University of Alberta

Nanopositioning Control of an Atomic Force Microscope

by

Hua Han 

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Electrical and Computer Engineering

Edmonton, Alberta
Fall 2004

Your file Votre référence
ISBN: 0-612-95760-8
Our file Notre référence
ISBN: 0-612-95760-8

Chapter 1

Introduction

Over the past decade, the atomic force microscope (AFM) techniques have been dramatically developed, providing an atomic-resolution material imaging and characterization for various fields of science. The key component, namely, the controller, which maintains a constant force between the tip and the sample for AFMs in the contact mode, has been developed increasingly.

Most current control algorithms employed on the AFM controllers are the conventional PID and the H_∞ controllers, which are designed based on the piezo scanner model. Very few designs incorporated the dynamic model of the cantilever due to the complexity of the tip-sample interaction. In this thesis, the macroscopic interaction between the tip and the sample has been examined thoroughly by using the Johnson-Kendall-Roberts (JKR) force model. In addition to this, three feedback controllers have been designed based on the comprehensive plant model which integrates the

piezo, the cantilever and the photodiode to obtain precise and high speed imaging. Finally, the experimental implementation of the designed controllers is discussed in detail.

1.1 Thesis Objectives

The goal of this thesis is to utilize the modern control H_∞ theory to achieve high-speed high resolution image of an atomic force microscope (AFM). To achieve this, the mathematic models of all components of the AFM are constructed and analyzed, and implementation of the controllers is performed.

1.2 Thesis Organization

This thesis is comprised of six chapters that detail the experimental and theoretical work performed and three appendices which include the programs of the designed controllers. Chapter 2 introduces the background knowledge and operation principles for AFMs working in the contact mode.

In Chapter 3, mathematical models of all components of the IBM Magnascope 2000 AFM, namely, the piezoelectric transducer, the cantilever and the photodiode, are developed based on experimental and theoretical studies. For this, the step response of the system combined with the macroscopic interactions between the tip and surface is analyzed. This plant model is used to design high performance con-

trollers which will provide more precise and faster position control of the AFM than the traditional controllers.

Three different controllers, namely, a phase-lag controller, a traditional PID controller and an H_∞ controller, are designed in Chapter 4 for the purpose of precisely tracking arbitrary input signals (the sample surface) with high bandwidth. Simulation results are compared by using MATLAB.

In Chapter 5, the experimental implementation of the designed controllers is addressed, which includes setting up the communication between the existing TMS320C32 DSP and the chosen DSP, the ADC board and the chosen DSP which stores the software programs of the controllers that have been designed in Chapter 4.

Chapter 6 summarizes and concludes the work performed in this thesis and provides a list of future tasks.

Chapter 2

Background

This chapter provides the background knowledge and principles required to understand the basic operation of an AFM. The chapter begins with a historical review of AFM development and the literature review of AFM controller design. The detailed description of each component of an AFM is followed. Finally, the principle of an AFM working in contact mode is introduced.

2.1 AFM History

The atomic force microscopy (AFM) has roots in scanning tunneling microscopy (STM) which was invented by IBM researchers, Gerd Binnig and Heini Rohrer, in 1981. In STM, the topography of surface electronic state is measured by using the tunneling current which is a function of the separation between the tip and sample. The limitation of STM is that it works only on electrically conductive samples.

The first AFM result was published in 1986 by Binnig *et. al* [3], in which they placed the STM tip which acted as a cantilever in a parallel direction so that its sharp edge was just above the surface. The high resolution image of the sample is achieved by measuring the minute deflection of the cantilever with its force-sensing edge while it scans the surface of the sample, which is not necessarily conductive. The spring constant of the cantilever should be smaller than the equivalent spring constant between the atoms on a surface so that the tip would not displace the surface atoms. A cantilever which is made of metallic wires with a length of several hundred micrometers and a diameter on the order of 10 microns will satisfy this requirement. The microfabricated Si, SiO_2 and Si_2O_3 cantilevers turned out to be a much better choice which had smaller spring constant and higher resonance frequency that enabled rapid raster scanning.

The next step in the development of AFMs came about when the optical methods for monitoring the deflection of the cantilever were developed by Meyer in 1988 [16]. The optical lever operates by bouncing a laser beam off the cantilever. The reflected laser beam strikes a position-sensitive split photodiode and the difference between the two photodiode signals indicates the angular deflection of the cantilever. Nowadays, most commercial AFMs employ optical sensors.

The first generation of AFMs (Binnig *et.al* 1986 [3], Rugar and Hansma 1990 [19]) operated in contact mode, that is, the tip mounted on the end of a flexible cantilever scans the sample surface. Since the lateral force exerted on the sample surface during

the scanning is quite large, soft biological samples can be easily damaged and therefore are difficult to be scanned using contact mode.

The non-contact mode which was first introduced in 1987 (Martin [15]) achieves more accurate images of soft biological samples. In the non-contact mode, the system vibrates a stiff cantilever close to its resonant frequency with magnitude of a few tens of angstroms at a distance ($1 - 10$ nm farther than that in contact mode) above the surface of the sample. The system keeps the resonant frequency or the vibrational amplitude of the cantilever constant, therefore keeps the tip-sample distance constant. This mode is desirable for soft or elastic samples since the total forces between the tip and sample are very low, generally about $10^{-12}N$.

The tapping mode was first developed in 1993 by Zhong *et. al* [28], which is the common mode used in AFMs nowadays. In this mode, the cantilever is oscillated at its resonant frequency (often hundreds of kilohertz) and positioned above the sample surface so that it only taps the surface for a very small fraction of its oscillation period. It still contacts with the sample but the lateral forces are dramatically reduced. The tapping mode maybe a far better choice than contact mode when imaging soft samples.

The ability of an AFM to produce images of atomic-scale resolution depends, among other factors, on a properly designed control system, which in tern depends on good mathematical models of the components involved.

Traditionally, a proportional and integral (PI), or proportional, integral, and deriv-

ative (PID) control strategy is employed to keep the vertical imaging force constant. These kinds of control systems enable AFMs to image the sample topography at high resolution but with relatively low speeds – it could take several minutes to acquire larger images at the micrometer scale. This would make it difficult for high-speed applications where the response speeds of positioning are crucial; for example, in the field of cell biology, scientists employ nano-probes to track events in the cell, which often have a time scale of microseconds. This motivates us to design high-speed AFM positioning control systems.

Low scan speeds of AFMs are mostly due to the low response speeds of the PI/PID feedback systems which maintain imaging forces constant. Several approaches have been taken to enable faster image acquisition. For an AFM with a stack based piezo configuration, a model-based H_∞ feedback law was employed by Salapaka *et.al.*[20] to increase the response speed of the control system and tracking accuracies of the z-axis piezo movement. The bandwidth of the system controlled by the H_∞ controller was 138 Hz, which showed an enormous improvement over the PI controller (bandwidth $<$ $3Hz$). By choosing appropriate weighting functions, the robust controller also showed improvements in eliminating the hysteresis, creep and nonrepeatability effects of the piezo scanner.

An H_2 and an H_∞ controller were designed by Hatch *et.al.* [12] for high speed AFMs while PID control designs cannot handle the increasing nonlinearities of the piezo actuators at high frequencies. The modeling of the piezo scanner incorporated

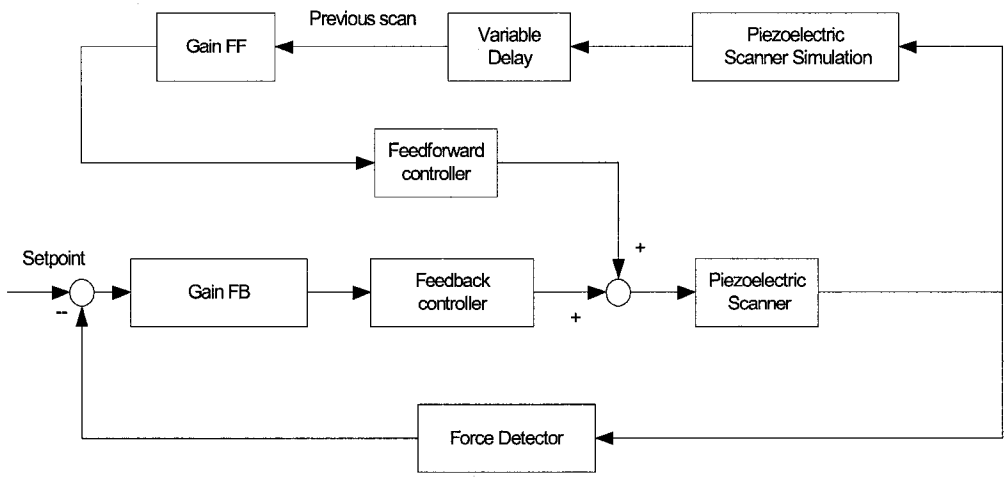

Figure 2.1: Block diagram of the 2-degree-of-freedom-controlled AFM system in [22].

both the dynamic and the hysteresis behaviors of the actuator. The microscopic relation between the applied voltage and the polarization was described by Helmholtz energy. The inverse of the constructed hysteresis model was employed as a filter in robust control design. Thus, the hysteresis effect of the piezo was compensated. Numerical examples showed that both controllers improved the tracking accuracies.

A new control strategy was implemented in [22], in which a two-degree-of-freedom (2DOF) controller was designed to improve the performance of a contact-mode AFM. The 2DOF controller consisted of an H_∞ feedback controller, which was used to maintain the constant tip-sample interaction force, and a feedforward H_∞ controller, which tracked the z-axis of the piezoelectric tube scanner to the topography of the previous scan line. The system block diagram is shown in Figure 2.1. Here, gain FB is used to adapt changes in the gain of the optical deflection detection system and gain

FF is used to define the rate of the feedforward trajectory applied to the piezo scanner. The combination of the feedback and feedforward law reduced the measurement error and enabled higher imaging speed. However, for the purpose of designing feedback controllers, effect of the cantilever was ignored in modeling. In this thesis, an overall AFM mathematical model is obtained, which incorporates a dynamic model for the cantilever when the AFM is in the contact mode. The following section will show how a typical contact-mode AFM is constructed and how it works.

2.2 AFM Instrumentation

As shown in Figure 2.2, an AFM consists of a sharp tip suspended by a cantilever, an optical detector to sense the displacement of the cantilever due to the interaction forces, a piezoelectric scanner and a control unit. An ideal AFM device should achieve tip-to-sample position control with nanometer precision, fine positioning capability in three dimensions and a scanning speed as high as possible. All these requirements have to be satisfied in the presence of electric noise, creep and hysteresis of piezoelectric scanner and other disturbances. In this section, the characteristics of each component of the AFM are discussed.

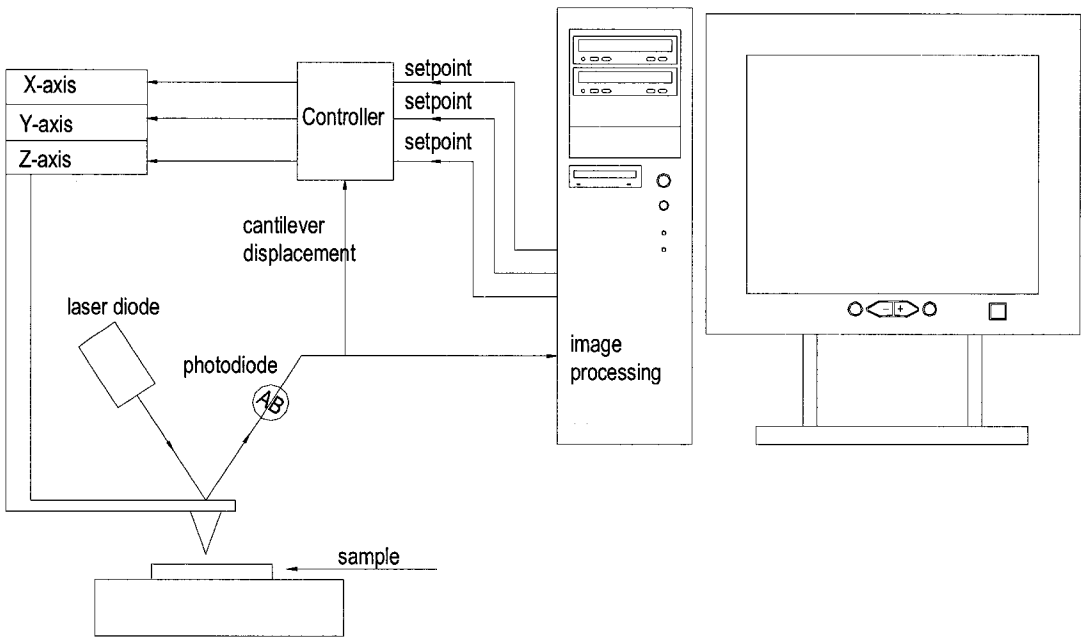

Figure 2.2: All components of an atomic force microscope.

2.2.1 Piezoelectric Ceramics

Piezoelectric material undergoes a dimensional change when it is placed in an electric field. This means when we apply a voltage across the piezo stacks, the piezo will expand or contract, as illustrated in Figure 2.3. The amount of motion and direction of motion depends on the type of piezoelectric material, the shape of the material, and the field strength.

More recently, with the development of new ceramic technologies, polycrystalline ceramic materials such as lead zirconate titanate (PZT) and barium titanate have been extensively developed for best performance in their piezoelectric properties.

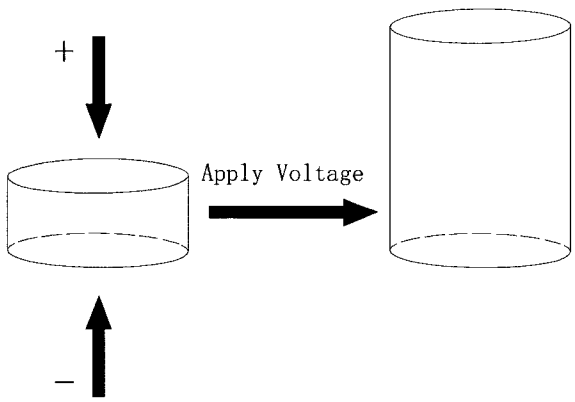

Figure 2.3: Piezo stack extends after being applied voltage.

These materials can convert electric voltage signals of $1mV$ to $1000V$ into mechanical motion in the range from less than one angstrom to several micrometers [2].

The piezoelectric scanners are characterized by the scan range and resonant frequency [21]. The scan range depends on the piezoelectric material, the scanner dimensions, and the applied voltage. It is better to increase the piezoelectric coefficient instead of the size of the scanner to increase the scan range. The resonant frequency associates with the scanning rate and determines the system stability in the presence of mechanical vibrations, and affects the electronic noise level in the system. An ideal AFM should achieve the higher scanner resonant frequency in system design.

The PZT piezoelectric scanners are in a variety of shapes and configurations, such as tubes, plates, stacks, bimorphs and strips. Most AFMs employ the piezoelectric tubes or stacks as the scanner. In a tube scanner, as shown in Figure 2.4, the outer or inner metal surface electrode is separated into four sections, which are electrically

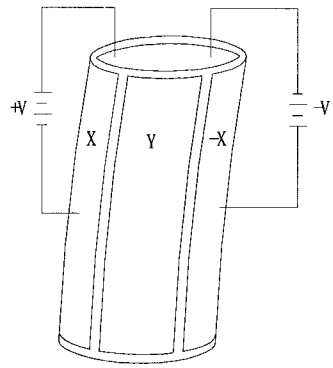

Figure 2.4: Schematic drawing of a piezoelectric tube scanner.

isolated from each other. The voltage is applied across the tube to move it laterally. For vertical motion, the voltage is between the inside and outside of the tube. However, there is the potential for coupling in different directions of positioning in this tube configuration.

The piezoelectric stack is the assembly of hundreds or thousands layers of piezoelectric disks in Figure 2.3. Applying a voltage to the top and bottom surface of this piezo stack causes the entire stack to expand. The stack piezos are cheaper and yield much larger scan range compare to the tube piezos. In the stack piezo based design, such as the IBM MagnaScope 2000 AFM in our laboratory, scanners in the x, y and z directions employ separate piezo actuators. Therefore, the stack piezo configuration has no coupling between x, y and z axises. However, the stack piezo shows more nonlinearities than the tube piezo.

In the IBM MagnaScope 2000 AFM, two MTP-75 piezoelectric translators from

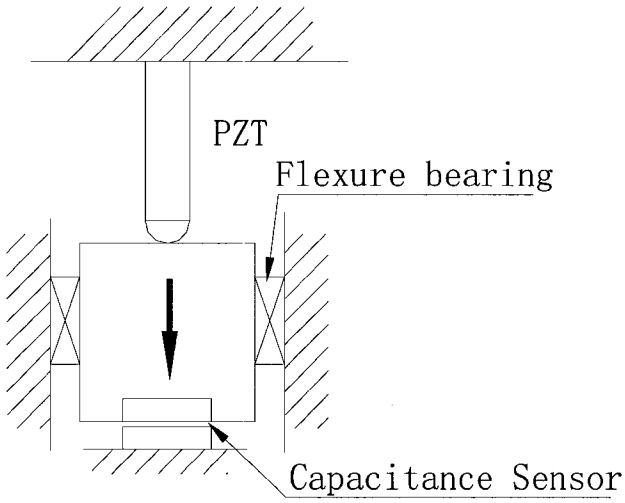

Figure 2.5: Structure of z-direction piezoelectric stage.

Queensgate Instruments Ltd. are used as actuators for x and y direction scanning respectively. The MTP actuator is a low-voltage piezoelectric translator and is capable of providing effectively infinite resolution, limited only by the noise on the drive voltage. Each actuator which has a scan range of $75\mu m$ is internally pre-loaded to ensure that a high pulling force can be obtained as well as a strong pushing force. The slewing speed is limited by the drive current of the electronics. Inherent hysteresis of 10% - 15% limits its accuracy when used without sensors while operates in the open-loop mode.

The piezo scanner in z direction is of our interest since the performance of this scanner has significant effects on the system precision of imaging on the nanometer scale. To compensate the nonlinearity and hysteresis of the piezo, the NPS-Z-15B

from Queensgate Instruments Ltd. is used in the IBM MagnaScope 2000 AFM. The NPS-Z-15B is a piezo scanned flexure guided stage with an integrated capacitance position sensor.

The capacitance sensor is a highly linear sensing device with linearity errors of $< 0.02\%$ over its specified operating range (normally between $100\tilde{}500\mu m$). It is fabricated from very stable materials like SuperInvar. To achieve a pure single axis motion, a flexure guiding mechanism is used, which implements constraints to any off-axis motions and combines the piezoelectric device and sensor together to form an integral stage system as shown in Figure 2.5. In the diagram, the displacement of the piezo measured by the sensor is fed back to the controller, which moves the piezo to minimize the difference between the sensed position and the command. In this case, the positioning precision in the control loop is mainly determined by the capabilities of the sensor and controller.

2.2.2 Force Sensor in z-Direction

The construction of an atomic force microscope requires a force sensor to measure the forces between a small tip and the sample being imaged. In IBM MagnaScope 2000 AFM, the force sensor consists of a cantilever, a tip, a laser diode and a photodiode as shown in Figure 2.6.

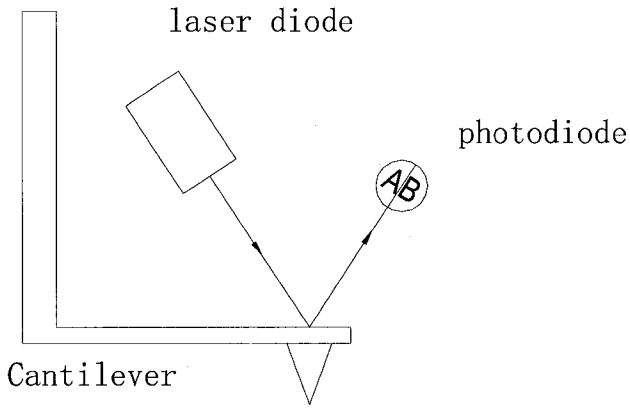

Figure 2.6: The force sensor uses a laser beam to monitor the deflection of the cantilever. When the cantilever moves up and down, the laser beam moves across the surface of the photodiode.

Cantilever

The relationship between the motion of a cantilever and the applied force is given by Hook's law, in the idealized case of a spring and assumes that the force increases linearly with distance:

$$F = -k * s \tag{2.1}$$

where k is the spring constant of the cantilever which depends on the material and dimensions of the cantilever, s is the displacement of the cantilever from equilibrium (where $F = 0$), and F is the force generated by the cantilever. For the cantilever we used in our IBM MagnaScope 2000 AFM, the spring constant k is about $0.2N/m$, which means for a cantilever movement of 1 nm, a force of $0.2nN$ is required.

A typical cantilever of most AFMs which operates in contact mode is made of silicon nitride. In the contact mode, in order to detect the small tip-sample force

(10^{-7} to $10^{-9}N$)[30], the cantilever must flex with a relatively low force constant (usually $< 1N/m$). Otherwise, the attractive force between the tip and the sample will not pull the tip to snap into the sample.

Mechanical resonant frequency of the cantilever is another factor we should take into account while building an AFM. Fast scanning rate is desirable in the design of AFM because the effects of thermal drift are more pronounced with slow scanning rate. If the scanning rate is too high or the cantilever resonant frequency is too low, the inertia of the cantilever will cause the tip to exert large force on the projecting parts of the sample and prevent the tip from tracking down the pit and reaching the bottom. In this case images of the sample will be distorted. Therefore, the AFM cantilever in contact mode should have a resonant frequency higher than 10kHz. The following equation shows that the cantilever can have both a low spring constant and a high resonant frequency if it has a small mass:

$$m = \frac{k}{\omega_0^2} \tag{2.2}$$

with m the mass of the cantilever reduced to its free end [21]. Therefore AFM cantilevers tend to be very small.

The resonant quality factor of the cantilever, known as the Q factor, is one of the important parameters in modeling the dynamics of the cantilever, which will be discussed in detail in Chapter 3. The Q factor is defined by [23]:

$$Q = \frac{\omega_0}{\Delta\omega} \tag{2.3}$$

where ω_0 is the resonant frequency of the cantilever and $\Delta\omega$ the bandwidth at $1/\sqrt{2}$ of the resonance peak height. This Q factor depends on the material and shape of the cantilever and can be changed by the surrounding medium. Increasing the Q factor will help to improve the signal-to-noise ratio. However, since the system damping time τ is the ratio of Q and the nature frequency of the cantilever [5], high quality factor will result in slow scanning rate and narrow detection bandwidths that complicate signal detection. All these consideration will be taken into account while designing the feedback controller which is based on the models of each component of the AFM.

Tip

Most tips are made of the same material as the cantilever. Currently, silicon micromachining technique is employed extensively in building AFM tips. There are two major types of tips: tips used in contact mode which is typically made of silicon nitride and tips used in non-contact mode usually made of silicon. The two basic shapes available are the pyramidal tip as shown in Figure 2.7 (up) which is most common for silicon nitride tips and the conical tip in Figure 2.7 (bottom) for silicon tips.

The AFM tip is characterized by its end radius r, the sidewall angles θ and its length l as shown in Figure 2.8. The tip's sidewall angle and length influence its ability to accurately image rough surfaces with deep and narrow holes. For example,

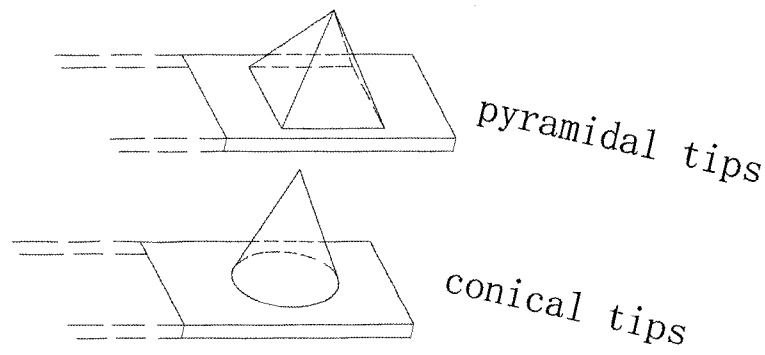

Figure 2.7: Typical shapes of cantilever tips [4].

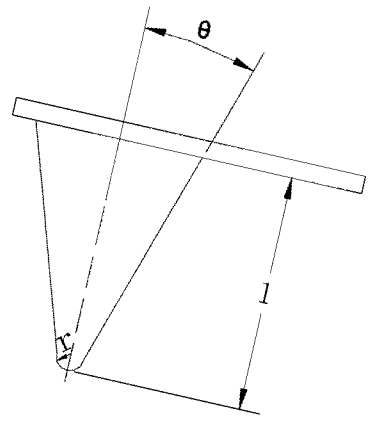

Figure 2.8: The AFM tip with tip length l, end radius r and sidewall angle θ [4].

to accurately image a $1\mu m$-deep pit on the sample surface, the tip must be longer than $1\mu m$. Furthermore, the tip should be narrow enough that it can completely go through the pit and reach the bottom; otherwise, the image of the sample would be distorted or underestimated.

