Tuning of PID Controllers using Advanced Genetic Algorithm

Ms. Reshmi P. Pillai
Department of Electronics Engg.
RamraoAdik Institute of Technology
Nerul, Navi Mumbai, India

Sharad P. Jadhav
Department of Electronics Engg.
RamraoAdik Institute of Technology
Nerul, Navi Mumbai, India

Dr. Mukesh D. Patil
Department of Electronics Engg.
RamraoAdik Institute of Technology
Nerul, Navi Mumbai, India

Abstract—**Proportional-Integral-Derivative (PID) controllers have been widely used in process industry for decades from small industry to high technology industry. But they still remain poorly tuned by use of conventional tuning methods. Conventional technique like Zeigler-Nichols method does not give an optimized value for PID controller parameters. In this paper, we optimize the PID controller parameter using Genetic Algorithm (GA), which is a stochastic global search method that replicates the process of evolution.**

Using genetic algorithm the tuning of the controller will result in the optimum controller being evaluated for the system every time. The GA is basically based on an iterative process of selection, recombination, mutation and evaluation. The performance of Advanced Genetic Algorithm (AGA) is compared with Guo Tao's Algorithm (GTA) and Elite Multi-Parent Crossover Evolutionary Optimization Algorithm (EMPCOA). AGA has a different replacement strategy as compared to EMPCOA which helps to maintain the population diversity and thus reducing the computational time which is proved by the results presented here.
The effectiveness of the AGA is also verified for a system with an unstable plant. The PID controller is also tuned with different error criteria viz. Integral Time Absolute Error (ITAE), Integral Square Error (ISE) and Integral Absolute Error (IAE).

Keywords-PID tuning; Genetic Algorithm; Multi-parent crossover; Elite crossover; Discrete recombination.

I. INTRODUCTION

The PID controller was patented in 1939 by Albert Callender and Allan Stevenson of Imperial Chemical Limited of Northwich, England. The PID controller is widely used in most industrial processes despite continuous advances in control theory. The main reason is due to their simplicity of operation, ease of design, inexpensive maintenance, low cost, and effectiveness for most linear systems. Recently, motivated by the rapidly developed advanced microelectronics and digital processors, conventional PID controllers have gone through a technological evolution, from pneumatic controllers via analog electronics to microprocessors via digital circuits [1]. Most conventional PID tuning methods require considerable technical experience to apply tuning formulas to determine the PID controller parameters. The conventional tuning methods require the process model to be reduced if it is too complicated originally [2]. In practical applications, most of the industrial processes exist to be non-linear, variability of parameters and uncertainty of models are very high, thus using conventional PID tuning methods the precise control of the process cannot be achieved. Due to this, PID controllers are rarely tuned optimally and thus require improved tuning technology. The above problems can be well addressed by the application of non-conventional methods for tuning of the PID controller. Most practical PID remains poorly tuned leading to deteriorated process performance [3]. The conventional tuning methods require considerable technical experience and are time consuming and do not work well for non-linear, higher order and time-delayed systems and the ones that do not have a precise mathematical model [1]. Non-conventional methods are especially useful for solving problems of computationally complicated and mathematically untraceable. Hence the need arises for an optimization algorithm like Genetic Algorithm (GA).

Genetic Algorithm is a stochastic search and optimization method that mimics the process of natural evolution [4]. John H. Holland formally introduced GA in his book, ‘Adaptation in Natural and Artificial Systems’ in the 1975 at the University of Michigan, Ann Arbor, United States. GA is one of the Evolutionary Algorithms (EA) methodologies. The key aspect distinguishing an evolutionary search algorithm from traditional algorithms is that it is population-based. Through the adaptation of successive generations of a large number of individuals, EA performs an efficient directed search. Evolutionary search is generally better than random search as EA inspired by the evolution process in nature and try to solve problems by evolving sets of search points. GA imitates natural evolution with survival of the fittest approach. It performs on coding of parameters hence does not depends on the continuity of parameter nor the existence of the derivatives of the functions, thus allowing it to handle multi parameters or multi-model type of optimization problems. GA can also work for non-deterministic systems or the systems that can be only partially modeled. GA uses random choice and probabilistic decision to guide the search, where the population improves toward near optimal points from generation to generation [5]. The main advantage of the GA formulation is that fairly accurate results may be obtained using a very simple algorithm. The GA is basically based on an iterative process of selection, recombination, mutation and evaluation. GA has parallel search techniques, which emulate natural genetic operations. Due to its high potential for optimization, GA has received great attention in control systems such as the search of optimal PID controller parameters.

