© 2018 IJSRST | Volume 5 | Issue 3 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

Tuning and Analysis of PID Controllers Using Soft Computing Techniques

Bajarangbali, Sujay H S, Suman R, Chaithanya S, Narayanan S, Shamanth U
Department of Electronics and Communication, PESIT-Bangalore South CampusHosur Road, Bangalore, Karnataka, India

ABSTRACT

In this paper, we are looking at a method to tune a PID controller given that we know the system transfer function beforehand. Here we will be tuning the controller using soft computing methods which involve Particle Swarm Optimization (PSO) and Genetic Algorithms (GA). These techniques like PSO and GA are very well known to find the best global minima. Thus, such a feature of these techniques is used to minimize the errors such as IAE (Integral Absolute Error), ITAE (Integral Time Absolute Error) and ISE (Integral Squared Error). By reducing the errors, the best value for the three PID parameters, K_p, K_i and K_d are determined. The PID controller will be tuned for linear processes with and without time delay.
Keywords: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), IAE (Integral Absolute Error), ITAE (Integral Time Absolute Error), ISE (Integral Squared Error).

I. INTRODUCTION

A PID (Proportional Integral Derivative) controller is a feedback type of controller which is used in many industrial applications. The PID controller continuously calculates the error and applies a control signal based on the Proportional, Integral and Derivative Gains and hence the name "PID Controllers".

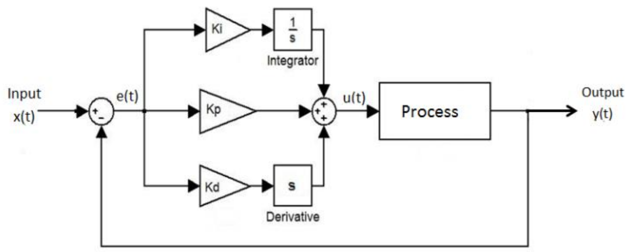

Figure 1. PID Controller Closed Loop System

The PID constitutes three critical parameters, proportional, integral and derivative gain denoted by K_p K_i and K_d respectively. In brief, effect of each of the coefficient is as explained below:

Proportional Parameter (K_p): The proportional gain term is directly proportional to the present error produced by the system. Here, the error is the deviation of the output from the set-point value. It is like the ratio of the output value to the error value. Increasing this increases oscillations which will never settle, but makes the system response much faster.

Integral Parameter (K_i): The integral gain parameter calculates the error over a certain period of time. Thus, the output produced by the I term is not only dependent on the present output but also on the past output values. This helps greatly in eliminating steady state error.

Derivative Parameter (K_d): The derivative gain term comes into action when the rate of change of error is large. Thus, the D term is regarded to see the future errors and reduce them. Increasing the derivative term makes the controller more aggressive and hence must be used with caution.

All the three terms, i.e., proportional, integral and derivative terms are summed to compute the output of the PID controller. If, u(t) is regarded as the controller output, the PID controller can be represented mathematically as:

$$u(t) = K_p.e(t) + K_i \int_0^{\infty} e(\tau)d\tau + K_d.\frac{de(t)}{dt} \qquad (1)$$

In the Laplacian domain, the PID controller is represented as:

$$L(s) = K_p + \frac{K_i}{s} + K_d.s \qquad (2)$$

System Performance Assessment

Academic measures are ones that are calculated on paper based on the expected response from the system. The three measures are IAE (Integral Absolute Error), ITAE (Integral Time Absolute Error) and the ISE (Integral Squared Error).
Mathematically, they are given as:

$$ISE = \int e^2(t)dt \qquad (3)$$

$$IAE = \int |e(t)|dt \qquad (4)$$

$$ITAE = \int t.e(t)dt \qquad (5)$$

ISE - This error type calculates the integral over the square of error in time. It is found to help in correcting sufficiently large errors that are present for a short period of time, rather than those small errors which can last either for short or long period of time. Hence quick responses can be achieved, but small percentage of steady state oscillations will be present.
IAE - Similar to ISE, it only integrates the absolute value of error over time instead of the square of error value. It is helpful in reducing constant errors or small persistent oscillations. It does however produce slower responses.
ITAE - This error calculates the integral of the time weighted absolute value of the error over time. It can be definitely more tedious to calculate, but is often used as it produces the best results in most cases. It eliminates errors that are persistent for large period of time as it weights heavily for those errors at a later period of time than those at the beginning. But, initial response is found to be quite sluggish which are quite essential to reduce the sustained oscillations.

II. IMPLEMENTATION

A. Soft Computing

Soft computing, in computer science is a method to find out the inexact solution to really hard problems. These solutions are generally an approximation that include uncertainty and are partially true.

Soft computing techniques include Evolutionary Computation (EC), Fuzzy Logic (FL), Probabilistic Reasoning (PR) and Machine Learning (ML).

