Soft Computing Based PID controller Design for Consistency Control in Papermaking

Nagaraj B.[1] & Vijayakumar P.[2]

1. Karpagam College of Engineering, Coimbatore, India
2. Tamilnadu News Print and Paper Ltd., P.O. Kagithapuram-639136 (India)

ABSTRACT

Proportional - Integral - Derivative (PID) control schemes continue to provide the simplest and effective solutions to most of the control engineering applications today. However, PID controller is poorly tuned in practice with most of the tuning done manually, which is difficult and time consuming. This research comes up with a soft computing approach involving Genetic Algorithm (GA), Evolutionary Programming (EP), Particle Swarm Optimization (PSO) and Bacterial Foraging Optimization (BFO). The proposed algorithm is used to tune the PID parameters and its performance has been compared with the conventional method of Ziegler Nichols. The results obtained reflect that use of soft computing based controller improves the performance of the process in terms of time domain specifications and performance index. This paper discusses in detail the Soft computing technique and its implementation in PID tuning for a controller of a consistency process in papermaking. Compared to other conventional PID tuning methods, the result shows that better performance can be achieved with the soft computing based tuning method. The ability of the designed controller in terms of tracking set point is also compared and simulation results are shown.

***Keywords**: Genetic algorithm, Particle swarm optimization, PID controller and bacterial foraging*

Introduction:

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 set point [1]. The PID controller calculation involves three separate parameters, namely 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 [2]. The goal of PID controller tuning is to determine parameters that meet closed loop system performance specifications. The robust performance of the control loop over a wide range of operating conditions should also be ensured. Practically, it is often difficult to simultaneously achieve all of these desirable qualities. For example, if the PID controller is adjusted to provide better transient response to set point change, it usually results in a sluggish response when under disturbance conditions [3]. In this paper, soft computing approach is applied to optimally design a PID controller, for a consistency control in Paper mill.

Development Of A Mathematical Model Of Real Time Process

Today consistency control remains the most fundamental and important measurement in the pulp and paper industry. Unfortunately, consistency control systems are generally the worst performing in the mill. Without uniform consistency, the pulp and paper process is practically impossible to control. Changes in consistency must be monitored continuously, accurately and online, if operators and engineers are to manage the process better. It is not only important to be able to sense variations in consistency, but equally important is to determine the absolute value of the consistency. The term consistency, as used in the pulp and paper industry, is defined as the percentage of weight of bone dry fibrous material in any combination of pulp and water, or stock (pulp and additives) and water. Based on the set point of the controller, if the consistency value exceeds the set point value, controller actuates the control valve to open accordingly and increases dilution water flow in the line. If the consistency value falls below the set point value, the control valve will close to reduce dilution water flow in the line and improve the consistency thereby maintaining the consistency of the chest/tower. The process and instrumentation diagram of blend chest consistency control is shown in figure 1.

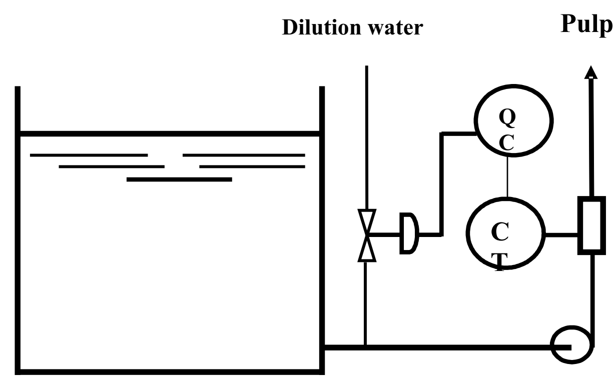


Fig.1 Blend chest consistency control

Real-time process setup as available in our mill is given in Figure 8.