The common silicon nitride tip used in contact mode has a nominal end radius between 20 to $60nm$ (the tip of our AFM has a end radius of $30nm$) and rather large sidewall angle of 35°. Therefore, these tips are well suited for relatively flat surfaces rather than rough samples with deep and narrow valleys.

2.2.3 Displacement Sensor in x and $y-$ Direction

To measure the AFM movement in x and y directions, a capacitance NanoSensor NX from Queensgate Instrument Ltd is used. The NX sensor is combined with the MTP piezoelectric actuator and the NPS 3330 controller to form a closed-loop control system. This sensor is a non-contact position measuring system based on the principle of capacitance micrometry. Two sensor plates (a Target and a Probe) form a parallel plate capacitor. The NX sensor has linearity error less than 0.02% and bandwidth up to 5 kHz.

2.2.4 Control System

When designing a control system for an AFM which moves piezoelectric driven cantilever within $100\mu m$ range with $0.1nm$ resolution requires a computer with more

than 20 bits resolution to implement the controller. A digital signal processor (DSP) can provide 32-bit resolution which meets the control need of the piezo positioning in the AFM. In the control system, a DSP-based controller NPS 3330 from Queensgate Instrument Ltd has been integrated into the design. The feedback control system for the AFM movement in the z-direction consists of the following equipment:

1. TMS320C6701 EVM DSP board from Texas Instruments in which the implementation of the designed H_∞ controller is stored.

2. NPS 3330 controller with built-in PID algorithm to implement the closed-loop control and move the piezoelectric stage to the command position. In NPS 3330, the linearity error of the piezo has been compensated to within 0.02%.

3. The high voltage amplifier which is integrated into NPS 3330 to drive the piezoelectric stage.

4. The photodiode as the motion sensor feedbacks the actual displacement of the cantilever to TMS320C6701 board.

5. ADS8364 Analog to Digital Converter EVM board converts the voltage signal from the photodiode to digital signals that the DSP board can recognize.

6. A computer which takes the signal from the control system for data acquisition and image processing. It is also used for setting the scanning parameters such as scan size, scan speed and feedback control parameters if the controller is

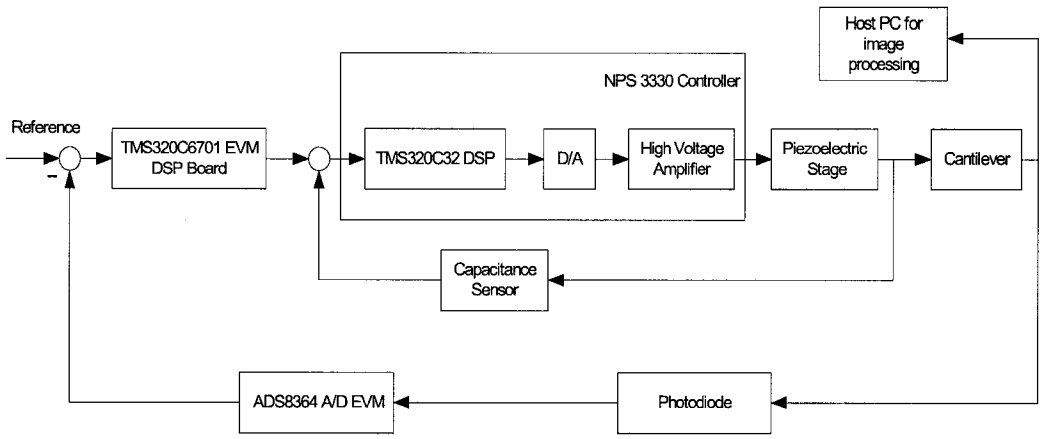

Figure 2.9: Block Diagram of control system for z-axis movement.

designed using PID algorithm.

The system diagram is show in Figure 2.9. The control system will be discussed in detail in Chapter 4.

For piezoelectric translators moving in x and y directions, two independent closed-loop PID control algorithms have been implemented in the TMS320C32 DSP of the NPS 3330 controller. The system diagram is illustrated in Figure 2.10.

2.3 Basic Operation of AFMs

When an AFM operates in the contact mode, the piezoelectric scanner moves the cantilever and the tip (or the sample) in x and y-direction. The cantilever is displaced in z-direction due to the interaction forces between the cantilever tip and the sample.

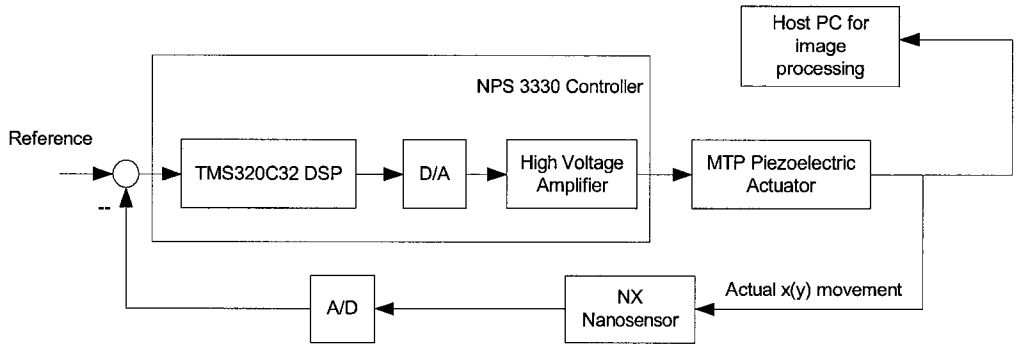

Figure 2.10: Block Diagram of control system for x(y)-axis movement

The cantilever displacement is detected by the deflection of the laser beam which is bounced off the back of the cantilever and into the position-sensitive photodiode. The photodiode transforms the displacement of the cantilever into voltage signal and sends it to the controller. The controller can either compensate the cantilever displacement by adjusting the z-extension of the piezo scanner (constant-force mode) or simply records the signal from the photodiode (the so-called constant-height mode). The signal from the control loop is processed by a computer and a three dimensional image of the sample surface topography is constructed.

Since the constant force mode can achieve a sample surface image with higher accuracy than the constant height mode, it is the most commonly used mode for topography measurement. The current controller design is based on the model of the AFM in constant force mode. Therefore, the vertical force exerted on the sample surface by the tip, that is, the tip-sample interaction should be examined thoroughly.

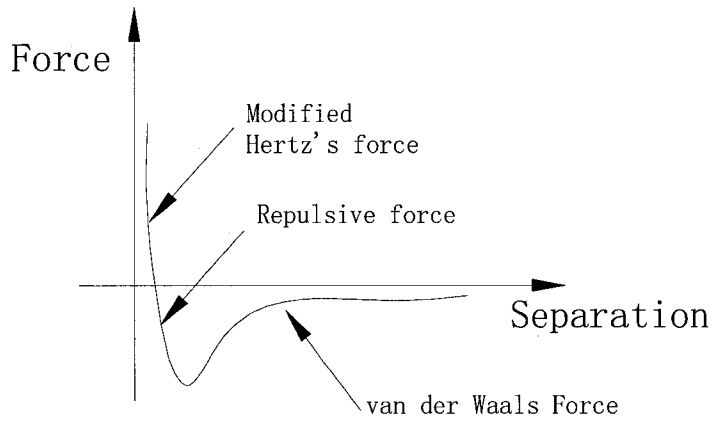

Figure 2.11: The force as a function of tip-sample separation.

van der Waals Type Force

At a distance of a few nanometers, the van der Waals force plays a major role in the fundamental interaction between the tip and the sample. Here, the weak force acting between the atoms at the very end of the tip and those atoms of the sample which are in contact with the tip will be examined, as illustrated in Figure 2.11. The van der Waals forces describe the intermolecular attraction forces which hold an individual molecule and the neighboring molecules together. The van der Waals forces consist of three components: dispersion, polarization and induction. Polarization refers to permanent dipole moments while induction force refers to the contribution of induced dipoles. The van der Waals dispersion force which will be discussed refers to the intermolecular attraction due to instantaneous fluctuations of dipoles of the molecular. This force is called the dispersion force because the electronic movement

that gives rise to it causes dispersion of light, that is the variation of refractive index of a substance with the frequency of the light. This force is attractive, that is, tip and sample are attracted to one another when the tip-sample separation is greater than $10\mathring{A}$, and is present regardless of the tip/surface setup used.

The van der Waals dispersion force has been established theoretically and experimentally by Israelanchevili [13]. The analytical expression for van der Waals potential is quite complicated. For simplicity the potential between two atoms can be written as:

$$W_6(z) = -\frac{C_6}{z^6} \tag{2.4}$$

where

$$C_6 = 4\epsilon\sigma^6 \tag{2.5}$$

is the interaction constant known as the London coefficient and is specific to the identity of the interacting atoms, z is the distance between two atoms, ϵ and σ are interaction parameters and determined by the materials of the tip and the sample. By assuming that additivity is permissible, we can get the macroscopic interaction between a sphere (tip end) and a plane (sample) by integrating the potential in Eq.(2.4):

$$W_6(z) = -\frac{AR}{6z} \tag{2.6}$$

Here, the Hamaker constant A depends on the geometry and materials of the tip and sample and is defined as

$$A = \pi^2 C\rho_1\rho_2 \tag{2.7}$$

where ρ_1 and ρ_2 are the number densities of the tip and the sample, C is the London coefficient, and R the end radius of the tip. By substituting Eq.(2.5) into Eq.(2.6), the van der Waals potential can be written as

$$W_6(z) = -4\pi^2\epsilon\rho_1\rho_2 R\frac{\sigma^6}{6z} \tag{2.8}$$

and the corresponding force is obtained by differentiating Eq.(2.8)

$$F(z) = -\frac{2}{3}\pi^2\epsilon\rho_1\rho_2 R\frac{\sigma^6}{z^2} \tag{2.9}$$

Repulsive Force

When the distance between the tip and the sample is less than the order of about $3\mathring{A}$ [21], the quantum-mechanical force which derives from Pauli's exclusion principle will dominate. Basically, Pauli's exclusion principle states that no two identical electrons in an atomic structure may occupy the same quantum state [17,18]. It is found experimentally that this repulsive force grows very steeply as the interatomic distance reduces, as seen in Figure 2.11. We can use the power-law potential to model the repulsive potential between two atoms:

$$W_{12}(z) = \frac{C_{12}}{z^{12}} \tag{2.10}$$

where the London coefficient is defined as

$$C_{12} = 4\epsilon\sigma^{12} \tag{2.11}$$

The parameters in Eq.(2.11) have the same meanings as in Eq.(2.5). Therefore, the total repulsive macroscopic interaction between the tip end and the sample surface

can be obtained by integrating the potential in Eq. (2.11):

$$W_{12}(z) = 4\pi^2 \epsilon \rho_1 \rho_2 R \frac{\sigma^{12}}{1260 z^7} \tag{2.12}$$

and the corresponding force is given by differentiating Eq.(2.12)

$$F(z) = \frac{2}{3}\pi^2 \epsilon \rho_1 \rho_2 R \frac{1}{30} \frac{\sigma^{12}}{z^8} \tag{2.13}$$

The above discussed van der Waals force model and the repulsive force are components of the Johnson-Kendall-Roberts (JKR) force model which can be shown in Figure 2.11. The third part of the JKR model, namely, the modified Herz's model will be addressed in Chapter 3 to construct the state-space model of the cantilever.

Chapter 3

System Modeling

Modelling the AFM in contact mode is a process of several stages and as such this chapter is split into three sections. The three sections can be summarized as follows:

- The Piezoelectric Transducer - establishing a realistic model of the piezo stage from experiment and theory.
- Cantilever - starting from explicitly modelling the macroscopic interactions between the tip and surface, then deriving the state-space model of the cantilever based on the tip-sample interaction.
- Modelling Photodiode - modelling the static behavior of the photodiode from experiment.

3.1 Modeling of Piezoelectric Transducer

As mentioned in Chapter 2, the piezoelectric stage which moves the cantilever in the $z-$direction is of importance since the image of the sample surface is achieved by detecting the vertical motion of the cantilever. The controller design will be partly based on the model of the piezo stage. The physical modeling of the piezo scanner is too complicated due to the nonlinear sensitivity, hysteresis, thermal drift and creep that present in the piezoelectric materials. However, to model the dynamic performance of the piezo stage, we will first simplify the system structure to a linear, second order model, then obtain the key parameters of the system model by using the measured response of the system to a step input signal without taking the system's material properties into account.

3.1.1 Data Collected from NPS 3330

In the open loop, the piezo stage can be simply considered as a mass – spring – damping second order linear system [25]. The response of the stage is generally dominated by the stage resonant frequency, which mainly depends on the mass of the moving part and the stiffness of the system. The transfer function of the system is given as:

$$G(s) = \frac{k\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{3.1}$$

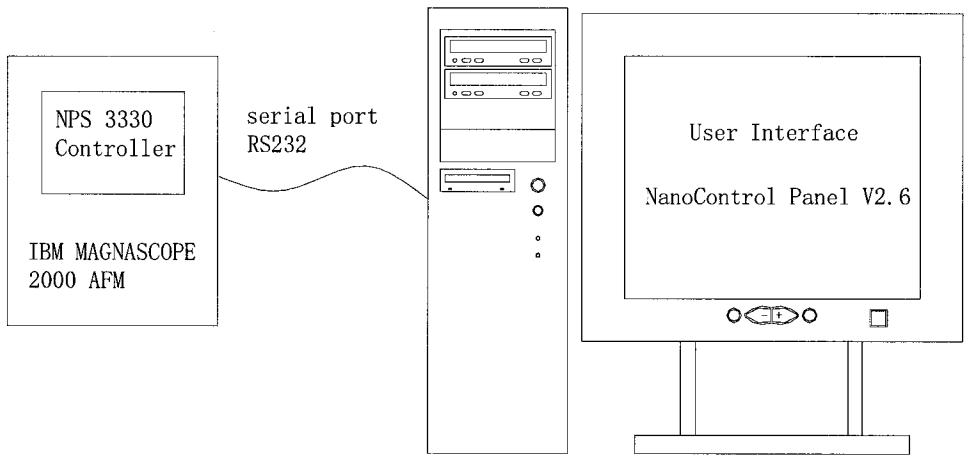

Figure 3.1: System setup for collecting data.

where k is the system gain, ω_n is the natural frequency and ζ is the damping ratio. The ζ value depends on the energy loss in the system, namely, damping. The value of ω_n and ζ will be determined by using the system identification method in Section 3.1.3.

To simulate the open-loop response of the system, the NanoControl Panel Software Version 2.6 of Queensgate is used. The system setup is illustrated in Figure 3.1 and the open-loop response is shown in Figure 3.3. The step signal lasted from $t = 0ms$ to $t = 8ms$. The starting point of the step response is not zero because of the drift of the piezo while the closed-loop mode is off. To compensate the drifting, we add the offset value back to the response curve as in Figure 3.2. All our analysis is based on the modified step response.

The value of k can be determined by applying a small step signal with magnitude

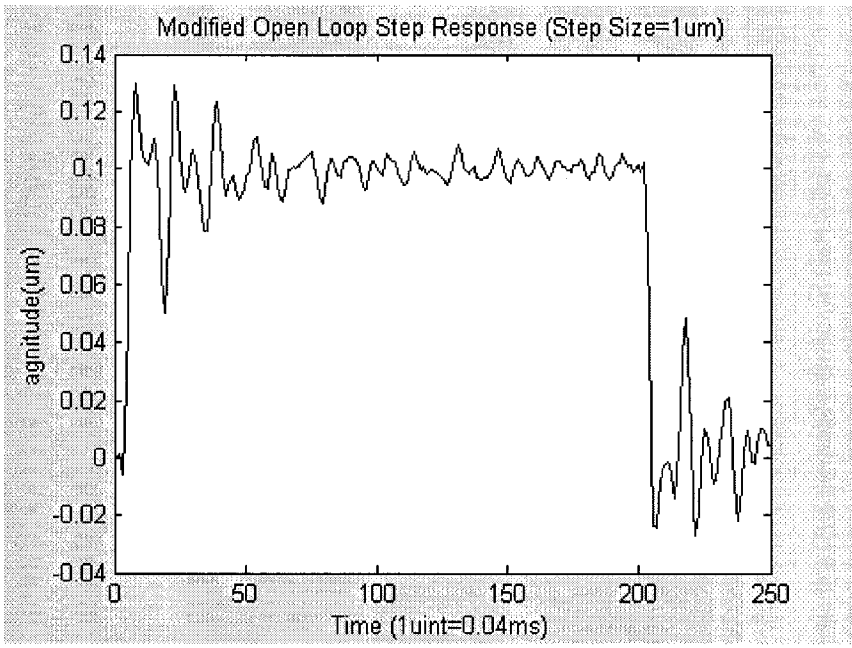

Figure 3.2: Data Collected from NPS 3330-open loop response of piezo stage with step input $1\mu m$.

of $1\mu m$ to the stage and measuring the corresponding output. It is calculated by:

$$k = \frac{y_\infty - y_0}{u_\infty - u_0} = 0.1 \tag{3.2}$$

where y_∞ and u_∞ are the final values of the output and input; and y_0 and u_0 are the initial values respectively.

3.1.2 Data Preprocessing

As examine the step response of the piezo stage closely, it is noticeable that there exists two undesirable peaks in each cycle, which indicates high frequency noise of the system. The noise in the closed-loop system as shown in Figure 2.9 (the inner

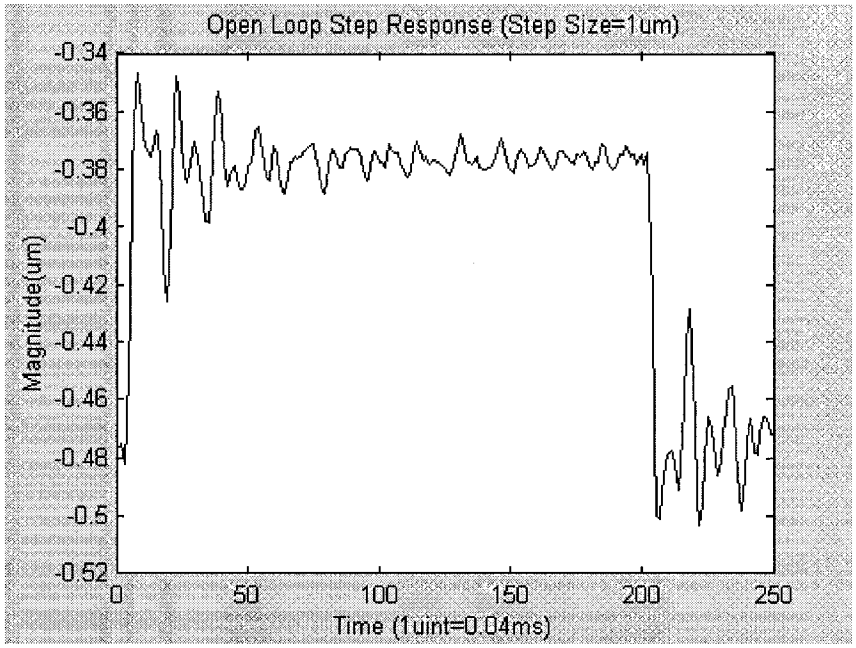

Figure 3.3: The open-loop response of the piezo stage to a step signal with magnitude of $1\mu m$.

loop) is, in general, composed of sensor, piezodrive, mechanical and acoustic noises. Since the step response is taken under the open loop condition, piezo drive voltage noise will mainly contribute to the stage positioning noise. Also, external mechanical inputs such as ground vibration and acoustic noise will cause the stage to move thus increase the noise level.

To minimize the effect of these noises, the original step response has been passed through a low-pass filter in the MATLAB identification toolbox. The low-pass filter is a standard IIR (Infinite Impulse Response) fifth-order Butterworth type filter. The IIR filter refers to the fact that its impulse response has infinitely many nonzero entries. It is known that the higher the order of the filter, the more closely the filter

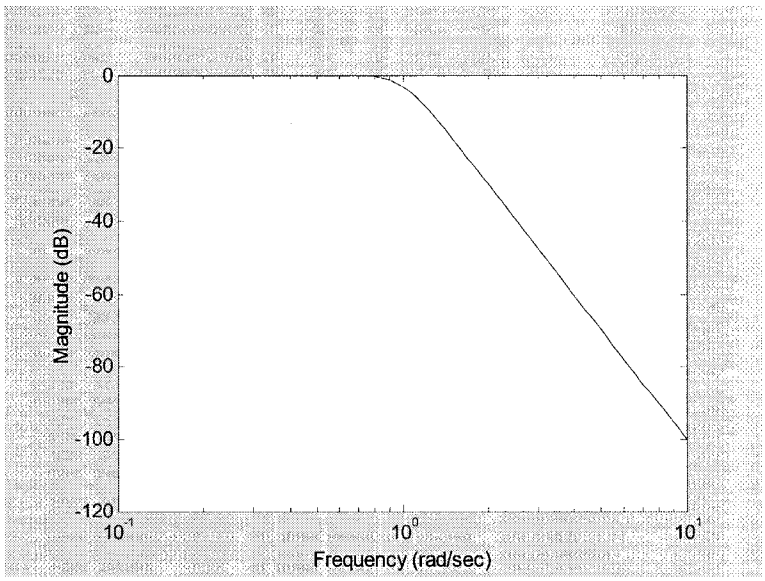

Figure 3.4: Magnitude response of Butterworth low-pass filter with $\omega_p = 1\ rad/s$.

can approximate any given ideal filter. However, more computation will be needed as the order of the filter increases. Therefore, the order of the filter is set to be five to get the satisfactory result while keeping the computation reasonable.

The way to carry out the filter design is to build an analog low-pass Butterworth prototype filter with 1 rad/s as its passband, then transfer 1 rad/s to our desired band-edge frequency 1.176 rad/s by frequency transformation, and finally transform it into a digital filter using bilinear transformation [6].

