Application of Ant Colony Algorithm for calculation and analysis of Performance Indices for adaptive control system

Prof. A.Q.Ansari
JAMIA MILLIA ISLAMIA
NEW DELHI

Prof. Ibraheem
JAMIA MILLIA ISLAMIA
NEW DELHI

Sapna Katiyar
JAMIA MILLIA ISLAMIA
NEW DELHI

ABSTRACT **– To achieve good performance from the system, performance index plays vital role in the objective function. In this paper various performance indices are used as the objective functions. The choice of the objective function is the most crucial and complicated step in applying any optimizing algorithm for an adaptive control system. They are used to evaluate fitness of items for iterations. The various objective functions like Mean of the Squared Error (MSE), Integral of Time multiplied by Absolute Error (ITAE), Integral of Absolute Magnitude of the Error (IAE), Integral of the Squared Error (ISE) and Integral of Time multiplied by the Squared Error (ITSE) have been analyzed and compared to find the most suitable one. Ant Colony Optimization algorithm is applied to tune a PID controller to find out best solution and to study the behavior of different performance indices.**

Keywords- Adaptive Control; PID Performance Index; Ant Colony Optimization; MSE; ITAE; IAE; ISE; ITSE.

I. INTRODUCTION

The most familiar controller in industry, about 95% of the control loops [1] in the area of control project, is the Proportional-Integral-Derivative (PID) controller. It has been successfully implemented for over 50 years, as it provides satisfactory robust performance despite of varying dynamic characteristics of a process plant. The general form of parallel PID, as shown in Fig 1.It has an advantage of having a simple algorithm, easy implementation, robust performance, universality. It allows its application to numerous processes viz. pharmaceutical, textile, clinical laboratories, process control, robotics, defense and space technologies etc. The correct implementation of the PID depends on the specification of three parameters [1]: proportional gain (Kp), integral time (Ti) and derivative time (Td). These three parameters are often tuned manually by trial and error, which has a major problem in the time needed to accomplish the task. Other limitations of a PID controller are that it cannot handle multivariable systems and complex systems

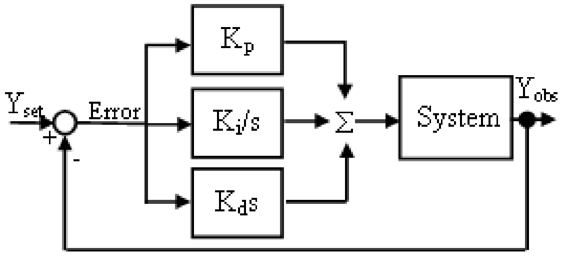

Fig 1: PID Controller Configuration

A well-known and commonly used continuous Proportional-Integral-Derivative (PID) controller is described using

$$u = K_P\left(e + \frac{1}{K_i}\int_0^t e d\tau + K_d \frac{de}{dt}\right) \quad (1)$$

where $e=T_{set}-T_{obs}$, u is the controller output, K_p is the proportional gain, K_i is the integral reset time, K_d is the derivative time, and e is the error between the set point and the process output. For a digital control of T_s sampling periods, we can write

$$u = K_P\left(e_n + \frac{1}{K_i}\sum_{j=1}^{n} e_j T_s + K_D \frac{e_n - e_{n-1}}{T_s}\right) \quad (2)$$

In present case the equation is slightly modified for better efficiency and is taken as

$$u_i = u + I \quad (3)$$

where u as given by equation (2) , u_i is i^{th} output term and I is the suitably chosen offset value.

The step response technique has been employed for tuning the PID parameters Kp, Ki and Kd. After getting these parameters they have been put in the main PID controller program and hence the PID controls the system effectively and efficiently.

An improvement in tuning can be achieved by using soft computing based optimization technique. It makes the system to adapt according to the changes in its environment. The role of adaptive control systems therefore becomes of prime importance. Soft computing techniques like e.g. Neural Networks, Fuzzy Sets, Genetic Algorithm [2], Ant Colony Optimization Algorithm [3-4] Particle Swarm algorithms [5], Bee Colony Optimization, Cuckoo Search, Bat Algorithm, Teaching learning based techniques and hybrid techniques with their fusion are commonly used.

