PSO and PSO-BFO Based Tuning of PID Controller: A Comparative Evaluation

Sapna Gupta¹, Rajeev Gupta²

Department of Electronics Engineering, University College of Engineering, Rajasthan Technical University, Rajasthan, India

Abstract—The aim of this paper is to study the tuning of a PID controller using swarm optimization techniques. In this paper, comparative performance of PSO and BF-PSO based PID controller is analyzed. PSO algorithm converges rapidly during the initial stages of a global search, but around global optimum, the search process slows down. In order to overcome this problem and to further enhance the performance of PSO, this paper implements a hybrid algorithm (BF-PSO) combining the features of Bacterial Foraging Optimization (BFO) and Particle Swarm Optimization (PSO) for tuning of PID controller. The simulation results show that the hybrid (BF-PSO) algorithm has less settling time and less overshoot than PSO algorithm. BF-PSO algorithm is better than Z-N tuning method and PSO in terms of set point tracking, simplicity, consistency, search and computational efficiency.

Keywords—BFO, BF-PSO, PID, PSO

I. INTRODUCTION

PID control is one of the most efficient and widely used feedback control strategy. This is due to its simplicity and satisfactory control performance. PID controller was introduced in 1910 and is widely used in oil refineries, chemical plants, and paper and pulp industry. Its use and popularity had grown particularly after the Ziegler-Nichols empirical tuning rules in 1942.

The optimally combined three terms functioning of PID controller can provide treatment for both transient and steady state responses. In fact, optimal control performance can only be achieved after identifying the finest set of three gains, that is, proportional gain (Kₚ), integral gain (Kᵢ) and derivative gain (Kₛ). Proportional gain reduces the overshoot, integral gain reduces the steady state error and derivative gain makes the controller act faster. Many approaches have been reported in literature for tuning parameters of PID controller [1-2]. The conventional PID tuning techniques include Z-N, Cohen Coon, and relay feedback methods. The modern techniques are based on artificial intelligence techniques such as neural network, fuzzy logic and evolutionary computation.

In Ziegler-Nichols tuning method, ultimate gain and time period is calculated and from the above information, PID parameters are calculated.

But the calculated gains of PID controller may or mayn’t be optimal for practical purposes. Therefore, researchers have used evolutionary algorithms [3] and swarm optimization based techniques [4-6] for PID tuning. Out of many swarm optimization algorithms hybrid particle swarm optimization is one of the emerging algorithms because of its faster convergence and optimal results. This paper presents a comparative performance analysis of PSO and BF-PSO algorithms for optimal PID tuning of shell and tube heat exchanger system. The simulation results show that the hybrid (BF-PSO) algorithm is performs better than particle swarm optimization algorithm (PSO) algorithm in terms of time and frequency domain specifications.

II. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is one of the stochastic optimization technique based on the movement and intelligence of swarms. PSO applies the concept of social interaction to problem solving. It uses a number of agents (particles) that constitute a swarm moving around in the search space looking for the best solution. Each particle is treated as a point in N-dimensional space which adjusts its “flying” according to its own flying experience as well as the flying experience of other particles. The position corresponding to best fitness is called pbest and overall best out of all the particles is called gbest. Initial position of pbest and gbest are different. However using different directions of pbest and gbest all agents are gradually get close to the global optimum [7-8].

The modified velocity and position of each particle can be calculated using current velocity and the distance between pbestₘ,ₙ and gbestₗ as per the following equations

\[ v_{j,g}^{(i+1)} = w v_{j,g}^{(i)} + c_1 r_1 ( (pbest_{j,g} - x_{j,g}^{(i)}) + c_2 r_2 ( (gbest_{g} - x_{j,g}^{(i)})) \]

\[ x_{j,g}^{(i+1)} = x_{j,g}^{(i)} + v_{j,g}^{(i+1)} \]

\[ j = 1, 2, ..., n \] and \[ g = 1, 2, ..., m \]

748
The general flow chart of PSO is shown in figure 1.

In principle, bacteria try to reach the nutrients and to avoid noxious materials and to find a way to exit the neutral and noxious nutrient environment. Therefore, understanding their behavior model helps engineers to use it for any nonlinear optimization problem. The foraging process of a bacterium, E.coli, living in the intestine, is modeled through four steps of chemotactic, swarming, reproduction, and elimination and dispersal. The BF algorithm is dependent on random direction which slows down the optimal solution process. Therefore, particle swarm optimization (PSO) algorithm is used to solve this drawback. So BF-PSO algorithm is used.
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. Where \( p_{best} \) is the best position of each bacterium and \( g_{best} \) is the global best bacterium.

