

University of Southern Queensland
Faculty of Health, Engineering and Sciences

**PIECEWISE ADAPTIVE CONTROLLER DESIGN
BASED ON ECP MODEL 730 MAGNETIC LEVITATION SYSTEM**

A dissertation submitted by

XINCHEN FAN

In fulfilment of the requirements of

Course ENG 4111 and ENG 4112 Research Project

Towards the degree of

Bachelor of Engineering (Electrical and Electronics)

Submitted: October, 2013

Abstract

Many new techniques have been developed in recent years. One of the most notable techniques is the magnetic levitation train, or Maglev. A Maglev uses electromagnetic control systems to levitate a vehicle in a short distance away from a guide way vertically. The idea was firstly patented in Germany by 'Transrapid'. After a few decades of development, the Maglev train has been already used for public service in China. The project studied a similar dynamic control system to control the lift of a maglev train. An ECP Mode 730 Magnetic levitation plant was used in the development of the control system. The system modelling was identified first and then a PID controller were designed, simulated and implemented. It was found that the characteristics of a PID controller are not good enough to such a maglev plant which requires a quicker response and almost no overshoot. A deadbeat controller later was designed to handle the maglev system which could give a much quicker response and no overshoot. The simulation results for a deadbeat controller suggested the overshoot of the system when the step input is 2cm is 0.013 (less than 0.0065%) which can be almost neglected. The settling time for the system response is 0.231 seconds and it has only 0.031 seconds' difference from the desired time. While applying the designed deadbeat controller to the plant, some real-world problems such as oscillations and control errors occurred. The problem was solved at the end and system performance became much better but the small oscillation still exists. It was believed that the small oscillation was coming from the hardware of maglev plant itself. In comparison with a classic PID controller, it was found the settling time has been improved at least 55% at the linearization point and the overshoot was reduced. However, when it comes up to a large movement from the linearization point there was no improved at all. There is a need to apply adaptive control techniques in the further work.

University of Southern Queensland
Faculty of Health, Engineering and Sciences
ENG4111/ENG4112 Research Project

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Health, Engineering & Sciences, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Health, Engineering & Sciences or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled “Research Project” is to contribute to the overall education within the student’s chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

XinChen Fan

Student Number: 0061013549

Signature

Date

Acknowledgements

It would not have been possible for me to finish my dissertation without the help and support of the following people.

Above all, I would like to thank my parents their unconditional love and support which always give me the courage to face each challenge throughout my life.

Besides my parents, I would like to express my deepest gratitude to Dr. Paul Wen for his continues support and patience during the project. Also, I would like to thank Dr Fu Bo for sharing his ideas with me.

XINCHEN FAN

University of Southern Queensland

October 2013

Contents

Abstract	ii
Limitations of Use	iii
Certification of Dissertation.....	iv
Acknowledgements	v
Chapter 1	1
Introduction.....	1
1.1 Project Aims and Objects	1
1.2 Background	3
1.21 Control System Applications.....	3
1.22 Magnetic Levitation Trains.....	4
1.3 Literature Review	7
1.4. Assessment of Consequential Effects and Ethical Responsibilities.....	8
1.5. Timelines and Methodologies	8
Chapter 2	11
ECP Model 730 Maglev Machine, Modelling and Linearization	11
2.1 Introduction	11
2.2 System Modelling.....	12
2.3 Linearization using Taylor' Series expansion	15
Chapter 3	18
PID Controller Design, Optimization and Implementation	18

3.1 Introduction	18
3.2 PID Controller Design	18
3.3 PID Controller Optimization	21
3.4 PID Controller Implementations.....	31
Chapter 4	37
Robust Deadbeat Controller Design, optimization and Implementation	37
4.1 Introduction	37
4.2 Robust Deadbeat Controller design	38
4.3 Robust Deadbeat Controller Simulation and Optimization.....	41
4.4 Robust Deadbeat Controller Implementation	47
Chapter 5	52
Conclusions and Further Work	52
5.1 Conclusions	52
5.2 Further work.....	55
References	56
Appendix A	59
Project specification	59
Appendix B.....	61
Matlab Code for Simulation.....	61
Appendix C	67
Simulink Model	67

Appendix D	69
C code for maglev plant.....	69
Appendix E.....	77
Risk Assessment.....	77

List of Figures

Figure 1-1 representation of nonlinear and linear system

Figure 1-2(a) and (b) guidance system during the launch of a rocket

Figure 1-3 guid way of a maglev train

Figure 1-4 different control systems applying on a maglev train

Figure 1-5 GPI control block diagram

Figure 2-1 Model 730 maglev system

Figure 2-2 system dynamic and configuration

Figure 2-3 magnetics displacement and control effort

Figure 2-4 estimated displacement and control effort curve

Figure 2-5 Taylor's linear approximation representation

Figure 3-1 PID controller block

Figure 3-2 PID controller block with plant

Figure 3-3 plant response without compensations

Figure 3-4 Z-N method S shaped curve

Figure 3-5 Z-N method exhibit sustained oscillation

Figure 3-6 P controller block diagram

Figure 3-7 step response with different K_p

Figure 3-8 step response with a suitable K_p

Figure 3-9 PD controller block diagram

Figure 3-10 K_p and K_d parameters turning 1

Figure 3-11 K_p and K_d parameters turning 2

Figure 3-12 PI controller block diagram

Figure 3-13 PI and P controller response

Figure 3-14 Root locus stability analysis

Figure 3-15 PID controller model

Figure 3-16 initial PID step response

Figure 3-17 full PID Simulink model

Figure 3-18 best simulation result for a PID controller at 2cm linearization point

Figure 3-19 effect for moving away from linearization point

Figure 3-20 implementation result for a 2 cm step response

Figure 3-21 implementation result for a 1.5 cm step response

Figure 3-22 implementation result for a 1 cm step response

Figure 4-1 robust deadbeat controller block diagram

Figure 4-2 detailed robust deadbeat controller block diagram

Figure 4-3 MATLAB script for deadbeat parameters optimization

Figure 4-4 initial simulation result

Figure 4-5 impacts of desired time tolerance

Figure 4-6 System response when the desired time is 0.1 second

Figure 4-7 System response when the desired time is 0.2 second

Figure 4-8 System response when the desired time is 0.4 second

Figure 4-9 Idea deadbeat response

Figure 4-10 the best deadbeat response was achieved by optimization

Figure 4-11 effect for moving away 0.5 cm from linearization point

Figure 4-12 effect for moving away 1 cm from linearization point

Figure 4-13 implementation deadbeat controllers for a 2 cm step response

Figure 4-14 implementation deadbeat controllers for a 2.5 cm step response

Figure 4-15 implementation deadbeat controllers for a 1 cm step response

List of Tables

Table 1 – comparison of a maglev train with a traditional train

Table 2 – Time lines

Table 3 – variables used for system modelling

Table 4 – impact on the system response by turning three parameters for a PID controller

Table 5 – variables and constants for PID controller which are used to in the C language programming

Table 6 – robust deadbeat controller constants look up table

Table 6 – comparison when different desired time tolerances were chosen

Table 7 – Impact on the system when different desired time were chosen

Table 8 – variables and constants for deadbeat controller which were used to in the C language programming

Chapter 1

Introduction

1.1 Project Aims and Objects

Effective and efficient public transports systems play a crucial role in individuals' daily life, especially in urban areas. They offer many advantages over the personal alternatives when it comes to getting large numbers of people from A to B in style such as safety, less congestion, less pollution and lower costs. One of the most notable transportation is railways. Rail transport has been used by people for a long time since 1820s in England and it was mainly used in mining areas. After the industrial revolution, steam locomotive has been developed to export economies across the world. Afterwards, electric railways, diesel power rails and high speed rails have been developed to meet people's needs. At the moment, there are many researches and developments ongoing focused on Maglev trains. The idea is to lift a train by magnet fields and then apply a directional force to a train to keep it moving forward.

Since maglev trains are floating above the track, the friction between wheels and the track can be neglected and better aerodynamic design are required to overcome air frictions. By levitating trains above the track, less energy loss and higher speed can be achieved. Also as the speed increases, much bigger air friction would be. Some researchers suggest that to operate maglev trains inside a vacuum tunnel but it brings safety concerns. According to Transrapid®'s documents [1], the normal energy consumption of a maglev train is approximately 50-100Kw per section for levitation and travel. The design of maglev trains involves several parts, and the control system is the most important part of them. The control system collects current input and output and then adjusts the current go through the guide way and track. For Transrapid® Maglev train designed in Germany, the gap between the train and the guidance coil is around 10cm [1] and the free space of the train is estimated to be around 1-2cm, which is considerably small. A precise and accurate control system

must be designed to deal with the tolerant distance between the train and the rail track. Also, it must be very sensitive so that once the actual distance is larger than the tolerance, the current going through the coil will be adjusted properly to make sure that the train is operating at a safety range. Therefore, there is a need to understand and apply control law to the magnetic system in order to operate the train safely.

In order to understand and investigate complexity of the control system, a magnetic levitation system plant ECP 730 (Educational Control Products) is available at the control lab which is a piece of lab equipment based on maglev system. This maglev plant has only one degree of movement either moving up or down and it has two types of configuration: Multi input Multi output (MIMO) as well as Single input and Single output (SISO). Moreover, a laser sensor is monitoring the positions of disks in real time. More details will be provided in the next sections.

The maglev plant was designed for students to apply control theory. It illustrates the complexity of the magnetic control system and simulates the same situations as the maglev trains are lifted up by the bottom coils. Therefore, by applying control theory on this plant gives students the opportunity to understand how to apply the complex control theorem learnt on the class into a real world application. The Maglev plant has several functions and demo programs and after getting familiar with these demo experiments, students are able to write their own control algorithm by C language program. It is an ideal platform for applying control algorithm to simulate the same situation for a maglev train.

All the control system design and analysis methods are only applicable on linear systems. Unfortunately, a Maglev system is a nonlinear system because the output is not proportional to the input. A linear system has two properties which include superposition and homogeneity [2]. As shown below,

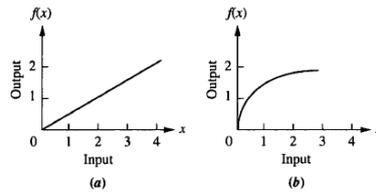


Figure 1-1 Nonlinear and linear system

Fig1-1 (a) is a linear system where the output is proportional to the input whereas Fig1-1 (b) does not have the same property. A designer can make a linear approximate to a nonlinear system [2]. One of the linearization methods will be introduced later.

A deadbeat control is defined as a control system which can reach the steady state in the shortest time for a given input. It is the idea control system but in real word it is hard to achieve because the error always occurs. A deadbeat controller will be designed to control the displacement of the disks. The system will be firstly simulated using MATLAB and SIMULINK to make sure that the desired response will be obtained before test on the real plant to reduce the risk of damaging the plant. The performance of the controller designed will be compared with the PID (Proportional, Integral and derivative) controller. To make sure the control system I designed can be applied in any sort of nonlinear system.

1.2 Background

1.21 Control System Applications

A control system is normally used to control the position, velocity, etc. Engineers should pay enough attention to the design of a control system to make sure it is safe and reliable. Once the system is applied to real life, the fault of the design will cause the loss of life and property. As can be read from the history, most of the tragedy is due to the careless and irresponsible design by some engineers. To demonstrate the

importance of control system design, an example is given as below,



Figure 1-2(a)

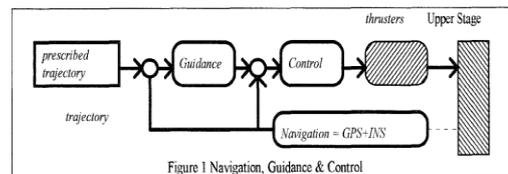


Figure 1-2 (b)

As can be seen from the fig above, a guidance system is crucial during the launch of a rocket. It compares prescribe trajectory with current position and send control signal to thrusters to make sure that the rocket is at the right track [3]. Obvious the failure of the control system will make the racket lose its track, causing huge loss of life and property. There are considerable huge amount of research papers have been published on how to design control laws (algorithms) for various purpose. The control techniques are roughly classified into five categories: classical control, optimal control, robust control, nonlinear control and intelligent control. Classic control is basically PID (Proportional – integral – derivative) control which developed in 1940s and mainly used for industrial process control. Optimal control including linear quadratic regular control, H_2 control and Kalman filter was developed in 1960s to achieve certain optimal performance. Robust control was developed in 80s and 90s to deal with systems with uncertainties, oscillations and disturbances with high performance. Nonlinear control is a hot research topic in the recent decades and it is designed to handle nonlinear systems such as maglev system, rockets, etc. Intelligent control such as adaptive control, neural and fuzzy control is developed in 1990s to handle system with unknown models. The purpose of this project is to design a controller based on the previous control techniques for a maglev system and compare the performance and evaluate the error occurs during the design process.

1.22 Magnetic Levitation Trains

The magnetic levitation (maglev) train is considered as a new generation transportation system which provides rapidity, reliability, safety and environmental friendly [2]. The development of Maglev trains has a long history. It was patented by

Herman Kemper from Germany since 1934. After a few decades' research and development, finally, accomplished practise public service [2]. Lee and Kim have done some research on the history and development of Maglev trains and they came up to the conclusion that Maglev trains have more advantages comparing to the conventional wheeled trains. We can see from the table below.

COMPARISON OF MAGLEV AND WHEEL-ON-RAIL SYSTEMS

	Maglev System	Iron Wheel-on-Rail System
Vibration & Noise	No mechanical contact 60-65 [dB]	Contact between wheels and rails, 75-80 [dB]
Safety	No possibility of derailment	Derails from a minor defect
Guideway	Light vehicle & distributed load → light-weight	Heavy & concentrated load → Hardy structure
Maintenance	Very little	Periodic replacement of wheels, gear, rails, etc
Grade	About 80-100/1000	About 30-50/1000
Curve	In 30 [m] in radius	In 150 [m] in radius

Table 1 comparison of Maglev train and traditional train (source:Lee&Kim)

As we can see from the table, Maglev trains overweigh the conventional trains in terms of low noise and vibration, less safety issues, light guidway, low maintenance lost small turning curve and high speed capability.

Since maglev train is only open to public service in three countries (China, South Korea and Japan), limited documents about the standards and design aspects were found. The technical document was found is from Transrapid[®] which is a German high speed maglev train. It was first adopted for commercial use in 2004 by Shanghai Maglev train [4]. According to its technical documents [5], there are five systems involved in the maglev train operation which are levitation system, vehicles, propulsion system, operation control system and guide way.

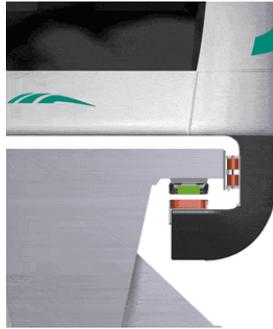


Figure 1-3 levitation train guid way

Figure 1-3 shows how the levitation system works by controlling the current goes through the coil both vertically and horizontally.

Transrapid® vehicles comprise a minimum of two sections, each with approx. 90 seats on average. According to application and traffic volume, trains may be composed of up to ten sections (two end and eight middle sections). The disadvantage is that the loads of the maglev trains are limited and one track can only have one vehicle at same time. The vehicle is propelled by a synchronous longstator linear motor and the speed can be modified by varying the frequency of AC power supply. Fig4 shows the operation control system is like a guidance to monitor the operation process and received the all the data by using an on board system and send them back to the centre control centre by fibre optics. The guid way is basic the track that a maglev train is floating on it.

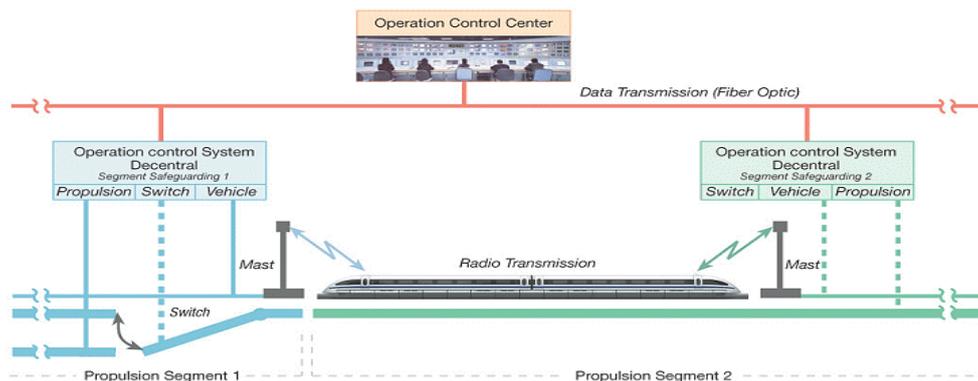


Figure 1-4 Detailed control systems for a maglev train

It is clearly that the control system plays an important role in the operation of a maglev train. Hence, to further understand and investigate the characteristics of the control system is crucial. The aim for the project is to simulate and implement the design on the Model 730 Maglev Machine.

