DYNAMIC MODELLING AND CONTROL OF MASS SPRING SYSTEM WITH LARGE LOAD UNCERTAINTY

AIZUDDIN ABDUL RAHIM

UNIVERSITI MALAYSIA PAHANG

DYNAMIC MODELLING AND CONTROL OF MASS SPRING SYSTEM WITH LARGE LOAD UNCERTAINTY

AIZUDDIN ABDUL RAHIM

This thesis is submitted as partial fulfillment of the requirement for the award of the degree of Bachelor of Electrical Engineering (Control and Instrumentation)

Faculty of Electrical & Electronic Engineering
University Malaysia Pahang

MAY, 2008

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **DYNAMIC MODELLING AND CONTROL OF MASS SPRING SYSTEM WITH LARGE LOAD UNCERTAINTY**

SESI PENGAJIAN: 2007/2008

Saya **AIZUDDIN A. RAHIM (850106-10-5195)**

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

□ **SULIT** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

□ **TERHAD** (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

☑ **TIDAK TERHAD**

Disahkan oleh:

(TANDATANGAN PENULIS)	(TANDATANGAN PENYELIA)

Alamat Tetap:

NO 18, JALAN 18/34 D,
40200 SHAH ALAM,
SELANGOR.

MISS NAJIDAH BINTI HAMBALI
(Nama Penyelia)

Tarikh: **6 MAY 2008**

Tarikh: : **6 MAY 2008**

CATATAN:

* Potong yang tidak berkenaan.

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.

♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the degree of Bachelor of Electrical Engineering (Control & Instrumentation)"

Signature : ________________________

Name : NAJIDAH BINTI HAMBALI

Date : 6 MAY 2008

To my beloved father and mother

En. Abdul Rahim bin Mohamed

And

Pn. Rokiah binti Ishak

"All the trademark and copyright use here in are property of their respective owner. Reference of information from other sources is quoted accordingly; otherwise the information presented in this report is solely work of the author."

Signature : ____________________

Author : AIZUDDIN A. RAHIM

Date : 8 MAY 2008

ACKNOWLEDGEMENT

In the name of Allah S.W.T. the Most Gracious, the Most Merciful. Praise is to Allah, Lord of the Universe and Peace and Prayers be upon His final Prophet and Messenger, Muhammad S.A.W.

I would like to take this opportunity to sincerely express my highest gratitude to my supervisor, Miss Najidah Hambali for her guidance, ideas, advices and times during this project fulfillment. Without her, this project could not be done successfully.

My sincere gratitude also goes to my fiend for their ideas, advices and aids, who are involves directly and indirectly in this project. My lecturers who are understand and encouraged me to finish up this project.

Last but not least, my special thanks to my beloved parents, Mr. Abdul Rahim Mohamed and Madam Rokiah Ishak; my lovely Aishah Hijriah and my beloved brothers and sister who had given me moral support and always pray for my future undertakings.

ABSTRACT

In this study, we introduce MATLAB software package for modelling, simulating and analyzing dynamic systems. The purpose of this project is to construct modelling of a mass spring system with a linear control design which is the Proportional Integral Derivatives (PID). To evaluate the performance of this system, PID is chosen as a control strategy and will be compared with the uncontrolled by performing a MATLAB Simulink® simulation. This illustrate the use of Simulink® which concern of modelling and simulating of engineering systems. This system can be divided into two sections which are to obtain the equivalent transfer function of the model and to obtain the control of the model’s output. The purpose of the controller is to control the output so it will be in specific condition that it’s required. All of this system is implemented in MATLAB Simulink®.

ABSTRAK

Dalam tesis ini, perisisan MATLAB digunakan dalam menganalisa dan simulasi dalam sesuatu sistem. Objektif utama projek ini adalah untuk membentuk satu model sistem spring menggunakan Proportional Integral Derivative (PID). Dalam menilai sistem ini, sistem yang mempunyai kawalan akan di bandingkan bersama sistem yg tidak di kawal. Ini akan turut memaparkan penggunaan Simulink® sebagai medium bagi membentuk model dan simulasi. Sistem ini boleh dibahagikan kepada dua bahagian, dimana bahagian pertama ialah untuk membentuk persamaan bagi model itu dan mengawal keluaran model sistem tersebut. Tujuan utama sistem kawalan ini adalah untuk mengawal keluaran supaya berada pada keadaan yg diinginkan dan stabil. Keseluruhan sistem ini di proses didalam MATLAB simulink®.

.

TABLE OF CONTENTS

CHAPTER	CONTENTS	PAGE
	TITLE	
	DEDICATION	i
	DECLARATION	ii
	ACKNOWLEDGEMENT	iii
	ABSTRACT	iv
	ABSTRAK	v
	TABLE OF CONTENTS	vi
	LIST OF FIGURE	ix
	LIST OF TABLE	x
	LIST OF ABBREVIATIONS	xi
	LIST SYMBOLS	xii
	LIST OF APPENDIX	xiii
CHAPTER 1	**INTRODUCTION**	1
	1.1 Project Overview	1
	1.2 Objective	2
	1.3 Project Scope	3
	1.4 Thesis Organization	3

CHAPTER 2 LITERATURE REVIEW 4

2.1 Introduction 4
2.2 Mass Spring System 4
2.3 Proportional Integral Derivative 5
2.4 Development Simulink® Using MATLAB 10
2.4.1 Concept of Signal and Logic Flow 10
2.4.2 Sources and Sinks 11
2.4.3 Continuous and Discrete System 11

CHAPTER 3 METHODOLOGY 13

3.1 Introduction 13
3.2 Methodology 13
3.3 MATLAB Modeling Design 16
3.4 Proportional Integral Derivative Design 18
3.5 Tuning Proportional Integral Derivative 21
3.5.1 Manual Tuning 21

CHAPTER 4 RESULT AND DISCUSSION 22
4.1 Introduction 22
4.2 Mass Spring System 22
4.3 Proportional Integral Controller 23
4.3.1 Proportional control 24
4.3.2 Proportional-Derivative control 26
4.3.3 Proportional-Integral control 27
4.3.4 Proportional-Integral-Derivative control 29

4.4 Simulation 30

4.4.1 Model with Controller 30

4.4.2 Model without Controller 31

4.5 Comparison 32

4.6 Conclusion 33

CHAPTER 5 CONCLUSION 34

5.1 Conclusion 34

5.2 Future Recommendation 35

REFERENCE 36

APPENDIX A 37

APPENDIX B 39

LIST OF FIGURE

FIGURE	TITLE	PAGE
2.1	PID controller diagram	7
2.2	Car Suspension Diagram	9
3.1	Flow Chart of whole project	15
3.2	Mass Spring (suspension) system	16
3.3	Block Diagram of Mass Spring System	17
3.4	PID controller	18
4.1	Full Model of Mass spring system	23
4.2	Proportional Integral Derivative controller	24
4.3	Setting the proportional control	25
4.4	Graph of the proportional controller	25
4.5	Setting the proportional-Derivative control	26
4.6	Graph of the proportional- Derivative controller	27
4.7	Setting the proportional-Integral control	28
4.8	Graph of the proportional- Integral controller	28
4.9	Setting the P-I-D control	29
4.10	Graph of the P-I-D controller	30
4.11	Graph of the output without controller	31
4.12	Graph of the output with controller	31
4.13	Comparison	32
4.14	Graph of the output with coulomb friction	33

LIST OF TABLE

TABLE	TITLE	PAGE
2.1	Principle of Road Behavior	12
3.1	Guideline for The Project	14
3.2	Parameter value for mass spring system	16
3.3	Effect of the controller to the system	21

LIST OF ABBREVIATION

FLC	-	Fuzzy Logic Control
LQR	-	Linear quadratic Regulator
PISMC	-	Proportional Integral Sliding Mode Control

LIST OF SYMBOLS

M	-	Mass
K	-	Spring Constant
C	-	Damping Ratio
P_{out}	-	Proportional output
K_p	-	Proportional Gain, a tuning parameter
E	-	Error = $SP - PV$
T	-	Time or instantaneous time (the present)
I_{out}	-	Integral output
K_i:	-	Integral Gain, a tuning parameter
K_D	-	Derivative Gain, a tuning parameter

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Block Diagram of an Uncontrolled Mass Spring System	37
B	Proportional Integral Derivative Design	39

CHAPTER 1

INTRODUCTION

1.1 Overview

The active suspension system control has been one of the most popular subjects in the automotive research area in order improving the ride comfort and handling the performance [1].The suspension system should isolate the body from road disturbance and inertial disturbance associated with the cornering and braking or acceleration [2].

