PSO Based PID Controller for Hard Disk Controller

Dheeraj Mor[a]*, Yuvraj Singh Rathore[b] and Charvi Mittal[c]
[a]*Department of EIE,* [b]*Department of EEE,* [c]*Department of ECE*
VIT University, Vellore, Tamil Nadu, India.
* mordheeraj91@gmail.com

Abstract

In today's fast growing world, there is an urgent need to increase the production rate in process industries which consists of numerous non-linear processes. Hence it is important to design a stabilizing controller. The aim of the paper is to design optimum Proportional- Integral- Derivative (PID) controller using Particle Swarm Optimization (PSO). In this paper the algorithm is applied to control the position head of the hard disk controllers using both PID and PSO-PID controllers. The performance of PID controller is compared with PSO-PID controller by calculating Overshoot value, Rise time, Settling time and Steady state error in both the cases. According to simulation and experimental results PSO-PID controller is proven to be more efficient and result also suggests that PSO converges with less number of functional evaluations.

Keywords - PSO, PID, Ziegler-Nichols Rule

Introduction

PID controllers are used in more than 95% of closed-loop industrial processes due to its simplicity and excellent if not optimal performance in many applications (Willjuice *et al.*, 2007).It can be tuned by operators without extensive background in Controls, unlike many other modern controllers that are much more complex but often provide only marginal improvement. The PID controller has three principal control effects. The proportional (P) action gives a change in the input (manipulated variable) directly proportional to the control error. The integral (I) action gives a change in the input proportional to the integrated error, and its main purpose is to eliminate steady state error. The overall controller output is the sum of the contributions from these three terms.

There have been a lot of approaches to search the parameters of PID controllers, including time response tuning (Baskar, 2007), time domain optimization (Lin *et al.*, 2003), frequency domain shaping (Hang *et al.*, 1991) and genetic algorithms (Karl and Hagglund, 1995). There are several methods for tuning of controller parameters in PID controllers such as (Voda *et al.*, 1995).

1. Ziegler-Nichols Rule
2. CHR Rule
3. Cohen-Coon Rule

Some of the methods like the Good Gain method and the Ziegler-Nichol's method are experimental and it require experiments to be made on the process to be controlled. Other method Skogestad's method is model-based, i.e. you can get good PID parameter values directly from the transfer function model of the process, without doing any experiment. Still, it requires verification that the PID tuning is proper by simulating. But all the above tuning algorithms are comparatively complicated and difficult to make the response optimized with the worse vibration and overshoot.

A lot of researches were devoted to improve the optimization of PID controllers and in this paper we propose one such method of Particle swarm optimization for the parameter search of PID controller. The results of PSO-PID are compared with various tuning algorithms of PID in this paper and it was concluded that PSO-PID shows better performance. Thus, PSO-PID algorithm must be recommended to implement the tuning of PID controller.

Algorithm

A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. Each particle is treated as a point in an N-dimensional space which adjusts its "flying" according to its own flying experience as well as the flying experience of other particles (Karl and Hagglund, 1995). The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

Formally, let f: $\mathbb{R}n \rightarrow \mathbb{R}n$ the cost function which muse be minimized. The function takes a condidate solution as argument in the form of a vector of real numbers and produces a real number as output which indicates the objective function value of the given candidate solution. The gradient of f is not known. The goal is to find a solution for which f(a) ≤ f(b) for all b in the search-space, which would mean a is the global minimum. Maximization can be performed by considering the function h = -f instead.

Let S be the number of particles in the swarm, each having a postiion $X_i \in \mathbb{R}_n$ in the search-space and a velocity $v_i \in \mathbb{R}_n$.

Let pi be the best known position of particle i and let g be the best known position of the entire swarm.

A basic PSO algorithm is then:

1. Initialize the particle's position with a uniformly distributed random vector: $x_i \sim U(b_{lo}, b_{up})$, where b_{lo} and b_{up} are the lower and upper boundaries of the search-space.
2. Initialize the particle's best known position to its initial position: $p_i \leftarrow x_i$
3. If $(f(p_i) < f(g))$ update the swarm's best known position: $g \leftarrow p_i$
4. Initialize the particle's velocity: $v_i \sim U(-|b_{up}-b_{lo}|, |b_{up}-b_{lo}|)$

Until a termination criterion is met (e.g. number of iterations performed, or a solution with adequate objective function value is found), repeat:

For each particle i = 1, ..., S do:

a) For each dimension d = 1, ..., n do:

Pick random numbers: $r_p, r_g \sim U(0,1)$

Update the particle's velocity: $v_{i,d} \leftarrow \omega\, v_{i,d} + \varphi_p\, r_p\, (p_{i,d}-x_{i,d}) + \varphi_g\, r_g\, (g_d-x_{i,d})$

b) Update the particle's position: $x_i \leftarrow x_i + v_i$

c) If $(f(x_i) < f(p_i))$ do:

Update the particle's best known position: $p_i \leftarrow x_i$

If $(f(p_i) < f(g))$ update the swarm's best known position: $g \leftarrow p_i$

Now g holds the best found solution.