1.3 Literature Review

Literature reviews are required before the project commences in order to have a good grasp and fully understanding the concepts of maglev systems, nonlinear control and different controller design techniques. The traditional controller is not accurate enough to satisfy modern maglev system due to the linearization errors which occur when there is a huge displacement refers to the operation points.

To resolve the problem, an alternative controller technique known as fast online algebraic identification could be used to control nonlinear magnetic levitation system. Rafeal Morales, Vicente Feliu and Hebertt Sira- Rairez proposed a method which is an adaptive control based on fast, on line, algebraic parameter estimation, exact linearization and generalized proportional integral (GPI) output feedback control. Also, in the paper, the adaptive controller is implemented on a laboratory prototype with excellent experimental results for both, stabilization and trajectory tracking tasks. This control closed-loop adaptive system showed below is a potential solution to solve such a nonlinear and unstable maglev system. However, the control algorithm itself is very complex. More details will be provided in the next few chapters.

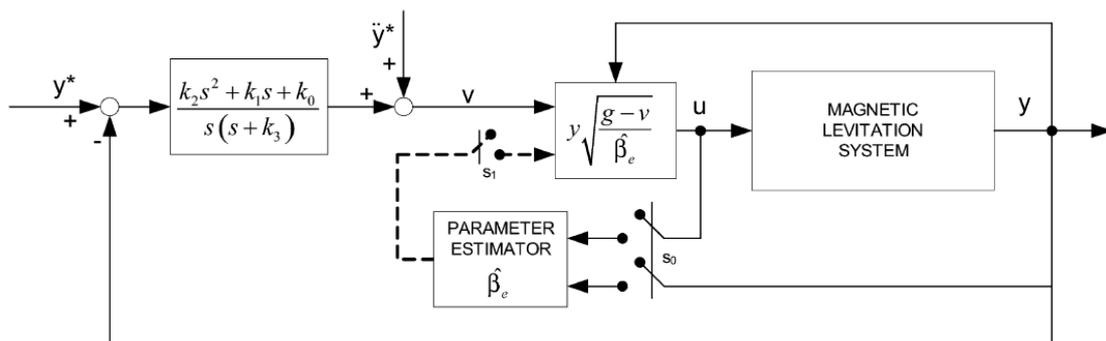


Figure 1-5 GPI controller

1.4. Assessment of Consequential Effects and Ethical Responsibilities

The risk assessment has been undertaken to ensure that the safe use of Maglev plant and all the equipment in the control lab should be operated and utilized in a safe, reliable and cost effective manner. As mentioned in the outline, there is no manufacturing required other than the plant which have been used. The design of the Maglev Apparatus complies with appropriate Australian standards. The power consumption may become a negative impact in terms of environment friendly. Therefore, it is necessary to turn off the plant once experiments have done. Moreover, the user manual should be read before start working on the plant thereby avoiding unnecessary damage on the maglev plant. When it comes to real life applications, the adverse effects of maglev trains on environment should not be ignored which include noise pollution, ecological effects and electromagnetic radiation. Hence, environmental assessment should be taken. According to Engineers Australia code of ethics [6], 'practise engineering to foster the health, safety and wellbeing of the community and the environment' is one of the ethics which should be followed. Also, the ethical responsibilities of a professional engineer include demonstrate integrity, practise competently and promote sustainability. By keeping code of ethics in the mind, engineers use their knowledge and skills for the benefit of the community to create engineering solutions for a sustainable future. A detailed risk assessment is provided in the Appendix.

1.5. Timelines and Methodologies

Timelines

In order to finish the project before the due date, the following table shows the tasks that have been distributed evenly for each month.

Milestone/Task	Prerequisite	Time slot
	None	Feb to March

1) Investigate the ECP Model 730 hardware and software		
2) Test the system and study the existing demo program and experiments	1)	March to April
3) Write a script to linearize the plant using Taylor's theorem	1)	April to May
4) Design and simulate a PID controller for the Maglev system	1), 3)	June
5) Implement, test and evaluate the design using the Model 730 Maglev plant	2), 4)	June
6) Design and simulate a deadbeat controller for the plant	4)	Ongoing
7) Implement, test and evaluate the deadbeat controller using the Model 730 Maglev plant	5), 6)	September

8) Analyse the results	7)	September
9) Compile Dissertation		September to 24th of October

Methodologies

Basically, there are three parts of tasks are involved in this project, research, computer simulation, and the physical implement on the real maglev plant. The first task requires literature reviews and MATLAB simulations based on the controller which will be implemented on the real plant. After the desired results are obtained, the designed controller could be applied to the real maglev plant. Also, a detailed assessment will be provided for comparison with the specifications and characters of the controllers that has been designed.

Chapter 2

ECP Model 730 Maglev Machine, Modelling and Linearization

2.1 Introduction

In order to design a control system which is able to be control a maglev train, the ECP Model 730 Maglev machine should be used. Educational Control Products have designed such a system which is ready to use, and all the experiments and implementations is based on it.

The Model 730 Magnetic Levitation (MagLev) can be transformed to either single input single output (SISO) or multi-input multi-output (MIMO) configuration. By using repulsive force from generated from bottom coil to levitate a single magnet, a SISO system is built. Also, both lower and upper coils can be used to set up a MIMO system. Consequently, the force between two magnetic disks needs to be taken into consideration.

Moreover, the plant has inherently strong nonlinearities due to the natural properties of magnetic fields. The nonlinear properties need to be compensated and linearized by different algorithms so that the control system may be regards as a linear system in a certain range. More details will be provided in the next section.

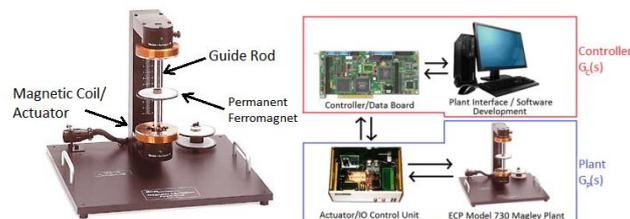


Figure 2-1

As showed in Figure 2-1, the Model 730 Maglev machine consists of three parts [7], the physical plant, which uses a coil with an aligned magnetic field parallel to a magnetic on a guide rod. Hence, it is a one degree control system while the maglev train is a two dimensional control system. A laser sensor is used to measure the displacement of the disk. The control box controls the physical plant, it converts control effort into current go through the coil and there are several DAC and protection units are involved. The control box is also sending and receiving signals from plant. While the control board is like a bridge between a PC and control box, it has a real time DSP on board. It produces control signals from the PC and sends them to the control box. The control algorithm is written in C language. The last component is the PC which has a user interface inbuilt.

2.2 System Modelling

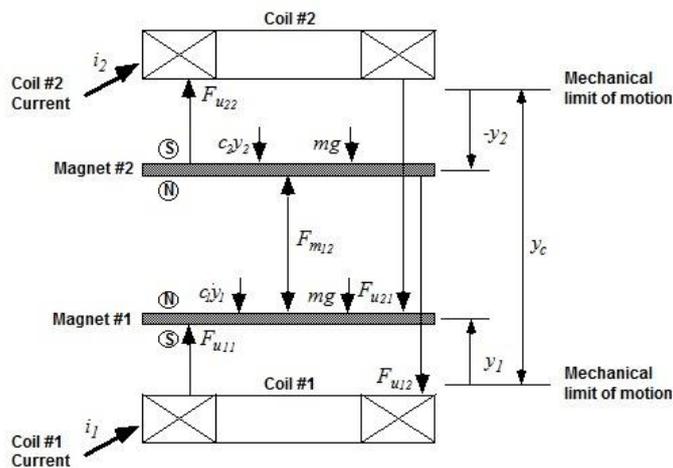


Figure 2-2

Constant or Variable	Definition
----------------------	------------

c_1, c_2	Friction/wind resistance
F_{m12}	Force between each magnet
F_{u22}	Force applied to magnet 2 from coil 2
F_{u11}	Force applied to magnet 1 from coil 2
F_{u21}	Force applied to magnet 2 from coil 1
F_{u12}	Force applied to magnet 1 from coil 1
m	Mass of the magnet, 120g
g	Force of gravity, assumed 9.81m/s ²
y_1	Distance between magnet 1 and coil 1. The linear sensor gives 10000 counts/cm
y_2	Distance between magnet 2 and coil 2. The linear sensor gives 10000 counts/cm
y_c	Distance between the coils
i_1, i_2	Current through the respective coils (input)
U_1, U_2	Control effort, proportional to current. Linearization gives 10000 counts/N
a, b, c, d	Constant

Table 2

Figure 2-3 shows Free Body Diagram & Dynamic Configuration and Table 2 shows variables were used for the modelling,

Firstly, SISO configuration, where only bottom magnet and bottom coil are considered, $F=ma$, a is the second derivative of accretion

$$m\ddot{y}_1 = F_{u11} - mg$$

Where,

$$F_{u11} = \frac{I_1}{a(y_1 + b)^4}$$

In my case I used 4th order approximation, and a , b is basically constants. It is measured by hand and then figured out by using MATLAB coding.

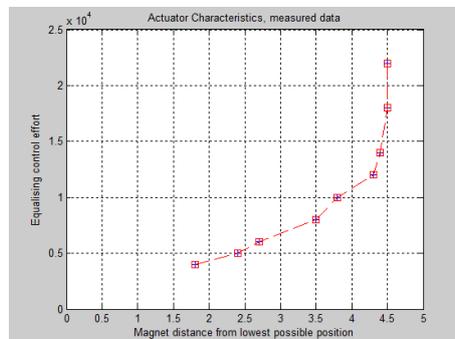


Figure 2-3

Figure 2-3 shows the measured data, as we can see it does not have linear relation

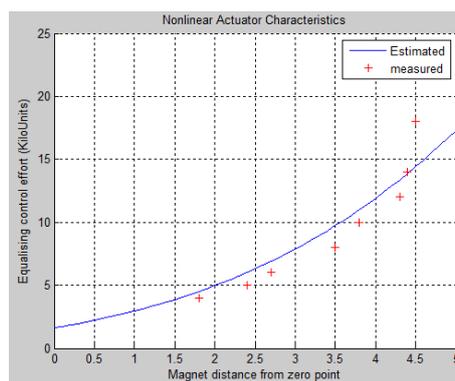


Figure 2-4

Figure 2-5 shows the Estimated values with a and b selected by experiments and the measured values.

2.3 Linearization using Taylor' Series expansion

One of the effective methods to linearize is to apply Taylors' Series expansion, firstly an operation point should be chosen and this will give the ability to apply linear approximately to the system.

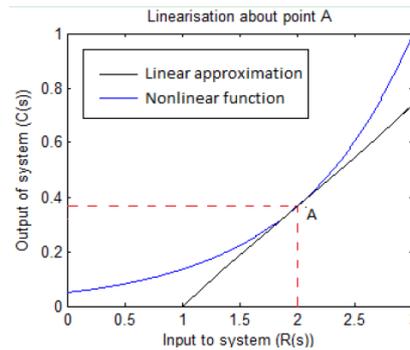


Figure 2-6 Taylor series representation

Figure 2-6 shows the idea of Taylor's approximation, an operation is chosen and when it moves away from the operation point, the error occurs. This error will bring problems for the further controller design.

Taylor's series expansion method is showed below,

$$f(x) - f(x_0) \approx \left. \frac{df}{dx} \right|_{x=x_0} (x - x_0)$$

or, $\delta f(x) \approx m|_{x=x_0} \delta x$

According to the documentation, the differential equation for the control system SISO can be representation as,

$$\frac{d^2 y_1}{dt^2} + \frac{c_1}{m} \frac{dy_1}{dt} + \frac{c}{m(y_{12} + d)^4} + \frac{U_2}{ma(y_c - y_1 + b)^4} + g + \frac{U_1}{ma(y_1 + b)^4} = 0$$

The above equation represents that when the control system is at a balance state ($y_{10} = 2 \text{ cm}$ and $u_{10} = 8000 \text{ counts}$), the friction between rod and magnetic disks is negligible at this point.

By applying Taylor's series expansion at the state point, system can be represented in state space form:

$$\dot{x} = Ax + BU(t)$$

$$Y = Cx$$

For SISO,

$$x = \begin{bmatrix} y_1 \\ \dot{y}_1 \end{bmatrix}, A = \begin{bmatrix} 0 & 1 \\ \frac{4u_1}{ma(y_{10}+b)^5} & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ \frac{1}{ma(y_{10}+b)^4} \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Similarly, for MIMO,

$$\begin{aligned} m\ddot{y}_1 + \left(\frac{4c}{(y_{120} + d)^5} + \frac{4u_{10}}{a(y_{10} + d)^5} \right) (y_1 - y_{10}) - \frac{4c}{(y_{120} + d)^5} (y_2 - y_{20}) \\ = \frac{1}{a(y_{10} + b)^4} (u_1 - u_{10}) \end{aligned}$$

$$x = \begin{bmatrix} y_1^* \\ \dot{y}_1^* \\ y_2^* \\ \dot{y}_1^* \end{bmatrix}, A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -(k_1' + k_{12}') & 0 & \frac{k_{12}'}{m} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_{12}'}{m} & 0 & -(k_2' + k_{12}') & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 \\ k_{u11}'/m & 0 \\ 0 & 0 \\ 0 & k_{u22}'/m \end{bmatrix}, C = \begin{bmatrix} C_1 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 \\ 0 & 0 & C_3 & 0 \\ 0 & 0 & 0 & C_4 \end{bmatrix}, U(t) = \begin{bmatrix} u_1(t)^* \\ u_2(t)^* \end{bmatrix}$$

Where,

$$y_i^* = y_i - y_{i0}, \quad i = 1,2$$

$$u_i^* = u_i - u_{i0}, \quad i = 1,2$$

$$k_1' = \frac{4u_{10}}{a(y_{10} + b)^5}$$

$$k_1' = \frac{4u_{20}}{a(-y_{20} + b)^5}$$

$$k_{12}' = \frac{4c}{(y_{12_o} + b)^5}$$

$$k_{u1}' = \frac{1}{a(y_{1_o} + b)^4}$$

$$k_{u2}' = \frac{1}{a(-y_{2_o} + b)^4}$$

Equilibrium control effort values

$$u_{1_o} = a(y_{1_o} + b)^4 \left(\frac{c}{(y_{12_o} + d)^4} + mg \right)$$

$$u_{2_o} = a(y_{2_o} + b)^4 \left(-\frac{c}{(y_{12_o} + d)^4} + mg \right)$$

The linearization for the plant is based on the desired operation point, the accuracy decreases dramatically as the magnet moves away from the point. Therefore, the fact that the system model is accurate and precise in small magnetic movements should be addressed. Since the system model and linearization have been done, control theory can be applied to the physical system.

Chapter 3

PID Controller Design, Optimization and Implementation

3.1 Introduction

PID controller is an abbreviation of proportional-integral-derivative controller which is one of the most common and conventional controller which is widely used in today's process plants around world, not to mention the fact that the earliest examples of a PID controller was developed in 1911 and the first theoretical analysis of a PID controller was published in 1922 [8]. Despite the abundance of sophisticated tools, including advanced controllers, the PID controller is still the most widely used in modern industry, controlling more than 95% of closed-loop industrial process [9]. It can be tuned by operators without extensive background in Controls, unlike many other modern controllers that are much more complex but often provide only marginal improvement. In fact, most PID controllers are tuned on-site.

Hence, PID controller is the one that chosen to be applied on the maglev plant at the beginning in order to observe and acquire more details about the physical modelling and on the purpose of developing more complicated controller design in the future. In this chapter, the details of PID controller design and optimization will be explained.

