PSO Based Optimization of A PI Controller for A Real Time Pressure Process

K Vijaya Shanthi[1] P Aravind[2] R Ramya[3] K Swathy[4]
[1,3,4]Student [2]Assistant Professor
[1,2,3,4]Department of Instrumentation & Control Engineering
[1,2,3,4]Saranathan College of Engineering, Trichy, India

Abstract— Controlling pressure is one of the significant control actions that has been carried out in the industries. The objective of the work is to uphold the pressure in the closed loop at desired set value. The pressurized tank has the features of nonlinearity, sluggishness by tuning conventional PI methods. This paper focus on the development and implementation of an intelligent optimization to obtain an optimum PI controller setting for pressure control process. System identification of the process is done by process reaction curve method and found to be First Order Plus Dead Time (FOPDT) model .At first, a Proportional Integral (PI) controller based on Ziegler Nichols (ZN)-PI setting is designed and the results are compared with particle swarm optimization(PSO) based PI controller settings. The performance comparison of the designed controller settings are analyzed by time domain specification. The simulation results confirm that PSO-PI controller has improved dynamic performance on disturbance rejection.

Key words: pressure process, PI controller, PSO, Ziegler Nichols, MATLAB

I. Introduction

To achieve the effective control standards, design of PSO based PI control is proposed in this work to maintain the pressure process at desired value. The PI control due to its simple arrangement, good robustness and wide application range, gradually highlighted in the control theory. PI tuning has certainly been the key to reasonable performance and robustness. PI controller setting is proposed for several process model, especially for First Order plus Time Delay (FOPDT). There are two commonly used method of PI tuning, they are Ziegler-Nichols setting [1] and Cohen-coon setting [2], which still used in several industrial applications. . It has been observed that, however, the PI controllers with the above mentioned tuning techniques may not perform well in the complex control processes, such as the higher-order system and time-delay system. Therefore, numerous effective PI controller design and tuning methods for complex processes have been stated.

Owing its simple structure and their real time implementation and tuning, the research community as well as the industrial pay attention towards computation intelligence (Muller SD, Marchetto J, Airaghi S, Koumoutsakos P (2002), JavedAlam Jan, BohumilSulc (2002), Y Zheng, Liyan Zhang, JixinQianLonghua Ma (2003)) to obtain the PI controller parameters. The computation efficiency is the advantage of particle swarm optimization algorithms over other tuning techniques. Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling [3] and [4]. PSO shares many similarities with other evolutionary computation techniques such as Genetic Algorithms (GAs)[5]. Compared to GA, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO has successfully applied in many areas such as function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied. PSO has already been a new and fast developing research topic. In this work the process dynamics is modeled by process reaction curve method [6]. For the developed model a ZN based PI control structure is designed and its performance measure is based on rise time, settling time and various performance indices are compared with PSO based PI controller. Paper is organized as follows. Section II describes the Mathematical Modeling of the pressure process. The implementation of ZN based PI controller and PSO based PI controller [7] and [8] are discussed in Section III. Section IV presents the comparison result. Finally, Section V presents Conclusion.

II. Experimental Setup

The physical experimental system comprises of process tank, pressure transmitter, control valve, pressure controller, air supply, current to pressure converter, vent valve, compressor, and air regulator

1. set point
2. vent valve
3. tank pressure
4. pressure transmitter
5. air supply
6. process tank
7. control valve
8. air supply
9. supply pressure
10. signal pressure
11. air regulator
12. I/P converter
13. pressure controller

TABLE 1: Components of experimental setup

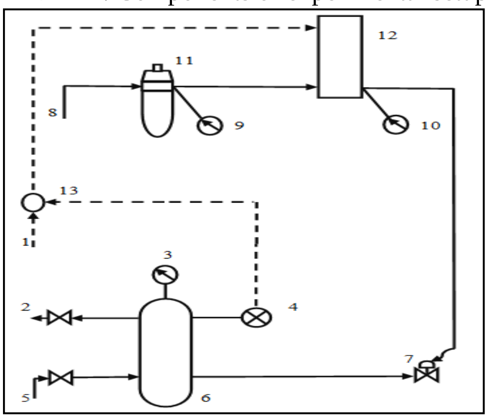

Fig. 1: Piping and Instrumentation Diagram for Real time pressure process

Part name	Description
Process tank	Opaque Pressure vessel
Pressure transmitter	Input 4-20mA, Output 4-20 psig
Air filter regulator	Range 0-2.5 kg/cm2
Control valve Type:	Pneumatic, Size: ¼", Input: 3-15 psig, Air to close, Linear
Pressure gauge	Range 0-2.5 kg/cm2 (1), Range 0-7 kg/ cm2 (2)
Current to Pressure converter	Input 4-20 mA, Output 3-15 psi
Communication	RS232

