

# Particle swarm optimization based PID controller tuning for level control of two tank system

**Anju K Vincent and Ruban Nersisson**

School of Electrical Engineering, VIT University, Vellore - 632014, Tamil Nadu, India

Email: nruban@vit.ac.in

**ABSTRACT:** Automatic control plays a vital role in industrial operation. In process industries, in order to have an improved and stable control system, we need a robust tuning method. In this paper Particle Swarm Optimization (PSO) based algorithm is proposed for the optimization of a PID controller for level control process. A two tank system is considered. Initially a PID controller is designed using an Internal Model Control (IMC). The results are compared with the PSO based controller setting. The performance of the controller is compared and analyzed by time domain specification. In order to validate the robustness of PID controller, disturbance is imposed. The system is simulated using MATLAB. The results show that the proposed method provides better controller performance.

## 1. INTRODUCTION

PID controllers are broadly utilized as a part of process control framework. The tuning of PID controller is very important to have an optimum control. The most common methods are Ziegler-Nichols method and Cohen-Coon methods [1]. In physical system the parameters of the system change as the operation conditions fluctuate. Usually most of the controllers are optimized for one operating condition. Therefore the controller shows poor performance when it is made to operate at different operating condition.

So the controllers should be retuned regularly [8]. Thus many new tuning methods were proposed to achieve better performance. To avoid the retuning of PID controller, a Particle Swarm Optimization (PSO) based robust method is used. PSO uses swarm intelligence. The method got motivation by observing the social interaction, behaviours of animals seen among birds and fishes [3].

## 2. SYSTEM MODELLING

Consider a non-interacting two tank system as shown in figure 1. Let  $h_1$  and  $A_1$  be the height and area of the tank 1. Let  $h_2$  and  $A_2$  be the height and area of tank 2. Let  $R_1$  and  $R_2$  be the resistance of the flow valves and  $Q_{in}$ ,  $Q_1$  and  $Q_o$  be the flow rates in  $\text{ft}^3/\text{min}$ . Here we consider the density of the liquid to be constant [10]. Therefore by mass balance equation:

Tank 1

$$A_1 \frac{dh_1}{dt} = Q_{in} - Q_1 \quad (1)$$

$$= Q_{in} - \frac{h_1}{R_1} \quad (2)$$

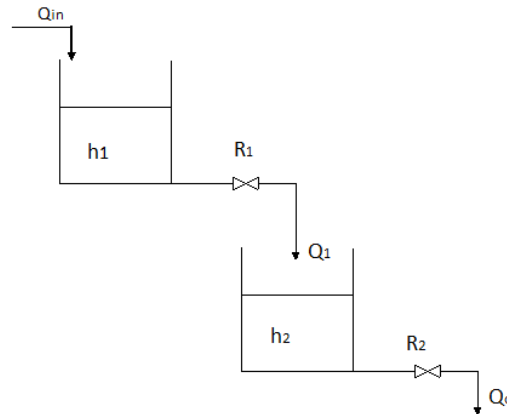
By taking Laplace transform we get,

$$A_1 s H_1(s) = Q_{in}(s) - \frac{H_1(s)}{R_1} \quad (3)$$



$$\frac{H1(s)}{Qin(s)} = \frac{R1}{\tau1s+1} \quad (4)$$

Where  $\tau1 = A_1 R_1$



**Figure 1. A two tank system**

Similarly for tank 2

$$\frac{H2(s)}{H1(s)} = \frac{R2}{R1(\tau2s+1)} \quad (5)$$

Where ;  $\tau2 = A_2 R_2$

Therefore the transfer function of the system is obtained by multiplying equation 4 and 5. The final transfer function is

$$\frac{H2(s)}{Qin(s)} = \frac{R2}{(\tau1s+1)(\tau2s+1)} \quad (6)$$

### 3. METHODOLOGY

#### 3.1 Internal Model Control (IMC)

IMC gives a controller with a single tuning parameter, the IMC filter coefficient ( $\lambda$ ). IMC is a technique, which is used for various controller tuning and designs. IMC is used to design a controller that satisfies basic demands [8]. Thus IMC is widely used in industries where there is need of single-loop adjusts. IMC system is used to track the set point and reject disturbance. The IMC-based PID controller can solve more industrial automation processes. IMC PID controller algorithm is found to be simple and it also gives a better solution in process that has large time delay [9].

