UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF NATURAL &
APPLIED SCIENCES

PID CONTROLLER TUNING WITH GENETIC ALGORITHMS: POSITION CONTROL OF PLANAR MECHANISMS

M. Sc. THESIS
IN
MECHANICAL ENGINEERING

BY
SERKAN AL
JANUARY 2009

PID Controller Tuning with Genetic Algorithms: Position Control of Planar Mechanisms

M.Sc. Thesis
in
Mechanical Engineering
University of Gaziantep

Supervisor
Prof. Dr. L. Canan DÜLGER

by

Serkan AL

January 2009

T.C.
GAZİANTEP UNIVERSITY
GRADUATE SCHOOL OF
NATURAL & APPLIED SCIENCES
MECHANICAL ENGINEERING

Name of the thesis: PID Controller Tuning with Genetic Algorithms, Position Control of Planar Mechanisms

Name of the student: Serkan AL

Exam date: 12.01.2009

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Ramazan Koç
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Sedat BAYSEÇ
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. L.Canan Dülger
Supervisor

Examining Committee Members

Prof. Dr. Sedat BAYSEÇ ______________________

Prof. Dr. L. Canan DÜLGER ______________________

Prof. Dr. Sadettin KAPUCU ______________________

Prof. Dr. İbrahim GÜZELBEY ______________________

Assoc. Prof. Dr. Mehmet TOPALBEKİROĞLU ______________________

ABSTRACT

PID CONTROLLER TUNING WITH GENETIC ALGORITHMS: POSITION CONTROL OF PLANAR MECHANISMS

AL Serkan
M.Sc in Mechanical Engineering
Supervisor: Prof.Dr.L.Canan DÜLGER
January 2009, 74 pages

Genetic Algorithms (GAs) are adaptive methods based on the genetic processes of biological organisms. GAs are included in family of computational models inspired by evolution. Three different phases of search happens in GA; an initial population is created, a fitness function is evaluated and a new population is produced. Three operators are available; reproduction, crossover, mutation.

In this study, GAs are used to determine Proportional-Integral-Derivative (PID) controller parameters. The results are implemented to position control of planar mechanisms. MATLAB Genetic algorithm Toolbox is used during the study. Two example applications are performed; four bar mechanism (SISO) system and the hybrid actuator (MISO) system. Series of numerical simulations are carried out on the mathematical models inclusion of the actuator dynamics; Brushless dc servo motor (BLDC) and permanent magnet dc motor (PMDC) are taken in the application and satisfactory results are obtained in position control of Planar Mechanisms.

Keywords: Genetic algorithms, PID tuning, hybrid actuator, motion control, four bar mechanism

ÖZET

GENETİK ALGORİTMALAR İLE PID DENETLEYİCİ AYARI: DÜZLEMSEL MEKANİZMALARIN KONUM DENETİMİ

AL Serkan
Yüksek Lisans Tezi Makina Mühendisliği
Danışman: Prof.Dr.L.Canan DÜLGER
Ocak 2009, 74 sayfa

Genetik Algoritma (GA) biyolojik organizmalarda rastsal mutasyon ve doğal seçime dayanan bir yöntemdir. GA evrim modelinden esinlenerek ortaya konulmuş bir hesaplama tabanlı algoritmadır. GA da üç farklı faz bulunur; önce başlangıç popülasyonu oluşturulur, bir uygunluk değeri yaratılır ve yeni bir populasyon oluşturulur. Üç genetik operatör kullanılır; çoğalma, çaprazlama ve mutasyon.

Çalışmada Genetik algoritmalar Oransal-Integral-Türevsel (PID) denetleyici parametrelerini bulmak için kullanılmıştır. Sonuçlar daha sonra düzlemsel mekanizmaların konum denetimine uygulanmıştır. Bu çalışmada MATLAB yazılımında Genetic Algoritma Araç Kutusu kullanılmıştır. İki örnek mekanizma; dört çubuk mekanizması (TGTÇ-tek giriş tek çıkış) ve hybrid eyleyici (İGTÇ-iki giriş tek çıkış) üzerinde elde edilen benzetim sonuçları sunulmuştur. Modellerde motor dinamiği de eklenmiş olup, fırçasız doğru akım servo motor (BLDC) ve sabit mıknatıslı doğru akım servo motor (PMDC) kullanılmış seçilen düzlemsel mekanizmaların konum denetiminde tatmin edici sonuçlar alınmıştır.

Anahtar sözcükler: Genetik algoritmalar, PID denetleyici ayarı, hibrid eyleyici, hareket denetimi, dört çubuk mekanizması

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Prof. Dr. L. Canan Dülger, for her encouragement, guidance, help and suggestions throughout the course of this study.

I would like to thank Research Assistant in Mechanical Engineering Department Ali Kılıç for his help and support throughout the course of this study.

Special thanks to my family for their motivation, support and encouragement during the preparation of this thesis.

LIST OF CONTENTS

ABSTRACT iii
ÖZET iv
ACKNOWLEDGEMENTS v
LIST OF CONTENTS vi
LIST OF TABLES viii
LIST OF FIGURES ix

CHAPTER 1 INTRODUCTION
1.1 Introduction 1
1.2 Research Objectives 2
1.3 Motion Control 2
1.6 Outlines of Thesis 3

CHAPTER 2 LITERATURE SURVEY
2.1 Introduction 4
2.2 Genetic Algorithms in Control Engineering 4
2.3 Literature review on Tuning PID Controllers 5
2.4 Characteristics of Genetic Algorithm 7
2.5 Operation of Genetic Algorithm 9
2.6 Objective Function and Fitness Function 15
2.7 Summary on Genetic Algorithm 16

CHAPTER 3 GENETIC ALGORITHM FOR PID TUNING
3.1 Introduction 18
3.2 PID Controller 18
3.3 PID Tuning Methods 20

3.4 PID Tuning with Genetic Algorithms 21
3.5 Genetic Algorithm Toolbox 22
3.5.1 Initializing the Genetic Algorithm 22
3.5.2 Setting GA Parameters 24
3.5.3 Performing GA 26
3.5.4 Objective Function of Genetic Algorithm 26
3.6 Application Examples of PID with GA 30
3.6.1 PID Controller for BLDC motor 30
3.6.1.1 Discussions on PID Tuning for BLDC motor 35
3.6.2 PID Controller for PMDC motor 38
3.6.2.1 Discussions on PID Tuning for PMDC motor 41

CHAPTER 4 POSITION CONTROL OF PLANAR MECHANISMS
4.1 Introduction 45
4.2 Four Bar Mechanism 45
4.2.1 Control Structure of Four Bar Mechanism 47
4.2.2 Mathematical Structure of Four bar Mechanism 47
4.2.3 Application of GA to PID Controller 50
4.3 GA based Results-Four Bar Mechanism 51
4.4 The Hybrid Actuator 54
4.4.1 Control Structure for the Hybrid Actuator 55
4.4.2 Mathematical Structure of Hybrid Actuator 57
4.4.2.1 Seven link mechanism 57
4.4.2.2 The System Equations 59
4.4.2.3 Application of GAs to PID Controller 61
4.5 GA based Results - The hybrid actuator 62
4.6 Discussions on position control applications 66

CHAPTER 5 CONCLUSIONS
5.1 Discussions on the Present study 67
5.2. Suggestions for further work 70

REFERENCES 71

LIST OF TABLES

Table 3.1 Brushless Servo Motor Data 31
Table 3.2 Parameters of GA for BLDC motor 32
Table 3.3 GA based PID parameters for BLDC motor (Direct Drive) 33
Table 3.4 GA based PID parameters for BLDC motor (Hybrid Actuator) 36
Table 3.5 PM Servo Motor Data 38
Table 3.6 Parameters of GA for PMDC motor 39
Table 3.7 GA based PID parameters for PMDC motor (Direct Drive) 40
Table 3.8 GA based PID parameters for PMDC motor (Hybrid Actuator) 43
Table 4.1 Four Bar Mechanism Parameters 49
Table 4.2 The Parameters of GA-Four Bar Mechanism 51
Table 4.3 Mechanical Properties of Seven Link Mechanism 62
Table 4.4 The Parameters of Genetic Algorithm 63

LIST OF FIGURES

Figure 2.1 Roulette Wheel Selection 12
Figure 2.2 Flow Diagram of Genetic Algorithm 17
Figure 3.1 Block Diagram of PID 19
Figure 3.2 Initializing GA 22
Figure 3.3 Setting the parameters of GA 24
Figure 3.4 Performing the Genetic Algorithm codes 26
Figure 3.5 Stability of the System according to the MSE error criteria 29
Figure 3.6 GA based closed loop response of BLDC motor 34
Figure 3.7 GA based response and PID parameters for Hybrid Configuration 37
Figure 3.8 GA based closed loop response of PMDC motor 42
Figure 3.9 GA based results for PMDC motor- Hybrid Configuration 44
Figure 4.1 (a) An assembly view-Four Bar Mechanism 46
Figure 4.1 (b) A top view-Four Bar Mechanism 46
Figure 4.2 GA based PID Controller-Four Bar Mechanism 47
Figure 4.3 Four Bar Mechanism and its parameters 48
Figure 4.4 PID Controller Gains-Four Bar Mechanism/ 10 runs 52
Figure 4.5 (a) GA based Response and Coupler Curves-The Crank Motion 53
Figure 4.5 (b) GA based Response and Coupler Curves-Coupler Points (X) 53
Figure 4.5 (c) GA based Response and Coupler Curves-Coupler Points (Y) 54
Figure 4.6 (a) An assembly view of Hybrid Actuator 56
Figure 4.6 (b) A Rotated (Top) view of Hybrid Actuator 56
Figure 4.7 Structure of Hybrid Actuator Control 57
Figure 4.8 Seven link mechanism referred 58
Figure 4.9 PID controller gains for Hybrid Actuator/ 10 different runs 63
Figure 4.10 (a) GA based BLDC motor response 64
Figure 4.10 (b) GA based PMDC motor response 65
Figure 4.10 (c) GA based Hybrid Output 65

CHAPTER 1
INTRODUCTION

1.1. Introduction

Today’s difficult problems make people to focus on fast and easy solution methods, some of them are based on evolutionary algorithms. Especially in the last years, soft computing techniques and the evolutionary algorithms are used instead of hard optimization techniques. Several algorithms have been applied in the literature under the group of evolutionary algorithms (EA) such as genetic algorithms (GA); the one based on Darwin’s Theory of Evolution, the ones based on swarm intelligence as Particle Swarm Optimization (PSO), Bees Algorithm (BA) and Ant Colony Optimization (ACO). [1]

GA is one of Evolutionary Algorithm, used in different type of applications. GA is a search technique applied in computing to get the exact or approximate solutions to optimization search problems. It is a stochastic global search method that utilizies the process of Natural Evolution used by Evolutionary Biology such as Mutation, Selection and Crossover.

The first research about Genetic Algorithms is made by John Holland in Michigan University in the department of Physiology and Computer science in the 1970’s, and is based on Darwin’s Evolution theory. Genetic Algorithms are used for function optimization, tabulation, mechanical learning, design applications and cellular production. Some research have been done for Genetic transfer and Genetic changes in Evolutionary animates to the Mechanical and Computer base in the Evolutionary theory. Learning cannot be done immediately and learning happens from old generations to new ones. The information and good properties of old generations are transferred to the new ones by this method and learning can also be done by this method. The first book about GA is published in 1975; the method is named as 'Genetic Algorithm'. GA started to use in every facet of Industrial applications. Since then many research studies have been explicitly performed world-wide for different problems. GA principles are well explained in many textbooks [1-4].

1.2. Research Objectives

In this study, main concentration is done on industrial basis. Position Control of Planar Mechanisms is considered. Two examples are chosen; a four bar mechanism and the hybrid actuator with included actuator dynamics. Some recent examples on GAs are performed off-line and on-line control applications, but the most popular ones are seen in off-line implementation. Some other examples on GA are counted as optimal controller tuning, robust stability analysis, system identification and classifier systems [5,6]. Here GA has been applied to controller design, especially for tuning PID parameters in off-line use.

The study of GA is largely dependent on numerical simulation. There are GA packages in the market. In this study, GA has been used as a population based optimizer for tuning PID controller parameters. GA Toolbox developed for MATLAB is taken and the numerical simulations are performed using its graphical facility. Optimum controller parameters are found and then used in numerical simulations based on the systems dynamic model with actuators. Simulations are compared to conventional PID with the system response curves.

1.3. Motion Control

In Industrial applications, to obtain high performance position control servo systems are being employed. Motors and their structures used in Industrial applications are different. Servomotors are mostly used in the systems which have high power, high moment and rapid reaction. Soft control of the motor in servo systems is easy when the motor is starting and stopping. DC servomotors are easy to control using adjustable DC voltage. Brushless DC (BLDC) and Permanent magnet DC (PMDC) servo motors are used in this application.

PMDC motors require commutator and brushes so periodical maintenance is needed such as changing brushes and resurfacing the commutator. These motors give a constant flux density and the speed is directly proportional to the applied voltage. PMDC motor limitation can be eliminated by replacing the mechanical commutator with an electrical alternative. Whenever low maintenance is essential, BLDC motors are preferred. BLDC motor consists of a permanent magnet rotor, a stator with three phases and a rotor position sensor. So lower rotor inertias, higher rotor speeds are resulted in compared to conventional ones. These motors are quiet and capable of running at higher speeds. They are widely used in robotics and machine tools. Both motors have similar structures mathematically. They are studied in motion control here. [6, 9]

1.4 Outlines of Thesis

This study is targeted to show use of Genetic Algorithms in control system applications. Chapter 1 gives research objectives and a literature survey on GAs in Control Engineering and Thesis Structure. Chapter 2 presents Genetic Algorithms, their applications with basic concepts. GAs for PID tuning is included in Chapter 3 together with application examples using GA Toolbox in MATLAB. Chapter 4 has presented position control of planar mechanisms. Two specific examples are chosen as a four bar mechanism and the hybrid actuator representing SISO and MISO systems respectively. System dynamic equations are numerically solved and GA based responses are presented for both examples. Finally discussions on the present work and suggestions for further study are included in Chapter 5.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

Genetic Algorithm is a method for solving optimization problems that is based on Natural Selection. GA is a stochastic global search method to get the correct result using Evolution operators, the process of Biological Evolution. John Holland has formally introduced this method in the United States in the 1970 [1,2, 21, 22]. GA starts without any knowledge of the best and correct solution and depends fully on responses from its environment. GA uses evolution operators such as reproduction, crossover and mutation to arrive at the best solution. It avoids local minima and converging to sub optimal solutions by searching in parallel and starting at several independent points. GA is capable of solving difficult problems to every facet of Industrial applications.

This chapter includes a literature survey on Genetic Algorithms in Control Engineering and its applications on PID controller tuning. A brief introduction on GA is included. Characteristics and operation of GA is given as separate sections. Objective function is used to evaluate the fitness of each chromosome. The details on its calculation and use are also included here in.

2.2 Genetic Algorithms in Control Engineering

The number of GA applications is continuously growing. Today numerous problems have been solved by using GAs commonly in industry. GA has been successfully implemented in many areas such as parameter and system identification, robotics, pattern recognition, speech recognition, engineering designs, planning and scheduling. In some studies, GA is integrated with other Artificial Intelligence Techniques either in neural network (NN) or in fuzzy logic (FL) [3].

GAs has been applied to the off-line design of controllers, such as PID since 1990's. Many successful studies were noted on its ability for parameter optimization. Some of the studies performed in last 10 years are taken into interest and used as references throughout the study. Studies are especially chosen on the base of PID controllers and DC electric drives in use. Recent applications of GAs in the control engineering are found in many studies.

Fleming and Conseca [7] have presented a study on GAs in control systems engineering. In the study a simple GA is presented with off-line design and on-line adaptation. GA application is given in terms of benefit to control with available tools. On-line applications are rare since the difficulties are seen in real time use. Fleming and Pursue [8] have reported a detailed study in IFAC Professional Brief on GA in control systems. Several GA toolboxes have been developed. The technical computing package MATLAB is the one commonly used. GA application for design; controllers with parameter optimization, structure optimization, and also fuzzy/neural control for complicated dynamic systems are studied. Later Wang et al [10] have presented an overview on GAs applied to control engineering problems. The use of GAs for specific control problems; multi objective control, optimal control, robust control, adaptive control, system identification, Hinfinity control, Eigen-structure assignment, Lyapunov's direct control, LQG control, stochastic robustness control, sliding mode control and tuning PID parameters. Later complex dynamic systems are studied with Fuzzy Logic Control (FLC), neural network control (NNC), also for online optimization.

2.3 Literature Review on Tuning PID Controllers

Genetic algorithms are inspired from the genetic evolution of mechanisms, and GAs have been applied successfully for solving problems in many engineering applications using a direct analogy of natural evolution since 1970's. Many different research studies are found on determination of PID controller parameters by genetic algorithms. Successful implementations are found in these studies. Some studies are noted for recent applications of GAs in control. They have been reported here and taken as references and used throughout this study. Man et al [3] have presented a detailed study on GAs as concept and applications. The collection of practical systems is introduced by using GA. Integration of NNs and fuzzy systems are explained.

A number of successful applications have also been reported in GA area. GAs have been addressed in parameter and system identification, control to improve the overall system performance, robotics in navigation, pattern recognition, speech recognition, engineering designs, planning and scheduling. Applying GA in time varying system, problems seen are mentioned. Examples are included mostly in industrial basis. Fleming and Pursue have reported these issues in a brief covering studies between 1991-2000.

Krohling et all [11] have designed PI/PID controllers for a motion control system based on genetic algorithms with a brief review. Optimal PI/PID parameters are found by minimization of the integral time multiplied squared error (ITSE) in the frequency domain with AC servo motor. Al-Hamouz et al [12] have presented a study on an application of variable structure controller (VSC) for DC motor with GAs. A procedure for the variable structure controller design is proposed. GAs has been applied to the speed control of a separately excited field controlled DC motor. Griffin [13] has done a work on-line PID controller tuning using GA in his MS Thesis. A PID controller is tuned for the ball and hoop system online using GA. Lin et al [14] have presented a tracing control design method for a linear brushless DC motor. Experimental and numerical results are presented to conform them in proposed control design approach.

Xia et al [15] have presented a study on speed control of brushless dc motor using genetic algorithm based fuzzy controller. An auto tuning method for fuzzy logic controller based on GA is presented. Simulation results on speed are given with changing load. Mohamed [16] has then performed PID controller design using GA. A turbine speed control system model is chosen as an example. The results of implemented PID control with GA are presented. Gundogdu [17] has presented optimum parameter tuning for driving an inertial load by a dc motor coupled to a gearbox.

Genetic algorithm based PID parameters are compared to Ziegler Nichols (ZN) settings. Better results are obtained in terms of system performance on step response. Kristiansson and Lennartson [18] have presented a study on optimal tuning of PI and PID controllers to improve the output performance of a closed loop system. Bagıs [19] has then explored a study on determination of the PID controller parameters by using modified genetic algorithm. Three different processes with different order are studied.

In a recent study, Kim et al [20] have performed auto tuning PID controller based on an improved GA for Reverse Osmosis (RO) plant. The performance of GA based controller is compared with the conventional ones using simulation.

2.4 Characteristics of Genetic Algorithm

Genetic Algorithm repeatedly modifies population of individual solutions. At each step, GA selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generations, the population evolves towards to an optimal solution. GA can be used to solve a variety of optimization problems that are not suitable for standard optimization methods.

Genetic Algorithms are different than other Optimization Techniques. These differences can be summarized by reviewing previous many studies [13, 16]. Some of these studies are thesis and projects. In most of the studies, advantages of the algorithm are emphasized for problem solving. However, there are couple of points to be kept in mind while applying GAs. They are given as;

1- GA generates a population of points at each iteration and the population approaches an optimal solution. The standard algorithms like; Ziegler Nichols, Kappa Tau, and Pole Placement generate a single point at each iteration and the sequence of points approaches an optimal solution.
2- GA selects the next population by computations that involve random choices. The standard algorithm selects the next point in the sequence by a deterministic computation.
3- GA does not work on only one search point, it works on one group of search points (these points are named as candidates) It works on global searching place and it does not work on local searching place.
4- GA makes searching the solution not in one point, it tries to find the solution in a group of solution.
5- GA uses objective-fitness function to find the fitness values of individuals.
6- GA uses stochastic rules to select and combine the individuals in these steps.
7- GA does not work on the parameters directly like the other optimization techniques. GA codes the parameters that will be optimized and works on these codes not with the parameters. Aim of this coding is to change the optimization problem to a combination problem.

Despite GA has several advantages, it has some disadvantages. These are given as the following [13];

1- GA includes information about what it does, it does not contain the information about how it does, and it is a blind method.

2- GA works according to the stochastic rules, how well the program is working cannot be known before, but it can be determined later.

There are some points which are encountered while applying GAs. Three phenomena are countered as deception, genetic drift (bias) and real time on-line issues. Depending on the objective function, some of objective functions are difficult to optimize. There is also no guarantee of getting the optimal point. The performance is not guaranteed in on line application. It is not advised to apply GA directly to a real system. Initially a simulation model is required to be built.