A suspension system can be classified as passive, active and semi-active system. The passive system is widely used which consists of spring and dampers (shock absorbers) A semi-active system is similar to the passive system with the exception that uses variable dampers. It can perform as good ride quality as active suspension system but it lacks the ability to control the car body motion during maneuvering [3]. Meanwhile, the active suspension uses actuators that create the desired force n the suspension system to reduce the sprung mass acceleration and providing sufficient suspension deflection to maintain tire-ground contact [4, 5]

The automotive active suspension control has been one of the greatest interests, both academically and in the automobile industry itself. Various control laws such as adaptive control, back steeping method, optimal state-feedback fuzzy control and sliding mode control have been proposed in the past years to control the active suspension system [6]. The control design of an active suspension system aims to maximize ride comfort (as measured by load of passenger) and under packaging constraints (as measured by suspension travel) [7]. The ride comfort is measured by the vertical acceleration of body because the passenger as a disturbance experiences the acceleration force.

As a mean of generating an active force to enhance the performance of the active suspension system, the optimal controller such as Proportional Integrated Derivation have been considered. Optimal control has been used in active suspension system since the 1960s [8] and still be the famous study among the researchers.

The purpose of this subject is to develop and apply the Proportional Integrated Derivative in active suspension system. Besides that the suspension travel and load of the passengers, will be considered.

1.2 Objective

The objectives of this project are as follows:

i. To develop a dynamic modelling and controller for a mass spring system such as a suspension
ii. To evaluate the performance of mass spring system in terms of large load.

1.3 Scopes of the project

The scopes of this project are as follows:

i. To develop the transfer function for a mass spring dynamic model system.
ii. To develop a controller to control the output of the system.

1.4 Thesis Organization

This thesis consists of five chapters covering introduction, system modeling, controller design, simulation and the last chapter is a conclusion and recommendation future work.

Chapter 1 presents the introduction of the active suspension system using Proportional Integrated Derivative (PID). This chapter also gives an overview of the project including the objectives and scopes of the project.

Chapter 2 gives a detail discussion on the model for overall systems. This includes the literature review for this project.

Chapter 3 discusses the detail of the design aspect for Proportional Integrated Derivative (PID) for an active suspension system. The mathematical modeling of the mass spring system is derived in this chapter. Meanwhile, Chapter 4 represents the computer simulation for the proposed controller. In this chapter, the performance ride comfort will be compared to the uncontrolled active suspension system.

Lastly, Chapter 5 presents the overall conclusion for this thesis and a few suggestion and recommendation for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter contains all the data that gets from any sources that helps to develop this project. The data are helping to familiarize with the mass spring (suspension) system and Proportional Integral Derivative (PID) which also will be useful for additional information in working and further study situation. Besides that, it is also been used as a reference to design the math modeling system.

2.2 Mass spring system

An ideal mass-spring-damper system with mass m (in kilograms), spring constant k (in Newton per meter) and viscous damper of damping coefficient c (in Newton-seconds per meter) can be described with the following formula:

$$F_s = -kx$$

Treating the mass as a free body and applying Newton's second law, we have:

$$\sum F = ma = m\ddot{x} = m\frac{d^2x}{dt^2}$$

Treating the mass as a free body and applying Newton's second law, we have:

$$\sum F = ma = m\ddot{x} = m\frac{d^2x}{dt^2}$$

2.3 Proportional Integral Derivative controller (PID controller)

A **proportional–integral–derivative controller (PID controller)** is a generic control loop feedback mechanism widely used in industrial control systems. 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.

The PID controller calculation (algorithm) involves three separate parameters; the Proportional, the Integral and Derivative values. The Proportional value determines the reaction to the current error, the Integral determines the reaction based on the sum of recent errors and the Derivative determines the reaction to the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve or the power supply of a heating element.

By "tuning" the three constants in the PID controller algorithm the PID can provide control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the set point and the degree of system oscillation. Note that the use of the PID algorithm for control does not guarantee optimal control of the system or system stability.

Some applications may require using only one or two modes to provide the appropriate system control. This is achieved by setting the gain of undesired control outputs to zero. A PID controller will be called a PI, PD, P or I controller in the absence of the respective control actions. PI controllers are particularly common, since derivative action is very sensitive to measurement noise, and the absence of an integral value may prevent the system from reaching its target value due to the control action.

Tutorials about PID are often very technical with a lot of mathematics that leave many people unable to comprehend. Our goal is try to explain PID controllers so that people can easily understand the theory behind them. On this first page we have to start out with terminology and some technical information so that you understand the basics of PID control. On the next page we will discuss PID in a more practical manner.

PID controllers are process controllers with the following characteristics: Continuous process control Analog input (also known as "measurement" or "Process Variable" or "PV") Analog output (referred to simply as "output") Set point (SP) Proportional (P), Integral (I), and / or Derivative (D) constants

How a PID Controller Works

The PID controller's job is to maintain the output at a level so that there is no difference (error) between the process variable (PV) and the set point (SP).

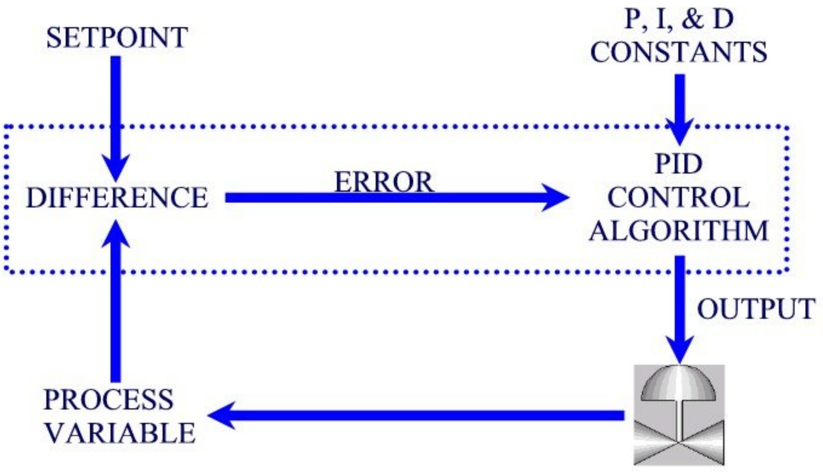


Figure 2.1: PID controller diagram

In this diagram the valve could be controlling the gas going to a heater, the chilling of a cooler, the pressure in a pipe, the flow through a pipe, the level in a tank, or any other process control system. What the PID controller is looking at is the difference (or "error") between the PV and the SP. It looks at the absolute error and the rate of change of error. Absolute error means -- is there a big difference in the PV and SP or a little difference? Rate of change of error means -- is the difference between the PV and SP getting smaller or larger as time goes on. When there is a "process upset", meaning, when the process variable OR the set point quickly changes -- the PID controller has to quickly change the output to get the process variable back equal to the set point. If you have a walk-in cooler with a PID controller and someone opens the door and walks in, the temperature (process variable) could rise very quickly. Therefore the PID controller has to increase the cooling (output) to compensate for this rise in temperature. Once the PID controller has the process variable equal to the set point, a good PID controller will not vary the output. You want the output to be very steady (not changing). If the valve (motor, or other control element) are constantly changing, instead of maintaining a constant value, this could case more wear on the control element. So there are these two contradictory goals. Fast response (fast change in output) when there is a "process upset", but slow response (steady output) when the PV is close to the set point we'll explore how car suspensions work, how they've evolved over the years and where the design of suspensions is headed in the future.

If a road were perfectly flat, with no irregularities, suspensions wouldn't be necessary. But roads are far from flat. Even freshly paved highways have subtle imperfections that can interact with the wheels of a car. It's these imperfections that apply forces to the wheels. According to Newton's laws of motion, all forces have both **magnitude** and **direction**. A bump in the road causes the wheel to move up and down perpendicular to the road surface. The magnitude, of course, depends on whether the wheel is striking a giant bump or a tiny speck. Either way, the car wheel experiences a **vertical acceleration** as it passes over an imperfection.

