A Comparative Study of PID Controller Tuning Using GA, EP, PSO and ACO

B.Nagaraj
Tamilnadu News Print and Papers Ltd
Karur, India
nagarajice@gmail.com

Dr.N.Murugananth
Director International Relations, Karpagam University
Coimbatore, India
dranand.2006@gmail.com

Abstract: The aim of this paper is to study the tuning of a PID controller using soft computing techniques. The methodology and efficiency of the proposed method are compared with that of traditional methods. Determination or tuning of the PID parameters continues to be important as these parameters have a great influence on the stability and performance of the control system. The results obtained reflect that use of soft computing based controller improves the performance of process interms of time domain specifications, set point tracking, regulatory changes and also provides an optimum stability.

Keywords: **Ant colony algorithm, Evolutionary programming, Genetic algorithm particle swarm optimization and soft computing.**

I. INTRODUCTION

Proportional-integral-derivative (PID) control schemes continue to provide the simplest and yet effective solutions to most of the control engineering applications today. Since it is ease to design and simplicity in structure these are frequently used in the heavy industries to regulate the time domain behavior of many different types of dynamic plants.PID controllers can be tuned in a variety of ways including hand tuning Ziegler Nichols tuning, Cohen-coon tuning and Z-N step response, but these have their own limitations [3]. Soft computing techniques like GA, PSO, EP and ACO methods have proved their excellence in giving better results by improving the steady state characteristics and performance indices.

II PROPORTIONAL INTEGRAL DERIVATIVE CONTROLLER

PID controller is a generic control loop feedback mechanism widely used in industrial control systems. It calculates an error value as the difference between measured process variable and a desired setpoint. The PID controller calculation involves three separate parameters proportional integral and derivative values.

The proportional value determines the reaction of the current error, the integral value determines the reaction based on the sum of recent errors, and derivative value determines the reaction based on the rate at which the error has been changing the weighted sum of these three actions is used to adjust the process via the final control element.

The block diagram of a control system with unity feedback employing Soft computing PID control action in shown in figure 1 [7].

$$Y(t) = [kpe(t) + Kd\frac{d(e)}{d(t)} + Ki\int_0^t e(t)d(t)] \text{-- (1)}$$

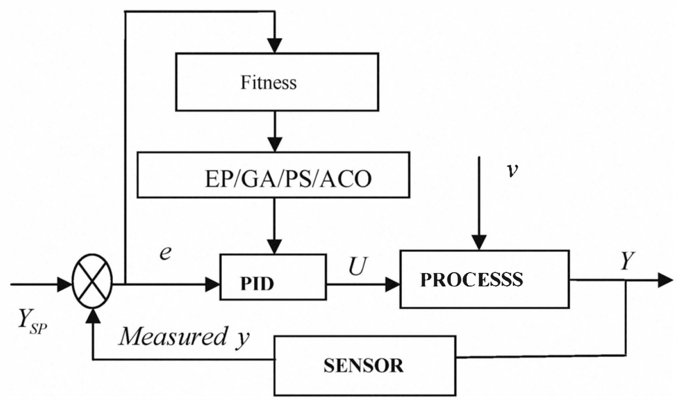

Fig 1.Block diagram of Intelligent PID controller

III REASON FOR SELECTING SOFT COMPUTING TECHNIQUES

A. Model type: Many methods can be used only when the process model is of a certain type, for example a first order plus dead time model (FOPDT). Model reduction is necessary if the original model is too complicated. [6]

B. Design criteria: These methods aim to optimize some design criteria that characterize the properties of the closed-loop system. Such criteria are, for example, gain and phase margins, closed-loop bandwidth, and different cost functions for step and load changes.[6]

C. Approximations: Some approximations are often applied in order to keep the tuning rules simple. [6]

The purpose of this project is to investigate an optimal controller design using the Evolutionary programming, Genetic algorithm, Particle swarm optimization techniques. In this project, a new PID tuning algorithm is proposed by the EP, GA, PSO and ACO techniques to improve the performance of the PID controller.

