Volume 3, Issue 7, July 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Speed Control of DC Motor Using Genetic Algorithm Based PID Controller

Megha Jaiswal , Mohna Phadnis (H.O.D. EX)
Electrical & Electronics Department
Malwa Institute of Technology ,Indore
R.G.P.V., Bhopal, India

Abstract- The tuning aspect of proportional–integral-derivative (PID) controllers is a challenge for researchers and plant operators. This paper proposes the tuning of PID controller of a DC motor using genetic Algorithm .Genetic algorithm is a soft computing technique which is used for optimization of PID parameters. The Algorithm functions on three basic genetic operators of selection, crossover and mutation. Based on the types of these operators GA has many variants like Real coded GA, Binary coded GA, These parameters have a great influence on the stability and performance of the control system. This Paper focuses the Binary coded GA & find the value of crossover , mutation of PID controller

Keywords- DC motor, Genetic algorithm, Crossover, Mutation, PID controller.

I. Introducion

Due to its excellent speed control characteristics, the DC motor has been widely used in industry even though its maintenance costs are higher than the induction motor . As a result, Speed control of DC motor has attracted considerable research and several methods have evolved. Proportional-Integral Derivative (PID) controllers have been widely used for speed and position control of DC motor. [1] This paper proposes a new method to design a speed controller of a DC motor by selection of PID parameters using GA To show the efficiency of GA. the results of this method are compared with simple PID Controller .Using genetic algorithms to perform the tuning of the controller results in the optimum controller being evaluated for the system every time. The objective of this paper is to show the Binary coded GA & find the value of crossover , mutation of PID controller .

II. Model of Dc Motor

As reference we consider a DC shunt motors as is shown in figure 1. DC shunt motors have the field coil in parallel (shunt) with the armature. The current in the field coil and the armature are independent of one another. As a result, these motors have excellent speed and position control. Hence DC shunt motors are typically used applications that require five or more horse power. The equations describing the dynamic behavior of the DC motor are given by the following equations;[1]

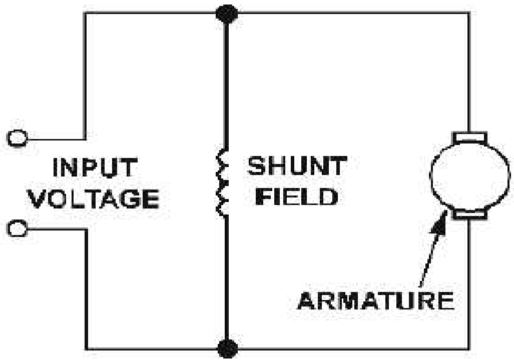

Fig.1. Diagram of DC shunt motor

$$V = R_i + L\frac{di}{dt} + e_b \quad (1)$$

$$T_m = K_t\, i \quad (2)$$

$$T_m = J\frac{d^2\theta}{dt^2} + B\frac{d\theta}{dt} \quad (3)$$

$$e_b = K_b\,\frac{d\theta}{dt} \quad (4)$$

$$\omega = \frac{d\theta}{dt} \quad (5)$$

After simplification and taking the ratio of ω(s)/v(s) we will get the transfer function as below:

$$\frac{\omega(s)}{V(s)} = \frac{K_b}{(Js+B)(Ls+R)+{K_b}^2+RB} \quad (6)$$

- R: Armature resistance in ohm
- L: Armature inductance in henry
- i: Armature current in ampere
- V: Armature voltage in volts
- e_b: Back emf voltage in volts
- K_b: Back emf constant in volt/ (rad/sec)
- K_T: Torque constant in N.m/Ampere
- T_m: Torque developed by the motor in N.m
- θ (t): Angular displacement of shaft in radians
- J: Moment of inertia of motor and load in Kg.m2/rad
- B: Frictional constant of motor and load in N.m/ (rad/sec)[2]

A. Numerical Values

The DC motor under study has the following specifications and parameters.

1) Specifications

2hp, 230 volts, 8.5 amperes, 1500rpm

2) *Parameters:*

Ra=0.45 ohm, La=0.035 H, Kb=0.5 volt/(rad/sec), J=0.022Kg-m2/rad, B=0.2*10^-3N-/(rad/sec).