3.2 PID Controller Design

The PID controller diagram showed as below, A PID controller is a simple three-term controller. The letters P, I and D stand for:

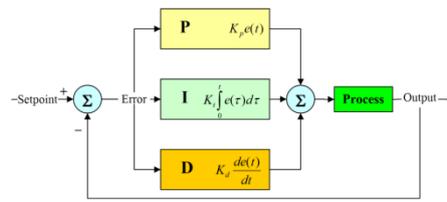


Figure 3-1 PID controller

- P – Proportional

In time domain: $c_p(t) = K_p e(t)$

P is proportional to the system error

- I – Integral

In time domain: $c_I(t) = K_i \int e(t) dt$

I is proportional to the summation of the system error

- D – Derivative

In time domain: $c_d(t) = K_d \frac{d_e(t)}{d_t}$

D is proportional to the change rate of the system error

Therefore, the combination of these three terms is the transfer function of PID controller in time domain, it shows as following:

$$c_{pid}(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{d_e(t)}{d_t} \right)$$

Alternatively,

$$c_{pid}(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{d_e(t)}{d_t}$$

Where $K_i = \frac{K_p}{T_i}$ and $K_d = K_p T_d$.

By taking the Laplace transform of the above equation, the transfer function in S domain can be obtained as,

$$C(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right)$$

The block diagram for PID controller and maglev plant can be represent as below,

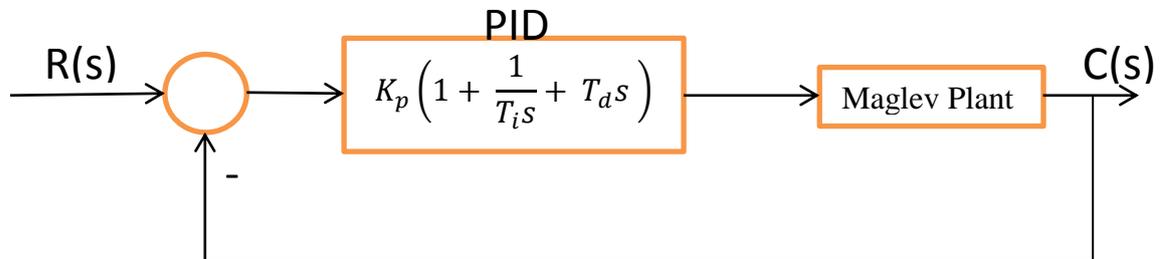


Figure 3-2 PID controller with a maglev plant

Once a system model of the plant can be derived mathematically, then it is possible to apply various design techniques for determining three parameters of the controller that will meet the transient and steady- state specifications of the close – loop system. Basically, the proportional term in the controller generally helps in establishing system stability and improving the transient response while the derivative term is often used when it is necessary to improve the closed loop response speed even further. Conceptually the effect of the derivative term is to feed information on the rate of change of the measured variable into the controller action.

There are four major characteristics of the closed-loop step response for a control system, they are:

1. **Rise Time:** the time it takes for the plant output y to rise beyond 90% of the desired level for the first time.
2. **Overshoot:** how much the peak level is higher than the steady state, normalized against the Steady-State.
3. **Settling Time:** the time it takes for the system to converge to its steady state.
4. **Steady-state Error:** the difference between the steady-state output and the desired output.

The increasing of each controller parameters K_p, K_i, K_d will definitely change the characteristics of the closed-loop step response. Table 3-1 summarized all the effects due to the increment of controller parameters.

Response	Rise Time	Overshoot	Settling Time	S-S Error	Stability
K_p	Decrease	Increase	NT	Decrease	Decrease
K_i	Decrease	Increase	Increase	Eliminate	Decrease
K_d	NT	Decrease	Decrease	NT	Improve

Table 4 – impact on the system response by turning three parameters for a PID controller

NT: No definite trend, minor change.

Typically, Table 4 could be used to design a PID controller as followings,

- Determine what characteristics of the system need to be improved.
- Use K_p to decrease the rise time.
- Use K_d to reduce the overshoot and settling time.
- Use K_i to eliminate the steady-state error.

3.3 PID Controller Optimization

As mentioned in the previous chapter, PID controllers are difficult to be tuned in practise with most of the tuning done manually which is difficult and time consuming. Hence, techniques of tuning the parameters of PID controllers are of great significants. After the 1930s, the PID controller has been widely accepted and used in process industries.

The maglev plant is nonlinear hence the linearized plant should be utilized in order to develop the control law. In system modelling section, the maglev plant was linearized at a magnet position of 2 cm by Taylor's theorem. Consequently, the linearized plant could only represent the maglev plant when its magnet position is at 2 cm. Any movements away from this operation point will increase its error and inaccuracy. The follow equation shows the result of linearization for a magnet position of 2 cm,

$$G_p(s) = \frac{1111}{s^2 + 4s + 478.5}$$

And the step response for the maglev plant showed blow,

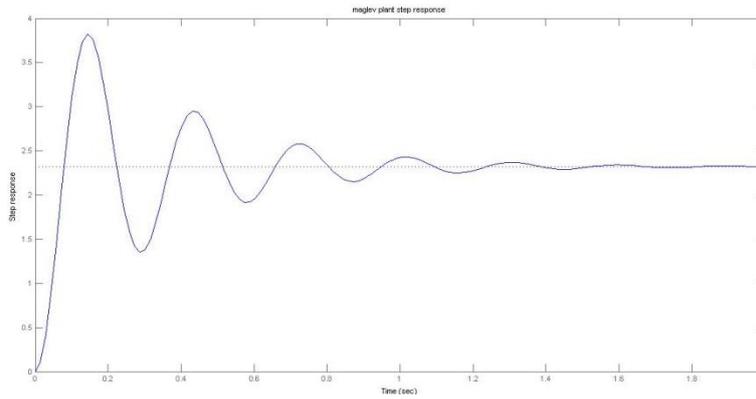


Figure 3-3 Maglev plant response

It can be easily noticed that the model of maglev plant is a second order underdamped system and without compensation, the offset error is considerable large. Hence, a PID controller is required to compensate the errors. Numerous of engineers and researchers have proposed different methods for PID controller parameters tuning such as Ziegler - Nichols tuning method, Kappa – Tau tuning method and analytical pole placement method etc. The first tuning method was chosen is Ziegler – Nichols tuning method. In 1942, Ziegler and Nichols, both employees of Taylor Instruments, developed simple mathematical procedures for tuning PID controllers. These procedures are now accepted as standard in control systems practise [2]. There are two methods called Ziegler – Nichols tuning rules, the first method and the second method. In the first method [9], the response of the plant to a unit-step input could be obtained experimentally. If the plant involves neither integrators nor dominant complex-conjugate poles, then its unit-step response curve may look S-shaped, as shown in Figure 3-4.

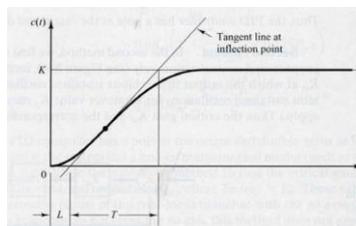


Figure 3-4 S curved plant response

It is clearly that the unit step response for the maglev system does not look like an S-shaped curve. Hence the first method is not applicable to the maglev system. For the second method, K_i is set to infinity and K_d is set to zero. Using the proportional

control action only, increase K_p from 0 to a critical value K_{cr} at which the output first exhibits sustained oscillations. The critical gain K_{cr} and the corresponding period P_{cr} are experimentally determined (see Figure 3-5). A look up table could be used to identify the three parameters. However, in the maglev system, no matter which K_p is chosen, the output does not exhibit sustained oscillations and it is not suitable to let the maglev system work on critical oscillation conditions. As a result, the second method could not be applied on the maglev plant model.

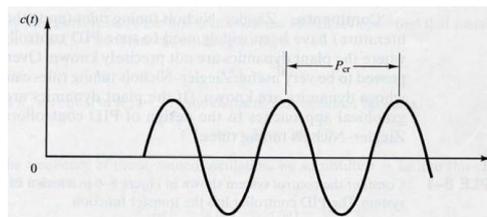


Figure 3-5 exhibits sustained oscillations Z-N method

Based on the previous analysis, it is obviously neither of Ziegler - Nichols tuning first and second method is applicable on the maglev plant model due to the natural characteristics and transient response of this maglev plant. Also, some other standard PID tuning methods such as Cohen-Coon method, Tyreus-Luyben method and Autotune method are further investigated but none of them is suitable for such a complex maglev plant. Hence, finally, the three parameters for a PID controller are tuned manually as following.

According to the tuning procedure, the P- Proportional parameter should be determined first.

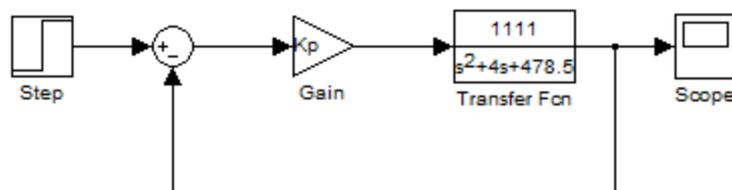


Figure 3-6 P controller block diagram

The previously Fig shows the a Proportional control system, the transfer function of the system is $C(s) = \frac{1111k_p}{s^2+4s+478.5}$, A MATLAB for loop is written to verify the increments of k_p , k_p was set up from 1 to 25, and step size is 1, by running the for loop, the response can be obtained as following.

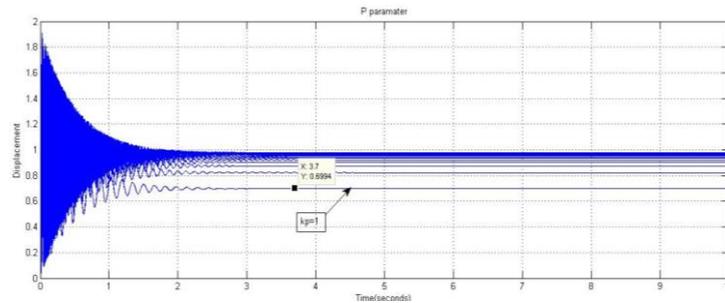


Figure 3-7 step response with different Kp

It is clear that as k_p increases, the state error decreases and the overshoot of the system raises dramatically also the increment of k_p declines the system stability since the system fluctuates rapidly at the beginning and settle done gradually afterwards. To observe the effect of increasing k_p more clearly, a threshold was set up to reach 90% of the steady state, in other words, 5% of error is acceptable and it showed below,

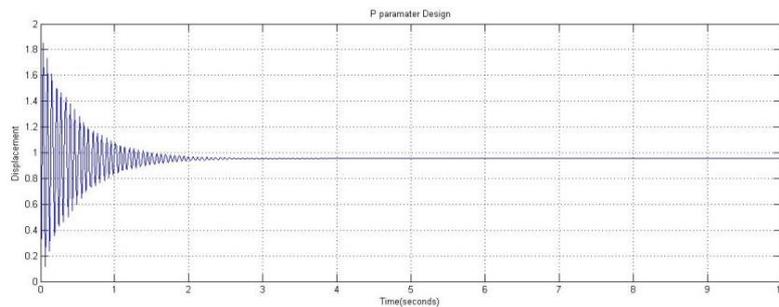


Figure 3-8 step response when Kp =4

Ideally, the minimal steady state error of the system response is preferred but there is always a compromise between the steady state error with overshoot and instability. Hence when $K_p = 4$, the DC gain is 0.90279 and the steady state error is $1 - 0.90279 = 0.09721$.

After the P parameter is defined, the next step is to determine the parameters for PD control. The PD controller adds a derivative term based on predefined P parameter which has already done in the previous step is able to improve the system performance. The PD control diagram shows below, it looks a bit complicated comparing to the P control algorithm.

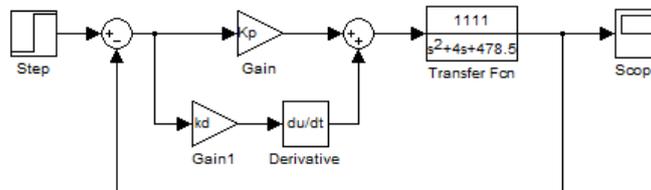


Figure 3 -9 PD controller Block Diagram

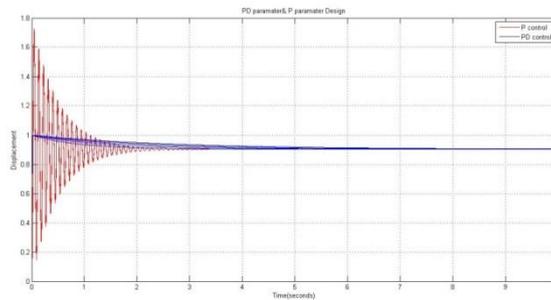


Figure 3-10 K_p and K_d parameters turning 1

As can be seen from figure 3-10, the P parameter remains same and D term increases from 1 to 3. The result suggests that PD term does not change the system steady – state error in fact, increasing K_d could reduce system overshoot and settling time effectively. It is possible to increase K_p to achieve a better steady state error. After tried to increase both K_d and K_p , it has been found that when K_d and K_p come a considerable large number, the overshoot, settling time and steady state error are nearly zero which can achieve a nearly perfect response. However, in real physical system, K_d and K_p are limited in a certain range, in other word, such an idea result does not occur in the real system which means these value shall not be used and Figure 3-11 is just a representation for the idea scenario.

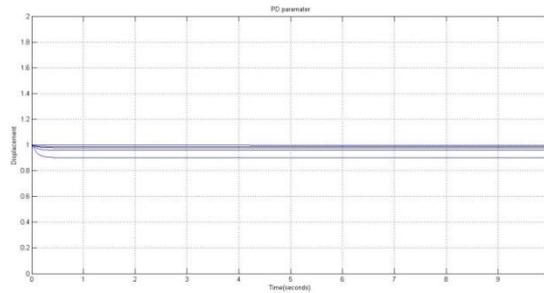


Figure 3-11 Kp and Kd parameters turning 1

Since PD parameters have been chosen, PI control parameters should also to be determined. PI control is basically a combination of a proportional term and an integral term. As mentioned in P parameter, the increment of P will reduce the steady state error but the side effect is the huge overshoot and instability. By introducing integral term into the system, those side effects could be eliminated. The following two figures show the PI control diagram and the comparison of single P parameter and PI control parameters,

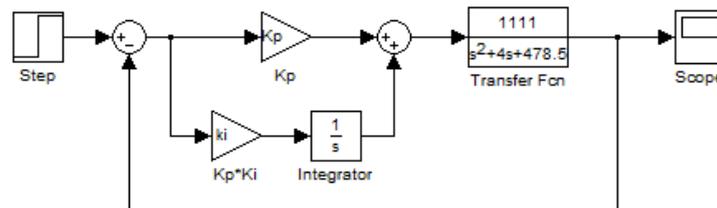


Figure 3-12 PI controllers Block Diagram

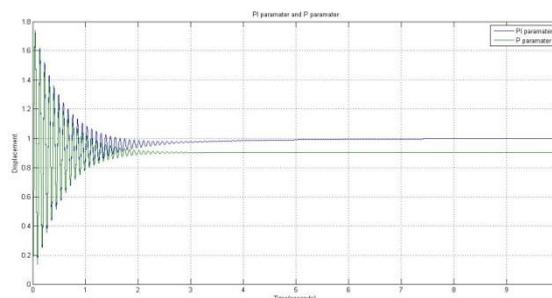


Figure 3-13 PI and P controller response

When looked at the figure 3-13, it is easily to tell that in PI control, the steady- state error decreased dramatically but system overshoot and stability have not improved at all. Also further increment of Integral term leads to more overshoot and produce more oscillations into the system. To solve the problem, it is necessary to compromise K_i with K_p , which is to increase K_i to make rise time shorter and get rid of steady state

error, at same time, reduces K_p to improve system performance. During the simulation, it was found that the system becomes unstable with K_i increase to a large value. Hence root locus technique was used to identify the problem as figure 3-14

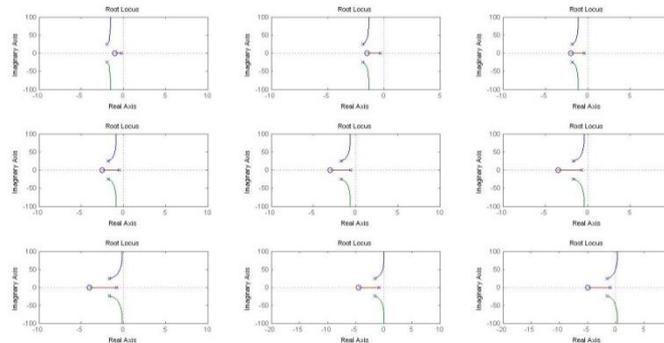


Figure 3-14 root locus stability analysis

The above figure shows when K_p is 0.5, K_i is from 1 to 5 and the step size is 0.5, as K_i reaches 4, the system is in a critical state and when K_i is bigger than 4, the system become unstable which means the value of K_i should not excess 4 when K_p is 0.5.

