International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 5, Issue 7, July 2016 ISSN 2319 - 4847

Parameter Estimation of PI Controller using PSO Algorithm for Level Control

[1]Bindutesh V.Saner ,[2]Bhagsen J.Parvat

[1,2]Department of Instrumentation & control Pravara Rural college of Engineering, Loni Maharashtra, India

ABSTRACT

In Industrial application mostly PI controller is use to achieve the desired output in case of closed loop control system. In PI controller sometimes it is difficult to obtain the proper value of controller parameter k_P &k_J. The paper delineate the design of control system model for PI controller using stochastic method named as Particle swarm optimization with the help of objective function. The solution provides minimum error which affects the control parameter such as rise time, maximum overshoot, settling time etc. of the system. The proposed method is demonstrated in simulation by tuning the PI parameter of FOPDT model.

Keywords:-PID (Proportional-Integral-Derivative), PSO (Particle Swarm optimization), optimization, Improved PSO, FOPDT model.

1.INTRODUCTION

PID controller is preferred due to its simplicity & reliability as well as robust performance. PID control algorithm is used to control almost every loop in process industries. The basic function of the controller is to execute an algorithm based on the control engineer's input and hence to maintain the output at a level so that there is no difference between the process variable and the set point [1]. Traditional method such as Zeigler Nicohls method [4], Cohen-Coon method [6], Astrom & Hugglund [5] are used for tuning PI-PID controller.

Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) etc. are generally used to tune the unknown parameters of PID controller. In this paper the transfer function of FOPDT model is attached with PI controller block. Tuning of parameters of the PI controller is carried out by using Particle Swarm Optimization (PSO).

The main objective of this paper is to implement the evolutionary algorithm, like particle swarm optimization as well as to understand in depth of particle swarm optimization. And finally the transient response of the system is taken and compares with the traditional method such as Ziegler Nichols method.

2.PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is an evolutionary computational technique introduced by Russell Eberhart and James Kennedy in 1995. This method is extremely robust in solving problems having non-linearity & non-differentiability. The basic operational principle of the particle swarm is applicable for the flock of birds or fish or for a group of people. While searching for food, the birds are either dispersed or go together before they locate the place where they can find the food. While the birds are searching for food from one place to another, there is always a bird that can smell the food very well, that is, the bird can observe the place where the food can be found. Because they are transmitting the information, especially the good information at any time while searching the food from one place to another, getting the good information, the birds will eventually flock to the place where food can be found [2-3].

In PSO particles move in a multi-dimensional search space with their own experience & experience of their neighboring particle. The goal is to find out a search solution space by swarming the particle towards best fitness solution available at the end of each iteration with the intension of finding a better solution through the course of process [7]. The performance of each particle is measured according to the predefined fitness function. The PSO algorithm operates by simultaneously maintaining several candidate solutions in the work space. It has got stable convergence characteristics and has a good computational efficiency.

According to the Kennedy & Eberhart the swarm is manipulated according to the following Equation:

The best previous positions of the i^{th} particle is represented as $P_i = (P_{i1}, P_{i2,} P_{i3} \dots \dots, P_{id}),\ P_{best}$. Therefore it is important to note that the personal best (P_{best}) is the best position that the individual particle has visited since the first time step.

The index of the best particle among the group is G_{best} represented as P_g the global best position is the best position discovered by any of the particles in the entire swarm. Velocity of the i^{th} particle is represented as $V_i = (V_{i1}, V_{i2,} V_{i3} \ \ldots \ldots, V_{id})$., The updated velocity and the distance(position) from $P_{i,d}$ to G_{id} is given as in equation(1) and (2) respectively.

$$V_i(t+1) = V_i(t) + C_1 R_1 (P_i(t) - X_i(t)) + C_2 R_2 (P_g(t) - X_i(t)) \qquad (1)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \qquad (2)$$

Where,
i=1, 2, 3, ….n is the particle's index,
C_1 and C_2 is positive acceleration constant,
R_1 and R_2 are random no. from uniform distribution which varies from 0 to 1.