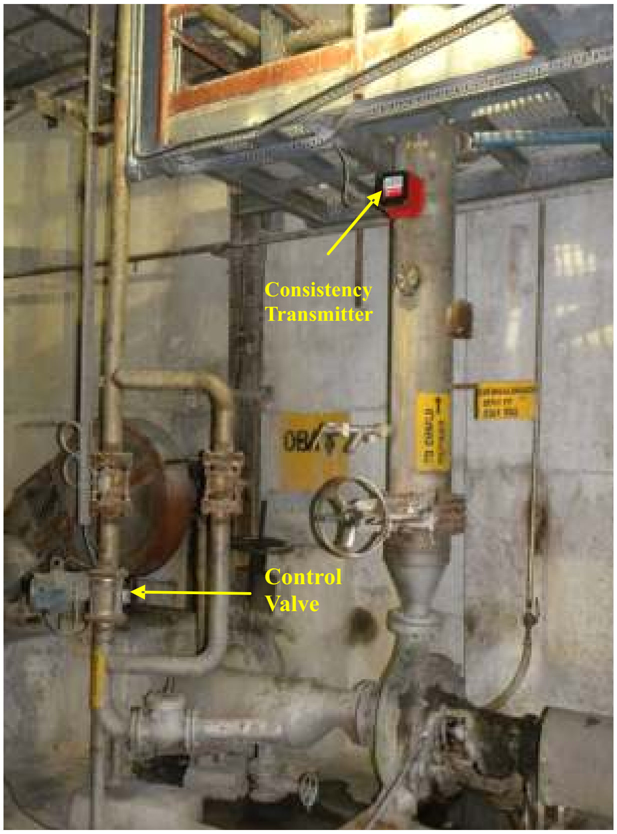


Fig. 8 Real-time Process setup Consistency control

Identification Of Process Parametrs

This case study concerns data collected from a ABB-DCS for blend chest consistency control. Thick stock is fed into the head box through main stock valve, after pressing and drying process, paper is formed. Before head box, consistency control in blend chest is very important to maintain the high quality paper production. In our mill, most of the consistency transmitter uses the shear force principle to measure consistency with a motion balance principle. The transmitter is mounted on the pipeline by means of welded stud and the sensing element a blade-is positioned directly in the pulp flow. Due to the shear force on the sensing blade, it moves the sensor to a certain distance. These signals are interpreted by means of a differential capacitor with electronic output at 4-20mA.

The input is the consistency of the blend chest or the output (4-20mA) from the consistency transmitter. The output is the flow rate of dilution water or dilution water control valve open position. One thousand input and output samples from this process are taken from the ABB DCS and measured data are used to identify the system open loop transfer function Eq. (1). Table 3 in shows the technical specifications of the setup.

$$G_{consis}(s) = e^{-3s} \frac{0.08726s + 0.1846}{s^2 + 0.4394s + 0.2404} \quad \text{Equation ------ (1)}$$

Table 3. Technical specifications of the real time setup

Part Name	Details
DCS System AC 450 Controller	Operate IT – HMI, ABB Analog Input/output :4- 20 mA AMPL – Programming
MEK Transmitter	Shear force measuring. Output – (4 – 20mA) BTG Make HART Protocol. 24V DC
Dilution Control Value	Size:4", Air to open & Fisher Make
Blend Chest Height	6 Meter
Machine Speed	700 MPM

After obtaining the system transfer function, it's tuned through the conventional ZN method as well as the soft computing techniques, with the help of MATLAB simulation using the unit step input. This section presents about the details of the arrangement of the Real time environment and experiment work carried out in Tamil Nadu News Print and Papers Ltd, for the consistency process. The Experimental setup used in this work as shown in Figure 8. BTG consistency transmitter is used to measure the consistency. The measured signal is transmitter into ABB DCS through analog input card AI-810. The ABB AC450 controller is connected with the field using analog output card AO-810. The connection established between the analog output card and AC450 through communication interface CI820. The difference of the measured consistency and desired consistency is transmitted from AC 450 controller to dilution valve to regulating the consistency. This valve actuated by a current signal (4-20 MA). By varying the current signal dilution water flow can be linearly varied.

Material And Methods

The classical method of Ziegler Nichos is employed to find out the values of *Kp, Ki* and *Kd*. Although the classical methods cannot provide the best solution [4], they give the initial values or boundary values needed to start the soft computing algorithms. Due to the high potential of evolutionary techniques such as EP, GA, PSO and BFO methods in finding the optimal solutions, the best values of *Kp, Ki* and *Kd* are obtained. The simulations are carried out using INTEL[R], Pentium [R] CPU 3 GHZ, 4GB RAM in MATLAB 7.10 environments. The Ziegler-Nichols tuning method using root locus and continuous cycling method were used to evaluate the PID gains for the system [4], using the "rlocfind" command in matlab, the cross over point and gain of the system were found respectively. The initial values of PID gain are calculated using conventional Z N method. The advantages of using evolutionary techniques for PID are listed below, these techniques can be applied for higher order systems without model reduction [5]. These methods can also optimize the design criteria such as gain margin, Phase margin, closed loop band width when the system is subjected to step change in load. [6]. Evolutionary techniques like Genetic Algorithm, Evolutionary Programming, Particle Swarm Optimization and Bacterial Foraging Optimization methods have proved their excellence in giving better results by improving the steady state characteristics and performance indices.