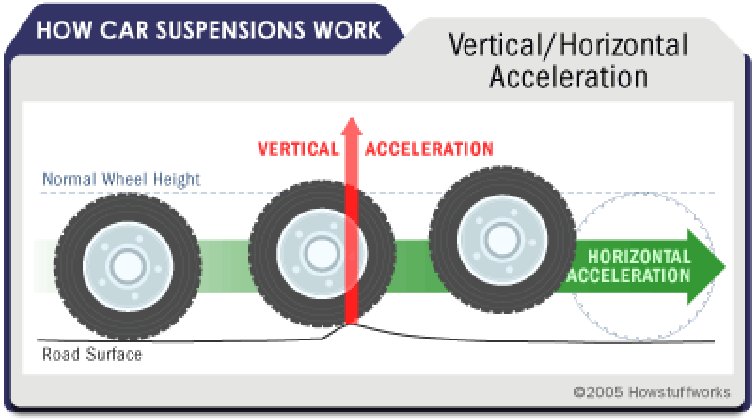


Figure 2.2: Car suspension diagram

Without an intervening structure, all of wheel's vertical energy is transferred to the frame, which moves in the same direction. In such a situation, the wheels can lose contact with the road completely. Then, under the downward force of gravity, the wheels can slam back into the road surface. What you need is a system that will absorb the energy of the vertically accelerated wheel, allowing the frame and body to ride undisturbed while the wheels follow bumps in the road.

The study of the forces at work on a moving car is called **vehicle dynamics**, and you need to understand some of these concepts in order to appreciate why a suspension is necessary in the first place. Most automobile engineers consider the dynamics of a moving car from two perspectives:

i. **Ride** - a car's ability to smooth out a bumpy road
ii. **Handling** - a car's ability to safely accelerate, brake and corner

These two characteristics can be further described in three important principles - **road isolation**, **road holding** and **cornering**. The table below describes these principles and how engineers attempt to solve the challenges unique to each.

2.4 Development Simulink® Using MATLAB

In Simulink, data/information from various blocks are sent to another block by lines connecting the relevant blocks. Signals can be ***generated*** and fed into blocks (dynamic / static). Data can be fed into functions. Data can then be dumped into ***sinks,*** which could be scopes, displays or could be saved to a file. Data can be connected from one block to another, can be branched, multiplexed etc. In simulation, data is processed and transferred only at ***Discrete*** times, since all computers are discrete systems. Thus, a SIMULATION time step (otherwise called an INTEGRATION time step) is essential, and the selection of that step is determined by the fastest dynamics in the simulated system. In the following sections, the different blocks that are available are explained. Figure 3.2 shows the overview of the Simulink libraries available. More toolboxes may be available based on what has been purchased

MATLAB is powered by extensive numerical analysis capability. Simulink® is a tool used to visually program a dynamic system (those governed by Differential equations) and look at results. Any logic circuit, or a control system for dynamic system can be built by using standard BUILDING BLOCKS available in Simulink Libraries. Various toolboxes for different techniques, such as Fuzzy Logic, Neural Network, DSP, Statistic etc. are available with Simulink, which enhance the processing power of the tool. The main advantage is the availability of templates/building blocks, which avoid the necessity of typing code for small mathematical processes

2.4.1 Concept of signal and logic flow

In Simulink, data/information from various blocks are sent to another block by lines connecting the relevant blocks. Signals can be ***generated*** and fed into blocks (dynamic / static). Data can be fed into functions. Data can then be dumped into

sinks, which could be scopes, displays or could be saved to a file. Data can be connected from one block to another, can be branched, multiplexed etc. In simulation, data is processed and transferred only at ***Discrete*** times, since all computers are discrete systems. Thus, a SIMULATION time step (otherwise called an INTEGRATION time step) is essential, and the selection of that step is determined by the fastest dynamics in the simulated system. In the following sections, the different blocks that are available are explained.

2.4.2 Sources and Sinks

The ***sources*** library contains the sources of data/signals that one would use in a dynamic system simulation. One may want to use a *constant* input, a *sinusoidal* wave, a step, a repeating sequence such as a *pulse train*, a *ramp* etc. One may want to test *disturbance* effects, and can use the random signal generator to simulate *noise*. The *clock* may be used to create a time index for plotting purposes. The *ground* could be used to connect to any unused port, to avoid warning messages indicating unconnected ports.

2.4.3 Continuous and Discrete System

All dynamic systems can be analyzed as continuous or discrete time systems. Simulink allows you to represent these systems using transfer functions, integration blocks, delay blocks etc

Discrete systems could be designed in the Z-plane, representing difference equations. Systems could be represented in State-space forms, which are useful in Modern Control System design.

Principle	Definition	Goal	Solution
Road Isolation	The vehicle's ability to absorb or isolate road shock from the passenger compartment	Allow the vehicle body to ride undisturbed while traveling over rough roads.	Absorb energy from road bumps and dissipate it without causing undue oscillation in the vehicle.
Road Holding	The degree to which a car maintains contact with the road surface in various types of directional changes and in a straight line (Example: The weight of a car will shift from the rear tires to the front tires during braking. Because the nose of the car dips toward the road, this type of motion is known as "dive." The opposite effect -- "squat" -- occurs during acceleration, which shifts the weight of the car from the front tires to the back.)	Keep the tires in contact with the ground, because it is the friction between the tires and the road that affects a vehicle's ability to steer, brake and accelerate.	Minimize the transfer of vehicle weight from side to side and front to back, as this transfer of weight reduces the tire's grip on the road.
Cornering	The ability of a vehicle to travel a curved path	Minimize body roll, which occurs as centrifugal force pushes outward on a car's center of gravity while cornering, raising one side of the vehicle and lowering the opposite side.	Transfer the weight of the car during cornering from the high side of the vehicle to the low side.

Table 2.1: Principles of road behavior

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter presents the methodology of this project. It describes on how the project is organized and to make sure that the development of the project is smooth. A good methodologies can described the structure or the flow of the project where by it can be the guideline in managing it. It also to avoid the project to alter course from the objectives that can have been stated or in the other words the project follow the guideline based on the objectives.

3.2 Methodology

There are three mains method in order to develop this project. Before the project is developing using MATLAB, it is needed to do the study on MATLAB. Table 3.1 illustrated the phase of guideline for this project.

Table 3.1: Guideline for the project

Phase I	Project preview and literature review i. Surfing internet ii. Books and writing materials iii. Discussion with lecturer
Phase II	Mathematical model
Phase III	MATLAB simulation
Phase IV	Analysis and Result i. Data collection ii. Data analysis iii. Data comparison

Figure 3.1 below show the flow of the whole project. The project begins after registering the PSM title with doing case study about the project. The flow of the project is to design a MATLAB modeling. Then the next task is to develop controller in MATLAB to control the model. After both part is done, the next step is doing the analysis until achieve the objective.

