An Investigation of Performance Limitations in Active Noise Reduction Headsets

by

Daniel C. Clatterbuck

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

William R. Saunders, Chair
Harry H. Robertshaw
William T. Baumann

April 27, 1998
Blacksburg, Virginia

Keywords: ANR, ANC, digital, headset, supra-aural

An Investigation of Performance Limitations in Active Noise Reduction Headsets

by
Daniel C. Clatterbuck
Dr. William R. Saunders, Chair
Department of Mechanical Engineering

(ABSTRACT)

Closed-loop performance of an active noise reduction (ANR) headset is limited by phase lag. Speaker dynamics, control hardware, and acoustic wave propagation from the control speaker to the error microphone all contribute to this phase lag. Understanding these sources of phase lag and their relative effects on performance allows for better design of an analog or digital ANR headset.

This thesis demonstrates that the three most significant sources of phase lag in a digital ANR headset are the dynamics of the control speaker's diaphragm, the anti-aliasing filter, and the smoothing filter. Additionally, it is demonstrated that the acoustic wave propagation from the control speaker to the error microphone is not a major contributor of phase lag. Based on these results, it was determined that attention should be focused on the anti-aliasing and smoothing filters when attempting to minimize phase lag and improve a digital ANR system's performance.

A design procedure was developed to calculate filter responses that contributed a minimal amount of phase lag to a headset. Using this minimal phase filter design procedure, a digital ANR headset was successfully built and tested. Initial testing revealed that the anti-aliasing filter was not as vital to performance as the smoothing filter. Further testing indicated that the anti-aliasing requirements could be effectively met through the use of only a smoothing filter. Therefore, in order to minimize the phase lag of a digital ANR headset, a smoothing filter may be utilized in the absence of an anti-aliasing filter for some applications.

ACKNOWLEDGEMENTS

Gretchen Ponzi	Thank you for being my crutch and inspiration over the past 7 years. Words can not describe what you've meant to me. The future looks so bright. E.L.U.
Mom and Dad	For the last 24 years, my parents have put myself and my brother ahead of themselves. Because of their hard work and dedication, we both have had opportunities beyond our wildest dreams. Thank you for giving us all we ever needed to be successful students, professionals, and individuals.
Will Saunders	I would like to thank Will for being my advisor, friend, and mentor. School can sometimes be a stressful place to try and exist, but this was not the case under Will. He let me develop on my own, providing slight nudges only when needed. Thank you for allowing me to mature academically in an independent, relaxed, and fun atmosphere. I'll always remember losing a game we had no business losing!
Harry Robertshaw	H^2 served as my boss for two semesters as the professor in charge of ME4006. In addition, he taught one of my graduate level control theory classes. It was the only 8:00 am class I never minded getting up to attend. Thank you for always having a smile on your face and time to chat. I will never forget Hell's Nostril or jumping over a camp fire!
Bill Baumann	As an ME student, you always hear horror stories about EE professors and classes. I would like to thank Bill for dispelling those myths. He was not a three headed monster that spit fire; rather he was one of the best professors I've ever had. His multivariable control class was difficult, but along with the hard work came the reward of knowledge and we all know that knowledge is king. Thanks for teaching me something.
Bob West	I owe Bob a large thank you for taking the time to bring me up to speed on the laser technology that he is developing. Without his knowledge and patience, the speaker diaphragm velocity measurements would have been impossible.
Chris Fannin Pete Pickett Mike Vaudrey	Thanks for sharing the office and helping me when needed. Many of the results in my thesis are due in large part to the minds of these three. It's been fun.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

A(s)	audio amplifier transfer function
AA(s)	anti-aliasing filter transfer function
AC(s)	acoustic transmission transfer function
ANC	active noise control
ANR	active noise reduction
a, a_{out}	speaker radius
a_{in}	speaker radius to inner edge of surround
a_o	radius of spherical source
C	electrical capacitance
C(s)	control transfer function
c	speed of sound
D	diaphragm displacement
e	input signal to anti-aliasing filter
f	frequency, Hz
f_{nyq}	Nyquist frequency, Hz
f_s	sampling frequency, Hz
k	free wavenumber
M(s)	microphone transfer function
M_{AA}	slope of minimal phase anti-aliasing filter
Mc_1, Mc_2	modal contribution percentage
n	bits, integer
O	point of observation in free space
P	acoustic pressure
Q	acoustic source strength
R	electrical resistance
r	vector from diaphragm center to point of observation
r'	vector from point source to point of observation
S(s)	speaker transfer function
SM(s)	smoothing filter transfer function

SPL	sound pressure level
s	laplace variable
T	sampling period
t	time
U, U_{in}, U_{out}	surface normal velocity amplitude
u	output signal from smoothing filter
u_{held}	piece-wise continuous signal output from D/A converter
u^*	impulse modulated signal
$\vec{u}$	particle velocity
V	voltage
V_{amp}	voltage output from audio amplifier
V_i	disturbance voltage at input to anti-aliasing filter
V_{mic}	voltage output from microphone
V_o	voltage output from smoothing filter
V_{spk}	minimum voltage capable of powering control speaker
x	highest frequency in range where control performance is designed
x_1, x_2	frequencies used to define slope
y_1, y_2	magnitudes used to define slope
Z	electrical impedance
zoh	zero order hold
ε,ε_{ad}	resolution
ϕ	phase lag
ϕ_{cone}, ϕ_{in}, ϕ_{out}	phase of diaphragm
θ	angle between z-axis and (r)
ρ	density
σ	vector from diaphragm center to point source
ω	angular frequency, rad/s
ω_i	angular frequency at which imposters are produced, rad/s
ω_{nz}, ω_{np}	natural frequency, rad/s

ω_o	angular frequency of input signal, rad/s
ω_s	sample frequency, rad/s
ψ	angle between y-axis and (σ)
ξ_z, ξ_p	damping ratio

1 CHAPTER: INTRODUCTION

1.1 Introduction and objectives

Active noise control (ANC) uses the principle of superposition to control sound. Every noise has an anti-noise which is equivalent in magnitude and 180° out of phase. The introduction of the anti-noise by means of a controlled source, therefore, causes destructive interference, negating the offensive noise. In this thesis, ANC is explored in relation to the design and performance of a digital active noise reduction (ANR) headset. Specifically, the system studied is one which places a microphone and a speaker in close proximity to a human's ear. Coupled via a digital feedback system, the speaker is used to drive the sound pressure level at the microphone to zero.

Analog ANR headsets have already been designed and are available on the market today. However, the versatility available with a digital control law makes exploration of a digital headset desirable. In order to design the digital equivalence of an analog headset, the basic principles of operation must first be understood. Understanding the dynamics of the system, especially the mechanisms which introduce phase lag, is imperative because the phase lag limits the performance of the control.

In this thesis, the author wishes to provide the background information necessary to begin designing a digital ANR headset. More specifically, this thesis attempts to answer the following questions:

- How does the acoustic propagation delay from control speaker to error microphone impact overall digital ANR headset performance?
- How do the control speaker's dynamics impact overall digital ANR headset performance?
- How do each of the components in a sampled data system behave and how do they impact overall digital ANR headset performance?

- What are the parameters required to design an anti-aliasing filter and a smoothing filter that contribute as little phase lag as possible to a digital ANR headset?
- For a digital ANR headset, is an anti-aliasing filter necessary?
- Can a digital ANR headset be implemented successfully?

1.2 Motivation

With the advent of high-speed digital signal processing techniques, there seems to be no limit to where ANC can be applied. From airport runways, to automobiles, to factories there is an endless supply of noise that needs to be controlled. However, in order for ANC technology to move forward, a firm foundation must first be created.

Although the concept of ANC has been around since the early 1930's, practical products were not a reality until the late 1970's. In the past twenty years, the growth of this technology has been phenomenal and it's rapid emergence as a powerful tool has created a gap in the literature. Tremendous strides have been made in the marketplace to advance and improve the technology, but little has been documented. This thesis is an attempt to begin filling the void in the literature.