In our study, we have chosen Evolutionary Computation. Evolutionary Computation involves algorithms that are inspired biologically, i.e., we have 'tried' to copy the way humans or living organisms interact with nature and one other. The copy is undoubtedly not accurate as a lot of research is still going on in the current field and only about 20% of the actual reason for human evolution, interaction with nature and one another could probably be justified. These algorithms can also be considered as a sub-field of artificial intelligence.
The algorithms we have considered under Evolutionary Computation are:

1. Genetic Algorithm
2. Particle Swarm Optimization

These algorithms look to solve many minimization problems (can also be maximization). They are believed to be really good at finding the global minima (or maxima) rather than ball parking in and around local minima's (maxima's). Thus they are employed where there is no known algorithm that can compute the exact solution in polynomial time. These algorithms initially start with random initialization of population (in case of genetic algorithm) or swarm (in case of Particle Swarm Optimization). Then the algorithm is repeated iterative steps until the best possible solution is found.

Thus to apply these algorithms to our Tuning of PID controller, we must thus define an "OBJECTIVE FUNCTION" which has to be minimized.

We have already discussed the four different types of errors (ISE, IAE, ITSE, ITAE). So, now we must define when and where which error must be used as the objective function.

B. Error Selection

To apply evolutionary computation for the tuning of PID controllers, the system transfer function must be known. Since the system is completely known before we begin our tuning process, we have to analyse the system. An analysis must be done such that it leads us to the correct selection of error definition to be used as our objective function.

In case the system response shows a large deviation from the set point, ISE criteria should be used because squaring the error term contributes more to the cost which eventually drags the optimization algorithm towards a set of controller parameters that ensures minimization of that cost. In case of small deviation errors, squaring the term would actually reduce its contribution to the cost. Hence IAE criterion is used for such cases. When the error persists for a long time, ITAE criterion helps because the presence of time as a multiplier to the error term actually augments its effect on the cost term at high values. Generally ITAE criterion is not used because time is not under anyone's control and squaring of time as it increases only shows larger errors and affects the objective (cost) function negatively.

To get a quantitative measure of the objective function, the following steps are followed:

1. For the given open loop transfer function, step response is plotted.
2. The step response is analysed.
3. Based on the above mentioned criteria, a suitable cost function is determined, i.e. either ISE or IAE or ITAE.
4. The selected cost function is minimized and thus the controller parameters are tuned.

Once, we have finalized the objective function, we run the algorithm and verify our results.

C. Genetic Algorithm

The Genetic Algorithm procedure is as follows:

Step 1: [Start] A random population of chromosomes are generated which represents the total number of solutions that are required for the problem.

Step 2: [Fitness] The fitness of each of the randomly created chromosome in the population is evaluated.

Step 3: [New population] A new population is created by repeating following steps till the new population is complete:

- [Selection] Select any two parent chromosomes from a population based on their fitness. It is generally taken as better the fitness, bigger is the chance of a chromosome to be selected as the parent.
- [Crossover] With a crossover probability rate of 4, crossover the parents to form new offspring i.e. children. If in case, no crossover is performed, offspring is taken as the exact copy of its parents.
- [Mutation] With a mutation probability of 8, mutate the new offspring at each of the locus.
- [Reproduction] Place newly created offspring in the new population.

Step 4: [Replace] Use new generated population in the previous step for a further run of the algorithm.

Step 5: [Test] If the end condition has been satisfied, stop and return the best obtained solution in the current population.

Step 6: [Loop] Go back to step 2. The generation is repeated for 100 iterations.

D. Particle Swarm Optimization

The Particle Swarm Optimization procedure is as follows:

Step 1: Initially, a random number of particles (agents) that belong to a swarm moving around in a search space looking for the best solution is setup.

Step 2: Each particle is then treated as a point in a N-dimensional vector space which will adjust its flying based on its own flying experience as well as the flying experience of the other agents.

Step 3: [pbest] Each agent always keeps track of its coordinates in the solution space which are associated with the best solution that has been achieved so far by that agent. The value of the best solution is called personal best, **pbest**.

Step 4: [gbest] Another best value that is kept track by the PSO is the best value obtained so far by any particle in the neighbourhood of that particle. This value is called **gbest**.

Step 5: PSO accelerates each and every agent toward its pbest and gbest locations hence finding the optimum solution.

III. RESULTS AND COMPARISON

Example 1

The following transfer function is considered from the paper-N A Rahman[3].

$$G(s) = \frac{-0.365s + 0.657}{s^2 + 1.642s + 1.982} \qquad (6)$$

In the paper, the PID parameter values were K_p =2.2324, K_i = 2.4070, K_d = 1.0736. Tuning the controller based on our proposed method, the values of K_p, K_i and K_d were respectively found to be 3.3856, 4.0761, 2.0501.