The overall transfer function of the system is given below,

$$\frac{\omega(s)}{V(s)} = \frac{0.5}{0.0077s^2 + 0.09007s + 0.25018}$$

III. Conventional Pid Controller

A. Fundamentals of PID controller

PID controllers are the most widely-used type of controller for industrial applications. They are structurally simple and exhibit robust performance over a wide range of operating conditions. In the absence of the complete knowledge of the process these types of controllers are the most efficient of choices. The three main parameters involved are Proportional (P), Integral (I) and Derivative (D). The proportional part is responsible for following the desired set-point, while the integral and derivative part account for the accumulation of past errors and the rate of change of error in the process respectively.

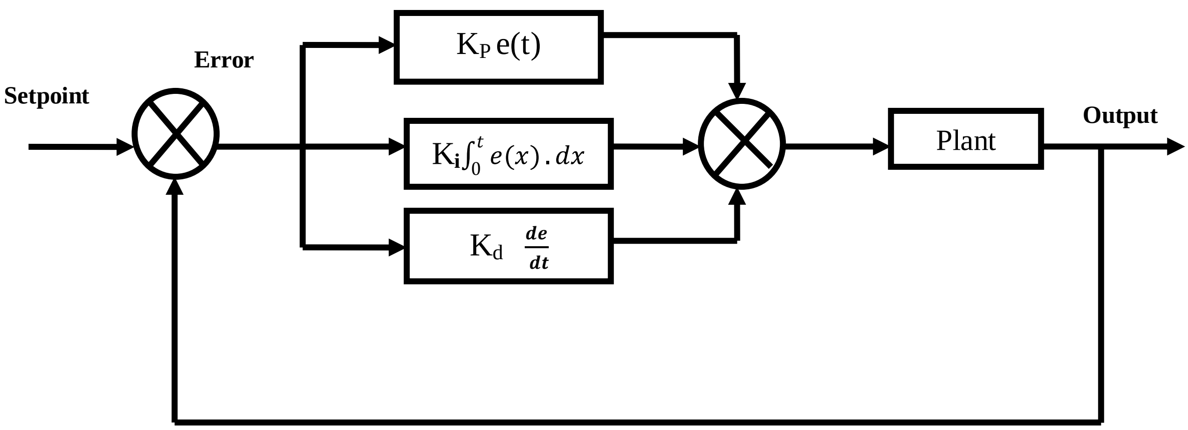

Fig.2 Block diagram of a conventional PID controller

For the PID controller presented in Fig. 2,
Output of the PID controller,

$$u(t) = K_p e(t) + K_i \int_0^t e(x).dx + K_d \frac{d\,e(t)}{dt} \qquad (7)$$

Where,
Error, e(t) =Set point- Plant output
K_p=proportional gain
K_i= integral gain
K_d = derivative gain

A. Analysis of conventionally tuned PID Controller without Genetic Algorithm

Here we are using DC motor with PID Controller without G.A. By the use of matlab programming PID controller is tuned & the result is given below