Through examination of acoustics, speaker dynamics, and digital control techniques, this thesis attempts to provide some of the fundamental knowledge needed to design a digital ANR headset. Understanding the fundamentals of how these three areas interact and affect the overall performance of the system will allow for future improvements to the basic components of such a headset. Such improvements will eventually lead to a much improved digital design that will replace the present analog designs.

1.3 Summary of findings

The basic equations governing the acoustic propagation in a headset were formulated and analyzed. It was found that modeling the control speaker diaphragm as a rigid piston was

not a valid assumption across the frequency range of interest, 0-6 kHz. Modeling the diaphragm's first "breakup mode" in addition to the rigid piston mode, however, proved to be accurate in predicting the acoustic propagation delay. Experimental data representing the transfer function from speaker input voltage to error microphone output voltage was measured in an effort to support the acoustic formulations. Mathematically removing the calculated acoustics from this data resulted in data that was representative of speaker dynamics. This provided proof that the acoustic formulations were accurate. Based on the acoustic formulations, it was concluded that the acoustic propagation delay incurred in a digital ANR headset is negligible in comparison to the phase lag induced by the higher-order speaker dynamics. For a speaker-microphone separation of 1 inch and a diaphragm radius of 1 inch, the acoustic phase lag was less than 10° at 1000 Hz, while the speaker dynamics created over 180° of phase lag by the same freqyency. Therefore, the dynamic response of the speaker diaphragm has a more profound effect than the acoustic wave propagation on the overall system performance.

After exploring the basic properties and functions of each of the components in a sampled data system, it was determined that judicious design of the anti-aliasing filter and smoothing filter would be necessary to implement a digital controller. The phase lag contributed by these filters limits the performance of the digital system more than any other single component. Minimizing the phase lag due to these filters allows greater flexibility in the controller design.

Once it was demonstrated that the anti-aliasing and smoothing filters were optimal candidates to reduce the phase lag in a digital ANR headset, the basic parameters necessary to design a set of minimal phase filters were formulated. Aliasing was eliminated by a minimal phase filter described by a 0 dB pass-band over the region of desired control and a roll-off rate that reduced the disturbance spectrum to the level of the resolution of the A/D converter by a certain frequency. That frequency was defined as the sampling frequency minus the highest frequency in the desired range of control. A minimal phase smoothing filter was developed and described by a 0 dB pass-band over the region of desired control and a roll-off rate that reduced the D/A imposters below a

certain voltage level defined by the control speaker. The voltage level was determined by the minimum voltage that would power the speaker and create audible sound. Defining the frequency responses using these criteria lead to the lowest order filters, minimal phase filters, that would eliminate aliasing and imposters.

A prototype single-channel headset was constructed and a digital controller was implemented to test the design procedure. Tests were done using only a minimal phase anti-aliasing filter, only a minimal phase smoothing filter, and using the combination of both filters. The tests revealed that the anti-aliasing filter was not necessary depending on the disturbance noise spectrum. In the case of a low pass disturbance with high frequencies of decreased magnitude, anti-aliasing requirements could be effctively met through the use of only a smoothing filter. Hence, an optimal design for some applications may only include a smoothing filter.

Finally, the tests proved the plausibility of designing a digital headset with only a minimal phase smoothing filter. A controller and minimal phase smoothing filter were implemented and subjected to a pink noise disturbance field. The controller was able to achieve in excess of 10dB reduction in a narrow band around 700Hz. Although the design was not optimal, it demonstrated that control of narrow-band noise is possible. When an elliptic smoothing filter was tested subject to the same noise field, spill-over in the closed-loop response increased. The minimal phase smoothing filter outperformed the elliptic smoothing filter for the particular plant studied.

1.4 Presentation of thesis

Following this introduction, there is a brief discussion of some of the literature pertaining to ANR headsets. In particular, literature is examined in three main areas: speaker dynamics, acoustics, and control theory. Each of these areas comprises a major part of an ANR headset and therefore, each area must be thoroughly understood. After this

background literature, some of the products that have been developed are discussed to illustrate the seemingly endless potential of ANR technology.

Chapter 3 examines the acoustics of an ANR headset. A qualitative discussion of the control speaker's near and far field, as well as the direct and reverberant field inside the headset, is first presented to provide background information. Next, the acoustic propagation due to a rigid piston approximation of the control speaker diaphragm is formulated and solved for the on-axis case. In an attempt to verify the acoustic predictions, the dynamics of the speaker diaphragm are studied and utilized. After demonstrating that the rigid piston approximation is not sufficient, the acoustic model is improved by including the first breakup mode of the diaphragm. Finally, the implications of the acoustic field calculations and speaker's dynamic response on ANR headset performance are discussed.

The components required for the control system in a digital ANR headset are examined in Chapter 4. The principles and properties of anti-aliasing filters, A/D converters, control filters, D/A converters, and smoothing filters are all discussed as they pertain to the phase lag in a digital controller. After presenting this theory, some experiments are presented to verify the theory and simulations are written to help analyze a digital control system. Lastly, a general method for designing an anti-aliasing filter and a smoothing filter that introduce as little phase as possible is presented.

Chapter 5 utilizes the results of the previous chapters to design a digital ANR headset. Testing is carried out on a single-channel prototype headset with only a minimal phase anti-aliasing filter, with only a minimal phase smoothing filter, and with a combination of both filters. Finally, the performance of a minimal phase smoothing filter is compared to the performance of an elliptic smoothing filter. The designs and their performance are evaluated on the basis of spill-over, acoustic masking, and acoustic bandpower. These tests demonstrate that a digital ANR headset is practical and that the phase lag may be minimized by using only a smoothing filter.

The final chapter summarizes the major developments of the thesis and presents some of the future work that must be done in this area before a full prototype is designed. Through improvement of the control speaker's dynamics, the plant response can be improved and the system's phase lag can be reduced. Reduction of this phase lag would enhance performance dramatically. Additionally, adaptive control algorithms should be explored as varying disturbance fields will create the need for a highly flexible system.

2 CHAPTER: LITERATURE REVIEW

2.1 Active noise control: an emerging technology

> If a tree falls in the woods and no person is around,
> does the tree create a sound?

Philosophers have long been plagued by such theoretical questions and the absolute answer to them does not appear to be coming within the foreseeable future. Scientists, however, view the world from a different perspective and they have chosen to redefine the question instead of answering it as posed.

> If a tree falls in the woods and a person is around,
> can the sound be made inaudible to the person?

The answer to this question has been answered by the many developments since the birth of ANC technology over sixty years ago. Since its beginning, considerable effort has been devoted to the theoretical and practical development of ANC systems with the major developments coming in the past twenty years. Although ANC has been around for quite some time, the technology is still under development and looking for widespread practical applications.

2.2 Theory

Due to the relative infancy of ANR technology, it is difficult to find many articles that attempt to explore the theory governing all aspects of the system. Instead, the interested reader is left to scour the library for articles on each of the subsystems and then to piece the puzzle together on his own.

Figure 2.1 illustrates the three major components, or subsystems, within an ANR headset. The disturbance noise is measured by a microphone which feeds the signal into a control subsystem. The output of the controller is then sent to the speaker subsystem which

creates an anti-noise. Finally, the anti-noise follows an acoustic transmission path and is summed with the disturbance noise at the microphone.

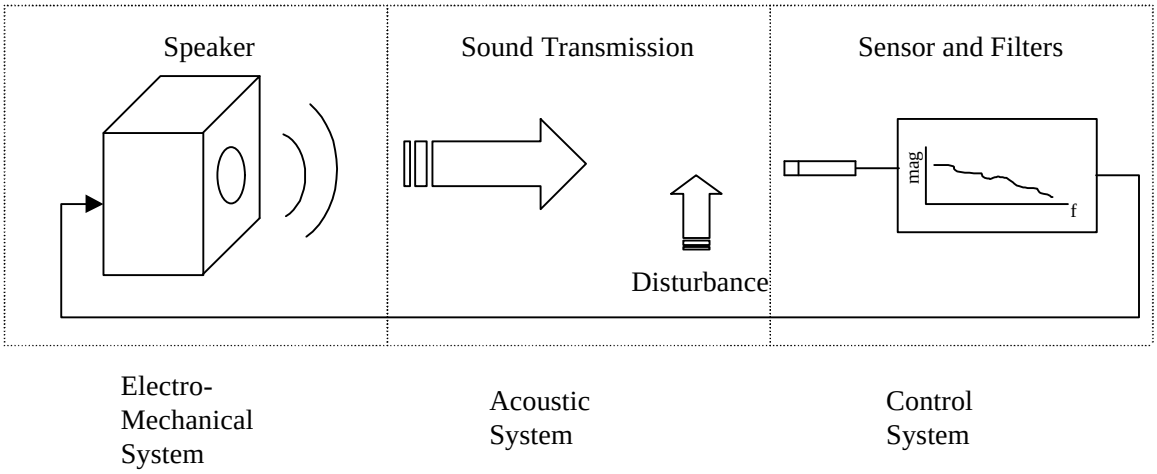

Figure 2.1 Three major sub-systems of an ANR headset.

