

**OPTIMIZATION OF PID CONTROLLER FOR INVERTED PENDULUM  
SYSTEM USING GENETIC ALGORITHM**

By

NURUL FIRDAUS BINTI JAMALUDDIN

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme  
in Partial Fulfillment of the Requirements  
for the Degree  
Bachelor of Engineering (Hons)  
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas  
Bandar Seri Iskandar  
31750 Tronoh  
Perak Darul Ridzuan

© Copyright 2005

by

Nurul Firdaus Binti Jamaluddin, 2005

# CERTIFICATION OF APPROVAL

## OPTIMIZATION OF PID CONTROLLER FOR INVERTED PENDULUM SYSTEM USING GENETIC ALGORITHM

by

Nurul Firdaus Binti Jamaluddin

A project dissertation submitted to the  
Electrical & Electronics Engineering Programme  
Universiti Teknologi PETRONAS  
in partial fulfilment of the requirement for the  
Bachelor of Engineering (Hons)  
(Electrical & Electronics Engineering)

Approved:



Ms. Zazilah May  
Project Supervisor

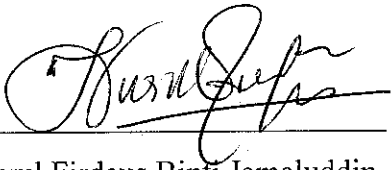
Zazilah May  
lecturer  
Electrical Engineering Department  
Universiti Teknologi PETRONAS  
750 Y...

UNIVERSITI TEKNOLOGI PETRONAS  
TRONOH, PERAK

December 2005

## CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

A handwritten signature in black ink, appearing to read 'Nurul Firdaus', written over a horizontal line.

Nurul Firdaus Binti Jamaluddin

## ABSTRACT

The *proportional-integral-derivative* controller or commonly known as PID Controller has been widely used in the industries since the 1940s and remains the most often used today. In this project, PID Controller of an Inverted Pendulum System is optimized using Genetic Algorithms (GA) approach. Currently, the Inverted Pendulum System as available in the laboratory is controlled by PID Controller. However, ample time is required by the pendulum to change its position from downward to upright and to be stabilized. Therefore, GA will be applied to overcome this problem. The main objective of this project is to find the optimum stable point, which is the optimum value of  $K_P$ ,  $K_I$  and  $K_D$  of the PID Controller using GA approach. The second objective of this project is to reduce the time required for the pendulum to be stabilized. In order to complete this project, a few stages need to be carried out. The stages include problem identification, research on GA, understand the principle of PID Controller and Inverted Pendulum, obtain stable region, create GA coding via MATLAB and conduct test on the real Inverted Pendulum System. Before optimization technique using GA can be applied, the stable region for the desired system needs to be obtained first. In this project, Nyquist Stability Criterion is utilized to obtain the stable region. Once the stable region is obtain, GA is then been applied where the optimum value of  $K_P$ ,  $K_I$  and  $K_D$  within the stable region are determined. For this project, MATLAB Software and Double Inverted Pendulum Trainer are required. Therefore, the understanding on those software and hardware are vital.

## ACKNOWLEDGEMENTS

First and foremost, thankfulness to Allah the Almighty who had given me the strength to complete this project. Without His help, this project will not come into reality.

I would like to dedicate my special thanks and appreciation to Ms. Zazilah May, my Project Supervisor who had tirelessly and relentlessly assisted me in conducting this project. Not forgetting the UTP lecturers, Ms. Noor Hazrin Hany Mohamad Haniff, Dr. Nordin Saad and Mr. Azhar Zainal Abidin, the Lab Technologist for their full support and guidance.

My deep appreciation also goes to my fellow friends Mohd. Fuad Zulkeflee and Maizura Ismail, for sharing precious knowledge and giving full support to me while conducting this project.

Last but not least, I would like to express my gratitude to my family who had given everlasting moral support.

## TABLE OF CONTENTS

ABSTRACT.....	i
LIST OF TABLES.....	v
LIST OF FIGURES.....	vi
LIST OF ABBREVIATIONS.....	vii
CHAPTER 1 INTRODUCTION.....	1
1.1 Background of Study.....	1
1.1.1 Introduction to optimization technique.....	1
1.1.2 Principle of Inverted Pendulum.....	2
1.1.3 Introduction to Genetic Algorithm (GA).....	3
1.2 Problem Statement.....	4
1.3 Objectives.....	5
1.4 Scope of Study.....	5
CHAPTER 2 LITERATURE REVIEW AND THEORY.....	6
2.1 Literature Review.....	6
2.2 Theory.....	10
2.2.1 Proportional Control.....	10
2.2.2 Integral Control.....	10
2.2.3 Derivative Control.....	10
2.2.4 Nyquist Stability Criterion.....	11
CHAPTER 3 METHODOLOGY.....	12
3.1 Procedure Identification.....	12
3.1.1 Problem Identification.....	13
3.1.2 Research on GA.....	13
3.1.3 Understand PID Controller and Inverted Pendulum System.....	13
3.1.4 Obtain stable region of the system.....	14
3.1.5 Create MATLAB Coding.....	16
3.1.6 Test on Inverted Pendulum Trainer.....	16
3.2 Tools Required.....	18
3.2.1 Hardware Requirements.....	18
3.2.2 Software Requirements.....	18
CHAPTER 4 RESULTS AND DISCUSSION.....	19
4.1 System Description.....	19

4.2 Performance of current PID Controller .....	22
4.3 Design Process.....	23
4.4 Obtaining Stable Region.....	24
4.5 Discussion.....	31
CHAPTER 5 CONCLUSIONS AND RECOMMENDATIONS.....	33
5.1 Conclusions.....	33
5.2 Recommendations for Future Work .....	35
REFERENCES .....	36
APPENDICES .....	37
APPENDIX A GENETIC ALGORITHM (GA).....	38
APPENDIX B INVERTED PENDULUM SYSTEM .....	41
APPENDIX C MATLAB CODING .....	45
APPENDIX D GANTT CHART OF COMPLETION .....	49

## LIST OF TABLES

<b>Table 1</b> Types of crossover in GA .....	4
<b>Table 2</b> Parameters of Double Inverted Pendulum System .....	21
<b>Table 3</b> Analysis of Nyquist Plot for $G(s) = \frac{(0.75s + 0.25)(4.365s^2 + 8.73s + 4.365)}{(s)(s^3 + 0.5739s^2 - 3.9652s + 2.2736)}$ .....	29
<b>Table 4</b> Allowable values of $K_p$ and $K_I$ for the Inverted Pendulum System .....	30



## LIST OF FIGURES

<b>Figure 1</b> Inverted Pendulum .....	2
<b>Figure 2</b> Roulette Wheel Selection .....	3
<b>Figure 3</b> Effect of mutation in GA.....	4
<b>Figure 4</b> PID control result when the GA is applied to select the PID gains.....	8
<b>Figure 5</b> The fitness value for Figure 3.....	8
<b>Figure 6</b> Minimum performance index of each generation .....	9
<b>Figure 7</b> Overall Project Flowchart .....	13
<b>Figure 8</b> Block diagram for a system using PID Controller .....	13
<b>Figure 9</b> Inverted Pendulum system using PID Controller and GA .....	14
<b>Figure 10</b> Flowchart for obtaining stable region .....	15
<b>Figure 11</b> Flowchart for GA programming.....	17
<b>Figure 12</b> Structure of the inverted pendulum system.....	20
<b>Figure 13</b> Cart Pole system.....	20
<b>Figure 14</b> Output from Inverted Pendulum System.....	22
<b>Figure 15</b> Typical block diagram for a system using PID Controller.....	23
<b>Figure 16</b> Inverted Pendulum System using PID Controller and GA.....	23
<b>Figure 17</b> Inverted Pendulum Subsystem .....	24
<b>Figure 18</b> Nyquist Plot for $G(s) = \frac{0.4635s}{s^3 + 0.5739s^2 - 3.9652s - 2.2736}$ .....	25
<b>Figure 19</b> Impulse Response for $G(s) = \frac{0.4635s}{s^3 + 0.5739s^2 - 3.9652s - 2.2736}$ .....	26
<b>Figure 20</b> Nyquist Plot for $G(s) = \frac{10(s+1)(s+1)0.4635s}{s(s^3 + 0.5739s^2 - 3.9652s - 2.2736)}$ .....	27
<b>Figure 21</b> Impulse Response for $G(s) = \frac{10(s+1)(s+1)0.4635s}{s(s^3 + 0.5739s^2 - 3.9652s - 2.2736)}$ .....	27
<b>Figure 22</b> Test Compensator.....	28
<b>Figure 23</b> Nyquist Plot for $G(s) = \frac{(0.75s + 0.25)(4.365s^2 + 8.73s + 4.365)}{(s)(s^3 + 0.5739s^2 - 3.9652s + 2.2736)}$ .....	28
<b>Figure 24</b> Stable region for $G(s) = \frac{4.365s^2 + 8.73s + 4.365}{s^3 + 0.5739s^2 - 3.9652s + 2.2736}$ .....	30

## LIST OF ABBREVIATIONS

PID Controller	Proportional Integral and Derivative Controller
GA	Genetic Algorithm
$K_P$	Proportional Term
$K_I$	Integral Term
$K_D$	Derivative Term

# CHAPTER 1

## INTRODUCTION

### 1.1 Background of Study

This report presents the utilization of GA approach to optimize the PID controller for the Double Inverted Pendulum System. In other words, the effectiveness of PID Controller will be enhanced using GA. Currently, PID controller is used to stabilize the inverted pendulum. However, in order to improve the performance, GA will be applied for optimization purpose. This section is divided into three subsections which mainly discussed about the optimization technique, inverted pendulum system and a brief introduction on the GA.

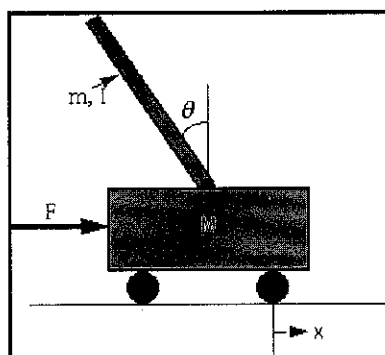
#### *1.1.1 Introduction to optimization technique*

Optimization can be defined as making something function at its best or most effective, or to use something to its best advantage. According to a book entitled “*Applied Optimization with MATLAB Programming*” by P. Venkayaraman, optimization is a branch of mathematics which is concerned with finding maxima and minima of real-valued functions.

GA is one of the optimization techniques used to enhance the performance of a system. With this technique, the best value for the objective function will be determined and hence it is also referred as the global optimum solution. Before the optimization process can be done, the basic principle of the Inverted Pendulum System and fundamental theory of GA need to be understood.

### 1.1.2 Principle of Inverted Pendulum

The inverted pendulum system is made up of a cart on top of which a pole is pivoted as shown in **Figure 1**. As mentioned before, GA is going to be applied to optimize the PID controller in order to find the optimum stable point of the Inverted Pendulum. The illustration of the inverted pendulum is shown below:



**Figure 1** Inverted Pendulum

Where;

- $M$  mass of the cart
- $m$  mass of the pendulum
- $I$  inertia of the pendulum
- $F$  force applied to the cart
- $b$  friction of the cart
- $l$  length to pendulum center of mass
- $\theta$  pendulum angle from vertical
- $x$  cart position coordinate

Initially, the inverted pendulum is pointing downward where rod hangs beneath the track. In order to self-erect the rod, the motor driven cart moves back and forth along the track. Once the rod of the pendulum rod is upwardly erect, the cart modifies its position as needed to balance the rod. Determination of the position of the cart on the track and angle of the rod are detected by two separate quadrature encoders. A weight which mounts on the cart must be used with the system in the self-erecting mode. If the weight is not used, the centrifugal force created by the swinging pendulum will cause the cart to lift off the track. Whenever disturbance is applied to the pendulum, the cart will be moving back and forth to stabilize the pendulum.

### 1.1.3 Introduction to Genetic Algorithm (GA)

GA is a useful search algorithm which is used for optimizing solutions to problems; especially those that are analytically intractable. GA is based on the mechanics of natural selection and natural genetics [1]. Basically, all living organisms consist of cells. In each cell there is the same set of chromosomes, which are the strings of deoxyribonucleic acid (DNA) and serve as a model for the organism. A chromosome consists of genes, blocks of DNA and each gene encodes a trait, for example colour of eyes [2]. In GA terminology, chromosomes can be said as string (solution) while genes are the feature or character of the solution.

