Hemispherical Tank Liquid Level Process Control

Parthasarathy, R¹, Balaji, P²

¹ Student, Dept. of EIE, St. Peter’s Institute of Higher Education and Research, Chennai.
² Professor and Head, Dept. of E&I, St. Peter’s Institute of Higher Education and Research, Chennai.

Abstract: An intelligent optimization method for designing PID controllers based on particle swarm optimization (PSO) is presented in this paper. The conventional gain tuning of PID controller (such as Ziegler-Nichols (ZN) method) usually produces a big overshoot, and therefore modern heuristics approach such as genetic algorithm (GA) and particle swarm optimization (PSO) are employed to enhance the capability of traditional techniques. However, due to the computational efficiency, only PSO will be used in this paper. The performance comparison of the ZNPI and PSO based PI controllers are compared based on performance indices like maximum peak overshoot, settling time, Integral Square Error (ISE) and integral absolute error (IAE).

The proposed PSO based PI controller is tested on the chosen hemispherical Tank level system and better controller performance can be envisaged by in the proposed methods than that of the ZNPI controller.

Keywords: Hemispherical Tank, Optimization, Particle Swarm Optimization, PI Controller.

I. INTRODUCTION

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behaviour of bird flocking or fish schooling’s shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. PSO algorithm in finding optimal values follows the work of this animal society. Particle swarm optimization consists of a swarm of particles, where particle represent a potential solution. [1-4] In past several years, PSO has been successfully used across a wide range of application fields as well as in specific applications focused on a specific requirement for the two reason following. First it is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods and second reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications [5-7]. The present work deals with the design of controller for hemispherical tank system. The contribution of this work consists mainly in the design of Kp, Ki, and Kd values are found using three types of Particle swarm optimization techniques to design the PID controller and compared with conventional one. The development and implementation of the proposed system and controllers was done using MATLAB/Simulink.

II. HEMISPHERICAL TANK LEVEL PROCESS

First a hemispherical tank laboratory level process whose parameters vary with respect to process variable is considered for simulation and real time implementation. Even though the selected process is simple, it has high nonlinearity. The state variables of the system considered for study can be measured easily and the system is pure single input single output system. A shift in the operating point towards top of the tank will increase the time constant and static gain. Similarly a shift in the operating point towards the bottom of the tank will decrease in time constant and static gain. Thus, the hemispherical tank level process whose time constant and gain are functions of the process variable becomes suitable for the present work.

A. Mathematical Modelling Of Hemispherical Tank Level Process

The hemispherical tank level process shown in Fig.1 is considered for study.

Height of the liquid level in this tank is h meter. Volume of liquid in the hemispherical tank is given by

\[ V = \frac{\pi h (r^2 + h^2)}{6} \]  \hspace{1cm} (1)

Where \( r \) is the radius of the liquid top surface. \( r \) may be eliminated by using the Pythagorean relation

\[ R^2 = r^2 + (R - h)^2 \]  \hspace{1cm} (2)

Where \( R \) is the total height as well as radius of the tank
Further elementary algebraic manipulation yields

\[ V = \pi \left( R^2 \frac{1}{3} h^3 \right) \]  \hspace{1cm} (3)

Where \( V \) is the volume of the liquid in the tank

Differentiating the above equation with respect to \( t \) we have

\[ \frac{dV}{dt} = \pi \left( 2Rh^2 \frac{1}{3} \frac{dh}{dt} \right) \]  \hspace{1cm} (4)

The mass balance equation governing the system by assuming constant density of the liquid is

\[ \frac{dV}{dt} = F_{\text{in}} - F_{\text{out}} \]  \hspace{1cm} (5)

Where \( F_{\text{in}} \) and \( F_{\text{out}} \) are the inflow and outflow rates of the process respectively. \( F_{\text{out}} \) is assumed to be proportional to \( \sqrt{h} \) and is given by

\[ F_{\text{out}} = c\sqrt{h} \]  \hspace{1cm} (6)