The Butterworth function has only poles (i.e., no finite zeros) and has the flattest magnitude possible in the passband. This function is also called maximally flat magnitude (MFM). The derivation of this function is illustrated by taking a general all-pole function of fifth-order with $-3dB$ passband edge frequency $\omega_p = 1\ rad/s$ as

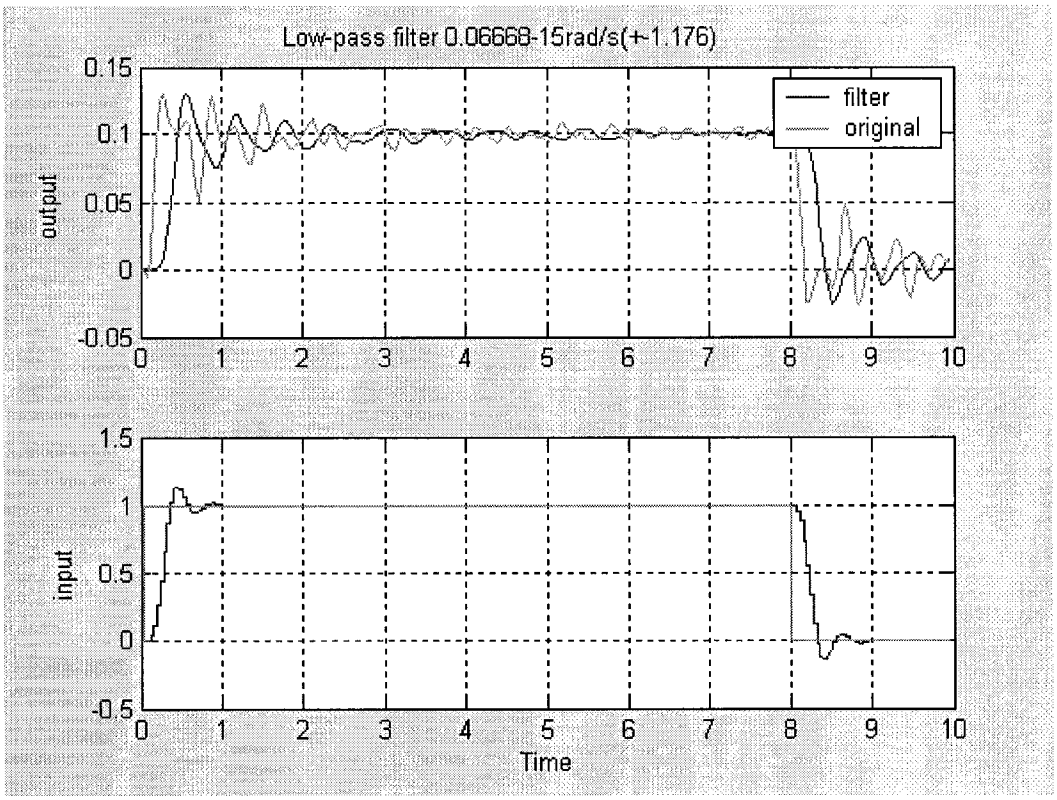

Figure 3.5: Step response signal passed through a $5th$ order butterworth low-pass filter.

follows:

$$G(s) = \frac{1}{s^5 + as^4 + bs^3 + cs^2 + ds + 1} \tag{3.3}$$

The coefficients a, b, c and d can be obtained by using MATLAB function **[z,p,k]=buttap(5)** and **[b,a]=zp2tf(z,p,k)**, where $zp2tf$ stands for zero-pole to transfer function, yielding the transfer function b/a, where b is the numerator and a the denominator. Its magnitude response is shown in Figure 3.4. By substituting

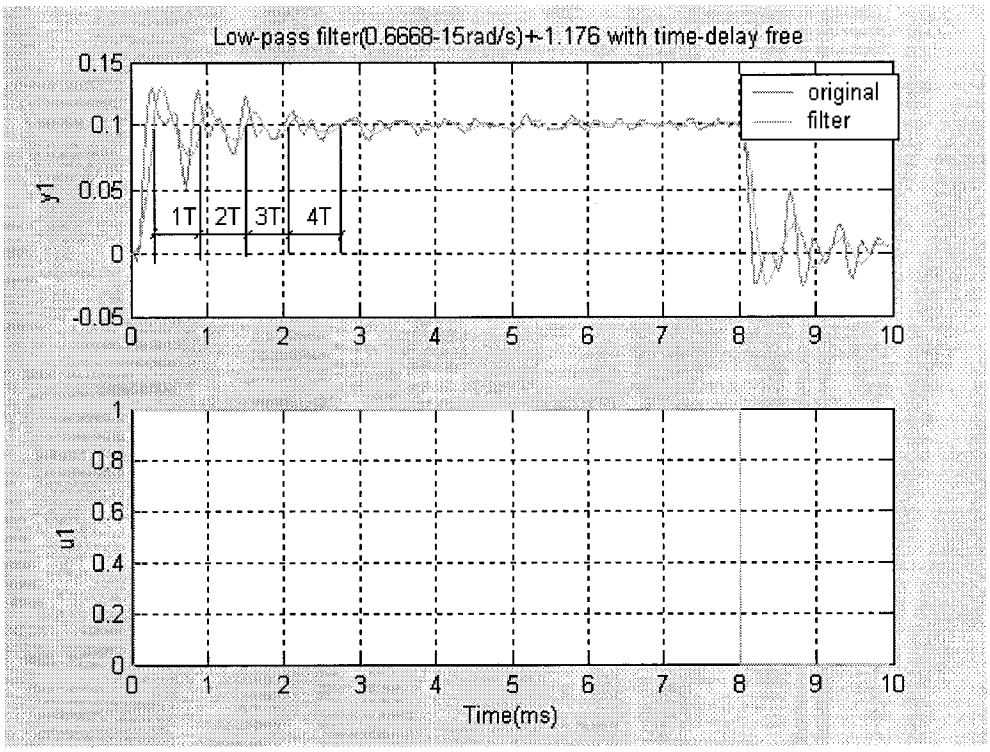

Figure 3.6: Filtered data with delay free.

a, b, c and d into Eq.(3.3), we get the $5th$ order Butterworth prototype filter

$$G(s) = \frac{1}{s^5 + 3.2361s^4 + 5.2361s^3 + 5.2361s^2 + 3.2361s + 1} \tag{3.4}$$

Now transfer the low-pass filter with passband edge frequency of $\omega_p = 1\ rad/s$ to $\overline{\omega_p} = 1.176\ rad/s$. The transformation that achieves this change of bandwidth must transform $\overline{\omega} = 0$ to $\omega = 0$ and $\overline{\omega} = \pm\infty$ to $\omega = \pm\infty$. The transformation defined as $s = \overline{s}/\overline{\omega_p}$ [6] can meet the requirement and maintain the order of the filter. The transfer function of the Butterworth filter with $\overline{\omega_p} = 1.176\ rad/s$ is given as:

$$G(\overline{s}) = \frac{2.2492}{\overline{s}^5 + 3.8057\overline{s}^4 + 7.2414\overline{s}^3 + 8.5159\overline{s}^2 + 6.1894\overline{s} + 2.2492} \tag{3.5}$$

Cycle	$1T$	$2T$	$3T$	$4T$
Time	$0.6ms$	$0.6ms$	$0.56ms$	$0.72ms$

Table 3.1: Cycle time of the step response.

This can also be obtained in MATLAB as **[b,a]=butter(5,1.176,'low','s')**.

The bilinear transformation will compress the analog frequency in $[0, \infty)$ into the digital frequency in $[0, \pi]$. Replacing $\overline{s}$ by $(z-1)/(z+1)$ in Eq.(3.5) yields the follow digital low-pass filter

$$H(z) = \frac{2.2492}{(\frac{z-1}{z+1})^5 + 3.8057(\frac{z-1}{z+1})^4 + 7.2414(\frac{z-1}{z+1})^3 + 8.5159(\frac{z-1}{z+1})^2 + 6.1894(\frac{z-1}{z+1}) + 2.2492} \tag{3.6}$$

We notice that there is a time-delay of five sampling unit, which equals $5 \times 0.04ms = 0.2ms$ (Figure 3.5) , introduced by the low-pass filter. Again, we shift the filtered signal back $0.2ms$ as shown in Figure 3.6.

3.1.3 Identification and Choosing the Best Fit

Now the work of the data pre-processing is done. We will estimate the system model by picking the appropriate values of ω_n and ζ. The relationship between ω_n and ζ is

$$\omega_n = \frac{2\pi}{T\sqrt{1-\zeta^2}} \tag{3.7}$$

where $T = 0.6\ ms$ is the average of four cycles time of the step response as shown in Figure 3.6. The four cycles time are presented in Table 3.1.

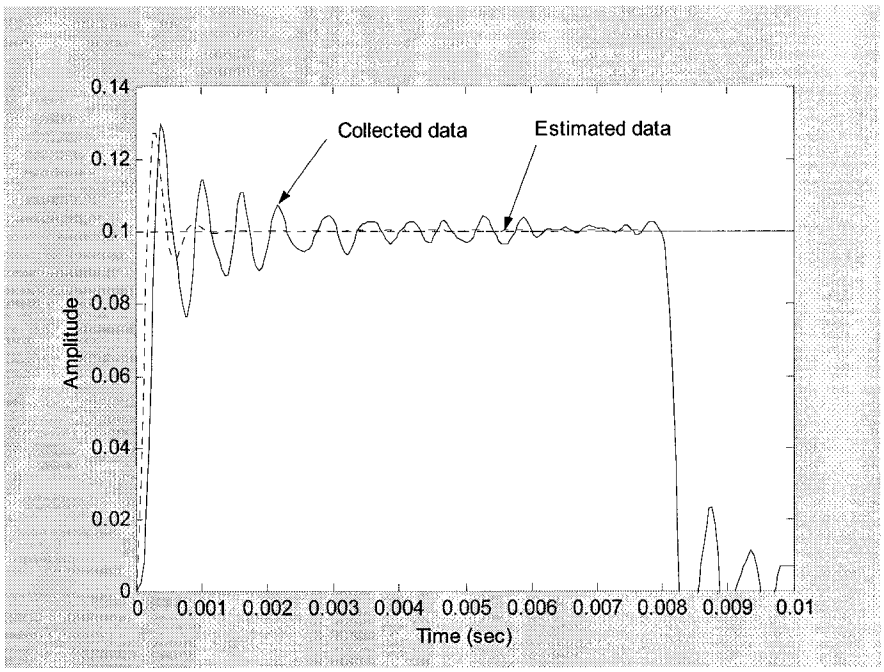

Figure 3.7: The step response of the estimated model with $\omega_n = 11.31 \times 10^3\ rad/s$ and $\zeta = 0.378$ (dotted line) compared with the collected data (solid line).

It is known that the value of damping ratio ζ is between 0 and 1. As ζ decreases, the system roots approach the imaginary axis and the step response becomes increasingly oscillatory. The estimate of the system parameters is based on this fact and the relation of ω_n and ζ described in Eq.(3.7). In other words, we will first choose a set of ω_n and ζ and get the transfer function of the piezo stage $G(s)$. Then the step response of $G(s)$ and the data collected from NPS 3330 as shown in Figure 3.2 is compared and modification of the value of ω_n and ζ will be made to get better fit of the collected data. We will repeat this procedure until the best fit is found.

As a first try, we take the resonant frequency $\omega_n = 1.8\ k\text{Hz} = 11.3 \times 10^3\ rad/s$, which is the first resonant frequency (no load) in Queensgate's datasheet of $NPS -$

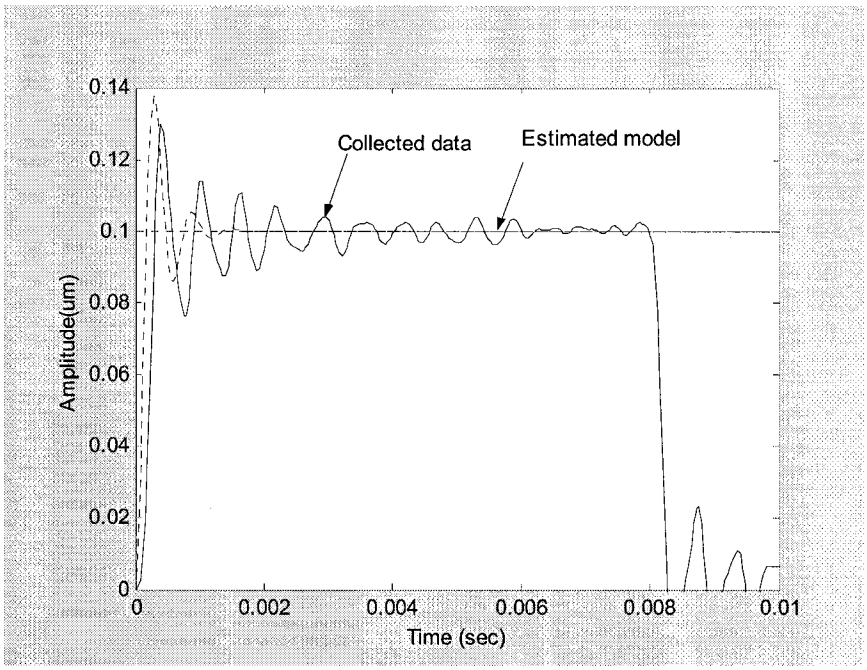

Figure 3.8: The step response of the estimated model with $\omega_n = 10.98 \times 10^3\ rad/s$ and $\zeta = 0.3$ (dotted line) compared with the collected data (solid line).

$Z - 15B$. The value of ζ calculated by Eq.(3.7) is 0.378. Therefore, the transfer function of the estimated system is

$$G(s) = \frac{1.279 \times 10^7}{s^2 + 8.55 \times 10^3 s + 1.279 \times 10^8} \tag{3.8}$$

and the step response of $G(s)$ is shown in Figure 3.7. In Figure 3.7, only the first cycle of the estimated model has 90 percent match to the actual model if the difference in the rising time is ignored. The comparison of the rest parts of the response shows that the estimated model is not acceptable. Because the output of the estimated model converges to the desired value (0.1 μm in this case) too quickly, ζ should be decreased to have the estimated model's output oscillate more, that is, more cycles

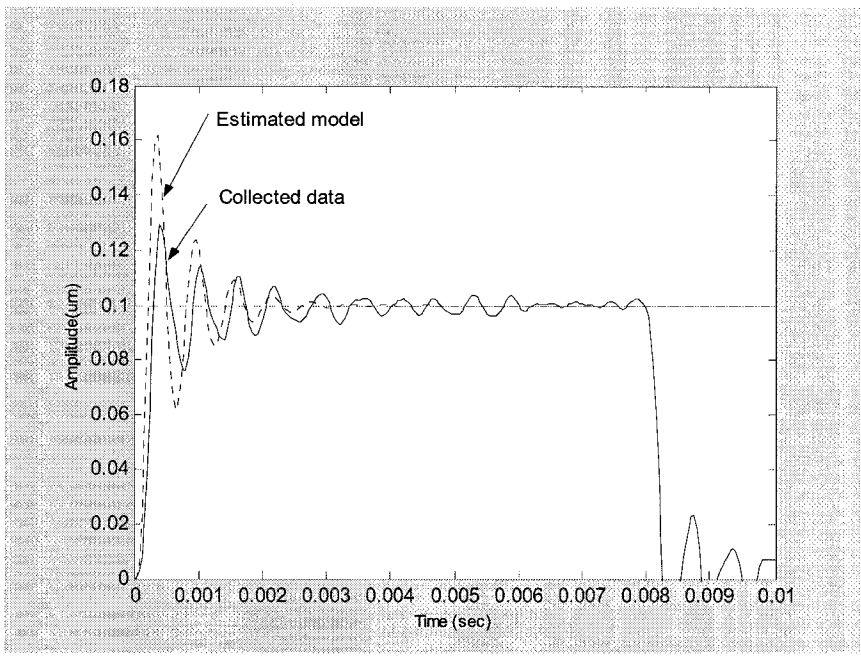

Figure 3.9: The step response of the estimated model with $\omega_n = 10.59 \times 10^3\ rad/s$ and $\zeta = 0.15$ (dotted line) compared with the collected data (solid line).

of the response have better match to the actual model.

By taking $\zeta = 0.3$ and $\omega_n = 10.98 \times 10^3\ rad/s$ into Eq.(3.7), the transfer function of the estimated system can be written as

$$G(s) = \frac{1.25 \times 10^7}{s^2 + 6.588 \times 10^3 s + 1.25 \times 10^8} \tag{3.9}$$

and the step response of $G(s)$ is shown in Figure 3.8. As we can see, although the first peak value of the estimated model is higher than that of Figure 3.7, the second peak value fits better. Now we will reduce the value of ζ further, i.e. from 0.3 to 0.15, to see if we can get better fits. The transfer function of the estimated system is

$$G(s) = \frac{1.12 \times 10^7}{s^2 + 3177 s + 1.12 \times 10^8} \tag{3.10}$$

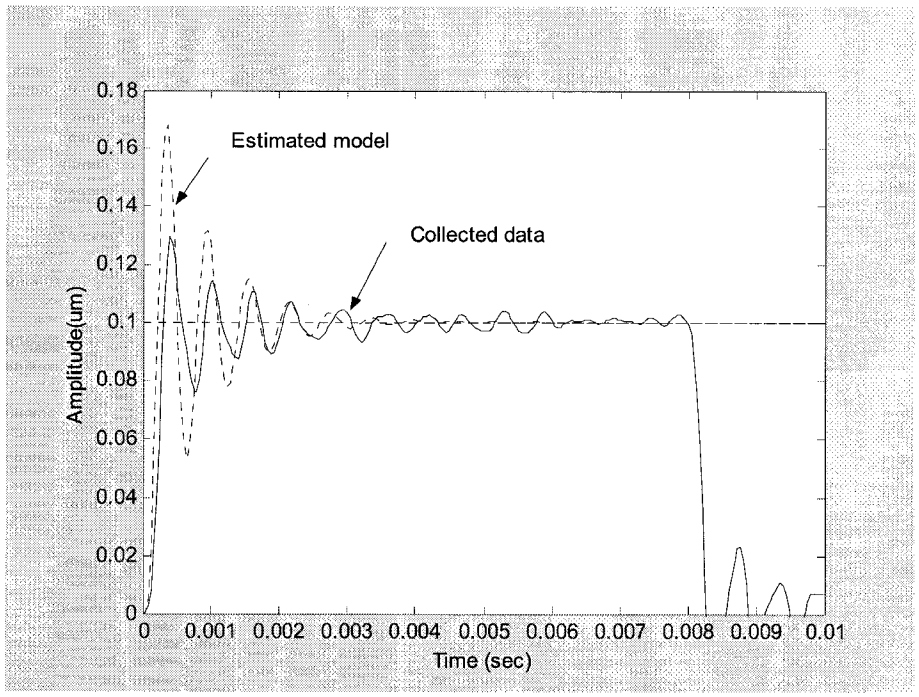

Figure 3.10: The step response of the estimated model with $\omega_n = 10.55 \times 10^3\ rad/s$ and $\zeta = 0.12$ (dotted line) compared with the collected data (solid line).

with $\omega_n = 10.59 \times 10^3\ rad/s$. The step response of $G(s)$ is shown in Figure 3.9. Here, the first three cycles of the estimated model match well to the original one although this is achieved at the price of increasing magnitude distortion of the first cycle. We can decrease the ζ value again to have more cycles fit as shown in Figure 3.10 with $\omega_n = 10.55 \times 10^3\ rad/s$ and $\zeta = 0.12$. The corresponding $G(s)$ is given as

$$G(s) = \frac{1.113 \times 10^7}{s^2 + 2532s + 1.113 \times 10^8} \tag{3.11}$$

It is clear that as the ζ reduces the system is even more oscillatory than the original system (Figure 3.11) which is not acceptable. Therefore, the estimated system with transfer function in Eq.(3.10) is chosen as the best fit and will be considered as the

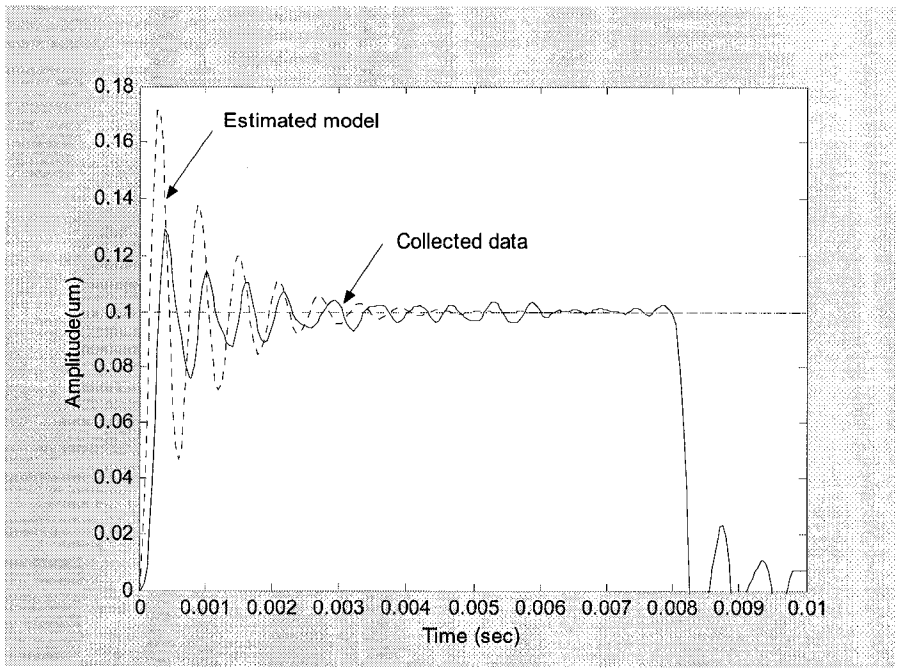

Figure 3.11: The step response of the estimated model with $\omega_n = 10.525 \times 10^3\ rad/s$ and $\zeta = 0.1$ (dotted line) compared with the collected data (solid line).

model of the piezo stage.

There is another alternative, say, curve fitting, could be used to determine ω_n and ζ from the measured system step response. Curve fitting is the process of computing the coefficients of a function to approximate the values of a given data set within that function. The accuracy of the approximation or the fit is judged by using a mathematical function such as a least squares regression. For a second-order system, the system output to a unit step input can be mathematically calculated as

$$y(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n} \sin(\omega_n \sqrt{1-\zeta^2}t + \theta) \tag{3.12}$$

where $\theta = \sin^{-1}\sqrt{1-\zeta^2}$. The value of ω_n and ζ could be obtained by minimizing the

least square function

$$\min_{\omega_n,\zeta} \sum_{t=0} |y(t) - y_m(t)|^2, \quad t = 0, 1, 2, 3... \tag{3.13}$$

where $y_m(t)$ presents the measured data points in Figure 3.6. Since the method used in this section achieved acceptable system model, curve fitting is not implemented to obtain the piezo stage model in this thesis.

3.2 Cantilever Dynamics

As discussed in Chapter 2, the model of the cantilever dynamics is based on the modelling of the tip-sample interaction. The whole picture of the interaction, i.e., the JKR force model which incorporates the two types of forces mentioned in Chapter 2 together with the repulsive modified Hertz model will be presented in this section.

3.2.1 Tip-sample Interaction

As illustrated in Figure 3.12 [8], the cantilever can be modeled equivalently as a spring with a spring constant k. Here, z_1 is the tip-sample distance determined by the piezo stage. To describe the dynamics of the tip-sample interaction we developed a force model based on Johnson-Kendall-Roberts (JKR) adhesion theory. The JKR model is used to relate the surface energy of the interaction to the elasticity of a sphere in contact with a flat surface. The fundamental assumptions of the JKR Theory are [18]: a) the deformation of the contacting spheres is elastic. b) the contact radius

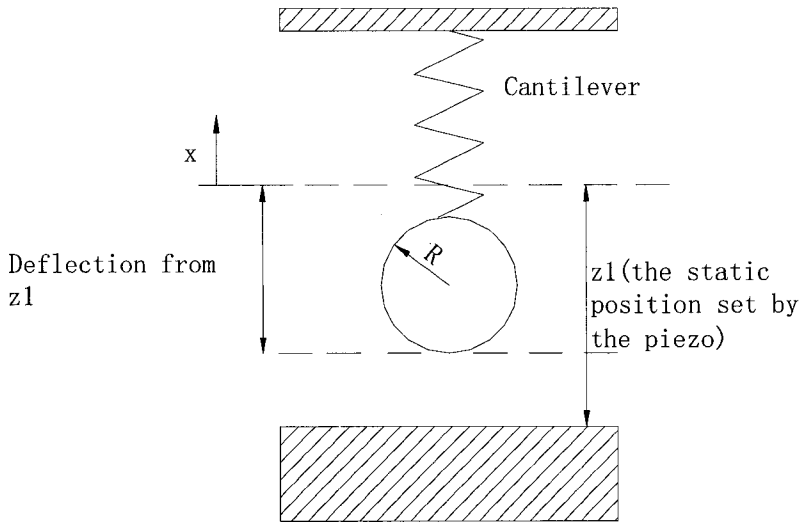

Figure 3.12: Equivalent diagram of the cantilever-tip-sample system.

is small compared to the particle radius. c) all interactions are localized to within the contact region. The JKR model consists of the Lennard-Jones potential and a modified Hertz model.

Consider that the equilibrium distance z_0, at which the force drops to zero, is given by $z_0 = 30^{-\frac{1}{6}}\sigma$, then for the tip-sample separation $z > z_0$, the Lennard-Jones potential dominates. Otherwise, the modified Hertz model will be used to model the repulsive force.

Lennard-Jones Potential

The Lennard-Jones potential applies strictly to the interaction between two molecules. However, the actual macroscopic interactions (i.e. the interaction between a point molecule and a planar surface separated by a distance z) can be approximated

R^*	Radius of the tip	$30nm$
ν_1[8]	Poisson ratio of the tip	0.5
ν_2[8]	Poisson ratio of the sample	0.5
E_1[21]	Young's modulus of the tip	$179nN(nm)^2$
E_2[21]	Young's modulus of the sample	$179nN(nm)^2$
z_0^{**}	Equilibrium point where $F(z) = 0$	$0.193nm$
σ[17]	The tip-sample separation where the energy is zero	$0.34nm$
ϵ[17]	Minimum energy of the Lennard-Jones potential	$3.79 \times 10^{-22}J$
ρ_1[23]	Number density of the tip	$5 \times 10^{28}m^{-3}$
ρ_2[23]	Number density of the sample	$5 \times 10^{28}m^{-3}$

Table 3.2: Parameters of the JKR model.

* Parameter of the cantilever used in the experiment.