2.5 Operation of Genetic Algorithm

GA passes through three main processes to obtain a new Generation. Initially evaluation of the old generation's fitness value of each individual is done. Selection of the new Generation's new individuals is performed by using fitness function according to the fitness value of old individuals finally to get better Generations. Genetic operators such as mutation and crossover are used for this purpose.

GA is typically initialized at a random population between 20 and 100 individuals. Population is also called as *mating pool.* Population is represented by a real-valued number or a binary string that is called as *Chromosome*. *Chromosome* is usually defined as the arrangement of the variables of the problem in a specified order. It is known as potential solution in Population [1, 13,16]. Chromosomes are the encoded representations of all the parameters of the solution. Each chromosome is compared with the other ones in the population and fitness rating shows that how successful this chromosomes to the later. The variable that is used to form *Chromosome* is called as *gene*. So according to the information above, *gene* is the smallest genetic structure that has the meaningful genetic information by itself. To make it understand, the following example is given by referring to Yakut's study [23];

There is a gene and it has a bit string of 101 and it is coded in $x - coordinate$ in binary system. A chromosome that is formed by using genes can be defined as the Genetic structure that all informations about the solution of the problem is stored in it. 100011101111 bit string is a chromosome and it gives information about the solution. In this thesis, PID gains will be optimized and there will be three strings that will be assigned to each member of the population. These members will be comprised of P, I and D string that will be evaluated throughout the course of GA. These three terms are entered into GA via the decleration of a three row matrix. In this matrix, the limitations of the choice of the chromosomes are done between upper and lower limits, and it is up to user of the program [13].

Performance of the Individuals is then determined by Objective Function. Tasks of the Individuals are assessed by the Objective Function. It assigns each individual to a corresponding number that is called as fitness of individual. Three stages of GA are considered as; *Reproduction, Crossover and Mutation.*

(a) Population Size

Population Size of GA is very important because it determines the performance of the Genetic Algorithm. But determining the number of population in GA does not have a thumb and fast rule according to the readings and according to the theories and experiments done before [13]. So determining the number of population is done by trial and error method. From different references [13, 16, 21, 22, 24], suggested safe population size is given between 30-100. In this study, population size is taken between 20-40-60-80 respectively and the results are observed successively. The safe Population size is seen between 40-80, population sizes below 20 and above 80 are not suitable for the solution of the problem. Thus increasing the population size has enabled GA to search more points and thereby obtain a better result. However, the larger the population size, the longer the GA takes to compute for each generation [13]. There must always be a compromise between the population size and the computational time while making a decision during application.

(b) Fitness Scaling

Fitness scaling converts the raw fitness scores that are returned by the fitness function to values in a range suitable for the Selection Function. It uses the scaled fitness values to select the parents of the next generation. The range of the scaled values is important and affects the performance of GA. If it varies too widely, individuals with the highest scaled values reproduce rapidly, taking over the population gene pool quickly and preventing the GA from searching other areas of the solution space. If it varies only a little all individuals have approximately the same chance of reproduction and operation goes on slowly [1, 2, 13].

(c) Reproduction

Fitness value of each chromosome is assessed during Reproduction. This value is used in Selection process. A fit chromosome has a higher possibility of being selected during Reproduction process like in Natural Evolution. Reproduction controls how GA creates the next generation. In this step of GA, there is a term about the children that is called as ***'Elite Count'.*** Elite Count is the number of individuals with the best fitness values in the current generation that are guaranteed to survive to the next generation [24]. They are called ***'Elite children'***. There is a fraction of Selection options for the next generation. This is the fraction of individuals in the next generation, other than elite children, that are created by Selection. It affects the performance of Genetic Algorithm. Fitness value is used in Selection process and there are several types of Selection techniques; Roulet Wheel Selection, Uniform Selection, Tournament Selection, Custom Selection and Stochastic Uniform. All selection methods are based on the same principal that is giving fitter chromosomes to a larger probability of selection.

All of them are used in GA and it is up to the user to select the appropriate one for each process but, Roulette Wheel is the most popular one.

In Roulette Wheel Selection, each individual in the population is allocated in a section of roulette wheel. Firstly, a pointer is spun. After that the individual to whom it points is selected. This procedure continues until the Selection process has been met. The probability of an individual that was selected is related to its fitness. More copies can be obtained by Selection and several copies of the same may be selected for Reproduction and fitter strings are started to begin dominate. But, for the situation that is shown in the figure below, it is possible for the weakest string in the roulette wheel 01001- 19% to dominate the selection process. The size of the section is proportional to the fitness of the individual, as shown in Figure 2.1 with binary coding[13].

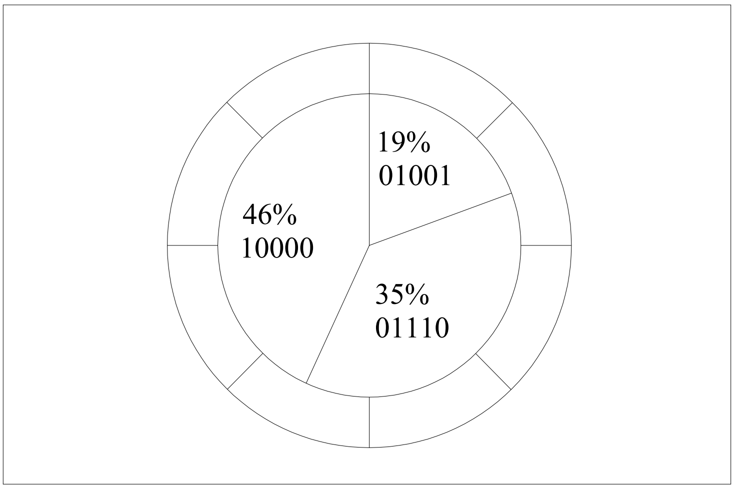


Figure 2.1 Roulette Wheel Selection

(d) Crossover

Crossover algorithm is initiated when Selection process is completed. The Crossover operators swaps certain parts of two selected strings to capture the good parts of old chromosomes and create better new ones. Genetic operators control the characters of chromosome directly and produce fitter individuals for the next generation by using the assumption of certain individual's gene codes. There is a probability that determines how often Crossover is performed. Probability is between 0% and 100%. A probability of 0% means that offspring will be the exact replicas of their parents and probability of 100% means that each generation will be composed of entirely new offspring. There are different types of Crossover techniques; *Single Point Crossover, Two Point Crossover, Intermediate Crossover, Heuristic Crossover* and *Uniform Crossover* [1, 2, 23].

The simplest Crossover technique is Single Point Crossover which includes two stages: Firstly, members of the newly reproduced strings in the mating pool are mated at random. After that, each pair of strings undergoes a crossover as follows:

An integer *k* is randomly selected between one and the length of string less one, [1,L-1]. Swapping all the characters between positions k+1 and L inclusively creates two new strings [16].

To make it understand, the following example is performed; If the strings 11000 and 01111 are selected for crossover and the value of k is randomly set to 3 then newly created strings will be 11010 and 01101.

110**I**00 results as 11010
011**I**11 results as 01101
llustration of Crossover

The other Crossover techniques; Multipoint Crossover (two point crossover) and Uniform Crossover are more complex than single point Crossover. Multi-point crossover is an extension of single point crossover algorithm. It operates on the principle that the parts of a chromosome that contribute most to its fitness might not be adjacent. There are three main stages in Multi-point crossover; firstly, members of the newly reproduced strings in the mating pool are mated at random. After that, multiple positions are selected randomly with no duplicates and sorted into ascending order. Finally, the bits between successive crossover points are exchanged to produce new offspring [16].

For Multi-point crossover, the following example is performed ; If the string 11111 and 00000 were selected for crossover and the multipoint crossover positions were selected to be 2 and 4 then the newly created strings will be 11001 and 00110. In Uniform Crossover, a random mask of ones and zeros of the same length as the parent strings is used as a procedure. Members of the newly reproduced strings in the mating pool are mated at random. A mask is placed over each string. If the mask bit is a one, the underlying bit is kept. If the mask bit is a zero then the corresponding bit from the other string is placed in this position. For Uniform Crossover, the following example is performed: If the string 10101 and 01010 were selected for crossover with the mask 10101 then newly created strings would be 11111 and 00000.

Uniform Crossover is the most disruptive method it affects the next generation's fit string badly so it is not preferred in this study.

(e) Mutation

In GA, Selection and Crossover processes are enough for a large amount of different strings. According to the initial population chosen, there may be not enough diversity in the initial strings to ensure that GA is searching the entire problem space. According to the bad choices of initial population, GA may converge on sub-optimum strings. These problems can be solved by the help of Mutation operator in GA. Mutation is a random alteration of a value of a string position but this happens occasionally because the probability of Mutation is low. If probability is high this destroys the fit strings and degenerate the algorithm into a random search in the searching space.

Probability of Mutation differs between 0.1% and 0.01%. Once a string is selected for Mutation, a randomly chosen element of the string is changed or the other name 'mutated'. For example when GA choses a bit position of 4 for Mutation operator in the binary string 10100 and the resulting string after Mutation operator is 10110.

10100 string after Mutation operator is 10110

(f) Elitism

In GA it is not definite to preserve the fittest string after the operators Crossover and Mutation. So there is high chance that the optimum solution could be lost in the processes of Crossover and Mutation. Elitist's models are often used to eliminate this problem.

The best individual from a population is saved before the operators Crossover and Mutation. It will be seen in the future that when a new population is formed and evaluated, the best individual has been preserved by Elitism [13, 16, 24].

2.6 Objective Function and Fitness Function

Writing an Objective Function is the most difficult part in GA. It changes the result directly in the problem and all the variables that are tried to be optimized pass through the Objective Function to get the result. In this study, Objective function is used to get the best PID controller parameters for both Four Bar Mechanism and Seven Link Hybrid mechanism. Objective Function is required to evalute the best PID parameters in the program. Each chromosome in the population is passed into the Objective Function one at a time and then the chromosome is evaluated and assigned a number to show its fitness. Bigger the Chromosome's fitness values better its fitness [1,13].

Objective Function is used to determine a measure of how individuals perform their tasks and how individuals perform in the solution space. Objective Function assigns each individual a corresponding number that is called as its fitness. Each chromosome's fitness is assessed and a survival of the fittest strategy is applied. In this study, fitness of each chromosome is assessed by the magnitude of the error, positional deviation. Four different types of error methods are used; Integral of Absolute Magnitude of the Error (IAE), Integral of Time Multiplied by Absolute Error (ITAE), Integral of Squared Error (ISE) and Mean Square Error (MSE). These error methods have been applied to minimize the error signal and get the most suitable one for the system [25]. They are explained with related formulations in Chapter 3.

Fitness Function is used to transfer the Objective Function value into a measure of relative fitness. Fitness Function value corresponds to the number of offspring that an individual can expect to produce in the next generation in many cases. [13]

2.7 Summary on Genetic Algorithm

There is flow chart of GA in Figure 2.2. Steps of creating and implementing GA are given as [13];

a- An initial, random population of individuals for a fixed size is generated
b- Fitness of the individuals is evaluated
c- Fittest members of the population is selected
d- Reproduction is done by using a probabilistic method (e.g Roulette Wheel, Tournament, Uniform etc…)
e- Crossover operation is done on Reproduced chromosomes
f- Mutation operator is done with a low probability
g- Second step is repeated if predefined criterion is met.

Predefined criterion is determined by the user of the program and needs of the program. The algorithm is applied MATLAB GA Toolbox. PID controller tuning based on GA is included in Chapter 3.

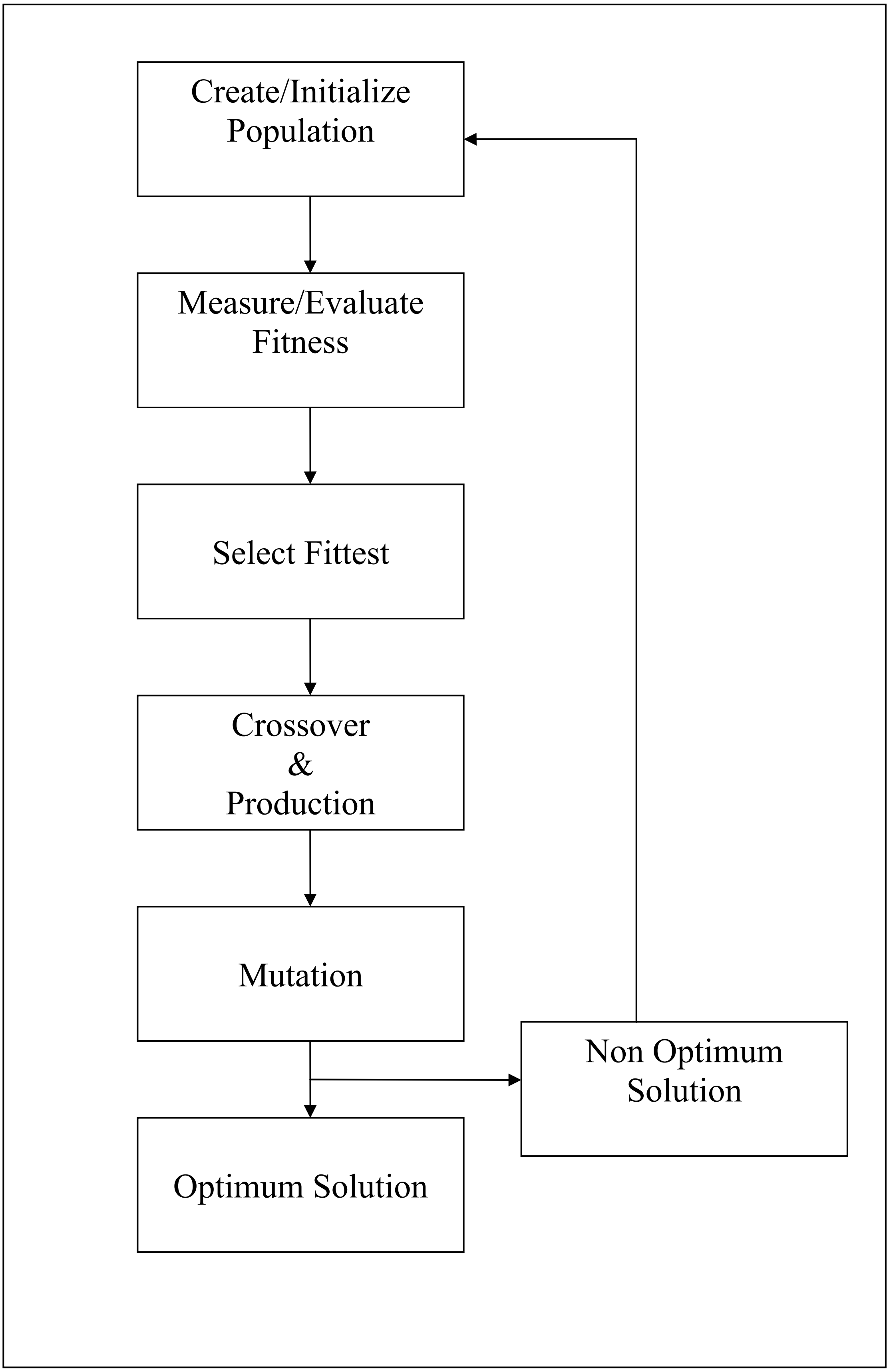


Figure 2.2 Flow diagram of Genetic Algorithm [13]

CHAPTER 3

GENETIC ALGORITHM FOR PID TUNING

3.1 Introduction

This chapter is organized in five sections. First, an introductory section is included on PID controller. PID Tuning with GA is then introduced with MATLAB scripts. The performance of GA is then tested with two motors by a unit step-input with different error methods. Tuned responses are presented for BLDC and PMDC motor with PID parameters found.

3.2 PID Controller

A proportional-integral-derivative controller (PID Controller) is a generic control loop feedback mechanism. It is one of the earlier control strategies. Many control systems with PID control have given satisfaction. So it has wide range of applications in industrial control. A PID controller attempts to correct the error between a measured process variable and a desired set point by calculating and then outputting a corrective action that can adjust the process accordingly [26, 27].

(a) Proportional Action

Proportional Action determines the current error and Proportional Controller output uses a proportion value of the error to control the system. But this proportion action introduces an offset error into the system. It is given in equation (3.1) where K_P represents the proportional gain.

$$P_{term} = K_p x Error \tag{3.1}$$

(b) Integral Action

Integral Action determines the reaction based on the sum of recent errors. The Integral Controller output is proportional to the amount of time that there is an error present in the system. This Action removes the offset introduced by Proportional Action but introduces a phase lag into the system. It is given in equation (3.2) where K_I represents the integral gain.

$$I_{term} = K_I x \int Errordt \tag{3.2}$$

(c) Derivative Action

Derivative Action determines the reaction to the rate at which the error has been changing. The derivative controller output is proportional to the rate of change of the error. It is given in equaiton (3.3) where K_D represents the derivative gain.

$$D_{term} = K_D x \frac{d(Error)}{dt} \tag{3.3}$$

In this action of PID controller, it introduces a lead into the system. The lag in the system is eliminated by introducing Integral action. There is a description of PID controller in feedback loop in Figure 3.1.

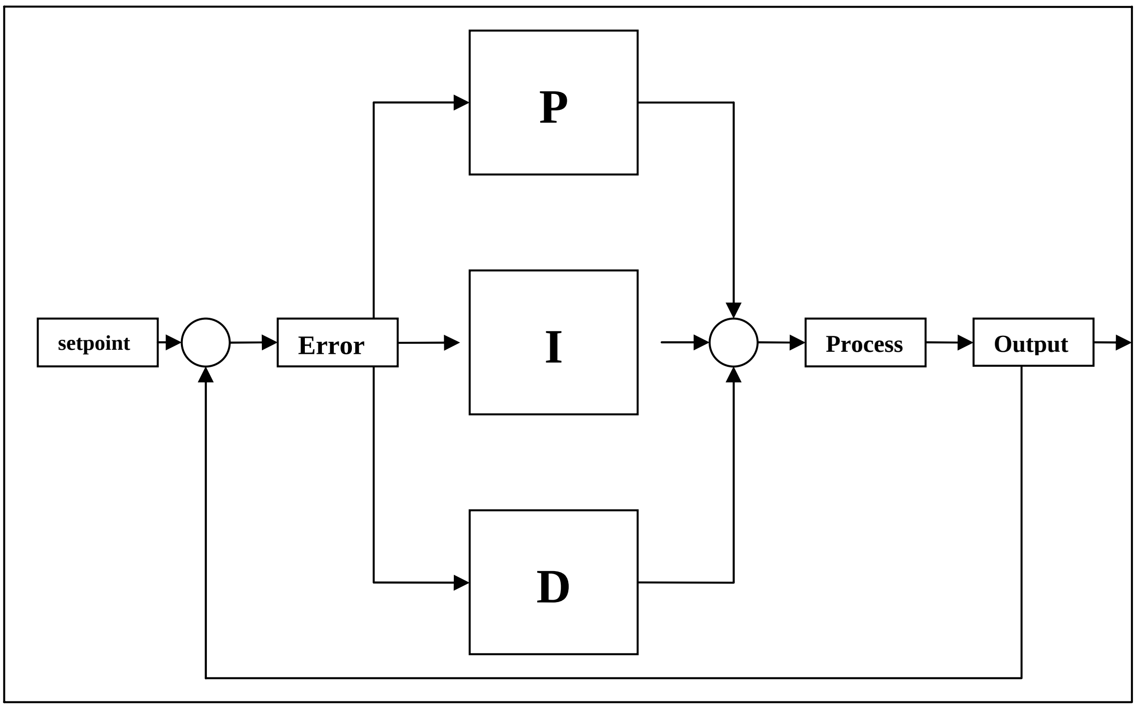


Figure 3.1 Block diagram of PID

K_p **is** the proportional gain K_i is the integration coefficient, Ti is known as the integral action time or reset time and K_i is the derivative action, Td is known as derivative action time or rate time.

By tuning the Proportional, Integral and Derivative constants, PID can do the specified action for the designed system given specific requirements.

A PID controller is described in the continuous s-domain by the following equation below:

$$G_c(s) = P + I + D = K_p + K_i(\frac{1}{s}) + K_d(s) \tag{3.4}$$

Or in it can be written in terms of Ti and T_d as:

$$G_c(s) = K_p(1 + (\frac{1}{T_i}s) + T_d s) \tag{3.5}$$

3.3 PID Tuning Methods

Tuning of PID controller is important to get specified response for the designed system. Several tuning methods are derived and used. These methods are Ziegler-Nichols (ZN) method, Kappa-tau tuning, Pole placement method, Nyquist based design, D-partioning tuning, OLDP method and GAs. There are some cases where these methods can not be applied. These cases are dynamic conditions which its parameters are changing constantly [26-28]. PID control algorithm is used for the control of several applications in process industries. PID loop has to be used and tuned properly to work properly for closed loops. Standard and conventional tuning methods have been used for many years but should be reevaluated for use on modern digital control systems [29, 30].

