ABSTRACT

The Paper Present to design a PID controller by selection of a PID parameters using Bacteria Foraging based Particle Swarm Optimization algorithm. The model of a DC motor is used as a plant. The conventional gain tuning of PID controller usually produces a big overshoot, and therefore modern heuristics approach such as Bacteria Foraging based Particle Swarm Optimization algorithm are employed to enhance the capability of traditional techniques. The methodology and efficiency of the proposed method are compared with the Bacteria Foraging based Particle Swarm Optimization. In any classical PID control problem, the 'required controller parameters should be optimally designed. These parameters can be optimally obtained via Bacteria Foraging based Particle Swarm Optimization. Furthermore, the process of coding and decoding not only impacts precision, but also increases the complexity of the Bacteria Foraging based Particle Swarm Optimization is a novel emerging intelligence which was flexible optimization. However, due to the computational efficiency, the BF-PSO will be used in this paper. The comparison between classical PID control problem and the Bacteria Foraging based particle swarm optimization is presented. By these technique produces minimum rise time and overshoot. The results show the advantage of the PID tuning using these optimization approaches. The output of the conventional PID system has a quite high overshoot and settling time. The main aim of this paper is to apply Bacteria Foraging based Particle Swarm Optimization to design and tuning of PID controller to get an output with better dynamic and static performance.

Keywords- PID tuning, MATLAB, Simulation, Bacteria Foraging based particle swarm optimization.

1. 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. In a PID controller, each mode (proportional, integral and derivative mode) has a gain to be tuned, giving as a result three variables involved in the tuning process[1]. There have been a lot of approaches to search the parameters of PID controllers, including time response tuning, time domain optimization, frequency domain shaping and genetic algorithms. The speed response of the drive with PID controllers designed with the above techniques may be satisfactory but not necessarily be the best, since they do not pose any constraint on settling time, overshoot undershoot etc[2]. In any classical PID control problem, the ' required controller parameters should be optimally designed. These parameters can be optimally obtained by Bacteria Foraging based Particle Swarm optimization[5]. Particle swarm optimization (PSO) is a novel emerging intelligence which was flexible optimization algorithm proposed in 1995. There are many common characteristics Bacteria Foraging based PSO. In this paper, an optimal PID controller for DC motor drive systems is developed using Bacteria Foraging based Particle Swarm optimization Technique[12]. The performance measure to be minimized contains the following objectives of the PID controller, that is Minimize the steady state error, Minimize the rise time, Minimize the maximum overshoot and Minimize the settling time.

2. PROBLEM FORMULATION

PID controller consists of Proportional, Integral and derivative gains. The feedback control system is illustrated in Fig.2.1 where \( r, e, y \) are respectively the reference, error and controlled variables.

![Fig 2.1: Unity Feedback control system](image)

In the diagram of Fig.2.1, \( G(s) \) is the plant transfer function and \( C(s) \) is the PID controller transfer function that is given as:

\[
C(s)=K_p+\frac{K_i}{s}+K_d s
\]  

(2.1)
Where \( K_p \), \( K_i \), \( K_d \) are respectively the proportional, integral, derivative gains/parameters of the PID controllers that are going to be tuned. The plant used here is a DC motor model which is a third order system written as:

\[
G(s) = \frac{Y(s)}{U(s)} = \frac{1}{s^3 + 9s^2 + 23s + 15} \quad (2.2)
\]

Furthermore, performance index is defined as a quantitative measure to depict the system performance of the designed PID controller. Using this technique an ‘optimum system’ can often be designed and a set of PID parameters in the system can be adjusted to meet the required specification. A set of good control parameters \( k_p \), \( k_i \), and \( k_d \) can yield a good step response that will result in performance criteria minimization in the time domain [1]. These performance criteria in the time domain include the overshoot \( M_p \), rise time \( t_r \), settling time \( t_s \), and steady-state error \( E_s \). Therefore, a new performance criterion \( W(K) \) is defined as follows:

\[
W(K) = (1 - e^{-\beta}) \cdot (M_p + E_s) + e^{-\beta} \cdot (t_s - t_r) \quad (2.3)
\]

where \( k \) is \([k_p, k_i, k_d]\), and \( \beta \) is the weighting factor. The performance criterion \( W(K) \) can satisfy the designer requirements using the weighting factor \( \beta \) value. We can set \( \beta \) to be larger than 0.7 to reduce the overshoot and steady-state error. On the other hand, we can set to be smaller than 0.7 to reduce the rise time and settling time. In this project, is set in the range of 0.8 to 1.5.

### 3. 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 set point. The PID controller calculation involves three separate parameters proportional integral and derivative values[2-5]. 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.

### 4. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is an algorithm modeled on swarm intelligence, that finds a solution to an optimization problem in a search space, or model and predict social behavior in the presence of objectives. The PSO is a stochastic, population-based computer algorithm modeled on swarm intelligence. Swarm intelligence is based on social-psychological principles and provides insights into social behavior, as well as contributing to engineering applications.