The ultimate gain and the ultimate period were determined from a simple continuous cycle experiment. The new tuning algorithm for the PID controller has the

978-1-4244-7770-8/10/$26.00 ©2010 IEEE

initial value of parameter Kp, T_i, T_d by the Ziegler-Nichols formula that used the ultimate gain and ultimate period from a continuous cycle experiment and we compute the error of plant response corresponding to the initial value of parameter.

The new proportional gain (K_p), the integral time (T_i), and derivative time (T_d) were determined from EP, GA, PSO and ACO. This soft computing techniques for a PID controller considerably reduced the overshoot and rise time as compared to any other PID controller tuning algorithms, such as Ziegler-Nichols tuning method and continuous cycling method.

III GENETIC ALGORITHM

Genetic Algorithms (GA.s) are a stochastic global search method that mimics the process of natural evolution. It is one of the methods used for optimization. John Holland formally introduced this method in the United States in the 1970 at the University of Michigan. The continuing performance improvement of computational systems has made them attractive for some types of optimization. The genetic algorithm starts with no knowledge of the correct solution and depends entirely on responses from its environment and evolution operators such as reproduction, crossover and mutation to arrive at the best solution [1]. By starting at several independent points and searching in parallel, the algorithm avoids local minima and converging to sub optimal solutions.

A. Objective Function of the Genetic Algorithm

This is the most challenging part of creating a genetic algorithm is writing the the objective function. In this project, the objective function is required to evaluate the best PID controller for the system. An objective function could be created to find a PID controller that gives the smallest overshoot, fastest rise time or quickest settling time. However in order to combine all of these objectives it was decided to design an objective function that will minimize the performance indices of the controlled system instead. Each chromosome in the population is passed into the objective function one at a time. The chromosome is then evaluated and assigned a number to represent its fitness, the bigger its number the better its fitness [3]. The genetic algorithm uses the chromosomes fitness value to create a new population consisting of the fittest members. Each chromosome consists of three separate strings constituting a P, I and D term, as defined by the 3-row bounds declaration when creating the population [3]. When the chromosome enters the evaluation function, it is split up into its three Terms. The newly formed PID controller is placed in a unity feedback loop with the system transfer function. This will result in a reduce of the compilation time of the program. The system transfer function is defined in another file and imported as a global variable. The controlled system is then given a step input and the error is assessed using an error performance criterion such as Integral square error or in short ISE. The chromosome is assigned an overall fitness value according to the magnitude of the error, the smaller the error the larger the fitness value.

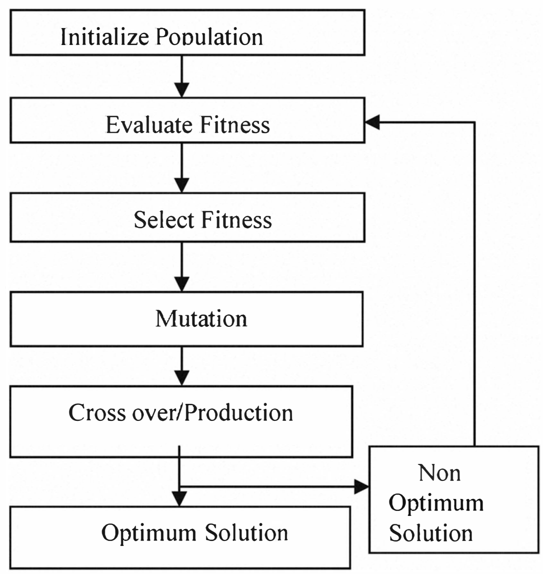

Fig 2 Flowchart of GA

IV EVOLUTIONARY PROGRAMMING

There are two important ways in which EP differs from GAs.

First, there is no constraint on the representation. The typical GA approach involves encoding the problem solutions as a string of representative tokens, the genome. In EP, the representation follows from the problem. A neural network can be represented in the same manner as it is implemented, for example, because the mutation operation does not demand a linear encoding [6].

Second, the mutation operation simply changes aspects of the solution according to a statistical distribution which weights minor variations in the behavior of the offspring as highly probable and substantial variations as increasingly unlikely.