II. IMPORTANCE OF USING SOFTCOMPUTING TECHNIQUES

The main features of soft computing are their capability to tolerate imprecision and inaccuracy unlike hard computing techniques. Soft computing techniques are also useful when a large and complex search space is available and system knowledge and accurate dynamic models are not available.

Some other reasons to choose soft computing techniques arise like when model reduction is necessary if the original model is too complicated. These aim of these methods is to optimize some design criteria that characterize the properties of the closed-loop system. Examples of such types of criteria are gain and phase margins, closed-loop bandwidth, and different cost functions for step and load changes.

All these soft computing techniques work with an objective function. Writing and choice of an objective function is the most difficult part of using any of the soft computation technique. In this paper, the objective function is required to evaluate the best cost function of PID controller. An objective function could be created to find a PID controller that gives the desired performance based on the system.

III. OBJECTIVE FUNCTIONS

Objective function is a mathematical expression which describes a relationship of the optimization parameters or the result of an operation that uses the optimization parameters as inputs. Here the PID controller is used to minimize the error signals and to find out the best possible solution. In general greater the value of objective function is the worse the control is. Hence fitness of the item is defined as: *performance index*

$$fitness\ \ value = \frac{1}{performance\ \ index}$$

Ant Colony Optimization algorithm is applied to tune a PID controller to study the behavior of different performance indices.

Here various objective functions are compared to find the most suitable one are Mean of the Squared Error (MSE), Integral of Time multiplied by Absolute Error (ITAE), Integral of Absolute Magnitude of the Error (IAE), and Integral of the Squared Error (ISE) and Integral of Time multiplied by the Squared Error (ITSE) to minimize the error signal. These indices are defined as follows:

$$MSE = \frac{1}{t}\int_0^t (e(t))^2\, dt$$

$$ITAE = \int_0^t t\left|e(t)\right| dt$$

$$IAE = \int_0^t \left|e(t)\right| dt$$

$$ISE = \int_0^t (e(t))^2\, dt$$

$$ITSE = \int_0^t te(t)^2\, dt$$

where e(t) is the error signal in time domain.

IV. PID CONTROLLER TUNING METHODS

A. Ziegler Nichols Method

This method is applicable to open loop transfer function and the type of response is typical of a first order system with transportation delay. The response is characterized by two parameters, *L* the delay time and *T* the time constant. These are found by drawing a tangent to the step response at its point of inflection and noting its intersections with the time axis and the steady state value Ziegler, J.G et al. (1942).

Table 1: Tuning parameters for Ziegler Nichols First method

Type of Controller	K_P	$T_i=K_p/K_i$	$Td=K_d/K_p$
P	T/L		
PI	0.9 T/L	L/0.3	
PID	1.2 T/L	2L	0.5 L

B. Ant Colony Optimization Algorithm

Ant Colony Optimization algorithms are especially suited for finding solutions to difficult optimization problems Dorigo (1996). A colony of artificial ants cooperates to find good solutions, based on their similarities with ant colonies in nature, ant algorithms are adaptive and robust and can be applied to different optimization problems. ACO algorithm has been shown in Fig 2.

The ants are driven by a probability rule to choose their solution to the problem, known as a tour. The probability rule between two nodes i and j, called Pseudo-Random-Proportional Action Choice Rule, depends on two factors: the heuristic and metaheuristic.

$$p_{ij} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{h\in s}[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}} \qquad (4)$$

where τ is the pheromone, η is the inverse of the distance between the two cities, q is a random variable uniformly distributed over [0, 1], q_0 is a tunable parameter in the interval [0, 1], and J belongs to the candidate list and is selected based on the above probabilistic rule.