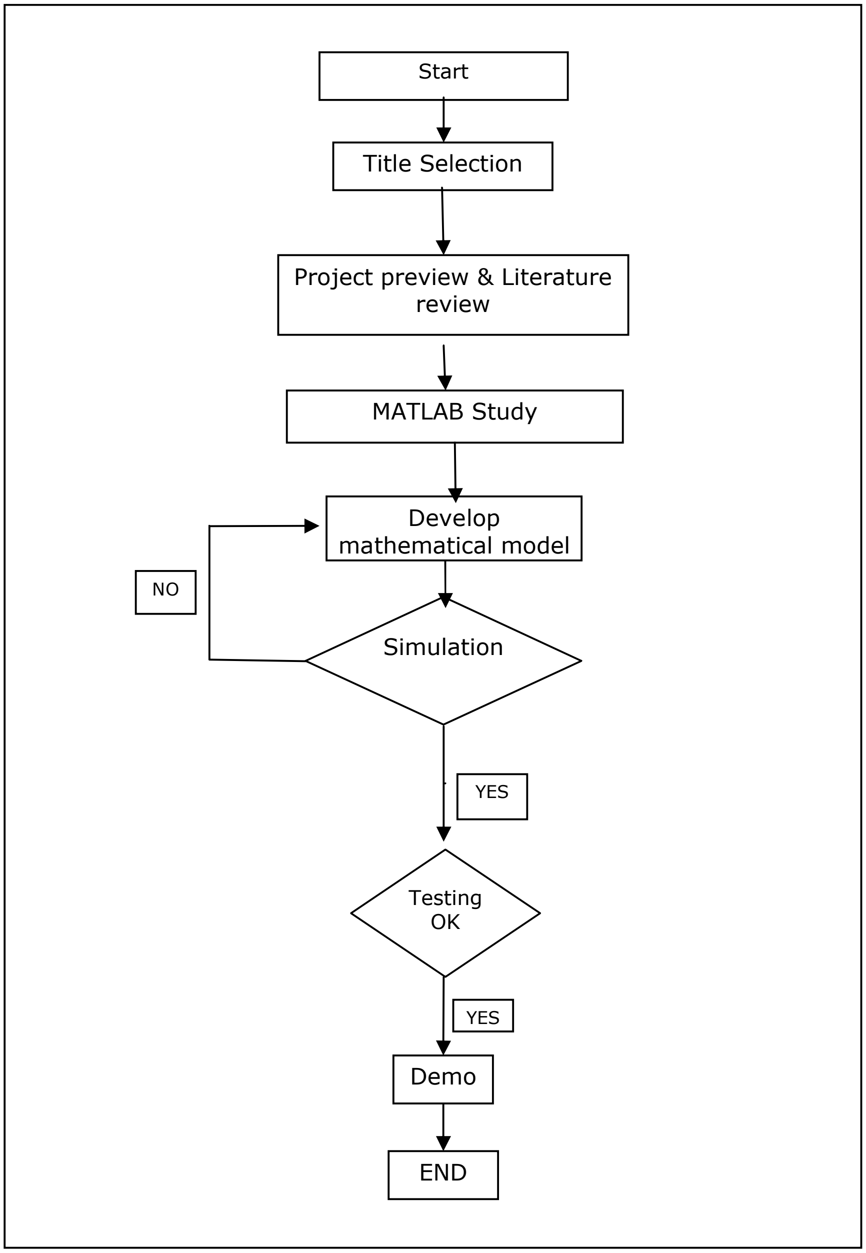


Figure 3.1: Flowchart for Whole Project

3.3 Build MATLAB Modeling

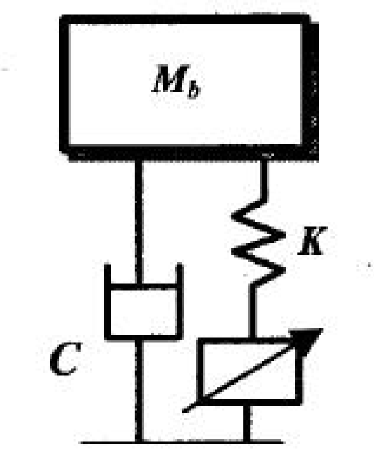


Figure 3.2: The mass spring (suspension) system

Table 3.2: Parameter value for the mass spring (suspension) system.

Mass for the car body, *Mb*	100 kg
Stiffness of the car body spring, *K*	36 N/m
Damping of the damper, *c*	2 Ns/m

Using Newton's second law of motion, the linear differential equations describing the dynamics of the semi-active suspension can be written as:

$$M\square + C\square + Kx = F \qquad (3.1)$$

Where;

M = Mass
K = Spring Constant
C = Damping Ratio

Laplace transform of the modeling equation

$$M s^2 + Cs + k = F(s) \qquad (3.2)$$

The transfer function of modeling (1)

$$\frac{X(s)}{F(s)} = Ms^2 + Cs + k \qquad (3.3)$$

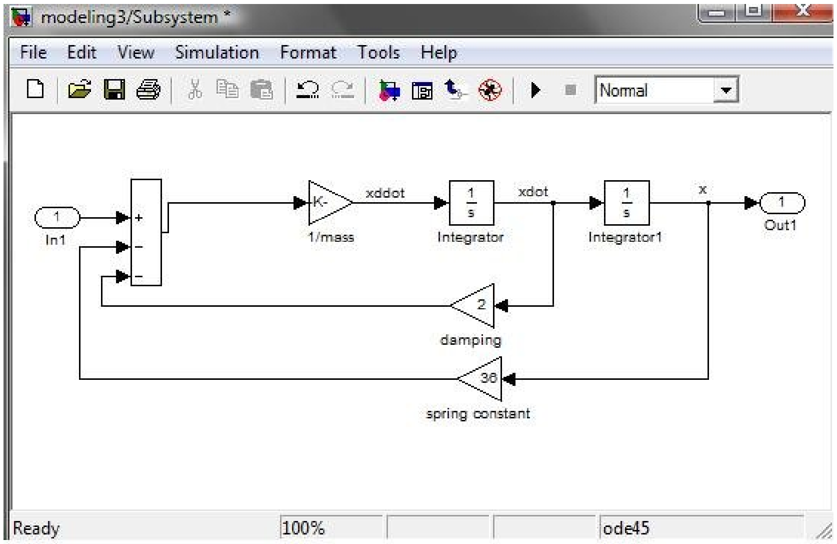
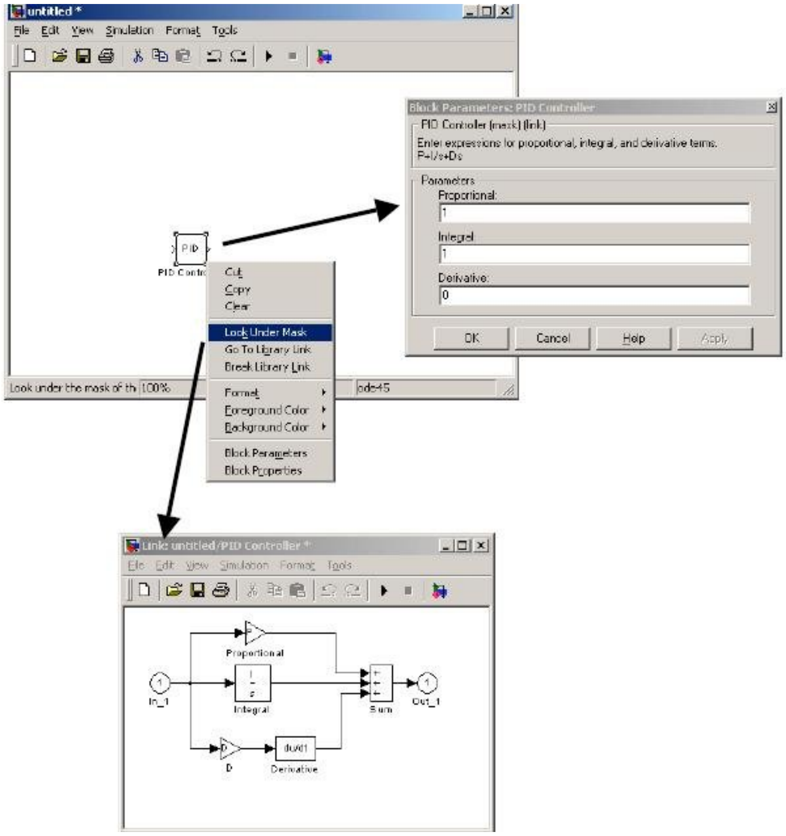


Figure 3.3: The block diagram of Mass Spring System**.**

3.4 Build Proportional Integral Derivative (PID) Controller

A **proportional–integral–derivative controller (PID controller)** is a generic control loop feedback mechanism widely used in industrial control systems. A PID controller attempts to correct the error between a measured process variable and a desired setpoint by calculating and then outputting a corrective action that can adjust the process accordingly.The PID controller calculation (algorithm) involves three separate parameters; the **Proportional**, the **Integral** and **Derivative** values.



Block Parameters: PID Controller

PID Controller (mask) (link)

Enter expressions for proportional, integral, and derivative terms.
P+I/s+Ds

Parameters

Proportional: 1

Integral: 1

Derivative: 0

Figure 3.4: Proportional Integral Derivative Controller

The Proportional value determines the reaction to the current error, the proportional term is given by:

$$P_{\mathrm{out}} = K_p\, e(t) \qquad (3.4)$$

Where;

- P_{out}: **Proportional output**
- K_p: **Proportional Gain**, a tuning parameter
- e: **Error** = $SP - PV$
- t: **Time** or instantaneous time (the present)

The Integral determines the reaction based on the sum of recent errors and the integral term is given by:

$$I_{\mathrm{out}} = K_i \int_0^t e(\tau)\, d\tau \qquad (3.5)$$

Where;

- I_{out}: **Integral output**
- K_i: **Integral Gain**, a tuning parameter
- e: **Error** = $SP - PV$
- τ: **Time** in the past contributing to the integral response

The Derivative determines the reaction to the rate at which the error has been changing whereby the derivative term is given by:

$$D_{\text{out}} = K_d \frac{de}{dt} \qquad (3.6)$$

Where;

- D_{out}: **Derivative output**
- K_d: **Derivative Gain**, a tuning parameter
- e: **Error** = $SP - PV$
- t: **Time** or instantaneous time (the present)

The output from the three terms, the proportional, the integral and the derivative terms are summed to calculate the output of the PID controller. Defining $u(t)$ as the controller output, the final form of the PID algorithm is:

$$\mathrm{u(t)} = \mathrm{MV(t)} = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de}{dt} \qquad (3.7)$$

3.5 Tuning Proportional Integral Derivative (PID) Controller

3.5.1 Manual Tuning

The last part of this chapter would be the tuning part, where the controller is been done with a manual tuning method. The tuning method is to first set the I and D values to zero. Increase the P until the output of the loop oscillates, and then the P should be left set to be approximately half of that value for a "quarter amplitude decay" type response. Then increase D until any offset is correct in sufficient time for the process. However, too much D will cause instability. Finally, increase I, if required, until the loop is acceptably quick to reach its reference after a load disturbance. However, too much I will cause excessive response and overshoot. A fast PID loop tuning usually overshoots slightly to reach the setpoint more quickly; however, some systems cannot accept overshoot, in which case an "over-damped" closed-loop system is required, which will require a P setting significantly less than half that of the P setting causing oscillation.Table 3.3 shows the effect of increasing the parameters.