Table 2: Technical description of experimental setup

III. SYSTEM IDENTIFICATION – PRC METHOD

Process Reaction Curve method is commonly used in system identification which is based on the step response. Process identification can be used to build a consistent model, after the process has been positioned in operation. There are several graphical user interface toolbox available for modelling. In this method, the small step change is introduced with the help of manual controller. For every input, the transient is recorded which is called process reaction curve. In the graph, a straight line is drawn tangent to the transient curve at the point of inflection. The tangent line intersects the curve in time axis at a point called Transportation lag (τd). The apparent time constant (τ) and the steady state gain (kp) are measured [6]. The representation of s-shaped transient curve by FOPTD is given by

$$G(s) = \frac{k_p e^{-\tau_d (s)}}{\tau s + 1}$$

Where kp is the steady state gain, τd is the apparent transport lag and τ is the apparent first order time constant[9]. The model is calculated with the real time parameters as

$$G(s) = \frac{0.113\, e^{-2s}}{16s + 1}$$

IV. CONTROLLER DESIGN

This paper reports the implementation of PI parameters in two design setting. The ZN-PI method and PSO-PI control method. With these techniques, tuning of PI parameters is accomplished to achieve a robust design with the desired response time. The tuner computes PID parameters that robustly stabilize the system. PI controller is tuned by physically regulating design criteria in two design modes. The response has approximately the similar overshoot as proportional control, but the period is longer; however, the response proceeds to the set point after a comparatively extended settling period. The most advantageous effect of the integral action in the controller is the removal of offset.

A. *ZN Based PI:*

The procedures were first suggested by Ziegler and Nichols. They established a closed-loop tuning technique still used today. The method is designated as a closed-loop method because the controller remains in the loop as an active controller in automatic mode. Ziegler and Nichols did not recommend that the ultimate gain Kcu and ultimate Period Pu be computed from frequency response designs based on the model of the process. They proposed that Kcu and Pu be acquired from a closed-loop experiment of the real time process. When the instructions were first suggested, frequency response methods and process prototypes were not usually available to the control engineers. The guidelines are offered further down and are in the form that one would use for real application to an actual process. After the process influences steady state at the usual level of operation, eliminate the integral and derivative modes of the controller, parting only proportional control. Select a value of proportional gain Kc, disturb the system, and detect the transient response. If the response falloffs, select an upper value of Kc and again spot the response of the system. Remain increasing the gain in minor phases till the response first shows a sustained oscillation. The importance of gain and the period of oscillation that resemble to the continued oscillation are the ultimate gain (Ku) and the ultimate period (Pu).

Controller Type	K c	τ *I*	τ ***D***
Proportional (P)	0.5Ku	-	-
Proportional-Integral (PI)	0.45Ku	Pu/1.2	-

TABLE 3: ZN-PI CONTROL PARAMETERS

B. *Optimization Using Pso:*

PSO is a robust stochastic optimization technique based on the movement and cooperation of swarms. The application of PSO algorithm was put onward by several researchers who developed computational simulations of the movement of organisms such as schools of fish and flocks of birds. Such simulations were heavily based on manipulating the distances between individuals, i.e., the synchrony of the behavior of the swarm was seen as an effort to keep an optimal distance between them. Sociobiologist Edward Osbourne Wilson outlined a link of these simulations for optimization problems

1) *Selection of PSO Parameters:*

To start up with PSO, certain parameters need to be defined. Selection of these parameters decides to a great extent the ability of global minimization. Population size=100
Number of iterations=100
Velocity constant, c1=1.2
Velocity constant, c2=2.

a) Particle Swarm Optimization:

The 'swarm' is initialized with a population of random solutions. In a PSO system, particles fly around in a multi-dimensional search space adjusting its position according to its own experience and the experience of its neighboring particle. The goal is to efficiently search the solution space by swarming the particles towards the best fitting solution encountered in previous iterations with the intention of encountering better solutions through the course of the process and eventually converging on a single minimum or maximum solution. The performance of each particle is measured according to a pre-defined fitness function, which is related to the problem being solved.

In PSO algorithm, the system is initialized with a population of random solutions, which are called particles, and each potential solution is also assigned a randomized velocity (S. M. GirirajKumar, R. Sivasankar, T.K.