Each ant modifies the environment in two different ways, Dorigo (1999):

- Local trail updating: As the agent moves between cities it updates the amount of pheromone on the edge by the eq (5):

$$\tau_{ij}(t) = (1-\rho).\tau_{ij}(t-1) + \rho.\tau_0 \qquad (5)$$

Where, ρ is the evaporation constant. The value τ_0 is the initial value of pheromone trails and can be calculated as

$$\tau_0 = (n.L_{nn})^{-1} \qquad (6)$$

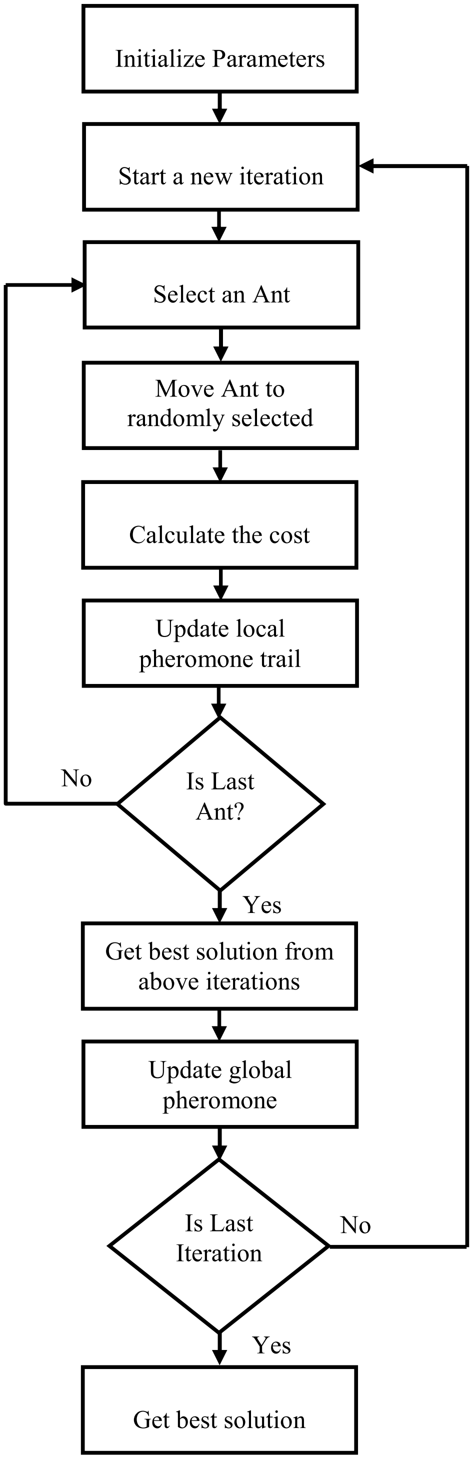

Fig.2. Flowchart for ACO algorithm

Where, n is the number of cities and L the length of the tour produced by one of the construction heuristics.

- Global trail updating: When all agents have completed a tour the agent that finds the shortest route updates the edges in its path using the following equation:

$$\tau_{ij}(t) = (1-\rho).\tau_{ij}(t-1) + \frac{\rho}{L^{+}} \qquad (7)$$

where, L^{+} is the length of the best tour generated by one of the agents.

V. SIMULATION RESULTS AND DISCUSSION

An Ant colony algorithm based adaptive PID control system has been designed and is tested at various set point temperatures under varying environmental conditions. The PID parameters of the controller have been automatically tuned to give the satisfactory results by Ant colony optimization technique (ACO).

Simulations are carried out by taking different orders of transfer functions. Theoretical analysis and simulation examples show that the ACO-PID controller can be used for proper tuning. A typical optimized graph obtained from MATLAB using ACO optimization is shown in Fig.3.