A.PSO algorithm parameters

There are few parameters is PSO algorithms that may affect its performance. For instance the some of the parameters values and choices may have a large impact on the efficiency and convergence of PSO method. The basic parameters are no. of particles or swarm size, no. of iterations, velocity components, & accelerations coefficients.

SWARM SIZE: It's a no. of particle 'n' in the swarm. A big swarm generates larger part of search space to be covered per iterations. Large no. of particle may reduce the no. of iteration which is needed for good optimization and vice versa. It is observed that particle size must be within the interval of 20 to 100.

ITERATION NUMBER: It is also one of the main parameters to obtain a good result. More no. of iteration may add complexity and more time needed while less no. of iteration may end the process prematurely.

VELOCITY COMPONENTS: It is a parameter responsible for updating particle's velocity in equation (1). The term $V_i\ (t)$ is called inertia component which provides the memory of the previous flight directions & which held responsible for changing the particular directions of the particle. It also represents the momentum.
The term $C_1\ R_1\ (P_i\ (t) - (t)\ )$ is called as cognitive component which measures the performance of the particles relative to past performances. This also represents the tendency of individuals to return to positions that satisfied them most in the past.
The term $C_2\ R_2\ (P_g\ (t) - X_i\ (t))$ is called as social component which measures the performance of the particles relative to a group of particles. Which help the particle to locate the best particle.

ACCELERATION COEFFICIENTS: Random values of R_1 and R_2 together with the acceleration coefficient C_1 and C_2 Maintain a stochastic influence of social and cognitive components of the particle velocity respectively.C_1express the particle confidence in itself while C_2 express the confidence of particle in its neighbor.
When,
C_1=C_2 all particles continue moving at their current speed and attracted towards the average of P_{best} and G_{best}
C_1>0 & C_2 =0, all particles are independent to move.
C_1>0 & C_2=0, all particles move towards single point.

When C_1 & C_2 are static with their optimized values being found empirically. Wrong initialization of C_1 and C_2 may result in divergent or cyclic behavior. From different empirical

researches it has been notice that the acceleration constant should be.C_1=C_2=2.
Presences of random number increases the zig zag movement of particle's tendency as well as slows down convergence thus it improves the state space explorations and also prevent the premature convergence.

B.Convergence improvements

In PSO, particle velocity is very meaningful, since it is the step size of swarm. At every iteration, all particle proceed by adjusting the velocity that each particle moves in every dimension of the given search space.
Exploitation and Exploration are two important characteristics which have to be balance to have well optimized algorithms. Exploitation is an ability to concentrate a search around a searching area for refining a particular area, while Exploration is an ability to explore a different area of search space for locating a good optimum. Therefore in order to reduce or minimize the divergence, particle velocities are reduced in order to stay within the boundary constraints.
The inertia weight denoted by 'w' is added to controls the momentum of the particle by weighing the contribution of the previous velocity. The inertia weight 'w' will at every step be multiplied by the velocity at the previous time step. Therefore the velocity equation of the particle with inertia will be changed and represented as follow:

$$V_i(t+1) = W * V_i(t) + C_1 R_1 \left(P_i(t) - X_i(t)\right) + C_2 R_2 \left(P_g(t) - X_i(t)\right) \quad (3)$$

Shi and Eberhart in 1999 first introduced the inertia weight to reduce velocities over time, to control the exploitation and exploration abilities of particle, and to converge the swarm more accurately. If $w \geq 1$ velocities increase over time and particles can hardly change their direction to move back towards optimum. $w << 1$ little momentum is only saved from the previous step and quick changes of direction are too set in the process. *W=0* particles velocity vanishes and move randomly without knowledge. Inertia weight can be implemented as a fixed value or a dynamically changing value [8]. In order to control the balance between local and global exploration, to have quick convergence, to reach an optimum the inertia weight whose value must be decreased linearly with the iteration no. set according to below formulae:

$$W = W_{max} - iter.\frac{W_{max} - W_{min}}{iter_{max}} \quad (4)$$

Where,
*iter*max is the maximum of iteration in the evolution process, *Wmax* is the maximum value of inertia weight,
Wmin is the minimum value of inertia weight, and
iter is current value of iteration