The PID parameters are found by using the following equations

$$K_p = \frac{\tau1 + \tau2}{k_p} \quad (7)$$

$$T_i = \tau1 + \tau2 \quad (8)$$

$$T_d = \frac{\tau1\tau2}{\lambda(\tau1+\tau2)} \quad (9)$$

**3.1.1 Selection of Filter Time Constant ( $\lambda$ ).** Here the controller is designed by taking values  $\lambda$  as 0.5, 1 and 3. It is found that the system shows better performance better time domain specification when  $\lambda$  is 1. So we take the value of  $\lambda$  as 1 [9].

#### 3.2 Optimization Using Particle Swarm Algorithm

PSO algorithm is based on (SI) swarm intelligence. The method got motivation by observing the social interaction, behaviours of animals seen among birds, fishes etc. PSO follows the method that is found in fishes, where they find food by competing and the cooperating among themselves [3]. The swarm has individuals which are called particles in which each particle represents various possible set of the parameters that are unknown which should get optimized. A 'swarm' is usually initialized by a population of random solutions [8]. In this system, particles fly around in a multi-dimensional search space. It keeps on adjusting its position with respect to its own experience and also by considering the experience of its neighboring particle. The goal of each particle is to search a solution very efficiently to achieve this [10]. The particles swarm among themselves and moves to the best function which is

called fitting function. Then it converges to a single min or max solution. A function is already defined and that function is used to analyse the performance of that particle. The accuracy of the controller that is tuned depends on model's accuracy. So the system model is important. The only objective of this work is to use the proposed PSO to attain the optimal parameter values of a PID controller that is used in a two tank process[14].

Here we initialize a system with a population that has random solutions. They are called particles. And a random velocity is assigned to each of them. PSO depends on the information that get exchanged between swarms (particles) [9]. Each swarm adapts its path to its best fitness function that has been achieved till that moment. This value is referred as pbest. Moreover swarms adjust its path accordingly by considering the best previous position that was achieved by its neighboring member. It is referred as gbest. In the search space the particles moves with a velocity which is adaptive in nature [8].

A function is used to analyze the performance of swarm; so that we can find whether it has attained the best solution. This function is called fitness function [11]. As the swarming takes place, each particle tries to attain its best function and by the end, particles shows stagnating trend. Through this process each particle gets optimized.

Consider  $D$  as the dimension of search space.

Where  $X_i = [x_{i1}, x_{i2}, \dots, x_{id}]^T$  which represents the present position  $i^{\text{th}}$  particle.

Then,  $X_{ipbest} = [x_{i1pbest}, x_{i2pbest}, \dots, x_{idpbest}]^T$  represents best position that it visited[14].

$X_{gbest} = [x_{1gbest}, x_{2gbest}, \dots, x_{idgbest}]^T$  represents gbest, the best position that has been achieved by the particle in the population.

$V_i = [v_{i1}, v_{i2}, \dots, v_{id}]^T$  refers to the velocity of  $i^{\text{th}}$  particle.

$V_{i\max} = [v_{i1\max}, v_{i2\max}, \dots, v_{id\max}]^T$  represent the upper bound velocity of the particle that is used to move from one step to next step [8].

$$V_{id} = W * V_{id} + c_1 * r_1 * (X_{id\text{pbest}} - X_{id}) + c_2 * r_2 * (X_{gbest} - X_{id}) \quad (10)$$

$$V_{id} = V_{d\max} \quad \text{if } V_{id} > V_{d\max} \quad (11)$$

$$V_{id} = -V_{d\max} \quad \text{if } V_{id} < -V_{d\max} \quad (12)$$

$$X_{id} = X_{id} + V_{id} \quad (13)$$

Where;

$c_1$  and  $c_2$  are constants; that is used to represent the cognitive and social parameter.

$r_1$  and  $r_2$  are random numbers in the range [0,1]

$w$  is the inertia weight to balance the ability of the search[5].