2.2.1 Speaker theory

In the 1900's, the study of loudspeakers and their characteristics has become quite popular. With the ultimate goal of creating a better speaker, many authors have examined the topics of diaphragm dynamics and modeling of the electro-mechanical properties of speakers. This knowledge has become vital to ANR technology as the system's performance may be limited by the dynamics of the control loudspeaker.

Loudspeaker diaphragms are not rigid bodies, instead they are characterized by multiple modes of vibration. Understanding the motion of the diaphragm for each of these modes is important to understanding the sound that is radiated by a speaker. Finite element (FE) techniques have been applied to the analysis of loudspeaker transducers' mid-frequency diaphragm modes (Binks et al., 1991). For a given drive force, the method calculated the mechanical vibration of the diaphragm. A drawback to performing such a detailed finite element analysis was that values of Young's modulus, density, damping, and Poisson's ratio for all the materials used in the speaker structure were needed. These parameters

are not always known for mass-produced speakers. Performance of the FE technique was evaluated by comparing measured diaphragm velocities with predicted theoretical diaphragm velocities. The FE results adhered to the measured results for low frequencies when few modes were involved in the diaphragm vibration. As higher frequencies were used to drive the diaphragm and higher-order diaphragm modes were developed, however, the modeling technique lost its accuracy.

Nomoto and Suzuki (1982) also applied the FE method to analyzing diaphragm vibrations. They utilized computer-aided analysis to illustrate the motion of the diaphragm for frequencies in the 0-10 kHz range. The results of their analysis suggested that the surround, flexible support around the edge of the speaker, started to vibrate out of phase with the central section of the diaphragm as high frequencies were observed. This change from a rigid structure to a flexible structure signified the first breakup mode of the diaphragm.

In order to overcome the problem of needing manufacturer's data, Gogate and Radcliffe (1992) proposed a laboratory methodology that attempted to identify the parameters needed to model speakers. First, a simple low frequency speaker model was developed which ignored the higher-order dynamics due to coil inductance. In order to evaluate the model, methods were described to measure the coil resistance, electromagnetic coupling factor, and system damping ratio. Experimental diaphragm velocity measurements were then compared to the predicted velocity response using the measured parameters in the speaker model. Below 200 Hz, the measured data and model predictions agreed well, varying by no more than 2dB, however, as higher frequencies were observed the agreement faded. Additionally, as an improvement to loudspeaker performance, a scheme by which cone velocity is fed back was proposed. The velocity sensor required only the use of a resistor and an operational amplifier. Use of high proportional gains on the velocity feedback flattened out the velocity magnitude response and decreased the phase lag as well.

In an attempt to alleviate the complication and time consumption associated with using FE methods to model and predict loudspeaker behavior, work was done to apply an experimental method (Struck, 1989). This modal analysis technique was based upon using a laser transducer to acquire diaphragm velocity response data. Curve-fitting of the velocity data synthesized the transfer function from speaker drive voltage to diaphragm velocity in the s plane. The curve fitting results were compared to a theoretical analysis in which the diaphragm was approximated as a stretched membrane. The natural frequencies derived from the curve-fitting technique agreed favorably to the predicted natural frequencies of the theoretical development. The first three natural frequencies of the curve-fit were all within 20 Hz of the predicted natural frequencies. This type of modal analysis demonstrated that the diaphragm responded in multiple mode shapes that could be approximated by the mode shapes of a stretched membrane.

2.2.2 Acoustic theory

Papers discussing both the vibrations of the speaker diaphragm and the resulting sound field in detail are a rarity. Generally, authors tend to tackle one of the two problems because both are detailed and complicated individually. Kaizer and Leeuwstein (1988) titled their paper "The calculation of sound radiation of a non-rigid loudspeaker diaphragm using the finite element method." However, they restricted themselves to discussing the calculation of the resulting sound radiation after the mechanical vibration of the diaphragm was known. Their formulation involved solving the Helmholtz equation and subsequently using Green's functions to develop a FE model of the radiated sound field. Complicated methods, such as this, can be computationally expensive as they use a FE model of the speaker diaphragm and another FE model of the resulting sound field.

A less computationally expensive method can be applied to formulating the resulting sound field if the dynamic response of the speaker is modeled by assuming the diaphragm is a rigid piston. Results of this approximation are accurate at low frequencies where no

diaphragm break-up modes are observed, but as higher frequencies of diaphragm vibration occur, this model losses its accuracy. On-axis sound pressure level calculations resulting from the model are not complicated and appear in most introductory acoustics books (Coppens et al., 1982). The theoretical development will be discussed in greater detail in Chapter 3.

Modeling the speaker diaphragm as a rigid piston simplifies the acoustic formulations for the on-axis scenario, but it still results in an unwieldy mathematical expression for the off-axis, near field case. A double integral results from the formulation and its solution is not demonstrated in even the most thorough books on acoustics. Pierce (1989) , Beranek (1954), and Rayleigh (1929) all analyze the acoustic radiation from an infinitely baffled piston radiator, but they don't discuss the off-axis case. They examine the sound field on and very near the axis of symmetry in the near field and move on to analyze the off-axis case in the far field.

Once the acoustic propagation is modeled properly, control techniques must be applied to reduce sound levels. Mazzola (1993) presented a discussion on the theory of sound absorption through active means. In his work, he examined the case where a transducer was used to absorb incident acoustic energy. A rigid, fixed plate overlaid on one side with a piezoelectric material formed the transducer. The surface motion of the transducer could be controlled in such a way, that it moved only when a plane wave was incident upon it. In this fashion, the transducer could be used to absorb completely the incident wave.

Elliot and Nelson (1992) offer a thorough examination of ANC techniques. Their book covers all facits of noise control providing a very useful introductory source of information on ANC. After developing some acoustic theory, they investigate single-channel feedforward and single-channel feedback control of sound. The authors then discuss 'zones of quiet' and their impact on global control of sound fields. Finally, local control of sound fields is explored.

2.2.3 ANR control techniques

There are many different configurations that may be used to implement an ANR headset. Control implementations include analog, digital, and adaptive digital feedback systems. In addition to these feedback methods, headsets may include an adaptive feed-forward control technique to enhance performance. All of these different techniques have their own advantages and disadvantages. To explore these techniques, a few papers have been written that document the characteristics of various types of control in ANC systems.

In the digital domain, a system identification approach (Allie et al., 1988) was theorized using adaptive digital signal processing. The authors presented the theory behind generating a sound canceling wave by passing an input signal through a computer based model that accurately represented the response of the acoustic plant. An advantage to this implementation was that with a proper model of the plant, the system could respond instantaneously to changes in the input signal caused by the disturbance. A disadvantage was that an accurate model of the plant was required. In actual field use, this technique was applied to rotating machinery, such as large fans, and achieved 30 dB reductions on narrow-band tonal sounds and 15 dB on broadband continuous spectra associated with random signals.

In a separate paper, Allie et al. (1988) discussed the hardware and software considerations for active noise controllers. The authors examined such topics as processor speed and sampling rate. A high sample rate is required to satisfy the Nyquist criterion if an anti-aliasing filter is not used and the high speed processing is required so the system may complete a cycle of calculations before input noise arrives at the loudspeaker. The authors also discussed the size of the model of the plant and its implications for the processor. In order to accurately model the amplitude and phase response of acoustical systems, a model must be relatively large and the computational time associated with processing a large model decreases the resolution of the system. Efficient coding can help diminish the effect of a large model by minimizing the necessary computation time. This minimization also allows for more weights to be used

in an adaptive algorithm. It is only through all of these considerations, and more, that a robust flexible digital control system can be implemented in a headset.

