Journal of Theoretical and Applied Information Technology

www.jatit.org

OPTIMIZATION OF PD-PI CONTROLLER USING SWARM INTELLIGENCE

[1]TUSHAR JAIN, [2]M. J. NIGAM

[1]Student, Department of Electronics and Computer, IIT Roorkee, Uttarakhand, India-247667

[2]Assoc. Prof., Department of Electronics and Computer, IIT Roorkee, Uttarakhand, India-247667

E-mail: [1]tushaar.iitr@gmail.com, [2]mkndnfec@iitr.ernet.in

ABSTRACT

Sensitivity and Robustness is the primary issue while designing the controller for non-linear systems. One of the performance objectives for controller design is to keep the error between the controlled output and the set-point as small as possible. A comparison between Evolutionary Algorithms namely GAs (Genetic Algorithms), and Swarm Intelligence i.e. PSO (Particle Swarm Optimization) and BG (Bacterial Foraging) has been carried out on the basis of performance indices: ITAE (Integral Time Absolute Error), ISE (Integral Square Error), IAE (Integral Absolute Error) and MSE (Mean Square Error) and settling time. In this paper, the idea of model generation and optimization is explored for PD-PI controller. Most commonly known, the highly nonlinear Inverted Pendulum system is used as a test system for this approach. The simulations are tabulated in section IV to analyze which technique gives promising results for the system.

Keywords: *PD-PI control, Particle Swarm Optimization, Genetic Algorithms, Performance Indices, Bacterial Foraging*

1. INTRODUCTION

The interactive or series structure of PD-PI controller is shown in Fig.1:

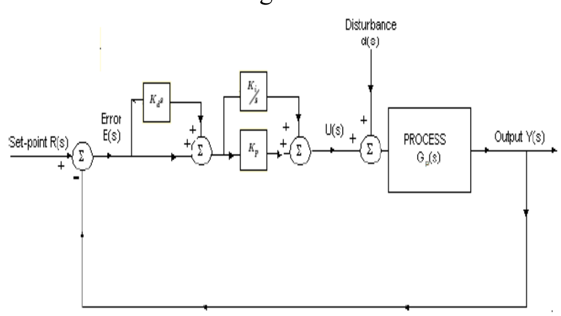

Fig.1 PD-PI control system

The general equation of PD-PI controller is:

$$U(s) = (1 + sK_d)\left(K_p + \frac{K_i}{s}\right)E(s) \qquad (1)$$

Where, K_p = Proportional Gain
K_i = Integral Gain
K_d = Derivative Gain

The step response of the PID controller is shown in Fig.2 and the effect of each component is discussed below [1]:

P (Proportional): error multiplied by the gain, K_p. This is an adjustable amplifier. In many systems K_p is responsible for "process stability": too low and the PV (process variable) can drift away; too high and the PV can oscillate.

I (Integral): the integral of error multiplied by a gain, K_i. It is responsible for driving error to zero, but to set Ki too high is to "invite oscillation" or instability or integrator windup or actuator saturation.

D (Derivative): the rate of change of error multiplied by a gain, K_d. K_d is responsible for system response: too high and the PV will oscillate; too slow and the PV will respond sluggishly. The signal derived is "infinitely high and narrow spike".

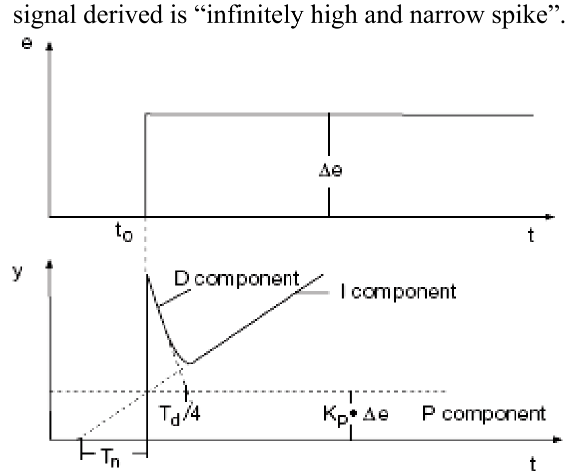

Fig.2 Step response of PID controller

2. NEED FOR TUNING

There are different types of objectives to be addressed by a controller; error can be minimized in different ways, as can CV (control variable) activity. Some systems have interactions and these may be of various strengths. Any interaction affects tuning of individual PD-PI. A need for a loop to be tuned if it responds slowly or if it oscillates too much or it has steady state error.