C.Algorithm flow chart

The searching procedures of the proposed PSO algorithm is shown below,

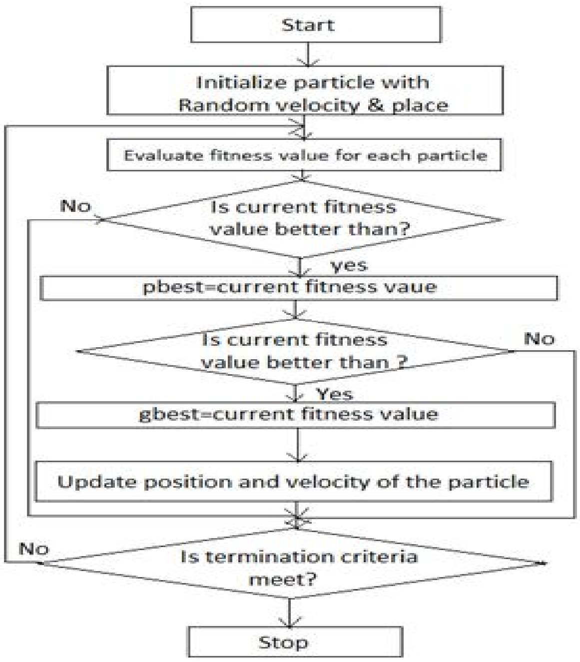

Figure 1 PSO algorithm flowchart

D.Evaluation of objective function

PSO search for the optimized value by minimizing or maximizing a given fitness function. Here in order to remove the negative error, square of error i.e. 1/ISE is used as a fitness function.

$$I_{ISE} = \int_0^t e^2(t)dt \quad (5)$$

E.Termination criteria

Termination of algorithm takes place as soon as the no. of iterations is over or else the required condition according to agreeable fitness function value. In this paper we have considered a minimization of fitness function as termination criteria.

3.IMPLEMENTATION OF PROPOSD METHOD

Considered a PI controller with the following transfer function as

$$G_c(s) = k_p + \frac{1}{\tau_i s} \qquad (6)$$

Where K_p and T_i are unknown to be determined, while K_p is proportional gain & T_i is the integral gain. Consider a stable first-order plus dead Time (FOPDT) process for level control application described by transfer function as

$$G_p(s) = \frac{0.74e^{-14\theta}}{137s+1} \qquad (7)$$

Using the Pade's approximation, the above transfer function is approximated. For the enhancement of the optimization process the PI controller is initially tuned with Zeigler-Nichols method to tune the parameter K_p and T_i later the closed loop transfer function is obtained. Thereafter the minimization of error is carried out with PSO Algorithm In figure 2. Block diagram of process is shown. Tuning of PI controller is done with Zeigler-Nichols second method where the value of K_u (ultimate gain) is 21.4108 and P_{cr} (Ultimate period) is 54.3 sec obtained.

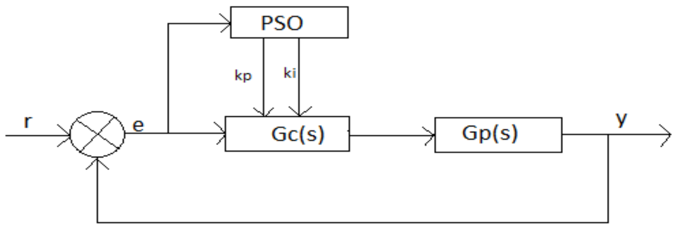

Figure 2 Block diagram of PSO-PI controller

4.SIMULATION & RESULT

The closed loop PI controller along with process is tuned with the optimal values of Kp and Ti using PSO algorithms. Few constraint ($1< K_p < 12$ & $0.01 < T_i < 0.08$) & below PSO parameter has been taken in consideration while optimizing PSO algorithms. The simulations results are carried out in matlab/simulink and their respective results are compared in below tables. In figure 5 the step response of the Z-N, & PSO is shown. While controller output and error deviation shown in figure 3 and 4 respectively.

