International Journal of Computer Applications (0975 – 8887)
Volume 43– No.9, April 2012

Optimization of I-PD Controller for a FOLIPD Model using Particle Swarm Intelligence

S.J. Suji Prasad
Assistant Professor (Sr.G.)
Electronics and Instrumentation
Kongu Engineering College, Erode

Susan Varghese
P.G Student
Electronics and Instrumentation
Kongu Engineering College, Erode

P.A. Balakrishnan
Dean (Academic and Research)
KCG College of Technology, Chennai

ABSTRACT
In this paper, I-PD controller is designed and controller parameters are optimized using particle swarm intelligence for a First Order Lag Integrating plus Time Delayed model (FOLIPD). One of the modifications of PID controller is I-PD controller, which can be used for eliminating the proportional and derivative kick occurs during set point change. The controller parameters play the major role in obtaining the desired performance of a process and that urges the importance of selecting the most suitable parameters. The simulation results show that particle swarm optimized I-PD controller gives better performance compared to traditional Ziegler Nichols tuning technique and tuning method proposed by Arvanitis.

Keywords
I-PD controller, FOLIPD model, PSO (Particle Swarm Optimization), Settling time, Rise time, and Peak overshoot.

1. INTRODUCTION
PID controllers and its modified structures provide a generic and proficient solution to real world control problems. Despite of its popularity the tuning of its parameters remains as a challenge to scholars and researchers. Its unfussiness and toughness stimulates the continuous research, developments and modification of the PID structure in order to improve the performance [1]. The three controller parameters of the PID controller determine the desired output response.

In recent years many evolutionary optimization techniques like GA (Genetic Algorithm), PSO (Particle Swarm Optimization), BF (Bacterial Foraging), ACO (Ant Colony Optimization) are proposed for optimizing the controller parameters [2][3]. In this paper Particle Swarm Intelligence is used to optimize the I-PD controller parameters for a first order lag integrating plus time delayed model.

This paper is divided in to six sections. I-PD controller structure and its importance are explained in section 2. Section 3 follows with the model used in this paper and the conventional tuning method, and Section 4 describes the concept of particle swarm intelligence. Results are discussed in section 5 and section 6 concludes the paper.

2. STRUCTURE OF I-PD CONTROLLER
I-PD controller is one of the modified forms of PID controller. In I-PD controller, the integral term is acting on the error and proportional and derivative terms are acting on the process variable, y(t). The error e(t) is the difference between the set point, r(t) and the measured process variable y(t). Here the u(t) is the output of the controller and the input to the process.

The output of I-PD controller is given by

$$u(t) = K_p\left(\frac{1}{\tau_i}\int e(t)dt - (y(t) + \tau_d\,\frac{dy(t)}{dt})\right) \quad (1)$$

where, K_p-controller proportional gain
τ_i -Integral time and
τ_d -Derivative time.

In conventional PID controller, the changes in set point cause an impulse signal or sudden change in the controller output as well as in output response [4]. This spike in the controller output is called proportional or derivative kick. The controller output is given to the final control elements like control valve, motor or electronic circuit in which the spikes create serious problem.

But in I-PD controller the proportional and derivative terms are acting only to the change in process variable not on the error as these terms are given in the feedback path. This structure may eliminate the proportional and derivative kick during any set point change [5].

The Laplace transform of equation (1) is

$$U(s) = K_p\left[\left(\frac{1}{\tau_i s}\right)E(s) - [\tau_d s + 1]Y(s)\right] \quad (2)$$

Figure 1 shows the block diagram of the I-PD controller given to a process.

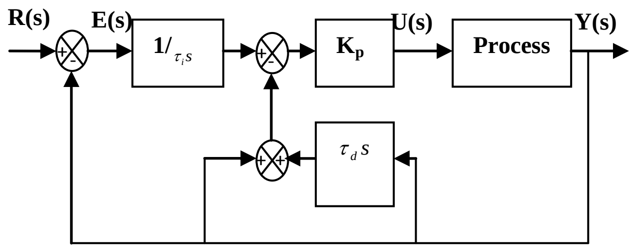

Fig 1: Block diagram of I-PD controller applied to the Process

3. FOLIPD MODEL AND TUNING
3.1 FOLIPD model
Most of the process may have time delay. First Order Lag Integrating plus Delayed model (FOLIPD) is one among them and is taken for the analysis [6].

