maximum peak value measured from maximum value to the final value.

Table 4. Comparison of Performance Indices

Performance Index	ZNT	GA	PSO	ACO
ISE	2.844	2.011	1.706	1.925
Rise Time (s)	20.06	28.56	50.47	41.29
Settling Time (s)	164.32	89.34	79.38	83.67
Peak Overshoot (%)	28.20	12.58	2.74	6.81

5) Comparative results of GA, ZNT and PSO

A single loop PID tuning of a CSTR system is done using three standard tuning techniques like Ziegler-Nichols Tuning, Genetic Algorithm and Particle Swarm Optimization. The step response of the three discussed methods is shown in the below.

The desired PID tuning parameters from these three methods has been tabulated below.

Table 5. Tuning parameters obtained for GA, ZNT and PSO

Tuning Algorithm	K_p	K_i	K_d
ZNT	0.9903	0.0877	0.4288
GA	0.8861	0.0587	0.6271
PSO	0.4140	0.02714	0.8487
ACO	0.6640	0.0411	0.7142

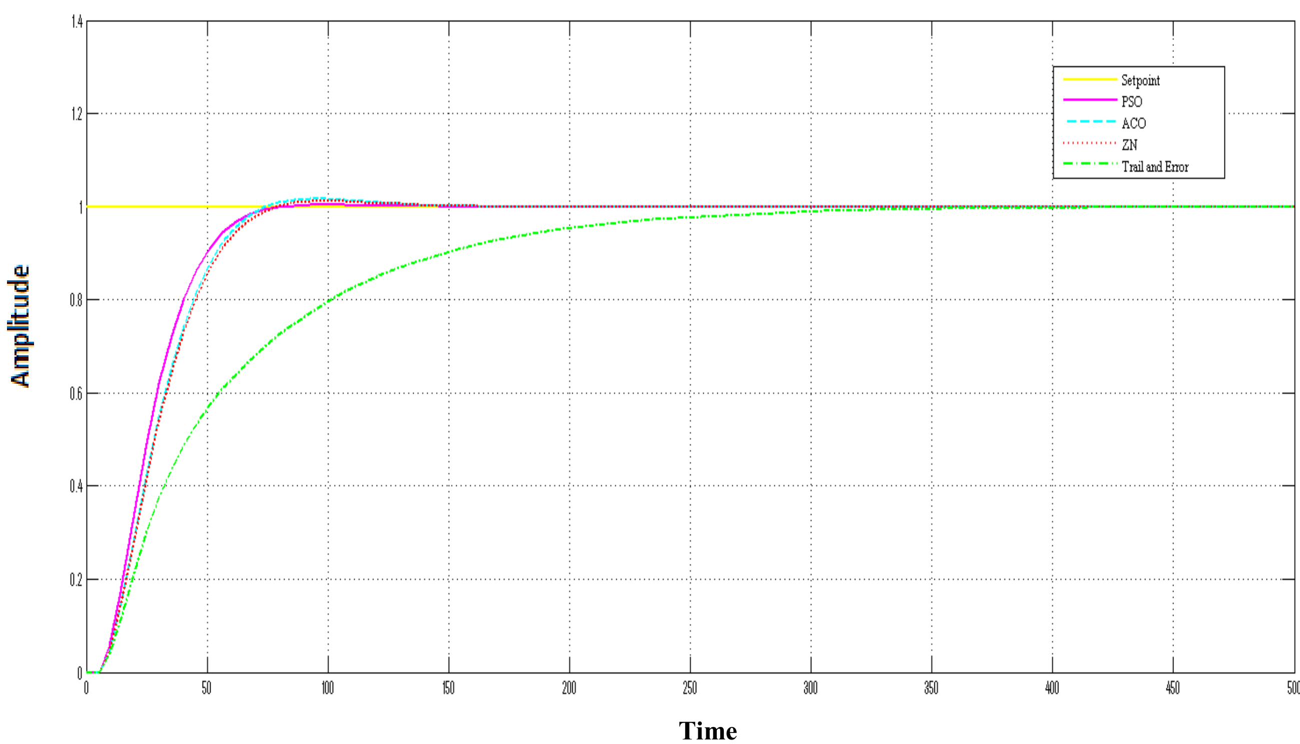

Figure 8. Step Response curve comparing results of ZNT, PSO and ACO

V. CONCLUSION

In this work, the PID controller tuning was done using a standard tuning algorithm (Ziegler-Nichols Tuning), an evolutionary computation algorithm (Genetic Algorithm) and a more advanced swarm intelligence approach (Particle Swarm Optimization) and (Ant Colony Optimization). By comparing all the three methods, it is found that PSO algorithm was the best implemented. Also, the PID controller parameters obtained from PSO algorithm gives better tuning result than the other 3 methods. This was also validated by checking for the robustness of PSO algorithm and it was concluded that the system exhibited best performance index for 200 iterations.

Though PSO algorithm is much advanced and simpler than other artificial intelligence based approach, it has its own short-comings. In this application, PSO algorithm was taken and applied for single set of data (K_p, K_i and K_d). Generally this can be seen as a limitation in terms of not being able to analyse multiple sets of data. Also these GA, ACO and PSO algorithms are giving better results, the best is yet to come. Right now with these probability and randomness based technologies, it cannot be claimed that the whole optimization domain can be traced. Moreover, there is a possibility that the algorithm could be trapped in the local optima. So it can be concluded that a much advanced algorithm could be developed in the future ironing out all the glitches from the existing algorithm.