The fitness function is given by;

$$F = (1 - e^{-\beta})(M_p + E_{ss}) + (e^{-\beta})(T_s - T_r) \quad (14)$$

$M_p$ : Peak Overshoot

$T_s$ : Settling Time

$T_r$ : Rise Time

$\beta$ : Scaling Factor(here I is taken as 1)

The parameters for PSO should be selected as it decides how much able is the algorithm, so that we get optimal value [7].

Here; The size of population = 100; Number of iterations=100;

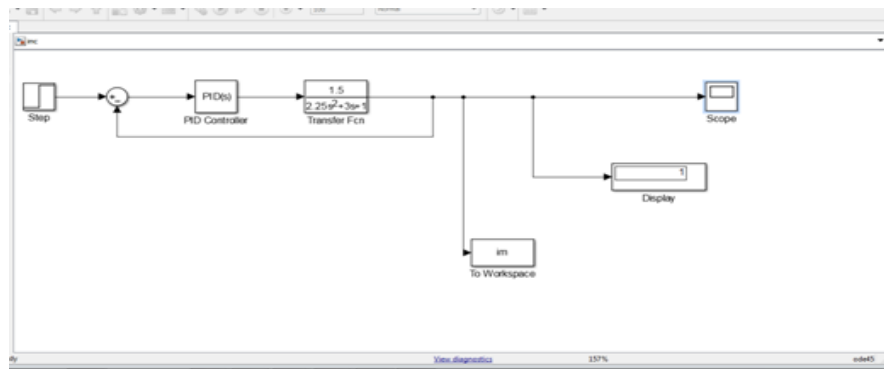
$c_1=1.2$ ;  $c_2=1.2$ ;  $r_1=1$ ;  $r_2=1$

### Algorithm

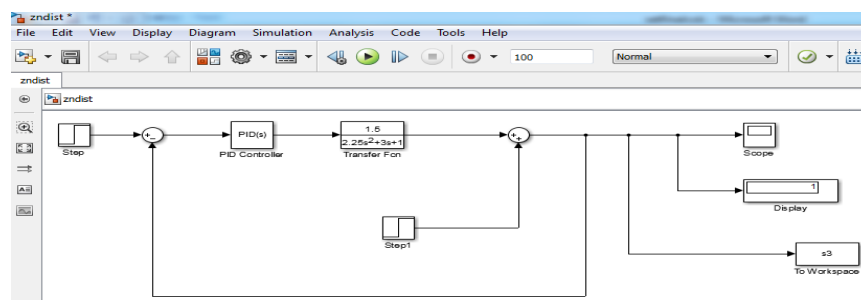
- Step i: Start  
 Step ii: Initialize the particles with a random place and velocity  
 Step iii: Evaluate the fitness function  
 Step iv: If the present value is more than pbest,  
           then goto → step v  
           else  
               to step viii.  
 Step v: Then Pbest is equal to present fitness value  
 Step vi: If the present fitness value is more than Gbest  
           then goto → step vii  
           else  
               to step viii  
 Step vii: Then Gbest is equal to the present value of fitness function  
 Step viii: Update the position & velocity values of the particles  
 Step ix: Exit if it meets end criteria  
           else  
               to step iii.

### 4. RESULTS

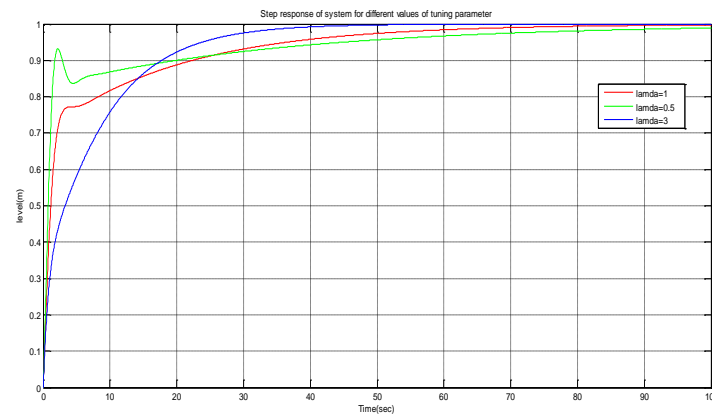
The system is modelled and implemented in simulink. PID controller is designed using IMC and PSO algorithm. When a disturbance is introduced to the system it is found that the system having PID controller designed using PSO algorithm shows better performance compared to that of IMC technique. When IMC is used the system takes 91.44 seconds to settle whereas when PSO algorithm is used the system takes 63.9 seconds to settle that is, when disturbance is introduced system with PSO algorithm settles fast. Thus it is found that PSO algorithm shows better disturbance rejection than IMC technique.



