# Predictive Tracking of Quasi Periodic Signals for Active Relative Motion Cancellation in Robotic Assisted Coronary Artery Bypass Graft Surgery

by

Jason Rotella

Submitted in partial fulfillment of the requirements

for the degree of Master of Science

Thesis Advisors: Dr. M. Cenk Çavuşoğlu

Dr. Wyatt S. Newman

Department of Electrical Engineering and Computer Science

CASE WESTERN RESERVE UNIVERSITY

January, 2005

# Contents

**List of Figures** **vi**

**List of Tables** **vii**

**Abbreviations** **viii**

**Abstract** **ix**

**1 Introduction and Background** **1**

1.1 Thesis Outline . . . . . 2

1.2 Coronary Heart Disease . . . . . 3

1.3 Surgical Solutions . . . . . 4

1.3.1 Angioplasty . . . . . 4

1.3.2 Coronary Artery Bypass Graft Surgery . . . . . 6

1.4 Robotic Surgical Solutions . . . . . 8

1.5 Current Heart Tracking Methods . . . . . 10

1.6 Future Prediction . . . . . 12

1.7 Heart Data . . . . . 13

1.8 Thesis Contributions . . . . . 15

**2 Test Bed Systems** **16**

2.1 System Objectives . . . . . 16

2.2 QNX Notes . . . . . 17
2.3 Subwoofer Speaker . . . . . 18
2.3.1 Assembly . . . . . 18
2.3.2 Speaker Specifications . . . . . 21
2.3.3 Amplifier Specifications . . . . . 21
2.3.4 Sensor Specifications . . . . . 22
2.3.5 Modeling and System Identification . . . . . 23
2.4 PHANToM Robot . . . . . 32
2.4.1 Robot Specifications . . . . . 33
2.4.2 Amplifier Specifications . . . . . 35
2.4.3 Modeling and System Identification . . . . . 38

**3 Control Algorithms** **46**
3.1 Observer Implementation . . . . . 46
3.2 Position Plus Derivative Control . . . . . 47
3.3 Pole-Placement Control . . . . . 50
3.4 Model Predictive Control . . . . . 53
3.5 Signal Estimated Model Predictive Control . . . . . 63

**4 Simulation and Results** **68**
4.1 Algorithm Implementation . . . . . 70
4.1.1 PD Control . . . . . 71
4.1.2 Pole Placement . . . . . 71
4.1.3 MPC . . . . . 73
4.1.4 Signal Estimated MPC . . . . . 75
4.2 C Simulations . . . . . 77
4.3 Simulation Results of Algorithm Testing . . . . . 77
4.3.1 PD Control . . . . . 78

4.3.2 Pole Placement . . . . . 80
4.3.3 MPC . . . . . 80
4.3.4 Signal Estimated MPC . . . . . 82

**5 Results of Hardware Experiments** **85**
5.1 Speaker Results . . . . . 85
5.1.1 PD Control . . . . . 86
5.1.2 Pole Placement . . . . . 86
5.1.3 MPC . . . . . 88
5.1.4 Signal-Estimated MPC . . . . . 89
5.2 PHANToM Results . . . . . 92
5.2.1 PD Control . . . . . 93
5.2.2 Pole Placement . . . . . 93
5.2.3 MPC . . . . . 96
5.2.4 Signal-Estimated MPC . . . . . 98

**6 Analysis and Conclusion** **100**
6.1 Results Revisited . . . . . 100
6.2 Conclusion . . . . . 101
6.3 Future Work . . . . . 101
6.4 Wrap up . . . . . 103

**Bibliography** **104**

# List of Figures


1.1 Portion of heart signal used for tracking . . . . . 13

1.2 Power Spectrum Density of Heart Signal . . . . . 14

2.1 Block diagram of speaker system setup . . . . . 19

2.2 Photograph of speaker setup . . . . . 20

2.3 Ultrasonic Sensor Offset Circuit . . . . . 21

2.4 Results of DC Step Test On Speaker : Hysteresis Curve . . . . . 24

2.5 Speaker Spring Plot . . . . . 26

2.6 Stepper experiment with least-squares 7th-order fit . . . . . 28

2.7 Flipped axis stepper experiment with 7th-order fit . . . . . 28

2.8 Stepper experiment with 7th-order inverse mapping function and fit . 29

2.9 Physical Block Diagram of Speaker . . . . . 31

2.10 Subwoofer Speaker Frequency Response and Fit . . . . . 31

2.11 PHANToM Haptic Device . . . . . 33

2.12 Stick Diagram of PHANToM in home position with appropriate joint labels . . . . . 34

2.13 DC Amplifier Transfer Function and Correction . . . . . 37

2.14 Amplifier 2 Bode Plot . . . . . 37

2.15 Amplifier 4 Bode Plot . . . . . 40

2.16 Experimentally measured frequency response of Joint 1. Results obtained from using different input magnitudes are superimposed. . . . 40

2.17 Motion trajectory used to measure Coulomb friction value of the PHANToM. The velocity has a trapezoidal profile. . . . . 41
2.18 Coulomb Friction . . . . . 42
2.19 Fit to PHANToM Bode Plot without $1/s^2$ . . . . . 43
2.20 Fit to PHANToM Bode Plot with $1/s^2$ . . . . . 44
2.21 Reduced Fit to PHANToM Bode Plot. This plot includes the $1/s^2$ term. . . . . 45

3.1 Observer Block Diagram . . . . . 48
3.2 Continuous PD Controlled Block Diagram . . . . . 49
3.3 Pole-Placement Controlled Block Diagram . . . . . 51
3.4 Coarse Block Diagram of MPC Control . . . . . 54
3.5 Attempting MPC control with a one cycle delay prediction after waiting for one cycle . . . . . 64
3.6 Estimated Signal and Actual Signal . . . . . 67

4.1 Diagram of Experimental Setup . . . . . 69
4.2 Resampling Effects . . . . . 70
4.3 Simulink model of PHANToM and PP controller used to determine effects of Coulomb friction and saturation on control . . . . . 73
4.4 Error correction functions with a 150 step error horizon. The furthest left function is 1st-order while the furthest right is 10th-order. . . . . 76
4.5 Simulated PD controlled system output and desired output . . . . . 79
4.6 Simulated PD error between desired signal and system output . . . . 81
4.7 Simulated PP controlled system output and desired output . . . . . 81
4.8 Simulated PP error between desired signal and system output . . . . 83
4.9 Known-future high-gain MPC tracking results . . . . . 83
4.10 Known-future low-gain MPC tracking results . . . . . 84

4.11 Signal estimated MPC tracking signal superimposed over the desired heartbeat signal . . . . . 84

5.1 PD controlled system output and desired output . . . . . 87
5.2 Speaker PD Control Effort . . . . . 87
5.3 Speaker PP Controlled System Output and Desired Output . . . . . 88
5.4 Speaker PP Control Effort . . . . . 89
5.5 Known-Future MPC System Output and Desired Output . . . . . 90
5.6 Speaker error between known-future MPC output and desired signal . 90
5.7 Signal-Estimated MPC System Output and Desired Output . . . . . 91
5.8 Speaker error between signal-estimated MPC and Desired Output . . 92
5.9 PHANToM PD-Controlled Tracking Error and Control . . . . . 93
5.10 PHANToM PP controlled tracking results . . . . . 94
5.11 PD and PP algorithms controlled system output utilizing identical feedback gains . . . . . 95
5.12 PHANToM MPC tracking results and error . . . . . 96
5.13 PHANToM MPC control effort . . . . . 97
5.14 Pole Placement By Algorithm . . . . . 98
5.15 PHANToM signal-estimated MPC Results . . . . . 99

# List of Tables

6.1 Complete Results of Simulation and Hardware . . . . . 100

# Abbreviations

CAB - Coronary Artery Bypass

CABG - Coronary Artery Bypass Graft

CHD - Coronary Heart Disease

CL - Closed Loop

CPB - Cardiopulmonary Bypass

DAC - Data Acquisition Card

DOF - Degree(s) Of Freedom

MPC - Model Predictive Control

OPCABG - Off Pump Coronary Artery Bypass Graft

OS - Operating System

PD - Position plus Derivative

PP - Pole Placement

PWM - Pulse-Width Modulation

SISO - Single Input Single Output

# Predictive Tracking of Quasi Periodic Signals for Active Relative Motion Cancellation in Robotic Assisted Coronary Artery Bypass Graft Surgery

Abstract

by

Jason Rotella

Traditional coronary artery bypass graft (CABG) surgery has undesirable side effects that range from cognitive loss to increased hospital stay that are believed to be related to artificial heart pumps. It has been proposed that a robotic surgical instrument can be developed to perform CABG surgery on the beating heart, therefore alleviating the need for the heart pump. By tracking beating-heart motion with a surgical robot, the relative motion between the heart and the robot can be cancelled and the surgeon can operate on a heart that appears to be stationary. The constraints on such a tracking system, however, are rigorous as failure to do so in surgery would be fatal. In this thesis, several control algorithms for high precision tracking of the heart motion have been developed, implemented, and tested on simulated and real systems. It was found that a model predictive controller (MPC) can be combined with estimated future information to produce the most accurate and robust controller. This novel variation of the MPC algorithm has been developed and tested in order to show the gain in tracking accuracy.

# Chapter 1

# Introduction and Background

In order to perform coronary artery bypass graft surgery, it is often necessary to use a heart pump and clamps to prevent heart motion during the operation. Using these surgical tools, however, can cause unwanted long term side effects. If the heart were able to beat freely during surgery, these tools would not be needed and it is possible that these effects might be alleviated. It has been proposed that a surgical robot may be able to maintain a constant distance between the heart and the surgical instruments. This would make the heart appear stationary to the eyes and instruments of the surgeon and hence make it possible to perform surgery while the heart is beating. Cancelling the motion allows the surgeon to operate on a relatively stationary heart and permits the heart to beat freely.

The major difficulty in developing such a robotic surgical tool is that traditional control methods do not produce high enough precision to effectively cancel out the motion. Therefore in this study, a novel control algorithm based on model predictive control is proposed and implemented. To perform tracking, the model predictive control algorithm calculates optimal gains that utilize the future-known values of a desired signal. By implementing control that utilizes more information than just the current position of the heart, it is possible to perform control that will produce higher

precision over classical methods. This is the basis behind the choice of using a model predictive control algorithm. Since the future heart signal will not be known and available to the model predictive control algorithm, an estimate of the future signal based on past information will be used.

## 1.1 Thesis Outline

This thesis is divided into 6 different chapters. The introduction will begin by giving background information on heart disease and the current surgical methods for treating it. It will continue by speaking of general surgical robotic methods followed by robotic surgeries done on the heart. Chapter 1 will conclude by discussing different tracking methods that have been attempted and will speak of others that have tackled the same problem.

Chapter 2 will give an overview of the subwoofer speaker and PHANToM manipulator systems that were used for testing of the control algorithms. This chapter will also explain the system-modelling methods as well as supply the general specifications for each of the systems. It will also give a description of the experimental setups.

Chapter 3 will supply in-depth descriptions of the observer implementation and the tracking control algorithms attempted. It will also discuss some of the intricacies involved within the tuning process.

Chapter 4 will discuss the simulations that were conducted before specific control algorithm testing on the actual systems began and show the results of those simulations.

Chapter 5 will give the results of the control algorithms on each of the hardware systems and speak of the general methods attempted for obtaining the best results.

Chapter 6 will discuss the results as a whole and describe future work to be done. It will sum up the results of the thesis and make a general statement about all of the

work performed herein.

## 1.2 Coronary Heart Disease

Coronary heart disease (CHD) affects more then 1 in 4 people around the world [1]. CHD is caused by the build up of plaque within the arteries. Plaque consists of cholesterol and other fatty deposits that are in the blood stream. As plaque gets lodged and stuck within the arteries, it hardens and a lesion is formed. Over time more and more plaque builds up until the artery becomes too clogged to allow the flow of blood into the needed areas. This concept is very similar to that of a clogged drain. Initially the drain sides are smooth and unobstructed. As time goes by, dirt and grime starts to build up on the side until eventually, the flow through the pipe is inhibited.

When a tissue is denied blood, it cannot obtain sufficient nutrients and oxygen needed for normal operation. If this tissue is brain tissue, the result is a stroke. If blood cannot flow into the limbs, loss of functionality and/or gangrene will occur. If the tissue is the heart, an angina or even a heart attack can occur.

When blood stops flowing to the heart, the deprived heart obtains its nutrients from locally stored reserves in an attempt to continue functioning. While operating anaerobically, the tissue begins to produce lactic acid. The lactic acid builds up because there is no blood flow to remove it. Like muscles that have been overworked, the heart becomes "sore" and depending on the severity of blood deprivation, a heart attack can occur. Unlike sore muscles which eventually are allowed to relax and replenish nutrients from blood flow after being used, the heart must constantly run and is unable to get blood and return to normal operation once an artery has been blocked.

An angina or temporary chest pain can occur based on the same principle. If a

person overworks themselves and the heart does not receive enough oxygen, lactic acid is temporarily produced and causes some minor chest pain. However, if there is not a blockage, enough blood will be able to eventually flow to recover and hence the pain is mild and temporary.

CHD is more prominent in people who smoke, are overweight, have high cholesterol or high blood pressure, and do not exercise regularly or eat nutritiously. People with a family history of heart disease are also more susceptible to the disease. By taking the appropriate measures to keep blood pressure and cholesterol down, a person can significantly decrease their chances of suffering from CHD.

## 1.3 Surgical Solutions

In general, heart disease can be caught and treated before a heart attack occurs. The treatments vary from a change of diet and medications to surgery. There are two primary types of surgery that can be performed. These are angioplasty and coronary artery bypass graft surgery.

### 1.3.1 Angioplasty

Angioplasty is one surgical option for dealing with coronary heart disease [2]. It consists of feeding a catheter into a clogged or partially blocked artery in order to open a passageway within the artery. The most common type of angioplasty uses a balloon inside the artery. The balloon is fed into the artery with a catheter and then inflated at the location of the blockage in order to compress the plaque and widen the artery. Generally the balloon must be inflated multiple times in order to accomplish this task. Another technique for angioplasty involves attaching a laser to the end of the catheter. The blockage is eliminated by heating the soft tissue surrounding the plaque and causing it to break down chemically, thereby releasing the plaque from

the side of the vessel wall. After the plaque has been detached, it is then removed with a catheter. This is only one technique that is used to remove the plaque from the artery. Yet another catheter technique grinds the hardened portions into micro particles that can safely be washed away by the blood stream.

Balloon angioplasty may sometimes also include a stent. A stent is a metal mesh that is locked into place around the inside of the artery. The stent is inserted folded around the balloon and then expands into its locking position when the balloon is blown up. It holds the artery open and clamps down the plaque. Some of the newer types of stents have been coated with drugs in order to help assure that the stent works correctly and that further blockage does not occur. Although a stent can help improve angioplasty, a stent can still fail and artery blockage can still occur.

No matter how the angioplasty is performed, it is important not to simply break up the plaque into chunks or dislodge it from the walls of the artery without removing it. These pieces can become lodged further down the blood stream and cause blood clots which can lead to heart attack or stroke.

Angioplasty is less traumatic and the recovery time is typically shorter when compared to bypass surgery. Unfortunately it sometimes acts as only a short-term solution. Restenosis or reclosing of the artery can often occur after a period of about 6 months. This is caused from the tissue of the artery regrowing after it had been damaged from the angioplasty. Often restenosis requires an additional surgery to be performed, whether it be another angioplasty or bypass surgery.

Furthermore, angioplasty is normally performed when there is only one clogged artery. Bypass surgery will typically be recommended for patients with multiple blockages. Ultimately the decision is made on a case by case basis.

### 1.3.2 Coronary Artery Bypass Graft Surgery

CABG surgeries are used for severe blockages in arteries and often used when multiple arteries are clogged [3]. CABG can be performed in two different methods, on-pump or off-pump. Though the details may differ for each of the methods, the intrinsic process is the same.

The surgeon first obtains a blood vessel from somewhere else in the body. This is typically from somewhere in the chest but can be from the arm or leg. After the vessel is obtained, the chest is opened and the blocked artery is cut near the blockage. The newly obtained vessel is sutured into the side of the cut vessel. A hole is then punched into the aorta and the other side of the vessel is attached. In order to attach the blood vessel to the aorta, often the aorta is locally clamped. Upon completion of this process, the surgeon checks on the blood flow through the newly attached line. These steps are repeated for every artery that is blocked.

The complications of CABG are damage to the aorta, creation of emboli (unwanted particles in blood stream [4]), bleeding, stenosis (closing of artery), arrhythmias (irregular heart beat), myocardial infarction (death of heart tissue due to lack of blood) and stroke or death.

A successful procedure will suppress symptoms of CHD, alleviate angina and reduce further heart problems. Ultimately it will prolong the patient's life. CABG surgeries were performed on more then 800,000 people last year alone. With a 90% success rate this surgery is not only one of the most complicated to perform but one of the most common major surgeries [3].

#### On-Pump

The on-pump CABG procedure utilizes a heart-lung machine in a cardiopulmonary bypass process (CPB). The machine operates by inserting tubes into the aorta and several of the major incoming veins. The CPB procedure involves taking the blood

in from the veins and feeding it through pumps within the machine. The heart-lung machine adds oxygen to the blood and maintains the blood at an appropriate temperature to renter the body through the aorta.

The heart is effectively turned off and is still during this procedure, which makes the surgery less complicated for the surgeon. During this process, clamps are used to restrict the flow of blood into the arteries to be bypassed.

### Off-Pump

Off-pump CABG (OPCABG) is performed while the heart is still beating and the CPB procedure is avoided. The surgeon will locally clamp or stabilize the portion of the heart being operated upon in order to perform surgery. This off-pump method makes the surgery more complicated but does have it benefits. Generally in this process, the bleeding is reduced and the blood flow through the body is more oxygen rich. This surgery generally caries less risk of side effects. OPCABG is a relatively new procedure and currently not the primary method used. It is also known as beating-heart surgery.

### CABG Complications

Two of the major risks of on-pump surgery include neurocognitive losses and severe aortic manipulation.

The neurocognitive effects are a noted decline in the patient's reasoning and thinking skills. This can also include sensory difficulties and cause personality changes of the patient. These effects generally will extend the needed recovery time. Though very common among people who have had CABG surgeries, the effects generally wear off after a period of 6-12 months (for most patients).

It is believed that the neurocognitive effects have a direct relationship with the amount of time spent on pump. Many studies have been conducted and documented

this relationship. One such study [5] concluded that cognitive decline had a higher occurrence in patients who had on-pump surgery rather than off-pump surgery. Furthermore, they said that those that had neurocognitive decline, due to on-pump surgery, were more likely to have a decline after a six-month period than those who were off pump.

It is believed that the emboli caused by the removing of the tubes from the heart after the bypass has taken place is the primary cause of this cognitive dissipation.

For the same reason, severe aortic manipulation is also a cause for concern during CABG. Since the aorta is clamped, it is possible to damage or loosen any plaque located in the aorta (creating emboli). Significant damage due to clamping can also cause aortic dissection (splitting of aorta).

Though OPCABG surgeries are preferable to on-pump CABG surgeries, the off-pump method is not as well developed and can not always be performed. Robotic surgery is a viable option to help expand the ability to perform off-pump surgery. However, currently the surgical methods are still somewhat limited. A team in Canada was able to perform the first robotic beating heart surgery in 1999 and later described the need to compensate for the movement disturbances of the vessels near the heart [6].

## 1.4 Robotic Surgical Solutions

Robotic surgery has been investigated for more than 20 years and practiced now for more than 10 years. One of the first surgeries used a robot to aid hip replacement. This robot precisely bored out a hole in which to fit the replacement hip [7] and hence minimized many of the problems associated with a poorly drilled cavity.

The ability to precisely place and use tools is just one of the many advantages that robotic surgery is able to provide. Robot repeatability allows a procedure to

be performed almost identically every time called upon. Newer robotic techniques relieve the surgeon of certain tasks such as holding and positioning of cameras [7]. Robots that are used for minimally invasive surgeries, are giving surgeons access to areas that previously would have required large incisions and openings. The benefits associated with just the minimally invasive operations include reduction of trauma and morbidity and shortened operation time as well as recovery time.

Robots for medical practice can be classified into two different subgroups, those that work along side of the surgeon and those that are working directly with the surgeon. The two groups are called surgical CAD/CAM and surgical assistants [8].

A surgical CAD/CAM uses information obtained before and during the procedure to carry out some part of the operation. The surgeon monitors the part of an operation that the robot performs. The surgeon acts as a "life-guard" during the robotic portion of the surgery. He can stop and or change the procedure at any time and ultimately has control over the robot. When using these types of robots, the surgeon and robot are not directly connected and the robot is actually performing the task under the eyes of the surgeon [8].

A surgical assistant works directly under the surgeon. The surgeon directly controls what the robot is doing, which includes everything from suturing to camera placement. The robot acts as more of a medium to facilitate the surgery. This group of robots is the group researched to perform CABG surgery.

The first successful robotic CABG surgery was conducted in May 1998 in France ([9],[10]). This surgery was done using a heart-lung machine in order to facilitate the procedure, but the first OPCABG was successfully implemented shortly thereafter. There are currently two systems that have been used successfully for heart surgery and specifically CABG. These two systems are the da Vinci system from Intuitive Systems and Zeus from Computer Motion Inc (Note that as of March of 2003 these companies have merged [11]). Though both of these machines have been used for

CABG, neither has been approved for such a procedure in the United States at this time.

Both robots are designed for laparoscopic surgery and have been FDA approved for a number of procedures. These procedures include but are not limited to mitral valve repair, gastric bypass surgery, esophageal surgery and radical prostatectomy [12]. Both machines utilize teleoperation in a master/slave configuration. Neither however, at this time is capable of doing heart tracking and motion cancellation [13]. They are able to perform OPCABG by using passive stabilizers on the heart such as the Medtronic Octopus ([9], [14], [13]).

## 1.5 Current Heart Tracking Methods

Like any new technology, smaller steps were taken in order to accomplish a larger goal. Before heart tracking was attempted, several groups attacked the problem of reducing the disturbance due to the respiratory system.

For instance, during radiosurgery, a tumor can move a significant amount due to breathing. This requires that a larger amount of radiation be applied to the patient in order to irradiate the tumor. In a pair of studies, an attempt was made to track the tumor motion and hence shrink down the dose of radiation. In order to lock on to that tumor, a surgical robot attempted to monitor and compensate for breathing motion ([15], [16]). Even though the surgical procedure is non contact, the task is made more complicated by the fact that the internal motion did not directly map to the external motion. Even as this was the case, both studies still concluded that robotic compensation could be accomplished.

Riviere et al. [17] attempted to cancel respiratory motion during percutaneous needle insertion surgery. The procedure consisted of carefully positioning and inserting a needle into a kidney with use of a robotic positioning system. In order for the

robot to maintain accuracy (even when performed without the robot), the breathing was suspended. Though the complications from stopped breathing are different from suspending the heart, the same moral still results: it is not preferable but it is the only way. The problem hence is parallel to that of this thesis. They used an adaptive controller that was able to model and predict the breathing motion of the patient. Their results supported the feasibility of doing respiratory motion cancellation. Furthermore, they speculated that this technology could be extended to heart motion tracking.

After attempts to cancel breathing motion were successful, others attempted to track the heartbeat motion. Trejos and Salcudean [18] performed a feasibility study on the ability to perform tasks on a moving a target versus performing the task on a motion-cancelled target. This study used human subjects instead of robots and the motion cancellation was done by attaching the person's hand to the oscillating target. The study reported that tasks could be performed using motion cancellation.

A patent was actually issued based on the relative motion idea [19]. A platform was designed for a surgeon for such a task. This platfrom was controlled in a linear fashion and forced to track heart beat motion. The surgeon strapped his hands to the platform then performed surgery. The heart was made periodic through use of a pacemaker.

Nakamura et al. did a similar heart tracking experiment using a PHANToM robot utilizing cameras as the visual system to do the heart-motion sensing [20]. The tracking error was too large to perform surgeries, but the error may have been due to the camera feedback system in place. Furthermore, Nakamura did not use any type of advanced algorithm using a future prediction for tracking purposes.

Thakral et al. [21] attempted to recreate the heart signal through online analysis techniques. They monitored rat heart motion and then used a recreated signal with adaptive control techniques in attempt to follow the motion. In this experiment, no

robot was used and only a displacement sensor was mounted onto a moving linear actuator to monitor the success of the algorithm.

There has been a parallel effort to this work going on in France using a similar approach to the one presented within this thesis [22]. It used current information in an attempt to predict the future signal. Their future prediction technique was very similar to that of [21]. The future signal was fed into a slightly modified model precitive controller in order to get higher precision tracking. This has been tested on the AESOP surgical robot [23] and again the heart position was monitored by a high-speed camera. The error in this case was better then Nakamura's experiment and the disturbance due to the organ motion was greatly reduced but not completely cancelled.

Tracking and subsequent motion cancellation is not purely for medical applications. Mehra et al. [24] did work towards cancelling out the disturbance due to the road beneath a vehicle with active suspensions systems. They stated that an MPC algorithm could be implemented by using future disturbances (knowledge of the road), and concluded that use of this knowledge made the ride smoother due to better suspension operation. Furthermore, a similar application attempted to cancel disturbance in movement and station keeping of autonomous underwater vehicles [25]. Again, an MPC algorithm was used that utilized future knowledge of the water disturbance to accomplish the autonomous tasks.

## 1.6 Future Prediction

The future prediction task that is associated with the MPC task within this thesis is also not a new idea. Kalman and Wiener made some of the initial steps to signal prediction in [26] and [27]. The goal in these works (and the numerous subsequent works) was to create a model that would among other tasks effectively predict the

future of a random signal.

## 1.7 Heart Data

The heart data used for tracking purposes in this thesis was collected using a sonomicrometry system. Piezoelectric crystals were placed around a beating heart. These crystals emitted and received ultrasonic waves. By measuring the time between emission and reception of the wave between crystals, the distances could be recorded. Note that it is also necessary to know the medium through which the waves are travelling. For more information, see [28].

The data was collected from an adult pig at a sampling rate of 257Hz. The peak displacement from the average value was 12.1 mm while the RMS displacement was only 5.1 mm. The collection was carried out by M. Cenk Cavusoglu. The desired heartbeat signal used for tracking can be seem in Figure 1.1.

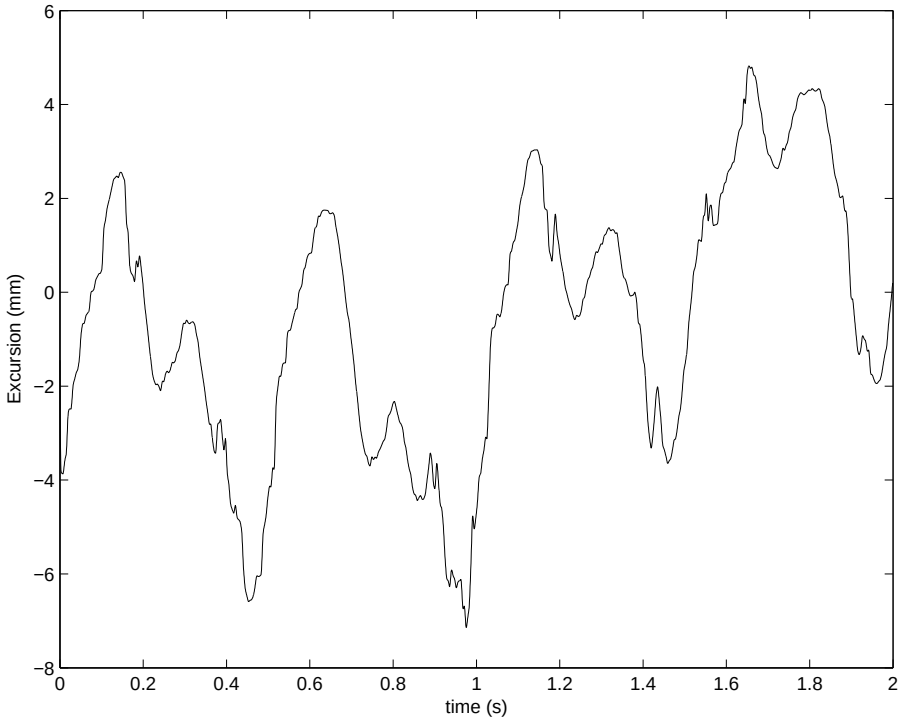



Figure 1.1: Portion of heart signal used for tracking

The signal has two primary modes of operation. The first occurs at a frequency of 0.37Hz. It corresponds to the breathing motion of the patient. The second mode is the primary heart motion. The heart was beating at 120 beats per minute when the data was collected (corresponding to a 2Hz mode). These modes can be seen in Figure 1.2. If the breathing motion is removed, the remaining signal consists entirely of the principle frequency of 2Hz and harmonics of that frequency. These harmonics carry significant information up to 20Hz. The primary motion however is in the principle heart beat and the breathing. Figure 1.2 is created in part with the `fft()` function in MATLAB.