Table 3.3: The effect of the controller to the system.

CL RESPONSE	RISE TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Small Change	Decrease	Decrease	Small Change

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

This chapter consists of the discussions on the results from the Simulink that has been developing using MATLAB. In this project can be divided to four parts which consist of full modeling system, controller and simulation.

4.2 Mass spring system

The main modeling of this project contains of mass spring system and Proportional Integral Derivative controller is shown in Figure 4.1.

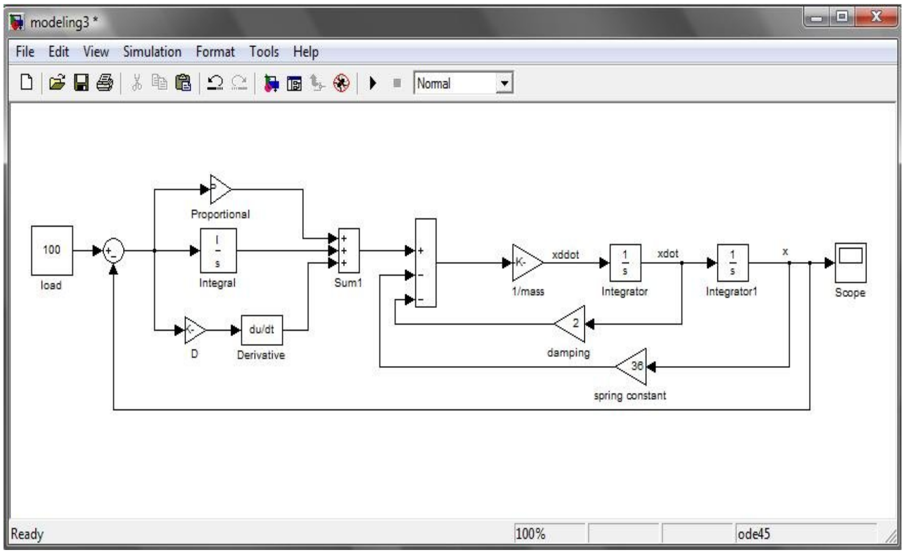


Figure 4.1: Full model of mass spring system and PID controller

The main modeling system is a combined of mass spring system and PID controller. As u can see above from the load, the system had to go through the PID controller first before entering the mass spring system. Here where feedback happens due to stabilizing the system.

4.3 Proportional Integral Controller

From the equation that had been derived before, the proportional integral derivative can now be set. With a single output and a single output, the controller will now set to control the system which will overcome the feedback.

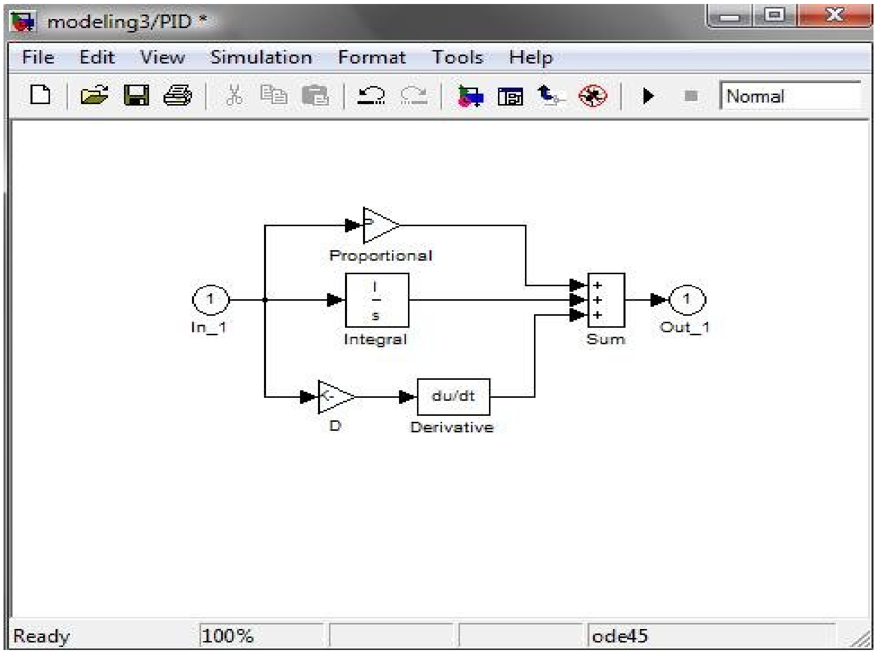
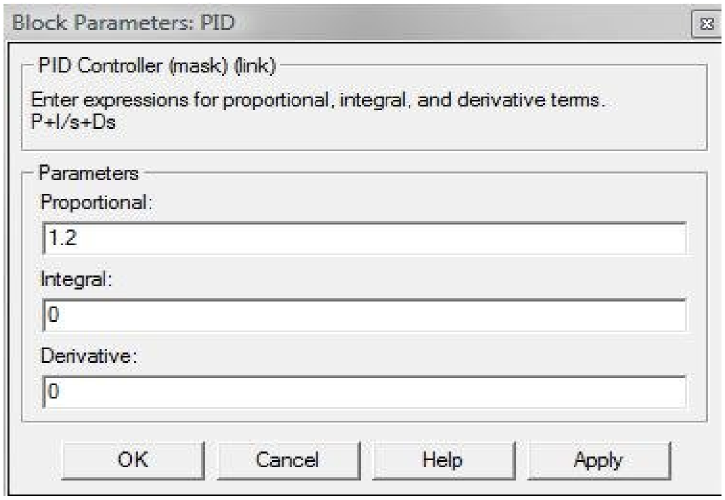


Figure 4.2: Proportional Integral Derivative (PID) controller diagram

4.3.1 Proportional control

From the table 3.3, we see that the proportional controller (Kp) reduces the rise time, increases the overshoot, and reduces the steady-state error. The closed-loop transfer function of the above system with a proportional controller is:

$$\frac{X(s)}{F(s)} = \frac{Kp}{4s^2 + 2s + (36 + Kp)}$$



Block Parameters: PID

PID Controller (mask) (link)

Enter expressions for proportional, integral, and derivative terms.
P+I/s+Ds

Parameters

Proportional: 1.2

Integral: 0

Derivative: 0

Figure 4.3: Setting the Proportional control

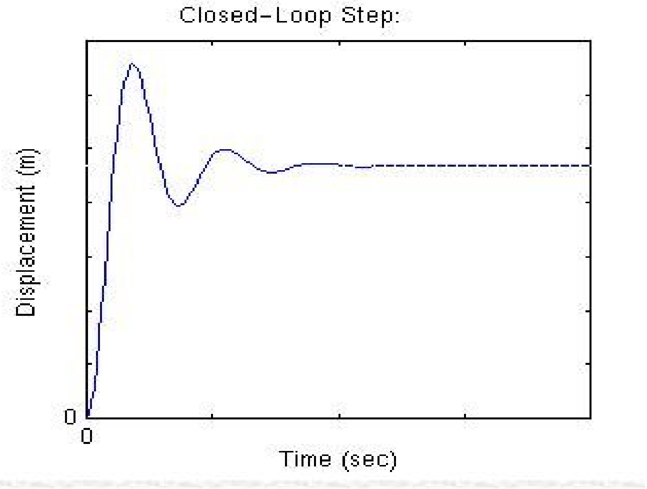
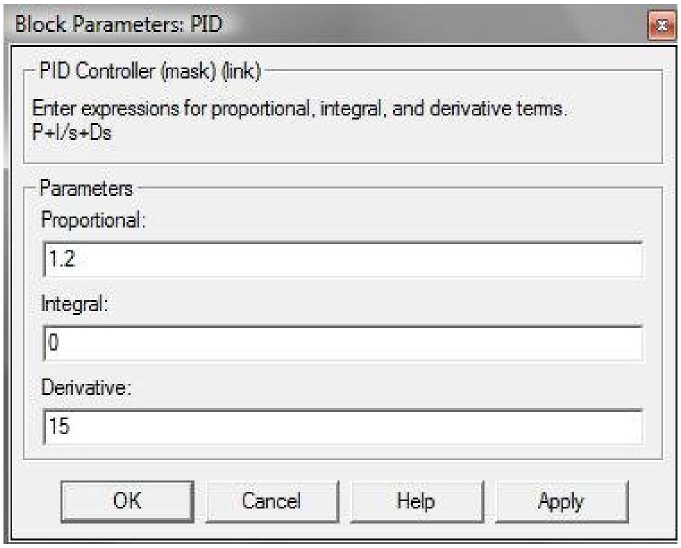