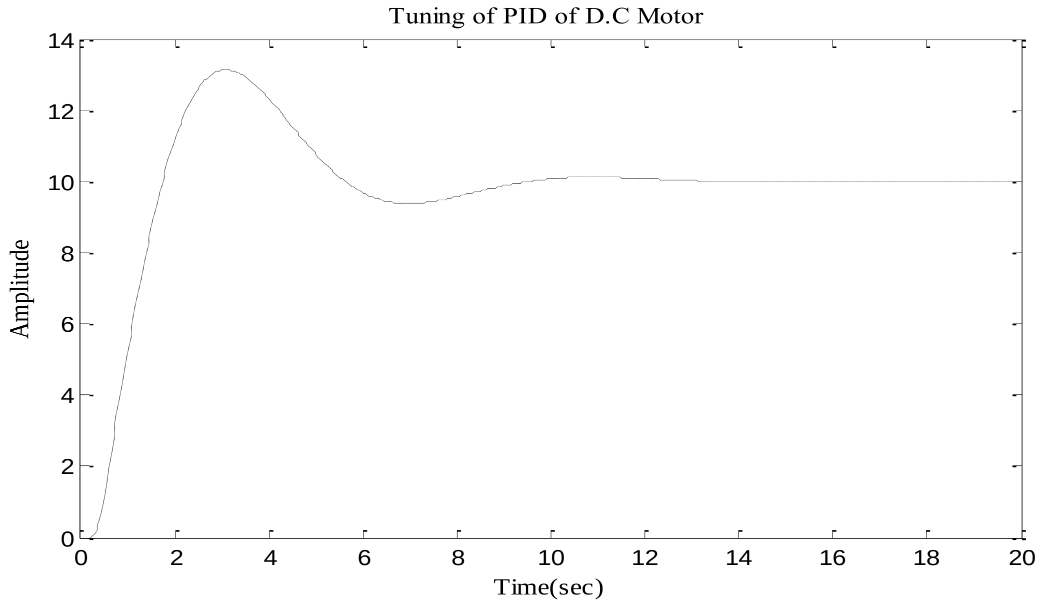

Fig.3 result without G.A based PID controller

From the above response ,we can analyze the system . we can analyze the following parameter*:*

- Rise Time, t_r
- Maximum Overshoot, M_p
- Settling time, t_s

The rise time, t_r is the time taken to reach 10 to 90 % of the final value is about 0.5 sec. The Maximum Overshoot, M_p of the system is approximately 1.01. Finally the Settling time, t_s is about 0.25sec. From the analysis above, the system has not been tuned to its optimum. So in order to achieve the following parameters we have to go for genetic algorithm approach. Our system requirements are given below,

TABLE 1
System requirements

Specification	M_p	t_r	t_s
	<1.01	< 0.5	<0.25sec

IV. Pid Controller Using Genetic Algorithm

A. Overview of Genetic Algorithm

GA is a stochastic global adaptive search optimization technique based on the mechanisms of natural selection. Recently, GA has been recognized as an effective and efficient technique to solve optimization problems. Compared with other optimization techniques. GA starts with an initial population containing a number of chromosomes where each one represents a solution of the problem which performance is evaluated by a fitness function. Basically, GA consists of three main stages: Selection, Crossover and Mutation. The application of these three basic operations allows the creation of new individuals which may be better than their parents. This algorithm is repeated for many generations and finally stops when reaching individuals that represent the optimum solution to the problem. The GA architecture is shown in Fig.4