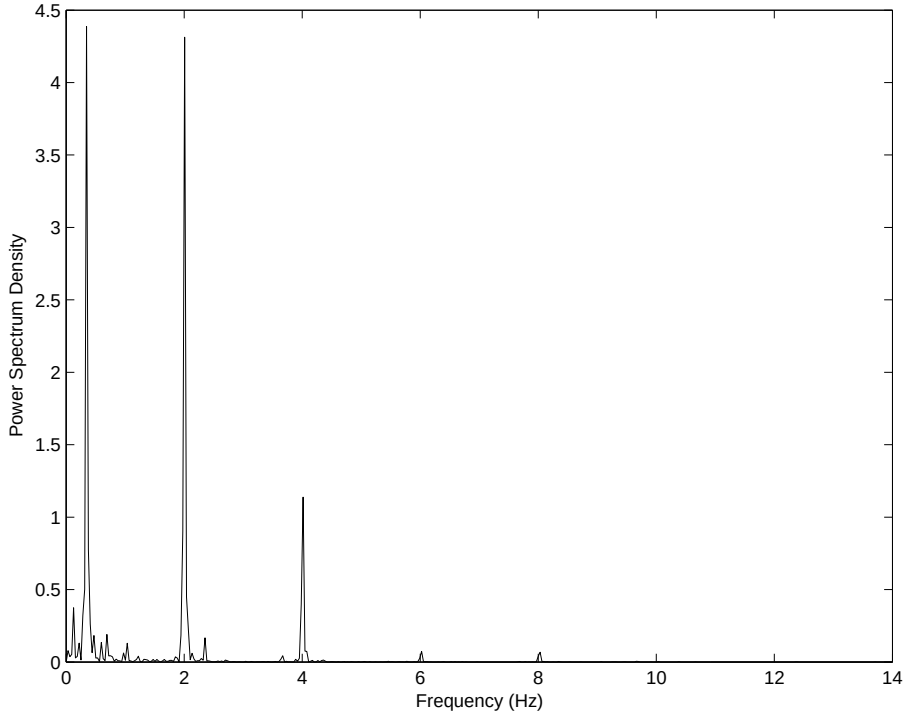



Figure 1.2: Power Spectrum Density of Heart Signal

This information is used as specifications for the test bed system and controller performance. The peak to peak range of the heart motion is around 20mm with significant frequency information up to 20Hz. Most vessels operated upon in surgery are 0.5mm - 2.0 mm in diameter. To be able to suture such arteries, accuracy down to 100 micrometers is needed [13].

## 1.8 Thesis Contributions

This thesis studies the effectiveness of different types of control algorithms to perform tracking on the position of a heart for robotic assisted coronary artery bypass graft surgery. It proposes and shows that the best algorithm for tracking is a model predictive control (MPC) algorithm. Due to the acausal properties of the model predictive control algorithm, the exact method cannot be used for surgery. A variation on this algorithm is presented and tested for accuracy on two different test bed systems. The systems are also tested by using the original MPC algorithm and classical control methods such as pole placement and position plus derivative control and the results are compared. This variation uses past knowledge of the heart motion along with a correction function in order to estimate a future signal. It will be shown that the signal estimated MPC algorithm presented here performs better then the classical algorithms. For this reason, it is believed that it is possible to obtain precision high enough through control to perform tracking of the heart for surgical applications.

# Chapter 2

# Test Bed Systems

In order to develop and test the algorithms it is necessary to obtain and model some type of hardware testbeds. This chapter discusses the two systems that are used as test beds in detail and explains the methods of obtaining frequency response models. To help give a better background for the setups, the operating system used to perform the control algorithms is also explained.

## 2.1 System Objectives

The algorithms developed in this project were implemented and tested on two different test bed systems. They consist of a subwoofer speaker and a PHANToM robot.

The speaker is an intrinsically stable device (meaning it is stable without the use of feedback) and possesses a highly repeatable nature. Though smaller speakers generally do not move with great amplitudes, large subwoofers have excursions of several inches. Speakers are also designed to move with high frequencies. This combination of high bandwidth and large excursions make a subwoofer speaker an ideal initial test bed for algorithm development.

The PHANToM robot possesses different characteristics from those of the subwoofer speaker and will provide more insight into the effectiveness of the algorithm

on a "real" system. The PHANToM is more likely to be similar to an actual robot used for surgery. The PHANToM's lightweight frame and drive system also allow for sufficient motion and speed to attempt to track the heartbeat signal.

The following sections will discuss each of these systems in detail and the processes used to accomplish the above mentioned tasks.

## 2.2 QNX Notes

The QNX system was chosen as an OS primarily because of its real-time capabilities. QNX is one of the only true real-time systems and is free for non-commercial research purposes. These facts made it ideal for the type of control experiments that would be performed.

The QNX operating system is a Unix-based real-time operating system (OS) that is distributed freely to students and educators. QNX utilizes a microkernel that helps reduce driver malfunction and system crashes by only using signals, timers and scheduling. Scheduling of the programs is based on an interrupt timer and a priority that is set for each software task. For control purposes, the interrupt timer is set to interrupt at the desired control frequency and generally the control program is given the highest priority. Giving the control program the highest priority forces the kernel to execute the control code before other processes are run and hence assures that the control is carried out before the next timer interrupt.

The control software has been set up such that the timer is initiated and the timer interrupt turned on. The code was then set to loop until the user terminated the program. Commands to start and stop control of the system were available within the software as well. Upon starting the controller, the interrupt timer became the control loop timer and the control was executed.

The QNX OS has a number of features to make it more user friendly. These

features include an X-windows based windowing system and software and libraries to create graphical user interface's (GUI). As with any non-Windows operating system, the number of available drivers for different computer hardware configurations was limited, but QNX has made extensive efforts to be compatible with basic protocols and other OS's. More specific information can be obtained from the QNX website [29].

## 2.3 Subwoofer Speaker

With the computer used for control, the subwoofer system was made up of the speaker itself, an ultrasonic sensor and an amplifier. A diagram of the setup can be seen in Figure 2.1 and will be described in the following section. After the setup is described, the specifications for each of the major components mentioned above will be given. The modeling process and the system model is subsequently reported at the end of this section.

### 2.3.1 Assembly

The subwoofer speaker was mounted onto a wooden enclosure that was part of an old speaker system. The wiring to the speaker was designated as 'signal' and 'return'. A large inductor was wired in series with the speaker into the signal side of the speaker cable. This inductor coil acted as a choke to the PWM amplifier and helped smooth out the incoming signal. The input channel of this amplifier served as the input to the system.

The ultrasonic sensor was mounted directly onto the speaker enclosure with four metal bars as can be seen in Figure 2.2. The sensor itself had been attached to a Plexiglas plate and the plate was attached to the frame. The two horizontal bars had rails that allow the sensor position to be adjusted in the horizontal plane while

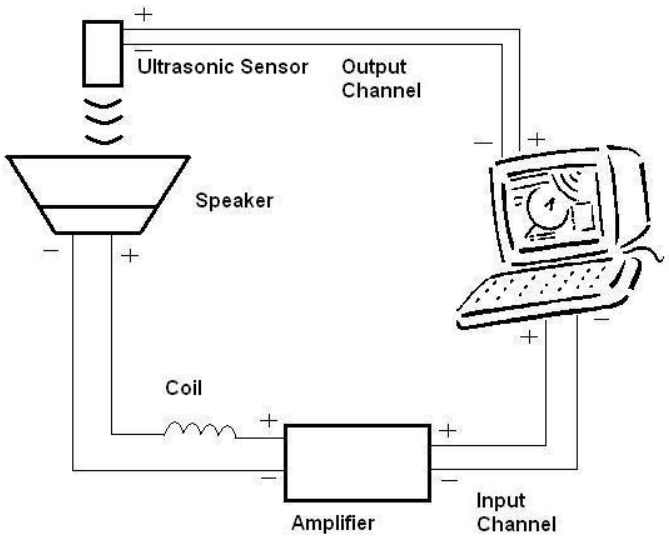



Figure 2.1: Block diagram of speaker system setup

the vertical bars allowed the sensor height to be adjusted. This setup allowed for the most freedom in positioning the sensor.

In order to ensure the sensor reading did not fault throughout its excursion, the following test was conducted. The sensor was positioned such that the 'target present' LED was on. Then the speaker cone was manually pressed down (such that the path between the sensor head and target area was not obstructed). If the 'target present' LED turned off or flickered, the sensor head position was readjusted until the LED remained on. The sensor was adjusted until the entire lower portion of the speaker excursion registered a reliable value with the sensor. This was repeated for the upper excursion by connecting the amplifier and commanding positions to move the speaker towards the sensor.

The 0 to 10V output range of the sensor was shifted to fit into the -5 to 5V range of the data acquisition card (DAC) (note that the DAC was used to connect the input and output signals of the system to the computer). This voltage shift was done with

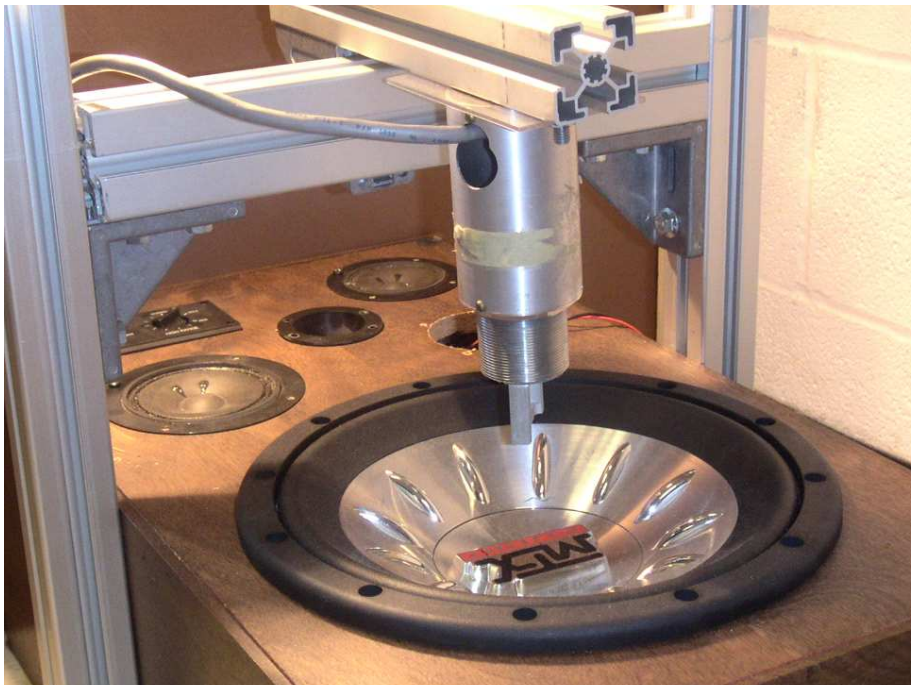


Figure 2.2: Photograph of speaker setup

a simple subtraction circuit comprised of a LF347 operational amplifier and several resistors as configured in Figure 2.3 with values of R1,R2,R3 equal to 5kΩ. The output signal of this circuit served as the output of the system. The circuit simplifies to $V_{in} - V_{offset} = V_{out}$. The voltage offset was 5 Volts.

Further offset calibration was necessary due to the sensor outputting a displacement relative to the sensor position. It was more convenient to use a position that had its origin within the speaker's range of motion. Therefore, a calibration procedure was conducted before the commencement of each experiment. The average value of the sensor output calculated over 100 samples was used as the offset value for the sensor. The sign of the sensor signal was also inverted due to the fact that as the speaker moved down, the distance reading of the sensor increased.

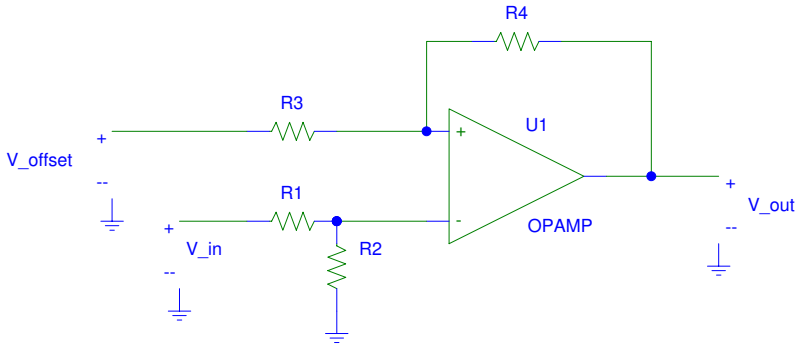



Figure 2.3: Ultrasonic Sensor Offset Circuit

### 2.3.2 Speaker Specifications

The speaker used for testing was an MTX Audio T8124A subwoofer. The speaker was 12 inches in diameter and approximately 6 inches deep. The linear excursion as rated by the data sheet was 12.2 mm. The linear excursion is defined as the maximum distance the center of cone can move from a center point (this referred to the peak value and not the peak-to-peak value). The speaker was rated at 4.0Ω and for 400 watts RMS. The frequency response range was between 23Hz and 150Hz. For a full list of specifications, see the MTX website [30].

### 2.3.3 Amplifier Specifications

The amplifier used to output current to the speaker utilized pulse-width modulation (PWM). To help smooth out the amplifier output, a coil was connected in series with the speaker input.

The amplifier was rated to output a peak current of 15 Amps with a bandwidth

specified at 3kHz. The amplifier switched at 25kHz.

PWM operates by emitting voltage pulses that control the current output of the amplifier. The duration and polarity of the pulse determine the amount of current supplied to the motor [31].

### 2.3.4 Sensor Specifications

The speaker displacement was measured using an ultrasonic non-contact sensor and specifically a Cleveland Motion Controls Pulsonic Sensor. The sensor was configured to monitor a 6.35 to 25.4 cm range with a repeatability of 0.0127 cm. The sensor output an analog voltage proportional to the distance of the target. The voltage output corresponded to 2.54 cm per volt. The sensor updated at a rate of 800Hz.

The sensor output a voltage that had a range preset according to jumpers on the side of the sensor controller. Even though the sensor was set for the 25.4 cm range, it had an upper distance limit of 3.05 meters away. The minimum allowable distance between the target and the sensor was 6.35 cm.

The sensor operated by emitting a pulse that was reflected off a target surface and then received back by the sensor. The time taken for the signal to make the trip was then calculated. By combining this information with the speed of sound in air, the distance between the sensor and the target was calculated. To compensate for any change of the speed of sound in air and as a result calibrate itself, the sensor had a precisely machined part attached directly in front of the surface that emit the waves. The distance of this part to the emitter was known. The metal part partially reflects the beam and as a result the speed of sound in air was determined automatically.

Since the sensor operated with reflected waves, it was necessary that the sensor head be positioned within 5 degrees of the perpendicular of the target surface (given the surface was already in range of the sensor). To get an accurate reading, the angle should be within 1.5 degrees of the perpendicular position . An adequate position of

the sensor head was verified by one of three LED's on the sensor controller unit.

Further information on the sensor is available from [32],[33].

### 2.3.5 Modeling and System Identification

#### Experimental Nonlinear Modeling

When attempting to perform the linear modeling experiment mentioned in the next section, the output wave was not a true sine wave. The wave was slightly distorted. In an attempt to map this nonlinearity, a program was created that would issue a series of sequential step commands to the amplifier and subsequently the speaker. Note that the command received by the amplifier was a voltage and that the speaker received a current which was converted into a force. For a step size of $v$, the program commanded for the first step $v$, then $2v$ then $3v$ and so on. When the program reached the upper limit, it started stepping back down until it reached the lower limit. Upon reaching the lower limit it switched directions again and returned to the zero value of output. Each step was held long enough to collect a settled position reading from the ultrasonic sensor. A settling time of about half a second was used. Even though the signal may not have been entirely settled after the half second delay, the short settling time helped ensure that the speaker coil would not over heat when subjected to larger DC values of current.

For purposes of the needed experiments, it was not necessary to create a software fuse to prevent the speaker coil from over heating. Simply keeping the step time short was enough to not damage the speaker. During normal operation, the speaker acted as a fan to itself and as a result could utilize much higher amplitudes of input current. As long as there was some kind of oscillating signal on the speaker, the movement of the cone helped the coil to remain cool. Upon completion of the data collection, the input voltage was plotted against the output position. This plot can be seen in

Figure 2.4.

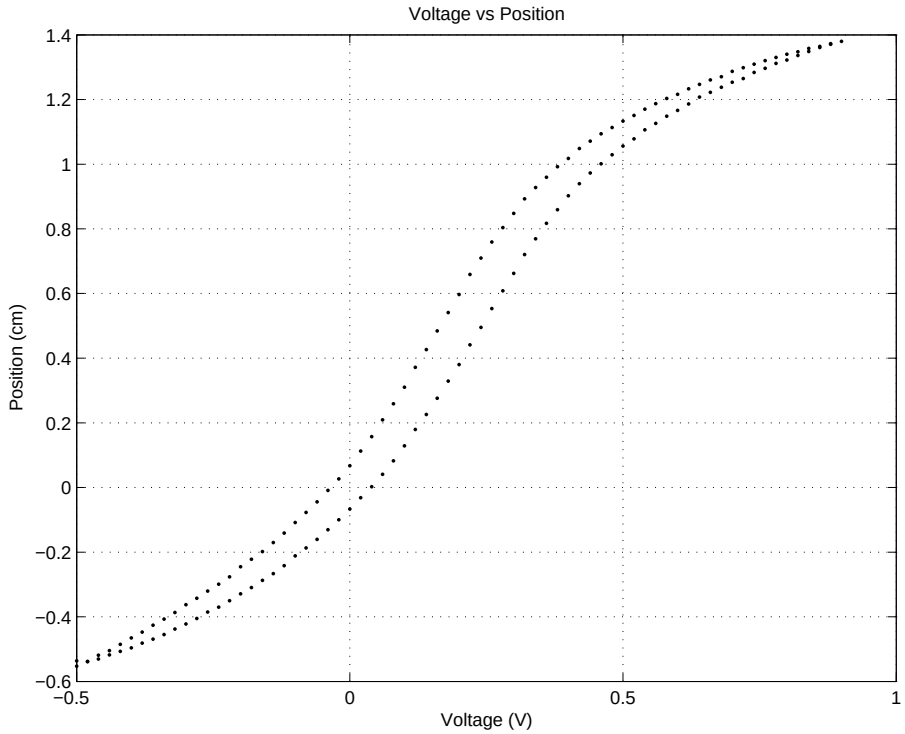



Figure 2.4: Results of DC Step Test On Speaker : Hysteresis Curve

This data suggested two kinds of nonlinearity: hysteresis and saturation. The curve was centered 0.4 cm away from the speaker's zero input position. The linear region of the speaker was originally specified as 1.2 cm (amplitude) in the subwoofer data sheet [30]. Disregarding the hysteresis, this curve suggested a linear region of about 0.4 cm. The short linear range was a source of problem as motions up to 1.0 cm were needed for the desired task. Steps were taken in order find the cause of the nonlinearity and then provide appropriate compensation.

Since the center of the curve was shifted, the lower and upper limits were adjusted and the experiment was repeated to obtain a symmetric curve (a symmetric curve is one that has the same amount of distortion on each end of the hysteresis curve). By finding a symmetric curve, it was possible to maintain a higher amount of operation in the linear region.

It is possible to code offsets into the input and output of the control code in order

to center this plot around a "zero output". Doing this would enable the system model and all data to be centered around (0,0) in Figure 2.4. However, this was undesirable, as a constant current would be needed to hold the speaker at the zero position. As stated before, this would heat the coil and possibly damage the speaker.

There are two possible explanations for the saturation nonlinearity. The voice coil motor and the speaker cone (which is similar to a spring) both contain linear regions near the center of their motion and tend to level off for high deflections. Determining the cause of the nonlinearity is important because, even though the effect looks the same, the correction techniques are different. The following test was performed in order to seek the cause of the saturation.

To ascertain the cause of the nonlinearity, it was necessary to isolate one of the devices (either the spring or the voice coil) and test it individually. Testing of the voice coil by itself proved difficult as it would require disassembling the speaker and tearing apart the cone. Testing the cone however could be done while the voice coil was not activated. To see any nonlinearities of the cone, weights were added while the position was monitored. A linear spring would have a straight-line relationship between the force and the displacement. To test the spring, the motion sensor was turned on and zeroed. A ring was then attached to the speaker cone. The ring was nearly 16 cm in outer diameter and possessed a 12.7 cm inside diameter. The ultrasonic sensor used the center of the speaker as a target for position reading. The ring allowed the weights to be placed on the cone without obstructing the sensor. The weight of the ring was measured and the deflection that occurred after adding the ring was recorded. Two blocks of known similar weight were then added to the ring in a symmetric fashion and the deflection of the speaker was again recorded. The blocks were added symmetrically in order to distribute the weight more evenly across the speaker in an attempt to assure a collinear deflection. The process was repeated by adding heavier blocks until the deflection reached the desired limit. The position

versus weight data was then plotted and fit to a linear function. This is shown in Figure 2.5.

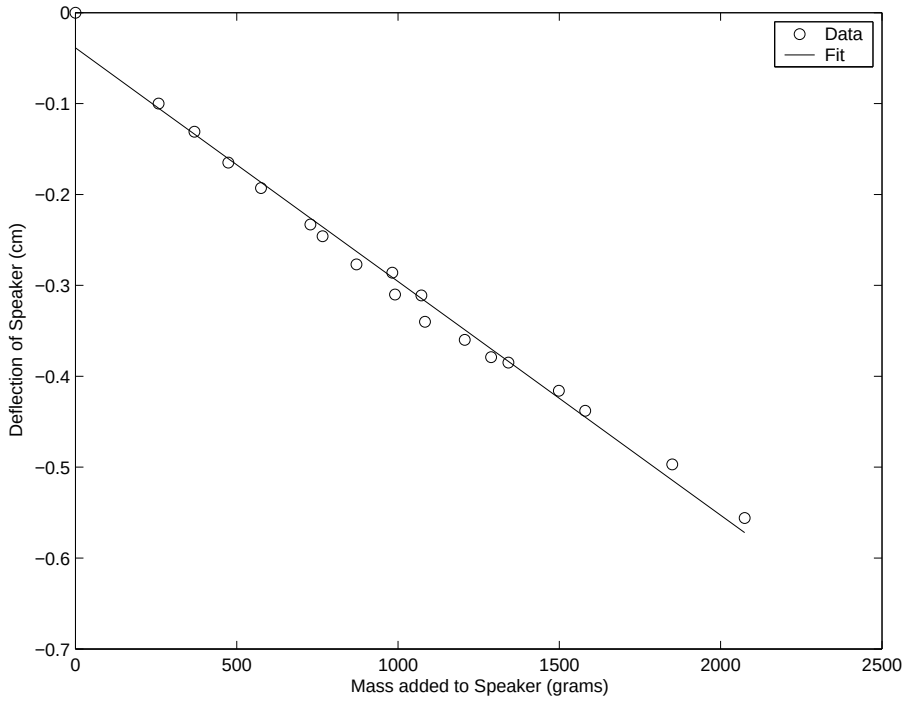



Figure 2.5: Speaker Spring Plot

From observing Figure 2.4, the saturation started to occur after the -0.2 cm portion of the plot. Figure 2.5, however, appeared very linear all the way out to the desired limit of motion. The linear fit provided an $R^2$ (square of the correlation coefficient) value of 0.989. This plot showed that the saturation was not caused by the spring and hence was most likely entirely from the voice coil.

The voice coil is a ring that carries a current in the tangential direction. The coil moves axially within an annular air gap. Within this gap, their is a rated magnetic field, which interacts with the coil current to provide an axial force. As the voice coil displaces outside the annulus, the $B$ field penetrating the coil decreases, resulting in decreased force.

The saturation problem can sometimes be ignored in cases where the saturation occurs outside the desired operating range of the system. The system can be classified

as locally linear and controlled accordingly. For the purposes of this experiment, this would not be adequate. The desired range of motion was 2.0 cm peak to peak which easily overextends any linear region of the plot.

A mapping function was used to correct for the saturation nonlinearity. First the hysteresis was ignored by just taking an average of the two curves in Figure 2.4. The resulting plot was a one-to-one function that still had the saturation nonlinearity. For a given desired position, there existed only one corresponding voltage. Therefore a desired position (DC) could be achieved by simply feeding in the voltage corresponding to the desired position in the plot. The relationship between the input and output was compensated for by using this mapping.

In order to obtain a function mapping, the following steps were executed. First the two axes were flipped and replotted as can be seen in Figure 2.7. The line dividing these data points was a least-squares polynomial fit. The best fit plot was the one that was the closest to the average of the upper and lower curves of 2.7.

It was found that a seventh-order fit worked well with this set of data and can be seen in Eqn 2.1.

$$\begin{aligned} y = {} & 0.19209x^7 - 0.48985x^6 + 0.54312x^5 - 0.2454x^4 \\ & +0.29328x^3 - 0.37788x^2 + 0.56939x - 0.0001 \end{aligned} \tag{2.1}$$

The results of the stepper test using the constructed inverse function is shown in Figure 2.8. The resulting relationship with the compensation was

$$y = 0.9954x + 0.0448. \tag{2.2}$$

The hysteresis problem still remained but the saturation problem was effectively removed. The hysteresis seems to be the result of a very large damping time constant. Currently the specific cause of this effect is not known. Due to sufficient accuracy

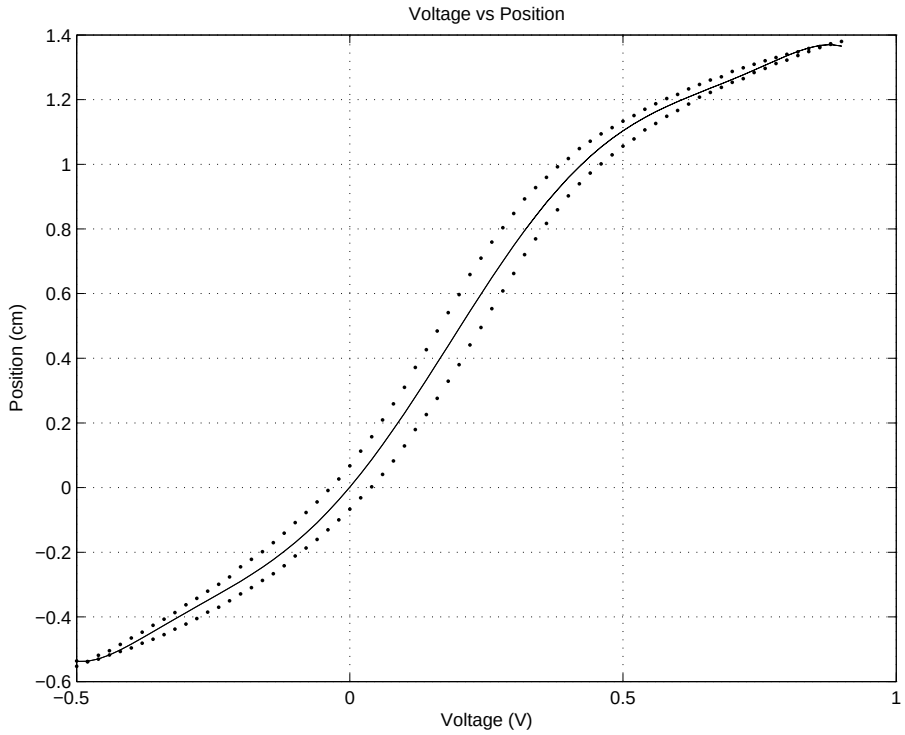



Figure 2.6: Stepper experiment with least-squares 7th-order fit

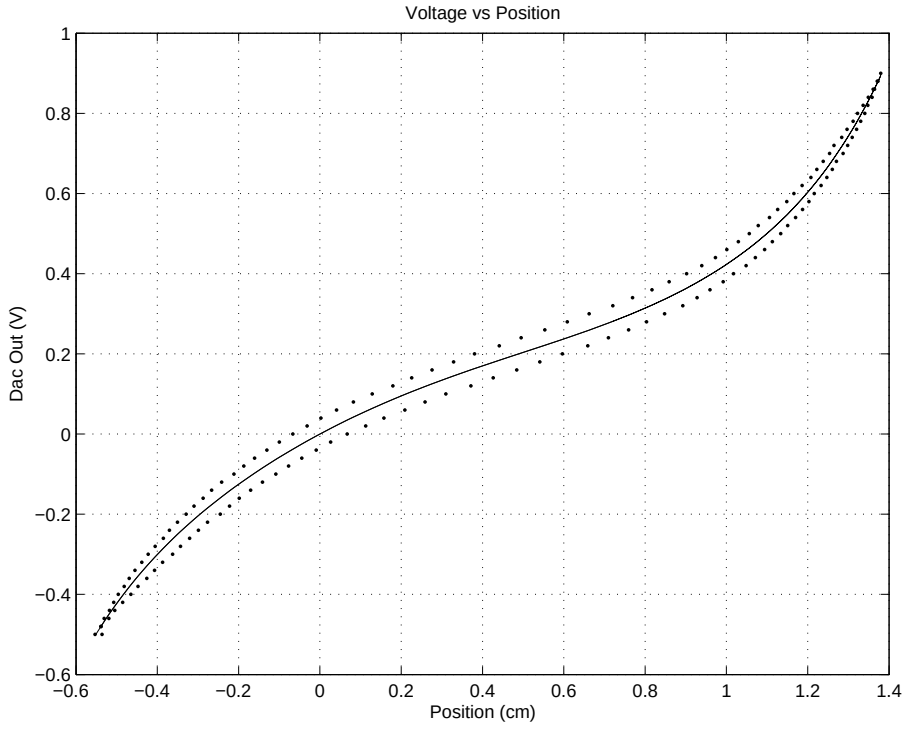



Figure 2.7: Flipped axis stepper experiment with 7th-order fit

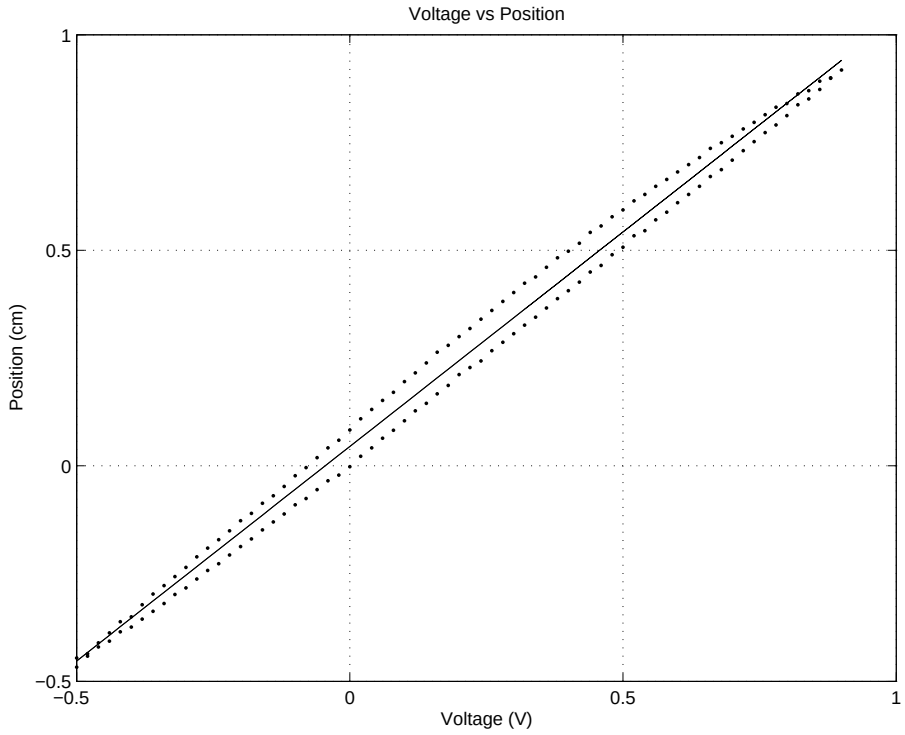



Figure 2.8: Stepper experiment with 7th-order inverse mapping function and fit

without the fix, the hysteresis problem was left uncompensated.