The step response of both the results are:

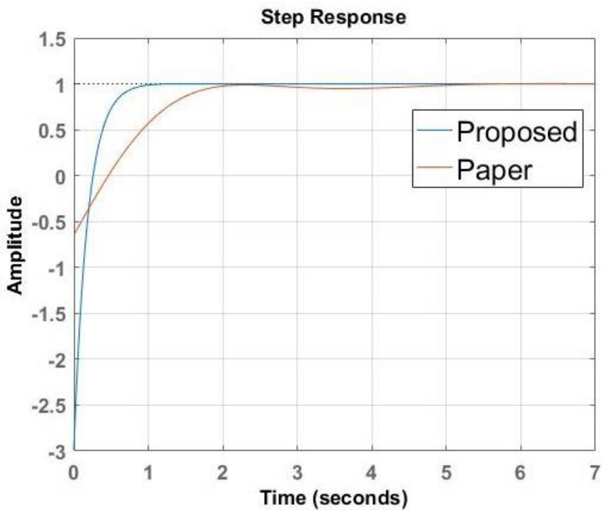

Figure 2. Example 1

Table 2: Comparison 1

	Type	ISE	IAE	ITAE	t_r	OS	t_s
Paper	GA	0.042	0.044	0.013	1.31	0.183	4.4
Prop.	PSO	0.080	0.031	0.004	0.413	NA	0.7

Here, we can see that the rise time has a significant decrease of 68%, the settling time of 84% and 0 overshoot compared to 0.183.

Example 2

The following transfer function is considered from the paper-RuchiJain[5].

$$G(s) = \frac{1.91 \times 10^6}{s^2 + 666.7s + 1.948 \times 10^6} \qquad (7)$$

In the paper, the PID parameter values K_p, K_i and K_d were 14.9, 29.93 and 29.96 respectively. Tuning the controller based on our proposed method, the values of K_p, K_i and K_d were respectively found to be 98.4, 99 and 97.6.

The step response of both the results are:

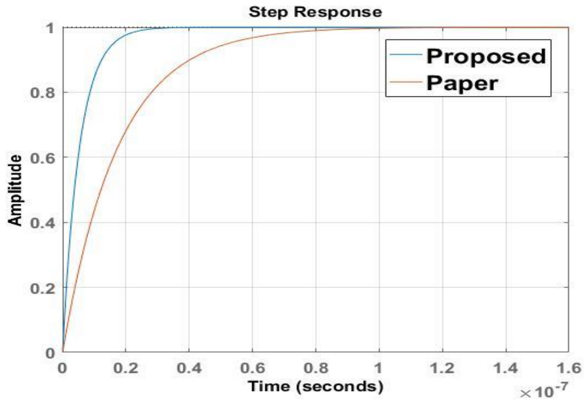

Figure 3. Example 2

Table 2. Comparison 2

	Type	ISE	IAE	ITAE	t_r	OS	t_s
Paper	PSO	0.001	0.008	0.023	3.84	0	6.84
Prop.	PSO	0.001	0.002	0.003	1.18	0	2.1

It can be seen from the results that an improvement of 69% in rise time and settling time is achieved in the proposed method.

IV. CONCLUSION

In this paper, we looked at improving the performance of the PID controller using modern techniques such as Machine Learning's subclass which include the Genetic Algorithm and Particle Swarm Optimization. In the results that we obtained, we can say with the right choice of error selection, a much better PID controller tuning can be done. Both the academic measures such as ISE, IAE and ITAE as well as the practical measures such as settling time, rise time and peak overshoot prove that the PID controller tuned using error minimization criteria proposed in this paper, yield good results.

V. REFERENCES

[1]. Cheng-Ching Yu; Autotuning of PID Controllers. A Relay Feedback Approach 2nd Edition; Published by Springer-Verlag London Limited 2006.

[2]. B.Nagaraj, S.Subha, B.Rampriya; Tuning Algorithms for PID Controller Using Soft Computing Techniques; IJCSNS International Journal of Computer Science and Network Security; VOL.8 No.4, April 2008.

[3]. Tengku Ahmad Faris Ku Yusoff, Mohd Farid Atan, Nazeri Abdul Rahman,Shanti Faridah Salleh and Noraziah Abdul Wahab; Optimization of PID Tuning Using Genetic Algorithm; Journal of Applied Science and Process Engineering;Vol. 2, No. 2, 2015.

[4]. Ahmad Ali and Somanath Majhi; Design of Optimum PID Controller by Bacterial Foraging Strategy; IEEE International Conference on Industrial Technology, 601 605, 2006.

[5]. Ruchi V. Jain, M.V. Aware, A.S. Junghare; Tuning of Fractional Order PID controller using particle swarm optimization technique for DC motor speed control; 1st IEEE International Conference on Power Electronics. Intelligent Control and Energy Systems (ICPEICES-2016).

[6]. Greg Birky, Thomas Mc Avoy, Bjorn Tyreus; DicodeAn Expert System for Designing Distillation Column Controls; IEEE, American Control Conference,1989.

[7]. Deepyaman Maiti, Ayan Acharya, Mithun Chakraborty, Amit Konar; Tuning PID and PID Controllers using the Integral Time Absolute Error Criterion; 4th International Conference on Information and Automation for Sustainability, IEEE 2008.

[8]. A. Visioli; Tuning of PID controllers with fuzzy logic; IEEE Proceedings - Control Theory and Applications, 2001.

[9]. J.G. Ziegler, N.B. Nichols; Optimum settings for Automatic Controller