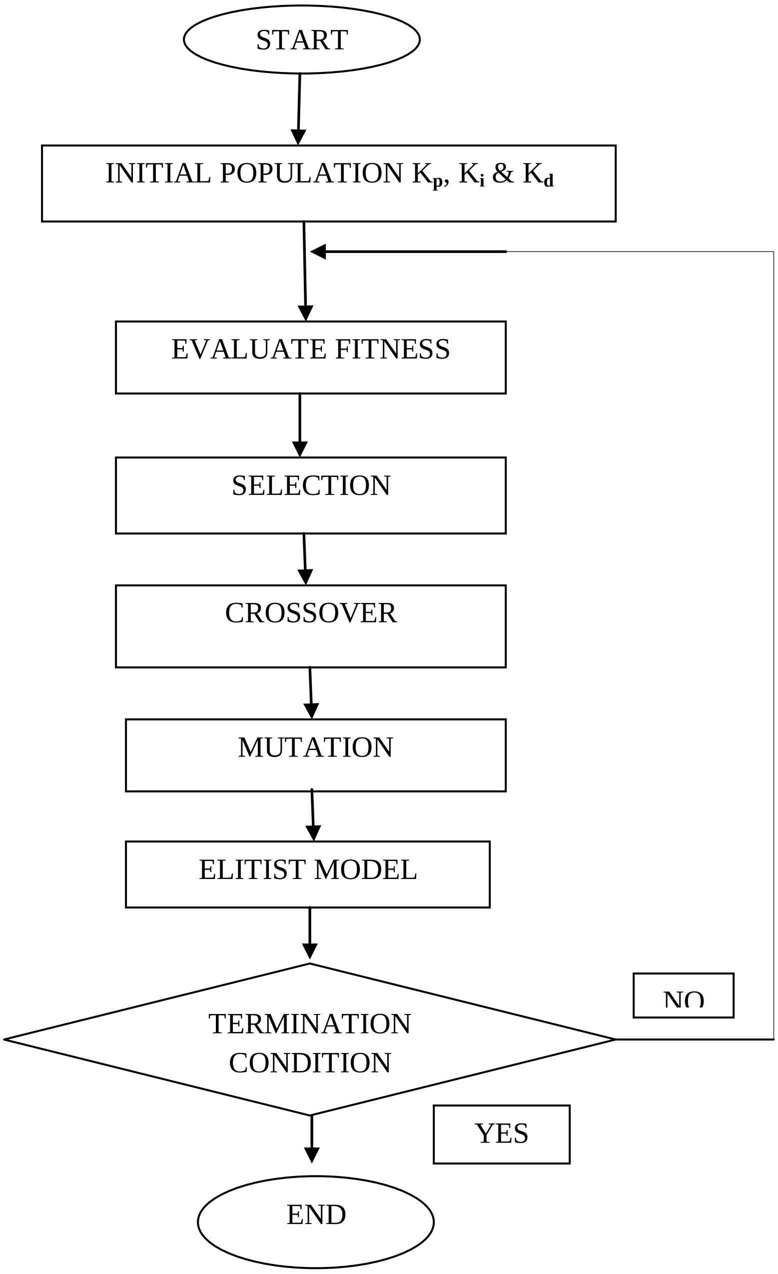

Fig.4 Flow Chart of Genetic Algorithm

B. Objective Function of the Genetic Algorithm

The most challenging part of creating a genetic algorithm is writing the objective functions. In this project, the objective function is required to evaluate the best PID controller for the system. An objective function could be created to find a PID controller that gives the smallest overshoot, fastest rise time or quickest settling time. However in order to combine all of these objectives an objective function is designed to minimize the performance indices of the controlled system instead. [4]

C. Overview of Binary Coded G.A

GA has many variants like Real coded GA, Binary coded GA, Saw tooth GA, Micro GA, Improved GA, Differential Evolution GA. This paper is based on Binary coded G.A.

The binary coded *genetic algorithm* is a probabilistic search algorithm that iteratively transforms a set (called a *population*) of mathematical objects (typically fixed-length binary character strings), each with an associated fitness value, into a new population of offspring objects using the Darwinian principle of natural selection and using operations that are patterned after naturally occurring genetic operations, such as crossover and mutation.[5]

1) Encoding

In genetic Algorithm ,coding is expressing the individual by the binary strings of 0's & 1's . In the instance one every individual has there dimension and every dimension is expressed by a 10- bit string of 0's & 1's.[6]

2) Selection

The selection operator selects chromosomes from the current generation to be parents for the next generation. In this method, a few good chromosomes are used for creating new offspring in every iteration. Then some bad chromosomes are removed and the new offspring is placed in their places. The rest of population migrates to the next generation without going through selection process.

3) Crossover

Crossover is the GA's primary local search routine. The crossover/reproduction operator computes two offspring for each parent pair given from the selection operator. The crossover operator is used to create new solutions from the existing solutions available in the mating pool after applying selection operator. This operator exchanges the gene information between the solutions in the mating pool. The most popular crossover selects any two solutions strings randomly from the mating pool and some portion of the strings is exchanged between the strings. The selection point is selected randomly. A probability of crossover is also introduced in order to give freedom to an individual solution string to determine whether the solution would go for crossover or not.

4) Mutation

Mutations are global searches. A probability of mutation is again predetermined before the algorithm is started which is applied to each individual bit of each offspring chromosome to determine if it is to be inverted. Mutation changes the structure of the string by changing the value of a bit chosen at random.[7] Mutation is the occasional introduction of new features in to the solution strings of the population pool to maintain diversity in the population. Though crossover has the main responsibility to search for the optimal solution, mutation is also used for this purpose. Mutation operator changes a 1 to 0 or vise versa, with a mutation probability of .The mutation probability is generally kept low for steady convergence. A high value of mutation probability would search here and there like a random search technique.

V. Proposed Work

A. Implementation of G.A based PID Controller

In the proposed work a DC Motor model is called by a program which is coded in Matlab for a fitness function i.e cost function. In order to use GA to tune the PID controller for DC motor. Variables K_p, K_i, & K_d are coded to solve string structures. Binary coded string having 1's & 0's are mostly used. The length of string is usually determined according to the desired solution accuracy. Here 10 bits are used to code each variable. We can use 8 bit & 4 bit also. Thereafter select the random strings from the population to form the mating pool. In order to use roulette-wheel selection procedure, we calculate the average fitness of the population. Then the mating pool strings are used in the crossover operation. The next step is to perform mutation on strings in the intermediate population . The resulting population becomes the new population. The whole process is coded in matlab & after running the program we get the optimized values of K_p, K_i & K_d . The simulation modal for the entire system is given below and also the genetic algorithm parameters are chosen for the optimization.