GA Based Tuning Of The Controller

This is the most challenging part of creating a genetic algorithm, and that is writing the objective functions. In this project, the objective function is required to evaluate the best PID controller for the 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 [6, 7]. 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 biggest its number the better its fitness. The genetic algorithm uses the chromosome 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. When the chromosome enters the evaluation function, it is split up into its three terms [8, 9]. The newly formed PID controller is placed in a unity feedback loop with the system transfer function. This will result in a reduction of the compilation time of the program. The system transfer function is defined

TABLE 1. PSO, GA, EP and BFO parameters

PSO Parameters	GA Parameters	EP Parameters	BFO Parameters
Population size:100	Population size:100	Population size:100	Number of Bacterium =5
Wmax=0.6/ Wmin=0.1	Mutation rate:0.1	Normal distribution	Number of iteration in a Chemotactic loop (N_c) =10
C1 = C2 = 1.5	Arithmetic Crossover	Mutation rate:0.01	Number of reproduction (N_{re}) =15 Number of Parameters (P) =3
Iteration:100	Iteration:100	Iteration:100	$W_{attract}$ =0.04 $D_{attract}$ =0.01
Fitness function:ISE	Fitness function:ISE	Fitness function:ISE	$H_{repellent}$ =0.01 $W_{repellent}$ =10 Fitness functions :ISE

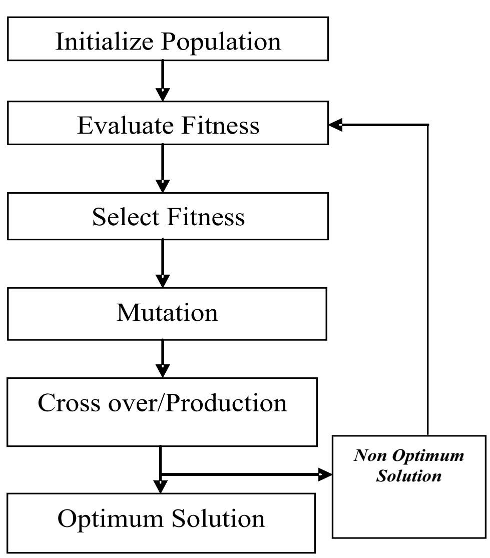


Fig. 2Flowchart of GA

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 squared error or in short ISE [10]. Initializing the values of the parameters is as per Table 1. The flowchart of the GA control system is shown in figure 2.

$$ISE= \int_{0} e^{2}(t)dt$$

EP Based Tuning Of Controller

There are two important ways in which EP differs from GA. 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 [5].

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:

1. Generate an initial, random population of individuals for a fixed size (according to conventional methods K_p, K_i, K_d ranges declared).
2. Evaluate their fitness (to minimize integral squared error).

$$ISE= \int_{0} e^{2}(t)dt$$

3. Select the fittest members of the population.
4. Execute mutation operation with low probability.
5. Select the best chromosome using competition and selection.
6. If the termination criteria reached (fitness function) then the process ends. If the termination criteria not reached, search for another best chromosome. The EP initializing parameters chosen are given in Table 1. The flowchart of the EP control system is shown in figure 3.

PSO Based Tuning of Controller

PSO is one of the optimization techniques and a 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 [11, 12, 13].

The algorithm proposed in [16] uses a 2-D approach for searching within the solution space. For this study the PSO algorithm will be applied to a 2-D or 3-D solution space in search of optimal tuning parameters for PI, PD and PID control.

Consider position $X_{i,m}$ of the *i-th* particle as it traverses a n-dimensional search space: The previous best position for this *i*-