2.1. TECHNIQUES FOR CONTROLLER TUNING

There are several recommendations for tuning PD-PI controller parameters and for experimental determination of process characteristics to obtain PV. These procedures can be applied when mathematical model of the process is known and also when it is unknown. In any case, these recommendations can be used for initial tuning of the controller and then user can perform fine tuning using more detail knowledge of the process [2]. The classical techniques are:

Ziegler-Nichols (Z-N) Oscillation Method: Z-N tuning is used for P, PI, PID controllers. It has to be noted that controllers tuned using this procedure are tuned for control, not for tracking. Critical gain is calculated at the frequency at which system starts oscillation and the PD-PI parameters can be measured. This method fails if transfer function is not monotonous, process has astatically mode of 1^{st} or higher order and if process is unstable. Criterion used by Z-N to tune parameters is actually IAE. A shortcoming of this method is that it requires that the plant be forced to oscillate; this can be dangerous and expensive.

Reaction Based Curve Methods: With this procedure, no process model is assumed and it is based on measurements only. It can be conducted with stable and unstable processes. the procedure is valid for the system having S-shaped step response of the plant. The design objective is to achieve a particular damping in the loop response to a step reference. Cohen- Coon offers a more homogeneous response in comparison to Z-N oscillation method. These tuning methods provide the starting point for finer tuning.

Modern Techniques: Another tuning method used for optimization is (EA) Evolutionary Algorithm [3] and (SI) Swarm Intelligence. EA are search methods that take their inspiration from natural selection and survival of the fittest in the biological world and SI is originated from the study of colonies, or swarms of social organisms. The flowchart of algorithm is shown in Fig.3.

One of the strengths of EAs is that they perform well on "noisy" functions where they may be multiple local optima. EAs tend not to get "stuck" on local minima and can often find a globally optimal solution. The algorithms are discussed in next sections.

GAs is considered as a best tuning method for designing the PD-PI control, the main aim of this work is to analyze the other techniques such as PSO and BG for optimizing the parameters of PD-PI controller in comparison to GAs.

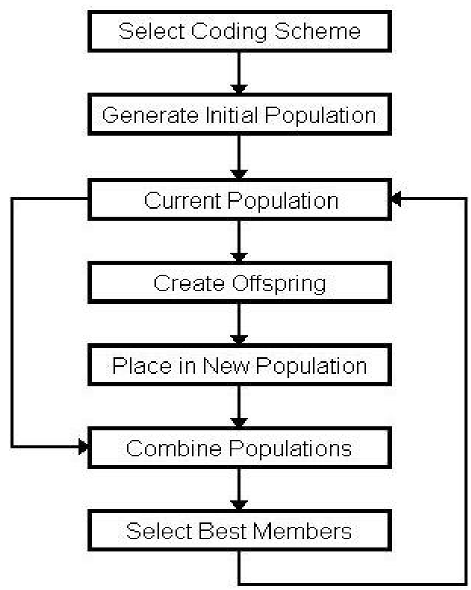

Fig.3 Evolutionary Algorithm

3. CASE STUDY

To investigate the impact of modern techniques on the design of PD-PI controller, a typical example of non-linear Inverted pendulum system is presented here in this section.

3.1. INVERTED PENDULUM

The inverted pendulum control problem [4] is usually presented as a pole balancing task. The system to be controlled consists of a cart and a rigid pole hinged to the top of the cart. The cart can move left or right on a one-dimensional bounded track, whereas the pole can swing in the vertical plane determined by the track. The linearized system equations around $\theta = \pi$ in the state space are:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u \qquad (2)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \qquad (3)$$

Where,
M = mass of cart = 0.5 kg
m = mass of pendulum = 0.2 kg
b = friction of cart = 0.1 N/m/sec
I = inertia of pendulum = 0.006 kgm^2
l = length of pendulum's center of mass = 0.3 m
F = force applied to cart

The state of the system is defined by values of four system variables: $(x, \dot{x}, \theta, \dot{\theta})$ the cart position, cart velocity, pendulum angle and angular velocity of the pendulum pole, respectively. Control force is applied to the system to prevent the pole from falling while keeping the cart within the specified limits.