Radhakrishnan, V. Dharmalingam and N. Anantharaman (2008)). PSO relies on the exchange of information between particles of the population called swarm. Each particle adjusts its trajectory towards its best solution (fitness) that is achieved so far. This value is called pbest. Each particle also modifies its trajectory towards the best previous position attained by any member of its neighborhood. This value is called gbest. Each particle moves in the search space with an adaptive velocity. The fitness function evaluates the performance of particles to determine whether the best fitting solution is achieved. During the run, the fitness of the best individual improves over time and typically tends to stagnate towards the end of the run. Ideally, the stagnation of the process coincides

with the successful discovery of the global optimum.

velocity = w *velocity + c1*(R1.*(L_b_position-current_position)) + c2*(R2.* (g_b_position-current_position)) (8)

where c1 and c2 are positive constants, represent the cognitive and social parameter respectively; R1 and R2 are random numbers uniformly distributed and w is inertia weight to balance the global and local search ability. In general the PSO technique can be given by the following algorithm,

b) Algorithm:

Step1: Start the program
Step2: Initialize particles with random place and velocity.
Step3: Evaluate fitness value for each particle
Step4: If current fitness value is better than pbest, goto Step5 else goto step8.
Step6: Pbest equal to current fitness value
Step7: If current fitness value is better than Gbest, goto to Step8 else goto step 8
Step9: Gbest is equal to current fitness value.
Step10: Update position and velocity of particles
Step11: Goto step10 if stop criteria met else goto step3.

c) Termination Criteria:

Optimization algorithm will automatically terminate execution either when the number of iterations gets over or with the attainment of acceptable fitness value. Fitness value, in this case is nothing but reciprocal of the error, since we consider for a minimization of objective function. In this paper the termination criteria is considered to be the attainment of maximum number of iterations. For each iteration the best among the 100 particles considered as potential solution is chosen. Therefore, the best values for 100 iterations for the model is sketched and shown in figure2 and figure3 with respect to iterations for Kp and ki.

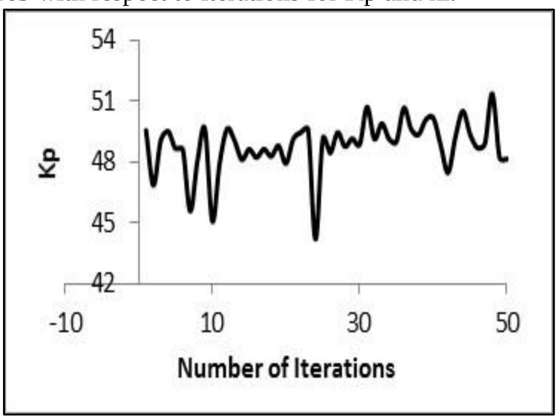

Fig. 2: Best solutions of Kp for 100 iterations

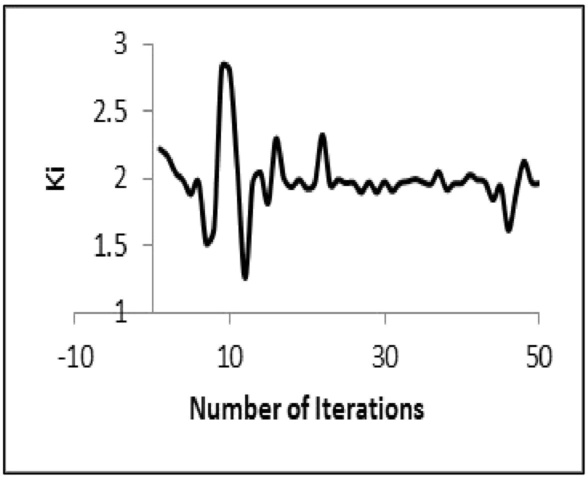

Fig. 3: Best solutions of Ki for 100 iterations

The PI controller was formed based upon the respective parameters for 100 iterations, and the gbest (global best) solution was selected for the set of parameters.

Controller	ZN	PSO
Kp	94.0265	49.5665
Ki	5.53097	2.2230

Table 4: Tuned Gain values of controller

Specifications	ZN	PSO
Rise Time (seconds)	5	7
Peak Time (seconds)	7	9
Peak overshoot(psi)	15.5	10.8
Settling time (seconds)	25	17

Table 5: Comparison of Time Domain Analysis

The performance estimation of proposed controllers are present in Table V based on time domain analysis. The response curve of the PSO based controller has the advantage of a better closed loop time constant, which enables the controller act faster with minimum settling time. The response of ZN controller is more sluggish and more oscillatory than the PSO based controller as shown in Fig 4.

V. Results and Comparison

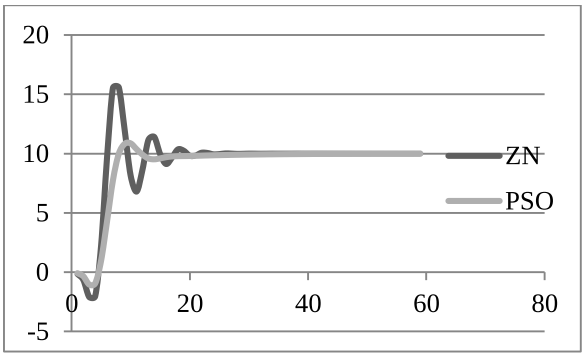

Fig. 4: Comparison of ZN and PSO-PI for a setpoint of 10 psi.

