ANT COLONY OPTIMIZATION AND ITS APPLICATION ON PID CONTROLLER: INTRODUCTION AND RECENT TRENDS

Harsha Yadu[1] and Mr. Prabhakar[2]
[1]BHILAI INSTITUTE OF TECHNOLOGY DURG

Abstract: This paper present to design method for determining the optimal proportional-integral-derivative (PID) controller parameters of plant system using the Ant Colony Optimization algorithm there are several methods which are used to tune the controller parameters. They are categorized into two types known as classical methods and modern methods. In this paper the use of ACO method tuned the PID parameter to make them more general and to achieve the steady state error limit, also to improve the dynamic behavior of the system. The performance and design criteria of automatic selection of controller constants are discussed below.

Keywords: ACO- Ant Colony Optimization, PID-Proportional Integral Derivatives

1. INTRODUCTION

During the past decades, the process control techniques in the industry have made great advances. Numerous control methods such as adaptive control, neural control, and fuzzy control have been studied. Among them, the best known is the proportional-integral-derivative (PID) controller, which has been widely used in the industry because of its simple structure and robust performance in a wide range of operating conditions. Unfortunately, it has been quite difficult to tune properly the gains of PID controllers because many industrial plants are often burdened with problems such as high order, time delays, and nonlinearities. For these reasons, it is highly desirable to increase the capabilities of PID controllers by adding new features. Many artificial intelligence (AI) techniques have been employed to improve the controller performances for a wide range of plants while retaining their basic characteristics. AI techniques such as neural network, fuzzy system, and neural-fuzzy logic have been widely applied to proper tuning of PID controller parameters PID controller consists of Proportional, Integral and Derivative gains. The PID feedback control system is illustrated in Fig. where r, e, y are respectively the reference, error and controlled variables [1]. Where Kp is proportional gain, Ki is integral gain and Kd is derivative gain.

A PID controller is described by the following transfer function in the continuous s-domain:

$$Gc = P + I + D$$

$$Gc = K_p + \frac{ki}{s} + K_d S$$

$$Gc = Kp\,(1 + \frac{1}{Ti} S + T_d S)$$

The PID control scheme is named after its three correcting terms, whose sum constitutes the manipulated variable (mv). The proportional, integral, and derivative terms are summed to calculate the output of the PID controller. Defining u(t) as the controller output, the final form of the PID algorithm is:

$$\mathrm{u}(t) = \mathrm{MV}(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt} \quad \ldots (1)$$

Where,
K_p: Proportional gain, a tuning parameter
K_i: Integral gain, a tuning parameter
K_d: Derivative gain, a tuning parameter
e: Error SP - PV
SP: Set Point
PV: Process Variable
t: Time or instantaneous time (the present)
τ : Variable of integration; takes on values from time 0 to the present t.

Equivalently, the transfer function in the Laplace Domain of the PID controller is:
$L(s) = K_p + K_i/s + K_d s$
Where
s: complex number frequency [2].

ACO TOPOLOGY:

Ant Colony Optimization is a search technique between source and destination developed by Macro-Dorigo (1991). It is a search technique to calculate the shortest path between source and destination. It helps in finding optimized path between source and destination. The first ant wanders randomly until it finds the food source (F),then it returns to the nest (N),laying a pheromone trail. It is a chemical substance released by the ants when it goes in search from food it is being released from trail from ant .

Other ants follow one of the paths at random,also laying pheromone trails. The ants on the shortest path lay pheromone trails faster, making it more appearing to future ants The ants become increasingly likely to follow this shortest path. The pheromone trails of the longer path evaporate.

ALGORITHM

At the beginning of the search process, a constant amount of pheromone is assigned to all arcs. When located at a node I and k uses the pheromone trail to compute the probability of Choosing j as the next node Where Ni^k is the neighborhood of ant k when in node i. When the arc(I,j) is traversed, the pheromone value changes as follows:

By using this rule, the probability increases, forthcoming ants will use this arc. After each ant k has moved to the next node the pheromones evaporate by the following equation to all the arcs:
Where p (0,1) is a parameter
Iteration cycle involves ants movement, pheromone evaporation and pheromone deposite.