## Experimental Linear Modeling

The transfer function for the speaker dynamics was obtained experimentally through a transfer function mapping procedure. The subwoofer speaker is an open loop stable device. By giving the system a sinusoidal input, the output should also be sinusoidal at the same frequency as the input wave (true for linear systems). The phase difference and the magnitude difference between the input wave and the output wave then can be calculated. By repeating this procedure at different frequencies of the input wave, the phase and magnitude response of the system can be calculated.

The Matlab `invfreqs()` function was utilized to fit a transfer function model to the experimental measured frequency response [34]. The first and second arguments of this function contain the experimentally obtained system information. The first

argument is created by combining the magnitude and phase information into one single vector. This is accomplished with the below equation.

$$complex = magnitude \times \exp\left(i\phi\frac{\pi}{180}\right) \tag{2.3}$$

The second argument consists of the vector of the frequencies corresponding to the phase and magnitude information. The last two arguments are the order of the numerator and denominator respectively of the system transfer function.

It is possible to add a fifth argument to the function which does data weighting. Weighting causes certain portions of the transfer function to be better fit then other portions. This extra argument was not utilized in this particular instance but is used in modeling of the PHANToM robot in the next section. The function returns two vectors which are coefficients of the numerator and the denominator of the fitted transfer function.

A transfer function with third-order denominator and zeroth-order numerator was fit to the experimental data. The order of the transfer function was chosen based on a physical model of the system constructed.

The order of the transfer function was specified to correspond with a logical model of the system. A speaker is made up of a cone and a voice coil, which can be construed as a mass. The mass is acted upon by forces on the coil and connected to ground through a spring and damper in parallel. The physical model can be seen in Figure 2.9.

The transfer function of Figure 2.9 is

$$\frac{x(s)}{F(s)} = \frac{1}{(Ms^2 + Bs + K)}. \tag{2.4}$$

Now if this transfer function is combined with the expected amplifier roll off (a low-pass filter effect), the expected transfer function is:

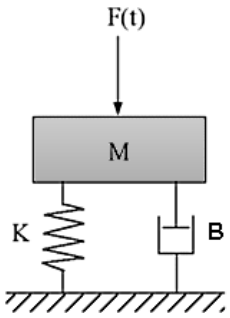



Figure 2.9: Physical Block Diagram of Speaker

$$G(s) = \frac{1}{(\frac{s}{c}+1)(Ms^2+Bs+K)}. \tag{2.5}$$

The parameter $c$ in Equation 2.5 is the amplifier roll off frequency.

To see the accuracy of the fit, the Bode plot of the transfer function was plotted over the frequency and magnitude information. This subwoofer Bode plot can be seen in Figure 2.10. The circles represent the experimentally measured magnitude and phase points while the continuous line is the fitting function.

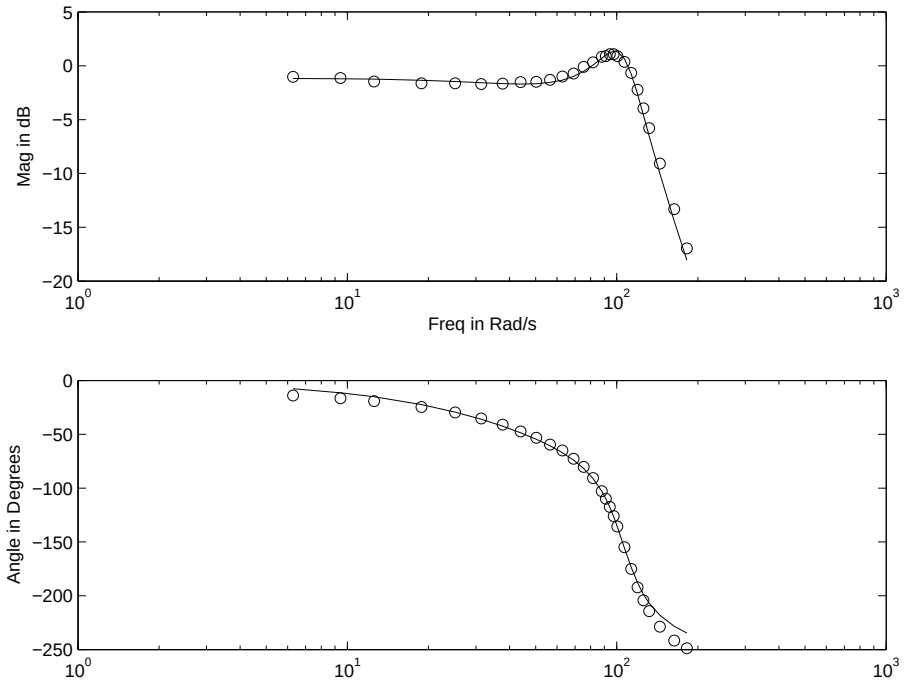



Figure 2.10: Subwoofer Speaker Frequency Response and Fit

The transfer function obtained with the frequency response method had a best fit of

$$G(s) = \frac{5.533 \times 10^5}{s^3 + 96.24s^2 + 1.339 \times 10^4 + 6.313 \times 10^5}. \tag{2.6}$$

The frequency response method of obtaining this transfer function was only valid if the system was linear over the operating region. This approximation only held true after the nonlinear modeling was enforced onto the subwoofer.

## 2.4 PHANToM Robot

The PHANToM robot (from Sensable Technologies [35]) is typically used by two different groups. The first is researchers using it as a haptic device. The second group is computer graphics designers who use the PHANToM's feedback along with Free Form Concept software to perform digital design. The PHANToM's design allows for 3-dimensional force feedback and full back-drivability. This makes it an ideal tool for anything requiring force feedback. The PHANToM is light weight and possesses a low inertia, which allows it to move more quickly than conventional industrial robots. Another advantage is the low friction that is associated with the motor drive system. Though limited to three degrees of freedom (DOF) to control, an adapter can expand the PHANToM's workspace to 6 DOF. A picture of a PHANToM setup can be seen in Figure 2.11.

In the area of haptic research, the PHANToM is used for algorithm development and position control, performing haptic training ([36] [37]), and measuring force for psychophysics experiments [38].

As in the previous section, the robot and amplifier will be discussed in detail followed by an explanation of the modeling procedure, and finally the model will be presented.

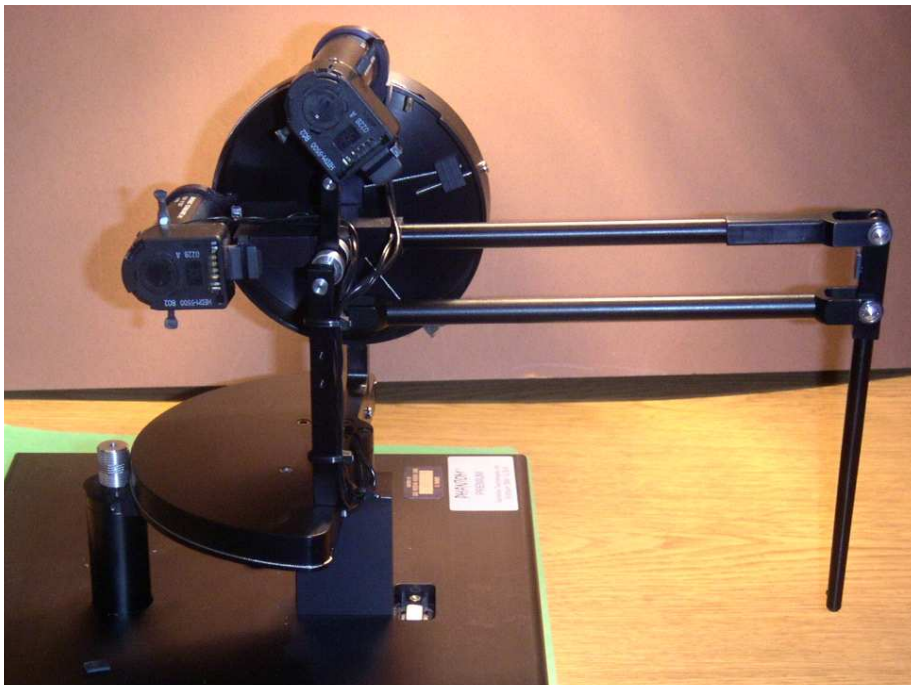


Figure 2.11: PHANToM Haptic Device

### 2.4.1 Robot Specifications

The primary novelty of the PHANToM robot is its motor drive system, which is based on the rotary mechanism design of Carson and Preonas [39]. It eliminates the use of gears through a cable pulley system. Each of the motors has a threaded capstan attached to its shaft. These threads have large enough grooves to hold a cable which is wrapped around the capstan several times. The motor is then positioned such that it is very close to a larger diameter cylindrical base. The cable from each end is then pinned tightly on each side of the larger cylinder. The effect is that spinning the motor causes the cable to pull and hence rotates the larger metal cylinder and allows the PHANToM to move.

The PHANToM possesses three joints that will be referred to as Joints 1, 2 and 3. These labels correspond to the $\theta_1, \theta_2, \theta_3$ respectively in Figure 2.12. Joint 1 rotates in a plane that is horizontal with the ground. The motors for Joint 2 and Joint 3 are actually attached to the same upper cylinder which is stationary. As a result

the rotation occurs in the same plane for each of these joints. The motors rotate themselves around this cylinder in order to move the outer segments of the robot.

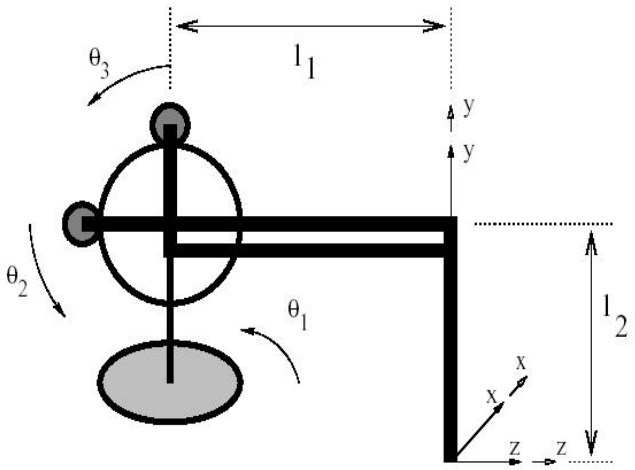



Figure 2.12: Stick Diagram of PHANToM in home position with appropriate joint labels

The encoders used for feedback are attached to each of the motors. For the large base rotation of the PHANToM, each encoder tick corresponds to 0.00117 radians. This fine resolution is because of the gear ratio between the motor and axis of rotation.

The PHANToM's design requires a low-weight frame and connecting pieces. These parts are made thinner and as a result, are not as strong. The actual position of the end-effector is distorted because of the inherent flexibility of the structure and hence not necessarily where the encoder value reads. This compliance of the PHANToM's beams is one source of dynamic complexity in the PHANToM.

The compliance can be easily seen by performing the following experiment. Joints 2 and 3 are clamped into place and hence not permitted to move. Joint 1 is activated with some type of sinusoidal motion. The end effector will oscillate within the Joint 1 range of motion and slightly in the vertical direction. Even though vertical motion

is seen, the encoders on each of the clamped motors will not have changed.

The encoder resolution on the motors of the PHANToM are 4000 counts per revolution. With the gear ratio between the Joint 1 motor and the actual Joint 1 angle of the PHANToM, this encoder count corresponds to a resolution of 0.000117 radians. Note that the gear ratio is different for Joint 2 and 3 which each have a resolution of 0.000137 radians.

Though not specifically specified by Sensable Technologies, the PHANToM's motors appear to be Maxon DC motors [40]. The motors are able to supply a torque of 0.1287 Nm maximum, with a 0.0293 Nm continuous torque limit. A form of software fuse was implemented in order to protect the motors against overheating while still allowing them to reach the maximum torque value.

### 2.4.2 Amplifier Specifications

The amplifiers used in conjunction with the PHANToM robot were Glentek Model GA4555P linear current amplifiers [31]. The auxiliary input channel was used for inputs in the range of +/- 13 Volts. The output signal of the amplifier was a proportional current with a peak current of 12 Amps and an RMS continuous current of 4 Amps.

The amplifier was protected with two safety devices. The first was a low speed electronic circuit breaker. This operated by integration of the output current over time. When this integrated value passes a setting determined by the user, the amplifier automatically switched off. The second device was a DC power fuse. This fuse blew if the output was shorted or excessively loaded (GA4555P Manual) [41].

The DC and frequency responses were tested on each of the amplifiers. These tests were conducted using identical setups. A power resistor was connected to the amplifier output while the input was connected to a voltage source. The voltage over the resistor was monitored with a DAC card and QNX OS.

The DC response test was coupled with setting the amplifier gain. First the output was monitored while inputting zero Volts to the amplifier command. The DC offset potentiometer was adjusted until the corresponding output voltage over the resistor was also zero. One Volt was output to the amplifier and the gain value was set by adjusting the auxiliary gain potentiometer until the desired amount of DC current was obtained on the output. Note that since the reading on the DAC and QNX machine was a voltage over the resistance, the actual current value was scaled by the resistor value. Once the gain value was set, several DC values over the range (+ and -) of the input were fed into the amplifier. The input/output plot can be seen in Figure 2.13 for the first amplifier.

This particular DC gain function shows that the amplifier was not symmetric. The negative gain was less then that of the positive gain, which gives the piecewise linear plot seen in Figure 2.13. This nonlinear DC gain relationship was linearized by simply using a different amplifier constant for multiplication of negative values of the first amplifier.

The frequency response was obtained with the same method as the speaker transfer function. A time-varying signal with a known frequency was input to the amplifier, and the voltage across the load resistor was monitored. The magnitude and phase of the output wave were then compared to that of the input wave. The Bode plots obtained from amplifiers two and four can be seen in Figure 2.14 and 2.15 respectively.

These Bode plots are typical for amplifiers. They start with nearly constant gain and then begin to roll-off at higher frequencies (similar to a low-pass filter). The second plot does have some irregularities at higher frequency components, but as the bandwidth required from the system was well below these frequencies, they did not have a significant effect. Both amplifiers had their roll-off start near 50Hz, and both started with very similar values of dc gain.

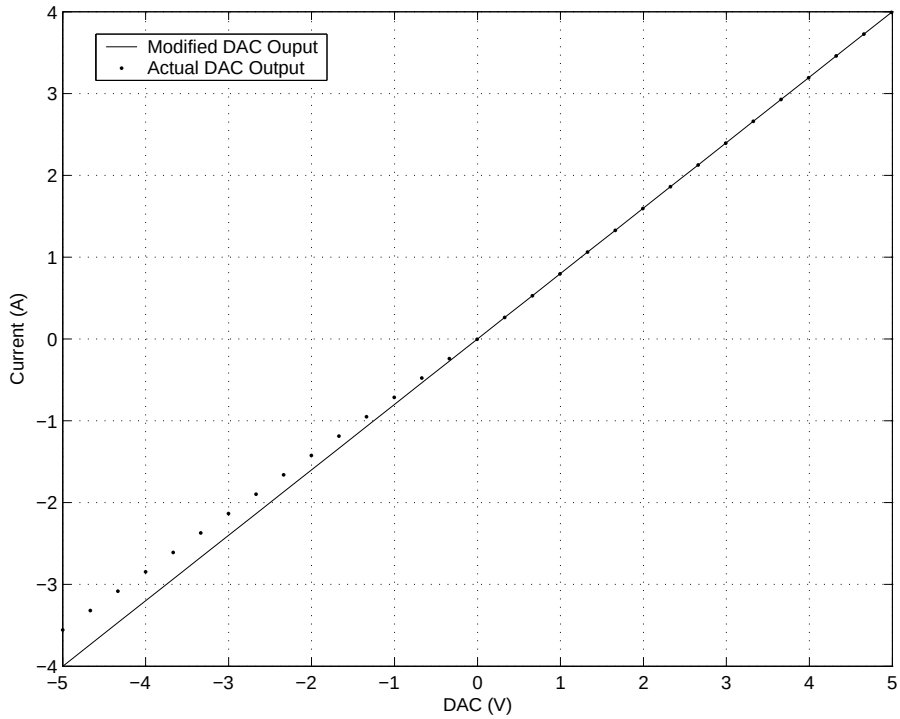



Figure 2.13: DC Amplifier Transfer Function and Correction

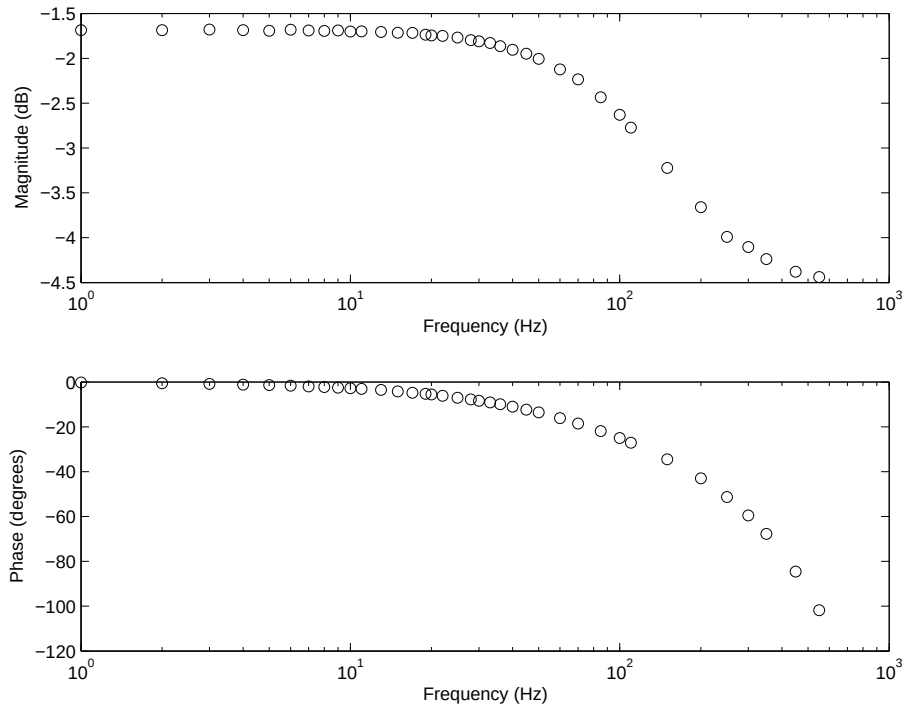



Figure 2.14: Amplifier 2 Bode Plot

### 2.4.3 Modeling and System Identification

Cavusoglu et al. in [42] have done work previously in order to create a model and transfer function for the PHANToM. This was a small-signal model in Cartesian coordinates. Though for very small signals, the joint angles could be approximated as Cartesian displacements, a new model was obtained in joint space utilizing a variation of the frequency-response method mentioned previously. This model was simpler due the fact that performing the tests on a single joint would help reduce the coupling dynamics that were seen between joints. In an attempt to further assure only the dynamics of Joint 1 were seen in testing, Joints 2 and 3 were clamped into position. The robot was clamped in the home position (seen in Figure 2.12).

The major difference between the speaker modeling and the PHANToM modeling was that the PHANToM was not a stable open-loop system. Therefore, using a strictly feedforward term as an input was not an option for this experiment. A proportional-plus-derivative feedback term was added to the input to stabilize the system response about the home position. The input equation then took the form of Eqn 2.7.

$$u = A\sin(\omega t) - k_p y - k_d \dot{y} \tag{2.7}$$

The $\dot{y}$ term was estimated using a first-order backwards-difference approximation. The gains were selected so that the system was stable but would not reject the sinusoidal term. The entire input term was recorded and then fit with a sinusoid. The same was done for the output.

The amplitude of the sinusoidal input term was chosen carefully depending on the frequency. For a linear system, the transfer function should not have any dependence on the magnitude of the input. This means that any amplitude sine wave should produce the same transfer function. At small frequencies the amplitude of the input needed to be small so as not to hit the mechanical stops of the system. However,

at the higher frequencies, a small amplitude is not sufficient to pull out meaningful information. There are simply not enough encoder readings per wave oscillation. To make the waveform fits more accurate, a minimum of 8 different values were desired in order to recreate the wave. Therefore, several data sets were taken at different values of amplitude in an attempt to get a good model of the system. A selection of these sets can be seen in Figure 2.16.

For frequencies under 50Hz the Bode plot was very consistent for different values of amplitude. However at frequencies above the 50Hz dip, the response became somewhat clouded and only a general outline could be obtained.

A complicating nonlinearity is Coulomb friction. Coulomb friction is the frictional force of the motor or gearing that opposes the motion of the robot. When a motor is in motion, this force or torque equivalent is a constant value that alters the effective force on the motor. While the motor is positively accelerating, the friction opposes that accelerating force and the effective force is the commanded force minus the friction term. If the motor is negatively accelerating, the friction term will be added to the commanded force. When the motor is not in motion, the Coulomb friction is equal and opposite to the amount of force supplied. It then follows that, in order for the motor to move, a force greater than Coulomb friction must be applied.

The friction term can be calculated using Newton's second law. The sum of forces for a non-accelerating system must be zero. If the motor is operating at a constant velocity, the only forces acting on it are the commanded force and the Coulomb friction. Therefore (in the absence of Viscous friction), simply recording the torque required to keep the PHANToM moving at a constant speed will give the Coulomb friction constant.

The following test was conducted in order to determine the Coulomb friction term of the PHANToM. The friction term as stated before can be easily measured by commanding a constant velocity to the motor. In order to command a constant velocity,

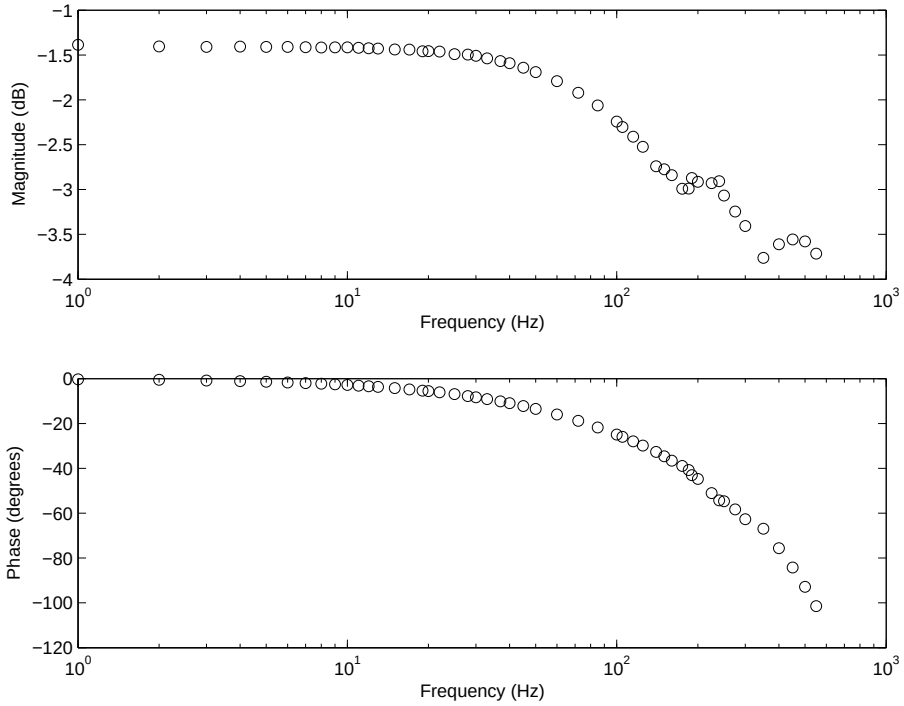



Figure 2.15: Amplifier 4 Bode Plot

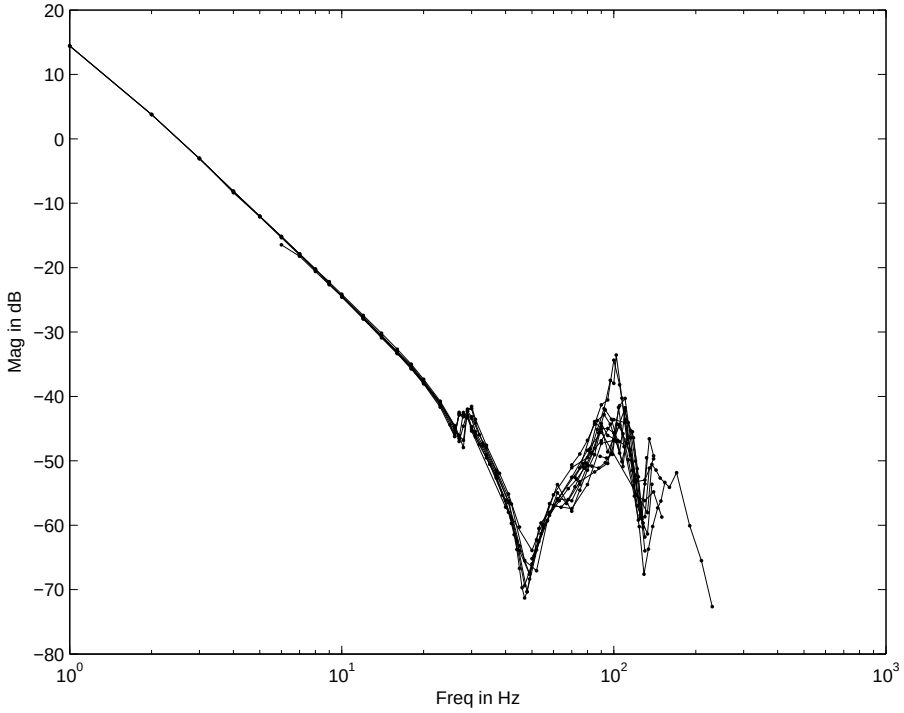



Figure 2.16: Experimentally measured frequency response of Joint 1. Results obtained from using different input magnitudes are superimposed.

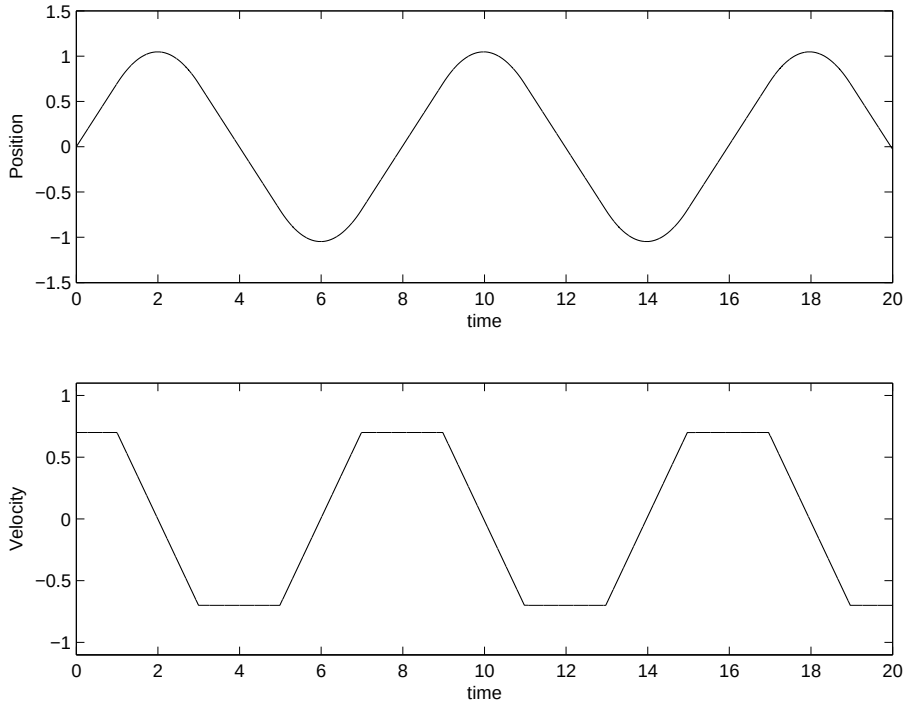



Figure 2.17: Motion trajectory used to measure Coulomb friction value of the PHANToM. The velocity has a trapezoidal profile.

a three-part displacement profile was created. The first and third part contained uniform acceleration from and to zero velocity. These portions ended and began at desired constant velocity that was set in code. After going forward through the profile, the code was set to return to the zero position through the reverse trajectory. This total profile will be described as a trapezoidal profile due to the shape of the velocity versus time curve (see Figure 2.17). The PHANToM was driven to track this motion trajectory using a PD controller.