**Calculated using data from Ref. [17].

by integrating the interaction across microscopic bodies. Here, the tip is treated as a sphere that has radius R and density ρ_1. The samplea is represented by a flat surface with density ρ_2. The Lennard-Jones potential consists of the attractive van der Waals part and the repulsive part (Pauli principle) in Section 2.3. It can be written as

$$F(z) = \frac{2}{3}\pi^2\epsilon\rho_1\rho_2^4\sigma^4 R\left[\frac{1}{30}\left(\frac{\sigma}{z}\right)^8 - \left(\frac{\sigma}{z}\right)^2\right] \tag{3.14}$$

Modified Hertz Model

The classical Hertz model provides for the elastic deformation of bodies in contact. This model has to be modified because the zero point of the resulting force has to be identical with the zero point of the Lennard-Jones potential z_0 [23]. According to this theory, the contact radius α is given by [8]

$$\alpha^3 = \frac{3}{4}\pi(k_1 + k_2)\frac{R_1R_2}{R_1 + R_2}F(z) \tag{3.15}$$

with

$$k_i = \frac{1-\nu_i^2}{\pi E_i} \tag{3.16}$$

where E_i is the Young's modulus. The parameters in Eq.(3.15) are described in Table 3.2. Assume that $R_1 << R_2$, i.e., the tip with E_1 and $R = R_1$ deforming a sample with E_2. A small indentation of the sample by $|z_0 - z| << R$ will result in an indentation radius $\alpha = \sqrt{2R(z_0 - z)}$. Inserting this value of α into Eq.(3.15) yields

$$F(z) = \frac{8\sqrt{2R}}{3\pi(k_1+k_2)}(z_0 - z)^{\frac{3}{2}} \tag{3.17}$$

Combining Eq.(3.14) and Eq.(3.17) we get the force model that describes the tip-sample interaction as

$$\begin{aligned} F(z) &= \frac{2}{3}\pi^2\epsilon\rho_1\rho_2^4\sigma^4 R\left[\frac{1}{30}\left(\frac{\sigma}{z}\right)^8 - \left(\frac{\sigma}{z}\right)^2\right] \quad z > z_0 \\ &= \frac{8\sqrt{2R}}{3\pi(k_1+k_2)}(z_0 - z)^{\frac{3}{2}} \qquad\qquad z \leq z_0 \end{aligned} \tag{3.18}$$

The force curve is shown in Figure 2.11. Since the AFM operates in the contact mode, only the modified Hertz model part of the force curve should be considered. By substituting the values of parameters in Table 3.2, the force model is simplified to

$$F(z) = \frac{8\sqrt{2R}}{3\pi(k_1+k_2)}(z_0 - z)^{\frac{3}{2}} = 2464 \times (0.193 - z)^{\frac{3}{2}} \tag{3.19}$$

3.2.2 State-space Model of the Cantilever

The system diagram is shown in Figure 3.13. The cantilever can be modeled as a soft damped single spring mass resonator, which is detuned by the tip-sample force

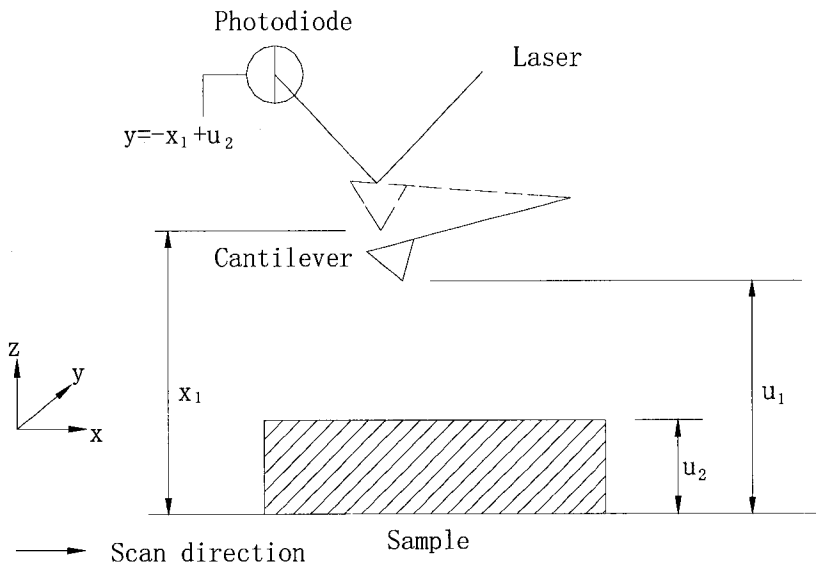

Figure 3.13: Schematic diagram of the state-space model of the cantilever-tip-sample system [23].

$F(z)$ in Eq.(3.19). The state space equations of the spring-mass system are:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -\tfrac{\omega_n}{Q} x_2(t) - \tfrac{k}{m}[x_1(t) - z_1] + \tfrac{1}{m} F(x_1(t)) \end{aligned} \tag{3.20}$$

where z_1 is the static height of the tip set by the piezo stage. The variables used in Eq.(3.20) are described in Table 3.3. To consider the movement of the piezo as the command input to the cantilever, the model has to be extended by an input u_1, which equals the sum of the displacement of the piezo and z_1. The sample's topography will be treated as a disturbance to the control system as the input u_2. The output of the system is the displacement of the tip which goes into the photodiode. All the states, inputs and the output are functions of time because of the scanning process. The new state space model that incorporates the effect of the movement of the piezo is

x_1	Position of the tip	
x_2	Velocity of the tip	
ω_n^*	Resonance frequency of the cantilever	$13k$Hz
k^*	Spring constant of the cantilever	$0.2N/m$
m^*	Reduced mass of the cantilever	$1.1834 \times 10^{-6}g$
z_1	Static height of the tip set by the piezo stage	
Q	Quality factor of the cantilever	

Table 3.3: Variables of the state space model of cantilever.

*Parameters chosen by us, reflecting a cantilever used in the experiment.

given by:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -\tfrac{\omega_n}{Q}x_2(t) - \tfrac{k}{m}[x_1(t) - u_1(t) + u_2(t)] + \tfrac{1}{m}F(x_1(t)) \\ y(t) &= -x_1(t) + u_2(t) \end{aligned} \tag{3.21}$$

Linear Approximation of Nonlinear Force $F(z)$

Since the tip-sample interaction force $F(z)$ is a non-linear function of the tip-sample separation, we should obtain a linear equivalent by assuming small-signal conditions. At the operating point of $z = 0.12nm$, $F(z)$ can be well approximated as a straight line with slope of $k_F = -1000$ nN/nm as illustrated in Figure 3.14. Thus, the linearized function of $F(z)$ is

$$\Delta F(z) = -1000\Delta z \tag{3.22}$$

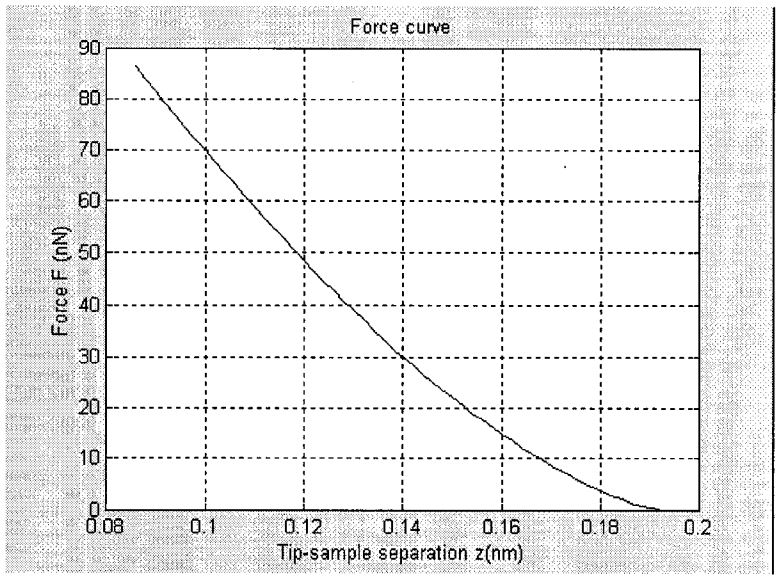

Figure 3.14: Tip-sample interaction force as a function of tip-sample separation.

Thus, the state-space model of the cantilever, namely, Eq.(3.21), becomes

$$\begin{bmatrix} \Delta\dot{x}_1 \\ \Delta\dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -8.45 \times 10^{11} & -\frac{1.3\times10^4}{Q} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1.69 \times 10^8 & -1.69 \times 10^8 \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix} \tag{3.23}$$

$$\Delta y = \begin{bmatrix} -1 & 0 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \begin{bmatrix} 0 & -1 \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix} \tag{3.24}$$

where Q is the parameter which can be adjusted while designing the controller. The dynamic of the cantilever can be approximated using Eq.(3.23) and (3.24) within the linear range of the tip-sample interaction force $F(z)$.

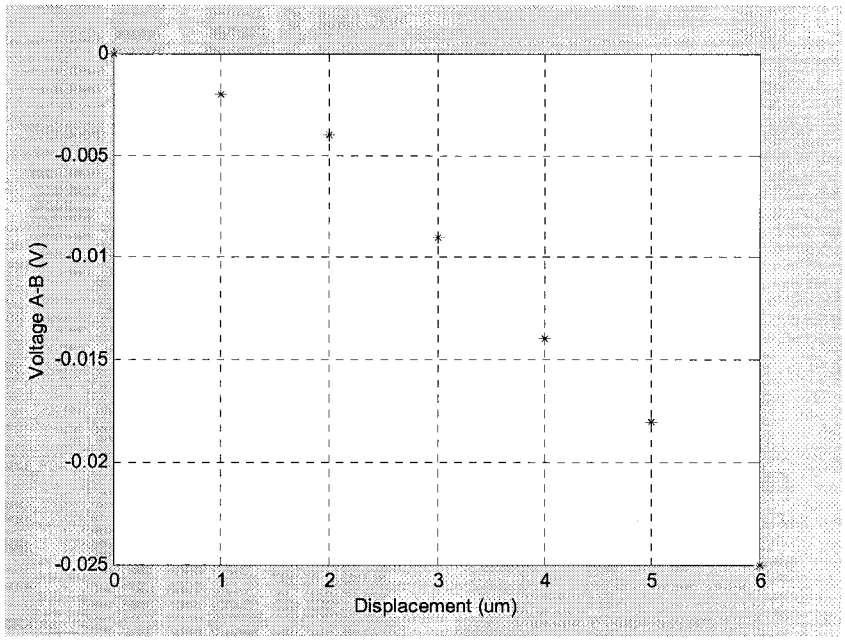

Figure 3.15: Measurement of photodiode gain.

3.3 Photodiode Modeling

We consider the photodiode as a constant gain in the sense that the output voltage of the photodiode is linear to the input displacement signal. From the experimental data we calculate the photodiode gain as 0.006 $V/\mu m$, as shown in Figure 3.15.

Chapter 4

Controller Design

In this chapter, three different controllers are designed for the purpose of precisely tracking arbitrary input signals (the sample surface) with high bandwidth. Two basic design methods, a phase-lag controller and a traditional PID controller, are presented in detail. This is followed by a more advanced H_∞ controller design, which demonstrates substantial improvements in the tracking speed and precision. Finally, comparisons of these three controllers are addressed in terms of the time-domain and frequency-domain control specifications.

4.1 Basic Controller Design

The model inferred for the integrated plant which consists of the piezoelectric stage, cantilever and photodiode is employed to design the feedback laws. The block diagram of the closed-loop system is shown in Figure 4.1. Here, r represents the

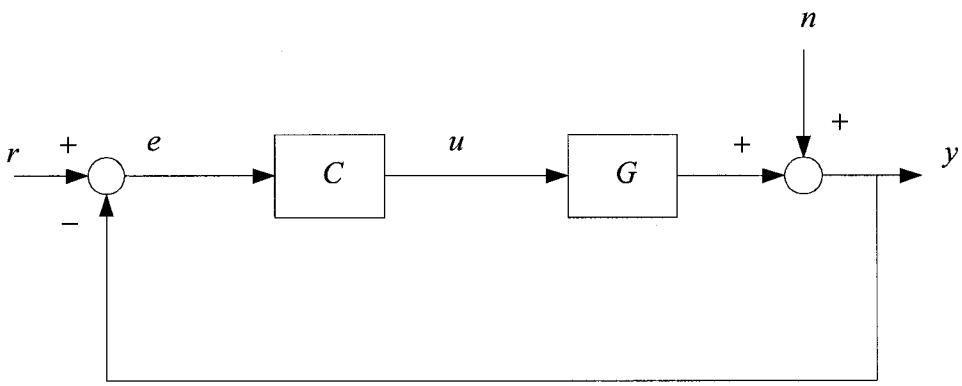

Figure 4.1: Block diagram of the closed-loop system.

reference or the command signal; e is the error signal; C denotes the controller which is to be designed; and G stands for the integrated plant model. The transfer function of the piezo stage and the photodiode is given as

$$G_p(s) = \frac{6.72 \times 10^4}{s^2 + 3177s + 1.12 \times 10^8} \tag{4.1}$$

The transfer function of the cantilever from input u_1 to output y (Fig. 3.13) is

$$G_c(s) = \frac{1.69 \times 10^8}{s^2 + 1300s + 8.45 \times 10^{11}}, \text{ with the quality factor } Q = 10 \tag{4.2}$$

Therefore, the integrated plant model $G(s)$ is

$$G(s) = \frac{1.136 \times 10^{13}}{s^4 + 4477s^3 + 8.45 \times 10^{11}s^2 + 2.685 \times 10^{15}s + 9.48 \times 10^{19}} \tag{4.3}$$

The phase-lag controller and the PID controller designs are based on this plant model.

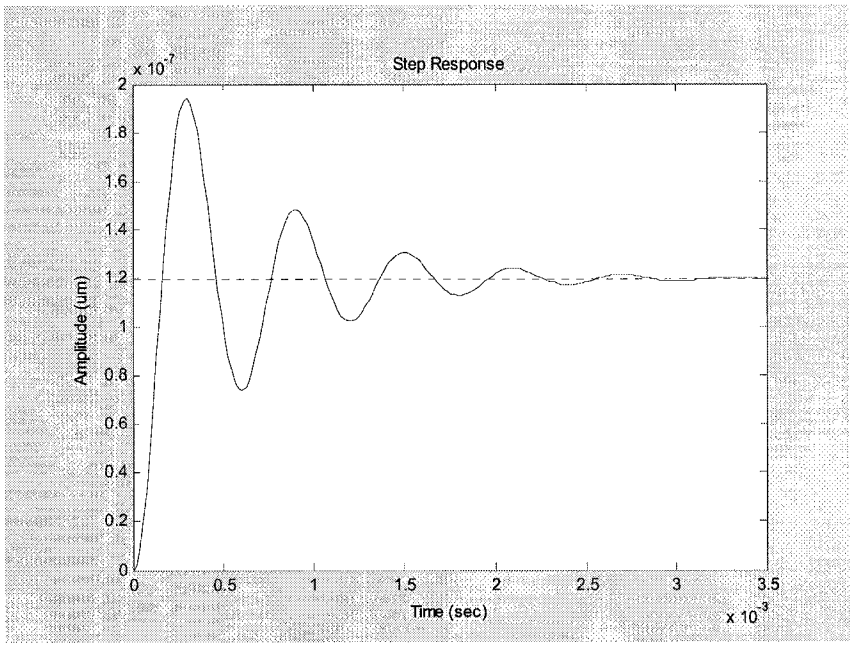

Figure 4.2: Step response of the open-loop transfer function G.

4.1.1 Phase-lag Controller Design and Simulation

The controller design problem should be addressed to meet a set of specifications. From the step response of the plant $G(s)$ as shown in Figure 4.2, we observe that there is a considerable steady-state error in the output magnitude comparing to the input signal with magnitude of $1\mu m$. In addition, the transient performance is not satisfactory, namely, the overshoot is quite high. Since the primary objective of the control design is to achieve precise tracking of the input signals with high bandwidth, the controller should meet the following specifications in the frequency domain:

1. Position error e_{ss} (the difference between the final output value and the input

signal) $\leq 1\%$.

2. Phase margin of the compensated system $PM \geq 60°$.

3. Gain-crossover frequency ω_{gc} which associates with the bandwidth of the compensated system as large as possible (within reasonable limit).

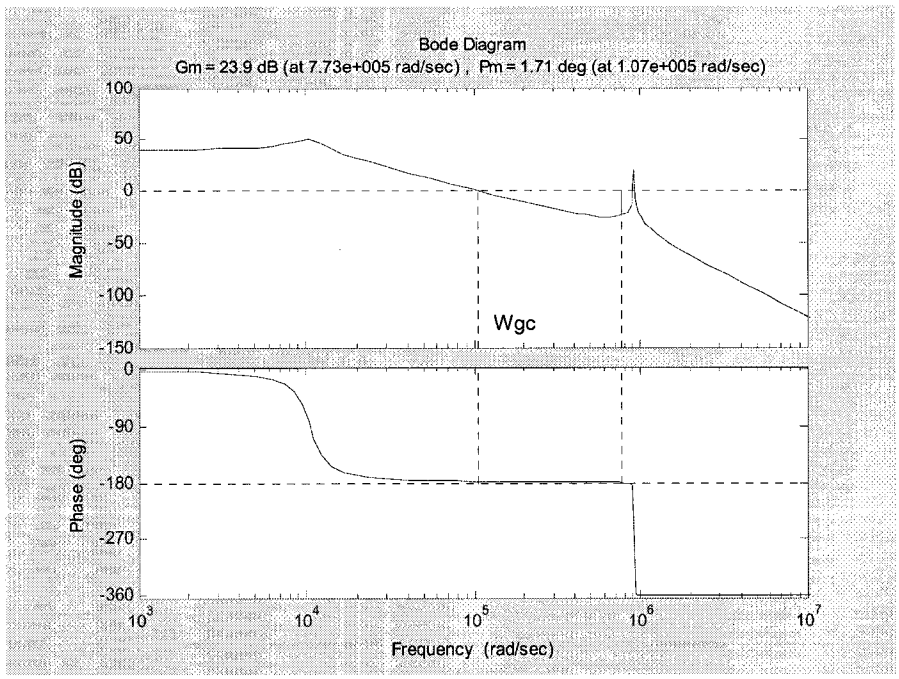

Figure 4.3: Bode plot of $k_c G(s)$.

A phase-lag controller is designed to meet the specifications described above. The transfer function of the phase-lag controller is given by

$$C(s) = k_c \frac{\alpha\tau s + 1}{\tau s + 1}, \quad \tau > 0, \quad 0 < \alpha < 1 \tag{4.4}$$

The procedure of designing phase-lag compensators can be summarized as [7]:

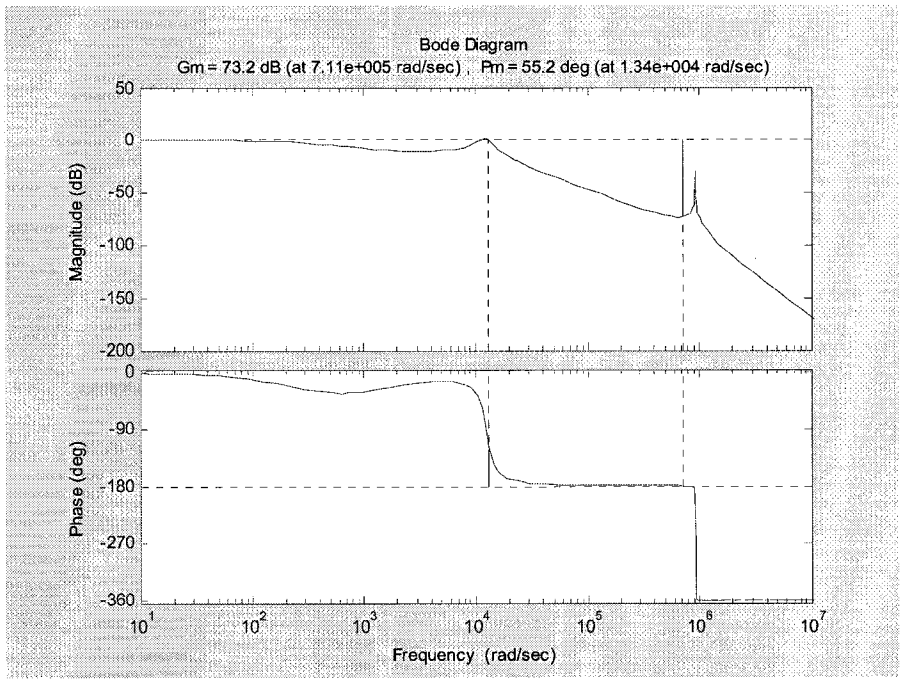

Figure 4.4: Bode plot of the closed-loop system controlled by the phase-lag controller.

1. Compute constant k_c from e_{ss}. $e_{ss} = \frac{1}{1+K_P} = 0.01 \implies K_p = 99$, where K_p is the position error constant. $K_p = \lim_{s\to 0} G(s)C(s) = \frac{k1.136\times 10^{13}}{9.48\times 10^{19}} \implies k_c = 8.26 \times 10^8$.

2. Plot the Bode plot of the plant with the required error constant, namely, $k_cG(s)$ in Figure 4.3. It can be read that the gain margin is 23.9 dB, the phase margin is 1.71° and $\omega_{gc} = 1.07 \times 10^5\ rad/s$.

3. Determine the required phase margin and the new ω_{gc}. If the required phase margin is 60°, then the frequency in the Bode plot of Figure 4.3 is 1.16×10^4 rad/s, where $\angle k_cG = -180° + 60° = -120°$. This frequency is designated as

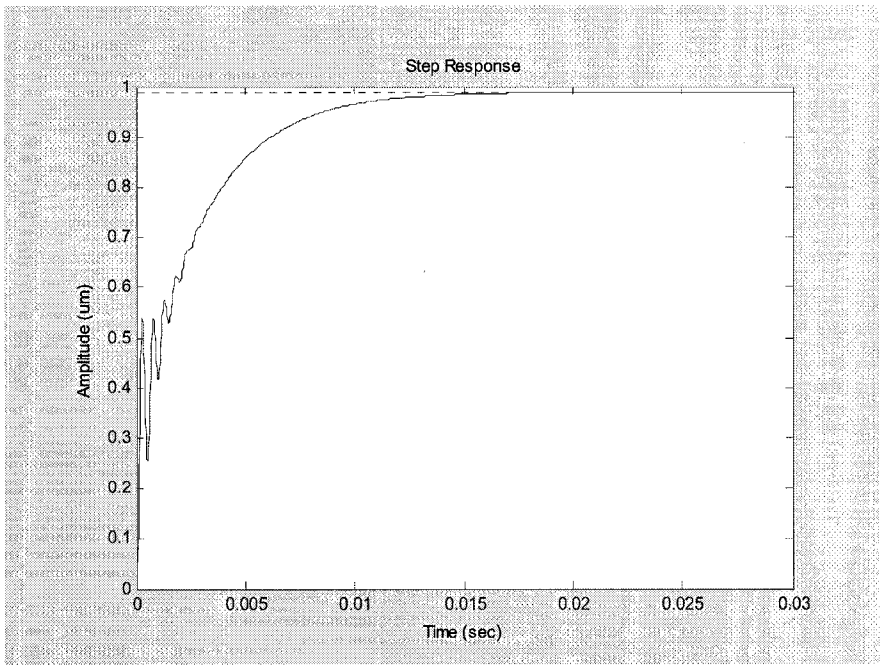

Figure 4.5: Step response of the closed-loop system controlled by the phase-lag controller.

the new gain-crossover frequency ω_{gc}.

4. Calculate the value of α needed to bring the gain plot down to new ω_{gc}. The gain at the new ω_{gc} is 48.2 dB. Then, $20\log\alpha = -48.2\ dB \implies \alpha = 0.004$.

5. Determine τ in Eq.(4.4). In practice, the corner frequency $1/\alpha\tau$ is placed one decade below the new gain-crossover frequency ω_{gc}, that is,

$$\frac{1}{\alpha\tau} = \frac{\omega_{gc}}{10} \tag{4.5}$$

The value of τ then is 0.216. The transfer function of the phase-lag controller is given as

$$C(s) = 8.26 \times 10^8 \frac{8.64 \times 10^{-4}s + 1}{0.216s + 1} \tag{4.6}$$

The phase margin of the closed-loop system is 55.2° as shown in Figure 4.4, which is very close to the specification (2). The step response of the closed-loop system is shown in Figure 4.5. The overall system is stable although some oscillations exist which are due to the high-frequency dynamics of the system. Figures 4.6-4.7. show the closed-loop system tracking of 1Hz and 5Hz sawtooth signal respectively. We can see that the tracking error grows while the frequency of the reference signal increases.

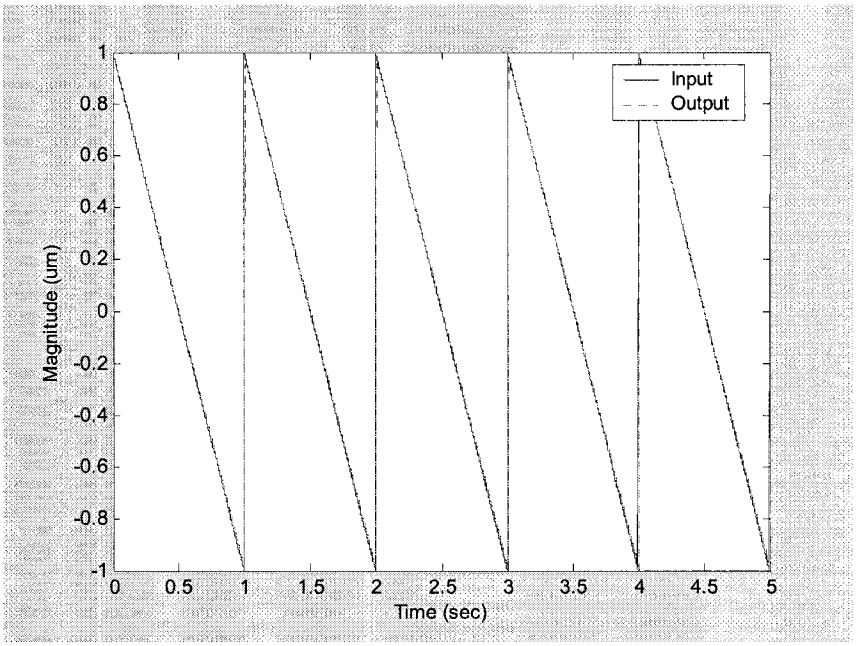

Figure 4.6: Tracking of 1Hz sawtooth signal with the phase-lag controller.