The steps involved in creating and implementing evolutionary programming are as follows:

- Generate an initial, random population of individuals for a fixed size (according to conventional methods Kp, Ti, Td ranges declared).
- Evaluate their fitness (to minimize integral square error).
- Select the fittest members of the population.
- Execute mutation operation with low probability.
- Select the best chromosome using competition and selection.
- If the termination criteria reached (fitness function) then the process ends. If the termination criteria not reached search for another best chromosome

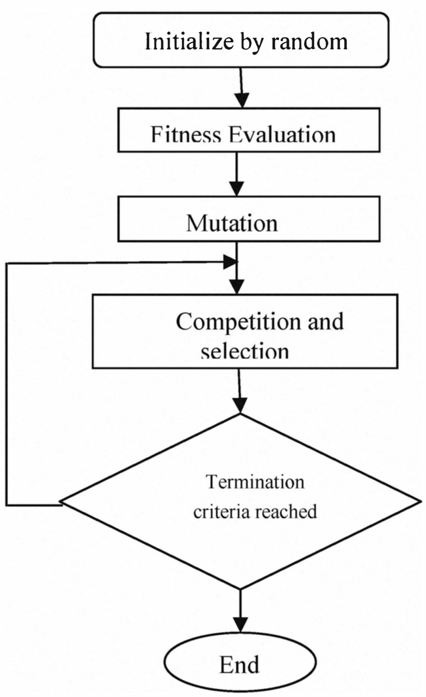

Fig 3 Flow Chart of EP

V PARTICLE SWARM OPTIMIZATION

PSO is one of the optimization techniques and kind of evolutionary computation technique .The technique is derived from research on swarm such as bird flocking and fish schooling. In the PSO algorithm, instead of using evolutionary operators such as mutation and crossover to manipulate algorithms, for a d-variable optimization Problem, a flock of particles are put into the d-dimensional Search space with randomly chosen velocities and positions knowing their best values. So far (p best) and the position in the d-dimensional space [7]. The velocity of each particle, adjusted accordingly to its own flying experience and the other particles flying experience [7].

For example, the i th particle is represented, as $x_i = (x_{i,1}, x_{i,2}, \ldots., x_{i,d})$

In the d-dimensional space. The best previous position of the i th particle is recorded as,

$$Pbest_i = (Pbest_{i,1}, Pbest_{i,2}, \ldots\ldots, Pbest_{i,d}) \text{---(2)}$$

The index of best particle among all of the particles in the group in g best d .The velocity for particle i is represented as

$$V_i = (V_{i,1}, V_{i,2}, \ldots\ldots., V_{i,d}) \quad \text{------------- (3)}$$

The modified velocity and position of each particle can be calculated using the current velocity and distance from P $best_{i,d}$ to $gbest_d$ as shown in the following formulas

$$V_{i,m}^{(t+1)} = W.V_{i,m}^{(t)} + c_1 * rand() * (Pbest_{i,m} - x_{i,m}^{(t)}) + c_2 * Rand() * (gbest_m - x_{i,m}^{(t)}) \quad \text{--- (4)}$$

$$x_{i,m}^{(t+1)} = x_{i,m}^{(t)} + v_{i,m}^{(t+1)}$$
$$i = 1,2,\ldots.,n$$
$$m = 1,2,\ldots..,d$$

Where
n= Number of particles in the group

d = dimension

t = Pointer of iterations (generations)