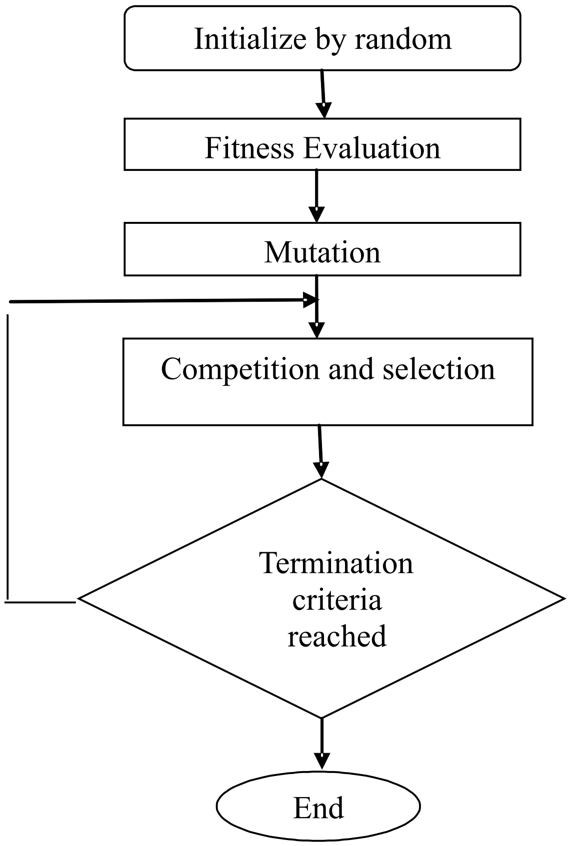


Fig. 3 Flow Chart of EP

th particle is recorded and represented as $pbest_{i,n}$. The best performing particle among the swarm population is denoted as $gbest_{i,n}$ and the velocity of each particle within the n-dimension is represented as $V_{i,n}$. The new velocity and position for each particle can be calculated from its current velocity and distance respectively [14, 15].

p best and the position in the d-dimensional space. The velocity of each particle, is adjusted accordingly to its own flying experience and the other particles flying experience [16].

For example, the i th particle is represented, as

$$X_i \quad X_{i,1}, X_{i,2}, \ldots\ldots X_{i,d})$$

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

$$Pbest_i \quad Pbest_{i,1}, Pbest_{i,2}, \ldots\ldots\ldots, Pbest_{i,d} \qquad \text{Equation (2)}$$

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

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

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

$$V_{i,m}^{t\ 1} \quad W.V_{i,m}^{t} \quad c_1 * rand * \ Pbest_{i,m} \quad X_{i,m}^{t} \quad c_2 * Rand * \ gbest_m \quad X_{i,m}^{t}$$

Equation (4)

$$X_{i,m}^{t\ 1} \quad X_{i,m}^{t} \quad V_{i,m}^{t\ 1} \qquad \text{Equation (5)}$$

i = 1,2, ... , n
m= 1,2, ... , 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

$X_{i,m}^{t}$ = 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

In the proposed PSO method each particle contains three members P, I and D. It means that the search space has three dimension and particles must 'fly' in a three dimensional space. Initializing the values of the parameters is as per Table 1. The flowchart of the PSO PID control system is shown in figure 4.