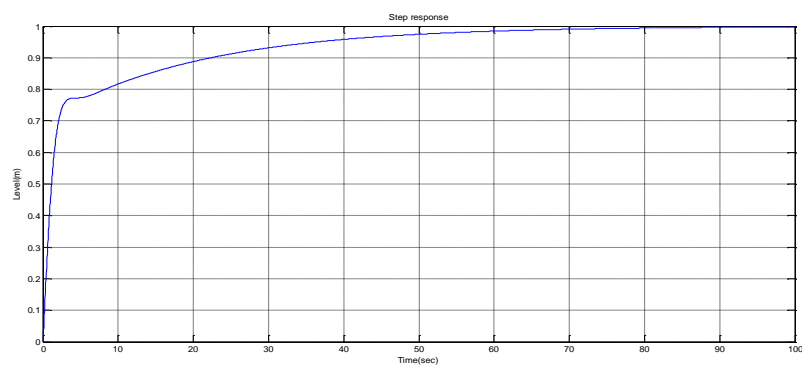
**Figure 2.** Simulink representation of the system



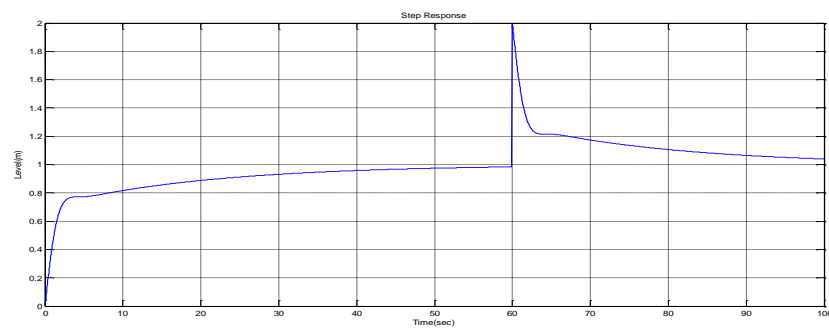
**Figure 3.** Simulink representation of the system with disturbance



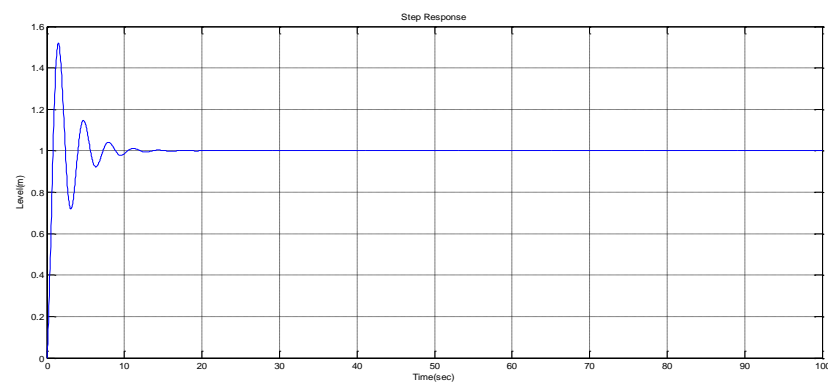
**Figure 4.** Step response of the system for different filter time constant ( $\lambda$ )



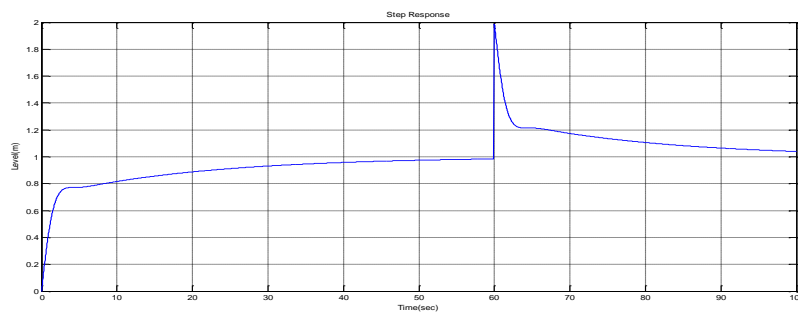
**Figure 5.** Step response of the system using IMC



**Figure 6.** Step response of the system using IMC with disturbance



**Figure 7.** Step Response of the system using PSO algorithm



**Figure 8.** Step Response of the system using PSO algorithm with disturbance

#### 4.1 Performance Comparison of the System:

Time domain specifications like rise time, settling time, peak overshoot is used to compare the performance of the system. Settling time and rise time is less for PSO based system. The IMC based tuning technique exhibits more overshoot.