The paper is organized as follows: Section II gives basic idea of a PID controller. Section III introduces Genetic Algorithm. The algorithm of implementation of Advanced GA is given in Section. The design steps are discussed in Section V. The results are presented in Section VI. Final conclusions are drawn in Section VII.

II. PID CONTROLLER

A PID controller aims at minimizing the error between a measured process variable of the controlled system and a reference, by calculating the error and generating a correction signal to the system from the error. The block diagram of a conventional PID controller is shown in the Fig. 1, where *r*(*t*)is the reference value, *y*(*t*) is the output of the controlled system, *e*(*t*) is the error between *r*(*t*) and *y*(*t*), whereas *u*(*t*) is the output control signal of the PID controller. A conventional PID controller consists of three components: the proportional part, the integral part and the derivative part as shown in Fig. 1. The proportional term produces an output value that is proportional to the current error value. The contribution from the integral term is proportional to both the magnitude of the error and the duration of the error. Derivative control is used to reduce the magnitude of the overshoot produced by the integral component and improve the combined controller-process stability. The output control signal of a PID controller is described as follows,

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}, \tag{1}$$

where, u(t) is the output control signal, e(t) is an error signal, and K_p, K_i, and K_d refers to the proportional gain, the integral gain and the derivative gain, respectively.

K_p, K_d and K_i should satisfy the following equations,

$$K_i = K_p T_i, \tag{2}$$

$$K_d = K_p T_{d,} \tag{3}$$

where T_i and T_d refers to the integration time and derivative time, respectively.

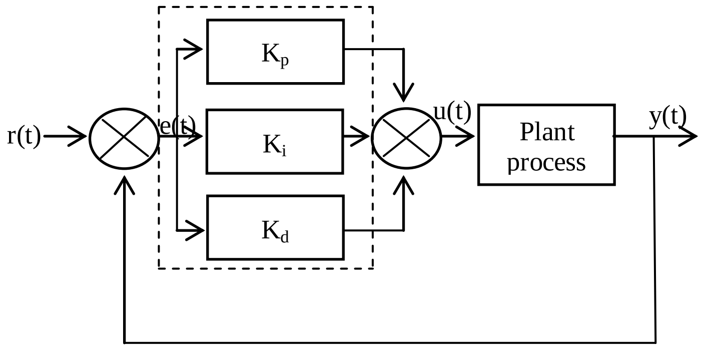

Fig. 1. PID Controller Block Diagram.

The individual effects of these terms on the closed-loop performance are summarized in Table I. Note that this table serves as a first guide for stable open-loop plants only. For optimum performance K_p, K_i and K_d are mutually dependent in tuning.

TABLE I. Effects Of Independent P, I And D Tuning [6]

Closed loop response	**Rise Time**	**Over-shoot**	**Settling Time**	**Steady State Error**	**Stability**
Increase K_p	Decrease	Increase	Small Increase	Decrease	Degrade
Increase K_i	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increase K_d	Small Decrease	Decrease	Decrease	Minor change	Improve

The quality of PID tuning rules is of considerable practical importance because a small percentage improvement in the operation of a plant can translate into large economic savings or other benefits.

III. GENETIC ALGORITHM

GA is a probabilistic optimization algorithm with a high probability of finding a good solution in a given search space. Genetic Algorithm can handle multiple variables and only requires the ability to develop a mathematical model to configure a set of inputs (the variables) in order for the model to produce an optimal output. After initialization of population, each string (individual) in the population is evaluated to determine the performance of the string. Then, the higher-ranking strings are mate. The process of crossover is performed by combining strings containing partial solutions. The algorithm favors fittest strings as parents, thus better strings will have more number of offspring. The GA exploits the regions of the solution space, because successive generations of reproduction and crossover produce increasing numbers of strings in those regions. In this paper the offspring replaces the weakest string, thus maintaining the population size same [7]. Lastly, mutations modify a small fraction of the strings. Mutation alone does not generally advance the search for a solution, but it does provide insurance against the development of a uniform population incapable of further evolution [8].