Ziegler-Nichols (ZN) Method is a PID tuning technique. It was designed by J. G. Ziegler and N.B. Nichols in 1942. It is PID tuning rule that attempts to produce better values for three PID gains parameters. There are two measured feedback loop parameters derived from measurements:

a- the period Tu of the oscillation frequency at the stability limit
b- the gain margin Ku for loop stability

These are used for good regulation. The K_i and K_d gains are set to zero for closed loop systems. Only K_p gain is increased to a value that an oscillation is started in the output. This value is known as critical value K_c=K. After that Ki and K_p values are started to multiply by the values in a table by using oscillation period [25]. ZN method is designed to give good responses to load disturbances.

3.4 PID Tuning with Genetic Algorithms

Genetic Algorithms do not have knowledge of the correct solution and it tries to get the correct solution from the responses from its environment to give an acceptable result [13]. For obtaining the PID tuning parameters in GAs, the performance index has to be minimized. To obtain the performance index, four different types of error criteria's are used; ISE (Integral of squared error), MSE (Mean square error),ITAE (Integral of Time Multiplied by Absolute Error) and IAE (Integral of Absolute magnitude of the error).

The most important step in Genetic Algorithm is to choose the Objective Function to evaluate the fitness of each chromosome. These performance indices are MSE, ITAE, IAE and ISE. Performance indices are given as in the following references [13, 16, 25]:

Mean of the Squared Error= MSE= $\frac{1}{t}\int_0^{\tau}(e(t))^2$ (3.6)

Integral of Time multiplied by Absolute Error: ITAE= $\int_0^{\tau} t|e(t)|dt$ (3.7)

Integral of Absolute magnitude of the Error: IAE: $\int_0^{\tau} |e(t)|dt\tau$ (3.8)

Integral of the squared error: ISE: $\int_0^{\tau} e(t)^2 dt$ (3.9)

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

In PID controllers, error signals are tried to be minimized. They are defined the term of error criteria and minimization the value of performance indices; like (MSE, ITAE, IAE, ISE) are tried to get. So smaller the performance indices, fitter the chromosomes will be and it is given according to the following equation [13];

$$\textbf{Fitness value} = \frac{1}{performanceindex} \tag{3.10}$$

3.5 Genetic Algorithm Toolbox

The study and evaluation of GA are largely depending on simulation. There are different GA software packages [7, 8]. GA toolbox in MATLAB is utilized in this study. The following section includes the codes used for simulation study [3, 24].

3.5.1 Initializing the Genetic Algorithm

Initially to proceed on GA, the algorithm has to be initialized. The PopulationSize, VariableBounds and evaluation function have to be done for proceeding of the GA [24]. These operations are the initial inputs required for the GA processes to start. Some special codes are used for initializing the GA and these codes are based on GA Optimization Toolbox. There are many studies introducing these codes for initializing GA. Couple of studies are taken for the following details [13, 16, 24]. Figure3.2 shows initializing GA with a MATLAB script.

```
%Initialising the Genetic Algorithm
populationSize=60;
variableBounds=[ 0 40; 0 40; 0 40 ];
evalFN='PID_objfun_IAE';
evalOps=[];
options=[1e-6 1];
initPop=initializega(populationSize,variableBounds,evalFN,…
evalops,options);
```

Figure 3.2 Initializing the GA

_populationSize: First step for writing the GA is to create a Population. This command defines the Population Size. In this study according to the several readings, safe Population size is taken between 20 and 80. If Population Size is below 20 or above 80, performance of the GA is decreased.

_variableBounds: In this study, GA is used for optimization of the gains of PID controller. So there are three strings that are assigned to each member of the Population. These members are compromised of P, I and D string. These strings are evaluated throughout the course of the Genetic Algorithm processes. These three terms are entered into the Genetic Algorithm, declaring three row **VariableBounds** matrix. Number of rows in **VariableBounds** matrix shows the number of terms in each member of the Population. Hence, it is selected between 0 and 40, it can be changed to different intervals between -100 and 100 according to the user.

_evalFN: This is a MATLAB function used to declare the Objective Function. It will fetch the file name of the Objective Function and execute the codes and return the values back to the main codes [24].

_**Options:** It is about the encoded types of strings. Two declarations are done in the **Options** command. First one is '1e-6' term; it is the floating point precision and the second term '1' indicates that real numbers to be used in the program. (0 indicates that binary encoding is used).

_initializega: In GA Toolbox, it combines the previously described terms and creates an initial population of 60 real valued members between 0 and 40 with six decimal place precision.

3.5.2 Setting GA Parameters

Some special codes are used for setting up the GA. A MATLAB script example is given in Figure 3.3. [13, 16, 24]

```
%Setting the Parameters for Genetic Algorithm
bounds=[ -100 100; -100 100; -100 100];
evalFN='PID_objfun_IAE';
evalOps=[];
startPop=initPop;
opts=[1e-6 1 0];
termFN='maxGenTerm';
termOps=240;
selectFN='normGeomSelect';
selectOps=0.1;
xOverFNs='arithXover';
xOverOps=4;
mutFNs='unifMutation';
mutOps=8;
```

Figure 3.3 Setting the Parameters of Genetic Algorithm

_bounds: This is different from the initialization of the population part Bounds, it defines entire search space for GA.

_startPop: It is the starting population of GA. Referring the first Initialization section 'initPop'.

_opts: It gives options for GA, 1e-6 defines the precision of the string values, '1' is the definition of real coded values, and '1' or '0' defines the request for the progress of GA to be displayed or not.

_termFN: This function is the declaration of the termination function for GA. It is used to terminate the GA when a stop condition or a certain criterion has been met. It means that when the number of Generations reaches a certain number. This is taken 240 in this study, Genetic Algorithm will be terminated. This function provides advantages to the program, one of the advantage is Genetic Algorithm stops when it reaches the certain criterion and this allows for more control over the compilation time for reaching the GA's termination criterion.

_termOps: It defines the options for TermFN function. This is equal to 240 which means that Genetic Algorithm will reproduce two hundred and forty generations before termination of GA. TermOps can be changed according to the user because if the value of termOps is small, the best result cannot be obtained because of low generation number so user has to increase the termOps value to get the best result for GA.

_selectFN: There are different types of selection functions in Genetic Algorithm Toolbox, Tournament Selection, Roulette Wheel Selection and Normalized geometric selection and etc. SelectFN is the Selection function of the GA. Roulette Wheel is used as selection function [13, 16].

_selectOps: This command defines the probability of selecting of the fittest chromosome of each generation and it is equal to 0.2 in this study. This parameter is important for the best solution for the problem.

_xoverFN: This command defines the crossover function for Genetic Algorithm. In this command it determines that which crossover function will be used. In this study arithmetic crossover technique is used because other methods of crossover are not suitable for this application.

_xOverOpts: In this command, where the crossover function used is determined.

_mutFNs: In this command, which type of mutation operator used in the program is determined.

_mutOps: In this command, probability of mutation is determined and additionally by using the '***multiNonUnifMutation***' the number of mutation, the maximum number of generations and the number reflects the variance of the distribution are detemined. Probability of Mutation is equal to '0.01' in this study.

3.5.3 Performing GA

GA is started to compile by using the command given below. According to the commands, the parameters of GA and initializing the GA, it will compile according to the settings done before the parameters. Performing the genetic algorithm command is done according to the following references [13, 16, 24]. The codes are given in Figure 3.4.

```
[x,endPop,bPop,traceInfo]=ga(bounds,evalFN,evalOps,startPop,opts,…
termFn,termOps,selecFN,selectOps,xOverFNs,xOverOps,mutFNs,mutOps);
```

Figure 3.4 Performing the Genetic Algorithm codes

It can be seen in the command window above that there are some words used before the '**ga'** command for performing GA. Some of definitions are given for them.

x: *it is the best population found during the Genetic Algorithm*

endPop: *it is the final population for Genetic Algorithm*

bPop (bestpopulation): *it is the best population for Genetic Algorithm*

traceInfo: *it is the best and avarage value for each generation*

3.5.4 Objective Function of Genetic Algorithm

Writing the Objective Function is the most important part of creating GA. In this application, Objective Function is written according to the optimization of PID controller. What are the requirements of a PID controller? What are the requirements for an objective to support this? An objective function could be created to find a PID controller that gives the smallest overshoot, fastest rise time or the quickest settling time in time domain. But to combine all these objectives, it was decided to create an objective function such that it minimizes the error of the controlled system, the position error here. In the population, each chromosome is passed into the objective function one at a time. After this chromosome is evaluated and assigned a number to represent its fitness, the bigger its number the better its fitness. These fittest members of chromosome's are used to create a new population. The codes for creating the Objective Function are written as MATLAB scripts [13, 16, 24] as the following;

```
function [x_pop, fx_val]=PID_objfun_IAE(x_pop, options)
global sys_controlled
global time
global sysrl
```

```
%Splitting the chromosomes into three seperate strings
Kp=x_pop(2);
Ki=x_pop(3);
Kd=x_pop(1);
```

```
%Creating the PID controller from current values
pid_den=[1 0];
pid_num=[Kd Kp Ki];
pid_sys=tf(pid_num, pid_den);
```

Each chromosome in the population consists of three separate strings constituting a P, I and D term as defined by the 3-row 'bounds' while creating the population. The equation below is used for creating a PID controller and P, I and D gains are used while creating population.

$$C_{PID} = \frac{(K_d s^2 + K_p s + K_i)}{s} \tag{3.11}$$

PID controller is placed in a unity feedback loop with the systems transfer function. This reduces the compilation time of program. Transfer function of the system is defined in another file and imported as a global variable. The controlled system is initially given a unity step input. The error of the system is then assessed using the error performance criterions such as Integral of Absolute Magnitude of the error (IAE), Integral of the Squared error (ISE), Mean of the Squared error (MSE) and Integral of Time multiplied by Absolute Error (ITAE) .These four error criterions are used to optimize the gains of PID controller. The chromosome is then assigned an overall fitness value according to the magnitude of the error, the smaller the error the larger the fitness value. The codes for error criterions are given for different criterias as the following.

a- Calculating the system error using ISE criteria

```
for i=1:301
error(i)= 1-y(i);
end
error= error*error;
ISE=sum(error);
```

b- Calculating the system error using MSE criteria

```
for i=1:301
error(i)=1-y(i);
end
error_sq=error*error;
MSE=error_sq/max(size(error));
```

c- Calculating the system error using ITAE criteria

```
for i=1:301
error(i)= (abs(1-y(i)))*t(i);
end
ITAE= sum(error)
```

d- Calculating the system error using IAE criteria

```
for i=1:301
error(i)= 1-y(i);
end
IAE=sum(abs(error))
```

These error criterias are used for both motor models for driving the systems; four bar mechanism and seven link hybrid systems. P, I and D gains are optimized for systems output optimization of two different types of motor models. Four different types of error criteria's are used to get the best result for two different types of motor models; brushless DC servo motor (BLDC) and Permanent magnet DC servo motor (PMDC). Several independent tests are made for the best error criterion for two types of motor model. All types of error criteria's are successful for both motor models but ISE error criteria is the best one for BLDC motor model and IAE error criteria is the best one for PMDC motor model. The following section presents the optimum values in a tabulated format for both motors. GA parameters for both motors are also included; as different number of population with same crossover fraction and mutation ratio used during independent runs of the program.

The system stability is also determined by using a code in MATLAB. Pole placement stability method is used in determination. Ensuring controlled system is stable or not, given equations are used below [13, 16, 24]. For all error criteria's same stability is used. Determination of the stability of the system is done according to the poles of the system. For all error criterians IAE, ISE, ITAE and MSE, the same method of stability is used. There is a script included as MATLAB code for the analysis for MSE error criteria as an example. Figure 3.5 shows MATLAB codes necessary for stability analysis.

```
poles= pole(sys_controlled);
if poles(1)>0
MSE=100e300;
elseif poles(2)>0
MSE=100e300;
elseif poles(3)>0
MSE=100e300;
elseif poles(4)>0
MSE=100e300;
elseif poles(5)>0
MSE=100e300;
end
```

$$f_x\_val = \frac{1}{MSE}$$

Figure 3.5 Stability of the system according to the MSE error criteria

3.6 Application Examples of PID with GA

To determine PID gains with GA, many independent trials are performed. At the beginning, some adjustments are done for the variables of GA. Before starting the selection, determination of parameters, and variables of GA, determination of the motor parameters have to be done. The motor inertia and the load inertia directly effects the system's transfer function.Two different types of motors are used, BLDC motor and PMDC motor in this study [9, 32].

3.6.1 PID Controller for BLDC motor

System is an electromechanical one with a motor and a rigid body (the output) directly coupled to motor as a single degree of freedom. The system equations are studied in detail in Chapter 4 with actuator dynamics. While tuning the controller, transfer function of BLDC motor and load are derived using System Control text books [25, 26]. The procedure is a routine one. Motor, load and controller are taken in an unity feedback. BLDC motor electrical data is taken from motor manufacturer's catalogue [32]. Table 3.1 shows the motor data for BLDC motor. In transfer function for the program, the total inertia is determined by using real motor and system parameters. Total inertia is given as the case representing the direct drive;

$$J_{total} = J_{load} + J_{motor} \tag{3.12}$$

Firstly, J_{load} is detemined as a direct drive case;

$$J_{load} = ratioxJ_{motor} \tag{3.13}$$

Different load inertias are then applied to imitate different loading cases as 25, 50 and 75 times the motor inertia. PID controllers are tested with a unit-step signal applied as reference signal to the control system. In first try, ratio is selected $J_{load} = 25xJ_{motor}$. Many trials are performed and the simulation results are obtained. Transient characteristics in this J_{load} value are good, but not the best one for the application.

Table 3.1 Brushless Servo Motor Data

Motor type	BLDC servo motor
Rotor moment of inertia	J_{m1}=6.8x10^{-4} kg.m^2
Motor torque constant	K_{t1}=0.76 Nm/A
Motor voltage constant	K_{e1}=90 V/krpm
Winding inductance	L_1=5.8 mH
Winding resistance	R_1= 0.8 Ω
Damping coefficient	b=0.05

In second try, $J_{load} = 50xJ_{motor}$ is applied and the results are seen. The system output is good but not the best one. Finally, $J_{load} = 75xJ_{motor}$ is applied and many trials are done. The results are then obtained. In this value for inertia load, the results are very good in terms of uniform heavy loading. Thus $J_{load} = 75xJ_{motor}$ is used for graphics to show unit step response. Having decided the loading, the other parameters for GA are to be determined next.

During the implementation of GA, PopulationSize is determined because it is the most important parameter for GA. Several trials are done for best PopulationSize. Safe PopulationSize is between 20- 80. Determination of PopulationSize is then started. Trials are performed by selecting Population Size at 20. At the same time, the other parameters of GA in the program are fixed. The other parameters ; Selection rate and Selection Type, Crossover rate and Crossover Type, Mutation rate and Mutation type that are fixed by selecting the Population Size are fixed. The simulation results for PopulationSize for 20, 40 and 60 are good but not the best. So PopulationSize is set 80 and the results are observed. Better results are seen in PopulationSize is 80.

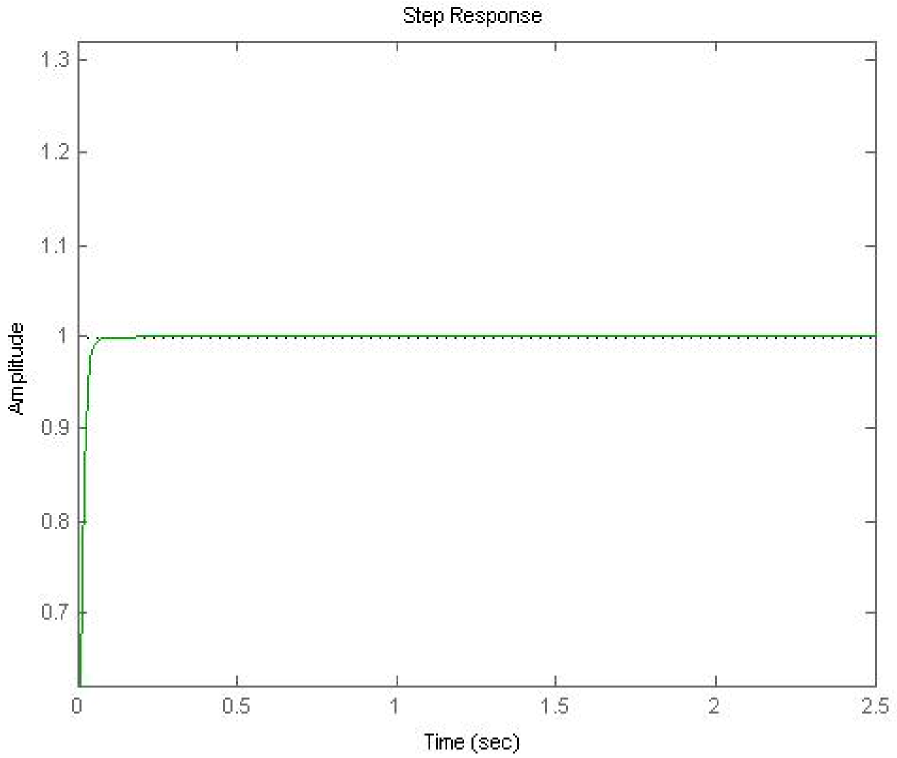
The same procedure is also applied when performing the error criteria and determining the other parameters of GA that was mentioned before. As already stated, there are four different types of error criteria; ITAE, IAE, ISE and MSE. Several independent runs are performed for selection of error criteria. All error criterias have given good results. However, one of the best is looked for in terms of transient response to a unit step. ISE and IAE error criteria's give the acceptably good results for BLDC motor. All parameters of GA are set like in Table 3.2. Tabulated results are given in Table 3.3 for BLDC motor. Ten independent runs are performed without changing any parameter only by changing the error criterion in use. The best results for each error criteria are presented in Table 3.3. ISE error criteria, the third result-the bold one in the Table 3.3 is the best one for direct drive for BLDC motor because according to the transfer function and type of the problem, ISE error criteria is the best one for this optimization problem. Other error criteria's results are good but according to the type of probem ISE error criteria is the best one for this study. The best results are presented in Figure 3.6. Thus Figures 3.6(a) and 3.6(b) represents the motor response and the corresponding PID parameters respectively.