REFERENCES

[1] M.Kodabandeh,H.Bolandi, "Model predictive control with State Estimation and Adaptation Mechanism for a Continuous Stirred Tank Reactor," *International Conference on Control, Automation and Systems*, vol. 12, pp. 1466-1471,2007.

[2] K. Srinivasan, J. Prakash, "Non-linear State Estimation for Continuous Stirred Tank Reactor using Neural Network State Filter " ,*IEEE annual conference*, pp 1-4 ,2006.

[3] Jiri Vojtesek , Petr Dostal, "Simulation analyses of continuous stirred tank reactor", *IEEE transactions*, Vol 24, pp 27-31, 2008.

[4] U. Sabura Banu,G Uma, "Modelling of CSTR by fuzzy clustering", *Proceedings of India International Conference on Power Electronics*, pp 371-376, 2006.

[5] Mohammad Ali Nekoui, Mohammad Ali Khameneh, Mohammad Hosein Kazemi, " Optimal design of PID controller for a CSTR system using Particle Swarm Optimization", *EPE-PEMC, 14th International Power Electronics and Motion Control Conference, Iran,*pp.T7-63-T7-66, 2010.

[6] Ziegler J. C, Nichols N. B, "Optimum settings for Automatic Controllers", *Trans. ASME,* vol.64,pp.759-768,1942.

[7] Sivanandham S. N, Deepa S. N, "Introduction to Genetic Algorithms", Springer-Verlag, Berlin Heidelberg,2008.

[8] Mohammed El-Said El-Telbany, "Employing Particle Swarm Optimizer and Genetic Algorithms for Optimal Tuning of PID Controllers – A comparative study", *ICGST – ACSE Journal,*Vol 7,Issue 2,pp. 49–54,November,2007.

[9] J. Kennedy, R. Eberhart, "Particle Swarm Optimization" *in Proc. IEEE Int. Conf. Neural Networks,* Vol.4,pp.1942-1948,1995.

[10] Clerc. M, Kennedy. J, "The Particle Swarm: Explosion, Stability, and Convergance in Multi-Dimensional Search Space",*IEEE Trans on Evolutionary Computation,*pp.58-73,2002.

[11] M.W.Iruthayarajan, S. Baskar, "Optimization Of PID Parameters Using Genetic Algorithm And Particle Swarm Optimization", *IET-UK International Conference on Information and Communication Technology in Electrical Sciences*, India, pp.81-86. 2007.