Pseudo-code

Initialize Trail

Do While (Stopping Criteria Not Satisfied)-

Cycle loop

{

Do Until (Each Ant Completes a Tour)-

Tour loop

{

Local Trail Update

}

End Do

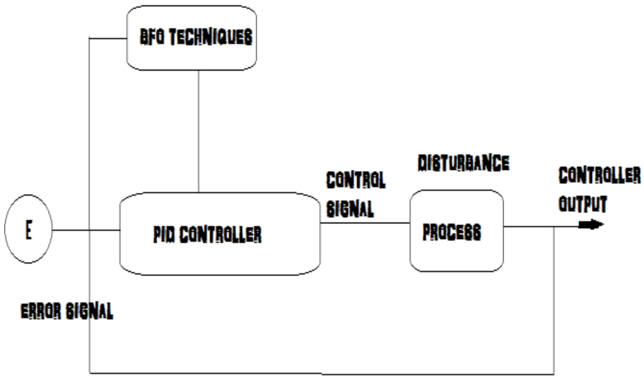
Analyze Tours

Global Trail Update

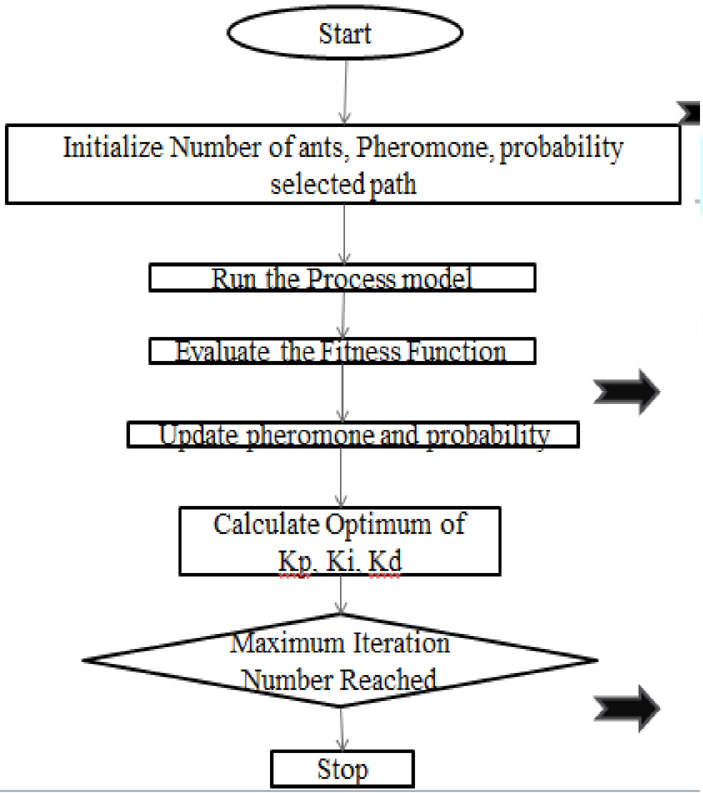
}

End Do

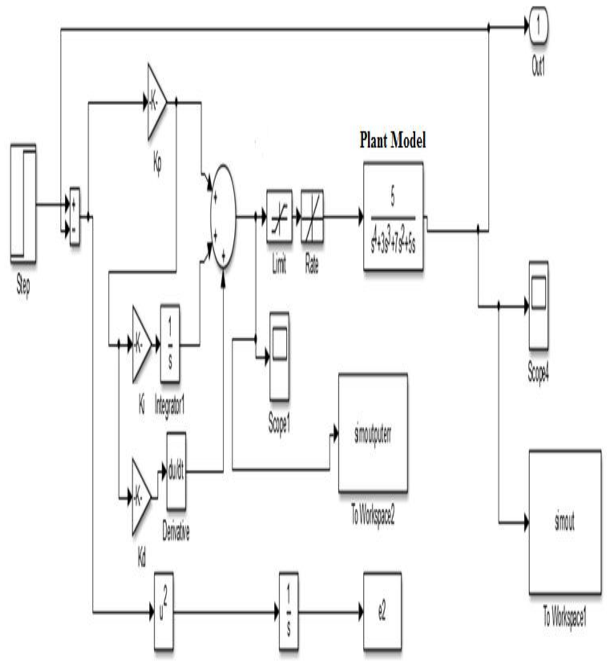
ACO BLOCK DIAGRAM



ACO FLOW CHART



SIMULATION & RESULTS

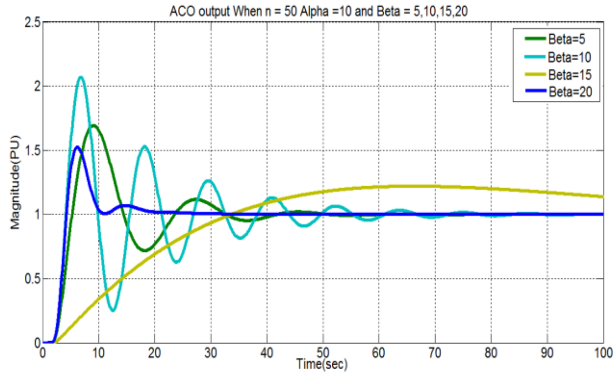


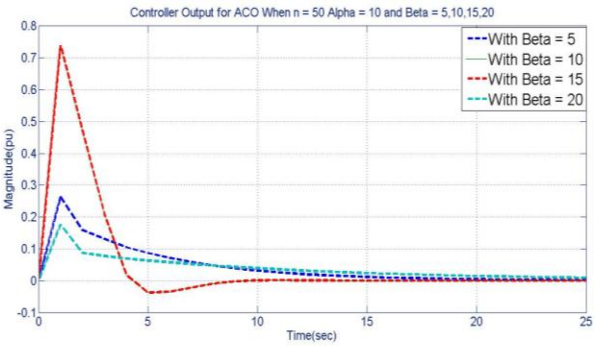