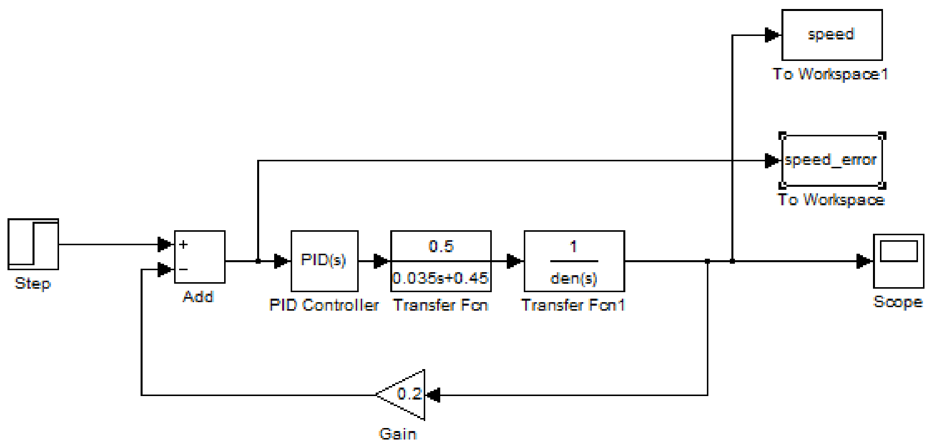

Fig:5 simulation modal of GA based PID controller

TABLE 2
G.A. PARAMETERS

Parameter	Value
Population size	20
Iteration	05
Crossover Probability	> 0.8
Mutation Probability	< 0.05

From the above table all the parameter values are apply to G.A the PID controller with DC motor is optimized . The system response is given below.

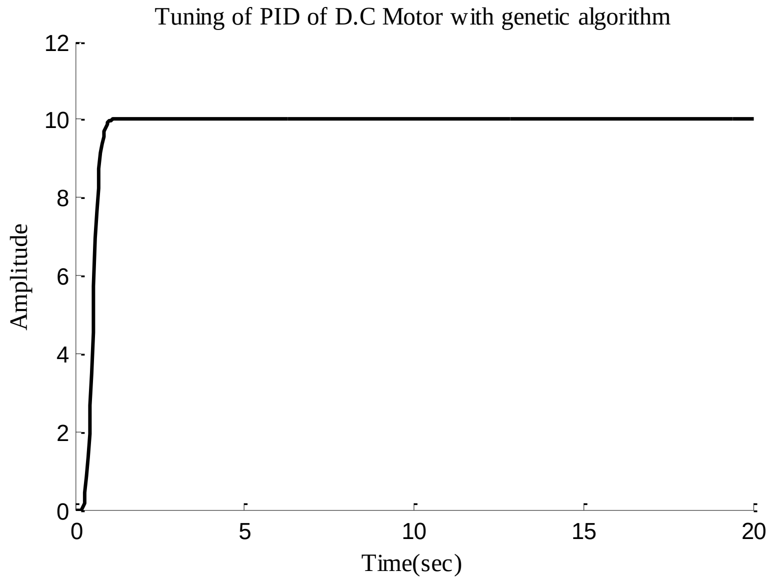

Fig.6 Result of G.A based PID controller

TABLE 3
G.A. PARAMETERS

Parameter value	M_p	t_r	t_s
PID with G.A	0.004	0.303	0.943

VI. Analysis Of Result

It is clear from both results that the simple PID controller is not getting the accurate results but the G.A based PID controller getting the proper optimized gain values of K_P, K_i and K_d. Below fig. shows the comparison of both the result.