Finally, A PID control system can be tuned after all these procedures above. PID control has all the combination of three terms which give the best response. The control diagram looks below,

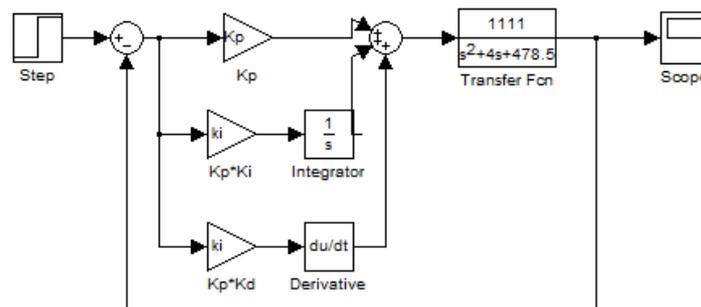


Figure 3-15 PID controller model

The behaviour of it is fairly complicate and it is not simply add the three terms together but combine the advantages of the three terms and make inner connect with them. After the tedious adjustment and compromise, the follow diagram shows the final result, it is probably not the best one but after the numerous attempts, it is the best one that has been found so far.

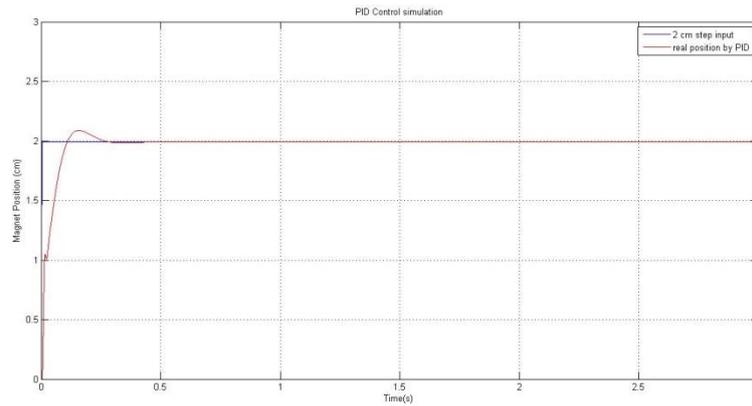


Figure 3-16 Best initial PID step response

During the design of the PID parameter, it is notable that tuning the PID parameter manually is not the best way to do since it is so time-consuming and difficult. But sometime it is the only way to identify the parameters. Also, in real world application, the initial guess and manual tuning is more common [2].

Before the parameters are applied to real maglev plant, it is simulated using Simulink package firstly to make sure that the maglev plant always works in a safe condition. The wrong input parameters and variables may cause the maglev plant to overload the coil or operate at a critical unstable condition there by damaging the coils or the magnets. This situation is not allowed to happen throughout the project. With the help of Simulink package, a more accurate and precise version of maglev plant can be simulated. As mentioned before, the nature characters of the maglev is not linear which means the maglev plant could not be simulated at 100% of accuracy. Hence, it is necessary to make some assumptions. Firstly, the plant parameters were be linearize at the desired position (2 cm). The error occurs when the magnet moves away from the desired point the accuracy of the linear model will decrease while the magnet position is close to the desired point, it linear model is comparably accurate. The simulation may not represent the physical behaviours precisely but it gives a basic idea of what the real system performance.

The maglev plant with PID controller model shows as figure 3-17, it can be noticed that two switches are part of the model trying to avoid the damaging due to the wrong variables and parameters. The input current (control effort) is limited in ± 2 amps.

Any value beyond it is a safety hazard. The manual state that the coil excitation durations for different current values. When the excitation current is 1 Amps, the excitation duration is indefinite. However, for 4 Amps excitation current, the duration is only 20 seconds. Hence, there is a need to limit the excitation current in a suitable value thereby protecting the plant from being damaged.

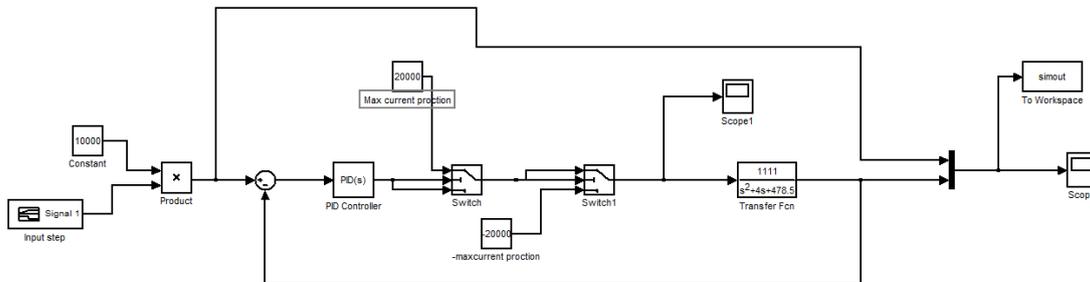


Figure 3-17 Full detailed PID Simulink model

Results for the simulation was plotted by using MATLAB is as following,

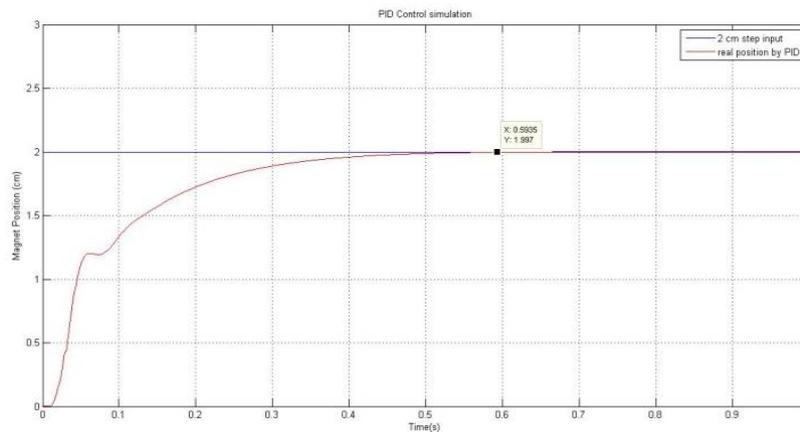


Figure 3-18 best simulation result for a PID controller at 2 cm linearization point

A PID controller achieves a desire response by calculating the error between a process variable and a desired position and attempts to minimize the error by adjusting the process input. It takes time to eliminate the error gradually until the steady state error is zero. There is always a compromise between settling time and overshoot. The following figure shows the step response for different inputs.

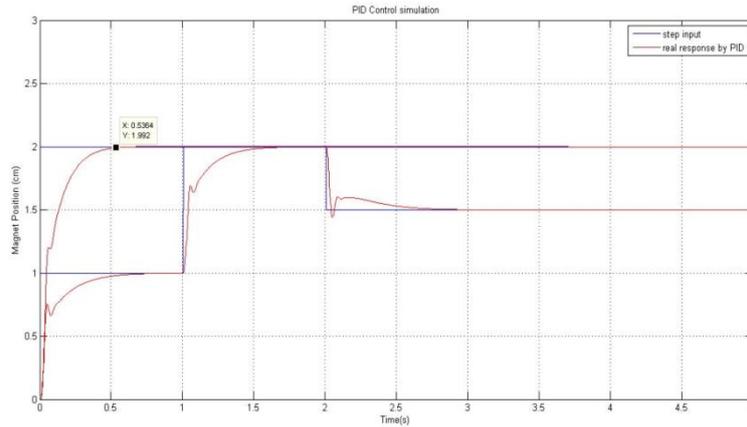


Figure 3-19 effect for moving away from linearization point

It can be seen that for a unit input is at the operation point (2cm), it gives better step response in terms of settling time and overshoot. However, when the step input moves away from the operation point, the system performance becomes worse. It is fairly obvious in the first second part. For a step input is 2 cm, the settling time for it is 0.6seconds while for a step input is 1cm, the steeling time becomes approximately 0.8 seconds. The settling time is longer and overshoot occurs. After the simulation, the PID control algorithms could be applied on the real plant.

The maglev plant is implemented using computer and digital signal process (DSP) chip, which is specially designed to carry out computation related to control algorithm realizations therefore a digital PID control law should be applied. As mentioned in the continues PID controller design,

- P – Proportional

Continues: $c_p(t) = K_p e(t)$

Discrete: $D_p(n) = K_p e(n)$

- I – Integral

Continues: $c_i(t) = K_i \int e(t) dt$

Discrete: $D_i(n) = K_i \sum_{j=0}^{n-1} e_j T = K_i T \sum_{j=0}^{n-1} e_j = K_i T (e_0 + e_1 + \dots + e_{n-1})$

- D – Derivative

Continues: $c_d(t) = K_d \frac{d e(t)}{d t}$

Discrete: $D_d(n) = K_d \frac{e(t_n) - e(t_{n-1})}{t_n - t_{n-1}} = K_d \frac{e_n - e_{n-1}}{T}$

Hence, in the discrete time, the digital control is defined as following,

$$D_{pid}(n) = K_p e(n) + K_i T (e_0 + e_1 + \dots + e_{n-1}) + K_d \frac{e_n - e_{n-1}}{T}$$

This equation would be applied for the following C language programming.

3.4 PID Controller Implementations

After the simulation part for a PID controller has done, it is the time to move on to the next part which is to implement the control algorithm on the real plant to produce the result as close to the simulated results as possible. As stated in Chapter 2, the Maglev machine has a digital signal processor (DSP) on-board and the DSP belongs to the M56000 processor family which has the capability to executing control laws at high sample rates making the implementation much easier. The control algorithms could be written into the DSP chip directly by the ECP Model 730 software package. The software allows users to create their own control algorithms, compile them and implement them via the DSP based controller. The control algorithms are coded into 'C – like' language which is totally different from MATLAB and SIMULINK because it has only the basic function such as comparators, conditional statements and basic calculations and for advanced calculations for example, derivative and integral cannot be applied in the ECP-written programming software. Also, matrixes calculations could not be done at once. In order to implement the PID control law, the 'C – like' program language should be fully understood.

According to the user manual, the user written control algorithm code is made of three different sections: the definition section, the variable initialization section and the servo loop or real time execution section. Once the ECPUSER program writes the

algorithm to the real-time controller, it uses the definition section to assign internal variables from q1 to q100 to the user predefined variables. In other words, all the variables which will be used in the following codes should be predefined in the definition section. By doing this, a huge amount of computation time could be saved since it is a real time execution. The variable initialization part is to be used to allocate some initial values and constant prior before being running inside the servo loop. The servo loop code starts with a 'begin' statement and end with an 'end' statement. All the code inside the servo loop will be executed every sample period. It means between the 'begin' and 'end' statements, the code will be executed automatically without using a 'for' loop. But there is an issue that the execution time for each sample period should not bigger than the sample period. The minimum sampling period is 0.000884 seconds which means the maximum sample frequency for the DPS controller is 1.1KHZ. The servo period could be an integral number multiply by the minimum sampling period 0.000884 seconds.

When moving from continuous control theory to discrete control theory, one of the most important concepts is Shannon's sampling theorem. The sampling theorem states if a signal is sampled twice faster than its highest frequency component, the sampled data can represent all the features of the signal. Hence, an assumption needs to be made that the sampling frequency is at least two times greater than the highest frequency component therefore, aliasing will not occur during the analog to digital conversion. In other word, the sampled signal is almost as same as the original signal. Then the PID control law can be edited in the setup control algorithm window. The following table shows all the variables and constants which are used to in the C language programming.

Variables/Constants	Explanations
$T_s=0.001768$	Sample period since its sample frequency is 1.1KHZ
$u_{10}=11800$	Weight of the bottom magnetic disk $mg=0.120*9.822*10000$ in N/10000

$y1o=25000$	Desired output
$Kp1, Kd1, Ki1$	PID parameters found in continues time
$kdd1=kd1/Ts$ $kid1=ki1*Ts$	discrete derivative term discrete integral term
$y1str=sensor1_pos-y1o$	Using calibrated sensor
$error=cmd1_pos-y1str$	The error between desired output and real output
$delta_y1=y1str-pos_last$	Difference between the error and the previous error $e_n - e_{n-1}$
$ui=kid1*error+ui_last$	$K_d \frac{e_n - e_{n-1}}{T}$
$u1str=kp1*error+kdd1*delta_y1+ui$	PID control law: $K_p e(n) + K_i T (e_0 + e_1 + \dots + e_{n-1}) + K_d \frac{e_n - e_{n-1}}{T}$

Table 5

The detailed C language codes can be found in Appendix. The ECPUSER program provides a function to generate figures for each plot however it is not satisfied for the comparison of two different plant responses. For the purpose of better observation, another function of ECIYSER program is used to export the data into txt files and these txt files are read by MATLAB to plot the response. The following three graphs showed the step responses for different inputs.