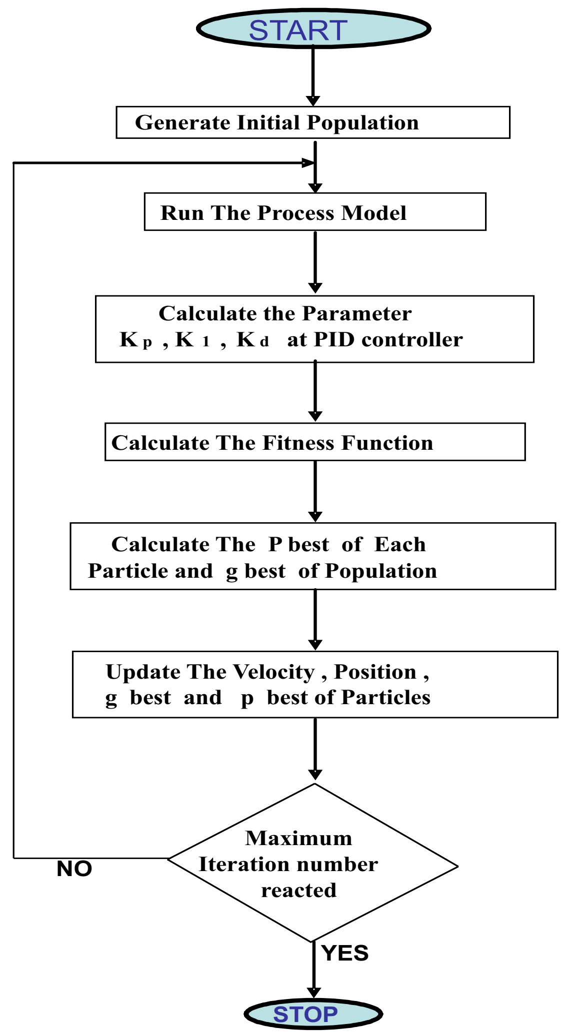


Fig. 4 Flowchart of PSO

Bacteria Foraging Otimization

The survival of species in any natural evolutionary process depends upon their fitness criteria, which relies upon their food searching and motile behavior. The law of evolution supports those species who have better food searching ability and either eliminates or reshapes those with poor search ability. The genes of those species who are stronger gets propagated in the evolution chain since they possess ability to reproduce even better species in future generations. So, a clear understanding and modeling of foraging behavior in any of the evolutionary species, leads to its application in any nonlinear system optimization algorithm. The foraging strategy of Escherichia coli bacteria present in human intestine can be explained by four processes, namely ,chemotaxis, swarming, reproduction, and elimination dispersal [17].

A. Chemotaxis

The characteristics of movement of bacteria in search of food can be defined in two ways, i.e. swimming and tumbling together known as chemotaxis. A bacterium is said to be 'swimming' if it moves in a predefined direction, and 'tumbling' if moving in a random direction. Mathematically, tumble of any bacterium can be represented by a unit length of random direction φ *(j)* multiplied by step length of that bacterium C(i). In case of swimming, this random length is predefined.

B. Swarming

For the bacteria to reach at the richest food location, it is desired that the optimum bacterium till a point of time in the search period should try to attract other bacteria so that together they conquer the desired location more rapidly. To achieve this, a penalty function based upon the relative distances of each bacterium from the fittest bacterium till that search duration, is added to the original cost function. Finally, when all the bacteria have merged into the solution point, this penalty function becomes zero. The effect of swarming is to make the bacteria congregate into groups and move as concentric patterns with high bacterial density [18].

C. Reproduction

The original set of bacteria, after getting evolved through several chemotaxis stages reaches the reproduction stage. Here, best set of bacteria gets divided into two groups. The healthier half replaces with the other half of bacteria, which gets eliminated, owing to their poorer foraging abilities. This makes the population of bacteria constant in the evolution process [18].

D. Elimination and dispersal

In the evolution process, a sudden unforeseen event can occur, which may drastically alter the smooth process of evolution and cause the elimination of the set of bacteria and/or disperse them to a new environment. Most ironically, instead of disturbing the usual chemo tactic growth of the set of bacteria, this unknown event may place a newer set of bacteria nearer to the food location. From a broad perspective, elimination, and dispersal are parts of the population level long distance motile behavior. In its application to optimization, it helps in reducing the behavior of stagnation often seen in such parallel search algorithms. The flow chart of BFO control system is shown in figure 5.

Results And Discussion

The process is modeled for designing blend chest consistency control loop by using DCS available in the our mill. A transfer function to validate the consistency control process is obtained with the real time data using Matlab system identification toolbox, and given by Eq. (1). The tuned values through the traditional, as well as the proposed techniques, are analyzed for their responses to a unit step input, with the help of Matlab simulation and, then, the real time application for the