Guo Tao's Algorithm (GTA) is a linear non-convex multi-parent crossover operator (GTX) which is used in optimization of nonlinear continuous functions [9]. The multi-parent crossover utilizes more number of candidate solutions and the replacement strategy implemented is supposed to be minimizing selection pressure. But major limitation of GTA is that it may ignore better solutions in population. To make use of better solutions in population elite preservation strategy is introduced by Xiaoyi Che, Youxin Luo and Zhaoguo Chen implemented in the elite multi-parent crossover evolutionary optimization algorithm (EMPCOA) [10]. The selection scheme and replacement strategy implemented in EMPCOA gives global optima with increase in an execution time. This is mainly due to the decrease in population diversity and therefore requires more number of iterations to converge.

Aimed at these shortages of GTA and EMPCOA we are motivated to implement the Multi-parent Crossover Algorithm with Discrete Recombination [7] with better parts of both algorithms like, fixed population size of GTA and elite preservation strategy of EMPCOA with multi-parent crossover. Here, we aim to reduce the number of iterations and execution time with improvement in transient (performance) response.

General steps involved in GA are.

- Representation
- Objective function
- Population initialization
- Parent selection mechanism
- Variation operator, crossover (recombination)
- Variation operator, mutation and
- Termination condition.

A GA is typically initialized with a randomly generated population consisting of candidate individuals. Each individual in the population is usually represented by a real-valued number or a binary string. Such strings are called as chromosomes. A set of chromosome or individual is a whole population. Performance of each individual is measured and assessed by the objective function. The objective function assigns each individual a corresponding number called its fitness value. If the termination criteria are not met with the current population then new individuals are created with genetic operators. A survival of the fittest strategy is applied on individuals. The fittest parents are found out by reproduction or selection operator. New individuals are generated by performing operations such as crossover and mutation on the individuals whose fitness has just been measured. The fitness of the offspring is then computed. The offspring is inserted into the population replacing the parents or low-fitness individuals producing a new generation. This cycle is performed until the termination criterion is reached. Such a single population GA is powerful and performs well on a wide variety of problem. Every iteration of GA loop is referred to as a generation. When termination criteria gets satisfied GA stops. The work flow of GA is as shown in Fig 2.

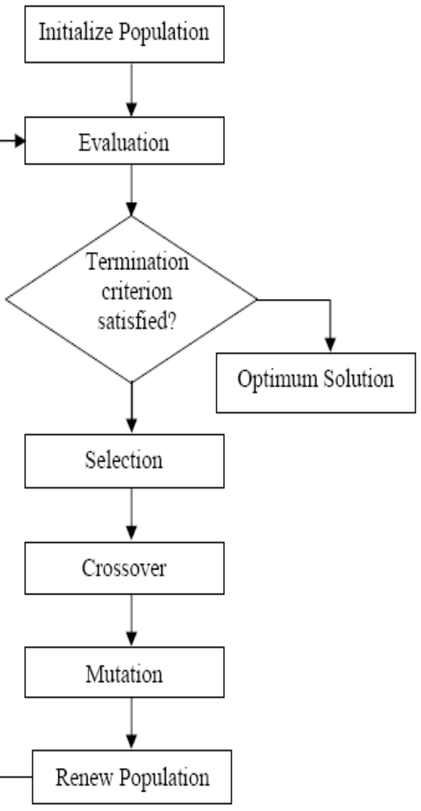

Fig. 2. Workflow of Genetic Algorithm [11].

IV. PID TUNING USING GA METHOD

The main features of the algorithm implemented are,

- It uses elite preservation strategy.
- It makes use of multi-parent crossover to create new offspring.
- The performance of the proposed algorithm is better than the existing algorithms like GTA and EMPCO in terms of number of iterations and computational time.
- It gives better transient response as compared to the existing algorithms like GTA and EMPCOA.

The design steps of the algorithm are

Step 1 Produce initial population $P_0 = X_1; X_2 \ldots X_N$ randomly at searching space S, N is number of individuals in populations and t = 0.

Step 2 Arrange the individuals in population P from good to bad according to the fitness of parent. Then still record as $P_t = X_1, X_2 \ldots X_N$ after the arrangement, X_1 is the best individual X_N is the worst one.

Step 3 Termination criteria is when the fitness difference between X_{worst} and X_{best} become less than or equal to fitness limit (ε) (i.e. the fitness of the worst individual is almost same as the best one), then go to step 7.

Step 4 Choose K (K≤m) best individuals $X_1, X_2 \ldots, X_k$ from population P_t, and then, choose (m-K) individuals $X_{k+1}, X_{k+2} \ldots, X_m$ from the rest (N-K) individuals randomly. A subspace V is formed from these m (m ≤ N) individuals. Then perform multi-parent crossover as given in (4).

$$V = x_c \in S, \qquad x_c = \sum_{i=1}^{m} a_i x_i, \tag{4}$$

$$\sum_{i=1}^{m} a_i = 1, \quad -0.5 \le a_i \le 1.5. \tag{5}$$

Step 5 Compare x_c with X_{worst}, x_c replaces X_{worst} if better (x_c, X_{worst}) condition is true else discard x_c.