**Table 1.** Comparison of time domain specification of the system without disturbance

TIME DOMAIN SPECIFICATION	IMC	PSO
Rise Time(sec)	21.7486	0.5783
Settling Time(sec)	53.0165	9.7825
Overshoot	0	51.9313
Peak Time(sec)	100	1.4706

**Table 2.** Shows the Comparison of time domain specification of the system with disturbance

TIME DOMAIN SPECIFICATION	IMC	PSO
Rise Time(sec)	31.1691	0.6583
Settling Time(sec)	91.4469	63.9942
Overshoot	90.8296	0
Peak Time(sec)	60.0000	60.0000

## 5. CONCLUSION

The effect of  $\lambda$  in tuning a controller has been analyzed. The controller is tuned using IMC and PSO. The result is compared by taking the time domain specification. It is found that the system having the controller that is designed using PSO technique is more robust and shows better disturbance rejection compared to the other technique.

## ACKNOWLEDGEMENT

The authors want to sincerely thank the VIT University for providing necessary resources and support to complete this work.

## References

- [1] J G Ziegler, N B Nicholas and J G Ziegler Optimum settings for automatic controllers *Trans ASME* **64** 759–768
- [2] G N Cohen and G A Coon Theoretical Consideration of Retarded Control *Trans ASME* **75** 827-834
- [3] Astrom K J and Hagglund T 1984 Automatic tuning of simple regulators with specifications on phase and amplitude margins *Automatica* **20** 645-651

- [4] Kennedy J F and Eberhart RC 1995 Particle swarm optimization. In: Proceedings of the IEEE International conference on neural networks **4**
- [5] Y Zheng, Liyan , Zhang, Jixin Qian and Longhua Ma 2003 Robust PID Controller Design using PSO *International Symposium on Intelligent Control IEEE*
- [6] T Bartz, Beielstein K E, Parsopoulos and M N Vrahatis 2004 Analysis of Particle Swarm Optimization Using Computational Statistics *International conference on numerical analysis and applied mathematics ICNAAM-2004*
- [7] Jun Zhao, Tianpeng Li and JixinQian 2005 Application of Particle Swarm Optimization Algorithm on Robust PID controller Tuning *Springerlink-Verlag Berlin Heidelberg* 948-957
- [8] S M GirirajKumar, R Sivasankar, T K Radhakrishnan, V Dharmalingam and N Anantharaman 2008 Particle Swarm Optimization Technique Based Design of PI Controller for a Real-Time Non-Linear Process Instrumentation *Science & Technology* **36(5)**
- [9] P Aravind M, Valluvan B and Sangeetha 2014 Model Based Controller Design for Level Process *International Journal for Computer Applications* **88(18)**
- [10] D R Coughanowar 1991 Process Systems Analysis and Control Tata McGraw Hill
- [11] Thomas, Beielstein, K E Parsopoulos and Michael N Vrahatis 2002 Tuning PSO Parameters Through Sensitivity Analysis, Technical Report of the Collaborative Research Center 531 Computational Intelligence, CI--124/02, University of Dortmund
- [12] Fan H 2002 A modification to particle swarm optimization algorithm *Engineering Computations* **19** 970-989
- [13] Kao, Chih-Cheng, Chin-Wen Chuang and Rong-Fong Fung 2006 The self-tuning PID control in a slider – crank mechanism system by applying particle swarm optimization approach *Mechatronics* **16.8** 513-522
- [14] Tayal M 2003 Particle swarm optimization for mechanical design *M. Sc. thesis* The University of Texas, Arlington, United States
- [15] W B Bequette 2003 *Process, Control Modeling and Simulation* Prentice Hall