$V_{i,m}^{(1)}$ = Velocity of particle I at iteration t

W= Inertia weight factor

c_1, c_2 = Acceleration constant

$rand()$=Random number between 0 and 1

$x^{(t)}_{i,m}$ = Current position of particle i at iterations

$Pbest_{i,}$ = Best previous position of the ith particle

$gbest_{m}$ = Best particle among all the particles in the population

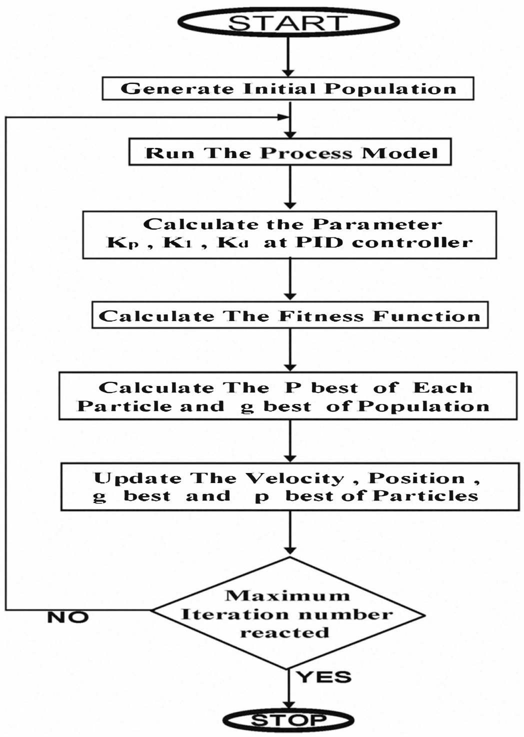

Fig 4. Flowchart of PSO

VI ANT COLONY OPTIMIZATION

ACO's are especially suited for finding solutions to different optimization problems. A colony of artificial ants cooperates to find good solutions, which are an emergent property of the ant's co-operative interaction .Based on their similarities with ant colonies in nature, ant algorithms are adaptive and robust and can be applied to different versions of the same problem as well as to different optimization problems

The main traits of artificial ants are taken from their natural model. These main traits are (1) artificial ants exist in colonies of cooperating individuals, (2) they communicate indirectly by depositing pheromone (3) they use a sequence of local moves to find the shortest path from a starting position, to a destination point (4) they apply a stochastic decision policy using local information only to find the best solution. If necessary in order to solve a particular optimization problem, artificial ants have been enriched with some additional capabilities not present in real ants.

An ant searches collectively foe a good solution to a given optimization problem. Each individual ant can find a solution or at least part of a solution to the optimization problem on its own but only when many ants work together they can find the optimal solution. Since the optimal solution can only be found through the global cooperation of all the ants in a colony, it is an emergent result of such this cooperation. While searching for a solution the ants do not communicate directly but indirectly by adding pheromone to the environment. Based on the specific problem an ant is given a starting state and moves through a sequence of neighboring states trying to find the shortest path. It moves based on a stochastic local search policy directed by its internal state, the pheromone trails, and local information encoded in the environment. Ants use this private and public information inorder to decide when and where to deposit pheromone. In most application the amount of pheromone deposited is proportional to the quality of the move an ant has made. Thus the more pheromone, the better the solution found. After an ant has found a solution, it dies; i.e.it is deleted from the system.

ACO uses a pheromone matrix $\tau = \{\tau_{ij}\}$

for the construction of potential good solutions. The initial values of ć are set $\tau_{ij} = \tau_0 \forall (i,j), where \tau_0 \rangle 0.$

The probability $P^{A}{}_{ij}\ (t)$ of choosing a node j at node I is defined in the equation (5). At each generation of the algorithm, the ant constructs a complete solution using (5), starting at source node.

$$p_{ij}{}^{A}(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{i,j\in T^{A}} \tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}} \quad if\ i, j \in T^{A} \qquad \text{-----------} (5)$$

Where

$$\eta_{ij} = \frac{1}{k_j}, \ j = [p, i, d]:$$

representing heuristic functions.

α and ß are constants that determine the relative influence of the pheromone values and the heuristic values on the decision of the ant.

T^A : is the path effectuated by the ant A at a given time.

The quantity of pheromone $\Delta \tau^{A}_{ij}$ on each path may be defined as,

$$\Delta \tau_{ij}^{A} = \begin{cases} \dfrac{L^{\min}}{L^{A}} & if \quad i, j \in T^{A} \\ 0 & \text{else} \end{cases} \quad -(6)$$

Where,

L^A is the value of the objective function found by the ant A.

L^{min} is the best solution carried out by the set of the ants until the current iteration.