The controller parameters are calculated and implemented for set point 10 psi and shown in figure4. The servo response of the system was observed by giving set points of 10psi and then 20 psi. The corresponding variation of pressure from a reference value is noted. The responses of the pressure tank for the two set points with controller settings are presented in the Fig 5. The deviation of pressure from reference point is noted.

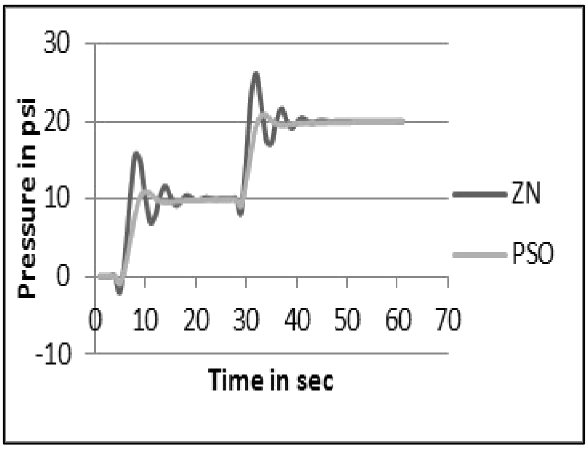

Fig. 5: Servo Response of a Process

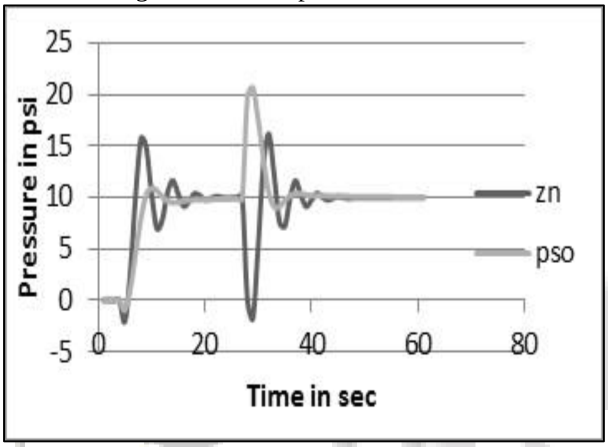

Fig. 6: Regulatory Response of a process

Figure 6 clearly states that how fast the PSO based controller reacts to disturbance compare to ZN based controller. A process is disturbed at the time of 30 seconds with 10% of setpoint, the proposed PSO based controller reacts faster and process variable attains steady state quicker than ZN based controller.

VI. CONCLUSIONS

In this work we have presented scenario based optimization algorithm through swarm intelligence. The PI controller parameters are obtained using PSO. The design, implementation and testing of such PSO based PI controller parameters are discussed in detail and compared with traditional tuning method. The simulation results are obtained which demonstrate the efficiency and effectiveness of the proposed tuning technique.

It is shown graphically that there is a substantial improvement in the time domain specification in terms of lesser settling time with the application PSO based PI settings. The performance of the proposed controller is also analyzed by applying setpoint change and load change and are presented in figure 5 and 6. Those results has proven that PSO-PI control setting is more effective in disturbance rejection and to enhance the stability of system.Thus we have proposed an effective method to design the PI controller that can be implemented in real time pressure process.

REFERENCES

[1] J. G. Ziegler and N. B. Nichols (1942), Optimum settings for automatic controllers, Trans. Amer. Soc. Mech. Eng., vol. 64, pp. 759–768, 1942
[2] G.H Cohen and G.A Coon, Theoretical Consideration of Retarded Control, Trans ASME 75, pp.827/834, 1953
[3] Kennedy JF, Eberhart RC.(1995), Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, vol. 4. Perth, Australia; PP. 1942– 48.
[4] Kennedy JF, Eberhart RC, Shi Y.(2001), Swarm intelligence. San Francisco: Morgan Kaufman.
[5] Hassen T.Dorrah, Ahmed M.El-Garhy, Mohammed E.El-Shimy, PSO based optimized fuzzy controller for decoupled highly Interacted distillation process, Ain shams Engineering Journal,
Volume 3, Issue 3, September 2012, pages 251-266.
[6] B.Diwakar, M.Gokul Anandh, R.Brindaa, S.R.Jayashree Devi, P.Aravind, IMC Based Design of PI controller for Real time pressure process, IJIREEICE, Vol 3, Issue 10 ,October 2015
[7] P.Aravind, M.Valluvan, PSO based optimization of a PID controller for a linear tank level control process, International Journal of Current Engineering and Technology, Vol 4, No.2, April 2014
[8] P.Aravind, M.Valluvan, B.Sangeetha(2014), Model based controller Design for level process, International Journal for computer applications, Volume 88-Number 18.

