Indian Journal of Science and Technology, *Vol 7(S7), 140–145, November 2014*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

PID Controller Tuning using Soft Computing Methodologies for Industrial Process- A Comparative Approach

T. K. Sethuramalingam[1]* and B. Nagaraj[2]

[1]Department of EEE, AMET University, Chennai - 603112, India; tksethuramalingam@gmail.com
[2]Karpagam College of Engineering, Coimbatore – 641032, India

Abstract:

Industrial Processes are non-linear, time variant and sensitive to disturbances. They are extremely complex and highly resistant to control. The dynamics of process offers great difficulty in controller design. Proportional, Integral, Derivative controller has been used in the process. In this context Z-N method has been used for controller tuning. Some disadvantages of Z-N method control technique for tuning a PID controller are: excessive number of rules to set the gain, inadequate dynamics of closed loop responses, and difficulty to deal with non-linear responses. The aim of this study is to develop a soft computing based PID control tuning methodology for optimizing the controller for the fitness function. In this approach, the transfer function of the marine processes has been taken for a simulation using MATLAB. From this simulation, outputs have proved their excellence in giving better results by improving the steady state characteristics and performance indices.

Keywords: Derivative controller, Integral, Non-Linear, Proportional, Soft Computing, Time Variant.

1 Introduction:

Conventional proportional integral derivative (PID) controller is widely used in many industrial applications due to its simplicity in structure and ease to design[1]. However, it is difficult to achieve the desired control performance. Tuning is an important parameter for the best performance of PID controllers.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[2]. Soft computing techniques like GA, PSO, and EP methods have proved their excellence in giving better results by improving the steady state characteristics and performance indices.

1.1. Proportional Integral Derivative Controller

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 the derivative value determines the reaction based on the rate at which the error has been changing the weighted sum of these three actions that 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 (Figure 1)[3].

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

2. Reason for Selecting Soft Computing Techniques:

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[4].

**Author for correspondence*

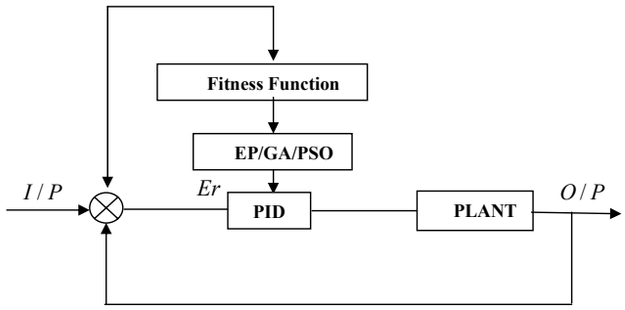

Figure 1. Block diagram of Intelligent PID controller.

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[4].

Approximations: Some approximations are often applied in order to keep the tuning rules simple[4].

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, and PSO 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 initial value of parameter Kp, T_i, T_d by the Ziegler-Nichols formula that uses 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, and PSO. These soft computing techniques for a PID controller considerably reduce the overshoot and rise time as compared with any other PID controller tuning algorithms, such as Ziegler-Nichols tuning method and continuous cycling method.[8]

2.1 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 converges to sub optimal solutions.

2.1.1 *Objective Function of the Genetic Algorithm*

The most challenging part of creating a genetic algorithm is writing the objective function. In this study, 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[2]. The genetic algorithm uses the chromosome's 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[2]. 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 reduction 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[8]. The smaller the error, the larger the fitness value. The operation flow for the genetic algorithm is shown in Figure 2.