During chemotaxis loop, tumble direction is updated by

\[
\phi(j + 1) = w\phi(j) + c_1(p_{best} - p_{current}) + c_2\text{rand}(g_{best} - p_{current})
\]

A. Pseudo Code for BF-PSO [9]

1. Initialization
   a. \( P \) dimension of search space
   b. \( S \) total number of bacteria in population
   c. \( N_c \) number of chemotactic steps
   d. \( N_s \) swimming length \( N_c > N_s \)
   e. \( N_{re} \) number of reproduction steps
   f. \( N_{ed} \) number of elimination-dispersal event
   g. \( P_{ed} \) probability of elimination-dispersal
   h. \( \theta^i \) is location of \( i^{th} \) bacterium
   i. \( C(i) \) size of steps taken by random direction
   j. \( c_1, c_2, w, \phi \) PSO parameters

2. Elimination-dispersal loop \( l = l + 1 \)

3. Reproduction loop \( k = k + 1 \)

4. Chemotaxis loop \( j = j + 1 \)

   for \( i = 1, 2 \ldots S \) (take a chemomatic step for bacterium as follows)
   a. Tumble using
      \[
      \phi(j + 1) = w\phi(j) + c_1(p_{best} - p_{current}) + c_2\text{rand}(g_{best} - p_{current})
      \]
   b. Computer Objective function (Obj)
   c. Swim
      i. \( m = 0 \) for swim counter
      ii. While \( m < N_s \)

1. Let \( m = m + 1 \)

2. If \( \text{Obj}(i, j + 1, k, l) < \text{Obj}_{last} \)
   Update \( S_i \)
   end if loop

end while loop

Go to “a” for next bacterium \((i+1)\) if \( i \neq S \)

5. If \( j < N_c \), go to step 4

6. Reproduction

7. If \( k < N_{re} \), go to step 3

8. Elimination-Dispersal

9. End

B. BF-PSO based PID Tuning

In BF-PSO tuning of PID controller, optimal value of \( [k_p, k_i, k_d] \) are obtained which exhibits less overshoot, has a moderate level of settling time, low rise time and zero steady state error. The objective function of BF-PSO consists of these terms and minimizes the objective function.

Objective function of BF-PSO considered in this paper is

\[
\min_{k_p, k_i, k_d} W(K) = (1 - e^{-\beta})(M_p + e_{ss}) + e^{-\beta}(t_s - t_r)
\]

Subject to,

\[
0 \leq K_p \leq K_{p_{(\max)}}
0 \leq K_i \leq K_{i_{(\max)}}
0 \leq K_d \leq K_{d_{(\max)}}
\]

\( \beta \) is weighing factor, \( M_p \) is peak overshoot, \( e_{ss} \) is steady state error, \( t_s \) is settling time and \( t_r \) is rise time.

\( K_{p_{(\max)}}, K_{i_{(\max)}}, K_{d_{(\max)}} \) are maximum values of \( K_p, K_i, K_d \) respectively
V. FEEDBACK CONTROL OF HEAT EXCHANGER SYSTEM

Heat exchanger is one of the most widely used equipment in chemical industry which performs mass transfer and heat transfer operations. The primary objective of the heat exchanger system is to maintain the outlet temperature to a desired temperature. To achieve the above mentioned control objective, feedback control is used. In feedback control, the steam flow is termed as manipulated variable, whereas input flow change can be termed as flow disturbances.

To develop the experimental model, the experimental data is considered which is reported in [10,11]. From the experimental data, following model is developed.

Transfer function of process is considered as

\[
\frac{G_p(s)}{T(s)} = \frac{5e^{-s}}{90s^2 + 33s + 1}
\]

Transfer function of inlet flow disturbance is considered as,

\[
\frac{G_d(s)}{D(s)} = \frac{1}{30s + 1}
\]

Transfer function of sensor is considered as

\[
\frac{G_s(s)}{B(s)} = \frac{0.16}{10s + 1}
\]

The complete transfer function can be represented as

\[
Y(s) = \frac{G_i(s)G_p(s)}{1 + G_i(s)G_p(s)G_s(s)}R(s) + \frac{G_d(s)}{1 - G_i(s)G_p(s)}D(s)
\]

The stability of the mathematical model is evaluated using bode plot which is shown in figure 5. From the calculated gain and phase margin it is found out that the system is a stable system.
VI. SIMULATION RESULTS

A controller has two main objectives, i.e. set point tracking and disturbance rejection. Unit step function is provided to the system to test the above mentioned objectives. Simulation results in figure 6 shows the set point tracking feature of the PID controller. In set point tracking objective, the PID controller exhibits a peak overshoot of 17%, settling time of 84.9 sec and zero steady state error.