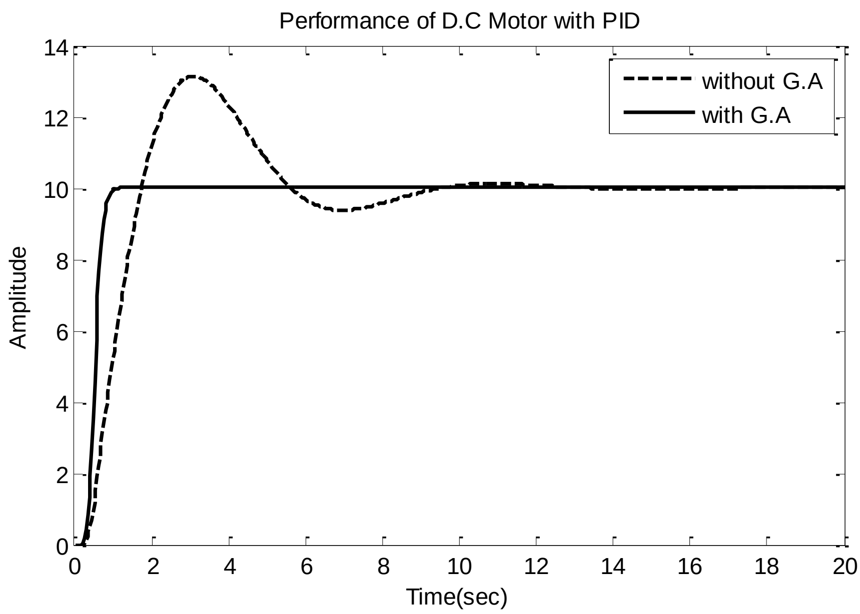

Fig.7 *Comparative results*

TABLE II

COMPARISON OF PARAMETER WITH G.A & WITHOUT G.A

Parameter	k_p	K_i	K_d	T_r	T_s	M_p
Without G.A	1	1	0.100	1.0603	8.66	3.13
With G.A	6.520	0.008	0.303	0.3773	0.9433	0.004

VII. Conclusion

Research work has been carried out to get an optimal PID tuning by using GA. This paper provides the complete original binary coded G.A program in mat lab, which can be directly run through Mat lab 7.10. G.A is applied to find optimal solution for the parameter of DC motor with PID controller & indicates that G.A is powerful global searching method. The G.A designed PID controller is much better in terms of rise time, settling time, overshoot then simple PID controller. In the future Binary coded G.A. is implement to other system of the plant for the speed control by using PID controller.

Refrences

[1] Neenu Thomas, Dr.P.Poongodi, 2009. Position Control of DC Motor Using Genetic Algorithm Based PID Controller,Proceedings of the World Congress on Engineering 2009 Vol II WCE 2009, London, U.K.

[2] T. O..Mahony, C J Downing and K Fatla, "Genetic Algorithm for PID Parameter Optimization: Minimizing Error Criteria", Process Control and Instrumentation 2000 26-28 July 2000, University of Stracthclyde, pg 148- 153.

[3] K Ogata, Modern Control Systems, University of Minnesota, Prentice Hall, 1987

[4] B.Nagaraj ,Dr.N.Murugananth" A Comparative Study of PID Controller Thning Using GA, EP, PSO and ACO" ICCCCT-10 IEEE Trans

[5] Phillip David Power, "Non Linear Multi Layer Perceptron Channel Equalisation",Chapter 4 „Genetic Algorithm Optimisation", *in IEEE Transactions,*The Queen University of Belfast,April 2001.

[6] Kalyanmoy Deb, Optimization for Engineering Design Algorithm & Examples ,Indian Institute Of Technology Kanpur, Prentice Hall, 2005.

[7] Liu Fan, Er Meng Joo" Design for Auto-tuning PID Controller Based on Genetic Algorithms" Nanyang Technological University Singapore IEEE Trans on ICIEA 2009

[8] Grefenstette, Optimization of control parameters for geneticalgorithms. IEEE Trans on SMC, 1986, 16(1):122-128.

[9] ASTROM K.J., HAGGLUND T.: 'The future of PID control', Control Eng. Pract., 2001, 9, (11), pp. 1163–1175.

[10] B. C. Kuo, "Automatic Control Systems," 5th ed. Englewood Cliffs Prentics-Hall, 1987.

[11] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," Trans. ASME, vol. 64, pp. 759-768, 1942.

[12] M. Ge, M. S. Chiu and Q. G. Wang, "Robust PID controller design via LMI approach," Journal of Process Control, vol. 12, no.1, pp. 3-13, 2002

[13] T. O..Mahony, C J Downing and K Fatla, "Genetic Algorithm for PID Parameter Optimization: Minimizing Error Criteria", Process Control and Instrumentation 2000 26-28 July 2000, University of Stracthclyde, pg 148- 153

