A Review Paper On "Particle Swarm Optimization and Genetic Algorithm Techniques For System Stability"

Harsha Yadu[1] Doordarshi Yadu[2] Chaman Yadav[3]

[1]*Department of Electronics and Telecommunication(IC & T)*
Bhilai Institute of Technology
Durg, India

[23]*Electronics and Tele communication Department*
Chhatrapati Shivaji Institute of Technology Durg, India

Abstract—*This paper deals with optimal tuning of a Proportional-Integral- Derivative (PID) controller used for system stabiltiy for controlling the output obtained. The main objective is to obtain a stable, robust and controlled system by tuning the PID controller using Particle Swarm Optimization (PSO) algorithm and Genetic Algorithm.*

Keywords— PSO Particle Swarm Optimization; GA Genetic Algorithm; PID Proportional Integral Derivative;

I. INTRODUCTION

In our daily life control systems play an important role. Due to the fault of some functions the system become unstable. So to maintain stability PID controller is used which gives optimized result. Recently, genetic algorithms (GA) and particle swarm optimization (PSO) technique have attracted considerable attention among various modern heuristic optimization techniques. The GA has been popular in academia and the industry mainly because of its intuitiveness, ease of implementation, and the ability to effectively solve highly non-linear, mixed integer optimization problems that are typical of complex engineering systems. PSO technique is a relatively recent heuristic search method whose mechanics are inspired by the swarming or collaborative behavior of biological populations. Since the two approaches are supposed to find a solution to a given objective function but employ different strategies and computational effort, it is appropriate to compare their performance. This paper presents the application and performance comparison of PSO and GA optimization techniques with the help of PID controller. The design objective is to enhance system stability

II. GENETIC ALGORITHM

Artificial intelligent techniques have come to be the most widely used tool for solving many optimization problems. Genetic Algorithm (GA) is a relatively new approach of optimum searching, becoming increasing popular in science and engineering disciplines. The basic principles of GA were first proposed by Holland, it is inspired by the mechanism of natural selection where stronger individuals would likely be the winners in a competing environment. In this approach, the variables are represented as genes on a chromosome. Gas features a group of candidate solutions (population) on the response surface. Through natural selection and genetic operators, mutation and crossover, chromosomes with better fitness are found. Natural selection guarantees the recombination operator, the GA combines genes from two parent chromosomes to form two chromosomes (children) that have a high probability of having better fitness that their parents . Mutation allows new area of the response surface to be explored. In this paper, a GA process is used to find the optimum tuning of the PID controller, by forming random of population of 50 real numbers double precision chromosomes is created representing the solution space for the PID controller parameters (KP, KI and KD), which represent the genes of chromosomes. The GA proceeds to find the optimal solution through several generations, the mutation function is the adaptive feasible, and the crossover function is the scattered**.**

A. GENETIC ALGORITHM OPERATION

To illustrate the working process of genetic algorithm, the steps to realise a basic GA are listed :

Step 1: Represent the problem variable domain as a chromosome of fixed length; choose the size of the chromosome population N, the crossover probability Pc and the mutation probability Pm.

Step 2: Define a fitness function to measure the performance of an individual chromosome in the problem domain. The fitness function establishes the basis for selecting chromosomes that will be mated during reproduction.

Step 3: Randomly generate an initial population of size N: *x1*, *x2* ,..., *xN*

Step 4: Calculate the fitness of each individual chromosome: f(x1), f(x2)…… f(xN)

Step 5: Select a pair of chromosomes for mating from the current population. Parent chromosomes are selected with a probability related to their fitness. High fit chromosomes have a higher probability of being selected for mating than less fit chromosomes.

Step 6: Create a pair of offspring chromosomes by applying the genetic operators.
Step 7: Place the created offspring chromosomes in the new population.
Step 8: Repeat Step 5 until the size of the new population equals that of initial
population, N.
Step 9: Replace the initial (parent) chromosome population with the new (offspring) population.
Step 10: Go to Step 4, and repeat the process until the termination criterion is satisfied.

A GA is an iterative process. Each iteration is called a generation. A typical number of generations for a simple GA can range from 50 to over 500. A common practice is to terminate a GA after a specified number of generations and then examine the best chromosomes in the population. If no satisfactory solution is found, then the GA is restarted.