Figure 4.4: This plot shows that the derivative controller reduced the rise time

4.3.2 Proportional-Derivative control

As for PD control, from the table 3.3 shown, we see that the derivative controller (Kd) reduces both the overshoot and the settling time. The closed-loop transfer function of the given system with a PD controller is:

$$\frac{X(s)}{F(s)} = \frac{Kds + Kp}{4s^2 + (2 + Kd)s + (36 + Kp)}$$



Block Parameters: PID

PID Controller (mask) (link)

Enter expressions for proportional, integral, and derivative terms.
P+I/s+Ds

Parameters

Proportional: 1.2

Integral: 0

Derivative: 15

Figure 4.5: Setting the Proportional-Derivative control

This plot in figure 4.6 shows that the derivative controller reduced both the overshoot and the settling time, and had small effect on the rise time and the steady-state error

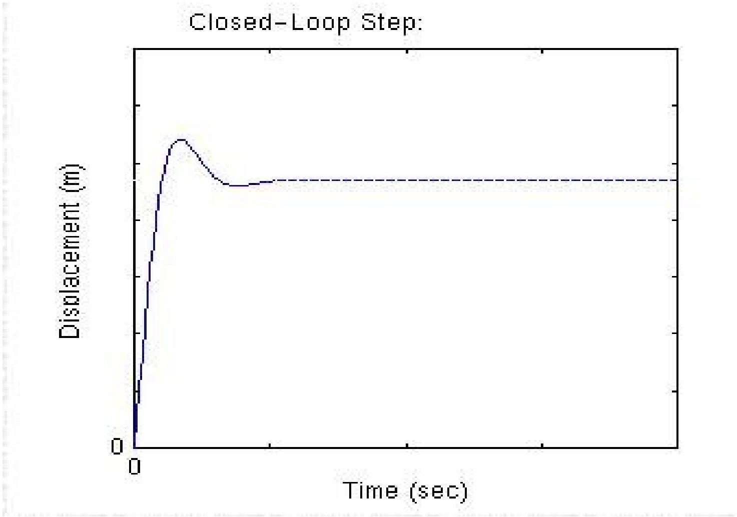
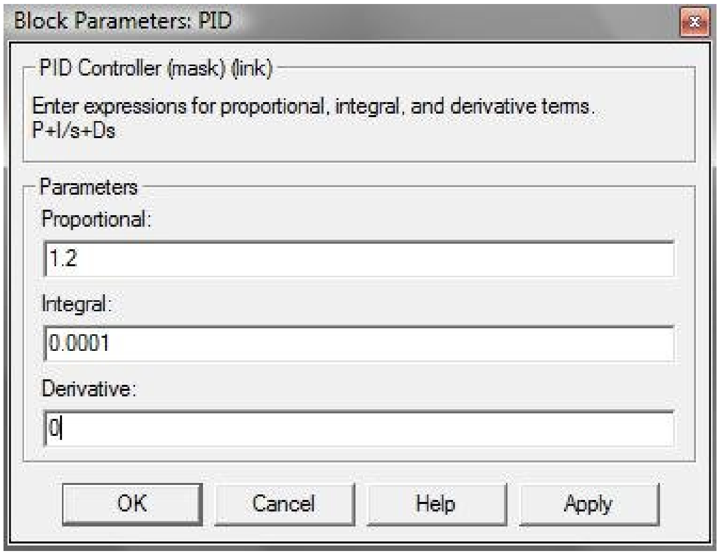


Figure 4.6: Plot shows response from Proportional-Derivatives control

4.3.3 Proportional-Integral control

From the table, we see that an integral controller (Ki) decreases the rise time, increases both the overshoot and the settling time, and eliminates the steady-state error. For the given system, the closed-loop transfer function with a PI control is:

$$\frac{X(s)}{F(s)} = \frac{Kps + Ki}{4s^3 + 2s^2 + (36 + Kp)s + Ki}$$



Block Parameters: PID

PID Controller (mask) (link)

Enter expressions for proportional, integral, and derivative terms.
P+I/s+Ds

Parameters

Proportional:
1.2

Integral:
0.0001

Derivative:
0

Figure 4.7: Setting the Proportional-Integral control

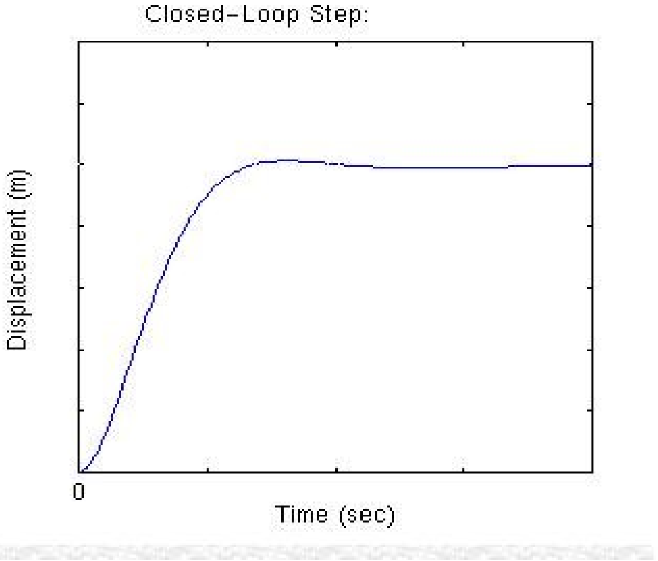


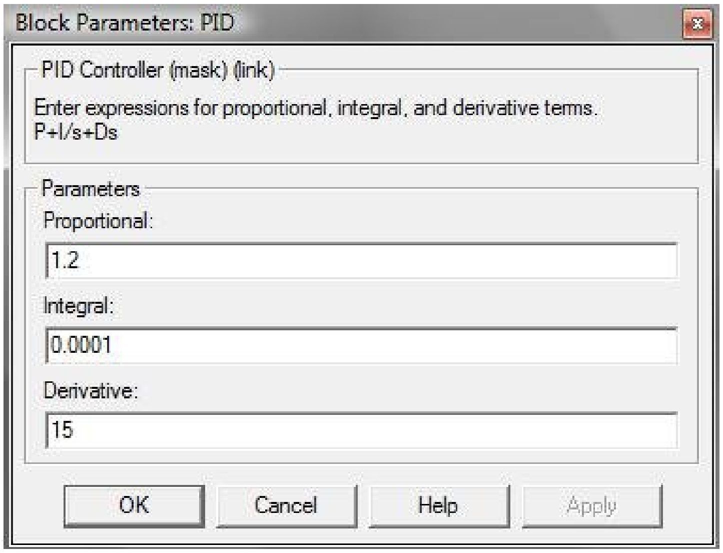
Figure 4.8: Plot shows response from Proportional-Integral control

The proportional gain (Kp) because the integral controller also reduces the rise time and increases the overshoot as the proportional controller does (double effect). The above response shows that the integral controller eliminated the steady-state error.

4.3.4 Proportional-Integral-Derivative control

The closed-loop transfer function of the given system with a PID controller is:

$$\frac{X(s)}{F(s)} = \frac{Kds^{2} + Kps + Ki}{4s^{3} + (2 + Kd)s^{2} + (36 + Kp)s + Ki}$$



Block Parameters: PID

PID Controller (mask) (link)

Enter expressions for proportional, integral, and derivative terms.
P+I/s+Ds

Parameters

Proportional: 1.2

Integral: 0.0001

Derivative: 15

Figure 4.9: Plot shows response from Proportional-Integral-Derivative control

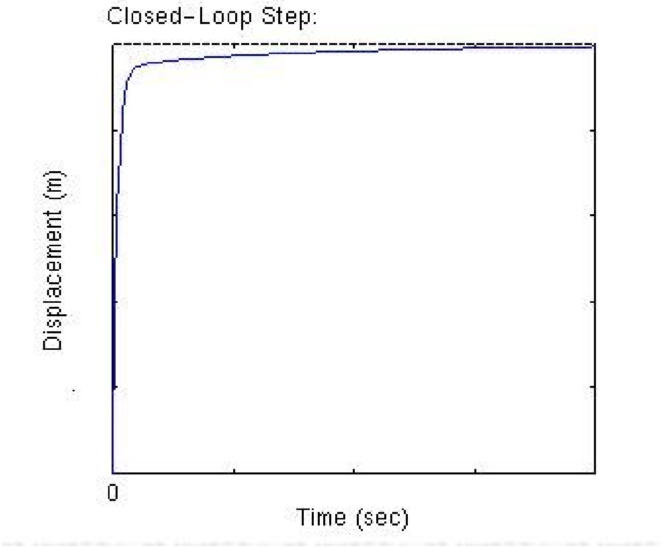


Figure 4.10 Plot shows response Proportional-Integral-Derivative control

4.4 Simulation

4.4.1 Model without Controller

At first, the modeling system is done without a controller which supposes to give an output that had show the disrupted output that consist overshoot, longer settling time.