In finding solution using GA, a few steps need to be taken. The steps involves are:

1. Population (set of potential solutions) need to be created. Each member of this set is called individual and evaluation need to be done upon them.
2. The fitness of each individual need to be calculated. The score that an individual achieves at performing the required task is called its fitness.
3. Selection is then performed. One of the popular selection algorithms is the “Roulette Wheel” [3]. A simple spin of Roulette Wheel yields new offspring. As in figure below, Chromosome 1 has the highest value of fitness and has more probability to be selected.

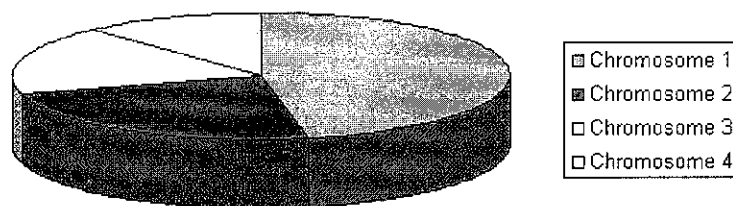


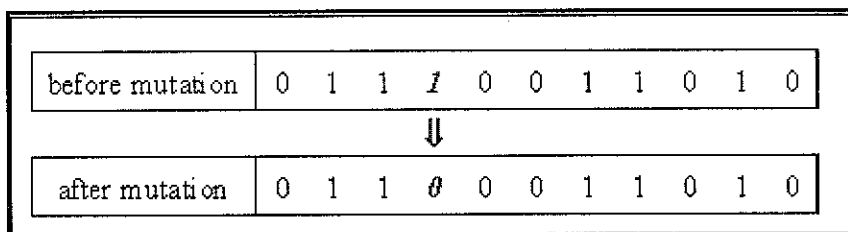
Figure 2 Roulette Wheel Selection

4. Once selection is finished, crossover is performed. There are three types of crossover which are Single Point Crossover, Multiple Point Crossover and Uniform Crossover. For Uniform Crossover, if bit of Mask is 1, Child 1 and Child 2 will follow the bit of Parent 1 and 2 respectively and vice versa if the bit of Mask is 0.

**Table 1** Types of crossover in GA

Parent 1 101 11100	Parent 1 10111100
Parent 2 110 10010	Parent 2 11010010
-----	-----
Child 1 101 10010	Mask 01001101
Child 2 110 11100	Child 1 10011110
	Child 2 11110000
<b>Single Point Crossover</b>	<b>Uniform Crossover</b>

5. The next operation that occurs is mutation, the random changing of bits in the chromosome. If mutation occurs to bit 1, the resulting bit is 0 and vice versa.



**Figure 3** Effect of mutation in GA

Step 3 until 5 will be repeated until the termination criteria is satisfied. The steps involved in realizing GA can be summarized in a flowchart as in Appendix A. For a clearer understanding, please refer to Appendix A.

## 1.2 Problem Statement

Currently, PID Controller is used to stabilize the inverted pendulum available in the laboratory. However, ample of time is required by the pendulum to change its position, from pointing downward to pointing upright and stabilized if PID Controller is utilized. Therefore, a mean of searching the best value for PID parameters, which is the optimum point need to be developed in order to reduce the time taken by the pendulum.

### **1.3 Objectives**

The objectives of this project are:

- To study and understand the behavior of current PID Controller for Inverted Pendulum System
- To understand the principle of GA
- To find optimum stable point for Inverted Pendulum by using GA approach
- To reduce the time required by the pendulum to erect and stabilize

### **1.4 Scope of Study**

The student will be concentrating on the GA to optimize PID Controller of the inverted pendulum system. The type of pendulum that will be considered is the Double Inverted Pendulum as available in the laboratory. A period of two semesters is allocated for this project. In the first semester, the student is required to carry out research, where the student need to have the understanding of GA and how it can be applied to find the optimum stable point for the inverted pendulum system. The design and implementation phase will be conducted in the second semester. The second semester is also allocated for the testing process upon the Double Inverted Pendulum System in the laboratory.

## CHAPTER 2

### LITERATURE REVIEW AND THEORY

#### 2.1 Literature Review

By referring to a book entitled "*A First Course of Fuzzy and Neural Control*" [4], the author explained an example of PID Controller used for the Inverted Pendulum System. Similar to the Inverted Pendulum available in the laboratory, PID Controller is used for controlling purpose. However, the Inverted Pendulum requires some time to change its position which is from pointing downward to upright when using PID Controller. Therefore, GA will be applied to optimize the PID Controller hence reducing the time for the Inverted Pendulum to stable.

From a journal entitled "*Genetic Algorithms Adaptive Control for under actuated System*" [5], written by Faical Mnif and Jawhar Ghommem, GA had been applied to stabilize a whirling pendulum. Whirling pendulum in this case is a system consisting of Inverted Pendulum connected to an arm rotating within a horizontal plane.

*A journal entitled "Stabilization of Inverted Pendulum by the Genetic Algorithm"* by Safaai Deris, Sigeru Omatu and Katsuhide Kitagawa had also been referred. The journal is basically on the usage of GA to find parameter values for neural network and initial value of PID gain. The main idea of this journal is tune the PID gain by using neural network to stabilize the inverted pendulum. In this case, the initial values of PID gain which are  $K_P$ ,  $K_I$  and  $K_D$  and also the connection weight for the neural network need to be identified. These values are solved using the GA.



Firstly, the GA is applied to attain the initial value of  $K_p$ ,  $K_I$  and  $K_D$ . The GA procedures are applied to obtain the values for PID gains. The PID gains are encoded into 8 bits strings as follows:

$$\begin{array}{ccc} 01010110 & | & 01000011 & | & 11010101 \\ K_p & & K_I & & K_D \end{array}$$

The PID gains are limited within  $[0, 250]$  while the population size is taken as 20. In this paper, the stabilization of the inverted pendulum is considered. Thus, the following fitness function is adopted:

$$F = 1/J^{10000}$$

Where;

$$J = \int_0^{\infty} (200x^3 + \theta^2 + \dot{\theta}^2 + 0.1\ddot{\theta}^2 + 0.005u^2) dt$$

Where  $x$  = cart position and  $\theta$  = pendulum angle

Two point crossover is applied in which two digits in the code string are selected and the bit strings enclosed by the two digits are exchanged between individuals is adopted with the crossover probability,  $P_c$  of 0.7. The mutation probability,  $P_m$  on the other hand is 0.04. By using MATLAB, the final values of the PID gains obtained by GA are  $K_p = 216.80$ ,  $K_I = 42.19$  and  $K_D = 72.66$ .

The initial values of connection weights for the Neural Network are also been determined using GA. The code of the connection weights is made by encoding each connection weight by 8 bits string as below:

$$\begin{array}{ccc} 01001111 & | & \dots & | & 01010011 \\ W_{ji} & & & & W_{jk} \end{array}$$

Where the connection weights  $W_{ji}$  and  $W_{jk}$  are limited to  $[-5, 5]$ . The population is taken as 10. Selection is based on the minimization of the fitness function as below:

$$F = 1/J^{10000}$$

Uniform crossover is applied to find the global minimum. By using uniform crossover, the suitable digits covered by random masks are exchanged between two individuals. The crossover probability,  $P_c$  used is 0.65 while the mutation probability,  $P_m$  is 0.03.

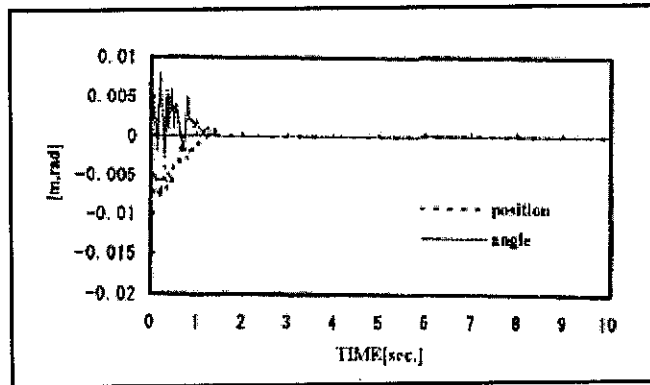


Figure 4 PID control result when the GA is applied to select the PID gains

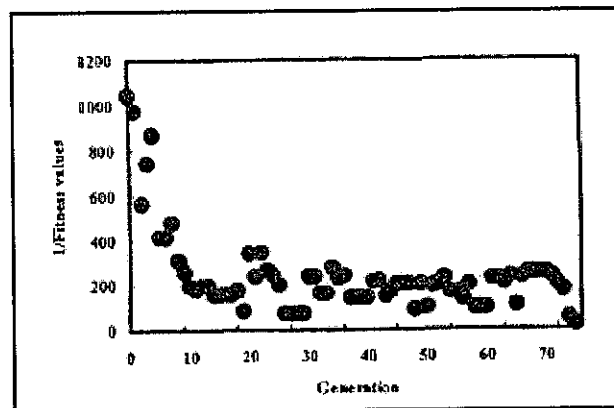


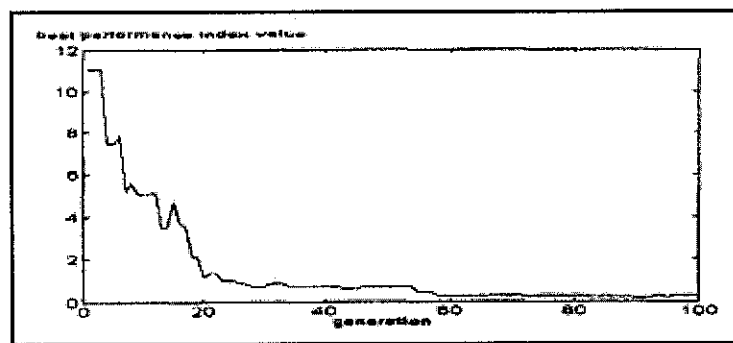
Figure 5 The fitness value for Figure 3

From the journal, it can be concluded that GA had eased the process of finding the initial value of  $K_p$ ,  $K_i$  and  $K_D$  and also the initial values of connection weights for the Neural Network. Besides, GA also provides less time consumption compared to the trial-and-error process.

Another example of journal that applying GA is “*A Genetic Approach to the Attitude Control of an Inverted Pendulum System*” by Kyung-Jae Ha and Hak-Man Kim. In this journal, GA is used for Linear Quadratic Regulator (LQR) problem to find the proper weighting matrices ( $Q$  and  $R$ ) and hence to obtain the optimal control gain ( $K$ ).

In this journal, the population size was chosen to be 100 and the probability crossover,  $P_c$  and probability mutation,  $P_m$  are 0.3 and 0.05 respectively while the sampling period is set to be 0.01second. In order to avoid significant difference between the linearized simulation model and the actual system, noises are added to consider the unstable aspect of the actual system due to modeling error.

The value of  $Q$ ,  $R$  and  $K$  can be determined by using MATLAB software as in figure below. From Figure 5, the best performance index value was found at the 91<sup>st</sup> generation. Thus, the value of  $Q$ ,  $R$  and  $K$  obtained at the 91<sup>st</sup> generation were used for the computer simulation using the inverted pendulum.



**Figure 6** Minimum performance index of each generation

In this paper, a new approach to the optimal LQR design which is the GA is introduced. For the optimal LQR design, one must begin with the choice of proper weighting matrices  $Q$  and  $R$  by trial and error. However, the process of choosing the best matrices can be simplified by using GA.