As a further development of the theory behind digital ANC systems, Leitch and Tokhi (1991) presented a general design method for systems using a primary and secondary source. In their formulation, a fixed control filter was discussed. This fixed filter system provided significant amounts of cancellation over broad frequency ranges if three conditions were met. First, the characteristics of the components in the system had to be time invariant. Second, the geometric arrangement of the system had to remain constant. Finally, the acoustic response of the medium had to stay linear and acoustic reflections had to be minimized. In practice, meeting all of these conditions simultaneously was very difficult so, the authors also discussed an adaptive control filter implementation which involved two processes. The first process involved identifying the system model through use of a system identification algorithm. The second process consisted of calculating the proper IIR filter based upon the system identification and design objectives. In their discussion, the authors briefly mentioned the fact that they used an anti-aliasing filter and because of this the increased system phase lag had to be compensated for by the controller. However, they did not do any testing in the absence of an anti-aliasing filter.

2.3 ANR applications

Having explored some of the theoretical background applied to a ANC systems, it is interesting to look at some of the products that have been developed throughout history. One of the earliest applications of using electronics to control physical systems was seen in ANC techniques. As early as the 1930's, scientists were investigating the possibilities of using active techniques in place of passive techniques to control sound. Paul Leug first filed for a patent in 1933 in Germany and later in 1936 he received US Patent No. 2,043,416. Leug proposed the active control of sound using a secondary source to create a sound wave out of phase with that due to the primary acoustic disturbance. His

approach was targeted at the control of plane sound waves propagating in a duct. Due to the complexities of practical implementation, however, ANC technology was not adapted to earmuffs until 1957 and not until 1978 did a practical system appear for use outside of the laboratory.

Dorey et al. (1978) developed an ANR system for use in aircrew flying helmets. The acoustic field inside the ear defender was detected using a microphone and a control signal was fed back to a telephone. Though the concept was not new, the implementation was a first. The major engineering development of the project was identifying and characterizing suitable microphones and telephones. Testing of speech intelligibility and noise attenuation both were promising for this revolutionary system.

In June of 1984, Bose Corporation applied for U.S. Patent No. 4,455,675. The patent was for a headphone system that achieved noise reduction while faithfully reproducing a music or speech signal. Through an efficient design of the cavity and a flexible cushioning to conform to the irregular shape of the ear, the headset provided high frequency attenuation. Lower frequency attenuation was achieved through active means. The company claimed to achieve a 15 dB attenuation across a frequency range from 100 to 800 Hz.

Active noise reduction techniques were applied to the automobile industry when Costin et al. (1989) attempted to reduce low frequency tire impact noise. A detector microphone, high speed digital controller, amplifier, and an analog smoothing filter were used in conjunction with a headphone to reduce transient-induced road noise in a vehicle interior. The system reportedly achieved a 10-20 dB reduction between 25 and 60 Hz. This system was significant in that it used digital technology in conjunction with analog circuitry. Due to the low frequency nature of the study, however, no anti-aliasing filter was required.

Goodfellow tested the performance of a prototype ANR in-ear hearing protector in 1993. When a conventional circum-aural headset is worn over a protective hood, an acoustic

leak occurs, degrading the hearing protection. Goodfellow suggested that in a noisy environment containing frequencies predominantly below 1 kHz, an in-ear hearing protector may offer an alternative. She measured acoustic performance at the ear and was able to achieve a reduction from 112 dB(A) to below 90 dB(A). The prototypes performed satisfactorily, but it was stated that they needed to be easier to fit and more robust if they were to be used in military applications.

Ambulances, police cars, and fire trucks all have sirens on them. As important as these sirens are to the public outside of the vehicle, the noise created inside the vehicle is not necessary. This noise problem for drivers of such vehicles was studied by Noise Cancellation Technologies Inc. A digital headset was developed that cancelled these siren signals. Casali et al. (1994) performed a review of the headset and reported on the real-ear and acoustical manikin insertion loss. The headset consisted of supra-aural headphones covered with open-cell polyurethane foam and connected by a lightweight metal band. The controller was a separate unit that was powered by 12-V dc power from a cigarette lighter. A digital synchronous feedback control implementation was used in which an adaptive LMS controller circuit captured and synchronized on the rate of change of the noise signal amplitude. The synchronized signal was then fed back to a digital signal generator which determined the harmonic frequency content of the signal to be controlled and created a cancellation signal to be sent to an earphone. The unit performed satisfactorily, achieving 8-22 dB real-ear attenuation at the peak siren frequency of 800 Hz. In addition, the unit only minimally influenced the amplitude of sound at frequencies other than those of the siren, thus offering better comprehension of wanted sounds (speech etc.) than is possible with conventional passive hearing protectors. This digital technology offered the potential for providing selective frequency hearing protection against tonal hazards with the benefits of the comfort of a lightweight supra-aural headset.

2.4 Conclusion

ANC using signal processing is an emerging technology that offers the unique ability to control spectral shape, to allow flexible system operation, and to provide low lifetime cost. Many products have pioneered the marketplace and provided promising results for both customer and inventor. As this new technology advances and infringes upon the areas formerly reliant upon passive techniques, the importance of understanding the detailed theory of operation becomes clear. It is not adequate enough to develop products without developing the theoretical background and literature on the products.

3 CHAPTER: ACOUSTIC/SPEAKER SYSTEM

3.1 Introduction

An ANR headset consists of three main components, or systems: the control speaker, the acoustic transmission path, and the control hardware. The control speaker and the acoustic transmission are intimately tied together, however. The dynamic response of the speaker diaphragm defines the resulting sound field's characteristics. Quantifying the phase lag caused by an acoustic wave propagating from the control speaker to the error microphone, therefore, requires determining the dynamics of the diaphragm.

To quantify this phase lag, some background information was studied and then the acoustic propagation delay was formulated by modeling the speaker diaphragm as a rigid piston. This rigid piston model was demonstrated to be accurate in predicting the acoustic propagation delay at low frequencies, but it was inaccurate as high frequencies were incurred. Consequently, using laser vibrometry, the dynamic response of the speaker diaphragm was studied. Inclusion of the first breakup mode in the modeling of the diaphragm was successful in providing validation of the acoustic propagation model. Using the acoustic formulation resulting from the rigid piston mode and first breakup mode of the diaphragm, conclusions were made as to the relative contributions of the speaker and the acoustic propagation to the overall system's phase lag.

3.2 Sound fields

The acoustic properties of the pressure signal received by a microphone differ depending upon the spatial location of the microphone relative to the signal source. In addition to the spatial location, the boundary conditions surrounding the sound field affect the pressure signal characteristics. Together, boundary conditions and spatial locations determine if the microphone will be in the direct or reverberant field as well as in the near or far field. Therefore, an accurate formulation of the acoustic transmission delay due to

a wave propagating from the control speaker to the error microphone depends on accurately defining the acoustic field of the microphone. By understanding the acoustic field, judicious positioning of the error microphone will lead to enhanced ANR headset performance.

3.2.1 Reverberant vs. direct field - a qualitative discussion

Any sound field that has an enclosure boundary with less than 100% absorptivity will have a sound field that is divided into reverberant and direct regions. The region nearest the source is referred to as the direct field and the region nearest the boundary is dubbed the reverberant field.

The direct field is a region in which the source dominates the resultant acoustic field. The sound waves that are reflected off of the boundary do not significantly affect the acoustic field in this region. As a result, the direct acoustic field may be analyzed as if there was no boundary present.

An acoustic field that is dominated by sound waves reflected off of the boundary of the medium is referred to as a reverberant field. In this region, the direct sound waves from the source interact with the reflected waves to create a reverberant environment. The level of the sound field in this region is primarily controlled by the absorptivity of the boundary. A high absorptivity will result in a reverberant field that closely resembles the direct field, while a low absorptivity will create a field that looks drastically different from the direct field.