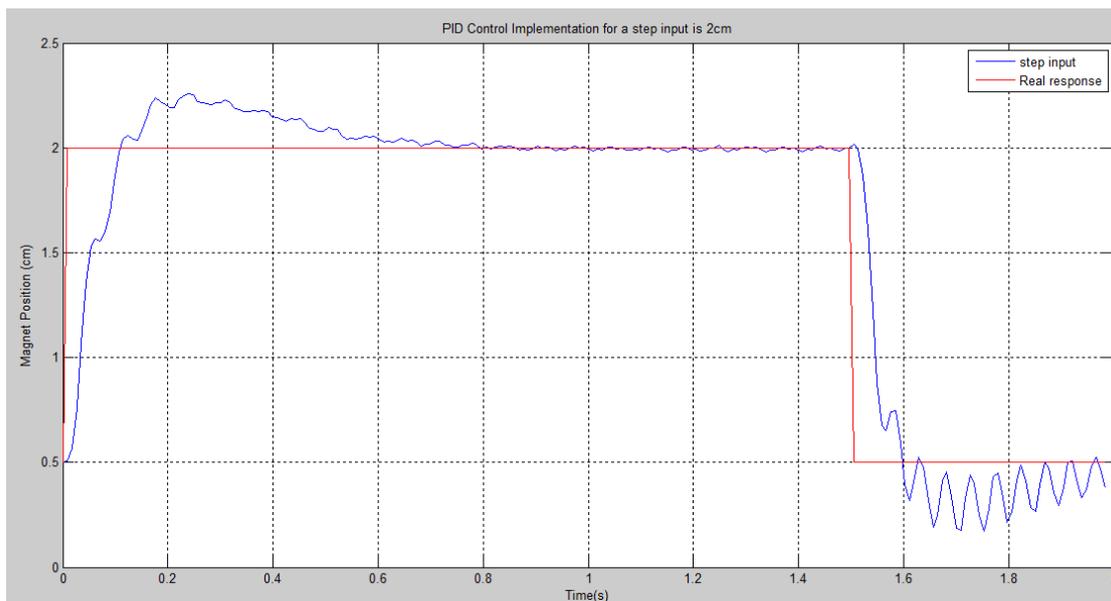


Figure 3-20 Implementation result of a 2 cm step response

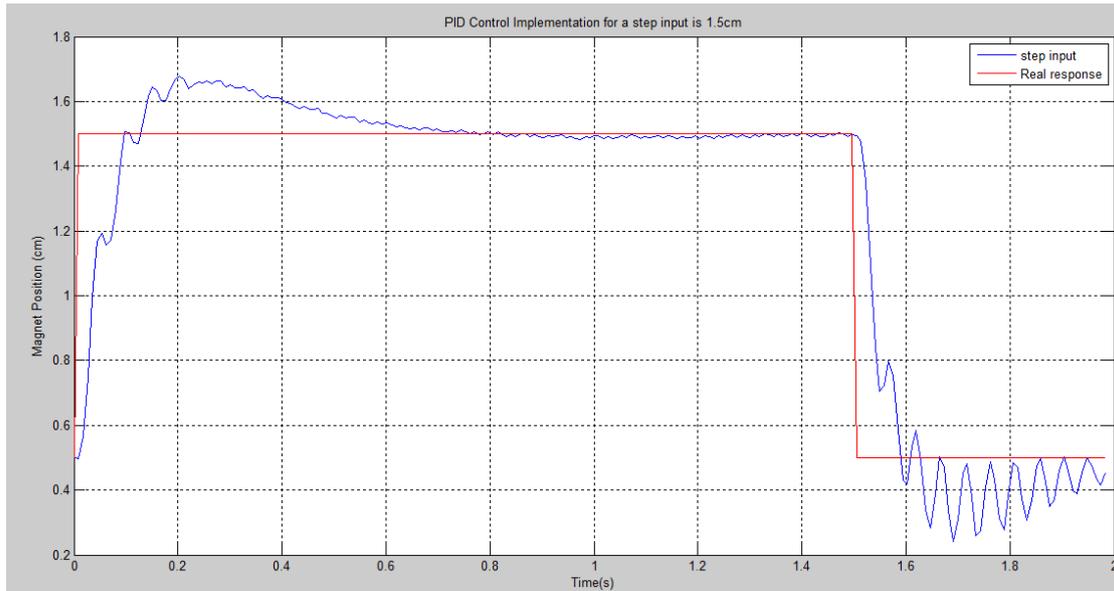


Figure 3-21 Implementation result of a 1.5 cm step response

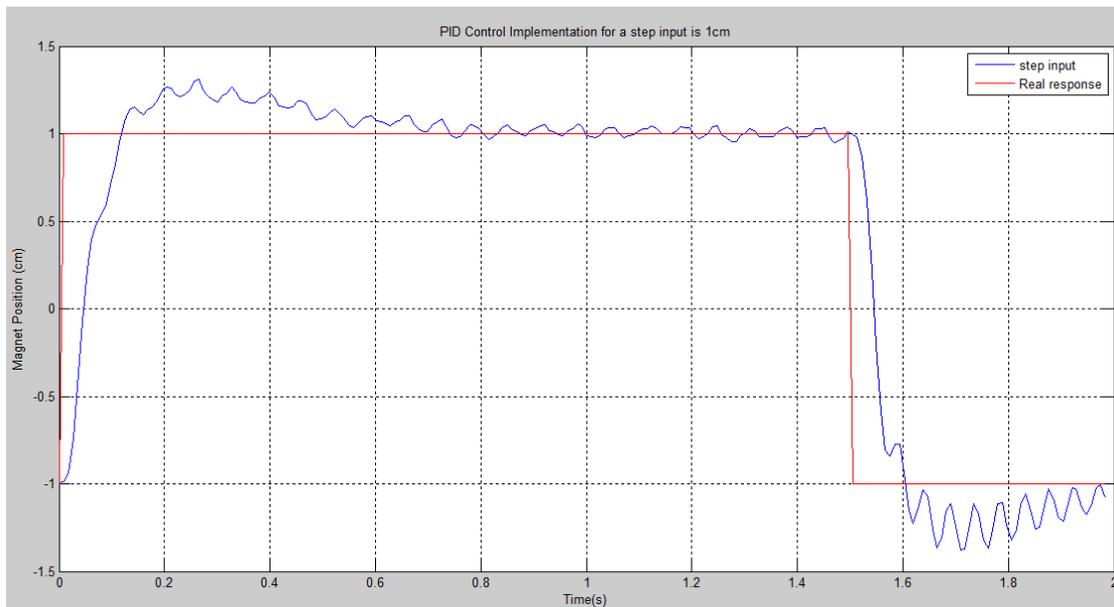


Figure 3-22 Implementation result of a 1 cm step response

By comparing these three figures, it can be seen easily that when step input is 2 cm, the step response of the Maglev plant is the best. As the step input is 1.5 cm, the step response does not look much different from the step response for a 2cm step input. However, when the step input becomes 1cm, the step response is the worst since it contains huge amount of oscillations and overshoot. The reason for that is predictable because all the PID parameters were selected based on the system modelling at an

operation point which is 2 cm. When the bottom magnetic disk moves away from the operation point, the accuracy of the system modelling will decrease. It becomes obviously that if a system modelling is not accurate, no matter how good the three parameters for PID controller were chosen, it would not give the perfect response. A conclusion can be drawn that for a small difference between assumed operation point and real operation point, such as 0.5 cm difference, it seems not affect the step response however, for a huge difference which equals or more than 1 cm, the step response becomes worse in terms of system performances. The settling time difference between the simulation and the real implement results is about 0.2 seconds which is acceptable. Also, it can be seen from the figures that even when the maglev plant was working at the linearization point, there are still some small oscillations in steady state. However, by look at the maglev machine visually, there are no oscillations at all. During the further investigation, it was found that the oscillations were coming for the position sensor itself. In Adam Clerk's thesis [10], he pointed out that the position sensor noise appeared to be a repeating sinusoidal function at 195 Hz which is close to a fourth harmonic of the 50Hz mains power supply 200Hz. Hence, it is believed that the interference is from both Maglev power supply and computer power supply coupled with Maglev circuits. Since the noise comes from the hardware and there is no way to improve it by modifying the hardware, low pass filter design techniques could be used to improve the response the step response in terms of eliminations of small oscillation. Once again, the sensor noise does not affect real step responses and the only thing it affects is the sensor position data it collected. Hence, it is not a big deal based on the fact that the actually plant works smoothly in steady state.

As a result, the characteristics of a PID controller are not good enough to such a maglev plant which requires a quicker response and almost no overshoot. Also, for a PID controller, it is not possible to set up a desired time during controller design and all the response are based on adjusting three parameters to see which combination of three variables can give a good response. However, a deadbeat control makes it possible to design the controller which can give the desired response in a predefined time. A deadbeat controller should be introduced to handle the maglev system which can give a much quicker response and no overshoot. A 'deadbeat' response is defined

as ‘controlling a system to the desired position in the shortest time’ [2]. By applying the deadbeat response concept into maglev system, a better response could be achieved. Also, the robustness character of a controller is necessary. According to Stenel,RF and Ray, LR, a robust control is defined as the ability to maintain stability or performance characteristic in the presence of all predefined system parameter variations[11].

Chapter 4

Robust Deadbeat Controller Design, optimization and Implementation

4.1 Introduction

Robust deadbeat control is defined as a control algorithm which can achieve both robust and deadbeat response where deadbeat response is defined as ‘having zero steady-state error, minimum settling time (T_s) at 90% rise time (T_{rm}), and a percent overshoot (P.O.) and percent undershoot (P.U.) less than 2%’ [11]. This control method was originally proposed by Dorf et al. in 1994 [12]. However, this technique works only on lower-order plants. As a result, there is a need for higher gain when applying on higher-order systems due to the cancellation between poles and zeros in the close loop function. More variable gain is able to independently specify the desired overshoot and settling time. This design with a proper high gain will result in systems that are insensitive to plant parameter variations of up to 50% [12]. When referring to the maglev system, robustness is required to immune the nonlinear characteristics of the magnet and external disturbances while deadbeat control is required to achieve better response [12]. Figure 4-1 shows the basic diagram of a robust deadbeat control system and Table 6 is the look up table for controller design.

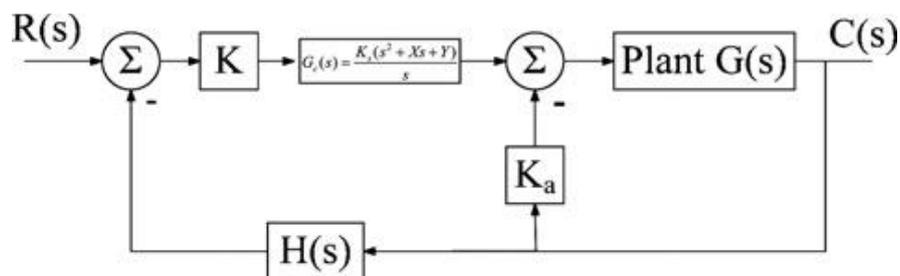


Figure 4-1 robust deadbeat controller block diagram

Order (n_p)	α	β	γ	δ	T_{r90}	T_s
2nd	1.82				3.47	4.82
3rd	1.90	2.20			3.48	4.04
4th	2.20	3.50	2.80		4.16	4.81
5th	2.70	4.90	5.40	3.40	4.48	5.43

Table 6 robust deadbeat controller constants look up table

4.2 Robust Deadbeat Controller design

According to Dorf et al[12], the design procedure is seven steps as following,

- 1) Use a PID controller as $G_c(s)$.
- 2) Add a cascade gain K before the PID controller.
- 3) Add a state variable feedback gain K_a . This will make the system over specified by at least one variable.
- 4) Determine n_p for $G_cG(s)$, where n_p equal's the number of poles in $G_cG(s)$.
- 5) Add the feedback:
 - $H(s) = 1$ for $n_p=2$.
 - $H(s) = 1 + K_b s$ for $n_p=3$ or 4.
 - $H(s) = 1 + K_b s + K_c s^2$ for $n_p=5$.
- 6) Select gains, using the coefficients from Table 1, to achieve deadbeat step response with the following requirements:
 - a) Set $K=1$.
 - b) Set $\omega_n = Ts' / (80\% \text{ of the desired settling time})$.
 - c) Set the characteristic equation of the closed loop transfer function equal to:
$$s^{n_p} + \alpha \omega_n s^{n_p-1} + \beta \omega_n^2 s^{n_p-2} + \dots + \omega_n^{n_p}$$
 - d) The roots of H(s) must be real and negative.
 - e) The smallest root of H(s) will set the desired settling time by the relationship: $\{4 / (\text{smallest root})\} = \text{desired settling time}$.
 - f) If it is not possible to satisfy 6d and 6e, then adjust the 80% in 6b to a lower value. Lower percentages will also result in higher values for K_a K_b s, X and Y. Higher percentages will decrease these values.

- 7) Increase K until the response becomes deadbeat and the settling time is approximately equal to the desired value.

As applying the above procedure, the system block diagram looks as following,

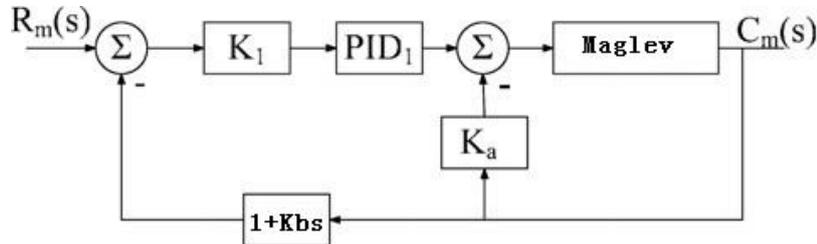


Figure 4-2 Detailed robust deadbeat controller block diagram

The model of maglev plant itself is a second order system hence the number of poles in $G_cG(s)$ is three. The coefficients of the equation α, β, T_{r90} and T_s are selected from Table 6 as following,

$$\alpha = 1.9, \beta = 2.2, T_{r90} = 3.48 \text{ and } T_s = 4.04.$$

And since $n_p = 3$, the characteristic equation is

$$\text{C.E} = s^3 + \alpha\omega_n s^2 + \beta\omega_n^2 s + \omega_n^3$$

For the initial design, if **desired settling time is chosen as 0.4** seconds, then,

$$\begin{aligned} \omega_n &= \frac{T_s'}{80\% \text{ of the desired settling time } T_s} \\ &= \frac{4.04}{0.8 \times 0.4} = 12.6250 \end{aligned}$$

Then substitute all these values α, β and ω_n into the characteristic equation,

$$\begin{aligned} \text{C.E} &= s^3 + \alpha\omega_n s^2 + \beta\omega_n^2 s + \omega_n^3 \\ &= s^3 + 23.9875s^2 + 350.6594s + 2012.30664 \end{aligned}$$

The closed loop function of the control block diagram is,

$$\frac{C(s)}{R(s)} = \frac{KG_c(s)G_p(s)}{1 + G_p(s)K_a + KG_c(s)G_p(s) \times (1 + K_b s)}$$

Where $G_c(s) = \frac{K_3(s^2 + Xs + Y)}{s}$, which is basically a PID controller.

$G_p = \frac{1111}{s^2 + 4s + 478.5}$, which is a linearized transfer function for maglev plant at 2cm operation point.

By substitute two transfer functions into the closed loop function of the whole robust control system, we obtained,

$$\frac{C(s)}{R(s)} = \frac{1111KK_3(S^2 + XS + Y)}{S^3 + \frac{4 + 1111KK_3 + 1111KK_3XK_b}{1 + 1111KK_3K_b} S^2 + \frac{1111K_a + 478.5 + 1111KK_3X + 1111KK_3XK_bY}{1 + 1111KK_3K_b} S + \frac{1111KK_3Y}{1 + 1111KK_3K_b}}$$

Note: K_b is defined $\{4/(\text{Root of } H(s))\}$ =desired settling time, $H(s) = 1 + K_b s$, so the root of $H(s)$ is $\frac{1}{K_b}$, it becomes $\frac{4}{\frac{1}{K_b}} = 0.4$, therefore, the value of $K_b = 0.1$. By comparing the denominator with the system characteristic equation, following equations could be obtained,

$$\frac{4 + 1111KK_3 + 1111KK_3XK_b}{1 + 1111KK_3K_b} = 23.9875$$

$$\frac{1111K_a + 478.5 + 1111KK_3X + 1111KK_3XK_bY}{1 + 1111KK_3K_b} = 350.6594$$

$$\frac{1111KK_3Y}{1 + 1111KK_3K_b} = 2012.30664$$

As $K=1$ is set up and K_b is calculated as 0.1. It is obviously that the other four variables X , Y , K_3 and K_a could not be solved by only three equations. However, according to Dorf et al [12], K and K_3 could be chosen arbitrarily. For the sake of convenient calculations, $K \cdot K_3$ is written as K_c . At first, all the valuables were calculated manually as following, $K=1$, $K_3 = 0.01$, $K_b = 0.1$ and then substitute these three equation into the three equations above to work out the values of K_a , K_b , X and Y .

4.3 Robust Deadbeat Controller Simulation and Optimization

For the sake of convenience, a MATLAB script was written to do the tedious calculations for all for these parameters then substitute them into the SIMULINK model that has been built and in the end plot the step response for a 2cm unit step at the linearization point. All the output was sent back to MATLAB workspace which allows the plotting of initial simulation results data and working out the settling time by the data cursor. The script is as following and the Simulink model will be included in the Appendix.

```

alpha=1.90; beta=2.20; Tsetd=4.04; Tr90=3.48;% from look up table
for Tdesired=0.4; %desired time
    omega=Tsetd/(0.9*Tdesired); %omega
    eq1=alpha*omega; %coefficients for s2
    eq2= beta*(omega^2); %coefficients for s1
    eq3=omega^3; %coefficients for s0
    for Kb=Tdesired/4 %Kb
        for K=1 % set K=1
            for K3=0.01 % set K3
                X=((eq1*(1111*K3*K*Kb + 1))-4-1111*K*K3)/(1111*K*K3*Kb) %Calculate X
                Y=eq3*(1111*K*Kb*K3+1)/(1111*K*K3) %Calculate Y
                Ka=(eq2*(1111*K*K3*Kb+1)-478.5-1111*K*K3*Kb*Y-1111*K*K3*X)/(1111) %Calculate Ka
            end
        end
    end
end
end
sim('deadbeatfail') %call simulation
t=linspace(0,1,length(simout(:,2)));
plot(t,simout(:,1)/10000);
hold on
plot(t,simout(:,2)/10000,'r');
xlabel('Time(s)')
ylabel('Magnet Position (cm)')
title('Initial Deadbeat Control Simulation')
title(['Initial Deadbeat Control Simulation when desired time=' ,num2str(Tdesired)], 'FontSize',15)
legend('step input','response for Deadbeat controller')
grid on

```

Figure 4-3 MATLAB script for deadbeat parameter optimization

The plotting initial result shows below,



Figure 4-4 Initial simulation result

As Figure 4-4 shows that the desired time was set up to 0.4 seconds but the real response is about 0.7 seconds which is somehow worse than the classical PID controller when talking about settling time. Dorf et al pointed out in his thesis that if the response is not satisfied, the 80% of the desired settling time could be reduced to a lower value to achieve a better performance. Figure 4-5 indicated the different response when different percentage of the desired settling time was chose.