## 2.2 Theory

The understanding on PID Controller is necessary as the optimization using GA will be conducted upon the PID Controller. Basically, proportional-plus-integral-plus-derivative also commonly known as PID comprises of three mode of controllers which are Proportional Control, Integral Control and Derivative Control. The explanation of each controller is described as follow:

### 2.2.1 Proportional Control

The first element of PID control is Proportional control. However, with proportional control alone, the controller will not bring the process measurement to the setpoint. The transfer function of Proportional Control is as in (2.1).

$$G_C(s) = K_C \quad (2.1)$$

### 2.2.2 Integral Control

The primary advantage of using Integral mode is its ability to achieve zero steady state offset. Steady state offset can be referred as the difference between the final steady state values with the set point. For a good performance, the steady state offset should be as close as possible to the set point value. Normally, the acceptable value of steady state error is  $\pm 5\%$  of the set point. Equation (2.2) shows the transfer function of Integral Control.

$$G_C(s) = K_C \left( 1 + \frac{1}{T_I s} \right) \quad (2.2)$$

### 2.2.3 Derivative Control

The derivative term looks at the rate of change of the input and adjusts the output based on the rate of change. With Derivative Control, fast response can be achieved. The transfer function of Derivative Control can be stated as in equation (2.3).

$$G_C(s) = K_C (1 + T_d s) \quad (2.3)$$

#### 2.2.4 Nyquist Stability Criterion

The Nyquist criterion actually relates the stability of a closed-loop system to the open-loop frequency response and open-loop location. By referring to a book entitled "*Feedback Control of Dynamic System*" written by *Gene F. Franklin, J. David Powell and Abbas Emami-Naeini*, the key concept of the Nyquist stability criterion is that, the net number of clockwise encirclements,  $N$  equals the number of zeros (closed-loop system roots) in the right half plane (RHP),  $Z$  minus the number of open-loop poles in the RHP,  $P$  [11]. This statement can be simplified by the equation 4.1. The system is considered as stable when the value of  $Z$  equals to zero.

$$Z = N + P \quad (2.4)$$

Where;

$Z$  = number of closed-loop zeros in the RHP

$N$  = number of clockwise encirclement of  $-1+j0$  point

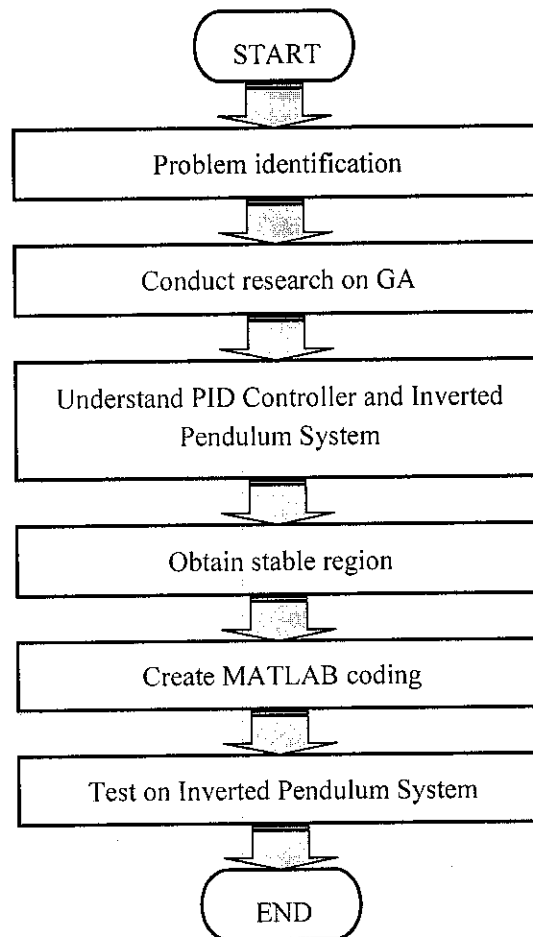
$P$  = number of open-loop poles in the RHP

## CHAPTER 3

### METHODOLOGY

#### 3.1 Procedure Identification

The overall methodology for this project can be deduced into the following flowchart. The details on each procedure are discussed next:



**Figure 7** Overall Project Flowchart

### **3.1.1 Problem Identification**

The first stage of this project is to identify the core problem for this project. The main problem arose in this project is that; the inverted pendulum which applies the PID Controller requires ample time to change its position, from pointing downward to pointing upright and stabilized. Therefore, a mean of searching the desired point by consuming less time need to be developed.

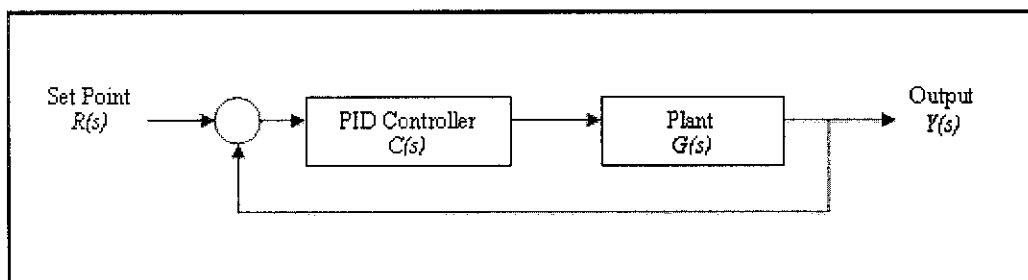
### **3.1.2 Research on GA**

Research on GA need to be conducted thoroughly as the student does not have any background knowledge about GA. Initially, research started with information gathering on basic principle of GA. The steps involve in applying GA and how it is applied in solving problems are also been studied.

### **3.1.3 Understand PID Controller and Inverted Pendulum System**

As the objective of this project is to optimize the PID controller for the inverted pendulum system, hence the student needs to have a strong understanding on the PID Controller and the Inverted Pendulum. The functionality and characteristics of PID Controller and Inverted Pendulum are also need to be understood.

A simple system that utilizes the PID Controller can be illustrated in a block diagram such as in **Figure 8**. The 'Plant' block represents the system that need to be control. In this case, it represents the Inverted Pendulum System.

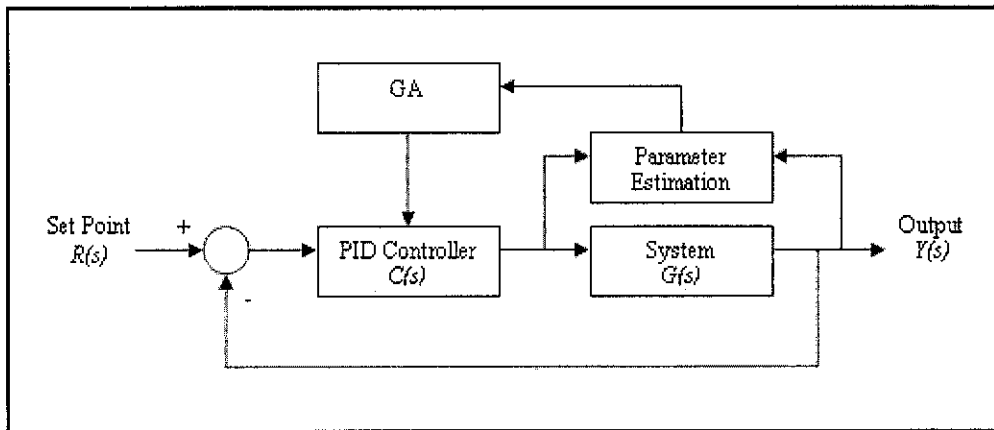


**Figure 8** Block diagram for a system using PID Controller

As stated in the diagram,  $C(s)$  is the PID Controller whose expression is

$$C(s) = K_p + K_D s + \frac{K_I}{s} \quad (3.1)$$

In order to implement GA to optimize the PID Controller, the block diagram as in **Figure 8** needs to be modified. The resulting block diagram after the modification is shown in **Figure 9**.

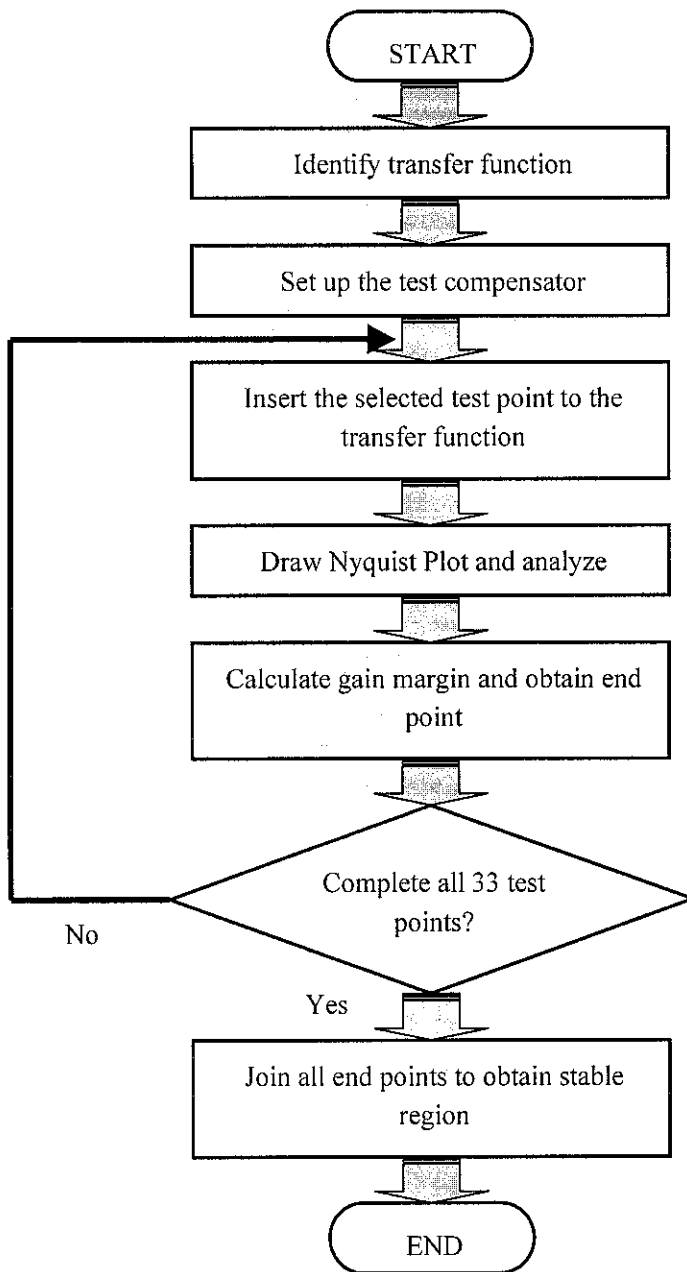


**Figure 9** Inverted Pendulum system using PID Controller and GA

### 3.1.4 Obtain stable region of the system

Before the GA can be applied, the stable region for the Inverted Pendulum system needs to be obtained. From the stable region, the allowed value of  $K_P$ ,  $K_I$  and  $K_D$  that ensure stability of the system can be determined. The Nyquist Stability Criterion is used for the purpose of finding the stable region. Here are the steps taken in obtaining the stable region:





**Figure 10** Flowchart for obtaining stable region

The transfer function of the desired system need to be identified first. Then the Test Compensator comprises of 33 points is set up using MATLAB. 33 points of Test Compensator is not a fixed value. However, the minimum number test points should be is 30 points in order to give a clear visualization of the stable region. The test compensator is represented in  $2D$  where,  $x$ -axis represents the  $K_P$  value while  $y$ -axis shows the  $K_I$  value.

The value of  $K_P$  and  $K_I$  obtained from the Test Compensator is then substituted into the transfer function. In other word, the Test Compensator is cascaded with the system represented by the transfer function. Here, a new transfer function will be obtained. By using the new transfer function, the Nyquist Plot is then been drawn and analyzed.

