IJESM
Consulting, help, relaxation
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES
&
MANAGEMENT

A COMPARATIVE STUDY OF TUNING PID BY DIFFERENT METHODS

Jai Karan Singh[*1], Mukesh Tiwari[2], Dinesh Lodwal[3]
Department of E&C, Sri Satya Sai Institute of Science and Technology, Sehore, India

Abstract

Abstract— A comparative study of optimal control of a transfer function is studied in this paper utilizing the auto tuning concept in conjunction with PID controller. A control scheme composing an auto tuning technique based Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and fuzzy logic is proposed.

Key Words— PID controller, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), fuzzy logic.

Introduction

A PROPORTIONAL–INTEGRAL–DERIVATIVE CONTROLLER (PID CONTROLLER) is a generic control loop feedback mechanism (controller) widely used in industrial control systems – a PID is the most commonly used feedback controller. A PID controller calculates an "error" value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process control inputs.

The PID controller calculation (algorithm) involves three separate constant parameters, and is accordingly sometimes called three term control: the proportional, the integral and derivative values, denoted P, I, and D. Heuristically, these values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change.[1] The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve, or the power supplied to a heating element.

*** Corresponding Author**
E. mail: jksingh81@yahoo.co.in

In the absence of knowledge of the underlying process, a PID controller has historically been considered to be the best controller.[2] By tuning the three parameters in the PID controller algorithm, the controller can provide control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the set point and the degree of system oscillation. Note that the use of the PID algorithm for control does not guarantee optimal control of the system or system stability.

Some applications may require using only one or two actions to provide the appropriate system control. This is achieved by setting the other parameters to zero. A PID controller will be called a PI, PD, P or I controller in the absence of the respective control actions. PI controllers are fairly common, since derivative action is sensitive to measurement noise, whereas the absence of an integral term may prevent the system from reaching its target vale due to the control action.

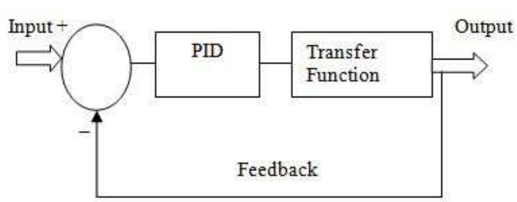

Fig.1 Control strategy by PID

Control Scheme

A. Genetic Algorithm

In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

B. PSO

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. 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.

C. Fuzzy Logics

Fuzzy logic is a form of many-valued logic or probabilistic logic; it deals with reasoning that is approximate rather than fixed and exact. In contrast with traditional logic theory, where binary sets have two-valued logic: true or false, fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false.[1] Furthermore, when linguistic variables are used, these degrees may be managed by specific functions.

Experimental Setup

MATLAB/ SIMULINK has used to perform the proposed experiment. The transfer function

$$\frac{X(s)}{Y(S)} = \frac{1}{s^4 + 6s^3 + 8s^2 + 6s + 1}$$

Is taken and the disturbance is controlled by PID. The Genetic algorithm Optimization toolbox, Particle Swarm Optimization toolbox and Fuzzy Logic toolbox are applied to find the respective values of Kp, Ki and Kd .

Results

Analysis shows that the design of proposed controller gives a better robustness, and, the performance is satisfactory over a wide range of process operations. Simulation results show performance improvement in time domain specifications for a step response. Using the proposed approaches, global and local solutions could be simultaneously found for better tuning of the controller parameters.

Controller Type	GA	PSO	Fuzzy Logic
Rise Time	1.2954	3.2320	1.6861
Overshoot	10.8042	19.9980	31.9026
Settling Time	0.7955	46.1964	29.3538
Peak Time	2.7000	7.7000	5.5000

Table 1: Obtained Parameters

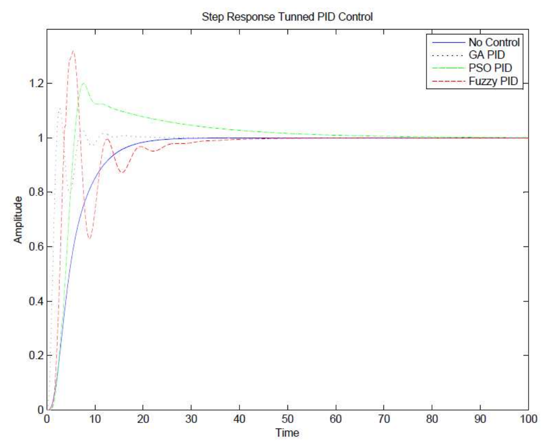

Fig. 2: Step response of the system with proposed methods

Conclusion

This paper presents a novel design method for determining the PID controller parameters using the PSO,GA and FUZZY methods. The proposed method integrates the algorithms with the new time-domain performance criterion into a GA-PID, PSO-PID or FUZZY-PID controller. Through the simulation, the results show that the proposed controller can perform an efficient search to obtain optimal PID controller parameter that achieve better performance criterion that are rise time, settling time, overshoot and steady state error condition.

References

1. Araki M. "PID Control".
2. Bennett, Stuart (1993). A history of control engineering, 1930-1955. IET. p. p. 48. ISBN 978-0-86341-299-8.
3. Chris Houck, Jeff Joines, and Mike Kay, "A Genetic Algorithm for Function Optimization: A Matlab Implementation" NCSU-IE TR 95-09, 1995.
4. A. Varsek, T. Urbacic and B. Filipic, .Genetic Algorithms in Controller Design and Tuning., IEEE Trans. Sys.Man and Cyber, Vol. 23, No. 5, pp1330-1339,1993.
5. Chen, M. and Linkens, D.A. (1998). A hybrid neuro-fuzzy PID controller. Fuzzy Sets and Systems, Vol.99, 27-36.
6. Chang, W.-D., Hwang, R.-C. and Hsieh, J. –G. (2003). A multivariable on-line adaptive PID controller using autotuning neurons. Engineering Applications of Artificial Intelligence, Vol.16, 57-63.