![Unit step response of PID controller](image)

**Fig. 6.** Unit step response of system using PID controller tuned using Z-N Method exhibiting set point tracking feature

To test the load disturbance objective of PID controller, two kinds of loads are used i.e step disturbance and pulse disturbance. The unit step response of PID controller compensating the step load disturbance at 250 sec is shown in figure 7. In this case, the PID controller has slightly high peak overshoot of 29.56%. The unit step response of PID controller compensating the pulse load disturbance of 50% duty cycle originating from 250 sec is shown in figure 8. In this case, the PID controller has slightly high peak overshoot of 34.72%.

![Unit step response of PID controller](image)

**Fig. 7.** Unit step response of system using PID controller tuned using Z-N Method compensating for step disturbance

![Unit step response of PID controller](image)

**Fig. 8.** Unit step response of system using PID controller tuned using Z-N Method compensating for pulse disturbance

From the above results it is observed when PID controller does the set point tracking operation, the peak overshoot is small and when it is used for load disturbance rejection, the peak overshoot increases. Table I summarizes the results of PID controller transient performance.
To qualitatively analyze controller performance, different error indices are used. Widely used error indices are

\[ IAE = \int_{0}^{\infty} |e(t)| \, dt, \quad ITAE = \int_{0}^{\infty} t \cdot |e(t)| \, dt, \]

\[ ISE = \int_{0}^{\infty} e^2(t) \, dt, \quad ITSE = \int_{0}^{\infty} te^2(t) \, dt \]

Table II indicates error indices of set point tracking, step disturbance and pulse disturbance rejection.

<table>
<thead>
<tr>
<th>Controller</th>
<th>IAE</th>
<th>ISE</th>
<th>ITAE</th>
<th>ITSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID (Pulse disturbance rejection)</td>
<td>7.67</td>
<td>0.42</td>
<td>1346</td>
<td>36.09</td>
</tr>
<tr>
<td>PID (Step disturbance rejection)</td>
<td>5.89</td>
<td>0.31</td>
<td>709.4</td>
<td>18.4</td>
</tr>
<tr>
<td>PID (Set point tracking)</td>
<td>3.93</td>
<td>0.31</td>
<td>131.8</td>
<td>3.98</td>
</tr>
</tbody>
</table>

Due to high overshoot of PID controller tuned using Z-N method, PSO and BF-PSO is used. Tuning of PID using PSO and BF-PSO is described in section IV. The set point tracking feature of PSO tuned PID controller and BF-PSO tuned PID controller is shown in figure 9 and figure 10 respectively. The transient behavior of Z-N tuned PID, PSO and BF-PSO is tabulated in Table III.

<table>
<thead>
<tr>
<th>Controller</th>
<th>% Overshoot</th>
<th>Settling Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID (Z-N)</td>
<td>17%</td>
<td>86.04 sec</td>
</tr>
<tr>
<td>PID (PSO)</td>
<td>0%</td>
<td>206 sec</td>
</tr>
<tr>
<td>PID (BF-PSO)</td>
<td>0.19%</td>
<td>33.2 sec</td>
</tr>
</tbody>
</table>

Table IV indicates error indices of set point tracking, step disturbance and pulse disturbance rejection.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Kp</th>
<th>Ki</th>
<th>Kd</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID (Z-N Tuning)</td>
<td>10</td>
<td>0.38</td>
<td>150</td>
</tr>
<tr>
<td>PID (PSO Tuning)</td>
<td>0.0788</td>
<td>0.0099</td>
<td>0.0058</td>
</tr>
<tr>
<td>PID (BF-PSO Tuning)</td>
<td>2.7200</td>
<td>0.0703</td>
<td>18.2497</td>
</tr>
</tbody>
</table>
Table V. indicates error indices of PID controller using Z-N tuning, PSO and BF-PSO algorithm.

<table>
<thead>
<tr>
<th>Controller</th>
<th>IAE</th>
<th>ISE</th>
<th>ITAE</th>
<th>ITSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID (Z-N Tuning)</td>
<td>3.93</td>
<td>0.31</td>
<td>131.8</td>
<td>3.98</td>
</tr>
<tr>
<td>PID (PSO Tuning)</td>
<td>2.85</td>
<td>0.14</td>
<td>106.8</td>
<td>2.54</td>
</tr>
<tr>
<td>PID (BF-PSO Tuning)</td>
<td>8.7</td>
<td>1.00</td>
<td>325.9</td>
<td>23.61</td>
</tr>
</tbody>
</table>

Extensive simulation results and analysis proves that in BF-PSO tuned PID controller, the overshoot is less than the Z-N tuning method but somewhat higher than the PSO algorithm and settling time of BF-PSO tuned PID controller is less than both Z-N tuning method and PSO algorithm i.e 33.12 sec. Hence it is proved that BF-PSO algorithm is better than Z-N tuning method and PSO in terms of set point tracking, simplicity, consistency, search and computational efficiency.

REFERENCES