The output of the plant model is determined by comparing the three algorithms according to the values of gain, maximum overshoot, Rise time, settling time.

Comparison between PSO, BFO and ACO for different values of Alpha and Beta

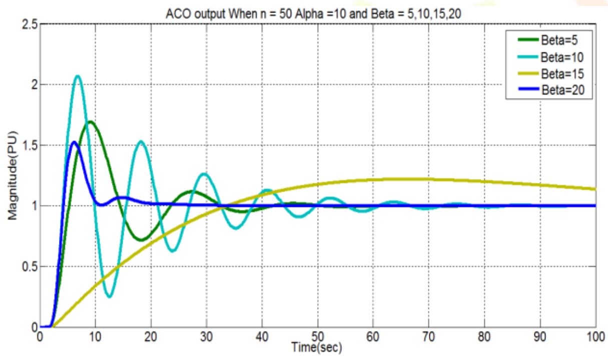
Comparison between PSO, BFO and ACO

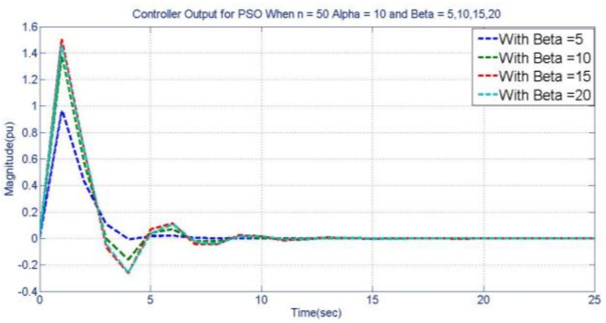
S. NO	GENERATION	ALPHA	BETA	KP	KI	KD	MP %	TS	TR
1	50	10	5	0.9682	0.592	0.854	2.051	6.928	4.262
2	50	10	10	0.8809	0.0699	0.6724	1.692	5.028	2.252
3	50	10	15	0.17585	0.2557	0.8559	0.262	5.121	2.345
4	50	10	20	0.7564	0.6543	0.8553	0.926	4.062	1.862