2.2 Evolutionary Programming

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

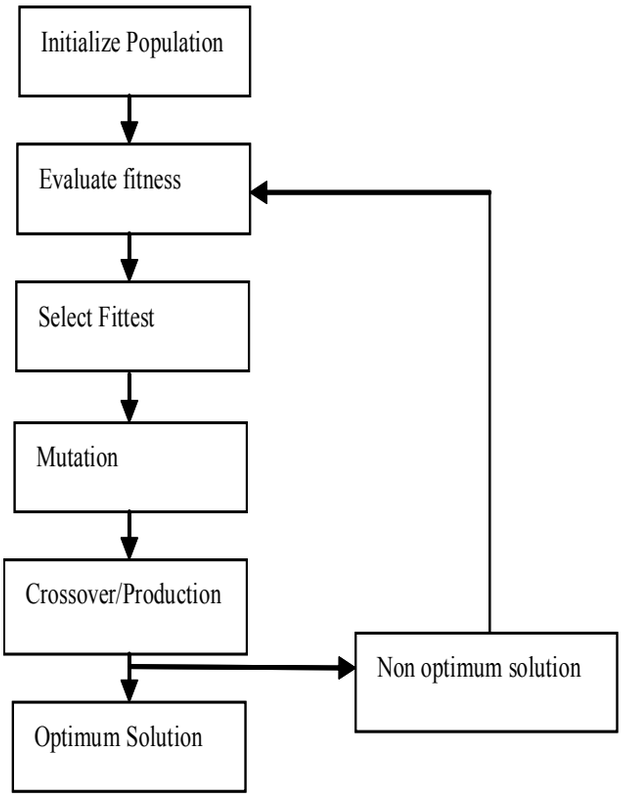

Figure 2. Flowchart of GA

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[4].

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 are not reached, search for another best chromosome.

2.3 Particle Swarm Optimization

PSO is one of the optimization techniques and a kind of evolutionary computation technique. The operation flow for PSO is shown in Figure 3. 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 is 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[3]. The velocity of each particle is adjusted accordingly to its own flying experience and the other particles flying experience[3].

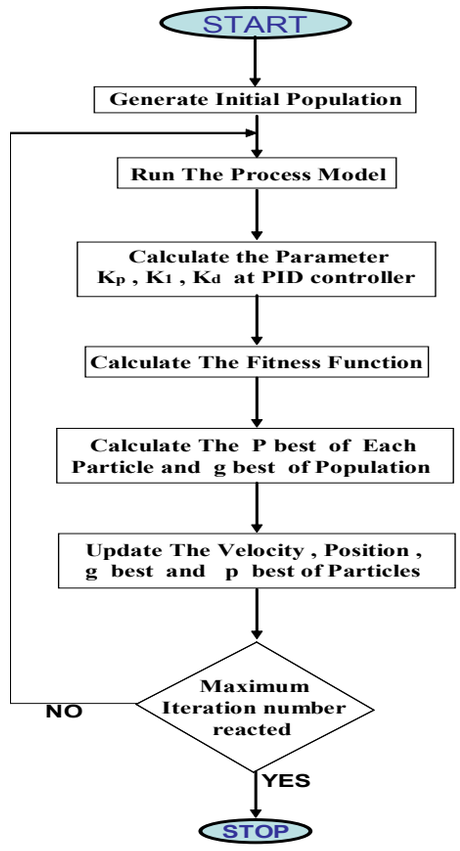

Figure 3. Flowchart of PSO.

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

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

$$Pbest_i = (Pbest_{i,1}, Pbest_{i,2}, ..., Pbest_{i,d}) \quad (2)$$

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

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

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:

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

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

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

3. Results and Discussions:

In order to cover typical kinds of common industrial processes have been taken from the below models.

Model-A[5] $$\frac{0.0147}{7.24e^{-006}s^2 + 0.000207s + 0.000437} \quad (5)$$

Model-B[2] $$\frac{1.2}{0.00077s^3 + 0.0539s^2 + 1.441s} \quad (2)$$

Model-C[6] $$\frac{32.31}{s^2 + 15.1s} \quad (6)$$

Model-D[1] $$\frac{46.21s + 206.1}{0.9372s^4 + 2.656s^3 + 75.87s^2 + 112.1s} \quad (1)$$

3.1 Implementation of Intelligent PID controller tuning

The Ziegler-Nichols tuning method using root locus and continuous cycling method was 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.