Pheromone updation is determined by,

$$\sum_{k} \Delta \tau^{k} = \Delta \tau^{k=1} = \frac{\zeta f_{best}}{f_{worst}} \quad -(7)$$

The pheromone evaporation is a way to avoid unlimited increase of pheromone trails. Also it allows the forgetfulness of the bad choices.

$$\tau_{ij}(t) = p\,\tau_{ij}(t-1) + \sum_{A=1}^{NA} \Delta \tau_{ij}^{A}(t) \quad -(8)$$

Where-

NA; Number of ants

P; the evaporation rate. $0 < p <= 1$.

A. *IMPLEMENTATION ALGORITHM*

Step 1

Initialize randomly a potential solutions of the parameters (k p, k i, k d) by using uniform distribution. Initialize the pheromone trail and the heuristic value

Step 2

Place the A th ant on the node. Compute the heuristic value associated in the objective (minimize the error).

Step 3

Use pheromone updation (7) given by eqn to avoid unlimited increase of pheromone trails and allow the forgetfulness of bad choices.

Step 4

Evaluate the obtained solutions according to the objectives.

Step 5

Display the optimum values of the optimization parameters.

Step 6

Globally update the pheromone, according to the optimum solutions calculated at step 5. Iterate from step 2 until the maximum of iterations is reached.

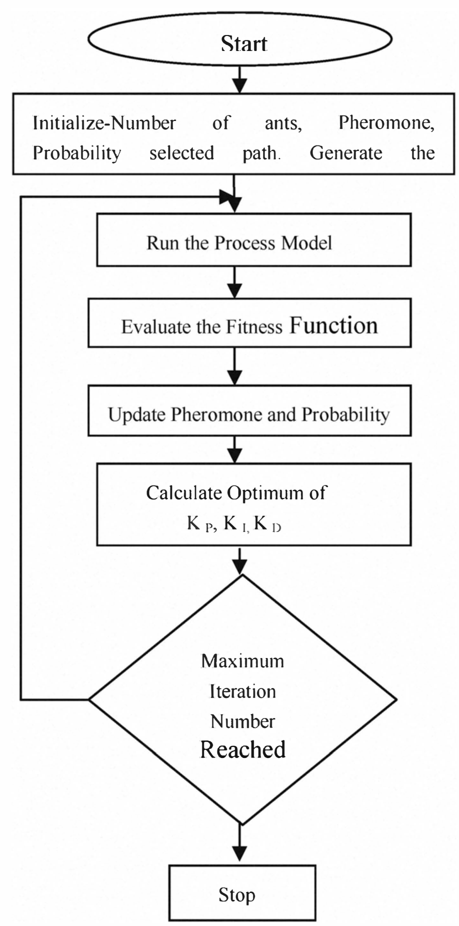

Fig 5. Flowchart of ACO

VII RESULTS AND DISCUSSIONS

In order to cover typical kinds of common industrial processes have been taken

Model1

$$G_1(S)=\frac{0.1433}{5.2*10^{-7}S^2+2.17*10^{-4}S+2.265} \quad [12]$$

Model2

$$G_2(S)=\frac{46.21S+206.1}{0.9372S^4+2.656S^3+75.87S^2+1121S} \quad [1]$$

Model3

$$G_3(S)=\frac{32.31}{s^2+51.1s} \quad [5]$$

VIII IMPLEMENTATION OF INTELLIGENT PID CONTROLLER TUNING

The Ziegler-Nichols tuning method using root locus and continuous cycling method were used to evaluate the PID gains for the system, using the “rlocfind” command in matlab, the cross over point and gain of the system were found respectively.

In this paper a time domain criterion is used for evaluating the PID controller. A set of good control parameters P, I, and D can yield a good step response that will result in performance criteria minimization in the time domain.

These performance criteria in the time domain include the over shoot rise time and setting time. To control the plant model the following ACO, PSO, EP and GA parameters are used to verify the performance of the PID controller Parameter

Performance characteristics of process model 1 to 3 were indicated and compared with the intelligent tuning methods as shown in the figure 6 to figure 8 and values are tabulated in table2 to table-5

Conventional methods of controller tuning lead to a large settling time, overshoot, rise time and steady state error of the controlled system. Hence Soft computing techniques is introduces into the control loop.