Where \( c \) is the constant of proportionality. A delay time \( (T_d) \) is introduced in the inflow \( F_{\text{in}} \) to incorporate a dead time in the process and \( u \) is linear function. The equation becomes

\[ \frac{dV}{dt} = u(t - T_d) - c\sqrt{h} \]  \hspace{1cm} (7)

Substitute Eq. (4) in (7), we have

\[ \frac{dh}{dt} = \frac{u(t - T_d) - c\sqrt{h}}{\pi(2Rh^2)} \]  \hspace{1cm} (8)

This is the mathematical model of the hemispherical tank level process

### III. Tuning of PI Controller

The goal of PI controller tuning is to determine parameters that meet closed loop system performance specifications, and to ensure the robust performance of the control loop over a wide range of operating conditions. Practically, it is often difficult to simultaneously achieve all of these desirable qualities. For example, if the PI controller is adjusted to provide better transient response to set point change, it usually results in a sluggish response when under disturbance conditions. On the other hand, if the control system is made robust to disturbance by choosing conservative values for the PI controller, it may result in a slow closed loop response to a set point change. A number of tuning techniques that take into consideration the nature of the dynamics present within a process control loop have been proposed (Ziegler and Nichols, 1942; Cohen and Coon, 1953; Aström and Hägglund, 1984; and Atherton, 1993).[13-15] All these methods are based upon the dynamical behavior of the system under either open-loop or
closed-loop conditions. The Simulink model of the hemispherical tank level process is shown in figure 2. The height and top radius of the hemispherical tank are assumed as 30 cm and 15 cm respectively. A dead time ($T_d$) of 30 seconds is introduced in the process through software. The time constant and the gain of the process increase as the level increases. To obtain the transfer function model, reaction curves for various magnitudes of input at 50% percentage of nominal operating point are obtained by MATLAB Simulink software as shown in the figures 3. Different step changes in input (say) 10%, 20%, and 30% are given to obtain reaction curves. The corresponding process gain and time constant are tabulated in the Tables 1.

![Fig. 2 Simulink model of hemispherical tank level process](image)

![Fig. 3 Reaction curve for different set values](image)

<table>
<thead>
<tr>
<th>Step change</th>
<th>Process gain $K_p$</th>
<th>Time constant $\tau_p$</th>
<th>Dead Time $T_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>30%</td>
<td>2.09</td>
<td>1424.58</td>
<td>41.72</td>
</tr>
<tr>
<td>20%</td>
<td>2.011</td>
<td>1422</td>
<td>22.35</td>
</tr>
<tr>
<td>10%</td>
<td>1.926</td>
<td>1383.75</td>
<td>8.75</td>
</tr>
<tr>
<td>-10%</td>
<td>1.76</td>
<td>1206.8</td>
<td>28</td>
</tr>
<tr>
<td>-20%</td>
<td>1.674</td>
<td>1098.96</td>
<td>40.18</td>
</tr>
<tr>
<td>-30%</td>
<td>1.589</td>
<td>989.55</td>
<td>52.45</td>
</tr>
</tbody>
</table>

The gain of the system varies from 1.589 to 2.09 and the time constant varies from 989.55 to 1424.58 sec as the level varies from 25% to 80%. For the simulation study, the reaction curve for +10% change at 50% nominal operating point is considered to tune the PI controller. The tuning parameters obtained are $K_c=9.123$ and $T_i=0.136$.

**IV. PARTICLE SWARM OPTIMIZATION**

PSO is an evolutionary computational technique based on the movement and intelligence of swarms looking for the most fertile feeding location. A “swarm” is an apparently disorganized collection (population) of moving individuals that tend to cluster.
together, while each individual seems to be moving in a random direction. PSO uses a number of agents (particles) that constitute a swarm moving around in the search space looking for the best solution [2,8,9]. Each particle is treated as a point in an n-dimensional space and adjusts its “flying” according to its own flying experience, as well as the flying experience of other particles. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) that has been achieved so far. This value is called pbest. Another best value called gbest is that obtained so far by any particle in the neighbours of the particle. The PSO concept consists of changing the velocity (or acceleration) of each particle toward its pbest and the gbest position at each time step. Each particle tries to modify its current position and velocity according to the distance between its current position and pbest, and the distance between its current position and the gbest. At each step n, by using the individual best position, pbest, and global best position, gbest, a new velocity for the ith particle is updated by,

\[ v_{i}^{k+1} = w_{k} v_{i}^{k} + c_{1} r_{1} \times (p_{\text{best}_i} - x_{i}^{k}) + c_{2} r_{2} \times (g_{\text{best}} - x_{i}^{k}) \]  