The parameters ω, φ_p, and φ_g are selected by the practitioner and control the behavior and efficacy of the PSO method.

Steps Involved

Following steps were involved in this methodology:-

1. Implied a Ziegler Nicholas criterion on the given transfer function to find out the tuning parameters of the PID algorithm.
2. Then find out the timing parameters of the system.
3. Since the tuning parameters found out by the Ziegler Nicholas criterion does not provide the optimized solution, so we applied PSO algorithm on the PID with the range of in the values of the kp, ki and kd.
4. Then the timing parameters where calculated for each iterations and the list of the parameters calculated are:
   a. Overshoot value
   b. Rise Time
   c. Settling time
   d. Steady State Error
5. Calculate the evaluation value of each individual in the population using the evaluation function

$$f=1/W(K)$$

where,

$$\min_{K:\ stabilizing} W(K)=(1-\exp(-\beta)).(M_p+E_{ss})+\exp(-\beta).(t_s-t_r)$$

6. Compare each individual's evaluation value with its pbest. The best evaluation value among the pbest is denoted as gbest.
7. Velocity Function as

$$v_{j,g}^{(t+1)}=w.v_j^{(t)}+c_1*rand()*(pbest_{j,g}\ \ k_{j,g}^{(t)})+c_2*Rand()*(gbest_g\ \ k_{j,g}^{(t)})$$

j=1,2,3….

g=1,2,3….

where

$$w=w_{max}\ \ ((w_{max}-w_{min})/iter_{max}).\ iter$$

where

n	number of particles in a group
m	number of members in a particle
t	pointer of iterations (generations)
$v_{j,g}^{(t)}$	velocity of particle j at iteration t,

$$V_g^{min} <= v_{j,g}^{(t)} <= V_g^{max}$$

w	inertia weight factor
c1, c2	acceleration constant

rand(), Rand()	random number between 0 and 1
$V_{j,g}^{(t)}$	current position of particle j at iteration
$pbest_j$	pbest of particle j
gbest	gbest of the group

8. If $v_{j,g}^{(t+1)} > V_g^{max}$, then $v_{j,g}^{(t+1)} = V_g^{max}$

   $v_{j,g}^{(t+1)} < V_g^{min}$, then $v_{j,g}^{(t+1)} = V_g^{min}$

9. Modify the member position of individual K

$$k_{j,g}^{(t+1)} = k_{j,g}^{(t)} + v_{j,g}^{(t+1)},$$

$$k_g^{min} \leq k_{j,g}^{(t+1)} \leq k_g^{max}$$

10. When the maximum number of iterations where reached the individual that generated the gbest was considered as the optimal solution and all the results were recorded.

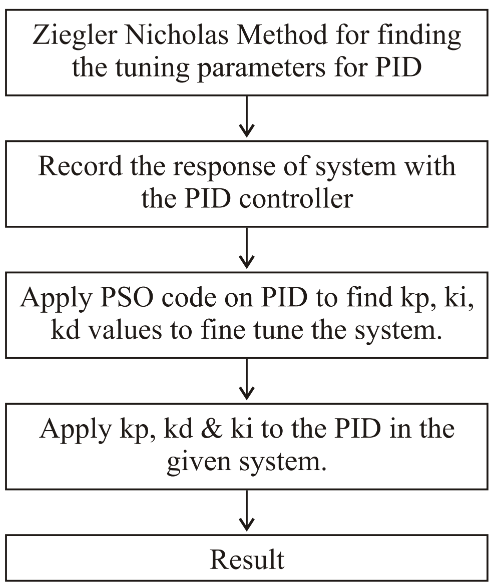

Figure 1: Block Diagram of the Approach used to solve this problem.