4.1.2 PID Controller Design and Simulation

The traditional proportional-integral-derivative (PID) controllers have been employed extensively on commercial AFMs. A well tuned PID controller can provide

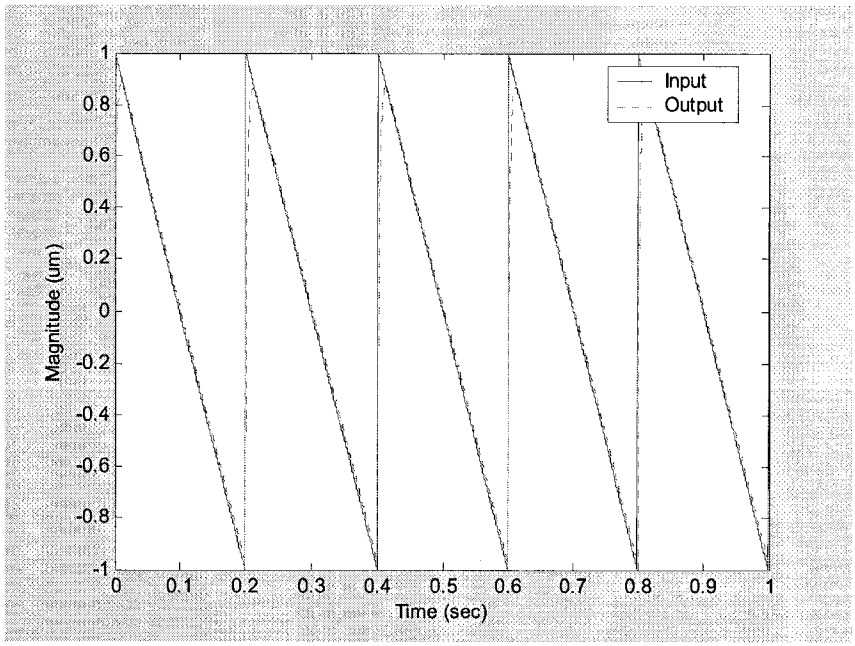

Figure 4.7: Tracking of 5Hz sawtooth signal with the phase-lag controller.

satisfactory overall performance for most industrial applications. However, finding the optimal PID parameters is not easy and it is a primarily trial and error task. Figure 4.8 shows the block diagram of the closed-loop system controlled by a PID controller [14, 36]. Here, the scalar parameters K_p, K_d and time constants τ_i, τ_d are to be determined for the plant in Eq. (4.3).

The first path of the controller is the integrator. The transfer function of the integral controller can be written as

$$C_i(s) = \frac{1}{\tau_i s} \tag{4.7}$$

where τ_i is the integral time constant. A saturation limit is placed in front of $1/\tau_i s$

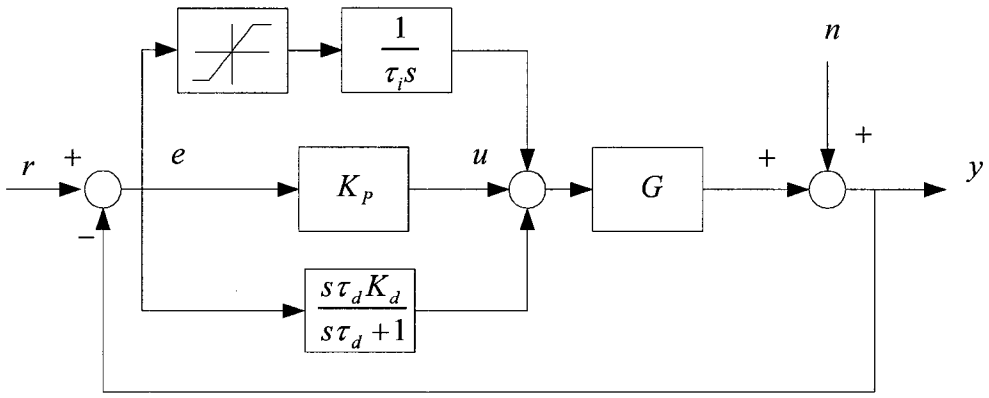

Figure 4.8: The block diagram of the closed-loop system controlled by the PID controller.

to prevent windup. For the type zero system such as the plant model in Eq.(4.3), the introduction of an integral controller will keep the steady-state error stay zero for any step reference input. However, it will make the stabilization of the feedback system more difficult. Even if it is stable, the system response speed may slow down and the response may be more oscillatory. To get a satisfactory system performance, the integral time constant τ_i has to be adjusted. The lower the value of τ_i, the faster the overall system response to a command input.

The second path in Fig.4.8 is the proportional controller. The transfer function of the proportional controller is simply a gain, say K_p. It is proportional to the measured signal and the input signal. As K_p increases, the step response may become faster and the feedback system may eventually become unstable.

The derivative controller is in the third path. The transfer function of the derivative term should be $K_d s$, where K_d stands for the derivative gain. However, $K_d s$ is

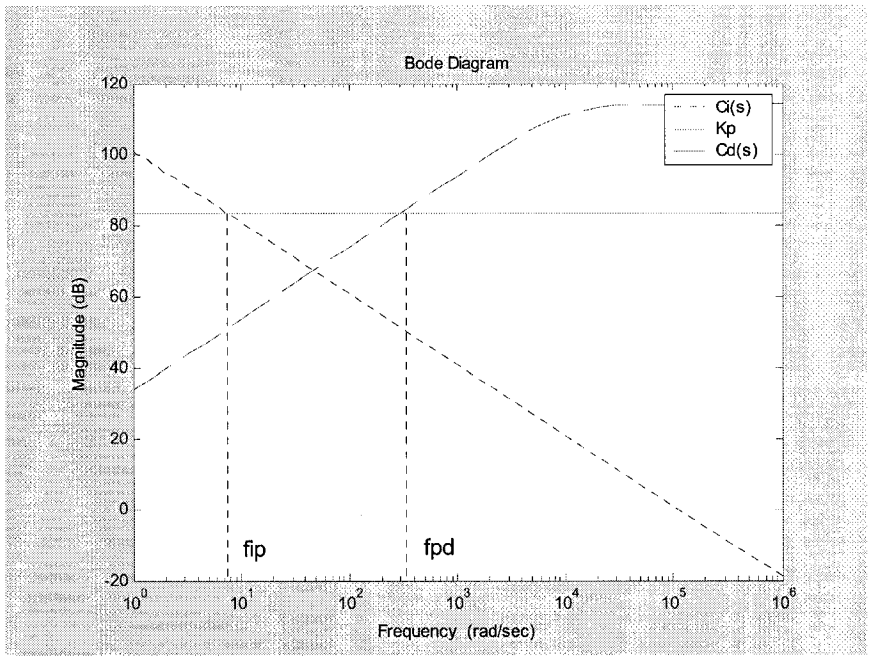

Figure 4.9: Bode plots for each component of the PID controller.

an improper transfer function and is difficult to implement. In practice, it is usually approximated by

$$C_d(s) = \frac{K_d \tau_d s}{\tau_d s + 1} \tag{4.8}$$

Here, τ_d represents the derivative time constant. The derivative terms, K_d and τ_d, are damping terms. Oscillations excited by step inputs can be reduced by increasing the derivative feedback term. The more oscillation you are trying to damp, the more derivative gain you will need. The damping term also limits the high-frequency gain; therefore, high-frequency noises will not be unduly amplified.

The PID parameters could be initially determined from the Bode diagram in

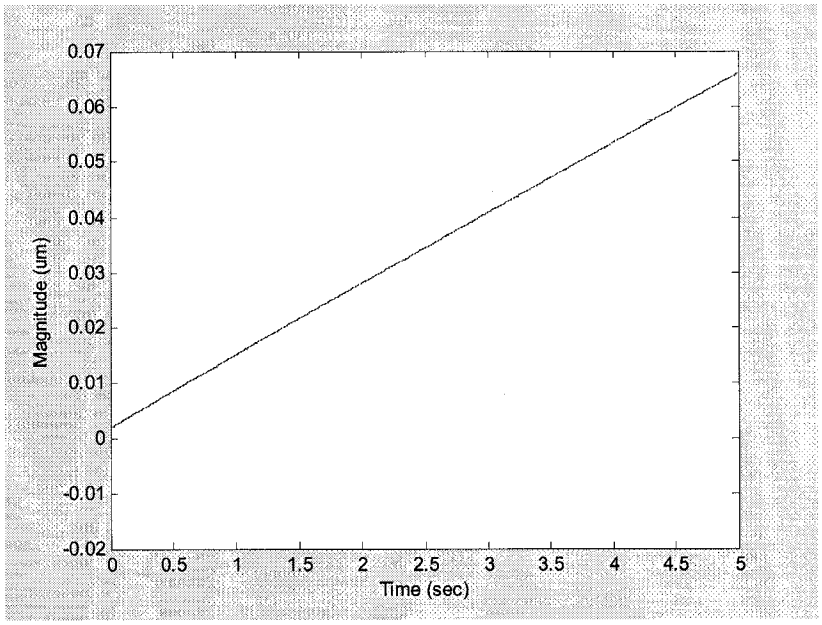

Figure 4.10: Step response of the closed-loop system controlled by the PID controller in Figure 4.9.

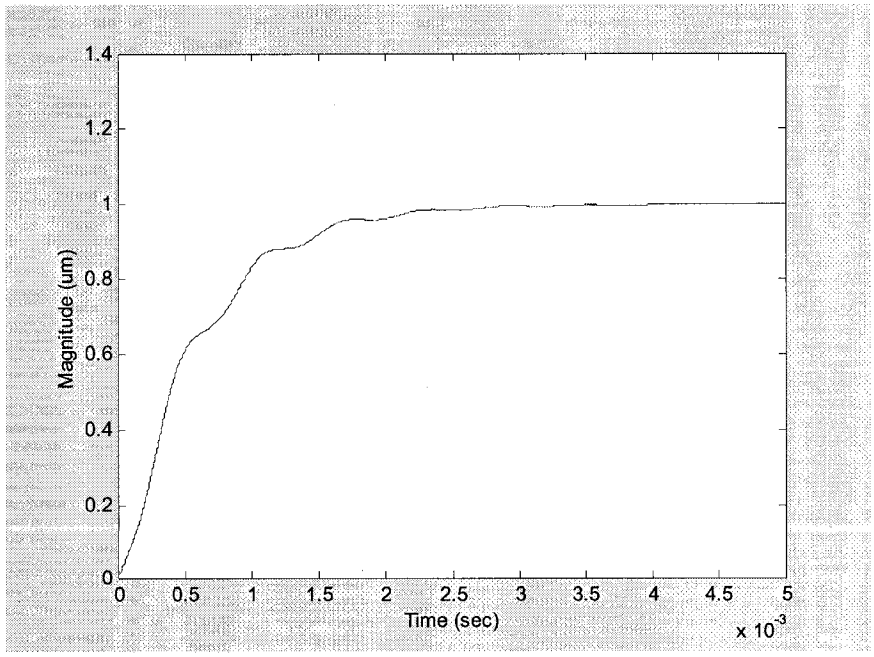

Figure 4.11: Step response of the closed-loop system controlled by the PID controller.

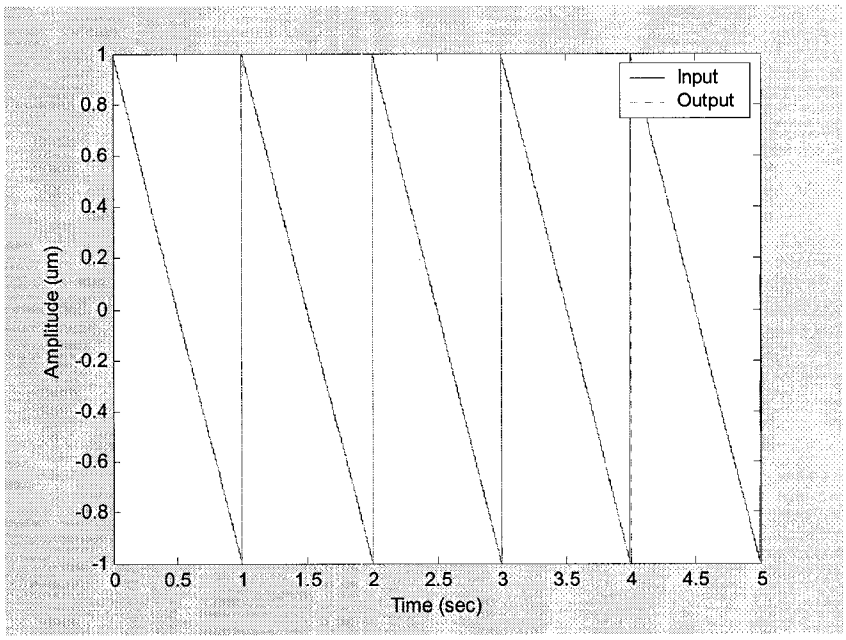

Figure 4.12: Tracking of 1Hz sawtooth signal with the PID controller.

Figure 4.9 [14]. The proportional gain K_p dominates at midrange frequencies, the integrator $C_i(s)$, at lower frequencies, and the derivative controller $C_d(s)$, at higher frequencies. Usually, the distance between the corner frequencies f_{IP} and f_{PD} is greater than 2 octaves. The pole of $C_d(s)$ is typically chosen to be from $3 \times 2\pi f_{PD}$ to $10 \times 2\pi f_{PD}$. According to these rules, f_{IP} in Figure 4.9 is initially chosen to be about 8 rad/sec and f_{PD} is around 230 rad/sec. Figure 4.10 shows the step response of the closed-loop system. It is clear that the system response speed is too slow–the system does not reach the final value of $1\mu m$ within five seconds. To improve the system performance, PID parameters are adjusted as mentioned earlier in this section.

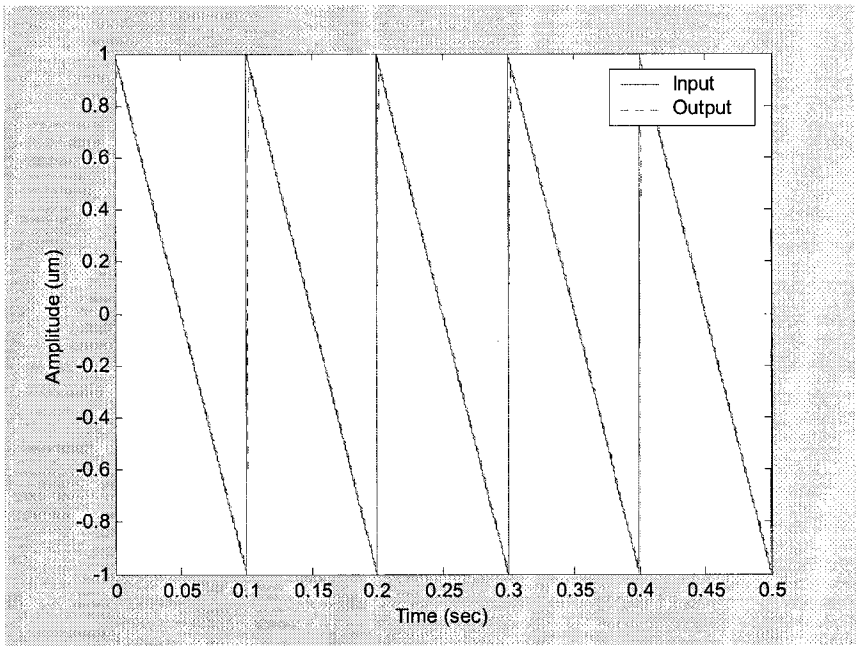

Figure 4.13: Tracking of 5Hz sawtooth signal with the PID controller.

The final tuned PID controller is chosen as

$$C(s) = \frac{1}{9 \times 10^{-11} s} + 1.5 \times 10^4 + \frac{50s}{1 \times 10^{-4} s + 1} \tag{4.9}$$

The step response is illustrated in Figure 4.11. Although the PID controller can provide acceptable system performance, all the parameters have to be tuned again if a change of the system load occurs. Figures 4.12-4.13 also show the closed-loop system tracking of 1Hz and 5Hz sawtooth signal respectively. There is noticeable mismatch in the system output while the frequency of the reference signal increases.

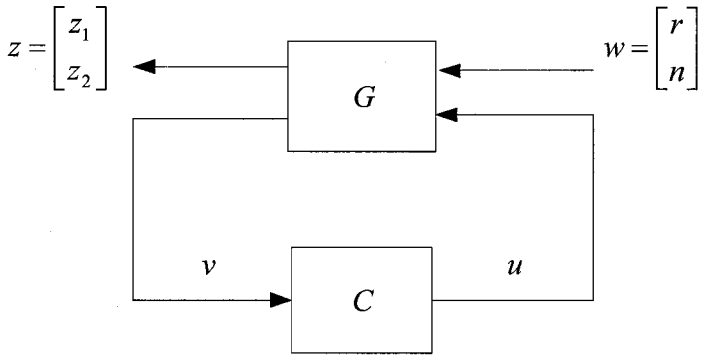

Figure 4.14: Standard system set-up.

4.2 Advanced Robust Controller

From the previous section it is clear that the control performance of tracking input signals of high frequencies needs to be improved. To achieve precise tracking of the arbitrary input signals with faster imaging speed, a more complex controller, say, an H_∞ controller, is used. This model-based H_∞ controller is able to handle the high order dynamics of the plant at high frequencies. Another advantage of this design is that it shows robustness against model uncertainties, *e.g.*, the change of the deflection detection caused by using another cantilever, and modeling errors in the identification.

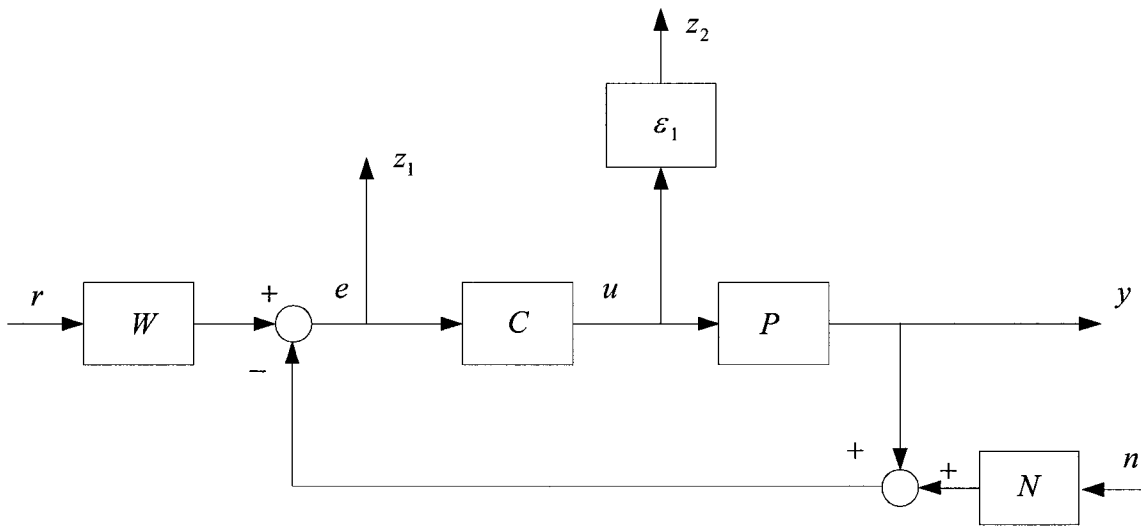

Figure 4.15: Schematic block diagram of the closed-loop system.

4.2.1 H_∞ Controller Design and Simulation

The H_∞ optimal controller is characterized by minimizing the H_∞ norm of the system, say $\left\| \hat{t}_{zw} \right\|_\infty$, for the standard setup of Figure 4.14. The procedure of designing an optimal H_∞ controller can be summarized in the following steps.

The first step towards H_∞ controller design is to form the generalized plant G as shown in Figure 4.14. In the system, the exogenous inputs consist of reference command r and the sensor noise n, the control input is u, the signals to-be-controlled are the sensed output v and the error signal e. In order to formulate the performance objectives and physical constraints into the controller design, appropriate weighting functions are used to regulate system inputs and outputs [9, 23, 28]. The weighting transfer functions in Figure 4.15 are:

- W is selected to be a low-pass filter with bandwidth ω which reflects the operating frequency range, say $[0, \omega]$, for good tracking. The weighting function W is chosen to reflect the basic control specification, that is, to make the magnitude of the sensitivity function S (the transfer function from r to e) small over the frequency range $[0, \omega]$.

- The weighting function N is chosen to reflect the frequency contents of the sensor noise. Minimizing the magnitude of T by appropriately choosing N can ensure low control gains at high frequencies since the frequency spectrum of the noise is usually much higher than that of the reference input. Here T represents the transfer function from n to y.

- The weight ϵ_1 which is imposed on the control signal u is a small positive scalar. It can restrict the magnitude of the control signal such that it is within the saturation limit.

To form the generalized plant G, we have to identify the inputs and outputs in Figure 4.14. Here, the regulated system outputs are

$$\begin{bmatrix} z \\ v \end{bmatrix} = \begin{bmatrix} e \\ \epsilon_1 u \\ v \end{bmatrix} = \begin{bmatrix} Wr - Pu - Nn \\ \epsilon_1 u \\ Wr - Pu - Nn \end{bmatrix} \tag{4.10}$$

and the system inputs are

$$\begin{bmatrix} w \\ u \end{bmatrix} = \begin{bmatrix} r \\ n \\ u \end{bmatrix} \tag{4.11}$$

Therefore, the generalized plant G can be described as

$$G = \begin{bmatrix} \begin{bmatrix} W & -N \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} -P \\ \epsilon_1 \end{bmatrix} \\ \begin{bmatrix} W & -N \end{bmatrix} & -P \end{bmatrix} \tag{4.12}$$

The selection of the weighting functions often involves trial and error and fine tuning. Initially, the weighting functions are determined based on the theory in [24]. Next, the closed-loop system will be simulated by using MATLAB and the parameters are adjusted to get better system response.

To achieve good tracking over $[0, 1.8k\text{Hz}]$ which is close to the nature frequency of the piezo stage, the weighting function W is chosen such that it has high gains at low frequencies and low gains at high frequencies. More precisely, W is in the form of a first order system

$$W(s) = \frac{1.2 \times 10^{-4}s + 5.2}{3.9 \times 10^{-4}s + 1} \tag{4.13}$$

From the Bode plots in Figure 4.16 we can see that the inverse of W(s) put an upper bound on the magnitude of the transfer function it is used to scale, that is, $1/W$ has a gain of 0.2 at frequencies which are less than 110Hz and a gain of 3.16 around $2.1k$Hz.

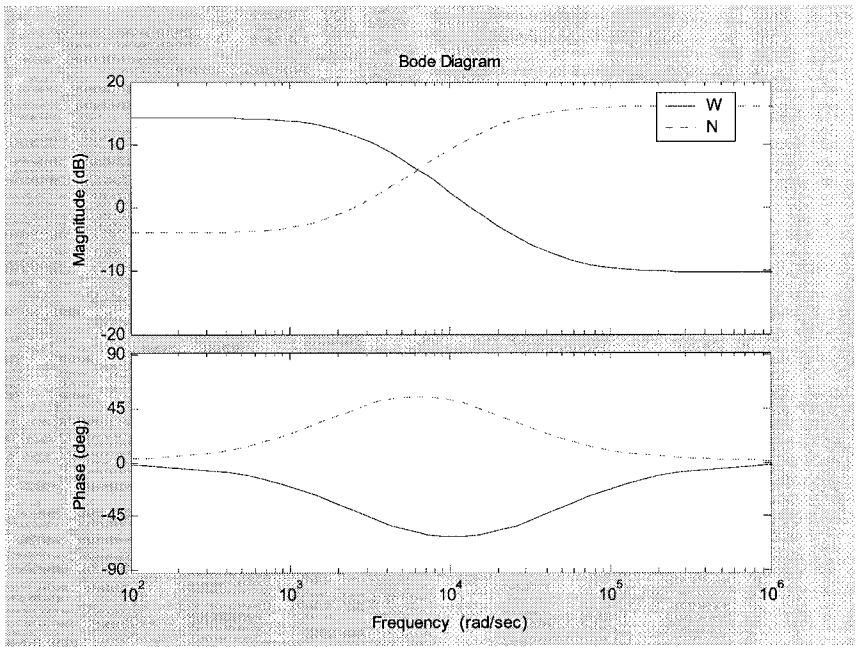

Figure 4.16: Bode diagrams of the weighting transfer functions.