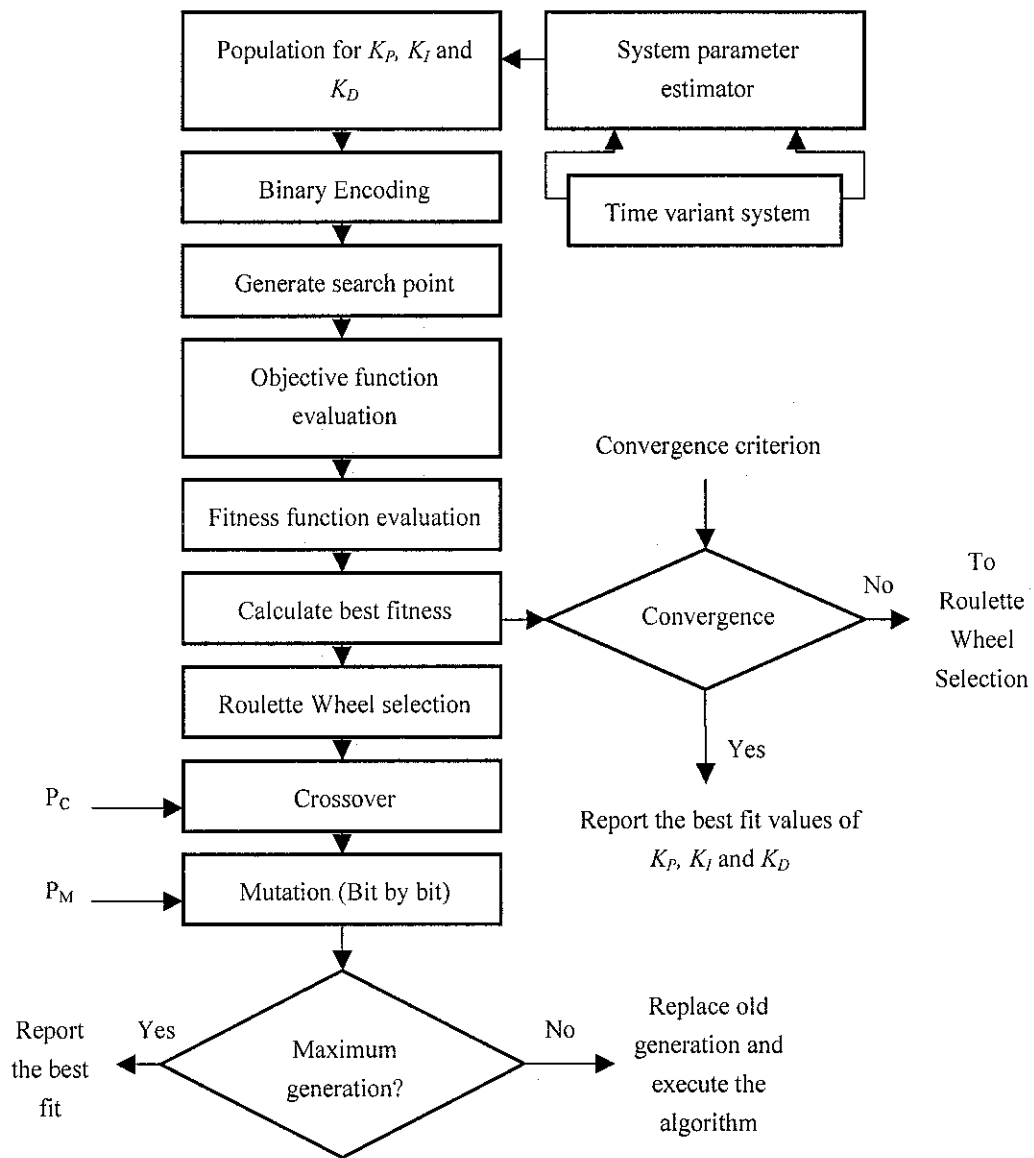
The next step is to calculate the gain margin and determine the end point in the stable region. The same procedures are repeated for other 32 points. At the end, all of the end point will be joined together to form a bounded region, called stable region. By changing the value of  $K_D$ , another layer of Test Compensator will be created. At the end, each layer of stable region with the corresponding value of  $K_D$  will be stacked on top of each other to form the 3D stable region.

### ***3.1.5 Create MATLAB Coding***

GA Coding using MATLAB will be created and the functionality of the coding will be determined by applying it to the Inverted Pendulum System available in the laboratory. The student needs to have deep understanding on the GA functions and tools. The GA programming process is summarized in **Figure 11**.

### ***3.1.6 Test on Inverted Pendulum Trainer***

MATLAB Coding will be applied to the Inverted Pendulum Trainer available in the laboratory to verify the functionality of the coding. The inverted pendulum should be able to stabilize and require less time compared to the conventional PID Controller.



**Figure 11** Flowchart for online GA programming

## **3.2 Tools Required**

This project requires the usage of MATLAB 7.0 Software and the Double Inverted Pendulum System Trainer which is available in the lab. The Double Inverted Pendulum System comprises of two main requirements which are the hardware and software requirements. The details for each category are stated as below:

### ***3.2.1 Hardware Requirements***

- Pentium personal computer
- RTDAC4/PCIc data acquisition board for controlling the system. the board should contain FPGA equipped with dedicated logic design
- Pendulum and cart set-up
- Power interface and wiring allowing electrical connection to the pendulum set

### ***3.2.2 Software Requirements***

- MS Windows 95/98/2000 or Windows XP
- MATLAB version 6.x with Simulink to develop the project
- Real Time Workshop to generate the code
- Real Time Windows Target toolbox
- MS Visual C++ to compile the generated code

## CHAPTER 4

### RESULTS AND DISCUSSION

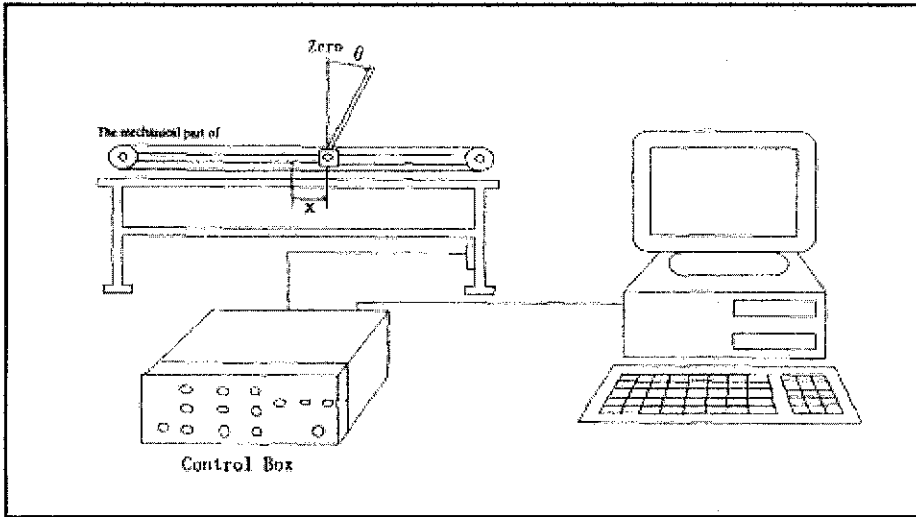
#### 4.1 System Description

The Inverted Pendulum System consists of a pole mounted on a cart in such a way that the pole can swing freely in the vertical plane. In order to swing and to balance the pole, the cart is pushed back and forth on a rail which is driven by a DC motor. Once the pole of the pendulum rod is upwardly erected, the cart modifies its position as needed to balance the rod. However, if disturbance is applied to the pendulum, the cart will be moving back and forth to stabilize the pendulum.

The parameters of the inverted pendulum which are required to be controlled need to be clearly identified. There are four main parameters of the pendulum that need to be taken into account. Those parameters are listed as below:

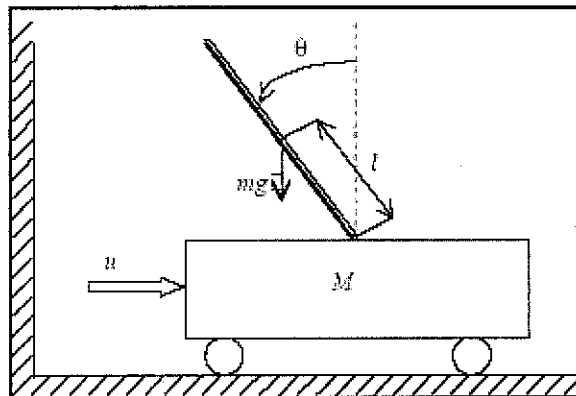
- Cart Position,  $x$
- Cart Velocity,  $\dot{x}$
- Pendulum Angle,  $\theta$
- Pendulum Velocity,  $\dot{\theta}$

**Figure 12** illustrates a typical structure of the Inverted Pendulum System. The pendulum angle,  $\theta$  is measured from a vertical line and the cart position,  $x$  is measured from the center of the cart track.



**Figure 12** Structure of the inverted pendulum system

The desired value of the parameter for the pendulum to be stable is  $\theta = 0$  and  $x = 0$  respectively. Whenever disturbance is applied to the pendulum, the cart will be moving back and forth to stabilize the pendulum. This is to ensure that the cart achieve the position of  $x = 0$  and the pendulum angle of  $\theta = 0$ .



**Figure 13** Cart Pole system

**Figure 13** illustrates a typical cart-pole system. By referring to this figure, input force of  $u$  is applied to the cart of mass  $M$ . The pendulum is free to rotate without friction in the plane. The pendulum is of length  $l$  and the pendulum mass ( $m$ ) is assumed to be concentrated at the end. The equations of motion of this system in terms of the cart displacement and pendulum angle can be written as:

$$(M + m)\ddot{x} - ml\ddot{\theta}\cos\theta + ml\dot{\theta}^2\sin\theta = u \quad (4.1)$$

$$m\ddot{\theta}\cos\theta + ml\dot{\theta}^2\sin\theta = mg\sin\theta \quad (4.2)$$

The Inverted Pendulum System is designed using MATLAB SIMULINK. From the above equations, it can be simplified as the equation below. In the SIMULINK, the system is built using these equations:

$$\ddot{x} = \frac{ml\dot{\theta}^2 \sin \theta - mg \sin \theta \cos \theta + u}{M + m \sin^2 \theta} \quad (4.3)$$

$$\ddot{\theta} = \frac{-(ml\dot{\theta}^2 \sin \theta \cos \theta - mg \sin \theta + u \cos \theta - Mg \sin \theta)}{l(M + m \sin^2 \theta)} \quad (4.4)$$

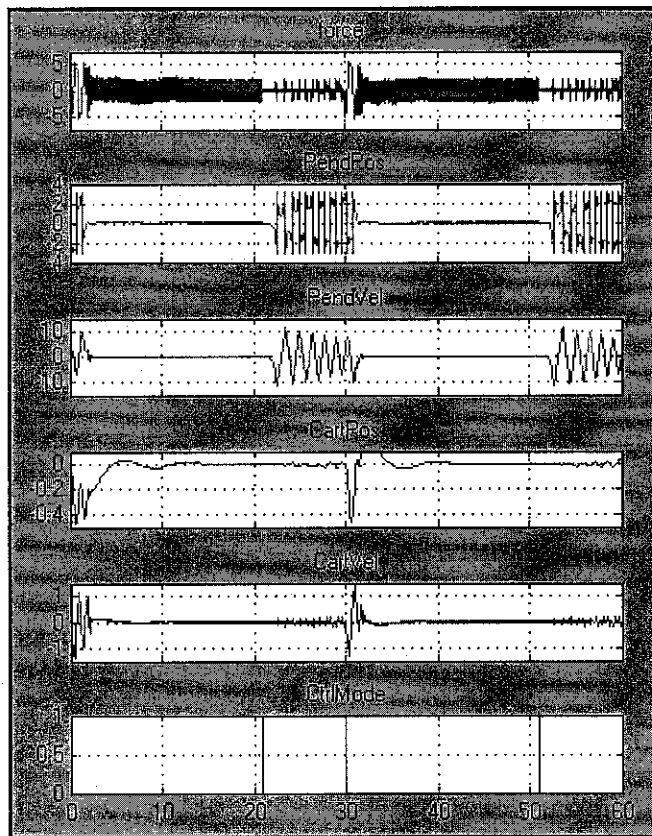
The SIMULINK model of the system is built with  $u$  as the input and  $x$  as the output. The state variables as chosen to be  $x, \dot{x}, \theta, \dot{\theta}$  and these state variables are used in the controller. The parameters for the Inverted Pendulum trainer in the lab are tabulated as in **Table 2**. Those values, such as the  $M, m$  and  $l$  are then substituted into the above equations.

**Table 2** Parameters of Double Inverted Pendulum System

NAME	DESCRIPTION	UNIT
$M$	Equivalent mass of cart and pendulum	0.872kg
$L$	Distance from axis of rotation to centre of mass of system	0.011m
$g$	Gravity	9.81m/s
$p_1$	Control force to PWM signal ratio	9.4N
$p_2$	Control force to cart velocity ratio	-0.548Ns/m
$m_c$	Equivalent cart mass	0.768kg
$m_{ps}$	Pole mass	0.038kg
$m_{pw}$	Load mass	0.014kg
$R_l$	Rail length	1.8m
$l_p$	Length of pole	0.5m
$l_{po}$	Distance between centre of pole mass and rotation axis	0.107m
$l_c$	Length of load	0.0m
$l_{pwo}$	Distance between centre of load mass and rotation axis	0.54m
$T$	Pendulum period	1.17s

## 4.2 Performance of current PID Controller

In order to have a clearer understanding on the Inverted Pendulum System, an experiment had been conducted using the Double Inverted Pendulum Trainer available in the laboratory. The purpose of having this experiment is to understand how the pendulum works and observe the output from the system. Once the process on stabilizing the pendulum is understood, the modification on the existing system can be done. From the experiment conducted, the output can be observed as in **Figure 14** below.