GA, EP, PSO and ACO based tuning methods have proved their excellence in giving better results by improving the steady state characteristics and performance indices.

TABLE I

PSO, GA, EP and ACO Parameters

PSO PARAMETERS	GA PARAMETERS	EP PARAMETERS	ACO PARAMETERS
Population size:100	Population size:100	Population size:100	Population size:100
Wmax=0.6	Mutation rate:0.1	Normal distribution	No. of Ants=10
Wmin=0.1	Arithmetic Crossover	Mutation rate:0.01	No. of Path=15
Iteration:100	Iteration:100	Iteration:100	Iteration:100
Fitnessfunction:ISE	Fitnessfunction:ISE	Fitnessfunction:ISE	Fitnessfunction:ISE

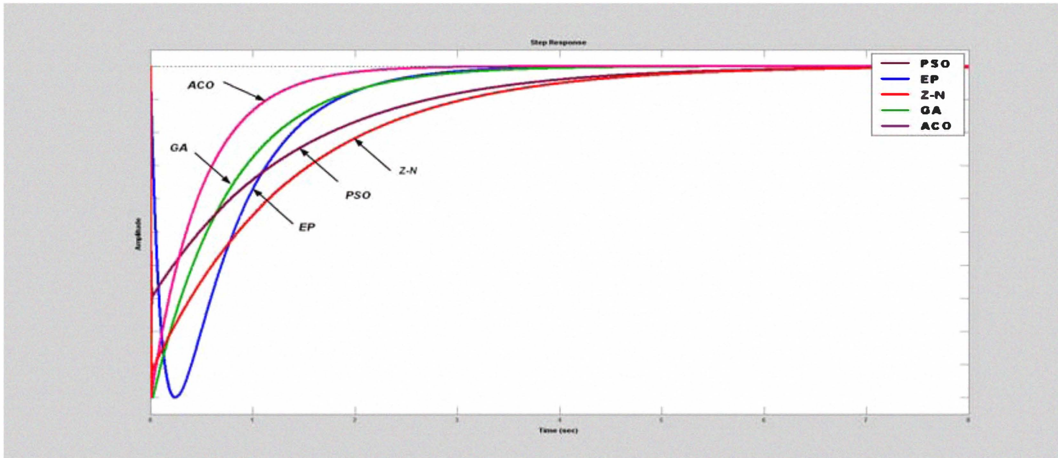

Fig 6 Comparison result of all methods for model 1

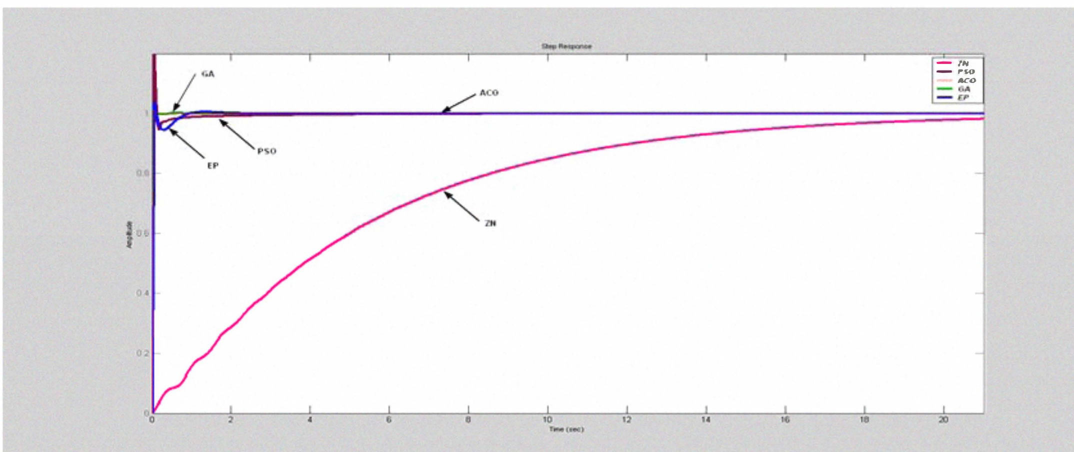

Fig 7 Comparison result of all methods for model 2

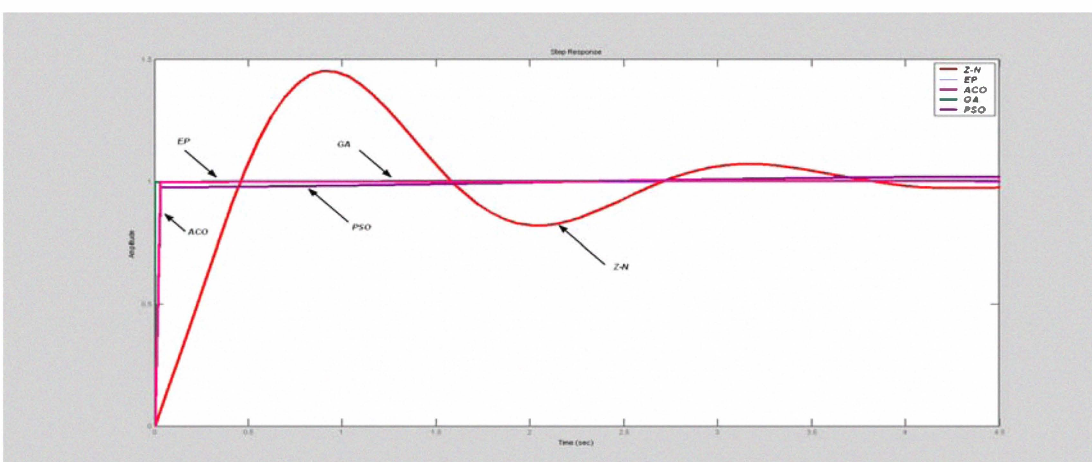

Fig 8 Comparison result of all methods for model 3

TABLE II

Comparison result of all methods for model 1

Characteristics	Z-N	GA	EP	PSO	ACO
Settling time (sec)	5.3	1.8	1.95	4.9	1.51
Rise Time (sec)	0.0	0.0	0.0	0.0	0.0
Over shoot (%)	1	0.0	0.0	0.0	0.0

TABLE III

Comparison result of all methods for model 2

Characteristics	Z-N	GA	EP	PSO	ACO
Settling time (sec)	20.4	0.023	0.43	0.0447	0.636
Rise Time (sec)	11.4	0.018	0.019	0.0365	0.0221
Over shoot (%)	1	0.6	23	1	3

TABLE IV

Comparison result of all methods for model 3

Characteristics	Z-N	GA	EP	PSO	ACO
Settling time (sec)	4.58	0.00315	0.0257	0.0301	0.767
Rise Time (sec)	0.34	0.00257	0.0209	0.0246	0.0263