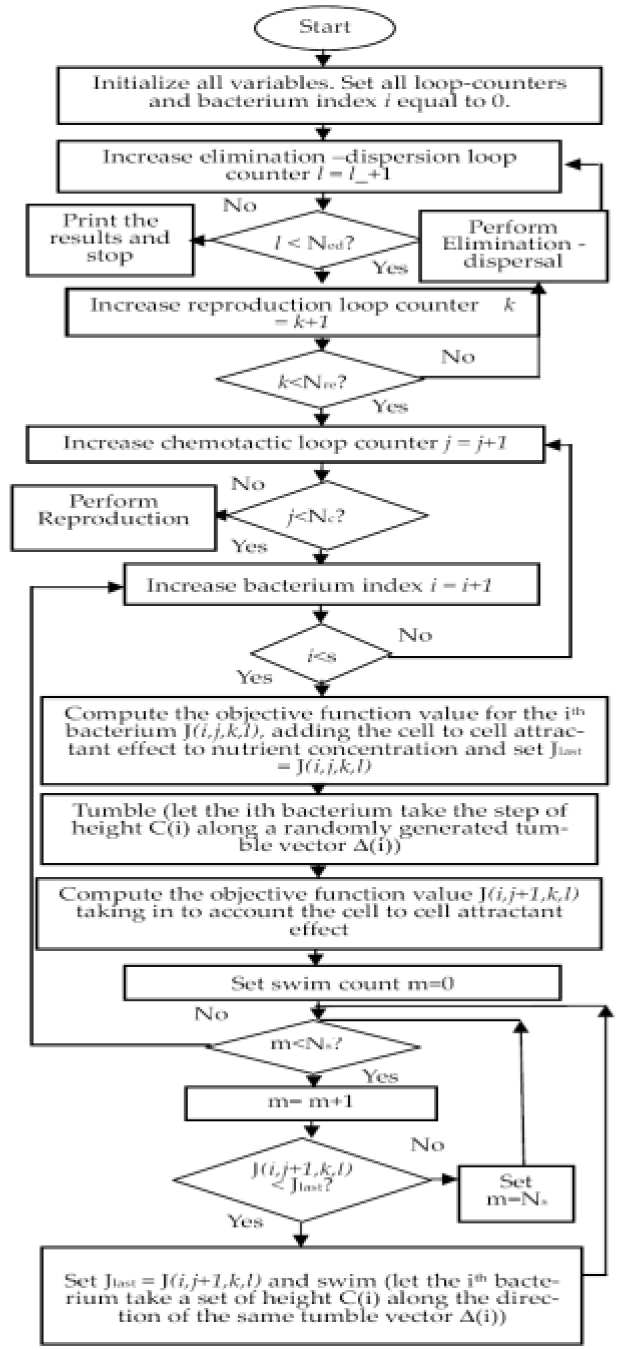


Fig.5. Flowchart for BFO

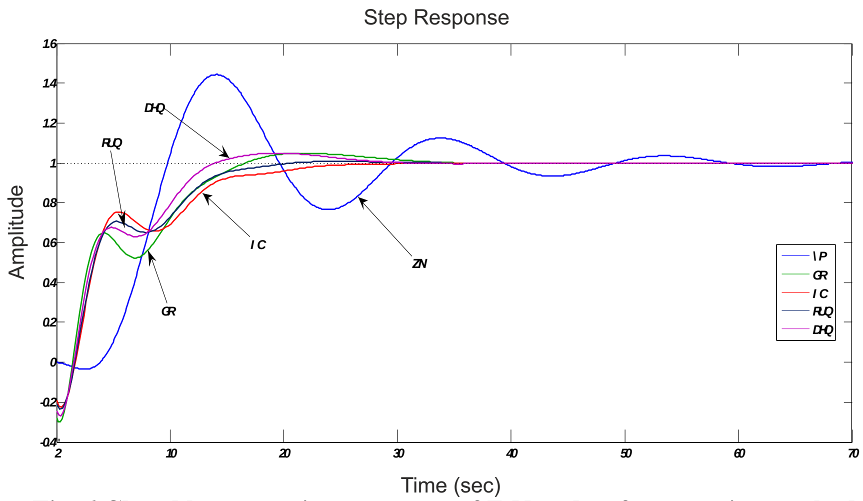


Fig. 6 Closed loop transient response of Z-N and soft computing methods

TABLE 2. PID Parameters and closed loop response specifications for consistency control