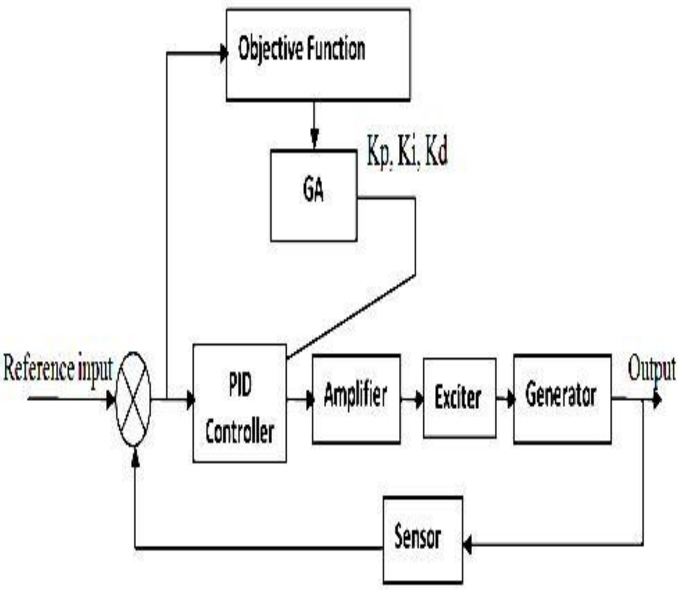
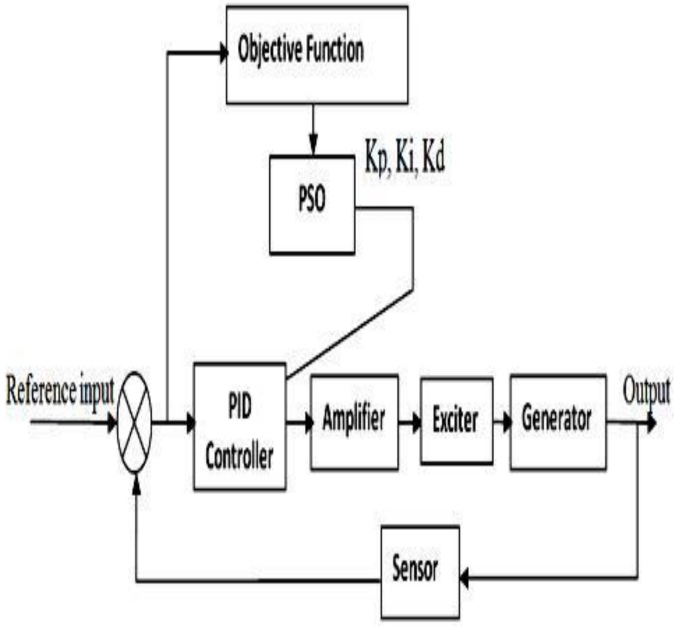


Fig : Block-diagram of AVR system with GA-PID controller

III. PARTICLE SWARM OPTIMISATION ALGORITHM

PSO optimizes an objective function by undertaking a population-based search. The population consists of potential solutions, named particles, which are a metaphor of birds
in flocks. These particles are randomly initialised and freely fly across the multidimensional search space. During flight, each particle updates its own velocity and position based on the best experience of its own and the entire population. The updating policy drives the particle swarm to move toward the region with the higher objective function value, and eventually all particles will gather around the point with the highest objective value.



IV. PID CONTROLLER

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

A. PID Controller Model

The transfer function of PID controller is

$$G_c(s) = k_p + k_d s + \frac{k_i}{s}$$

Where *kp*, *kd*, and *ki* are the proportion coefficient, differential coefficient, and integral coefficient, respectively. The derivative controller adds a finite zero to the open-loop plant

B	N0. Of generations	Type of controller	K_p	K_i	K_d	Mp(%)	t_s	t_r	Ess	Evaluation value
0.5	20	**GA-PID**	0.4370	0.2353	0.2391	0.0165	2.0316	0.7267	0	0.9387
		PSO-PID	0.4315	0.2356	0.1557	2.4064	1.7284	0.6801	0	1.0868
0.5	50	**GA-PID**	0.5395	0.2645	0.2466	0	1.2828	0.5225	0	1.3952
		PSO-PID	0.5846	0.2987	0.2438	0.4566	0.7555	0.4762	0	1.3837
0.5	100	**GA-PID**	0.3002	0.1682	0.1524	0.2869	2.3165	1.1758	0	1.4044
		PSO-PID	0.4497	0.2370	0.2636	0.2100	2.2400	0.7258	0	1.8399
0.5	150	**GA-PID**	0.6233	0.3728	0.2820	0.7377	0.6696	0.4207	0	1.3996
		PSO-PID	0.5374	0.3096	0.2572	0.1548	0.9808	0.5076	0	1.8398
1.0	20	**GA-PID**	0.3371	0.1656	0.2489	0.3370	3.4780	1.6281	0	0.7054
		PSO-PID	0.1035	0.0667	0.0228	0	4.0772	2.4701	0	0.7511
1.0	50	**GA-PID**	0.4373	0.2164	0.1865	0	1.3584	0.6969	0	1.4815
		PSO-PID	0.6938	0.3780	0.3216	0.0846	0.5772	0.3653	0	0.7689
1.0	100	**GA-PID**	0.6235	0.3212	0.2865	0	0.7266	0.4242	0	1.4847
		PSO-PID	0.3741	0.1864	0.1367	0	1.2991	0.8182	0	0.7600
1.0	150	**GA-PID**	0.5684	0.3148	0.2579	0	0.8192	0.4786	0	1.4860
		PSO-PID	0.5959	0.2745	0.2998	0	2.1816	0.4407	0	0.7678

transfer function and improves the transient response. The integral controller adds a pole at the origin, thus increasing system type by one and reducing the steady-state error due to a step function to zero. Range of PID controller parameters are shown in table .