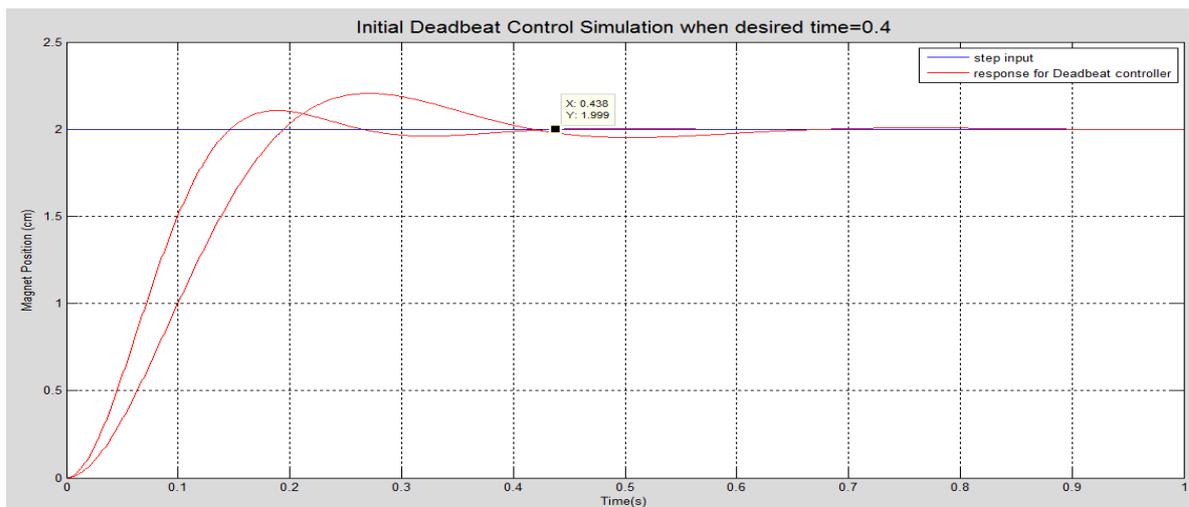


Figure 4-5 impacts of different desired time tolerance

	Test1	Test2
Desired Time	0.4 seconds	0.4 seconds
percentage	90%	60%
K	1	1
K₃	0.01	0.01

X	26.9138	59.3251
Y	268.5411	1.5661e+003
K_a	-0.4419	-0.8844
K_b	0.1	0.1
Settling Time	0.741 seconds	0.438 seconds

Table 7

Table 7 shows the other parameters will change when the reduce percentage of the desired time. By looking at Figure 4-5 and Table 6, it is noticeable that by reducing the percentage of the desired time, the system performance improved dramatically in terms of settling time and overshoot but it requires a considerable integrator gain(Y) to boost the system performance. This is one of the issues which need to be taken into the consideration during the implementation. Overall, after reducing percentage of the desired time, the system response is much better comparing to the classical PID controller because settling time and overshoot are reduce by 0.2 seconds and 15% respectively. Figure 4-6, 4-7, 4-8 and Table 8 demonstrate the response and parameters when desired time is chosen as 0.1 seconds, 0.2 seconds and 0.4 seconds respectively.

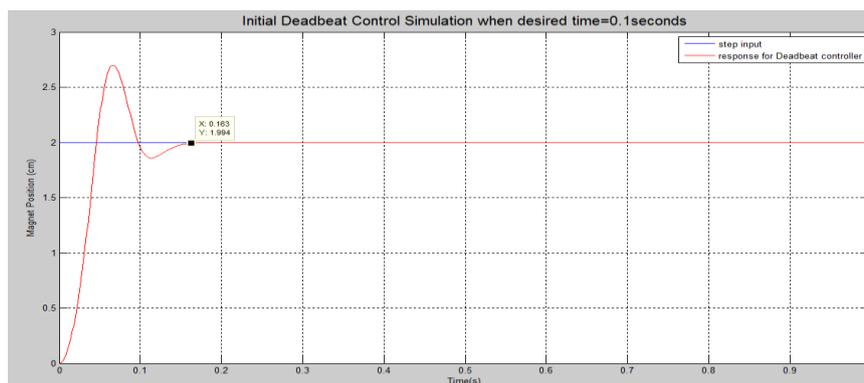


Figure 4-6 system response when the desired time is 0.1 second

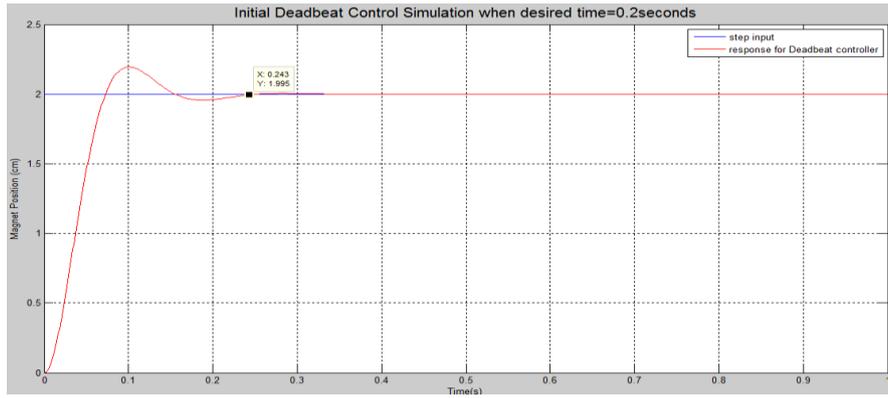


Figure 4-7 system response when the desired time is 0.2 seconds

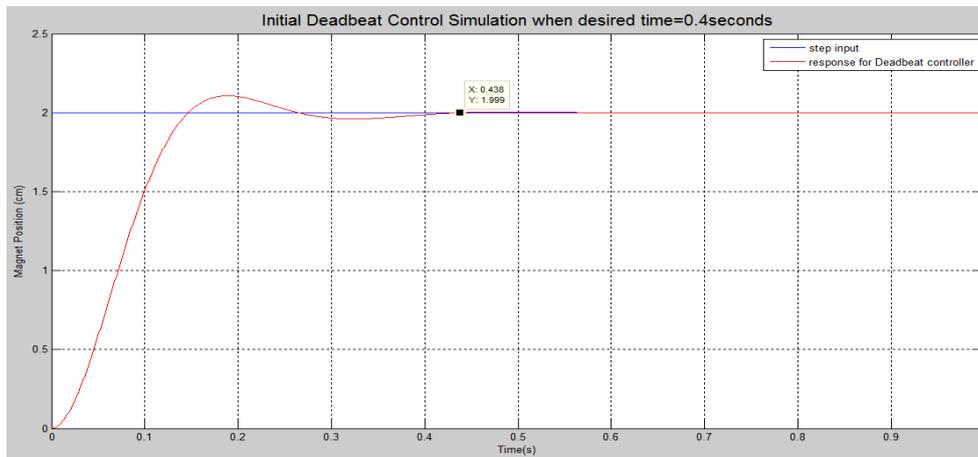


Figure 4-8 system response when the desired time is 0.4 seconds

Test Numb	Test 1	Test2	Test 3
Desired Time	0.1 seconds	0.2 seconds	0.4 seconds
percentage	60%	60%	60%
K	1	1	1
K_3	1	0.0001	0.01
X	92.3954	1.0839e+004	59.3251
Y	7.9066e+003	3.4538e+005	1.5661e+003
K_a	-32.1564	-0.9846	-0.8844

K_b	0.025	0.05	0.1
Settling Time	0.163 seconds	0.245 seconds	0.438 seconds

Table 8

It should be pointed out that K_3 and K values should be carefully chosen during the optimization because incorrect or unsuitable K_3 and K values would cause the instabilities of the system which is not a preferred outcome. Also, by looking at the three figures, it is obviously that the less desired time required, the more overshoot will occur. Accordingly, more overshoot means more current (control effort) is going through the coil. Hence, there is a need to observe the current to make sure that the current is in a certain limit. Also, from the system response figure above, the system response is not an idea deadbeat response as Figure 4-9 (desired time = 0.2) shows because there are still some oscillations and overshoot.

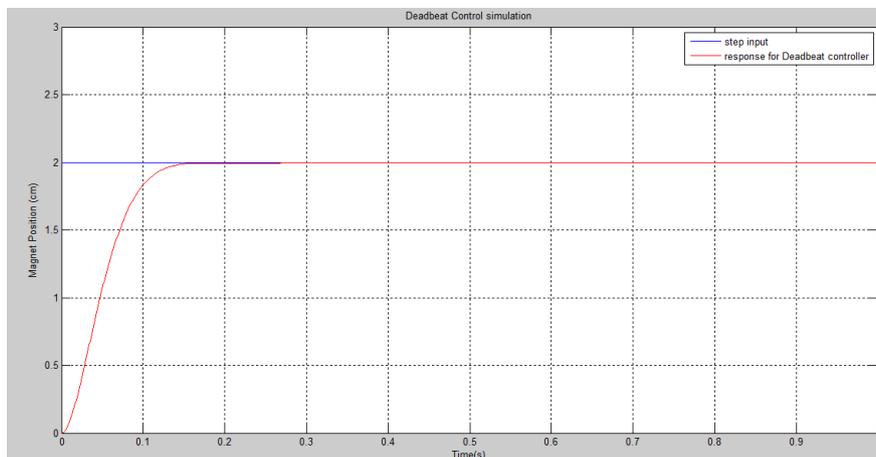


Figure 4-9 idea deadbeat response

However by changing the K and K_3 values slowly based on the observation of the system response, it is possible to get the system response which is very close or even almost same as the ideal deadbeat response as the following figure shows,

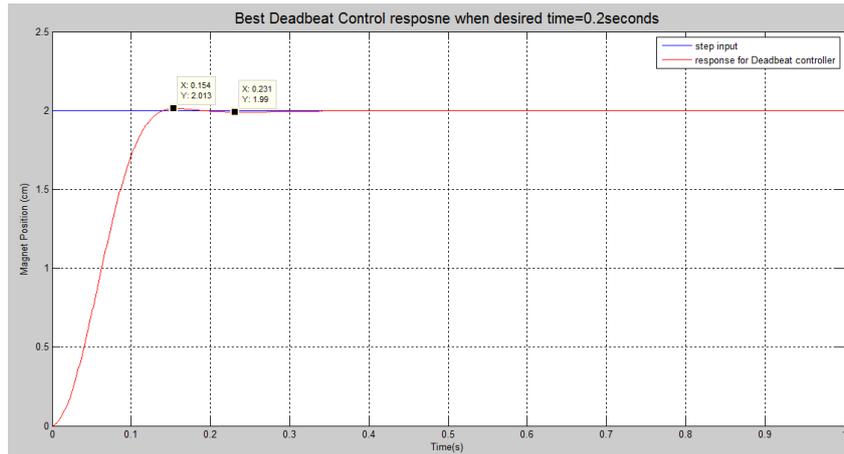


Figure 4-10 the best deadbeat response achieved by optimization

The overshoot of the system step response is 0.013 (less than 0.0065%) when a step input is 2cm which can be almost neglected. The settling time for the system response is 0.231 seconds and it has only 0.031 seconds' difference. The design and simulation of a robust deadbeat controller at the linearization point has achieved based on the simulation results. In comparison with PID controller, the robust deadbeat controller I designed gives much better step response and enables to achieve almost zero overshoot and quick response time. The next part is to simulate the system response when the input of the system goes beyond the linearization point.

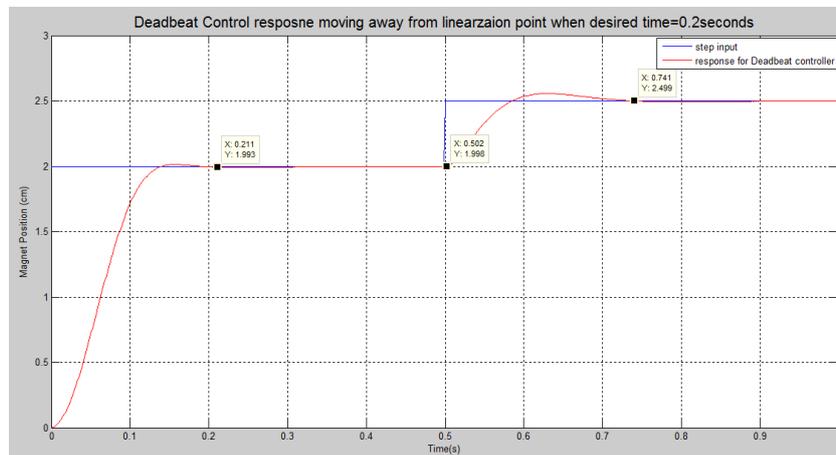


Figure 4-11 moving away 0.5 cm away from the linearization point

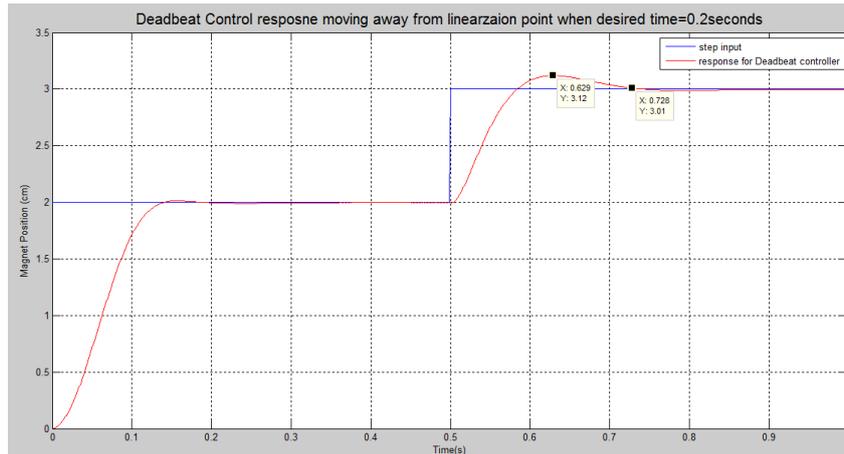


Figure 4-12 moving away 1 cm away from the linearization point

As the above shows that magnetic disk was moving from 2 cm to 2.5 cm, the overshoot becomes much bigger and the settling time is longer. When magnetic disk was moving from 2 cm to 3 cm, the overshoot and settling time is even worse. The outcome is predictable, same as the classical PID the step response becomes worse as the magnetic desk moves away from linearization point but the robust deadbeat controller still performance better in terms of resist the sudden change. By this point, all the simulation and optimization for a robust deadbeat controller has done and the result is pretty much ideal. The last thing is to implement the control law on the real plant to see if the real implement results are as good as the simulation result.

4.4 Robust Deadbeat Controller Implementation

The implementation for a robust deadbeat control law is a challenge, unlike a PID control algorithms, robust deadbeat control law is much more complicated because all the Simulink blocks needs to be convert to C Code which can be executed by the Maglev machine and the programming functions are limited. The Simulink model is as following,

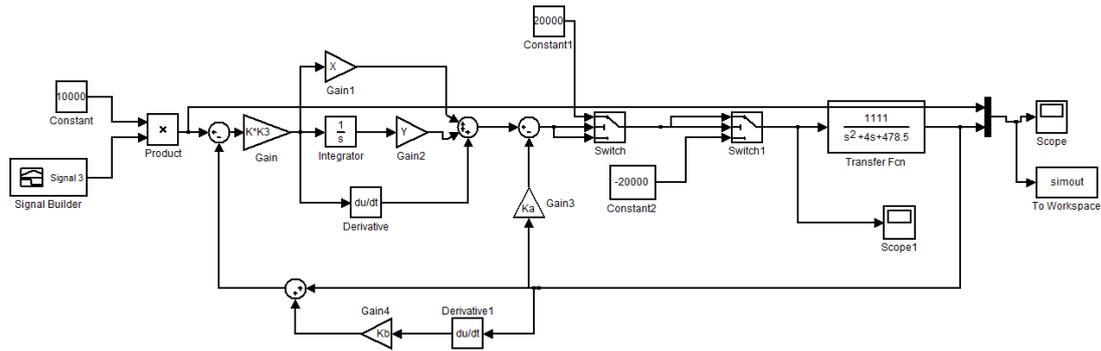


Figure 4-13 robust deadbeat controller Simulink model

The robust deadbeat Simulink model is consist of three part, normal PID controller block, state variable block and feedback zeros block. Also, a protection block was added to make sure the current go through the coil is not more than 2.5 amps otherwise it would damage the maglev plant. Although there is a protect circuit built in the control box, it is still necessary to limit the input current. The coding details are showed in the following table,

Blocks or values:	Expression:	Explanations:
H(s)	$H = Kb \frac{posn - posnl}{Ts} + posn$	H(s) = 1+KbS, S is basically the change rate in time domine.
State steady error	Error= desiredpos-H	State steady error
P	P = X*error	Proportional to error
I	$I = \frac{error+lasterror}{2} * Ts * Y + int$	Summation of error
D	$D = \frac{error+lasterror}{Ts}$	Derivative of error
Output:	Out= (P + I + D)KK3- Ka*posn	Output of the plant

As mentioned before in PID controller implementation, the system should be changed into discrete system during the development of C code. Also, after the servo loop, the current value will become the previous value automatically and not 'for' loops are required. The code is provided in Appendix.