Problem Statement

Transfer Function of the position hand of Hard Disk Controller (shown below)

$$G(s) = \frac{(K * e(-theta * s))}{(T * s + 1)}$$

where K=14.9, theta=80, T=360

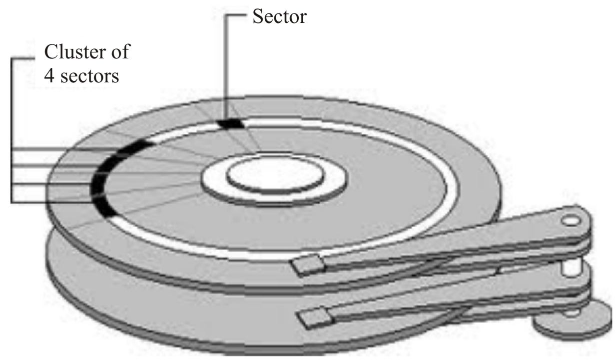

Figure 2: Disk Controller

Simulations

1. kpmin=0.4; kpmax=0.6
2. kimin=0.04; kimax=0.06
3. kdmin=2.97; kdmax=3.1
4. population size = 50
5. inertia weight factor wmin=0.4 and wmax=0.9

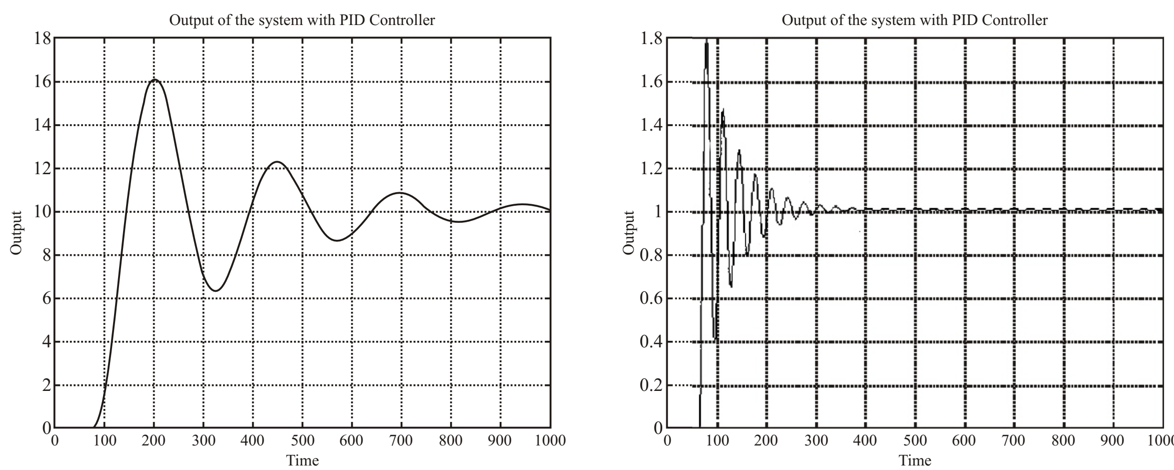

Figure 3: Case 1:- PID Algorithm result, Case II: - PSO-PID Algorithm result.

Table 1: For Case I and Case II.

Timing Parameter	Case I (PID)	Case II (PSO-PID)	Dominating Case
Rise Time	44.938011064	5.511056672	II
Settling Time	9.658126e+02	3.07139e+02	II
Overshoot	60.140771001	76.93830294	1
Peak	16.095665946	1.769383131	II
Peak Time	202	95	II

Conclusion

From the above we can observe clearly that in case II, Rise Time, Settling Time, Peak and Peak Time are much more improved than the case I. Though the overshoot value of the case I is less than that case II but by the various values changes it is clear that the PSO applied PID gives a much better controlling results then the normal PID controlling. These results help us to reduce the irregularities presented by the hard disk controller and thus it can be controlled in an efficient manner. It will also help to have an optimal control of the various control systems.

Acknowledgement

This work would not have been possible without the help of Prof. Jagannath Pandian. So, we would like to thank him for helping us throughout this research.

References

Baskar, S. 2007. Application of Evolutionary Computation to Engineering Optimization Problems. *Lecture Notes, Short term course on EMOA*.

Hang, CC. K. J., Astrom, W. K. Ho. 1991. Refinements of the Ziegler-Nichols tuning formula. *IEE Proc. Part D*, 138(2), 11- 18.

Karl J. Astrom, Tore Hagglund. 1995. PID Controllers: Theory, Design, and Tuning. *International Society for Measurement and Con*.

Kiam Heong Ang, Gregory Chang, Yun Li. 2005. PID control system Analysis, Design and Technology. *IEEE transaction on Control System Technology*, 13(4), 559-577.

Lin, C.L., Jan, H.Y., Shieh, N.C. 2003. GA-Based Multiobjective PID Control fuor a Linear Brushless DC Motor. *IEEUASME Trans. Mechatronics*, 8 (1), 56-65.

Voda, A. A., Landaq, I. D. 1995. A method of the auto calibration of PID controllers. *Autonmlica,* 31(1), 41-53.

Willjuice Iruthayarajan, M., Baskar., S. 2007. *IET-UK International Conference on Information and Communication Technology in Electrical Sciences.*, 81-86.

Zhuang, M., Atherton, D. P. 1994. Optimum Cascade PID Controller Design for SISO Systems. *IEEE Control*, 1, 606-611.