Throughout the trajectory, the position and torque were recorded. The Coulomb friction term was calculated by taking the average force applied over any one of the constant velocity portions show in Figure 2.18. Averages were calculated for the positive and negative portion to assure accuracy of the experimental value. The particular trapezoidal profile seen for this torque plot had a longer constant velocity cycle with very quick accelerations. The Coulomb friction value was 0.045 Nm. Regardless of

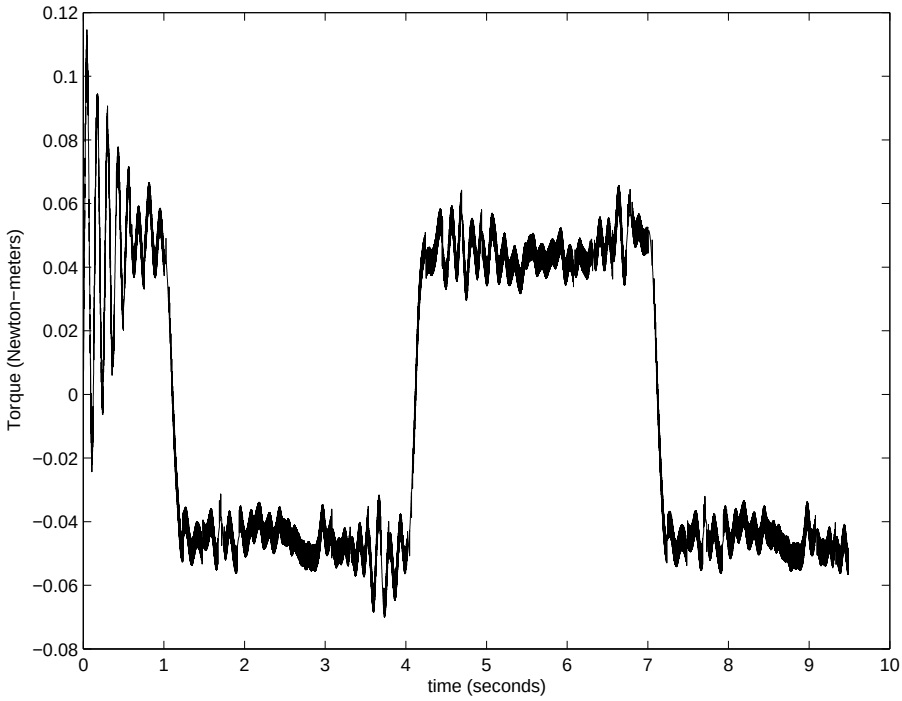



Figure 2.18: Coulomb Friction

the direction and the set speed, the Coulomb friction value should be the same (note again this is true for zero Viscous friction). This was tested by using several different top speeds, and the results were consistent.

If the Coulomb friction term was small enough, it was possible that it could be ignored within the control loop. To ascertain the total effect of this nonlinearity, a simulation was created where the Coulomb friction was included within the plant model. In order to simulate the plant while including the Coulomb friction, it was necessary to alter the control effort applied to the plant. As stated, the Coulomb friction term always opposed motion and thus the sign of the velocity of the plant defines the sign of the Coulomb friction. The sign of the velocity would always be in the direction of motion. In code, subtracting the sign of the velocity multiplied by the magnitude of the Coulomb friction would produce the desired effect for simulation.

On the actual plant, the compensation would be done in reverse. The Coulomb friction would be added to the already calculated control and then applied to the

plant. This would have a linearizing effect on the plant.

The model used for testing was the frequency response model that related input torque to output angle of the Joint 1 actuator. By examining the data from Figure 2.16, the plot appeared to have an immediate drop of -40dB/decade. This was the standard transfer function of a motor or

$$G(s) = \frac{1}{Js^2}, \tag{2.8}$$

where $J$ is the rotational inertia of the motor. With this presumption, a pair of poles at $s = 0$ was forced onto the current data by dividing out a complex $1/s^2$ and then performing the fit. The fit with the $1/s^2$ removed can be seen in Figure

2.19.

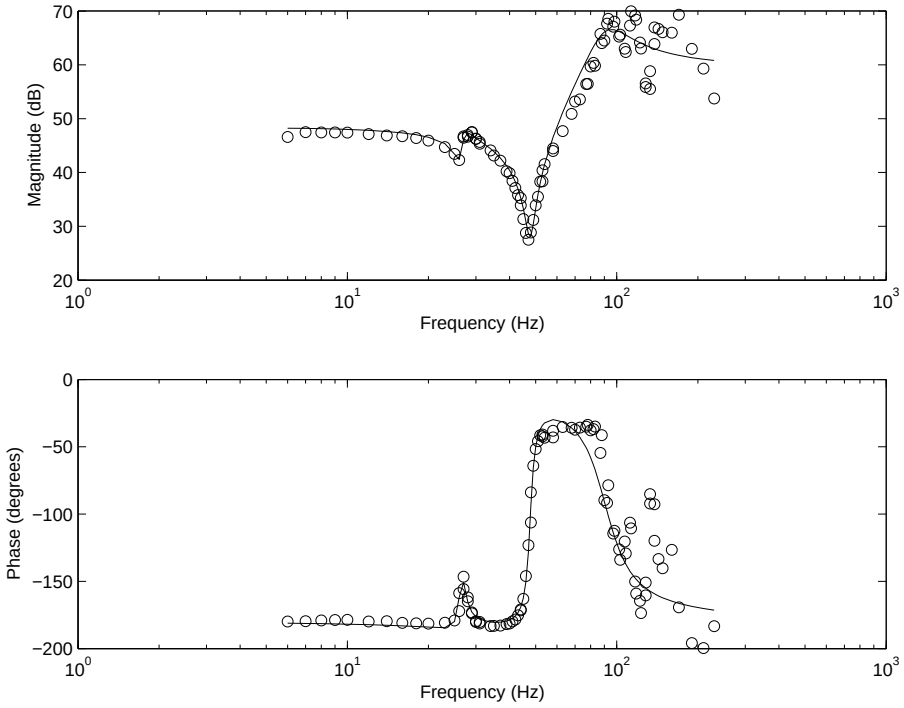



Figure 2.19: Fit to PHANToM Bode Plot without $1/s^2$

After the fit was obtained this model was then multiplied by $1/s^2$ and the Bode plot in Figure 2.20 and the transfer function below resulted.

$$G(s) = \frac{983.6s^4 + 3.037 \times 10^4 s^3 + 1.154 \times 10^8 s^2 + 1.502 \times 10^9 s + 2.402 \times 10^{12}}{s^6 + 214s^5 + 3.544 \times 10^5 s^4 + 1.036 \times 10^7 s^3 + 9.179 \times 10^9 s^2} \tag{2.9}$$

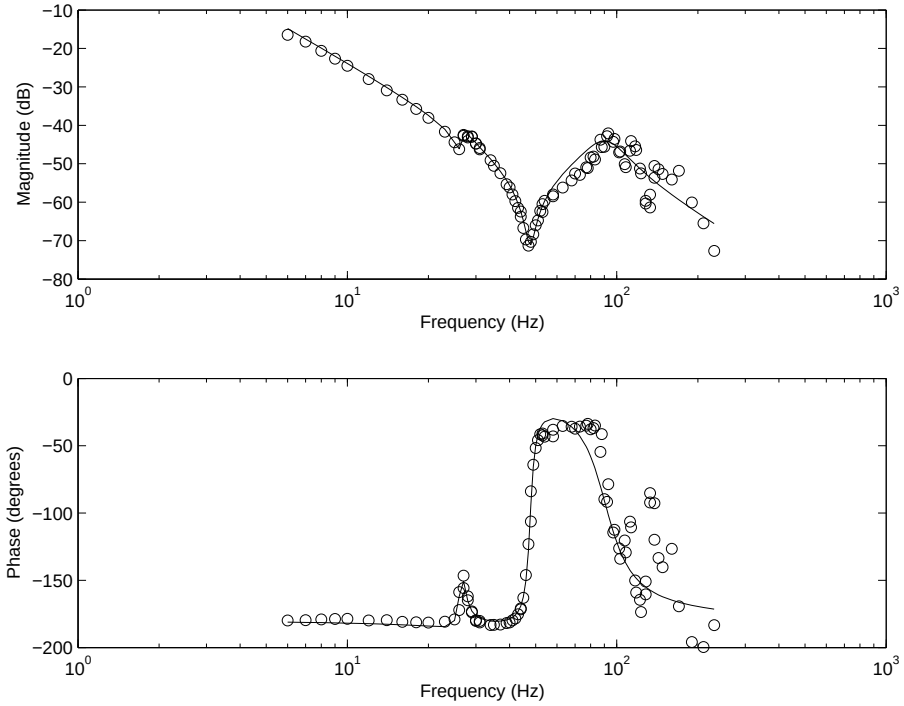



Figure 2.20: Fit to PHANToM Bode Plot with $1/s^2$

Initial testing with this model yielded inadequate results. The controllability matrix of the system was calculated, and it was observed that the system was only weakly controllable. This was causing poor conditioning of the state space matrices and as a result instability and poor control. A reduced realization was obtained using Matlab that produced a 4th-order transfer function, as detailed below.

First, the two poles at $s = 0$ were removed from the transfer function realization. When performing a model reduction such as the Schur reductive method, it is required that the plant model be stable. The Matlab function `schmr()` was then utilized to perform the reduction [43]. This function takes in a system and the number of states to be removed. It calculates the weakly controllable eigenvalues and then uses that

information to eliminate them and produce a new system.

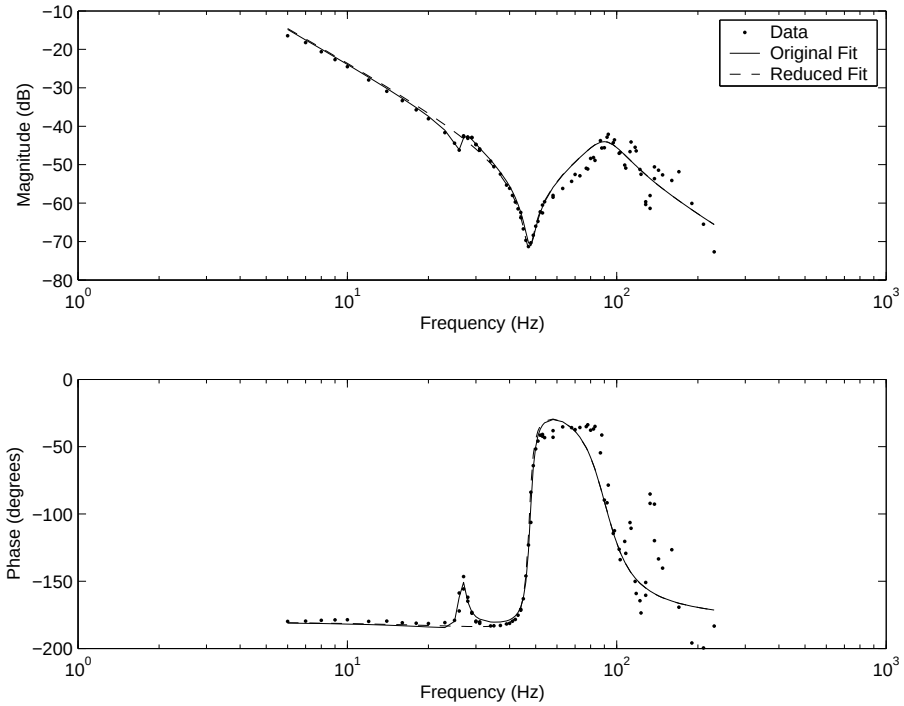



Figure 2.21: Reduced Fit to PHANToM Bode Plot. This plot includes the $1/s^2$ term.

From Figure 2.21, it can be seen that the small spike near 30Hz was causing the issues. The new system was much better conditioned and as a result produced better results. The reduced transfer function is

$$G(s) = \frac{983.6s^2 + 1.784 \times 10^4 s + 8.691 \times 10^7}{s^4 + 200.2s^3 + 3.234 \times 10^5 s^2}. \tag{2.10}$$

This was the model used for control purposes and was subsequently used with the different tracking algorithms.

# Chapter 3

# Control Algorithms

With the system models in place, the next task to perform was to implement and test the different tracking algorithms. In order to understand the operation of each of these algorithms, they are described one by one within this chapter.

This chapter starts by describing some general information about the observer used for the controllers. Subsequent sections include details of each of the tracking algorithms.

The overall setup for each algorithm is as follows. Generally, a desired signal will be present and included in the calculation of a feedforward gain for the algorithm. The feedback gain will be based on the system output. The system to be controlled is characterized by a discrete state-space realization as can be seen in Eqns. 3.1 and 3.2.

## 3.1 Observer Implementation

With the exception of PD control, all control algorithms were implemented using state-space realizations and utilized state feedback. Only positions were directly sensed. An observer was implemented for each of the plant models to obtain the full state vector for state feedback.

As the control algorithms were designed in discrete time using state-space difference equations, the observer was also implemented in discrete time. The predictor form of the observer was implemented [44], taking in the sensor value and calculating the next cycle's state based on the current state, the control effort and the error of the observer output. The observer poles were typically placed to avoid oscillation and in most cases the observer pole values were placed between 0.5 and 0.9. The actual corresponding observer bandwidth varied on an experiment to experiment basis.

The actual gains and corresponding time constants are presented in the results section in Chapter 5 and Section 4.4. See Figure 3.1 for a discrete block diagram of the observer. Note that in this diagram, the system model parameters are Phi, Gam and H corresponding to $\Phi$, $\Gamma$ and $\mathbf{H}$, respectively, in the discrete state-space representation seen in Eqn 3.1 and Eqn 3.2.

$$x[k+1] = \Phi x[k] + \Gamma u[k] \tag{3.1}$$

$$y[k] = \mathbf{H}x[k] \tag{3.2}$$

For an $n$th order system, the dimensions of of $x$ and $y$ are $n \times 1$ and $1 \times 1$ respectively.

## 3.2 Position Plus Derivative Control

The classical control method, position-plus-derivative (PD), was the first of the four control algorithms tested. The control effort was calculated according to Eqn 3.3.

$$u = k_p(y_{des} - y_{act}) - k_d\dot{y}_{act} \tag{3.3}$$

The control effort is characterized in Eqn 3.3 by $u$. The output of the system is $y_{act}$ and the desired output is $y_{des}$. The position and derivative gains are notated by $k_p$ and $k_d$ respectively. A continuous block diagram representation can be seen in Figure

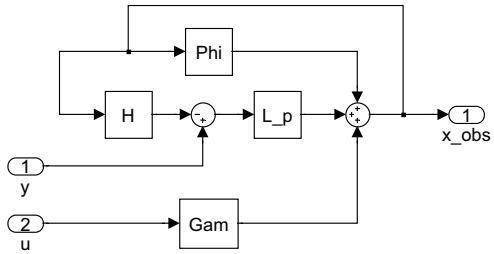



Figure 3.1: Observer Block Diagram

3.2.

Since the speed was not available through a sensor from the plant, it was calculated using the plant state-space model. The state-space plant model is represented using **A**, **B**, and **C** matrices. The **A** and **B** matrices relate the states and the control respectively to the derivative of the states. The **C** matrix relates the state to the system output. Note that all of the systems were strictly proper or forced to be strictly proper and hence did not contain a **D** term.

$$\dot{x} = \mathbf{A}x + \mathbf{B}u \tag{3.4}$$

$$y = \mathbf{C}x \tag{3.5}$$

$$\dot{y} = \mathbf{C}\dot{x} = \mathbf{C}(\mathbf{A}x + \mathbf{B}u) \tag{3.6}$$

By taking a time derivative of the output equation, the output speed can be related to the state speeds. For a given state and control effort, their derivatives were

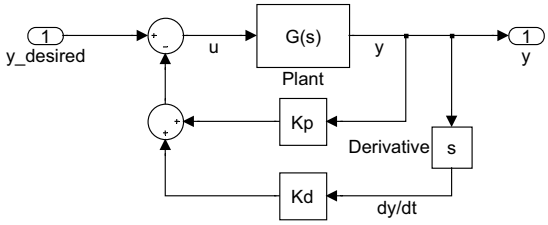



Figure 3.2: Continuous PD Controlled Block Diagram

known from the state equation.

The states maintained their values and meaning through the continuous-to-discrete transformation. It was necessary to obtain the discrete state-space realization from the continuous time state-space realization. Changing realizations would give different states and the speed calculation method would break down. The total process summed up to calculating the state using a discrete observer and then plugging that state into the continuous state-space functions. By obtaining the speed in this form, the noise that resulted from taking the first-order approximation to calculate the speed, diminished. Effectively, it was a filtered result.

The gains for this algorithm were tuned instead of calculated. The process of tuning the gains started by simply increasing the position gain until the ringing in the system was the primary cause of tracking error. At this point, the derivative gain was set such that the ringing was reduced. These two steps were repeated until adequate tracking was obtained or the system was on the verge of instability.

The gains were fine tuned by looking at the RMS control and RMS error between the desired and actual position. See the results section for the details of this tuning.

## 3.3 Pole-Placement Control

PD control can be seen as a variation of pole-placement control. In general, pole placement has $m$ degrees of freedom to do control (where $m$ is the order of the system). In other words, it is possible to move all the poles of the system to any specific locations in the unit circle. PD control, however, is limited to two degrees of freedom. For systems of order $m > 2$, PD is not exercising all the degrees of freedom possible to do control. PD control will often find an acceptable set of poles, but it is likely that the controller is suboptimal.

The pole-placement algorithm utilized state feedback with the states obtained by the observer. The poles were placed using the discrete plant transfer function and Ackerman's formula. (used by way of the Matlab function `acker()`).

The feedforward gain of the algorithm was calculated by extracting states that corresponded to the desired trajectory. These desired states are compared to the actual states in feedback. Both of these gains are calculated from the system model.

The pole-placement system block diagram can be seen in Figure 3.3. Note that this algorithm was taken directly from [44].

Steady-state error will occur with this algorithm for systems of type 0 or those systems that contain no poles at $s = 0$. The type of the system is defined as the number of poles at zero. Therefore, a feedforward steady-state term was added to the basic PP controller. To get the desired states, a vector exists that when multiplied by the desired signal, will yield an approximation of the states.

$$\mathbf{N}y_{des} = x_{des} \tag{3.7}$$

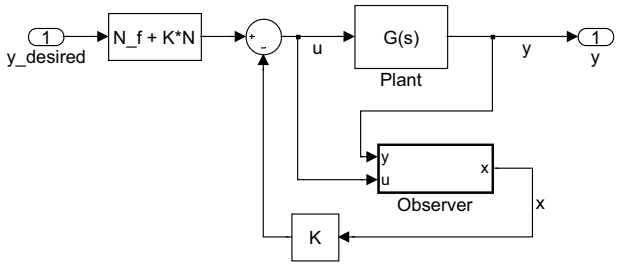



Figure 3.3: Pole-Placement Controlled Block Diagram

The output is determined by the states from

$$y_{act} = \mathbf{H}x. \tag{3.8}$$

For tracking, the desired signal should be identical to the actual and hence the desired states should be the same as the actual states. By making a substitution from Eqn 3.7 to Eqn 3.8, it can be seen that $\mathbf{N}$ is actually a pseudo-inverse of $\mathbf{H}$.

$$y_{act} = \mathbf{HN}y_{des} \tag{3.9}$$

$$\mathbf{HN} = \mathbf{I} \tag{3.10}$$

The steady-state term is defined by

$$u_{ss} = \mathbf{N}_f y_{des}. \tag{3.11}$$

When in steady state, the control should be tracking and hence the component from the state feedback will be completely eliminated by the state feedforward portion. This makes the state equation for steady state

$$x_{ss} = \Phi x_{ss} + \Gamma u_{ss}. \tag{3.12}$$

The $\Phi$ and $\Gamma$ matrices are the discrete state-space equivalent of the $\mathbf{A}$ and $\mathbf{B}$ matrices (respectively).

By making substitutions into the steady-state equation,

$$(\Phi - \mathbf{I})\mathbf{N} + \Gamma \mathbf{N}_f = \mathbf{0}. \tag{3.13}$$

Using Eqn 3.13 in conjunction with Eqn 3.10, a system of equations is constructed from which $\mathbf{N}$ and $\mathbf{N}_f$ can be obtained.

$$\begin{pmatrix} \mathbf{N} \\ \mathbf{N}_f \end{pmatrix} = \begin{pmatrix} \Phi - \mathbf{I} & \Gamma \\ \mathbf{H} & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \tag{3.14}$$

Once $\mathbf{N}_f$ and $\mathbf{N}$ have been solved, the feedfoward control is simply

$$u_{ff} = \mathbf{N}_f y_{des} + \mathbf{K}\mathbf{N} y_{des} \tag{3.15}$$

The matrix $\mathbf{K}$ is the set of feedback gains that are acquired through the `acker()` function. The complete control equation takes the form of

$$u = (\mathbf{N}_f + \mathbf{K}\mathbf{N}) y_{des} - \mathbf{K} x_{obs} \tag{3.16}$$

## 3.4 Model Predictive Control

Model predictive control (MPC) is an acausal algorithm used for trajectory tracking. The MPC algorithm uses the system model to predict future outputs. The future outputs are compared to a desired reference signal and used to calculate gains. Note that generally the reference signal is known a priori. MPC has been successful because of its ability to be applied to a number of different applications. Several variations have been developed and implemented to the extent where MPC has become a family of algorithms. However, all of the algorithms contain three characteristics: a model is utilized to predict future outputs of a system, the control sequence is calculated based on minimization of a cost function and the gains are calculated utilizing a receding horizon where the furthest point ahead considered moves one step ahead for every control cycle [45].

MPC has a wide range of industrial applications. Clark conducted studies using MPC to control a cement grinding mill, a spray drying tower and a compliant robotic arm [46]. He concluded that MPC is an effective method to use for all types of plants. The typical applications for MPC have been industrial processes (for a short history of the MPC algorithm see [45]).

As in the other tracking algorithms, there exist two parts to the algorithm, a feedback term and a feedforward term. These terms can be calculated separately from one another, yet depend on many of the same parameters. The following MPC algorithm was obtained from [47].

The block diagram for this controller shown in Figure 3.4, resembles that of the pole-placement algorithm. The major difference between the two is in the complexity of calculating the gains.

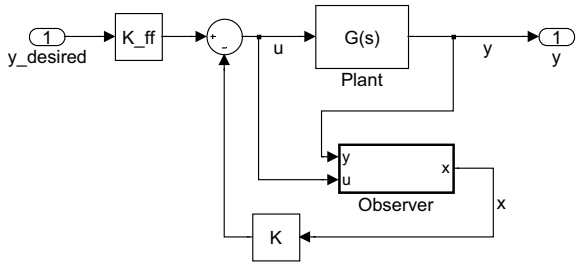



Figure 3.4: Coarse Block Diagram of MPC Control

## Feedback

To help better explain the feedback portion of MPC, optimal control (or linear quadratic regulation) will first be explained. Optimal control can be seen as a subclass of MPC. An optimal controller is a model predictive controller where the desired trajectory is zero. The idea behind the optimal controller is to find a control effort that will minimize a cost index equation. This equation takes the form of a quadratic in the control effort and in the state vector of the system. The goal for this type of control is to minimize Eqn 3.17 with respect to the control $u[k]$.

$$J^*[k] = \min_{u[k]}(x[k+1]\mathbf{Q}x[k+1] + u[k]\mathbf{R}u[k] + J^*[k+1]) \tag{3.17}$$

$J^*[n]$ is the optimal index at time $n$. $\mathbf{Q}$ and $\mathbf{R}$ are matrix weighting parameters. By altering the ratio between $\mathbf{Q}$ and $\mathbf{R}$, the emphasis of the optimization problem is shifted. Using a higher $\mathbf{Q}$ to $\mathbf{R}$ ratio will accentuate the state and hence regulate

more quickly. Using a lower $\mathbf{Q}$ to $\mathbf{R}$ ratio penalizes higher control values, so the regulation is slower but uses a smaller control effort. The $\mathbf{Q}$ and $\mathbf{R}$ matrices should be positive semi-definite and positive definite, respectively. Often for simplicity, the matrices are created as multiples of the identity matrix. For systems with many states and multiple inputs, this simplifies the tuning process to two parameters. However, it is possible to weight more heavily different indices of each matrix in an attempt to penalize particular states or inputs.

For any regulation problem the actual control equation takes the form

$$u[k] = \mathbf{K}[k]x[k]. \tag{3.18}$$

The optimal index equation can be written in terms of a quadratic of the state and the next optimal index by substituting in the control equation.

Recall also that this representation is discrete so the state-space equation is a difference equation. More directly this means, an equation for $x[k+1]$ exists that is written in terms of $x[k]$ and $u[k]$. By making this substitution, the cost equation can now be written completely in terms of the current state, $x[k]$ and the next cost index. Note that the $u[k]$ term is again eliminated through the control equation.

$$J^*[k] = \tag{3.19}$$
$$(\Phi x[k] + \Gamma \mathbf{K}[k]x[k])'\mathbf{Q}(\Phi x[k] + \Gamma \mathbf{K}[k]x[k]) + (\mathbf{K}[k]x[k])'\mathbf{R}(\mathbf{K}[k]x[k]) + J^*[k+1]$$

In an attempt to solve Eqn 3.19, Eqn 3.20 is assumed to be a solution.

$$J^*[k] = x[k]'\mathbf{P}[k]x[k] \tag{3.20}$$

Upon plugging in Eqn 3.20, nearly every term of Eqn 3.19 is a quadratic in terms of the current value of the state. By repeating the two substitutions used to create

Eqn 3.19, the $J^*[k+1]$ term can be written in terms of the current state as is seen in Eqn 3.21.

$$\begin{aligned} J^*[k+1] &= x[k+1]'\mathbf{P}[k+1]x[k+1] \\ &= (\Phi x[k] + \Gamma u[k])'\mathbf{P}[k+1](\Phi x[k] + \Gamma u[k]) \\ &= (\Phi x[k] + \Gamma \mathbf{K} x[k])'\mathbf{P}[k+1](\Phi x[k] + \Gamma \mathbf{K} x[k]) \end{aligned} \tag{3.21}$$

When eliminating the state from both sides of the equation, it can be seen that $\mathbf{P}[k]$ is actually a solution to a difference matrix Riccatti equation. This Riccatti equation can be solved by backwards iteration where the final value is $\mathbf{P}[T] = \mathbf{0}$. All the parameters in Eqn 3.19 are known except $\mathbf{K}$. It is necessary to solve algebraically for this gain in order to solve numerically the Riccatti equation.

The optimal gain $\mathbf{K}$ is solved for by taking the derivative of $J[k]$ with respect to the control effort $u[k]$ and setting it equal to zero. In order to take the derivative of the $J[k+1]$ term, it is replaced by $x[k+1]'\mathbf{P}[k+1]x[k+1]$ and then subsequently the $x[k+1]$ is exchanged with the state equation. Solving for $u[k]$ then produces Eqn 3.22.

$$u[k] = -(\mathbf{R} + \Gamma'(\mathbf{Q} + \mathbf{P}[k+1])\Gamma)^{-1}(\Gamma'(\mathbf{Q} + \mathbf{P}[k+1])\Phi)x[k] \tag{3.22}$$

$$\mathbf{K}[k] = -(\mathbf{R} + \Gamma'(\mathbf{Q} + \mathbf{P}[k+1])\Gamma)^{-1}(\Gamma'(\mathbf{Q} + \mathbf{P}[k+1])\Phi) \tag{3.23}$$

This derivation shows the optimal gain as a function of time. The Riccatti equation is now

$$\begin{aligned} \mathbf{P}[k] = \Phi'\{&\mathbf{Q} + \mathbf{P}[k+1] - \\ &(\mathbf{Q} + \mathbf{P}[k+1])\Gamma[\mathbf{R} + \Gamma'(\mathbf{Q} + \mathbf{P}[k+1])\Gamma]^{-1}\Gamma'(\mathbf{Q} + \mathbf{P}[k+1])\}\Phi \end{aligned} \tag{3.24}$$

This Riccatti equation is dependent only on the system model and the weighting

matrices $\mathbf{Q}$ and $\mathbf{R}$. As stated before, the value of the Riccatti equation at the horizon value ($\mathbf{P}[T]$) is equal to $\mathbf{0}$. This equation is independent of the system state/output value and the control value. This means that the gain is based only on static matrices and iterative parameters. The equation (and in turn the gains) can be solved before control is exercised on the plant using backwards iteration. The iteration is backwards in the sense that the starting point of the iteration is some horizon into the future and the calculation occurs backwards in time to the present.