Step 6 Go to Step 2

Step 7 Output is the best solution and end.

Here, we implement the algorithm for optimizing PID parameters. The following section describes how gains K_p, K_i and K_d are represented in form of chromosome or individual. The implementation of PID parameters optimization procedure using GA starts with the chromosome representation as shown in Fig. 3.

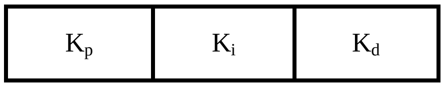

Fig. 3. Chromosome representation

The chromosome is formed by three values that correspond to the three gains to be adjusted in order to achieve a satisfactory behavior. The gains K_p, K_i and K_d are strings of chromosome as shown in Fig. 3. A set of chromosomes or individual forms a generation.

An objective function could be created to find a PID controller that gives the smallest overshoot, fastest rise time or quickest settling time. There are several variables used as the standard to measure system performance. In general, unit step input is used to test the systems, and the output signals is characterized by some standard performance measures likes settling time, percentage overshoot, rise time and peak time. All these measures are defined in the time domain response. Each chromosome in the population is passed into the objective function one at a time. The chromosome is then evaluated and assigned a number to represent its fitness, which is its fitness value. The GA uses the chromosomes fitness value to renew population consisting of the fittest members. When the chromosome enters the evaluation function, it is split up into its three terms. The newly formed PID controller is placed in a unity feedback loop with the system transfer function. This will result in reducing the compilation time of the program. The system transfer function is defined in another file and imported as a global variable. The controlled system is then given a step input and the error can be assessed using error performance criterion integral absolute error (IAE).

V. DESIGN PROCEDURE

- The initial population of 10 chromosomes is generated between 0 to 20 using random number generation.
- The objective function is used as error performance criterion. The error criteria used is Integral of absolute error (IAE). The magnitude of this error will be used to assess the fitness of each chromosome.
- The termination criterion for the algorithm is based on fitness limit. The algorithm terminates when the difference between the fitness of best solution X_{best} and worst solution X_{worst} is less than or equal to fitness limit $\varepsilon = 1$.
- A subspace is formed using elite preservation strategy and then multi-parent crossover takes place using 5 parents. The 2 best chromosomes are selected and the 3 random parents are selected from the remaining 8 chromosomes.
- In the proposed algorithm the offspring solution (x_c) is compared and replaced with X_{worst}.

VI. RESULT

A. Example 1

We implement all three algorithms on a system with transfer function given as follows.

$$G(s) = \frac{1}{0.004s^3 + 0.08s^2 + 0.5s + 1} \tag{6}$$

The controller parameter and the computational time for all three algorithms are given in Table II. It can be seen that AGA requires least amount of time to obtain the optimum controller gains.

TABLE II. Comparison Of Controller Gains & Computational Time For Example 1

	GTA	EMPCOA	AGA
K_D	16.1815	2.5092	0.948
K_P	14.9747	15.0449	6.847
K_I	17.3709	16.5410	14.72
No. of iterations	15	11	8
Computational time (s)	3.85	6.83	2.06

The closed loop step response of the system using different algorithms is as shown in Fig. 4. It is observed that the transient response of the system using AGA is best because of fastest settling time and least overshoot. The same is summarized in Table III.

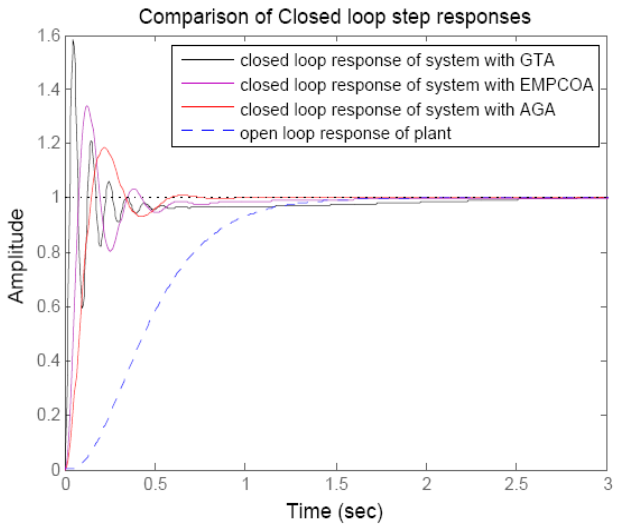

Fig. 4. Comparison of closed loop step responses for GTA, EMPCOA and AGA for the illustrative example 1.