Table-2: ACO parameters

Population Size	200
No. of Ants	50
No. of Paths	20
Iteration	200

Model 1:

$$G(s) = \frac{0.8}{s+1} e^{-0.1s}$$

Model 2:

$$G(s) = \frac{1}{s+1} e^{-0.3s}$$

Model 3:

$$G(s) = \frac{1}{s^2 + 10s + 20}$$

Model 4:

$$G(s) = \frac{e^{-2s}}{(3s+1)(5s+1)}$$

Model 5:

$$G(s) = \frac{e^{-0.5s}}{(2s+1)(s+1)(0.5s+1)}$$

Simulation results of all five models are shown from Fig. 3 to Fig. 17. Values of the PID parameters i.e. K_p, K_i, K_d obtained from different models have been shown in Table 3-8.

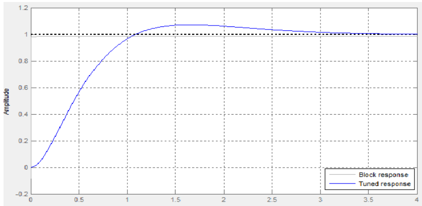

Fig.3

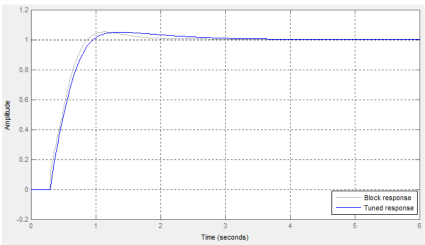

Fig.4

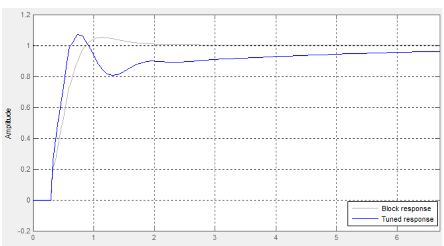

Fig.5

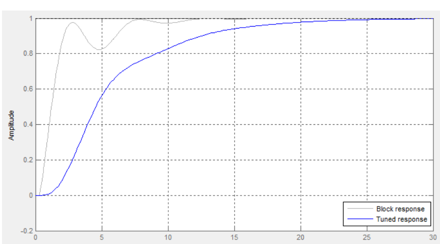

Fig.6

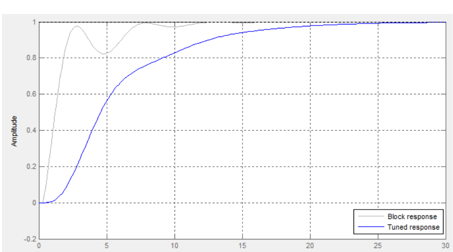

Fig.7

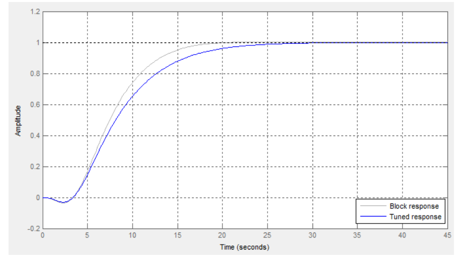

Fig.8

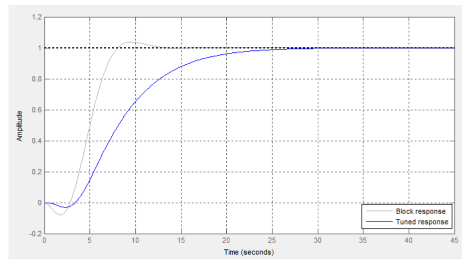

Fig.9

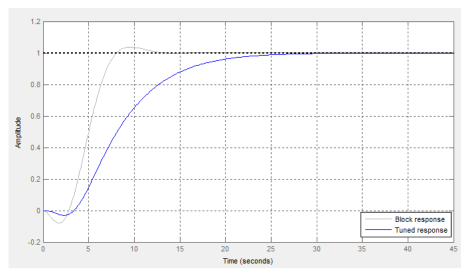

Fig.10

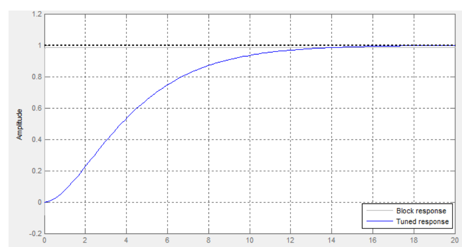

Fig.11

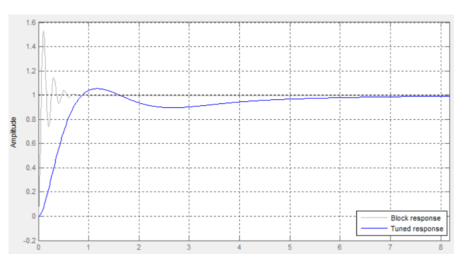

Fig.12

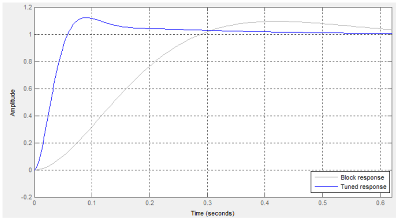

Fig.13

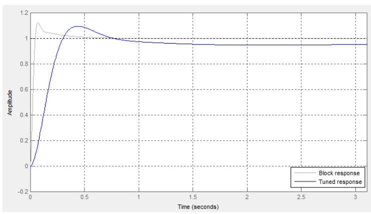

Fig.14

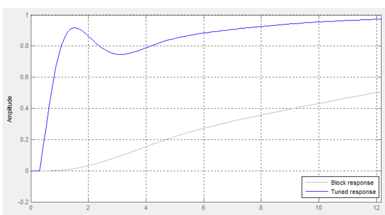