A FOLIPD model can be represented in the following form

$$G_m(s) = \frac{K_m e^{-s\tau_m}}{S(1+T_m S)} \quad (3)$$

where K_m - gain of the process model
τ_m - time delay of the process model and

T_m - time constant of the process model.

The FOLIPD system used in this paper is

$$G_m(s) = \frac{e^{-0.358s}}{S(1+1.33S)} = \frac{e^{-0.358s}}{1.33S^2+S} \quad (4)$$

3.2 Ziegler-Nichols Tuning

ZN (Ziegler-Nichols Tuning) [7] method is commonly used for the tuning of PID controller because of its effectiveness, openness and applicability to wide range of processes. At the cross over frequency ω_c, the system will show sustained oscillations, M is the amplitude ratio of the system. The critical gain is Kcr=1/M; and the period of sustained oscillations is Pcr=2*pi/ ω_c. The PID controller parameters can be calculated by following formulas

K_p =0.6Kcr,

$\boldsymbol{\tau_i}$ =0.5Pcr and

$\boldsymbol{\tau_d}$ =0.125Pcr.

3.3 Arvanitis Tuning

For the tuning of I-PD controller, this rule uses model parameters like gain of the process, time constant, time delay constant and damping factor [6].

The controller parameters for FOLIPD model are calculated by tuning rule is given in Table 1

Table 1 Arvanitis Tuning Rule for FOLIPD Process Model

K_p	τ_i	τ_d
$\frac{\tau_i}{K_m\tau_m^2[1+8\zeta^2]}$	$\tau_m + T_m + 4\tau_m\zeta^2$	$\frac{\tau_m T_m(1+4\zeta^2)}{\tau_i}$

The values of controller parameters for the FOLIPD model using ZN method and Arvanitis tuning rule are obtained and given in Table 2

Table 2 Controller Parameters from Ziegler-Nichols and Arvanitis Tuning Rule

Controller Parameters	FOLIPD $G_m(s) = \frac{e^{-0.358s}}{S(1+1.33S)}$	
	ZN	**Arvanitis**
K_p	1.7469	5.5484
τ_i	2.2645sec	1.7804sec
τ_d	0.5661sec	0.4062sec

4. PARTICLE SWARM OPTIMIZATION

This evolutionary optimization method is based on the social sharing of information by the birds or fishes during the search for their food. The whole population of the bird or fish is called the swarm and each member in a population is called a particle. Gradually the swarm will move to optimum place of the food [8] [9].

Current position and Velocity, Location of personal best fitness and Location of Global best fitness are the functions which determine the direction of swarm's movement. The swarm moves towards optimum solution of the multi dimensional problem plane.

The major steps in PSO algorithm are initialization and updation of position and velocity. These steps can be represented by the following equations.

- Initialization

Position $x_0^i = x_{min}$+rand *(x_{max}-x_{min}) (6)

Velocity v_0^i = 0.1*randn (dim, n) (7)

where

x_{max}& x_{min}- boundary of the position

dim - dimension of the design plane

n - no of particle

- Updation of Velocity

$v_{k+1}^i = w*v_k^i + c_1$*rand( ).*($p_i$-$x_k^i$)+$c_2$*rand( ).*($p_k^g$-$x_k^i$) (8)

where

w - inertia factor (in between 0.4 & 1.4)

c_1 - self confidence of the particle

c_2 - swarm confidence (in between 1 and 2)

p_i , p_k^g - the personal best and global best

- Updation of Position

$x_{k+1}^i = x_k^i + v_{k+1}^i$ (9)

For the optimizing the I-PD controller parameters, Position of each particle represents a set of K_p, τ_i, τ_d values[10][11]. So the updation of particle's position causes the particle to move in the potential areas of the problem plane that will give minimum settling time.

PSO parameters used in this paper are given below

Size of the Swarm or no of particles	= 30
No of iterations	= 30
Dimension of the problem space	= 3
Velocity constants C_1 and C_2	= 1.5
Inertia factor	= 1

Figure 2 shows the flow chart PSO algorithm implementation. As a result of PSO algorithm optimized values of Proportional gain, Integral time and Derivative time are obtained in accordance with minimum setting time.