TABLE III. Comparison Of Step Response For Example 1

	GTA	EMPCOA	AGA	Open loop plant
Settling time (s)	1.78	0.812	0.538	1.29
Rise time (s)	0.0186	0.0534	0.101	0.723
Overshoot (%)	59	34	18	0

B. Example 2

We implement all three algorithms on a system with unstable plant whose transfer function is given as follows.

$$G(s) = \frac{6}{s^2 - 0.5s + 6} \quad (7)$$

The controller parameter and the computational time for all three algorithms for example 2 are given in Table IV. It can be observed the using AGA the optimum controller gains are obtained in minimum number of iterations although the amount of time required is slightly greater than that of EMPCOA.

The closed loop step response of the system using different algorithms is as shown in Fig. 5. It is observed from the transient response of the system that AGA has minimumovershoot although the settling time required is more than that required by EMPCOA. The same is summarized in Table V.

TABLE IV. Comparison Of Controller Gains & Computational Time For Example 2.

	GTA	EMPCOA	AGA
K_d	8.3112	7.8441	17.8181
K_P	5.3240	16.1898	12.1858
K_i	11.2521	12.1148	15.9448
No. of iterations	6	11	4
Computational time (s)	2.46	6.49	3.3

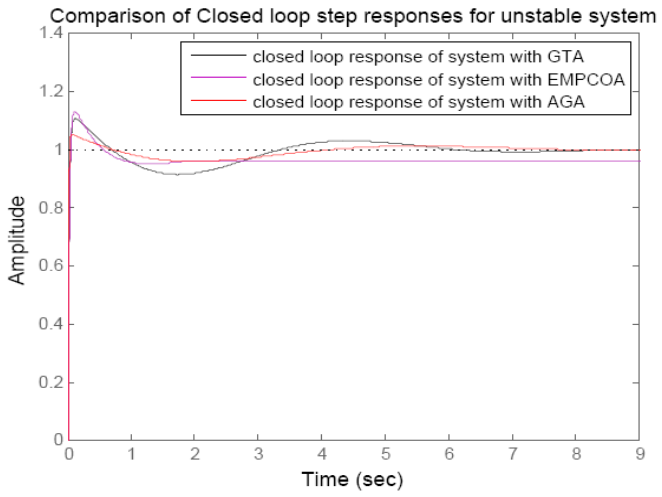

Fig. 5. Comparison of closed loop step responses for GTA, EMPCOA and AGA for the illustrative example 2.

TABLE V. Comparison Of Step Response For Example 2

	GTA	EMPCOA	AGA
Settling time (s)	5.15	2.27	3.31
Rise time (s)	0.0366	0.0366	0.0366
Overshoot (%)	10.6	12.9	5.05

C. Comparison with different error criteria

The parameters of PID controller are obtained with different error criteria for Example 1. The open loop transfer function is given by the (6). The performances of the designed controllers are compared. The controller gains (*Kp*, *Kd* and *Ki*) are obtained using Advanced Genetic Algorithm. The AGA is implemented using different error criteria, viz. ITAE, ISE and IAE as objective function for the system.

The controller gains so obtained with different error criteria are as shown in Table VI. AGA using ISE requires minimum number of iterations to compute the parameter value, but IAE gives best performance with minimum computational time while ITAE requires highest computational time.

A comparison of closed loop step response of the system using AGA with all three error criteria as objective function is as given in Fig.6. The same is summarized in Table VII, which shows that the system with AGA using IAE gives the best performance with fastest settling time and minimum overshoot

TABLE VI. Comparison Of Controller Gains & Computational Time For Example 1 Using Different Error Criteria.