Table 3.2 Parameters of GA for BLDC motor

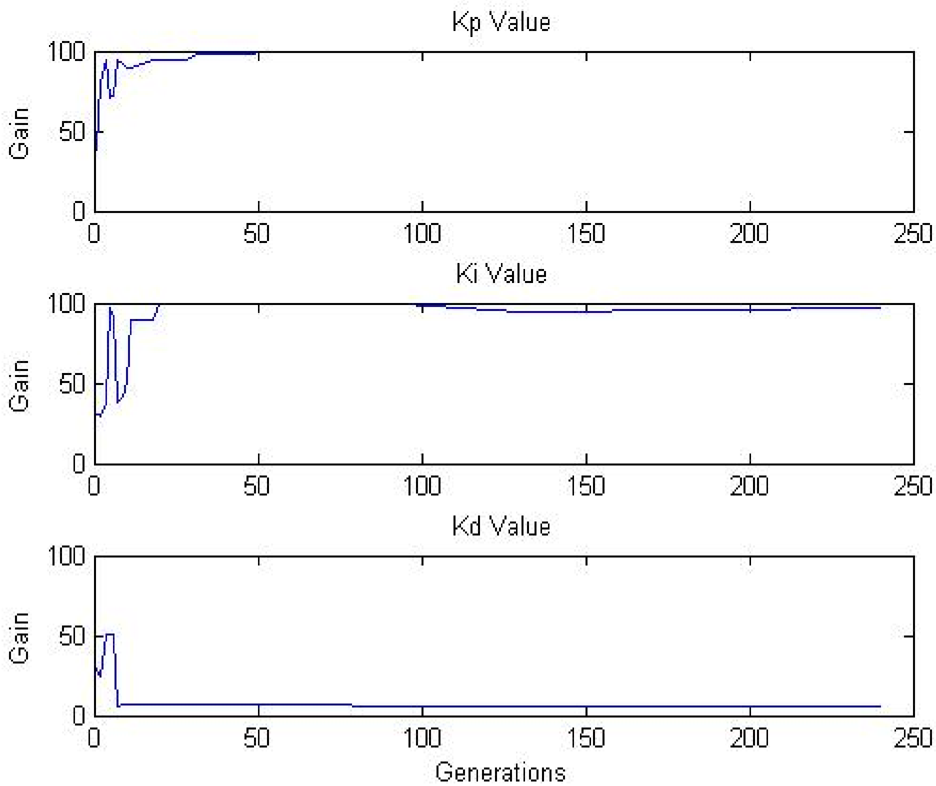
	BLDC servo motor
Crossover probability	0.1
Mutation probability	0.01
Generation numbers	240
Population size	80
Selection type	*Roulette Wheel*

Table 3.3 GA based PID Parameters for BLDC motor (Direct drive)

Error Methods	PopSize=80, Generation size=240, Selection option=0.2 Mutation Option= 0,01, $J_1 = 75xJ_{BLDC}$		
	Kd	**Kp**	**Ki**
ISE Error Method	4.53	88.32	83.94
	4.94	96.59	99.63
	5.22	**99.65**	**97.49**
	5.73	99.85	98.97
	2.84	57.5	54.39
	3.41	67.26	63.77
	5.1	98.72	93.62
	4.75	91.93	87.49
	4.90	95.05	90.18
	5.64	99.42	99.23
ITAE Error Method	1.77	32.55	30.81
	1.37	37.2	36.28
	2.03	39.95	37.92
	1.95	43.86	42
	1.41	29.61	28.25
	1.83	32.03	30.23
	1.73	36.09	34.32
	3.04	65.28	62.37
	2.05	37.04	35.26
	4.51	87.1	82.53
IAE Error Method	1.78	37.67	35.5
	1.96	42.7	42.74
	1,53	34.71	33.16
	2.33	41.5	39.16
	2.29	48.97	46.75
	2.69	49.89	47.28
	3.64	77.05	73.49
	2.21	47.19	42.22
	3.93	80.59	76.89
	3.39	63	59.71
MSE Error Method	1.91	43.42	39.8
	1.85	42.67	39.03
	1.99	44.25	41.04
	2.93	59.02	55.72
	2.01	44.76	41.35
	3.78	73.93	70.17
	2.76	56.28	53.08
	2.3	48.91	45.63
	2.12	46.34	43
	1.87	42.96	39.27



(a)



(b)

Figure 3.6 GA based closed loop response of BLDC motor

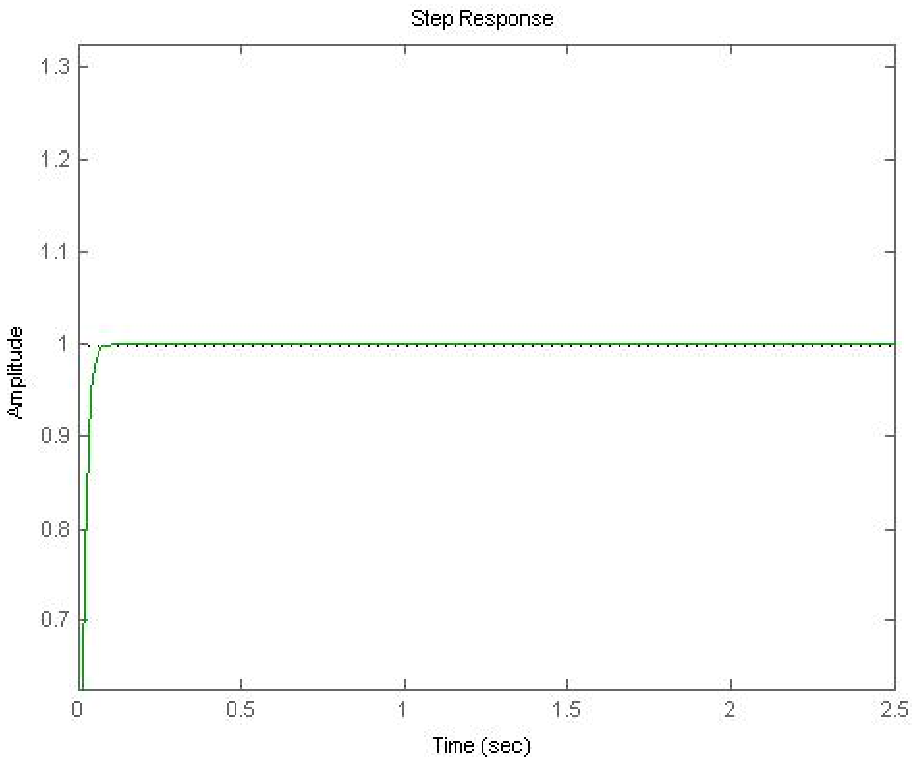
Having performed the above runs, the system transfer function is changed as one to be used as hybrid actuator which be explained in Chapter 4. This case presents lighter loading case, but the system is nonlinear. System is modeled with only taking linear representation with hybrid loading application. Then a unit-step is applied like in the previous example. Same GA parameter settings are used with different inertia in this problem. The results of PID are presented in Table 3.4. Since the same procedure is applied, the details of procedure are not given. In Table 3.4 for BLDC motor for Hybrid Actuator, the best results for each error criteria is typed. ISE error criteria (bold) is the best one for BLDC in this example. The simulation results are presented in Figure 3.7(a). Figure 3.7(b) shows PID parameters for this example. So in simulation results, an acceptable closed loop response for a unit step input is obtained by the minimization of ISE performance criteria.

3.6.1.1 Discussions on PID Tuning for BLDC motor

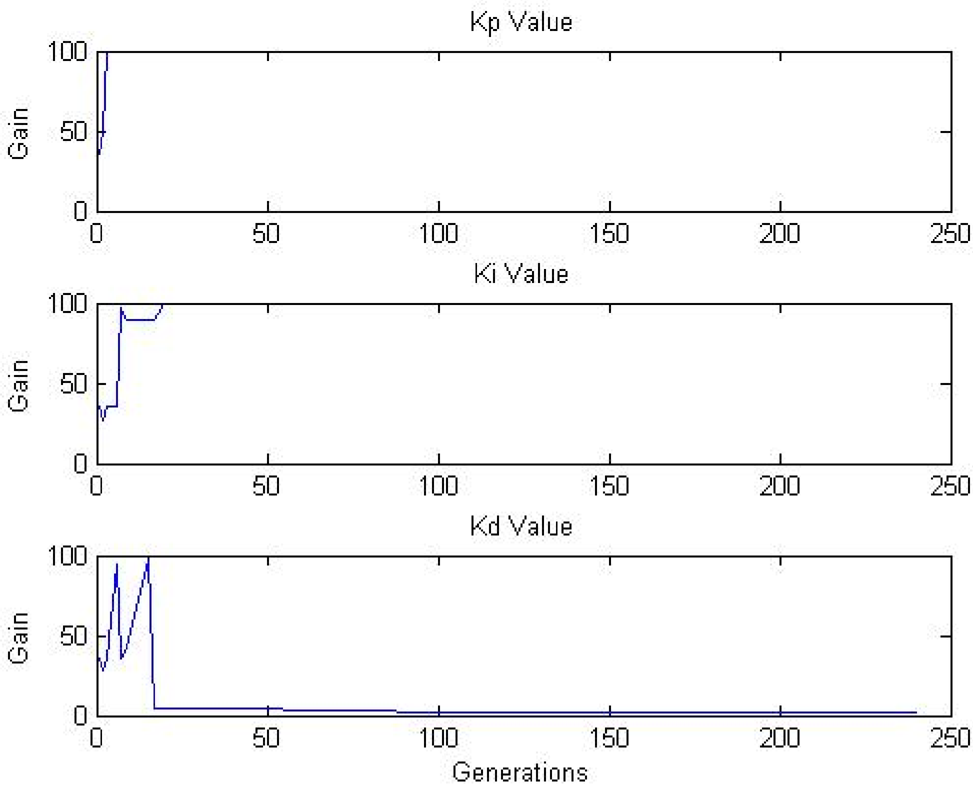
To increase the capabilities of PID controllers in use, GA based tuning is performed here. Application examples are chosen as BLDC and PMDC motors which are previously used in experimental systems. Motor structures are different, BLDC motor offers a lighter structure, with electronic computation. Here GA based results are presented in Table 3.3 and Table 3.4 where BLDC motor has been tried with an unity signal for different loading. Time domain specifications are taken into consideration as no overshoot, fast settling time etc. Many runs are performed in Matlab GA Toolbox, different error methods are studied. Since it is not practical to give all here, they are tabulated and PID gain values are presented to show consistency in optimization. According to the motor structure difference and type of the problem, ISE is chosen for BLDC motor, representing bold line in Table 3.3 and Table 3.4. The intention is to offer a design guide to the reader on the subject.

Table 3.4 GA based results for BLDC motor (Hybrid Actuator)

Error Methods	PopSize=80, Generation size=240, Selection option=0.2 Mutation Option= 0.01		
	Kd	**Kp**	**Ki**
ISE Error Method	1.42	75.1	71.54
	1.98	**99.76**	**95.49**
	1.5	78.08	74.38
	1.79	97.58	93.01
	1.42	75	71.6
	0.96	55.17	52.46
	1.95	99.93	95.5
	1.22	65.88	62.28
	2.32	99.62	99.78
	2.05	99.2	95.79
ITAE Error Method	1.25	29.2	27.02
	0.74	33.24	31.4
	0.68	36.27	34.46
	0.98	23.95	23.03
	0.16	19.87	18.91
	0.77	54.88	52.27
	1.22	90	85.77
	0.48	36.54	34.78
	1.33	86.06	81.77
	0.9	41.33	39.03
IAE Error Method	0.53	36.15	34.4
	0.79	54.89	52.24
	1.09	61.07	57.73
	0.09	28.69	27.6
	1	63.23	61.61
	0.25	34.75	33.29
	0.77	53.75	51.39
	1.29	87.34	83.15
	0.58	56.82	54.3
	1.5	99.42	94.52
MSE Error Method	1.33	83.48	79.55
	0.92	99.58	98.23
	0.09	64.61	59.79
	0.096	28.93	27.45
	0.81	57.14	54.33
	0.5	78.13	74.77
	1.06	99.87	94.47
	2.01	99.67	99.49
	1.25	82.96	78.05
	1.59	97.91	93.01



(a)



(b)

Figure 3.7 GA based response and PID parameters for Hybrid Configuration

3.6.2 PID Controller for PMDC motor

Similar procedure is applied while tuning PMDC motor. System is still an electromechanical one with a motor and a rigid body (the output) directly coupled to motor as a single degree of freedom. The motor equations are similar to BLDC motor. While tuning the controller, transfer function of PMDC motor and load are derived by using System Control Text books [25, 26]. PMDC motor electrical data is taken from motor manufacturer's catalogue [32]. Table 3.5 shows the motor data for PMDC motor. While determining the transfer function required for the program, the total inertia is determined by using real motor and system parameters.

Table 3.5 PM Servo Motor Data

Motor types	PMDC servo motor
Rotor moment of inertia	J_{m2}=1.3x10^{-3} kg.m^2
Motor torque constant	K_{t2}=0.63 Nm/A
Motor voltage constant	K_{e2}=38 V/krpm
Winding inductance	L_2=1.1 mH
Winding resistance	R_2=0.45 Ω
Damping coefficient	b=0.05

In transfer function for the program, the total inertia is determined. Total inertia is found like in the previous part as;

$$J_{total} = J_{load} + J_{motor} \tag{3.12}$$

J_{load} is determined as;

$$J_{load} = ratioxJ_{motor} \tag{3.13}$$

In the first try, the ratio is selected as $J_{load} = 25xJ_{motor}$. Many trials are separately done and the responses are obtained. Results in this J_{load} value are good, but not the best one for the application. In the second try, $J_{load} = 50xJ_{motor}$ is applied and the results are seen. Again the results are good but not the best one. Finally, $J_{load} = 62.5xJ_{motor}$ is applied and more trials are performed by observing transient responses for a unit step input. In the last ratio, representing heavier loading than the previous ones, the results are acceptably good. So $J_{load} = 62.5xJ_{motor}$ is presented here. When the mechanical parameters are decided, then other parameters for GA are tried to be set.

Firstly, PopulationSize is determined because it is the most important parameter for GA. Several trials are done for best PopulationSize. Safe PopulationSize is between 20- 80. Trials are started by selecting the Population Size 20. The other parameters of Genetic Algorithm in the program are fixed; Selection rate and Selection type, Crossover rate and Crossover type, Mutation rate and Mutation type. Then the results for different population sizes are observed, for example 20, 40 and 60. PopulationSize is later set 80 and the results are reasonably good but computation has taken more time. Similar procedure is applied while selecting the error criterias; ITAE, IAE, ISE and MSE and determining the other parameters of GA are determined like in BLDC motor before. More and more trials are done for selection of error criteria. All error criterias have given good results but ISE and IAE error criterias give the better results for PMDC motor. Consequently IAE error has been decided. Parameters of GA for PMDC motor is given Table 3.6.

Table 3.6 Parameters of GA for PMDC motor

	PMDC servo motor
Crossover probability	0.1
Mutation probability	0.01
Generation numbers	240
Population size	80
Selection type	*Roulette Wheel*

All error calculations are presented in Table 3.7 for PMDC motor. Many more trials are done in MATLAB environment with graphical facility. Figure 3.8(a) and (b) show the response and PID parameters found for this loading case.

Table 3.7 GA based PID results for PMDC motor (Direct drive)

Error Methods	PopSize=80, Generation size=240, Selection option=0.2 Mutation Option= 0,01, $J_1 = 62.5xJ_{PMDC}$		
	Kd	**Kp**	**Ki**
ISE Error Method	2.24	48.86	44.6
	5.39	98.22	91.75
	1.57	38.27	33.28
	3.16	59.79	55.67
	1.81	40.94	36.78
	3.83	70.73	66.1
	2.68	52.41	48.27
	5.55	99.83	94.63
	1.83	40.96	36.87
	2.73	53.26	49.17
ITAE Error Method	1.85	29.88	27.88
	1.89	30.21	28.08
	2.59	37.94	35.02
	1.75	40.15	38.07
	2.07	37.1	34.67
	2.34	42.62	39.64
	2.57	41.53	38.62
	2.23	32.66	30.17
	5.17	93.13	87.14
	3.74	79.56	75.35
IAE Error Method	4.2	77.21	72.37
	5.1	95.4	89.54
	5.43	**97.44**	**90.88**
	2.71	49.16	46.05
	4.85	89.18	83.61
	3.51	65.44	61.63
	5.31	96.64	90.57
	5.47	99.62	93.15
	5.12	80.59	76.89
	3.39	63	59.71
MSE Error Method	4.49	91.92	77.51
	5.56	99.78	95.39
	4.71	85.73	80.33
	5.37	97.74	91.78
	3.45	64.31	59.74
	2.64	51.7	47.53
	5.69	98.96	96.14
	5.43	98.53	92.3
	4.36	79.56	74.52
	2.43	48.84	44.6

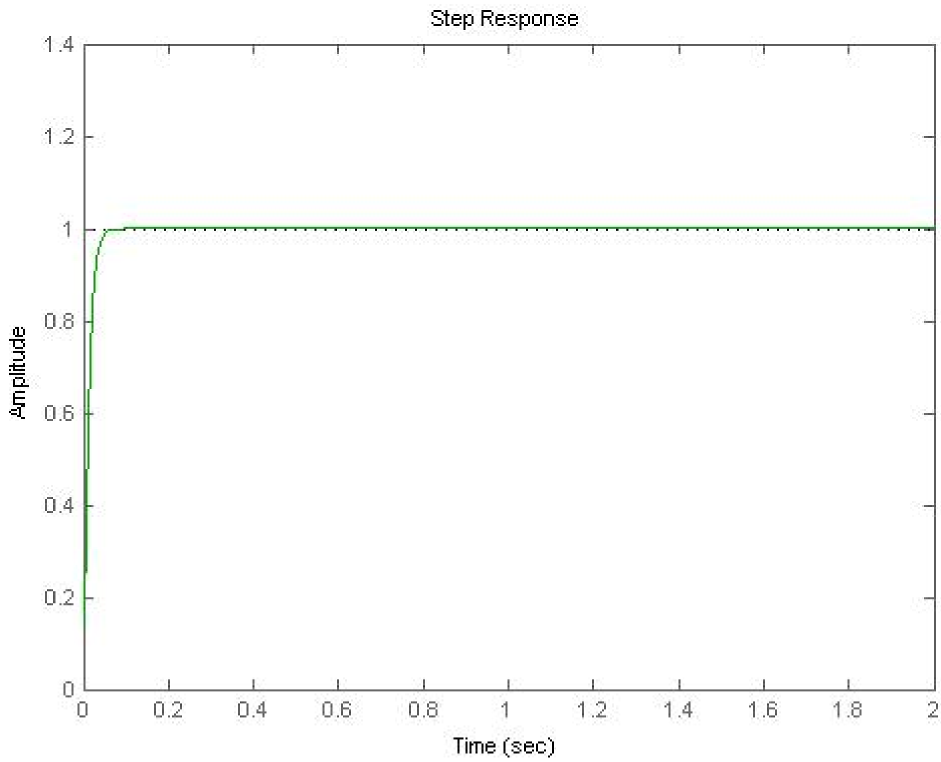
In Table 3.7, the results for PMDC motor is performed and presented for PMDC motor Direct Drive. Direct Drive best result for PMDC motor is given in Figure 3.8 (a) and Figure 3.8 (b) and represented by bold letter in Table 3.7. There are many different values of Kp, Kd and Ki satisfying the condition of minimum error. Best tuning is dependent on the smallest error.

Similarly a set of results are obtained using the error methods for PMDC motor with Hybrid Actuator (2^{nd} axis). IAE error criteria is applied. Different values of PID parameters are obtained. They are given in Table 3.8. Figure 3.9 (a) and (b) show GA based results for this motor and Hybrid Actuator alternative. Again an unity step input signal is applied during tests.

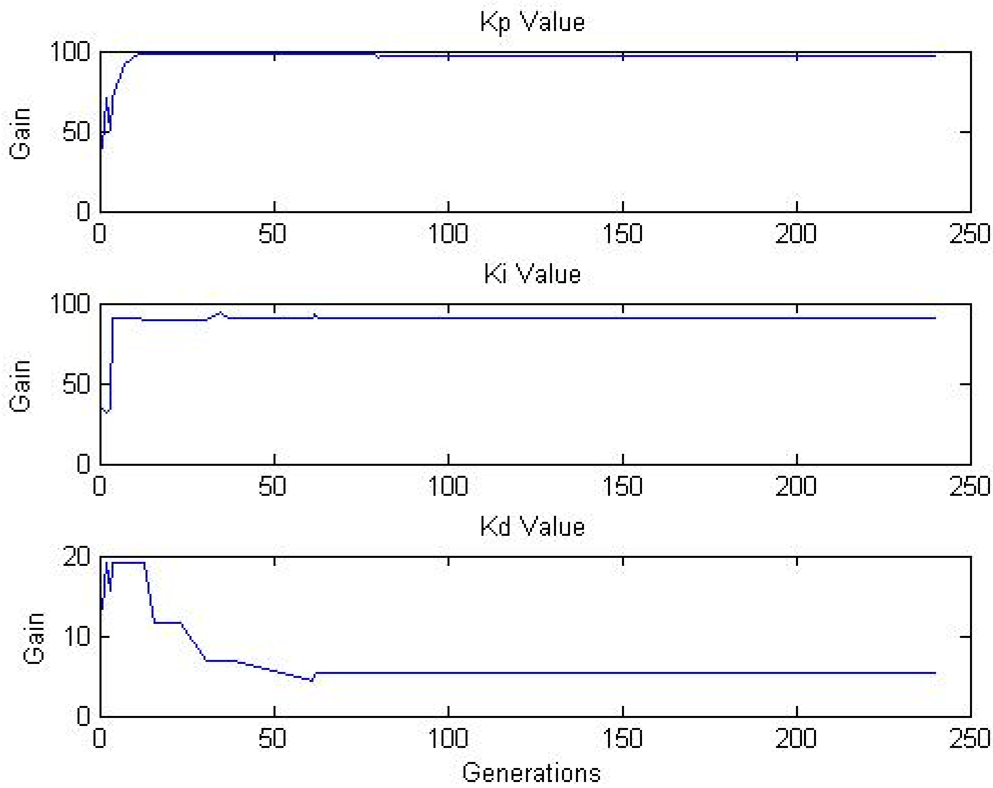
At this point, both application examples provide PID parameters to be used in the following Chapter. In Chapter 4, the results found in Table 3.3 are applied while building section on position control of Four- bar mechanism. Similarly, PID values in Table 3.4 and Table 3.8 are directly used during simulation in section on the position control of the Hybrid Actuator in Chapter 4.

3.6.2.1 Discussions on PID Tuning for PMDC motor

There are four different types of error criteria's and these are ISE, IAE, ITAE and MSE. All error criteria's are performed for the optimization problem but best result is obtained by IAE error criteria. Other error criteria's has given good results but the best one for PMDC motor for Direct Drive and Hybrid Actuator is IAE error criteria. It is related with the type of problem and properties of the motor structure of PMDC motor. All results for four different types of error criteria's are given in Table 3.7 and Table 3.8 for Direct Drive and Hybrid Actuator, respectively. PID parameters are found to be different as shown in bold using IAE error criteria. They are referred as optimum values during simulations in Chapter 4.