PSO is optimization algorithm based on evolutionary computation technique. The basic PSO is developed from research on swarm such as fish schooling and bird flocking. After it was firstly introduced in 1995, a modified PSO was then introduced in 1998 to improve the performance of the original PSO. A new parameter called inertia weight is added. This is a commonly used PSO where inertia weight is linearly decreasing during iteration in addition to another common type of PSO which is reported by Clerc. The later is the one used in this project. In PSO, instead of using genetic operators, individuals called as particles are “evolved” by cooperation and competition among themselves through generations. A particle represents a potential solution to a problem. Each particle adjusts its flying according to its own flying experience and its companion flying experience. Each particle is treated as a point in a D-dimensional space. The th particle is represented as \( Xi=(x1,x2,\ldots,xD) \). The best previous position (giving the minimum fitness value) of any particle is recorded and represented as \( Pf=(p1,p2,\ldots,pD) \), this is called pbest. The index of the best particle among all particles in the population is represented by the symbol \( g \), called as gbest. The velocity for the particle \( i \) is represented as \( Vi=(v1,v2,\ldots,vD) \). The particles are updated according to the following equations:

\[
V_{i,m}^{(t+1)} = w \cdot V_{i,m}^{(t)} + c_1 \cdot rand() \cdot (P_{best,i}^{(t)} - X_{i,m}^{(t)}) + c_2 \cdot rand() \cdot (G_{best}^{(t)} - X_{i,m}^{(t)}) \quad (4.1)
\]

\[
X_{i,m}^{(t+1)} = X_{i,m}^{(t)} + V_{i,m}^{(t+1)} \quad (4.2)
\]

where \( c_1 \) and \( c_2 \) are two positive constant. As recommended in Clerc’s PSO, the constants are \( c_1=c_2=1.5 \). While \( rand() \) is random function between 0 and 1, and \( n \) represents iteration. Eq.(4.1) is used to calculate particle’s new velocity according to its previous velocity and the distances of its current position from its own best experience (position) and the group’s best experience. Then the particle flies
toward a new position according to Eq.(4.2). The performance of each particle is measured according to a pre-defined fitness function (performance index), which is related to the problem to be solved. Inertia weight, w is brought into the equation to balance between the global search and local search capability. It can be a positive constant or even positive linear or nonlinear function of time. A guaranteed convergence of PSO proposed by Clerc set \( w = 0.5 \). It has been also shown that PSO with different number of particles (swarm size) has reasonably similar performance. Swarm size of 10-50 is usually selected. Here, we set 50.

Stochastic Algorithm can be applied to the tuning of PID controller gains to ensure optimal control performance at nominal operating conditions. PSO is employed to tune PID gains/parameters (\( K_p, K_i, K_d \)) in offline using the model in above Eqn. PSO firstly produces initial swarm of particles in search space represented by matrix. Each particle represents a candidate solution for PID parameters where their values are set in the range of 0 to 100. For this 3-dimensional problem, position and velocity are represented by matrices with dimension of 3xSwarm size. The swarm size is the number of particle where 40 are considered a lot enough. A good set of PID controller parameters can yield a good system response and result in minimization of performance index.

### 5. BACTERIAL FORAGING OPTIMIZATION

Introduction Based on the research of foraging behaviour of E.colli bacteria Kevin M.Passino and Liu exploited a variety of bacterial foraging and swarming behaviour, discussing how to connect social foraging process with distributed non-gradient optimization. In the bacterial foraging optimization process four motile behaviours are mimicked: -

**CHEMOTAXIS:**
This process is achieved through swimming and tumbling. Depending upon the rotation of the flagella in each bacterium, it decides whether it should move in a predefined direction (swimming) or an altogether different direction (tumbling), in the entire lifetime of the bacterium. To represent a tumble, a unit length random direction, \( \theta(j) \) is generated; this will be used to define the direction of movement after a tumble. In particular, \( \theta(j+1, k, l) = \theta(j, k, l) + C(i) \ast \theta(j) \)  

\[ (5.1) \]

where \( \theta(j, k, l) \) represents the ith bacterium at jth chemotactic kth reproductive and lth elimination and dispersal step. \( C(i) \) is the size of the step taken in the random direction specified by the tumble. “C” is termed as the “run length unit”.

**SWARMING:**
It is always desired that the bacterium that has searched the optimum path of food should try to attract other bacteria so that they reach the desired place more rapidly. Swarming makes the bacteria congregate into groups and hence move as concentric patterns of groups with high bacterial density. Mathematically, swarming can be represented by

\[ I_m(\theta(j, k, l)) = \sum_{i=1}^{n} [-d_{attractive} \exp(-W_{attractive} \sum_{m=1}^{p} (\theta_m - \theta_i)^2)] \]

\[ + \sum_{i=1}^{n} [-d_{repellent} \exp(-W_{repellent} \sum_{m=1}^{p} (\theta_m - \theta_i)^2)] \]

\[ (5.2) \]

Where is the cost function value to be added to the actual cost function to be minimized to present a time varying cost function. “S” is the total number of bacteria. “p” is the number of parameters to be optimized that are present in each bacterium. Attractant \( d \), attractant \( w \), repellent \( h \), and repellent \( w \) are different coefficients that are to be chosen judiciously.