Both the reverberant field and the direct field may affect performance of an ANR headset. Most of the equipment on the market to date utilizes a closed cavity around the ears. By putting an enclosure around the control speaker and error microphone, the equipment introduces a reverberant field. Typically, these enclosures are lined with a highly absorptive material to reduce the reverberant field. An even better way, however,

to reduce this reverberant field would be to eliminate it all together. This can be accomplished by eliminating the enclosure itself. However, as well as keeping some of the generated sound within the cavity, the enclosure tends to decrease the levels of the unwanted sound that permeate the cavity from the outside. Clearly there is a trade off. The open air headset, referred to as a supra-aural headset, provides a zero reverberant field and exposes the microphone to the full exterior disturbance noise field. A closed cavity headset, referred to as a circum-aural headset, introduces a reverberant field, but muffles the exterior environment noise via passive absorption and transmission loss across the cavity walls. If an enclosure is used to gain the benefits of passive noise reduction, the casing should be made quite large to reduce the effects of the reverberant region. The supra-aural headset, on the other hand, would allow for flexibility in microphone location. This added flexibility could be used to allow the user to change microphone locations, resulting in improved control performance.

3.2.2 Far field vs. near field - a qualitative discussion

In addition to the boundaries of the acoustic field creating direct and reverberant fields, the acoustic field generated by a particular source is divided into near and far fields. The near field is quite complicated. Interference between contributing waves originating from various points on the source leads to sound pressures that exhibit rapidly varying maximum and minimum levels as measured along an axis outward from the source. Conversely, in the far field, the sound pressure levels simply decay monotonically with each doubling of the distance from the source.

The near field is the sound field immediately adjacent to the source characterized by two different regions. The region nearest the source is called the hydrodynamic near field and the region slightly further from the source is called the geometric near field. The hydrodynamic region is characterized by fluid motion which is not directly related to sound propagation. In the geometric near field, the particle velocity of each point on the source and the resulting pressure wave are in phase. However, the pressure and particle

velocity of the combined wave due to all the point sources on the source may not be in phase. This leads to maximums and minimums in the sound field.

The far field is described by Euler's equation, (3.1). It is characterized by a fluid density (ρ_o) and particle velocities ($\vec{u}$) that are in phase with pressure waves (P). As a result, the

$$\rho_o \frac{\partial \vec{u}}{\partial t} = -\nabla P \qquad (3.1)$$

sound pressure levels do not exhibit maximums and minimums. Instead, the sound pressure levels decay monotonically with each doubling of the frequency.

In light of the different characteristics of the near and far fields, it is desirable for the error microphone in an ANR headset to be located in the far field. This is due to the existence of the rapidly varying maximum and minimum pressure levels exhibited by the near field. Location of the microphone at one of these extreme pressure levels would decrease performance as the error microphone would not be exposed to the same acoustic pressure levels as the user's ear. Placement of the microphone in the far field would negate the possibility of improperly placing the microphone at one of these extreme pressure locations.

3.2.3 Defining the case of interest - a supra-aural headset

The existence of these four characteristics of acoustic fields has important implications for control applications. In order to predict the sound field at the error microphone, the geometric relation between the speaker and the microphone must be accurately defined as well as the geometric relation between the microphone and the boundaries. The effects of the boundary conditions will not be included in the following analysis of the acoustic propagation delay because the particular interest is in supra-aural ANR headsets. Therefore, the error microphone will always be located in the direct field of the control speaker.

3.3 Rigid diaphragm speaker model

In the field of acoustics, it is standard practice to model a speaker as a piston radiator. This model simplifies the analysis associated with the loudspeaker yet retains enough accuracy to prove useful in estimating sound fields. In this model, the speaker is considered to be a rigid structure, thus eliminating the higher-order break-up modes of actual loudspeaker diaphragms. Because of its simplicity, the piston radiator model provides a very convenient starting point to analyze the phase lag due to acoustic propagation in an ANR headset. With this simplified model, the acoustic field can be analyzed with the caveat that the analysis is only approximate. In the low frequency range, the model will be very accurate as the speaker's diaphragm vibrates in close agreement with a rigid approximation. As the diaphragm is forced with higher frequencies, the appearance of break-up modes results in a diaphragm that does not resemble a rigid body anymore and the model will lose accuracy.

3.4 Off-axis SPL

Now that the different characteristics of sound fields have been briefly discussed and the source defined, the analytical expressions for the sound pressures can be developed. The derivation starts by noting that an infinite number of pulsating point sources make up the surface of a piston radiator. By analyzing the sound field due to each one of these point sources and then summing their effects, a general expression for the sound field radiated by an infinitely baffled piston can be developed.

Figure 3.1 illustrates the case being studied where the cumulative effect of point sources, each with area $\approx \sigma d\sigma d\psi$, is described. Each point source is assumed to vibrate harmonically with a given velocity amplitude. The incremental contribution of each point source at the point of observation (O), which is a distance (r') from the point source, is expressed in terms of the density (ρ) of the medium (assumed to be air), the speed of

sound (c) in the medium, the free wave number (k) in the medium, the velocity amplitude (U) of the source, and the frequency (ω) of source excitation.

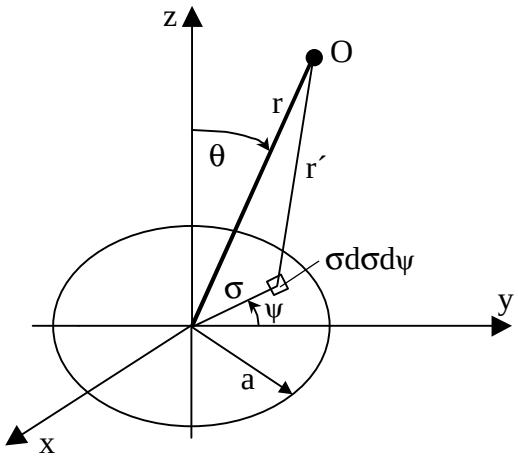

Figure 3.1 A piston of radius (a) lies in the x-y plane and vibrates parallel to the z-axis. It has a velocity amplitude (U). The square represents an infinitesimally small source of cross-sectional area ≈σdσdψ.

The first step in formulating the total sound pressure created by the piston source is to calculate the incremental pressure radiated by each pulsating point source. Each point source is initially approximated by a sphere with radius (a_o) radiating harmonically in free space.

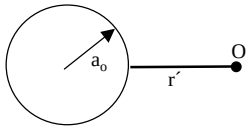

Figure 3.2 Each point source on the diaphragm is initially approximated by a speherical radiator with radius (a_o). The distance to the point of observation (O) is (r').

A spherical source radiates spherical waves and its normal surface velocity must equal the acoustic radial velocity. Applying this criteria yields (3.2). The equation describes the pressure (P) at a radial distance (r') from a spherical source of radius (a_o). The source strength (Q) has been used to replace the velocity amplitude and source surface area.

$$P(r') = \frac{i\rho\omega Q}{4\pi r'(1 - ika_o)} e^{i(\omega t - k(r' - a_o))} \qquad (3.2)$$

This equation is for a spherical source, but the sources that make up the surface of the diaphragm are point sources. Therefore, the radius of each of these sources tends towards zero. In addition, this equation is for a source in free space. A baffled source will radiate twice this amount of acoustic pressure. Taking the limit of (3.2) as $a_o \rightarrow 0$ and doubling the pressure results in the pressure due to a single point source that is infinitely baffled.

$$P(r') = \frac{i\rho\omega Q}{2\pi r'} e^{i(\omega t - kr')} \tag{3.3}$$

In order to find the total pressure radiated by all of the point sources, the above pressure must be discretized and then integrated over the surface area of the source. Discretizing (3.3) is accomplished by expressing the source strength (Q) as the product of a velocity amplitude (U) and a differential surface area (ds).

$$dP(r') = \frac{i\rho\omega U}{2\pi r'} e^{i(\omega t - kr')} ds \tag{3.4}$$

As shown in Figure 3.1, the vector locating the point of observation relative to each point source is given by (r'). To complete the analysis, this distance is expressed in terms of the distance from the center of the diaphragm. The position vector can be written in terms of the angles (ψ) and (θ) and the distances (r) and (σ) using the law of cosines.

$$r' = \sqrt{\sigma^2 + r^2 - 2r\sigma \sin\theta \cos\psi} \tag{3.5}$$

Finally, (3.5) is inserted into (3.4) and the result is integrated over the surface of the piston to obtain an expression for the sound pressure radiation at a point of observation for an infinitely baffled piston source.

$$P(r,\theta,t) = \frac{i\rho\omega U e^{i\omega t}}{2\pi} \int_0^a \sigma d\sigma \int_0^{2\pi} \frac{e^{-ik\sqrt{\sigma^2 + r^2 - 2r\sigma \sin\theta \cos\psi}}}{\sqrt{\sigma^2 + r^2 - 2r\sigma \sin\theta \cos\psi}} d\psi \tag{3.6}$$

Obtaining a closed form analytical solution to (3.6) is not a trivial matter. MATLAB and MATHEMATICA were both used in an attempt to perform the integration and neither program converged to a solution. Therefore, a second method of solution was attempted.