(a)

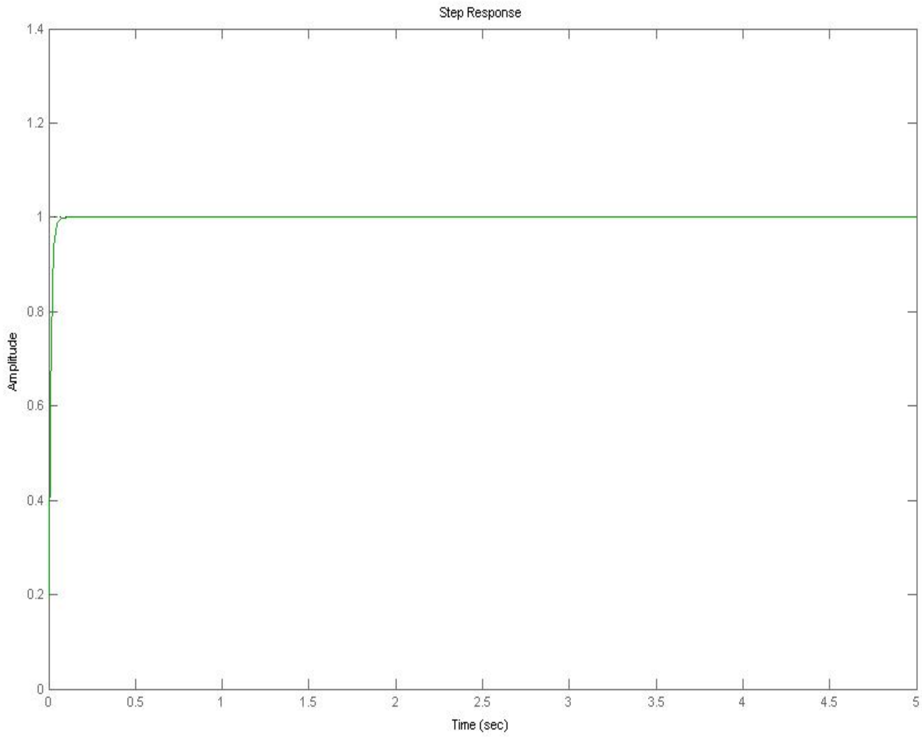


(b)

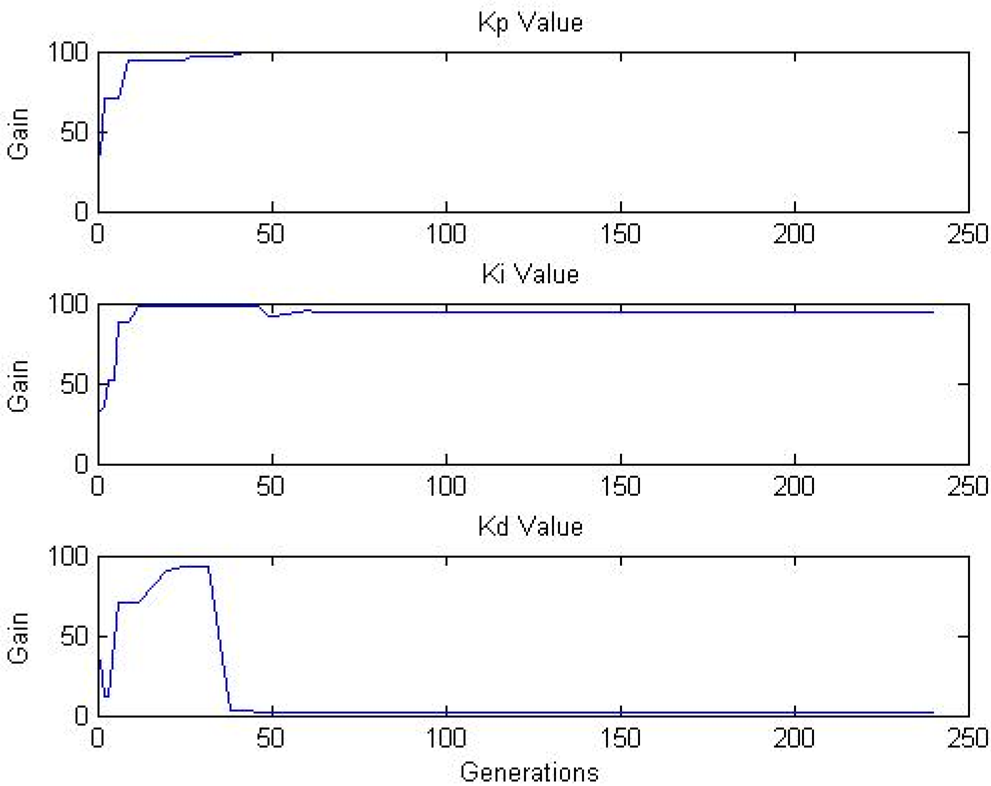
Figure 3.8 GA based closed loop response of PMDC motor

Table 3.8 GA based PID results for PMDC motor (Hybrid Actuator)

Error Methods	PopulationSize=80, Generation Size=240, Selection Option=0.2		
ISE Error Method	**Kd**	**Kp**	**Ki**
	0.11	94.77	88.95
	0.03	91.38	85.8
	0.31	98.49	92.39
	0.077	78.73	73.87
	0.53	99.97	95.07
	0.14	96.29	90.32
	0.29	32.7	30.69
	0.41	78.07	73.63
	1.26	99.99	91.37
	1.65	89.23	85.64
ITAE Error Method	0.4	36.16	34.35
	1.08	66.73	62.77
	0.25	25.58	24.16
	0.87	38.55	35.67
	0.93	90.34	85.86
	0.19	71.84	68.37
	0.71	42.19	39.64
	1.63	80.6	75.42
	0.33	32.53	30.73
	1.43	69.71	65.21
IAE Error Method	1.69	94.66	88.84
	1.19	70.25	65.65
	1.5	42.02	38.58
	0.38	34.02	32.05
	0.68	83.93	79.48
	1.16	85.58	79.83
	1.82	**98.09**	**91.98**
	0.7	42.86	40.74
	1.46	76.99	72.13
	1.73	98.33	92.23
MSE Error Method	0.5	80.62	76.21
	1.04	59.64	55.83
	0.5	39.41	36.82
	0.88	53.01	49.82
	0.37	35.27	33.07
	1.97	99.93	94.51
	1.99	99.51	94.06
	1.48	81.82	76.7
	0.62	78.45	73.4
	1.21	67.24	63.34



(a)



(b)

Figure 3.9 GA based results for PMDC motor-Hybrid Configuration

CHAPTER 4

POSITION CONTROL OF PLANAR MECHANISMS

4.1 Introduction

This chapter is presented position control of planar mechanisms with GA. Two examples are taken as ‘four bar mechanism’ and ‘the hybrid actuator’. Control and mathematical structure and actuator dynamics for both systems are included. Computer simulation is developed using the system’s dynamic equations with GA based PID controller and actuators . So the performance of GA is tested both applications given.

4.2 Four Bar Mechanism

A planar mechanism describes plane curves in space. A four bar mechanism is a widely used single degree of freedom mechanism with different configurations as double crank, crank rocker and double rocker in industry. Many studies can be found on four bar mechanisms, their configurations as crank rocker mechanism, their kinematics and dynamics with control issues [33, 34, 36, 37].

Here a previously built four bar mechanism, at Dynamics System Laboratory is used. Figures 4.1.(a) and 4.1.(b) show assembly drawings for this mechanism with original dimensions driven by a BLDC motor (BRU 200, S-4075 motor, DM 30 drive). It represents a direct drive configuration with no other coupling elements between the motor and the crank. The coupler points are taken on the coupler link.

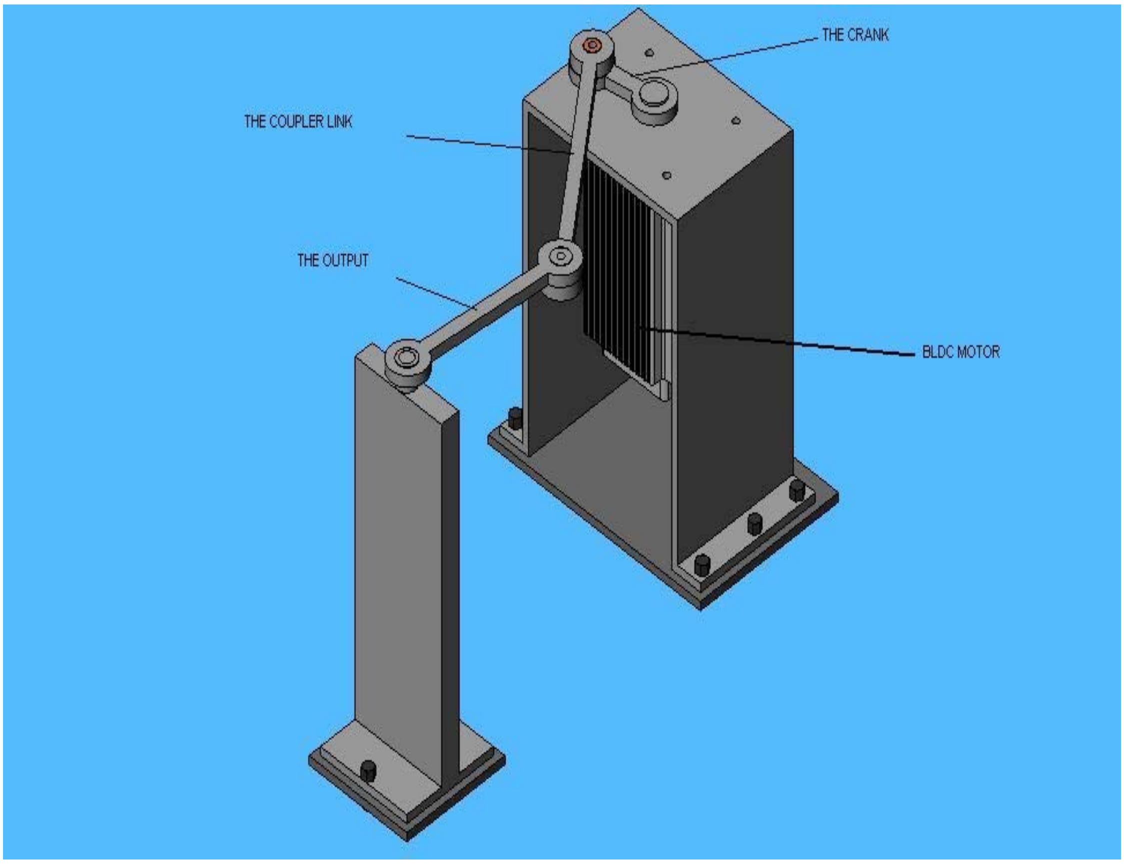


Figure 4.1 (a) An assembly view – Four Bar Mechanism

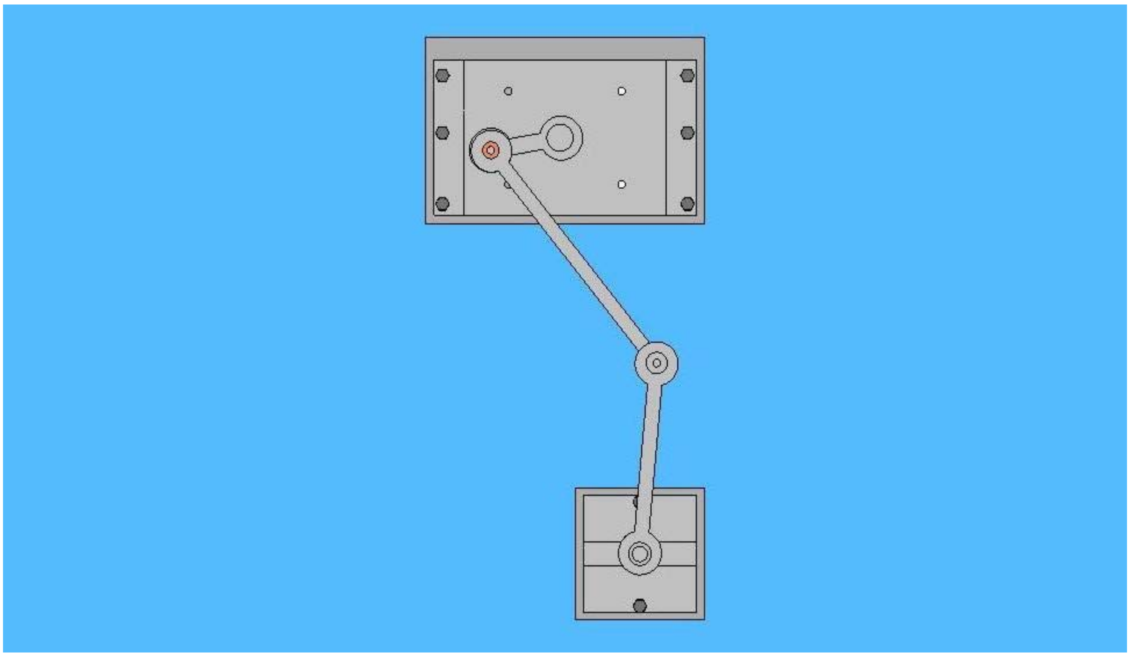


Figure 4.1 (b) Top view

4.2.1 Control Structure of Four Bar Mechanism

The block diagram representation of GA based PID controller, BLDC motor and four bar mechanism is given in Figure 4.2. In a previous study, Dülger and Uyan [36] have done modelling, simulation and control of this four bar mechanism with a BLDC motor. This study has involved a traditional PD controller. An experimental set up was built and simulation results were presented. Later another study [37] was performed on the four bar mechanism with PMDC motor and a gear unit, and PID controller tuning is performed by PSO algorithm. A PSO-PD controller is designed and applied; position control of this mechanism is achieved. Here PID controller is tuned with GA, position control of the mechanism is achived. The mechanism and the motor are same, but tuning algorithm is different.

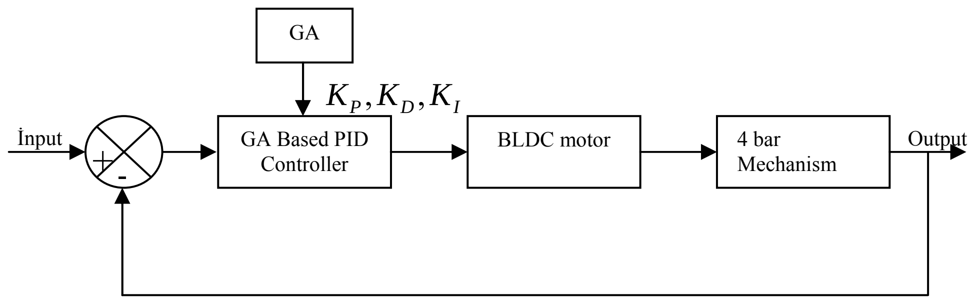


Figure 4.2 GA based PID Controller-Four Bar Mechanism

4.2.2 Mathematical Structure of Four bar Mechanism

Figure 4.3 illustrates a sketch of the four bar mechanism which is used for derivation of the mathematical model. The mechanism operates in the horizontal plane. Freudenstein [33, 34] and Lagrangian [35] equations are used for the kinematical analysis and also deriving the equation of motion of the system. Initially dimensions of the mechanism are chosen in order to give full rotation of the crank (θ_2). Thus, the input link (θ_2) rotates a full cycle and the output link (θ_4) moves in a certain angles resulting in swinging motion. Here link 1 is the ground, link 2 is the crank, link 3 is the coupler and link 4 is the output. In link 3, a coupler link is also fixed to get coupler curves on link 3. It is possible to change coupler curves by altering dimensions of the point fixed.

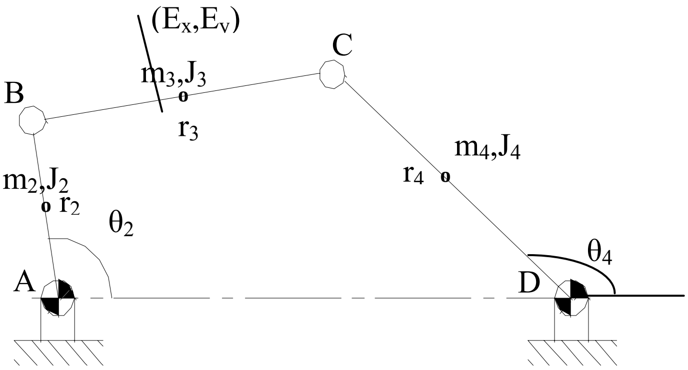


Figure 4.3 Four Bar Mechanism and its parameters

$\theta_2, \theta_3, \theta_4$	the angular displacements (rad),
$\dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4$	the angular velocities (rad/s),
$\ddot{\theta}_2, \ddot{\theta}_3, \ddot{\theta}_4$	the angular accelerations (rad/s^2),
a_1, a_2, a_3, a_4	the link lengths (m),
r_2, r_3, r_4	the lengths of the distance w.r.t. center of gravity (m),
m_2, m_3, m_4	the link masses (kg),
J_2, J_3, J_4	the link inertias (kg.m^2)
E_x, E_y	the coupler coordinates (m)

The system is studied with actuator-mechanism configuration by including actuator dynamics. So electrical circuit equation of electrical motor is expressed first order nonlinear equation with constant coefficients. Motor electrical equation is written by using Kirchoff's law[25], and given in equation (4.1). In this equation, K_e is the motor voltage constant (V/rad/s), L is the motor inductance (H), R is the motor resistance (Ω), I is the motor current, and V is the applied voltage to the motor coil. Electrical parameters which are important for motor-coil equation are taken from the producer catalogue [32]. Mathematical expression for traditional PID controller is given in equation (4.3) for the applied voltage, where K_p, K_v and K_i represent the proportional, the derivative, and the integral constants respectively.

The system dynamic equations are given in closed form in equation (4.2). So electromechanical system is completely described by equations (4.1) and (4.2), relating the motor torque, the angular displacements, velocities and accelerations.

$$L\frac{dI}{dt} + RI + K_e \frac{d\theta}{dt} = V \quad (4.1)$$

$$\left. \begin{aligned} &\left[J_m + J_g + \left(\frac{m_2 r_2^2 + J_2}{N^2} \right) \right] \ddot{\theta}_{2m} + \left[(m_3 r_3^2 + J_3) \left(\frac{\partial \dot{\theta}_3}{\partial \dot{\theta}_{2m}} \right) \right] \ddot{\theta}_3 + \left[(m_4 r_4^2 + \right. \\ &\left. J_4) \left(\frac{\partial \dot{\theta}_4}{\partial \dot{\theta}_{2m}} \right) \right] \ddot{\theta}_4 + \left[(m_3 r_3^2 + J_3) \dot{\theta}_3 \frac{d}{dt} \left(\frac{\partial \dot{\theta}_3}{\partial \dot{\theta}_{2m}} \right) \right] + \left[(m_4 r_4^2 + J_4) \dot{\theta}_4 \frac{d}{dt} \left(\frac{\partial \dot{\theta}_4}{\partial \dot{\theta}_{2m}} \right) \right] = T \end{aligned} \right\} \quad (4.2)$$

$$V = K_p e(t) + K_v \frac{d}{dt} e(t) + K_i \int_0^t e(t)dt \quad (4.3)$$

where e(t)= θ_{2c}-θ_2. Equation (4.3) is second order nonlinear expression with variable inertia terms that are motion dependent on mechanism parameters; where J_m is the inertia of rotor (kg.m^2), N is the gear box ratio and J_g represents the inertia of gear box. Gear box ratio N is used in calculations as N= θ_{2m}/θ_2 and is taken N=1 for direct drive motor-mechanism in this study. Gear and motor losses, frictional effects are not included in mathematical calculations to simplify the mathematical model. Additionally, torque is proportional with motor current represented in equation (4.4).

$$\tau = K_t I \quad (4.4)$$

Where K_t is the moment constant (Nm/A).

Motor-mechanism equations are written in terms of three first order equations. Runge Kutta is used for numerical solution. Electrical parameters of BLDC servo motor are taken from [32]. Link lenghts, masses and inertias of four bar mechanism are given in Table 4.1.

$a_1 = 300mm$	$a_2 = 90mm$	$a_3 = 360mm$	$a_4 = 260mm$
	$r_2 = 68.16mm$	$r_3 = 180mm$	$r_4 = 145mm$
	$m_2 = 0.19kg$	$m_3 = 0.198kg$	$m_4 = 0.306kg$
	$J_2 = 407.26kg.mm^2$	$J_3 = 3503.1kg.mm^2$	$J_4 = 3673.8kg.mm^2$
	$E_x = 180mm$	$E_y = 150mm$	

Table 4.1 Four Bar Mechanism Parameters

A traditional PID controller is applied on the mathematical model with the motor-constant load. Routh-Hurwitz criteria is applied on to motor-load model. Stability of the system is also examined in closed loop control by using MATLAB GA Toolbox.

4.2.3 Application of GA to PID Controller

GA is implemented in MATLAB with Genetic Algorithm Toolbox. Here each chromosome has three strings involving three terms in controller. In PID controller tuning, each chromosome has genes as Chromosome=[K_p K_d K_i]. There are three strings assigned to each member of the population, these members will be comprised of P, I, and D string. These three terms are entered into GA as three row matrix in program. Here the limitations of the chromosome choice are performed between upper and lower limits. This is completely performed by depending on the user [13]. The chromosome fitness is then evaluated using the error performance criteria. They are previously explained in Chapter 3.