The initial implementation of the PID- based robust deadbeat control algorithm on the real plant by using the optimised values failed. Once the control law was implemented, the maglev disk started oscillation and vibrating which makes the whole maglev system extremely unstable. The first reaction was to check if there are some errors in my coding. After plot the export data from Maglev plant and comparing them with each point from the Simulink model using scopes, it turned out that the real raw data was very close to the Simulink model. Then, the parameters of the robust deadbeat controller were checked, and it was found that during the simulation, the higher K and K3 were chosen to improve the system step response so that an idea deadbeat response was achieve. However, in the real practical scenario, high K and K3 would cause huge increments of Y and X parameters which leads to some issues such as the overloading of the maglev plant, system oscillations and instability. The solution to tackle such a problem is to reduce the system performance to a certain level at which the maglev plant can actually handle. Further experiments were conducted to figure out at which boundary the maglev system could work much more stable. Based on the test results, it showed that when the multiplying of K and K3 is small than 0.0004, the system may work in a fairly stable condition. In other words, the simulated idea response time and overshoots cannot be achieved when the PID-based robust deadbeat control law was applying on the real plant. After a few further tests, it was found that when the multiplying of K and K3 were limited to 0.0004, the real settling time for the maglev plant become longer than the desired time and the overall output of the maglev system was more stable. However, it seems impossible to eliminate the output oscillation totally no matter how small K and K3 are chosen and reducing the values of K and K3 can only reduce the amount of vibrations and oscillations to a certain level. It deems that the oscillations are coming from the hardware of the maglev plant itself although the oscillations become very tiny but they still exist.

Also, the initial implantation results showed that the steady state was always 3 cm higher than the designed displacement which is 2 cm for different increment of movements. Based on further investigation, it was found that the position sensor itself has a huge offset. When the magnetic disk is at 0 cm the reading of position sensor is 30000 (3 cm) and that is where the offset comes from. The final output should be compensated by minus the offset in the C code to get rid of the huge offset error.

There may be other issues for the oscillations but the overall results are acceptable. The following figures showed the magnetic response.

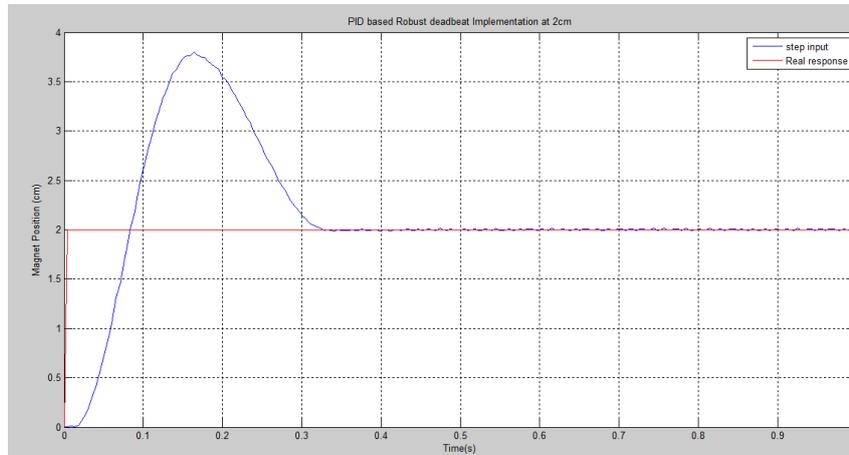


Figure 4-14 implementation of a deadbeat controller for a 2cm step response

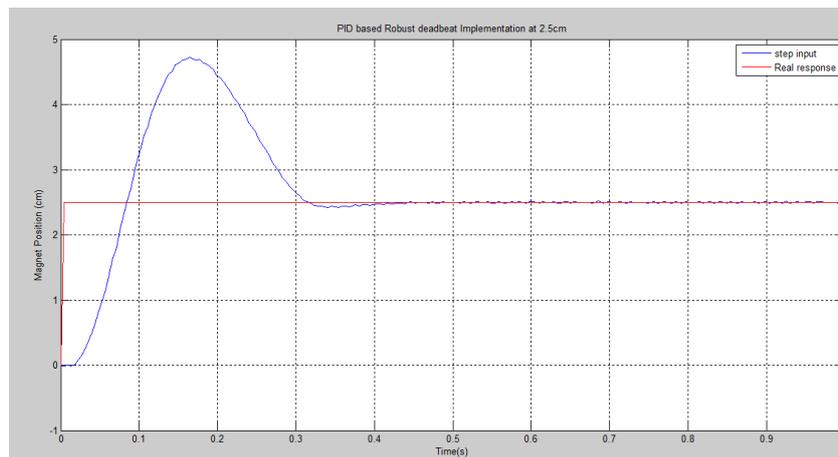


Figure 4-15 implementation of a deadbeat controller for a 2.5cm step response

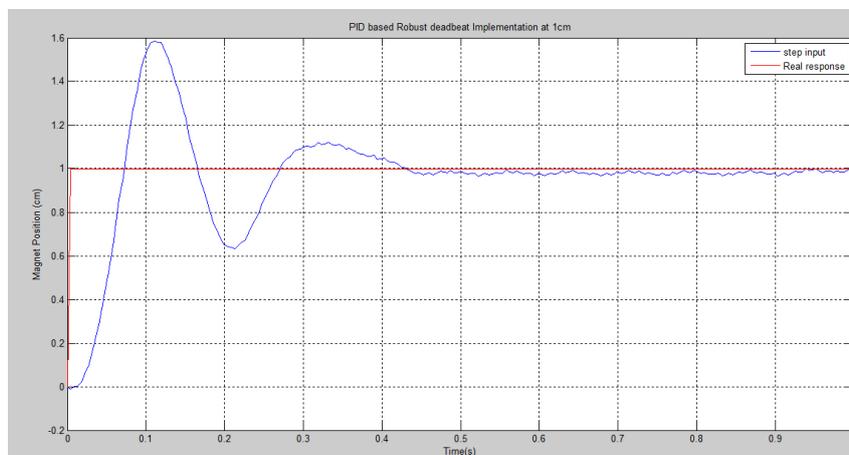


Figure 4-16 implementation of a deadbeat for a 1 cm step response

It can be seen from the figures that the impact of moving away from the linearization point of 2cm was not as huge as classical PID controllers and since the desired time for deadbeat control laws can be designed beforehand, the settling time has been improved at least 55% at the linearization point. Also, for a small movement, the system response looks not too bad considering that there is only 0.1 second increasing in settling time which tells that the PID based deadbeat controller work well for a movement of 0.5 cm. However, a movement of 1 cm from the linearization point lead to some oscillations although the oscillations disappeared after 0.5 seconds and the settling time becomes much longer. Further movements away from the linearization point would cause serious oscillations problems which may damage the maglev plant so the further test for an even larger displacement has not been conducted. Comparing with a normal PID controller, the PID based robust deadbeat controller has excellent performance in terms of overshoot and settling time but when it comes up to a huge movement away from the linearization point, frankly speaking, there is nothing improved at all. It was also found that theoretically, a perfect deadbeat response could be achieved by the PID based robust controller but in practical, an idea deadbeat response requires an extremely huge gain which makes it impossible to be achieved on that maglev plant due to its own limitations.

Chapter 5

Conclusions and Further Work

5.1 Conclusions

In this project, a normal PID control and a PID-based deadbeat controller were designed, simulated and finally applied on the real Maglev plant. Throughout the project, all the design process was showed step by step and the results and performances for each control law were evaluated at the end. The project also showed that there had always been issues when applying the theoretical simulation results into a practical plant directly but the issues were solved in the end.

The project was started with literature review. The purpose of the literature review is to obtain some background information about the existing controller design techniques, maglev trains, nonlinear control methods and ECP Model 730 maglev plant. Based on the literature review, it was found that the information and document about Model 730 maglev plant was so limited and there was only one paper about adaptive control with neural network and in that paper the author implemented the algorithm on the Model 730 plant. Also, some of papers pointed out that 90% of the process controls are still using classic PID to compensate the plant. Hence, a decision was made to design a normal PID controller first. There are two reasons for it, firstly, PID control algorithm is fairly easy to handle and it is a good start point. Also, the PID control can be used as a reference to compare with the deadbeat controller which was designed later. In one of the latest transition paper, a generalized proportional integral controller combined with fast online close-loop identification was applied on the magnetic levitation system and the results based on experiment were excellent for both stabilization and trajectory tracking tasks. The research direction and area is very similar to the work I was doing and the only difference is that we used different maglev platforms. However, in this paper, a huge amount of high level mathematics

or algorithms were in the context and some of them were far more beyond the understanding as a bachelor student.

After the literature reviews, the maglev plant was investigated. The manual was provided by ECP and all of the characters and details about the maglev plant dynamics were included. In this manual, the 'C like' programming language was explained in details along with several experiment demos which were very helpful for the further implantation. The experiment demo was studied and tested on the plant thereby having a better understanding of both the programming language and the physical maglev plant. After read through the manual, an approximate system modelling was obtained. The maglev plant was modelled as a second order system and two parameters were identified through experiments. It should be noticed that some of the impacts were negated on the purposed of reducing the complexities of the project.

Further research found that the system was not linear in wide operation range. To tackle the problem, a linear approximation was required. There are several linearization methods were available to use such as Euler's method, finite difference method, Newton's method and Taylor theorem. Taylor theorem turned out was the easiest one to be applied. By applying Taylor theorem, a system transfer function could be found at a linearization point and the linearization point was chosen when the magnet force was exactly equal to the weight of the bottom magnet disk. However, for such a linearized transfer function, once the operation point moves away from the linearization point, a linearization error occurs. In other word, the system transfer could not represent the system anymore when there is a movement away from the linearization point. This problem occurred in both simulation and implementation and the only way to deal with it is to use the adaptive control in the future work.

Once the transfer function has obtained, the implementations for a PID and a deadbeat controller could be done. The PID control law was applied first. Some of the existing PID parameters tuning laws could not be applied such as Ziegler - Nichols tuning

method because these methods required that the plant response to be an S curve. Therefore, the three parameters were identified manually started with individual parameters and then the combination of three parameters by observing the response at the same time. In the end, the three parameters was defined and simulated by the Simulink model to make sure that the control effort would not go beyond the input current limitation. Then the PID control law was applied on the real maglev plant. It was found that the characteristics of a PID controller are not good enough to such a maglev plant which requires a quicker response and almost no overshoot. Also, for a PID controller, it is not possible to set up a desired time during controller design and all the responses are based on adjusting three parameters to see which combination of three variables can give a good response. In other word, a PID controller is an error sensitive but not a time sensitive controller.

Based on the simulations and implements of a PID controller, a deadbeat controller should be designed to handle the maglev system which can give a much quicker response and no overshoot. Firstly, a normal deadbeat controller was designed but during the design process, it was found that it was impossible for a normal deadbeat controller to handle and control a maglev system which has such a special plant response. Therefore, the literature review was started again, and two of the papers indicated that a PID based robust deadbeat controller could handle the problem and there were several experiment results in the papers. The robust deadbeat controller was developed in 1994 and an original paper was found with all the design procedures. The parameters for a robust deadbeat controller can be found using 'for' loops numerically. During initial simulation, the response for a robust deadbeat controller was good but it was not an ideal deadbeat response. However, changing the parameters slowly based on the observation of the system response, it is possible to get the system response which is very close or even the same as the ideal deadbeat response. The final simulation results suggested the overshoot of the system step response when the step input is 2cm is 0.013 (less than 0.0065%) which can be almost neglected. The settling time for the system response is 0.231 seconds and it has only 0.031 seconds' difference from the desired time. Just like many things happened in the real world, practical implementations are totally different from the theoretical simulation. The initial implementation of the PID-based robust deadbeat control

algorithm on the real plant using the optimised values failed because the maglev disk started oscillation and vibrating. The reason was figured out later that the ideal deadbeat response required an extremely high gain for the maglev system at which a maglev plant could not handle it. The obvious solution was to reduce the gain to a certain level range. By optimising the parameters the implemented system responses became much better but the small oscillation still exists and it was believed that the small oscillations were coming from the hardware of maglev plant itself.

In the comparison with a classic PID controller, it was found the settling time has been improved at least 55% at the linearization point and the overshoot was reduced as well using deadbeat control law. However, when it comes up to a huge movement away from the linearization point, frankly speaking, there was nothing improved at all. The further step was to investigate the adaptive control design based on fast, online, algebraic parameter estimation, exact linearization and generalized proportional integral output feedback control design which proposed by Moarles, R Feliu, V and Hebbertt in one of the latest IEEE transaction.

5.2 Further work

It was planned to simulate and implement a generalized proportional integral controller and it was assumed using GPI controller with the adaptive control techniques, would perform much better and has the capabilities to eliminate the linearization errors which means it could handle the huge movement from linearization point.

References

[1] Lee, H & Kim, K & Lee, J 2006, *Review of Maglev Train Technologies*, Hanyang University, Korea, viewed 10 May 2013, <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1644911>>.

[2] Nise, N 2011, *Control Systems Engineering*, 6th edn, California State Polytechnic University, Pomona

[3] NASA, 201, *example of controls*, viewed 10 May 2013, <<http://exploration.grc.nasa.gov/education/rocket/rktcontrl.html>>.

[4] Shanghai Maglev Transportation Development, 2013, *Shanghai Maglev project background*, viewed 10 May 2013, <<http://www.smtdc.com/en/gycf4.asp>>.

[5] Transrapid, Germany, view on May 2013, <http://www.transrapid.de/cgi-tdb/en/basics.prg?session=7cb383485265fd73_594179>

[6] Engineers Australia 2012, *Institution of Engineers*, Australia, viewed 20 May 2013, <<http://www.engineersaustralia.org.au/sites/default/files/shado/About%20Us/Overview/Governance/CodeOfEthics2000.pdf>>.

[7] Educational Control Products 2002, *Educational Control Products California*, America, viewed 14 March 2013, http://www.ecpsystems.com/controls_ifsw.htm

[8] Building Automation Consultant, *our fascinating look at building automation history starts in 270 B.C*, viewed 14 August 2013, <<http://www.building-automation-consultants.com/building-automation-history.html>>

[9] Astrom K. J. and Hagglund T. H., 'New tuning methods for PID controllers', *Proceedings of the 3rd European Control Conference*, 1995

[10] Clarke, A 2012, '*ROBUST DEADBEAT CONTROL OF THE NON-LINEAR MAGNETIC LEVITATION SYSTEM USING ADAPTIVE CONTROL TECHNIQUES*', Bachelor thesis, University of Southern Queensland, Australia.

[11] R. F. Stengel & L. R. Ray, '*Stochastic robust-ness of linear time-invariant control systems*', IEEE Trons. On AC, Vol. 36, pp.82-87, 1991.