3.2. GENETIC ALGORITHMS

The GAs control parameters play an important role in the procedure for optimizing the parameters of the PD-PI controller. Some worthwhile discussions of the GAs parameters are made as follows [5]:

- *Encoding form:* The linear encoding form is used. The length of binary coding string for each variable is important for the GAs. There is always a compromise between complexity and accuracy in the choice of string length. Here, a 16-bit binary coding is used for each parameter.
- *Crossover and mutation rates:* Crossover and mutation rates are not fixed during evolution period. At the beginning, crossover and mutation rates are, respectively, fixed to 0.9 and 0.1, then decrease 10 percent in each generation until crossover rate is 0.5 and mutation rate is 0.01.
- *Population size:* The population size has to be an even number and is kept fixed throughout. Generally, the bigger the population size, the more design features are included. The population size should neither be too small nor too big. The procedure of optimizing will be slow when the population size is big.

3.3. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is a population based stochastic optimization technique inspired by social behavior of bird flocking or fish schooling [6]. PSO learns from the scenario and uses it to solve the optimization problems. In PSO, each single solution is a "bird" in the search space. We call it "particle". All particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles.

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In each iteration, every particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest.

After finding the two best values, the particle updates its velocity and positions with following equation (4) and (5).

v [] = v [] + c1 * rand () * (pbest [] - present []) + c2 * rand () * (gbest [] - present []) (4)

present [] = present [] + v [] (5)

v[] is the particle velocity, present[] is the current particle (solution). pbest[] and gbest[] are defined as stated before. rand () is a random number between (0, 1). c1, c2 are learning factors usually c1 = c2 = 2.

3.4. BACTERIAL FORAGING

This paper considers the foraging behavior of E. coli, which is a common type of bacteria [7]. Its behavior to move comes from a set of up to six rigid 100-200 rps spinning flagella, each driven as a biological motor. An E. coli bacterium alternates between running and tumbling. Running speed is 10-20 μm/sec, but they cannot swim straight. The main goal based on bacterial foraging is to apply in order to find the minimum of $J(\phi), \phi \in R^n$, not in the gradient $\nabla J(\phi)$. here, when ϕ is the position of a bacterium, and $J(\phi)$ is an attractant-repellant profile. That is, it means where nutrients and noxious substances are located, so $J(\phi) < 0$, $J(\phi) = 0$, $J(\phi) > 0$ represent the presence of nutrients, a neutral medium and the presence of noxious substances, respectively can be shown by

$$P(j,k,l) = \{\phi^i(j,k,l) \mid i = 1,2,...., N\} \qquad (6)$$

Equation (6) represents the positions of each member in the population of the N bacteria at the jth chemotactic step, kth reproduction step, and lth elimination-dispersal event.

Basically, chemotaxis is a foraging behavior that implements a type of optimization where bacteria tries to climb up the nutrient concentration (find lower and lower values of $J(\phi)$), avoid noxious substances, and search for ways out of neutral media. It implements a biased random walk. This paper describes the method in the form of an algorithm to search optimal value of PD-PI parameter.

4. RESULTS AND DISCUSSION

The open loop step response of pendulum angle is shown in Fig.4.