## Feedforward

The feedforward portion of the MPC algorithm again is what separates MPC from optimal control.

An auxiliary system is defined for the desired input signal such that it possesses the same states as the plant. The relationship between this desired state and the desired output can be described with a difference equation and an output equation.

$$\begin{aligned} x_{des}[k+1] &= \mathbf{F}x_{des}[k] \\ y_{des}[k] &= \mathbf{G}x_{des}[k] \end{aligned} \tag{3.25}$$

For tracking purposes, the cost index should attempt to minimize the error between the newly defined desired state and actual system state. The equation would hence take the form

$$J^*[k] = \min_{u[k]}((x[k+1] - x_{des}[k+1])^T\mathbf{Q}(x[k+1] - x_{des}[k+1]) + u[k]\mathbf{R}u[k] + J^*[k+1]) \tag{3.26}$$

This form can be forced into the original form (Eqn 3.17) by creating a new state vector and carefully choosing the weighting parameter $\mathbf{Q}$.

A new state vector is defined that augments the current state vector with the

states of the auxiliary system. The new state vector now looks like

$$x_{new} = \begin{pmatrix} x \\ \\ x_{des} \end{pmatrix}. \tag{3.27}$$

By choosing $\mathbf{Q}$ as seen below, the cost index equation can attempt to regulate $x - x_{des}$ and mimic the form of the regulator problem which already has a solution. Note the subscripts are dropped for simplification. The matrix $\mathbf{Q}$ is originally an $n \times n$ matrix where $n$ is the order of the system. The augmented system is now $2n \times 1$ and hence $\widetilde{\mathbf{Q}}$ is a $2n \times 2n$ matrix.

$$(x - x_{des})^T \mathbf{Q}(x - x_{des}) = x^T \mathbf{Q} x - x^T \mathbf{Q} x_{des} - x_{des}^T \mathbf{Q} x + x_{des}^T \mathbf{Q} x_{des} \tag{3.28}$$

$$\widetilde{\mathbf{Q}} = \begin{pmatrix} \widetilde{q}_{11} & \widetilde{q}_{21} \\ \\ \widetilde{q}_{12} & \widetilde{q}_{22} \end{pmatrix} \tag{3.29}$$

$$\begin{pmatrix} x^T & x_{des}^T \end{pmatrix} \widetilde{\mathbf{Q}} \begin{pmatrix} x \\ \\ x_{des} \end{pmatrix} = x^T \widetilde{q}_{11} x + x^T \widetilde{q}_{21} x_{des} + x_{des}^T \widetilde{q}_{12} x + x_{des}^T \widetilde{q}_{22} x_{des} \tag{3.30}$$

$$\widetilde{\mathbf{Q}} = \begin{pmatrix} \mathbf{Q} & -\mathbf{Q} \\ \\ -\mathbf{Q} & \mathbf{Q} \end{pmatrix} \tag{3.31}$$

Since the state has been augmented, it will be necessary to define a new state equation. Using Eqn 3.25, the below augmented state-space is obtained.

$$x_{new}[k+1] = \widetilde{\Phi} x_{new}[k] + \widetilde{\Gamma} u[k] \tag{3.32}$$

$$\widetilde{\Phi} = \begin{pmatrix} \Phi & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{pmatrix}; \widetilde{\Gamma} = \begin{pmatrix} \Gamma \\ \mathbf{0} \end{pmatrix} \tag{3.33}$$

At this point the tracking problem can be treated as the regulator problem. The control effort equation will also retain the form of $u[k] = \widetilde{\mathbf{K}}[k]x[k]$, where $x[k]$ is now the augmented state. The gain $\widetilde{\mathbf{K}}[k]$ can be written as seen below.

$$\widetilde{\mathbf{K}}[k] = -(\widetilde{\Gamma}'(\widetilde{\mathbf{P}}[k+1] + \widetilde{\mathbf{Q}})\widetilde{\Gamma} + \mathbf{R})^{-1}\widetilde{\Gamma}'(\widetilde{\mathbf{P}}[k+1] + \widetilde{\mathbf{Q}}) \tag{3.34}$$

The $\widetilde{\mathbf{P}}[k]$ is the Riccatti equation parameter. For simplification the $\widetilde{\mathbf{Q}}$ weighting matrix will be combined with $\widetilde{\mathbf{P}}[k]$ to form a new Riccatti parameter which is defined by the equation below.

$$\mathbf{S}[k] = \widetilde{\mathbf{P}}[k] + \widetilde{\mathbf{Q}} \tag{3.35}$$

$$\mathbf{S}[k] = \widetilde{\Phi}'(\mathbf{S}[k+1] - \mathbf{S}[k+1]\widetilde{\Gamma}(\mathbf{R} + \widetilde{\Gamma}'\mathbf{S}[k+1]\widetilde{\Gamma})^{-1}\widetilde{\Gamma}'\mathbf{S}[k+1])\widetilde{\Phi} + \widetilde{\mathbf{Q}} \tag{3.36}$$

The control effort equation can be broken down into block form using 4 terms to make up the Riccatti parameter and expanding each of the state-space terms.

$$\mathbf{S}[k] = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \tag{3.37}$$

Note that all of the matrix blocks within $\mathbf{S}$ are functions of $k$ but have been written with subscripts alone. Then,

$$\widetilde{\mathbf{K}}[k] = \tag{3.38}$$
$$-\left[\begin{pmatrix} \Gamma & \mathbf{0} \end{pmatrix}\begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}\begin{pmatrix} \Gamma \\ \mathbf{0} \end{pmatrix} + \mathbf{R}\right]^{-1}\begin{pmatrix} \Gamma & \mathbf{0} \end{pmatrix}\begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}\begin{pmatrix} \Phi & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{pmatrix}$$

By multiplying out the matrices, Eqn 3.38 is simplified into the equation below.

$$\widetilde{\mathbf{K}}[k] = -(\Gamma' S_{11}\Gamma + \mathbf{R})^{-1} \begin{pmatrix} \Gamma' S_{11}\Phi & \Gamma' S_{12}\mathbf{F} \end{pmatrix} \tag{3.39}$$

*Note: $S_{xx}$ for Eqn 3.38 and Eqn 3.39 are all functions of $k+1$.*

$S_{11}$ and $S_{12}$ need to be calculated in order to find the optimal gains. As was done for the gain matrix in Eqn 3.39, the Riccatti equation can also be expanded and simplified. The results of this simplification can be seen below.

$$\mathbf{S}[k] = \widetilde{\mathbf{Q}} + \tag{3.40}$$
$$\begin{pmatrix} \Phi'(S_{11} - S_{11}\Gamma[\Gamma' S_{11}\Gamma + \mathbf{R}]^{-1}\Gamma' S_{11})\Phi & \Phi'(S_{12} - S_{11}\Gamma[\Gamma' S_{11}\Gamma + \mathbf{R}]^{-1}\Gamma' S_{12})\mathbf{F} \\ \mathbf{F}'(S_{21} - S_{11}\Gamma[\Gamma' S_{11}\Gamma + \mathbf{R}]^{-1}\Gamma' S_{21})\Phi & \mathbf{F}'(S_{22} - S_{21}\Gamma[\Gamma' S_{11}\Gamma + \mathbf{R}]^{-1}\Gamma' S_{21})\mathbf{F} \end{pmatrix}$$

*Note: $S_{xx}$ on the right hand side of Eqn 3.40 is a function of $k+1$.*

By examining index (1,1) of Eqn 3.40, a quick substitution back to $\mathbf{P}$ makes the equation identical to the optimal control Riccatti equation, which was already calculated (Eqn 3.25). However, the gain matrix $\widetilde{\mathbf{K}}$ is still dependent on $S_{12}$ and $\mathbf{F}$, which have not been defined or derived.

Fortunately, both of this unknown parameters can be eliminated with a single substitution. A new parameter $\mathbf{M}$ is defined as

$$\mathbf{M}[k] \triangleq S_{12}[k] x_{des}[k] \tag{3.41}$$

A simple multiplication of $x_{des}$ will accomplish the task of obtaining the $\mathbf{M}$ term within the $S_{12}$ equation.

$$S_{12}[k] = \Phi'(S_{12}[k+1] - \tag{3.42}$$
$$S_{11}[k+1]\Gamma[\Gamma' S_{11}[k+1]\Gamma + \mathbf{R}]^{-1}\Gamma' S_{12}[k+1])\mathbf{F} - \mathbf{Q}$$

$$S_{12}[k]x_{des}[k] = \Phi'(S_{12}[k+1]\mathbf{F}x_{des}[k]- \tag{3.43}$$
$$S_{11}[k+1]\Gamma[\Gamma'S_{11}[k+1]\Gamma + \mathbf{R}]^{-1}\Gamma'S_{12}[k+1]\mathbf{F}x_{des}[k]) - \mathbf{Q}x_{des}[k]$$
$$\mathbf{M}[k] = \Phi'(\mathbf{M}[k+1]- \tag{3.44}$$
$$S_{11}[k+1]\Gamma[\Gamma'S_{11}[k+1]\Gamma + \mathbf{R}]^{-1}\Gamma'\mathbf{M}[k+1]) - \mathbf{Q}x_{des}[k]$$

The only unknown parameter of Eqn 3.44 at this point is the relationship between the desired state and desired output. This relationship can be obtained by taking a pseudo-inverse of the output equation.

$$y = \mathbf{H}x$$
$$x = \mathbf{L}y$$
$$\mathbf{L} = \mathbf{H}'(\mathbf{H}\mathbf{H}')^{-1} \tag{3.45}$$
$$x_{des} = \mathbf{L}y_{des} \tag{3.46}$$

Eqn 3.44 can now be calculated iteratively in the same way as the feedback Riccatti equation using the final condition $\mathbf{M}[T] = \mathbf{0}$. The final condition for $S_{11}[T]$ is equal to $\mathbf{Q}$.

The parameter describing the number of iterations used to calculate $\mathbf{M}$ and $S_{11}$ is called the horizon value. Every iteration corresponds to one control cycle set of gains. In effect, calculating $T$ iterations is like calculating time-varying gains up to $T$ steps ahead even though the only gain that is used is one for the current time value. This type of control is also known as receding horizon control [45]. With every new control cycle, a new point on the desired signal is used and an old point is dropped in the gain calculation. The calculation is meant to be redone for every control cycle.

In the feedback Riccatti equation, the iteratively calculated matrices are not dependent on any time-varying values save the previous iteration matrix. If the iteration is executed every control cycle, the numerical results will be identical. As a result,

the feedback gains are not time dependent and instead are constant values for a given horizon. Therefore the horizon can be set for this calculation and the iteration can be carried out off line, before attempting any control. Hence the feedback gains are known before the control ever starts and do not need to be calculated for every control cycle.

The feedforward Riccatti equation however is dependent on a time dependent input signal (in this case the desired heart trajectory). As a result these gains must be iteratively calculated on the fly every control cycle. Furthermore, it is necessary to save the $S_{11}$ matrix values for each time increment up to the horizon as they are needed for the feedforward calculation.

The horizon along with the weighting matrices are the parameters that can be used to tune the algorithm. Though rather intuitive to tune, altering the horizon does make a difference in the results. A larger horizon generally results in more accuracy of the feedfoward term (primarily because of greater foresight into the future and more iterations to calculate gains). Because of the larger number of iterations, the calculations now take longer. Therefore, a horizon must be chosen such that the gains can be iteratively calculated within one cycle of the control loop. If the gains require more calculation time than the length of the control-loop cycle, the control effort value will be emitted not precisely at the end of the cycle and the next gain calculation will start late. This inadvertently changes the sample rate of the control loop and throws off the control algorithm calculation which causes instability in the system.

This MPC algorithm can handle time-varying systems and weighting matrices. For the applications used herein, the problem was not particularly complicated. Constant weighting matrices were used along with constant state-space models. The only true time-varying gain within the algorithm is the feedforward term, which is calculated from the heart-beat data.

## 3.5 Signal Estimated Model Predictive Control

The particular MPC algorithm described in the previous section contains one flaw. The algorithm requires the future of the trajectory signal to calculate its feedforward gains. For some processes, this is not a problem as the desired trajectory is known before the control ever takes place. However, this is not the case for the heart tracking problem. The MPC algorithm has high enough precision to perform the necessary tracking if the signal is known. As a result, the problem is reduced to effectively predicting the future desired signal.

The heartbeat is a quasiperiodic signal that varies slightly from beat to beat in actual motion (disregarding arrhythmias and other irregularities). If one cycle of the heartbeat is known, it can be used as an estimate of the next cycle.

Every time a value is read in from the heart position, that value can be approximated forward one cycle as long as the period is known. One cycle of heart motion ahead is more then enough information to perform tracking with the MPC algorithm. The problem however is still that the hypothesized cycle ahead is not near enough to the actual signal. An attempt to control using MPC with a one heartbeat cycle guess ahead has been attempted in Figure 3.5. The error is unacceptably large.

However, comparing the current cycle to the previous delayed cycle shows that the major features of the waves do in fact line up. Hence, aside from small differences in the actual signal, the largest error can be attributed to an offset, which is due to the breathing motion within the data. This offset is just the error between the current heart position and the guessed heart position for values in the near future. For values in the distant future, the current error is probably not a good estimate of the future error. Therefore, only a percentage of the current error is added to distant future values of the estimated signal. In Eqn 3.47, $m$ is the number of steps ahead that the signal is calculated, while $k$ is the current time. The function $f(m)$ is the percentage

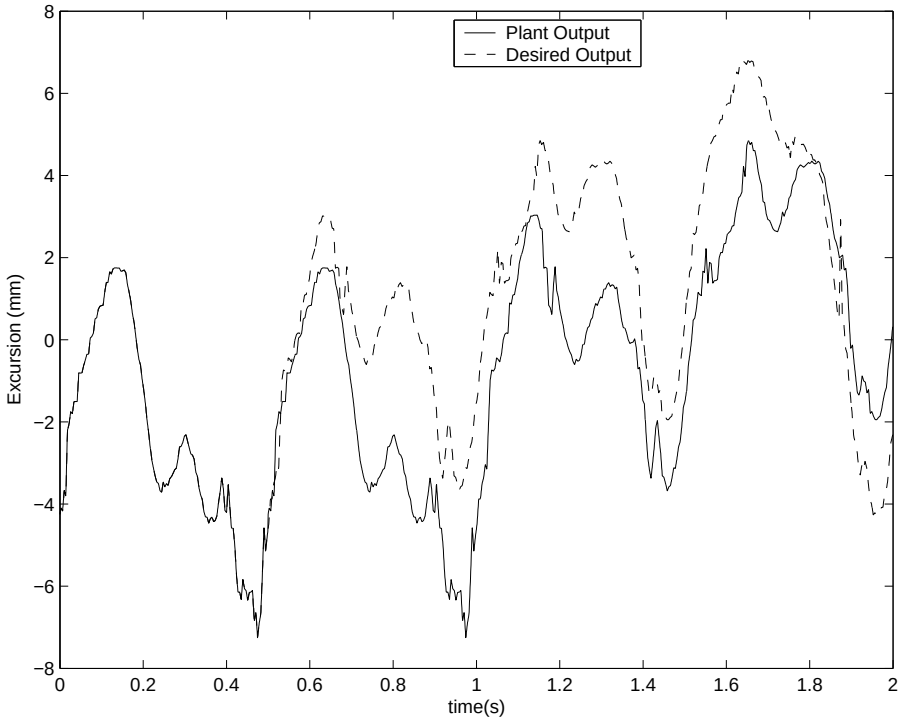



Figure 3.5: Attempting MPC control with a one cycle delay prediction after waiting for one cycle

of the error to be added.

$$y_{est}[k+m] = y_{est}[k+m] + f(m)y_{error}[k] \tag{3.47}$$

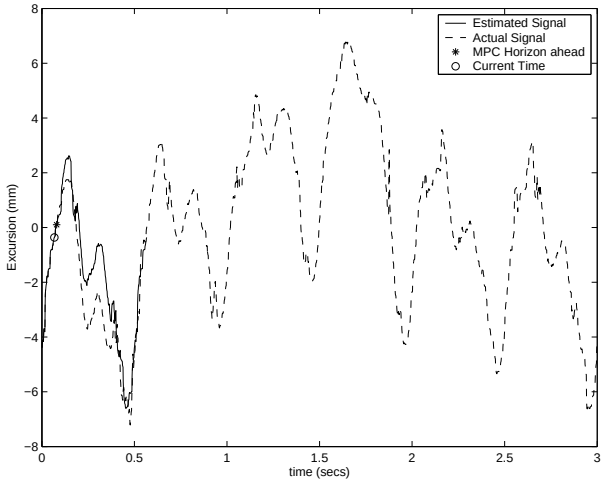
This calcualtion is carried out for all indices up until the $k+M$ index is reached where $M$ is the error horizon value.

The percentage can take the form of a function with respect to the number of error horizon steps ahead. Several different profiles of percentages were attempted for different error horizons. The profile needed to slope down to zero percent in order to avoid discontinuities in the guessed signal. The reason for the percentage to fall to zero is as follows. The estimated signal is being constantly updated. If 100% of the error were added to the entire signal, the new signal would be offset from the old signal. On the next control cycle, the estimated signal would be extended forward

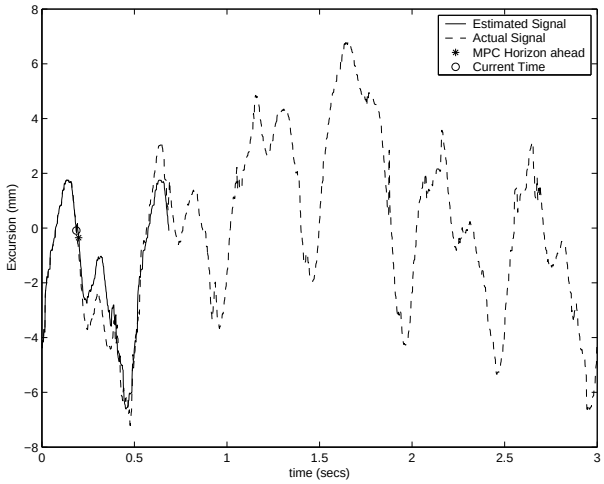
in time by augmenting the end of estimated signal with the current actual signal (one heartbeat cycle difference). This augmented point would not include the offset additions from the previous cycle and hence a discontinuity would be introduced.

By adding a smaller percentage of error offset for the distant-future points, the discontinuity between the last and next-to-last point becomes smaller, and extra noise is not introduced into the estimated signal. Several different percentage profiles were attempted in code and will be reported on in the simulation results section (Section 4.3.4). The signal estimated MPC not only takes care of the offset issue that is present from cycle to cycle but also forces the estimated signal to converge onto the actual signal for a small horizon ahead.

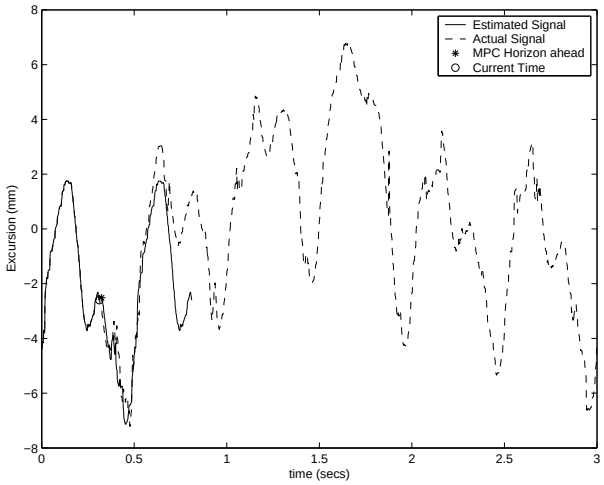
In figure 3.5, the actual signal and the estimated signal can be seen as the control executes. The current time for which the control is being executed, is noted by the circle in the plot. The asterisk shows the final horizon value that is being used for the MPC controller. Ideally the estimated signal should be exactly on top of the actual signal but the estimate is not very accurate for distant values into the future. For near future values (those which are within the MPC horizon), the horizon or asterisk in the plot needs to be on top of the actual signal line in order for the MPC to reliably operate.


Estimated Signal
Actual Signal
MPC Horizon ahead
Current Time
Excursion (mm)
time (secs)

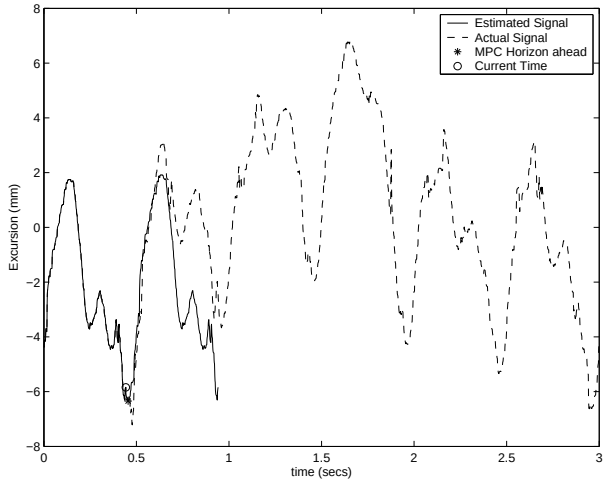
(a)


Estimated Signal
Actual Signal
MPC Horizon ahead
Current Time
Excursion (mm)
time (secs)

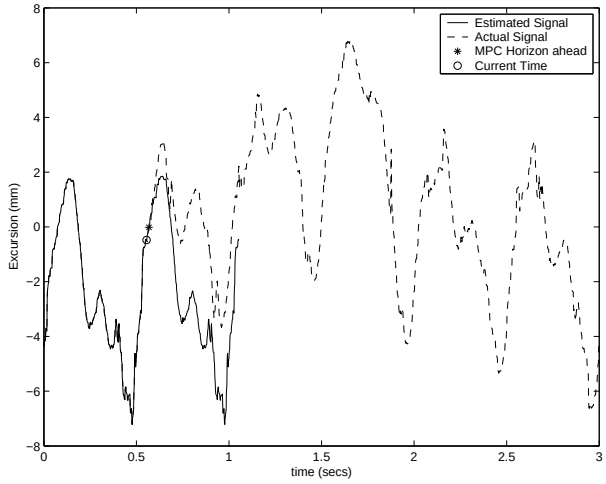
(b)


Estimated Signal
Actual Signal
MPC Horizon ahead
Current Time
Excursion (mm)
time (secs)

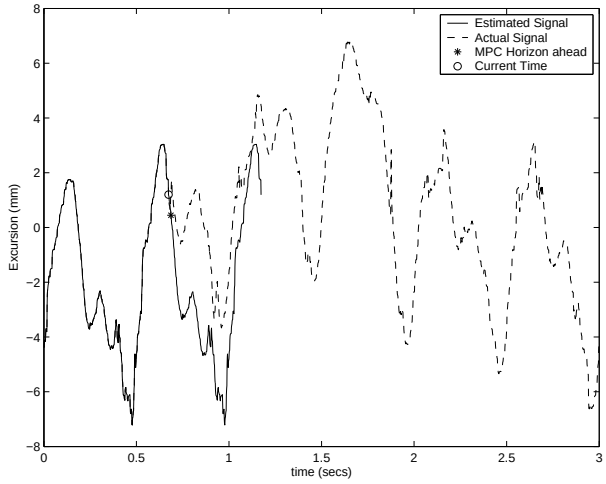
(c)




(d)




(e)




(f)

Figure 3.6: Estimated Signal and Actual Signal

# Chapter 4

# Simulation and Results

All the control algorithms mentioned in the previous section were initially developed and tested in simulation. These simulations were conducted in Matlab and written using the executable .m Matlab files. Simulations were conducted for two primary reasons. The first was to test the algorithms to see if the proposed control method actually gave improved performance over the traditional approaches. The simulation results section presented later in this chapter will divulge these findings. The second reason was for hardware implementation. By using the .m files, code could be more easily ported over to the eventual test bed that utilized C code. This made debugging easier and reduced translation errors.

## Experimental Description

The diagram in Figure 4.1 shows the experimental setup for all of the algorithm testing. The heart signal in some capacity is fed into the controller along side of the system output. The control is calculated based on these two inputs and hopefully the hardware position will match the heart position.

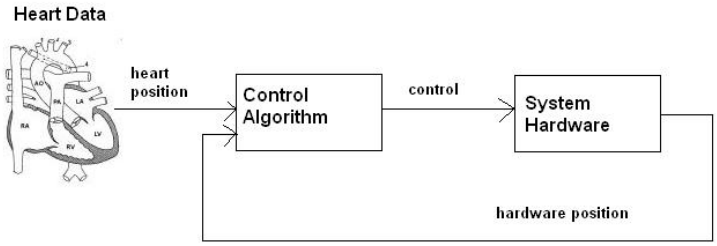



Figure 4.1: Diagram of Experimental Setup

## Heart Data Resampling

All controllers were implemented with either a 2Khz and or 1Khz controller. The original heart motion data collected as described in Section 1.7 was sampled at approximately 257 Hz. Resampling this data by using a zero-order hold would add unnecessary noise and most likely cause larger error for tracking.y As a result, a second data set was written that used a first-order approximation between the original sampled points.

The original data was sampled at 257Hz and produced values $y(t)$ where $y(t)$ is the position at time $t$. The time step corresponding to 257Hz is defined as $\delta t$. A straight line can be fit between the two points $y(t)$ and $y(t + \delta t)$. This straight line fit is then sampled at a higher rate for values of time between $t$ and $t + \delta t$ instead of using just $y(t)$.

The plot in Figure 4.2 shows the effects of resampling. The line without steps is the desired signal while the line with larger steps is the resampling with just a

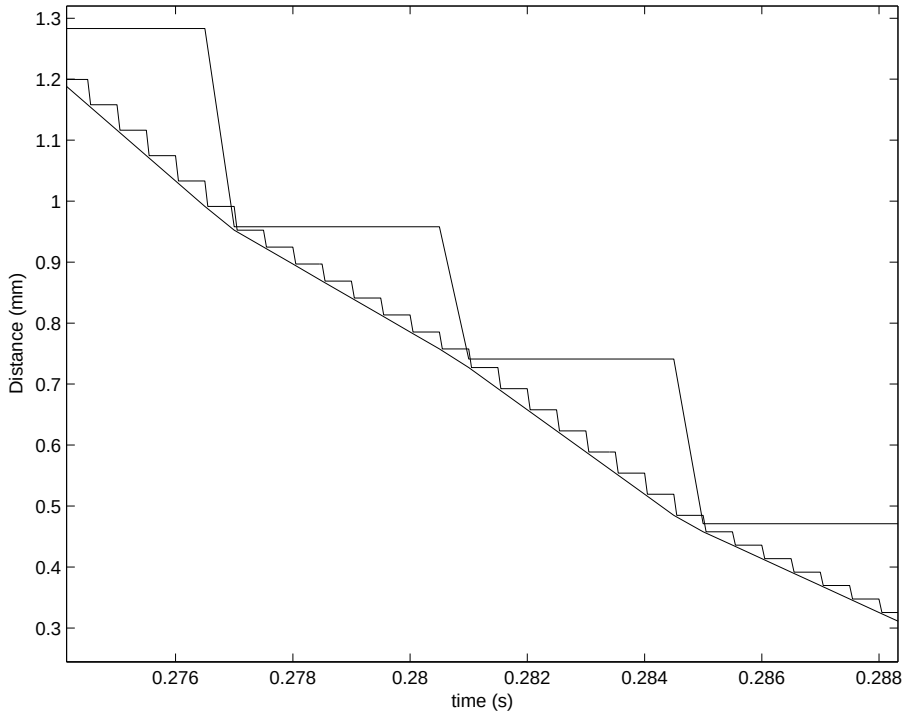



Figure 4.2: Resampling Effects

zero-order hold approximation of the original data. The smaller-step line is using the resampled first-order approximation data. This method produced position data that was utilized in the Matlab simulation as well as the hardware implementation.

To further promote stability, the heart signal was ramped up to its actual value. This means that the first half-second of the reference signal was linearly scaled up in amplitude to the signal full value. Instantly tracking the signal proves to be difficult as it is already "moving" and the sytem starts at rest. This implementation came later as a precaution for the robot and was not implemented with the speaker or in simulation. It does not effect the results because the calculations were only carried out on the portion of the data that would be considered steady state.

## 4.1 Algorithm Implementation

A brief description of how each of the algorithms was implemented in Matlab follows.