There are some points to be decided before applying GA in PID tuning. Some time is devoted for discussing the importance of encoding, information exchange and the population size for GA. Four selection methods are available; Roulette wheel, stochastic universal sampling, normalized geometric selection, and tournament selection. All of them are based on same principal. However, in some of reference studies [10-16], a common selection scheme is seen as Roulette Wheel. Thus it has been applied here to produce qualified individuals. Many independent trials are performed at the decision stage using GA Toolbox. In addition, Population type is also very important as the type of input to the fitness function. Three possibilities are seen as; double vector, binary string and custom. Here a random population 20-100 is used for initializing GA, each chromosome is represented by binary string using less memory. Since the fitness is related to performance measures as the integral error. PID tuning results found are tabulated for BLDC motor with four bar mechanism, a look up Table 3.3. The GA parameters used in the model are given in the following section.

4.3 GA based Results-Four Bar Mechanism

The GA is tested by using different motion applications. Firstly, PID controller parameters are tuned with traditional methods. Reference, PID and GA based PID controller representations are used respectively in figures. Equations of the system are solved by using Runge-Kutta method. Gain values are taken approximately with trial and error. Actually they are initially found by using settings of Ziegler-Nichols [25]. Pascal programming language is used for the solution of model. Many trials are performed; here 10 trials are presented to show PID gains for BLDC motor driven four bar mechanism. Transfer function of the system is derived using the system equations. Linear system model is considered while applying algorithm for PID values. However, non-linear system equations are solved. Table 4.2 shows the parameters of GA applied in GA Toolbox.

Table 4.2. The Parameters of GA-Four Bar Mechanism

GA Parameters	*BLDC servo motor*
Crossover probability	0.1
Mutation probability	0.01
Generation numbers	240
Population size	80
Selection type	Roulette wheel

Figure 4.4 shows the PID parameter gains for BLDC motor/four bar combination. During numerical simulations, it is seen that all values found during trials have given satisfaction. One of them is taken and plotted here.

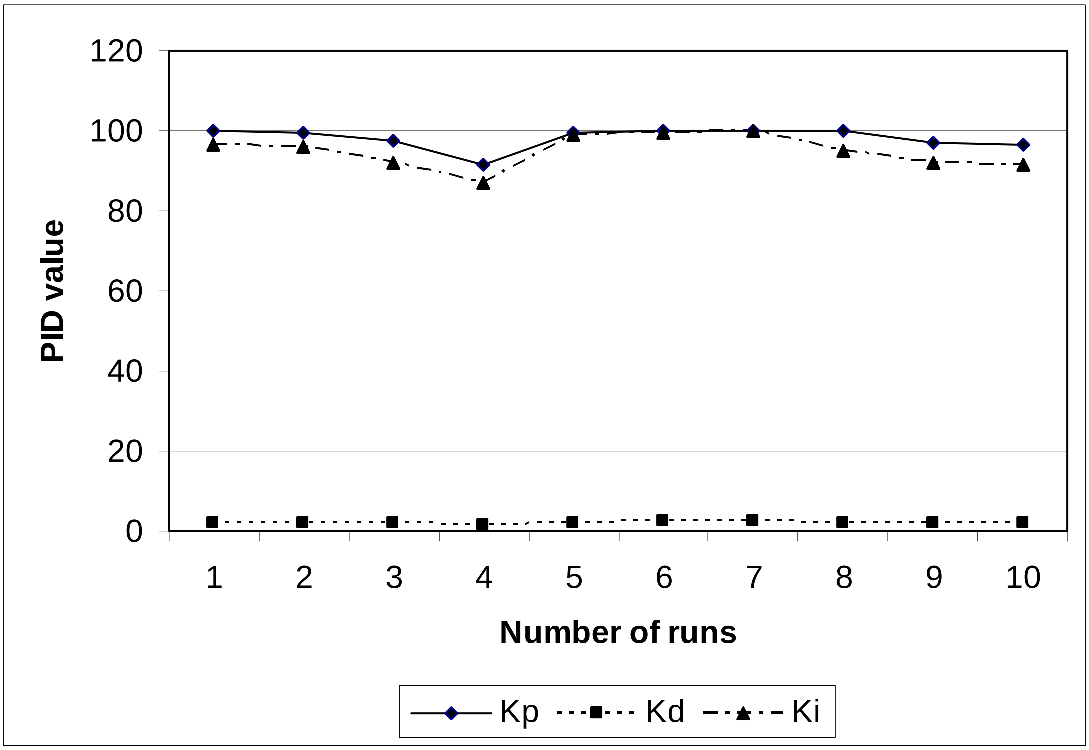
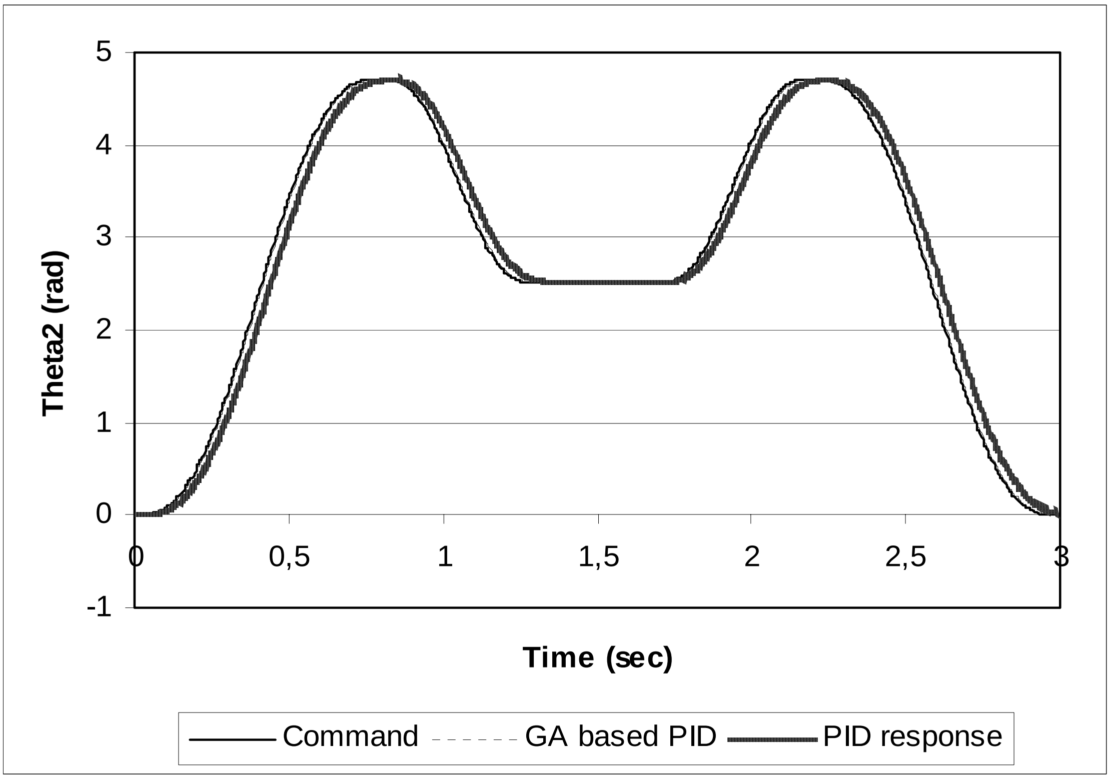


Figure 4.4 PID Controller Gains-Four Bar Mechanism/ 10 runs

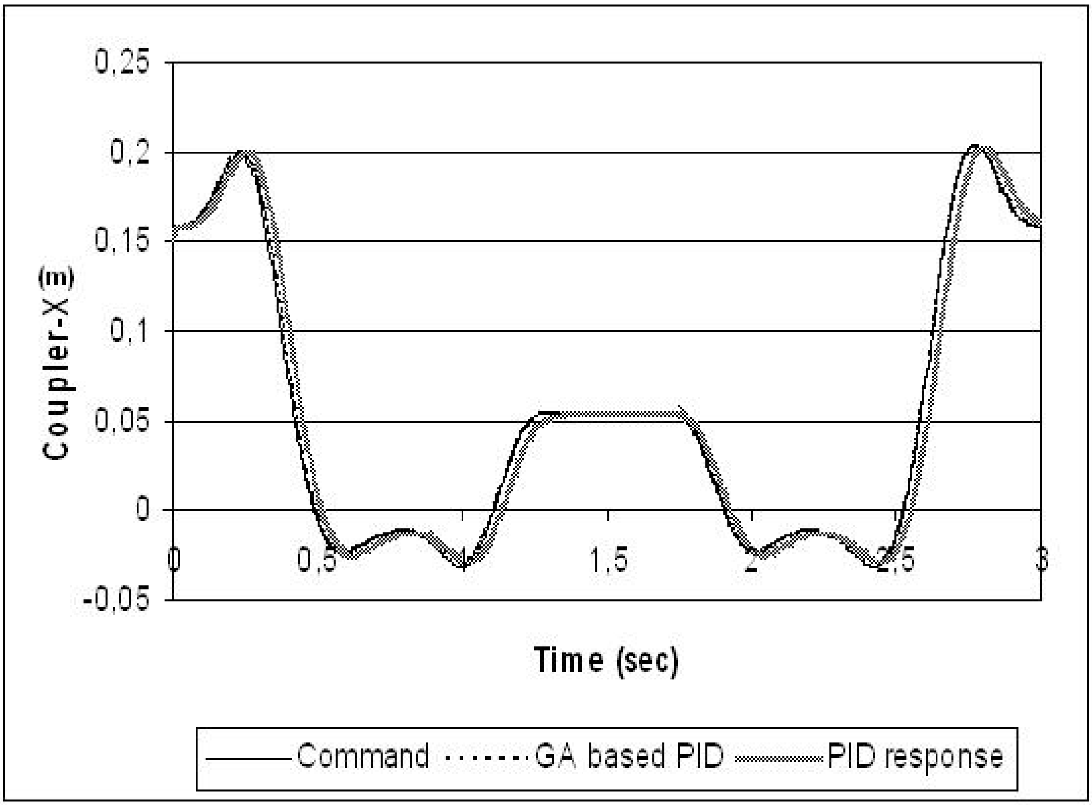
Whole crank motion is performed in 3 seconds. During integration, 720 points are used. GA based PID parameters are taken by using ISE error as K_p=99.96, K_d=5.22 and K_i=97.49. The coupler points are also plotted. They are calculated referring to Figure 4.3 by using schematic drawing as

$$E_x = a_2 \cos\theta_2 + p\cos\theta_3 - q\sin\theta_3$$
$$E_y = a_2 \sin\theta_2 + p\sin\theta_3 + q\cos\theta_3 \qquad (4.5)$$

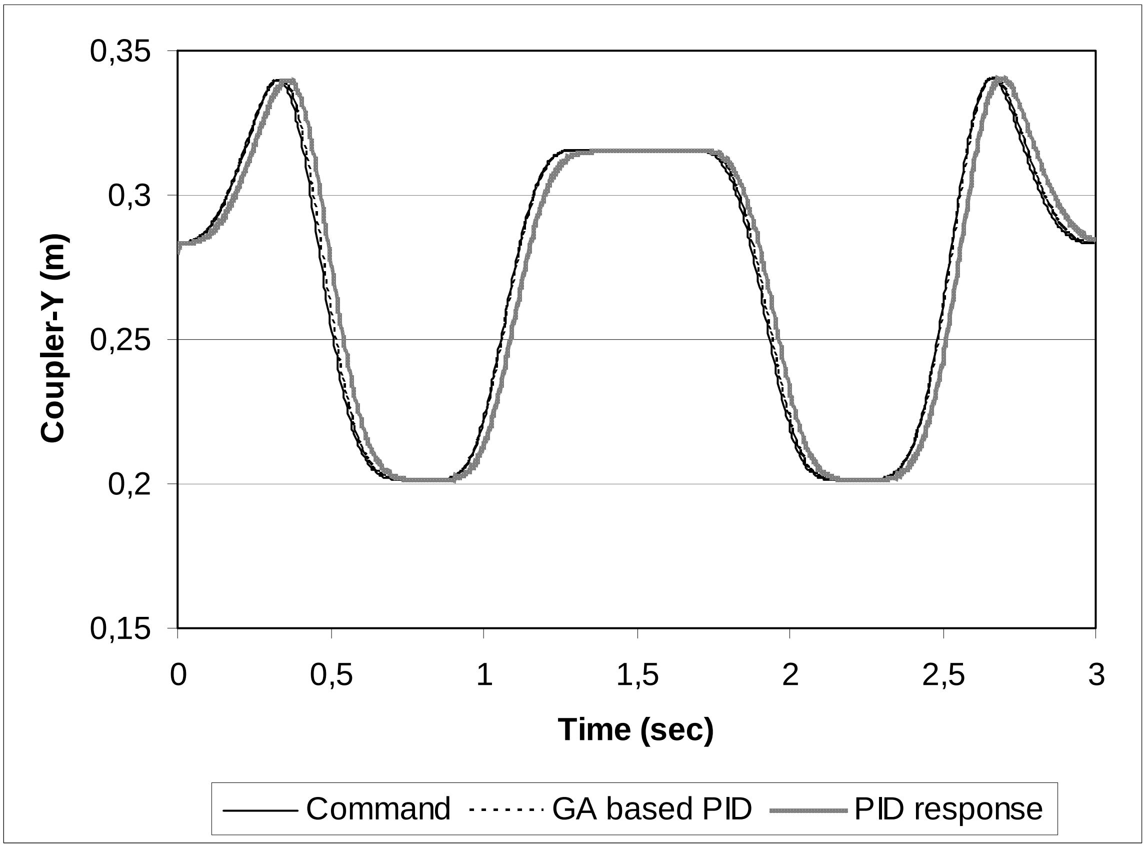
Figures 4.5(a), 4.5(b) and 4.5(c) show the motion curves and the coupler curves resulted in this application. A motion including rise, dwell and return is designed and taken. Graphics are plotted in order of the command, GA based PID and PID response respectively.



(a) The crank motion



(b) Coupler points (X)



(c) Coupler points (Y)

Figure 4.5 GA based Response and Coupler Curves

4.4 The Hybrid Actuator

The idea of hybrid machine is not new, initially studied by Tokuz and Jones [38]. An arrangement combining the motions from a constant speed motor and a servo motor with a differential gear unit is used to drive a system, like a slider-crank mechanism. This configuration is used for a model of soap bar embossing. Other alternative applications are counted to be packaging and printing machinery to improve overall performance. The following benefits are offered for hybrid machines as reduced changeover time with programmability, reduced servomotor size and improved motion tuning. Greenough et al [39] have presented a study on design of hybrid machines with a literature review on the subject. A research overview on hybrid machine configurations can be found in the following studies. Kirecci et al [40] have later explored a study on a hybrid actuator proposing an arrangement. Kirecci et al [41] have then presented a study on motion design and implementation for a hybrid drive system. A large constant speed motor is used with a small servo motor to get a programmable output at the end. Dülger et al [42] have studied on modeling and simulation of this so called hybrid actuator. Detailed study on the kinematics and dynamics are also included to explore their potential use. Mathematical model was derived by using Lagrangian approach. A simulation work was also carried out by using a traditional PID controller on both motors in use (BLDC and PMDC motors). Here this mechanism is taken and used with a PID controller tuned by GA.

4.4.1 Control Structure for the Hybrid Actuator

The motions of two characteristically different electric motors are combined in this configuration. Since the hybrid actuator is previously built as a research project, its dimensions are taken directly on it, Dynamic System Laboratory, Gaziantep University [40]. Hybrid actuator represents MISO system. It is drawn in assembly view in 3-d for the sake of clarity with its real dimensions, and shown in Figures 4.6.(a) and 4.6.(b) respectively. Figure 4.6.(b) represents rotated top view of the actuator system. The control structure is also represented in Figure 4.7. Position control on both motors is performed by incremental encoders with negative feedback in experimental system. A previously built experimental set up is taken during the study.

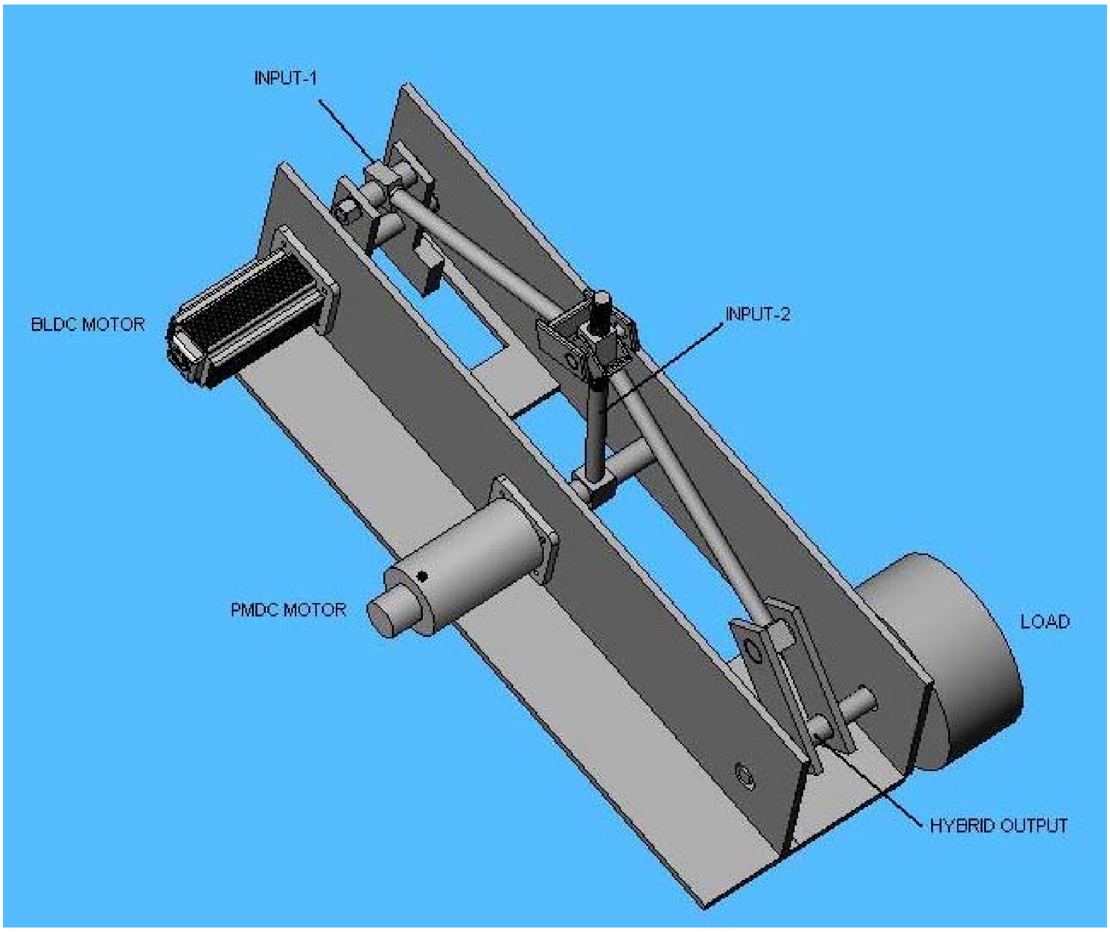


Figure 4.6 (a) An assembly view for the hybrid actuator

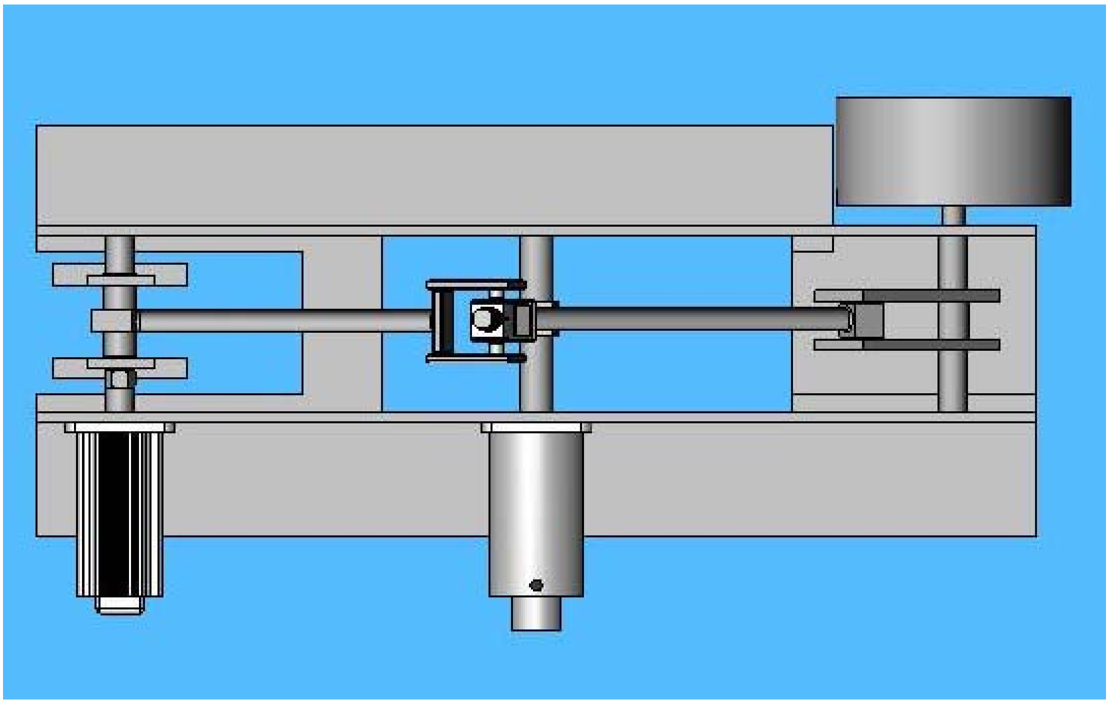


Figure 4.6 (b) Rotated (Top view)