**Figure 14** Output from Inverted Pendulum System

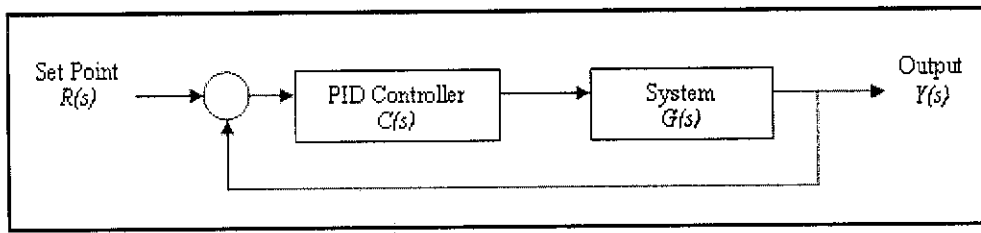
The output of the Inverted Pendulum System comprises of six scopes as in **Figure 14**. Only four main scopes will be considered in this project, which are the pendulum position, pendulum velocity, cart position and cart velocity.



From the scope, we can see that, at certain period of time, the pendulum position and the cart position has the value of 0 respectively. This behavior shows that the Inverted Pendulum System is in stable condition as  $\theta = 0$  and  $x = 0$ . As mentioned before, the pendulum becomes stable if both  $\theta$  and  $x$  have zero values. However, the real Inverted Pendulum System requires some time to be stabilized and it will only be stable for a few seconds.

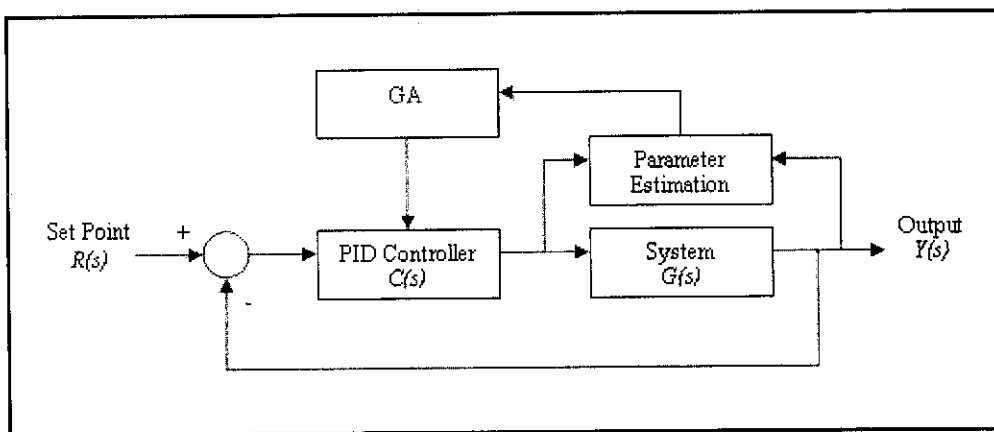
### 4.3 Design Process

Basically, a simple system which utilizes the PID Controller is shown in block diagram such as in **Figure 15**. For this project, the system block diagram represents the Cart Model block diagram, which is the inverted pendulum subsystem. Please refer to **Figure 17** for the detailed block diagram of the Cart Model.



**Figure 15** Typical block diagram for a system using PID Controller

For the optimization purpose, GA block will be placed in the Inverted Pendulum System using PID Controller as in **Figure 15**. The modified Simulink model after considering the GA block is shown in **Figure 16**.



**Figure 16** Inverted Pendulum System using PID Controller and GA

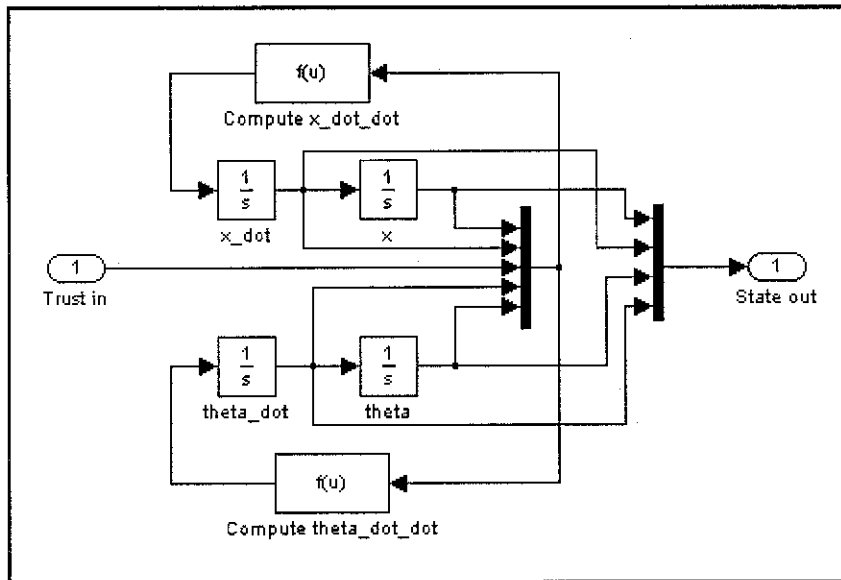


Figure 17 Inverted Pendulum Subsystem

Figure 17 illustrates the Inverted Pendulum Subsystem. This system is a standard Simulink model for any typical Inverted Pendulum System. The User Defined Function,  $f(u)$  for the 'compute  $x\_dot\_dot$ ' block is given by the equation (4.3) as mentioned before while the function for 'compute  $theta\_dot\_dot$ ' block is determined by the equation (4.4).

#### 4.4 Obtaining Stable Region

In order to determine the stable region of a system, Nyquist Stability Criterion can be used. The Nyquist criterion actually relates the stability of a closed-loop system to the open-loop frequency response and open-loop location.

The first step to obtain stable region is by determining the transfer function of the system. The transfer function of the Inverted Pendulum System is:

$$G(s) = \frac{\frac{ml}{q}s}{s^3 + \frac{b(I + ml^2)}{q}s^2 - \frac{(M + m)mgl}{q}s - \frac{bmgl}{q}} \quad (4.5)$$

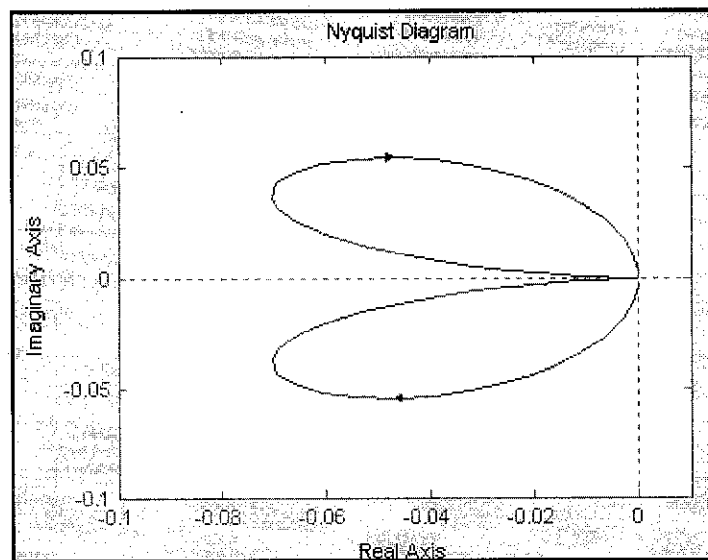
Where;

$M$ = mass of the cart	= 0.768 kg
$m$ = mass of the pendulum	= 0.104 kg
$I$ = inertia of the pendulum	= 0.00282 kgm <sup>2</sup>
$b$ = friction of the cart	= 0.5 Ns/m
$l$ = length to pendulum center of mass	= 0.011 m
$q = [(M + m)(I + ml^2) - (ml)^2]$	= 0.002468

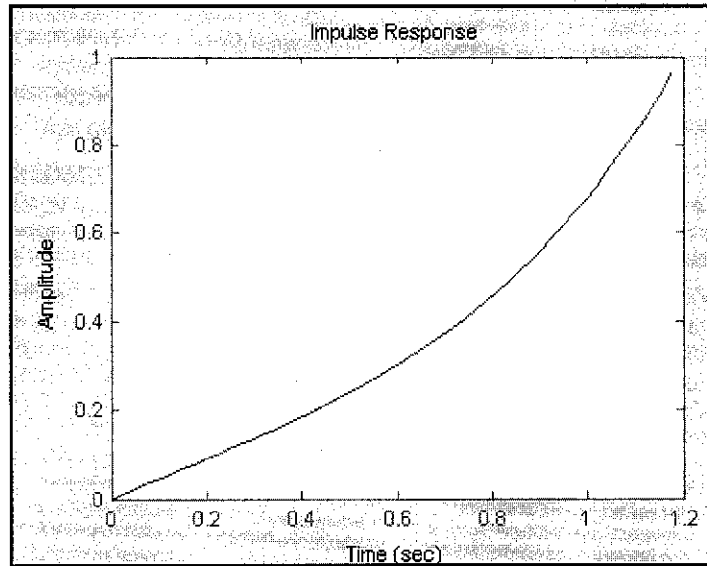
The resulting transfer function after substituting the variable values is as below:

$$G(s) = \frac{0.4635s}{s^3 + 0.5739s^2 - 3.9652s - 2.2736} \quad (4.6)$$

The Nyquist plot for the above transfer function can be drawn using MATLAB. The MATLAB coding to draw the Nyquist plot is attached in the Appendix. The Nyquist plot for system in equation 4.6 is as in **Figure 18**.



**Figure 18** Nyquist Plot for  $G(s) = \frac{0.4635s}{s^3 + 0.5739s^2 - 3.9652s - 2.2736}$



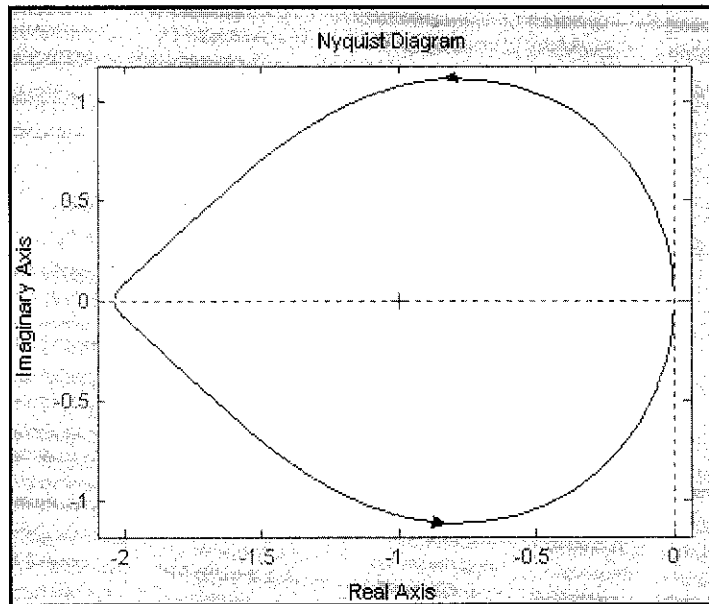
**Figure 19** Impulse Response for  $G(s) = \frac{0.4635s}{s^3 + 0.5739s^2 - 3.9652s - 2.2736}$

By referring to **Figure 18** and **Figure 19**, it can be seen that the system with equation 4.6 is not stable since the transfer function has one positive pole and two negative poles which are 1.9911, -1.9916 and -0.5733 respectively. The first step to make the system stable is by adding an integrator to cancel the extra zero at the origin and it will then have two poles and two zeros at the origin.

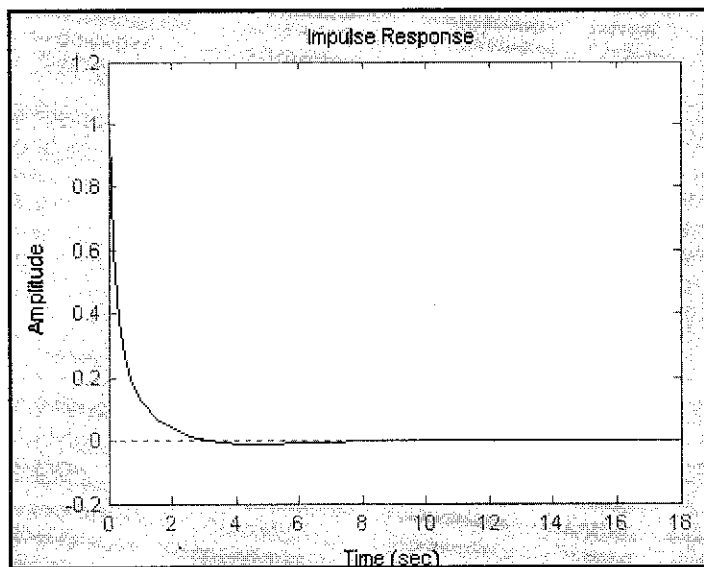
Now, there are two poles in the right-half plane ( $Z = P + N = 1 + 1$ ). In order to get an anti-clockwise encirclement ( $N = -1$ ), a zero is added to the controller. For starter, zero is placed at -1. However, the encirclement around -1 is still clockwise ( $N = +1$ ). Thus, a second zero needs to be added. Yet, the encirclement around -1 is still clockwise. nevertheless, by adding some gain, the system will become stable by shifting the Nyquist plot to the left, moving the anti-clockwise circle around -1, so that  $N = -1$ .