### 4.1.1 PD Control

The simulated system was implemented using the state-space model. The derivative of the system was calculated by doing a first-order backwards difference approximation (the current output subtracted by the previous output over the time step of the simulation). The gains were directly tuned as stated in Section 3.2, and as a result, no prior loop calculations were necessary. The actual simulation code can be seen below.

```
T = 1/control_frequency;
y = 0; y_old = 0; y_dot = 0; y_vec =[];
x_old(1:length(phi)) = 0;

for k = 1:total_time/T-1;
    %Create time vector
    time(k) = (k-1)*T;

    %Calculate Control
    u(k) = k_p*(y_desired(k) - y) - k_d*y_dot;

    %Simulate in Plant
    y = h*x_old + j*u(k);
    y_vec = [y_vec; y];
    x_new = phi*x_old + gamma*u(k);
    x_old = x_new;

    y_dot = (y-y_old)/T;
    y_old = y;

end
```

### 4.1.2 Pole Placement

For this algorithm, the simulated system utilized the state-space representation. The feedforward gains were calculated by breaking down the $\mathbf{N}$ and $\mathbf{N}_f$ from Eqn 3.14. The systems being used are all single-input, single-output (SISO) systems. Therefore, the identity matrix from Eqn 3.10 is a $1\times1$ matrix. The vectors corresponding to Eqn

3.14 are then $(n+1)\times 1$ matrices. More directly, the steady state gain $\mathbf{N}_f$ is $1\times 1$ and $\mathbf{N}$ is $n\times 1$ (Note that $n$ is the order of the system). This code can be seen below.

```
%Feedback Gain Calculation
KK = acker(phi,gamma,desired_poles)

%Feedforward Gain Calculation
Nmatrix = [[phi-eye(length(phi))  gamma];[h 0]];

temp(1:length(phi)) = 0; temp(1:length(phi)+1) = 1;

N = inv(Nmatrix)*temp';

Nx = N(1:length(phi)); Nu = N(length(phi)+1); N_bar = Nu + KK*Nx;

T=1/control_frequency;
y = 0; y_vec =[]; x_old(1:length(phi)) = 0;

for k = 1: total_time/T-1;
    %Create time vector
    time(k) = (k-1)*T;

    %Calculate New Control
    u(k) = N_bar*y_desired(k) - KK*x_new;

    %Simulate in Plant
    y = h*x_old + j*u(k);
    y_vec = [y_vec; y];
    x_new = phi*x_old + gamma*u(k);
    x_old = x_new;

end
```

In order to alleviate some of the difficulties that were encountered with the PHANToM, another type of pole placement simulation was set up that utilized Simulink. Simulink is a platform for multidomain simulation and model-based design of dynamic systems. It provides an interactive graphical environment and a customizable set of block libraries that lets the user accurately design, simulate, implement, and test control, signal processing, communications, and other time-varying systems [48].

Using Simulink allowed effects such as the saturation of the actuators and the

Coulomb friction terms to be incorporated easily into the model and to test for implementation accuracy. The diagram used in the simulation can be seen in Figure 4.3.

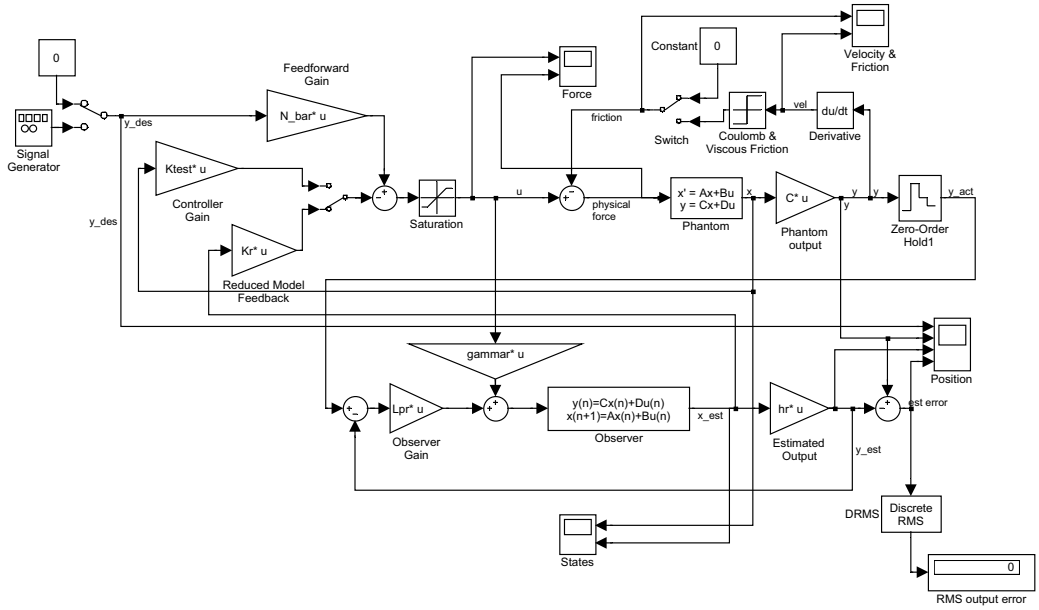



Figure 4.3: Simulink model of PHANToM and PP controller used to determine effects of Coulomb friction and saturation on control

The constants and gains used by Simulink were set by using a Matlab executable file. The poles were placed using the `acker()` function for the observer and feedback.

### 4.1.3 MPC

The Matlab simulated MPC was calculated in a slightly different method than actually implemented. As presented, the feedback gains of the algorithm can be calculated before the control loop, while the feedforward portion was iteratively calculated during the control. The feedback and feedforward portion both use the same parameters

in their respective calculations. In the Matlab simulation, these parameters were calculated before the control was executed and then multiplied together to get gains and the control effort during the control loop. The parameter `b` seen in the code below is analogous to the defined matrix $\mathbf{M}$ in Eqn 3.41. The difference between the two is a matrix multiplication in the iteration calculation. In hardware implementation, all calculations that could be done outside of the control loop were performed outside of the control loop.

The indexing for this algorithm can become particularly complicated because of the backwards iteration. In order to compensate for this irritation, an interim counter was created within the iteration loops. As was presented, the parameter $\mathbf{S}$ was used in place of the Riccatti parameter $\mathbf{P}$ (where $\mathbf{S} = \mathbf{Q} + \mathbf{P}$) to simplify the code. The actual code used can be seen below.

```
L = h*inv(h'*h);

%Iteration of Optimal Parameters
Q = (eye(length(phi))-L*h')*Q1*(eye(length(phi)) - L*h')' +
    h*Q2*h';
S(:,:,horizon) = (eye(length(phi))-L*H')*Q1*(eye(length(phi)) -
    L*h')' + h*Q2*h';
b(:,horizon) = zeros(length(phi),1);

for counter = 1:horizon-1
    a = horizon - counter;

    S(:,:,a) = phi'*(S(:,:,a+1) - S(:,:,a+1)*gamma*inv(gamma'*S(:,:,a+1)
        *gamma + R)*gamma'*S(:,:,a+1))*phi + Q;
    K(:,a) = (-inv(gamma'*S(:,:,a+1)*gamma + R)*gamma'*S(:,:,a+1)*phi)';

end
T=1/control_frequency;

x_old(1:length(phi) = 0; u = 0; y_vec = [];

for k = 1: total_time/T-1;
    %Iteration of b
    b(:,horizon) = zeros(length(phi),1);
```

```
    if (length(y_desired) >= (horizon-1+k))
        for counter = 1:horizon-1
            a = horizon - counter;
            b(:,a) = (phi' + K(:,a)*gamma')*b(:,a+1) - Q*L*y_desired(k+a);
        end
    end

    %Create time vector
    time(k) = (k-1)*T;

    %Calculate Control
    u(k) = -inv(gamma'*S(:,:,1)*gamma + R)*gamma'*(S(:,:,1)*phi*x_old
         + b(:,1));

    %Simulate in Plant
    y = h*x_old + j*u(k);
    y_vec = [y_vec; y];
    x_new = phi*x_old + gamma*u(k);
    x_old = x_new;

end
```

### 4.1.4 Signal Estimated MPC

The signal estimate code has two parameters that can be used for tuning. The first parameter was how far ahead to start correcting the estimate or the error horizon parameter. The second was for setting the power of the error-percentage function.

The error-percentage function is the function that showed the percentage of current error of the guessed signal to be added versus the number of steps ahead. This function passed through two points. The first was 0% and the error horizon value. The second was 100% and 0 steps ahead. This second parameter is the order of the fit between these two points. A first-order fit is just a straight line between the two points. A second-order fit is a parabolic function between the two points and so on for higher order fits. Fits one through ten can be seen in Figure 4.4. Note that they are labelled from left to right 1-10 and in this particular plot, the error horizon value was 150.

The code for the signal estimater can be seen below. One extra vector was set up

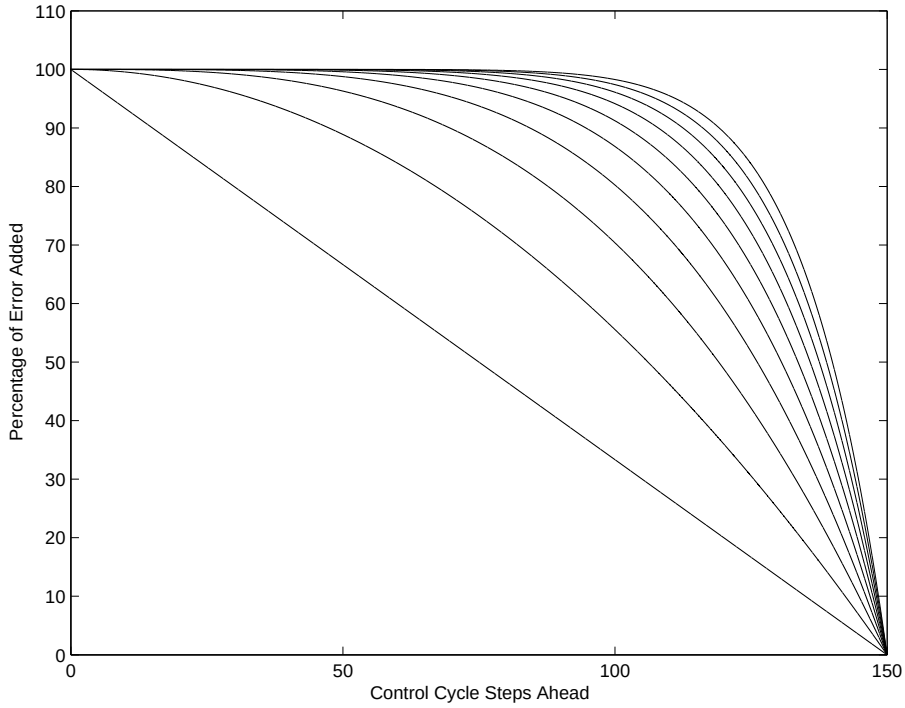



Figure 4.4: Error correction functions with a 150 step error horizon. The furthest left function is 1st-order while the furthest right is 10th-order.

to execute this code.

```
%Calculate Current Error
y_error(k) = y_actual(k) - y_guessed(k);
for i = 1:error_horizon
    y_guessed(k+i-1) = y_guessed(k+i-1) +
        (1 - ((i-1)/error_horizon)^nth_order )*y_error(k);
end

%Put in actual Heart Value as next cycle
y_guessed(k+heart_period/T) = y_actual(k);
```

The `nth_order` and `error_horizon` variables are the tuned parameters. As in the previous coding examples, `k` is the current state and `T` is the control algorithm period. The parameter `heart_period` refers to the period of the heartbeat. This code was directly inserted into the MPC control code with the addition that the `y_guessed` variable seen here was used as the `y_desired` variable in the iteration of `b`.

## 4.2 C Simulations

Simulations in C were conducted for the sole purpose of minimizing coding errors. The computations often required matrix calculations which are not always simple calculations when coding in C. These simulations mirrored those executed in Matlab and the resulting control signals, plant response and even system state were compared for correctness. Identical outputs assured that the code was transferred from MATLAB to C successfully.

To facilitate the coding, functions were made that performed different tasks of the simulation. Functions that simulated the plant, observed the plant, and calculated the control effort were all implemented. Constants and gains that could be calculated before the control loop were set as static variables within their respective functions.

The observer function, simulated state function and MPC and pole placement algorithms required matrix calculations that were easily executed in Matlab. To do these calculations in C, a freely distributed linear algebra library was used. The Meschach library is a linear algebra toolbox that allows users to call the general matrix manipulations. The open-source code is intended to be used by C developers to make more complicated matrix calculations by providing basic matrix functionality. The code executes particularly quickly and for the purposes of this project, possessed all the required functionality. For a complete reference and more information on the library, see [49].

## 4.3 Simulation Results of Algorithm Testing

Each of the algorithms described was tested and tuned in simulation. This section describes the results of those tests.

The results in this section and all of the following sections have been measured in the following fashion. The desired accuracy of tracking will be presented in the

appropriate units and suffice as a metric for comparison of all of the algorithms. All errors in tracking have been reported as RMS errors. The maximum error may be reported but is not considered in algorithm effectiveness. For the purposes of this project, the control effort is generally not considered. The original goal was to acquire accuracy at the cost of any control. However, in simulation, it is possible to drive errors down to very small levels while the control becomes unreasonably large (often regardless of the algorithm being tested). When this occurs the differences in errors are insignificant and as a result, nothing can be determined. The control in this case must be considered for comparison purposes. As a result, the algorithms were tested for accuracy while monitoring the RMS control used. Minimizing the error at a given control for each of the algorithms would be the optimal method for comparing the effectiveness of each of the algorithms and would probably be a necessary task when comparing small improvements between algorithms.

The transfer function used in the initial simulations was a small-signal Cartesian model obtained by Cavusoglu et al. [42] of the PHANToM manipulator. The model related the $x$-direction position to the force. The transfer function can be seen in Eqn 4.1.

$$\frac{x(s)}{F(s)} = \frac{s^2 + 5.716s + 9.201 \times 10^4}{3.329 \times 10^{-6}s^4 + 0.001226s^3 + 1.536s^2} \tag{4.1}$$

The units of the model are in mm/N. The desired accuracy is within 100 $\mu$m RMS.

### 4.3.1 PD Control

The tuning process for this control method was straightforward. Using the continuous state-space model to approximate the velocity term (as discussed in Section 3.2) allowed higher gains to be chosen for the algorithm and better results were achieved over the first-order velocity approximation method. The algorithm however still broke

down and became unstable for excessively high gains.

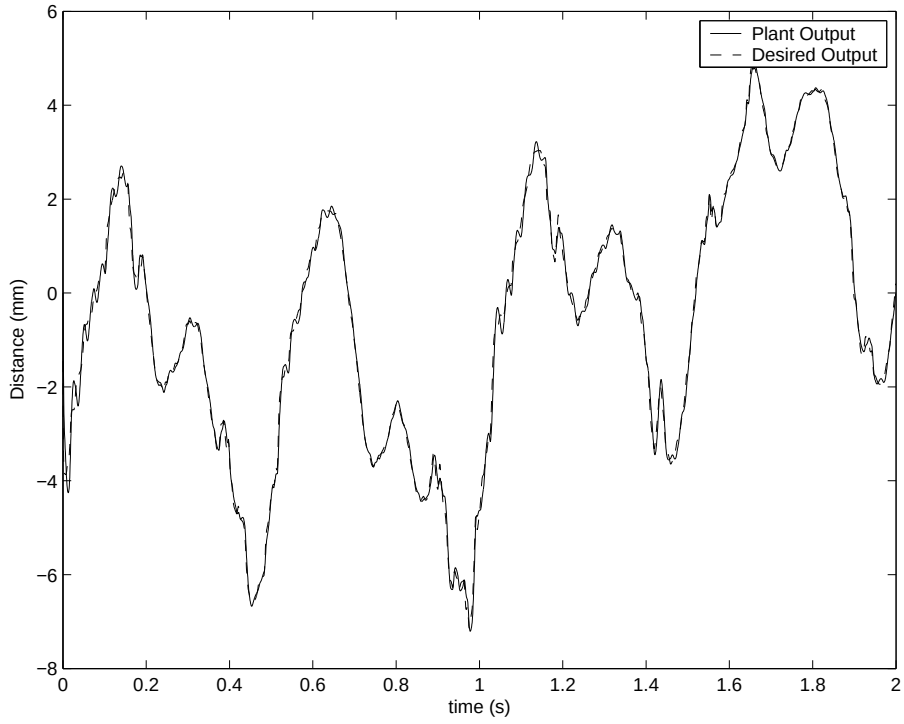



Figure 4.5: Simulated PD controlled system output and desired output

Figure 4.5 is a plot of the desired trajectory and the simulated output. The tracking appears to be effective. However, the RMS tracking error was 0.17 mm, which is not below the desired specification of 0.10 mm. The control required for this algorithm was 0.70 N RMS.

By examining Figure 4.5 more closely, it appears as if the simulated plant output oscillates over the heartbeat signal while attempting to track. In the later beats, the plant output settles down and performs better. This oscillation is due to the initial position of the desired trajectory. The simulated plant starts at a position of zero while the desired trajectory starts near -3.5 mm. In an attempt to track this step, the simulation overshoots and then oscillates onto the desired signal. Some of this oscillation can be seen in Figure 4.6.

It would be possible to push the gains higher had this initial step been removed. This could be done by starting the simulation of the plant at -3.5 mm or choosing a

desired trajectory that started at zero. This was not done for any of the algorithms, however, and the first beat was not considered when calculating error and control effort.

### 4.3.2 Pole Placement

The presence and location of zeros has a direct effect on the system response. A transfer function with dominant effects in the numerator will adversely affect the system output [50].

In general, the zeros of a system cannot be placed, so the inherited dynamics must be compensated for using some other technique. For stable zeros, a common option is to simply cancel out the zero dynamics by placing a pole in the same position. With the zero's dynamics out of the way, it is possible to design a much better controller where the dominant effects can be directly controlled.

Upon attempting to place the poles at the zeros of the transfer function and then placing the remaining poles on the positive real axis (in the z plane), the following results were obtained. The RMS error was 0.15 mm and the control was 0.97 N RMS. The tracking response and error plots can be seen in Figures 4.7 and 4.8 respectively. After the poles and zeros effectively cancelled each other out, the new bandwidth of the system was at the remaining placed pole location. These poles had a bandwidth of 180Hz.

### 4.3.3 MPC

Using a future known signal for the MPC algorithm is equivalent to having nearly perfect tracking. It is similar to driving in fog versus driving on a clear day. It is much easier when the driver can see what is coming. As stated, using a future known signal is not feasible in this particular scenario but this algorithm was used to show the baseline performance.

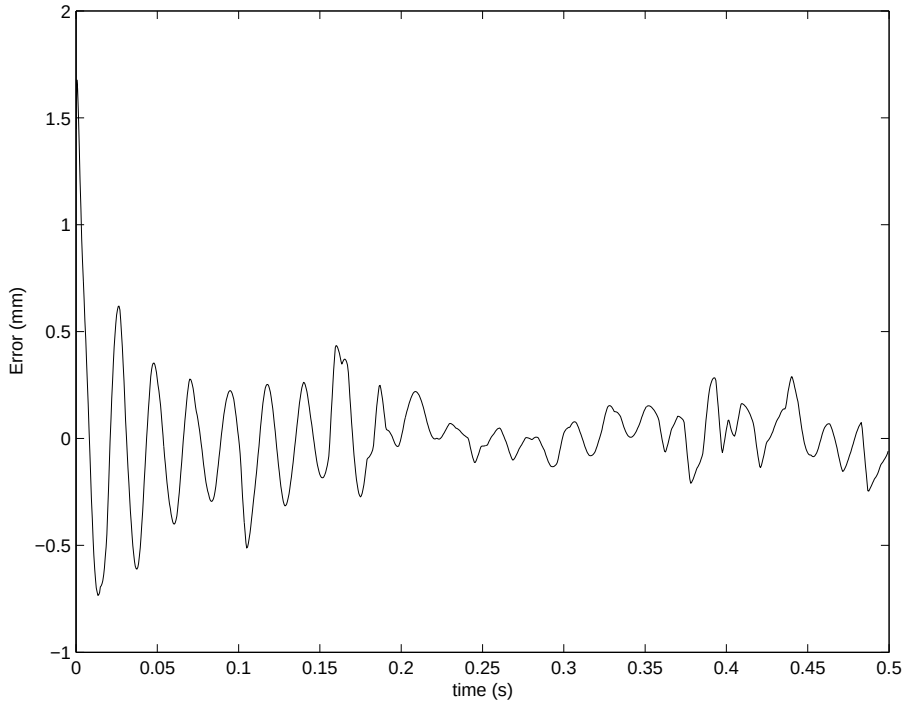



Figure 4.6: Simulated PD error between desired signal and system output

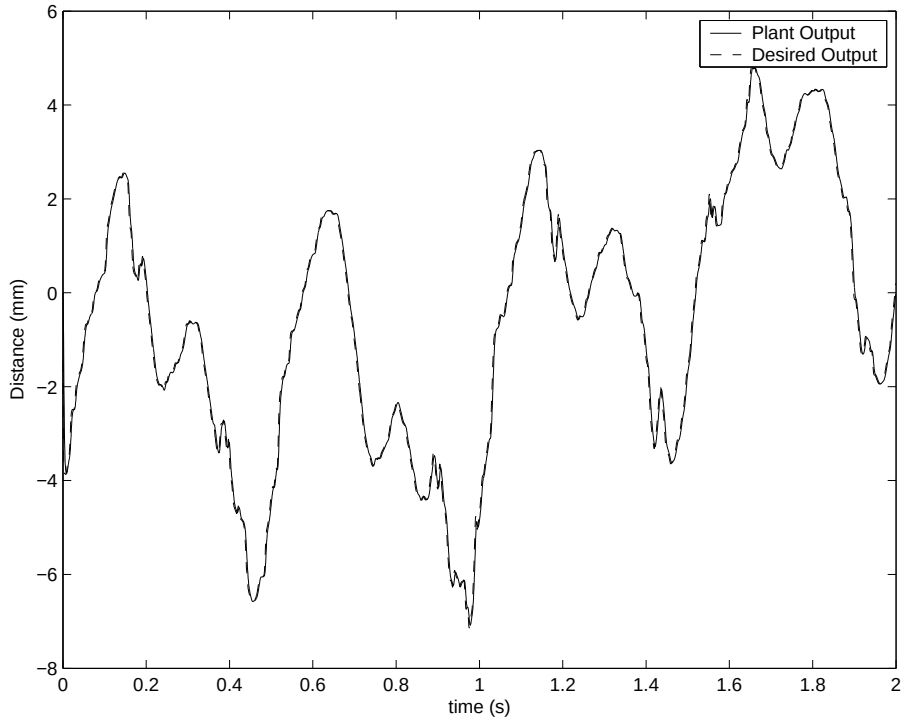



Figure 4.7: Simulated PP controlled system output and desired output

This was quite evident upon simulation testing of the algorithm. Figure 4.9 is actually the plant output superimposed on the desired trajectory. The only portions where the two plots differ are at the beginning of the simulation and near some of the high frequency regions of the desired signal. In this simulation, the gains were tuned particularly tightly, yet the RMS control was only 0.93 N. The RMS error was 0.023 mm. Tightly tuned gains in this algorithm mean that the ratio of $\mathbf{Q}$ to $\mathbf{R}$ was particularly high.

Figure 4.10 shows the same simulation results with weaker gains. There is a small noticeable difference near the peaks of the desired waveform, but the plots again match up better than in the previous two algorithms. The RMS control here was 0.30 N, with an error value of 0.080 mm RMS. These results show that this algorithm has a control on par with the pole-placement method while possessing three times better accuracy.

### 4.3.4 Signal Estimated MPC

The signal estimated MPC was able to produce very favorable results. From Figure 4.11, the tracking appears to be very accurate. The actual numbers support this with an RMS error of 0.09 mm. The corresponding control effort was 0.78 N RMS. This algorithm was the only causal algorithm that was able to get the error below the desired specification while maintaining a reasonable control level.

The horizon value and weighting matrices used for tuning were the same as those that were used for the tightly tuned MPC algorithm. Recall that the tuning parameters for the error correction function utilized a fit order and error horizon value. The error correction function utilized only a 200 step look-ahead with a second-order fit. It is possible that the function could have been made better with a higher-order fit and possibly a larger error horizon, but the control and RMS error were within the desired specifications.

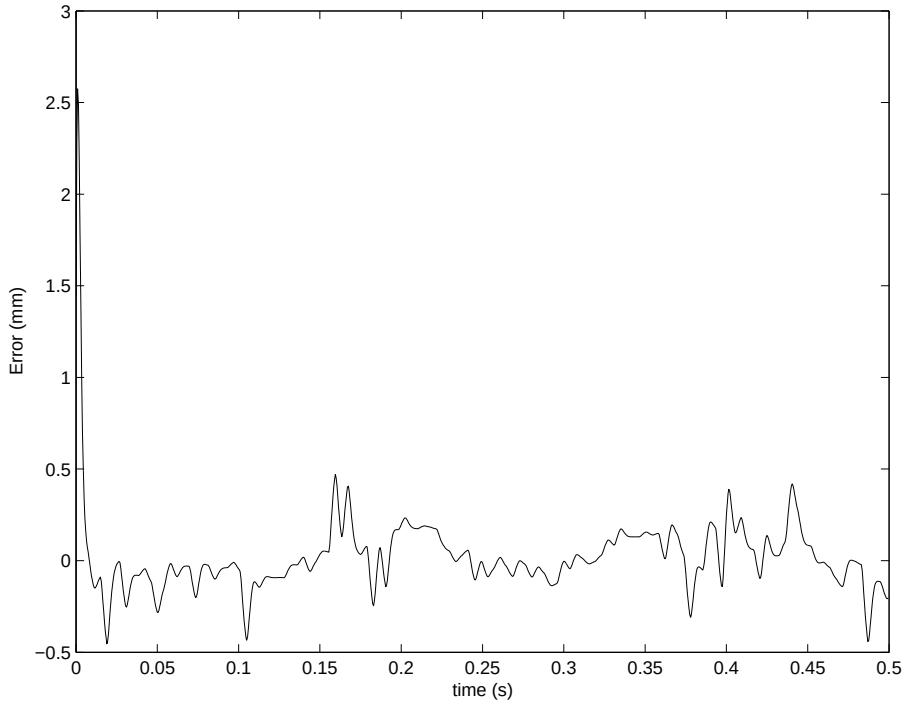



Figure 4.8: Simulated PP error between desired signal and system output

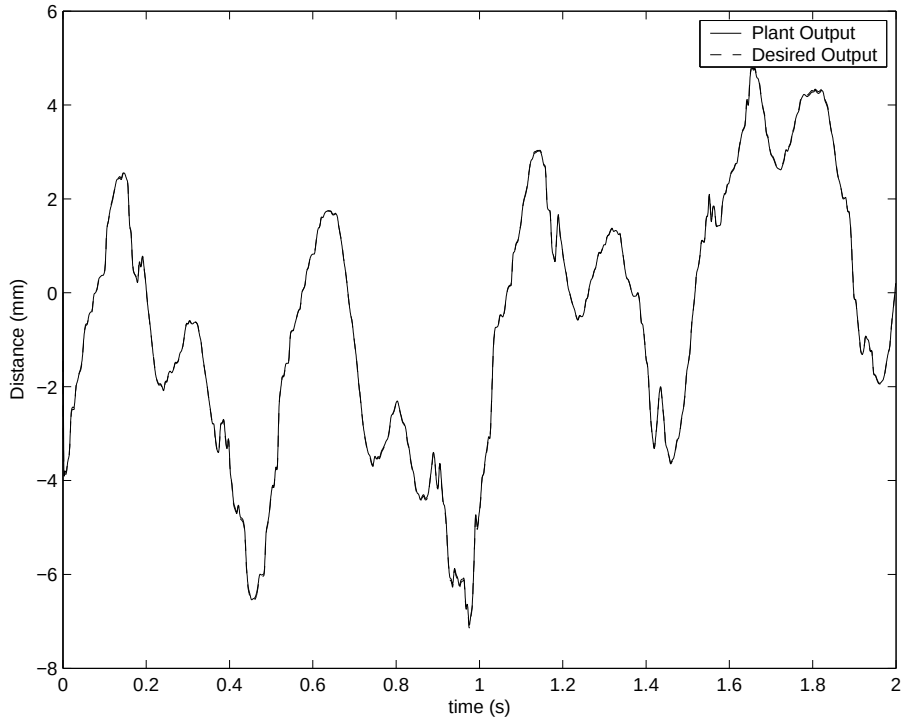



Figure 4.9: Known-future high-gain MPC tracking results

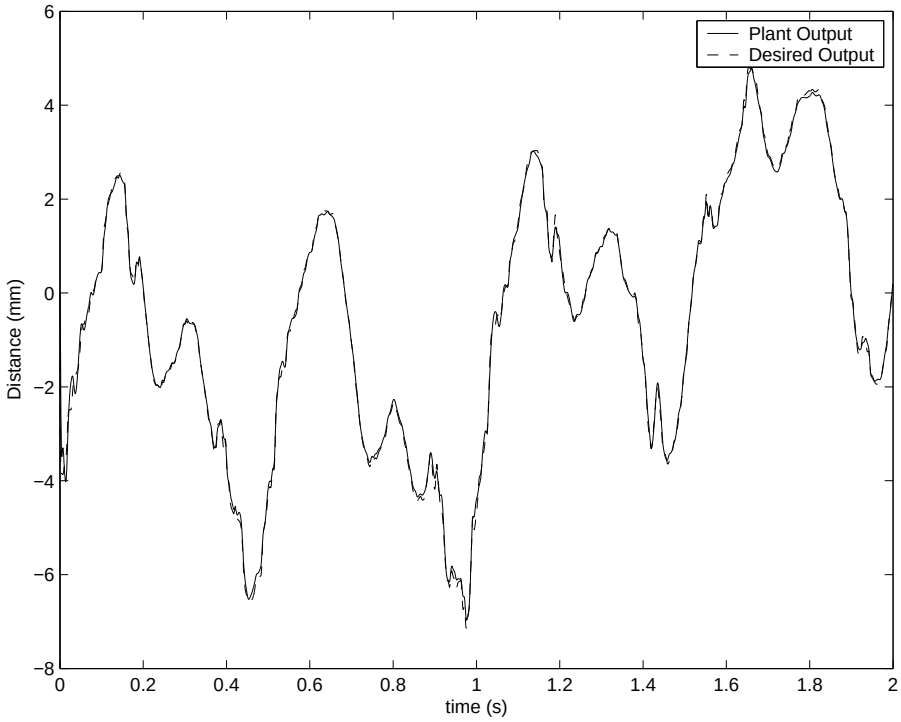



Figure 4.10: Known-future low-gain MPC tracking results

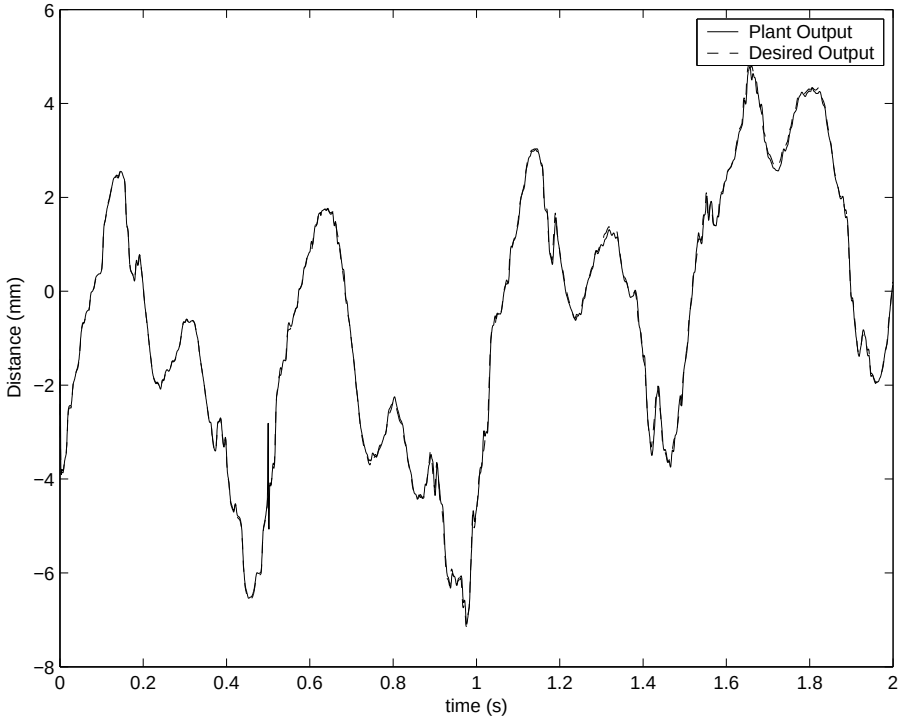



Figure 4.11: Signal estimated MPC tracking signal superimposed over the desired heartbeat signal

# Chapter 5

# Results of Hardware Experiments

The hardware results are presented in the same order as the previous control and simulation sections. Again, the error specification is quantified for each of the two test beds before presentation of results. Each experiment was run for at least 12 heart beat cycles and a portion of those cycles can be seen in plots for each section. The heart signal used as a reference for tracking is spoken about in detail in Section 1.7.