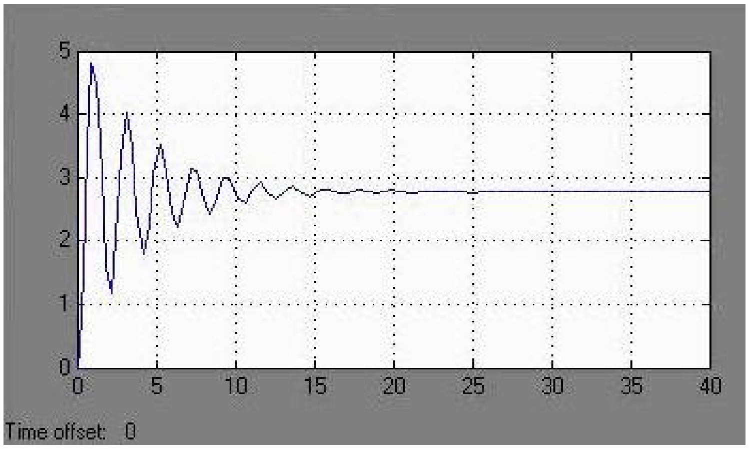


Figure 4.11: Graph shown the output without a controller system

4.4.2 Model with Controller

Figure below show the output for the feedback from the PID controller which had fasten the settling time and overcome the overshoot.

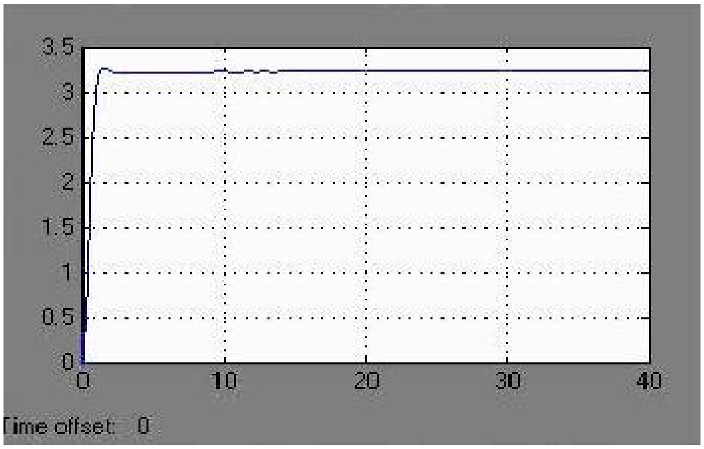


Figure 4.12: Graph shown the output with a controller system

4.5 Comparison

A mass-spring-damper system is created with ***Coulomb friction*** for the damper force. The Coulomb friction (from the non-linear library block) is represented as an offset at zero velocity. The offset for our example is given as 0.5 (with a slope of 1). The coding is shown in Figure 4.5. The output for a combination input = ramp (2t) + step + ramp (5t) is shown in Figure 4.6. The combination input is available as the ***repeating sequence*** in the sources library block. As expected, the Coulomb Friction creates undesired response in the output of the system.

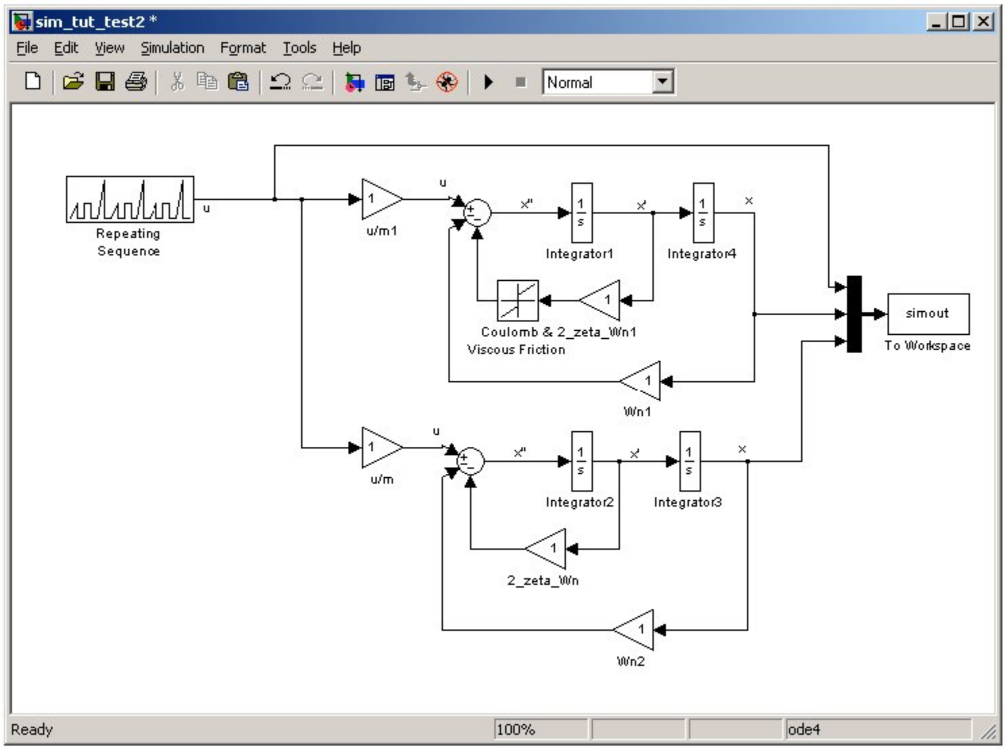


Figure 4.13: Mass-Spring-Damper system with Coulomb friction

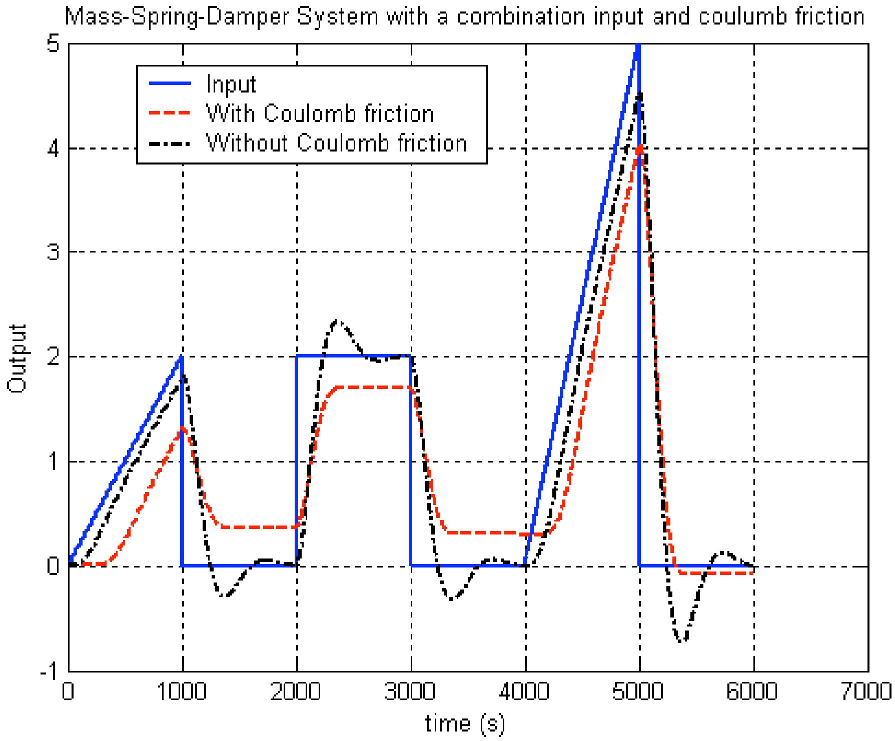


Figure 4.14: Output of mass-spring-damper system with coulomb friction

4.4 Conclusion

In this chapter, the performance of PID controller has been shown compared to the uncontrolled mass spring system. The result determined that the PID controller improved the ride comfort and road handling performances compared to the uncontrolled mass spring system. In conclusion the PID can not overcome the mass spring system problems effectively due to the slightly results. PID controller only can improve the performance of the mass spring system from the uncontrolled system.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The design and implementation of Motor control GUI has been presented in this project. The development of the Simulink® using MATLAB was done after detail study and analysis. Through the development of this project it has conclude that the Proportional Integral Derivative (PID) controller can control the system and interface with the feedback with the proper action

The Proportional Integral Derivative (PID) controller has been successfully implemented in the mass spring system (suspension). The result shows that the PID controller improved the ride comfort compared to the uncontrolled mass spring system.