The resulting transfer function after adding an integrator, two poles and gain are as below:

$$G(s) = \frac{10(s+1)(s+1)0.4635s}{s(s^3 + 0.5739s^2 - 3.9652s - 2.2736)} \quad (4.7)$$



**Figure 20** Nyquist Plot for  $G(s) = \frac{10(s+1)(s+1)0.4635s}{s(s^3 + 0.5739s^2 - 3.9652s - 2.2736)}$



**Figure 21** Impulse Response for  $G(s) = \frac{10(s+1)(s+1)0.4635s}{s(s^3 + 0.5739s^2 - 3.9652s - 2.2736)}$

The next step in finding the stable region is by producing a set of test compensator as in **Figure 19**. The test compensator can be obtained using MATLAB. The test compensator is represented in  $2D$  where,  $x$ -axis represents the  $K_P$  value while  $y$ -axis shows the  $K_I$  value. The value of  $K_D$  will be represented in  $z$ -axis. For each layer of test compensator, the value of  $K_D$  will be fixed.

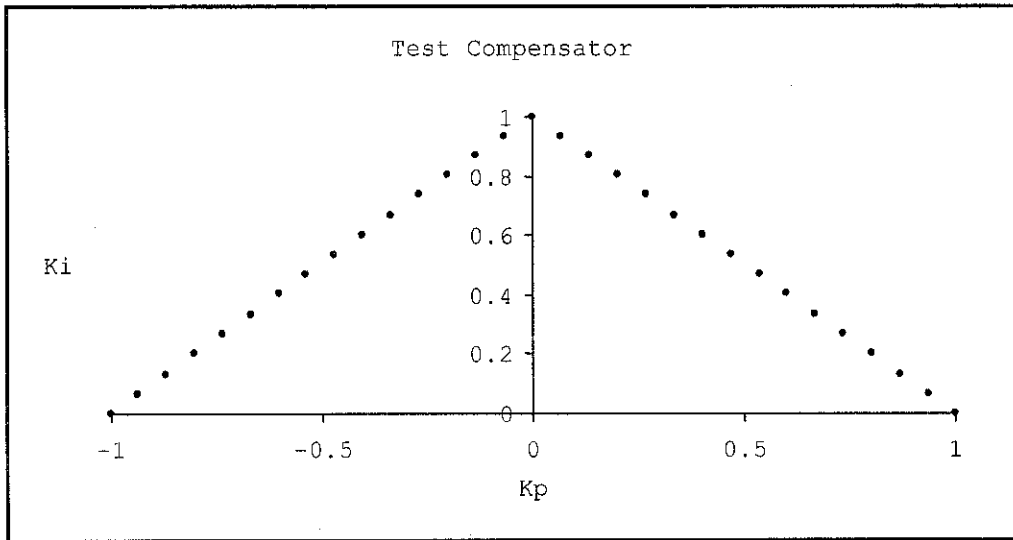


Figure 22 Test Compensator

The value of  $K_p$  and  $K_i$  for Test Compensator,  $C(s) = \frac{K_p s + K_i}{s}$  are to be determined. For explanation purpose, let consider one of the points in test compensator, which is at  $K_p = 0.75$  and  $K_i = 0.25$  that corresponds to  $C(s) = \frac{0.75s + 0.25}{s}$ . This will result the transfer function as in equation 4.8.

$$G(s) = \frac{(0.75s + 0.25)(4.365s^2 + 8.73s + 4.365)}{(s)(s^3 + 0.5739s^2 - 3.9652s - 2.2736)} \quad (4.8)$$

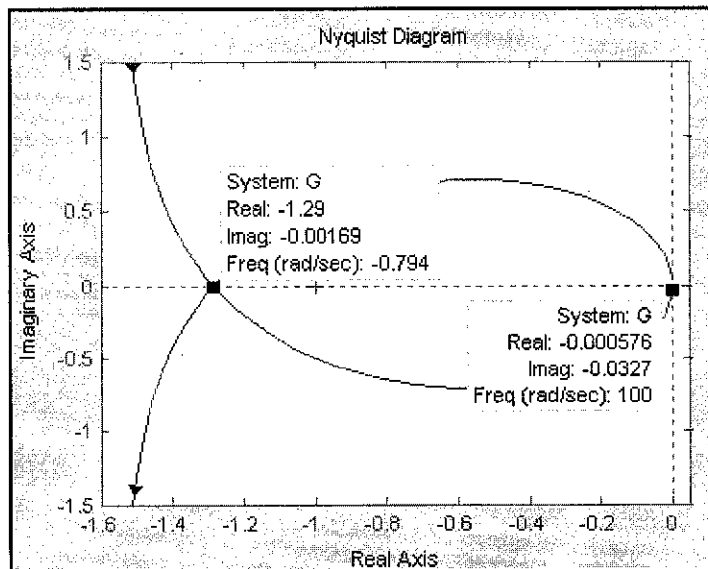


Figure 23 Nyquist Plot for  $G(s) = \frac{(0.75s + 0.25)(4.365s^2 + 8.73s + 4.365)}{(s)(s^3 + 0.5739s^2 - 3.9652s - 2.2736)}$

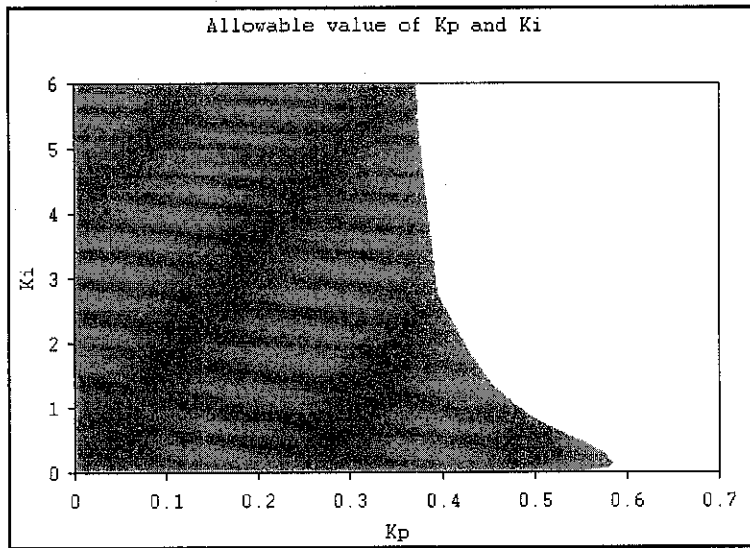
From equation 4.8, the Nyquist Plot can be obtained as in **Figure 23**. The Nyquist Plot can be analyzed by splitting it into three segments which are from  $-\infty$  to  $-1.29$ ,  $-1.29$  to  $0$  and  $0$  to  $\infty$ . From **Figure 23**, the stability of each segment can be determined by obtaining the value of  $Z$ . From Equation 2.4, the value of  $Z$  equals to the summation of  $N$  and  $P$ . In order to achieve stability,  $Z$  must be equal to zero. The analysis of **Figure 23** is summarized as in **Table 3**.

**Table 3** Analysis of Nyquist Plot for  $G(s) = \frac{(0.75s + 0.25)(4.365s^2 + 8.73s + 4.365)}{(s)(s^3 + 0.5739s^2 - 3.9652s - 2.2736)}$

From	To	$P$	$N$	$Z$	Stability	Gains	$x$	$y$	Point 1	Point 2
$-\infty$	$-1.29$	0	0	0	Stable	0 0.7752	0.75	0.25	(0,0)	(0.5814, 0.1938)
$-1.29$	0	0	-2	-2	Not stable	0.7752 $\infty$	0.75	0.25		
0	$\infty$	0	1	-1	Not stable	$\infty$ 0	0.75	0.25		

By referring to **Table 3**, it can be seen that only the first segment ( $-\infty$  to  $-1.29$ ) is considered as stable with the gain of 0 and 0.7752. This segment is stable from (0, 0) to (0.5814, 0.1938), where the  $x$ -coordinate and  $y$ -coordinate is represented by  $K_p$  and  $K_f$  respectively.

As mentioned before, the analysis that had been discussed before is based on one of the points in test compensator, which is  $K_p = 0.75$  and  $K_f = 0.25$ . Since the test compensator comprises of 33 points, analysis need to be done on 32 other points. Towards the end, all of the final point will be joined together to form a bounded region, called stable region as in **Figure 24**.



**Figure 24** Stable region for the Inverted Pendulum System

The shaded region is called the stable region for the Inverted Pendulum System with

the transfer function  $G(s) = \frac{4.365s^2 + 8.73s + 4.365}{s^3 + 0.5739s^2 - 3.9652s - 2.2736}$ . The allowable values

of the  $K_P$  and  $K_I$  are simplified as in **Table 4**.

**Table 4** Allowable values of  $K_P$  and  $K_I$  for the Inverted Pendulum System

TEST COMPENSATOR		GAIN	STABLE REGION	
$K_P$	$K_I$		$K_P$	$K_I$
0	1	2293.578	0	2293.578
0.0625	0.9375	5.952381	0.372024	5.580357
0.125	0.875	3.154574	0.394322	2.760252
0.1875	0.8125	2.283105	0.428082	1.855023
0.25	0.75	1.814882	0.453721	1.361162
0.3125	0.6875	1.526718	0.477099	1.049618
0.375	0.625	1.331558	0.499334	0.832224
0.4375	0.5625	1.189061	0.520214	0.668847
0.5	0.5	1.075269	0.537634	0.537634
0.5625	0.4375	0.990099	0.556931	0.433168
0.625	0.375	0.909091	0.568182	0.340909
0.6875	0.3125	0.840336	0.577731	0.262605
0.75	0.25	0.775194	0.581395	0.193798
0.8125	0.1875	0.719424	0.584532	0.134892
0.875	0.125	0.662252	0.57947	0.082781
0.9375	0.0625	0.60241	0.564759	0.037651
1	0	0.520833	0.520833	0



## 4.5 Discussion

The main purpose of developing this project is to optimize the PID Controller for Inverted Pendulum System. The PID Controller needs to be optimized because of some reasons. The major reason of optimizing the PID Controller is because; ample time is required for the Inverted Pendulum to stabilize. The cart of the Inverted Pendulum is required to move back and forth for quite some time before the stable region is achieved. Therefore, this project is carried out to enhance the performance of the PID Controller and hence, reduces the time taken by the pendulum to stabilize.

There are various types of optimization techniques that can be used to optimize a system. However, in this project, GA is chosen as an optimization technique for optimizing the PID Controller. GA is preferred because of several reasons. One of the reasons that lead to the usage of GA is its parallelism, where it travels in a search space using more individuals so that it is less likely to get stuck in a local extreme like other methods.

The main concern in this project is to determine the stable region of the Inverted Pendulum System. Stability is very crucial in any system design. The concept of stability is so important since every control system must be primarily stable and only then other properties can be studied [13]. An unstable system cannot be designed for a specific transient response or steady state requirement [14] and it has no practical significance.

In this project, the stable region for the Inverted Pendulum System is determined using Nyquist Stability Criterion. Nyquist Criterion actually relates the stability of a closed loop system to the open loop frequency response and open-loop location [14]. Besides Nyquist Stability Criterion, other methods that can be used to determine stability are Routh-Hurwitz Criterion, Root Locus technique and Lyapunov Stability.

The first step that needs to be taken in obtaining stable region for a system is to determine the transfer function of the system. In this project, the transfer function for the Inverted Pendulum System is quite impossible to obtain. According to Mr. A

Turnau, the Chief Executive of INTECO, the inverted pendulum system is a nonlinear system therefore the transfer function does not exist. However, if the system is linearized, the system will have one positive root among four roots.

The process of determining the transfer function of Inverted Pendulum System is very complicated. Thus, the transfer function of a typical Inverted Pendulum System is considered in this project. However, the real values for parameters of the Inverted Pendulum System are used and substituted in the typical transfer function.

The stable region of the Inverted Pendulum System is very important since it has a great relation to GA. The allowable values of  $K_p$  and  $K_i$  in the stable region are essential in order to determine the population, which is the first step in GA implementation. Only then, the process of selection, crossover and mutation will take place. These processes will be repeated until the termination criterion is met.