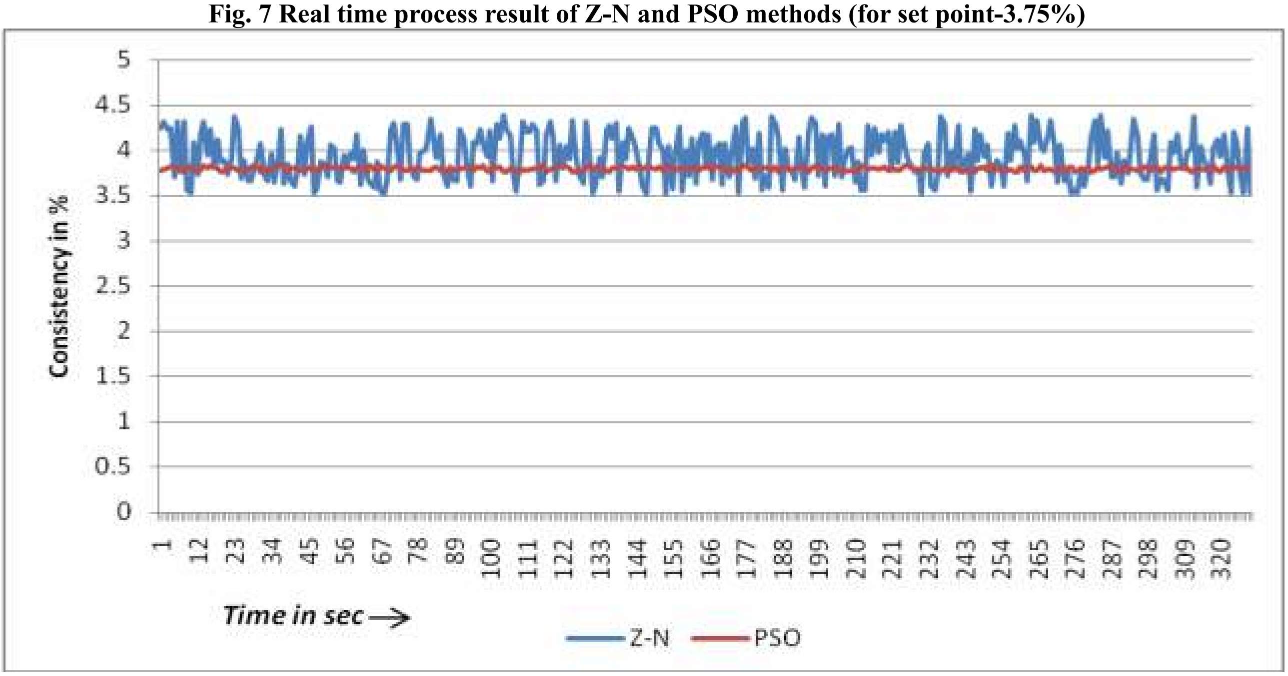
Tuning Method	K_p (Proportional gain)	K_i (Integral gain)	K_d (Derivative Gain)	Dynamic performance specifications Tr (Rise time)	Ts (Settling time)	Mp(%) (Peak overshoot)	Performance Index ISE (Integral square error)
ùb	0.0609	0.2363	0.0039	8.2	56.62	44.283	79.641
EP	0.4	0.22	2.4976	11.3	28	4.8%	52
GA	0.70	0.1986	1.8548	11.9	21.9	0%	42
PSO	0.5934	0.2148	1.986	9.9435	13.3104	1.8969	39.5811
BFO	0.14748	0.2506	2.2640	9.9479	24.6194	4.9	40.90

consistency control in the blend chest is presented. A tabulation of the time domain specifications comparison and the performance index comparison for the obtained models with the designed controllers is presented. The classical methods Zigler Nichols is employed to find out the values of K_p, K_i and K_d. Although the classical method is not able to provide the best solution, they give the initial values or boundary values needed to start the soft computing algorithms. Due to the high potential of evolutionary techniques, such as, EP, GA, PSO, BFO methods in finding the optimal solutions, the best values of *Kp, Ki* and *kd* are obtained. From Table 2, the PSO tuned system displays a best performance among the BFO, GA, EP and ZN by achieving an ISE of 39.5811. The closed-loop step response (transient response) for the different tuning methods is illustrated in Figure 6. The response specifications and performance index for the consistency control loop are given in Table 2. From Figure 6 and Table 2, the BFO, GA and EP method yields a system with marginally higher overshoot, longer settling and rise time in comparison to the PSO method. The closed-loop response for the Z-N method yields higher overshoot and longer settling time. The PSO method delivers superior control performance with improved dynamic performance specifications over the other tuning methods.

Real Time Response Of Blend Chest Consistency Control

Figure 6 illustrates that PSO has proved its excellence by producing high quality of the solution when compared to GA, EP and BFO. With these optimized values of K_p, K_i and K_d obtained as a result of PSO, the system settles down within 13.3104 seconds and with minimum peak overshoot 1.8969%.

Fig. 7 Real time process result of Z-N and PSO methods (for set point-3.75%)



The most important aspect of the paper is presented in this section. Further, to prove to the potential of soft computing methods in solving the real-time problems, the experimentation is done in ABB DCS of TNPL plant for blend chest control loop. The designed settings for the process were implemented for one set point. The ABB DCS is fed with optimized value PSO based controller parameter (K_p, K_i and K_d) for the consistency control process. The real time response of the system was observed by giving a set point of 3.75%, and the corresponding variation of consistency from a set point was recorded. The response of the consistency process for a set point (3.75) is presented in Figure 7. It is clear from the responses that the PSO based controller has the advantage of a better closed loop time constant, which enables the controller to act faster with a balanced overshoot and settling time. The response of the conventional controller is more sluggish than the PSO based controller.