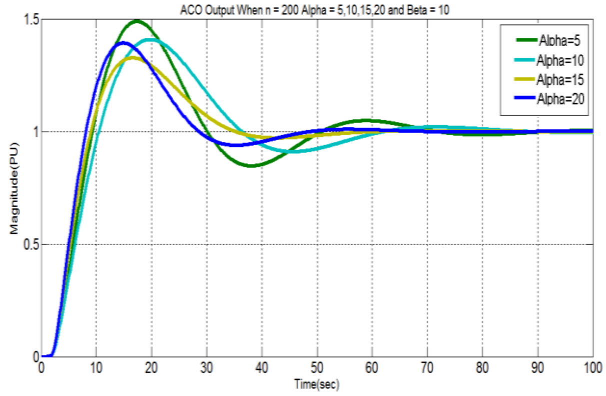


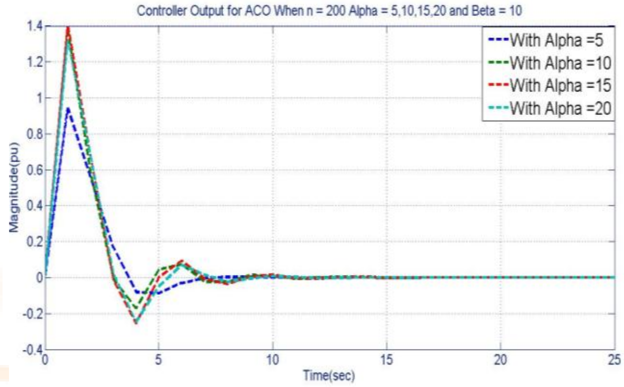
S.NO	GENERATION	ALPHA	BETA	KP	KI	KD	MP%	TS	TR
1	50	10	5	0.7463	0.0582	0.89754	0.4673	2.267	4.4519
2	50	10	10	0.9874	0.6452	0.7564	0.0396	3.029	2.268
3	50	10	15	0.8403	0.625	0.7790	0.692	3.267	4.667
4	50	10	20	0.43961	0.0449	0.18573	2.528	0.29	0.267





S.No	Generation	Alpha	BETa	Kp	Ki	Kd	Mp%	Ts	Tr
1	200	5	10	0.8456	0.0023	0.7856	0.1698	3.267	5.672
2	200	10	10	0.5489	0.0451	0.0487	0.893	4.252	6.293
3	200	15	10	0.1798	0.1054	0.3454	0.2713	8.928	3.028
4	200	20	10	0.9209	0.0924	0.9178	0.0212	5.292	4.028





CONCLUSION

From the closed discussion it is seen that by applying ACO algorithm it provides optimal values for PID parameters for better system performance. Using ACO it can be seen that the best overshoot is achieved many times along with good rise time as well as settling time. BFO algorithm is next optimization technique applied for optimization of PID parameters for stability enhancement of plant model. Overshoot, rise time and settling time are achieved in specified range but as compare to PSO it is not.

REFERENCES

[1] V. Rajinikanth, K. Latha; 'I-PD Controller Tuning for Unstable System using Bacterial Foraging Algorithm:A study based on various Error Criterion ', Applied Computational Intelligence and Soft Computing, Volume 2012.

[2] V. K. Kadiyala; R. K. Jatoth; S. Pathalaiah; '*Design and implementation of fractional order PID controller for Aero fin control system*';IEEE; NaBIC 2009; pp. 696-701.

[3] C. H. Liu; Y. Y. Hsu; 'Design of a Self-Tuning PI Controller for a STATCOM Using Particle Swarm Optimization'; IEEE 2010; pp. 702-715.

[4] D. Maiti, A. Acharya, M. Chakraborty, A. Konar, R. Janarthanan; 'Tuning PID and *PIlDd* Controllers using the Integral Time Absolute Error Criterion'; IEEE 2008; pp. 457- 462.

[5] C. M. Lin, M. H. Lin, C. H. Chen, D. S. Yeung; 'Robust PID control system design for chaotic systems using particle swarm optimization algorithm'; IEEE 2009; pp. 3294-3299.

[6] W.W. Cai, L. X. Jia, Y. B. Zhang, N. Ni; 'Design and simulation of intelligent PID controller based on particle swarm optimization'; IEEE 2010.