## 5.1 Speaker Results

The results for this speaker are given within this section along with any subsequent discussion of the results. The target RMS error is 0.1 mm again for the subwoofer. Recall that the speaker model was obtained using the frequency-response method. As was stated earlier, a voltage was output from the DAC into the system, causing the speaker to move. The resulting transfer function relates voltage input to position output. Therefore, the control is quantified in Volts while the position will be presented in mm.

### 5.1.1 PD Control

The PD control produced the worst results of all the algorithms tested and was only able to produce accuracies with 1.4 mm RMS error. This result is more then ten times larger the desired specification of 0.1 mm. The RMS control did remain low with a value of 0.23V. A plot of the tracking results can be seen in Figure 5.1.

The system output was never able to reach the desired trajectory maximum and minimum values. The control corresponding to this experiment (Figure 5.2) shows the system attempting to reach these high peaks. From the large amount of noise seen on the control effort signal, it is likely that the velocity approximation still limited the system's ability to track more accurately. Recall that in this instance, the velocity was estimated using the observer and state space approximation. It is possible that observer gains were tuned too tightly for a good filtered approximation of the velocity. In either case this algorithm performed particularly poorly and even a better velocity approximation may not have been enough to make the results comparable to the other algorithm results.

### 5.1.2 Pole Placement

The pole-placement algorithm was able to obtain an RMS error value of 0.82 mm with a corresponding control of 0.37V. Even though the control was much higher for this method relative to PD control, the value still was well within the operating range of the speaker. A plot of the desired signal and actual system output can be seen in Figure 5.3.

Recall that in simulation the PP algorithm approached instability for very high gains, but when the gains were reduced, a persistent delay existed. This is clearly seen through implementation on the speaker. Being on the verge of instability in simulation, however, corresponded to instability in the real system. As a result, the gains were reduced until a stable system was acquired, which resulted in the

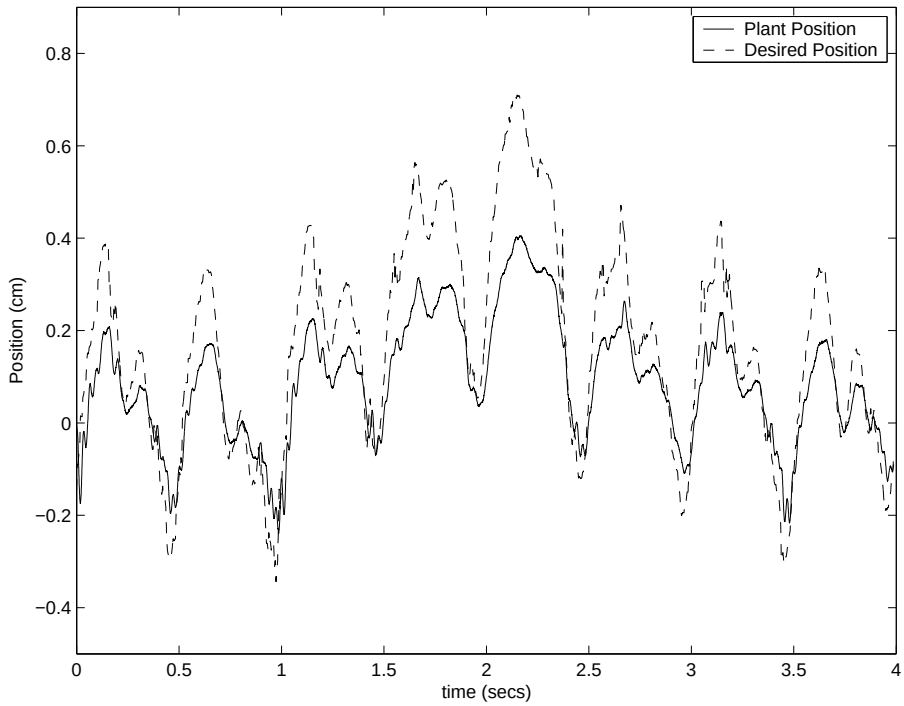



Figure 5.1: PD controlled system output and desired output

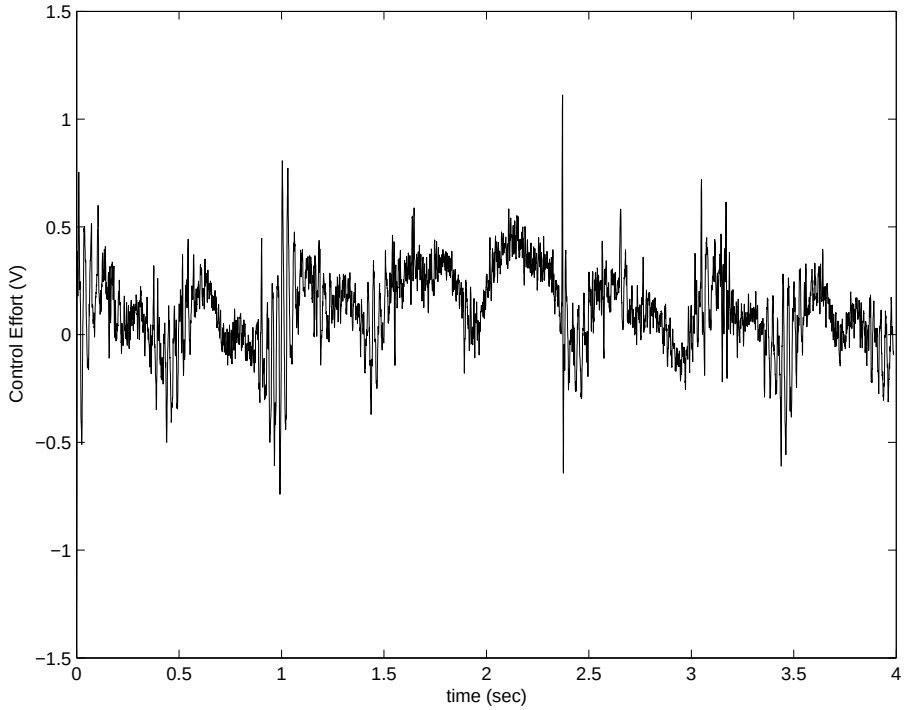



Figure 5.2: Speaker PD Control Effort

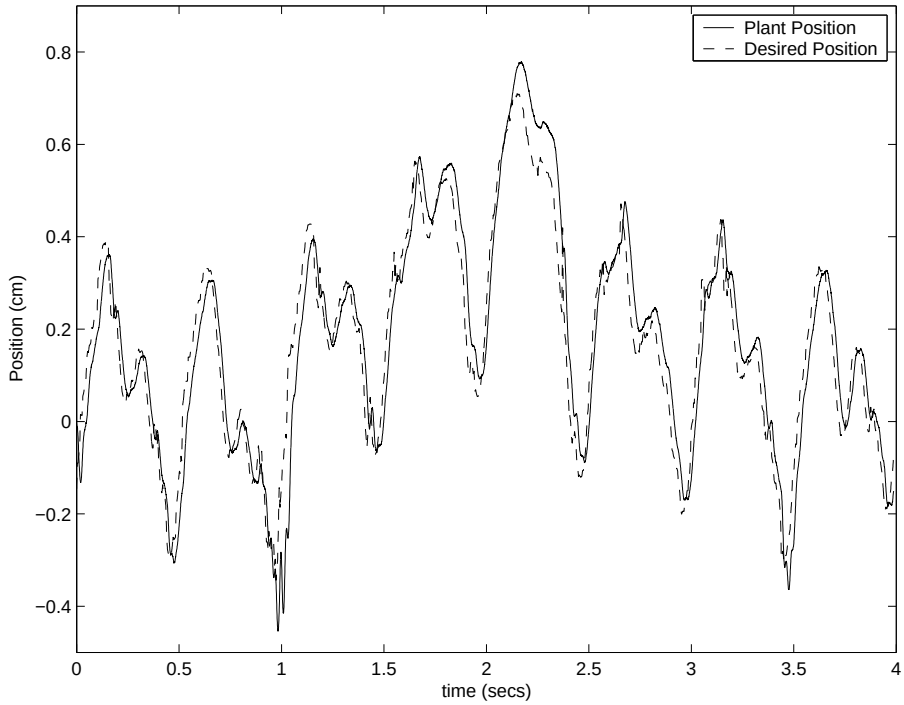



Figure 5.3: Speaker PP Controlled System Output and Desired Output

unwanted delay. This delay comprises a large portion of the error for most of the tracking process.

### 5.1.3 MPC

The computer (used for this experiment) could execute code at a maximum of approximately 40 steps ahead within the control sample period of 0.5 msec. No tests were attempted above that horizon. It is possible that with a faster computer, the results could have been improved with a larger horizon.

The known-future MPC algorithm again produced the best results, as expected. Through tuning of each of the weighting parameters, a combination was found that produced an RMS error value of 0.28 mm and a corresponding control of only 0.33 V RMS. Though the tracking error is still larger then the required response, the speaker system is not meant to perform surgery. This particular test serves more as a relative performance evaluation. It is likely that the best this particular system could track

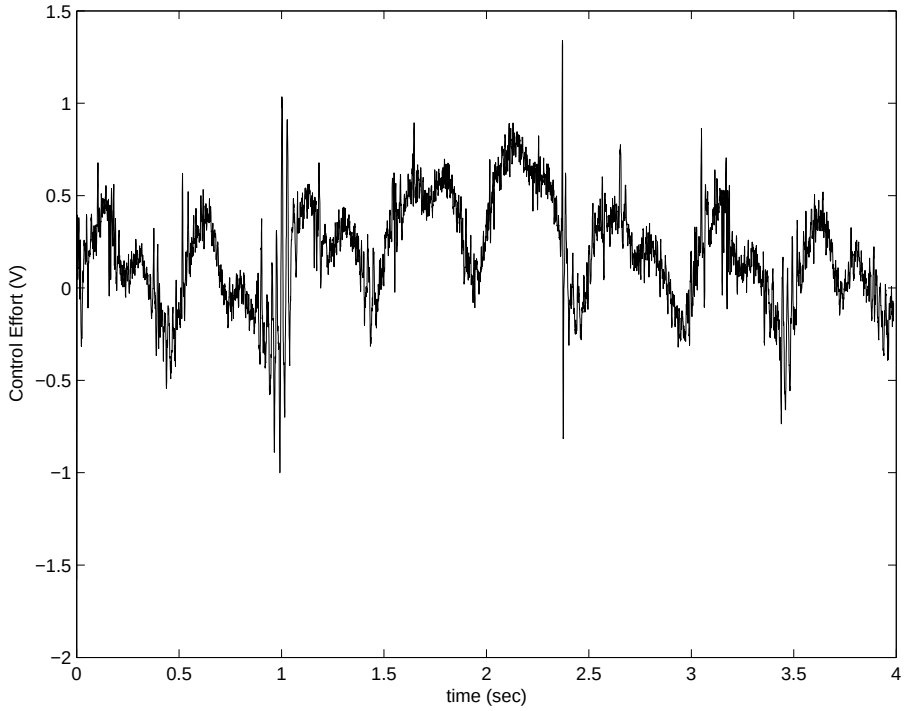



Figure 5.4: Speaker PP Control Effort

would be near 0.3 mm RMS error. The actual output can be seen in Figure 5.5.

Since the error is difficult to see between the two plots, the error plot is provided (Figure 5.6).

### 5.1.4 Signal-Estimated MPC

The signal-estimated MPC algorithm again allowed for better results than obtained in either PD control or PP control. Consistent with simulation, the results were worse than those found using the known-future MPC.

The error-correction function was tuned with virtually the same values of the gain parameters and the horizon used in the known-future MPC. The error horizon worked best when looking 750 steps ahead. This corresponds to a look-ahead of 0.375 seconds. Along side of the 750 step error horizon, the function was raised to the 8th power. With this combination of parameters, the RMS error was 0.46 mm. The control was 0.33V RMS.

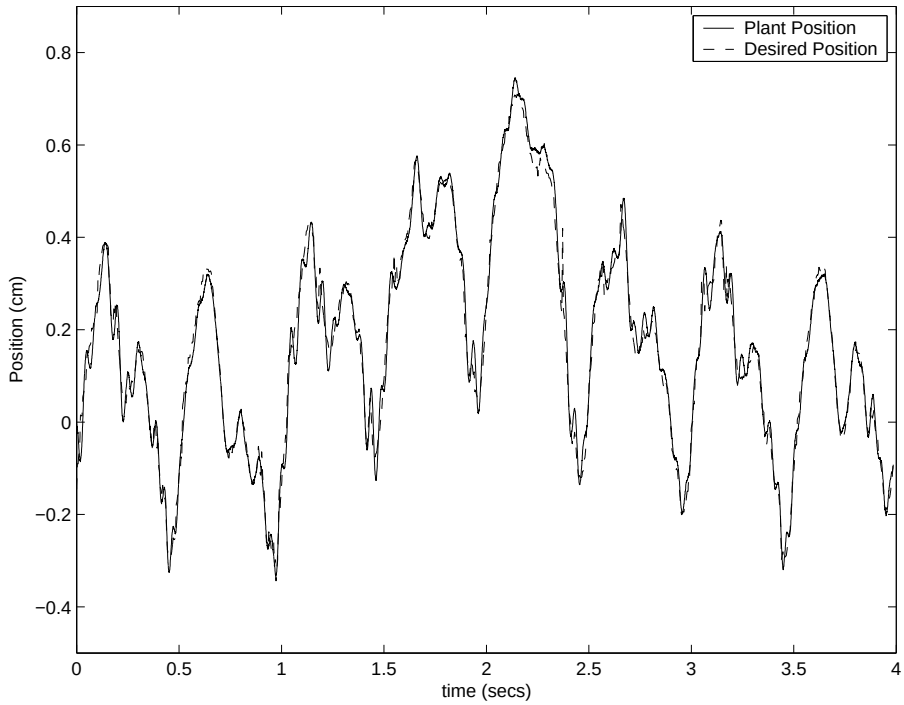



Figure 5.5: Known-Future MPC System Output and Desired Output

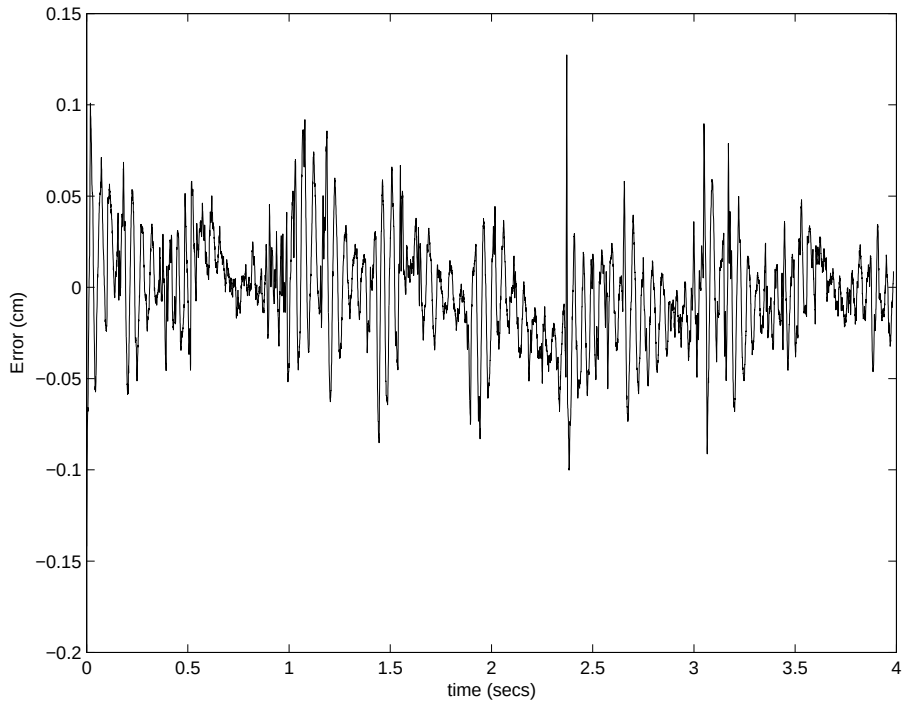



Figure 5.6: Speaker error between known-future MPC output and desired signal

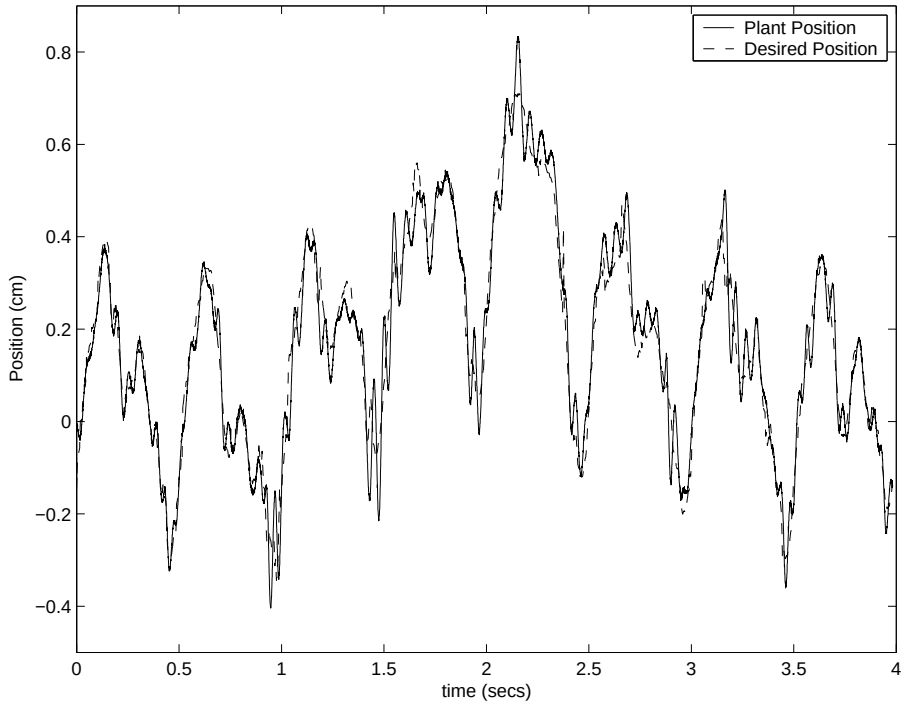



Figure 5.7: Signal-Estimated MPC System Output and Desired Output

The signal-estimated MPC method works by extrapolating forward one heartbeat; it would be possible to extrapolate forward more than one beat but this may lead to larger errors as consecutive cycles are likely more similar then every other cycle. For a sampling frequency of 2kHz, one heart beat ahead corresponded to a maximum error horizon of 1000 steps (0.5 seconds). The error horizon was attempted at that level; however, the results were close to but not quite as accurate as those using just the 750-step method. Though this method is not overly time consuming, it makes sense to attempt to keep the overall algorithm execution as short as possible. If there is no gain in accuracy for more computation (as was the case in this instance), it makes sense to save the computation time for the MPC gain calculation.

The signal-estimated MPC error plot is shown in Figure 5.8 for comparison to the known-future MPC algorithm.

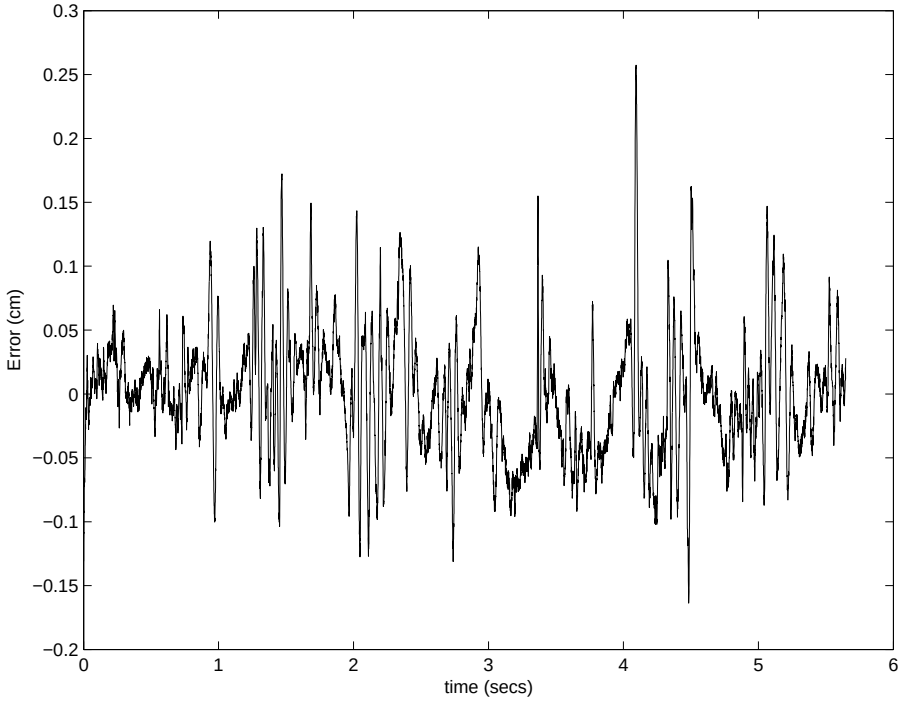



Figure 5.8: Speaker error between signal-estimated MPC and Desired Output

## 5.2 PHANToM Results

The PHANToM results have been measured and recorded in radians. The input has been recorded as a torque with units of Newton-meters. In order to get an equivalent experiment in the circular domain, the desired trajectory signal was modified by dividing it by the length of the arm. Using Eqn 5.1, the desired RMS error can also be calculated.

$$s = r\theta \tag{5.1}$$

The parameter $s$ is the arc length in mm while $r$ is length of the robot arm. A desired RMS error of 0.1 mm corresponds to an angle of 0.465 mrad. Though this is not a straight-line length, the arc length of rotation does traverse the correct amount of distance for the experiment.

### 5.2.1 PD Control

As was done in the previous two experimental setups, the PD control was tested first. The best PD results were able to produce an RMS error of 1.217 mrad (0.2617 mm) and a corresponding control of 0.3274 Nm RMS. This error is nearly three times the desired specification. A plot of the control effort and the error signal can be seen in Figure 5.9.

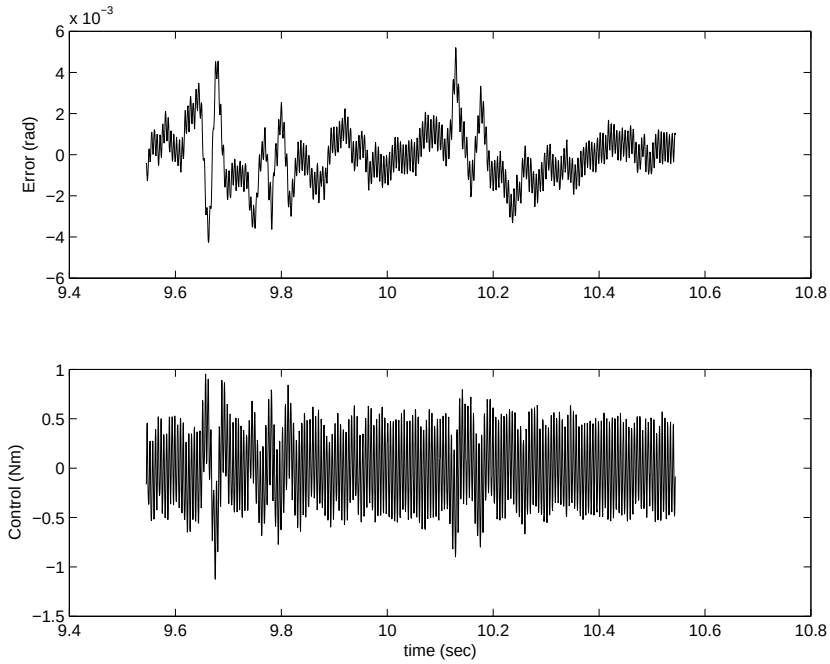



Figure 5.9: PHANToM PD-Controlled Tracking Error and Control

### 5.2.2 Pole Placement

Initial results and tuning of the pole placement algorithm were worse than those of the PD control. The poles were placed on the positive real axis as close to $z = 0$ as possible while still maintaining stability. The system became unstable for poles that were placed closer than $z = 0.8$. Poles at this location correspond to a bandwidth of approximately 70Hz, which should be adequate to track a 20Hz reference signal. The results of this pole placement can be seen in Figure 5.10.

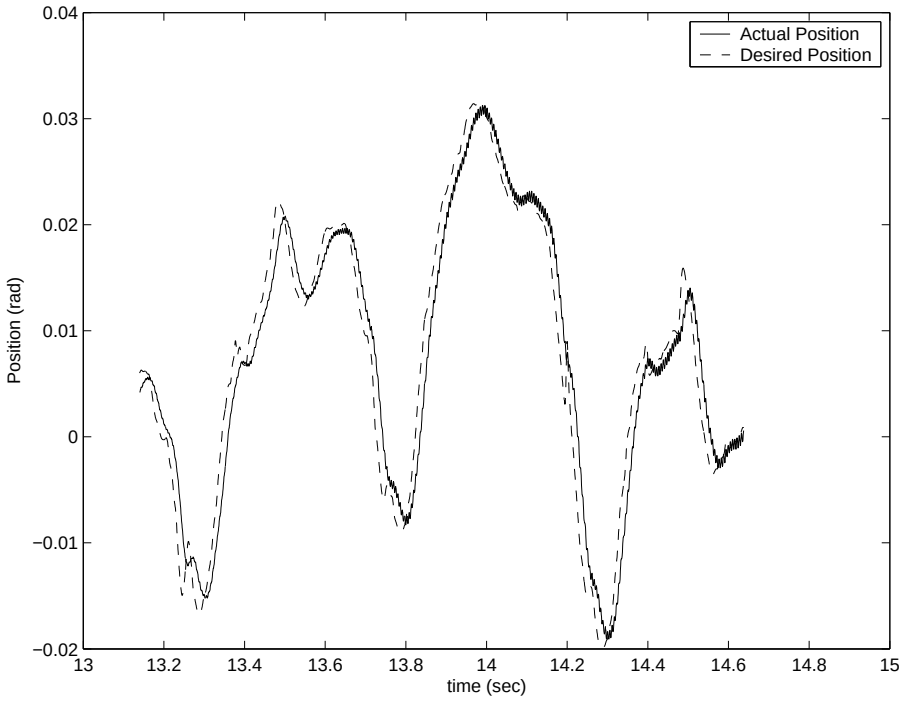



Figure 5.10: PHANToM PP controlled tracking results

The experiment in Figure 5.10 utilized poles placed at $z = 0.85$ and had an RMS error of 3.1 mrad (0.6665 mm) and a control RMS of 0.1561 Nm. Results with poles placed at $z = 0.8$ produced a similar value for error but a much larger control value. For those gains, the system was nearly unstable and hence the results were not improved.