Fig.15

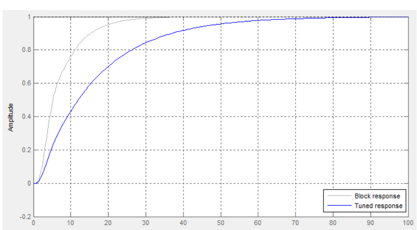

Fig.16

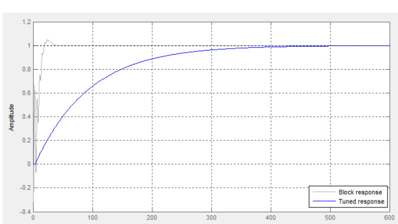

Fig.17

Table-3: Parameters: Model 1

Parameter	MSE	ITAE	IAE	ISE	ITSE
K_p	190.0047	190.0047	190.0047	190.0047	190.0047
K_i	62.165	62.165	62.165	62.165	62.165
K_d	42.4905	42.4905	42.4905	42.4905	42.4905
N	86.2646	86.2646	86.2646	86.2646	86.2646
Overshoot%	1.45	0.35	1.56	0.989	1.335
Rise time(s)	562	1140	1145	1350	1260
Settling Time(s)	1730	1800	1538	1680	1790

Table-4: Parameters: Model 2

Parameter	MSE	ITAE	IAE	ISE	ITSE
K_p	190.0047	190.0047	190.0047	190.0047	190.0047
K_i	62.165	62.165	62.165	62.165	62.165
K_d	42.4905	42.4905	42.4905	42.4905	42.4905
N	86.2646	86.2646	86.2646	86.2646	86.2646
Overshoot%	2.3	0.6	1.9	1.6	2.1
Rise time(s)	3120	2020	2650	2800	2840
Settling Time(s)	3300	3150	2940	2600	2720

Table-5: Parameters: Model 3

Parameter	MSE	ITAE	IAE	ISE	ITSE
Kp	0.18014	0.18014	0.18014	0.18014	0.18014
Ki	0.19697	0.19697	0.19697	0.19697	0.19697
Kd	0.01565	0.01565	0.01565	0.01565	0.01565
N	100	100	100	100	100
Overshoot%	0.6	0.02	0.9	0.4	0.9
Rise time(s)	3610	3630	3580	4210	3900
Settling Time(s)	1120	1200	1150	1080	1230