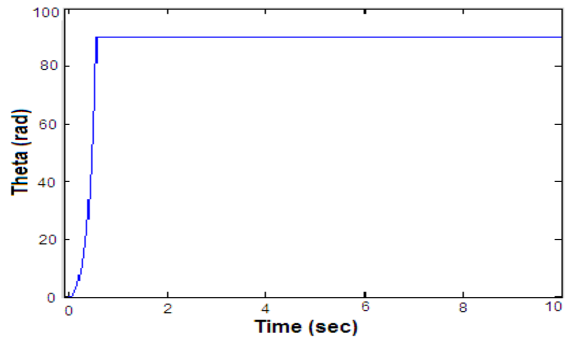

Fig.4 Step Response of pendulum angle

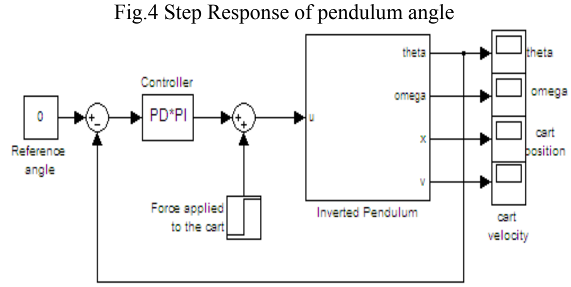

Fig. 5 Block diagram of PD-PI control system

The performance of the PD-PI control can be judged by the values of its parameters. The algorithms as explained have been used to estimate the parameters. Empirical tuning methods cannot be applied to the system under consideration as the conditions (i.e. open loop stability, S-shaped response etc.) for applying them are not satisfied.

The performance with GAs, PSO and BG is analyzed on the basis of ITAE, ISE, MSE, IAE and the settling time of the pendulum to maintain its initial position i.e. 0^o for the force applied to the cart. For settling time, a band of -0.0005 rad to +0.0005 rad are considered around 0 rad. The simulated results of PD-PI control using GAs, PSO and BG for different types of errors are shown in Fig.6, Fig.7 and Fig.8 respectively.

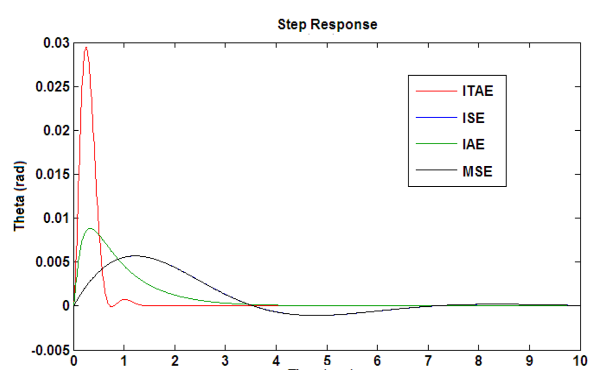

Fig.6 Step responses (ITAE, ISE, IAE, MSE) using Gas

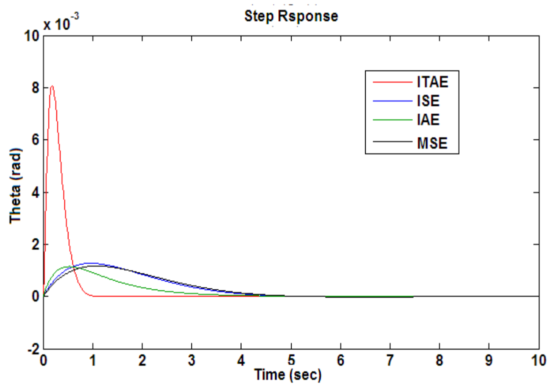

Fig.7 Step responses (ITAE, ISE, IAE, MSE) using PSO

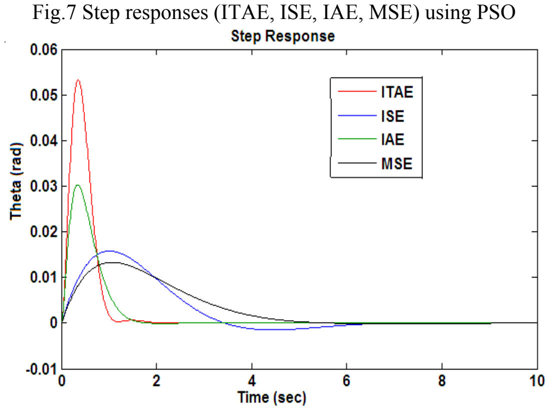

Fig.8 Step Response (ITAE, ISE, IAE, MSE) using BG

Table.1 Comparison of Performance Indices

Performance Indices	ITAE	ISE	IAE	MSE
Using Genetic Algorithms	0.045 0	0.00083	0.211 1	2.1e-06
Using Particle Swarm Optimizatio n	0.004 2	0.00002 3	0.015 3	8.24e-08
Using Bacterial Foraging	0.370 3	0.0039	0.232 4	1.469e -5