An approach to finding better poles was to look to the PD control. By using the experimentally-determined gains and experimentally-obtained system model, the closed-loop response can be obtained. In this instance, the closed-loop response produced poles that were relatively close to the system zeros. In an attempt to match the results obtained by PD control, a pole placement design was performed with poles specified identical to those resulting from the tuned PD controller. Theoretically, the performance will be very similar to the PD controller.

Figure 5.11 shows the position-plus-derivative and pole-placement responses for calculated identical feedback gains. From this plot, it is evident that the controls

are, in fact, similar. The response is not exactly the same primarily because the algorithms differ in feedforward gains (explained later in this chapter).

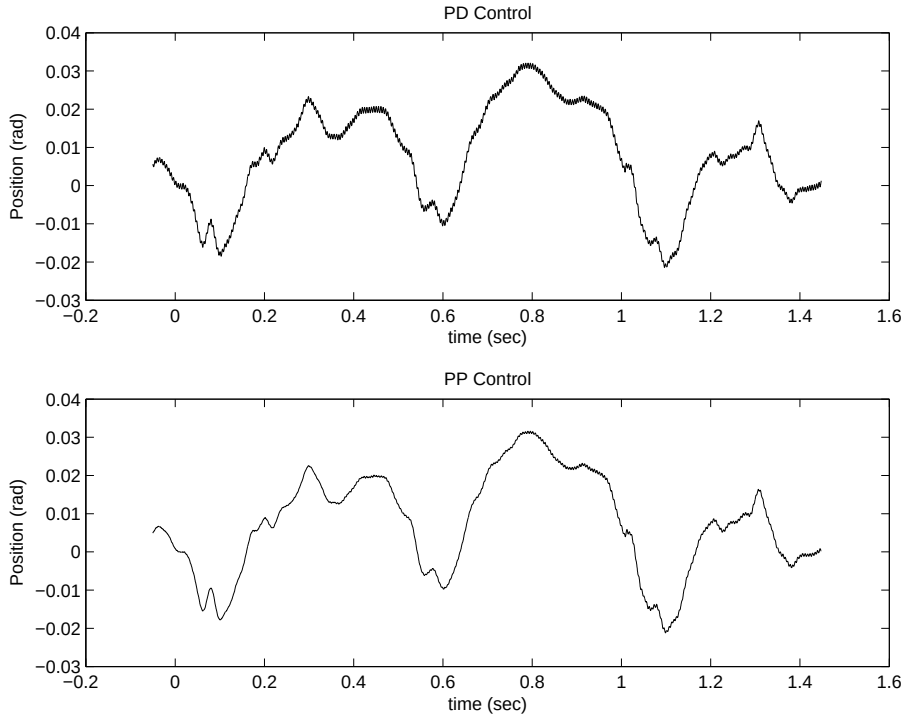



Figure 5.11: PD and PP algorithms controlled system output utilizing identical feedback gains

The RMS error was 1.400 mrad (0.30 mm) for PD and 1.437 mrad (0.31 mm) for pole placement. The control efforts differed by a larger margin with the PD being 0.3566 Nm RMS and pole placement only using 0.1844 Nm RMS. The difference in the controls is evident from Figure 5.11 as the PD plot has more noise associated with the signal than in the pole-placement output.

Since nearly identical results were obtained in this test (as anticipated), attempts to improve the results were taken by editing the position of the poles relative to these seed locations. The PHANToM transfer function contained four poles, which PD control placed as two sets of complex poles.

Another attempt was to place the dominant poles over the zeros of the numerator of the transfer function. This was augmented with a movement of the faster poles to

the real axis and towards the origin. The results of this design were similar to those obtained by PD; the RMS error was 1.406 mrad (0.3 mm).

### 5.2.3 MPC

Experimentally on the PHANToM, the MPC was able to out-perform the other algorithms. The results of the known-future MPC were 0.683 mrad RMS (0.1468 mm) and a corresponding control of only 0.0526 Nm RMS. The results and figures are seen in Figures 5.12 and 5.13.

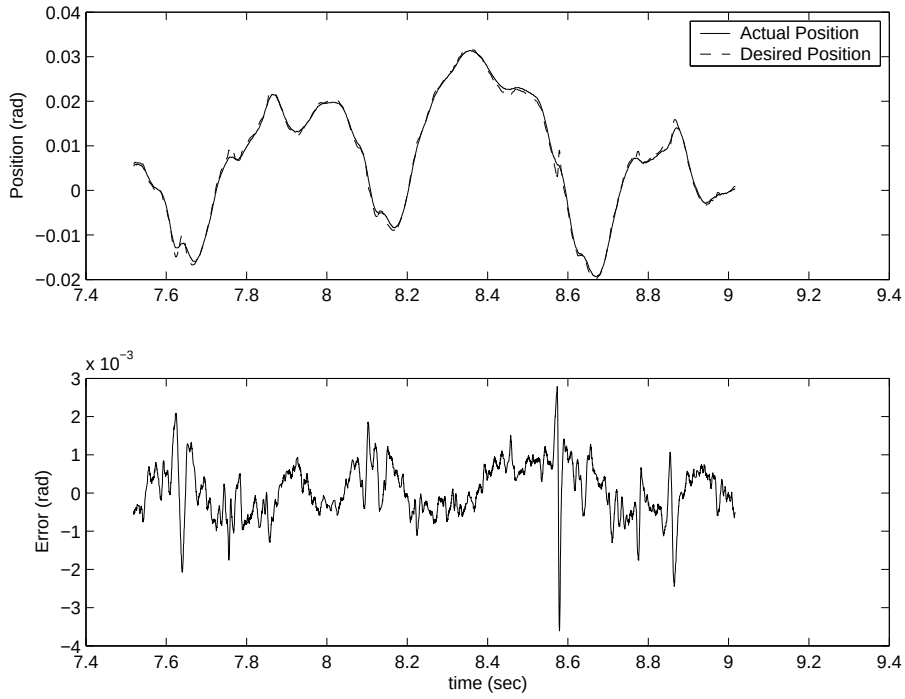



Figure 5.12: PHANToM MPC tracking results and error

These results serve as the relative figure of merit. The 0.683 mrad RMS is 50% more than the desired specification. This may be the best that this particular robot could perform. In addition to achieving the best tracking, this algorithm also had a low control effort. The tracking response was virtually silent for this algorithm and particularly stable.

Oddly, the closed-loop poles that result from this algorithm are not the same as

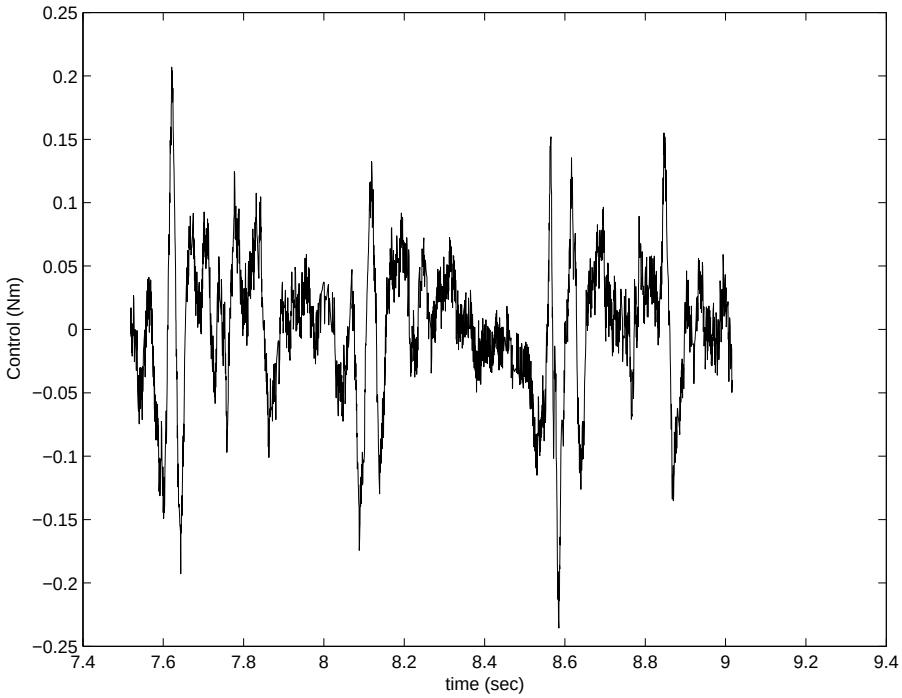



Figure 5.13: PHANToM MPC control effort

those that produced the best results in the pole-placement and PD control methods.

The closed-loop poles for the MPC were calculated and these poles were used as the poles to place in the pole placement algorithm. The resulting system output error and control resembled the results seen in Figure 5.9. The overall response was much more oscillatory and generally undesirable. Since the feedback gains were identical, the major influence must come from the feedforward terms in the algorithms.

A plot of the closed-loop (CL) poles of the control algorithms, relative to the open-loop poles and zeros, is shown in the $z$ plane in Figure 5.14.

Singer et al. performed input-shaping experiments in the early 90's [51]. The idea was to model the system and produce open-loop feedfoward control components that would eliminate unwanted oscillation. Therefore a specific control input is shaped in order to get a desired response. This has been successful in practice [52] and shows that with sufficient system information, it is possible to improve the controlled response without altering the system response through feedback. This in part explains

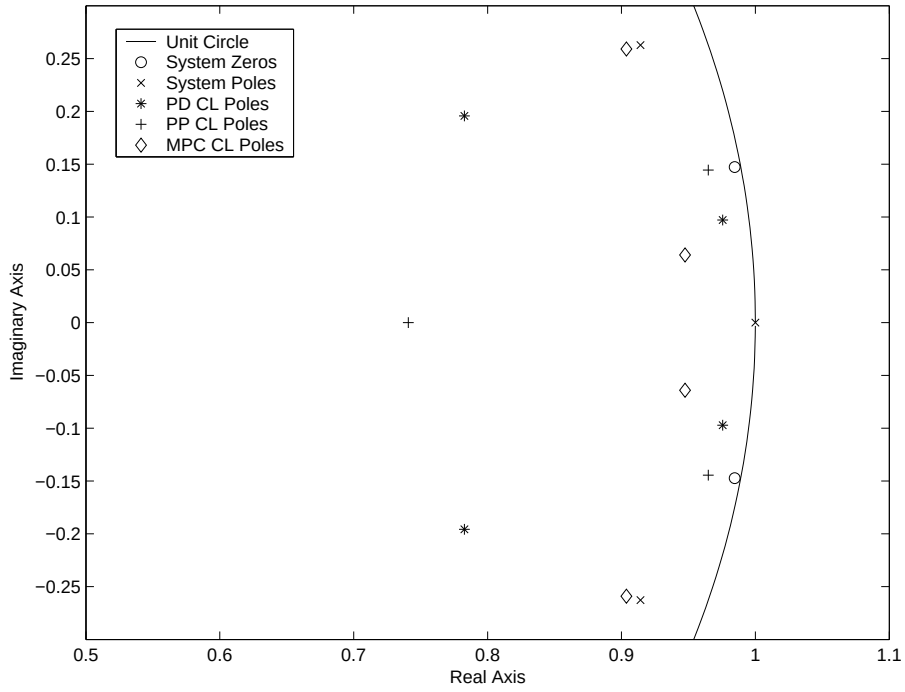



Figure 5.14: Pole Placement By Algorithm

why the MPC algorithm produces such accurate results with the calculated gains. It is taking the system dynamics into consideration along with the desired output and prescribing an appropriate feedforward control.

### 5.2.4 Signal-Estimated MPC

The signal-estimated MPC in this instance did not produce significantly better results than that of the other causal algorithms. The RMS error was 1.088 mrad (0.23 mm) with a corresponding RMS torque of 0.1961 Nm. Plots of these results can be seen in Figure 5.15.

The error-correction parameters for these results corresponded to an error horizon of 350 steps and the error function raised to the 8th power. In general, the same weighting matrices and horizon values were used when switching control algorithms from MPC to Signal-Estimated MPC. In this case, the known-future MPC weighting matrices and horizon did not produce the best results for the Signal-Estimated

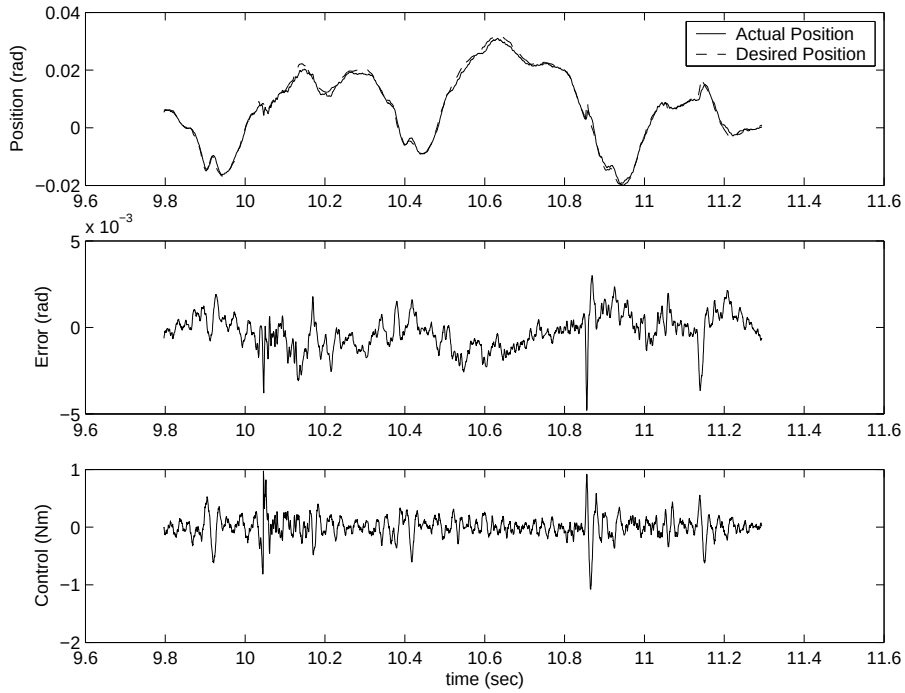



Figure 5.15: PHANToM signal-estimated MPC Results

MPC and as a result, an alternate set of MPC parameters was tuned for the Signal-Estimated MPC. An MPC horizon of 13 steps was used in conjunction with a state-error weighting matrix that was three times the size of previous weighting parameter.

The choosing of parameters in the error-correction function (the error horizon and power of error function) turned out to be much more difficult as well. Several local minima were found before a satisfactory solution was obtained. These minima did not follow an obvious pattern, so any changing of the MPC gains resulted in large changes in the error-correction function parameters.

# Chapter 6

# Analysis and Conclusion

## 6.1 Results Revisited

The complete results of the algorithms have been organized into Table 6.1. The known-future MPC produced the best results in all cases. It was able to maintain a reasonable control effort while showing accuracy below or very near the desired specification. Furthermore, the estimated signal MPC method also produced results better than those of traditional pole placement and PD control methods.

| ***Tracking Results*** | *Simulation* | | *Speaker* | | *PHANToM* | |
|---|---|---|---|---|---|---|
| | Error | Control | Error | Control | Error | Control |
| *Units* | mm | N | mm | V | mrad | Nm |
| *Desired Specification* | 0.10 | - | 0.10 | - | 0.465 | - |
| **PD** | 0.17 | 0.70 | 1.4 | 0.23 | 1.22 (0.262mm) | 0.327 |
| **PP** | 0.15 | 0.97 | 0.82 | 0.37 | 1.37 (0.295mm) | 0.304 |
| **MPC** | 0.023 | 0.93 | 0.28 | 0.33 | 0.683 (0.147mm) | 0.0526 |
| **Estimated Signal MPC** | 0.090 | 0.78 | 0.46 | 0.33 | 1.09 (0.234mm) | 0.196 |

Table 6.1: Complete Results of Simulation and Hardware

## 6.2 Conclusion

The feasibility of motion cancellation during heart surgery has been studied. Though current methods are still raw, it is very likely that an algorithm and system exists that will be able to produce the accuracy restraints needed in order to perform motion cancelled heart surgery.

The estimated signal MPC algorithm showed that it was able to work better than the other algorithms while using an acceptable amount of control. The improvement, though significant in simulation was only "just better" when tested on the actual hardware. The PHANToM improvement could even be considered marginal and as a result, implementation of the algorithm would not be worth the extra effort.

The estimated signal MPC algorithm as a whole proved to be difficult to tune, being sensitive to changes in the MPC horizon or weighting parameters. An in-depth study needs to be conducted in order to optimize such a method for general use. The guess-and-check method implemented was able to produce the desired results, but it is unlikely that these results were optimal. Furthermore, a more theoretical approach to the problem may divulge some insight into the tuning process.

What is undeniable however, is that the known-future MPC algorithm was significantly better than the alternatives tested, both in simulation and in experiments. By using future information about the desired signal, it is possible to obtain improved accuracies.

## 6.3 Future Work

Unfortunately it is not known if the estimated signal MPC algorithm would work with a real surgical setup. The error was low but not as low as what was possible with the MPC method. Generally the accuracy can be attributed to the system first and then the control second. A responsive system will produce good results regardless of what

type of control is used; while a poorly designed system may have trouble producing even reasonable results when the best control algorithm is used.

The MPC algorithm obviously can not work by itself, as it can not produce the future desired trajectory. The estimated signal MPC may be one answer to such a problem, but specific tests on a real surgical system would have to be conducted to determine its overall effectiveness and usefulness.

It is likely that some more information needs to be included in order to produce a better signal prediction for the MPC algorithm. An inclusion of ECG data would be one possible solution. The ECG data is an electric signal passed to the heart, which often indicates when the heart will beat. This information generally can be obtained before the heartbeat occurs and as a result provide some future information about the heart beat signal. Research into this area will provide more insight into this alternative.

As far as the estimated signal MPC algorithm is concerned, as stated previously, the tuning needs to be improved. The method does seem to have some promise and its simplicity in concept is appealing. Likely a mapping between results and the tuning parameters could be developed, which would enable more effective control design.

Another setup could be implemented where the speaker was used in a feedforward-only setup and was treated as a hardware simulated heart. The speaker excursion could be read in by the ultrasonic sensor and then used for trajectory tracking with the PHANToM. This would constitute a more realistic emulation. The tracking and different algorithms could be physically seen. Since the heart signal may not be as good when implemented entirely feedforward, other aspects such as a slight randomness in the signal is added which would probably help see the robustness of the algorithm.

The process could be made more realistic by doing the tracking on the robot on the same axis as the subwoofer. Attaching a camera onto the end of the PHANToM

looking towards the speaker would give some idea to how good the tracking actually is and provide some insight into how difficult the surgery with motion cancellation might be.

With regards to the algorithm, advancements could be made using $\mu$ analysis [53] or $H\infty$ controllers [54]. These control methods may be able to provide more robustness to the algorithm.

## 6.4 Wrap up

This thesis has presented control system simulations and experiments to evaluate prospects for future motion cancellation surgical systems. The results suggest that the dynamic requirements may be within reach of current technology with careful attention to design of the electromechanical system and the control algorithm.

# Bibliography


[1] "American Heart Association," American Heart Association, July 2004. [Online]. Available: http://www.americanheart.org

[2] "Angioplasty.org patient center," Angioplasty.org, July 2004. [Online]. Available: http://angioplasty.org/devices.html

[3] "Coronary bypass graft surgery," St. Jude Medical, July 2004. [Online]. Available: http://www.sjm.com/procedures/procedureindex.aspx

[4] "Emboli - medical dictionary," MedicineNet.com, July 2004. [Online]. Available: http://www.medterms.com/script/main/art.asp?articlekey=15687

[5] D. van Dijk, K. G. M. Moons, A. M. A. Keizer, E. W. L. Jansen, R. Hijman, J. C. Diephuis, C. Borst, P. P. T. de Jaegere, D. E. Grobbee, and C. J. Kalkman, "Association between early and three month cognitive outcome after off-pump and on-pump coronary bypass surgery," *Heart*, vol. 90, no. 4, pp. 431–434, 2004.

[6] A. Chiu, D. Boyd, and T. Peters, "3-D visualization for minimally invasive robotic coronary artery bypass (MIRCAB)," *Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE*, vol. 3, no. 4, pp. 1728–1730, 2000.

[7] H. Paul, B. Mittlestadt, W. Bargar, B. Musits, *et al.*, "A surgical robot for total hip replacement surgery," in *Proc. 1992 IEEE International Conference on Robotics and Automation*, May 1992, pp. 606–611.

[8] R. H. Taylor and D. Stoianovici, "Medical robots in computer-integrated surgery," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 765–781, 2003.

[9] D. H. Boehm, H. Reichenspurner, C. Detter, M. Arnold, H. Gulbins, B. Meiser, and B. Reichart, "Clinical use of a computer-enhanced surgical robotic system for endoscopic coronary artery bypass grafting on the beating heart," *Thoracic and Cardiovascular Surgery*, vol. 4, no. 48, pp. 198–202, 2000.

[10] A. Carpetentier *et al.*, "First computer assisted open heart operation," *Life Sciences*, no. 321, pp. 437–442, 1998.

[11] "Intuitive Surgical and Computer Motion announce merger agreement," Intuitive Surgical, July 2004. [Online]. Available: http://intuitivesurgical.com/news_room/press_releases/pr_030703.html

[12] "Intuitive Surgical," Intuitive Surgical, July 2004. [Online]. Available: http://www.intuitivesurgical.com

[13] M. C. Cavusoglu, J. Ustin, F. Tendick, S. Choi, and S. S. Sastry, "Toward a robotic telesurgical workstation with active relative motion cancelling for off-pump (beating heart) coronary artery bypass graft surgery," 2003, unpublished.

[14] "Octopus System (Starfish and Octupus 3)," Medtronic, July 2004. [Online]. Available: http://www.medtronic.com/cardsurgery/products/mics_octosystem.html

[15] K. Sharma, W. Newman, M. Weinhous, G. Glosser, and R. Macklis, "Experimental evaluation of a robotic image-directed radiation therapy system," in *IEEE International Conference on Robotics and Automation ICRA '00*, vol. 3, 2000, pp. 2913 –2918.

[16] A. Schweikard, G. Glosser, M. Bodduluri, M. Murphy, and J. Adler, "Robotic motion compensation for respiratory movement during radiosurgery," *Computer Aided Surgery*, vol. 5, no. 4, pp. 263–77, 2000.

[17] C. Riviere, A. Thakral, I. I. Iordachita, G. Mitroi, and D. Stoianovici, "Predicting respiratory motion for active canceling during percutaneous needle insertion," in *Proc. 23rd Annual Intl. Conf. IEEE Engineering in Medicine and Biology Society*, October 2001, pp. 3477–3480.

[18] A. Trejos, S. Salcudean, F. Sassani, and S. Lichtenstein, "On the feasibility of a moving support for surgery on the beating heart," in *Proc. Medical Image Computing and Computer-Assisted Interventions (MICCAI99)*, 1999, pp. 1088–1097.

[19] P. W. Mayer, "Relative motion cancelling platform for surgery," U.S. Patent 5,871,017, 1999.

[20] Y. Nakamura, K. Kishi, and H. Kawakami, "Heartbeat synchronization for robotic cardiac surgery," *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 2, pp. 2014– 2019, 2001.

[21] A. Thakral, J. Wallace, D. Tomlin, *et al.*, "Surgical motion adaptive robotic technology (S.M.A.R.T.): Taking the motion out of physiological motion," in *Proc. of 4th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, October 2001, pp. 317–325.

[22] R. Ginhoux, J. Gangloff, M. de Mathelin, L. Soler, J. Leroy, and J. Marescaux, "A 500 hz predictive visual servoing scheme to mechanically filter complex repetitive organ motions in robotized surgery," *Intelligent Robots and Systems, 2003.*

*(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 4, pp. 3361– 3366, 2003.

[23] J. Sackier and Y. Wang, "Robotically assisted laparoscopic surgery. from concept to development," *Surgical Endoscopy*, vol. 8, pp. 63–66, 1994.

[24] R. K. Mehra and J. N. Amin, "Active suspension using preview information and model predictive control," in *IEEE International Conference on Control Applications*, 1997, pp. 860–865.

[25] J. S. Riedel and A. J. Healey, "Model based predictive control of auvs for station keeping in a shallow water wave environment." [Online]. Available: citeseer.ist.psu.edu/397286.html

[26] R. Kalman, "A new approach to linear filtering and prediction problems1," *Basic Engineering*, no. 82, pp. 35–45, 1960.

[27] N. Wiener, *The Extrapolation, Interpolation and Smoothing of Stationary Time Series*. John Wiley and Sons, Inc., 1949.

[28] "Sonometrics Corp." Sonometrics Corp, July 2004. [Online]. Available: http://www.sonometrics.com/

[29] "QNX real time operating system overview," QNX Software Systems, July 2004. [Online]. Available: http://www.qnx.com/products/rtos/

[30] "MTX car audio," MTX Audio, July 2004. [Online]. Available: http://www.mtxaudio.com/caraudio/archive/index.cfm

[31] "Glentek Inc." Glentek Inc., July 2004. [Online]. Available: http://www.glentek.com/

[32] "Cleveland Motion Controls," Cleveland Motion Controls, July 2004. [Online]. Available: http://www.cmccontrols.com

[33] *Pulsonic Non Contact Measuring System*, Cleveland Motion Control, 7550 Hub Parkway Cleveland, Ohio 44125, January 1990.

[34] *Signal Processing Toolbox User's Guide For Use With Matlab*, 6th ed., The Mathworks Inc., 3 Apple Hill Drive Natick, MA 01760-2098, June 2004, pp. 7-365 - 7-368.

[35] "Sensable Technologies," Sensable Technologies, July 2004. [Online]. Available: http://www.sensable.com/

[36] D. Feygin, M. Keehner, and F. Tednick, "Haptic guidance: Experimental evaluation of a haptic training method of perpetual motor skill." in *10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2002, pp. 40–47.

[37] D. Tzovaras, G. Nikolakis, G. Fergadis, S. Malasiotis, and M. Stavrakis, "Design and implementation of haptic virtual environments for the training of the visually impaired," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 12, no. 2, pp. 266–278, June 2004.

[38] N. Dhruv and F. Tendick, "Frequency dependence on compliance contrast detection," in *Proceedings of the Symposium on Haptic Interfaces for Virtual Envrionment and Teleoperator Systems, part of the ASME Int'l Mechanical Enigerring Congress and Exposition*, vol. DSC 69-2, 2000, pp. 1087–1093.

[39] D. G. Carson and D. D. Preonas, "Rotary drive apparatus having one member with smooth outer peripheral surface," U.S. Patent 5,105,672, 1992.

[40] "Maxon precision motors." Maxon Motor USA, July 2004. [Online]. Available: http://www.maxonmotorusa.com/

[41] *Operation and Service Manual for Model GA4555P Brush Type Linear Servo Amplifier*, Glentek Inc., 208 Standard St., El Segundo, California 90245.

[42] M. C. Cavusoglu, D. Feygin, and F. Tendik, "A critical study of the mechanical and electrical properties of the PHANToM haptic interface and improvement for high performance control," *Presence*, vol. 11, no. 6, pp. 555–568, 2002.

[43] *Robust Control Toolbox User's Guide For Use With Matlab*, 2nd ed., The Mathworks Inc., 3 Apple Hill Drive Natick, MA 01760-2098, June 2004, pp. 2-17 - 2-18.

[44] G. F. Frankln, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Addison Wesley Longman, 1998.

[45] E. Camacho and C. Bordons, *Model Predictive Control*. Springer, 1999.

[46] D. Clarke, "Application of generalized predictive control to industrial processes," *Control Systems Magazine, IEEE*, vol. 8, no. 2, pp. 49–55, 1988.

[47] B. D. Anderson and J. B. Moore, *Optimal Control Linear Quadratic Methods*. Prentice Hall, 1990.

[48] "The Mathworks - Simulink - Simulation and model-based design." The Mathworks, July 2004. [Online]. Available: http://www.mathworks.com/products/simulink/

[49] "Meschach," Meschach Library, July 2004. [Online]. Available: http://www.netlib.org/c/meschach/ and http://www.netlib.no/netlib/c/meschach/readme

[50] F. M. Callier and C. A. Desoer, *Linear System Theory*. Springer-Verlag, 1991.

[51] N. Singer and W. Seering, "Preshaping command inputs to reduce system vibration," *Journal of Dynamic Systems, Measurement and Control*, vol. 112, pp. 76–82, March 1990.

[52] W. Singhose and N. Singer, "Effects of input shaping on two-dimensional trajectory following," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 881–887, December 1996.

[53] G. J. Balas, J. C. Doyle, K. Glover, A. Packard, and R. Smith, $\mu$ - *Analysis and Synthesis Toolbox*. Mathworks Inc., 1995.

[54] Zhou, Doyle, and Glover, *Robust and Optimal Control*. Prentice Hall, 1996.