Sometimes re-writing a complicated integral as a power series simplifies the integration. The first step to writing the power series form of (3.6) is to express the exponential term as a complex value.

$$P(r,\theta,t) = \frac{i\rho\omega U e^{i\omega t}}{2\pi}\int_0^a \sigma d\sigma \int_0^{2\pi}\left[\frac{\cos(kz)}{z} + j\frac{\sin(kz)}{z}\right]d\psi \qquad (3.7)$$
$$z = \sqrt{\sigma^2 + r^2 - 2r\sigma\sin\theta\cos\psi}$$

Next, the power series expansion for the cosine and sine terms is carried out.

$$P(r,\theta,t) = \frac{i\rho\omega U e^{i\omega t}}{2\pi}\int_0^a \sigma d\sigma \int_0^{2\pi}\left[\begin{array}{l}\frac{1}{z} - \frac{(kz)^2}{2!z} + \frac{(kz)^4}{4!z} - \frac{(kz)^6}{6!z} + \ldots \\ \\ + jk - \frac{j(kz)^3}{3!z} + \frac{j(kz)^5}{5!z} - \frac{(kz)^7}{7!z} + \ldots\end{array}\right]d\psi \qquad (3.8)$$

The result is a series of simpler integrations that are performed individually. However, even in this simplified expansion, the computer software programs were unable to calculate a solution. For this reason, most analyses that require knowledge of the sound field radiated by a source use finite element (Kaizer and Leeuwestein, 1988) or boundary element techniques (Geddes et al. 1987). Such techniques are capable of predicting the radiation field for all position vectors (r), but the computation is very time consuming and they extend beyond the scope of this thesis. Instead, the concern is with the phase lag of the propagation of acoustic waves generated by a control speaker and measured at a field point (r,θ). To gain some insight into this phase lag and its relative magnitude, the on-axis pressure field can be formulated instead. Solution to the on-axis case is trivial and provides information that may be used to discuss the off-axis case.

3.5 On-axis SPL

Formulation of the on-axis pressure field begins with (3.4). The piston is still considered to be infinitely baffled and each point source has an infinitesimal radius, ($a_o \rightarrow 0$).

$$dP(r') = \frac{i\rho ckU}{2\pi r'} e^{i(\omega t - kr')} ds \tag{3.9}$$

Due to the on-axis assumption, the expression for the distance (r') from the point source to the point of observation is no longer a function of the angle (ψ). Instead, it is merely a function of the distance (r) from the center of the piston to the point of observation and the distance (σ) from the center of the piston to the point source. Inserting the expression for (r'), substituting in the differential source area, and integrating over the cross-sectional area of the piston results in the integrand used for calculating the on-axis pressure field.

$$P(r) = \frac{i\rho ckU}{2\pi} \int_0^a \frac{e^{i(\omega t - k\sqrt{\sigma^2 + r^2})}}{\sqrt{\sigma^2 + r^2}} 2\pi\sigma d\sigma \tag{3.10}$$

Unlike the case for the off-axis pressure, the on-axis pressure is calculated by performing only one integration. This single integral is a perfect differential and can be performed by hand. Performing the integration and simplifying produces an expression for the on-axis pressure field of an infinitely baffled piston radiator.

$$P(r) = \rho cU e^{i\omega t}[e^{-ikr} - e^{-ik\sqrt{\sigma^2 + a^2}}] \tag{3.11}$$

Equation (3.11) is interesting when analyzed. One interpretation of the expression is that the total acoustic pressure at a point along the axis of the piston is formed by the interaction of two plane waves. The first plane wave emanates from the center of the piston and travels a distance of (r). The second plane wave emanates from the edge of

the piston source and travels a distance of ($\sqrt{\sigma^2+a^2}$). This realization explains the phenomena of maximum and minimum SPLs in the radiation field. At certain locations along the axis, the two plane waves will travel distances that differ by one half of a wavelength. The effect is for the waves to be 180° out of phase and of equal amplitude, resulting in destructive interference. Hence, a local pressure minimum is created. At locations where the distances differ by one full wavelength, constructive interference occurs and a local maximum is created. As discussed previously, these minimums and maximums can deteriorate performance of an ANR headset. If the error microphone is located at one of these extreme pressure locations, the error microphone signal being controlled will not be an accurate representation of what the ear is actually hearing. For this reason, particular care must be taken to ensure that the microphone is not located at a point of extreme pressure.

3.5.1 Far field vs. near field revisited

The far field and the near field were previously discussed without examining the detail of the differences in radiation patterns. The on-axis pressure presents an opportune time to discuss these variations. Transition between the near field and the far field occurs when the field point of observation is distant compared to both the radius of the piston and the wavelength of radiation.

$$r/a \gg 1 \text{ and } r/a \gg ka \tag{3.12}$$

The local minimums and maximums created on the axis of the piston radiator are only associated with the near field. As transition to the far field occurs, the pressure tends to exhibit monotonically decaying behavior, asymptotically approaching a 1/r dependence. When (3.11) is re-written to reflect the far field assumption, it takes on the form of (3.13).

$$P(r) = \frac{\rho c U a}{2r} ka\, e^{j(\omega t - kr)} \tag{3.13}$$

This equation demonstrates an interesting concept in regards to the phase of the pressure field on-axis in the far field. There is no longer an interaction between two traveling plane waves. Instead, there is effectively only one plane wave. This is due to the fact that the angle between the two waves asymptotically approaches zero as the distance from the source grows large. Therefore, the two waves approach being in phase with each other. The total phase of the combined pressure wave is simply the phase due to one of the waves. As a result, it is important to calculate the ratio (r/a) and the product (ka) for the position of the error microphone in a headset to determine if near field or far field acoustic formulations apply. Performance is, therefore, dependent upon the microphone position and radiation wavelength.

The on-axis pressure magnitudes for two diaphragms are illustrated in Figure 3.3. The figure demonstrates how a different value for (ka) determines whether or not typical near

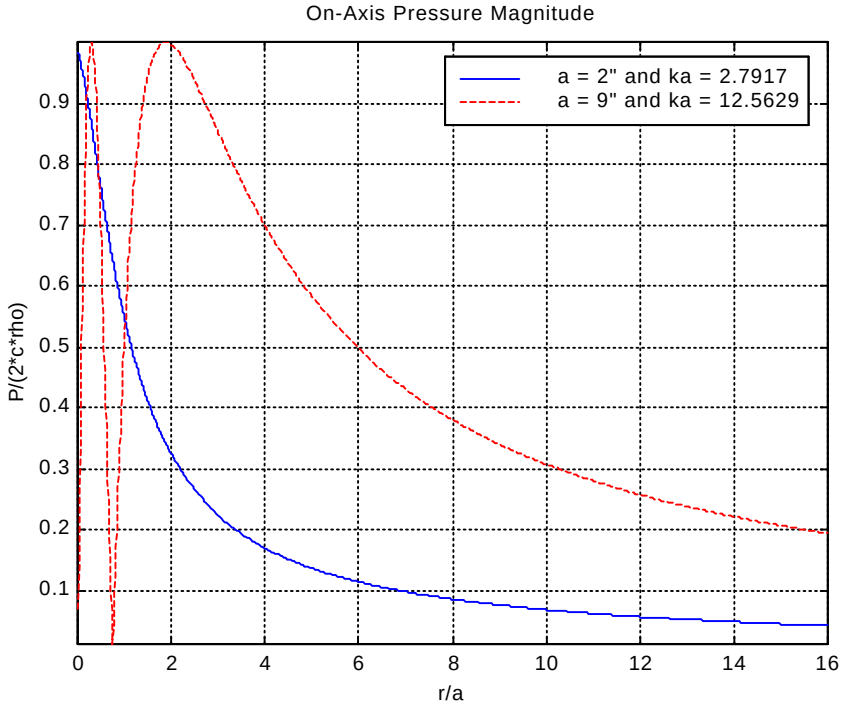

Figure 3.3 On-axis pressure magnitude resulting from an ideal piston radiator. The piston is excited at 3000 Hz.

field qualities exist. For the low value of (ka), the pressure magnitude does not exhibit extremes. A near field exists, but it does not display the characteristic maximum and minimum pressures. As the radius of the piston grows larger, however, (ka) grows in value and the extreme pressures begin to appear, causing the discussed near field condition to exist. In both cases, a (1/r) dependence is evident as (r/a) increases and the far field is reached.