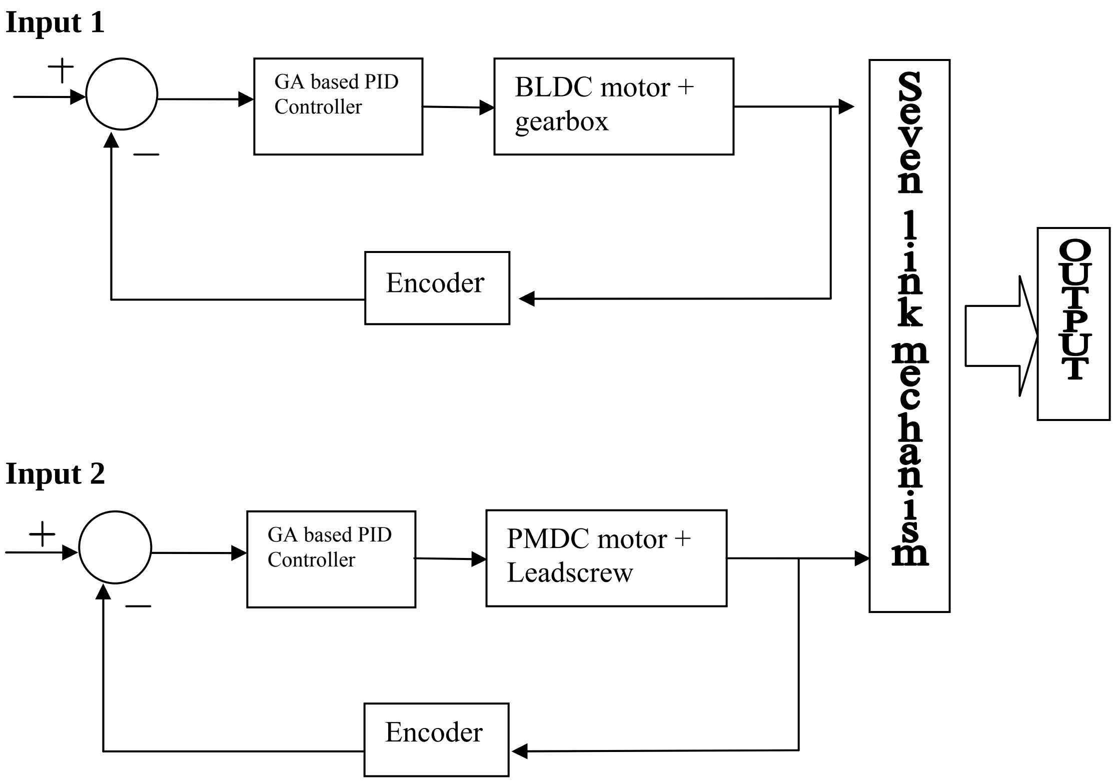


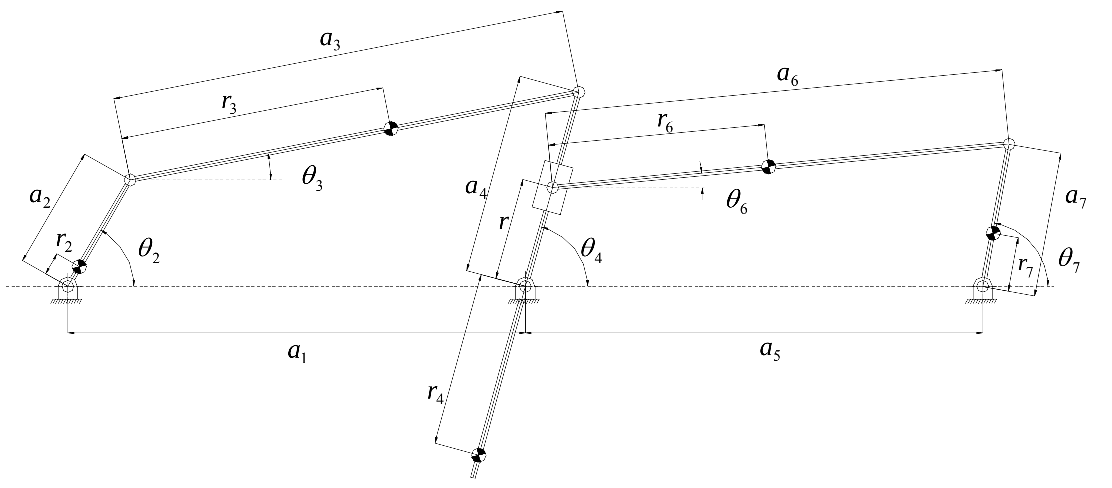
Figure 4.7 Structure of Hybrid Actuator Control

4.4.2 Mathematical Structure of Hybrid Actuator

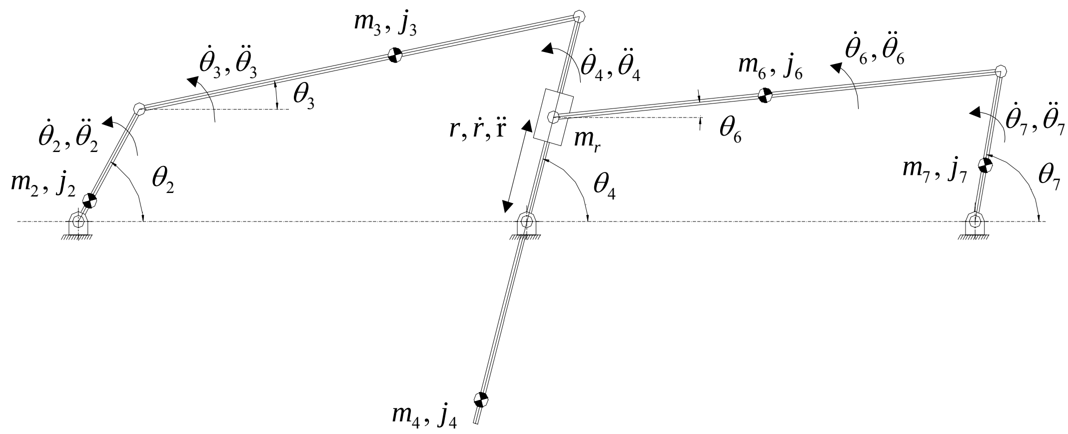
Mathematical structure of hybrid actuator includes two parts; the seven link mechanism and the electrical actuators (motors). Kinematics and dynamic issues are considered together while using the system equations.

4.4.2.1 Seven link mechanism

Kinematics of seven link mechanism referred is previously studied [40-42]. In this configuration, two four bar mechanisms are connected together such that 2^{nd} input has an adjustable crank and is provided by a power screw mechanism during application. System output is taken by link 7. Seven link mechanism is shown in Figure 4.8.(a) and 4.8.(b) schematically with notations used in the model.



(a)



(b)

Figure 4.8 Seven link mechanism referred [40-42]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7$ - link lengths of the mechanism

$\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7$ - angular displacement of the links and the ground (rad)

$\dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4, \dot{\theta}_6, \dot{\theta}_7$ - angular velocity of the links (rad/s)

$\ddot{\theta}_2, \ddot{\theta}_3, \ddot{\theta}_4, \ddot{\theta}_6, \ddot{\theta}_7$ - angular acceleration of the links (rad/s^2)

r_2, r_3, r_4, r_6, r_7 - positions to the centre of gravity (mm)

m_2, m_3, m_4, m_6, m_7 - masses of the links (kg)

j_2, j_3, j_4, j_6, j_7 - link moment of inertias (kg.mm^2)

j_{m1}, j_{m2}, j_g, - motor and gearbox moment of inertias (kg.mm^2)

m_r, j_r - mass and moment of inertia of the slider (kg, kg.mm^2)

P - pitch of the lead screw (mm)

$r, \dot{r}, \ddot{r}$ - displacement, velocity, and acceleration of the slider on link4 (mm, mm/s, mm/s^2).

4.4.2.2 The System Equations

This section presents a previously established dynamic model of seven link mechanism with actuators using Lagrange's method before numerical simulation. Mathematics involved while deriving the equations of motion is included in [42]. Here they are all taken from the previous work and arranged as the following. The actuator has 2 degrees of freedom representing a MISO system in application. The equations are given for the first and the second axes as well. Since the actuator dynamics is to be included here, initially the electrical motor equations for BLDC and PMDC motor are written. The equations describing the dynamic behavior of the motors include the armature's electrical and the dynamic equation.

The electrical equation of a brushless and permanent magnet dc motor [32] is:

$$L_k \frac{dI_k}{dt} + R_k I_k + K_{e_k} \dot{\theta}_n = V_k \tag{4.6}$$

Where L_k is the motor inductance , R_k is the motor resistance, K_{ek} is the motor voltage constant, I_k is the current passing through the armature, and V_k is the voltage applied to the armature. This voltage will be calculated with PID controller in which its parameters will be found by applying GAs for both motors in optimum. Here k is taken as an index 1 and 2 for BLDC and PMDC servo motors respectively. θ_n is replaced by two generalized coordinates; θ_{2m} and θ_s during simulation for the input 1 and input 2 respectively.

The equation of motion is expressed in the vector-matrix form for the first axis as:

$$\begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} & M_{15} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_{2m} \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \\ \ddot{\theta}_6 \\ \ddot{\theta}_7 \end{bmatrix} + [C_{11} \quad C_{12} \quad C_{13} \quad C_{14} \quad C_{15}] \begin{bmatrix} \dot{\theta}_{2m} \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \end{bmatrix} + [G_1] = [\tau_1] \tag{4.7}$$

where

$$M_{11} = \left[J_{m1} + J_g + \left(\frac{m_2 r_2^2 + J_2}{N^2} \right) \right] M_{12} = \left[(m_3 r_3^2 + J_3) \left(\frac{\partial \dot{\theta}_3}{\partial \dot{\theta}_{2m}} \right) \right]$$

$$M_{13} = \left[(m_4 r_4^2 + J_4) \left(\frac{\partial \dot{\theta}_4}{\partial \dot{\theta}_{2m}} \right) \right] M_{14} = \left[(m_6 r_6^2 + J_6) \left(\frac{\partial \dot{\theta}_6}{\partial \dot{\theta}_{2m}} \right) \right]$$

$$M_{15} = \left[(m_7 r_7^2 + J_7) \left(\frac{\partial \dot{\theta}_7}{\partial \dot{\theta}_{2m}} \right) \right]$$

and

$$C_{11}=0 \quad C_{12} = \left[(m_3 r_3^2 + J_3) \frac{d}{dt} \left(\frac{\partial \dot{\theta}_3}{\partial \dot{\theta}_{2m}} \right) \right] C_{13} = \left[(m_4 r_4^2 + J_4) \frac{d}{dt} \left(\frac{\partial \dot{\theta}_4}{\partial \dot{\theta}_{2m}} \right) \right]$$

$$C_{14} = \left[(m_6 r_6^2 + J_6) \frac{d}{dt} \left(\frac{\partial \dot{\theta}_6}{\partial \dot{\theta}_{2m}} \right) \right] C_{15} = \left[(m_7 r_7^2 + J_7) \frac{d}{dt} \left(\frac{\partial \dot{\theta}_7}{\partial \dot{\theta}_{2m}} \right) \right]$$

and the gravitational terms

$$G_1 = m_2 g r_2 \cos\left(\frac{\theta_{2m}}{N} \right) \frac{1}{N} + m_3 g \left[a_2 \cos\left(\frac{\theta_{2m}}{N} \right) \frac{1}{N} + r_3 \cos\theta_3 \left(\frac{\partial \theta_3}{\partial \theta_{2m}} \right) \right] - m_4 g r_4 \cos\theta_4 \left(\frac{\partial \theta_4}{\partial \theta_{2m}} \right) +$$
$$m_r g r \cos\theta_4 \left(\frac{\partial \theta_4}{\partial \theta_{2m}} \right) + m_r g \sin\theta_4 \left(\frac{\partial r}{\partial \theta_{2m}} \right) + m_6 g \left[r \cos\theta_4 \left(\frac{\partial \theta_4}{\partial \theta_{2m}} \right) + r_6 \cos\theta_6 \left(\frac{\partial \theta_6}{\partial \theta_{2m}} \right) \right] +$$
$$m_7 g r_7 \cos\left(\frac{\partial \theta_7}{\partial \theta_{2m}} \right)$$

and the relation between the motor and the crank is $N=\theta_{2m}/\theta_2$ as the coupling ratio. τ_2 represents the torque generated by the motor.

For the second input axis, the equation of motion is expressed in vector-matrix form as

$$\begin{bmatrix} N_{11} & N_{12} & N_{13} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_s \\ \ddot{\theta}_6 \\ \ddot{\theta}_7 \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12} & D_{13} \end{bmatrix} \begin{bmatrix} \dot{\theta}_s \\ \dot{\theta}_6 \\ \dot{\theta}_7 \end{bmatrix} + [G_2]=[\tau_2] \qquad (4.8)$$

$$N_{11}=\left[m_r\left(\frac{P}{2\pi}\right)^2+J_r+J_{m2}\right] N_{12}=\left[\left(m_6r_6^2+J_6\right)\left(\frac{\partial\dot{\theta}_6}{\partial\dot{\theta}_s}\right)\right] N_{13}=\left[\left(m_7r_7^2+J_7\right)\left(\frac{\partial\dot{\theta}_7}{\partial\dot{\theta}_s}\right)\right]$$

$D_{11}=0$ $$D_{12}=\left[(m_6r_6^2+J_6)\dot{\theta}_6\frac{d}{dt}\left(\frac{\partial\dot{\theta}_6}{\partial\dot{\theta}_s}\right)\right] D_{13}=\left[(m_7r_7^2+J_7)\dot{\theta}_7\frac{d}{dt}\left(\frac{\partial\dot{\theta}_7}{\partial\dot{\theta}_s}\right)\right]$$

And the gravitational terms

$$G_2=m_r gr\cos\theta_4\left(\frac{\partial\theta_4}{\partial\theta_s}\right)+m_6 g\left[r\cos\theta_4\left(\frac{\partial\theta_4}{\partial\theta_s}\right)+r_6\cos\theta_6\left(\frac{\partial\theta_6}{\partial\theta_s}\right)\right]+m_7 gr_7\cos\theta_7\left(\frac{\partial\theta_7}{\partial\theta_s}\right)$$

and a linear relation between the motor and link 5 (r) is $\theta_s=(2\pi/P)r$. In equations (4.7) and (4.8), τ_1 ans τ_2 represent the torques generated by the motors. These torques are calculated by using $\tau_1=K_{t1}I_1$ and $\tau_2=K_{t2}I_2$ by solving equation (4.6) for both servo motors with the equations (4.7) and (4.8).

The system equations are represented by the equations (4.6), (4.7) and (4.8). Equation (4.6) is then modified by using the motors electrical data, BLDC and PMDC. So the equation (4.6) describes BLDC and PMDC mathematics with given motor data with PID controller (Figure 4.6). In simulation work, equations (4.6), (4.7) and (4.8) are taken to get the electrical and dynamic structure of the system representing nonlinear characteristics.

4.4.2.3 Application of GAs to PID Controller

The procedure is similar one performed in previous example. In use of GA with Toolbox, each chromosome has the genes, the proportional, derivative and integral gain values. So it is represented by Chromosome=[K_p K_d K_i]. Chromosome fitness is evaluated using fitness IAE method. So here the best tuning is related to the smallest IAE, which corresponds to the position error for PMDC motor. ISE performance criterion is applied for BLDC motor The objective function is minimized by genetic algorithm utilizing population size of 80. The process is terminated after 240 generations. The choice of crossover rate and mutation rate are dependent upon the nature of the objective function. As GA progresses, the optimum values are found for the same transfer function in each run. Roulette wheel selection with binary coding is chosen [13-16]. Similar procedure is performed like in previous application.

4.5 GA based Results - The hybrid actuator

Mechanical properties of seven link mechanism; link lengths, positions to the centre of gravity of each link, link masses, and link inertias are shown in Table 4.3. BLDC and PMDC servo motor data taken from motor manufacturers catalogues [32] are given in Table 3.2 and Table 3.5 respectively (Chapter 3).

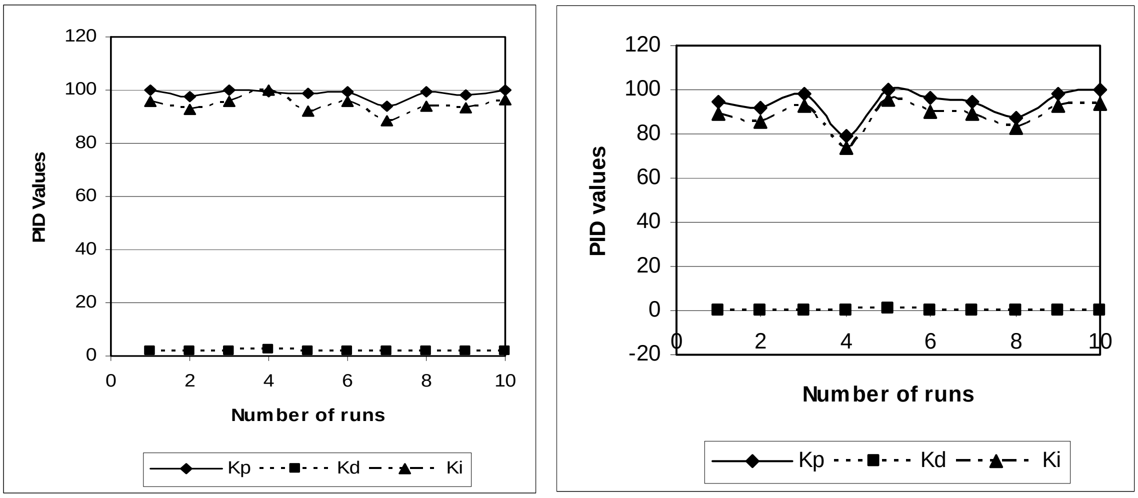
Table 4.3 Mechanical Properties of Seven Link Mechanism

A_1= 375 mm	a_2=102 mm	A_3=375 mm	a_4=167 mm
A_5=375 mm	a_6=375 mm	A_7=120 mm	
R_2=19 mm	r_3= 218 mm	r_4= -145 mm	r_6=178 mm
R_7= 45 mm			
M_2=3.9 kg	M_3=1.385 kg	m_4=13.295 kg	m_6=1.1 kg
M_7=25.186 kg	m_r= 0.425 kg		
j_2=12300 kg.mm^2	j_3=24900 kg.mm^2	j_4=34300 kg.mm^2	j_6=18700 kg.mm^2
j_7=131500 kg.mm^2	j_r=263 kg.mm^2	j_g=15000 kg.mm^2	P=1 mm

Initially the hybrid actuator system is thought separately as a BLDC and a PMDC driven linear load. Related transfer functions are derived for each motor. The parameters used in GA are given in Table 4.4. For example; many independent trials are performed, 10 independent trials are chosen for each motor. Parameters for PID are then shown. While performing GA algorithm, motor and fixed load terms are taken as a start. Equations (4.7) and (4.8) for each axis dynamics, M_{11} and N_{11} are calculated using real mechanism parameters. Figure 4.9.(a) and 4.9.(b) show Kp, Kd and Ki values to see their distribution in different runs. For each gain value of PID, the motors can definitely show satisfactory responses for step inputs. Here simulation results are presented by taking only one of them from look up Table. (Chapter 3)

Table 4.4 The Parameters of Genetic Algorithm

	BLDC servo motor	*PMDC servo motor*
Crossover probability	0.1	0.1
Mutation probability	0.01	0.01
Generation numbers	240	240
Population size	80	80
Selection type	Roulette wheel	Roulette wheel



(a) BLDC motor **(b) PMDC motor**

Figure 4.9 PID controller gains for Hybrid Actuator/ 10 different runs

Simulations on the system responses for both axes applying unit step input clearly have shown the efficiency of the method. MATLAB and in some parts, Pascal are applied for solution of the systems equations. In the responses obtained using GA based tuning, no overshoot is required. GA tuning is then tested with the hybrid system's equations with the parameters given in Table 4.3, Table 3.2 and Table 3.5. Better transient responses are definitely seen. Although the system equations are coupled and nonlinear, tuning values performed are given very good responses from both motors. BLDC servo motor rotates at 600 rpm with 1/30 reduction unit, so the crank has 20 rpm at the end. Whole motion includes 360 points representing 3 seconds of execution time. The output motion is designed using motion design software [43]. MATLAB codes used in GA are not given here [24].