**REPRODUCTION:**
The least healthy bacteria die, and the other healthiest bacteria each split into two bacteria, which are placed in the same location. This makes the population of bacteria constant.

**ELIMINATION AND DISPERSAL:**
It is possible that in the local environment, the life of a population of bacteria changes either gradually by consumption of nutrients or suddenly due to some other influence. Events can kill or disperse all the bacteria in a region. They have the effect of possibly destroying the chemotactic progress, but in contrast, they also assist it, since dispersal may place bacteria near good food sources. Elimination and dispersal helps in reducing the behavior of stagnation (i.e., being trapped in a premature solution point or local optima).

### 6. BACTERIA FORAGING BASED PARTICLE SWARM OPTIMIZATION (BF-PSO)

BF-PSO algorithm combines both BFO and PSO. The aim is to make PSO ability to exchange social information and BF ability in finding new solution by elimination and dispersal, a unit length direction
of tumble behavior is randomly generated. Random direction may lead to delay in reaching the global solution. In "BF-PSO" algorithm the unit length random direction of tumble behavior can be decided by the global best position and the best position of each bacterium. During the chemotaxis loop tumble direction is updated by:

$$\phi_j = \phi_{pbest} - \phi_{pcurrent} + C2 * \text{rand} * (gbest - \phi_{pcurrent})$$

Where pbest is the best position of each bacterium and gbest is the global best bacterium.

Fig 6.1 Flow chart of BF-PSO algorithm.

7. SIMULATION RESULTS

The transfer function of the DC Motor is given below:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{s^3 + 9s^2 + 23s + 15}$$

In the classical PID controller, the plant response produces high overshoot, but a better performance obtained with the implementation of BF-PSO-based PID controller tuning.

Table 7.1 The Comparative Results of these Techniques.

<table>
<thead>
<tr>
<th>Parameter and Gain</th>
<th>BFO</th>
<th>PSO</th>
<th>Bacteria Foraging based PSO (BF-PSO)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_p$</td>
<td>33.49</td>
<td>33.0360</td>
<td>44.5033</td>
</tr>
<tr>
<td>$K_i$</td>
<td>24.50</td>
<td>24.7045</td>
<td>29.0378</td>
</tr>
<tr>
<td>$K_d$</td>
<td>11.24</td>
<td>8.8183</td>
<td>11.0689</td>
</tr>
<tr>
<td>Rise Time</td>
<td>1.1147</td>
<td>0.9878</td>
<td>0.7207</td>
</tr>
<tr>
<td>Settling Time</td>
<td>2.1334</td>
<td>1.6357</td>
<td>1.1170</td>
</tr>
<tr>
<td>Overshoot</td>
<td>0.4059</td>
<td>0.2000</td>
<td>0</td>
</tr>
</tbody>
</table>

Comparative results for the All Technique are given in Table (7.1), where the step response performance is evaluated based on the overshoot, settling time and Rise time. The step responses of BFO-PID controller are shown in Fig. 7.1.

Fig.7.1: Step response for BFO-PID controller

The step responses of PSO-PID controller are shown in Fig. 7.2. it also have a good Response for the system.

Fig.7.2: Step response for PSO-PID controller

And The step response of BF-PSO-PID controller have less overshoot are shown in Fig. 7.3. and Bacteria Foraging based PSO are the recent and efficient optimization technique.

Fig.7.3: Step response for BF-PSO-PID controllers
8. CONCLUSIONS

From the results, the designed PID controllers using Bacteria Foraging based PSO based optimization have less overshoot compared to that of the classical method. However, the classical method is good for giving us as the starting point of what are the PID values. Therefore the benefit of using a modern optimization approach is observed as a complement solution to improve the performance of the PID controller designed by conventional method. Of course there are many techniques can be used as the optimization tools and Bacteria Foraging based PSO are the recent and efficient optimization techniques.

Fig.7.4: The comparative Step responses for BFO, PSO, BF-PSO controller

9. REFERENCES

[5] Adel A. El-Gammal1 Adel A. El-Samahy “A Modified Design of PID Controller For DC Motor Drives Using Particle Swarm Optimization,” Ienergy Research Centre, University of Trinidad and Tobago UTT (Trinidad and Tobago), Lisbon, Portugal, March 18-20, 2009
[7] Wen-wen Cai, Li-xin Jia, Yan-bin Zhang, Nan Ni “Design and simulation of intelligent PID controller based on particle swarm optimization” School of Electrical Engineering Xi'an Jiao Tong University Xi'an, Shaanxi, 710049, P. R. China Caiwenwen0533@yahoo.com.cn