In the simulation, the performance of PID controller was demonstrated which by using proposed controller; the mass spring (suspension) system can will be perform better than uncontrolled mass spring system (suspension) system.

In estimation works, the manual tuning in determining the value of PID controller, also gives the best result to get the better performance of proposed controller.

5.2 Future Recommendations

For the future development, the optimal controller can be replace by a robust controller such as Proportional Integral Sliding Mode Control (PISMC) strategy or Fuzzy Logic Controller (FLC). These controllers may be being able to overcome the mass spring system which suffers from the mismatched condition. Furthermore, the performance of the mass spring system will be investigated using proposed controller and other controllers.

REFERENCE

[1] Seok-il Son and Can Isik, “Fuzzy Control of an Automotive Active Suspension, “IEEE Transaction, pp 377-381, 1996.

.

[2] Sam, Y.M Osman J.H.S., and Ghani, M.R.A. “ A class of Proportional Integral sliding Mode Control with Application to Suspension system, “ System and Control letter (2003), pp 217-223.

[3] D.A. Crolla, “Mass spring Suspension system control Model, “SAE 892482, pp 557-567, 1989.

[4] Mehdi Farahmand, Caro. Lucas and M.J. Yazdanpanah, “Design an Optimal control for an Automotive suspension system,” IEEE Transaction, pp 688-692, 2003.

[5] L.Wang and T.J. Kamazmierski, “VHDL-AMS based genetic optimization of a logic controller for automotive suspension system,”IEEE Transaction, pp 124-127, 2005

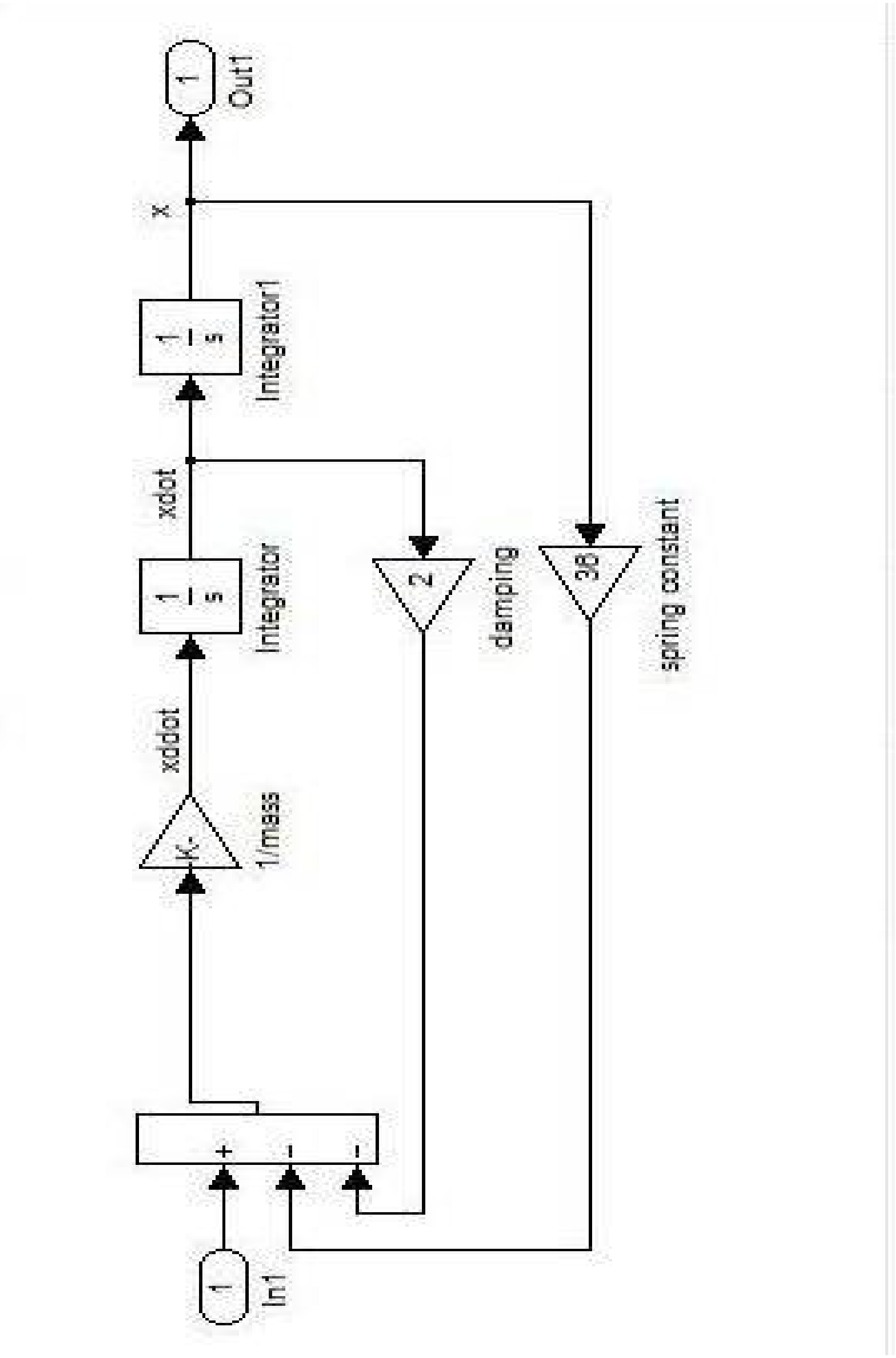
[6] Andrew J. Barr and Jeffrey I.Ray, “Control of an Active suspension system using Proportional Integral derivative, “IEEE Transaction, pp 42-48, 2006

[7] A.J Truscott, “Composite Active Suspension for a mass spring system,” Computing and control Engineering Journal, pp 149-154, 1994

[8] D.Horovat, “Applications of Optimal Control to Advanced suspension design,” Transaction of ASME, journal of Dynamic systems, Measurement and Control, Vol 115, pp 328-342, June 1993.

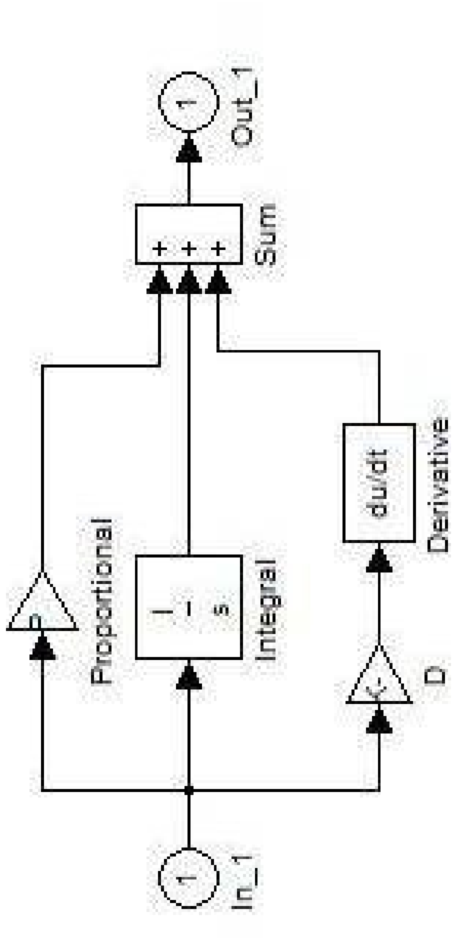
APPENDIX A

BLOCK DIAGRAM OF AN UNCONTROLLED MASS SPRING SYSTEM


1
In1
+
-
-
-K-
1/mass
xddot
1
s
Integrator
xdot
1
s
Integrator1
x
1
Out1
2
damping
38
spring constant

APPENDIX B

PROPORTIONAL INTEGRAL DERIVATIVE DESIGN


1
In_1
P
Proportional
1
s
Integral
K-
D
du/dt
Derivative
+
+
+
Sum
1
Out_1