Conclusion

Research work has been carried out to get an optimal PID tuning by using GA, EP, PSO and BFO for a blend chest consistency control process. The Soft computing technique is applied to a real time control of a blend chest consistency system using ABB AC450 DCS. The performance of the soft computing based controller is compared with conventional PID controller tuning settings. The performance is compared for set point 3.5% consistency. For the conventional controller, set point tracking performance is characterized by lack of smooth transition as well it has more oscillations. Also, it takes much time to reach set point. The Soft computing based controller tracks the set point faster and maintains steady state. It was found for a consistency control in blend chest process for 3.75% set point, that the performance of the Soft computing based controller was much superior to the conventional control. 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, PSO tuned system has good steady state response and performance indices.

References

[1] Lelic M. A. Reference Guide to PID Controllers in the Nineties, In Proceedings of IFAC Workshop: Past, Present and Future of PID Control.75-78, 2000.

[2] Ziegler, J.G. and Nichols. N.B. Optimum settings for automatic controllers. Trans. ASME. 64, 1942.

[3] Astrom K.J. and Hagglund .T PID Controllers: Theory, Design, and Tuning. ISA Triangle Park, 140-148, 1995.

[4] Ellis G. Control Systems Design Guide. Academic Press, London.1991.

[5] Jukka Lieslehto. PID Controller Tuning Using Evolutionary Programming, Proceedings of the American Control Conference Arlington.2828-2833,2001.

[6] Giriraj Kumar S.S, Jain .R, Anantharaman .N, Dharmalingam .V, Sheriffa Begum. Genetic algorithm based PID controller tuning for a model bioreactor, Indian Chemical Engineer Journal.50 (3),214-226,2008.

[7] Nara .S, Khatri .P, Garg .J. Proportional - Integral - Deriative Controller Tuning of Temperature Control System Using Genetic Algorithm, International Journal of Electronics, Information and System,12, 37-42,2010.

[8] Andrey Popov, Adel Parag and Herbert Werner. Tuning of a PID controller Using a Multi-objective Optimization Technique Applied to A Neutralization Plant, Proceeding of the 44th IEEE Conference on Decision and Control, and the European Conference, 7139-7143,2005.

[9] Jin-Sung Kim, Jin-Hwan Kim, Ji-Mo Park, Sung-Man Park, Won-Yong Choe. Auto Tuning PID controller based on Improved Genetic Algorithm for Reverse Osmosis Plant, World Academy of Science , Engineering and Technology .47,384-389,2008

[10] Neenu Thomas, Poongodi .P. Position Control of Motor Using Genetic Algorithm Based PID Controller, Proceeding of the World Congress on Engineering. 2,2009.

[11] Boumediene Allaoua, Brahim Gasbaoui, Brahim Mebarki. Setting up PID DC motor speed control alteration parameters using particle swarm optimization strategy, Leonardo Electronic Journal of Practices and Technologies.14,19-32, 2009.

[12] Mehdi Nasri, Hossein Nezamabad-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 26,210-218, 2007.

[13] Pillay.N .A. Particle Swarm Optimization approach Tuning of SISO PID control loops, Master thesis Light Current Electronics, Durban University of Technology,2009.

[14] Girirajkumar S.M., Deepak Jayaraj, and Anoop.R. Kishan. PSO based Tuning of a PID Controller for a High Performance Drilling Machine,International Journal of Computer Applications, 1(19),12-18,2010

[15] Mohammed EI-Said EI-Telbany. Employing Particle Swarm Optimizer and Genetic Algorithms for Optimal Tuning of PID Controllers: A Comparative Study,ICGST-ACSE Journal, 7(2),49-54,2007.

[16] Giriraj Kumar S.M., Sivasankar .R, Radhakrishnan .T.K, Dharmalingam .V and Anantharaman .V. Particle Swarm Optimization Technique Based Design of Pi Controller for a Real-Time Non-Linear Process, Instrumentation Science and Technology.36,525-542,2008.

[17] Salim Ali E. and S.M. Abad-Elazin. Optimal PID Tuning for Load Frequency Control Using Bacteria Foraging Optimization Algorithm, Conference Proceedings (MEPCON'10), Cairo University, Egypt,19-21, 2010.

[18] Dong Hwa Kim, Ajith Abraham and Jae Hoon Cho. A hybrid genetic algorithm and bacterial foraging approach for global optimization, Information Sciences 177, 3918-3937,2007.

[19] Jain .T and Nigam .M.J. Optimization of PD-PI Controller Using Swarm Intelligence, Journal of Theoretical and Applied Information Technology,1013-1018,2007.