In this study, 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 PSO, EP, and GA parameters are used to verify the performance of the PID controller Parameter. The parameters of PSO, GA, and EP are tabulated in Table 1.

Performance characteristics of process models A to D were indicated and compared with the intelligent tuning methods as shown in the Figure 4-7 and values are tabulated in Table 2-5.

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

Table 1. PSO, GA, and EP parameters

PSO Parameters	GA Parameter	EP Parameters
Population size:100	Population size:100	Population size:100
Wmax=0.6	Mutation rate:0.1	Normal distribution
Wmin=0.1	Arithmetic Crossover	Mutation rate:0.01
Iteration:100	Iteration:100	Iteration:100
Fitness function:ISE	Fitness function:ISE	Fitness function:ISE

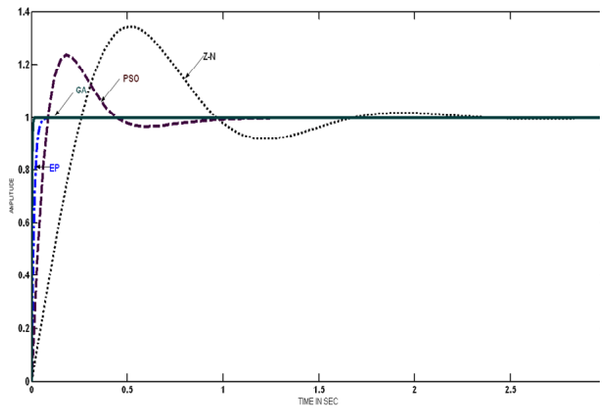

Figure 4. Comparison of all methods for model-A.

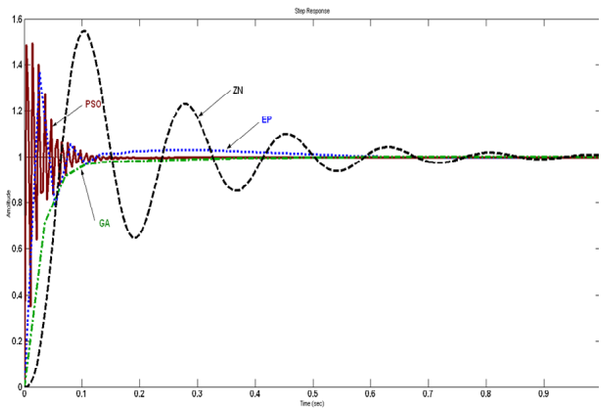

Figure 5. Comparison of all methods for model-B.

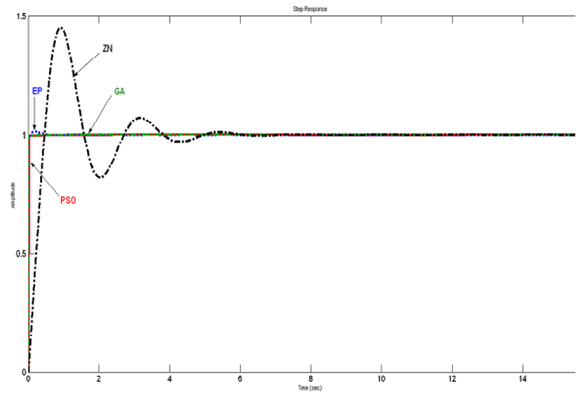

Figure 6. Comparison of all methods for model-C.

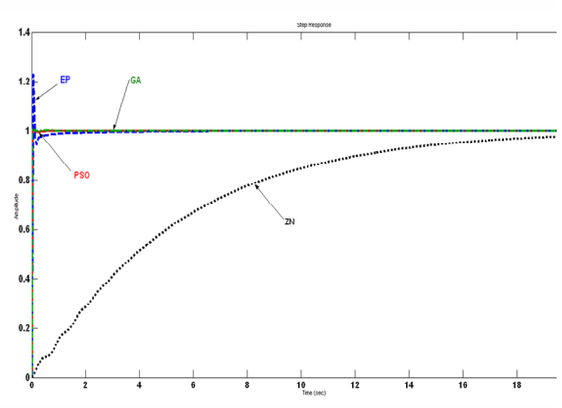

Figure 7. Comparison of all methods for model-D.