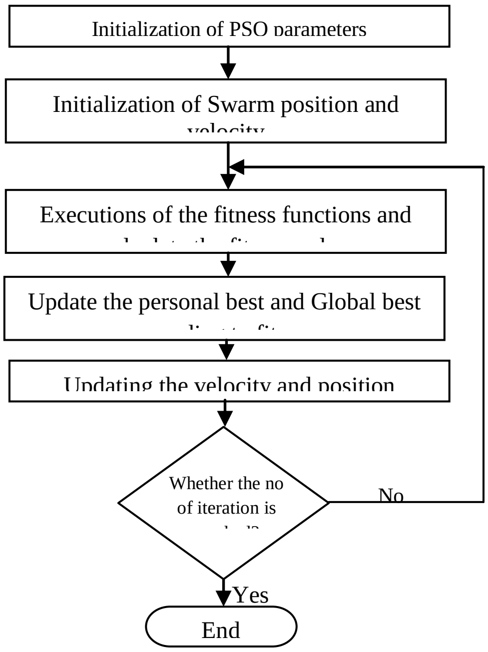

Fig 2. Flow Chart of PSO Algorithm

The minimizing function used for the optimization is the settling time with 2% tolerance. After 30 iterations the particles move to the optimum value of the controller parameters that will result in minimization of time required to reach the set point.
The Table 3 shows the results of Particle swarm optimization.

Table 3 Controller Parameters Obtained from PSO

Controller Parameters	Particle Swarm Optimization
	FOLIPD $G_m(s) = \frac{e^{-0.358s}}{S(1+1.33S)}$
K_p	4.6691
τ_i	1.7101sec
τ_d	0.6034sec

5. RESULTS AND DISCUSSION

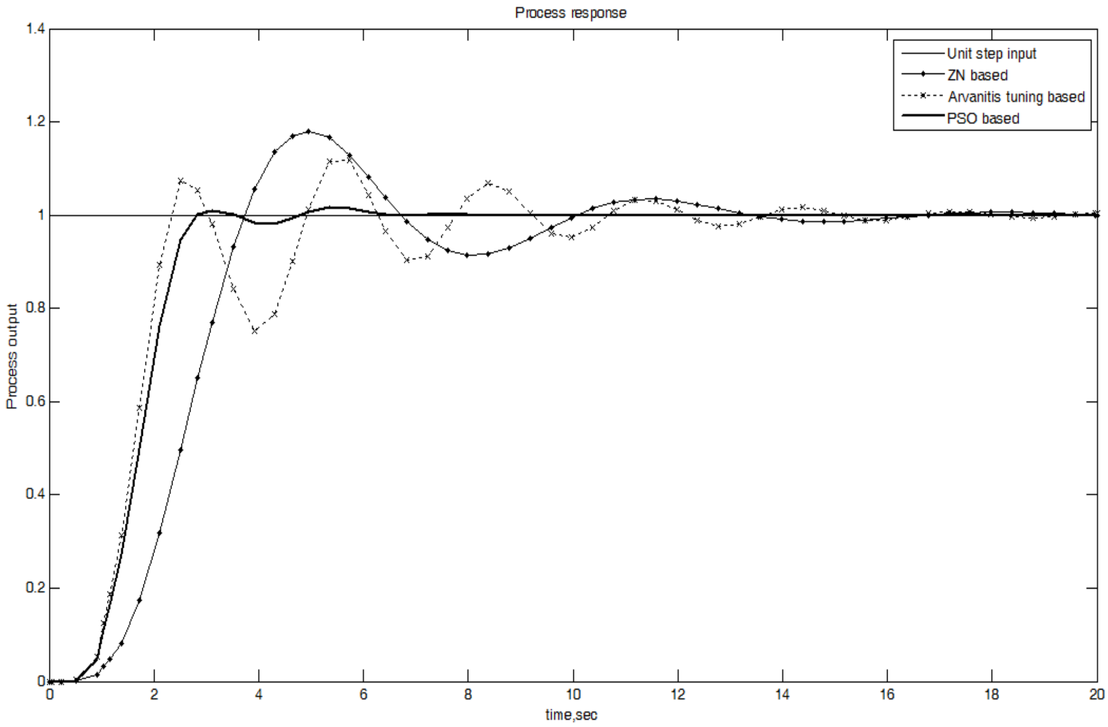

Fig 3. Output Response of the FOLIPD Model

The Figure 3 shows output response of PSO based I-PD controller for FOLIPD process model compared with ZN and Arvanitis tuning methods.