A.Selection of PSO parameter

To initializes PSO certain predefined parameters are must for global minimization. Selection of these parameter help to retain the particle velocity in a specific bound, or else they can affect the global optimization. Size of swarm also helps to retain global optimization. Parameter of PSO is chosen as shown in table 1.

TABLE.1 PSO PARAMETERS VALUES

PSO Parameters	Values
Acceleration constant (C_1,C_2)	2
Particle size	100
No. of iterations	55
W_{max},, W_{min}	0.9,0.4
Random no. (R_1 & R_2)	0 to 1

TABLE.2 PI VALUE OBTAINED FOR ZN & PSO METHODS

PI parameters	ZN-PI	PSO-PI
Proportional gain	9.63485	7.8349
Integral(sec)	0.2129	0.0579

TABLE.3 COMPAISION OF TRANSIENT RESPONSE

Specifications	**Z-N method**	**PSO method**
Rise Time(sec)	14.017	20.733
Settling Time(sec)	159.05	80.599
Overshoot	54.84	11.208

TABLE.4 ROBUSTNESS COMPARISION

Performance index	**Z-N method**	**PSO method**
ISE	27.54	22.22
IAE	43.80	29.66
ITAE	1703	632.7

B.Controller output of Z-N and PSO.

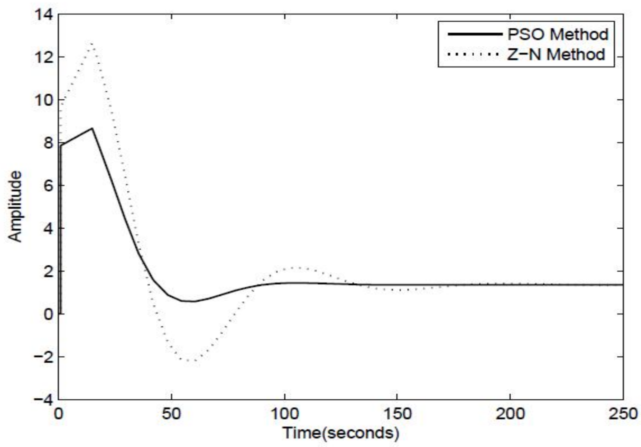

Figure 3 Controller Output (u)

C.Error deviation of Z-N and PSO.

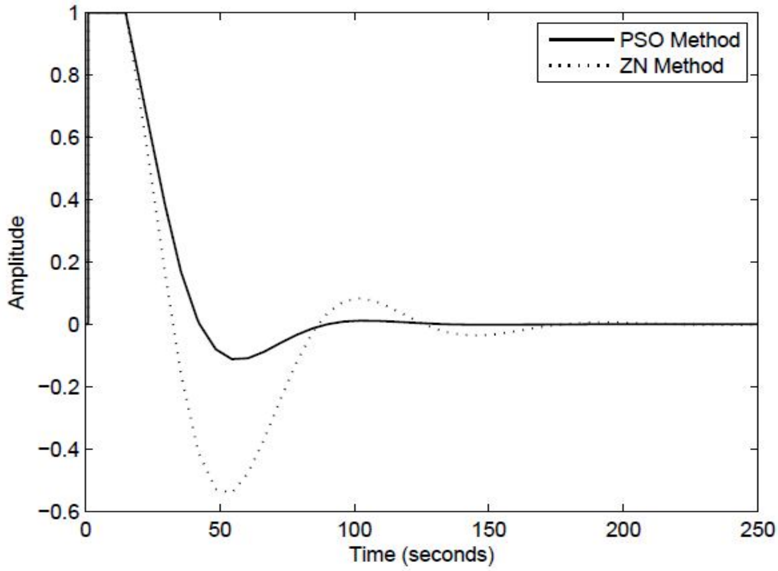

Figure 4 Error Deviation

D.Unit step response of Z-N and PSO.