This scaling enables that the sensitivity function S is small at low frequencies, thus assuring good tracking over the desired frequency range.

The objective of adding the weighting function N is to prevent high frequency sensor noise amplification. The sensor noise in the closed-loop system is mainly from the electronic circuit which transforms the cantilever displacement signal into the voltage signal. This sensor noise, which combines the mechanical and acoustic noises from the environment will reduce the image resolution of the AFM. To reject the sensor noise, the weighting function N is employed on the complementary sensitivity

function T. Here, N is chosen as

$$N(s) = \frac{3.155 \times 10^{-4}s + 0.631}{5 \times 10^{-5}s + 1} \tag{4.14}$$

which has high gains at high frequencies as shown in Figure 4.16. Unlike the weighting function W which puts a lower bound on the system bandwidth, N specifies the roll-off of the open-loop system by putting an upper bound (the inverse of N) on the magnitude of T.

To restrict the magnitude of the control signal u, an upper bound $1/\epsilon_1$ is placed on the magnitude of the transfer function CS. In this case we take $\epsilon_1 = 0.1$.

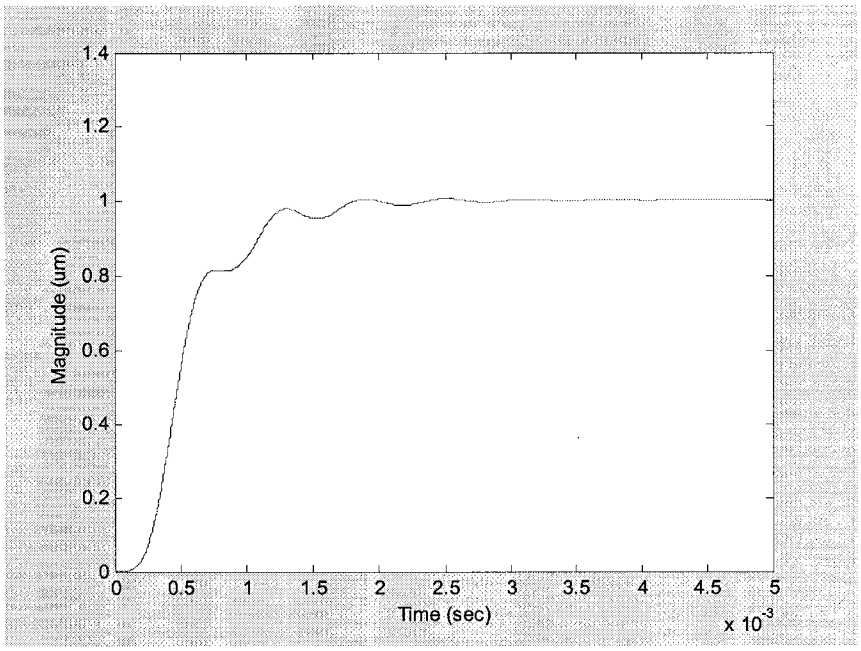

Figure 4.17: Step response of the closed-loop system controlled by the H_∞ controller.

Now the generalized plant model has been built as seen in Eq.(4.12). Then the control problem is to find an optimal controller C such that $\left\| \hat{t}_{zw} \right\|_\infty$ is minimized.

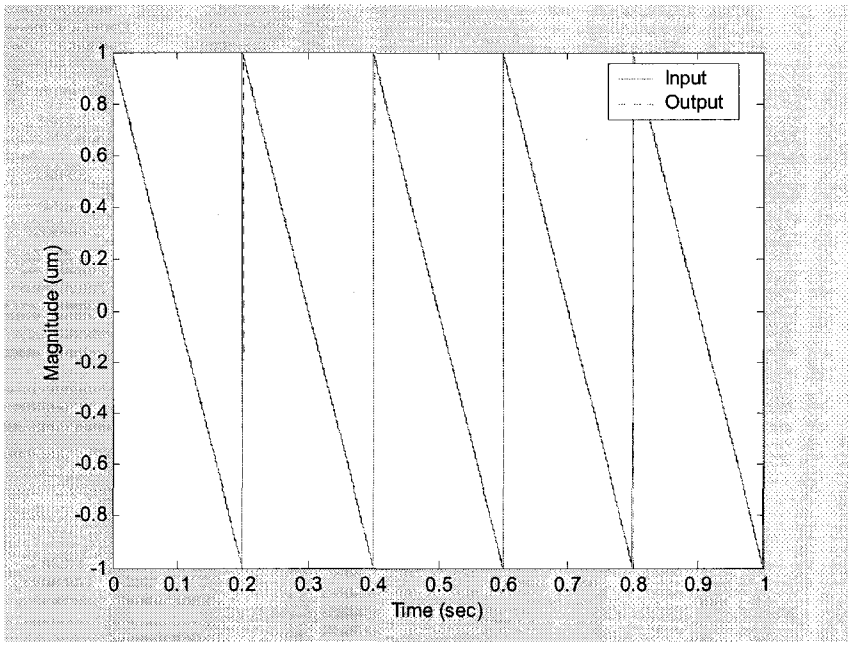

Figure 4.18: Tracking of 5Hz sawtooth signal with the H_∞ controller.

In practice, we usually simplify the design problem to find a suboptimal controller that is very close in the norm sense to the optimal ones. In other words, for any $\gamma > \gamma_{opt} > 0$, we can find a controller that satisfies $\left\| \hat{t}_{zw} \right\|_\infty < \gamma$. By using the μ-toolbox and *hinfsyn* function in MATLAB we calculate the controller C as

$$\begin{aligned} C(s) = 2.5 \times 10^{11} & \frac{6.3 \times 10^{-5} s^{11} + 341.7 s^{10} + 1.2 \times 10^{8} s^{9} + 5.8 \times 10^{14} s^{8} +}{s^{12} + 4.2 \times 10^{4} s^{11} + 1.7 \times 10^{12} s^{10} + 6.8 \times 10^{16} s^{9} +} \\ & \frac{6.2 \times 10^{19} s^{7} + 2.4 \times 10^{26} s^{6} + 7 \times 10^{30} s^{5} + 1.0 \times 10^{35} s^{4} +}{7.2 \times 10^{23} s^{8} + 2.8 \times 10^{28} s^{7} + 5.4 \times 10^{32} s^{6} + 8.5 \times 10^{36} s^{5} +} \\ & \frac{1.5 \times 10^{39} s^{3} + 9.9 \times 10^{42} s^{2} + 7.8 \times 10^{46} s + 1.6 \times 10^{50}}{8.8 \times 10^{40} s^{4} + 7.1 \times 10^{44} s^{3} + 4.0 \times 10^{48} s^{2} + 1.4 \times 10^{52} s + 1.96 \times 10^{55}} \end{aligned} \tag{4.15}$$

for $\gamma = 6.3176$. The step response of the closed-loop system is seen in Figure 4.17. We notice that there is considerable improvement in the rising time compared with

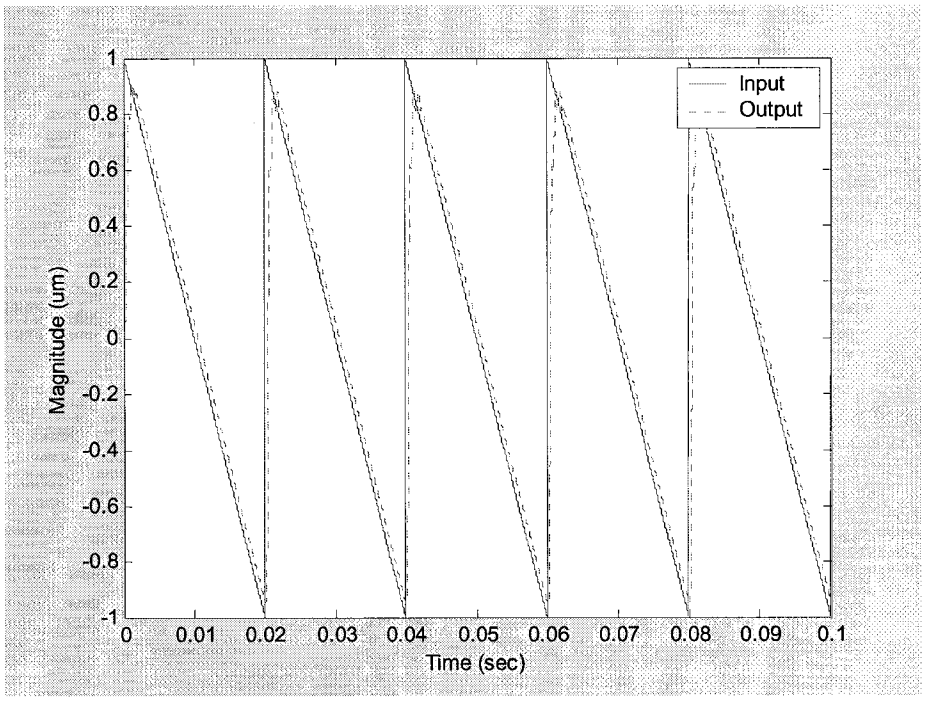

Figure 4.19: Tracking of 50Hz sawtooth signal with the H_∞ controller.

that of the PID controller. We will compare the closed-loop bandwidth with different controllers in section 4.3. The closed-loop system tracking of the reference sawtooth signal with frequencies 5Hz and 50Hz are shown in Figures 4.18-4.19. We can find considerable improvement in the tracking errors especially for the high frequency input signal compared with the PID controller.

4.2.2 Consideration of Model Uncertainty

In practice, the difference between the actual system and the system model which is used to design the controller always exists. The H_∞ optimal control is said to be

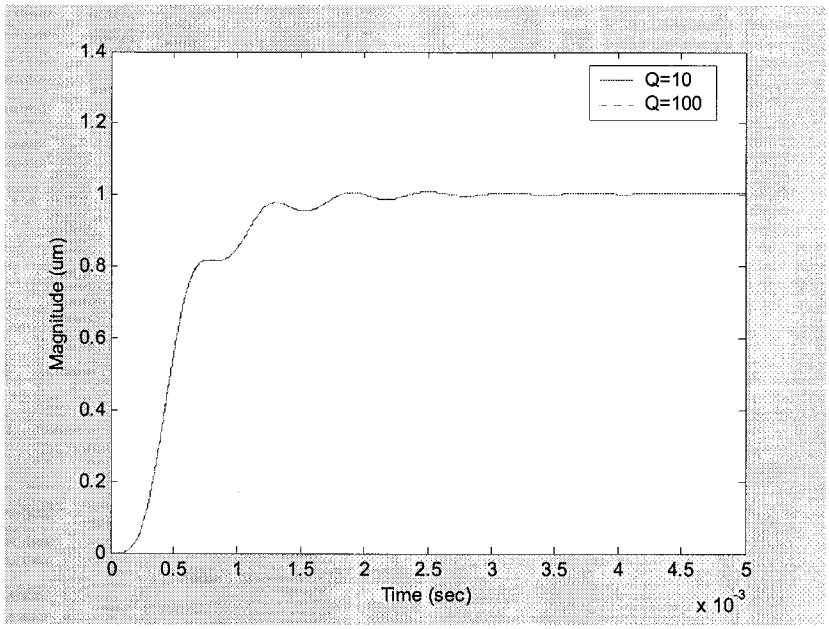

Figure 4.20: Step responses of the closed-loop system with different Q factors controlled by the same H_∞ controller.

robust if it is insensitive to these differences or model uncertainties. In this section, we will test the H_∞ controller which has been designed in the last section on the AFM system with parametric uncertainties [24]. The model uncertainties can be presented as follows:

- The change of the cantilever model by using a different cantilever
- The uncertain parameter (Q factor) of the cantilever.

Two cantilevers are employed in the simulation to test the robustness of the designed H_∞ controller, one is the cantilever in the lab which has been modeled in Chapter 3 and the other is from Digital Instruments [32]. The parameters of the

		Cantilever used	Cantilever from Digital Instruments
ω_n	Resonance frequency	$13k$Hz	$16k$Hz
k	Spring constant	$0.2N/m$	$0.2N/m$
R	Radius of the tip	$30nm$	$10nm$

Table 4.1: Parameters of the cantilevers

	Cantilever used	Cantilever from Digital Instruments
$\|e\|_2^2$ of tracking input of 5Hz	3.65×10^{-3}	3.64×10^{-3}
$\|e\|_2^2$ of tracking input of 10Hz	6.9×10^{-3}	6.89×10^{-3}

Table 4.2: Tracking errors of the closed-loop systems with different cantilevers.

cantilevers are listed in Table 4.1. The transfer function of the Digital Instruments cantilever from input u_1 to output y is

$$G_c(s) = \frac{2.56 \times 10^8}{s^2 + 1600s + 1.28 \times 10^{12}}, \text{ with the quality factor } Q = 10 \tag{4.16}$$

We measure the system performance by comparing the 2-norm of the tracking error, which is defined as

$$\|e\|_2^2 = \int_{-\infty}^{\infty} \|r(t) - y(t)\|^2 \, dt \tag{4.17}$$

The values of the error square of tracking sawtooth input signals with different frequencies are listed in Table 4.2, where the integral time range is [0, $500ms$]. The H_∞ controller shows robustness to the change of the plant model.

When we modeled the dynamics of the cantilever in Chapter 3, the quality factor Q is undetermined and is taken as 10 which is a normal value of cantilevers for contact mode AFMs. To test the robustness of the designed H_∞ controller against model uncertainties, we use two different values of the Q factor ($Q = 10$ and $Q = 100$).

	Cantilever with $Q = 10$	Cantilever with $Q = 100$
$\|e\|_2^2$ of tracking input of 5Hz	3.65×10^{-3}	3.66×10^{-3}
$\|e\|_2^2$ of tracking input of 10Hz	6.9×10^{-3}	6.93×10^{-3}

Table 4.3: Tracking errors of the closed-loop systems with different Q factors.

Figure 4.20 shows the step response. The tracking errors of the closed-loop system controlled by the same H_∞ controller are compared in Table 4.3, where the integral time range is [0, $500ms$]. The values of the two errors are almost identical which indicates that the H_∞ controller show considerable robustness against the effect of uncertainties in the cantilever dynamics.

4.3 Comparison of Different Controllers

In this section, the performance of the three controllers, namely, the phase-lag controller, PID controller and the H_∞ controller will be compared in two ways [19, 21]:

- Performance in the time domain and frequency domain.
- Tracking the input signals with different frequencies

The transient and steady-state performance of the three controllers are listed in Table 4.4. The H_∞ controller has the shortest rising time and the scanning time is reduced due to its high bandwidth. Thus, the H_∞ controller shows great improvement in the overall performance compared with the conventional PID controller.

	Phase-lag	PID	H_∞
Rising time (ms) 2%	13	2.8	1.3
Steady-state error	$\leq 1\%$	0	$\leq 0.3\%$
Bandwidth of closed-loop system (rad/s)	353	1.37×10^3	2.93×10^3

Table 4.4: Comparison of the closed-loop systems controlled by different controllers.

Figures 4.6, 7, 12, 13, 18, 19 in the previous sections show the performance of the three controllers for 1, 5, 50Hz reference signals. It can be seen that all the controllers track the 1Hz signal well. However, the H_∞ controller outperforms the other two controllers while tracking high frequency input signal ($>$ 5Hz). The fact that the H_∞ controller tracks high frequency input signals with the least distortion is consistent with the comparison result of the previous part. In addition to the above advantages, the H_∞ design incorporates the performance objectives like bandwidth and resolution in its weighting functions in contrast to the often tedious way of tuning parameters in the PID design. Therefore, we can reach the conclusion that H_∞ control is better for the contact mode AFMs than the other two designs we implemented.

Chapter 5

Implementation

In this chapter, experimental implementation of the designed controllers will be addressed. The overview of the control system and the definition of the design problems which include choosing the appropriate DSP board and the Analog-to-Digital Converter (ADC) board are introduced first. This is followed by setting up the communication between the existing TMS320C32 DSP and the chosen DSP, the ADC board and the chosen DSP which stores the software programs of the controllers that have been designed in the last chapter.

5.1 The Control System Overview

The overall control system of the IBM Magnascope 2000 AFM which adjusts the vertical movement of the piezo stage and the cantilever is shown in Figure 5.1. The controller box mainly includes the following two parts:

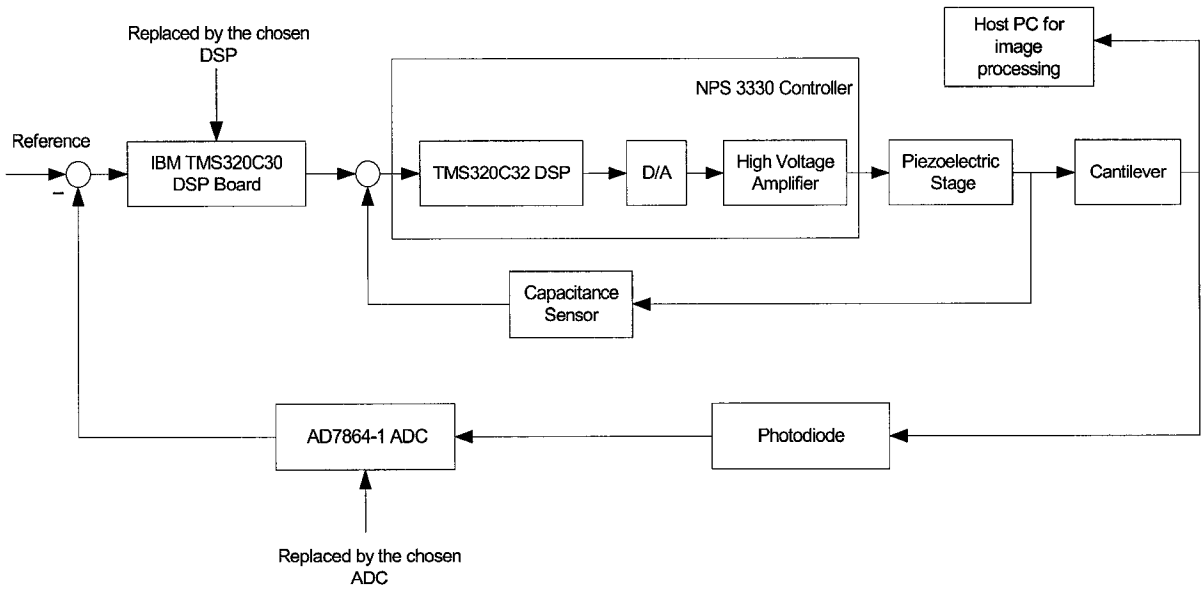

Figure 5.1: The block diagram of the IBM Magnascope 2000 AFM.

- The NPS 3330 controller is manufactured by Queensgate Instrument Ltd. It has a built-in TMS320C32 DSP board and a high voltage amplifier which amplifies the voltage signal from the output of TMS320C32 to drive the piezoelectric stage. Since the NPS 3330 has been developed for precise positioning of the piezo stage, it has PID algorithms stored in the on-board memory which provide the closed-loop control of the piezo movement in x, y and z axises respectively. It comes with a virtual front panel software (NanoControl Panel 2.6) which runs on the platform of Labview. This software enables users to optimize the PID loop parameters for any chosen channel (axial movement of the piezo) and set the command position of the piezo in three dimensions as well. However, the most

important feature of the NPS 3330 is that it can be integrated into customized application design by connecting to other DSP boards through its high speed DSP interface. It enables other DSPs get access to the position control of the piezo stage and integrates NPS 3330 into specific application environment through DLL command language which is provided by Queensgate.

- The customized DSP board which was built by IBM accommodates two floating-point TMS320C30 DSPs that form the closed-loop control to maintain the constant force between the sample and the cantilever tip. This DSP board also has four plug-in cards which perform the auxiliary functions such as driving the motor that moves the sample-holding stage and controlling the optical camera that is used for fine positioning of the tip before the scanning starts, *etc.* As one of the four plug-in cards, the four-channel 12-bit Analog-to-Digital converter card AD7864-1 ADC from Analog Devices converts the voltage signal from the split photodiode of $\pm 5V$ into the digital signal that can be utilized by the C30 processors.

However, we have little knowledge of how this controller box works due to the lack of corresponding driver software and user manuals. As a result, we were not able to get access to the DSP board or make any change to the program to meet our control requirement. To make this whole system work effectively as a basic contact mode AFM, we replaced the core components of the AFM– the IBM TMS320C30 DSP board and the ADC card with the ones we can run our programs for the designed

control algorithms.

5.2 Choose the Appropriate DSP Board and the ADC Card

The DSP board which will suit our needs should undertake the following tasks:

1. Can be programmed using the H_∞ and the PID control algorithms.
2. Connect to the PC which runs the operator interface software, such as adjusting control parameters and scanning commands.
3. Set commands to TMS320C32 DSP board in NPS 3330 to move the piezo stage.
4. Get the digital signal from the ADC board.

The two DSP boards used are: TMS320C6701 EVM board from Texas Instruments and Motorola DSP 56002 EVM board which is listed as a connection example to NPS 3330 by Queensgate. The features of these two boards are listed in Table 5.1.

Although the Motorola DSP 56002 EVM has an on-board ADC (CS4215), it cannot be used as a replacement of the 4-channel AD7864-1 ADC since it only has two channels which have been designed for processing audio signals. Therefore, we need an additional ADC card that can be connected to DSP 56002 EVM and function

Feature	**TMS320C6701 EVM** [41]	**DSP 56002 EVM** [34]
CPU clock rate	120MHz	66MHz
Processor type	32-bit floating point	24-bit floating point
PC Interface	PCI plug-in interface	OnCE serial port
ADC Interface	80-pin McBSP connector	Serial port C $J10$
DSP Interface	Six pins of the McBSP connector	Serial port C $J10$

Table 5.1: The comparison of TMS420C6701 EVM and DSP56002 EVM.

similarly to AD7864-1. The 56002 EVM provides the synchronous serial interface (SSI) as a full-duplex serial port for serial devices including other DSPs and industry-standard ADCs. However, after examined the wiring diagrams, it was found that there was only one connector (J10) which could be configured as a serial port to interface with either a DSP or the ADC card, not for both of them at the same time. It is impossible to build a control system without adding extra electronic circuits to DSP 56002 EVM. Thus, we turn to the other option, namely, TMS320C6701 EVM.

The high-performance 32-bit DSP board has an on-board expansion peripheral interface which allows serial devices to be added to the EVM via a daughterboard. This interface includes two multichannel buffered serial ports (McBSP) where each port can be configured independently to meet specific needs. Thus, the McBSP 1 is assigned to connect the chosen ADC daughterboard while McBSP 0 is treated as an interface to NPS 3330. The second connection will be addressed in detail in section 5.3.

The 16-bit ADS8364 EVM daughterboard from Texas Instruments is chosen as a replacement of AD7864-1. This board operates at 250 KHz and allows 6-channel

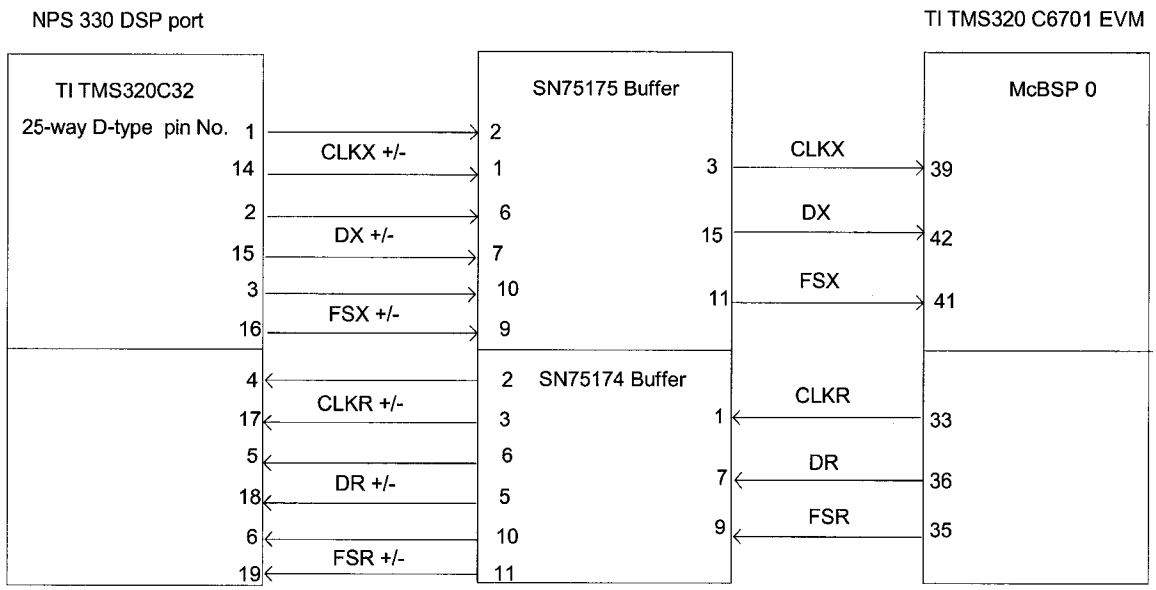

Figure 5.2: The inter-connection block diagram of C6701 and C32 DSP.

simultaneous sampling. Most importantly, it can interface seamlessly with C6701 platform through the 80-pin McBSP connector. The hardware communication setup between ADS8364 EVM and C6701 EVM will be discussed in section 5.4.