Table.2 Comparison of steady state response (ITAE)

Characteristi cs	GAs	PSO	BG
Settling time (sec)	0.9449	0.5652	1.76
Peak angle (rad)	0.02949	0.00808 3	0.0608

Table.3 Comparison of steady state response (ISE)

Characteristic s	GAs	PSO	BG
Settling time (sec)	3.36	2.4	5.93
Peak angle (rad)	0.00569	0.00127	0.0158

Table.4 Comparison of steady state response (IAE)

Characteristic s	GAs	PSO	BG
Settling time (sec)	2.69	1.15	1.53
Peak angle (rad)	0.00882 4	0.00114 6	0.0302

Table.5 Comparison of steady state response (MSE)

Characteristic s	GAs	PSO	BG
Settling time (sec)	3.33	1.97	4.78
Peak angle (rad)	0.00569 9	0.00116 8	0.0133

5. CONCLUSIONS

For the system under consideration, the simulation results with PSO techniques prove to be more effective than with GAs and BG. In GAs, the limits defined by the number of parameters gives the search region while in PSO, the search region is independent of the number of parameters, given by the distance between the randomly selected initial position and the position corresponding to optimal fitness value. The speed of computation is determined by the velocity initializing the PSO algorithm with which it reaches to the best solution. It is also observed that the speed of computation in PSO is very less in comparison to GAs and BG.

A disadvantage of PD-PI tuning using PSO is that the values of K_p, K_i, and K_d are quite high in comparison to the values determined using GA and BG.

Empirical tuning method discussed in section II cannot be applied in this case as the system considered is not open loop stable. From the step response, it can be seen that BG gives good results in terms of settling time in comparison to GAs.

PSO algorithms combined with other intelligent techniques, such as neural networks, expert systems, and fuzzy logic control systems open a new way to design and construct intelligence control systems adapted to complex processes.

REFRENCES

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

[2] Graham C. Goodwin, Stefan F. Graebe, Mario E. Salgado: "Control System Design", 2nd edition, January 2000.

[3] Greffenstette, J. J. "Optimization of Control Parameters for Genetic Algorithms," *IEEE Transactions on Systems, Man, and Cybernetics SMC-16*, **1**, 1986, 122-128.

[4] Dadios, Elmer P.; Williams, David J., "A Fuzzy-Genetic Controller for the Flexible Pole-Cart Balancing Problem", Proceedings of IEEE International Conference on Evolutionary Computation, pp 223, ISBN: 0 7803 2902 3, Nagoya, Japan, 1996.

[5] Goldberg, D. E.: "Genetic algorithms in search, optimization and machine learning" (Addison-Wesley, 1989).

[6] James Kennedy, Russell Eberhart: "Particle Swarm Optimization", IEEE Transactions, pp. 1942-1948, 1995.

[7] Kevin M. Passino, "Biomimicry of Bacterial Foraging for Distributed Optimization and Control", IEEE Control Systems Magazine, June 2002, p.p. 52-67

Tushar Jain is an M.Tech student with specialization in Control and Guidance, Department of Electronics and Computer at Indian Institute of Technology Roorkee. He received his B.Tech degree from Babu Banarsidas Institute of Technology, Ghaziabad in 2007. His research areas of interest are Fuzzy Control, Intelligent Control and Monitoring Systems and Robotics.

M. J. Nigam is Associate Professor in Control & Guidance group, Department of Electronics and Computer at Indian Institute of Technology Roorkee. He received his B.Tech degree from Regional Engineering College (now National Institute of Technology), Warangal in 1976 and M.E. and Ph.D. from University of Roorkee (now Indian Institute of Technology Roorkee) with specialization in Control and Guidance in 1978 and 1992 respectively. His research areas of interest are Digital Image Processing, Robotics, and Filtering Techniques. A number of research papers in the above areas have also been published/ presented in various journals and Conferences.