Table 4 shows the performance comparison of the response.

Table 4 Performance Comparison of the Process Models

OBSERVATION FROM THE RESPONSE OF FOLIPD PROCESS MODEL			
PERFORMANCE SPECIFICATION	ZN	Arvanitis	**PSO**
Settling time, sec	12.5	15.65	**2.78**
Rise time, sec	2.52	1.61	**1.71**
Peak overshoot %	18.52	11.59	**1.37**
IAE (Integral Absolute Error)	5.7612	6.5352	**2.6144**
ISE (Integral Square Error)	0.0508	0.0553	**0.0242**

Particle swarm optimized controller response of FOLIPD model gives 78% reduced settling time, 32% reduced rise time, 93% reduced peak overshoot, 55% reduced IAE, and 52% reduced ISE compared to ZN.

6. CONCLUSION

I-PD controller applied on a time delayed theoretical FOLIPD process model is optimized using Particle swarm intelligence. The settling time, rise time, peak overshoot, integral square error and integral absolute error in the PSO optimized process response is significantly reduced when compared to the process response using ZN and Arvanitis tuning method.

Particle Swarm Intelligence can be extended to multi objective optimization and also to optimize the parameters of I-PD controller for real time industrial systems

7. REFERENCES

[1] Astrom K.J., and T. Hagglund. “The future of PID control”, Control Engineering Practice, vol. 9, November 2001, pp. 1163–1175.

[2] Nagaraj B, P. Vijayakumar, 2011, ‘A Comparative Study Of PID Controller Tuning Using GA, EP, PSO AND ACO’, Journal of Automation, Mobile Robotics & Intelligent Systems,Volume 5, No 2 pp 42-48.

[3] Kiam Heong Ang, Gregory Chong and Yun Li (2005), ‘PID Control System Analysis, Design, and Technology’, IEEE Transactions on control systems technology, vol. 13, no. 4.

[4] Rania Hassan, Babak Cohanim and Olivier de Weck 2004, ‘A Comparison of Particle Swarm Optimization and the Genetic Algorithm’, American Institute of Aeronautics and Astronautics.

[5] Yun li, Kiam Heong Ang, and Gregory C.Y. Chong (2006), ‘PID Control System Analysis and Design - Problems, Remedies and Future Directions’, IEEE Control systems magazine, pp32-41.

[6] Aidan O'Dwyer, 2006, ‘Handbook of PI and PID Controller Tuning Rules’, (2nd Edition), Published by ICP.

[7] Zeigler, J. G., and Nichols, N.B.(1992) " Optimum Settings for Automatic Standard methods used for tuning PID controllers are hand Controllers", ASME Transaction pp 759-768, 1992.

[8] Riccardo Poli, James Kennedy and Tim Blackwell, 2007, ‘Particle swarm optimization-An overview’, Springer Science, Business Media, LLC.

[9] Russell C Eberhart and Yuhui Shi, 2001, ‘Particle Swarm Optimization: Developments, Applications and Resources’, IEEE conference.

[10] Wen-wen Cai, Li-xin Jia, Yan-bin Zhang,Nan Ni, 2010, ‘Design and Simulation of Intelligent PID Controller Based on Particle Swarm Optimization’, IEEE conferences.

[11] Giriraj Kumar S.M, Deepak Jayaraj and Anoop R Kishan 2010, ‘PSO based Tuning of a PID Controller for a High Performance Drilling Machine’ International journal of computer applications, volume I-No.19.

[12] Kiam Heong Ang, Gregory Chong and Yun Li, 2005, ‘PID Control System Analysis, Design, and Technology’, IEEE Transactions on control systems technology, vol. 13, no. 4.

[13] Tushar Jain and M. J. Nigam, 2008, ‘Optimization of PD-PI Controller Using Swarm Intelligence’, International journal of computational cognition, vol. 6, no. 4.

[14] Wael Korani ,2009, ‘Bacterial Foraging Oriented by Particle Swarm Optimization Strategy for PID Tuning’, IEEE International Symposium.