## CHAPTER 5

### CONCLUSIONS AND RECOMMENDATIONS

#### 5.1 Conclusions

As a conclusion, GA is one of the methods that can be used to optimize the performance of a system [7]. GA is chosen as the optimization technique for PID Controller because of several reasons. In contrast with other optimization techniques, GA operates on entire populations of candidate solutions in parallel. In fact, the parallel nature of a GA's stochastic search is one of the main strengths of the genetic approach. This parallel nature implies that GA is much more likely to locate a global peak because it is much less likely to get stuck at local optima.

In this project, GA is utilized to optimize PID Controller for the Double Inverted Pendulum System. Before the GA can be applied, the stable region of the system needs to be identified first. Nyquist Stability Criterion is one of the convenient ways in obtaining stable region. Once the stable region is obtained in 2-Dimension, GA can be applied to search for the optimum point within the stable region. By obtaining the optimum stable point, the values for  $K_P$ ,  $K_I$  and  $K_D$  for PID Controller can be determined.

The first two objectives of this project had successfully achieved by the student. The student is able to study and understand the behavior of current PID Controller for Inverted Pendulum System. Experiment on the Inverted Pendulum System available in the lab had been conducted and the performance of the PID Controller had been identified.

Thorough research had been conducted on the principle of GA. By performing this project, the student is now able to understand the basic principle of GA and how it is implemented. It is an advantage for the student to have the knowledge on GA since this artificial intelligence has a great prospect in the future.

However, the student is still working on the implementation of GA. Once the optimization technique using GA is successful, the time required by the pendulum to erect and stabilize will be reduced and hence improving the performance of PID Controller.

## 5.2 Recommendations for Future Work

It is recommended that Graphic User Interface (GUI) for the GA to be implemented in this project. The main reason of having GUI is to make GA more user-friendly and convenient to the user.

It is also recommended that the PID Controller together with GA will be tested on the real Inverted Pendulum available in the laboratory. The testing process is to make sure that the control design together with the MATLAB coding is functioning. This is the final step in completing the project. If the designed controller is succeeded on the real Inverted Pendulum, this project can be described as successful.

With the success of this project, the Inverted Pendulum System can be replaced by a more complex system such as a Multi Input Multi Output (MIMO) system. GA can be applied to the system as an optimization technique in order to enhance the performance of the system.

## REFERENCES

- [1] Thomas E. Marlin, "*Process Control Designing Processes and Control Systems for Dynamic Performance*", McGraw-Hill, 2000.
- [2] David E. Goldberg, "*Genetic Algorithm in Search, Optimization and Machine Learning*", Addison-Wesley, 1989.
- [3] K.F. Man, K.S Tang and S. Kwong, "*Genetic Algorithms*".
- [4] Emanuel Falkenauer, "*Genetic Algorithms and Grouping Problems*", John Wiley & Sons.
- [5] Hung T. Nguyen, Nadipuram R. Prasad, Carol L. Walker and Elbert A. Walker, "*A First Course of Fuzzy and Neural Control*", Chapman & Hall.
- [6] Faical Mnif and Jawhar Ghommem, "*Genetic Algorithms Adaptive Control for under actuated System*", Yang's Scientific Research Institute, LLC.
- [7] E.S. Sazonov, P. Klinkhachorn and R. L. Klein, "*Hybrid LQG-Neural Controller for Inverted Pendulum System*".
- [8] Lance Chambers, "*The Practical Handbook of Genetic Algorithms Application*", Chapman & Hall, Pg. 99.
- [9] Safaai Deris, Sigeru Omatu and Katsuhide Kitagawa, "*Stabilization of Inverted Pendulum by the Genetic Algorithm*".
- [10] Kyung-Jae Ha and Hak-Man Kim, "*A Genetic Approach to the Attitude Control of an Inverted Pendulum System*".
- [11] Gene F. Franklin, J. Davis Powell and Abbas Emami-Naeini, "*Feedback Control of Dynamic Systems*", Fourth Edition, Pearson Education International.
- [12] Zazilah May, "*The systematic Design of Simply Structured Compensator*", MSc in Advanced Control thesis, University of Manchester Institute of Science and Technology, United Kingdom, September 2000.
- [13] Zoran Vukic, Ljubomir Kuljaca, Dali Donlagic, and Sejid Tesnjak, "*Nonlinear Control System*", Marcel Dekker Inc.

## **APPENDICES**

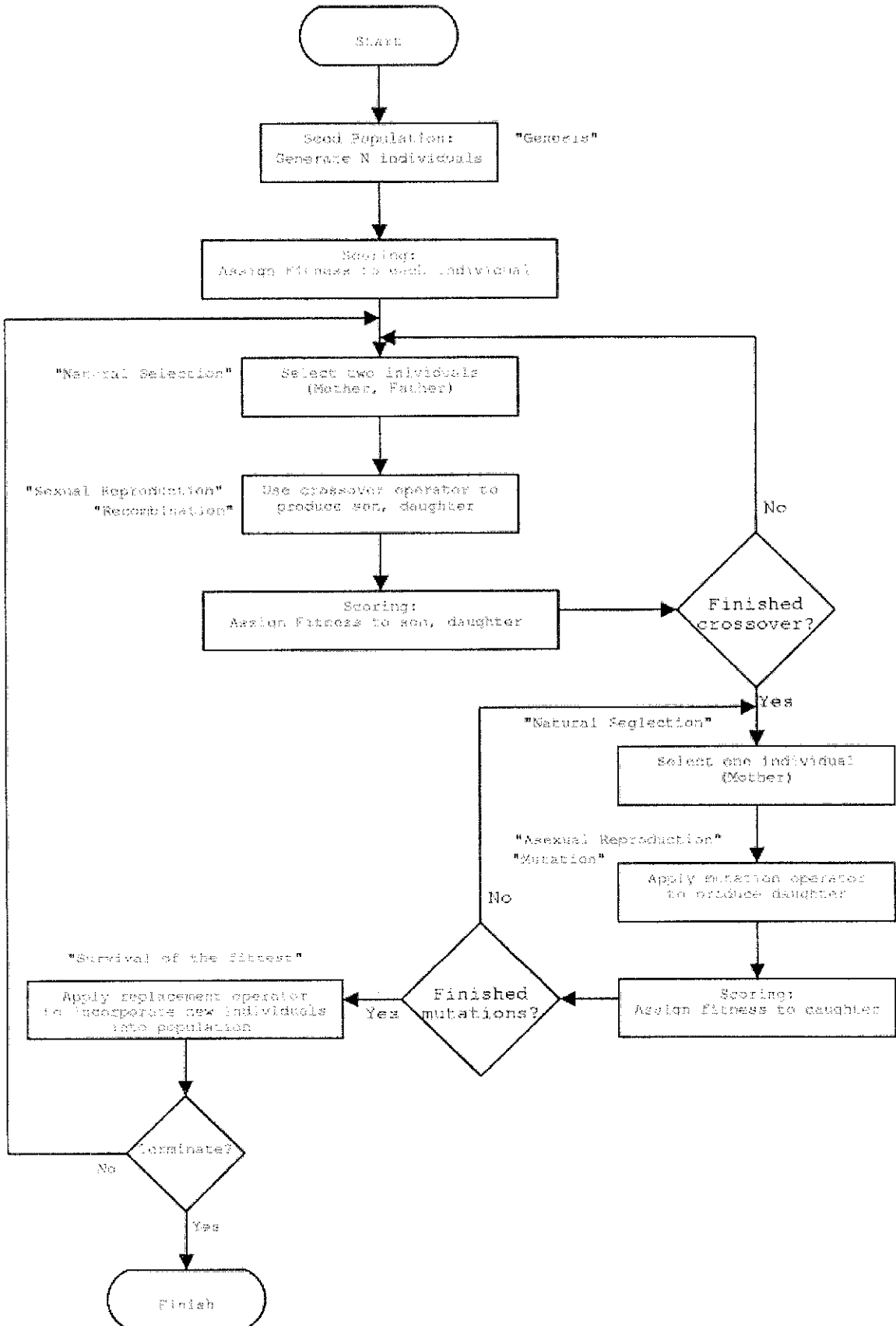
### **APPENDIX A GENETIC ALGORITHM (GA)**