3.5.2 ANR headset

Where should the microphone be placed in an ANR headset? The answer to this question requires knowledge of the radius of the control speaker and the highest frequency of interest so a maximum value for (ka) can be calculated. Based on this calculation, the microphone may be placed at a particular value of (r/a). A conservative estimate of the highest frequency that an ANR headset would control is 3000 Hz. Typically, active noise control is applied to low frequency spectrums where plane waves prevail and passive reduction is more difficult to achieve. Suppression at these low frequencies can be highly successful, but as higher frequencies are encountered plane wave propagation is overshadowed by higher-order waves. These higher-order waves do not lend themselves to active control and thus, passive techniques are usually employed. In addition to a maximum frequency of 3000 Hz, a 1/2 inch radius provides a large, conservative estimate of the speaker dimension. Generally, it is desirable to have the speaker be as small and lightweight as possible. Evaluating (3.11) with these two parameters leads to an on-axis pressure magnitude illustrated by Figure 3.4.

The pressure field does not exhibit the usual near field characteristics for the 1/2 inch driver. There are no minimum or maximum pressure locations. This is due to the fact that the wavelength at 3000 Hz is still relatively large (4.5 inches). Comparing this to the source radius of 1/2 inch demonstrates that there is no typical near field on-axis. The figure also demonstrates that the near field does not begin to exhibit extreme pressures

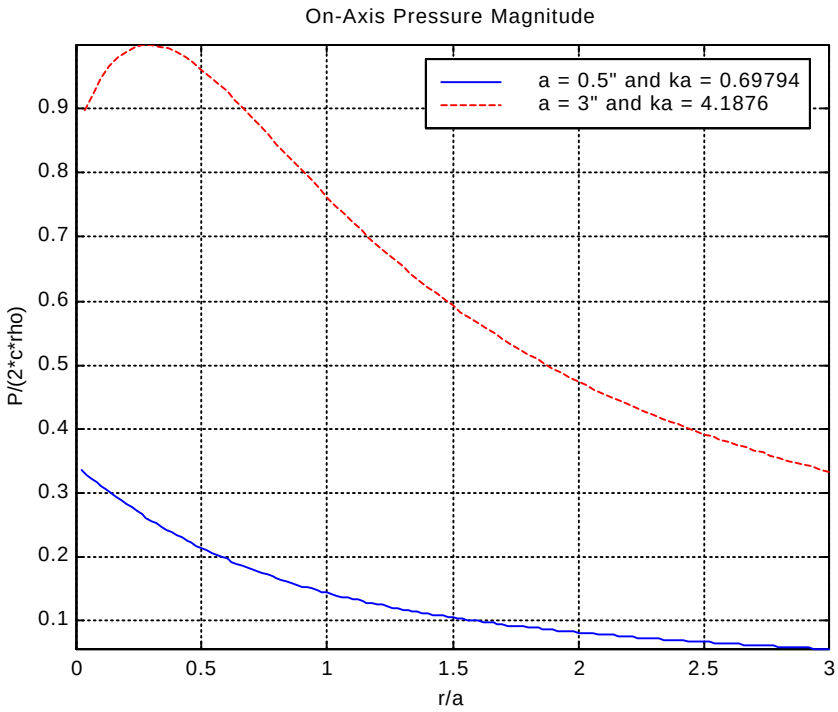

Figure 3.4 On-axis pressure magnitude resulting from an ideal piston radiator. The piston is excited at 3000 Hz.

until the source radius is increased to 3 inches. This bodes well for an ANR headset. As long as the speaker diameter is kept below 1 inch and the frequencies of concern remain below 3000 Hz, the microphone placement does not need to be concerned with on-axis maximum and minimum sound pressure levels.

3.5.3 Phase considerations

The goal in analyzing the sound field radiated by a piston was to quantify the phase lag caused by acoustic propagation from the control speaker to the error microphone in an ANR headset. Insight can be gained into this phase lag by analyzing the phase of the on-axis case described by (3.11). The phase lag is calculate by formulating the complex pressure vector.

$$P(r) = \rho cU[\cos(kr) - \cos\left(k\sqrt{r^2 + a^2}\right) + i\sin\left(k\sqrt{r^2 + a^2}\right) - i\sin(kr)]e^{i\omega t} \qquad (3.14)$$

The next step is defining a time and position reference for the phase. In this analysis, the phase is referenced to the displacement of the center of the diaphragm at an initial time ($t = 0$). Equation (3.14), however, relates the pressure to the velocity of the diaphragm. This is accounted for by calculating the phase of (3.14) and subtracting the 90° that velocity leads displacement. The final matter of convention in defining the phase lag is to make the displacement of the diaphragm a vector quantity with direction and magnitude. To accomplish this, a positive sign is associated with an outward excursion of the diaphragm as illustrated in Figure 3.5.

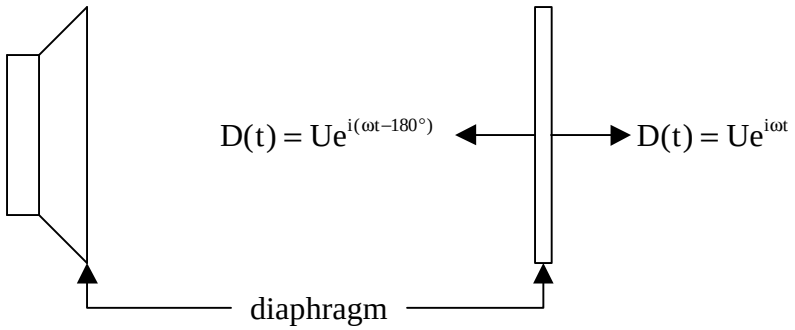

Figure 3.5 Vector representation of diaphragm displacement used to reference phase.

Applying the defined convention, the phase of (3.14) is calculated by (3.15).

$$\phi(r,k,a) = \tan^{-1}\left[\frac{-\sin(kr)+\sin(k\sqrt{r^2+a^2})}{\cos(kr)-\cos(k\sqrt{r^2+a^2})}\right] - 90° + \phi_{cone} \qquad (3.15)$$

In this equation, (ϕ_{cone}) represents the phase of the diaphragm displacement as defined in Figure 3.5. Therefore, if the center of the diaphragm is moving outward at time zero, (ϕ_{cone}) will be 0° and if the cone is moving inward, (ϕ_{cone}) will be -180°.

The expression for the phase lag is a function of excitation frequency, source radius and distance from the source. As depicted in Figure 3.6, the phase lag increases as the wavelength diminishes, or frequency increases, and as the distance from the source increases. Therefore, in order to minimize phase lag, good ANR headset design will locate the microphone in very close proximity to the control speaker.