Table-6: Parameters: Model 4

Parameter	MSE	ITAE	IAE	ISE	ITSE
Kp	2.1419	2.1419	2.1419	2.1419	2.1419
Ki	1.971	1.971	1.971	1.971	1.971
Kd	0.11946	0.11946	0.11946	0.11946	0.11946
N	236.12	236.12	236.12	236.12	236.12
Overshoot%	1.14	0.8	1.2	0.8	1.0
Rise time(s)	1610	1320	1445	1550	1360
Settling Time(s)	1080	1100	1200	1350	1290

Table-7: Parameters: Model 5

Parameter	MSE	ITAE	IAE	ISE	ITSE
Kp	0.9262	0.9262	0.9262	0.9262	0.9262
Ki	0.5416	0.5416	0.5416	0.5416	0.5416
Kd	0.3862	0.3862	0.3862	0.3862	0.3862
N	118.64	118.64	118.64	118.64	118.64
Overshoot%	1.12	0.2	0.8	1.0	1.0
Rise time(s)	1610	1420	1445	1750	1660
Settling Time(s)	1180	1100	1240	1350	1380

Table shows an approximate evaluation of the cost functions for each model. The evaluation parameters are the overshoot (OS), rise time (RT), settling time (ST).

Table-8: Comparison of performance indices of different objective functions

Performance Index	Model 1	Model 2	Model 3	Model 4	Model 5
MSE	0.53	0.63	0.71	0.34	0.21
ITAE	0.018	0.016	0.021	0.029	0.014
IAE	0.15	0.27	0.29	0.19	0.14
ISE	0.15	0.28	0.29	0.31	0.26
ITSE	0.47	0.56	0.63	0.29	0.32

VI. CONCLUSION

Ant colony optimization algorithm was applied to adaptive PID controller and after their implementation in all five different models their step response was simulated using the Matlab/Simulink™ software.

According to the simulations results, the advanced control algorithms using artificial algorithm perform satisfactory step behavior with good set point tracking and smooth steady state approach. System also sustains their robustness and performance during the introduction of input constraints or measured disturbances.

With the analysis and comparison done in the present paper one can conclude that out of the five objective functions, ITAE is the best choice for the optimization of the PID parameters for the adaptive control system under consideration.

Further other soft computing techniques can also be applied to compute best objective function and their results can be compared with ant colony optimization to find out the best soft computing techniques.

REFERENCES

[1] Ibtiseem Chiha, Pierre Borne: "Multi-Objective Ant Colony Optimization to Tuning PID controller" proceedings of the International Journal of Engeineering, Volume III, Issue no 2, March 2010.

[2] Neetu Thomas, Dr. P.Poongodi, " Position Control of DC Motor using Genetic Algorithm based PID Controller" Proceedings of the World Congress on Engineering , Vol II WCE 2009, July 1-3, 2009, London, U.K.

[3] YIN Hong-peng, CHAI Yi, "Parameters optimization design of PID controller based on Ant Colony Algorithm", Computer Engineering and Applications, Beijing 2007, 43(17), pp,4-7

[4] HE Guang-Hui, TAN Guan-zheng, "An Optimal Nonlinear PID Controller Based on Ant Algorithm", Programmable Controller & Factory Automation, Beijing, 2007, pp, 98-104

[5] Bequette, B. W., Process Control: Modelling, Design and Simulation. Prentice Hall, Upper Saddle River, NJ, 2003

[6] Dorigo M., Maniezzo V. and Colorni A., (1996) Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics.

[7]Kennedy J., and Eberhart. R., "Particle swarm optimization," Proc IEEE Int. Conf on Neural Network, pp. 1942-1945, 1995

[8]Åström, K. J. and B. Wittenmark, "Adaptive Control", Addison-Wesley, Reading, MA, 1989.

[9] Goldberg, D.E., "Genetic Algorithms in Search Optimization and Machine Learning", Addison-Wesley, 1989.