**APPENDIX A.1 STEPS IN GA IMPLEMENTATION**

**APPENDIX A.2 TWO POINT Crossover IN GA**

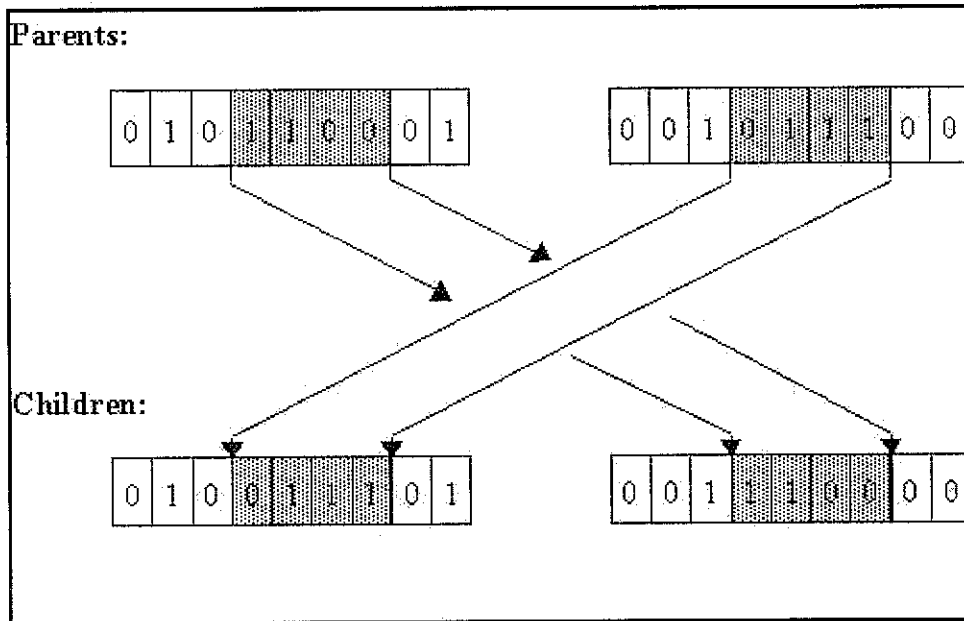
**APPENDIX A.3 MUTATION IN GA**

## APPENDIX A.1 STEPS IN GA IMPLEMENTATION

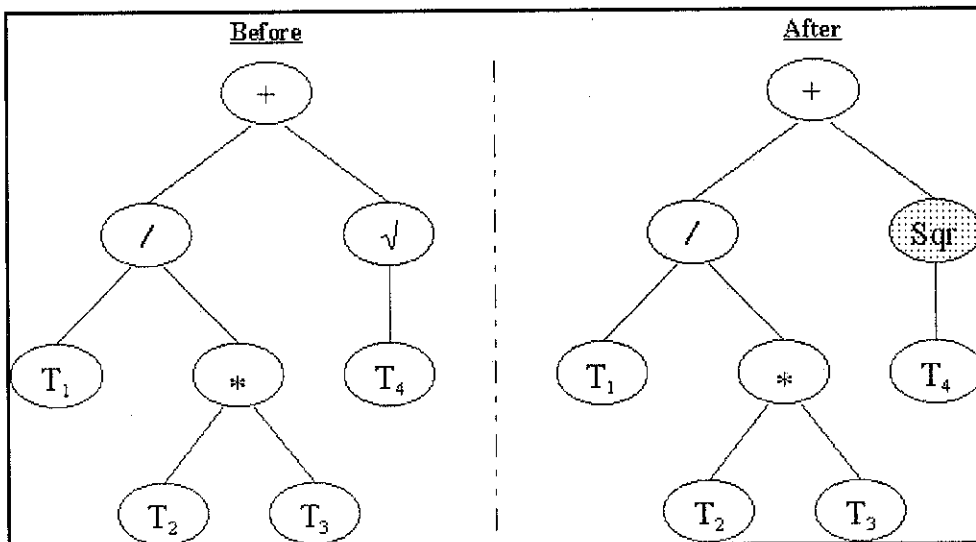




## APPENDIX A.2 TWO POINT CROSSOVER IN GA



## APPENDIX A.3 MUTATION IN GA



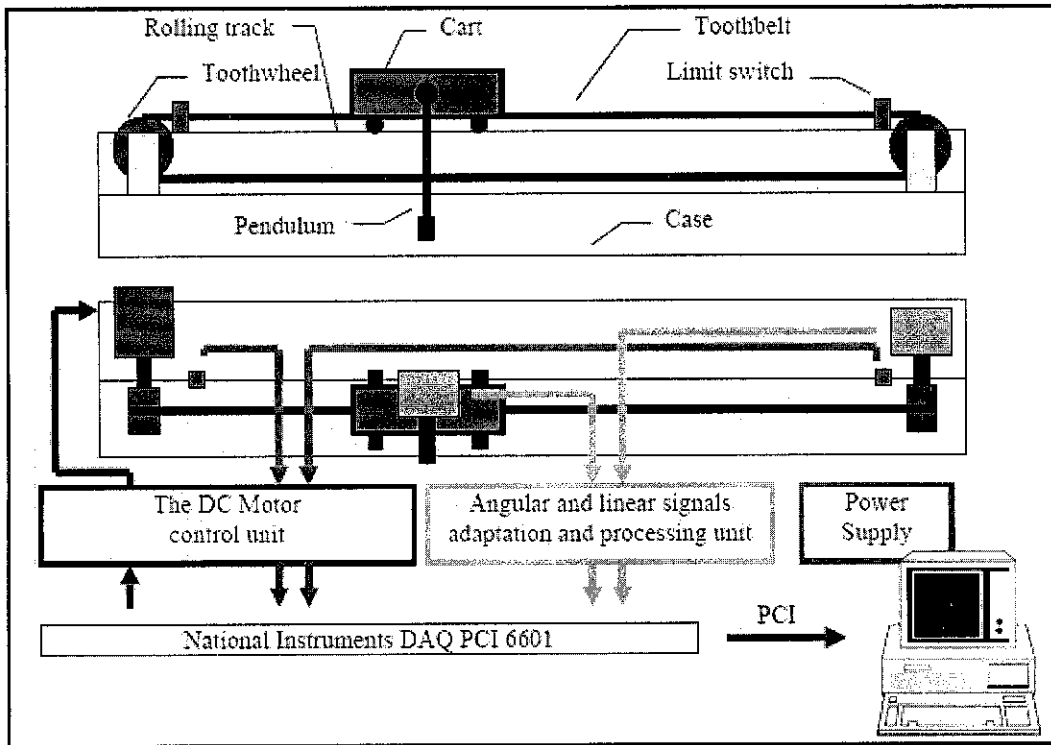
**APPENDIX B**  
**INVERTED PENDULUM SYSTEM**

**APPENDIX B.1 STRUCTURE OF INVERTED PENDULUM  
SYSTEM**

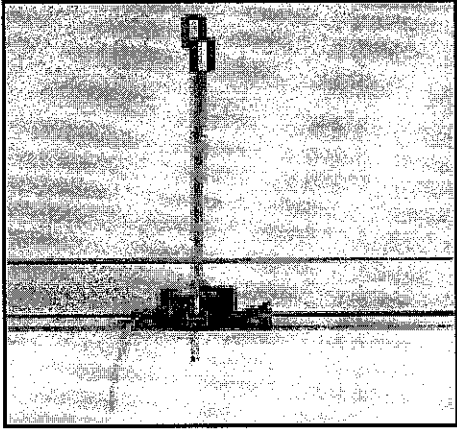
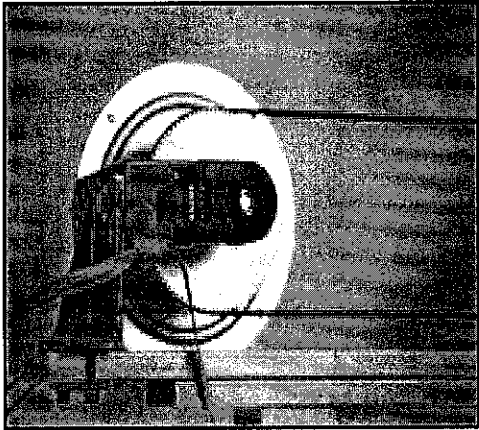
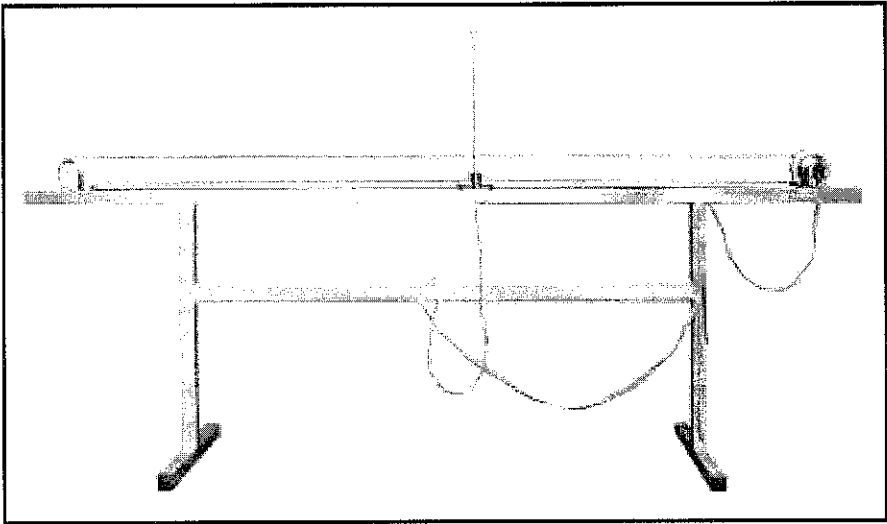
**APPENDIX B.2 DOUBLE INVERTED PENDULUM SYSTEM**

**APPENDIX B.3 PARAMETERS OF DOUBLE INVERTED  
PENDULUM SYSTEM**

# APPENDIX B.1 STRUCTURE OF INVERTED PENDULUM SYSTEM



**APPENDIX B.2 DOUBLE INVERTED PENDULUM SYSTEM**



### APPENDIX B.3 PARAMETERS OF DOUBLE INVERTED PENDULUM SYSTEM

NAME	DESCRIPTION	UNIT
$M$	Equivalent mass of cart and pendulum	0.872kg
$L$	Distance from axis of rotation to centre of mass of system	0.011m
$f_c$	Dynamic cart friction of the cart	0.5Ns/m
$f_s$	Static friction of cart	1.203N
$f_p$	Rotational friction coefficient	$6.65 \times 10^{-5}$ Nms/rad
$J_p$	Moment of inertia of pendulum with respect to axis of rotation	$0.00292 \text{kgm}^2$
$g$	Gravity	9.81m/s
$p_1$	Control force to PWM signal ratio	9.4N
$p_2$	Control force to cart velocity ratio	-0.548Ns/m
$u_{max}$	Maximum value of PWM signal	0.5
$m_c$	Equivalent cart mass	0.768kg
$m_{ps}$	Pole mass	0.038kg
$m_{pw}$	Load mass	0.014kg
$R_l$	Rail length	1.8m
$l_p$	Length of pole	0.5m
$l_{po}$	Distance between centre of pole mass and rotation axis	0.107m
$l_c$	Length of load	0.0.m
$l_{pwo}$	Distance between centre of load mass and rotation axis	0.54m
$T$	Pendulum period	1.17s
$J$	Moment of inertia related to the mass centre	$0.00282 \text{kgm}^2$

## **APPENDIX C**

### **MATLAB CODING**

- APPENDIX C.1 CODING FOR OBTAINING TEST  
COMPENSATOR**
- APPENDIX C.2 CODING FOR NYQUIST PLOT**
- APPENDIX C.3 CODING FOR GA**

## APPENDIX C.1 M FILE FOR OBTAINING TEST COMPENSATOR

```
N = 30;
z = N/2;
x = linspace(-1,1,N);
xx = fliplr(x);
y = linspace(0,50,N);
y1 = -50*xx(1:z)+50;
y2 = 50*xx(z+1:N)+50;
plot([xx(1:z);xx(1:z)],[zeros(1,z);
(-50*xx(1:z)+50)],'kx:',[xx(z+1:N);xx(z+1:N)],[zeros(1,z);(50*xx(z+1:N)+50)],'kx:');
xlabel('Kp');
ylabel('Ki');
title('Test Compensator');
```

## APPENDIX C.2 M FILE FOR NYQUIST PLOT

```
numg = conv([0.75 0.25], [0.4635 0]);
deng = conv([1 0], [1 0.5739 -3.9652 -2.2736]);
G = tf(numg,deng)
Nyquist(G);
```

## APPENDIX C.3 M FILE FOR GA

*% Decimal to Binary Algorithm*

```
function b2 = b10to2(b10,bits)
%converts base 10 base 2
%Originally from Mathworks Inc.

bit_count = 0;
b2_index = [];
a = length(bits)
bits_index = 1:length(bits);
for i=bits_index,
    bit_count = bit_count + bits(i);
    b2_index = [b2_index bit_count];
end

for i=1:max(bits),
    r = rem(b10,2)
    b2(:,b2_index) = r;

    b10 = fix(b10/2);
    tbe = find (all(b10==0) | (bits(bits_index==i)));
    if ~isempty(tbe),
        b10(:,tbe) = [];
        b2_index(tbe) = [];
        bits_index(tbe) = [];
    end
end
```

```

    if isempty(bits_index),
        return
    end

    b2_index = b2_index-1;

end

% Decoding Algorithm

function [x,coarse] = decode(gen,vlb,vub,bits)
%converts from binary to variable representation

bit_count = 0;
two_pow = 2.^(0:max(bits))
for i=1:length(bits),
    pow_mat((1:bits(i)) +bit_count,i) = two_pow(bits(i):-1:1);
    bit_count = bit_count+bits(i);
end

gen_row = size(gen,1);
coarse = (vub-vlb)/(2.^bits-1);
inc = ones(gen_row,1)*coarse;
x1 = ones(gen_row,1)*vlb + (gen&pow_mat).&inc;
x = round(x1);

% Encoding Algorithm

%This function is to encode a non-binary parameters to the binary
%parameters.
%This code is taken from Mathwork Inc.

function [gen, lchrom, coarse, nround] = encode (x, vlb, vub, bits)

bits = 4;
vlb = 10;
vub = 1;
x = 7;

lchrom = sum(bits);
coarse = (vub-vlb)/(2.^bits-1);
[x_row,x_col] = size(x);
gen = [];
if ~isempty(x),
    temp =(x-ones(x_row,1)*vlb./...
        (ones(x_row,1)*coarse));
    b10 = round(temp);
    nround = find(b10-temp>1e-4);
    gen = b10to2(b10,bits);
end

%Mutation Algorithm

function [new_gen,mutated] = mutate(old_gen,Pm)
%Mutation Operators
%This code was originated by Mathwork Inc.

mutated = find(rand(size(old_gen))<Pm);
new_gen = old_gen;
new_gen(mutated) = 1 - old_gen(mutated);

```



```

% Crossover Algorithm

function [new_gen,sites] = xover(old_gen,Pc)
%Crossover function
lchrom = size(old_gen,2);
sites = ceil(rand(size(old_gen,1)/2,1)*lchrom-1);
for l = 1:length(sites),
    new_gen([2*i-1 2*i],:) = [old_gen([2*i-1 2*i],1:sites(i))...
        old_gen([2*i2*i-1], sites(i)+1:lchrom)];
end

%Reproduction Algorithm

function [new_gen, selected] = reproduc(old_gen,fitness)
%Reproduction Operators
%This code is originated from Mathwork Inc.

norm_fit = fitness/sum(fitness);
selected = rand(size(fitness));
sum_fit = 0;
for i=1:length(fitness);
    sum_fit = sum_fit + norm_fit(i);
    index = find(selected<sum_fit);
    slected(index) = i*ones(size(index));
end
new_gen = old_gen(selected, :);

%Plot Algorithm
function plotfunc=plot(z)
end

```

**APPENDIX D**  
**GANTT CHART OF COMPLETION**



Milestone for the Second Semester of 2 Semester Final Year Project

No.	Detail/ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	19	21
1	Design MATLAB coding for GA		█	█	█	█	█	█	█	█							
2	Research on Nyquist Stability Criterion		█														
3	Submission of Progress Report 1			●													
4	Design test compensator				█												
5	Determine transfer function					█	█	█	█	█							
5	Obtain Stable Region					█	█	█	█	█							
6	Submission of Progress Report 2								●								
7	Optimization using GA									█	█	█	█				
8	Submission of Dissertation Final Draft													●			
9	Oral Presentation															●	
10	Submission of Project Dissertation																●