Table 2. Comparison results of all methods for model –A

Characteristics	Z-N	GA	EP	PSO
Settling time (sec)	1.57	0.0098	0.0474	0.787
Rise Time (sec)	0.2	0.0055	0.0275	0.0663
Over shoot (%)	34	0.0042	0.528	23

Table 3. Comparison results of all methods for model–B.

Characteristics	Z-N	GA	EP	PSO
Settling time (sec)	0.738	0.152	0.385	0.112
Rise Time (sec)	0.0375	0.063	0.015	0.001
Over shoot (%)	54.6	0.1	36	49.4

Table 4. Comparison results of all methods for model -C

Characteristics	Z-N	GA	EP	PSO
Settling time (sec)	4.58	0.00315	0.134	0.0301
Rise Time (sec)	0.361	0.00257	0.0196	0.0246
Over shoot (%)	45	0.0365	26.6	0.224

Table 5. Comparison results of all methods for model-D.

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

4. Conclusion

The GA, EP, and PSO algorithm for PID controller tuning presented in this research offers several advantages. One can use a-high-order process model in the tuning, and the errors resulting from model reduction are avoided. 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. Conventional methods of controller tuning lead to a large settling time, overshoot, rise time, and steady state error of the controlled system. Compared to conventionally tuned system, GA, EP, and PSO tuned system has good steady state response and performance indices.

5. References

1. Griffin I, Bruton J. On-Line Pid controller tuning using genetic algorithm. Available from www.eeng.dcu.ie/~brutonj/Reports/IGriffin_MEng_03.pdf
2. Thomas N, Poongodi P. Position control of DC motor using genetic algorithm based PID controller. Proceedings of the World Congress on Engineering. Jul 1-3, WCE, London, U.K. 2009; 2.
3. Nasri M, Nezamabadi-pour H, Maghfoori M. A PSO-based optimum design of PID controller for a linear brushless DC motor. World Academy of Science, Engineering and Technology; 2007.
4. Lieslehto J. PID controller tuning using Evolutionary programming. American Control Conference, VA Jun25-27; 2001.
5. Sharifian MBB, Rahnavard R, Delavari H. Velocity control of DC motor based intelligent methods and optimal integral state feedback controller. Internat J Comput Theor Engin. Apr 2009; 1(1).
6. Saha S. Performance comparison of PID base position control system with FLC based position control system. TIG Res J. Sep 2008; 1(2).
7. Ali khan A, Rapal N. Fuzzy pid controller: Design,Tuning and comparison with conventional PID controller. 1-4244-0457-6/06/$20 2006 IEEE.
8. Nagaraj B, Subha S, Rampriya B. Tuning algorithms for PID controller using soft computing techniques. Internat J Comput Sci Netw Secur. (IJCSNS). Apr 2008; 8(4).
9. Hwang HS, Choi JN, Lee WH. A tuning algorithm for the PID controller utilizing fuzzy theory. IEEE Proceedings 0-7803-5529-6/99/ Q1999 IEEE. pp.2210–2215.
10. Jantzen J. Tuning of fuzzy PID controllers. Denmark Tech. Report no 98- H 871(fpid), 30 Sep 1998; pp. 1-22.
11. Ang KH, Chong G. PID control system analysis, design, and technology. IEEE Transactions on Control Systems Technology. Jul 2005; 13(4):559-576.
12. Nagaraj B, Murugananth N. A comparative study of PID controller tuning using GA, EP, PSO and ACO. IEEE International Conference on Communication Control and Computing Technologies (ICCCCT); 2010