5.3 Set Up Communication Between TMS320C6701 EVM and NPS 3330

5.3.1 Hardware Connection

Since both C32 and C6701 are serial devices, data communication can be set up through the serial ports on each platform. However, the serial port (or the DSP port

defined by Queensgate) of NPS 3330 is a 25-way D-type port which accepts differential inputs and outputs [37]. In other words, the six single-ended signals which are used for McBSP communication on C6701 have to be transformed into differential forms.

In the experiment, we used a quadruple differential line receiver SN75175 for the receive section of McBSP 0 and a line driver SN75174 for the transmit section. The inter-connection block diagram is shown in Figure 5.2.

5.3.2 Serial Communication Software for TMS320C32

The serial port on C32 has two sections: receive and transmit. They can be configured to transfer 8, 16, 24 or 32 bits of data per word simultaneously in both directions. The serial port is setup in the handshake mode which allows for direct connection between DSPs. Once the serial port of C32 receives a word (C32 only needs to receive commands from C6701 in our case) in this mode, it sends a single acknowledge 0 bit to the receiving serial port on C6701 EVM. However, since C6701 platform operates in its standard mode other than the handshake mode, this 0 bit will be ignored by the receiving McBSP 0 without losing any information. The six serial I/O pins which are used for serial communication are listed in Table 5.2.

The serial communication is controlled by six control registers. We have to configure the selected bits of these registers to build effective serial communication. These registers are:

- Global control register which is a 32-bit register that contains the global control

Pin Name	Function	Type
CLKX	Transmit serial block	Output
DX	Transmitted serial data	Output
FSX	Transmit frame synchronization	Output
CLKR	Receive serial block	Input
DR	Received serial data	Input
FSR	Receive frame synchronization	Input

Table 5.2: Serial interface pins.

bits for the serial port.

- Two control registers (receive and transmit) for the six serial I/O pins.
- Receive/Transmit timer control register which contains the control bits for the timer module.
- Receive/Transmit timer counter register.
- Receive/Transmit timer period register which contains the value that sets the transfer clock.

The C codes which configure the control registers of C32 are listed in Appendix A. In the program, both receive and transmit registers have been configured to transfer 32 bits per word. Moreover, since the instruction cycle time of C6701 (120MHz) is much shorter than that of C32 (25MHz), C6701 has to slow down its transmit clock frequency to maintain the proper communication with C32. As a receiver, C32 uses an external signal from CLKR as clock resource which has a maximum frequency less than CPU clock/2.6 =9.165MHz [42]. Thus, the maximum receive frequency of C32

is divided by the value in the receiver timer period register, namely

$$R = 9.165/2 = 4.808\text{MHz} \tag{5.1}$$

5.3.3 Serial Communication Software for TMS320C6701

The standard serial operation of the McBSP on C6701 is different from the serial port of C32 in the following ways [35.36]:

- Triple-buffered receive section allows a continuous data stream.
- Either the CPU or the DMA (Direct Memory Access) /EDMA (Enhanced Direct Memory Access) controller can read the received data from the data receive register (DRR) and write the data to the data transmit register (DXR). Thus, the serial transfer interrupt request can be handled by DMA/EDMA without interrupting the normal operation of the CPU while for C32 only the CPU can process the data transfer interrupt.
- The size of the frame has to be determined as well as the size of every word before starting the data transfer. The frame refers to a data block and in our case a single-phase frame mode is chosen. In other words, each frame contains a word and each word has 32 bits.
- The data transfer frequency shift clock is programmable. As discussed in the previous section, the data transmit speed of C6701 has to be adjusted to ensure

that no data is lost during the transfer. The transmit clock of C6701 McBSP 0 is driven by the internal sample rate generator output clock (CLKG). CLKG has a frequency which can be calculated as

$$S = \text{Sample rate generator iutput clock}/(\text{CLKGDV}+1) \qquad (5.2)$$

where the input clock is the same as the CPU clock and CLKGDV is a scalar which should be assigned through the sample rate generator register (SRGR). The value of CLKGDV is chose to be 24 which guarantees that S is less than the receive rate R of C32. The communication program written by using Code Compose Studio (CCS) is included in Appendix A.

5.4 Interface A/D Converter Daughterboard ADS8364 EVM to C6701 EVM

5.4.1 Hardware Connection

The ADS8364 is a 6-channel, simultaneous sampling, 16-bit parallel ADC. The device incorporates an internal buffer that can be powered from the same 3.3 V supply as the DSP which enables its direct interfacing to the 3.3 V C6701 DSP. The block diagram in Figure 5.3 [44] is a representative of how the ADS8364 would be connected in an actual system using software control with the TMS320C6701 DSP.

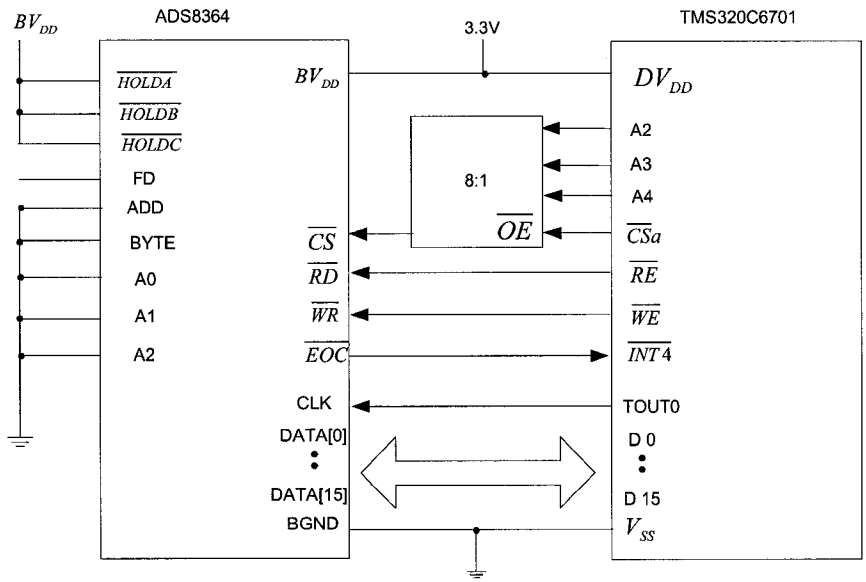

Figure 5.3: System block diagram of interfacing ADS8364 EVM with TMS320C6701 EVM .

5.4.2 Set Up the ADS8364 EVM Interface

The ADS8364 is operated from a clock frequency of 4MHz and all six channels of the ADS8364 provide a maximum throughput rate of 200 $KSPS$ [43]. The ADS8364 EVM uses four address lines ($A2$ through $A5$ with $W7$ closed) to access the data converter. The lower address lines control the $A0$, $A1$, and $A2$ pins of the ADC, while the upper most address line is sent through a single gate inverter to act as chip select ($\overline{CS}$).

The McBSP 1 of the C6701 EVM is configured as a general-purpose I/O port to interface with the ADS8364. The McBSP 1 is used to control the $\overline{HOLDx}$ pins, the ADD pin and the RESET pin. The first data (FD) signal is sent to the McBSP 1 as

an input to indicate when data from channel $A1$ is present. The C6701 EVM's read enable ($\overline{RE}$), write enable ($\overline{WE}$), and daughtercard chip select ($DC\_\overline{CS}a$) are all used in the parallel interface to read data from and write commands to the ADC.

5.4.3 Set Up the TMS320C6701 EVM

Initially, McBSP1 is set up as a general-purpose I/O port. For this, the associated sample rate generator (SRGR) and port control register (SPCR) are cleared and the pin control register (PCR) for the desired functions is set. The ADS8364 EVM uses CLKXa, CLKRa, FSXa, FSRa, and DXa as general-purpose outputs, while using DRa as a general-purpose input. To achieve the setup shown in Figure 5.3, the HOLDx lines need to be set, the ADD line needs to be cleared, and FD needs to be set as an input [40].

Next, the conversion clock is set up. The C6701 EVM's internal timer, TOUT0, can be used as the conversion clock source for the ADS8364 EVM. TOUT0 provides approximately a 4.5 MHz conversion clock to the converter.

Finally, the Chip Select ($\overline{CS}$), Read ($\overline{RD}$) and Write ($\overline{WR}$) signals have to be configured. Chip select to the ADS8364 is an active-low input signal. When $\overline{CS}$ is low, the parallel data lines reflect the current state of the output buffers. The ADS8364 EVM uses the C6701 EVM's $DC\_\overline{CS}a$ signal as a master chip select for the board. $DC\_\overline{CS}a$ controls the output enable ($\overline{OE}$) of a 4-bit 1-of-2 FET multiplexer, which feeds the address bus of the DSP to the ADC. Jumper W7 controls address line

selection—A2 through A4.

The C code of interfacing the ADS8364 EVM to the TMS320C6701 EVM was developed with Code Composer Studio on the C6701 EVM platform which is included in Appendix B. This program shows how to configure the DSP, initialize the data converter, and process an interrupt (via the EOC pin) from the data converter.

Chapter 6

Conclusion

In this thesis, the theory and operation of AFMs which work in the contact mode are described in detail. A plant model is outlined and employed to design the optimal feedback controller which achieves as fast as possible scanning rate with high resolution. For this, models of the three parts of the AFM, namely, the piezo stage, the cantilever and the photodiode, are integrated to provide a more comprehensive description of the performance of the particular AFM–IBM Magnascope 2000.

The model of the piezoelectric stage has been derived from the step response of the system while the photodiode model is built based on the experimental data. The Johnson-Kendall-Roberts (JKR) force model is used to develop the model which describes the macroscopic interactions between the tip and the sample surface. We note that only repulsive force has been integrated into the state-space model of the cantilever which has been linearized at the operating point of $z = 0.12\ nm$, where z

represents the tip-sample separation. The controller design is based on the linearized system which operates around that operating point.

After the plant model has been constructed, three feedback laws are designed to meet the specific control requirements. The requirements are specified in the timedomain, defined for the step response of the closed-loop system, and in the frequencydomain. A phase-lag and a PID controllers are designed to fulfill the time-domain specifications while an H_∞ controller is mainly designed to follow the high frequency arbitrary inputs (the sample surface). The MATLAB simulation results show that the H_∞ controller has the shortest rising time (0.2 ms), the highest bandwidth ($2.51 \times 10^4 rad/s$) and the smallest tracking errors especially for the input signal with frequencies more than 100Hz. In general, the H_∞ control algorithm outperforms its two counterparts in achieving high speed scanning with the smallest tracking error. In addition to this, the H_∞ controller eliminates the tedious task of tuning the parameters in the traditional PID architecture since it incorporates the performance objectives like bandwidth and resolution in its design.

The experimental implementation of the designed PID and H_∞ controllers is described in detail. The C codes of the controllers which are programmed by using the Code Composer Studio (CCS) from TI are stored in TMS320C6701 EVM. Since the program that drives the piezo stage was built in the TMS320C32 DSP board of the NPS 3330 controller from Queensgate, the serial communication between C6701 EVM and C32 DSP board need to be set up both in hardware connection and software

configuration. Additionally, an Analog-to-Digital converter (ADC) daughterboard, namely, ADS8364 EVM from TI is interfaced with C6701 EVM through the McBSP 1 port. The hardware setup and the software configuration have been addressed in detail.

After building the hardware connection and programming C codes for the serial communication between C6701 and C32, we found that C6701 can transmit data properly during the debugging stage. However, the C codes of C32 data receiving cannot run correctly due to the missing C head files (*.h) and C library files (*.lib) which should be provided by Queensgate Instrument to integrate NPS 3330 into customized applications. As a result, the serial data transfer between C6701 and C32 cannot be set up.

The C codes for the H_∞ controller and the PID controller (the PID parameters may need to be adjusted during the operation of the AFM) are included in Appendix C.

To run the AFM system in the future, the following tasks have to be carried out:

- Solve the DSP serial communication problem by either having corresponding C files from Queensgate or obtaining an emulator from TI to get direct access to the C32 processor. The later method will involve reading compiled machine code from the C32 processor which might need a lot of work to understand.
- Develop an image processing software or purchase the package from a third party.

Bibliography

[1] T. R. Albrecht, S. Akamine, T. E. Carver and C. F. Quate, "Microfabrication of cantilever styli for the atomic force microscope," *J. Vac. Sci. Technol.* A 8(4), pp. 3386-3396 ,1990.

[2] C. Bai, *Scanning Tunneling Microscopy and Its Applications, 2nd* Ed. New York: Springer, pp. 63-71, 2000.

[3] G. Binnig, C. F. Quate and C. Gerber, "Atomic force microscope," *Phys. Rev. Lett.*, vol . 56, pp. 930-933, 1986.

[4] D. A. Bonnell, *Scanning Probe Microscopy and Spectroscopy-Theory, Techniques, and Applications, 2nd* Ed. Wilwy-VCH, Inc., pp. 14-25, pp. 172-182, 2001.

[5] K. J. Bruland, J. L. Garbini, W. M. Dougherty and J. A. Sidles, "Optimal control of ultrasoft cantilevers for force microsopy," *J. Appl. Phys.*, vol.. 83, pp. 3972-3977, 1998.

[6] C. T. Chen, *Digital Signal Processing–Spectral Computation and Filter Design*, New York: Oxford University Press Inc., pp. 355-395, 2001.

[7] C. T. Chen, *Analog & digital control system design: transfer-function, state-space, and algebraic methods,* Saunders College Publishing, pp. 312-332, 1993.

[8] J. Chen, R. K. Workman, D. Sarid and R. Höper, "Numerical simulations of a scanning force microscope with a large-amplitude vibrating cantilever," *Nanotechnology*, vol. 5, pp. 199-204, 1994.

[9] T. Chen and B. Francis, *Optimal Sampled-Data Control Systems*, Springer, pp. 21-32, 1995.

[10] G. A. G. Cidade, G. Weissmuller, and P. M. Bisch, "A Microcontroller-based System for Piezoscanner Nonlinearity Correction: Atomic Force Microscope," *Rev. Sci. Instrum.*, vol. 69, pp. 3593-3597, 1998.

[11] R. C. Dorf, *Modern Control Systems*, 5*th* Ed. Addison-Wesley Publishing Company, ,pp. 14-25, 1989.

[12] A. G. Hatch, R. C. Smith and T. De, "Model Development and Control Design for High Speed Atomic Force Microscopy," SAMAI technical report #2004-3, 2004.

[13] J. N. Israelachvili, *Intermolecular and surface forces,* New York: Academic Press, 1985.

[14] B. J. Lurie and P. J. Enright, *Classical Feedback Control with MATLAB*, Marcel Dekker Inc., pp. 190-192, 2000.

[15] Y. Martin, C. C. Williams and H. K.Wickramasinghe, "Atomic force microscope–force mapping and profiling on a sub 100-Å scale," *J. Appl. Phys.*, vol. 61, pp. 4723-4729, 1987.

[16] E. Meyer, N. M. Amer, "Novel optical approach to atomic force microscopy," *Appl. Phys. Lett.*, vol. 53, pp. 1045-1047, 1988.

[17] E. Meyer, H. Heinzelmann, P. Grutter, Th. Jung, Th. Weisskopf, H. R. Hidber, R. Lapka, H. Rudin and H. J. Guntherodt, "Comparative study of lithium fluoride and graphite by atomic force microscopy (AFM)," *J. Microsc.*, vol. 152, pp. 269-280, 1988.

[18] D. Rimai, Particle-Substrate Interactions: Microscopic Aspects of Adhesion [Online], Available:www.clarkson.edu/projects/fluidflow/courses/me537/Rimai_2.pdf.

[19] D. Rugar and P. K. Hansma, "Atomic force microscopy," *Physics Today*, vol. 43, pp. 23-30, 1990.

[20] S. Salapaka, A. Sebastian, J. P. Cleveland and M. V. Salapaka, "High bandwidth nano-positioner: A robust control approach," *Rev. Sci. Instrum.*, vol. 73, pp. 3232-3241, 2002.

[21] D. Sarid, *Scanning Force Microscopy With Applications to Electric, Magnetic, and Atomic Forces*, New York: Oxford University Press Inc., pp. 14-32, 1991.

[22] G. Schitter, F. Allgower and A. Stemmer, "A new control strategy for high-speed atomic force microscopy," *Nanotechnology*, vol. 15, pp. 108-114, 2004.

[23] G. Schitter, P. Menold, H. F. Knapp, F. Allgower and A. Stemmer, "High performance feedback for fast scanning atomic force microscopes," *Rev. Sci. Instrum.*, vol. 72, pp. 3320-3327, 2001.

[24] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control Analysis and Design*, Jonhn wiley & Sons Ltd, pp. 56-62, 253-258, 1996.

[25] Y. Xu, P. D. Atherton, T. R. Hicks and M. McConnell, *Design and Characterisation of Nanometer Precision Mechanisms (I)*, Queensgate Instruments Ltd., Waterside Park, Bracknell, RG12 1RB, UK,1999.

[26] Y. Xu, P. D. Atherton, P. Rhead, T. R. Hicks and M. McConnell, *Design and Characterisation of Nanometer Precision Mechanisms (II)*, Queensgate Instruments Ltd., Waterside Park, Bracknell, RG12 1RB, UK,1999.

[27] S. Yang and W. Huang, "Three-dimensional displacements of a piezoelectric tube scanner," *Rev. Sci. Instrum.,* vol. 69, pp. 226-229, 1998.

[28] Q. Zhong, D. Innis, K. Kjoller, and V. B. Elings, "Fractured polymer/silica

fiber surface studied by tapping mode atomic force microscopy," *Surface Science Letters*, vol. 290, pp. L888-L692 1993.

[29] K. Zhou, *Essentials Of Robust Control*, Prentice Hall, 1997, pp. 269-279.

[30] Alberta Microelectronic Centre, *Introduction to Atomic Force Microscopy-AMC Short Course*, 1996.

[31] Atomic Force Microscopes Tutorial [Online]. Available: http://www.pacificnanotech.com/3_Article_A_13.html

[32] Contact Etched Silicon Probes [Online], Veeco Metrology LLC, Available:www.veeco.com/ESPW_e.pdf.

[33] How AFM Works [Online]. Available: http://stm2.nrl.navy.mil/how-afm/how-afm.html.

[34] Motorola Inc., DSP56002 24-Bit Digital Signal Processor User's Manual, pp. 61-236.

[35] Pauli's Principle [Online]. Available: http://www.ethbib.ethz.ch/ exhibit/ pauli/ ausschliessung_e.html.

[36] Pauli exclusion principle [Online]. Available: http://en.wikipedia.org/ wiki/ Pauli_principle.

[37] Queensgate Instrument Ltd., *NPS3110, NPS3220, NPS3330 Operating Manual,* pp. 112-113, 2000.

[38] Texas Instruments Inc., TMS320C6000 Multichannel Buffered Serial Port (McBSP) Reference Guide SPRU580B, pp. 11-59, 2003.

[39] Texas Instruments Inc., Using the TMS320C6000 McBSP as a High Speed Communication Port, SPRA455A, pp. 1-32, 2001.

[40] Texas Instruments Inc., TMS320C6000 DSP Peripherals Overview Reference Guide SPRU190E, pp. 14-15, 2003.

[41] Texas Instruments Inc., TMS320C6201/6701 Evaluation Module Technical Reference SPRU305A, pp. 19-43, 1999.

[42] Texas Instruments Inc.,TMS320C3x User's Guide SPRU031E, pp. 397-42, 1997.

[43] Texas Instruments Inc., ADS8364 User's Guide SLAU084, pp. 5-28, 2002.

[44] Texas Instruments Inc., Software Control of the ADS8364 SLAA155, pp. 1-8, 2002.

Appendix A

Simulink Diagram

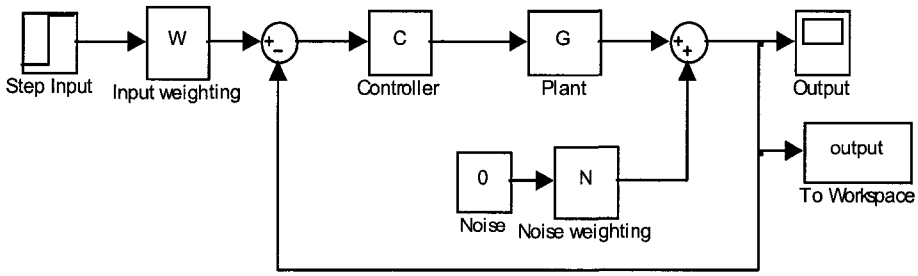

Figure A.1: Simulink diagram of the closed-loop system controlled by H_∞ controller.