(9)

\[ x_{i}^{k+1} = x_{i}^{k} + v_{i}^{k+1} \]  

(10)

With regards to (9):

- \( w_{k} \) = Inertial Weight
- \( v_{i}^{k} \) = current velocity of agent i at iteration k
- \( v_{i}^{k+1} \) = new velocity of agent i at iteration k+1
- \( c_{1}, c_{2} \) = adjustable social acceleration constant (swarm confidence),
- \( r_{1}, r_{2} \) = random number between 0 and 1,
- \( x_{i}^{k} \) = current position of agent i at iteration k,
- \( \text{pbest}_i \) = personal best of agent i,
- \( \text{gbest} \) = global best of the population.

For (10):

- \( x_{i}^{k+1} \) = position of agent i at the next iteration k+1,

The parameter ‘W’ in Equation (2) is inertia weight that increases the overall performance of PSO. It is reported that a larger value of ‘W’ can favour higher ability for global search while lower value of W implies a higher ability for local re-search. To achieve a higher performance, we linearly decrease the value of inertia weight W over the generations to favour global re-search in initial generations and local re-search in the later generations. The linearly decreasing value of inertia is expressed in Equation (11).

\[ w = w_{\text{max}} - \text{iter} \times \frac{w_{\text{max}} - w_{\text{min}}}{\text{iter}_{\text{max}}} \]  

(11)

Where \( \text{iter}_{\text{max}} \) is the maximum of iteration in evolution process, \( w_{\text{max}} \) is maximum value of inertia weight, \( w_{\text{min}} \) is the minimum value of inertia weight, and iter is current value of iteration.

Once the particle computes the new \( x_{i} \) it then evaluates its new location. If fitness (\( x_{i} \)) is better than fitness (\( p_{\text{best}} \)), then \( p_{\text{best}} = x_{i} \) and fitness (\( p_{\text{best}} \)) = fitness (\( x_{i} \)). In the end of iteration the fitness (\( g_{\text{best}} \)) = the better fitness (\( p_{\text{best}} \)), and \( g_{\text{best}} = p_{\text{best}} \).

The PSO algorithm method has been implemented as M file by MATLAB which is interconnected to the Simulink model, where the PID controller parameters are computed and fed to the GUI of the controller. The optimization performed with this initial parameter, number of particles 30, number of dimensions 3, maximum iteration 50, \( C_{1}=1 \), \( C_{2}=3 \), with the objective function ITAE or ISE. The initial values of three parameters \( K_{p} \), \( K_{i} \) and \( K_{d} \) of the PID controller will be generated in PSO program and submitted and running the simulation automatically then compute the objective function ITAE and go back with value of ITAE to PSO program to improve the value of \( K_{p} \), \( K_{i} \) and \( K_{d} \), and go on. In the end of iteration the parameters of the PID controller \( K_{p} \), \( K_{i} \), \( K_{d} \) has been obtained directly according to the minimum value of objective function ITAE. Fig. 4, shows the flowchart of PSO based PID tuning algorithm.
A. The design steps of PSO based PID controller

1) Initialize the algorithm parameters like a number of generations, population, inertia weight, cognitive and social coefficients.
2) Initialize the values of the parameters $K_p$, $K_i$ and $K_d$ randomly.
3) Calculate the fitness function of each particle in each generation.
4) Calculate the local best of each particle and the global best of the particles.
5) Update the position, velocity, local best and global best in each generation.
6) Repeat the steps 3 to 5 until the maximum iteration reached or the best solution is found.

V. PERFORMANCE OF HEMISPHERICAL TANK WITH PSO BASED PI CONTROLLER

The performance of Particle Swarm Optimization based PI controller for hemispherical tank level process is compared with conventional ZN PI.