The tracking performance of both axes is shown in Figure 4.10 (a) and 4.10 (b). Figure 4.10(c) also shows the output motion designed for the hybrid actuator. All results are plotted against time. It has been obtained by using hybrid configuration (Figure 4.1) and the motor responses from Figure 4.10 (a) and 4.10 (b). The tracking outputs are presented for the 1^{st} cycle referring to given initial conditions for both motors. Here GA conducts the tuning process and ends with optimum PID values for each representative transfer function. During simulations, ISE error calculation is performed for BLDC motor, and GA based PID parameters are found as K_p=99.76, K_d=1.98 and K_i=95.49. Similarly, IAE error criterion has been applied for PMDC motor, GA based PID parameters are taken as K_p=98.09, K_d=1.82 and K_i=91.98. It can be seen that both motors follow a predetermined track to acceptable accuracy and achieve good performance under variable load with non-linear equations. Conventional PID results are taken from previous study [42]. They are given as command, PID and GA based PID respectively.

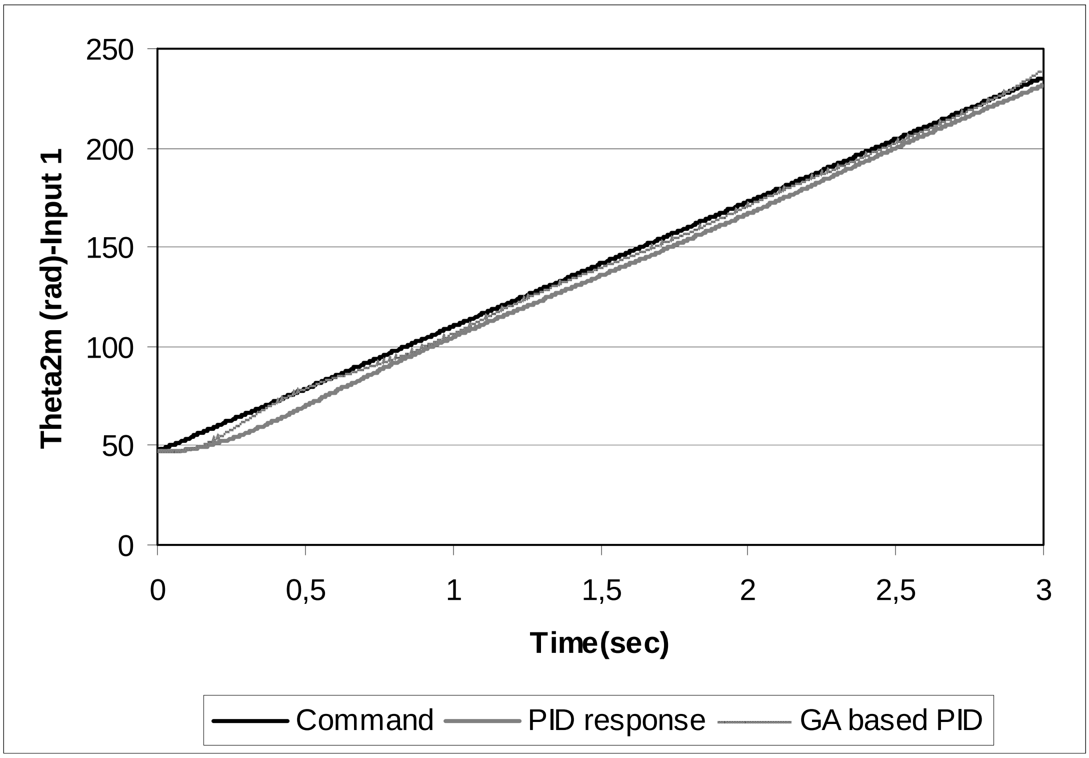


Figure 4.10 (a) BLDC Motor

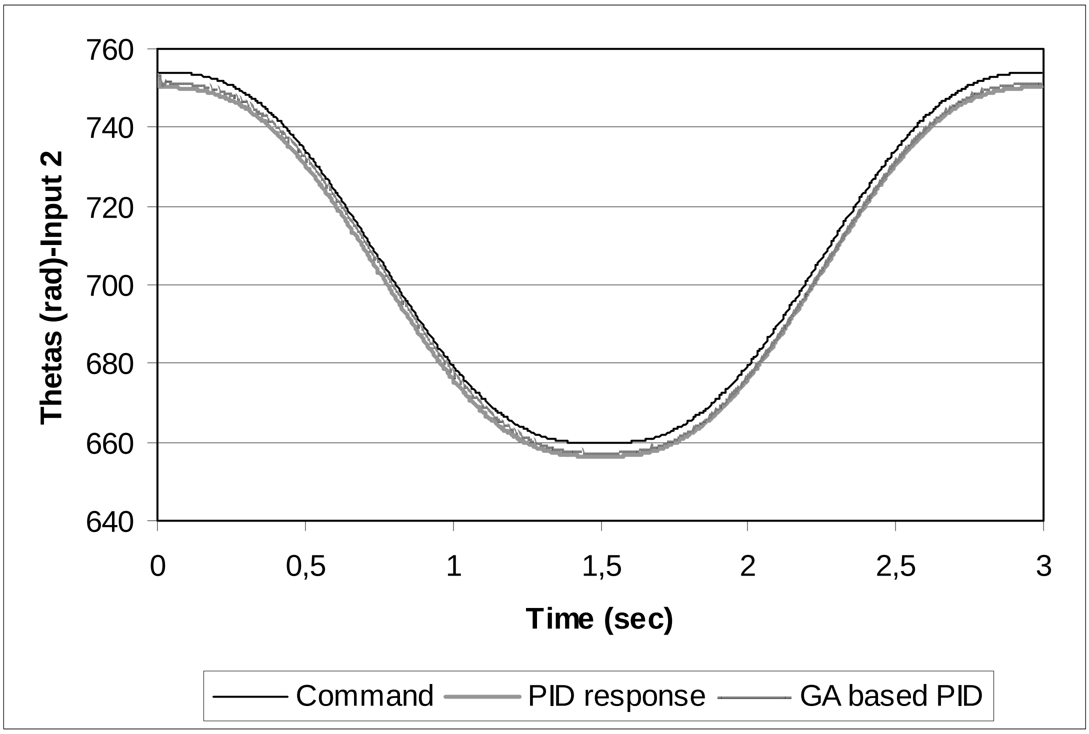


Figure 4.10 (b) PMDC Motor

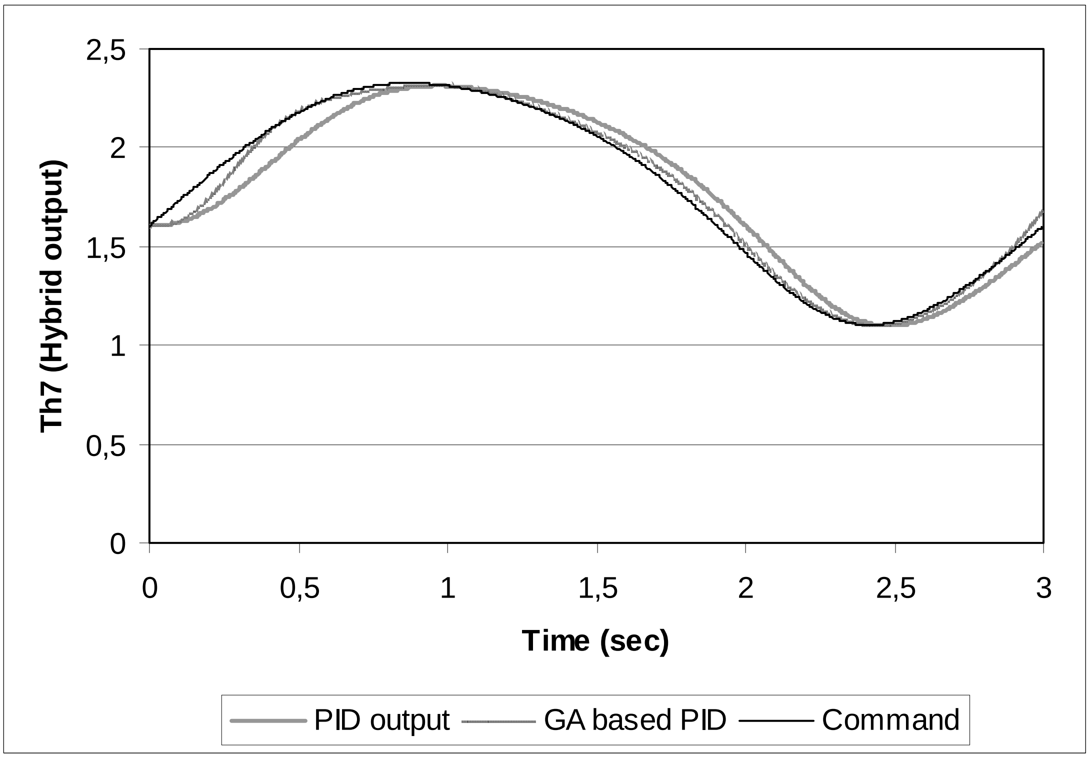


Figure 4.10 (c) The Hybrid Output

Figure 4.10 GA based BLDC and PMDC motor responses and Hybrid Output

4.6 Discussions on position control applications

Two different mechanism applications in position control; four bar mechanism and the hybrid actuator are studied in this chapter. A procedure involving the control and mathematical structure of both systems are presented in detail. Position control for both systems are definitely performed by tuning PID controlller with GAs. Angular position of each system are presented through Figures 4.5 (a, b, c) and Figures 4.10 (a, b, c). Two different motor structures are used in the Hybrid Actuator, BLDC motor performs a ramp output corresponding to constant velocity (ISE error) and PMDC motor performs a modulated profile (IAE error) to get the hybrid output designed. Comparisons are performed here with conventional PID versus GA based PID.

Tracking performances are acceptable for both axes. This tuning can be applied to different planar mechanisms if required. Transients are happening at the beginning, as seen from the figures. In simulation, motion profiles are run for 2 motion cycles. No transients are observed in the 2^{nd} cycle. Here only one motion cycle is presented for both systems.

CHAPTER 5

CONCLUSIONS

5.1 Discussions on the Present study

This study has presented use of GA for PID controller tuning by using computer simulation. The development of thesis is built in five chapters. The work is started with an introduction to GA and motion control issues. A brief survey is included about GA and its application on tuning of controllers in control engineering. Having given PID application by GA toolbox MATLAB, position control examples are presented using two different configurations; four bar mechanism and the hybrid actuator. These mechanisms were designed and studied in previous projects and available at Dynamic Systems Laboratory. They are reused and drawn in Solid Works with their original dimensions and their position control is performed by GA here. Two different types of motors are used to drive these mechanisms; BLDC motor and PMDC motor. Several trials are performed for both motors to obtain the best result in optimization. Four different types of error criteria's are used; MSE, ITAE, IAE and ISE. All error criteria's have given good results. Having performed many independent runs for both motors, ISE and IAE error criteria's are taken for our application. Simulation results have demonstrated effectiveness of the algorithm in tracking desired position trajectory.

Conclusions on this study are summarized with three headings as;
GA in control, GAs tuning of electrical actuators with mechanisms and problems seen in GAs application.

(i) About GA in Control

GA is a method for solving optimization problems and it is based on Natural Selection. It is a stochastic global search method for optimization problems using Evolution Operators. GAs has been successfully applied to many problems in engineering. GAs have been addressed in parameter and system identification, control to improve the overall system performance, robotics in navigation, pattern recognition, speech recognition, engineering designs, planning and scheduling. In addition to these, studies on the application of GAs in control engineering are seen in a wide range of control problems; PID control, optimal control, adaptive control, robust control and identification.

Many successful applications of GAs have been seen in PID controller design. It is simply to perform fitness. GA is implemented in MATLAB. Two important points are noted as chromosome coding and definition of evaluation criteria. It is seen that by using commands in GA toolbox, optimization can be done to different types of problems without any difficulty. The problem just becomes to understand Toolbox and GA itself. Although the system is a highly nonlinear one, regular tuning based on a linear model has given satisfaction in transient response. It can definitely handle position control of the systems. So GA can definitely result in a powerful tool to solve optimum controller tuning.

(ii) About GAs tuning of electrical actuators with mechanisms

This study has attempted to present an application in position control of planar mechanisms with actuator dynamics. GAs are showing a good potential for tuning electric drives with different characteristics, as BLDC and PMDC motor. Both motors are used to follow a desired position profile with variable load. They can be definitely used for tuning of controllers by addressing any dynamic system of interest in different industrial applications.

The use of GA with hybrid actuator control is also ready to get more attention. Once the actuators are tuned with PID parameters in linear model, the hybrid system achieves position control under real loading examples. The control system follows the command closely. Many nonlinear factors are then imposed by the load and drive. In general, high nonlinearity of system equations are not caused any problem during implementation.

(iii) Problems seen in GAs application

GA is different from other optimization methods; it tries to find the solution in a group of solution not in one point. GA codes the parameters and it does not use the parameters directly, GA does not contain the information about how it solves the problem, it is a blind method. It can be applied to different types of problems but its limits of its use has to be known well.

GA is generally seen in off line implementation. Some studies were performed on line issues. But they are limited compared to off line ones. Another problem comes with its real time application. GA needs a lot of computation each time. Its real time realization must be done on the physical system.

5.2 Suggestions for further work

Although this study can be regarded as a tool for PID controller with GA, there must be an experimental part to verify simulation results in a real system. The mechanisms were already available at the laboratory. However, motor control systems were not working at the moment. Therefore experimental set up was not operated here. This can be considered as a shortcoming in this study. In previous studies, these mechanisms were actually driven and tested with traditional PD and PID controller.

There can be also a study based on hybrid computation using other evolutionary algorithms on controller structure GA-Fuzzy, GA-NN and GA-PSO for example. Some successful studies are noted on this subject. GA performance is improved with hybrid evolutionary algorithms. Later some of these algorithms can be combined and studied. Then they can be applied for position control of mechanisms like one given here.

REFERENCES

[1] D. Goldberg, ‘Genetic Algorithms in Search, Optimization and Machine Learning’, Addison-Wesley, Reading MA, 1989.

[2] David A Coley, ‘An Introduction to Genetic algorithms for Scientists and Engineers’, World Scientific Publishing, 2005.

[3] K.M.Man, K.S.Tang, S. Kwong, ‘Genetic Algorithms: Concepts and Applications’, IEEE Transactions on Industrial Electronics, Vol. 43, No:5, 1996, p.519-533.

[4] Zekai ŞEN, ‘Genetik Algoritmalar ve En iyileme yöntemleri’, İ.T.Ü Uçak ve Uzay Bilimleri Fakültesi, Su Vakfı Yayınları, 2004

[5] W. Bolton, ‘Mechatronics: Electronic Control Systems in Mechanical and electrical Engineering, McGraw Hill, 2003.

[6] C. Fraser, J. Milne, ‘Integrated Electrical and Electronic Engineering for Mechanical Engineers’, McGraw Hill,1999.

[7] P.J. Fleming and C.M. Fonseca, ‘Genetic Algorithms in Control Systems Engineering: A Brief Introduction’, The Institution of Electrical Engineers, p. 1-5, 1993.

[8] P.J.Fleming, R.C.Purshouse, IFAC Professional Brief, ‘Genetic algorithms in Control System Engineering’.

[9] Electro-Craft, Robbins and Myers, DC motors, Speed controls, Servo systems, 5th edition.

[10] Q. Wang, P. Spronck, R. Tracht, ‘An Overview of Genetic algorithms applied to control engineering problems’, Pr. Of 2nd Int. Conf on Machine Learning and Cybernetics-IEEE, November 2003, p.1651-1656.

[11] R.A. Krohling, H. Jaschek, J.P.Roy, ‘Designing PI/PID Controllers for a Motion Control System based on genetic algorithms’, Pr. of the 12th IEEE Symposium on Intelligent Control, (1997), p.125-130.

[12] Z.M. Al-Hamouz, H.N. Al-Duwaish, ‘A New Variable Structure DC Motor Controller Using Genetic Algorithms’, (1998), p. 1669-1674.

[13] I. Griffin, ‘On-line PID controller tuning using Genetic algorithms’, M.Eng in Electronic systems, 2003.

[14] C.L. Lin, H.Y. Jan, N.C. Shieh, ‘GA based Multi Objective PID Control for a Linear Brushless DC Motor’, IEEE/ASME Transactions on Mechatronics’, Vol.8, No.1, p.56-65, 2003.

[15] C. Xia, P. Guo, T. Shi, M.Wang, ‘Speed control of brushless dc motor using genetic algorithm based fuzzy controller’, Pr. Of the 2004 Int. Conf. on Int. Mechatronics and Automation, 2004-IEEE, p. 460-464.

[16] S. Bin Mohamed İbrahim, ‘The PID controller design using genetic algorithm’, University of Southern Queensland, October 2005.

[17] Ö. Gündoğdu, ‘Optimal Tuning of PID controller gains using Genetic Algorithms’, Pamukkale University, Journal of Engineering Sciences, Vol.11, No.1, 2005, p.131-135.

[18] B. Kristiansson, B.Lennartson, ‘Robust tuning of PI and PID controllers’, IEEE Control Systems Magazine, pp.55-69, February 2006,

[19] A Bagıs, ‘Determination of the PID controller parameters by modified genetic algorithm for improved performance’, J. of Information Science and Engineering 23 (2007), p.1469-1480.

[20] J.S. Kim, J.H. Kim, J.M. Park, S.M. Park, W.Y. Choe, ‘Auto Tuning PID controller based on Improved Genetic Algorithm for Reverse Osmosis Plant’, PWASET, Vol.30, p.1001-1006, 2008.

[21] Murat Sürücü, ‘Genetic Algoritma ile Mikrodalga Kuvvetlendiricisi Optimizasyonu’, Süleyman Demirel Üniversitesi Mühendislik Mimarlık Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümü, Haziran- 2002

[22] Gül Gökay Emel ve Çağatan Taşkın, Genetik Algoritmalar ve Uygulama alanları, Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, Cilt 21 Sayı 1;2002, s.129-152

[23] Oğuz Yakut, ‘Yapıların sismik izolasyonunda ieri denetim algoritmalarının uygulanması’, Fırat Üniversitesi, Doktora Tezi, Eylül 2007

[24] Matlab R2007a Genetic Algorithm Toolbox; Matlab User’s Guide

[25] R.C.Dorf, R.H. Bishop, ‘Modern Control Systems’, 10th Edition, Pearson Prentice Hall, 2005.

[26] K. Ogata, ‘Modern Control Engineering’, Fourth Edition, Prentice Hall, 2002.

[27] P. Cuminos, N.Munro, ‘PID Controllers: Recent Tuning methods and design to specification’, IEEE Proc. Control Theory Applications, 49,1, pp.46-56, 2002

[28] G.K.I. Mann, B.G.Hu, R.G.Coosine, ‘Time-domain based design and analysis of new PID tuning rules’, IEEE Proc. Control Theory Applications’, Vol.148, No.3, pp.251-261, 2001.

[29] Wikipedia, ‘PID Tuning Methods’

[30] J.A. Show, ‘PID Algorithm and Tuning Methods’, Rochester, New-York, http://www.jashow.com/pid/tutorial

[31] D. Xune, Y.Q. Chen, D.P.Atherton, Linear Feedback Control-PID Controller Design’, Society for Industrial and Applied Mathematics, 2007.

[32] Electro-Craft, Robbins and Myers, DC Motors, speed Controls Catalogue

[33] E. Söylemez, ‘Mechanisms’, METU,1999.

[34] J. E. Shigley, J.J. Vicker, ‘Theory of Machines and Mechanisms’, Second edition, Mc Graw Hill-1995.

[35] H.V.Vu, R.S. Esfandiari, ‘Dynamic Systems: Modeling and Analysis’, Mc Graw Hill-1997.

[36] L.C.T. Dülger, S. Uyan, ‚Modelling, Simulation and control of a four bar mechanism with a brushless dc motor’, Mechatronics, Vol./, No:4, pp.369-383, 1997.

[37] M.T.Daş, L.C.Dülger, ‘Particle Swarm Optimization (PSO) Algorithm: Control of a four bar mechanism’, Proceedings of EU/ME 2007-Metaheuristics in the Service Industry’, pp.96-101, 2007

[38] L.C.Tokuz, J.R.Jones, ‘Programmable modulation of motion using hybrid machines’, EUROTECH DIRECT 91, IMechE, Conference paper-C414/071.

[39] J.D.Greenough, W.K. Bradshaw, M.J.Gilmartin, S.S.Douglas, J.R.Jones, ‘Design of hybrid machines’,Proc. 9th World Congress on the theory of Machines and Mechanisms, Milan, pp. 2501-2505, 1995.

[40] A. Kirecci, L.C.Dülger, ‘A study on hybrid actuator’, Mechanism and Machine Theory, 35 (2000), p.1141-1149.

[41] A. Kirecci, L.C.Dülger, ‘Motion design and implementation for a hybrid drive system’, Proc. Instn. Mech. Engrs, Vol.217, Part I: J. Systems and Control Eng.,(2003), p.299-302.

[42] L.C.Dülger, A. Kirecci, M. Topalbekiroğlu, ‘Modeling and simulation of a hybrid actuator’, Mechanism and Machine Theory 38, 2003, p. 395-407.

[43] A Kirecci, M. Gilmartin, ‘Trajectory Planning for Robotic Manipulators’, 6th Int. Machine Design and Pr. Conference, METU-TURKEY, 1994.