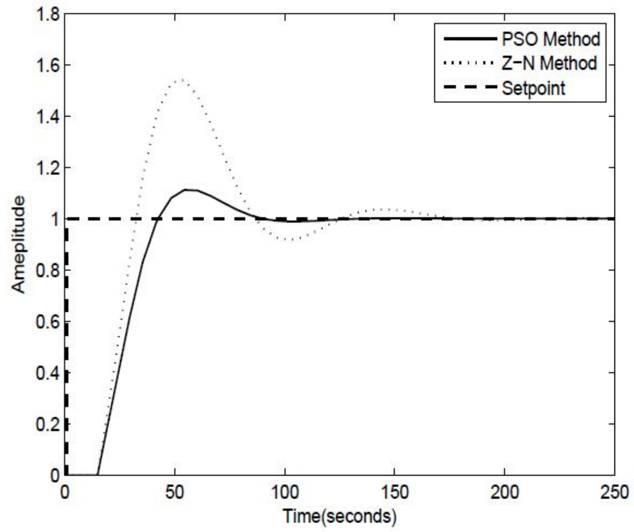

Figure 5 Step Response

5.CONCLUSION

In this paper the PSO-PI is tuned for level control application and has a great advantage in parameter estimations. The response of Improved Particle Swarm Optimization gives a better performance as compared to traditional method such as Ziegler and Nichols methods which helps to overcome the problem of overshoot & settling time. In PSO Gradual increase in no. of iteration gives good results as well as convergence takes place well. Therefore the benefit of using a modern optimization approach is observed as a complement solution to improve the performances of the PI controller designed by conventional method.

REFERENCE

[1]. M Araki: Control systems, Robotics and Automation-Vol II – PID Control –,Kyoto University ,Japan.

[2]. J. Kennedy and R.C. Eberhart, Swarm Intelligence , ISBN 1-55860-595-9 , Academic Press (2001)

[3]. J. Kennedy,"The Behaviour of Particles" Proceedings of the 7th International Conference on Evolutionary Programming,San Diego,california, USA,March,(1998);581-587.

[4]. Ziegler, G. and Nichols, N.B, 1942.Optimum settings for automatic controllers, Trans. ASME,64,759-768.

[5]. Astrom, K J.;.Hagglund .T,1984, Automatic tuning of simple regulators with specifications on phase and amplitude margins, Automatica, 20,645-651.

[6]. G.H Cohen and G.A Coon: Theoretical Consideration of Retarded Control, Trans ASME 75,pp.827/834,(1953).

[7]. T.Bartz–Beielstein K.E. Parsopoulos and M.N. Vrahatis,''Analysis of Particle Swarm Optimization using Computional Statistics",International conference on numerical analysis and applied mathematics ICNAAM-(2004).

[8]. M.A.Johnson and M.H. Moradi. PID Control. Springer Science+Business Media, 2005.

[9]. K.Valamathi, D.Devraj and T.K.Radhakrishnan,"particle swarm optimization based PI controller tuning for Fermentation process"International Conference on Computational Intelligence for Modelling, Control and Automations and Intelligent Conference and Intelligent Agents,Web Technologies and Internet Commerce.(2005).

[10]. R.C Eberhart and Y.Shi, 'Comparison between genetic algorithm and particle swarm optimization.' Proceedings. IEEE International Conference Evolutionary Computation, pp 611–616,May 1998.

[11]. BJ Parvat, et al.,'Design and Implementation of Sliding mode control for level control', In IOSR Journal of Electronics and Communication Engineering page no.2278-2834,2010.

[12]. Marco Dorigo et al., "Particle swarm optimization," Scholarpedia, vol. 3, no. 11, p. 1486, 2008.

[13]. Merlin,T.E(1995),process Control Designing Processes and Control systems for Dynamic Performance,Mc Graw-Hill.

[14]. NK Patil, BJ Parvat,'Self Tuning Mechanism using input scaling Factor of PI like Fuzzy Controller for Improved Process Performance', in International Journal of Advanced Research in Computer Science and Elcetronics Engineering(IJARCSEE), vol.2 no.11pp729-733 Nov 2013.