Appendix B

C Code of Communication Between TMS320C6701 EVM and NPS 3330

B.1 Serial communication of TMS320C6701

```
/* Chip definition, change this accordingly */
#define CHIP_6701 1
#include <c6x.h>
#include <csl.h>
#include <csl_dma.h>
#include <csl_edma.h>
#include <csl_irq.h>
#include <csl_mcbsp.h>
#define FALSE 0
#define TRUE 1
#define DMA_XFER 8
#define XFER_TYPE DMA_XFER
#define BUFFER_SIZE 256
#define ELEMENT_COUNT 32
volatile int xmit0_done = FALSE;
MCBSP_Handle hMcbsp0;
#if (TRANSMIT)
DMA_Handle hDma2;
#endif
#if (RECEIVE)
DMA_Handle hDma1;
#endif
```

```
void init_mcbsp0_master(void);
void set_interrupts_dmax(void);
void init_mcbsp0_slave(void);
void set_interrupts_dmar(void);
extern far void vectors();
void main(void)
{
static int element_count, xfer_type;
static Uint32 dmaOutbuff[BUFFER_SIZE];
static Uint32 edmaOutbuff[BUFFER_SIZE];
IRQ_setVecs(vectors);
element_count = ELEMENT_COUNT;
xfer_type = XFER_TYPE;
CSL_init();
//McBSP act as a transmitter
#if (TRANSMIT)
init_mcbsp0_master();
MCBSP_enableSrgr(hMcbsp0);
DMA_reset(INV);
hDma2 = DMA_open(DMA_CHA2, DMA_OPEN_RESET);
DMA_configArgs(hDma2,
DMA_PRICTL_RMK(
DMA_PRICTL_DSTRLD_DEFAULT,
DMA_PRICTL_SRCRLD_DEFAULT,
DMA_PRICTL_EMOD_DEFAULT,
DMA_PRICTL_FS_DEFAULT,
DMA_PRICTL_TCINT_ENABLE,
DMA_PRICTL_PRI_DMA,
DMA_PRICTL_WSYNC_XEVT0,
DMA_PRICTL_RSYNC_DEFAULT,
DMA_PRICTL_INDEX_DEFAULT,
DMA_PRICTL_CNTRLD_DEFAULT,
DMA_PRICTL_SPLIT_DEFAULT,
DMA_PRICTL_ESIZE_32BIT,
DMA_PRICTL_DSTDIR_DEFAULT,
DMA_PRICTL_SRCDIR_INC,
DMA_PRICTL_START_DEFAULT
),
DMA_SECCTL_RMK(
DMA_SECCTL_DMACEN_DEFAULT,
DMA_SECCTL_WSYNCCLR_DEFAULT,
DMA_SECCTL_WSYNCSTAT_DEFAULT,
```

```
DMA_SECCTL_RSYNCCLR_DEFAULT,
DMA_SECCTL_RSYNCSTAT_DEFAULT,
DMA_SECCTL_WDROPIE_DEFAULT,
DMA_SECCTL_WDROPCOND_DEFAULT,
DMA_SECCTL_RDROPIE_DEFAULT,
DMA_SECCTL_RDROPCOND_DEFAULT,
DMA_SECCTL_BLOCKIE_ENABLE,
DMA_SECCTL_BLOCKCOND_DEFAULT,
DMA_SECCTL_LASTIE_DEFAULT,
DMA_SECCTL_LASTCOND_DEFAULT,
DMA_SECCTL_FRAMEIE_DEFAULT,
DMA_SECCTL_FRAMECOND_DEFAULT,
DMA_SECCTL_SXIE_DEFAULT,
DMA_SECCTL_SXCOND_DEFAULT
),
DMA_SRC_RMK((Uint32)dmaOutbuff),
DMA_DST_RMK(MCBSP_ADDRH(hMcbsp0, DXR)),
DMA_XFRCNT_RMK(
DMA_XFRCNT_FRMCNT_DEFAULT,
DMA_XFRCNT_ELECNT_OF(element_count)
)
);
set_interrupts_dmax();
DMA_start(hDma2);
MCBSP_enableXmt(hMcbsp0);
MCBSP_enableFsync(hMcbsp0);
while (!xmit0_done);
MCBSP_close(hMcbsp0);
DMA_close(hDma2);
#endif
#if (RECEIVE)
init_mcbsp0_slave();
MCBSP_enableSrgr(hMcbsp0);
DMA_reset(INV);
hDma1 = DMA_open(DMA_CHA1, DMA_OPEN_RESET);
DMA_configArgs(hDma1,
DMA_PRICTL_RMK(
DMA_PRICTL_DSTRLD_DEFAULT,
DMA_PRICTL_SRCRLD_DEFAULT,
DMA_PRICTL_EMOD_DEFAULT,
DMA_PRICTL_FS_DEFAULT,
DMA_PRICTL_TCINT_ENABLE,
```

```
DMA_PRICTL_PRI_DMA,
DMA_PRICTL_WSYNC_REVT0,
DMA_PRICTL_RSYNC_DEFAULT,
DMA_PRICTL_INDEX_DEFAULT,
DMA_PRICTL_CNTRLD_DEFAULT,
DMA_PRICTL_SPLIT_DEFAULT,
DMA_PRICTL_ESIZE_32BIT,
DMA_PRICTL_DSTDIR_DEFAULT,
DMA_PRICTL_SRCDIR_INC,
DMA_PRICTL_START_DEFAULT
),
DMA_SECCTL_RMK(
DMA_SECCTL_DMACEN_DEFAULT,
DMA_SECCTL_WSYNCCLR_DEFAULT,
DMA_SECCTL_WSYNCSTAT_DEFAULT,
DMA_SECCTL_RSYNCCLR_DEFAULT,
DMA_SECCTL_RSYNCSTAT_DEFAULT,
DMA_SECCTL_WDROPIE_DEFAULT,
DMA_SECCTL_WDROPCOND_DEFAULT,
DMA_SECCTL_RDROPIE_DEFAULT,
DMA_SECCTL_RDROPCOND_DEFAULT,
DMA_SECCTL_BLOCKIE_ENABLE,
DMA_SECCTL_BLOCKCOND_DEFAULT,
DMA_SECCTL_LASTIE_DEFAULT,
DMA_SECCTL_LASTCOND_DEFAULT,
DMA_SECCTL_FRAMEIE_DEFAULT,
DMA_SECCTL_FRAMECOND_DEFAULT,
DMA_SECCTL_SXIE_DEFAULT,
DMA_SECCTL_SXCOND_DEFAULT
),
DMA_SRC_RMK(MCBSP_ADDRH(hMcbsp0, DRR)),
DMA_DST_RMK((Uint32)dmaInbuff),
DMA_XFRCNT_RMK(
DMA_XFRCNT_FRMCNT_DEFAULT,
DMA_XFRCNT_ELECNT_OF(element_count)
)
);
set_interrupts_dmar();
DMA_start(hDma1);
MCBSP_enableRcv(hMcbsp0);
while (!recv0_done);
MCBSP_close(hMcbsp0);
```

```
DMA_close(hDma1);
#endif
}
void
init_mcbsp0_master(void)
{
MCBSP_Config mcbspCfg0 = {
MCBSP_SPCR_RMK(
MCBSP_SPCR_FRST_DEFAULT,
MCBSP_SPCR_GRST_DEFAULT,
MCBSP_SPCR_XINTM_DEFAULT,
MCBSP_SPCR_XSYNCERR_DEFAULT,
MCBSP_SPCR_XRST_DEFAULT,
MCBSP_SPCR_DLB_DEFAULT,
MCBSP_SPCR_RJUST_DEFAULT,
MCBSP_SPCR_CLKSTP_DEFAULT,
MCBSP_SPCR_RINTM_DEFAULT,
MCBSP_SPCR_RSYNCERR_DEFAULT,
MCBSP_SPCR_RRST_DEFAULT
),
MCBSP_RCR_RMK(
MCBSP_RCR_RPHASE_DEFAULT,
MCBSP_RCR_RFRLEN2_DEFAULT,
MCBSP_RCR_RWDLEN2_DEFAULT,
MCBSP_RCR_RCOMPAND_DEFAULT,
MCBSP_RCR_RFIG_DEFAULT,
MCBSP_RCR_RDATDLY_DEFAULT,
MCBSP_RCR_RFRLEN1_DEFAULT,
MCBSP_RCR_RWDLEN1_DEFAULT
),
MCBSP_XCR_RMK(
MCBSP_XCR_XPHASE_SINGLE,
MCBSP_XCR_XFRLEN2_DEFAULT,
MCBSP_XCR_XWDLEN2_DEFAULT,
MCBSP_XCR_XCOMPAND_DEFAULT,
MCBSP_XCR_XFIG_DEFAULT,
MCBSP_XCR_XDATDLY_1BIT,
MCBSP_XCR_XFRLEN1_DEFAULT,
MCBSP_XCR_XWDLEN1_DEFAULT
),
MCBSP_SRGR_RMK(
MCBSP_SRGR_GSYNC_DEFAULT,
```

```
MCBSP_SRGR_CLKSP_DEFAULT,
MCBSP_SRGR_CLKSM_INTERNAL,
MCBSP_SRGR_FSGM_FSG,
MCBSP_SRGR_FPER_DEFAULT,
MCBSP_SRGR_FWID_DEFAULT,
MCBSP_SRGR_CLKGDV_OF(0x0)
),
MCBSP_MCR_RMK(
MCBSP_MCR_XPBBLK_DEFAULT,
MCBSP_MCR_XPABLK_DEFAULT,
MCBSP_MCR_XMCM_DEFAULT,
MCBSP_MCR_RPBBLK_DEFAULT,
MCBSP_MCR_RPABLK_DEFAULT,
MCBSP_MCR_RMCM_DEFAULT
),
MCBSP_RCER_RMK(
MCBSP_RCER_RCEB_DEFAULT,
MCBSP_RCER_RCEA_DEFAULT
),
MCBSP_XCER_RMK(
MCBSP_XCER_XCEB_DEFAULT,
MCBSP_XCER_XCEA_DEFAULT
),
MCBSP_PCR_RMK(
MCBSP_PCR_XIOEN_DEFAULT,
MCBSP_PCR_RIOEN_DEFAULT,
MCBSP_PCR_FSXM_INTERNAL,
MCBSP_PCR_FSRM_DEFAULT,
MCBSP_PCR_CLKXM_OUTPUT,
MCBSP_PCR_CLKRM_DEFAULT,
MCBSP_PCR_CLKSSTAT_DEFAULT,
MCBSP_PCR_DXSTAT_DEFAULT,
MCBSP_PCR_FSXP_DEFAULT,
MCBSP_PCR_FSRP_DEFAULT,
MCBSP_PCR_CLKXP_DEFAULT,
MCBSP_PCR_CLKRP_DEFAULT
)
};
hMcbsp0 = MCBSP_open(MCBSP_DEV0, MCBSP_OPEN_RESET);
MCBSP_config(hMcbsp0, &mcbspCfg0);
}
void
```

```
set_interrupts_dmax(void)
{
IRQ_nmiEnable();
IRQ_globalEnable();
IRQ_disable(IRQ_EVT_DMAINT2);
IRQ_clear(IRQ_EVT_DMAINT2);
IRQ_enable(IRQ_EVT_DMAINT2);
return;
}
void
init_mcbsp0_slave(void)
{
MCBSP_Config mcbspCfg0 = {
MCBSP_SPCR_RMK(
        MCBSP_SPCR_FREE_DEFAULT,
        MCBSP_SPCR_SOFT_DEFAULT,
MCBSP_SPCR_FRST_DEFAULT,
MCBSP_SPCR_GRST_DEFAULT,
MCBSP_SPCR_XINTM_DEFAULT,
MCBSP_SPCR_XSYNCERR_DEFAULT,
MCBSP_SPCR_XRST_DEFAULT,
MCBSP_SPCR_DLB_DEFAULT,
MCBSP_SPCR_RJUST_DEFAULT,
MCBSP_SPCR_CLKSTP_DEFAULT,
MCBSP_SPCR_DXENA_DEFAULT,
MCBSP_SPCR_RINTM_DEFAULT,
MCBSP_SPCR_RSYNCERR_DEFAULT,
MCBSP_SPCR_RRST_DEFAULT
),
MCBSP_SPCR_RMK(
MCBSP_SPCR_FRST_DEFAULT,
MCBSP_SPCR_GRST_DEFAULT,
MCBSP_SPCR_XINTM_DEFAULT,
MCBSP_SPCR_XSYNCERR_DEFAULT,
MCBSP_SPCR_XRST_DEFAULT,
MCBSP_SPCR_DLB_DEFAULT,
MCBSP_SPCR_RJUST_DEFAULT,
MCBSP_SPCR_CLKSTP_DEFAULT,
MCBSP_SPCR_RINTM_DEFAULT,
MCBSP_SPCR_RSYNCERR_DEFAULT,
MCBSP_SPCR_RRST_DEFAULT
),
```

```
MCBSP_RCR_RMK(
MCBSP_RCR_RPHASE_SINGLE,
MCBSP_RCR_RFRLEN2_DEFAULT,
MCBSP_RCR_RWDLEN2_DEFAULT,
MCBSP_RCR_RCOMPAND_DEFAULT,
MCBSP_RCR_RFIG_DEFAULT,
MCBSP_RCR_RDATDLY_1BIT,
MCBSP_RCR_RFRLEN1_DEFAULT,
MCBSP_RCR_RWDLEN1_DEFAULT
),
MCBSP_XCR_RMK(
MCBSP_XCR_XPHASE_DEFAULT,
MCBSP_XCR_XFRLEN2_DEFAULT,
MCBSP_XCR_XWDLEN2_DEFAULT,
MCBSP_XCR_XCOMPAND_DEFAULT,
MCBSP_XCR_XFIG_DEFAULT,
MCBSP_XCR_XDATDLY_DEFAULT,
MCBSP_XCR_XFRLEN1_DEFAULT,
MCBSP_XCR_XWDLEN1_DEFAULT
),
MCBSP_SRGR_RMK(
MCBSP_SRGR_GSYNC_DEFAULT,
MCBSP_SRGR_CLKSP_DEFAULT,
MCBSP_SRGR_CLKSM_DEFAULT,
MCBSP_SRGR_FSGM_DEFAULT,
MCBSP_SRGR_FPER_DEFAULT,
MCBSP_SRGR_FWID_DEFAULT,
MCBSP_SRGR_CLKGDV_DEFAULT
),
MCBSP_MCR_RMK(
MCBSP_MCR_XPBBLK_DEFAULT,
MCBSP_MCR_XPABLK_DEFAULT,
MCBSP_MCR_XMCM_DEFAULT,
MCBSP_MCR_RPBBLK_DEFAULT,
MCBSP_MCR_RPABLK_DEFAULT,
MCBSP_MCR_RMCM_DEFAULT
),
MCBSP_RCER_RMK(
MCBSP_RCER_RCEB_DEFAULT,
MCBSP_RCER_RCEA_DEFAULT
),
MCBSP_XCER_RMK(
```

```
MCBSP_XCER_XCEB_DEFAULT,
MCBSP_XCER_XCEA_DEFAULT
),
MCBSP_PCR_RMK(
MCBSP_PCR_XIOEN_DEFAULT,
MCBSP_PCR_RIOEN_DEFAULT,
MCBSP_PCR_FSXM_DEFAULT,
MCBSP_PCR_FSRM_DEFAULT,
MCBSP_PCR_CLKXM_DEFAULT,
MCBSP_PCR_CLKRM_DEFAULT,
MCBSP_PCR_CLKSSTAT_DEFAULT,
MCBSP_PCR_DXSTAT_DEFAULT,
MCBSP_PCR_FSXP_DEFAULT,
MCBSP_PCR_FSRP_DEFAULT,
MCBSP_PCR_CLKXP_DEFAULT,
MCBSP_PCR_CLKRP_DEFAULT
)
};
hMcbsp0 = MCBSP_open(MCBSP_DEV0, MCBSP_OPEN_RESET);
MCBSP_config(hMcbsp0, &mcbspCfg0);
}
void
set_interrupts_dma(void)
{
IRQ_nmiEnable();
IRQ_globalEnable();
IRQ_disable(IRQ_EVT_DMAINT1);
IRQ_clear(IRQ_EVT_DMAINT1);
IRQ_enable(IRQ_EVT_DMAINT1);
return;
}
interrupt void
c_int09(void)
{
recv0_done = TRUE;
return;
}
interrupt void
c_int11(void)
{
xmit0_done = TRUE;
return;
```

```
}
```

B.2 Serial communication of TMS320C32

```
void enable_dsp_port()
{
*SER0_GC = 0x023c0064;
*SER0_XPC = 0x00000111; /* Set Tx port control register (CLKXFUNC =
DXFUNC = FSXFUNC = 1) */
*SER0_RPC = 0x00000111; /* Set Rx port control register (CLKRFUNC =
DRFUNC = FSRFUNC = 1) */
*SER0_TC = 0x0000000D; /* Set timer control register (XGO = 1, XC/P = 1,
XCLKSRC = 1) */
*SER0_CNT = 0x00000000; /* Clear timer counter register */
*SER0_PER = 0x00000002; /* Set timer period register (f = 12.5MHz / (2 x
period register)) */
*SER0_TC |= 0x00000002; /* Start transmit counter (XHLD = 1) */
*SER0_GC |= 0x0c000000; /* Start serial port (XRESET = RRESET = 1) */
    enable_interrupt(RINT0);
}
```

Appendix C

C Code of Communication Between TMS320C6701 EVM and ADS8364 EVM

C.1 Communication of TMS320C6701 EVM

```
#define CHIP_6701      1
#include <c6x.h>
#include "c6x11dsk.h"
#include "ADS8364.h"
#include <csl.h>
#include <csl_legacy.h>
#include <csl_emif.h>
#include <csl_timer.h>
#define SP0_SRGR_V 0x00000000
#define SP0_SPCR_V 0x00000000
#define SP0_PCR_V 0x00000000
#define SP1_SRGR_V 0x00000000
#define SP1_SPCR_V 0x00000000
#define SP1_PCR_V 0x00003F0F
void init_GPIO(void)
{
    * (UINT32 *) McBSP0_SRGR = (UINT32) SP0_SRGR_V;
    * (UINT32 *) McBSP0_SPCR = (UINT32) SP0_SPCR_V;
    * (UINT32 *) McBSP0_PCR = (UINT32) SP0_PCR_V;
    * (UINT32 *) McBSP1_SRGR = (UINT32) SP1_SRGR_V;
    * (UINT32 *) McBSP1_SPCR = (UINT32) SP1_SPCR_V;
    * (UINT32 *) McBSP1_PCR = (UINT32) SP1_PCR_V;
```

```
        return;
}
void dsk_init()
{
unsigned int i;
CSR = 0x100;
IER = 0x2;
ICR = 0xffff;
*(unsigned volatile int *)EMIF_GCR = 0x3300;
*(unsigned volatile int *)EMIF_CE0 = 0x30;
*(unsigned volatile int *)EMIF_CE1 = CE1_32;
*(unsigned volatile int *)EMIF_SDCTRL = 0x57116000;
*(unsigned volatile int *)EMIF_SDRP = 0x61a;
*(unsigned volatile int *)EMIF_SDEXT = 0x54529;
*(unsigned volatile int *)EMIF_CE2 = 0x81b2d82a;
ISTP = (unsigned int)&intvectable;
*(unsigned volatile int *)TIMER0_CTRL |= 0x303;
*(unsigned volatile int *)TIMER0_PRD = CONVCLK_PERIOD;
*(unsigned volatile int *)TIMER0_COUNT = 0;
TIMER_Run(TIMER0_CTRL);
*(unsigned volatile int *)INT_POLARITY |= 1;
USER_LEDS_On;
for(i=2417795; i; i–);
USER_LEDS_Off;
}
```

C.2 Communication of ADS8364 EVM

```
#include <c6x.h>
#include "c6701.h"
#include "ADS8364.h"
#define ADD_Mode 0
unsigned int adc_ready;
unsigned int adc_readNdx;
unsigned int ADC_address[6];
unsigned int A0_temp[ADC_SAMPLE_SIZE];
unsigned int A1_temp[ADC_SAMPLE_SIZE];
unsigned int B0_temp[ADC_SAMPLE_SIZE];
unsigned int B1_temp[ADC_SAMPLE_SIZE];
unsigned int C0_temp[ADC_SAMPLE_SIZE];
unsigned int C1_temp[ADC_SAMPLE_SIZE];
```

```
int main()
{
    dsk_init();
    software_init_ADC();
    enable_interrupt(INT_EXT4);
software_cnvt(0, 1);
    while(1) {
        if(adc_ready) {
    software_cnvt(CMD0, 0);
    delay(0);
        adc_ready = 0;
        }
    }
}
void software_init_ADC()
{
    adc_ready = 0;
            if (ADD_Mode){
        *(unsigned volatile int *) McBSP1_PCR = (CTL_BASE | ADD | INIT_SOFT);}
        else{
    *(unsigned volatile int *) McBSP1_PCR = (CTL_BASE |INIT_SOFT);}
    delay(10);
    software_cfig(CMD1);
adc_readNdx = 0;
adc_ready = 1;
}
void software_cfig(int cmd1)
{
*(unsigned volatile int *)ADC_Base = cmd1;
}
void software_cnvt(int cmd0, int reset)
{
if(reset) {
*(unsigned volatile int *)ADC_Base = *(unsigned volatile int *)GBL_Reset;

}
*(unsigned volatile int *)ADC_Base = cmd0;
}
void software_read(int a0, int a1, int b0, int b1, int c0, int c1,
                        int cycle, int fifo)
{
if (a0){
```

```
        *(unsigned volatile int *)ADC_Base = A0 | CR1;
        if (ADD_Mode){ADC_address[0] = (*(unsigned volatile int *)ADC_Base);}
        A0_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);}
    if (a1){
        *(unsigned volatile int *)ADC_Base = A1 | CR1;
        if (ADD_Mode){ADC_address[1] = (*(unsigned volatile int *)ADC_Base);}
        A1_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);}
    if (b0){
        *(unsigned volatile int *)ADC_Base = B0 | CR1;
        if (ADD_Mode){ADC_address[2] = (*(unsigned volatile int *)ADC_Base);}
        B0_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);}
    if (b1){
        *(unsigned volatile int *)ADC_Base = B1 | CR1;
        if (ADD_Mode){ADC_address[3] = (*(unsigned volatile int *)ADC_Base);}
        B1_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);}
    if (c0){
        *(unsigned volatile int *)ADC_Base = C0 | CR1;
        if (ADD_Mode){ADC_address[4] = (*(unsigned volatile int *)ADC_Base);}
        C0_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);}
    if (c1){
        *(unsigned volatile int *)ADC_Base = C1 | CR1;
        if (ADD_Mode){ADC_address[5] = (*(unsigned volatile int *)ADC_Base);}
        C1_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);}
    if (cycle){
        if (ADD_Mode){ADC_address[0] = (*(unsigned volatile int *)ADC_Base);}
        A0_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);
        if (ADD_Mode){ADC_address[1] = (*(unsigned volatile int *)ADC_Base);}
        A1_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);
        if (ADD_Mode){ADC_address[2] = (*(unsigned volatile int *)ADC_Base);}
        B0_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);
        if (ADD_Mode){ADC_address[3] = (*(unsigned volatile int *)ADC_Base);}
        B1_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);
        if (ADD_Mode){ADC_address[4] = (*(unsigned volatile int *)ADC_Base);}
        C0_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);
        if (ADD_Mode){ADC_address[5] = (*(unsigned volatile int *)ADC_Base);}
        C1_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);}
    if (fifo){
        if (ADD_Mode){ADC_address[0] = (*(unsigned volatile int *)ADC_Base);}
        A0_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);
        if (ADD_Mode){ADC_address[1] = (*(unsigned volatile int *)ADC_Base);}
        A1_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);
        if (ADD_Mode){ADC_address[2] = (*(unsigned volatile int *)ADC_Base);}
```

```
    B0_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);
    if (ADD_Mode){ADC_address[3] = (*(unsigned volatile int *)ADC_Base);}
    B1_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);
    if (ADD_Mode){ADC_address[4] = (*(unsigned volatile int *)ADC_Base);}
    C0_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);
    if (ADD_Mode){ADC_address[5] = (*(unsigned volatile int *)ADC_Base);}
    C1_temp[adc_readNdx] = (*(unsigned volatile int *)ADC_Base);}
    adc_readNdx = adc_readNdx+1;
    adc_ready = 1;
}
void enable_interrupt(unsigned int mask)
{
ICR = (IFR | mask);
IER |= (mask | 0x2);
CSR = 0x101;
}
void disable_interrupt(unsigned int mask)
{
ICR = (IFR | mask);
IER &= ~mask;
}
interrupt void ext4_isr()
{
ICR = (IFR | INT_EXT4);
    software_read(0,0,0,0,0,0,1,0);
ICR = (IFR | INT_EXT4);
IER |= (INT_EXT4 | 0x2);
    if(adc_readNdx == ADC_SAMPLE_SIZE)
{
    adc_readNdx = 0;
    software_cnvt(0, 1);
    software_cfig(CMD1);
    }
}
interrupt void dummy_isr()
{
    USER_LEDS_On;
    while(1);
}
void delay(int time)
{
int i;
```

```
for (i=0; i<time; i++);
}
```

Appendix D

C Code of the PID and H_∞ Controller

D.1 PID controller

```
struct _pid {
int *pv;
int *sp;
float integral;
float pgain;
float igain;
float dgain;
int deadband;
int last_error;
};
void pid_init(a, pv, sp)
struct _pid *a;
int *pv, *sp;
{
a->pv = pv;
a->sp = sp;
}
void pid_tune(a, p_gain, i_gain, d_gain, dead_band)
struct _pid *a;
float p_gain, i_gain, d_gain;
int dead_band;
{
a->pgain = p_gain;
a->igain = i_gain;
```

```
a->dgain = d_gain;
a->deadband = dead_band;
a->integral=0.0;
a->last_error=0;
}
void get_gains(a, p_gain, i_gain, d_gain, dead_band)
struct _pid *a;
float *p_gain, *i_gain, *d_gain;
int *dead_band;
{
*p_gain = a->pgain;
*i_gain = a->igain;
*d_gain = a->dgain;
*dead_band = a->deadband;
}
void pid_setinteg(a,new_integ)
struct _pid *a;
float new_integ;
{
a->integral=new_integ;
a->last_error=0;
}
pid_bumpless
void pid_bumpless(a)
struct _pid *a;
{
a->last_error = *(a->sp) - *(a->pv);
}
#include "control.h"
main() {
struct _pid PID;
int process_variable, set_point;
pid_init(&PID, &process_variable, &set_point);
pid_tune(&PID, 4.3, 0.2, 0.1, 2);
set_point = 500;
pid_setinteg(&PID,30.0);
process_varialbe = read_temp();
pid_bumpless(&PID);
for(;;) {
process_variable = read_temp();
output( pid_calc(&PID) );
wait(1.0);
```

```
}
}
float pid_calc(a)
struct _pid *a;
{
int err;
float pterm, dterm, result, ferror;
err = *(a->sp) - *(a->pv);
if (abs(err) > a->deadband) {
ferror = (float) err;
pterm = a->pgain * ferror;
if (pterm > 100 || pterm < -100)
a->integral = 0.0;
else {
a->integral += a->igain * ferror;
if (a->integral > 100.0) a->integral=100.0;
else if (a->integral < 0.0) a->integral=0.0;
}
dterm = (err - a->last_error) * a->dgain;
result = pterm + a->integral + dterm;
}
else result = a->integral;
a->last_error=err;
return (result > 100.0 ? 100.0 : (result < 0.0 ? 0.0 : result));
}
```

D.2 H_∞ Controller

```
void hifinity(float x_k)
{
float y_k=0,y_(k-1)=0,y_(k-2)=0,y_(k-3)=0,y_(k-4)=0,y_(k-5)=0,y_(k-6)=0;
float x_(k-1)=0,x_(k-2)=0,x_(k-3)=0,x_(k-4)=0,x_(k-5)=0;
y_k=0.002606*y_(k-1)-1.598e-004*y_(k-2)+5.865e-017*y_(k-3)-1.387e-034*y_(k-
4)-3.49e-054*y_(k-5)+1.739e-073*y_(k-6)+0.0274*x_k-7.269e-005*x_(k-1)+4.38e-006*x_(k-
2)+2.51e-017*x_(k-3)-4.792e-0.34*x_(k-4)-1.106e-051*x_(k-5);
y_(k-6)=y_(k-5);
y_(k-5)=y_(k-4);
y_(k-4)=y_(k-3);
y_(k-3)=y_(k-2);
y_(k-2)=y_(k-1);
y_(k-1)=y_k;
```

```
x_(k-5)=x_(k-4);
x_(k-4)=x_(k-3);
x_(k-3)=x_(k-2);
x_(k-2)=x_(k-1);
x_(k-1)=x_k;
}
```