Over shoot (%)	45.3	0.0365	1.32	0.22	1.93

IX CONCLUSION

Research work has been carried out to get an optimal PID tuning by using GA, EP, PSO and ACO. Simulation results demonstrate the tuning methods that have a better control performance compared with the conventional ones. It is possible to consider several design criteria in a balanced and unified way. Approximations that are typical to classical tuning rules are not needed.Soft computing techniques are often criticized for two reasons: algorithms are computationally heavy and convergence to the optimal Solution cannot be guaranteed. PID controller tuning is a small-scale problem and thus computational complexity is not really an issue here. It took only a couple of seconds to solve the problem. Compared to conventionally tuned system, GA, EP, PSO and ACO tuned system has good steady state response and performance indices.

REFERENCES

[1].Ian griffin,Jennifer bruton "On-Line Pid controller tuning using genetic algorithm" URL; www.eeng.dcu.ie/~brutonj/Reports/IGriffin_MEng_03.pdf

[2]. M.B.B. Sharifian,R.Rahnavard and H.Delavari "Velocity Control of DC Motor Based Intelligent methods and Optimal Integral State FeedbackController" Internatonal Journal of Computer theory and engineering,vol.1,No.1,april 2009.

[3] Neenu Thomas, Dr. P. Poongodi "Position Control of DC Motor Using Genetic Algorithm Based PID Controller"Proceedings of the World Congress on Engineering 2009 Vol II WCE 2009, July 1 - 3, 2009, London, U.K.