TABLE I: RANGE OF THREE CONTROLLER PARAMETERS

Controller parameters	Min. value	Max. value
Kp	0	1.5
Ki	0	1
Kd	0	1

1) *For author/s of only one affiliation (Heading 3):* To are prescribed.

V. COMPARISON

A. *Tuning of PID Using GA-Based Optimization*

A genetic algorithm (GA) is a local search technique used to find approximate solutions to optimization and search problems. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination).Genetic algorithms are typically implemented as a computer simulation, in which a population of abstract representations (called chromosomes) of candidate solutions (called individuals) to an optimization problem evolves toward better solutions. The evolution starts from a population of completely random individuals and occurs in generations. In each generation, the fitness of the whole population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (mutated or recombined) to form a new population. The new population is then used in the next iteration of the algorithm.

B. *Tuning of PID Using PSO-Based Optimization*

The basic Particle Swarm Optimization (PSO) was developed by researching on swarm behaviour as fish schooling and bird flocking which has been proved an evolutionary optimization algorithm.PSO utilizes the individuals called as particles which get “evolved” by two processes such as cooperation and competition among themselves through generations. A particle denotes a particular solution to a problem that adjusts its flying manner according to its own experience. Each particle is assumed as a position in a D-dimensional space.Over the number of iterations, a group of variables have their value adjusted closer to the particle who is nearest to the food source. Imagine a flock of birds circling hovering over an area where they can smell food. The bird that is closest to the food source chirps loudest and other birds swing along in its direction.If any other bird comes closest to the food source ,it chirps even louder letting other birds know the location and the flock veer towards that direction.This orbital pattern of the birds continue until one of the bird finds the food system, the results show that the proposed controller can perform an efficient search for the optimal PID controller parameters. In addition, in order to verify it being superior to the GA method, many performance estimation schemes are performed, such as multiple simulation examples for their terminal voltage step responses; convergence characteristic of the best evaluation value; dynamic convergence behavior of all individuals in population during the evolutionary processing; Computation efficiency. The amount of overshoot for the output response was successfully decreased using the controller.

I. RESULT AND CONCLUSION

From this table it represents the better performance of PSO-PID as compared to GA-PID technique. The no. of generation is increased the performance is increased in both methods. It is clear from the results that the proposed PSO method can avoid the shortcoming of premature convergence of GA method and can obtain higher quality solution with better computation efficiency. The proposed method integrates the PSO algorithm with the new time-domain performance criterion into a PSO-PID controller. Through the simulation of a practical AVR

REFERENCES

As the number of iterations (generations) in PSO Algorithm and also the no. of generations in GA went on increasing the perform ance of the system also went on improving. The performance char acteristics of the PID controller by using PSO Algorithm give the better results as compared to Genetic Algorithm.

[1] Kennedy, J. and R.C. Eberhart. 2001. Swarm Intelligence. Morgan Kaufm
an Publishers.

[2] Kennedy, J. and R.C. Eberhart. 1995. Particle Swarm Optimisation. Proceedings IEEE International Conference on Neural Networks, IV, p. 1942-1948.

[3] Eberhart, R.C., Simpson, P. and Dobbins, R. 1996. Computational Intellige
nce PC Tools. Academic Press.

[4] Koza, J. R. 1991. Genetic evolution and co-evolution of computer programs.
In Langton, C. G. C. Taylor, J. D. Farmer and S. Rasmussen (Eds) *Artificial Life II: SFI Studies in the Sciences of Complexity*.

[5]. Hyung-soo Hwang, Jeong-Nae Choi, Won-Hyok Lee: "A Tuning Algorithm for The PID controller Utilizing Fuzzy Theory", IEEE Proceedings 0- 7803-5529-6/99/ Q1999 IEEE.PP:2210-2215.

[6]. Jan Jantzen: "Tuning of fuzzy PID controllers"Denmark.Tech. Report no 98-H871(fpid), 30 Sep 1998.PP:1-22.

[7]. Kiam Heong Ang, Gregory Chong: "PID Control System Analysis, Design, and Technology" IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, VOL.13, No.4, July 2005 PP: 559-576.

[8]. Abdel-Azim S.Ibrahim: " Non-Linear PID Controller Design Using Fuzzy Logic" IEEE MELECON 2002, may 7-9, 2002., pp:595-599