Fig. 4 flow chart of PSO based PID tuning

(a) Servo Response  
(b) Regulatory Response

Fig. 5 Hemispherical tank level for 10% increment and decrement in load from nominal operating load of 25% using PSO tuned PI controller
Fig. 6 Hemispherical tank level for 10% increment and decrement in load from nominal operating load of 50% using PSO tuned PI controller

Fig. 7 Hemispherical tank level for 10% increment and decrement in load from nominal operating load of 75% using PSO tuned PI controller

The performance of Particle Swarm Optimization based PI controller for hemispherical tank level process is compared with conventional ZN PI. Fig. 5(a) shows the servo response of 10% increase in set point from nominal operating point and 10% decrement in setpoint from the nominal operating point of 25%. Fig. 5(b) shows the regulatory response of both positive and negative load change of 10% at a nominal operating load of 25%.

Fig. 6(a) shows the servo response of 10% increment in set point from nominal operating point and 10% decrement in set point from the nominal operating point of 50%. Fig. 6(b) shows the regulatory response of both positive and negative load change of 10% at a nominal operating load of 50%.

Fig. 7(a) shows the servo response of 10% increment in set point from nominal operating point and 10% decrement in set point from the nominal operating point of 75%. Fig. 7(b) shows the regulatory response of both positive and negative load change of 10% at a nominal operating load of 75%. PSO based PI controllers give responses with no oscillations, smaller ISE and IAE. In the case of all PSO PI, it is settling time better than ZN PI controller as given in Tables. 2.

Table 2 (a) Performance index for Hemispherical Tank at various nominal operating points for Servo Response

<table>
<thead>
<tr>
<th>Nominal operating point</th>
<th>Controller</th>
<th>Servo Response 10% increment in SP</th>
<th>Servo Response 10% decrement in SP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>over shoot (%)</td>
<td>Settling time (s)</td>
</tr>
<tr>
<td>25%</td>
<td>ZNPI</td>
<td>58</td>
<td>2479</td>
</tr>
<tr>
<td></td>
<td>PSO PI</td>
<td>18.4</td>
<td>920</td>
</tr>
<tr>
<td>50%</td>
<td>ZNPI</td>
<td>66.2</td>
<td>4415</td>
</tr>
<tr>
<td></td>
<td>PSO PI</td>
<td>24.2</td>
<td>1380</td>
</tr>
<tr>
<td>75%</td>
<td>ZNPI</td>
<td>70.5</td>
<td>4052</td>
</tr>
<tr>
<td></td>
<td>PSO PI</td>
<td>27</td>
<td>1163</td>
</tr>
</tbody>
</table>
Table 2 (b) Performance index for Hemispherical Tank at various nominal operating points for Regulatory response

<table>
<thead>
<tr>
<th>Nominal operating point</th>
<th>Controller</th>
<th>Regulatory Response</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>10% increment in load</td>
</tr>
<tr>
<td></td>
<td></td>
<td>under shot (%)</td>
</tr>
<tr>
<td>25%</td>
<td>ZNPI</td>
<td>93.8</td>
</tr>
<tr>
<td></td>
<td>PSO PI</td>
<td>33.4</td>
</tr>
<tr>
<td>50%</td>
<td>ZNPI</td>
<td>74.7</td>
</tr>
<tr>
<td></td>
<td>PSO PI</td>
<td>24.2</td>
</tr>
<tr>
<td>75%</td>
<td>ZNPI</td>
<td>102.6</td>
</tr>
<tr>
<td></td>
<td>PSO PI</td>
<td>27.4</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

PSO based PI controller are used to control the level in the hemispherical tank. It has been shown that the speed of responses of the level control system with and without load interrupt in the tank are fast. In order to appraise the performance of the controller, the proposed controller was done with MATLAB/Simulink. The PSO tuned PI controller offers enhanced process characteristics such as better time domain specifications, smooth reference tracking, supply disturbance rejection, and error minimization compared with ZN PI. In addition, the PSO - PI controller enhanced the flexibility and stability of the hemispherical tank level process.

REFERENCES