[4]Asim ali khan,Nushkam rapal "Fuzzy pid controller:Design,Tuning and comparison with conventional PID controller" 1-4244-0457-6/06/$20 2006 IEEE

[5]Sankhadip saha "Performance Comparison of Pid base Position control system with FLC based position control system" TIG research Journal,vol.1,No.2 sep 2008.

[6]Jukka Lieslehto "PID controller tuning using Evolutionary programming" American Control Conference, VA June 25-27,2001

[7]Mehdi Nasri, Hossein Nezamabadi-pour, and Malihe Maghfoori "A PSO-Based Optimum Design of PID Controller for a Linear Brushless DC Motor" World Academy of Science, Engineering and Technology 26 2007.

[8]B.Nagaraj S.Subha B.Rampriya "Tuning Algorithms for PID Controller Using Soft Computing Techniques" IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.4, April 2008

[9]Hyung-soo Hwang, Jeong-Nae Choi, Won-Hyok Lee: "A Tuning Algorithm for The PID controller Utilizing Fuzzy Theory", IEEE Proceedings 0-7803-5529-6/99/ Q1999 IEEE.PP:2210-2215.

[10].Jan Jantzen: "Tuning of fuzzy PID controllers"Denmark.Tech. Report no 98- H 871(fpid), 30 Sep 1998.PP:1-22.

[11].Kiam Heong Ang, Gregory Chong: "PID Control System Analysis, Design, and Technology" IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, VOL.13, No.4, July 2005 PP: 559-576.

[12].Ibtissem Chiha, Pierre Borne: "Multi-Objective Ant Colony Optimization to tuning PID Controller" proceedings of the International journal of engineering, Volume III, Issue no 2,march 2010.

[13]G.Dicaro and M.Dorigo "Ant colonies for adaptive routing in packet switched communications network". In A.E.Eiben , T.Back, M.Schoenauer, and H-P. Schwefel ,editors, proceedings of PPSN-V, 5th international conference on parallel problem solving from nature, volume 1498 of lecture notes in csc, pages 673-682. Springer verlag, Berlin, Germany, 1998.

[14]G.Zhou and J.D.Birdwell, "Fuzzy logic- based PID autotuner design using simulated annealing", in proceedings of the IEEE/IFAC joint symposium on computer-aided control system design pp.67-72, march 7-9,1994.

[15]H.Ying-Tung, C.Cheng-Long and C.Cheng-Chih, "Ant colony optimization for designing of PID controllers",IEEE International Symposium on computer aided control systems design Taipei,Taiwan,September 24,2004.

[16] B.Nagaraj and N.Murugananth"A comparative approach approach of soft computing methodologies for industrial process tuning", published in KJER,Japan, vol II, Dec 2009.

B.Nagaraj is a Assistant plant engineer at Tamilnadu nadu News print and paper ltd. He received his B.E. and M.E. degrees in Instrumentation and control engineering from Arulmigu Kalasalingam College of Engg in 2004 and 2006, respectively.He is pursuing Ph.D degree in Karpagam university. He is a active member of instrument society of India, and Indian society for technical education. He has Published his research work in 4 International and 2 National journal.

Dr.N.Murugananth.,Ph.D.,D.Eng., is a graduate of University of South Florida (MS-1996), FSU (Ph.D-1998). Picasse award (1999), Edison award (2001,2002,2003), BNL fellowship award (2003), Carl Bucholt award (2005). Dr. Murugananth is vice president, GE, Energy and has contributed to various research innovations. He is co-director of NRL, Secretary- World Research scientist's organization, Director-Futuristic Technology, MIT, Co-ordinator, Defense Research Board, US Govt. He holds 24 patents in various fields of research, 2 patents on concepts, 19 publications. He is the Honorary member of the Jack Welsh Science and technology, JSF, Asian federation of engineers, GYSO, OSA, Florida optical forum, to name a few. His research interests are futuristic tech. in teleportation, photonics, adonics, Mobionics, UPP, Energy, space system, etc. He received the Honorary Doctorate in Power & Energy from Stanford University, USA.