[12] Dawes, J & Ng, L & Dorf, R & Tam, C 1994, '*Design of Deadbeat Robust Systems*', pp1597-1598, conference paper, University of California, Davis, viewed 3 August 2013, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=381478&tag=1

[13] Wai, Rj, Lee, JD&Chuang, KL 2011, 'Real-Time PID Control Strategy for Maglev Transportation System via Particle Swarm Optimization' *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, VOL. 58, NO. 2, FEBRUARY 2011, PP.620-645

[14] J. Kaloust, C. Ham, J. Siehling, E. Jongekryg & Q. Han 2004, 'Nonlinear robust control design for levitation and propulsion of a maglev system', *EE Proc.-Control Theory Appl.*, Vol. 151, No. 4, July 2004, 460-464

[15] Wei, Jp, Guo, Y, Huang, F&Wu, Yi 2012, 'Experimental Studies on H^∞ Control for Magnetic Levitation System with Friction Compensation', *Proceedings of 2012 International Conference on Modelling, Identification and Control*, Wuhan, China, June 24-26, 2012

[16] Lee, TE, Su, JP&Yu, KW 2006, 'Nonlinear Robust Control of a Magnetic Levitation System', *Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06)*

[17]Ma, J, Fan, W, He, F 2008, 'Parameters Self-adjusting Fuzzy PID Control in Magnetic Levitation System', *Harbin Institute of Technology* 150001, China

Appendix A

Project specification

University of Southern Queensland

FACULTY OF ENGINEERING AND SURVEYING

ENG4111 Research Project
PROJECT SPECIFICATION

FOR: **XinChen Fan**

TOPIC: **PIECEWISE ADAPTIVE CONTROLLER DESIGN
BASED ON ECP MODEL 730 MAGNETIC
LEVITATION SYSTEMS**

SUPERVISORS: **Dr. Paul Wen**

PROJECT AIM: **This project seeks to investigate, develop and
implement precise control systems using the ECP 730
Magnetic Levitation control/development instrument.**

PROGRAMME: Issue A, 5th March 2013

1. Investigate the ECP Model 730 hardware and software
2. Test the system and study the existing demo program and experiments
3. System model identification and linearization
4. Design a dead-beat controller for SISO the Maglev system
5. Simulate the design using MATLAB and Simulink
6. Implement, test and evaluate the design using the Model 730 Development System
7. Piecewise system model development and implementation
8. Design piecewise adaptive controllers for the levitation and guideline sub system
9. Simulate and improve the design using MATLAB and Simulink

APPROVED: **XinChen Fan (Student) 5/03/2013**

Paul Wen (Supervisor)

Examiner/Co-Examiner: _____

Appendix B

Matlab Code for Simulation

Find the force and distance relationship constant

```
%XinChen Fan - Final year project
clc
clear all
b=6.2; %offset parameter,
magMass=0.121; %Specify magnet weight, measured
gravity=9.81; %Specify gravity applied, known
magWeight=magMass*gravity; %calculate magnet weight
% Input the measured data from the Maglev plant
%Control effort applied to the machine coil
controlEffort=[4000; 5000; 6000; 8000; 10000; 12000; 14000; 18000;
22000];
%Physically measured output
measuredDisplacement=[1.8 2.4 2.7 3.5 3.8 4.3 4.4 4.5 4.5]';
%Apply mean regression (gradient) to function to obtain a value
a=mean(controlEffort./((measuredDisplacement+b).^4.*magWeight));
% plot experimental data
figure (1)
plot(measuredDisplacement,controlEffort,'+',measuredDisplacement,controlEffort,'--rs')
axis([0 5 0 25000]) %set the current axis
title('Actuator Characteristics, measured data');
xlabel('Magnet distance from lowest possible position');
ylabel('Equalising control effort')
grid on
% plot numerically estimated curve data
estimatedDisplacement=0:0.1:5; %matrix of test points
%evaluate test points to show numerical solution
estimatedControlEffort=magWeight.*a.*((estimatedDisplacement+b).^4);
figure (2)
hold on
plot(estimatedDisplacement,estimatedControlEffort/1000)
plot(measuredDisplacement,controlEffort/1000,'r+')
axis([0 5 0 25])
title('Nonlinear Actuator Characteristics');
xlabel('Magnet distance from zero point');
ylabel('Equalising control effort (KiloUnits)')
display(['For a displacement of b=', num2str(b), ' units, the other constant a=', num2str(a), ' units'])
display('This satisfies the numerical estimation of Full=U1/(a(y1+b)^4)')
legend('Estimated','measured');
grid on
```

PID parameters turning

```
%% P controller
clc
clear all;
d=1111;
n=[1 4 478.5];
t=0:0.001:10;
for Kp=0.1:1:0
    d1=Kp*d;
    g0=tf(d1,n);
    g=feedback(g0,1);
    y=step(g,t);
    dc=dcgain(g);
    if dc>0.9
        plot(t,y),disp(['Kp = ', num2str(Kp)]),disp(['DC gain = ',
num2str(dc)]);
        break
    end
end
title('P paramater Design')
xlabel('Time(seconds)')
ylabel('Displacement')
grid on

%% PD1
clc
clear all
t=0:0.001:10;
Kp=0.1;
d=1111;
d0=Kp*d;legend
n=[1 4 478.5];
s0=tf(d0,n);
s=feedback(s0,1);
k=step(s,t);
plot(t,k,'red');
hold on
for Kd=0.1:0.5:3
    d1=[d0*Kd,d0];
    g0=tf(d1,n);
    g=feedback(g0,1);
    y=step(g,t);
    plot(t,y);
    hold on
end
grid on
title('PD paramater& P paramater Design ')
xlabel('Time(seconds)')
ylabel('Displacement')
grid on
legend('P control','PD control',1);

%% PD2
clear all;
clc;
t=0:0.001:10;
n=[1 4 478.5];
for Kp=30
```

```

        for Kd=30
            d=[1111*Kp*Kd 1111*Kp];
            g0=tf(d,n);
            g=feedback(g0,1);
            y=step(g,t);
            plot(t,y);
            hold on
        end
    end
end
grid on
title('PD paramater')
xlabel('Time(seconds)')
ylabel('Displacement')
ylim([0 2])
%% PI Controller
clc;
clear all;
t=0:0.001:10;
Kp=4;
n1=[1 4 478.5 0];
d0=1111;
Ki=0;
d=[d0*Kp, d0*Kp*Ki];
g0=tf(d,n1);
g=feedback(g0,1);
y1=step(g,t);
np=[1 4 478.5];
dp=Kp*d0;
g1=tf(dp,np);
gp=feedback(g1,1);
y2=step(gp,t);
plot(t,y1,t,y2);
grid on
legend('PI paramater','P paramater')
title('PI paramater and P paramater')
xlabel('Time(seconds)')
ylabel('Displacement')
%% PI
clc;
clear all;
d0=1111;
Kp=0.5;
n1=[1 4 478.5 0];
Ki=1:0.5:5;
for m=1:9;
    d=[d0*Kp, d0*Kp*(Ki(m))];
    g0=tf(d,n1);
    g=feedback(g0,1);
    subplot(3,3,m);
    rlocus(g);
end
%%
t=0:0.01:10;
Ki=5;
Kd=0.1;
d=1111;
for Kp=0.5;
    % 0.1:1:1;
    D=[Kp*Kd*d, Kp*d,Ki*Kp*d];
    N=[1,4,478.5,0];
    g0=tf(D,N);

```

```

        g=feedback(g0,1);
        y=step(g,t);
        plot(t,y);
        hold on
end
title('PID')
xlabel('Time(seconds)')
ylabel('Displacement')
ylim([0 1.5])
grid on

```

plot result for PID simulation

```

% activate simulink
sim('pid')
t=linspace(0,5,length(simout(:,2)));
plot(t,simout(:,1)/10000);
hold on
plot(t,simout(:,2)/10000,'r');
xlabel('Time(s)')
ylabel('Magnet Position (cm)')
title('PID Control simulation')
legend('step input','real response by PID')
ylim([0 3])
grid on

```

Deadbeat parameters optimization and plotting

```

alpha=1.90; beta=2.20; Tsetd=4.04; Tr90=3.48;% from look up table
for Tdesired=0.2; %desired time
    omega=Tsetd/(0.7*Tdesired); %omega
    eq1=alpha*omega; %coefficients for s2
    eq2= beta*(omega^2); %coefficients for s1
    eq3=omega^3; %coefficients for s0
    for Kb=Tdesired/4 %Kb
        for K=0.1 % set K=1
            for K3=0.08 % set K3
                X=((eq1*(1111*K3*K*Kb + 1))-4-
1111*K*K3)/(1111*K*K3*Kb) %Calculate X
            end
            Y=eq3*(1111*K*Kb*K3+1)/(1111*K*K3) %
            Calculate Y
            Ka=(eq2*(1111*K*K3*Kb+1)-478.5-1111*K*K3*Kb*Y-
1111*K*K3*X)/(1111) %Calculate Ka
        end
    end
end
end

sim('deadbeatfail') %call simulation
t=linspace(0,1,length(simout(:,2)));
plot(t,simout(:,1)/10000);
hold on
plot(t,simout(:,2)/10000,'r');
xlabel('Time(s)')
ylabel('Magnet Position (cm)')
title(['Deadbeat Control resposne moving away from linearzaion point
when desired time=' ,num2str(Tdesired),'seconds'],'FontSize',15)
legend('step input','response for Deadbeat controller')

```

```
grid on
```

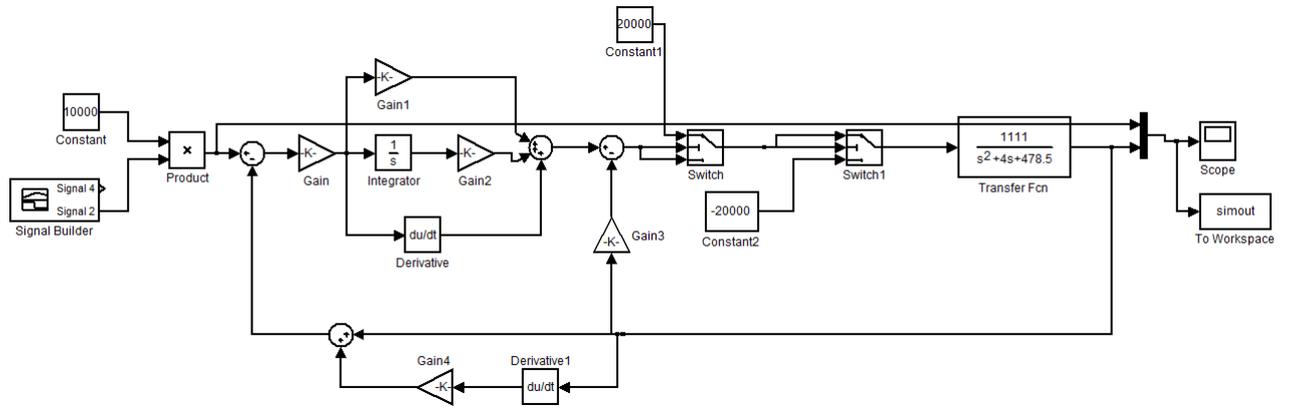
Deadbeat implementation

```
clear all
close all
clc
importData=load('deadbeat0.5.txt');
sampleTime1=importData(1:226,2);
output=importData(1:226,5);
input=importData(1:226,3);
figure(1)
plot(sampleTime1,(output/10000-3))
hold on
plot(sampleTime1,input/10000,'r')
xlabel('Time(s)')
ylabel('Magnet Position (cm)')
title('PID based Robust deadbeat Implementation at 1cm')
legend('step input','Real response')
grid on
```

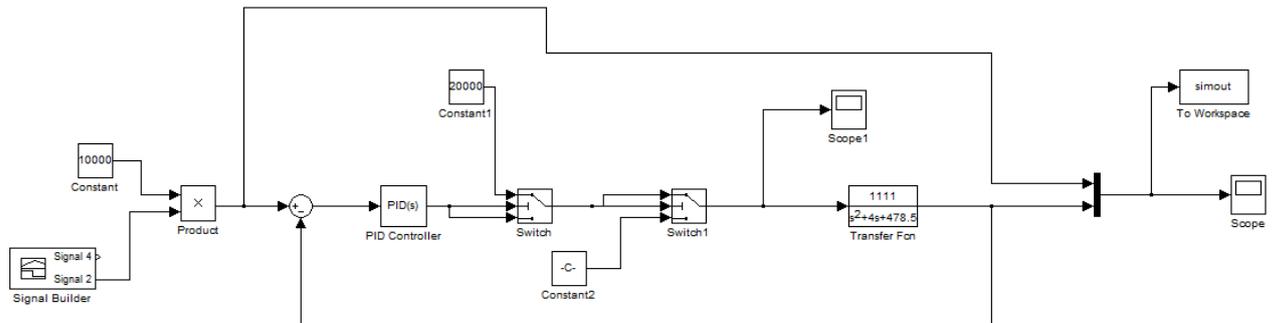
Appendix C

Simulink Model

Deadbeat model



PID model



Appendix D

C code for maglev plant

PID controller design

;Set Ts=0.001768 s

*****Declare variables*****

#define y1cal q2

#define y1rawo q3

#define kp1 q4

#define kd1 q5

#define kdd1 q6

#define Ts q7

#define y1str q8

#define comp_effort q9

#define pos_last q15

#define u1str q16

#define u1o q17

#define u1 q18

#define laser1 q19

#define y1o q20

#define uterm1 q21

#define uterm2 q22

#define error q23

#define ki1 q24

#define kid1 q25

#define ui q26

#define ui_last q27

#define delta_y1 q28

*****Initialize*****

Ts=0.001768 ;for local use only must set Ts in dialog box for sampling period

```

;Specify Parameters
u1o=11800 ;gravity offset in N/10000
y1o=30000
kp1=1.7
kd1=0.091
;ki1=8
ki1=20
kdd1=kd1/Ts ;Discrete time terms, compute here to save real-time computation
kid1=ki1*Ts
ui_last=0
control_effort2=0
;*****Begin Real-time Algorithm
begin
y1str=sensor1_pos-y1o ; Use calibrated sensor, sensor1_pos=y1cal
error=cmd1_pos-y1str
delta_y1=y1str-pos_last
ui=kid1*error+ui_last
;CONTROL LAW
u1str=kp1*error-kdd1*delta_y1+ui
;OUTPUT
u1=u1str+u1o ;Add gravity offset
u1term1=6.2+sensor1_pos/10000 ;nonlinear actuator compensation in three steps
u1term2=u1term1*u1term1
comp_effort=0.000165*u1term2*u1term2*u1
control_effort1=comp_effort

;UPDATE

```

```
pos_last=y1str
ui_last=ui
q10=sensor1_pos
end
```

Deadbeat controller implementation

```
;Set Ts=0.000884 s
;*****Declare variables*****
#define Ts q4
#define plot q10
#define perrorplot q11
#define ierrorplot q12
#define derrorplot q13
#define uterm1 q17
#define error q18
#define derror q19
#define perror q20
#define ierror q21
#define lastpos q22
#define dcalc q23
#define icalc q24
#define intlact q25
#define intnow q26
#define pid q27
#define kpid q28
#define output q29
#define K q30
```

```
#define Ka q31
#define K3 q32
#define X q33
#define Y q34
#define Kb q35
#define H1 q36
#define lasterror q37
#define previous1 q38
#define previous2 q39
#define previous3 q40
#define previous4 q41
#define previous5 q42
#define previous6 q43
#define previous7 q44
#define previous8 q45
#define previous9 q46
#define previous10 q47
#define currentpos q48
#define calc q49
#define index q50
```

```
*****Initialize*****
```

```
Ts=0.000884 ;for local use only must set Ts in dialog box for sampling period
```

```
;;Specify Parameters
```

```
control_effort2=0
```

```
control_effort1=0
```

intlact=0

currentpos=0

lastpos=0

lasterror=0

K=0.01

K3=0.01

Ka=-0.5133

Kb=0.25*0.16

X=6909

Y=201600

previous1=0

previous2=0

previous3=0

previous4=0

previous5=0

previous6=0

previous7=0

previous8=0

previous9=0

previous10=0

index=0

```

;*****Begin Real-time Algorithm

begin

H1=Kb*(currentpos-lastpos)/Ts+currentpos
error=(cmd1_pos+30000-H1)*K*K3
dcalc=error-lasterror
derror=(dcalc/Ts)/(0.0015)
intnow=(error+lasterror)*Ts*0.5*Y
ierror=intnow+intl原因
perror=X*error
pid=perror+ierror+derror
kpid=pid
output=kpid-Ka*currentpos

if (output>24000)
control_effort1=24000
endif
if (output<-24000)
control_effort1=-24000
endif
if (output>-24000 and output<24000)
control_effort1=output
endif

plot=currentpos-30000
perrorplot=lasterror

```

```
ierrorplot=intnow
derrorplot=ierror

intlast=ierror
lasterror=error
lastpos=currentpos
index=0
;endif
index=index+1

previous10=previous9
previous9=previous8
previous8=previous7
previous7=previous6
previous6=previous5
previous5=previous4
previous4=previous3
previous3=previous2
previous2=previous1
previous1=sensor1_pos
end
```

Appendix E

Risk Assessment

The professional engineer has the responsibility to take care to others, themselves and the environment. A risk assessment is required to identify the risk which is required throughout the duration of the project. The purpose of a risk analysis is to identify all possible risks and hazards, and reduce the likelihood and consequence severity of these issues to as low as reasonably practicable.

A **hazard** is an object which has the potential to cause harm to someone or the environment.

A **risk** is the likelihood of the stated hazard causing harm. Levels include:

- Extremely slight
- Very slight
- Slight
- Significant
- Substantial

The **exposure** is how often people are exposed to the risk

- Very rarely
- Rarely
- Occasionally
- Regularly
- Frequently
- continuously

The **consequences** are the repercussions which may occur should the hazard occur, levels include:

- minor equipment/component damage
- major destruction of equipment
- minor injury/illness
- major injury/illness

- possible death

Risk Assessment	
Hazard Description:	Hazard is the unstable movement of the magnet on the Maglev machine, which is exposed to the open world
Risk:	Some of the students may touch the magnetic
Consequence	Uncontrolled movement can cause small crushing and pinching forces. minor equipment/component damage and minor injury may occur.
Measures could be taken to reduce the risk	Abortion the plant once it is working in a unstable condition

Risk Assessment	
Hazard Description:	Two laser sensors are used to detect the position of the magnet
Risk:	Possible, The laser module is hidden from sight
Consequence	Would damage the eyesight or even becomes blind.
Measures could be taken to reduce the risk	Do not look at the laser directly and to put laser hazard label around the workplace