	ITAE	ISE	IAE
K_D	6.3486	9.2521	0.948
K_P	16.2908	7.3565	6.847
K_I	15.7815	13.5914	14.72
No. of iterations	42	3	8
Computational time (s)	100.42	3.55	2.06

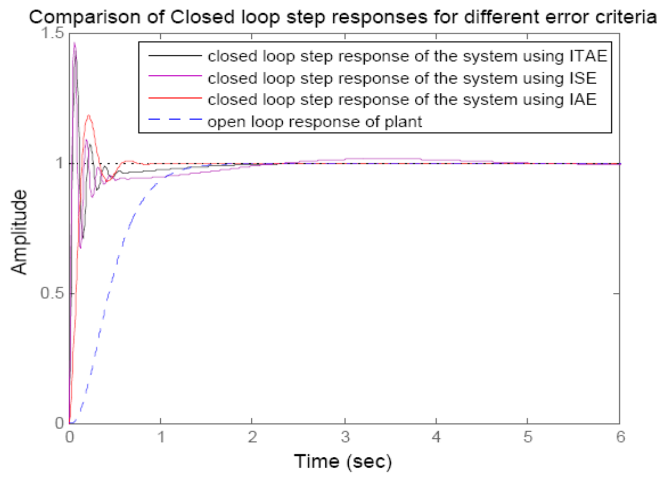

Fig. 6. Comparison of closed loop step responses for AGA using different error criteria for the illustrative example 1.

TABLE VII. Comparison Of Step Response For Example 1 Using Different Error Criteria.

	ITAE	ISE	IAE	Open loop plant
Settling time (s)	1.18	1.8	0.538	1.29

Rise time (s)	0.077	0.0258	0.101	0.723
Overshoot (%)	43.1	46.4	18	0

VII. CONCLUSION

We have implemented Advanced Genetic Algorithm (AGA) along with the existing genetic algorithms - Guo Tao's Algorithm (GTA) and Elite Multi-Parent Crossover Optimization Algorithm (EMPCOA). From the results presented it is observed that AGA requires lesser computational time as compared to GTA and EMPCOA. This is mainly because of the different replacement strategy in AGA which improves the population diversity and hence lesser number of iterations isrequired. Also from the graph it can be seen that the transient response of the system using AGA is better than that of GTA and EMPCOA. The closed loop step response of the system has a better settling time and maximum overshoot.

To prove the effectiveness of the algorithm, it is tested on an unstable system. From the results presented it is observed that it requires lesser number of iterations and a better transient response due to fast settling time and lesser overshoot.

Further, the PID controller is tuned with different error criteria. The performances of the controllers tuned using different error criteria are compared.

REFERENCES

[1] K. S. Tang,Kim Fung Man, Guanrong Chen and Sam Kwong, "An optimal fuzzy PID ontroller", IEEE Transactions On Industrial Electronics, vol 48, no 4, pp. 757-765, 2001

[2] W.K. Ho, O.P.Gan,E.B. Tay and E.L.Ang, "Performance and gain and phase margins of well-known PID tuning formulas", IEEE Transactions on Control Systems Technology, vol. 4, no. 4, pp. 473–477, 1996.

[3] Aidan O'Dwyer, "PI and PID controller tuning rules: an overview and personal perspective," Proceedings of the IET Irish Signals and Systems Conference, 2006c

[4] J.H.Holland, "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence", Ann Arbor, MI: University of Michigan Press 1975, Second edition, Cambridge, MA: The MIT Press, 1992.

[5] Teo Lian Seng, Marzuki Bin Khalid and Rubiyah Yusof,"Tuning of a Neuro-Fuzzy," IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, 1999

[6] Kiam Heong Ang, Gregory Chong and Yun Li, "PID control system analysis, design and technology", IEEE Transactions on Control Systems Technology, vol 13, no 4, pp. 555-576, 2005.

[7] Dazhi Jiang, Yulin Du, Jiali Lin, Zhijian Wu,"Multi-parent crossover algorithm with discrete recombination,"IEEE, 2010

[8] R. Michalski, J. Carbonell and T. Mitchell, "machine learning: an artificial intelligence approach", Los Altos, Morgan Kaufmann, 1986.

[9] Guo Tao and Kang Li-Shan, "A new evolutionary algorithm for function optimization", Wuhan University Journal of Natural Sciences, vol.4, no.4, pp.409- 414, 1999.

[10] Xiaoyi Che, Youxin Luo and Zhaoguo Chen, "Optimization for PID control parameters on hydraulic servo control system based on the elite multi-parent crossover evolutionary algorithm" , International Conference on Measuring Technology and Mechatronics Automation, IEEE Computer Society, pp. 845- 848, 2010.

[11] B.Nagaraj, P.Vijayakumar, "A comparative study of PID controller tuning using GA, EP, PSO and ACO," Jounal of Automation , Mobile Robotics and Intelligent systems, vol 5, no.2, February 2011