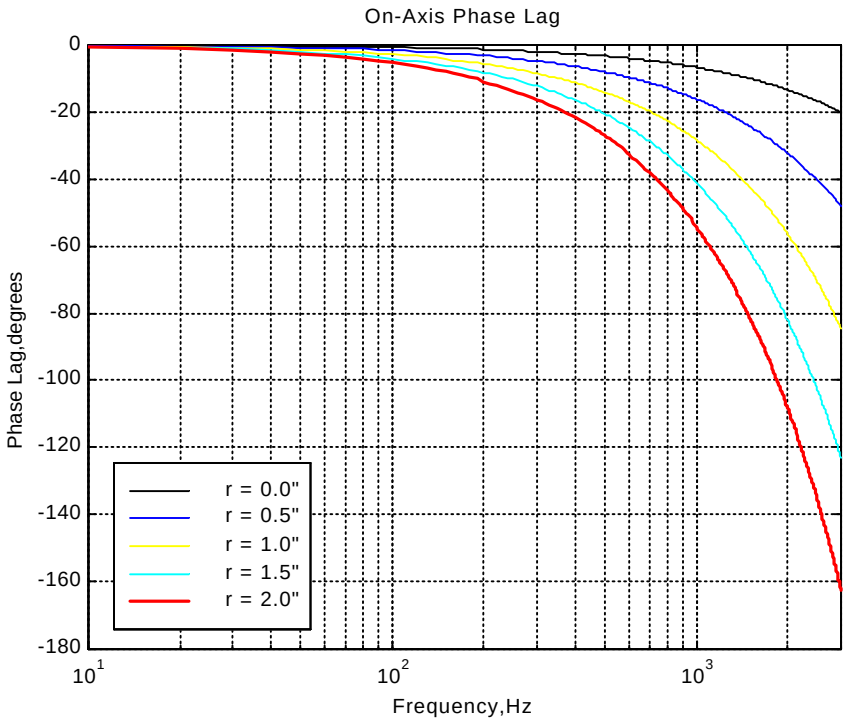

Figure 3.6 On-axis acoustic phase lag due to a 1/2 inch radius ideal piston radiator.

An alternate representation of the phase lag associated with acoustic propagation of waves employs a time delay model. It physically takes time for waves created by a source to propagate to a point of observation. Simplifying the phase lag into a time delay model produces a close approximation to the phase lag calculated by (3.11). The phase lag as a time delay (ϕ_{td}) is expressed in terms of the distance (r) from the center of the source to the observation point, the frequency (f) of propagation, and the speed of sound (c). Forming the ratio of the propagation time to the period of the wave and simplifying yields (3.16).

$$\phi_{td} = -\frac{rf}{c}360° \qquad (3.16)$$

Figure 3.7 illustrates how accurate the time delay approximation is in predicting the phase lag calculated by (3.11). At frequencies as high as 3000 Hz, the time delay model is only in error by 10°. This accuracy supports approximating the phase lag created by the acoustic propagation in a ANR headset by a time delay. The discrepancy between the

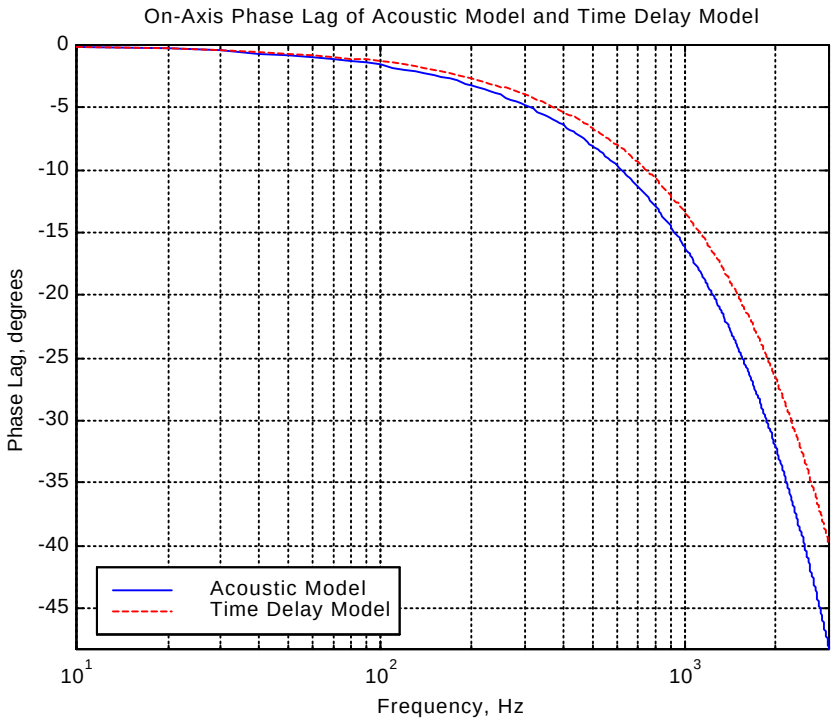

Figure 3.7 On-axis acoustic phase lag due to a 1/2 inch radius ideal piston radiator. Acoustic model is predicted by equation (3.11) and time delay model is predicted by equation (3.16). (r/a = 1)

two predictions arises because the equation plotted is strictly valid for the near field, ($r/a \leq 1$), in which their are two plane waves interacting. The time delay model does not account for two waves. The time delay model's accuracy is immediately obvious, however, when the far field, on-axis case is examined. In (3.13), the phase lag associated with the pressure is determined by the exponential term.

$$\phi = -kr \tag{3.17}$$

With a little simplification the phase lag is re-written in a familiar form.

$$\phi = -\frac{fr}{c}360° \tag{3.18}$$

Therefore, in the far field, on-axis the phase lag is exactly equivalent to a time delay. So, a time delay model can be used fairly accurately in the near field because of its simplicity and as transition to the far field takes place the model will approach 100% accuracy.

3.6 Theory for experimental verification of rigid diaphragm acoustics

The acoustic phase lag of a wave propagating from the control speaker to the error microphone has been formulated for the on-axis case when the speaker diaphragm is approximated by a rigid piston, but does the formulation predict the actual acoustic phase lag as measured in the lab? Verification of the model was accomplished by experimentally measuring the transfer function from audio amplifier input voltage to microphone output voltage, Figure 3.8. The transfer function describing the speaker dynamics was then isolated from this data by mathematically removing the individual transfer functions of the audio amplifier, acoustic propagation, and microphone. Finally, comparison of this measured speaker transfer function to a standard speaker model enabled conclusions to be made about the accuracy of the acoustic formulations. If the measured and modeled speaker dynamics matched, it would have been concluded that the acoustics that were removed from the data were accurate.

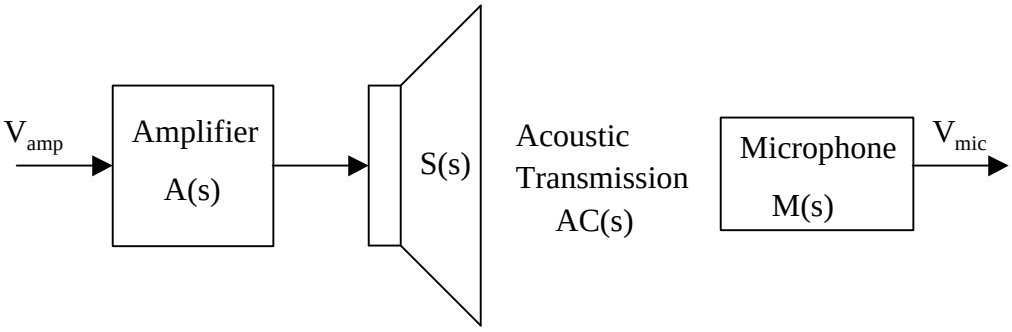

Figure 3.8 The plant for an ANR headset.

Experimental verification of the acoustic propagation phase lag consists of first, measuring the overall system transfer function from input voltage to the audio amplifier (V_{amp}) to output voltage from the microphone (V_{mic}) as shown in Figure 3.8.

$$\frac{V_{mic}}{V_{amp}} = A(s)S(s)AC(s)M(s) \tag{3.19}$$

This overall transfer function, (3.19), is composed of four individual transfer functions, but only two of them have associated dynamics. A Dynaco audio amplifier was used to power the control speaker. The amplifier had a flat frequency response with a fixed gain of 26 dB and 0° of phase lag across a frequency span 100-10,000 Hz. The condenser microphone had a similar flat frequency response with a fixed gain and 0° of phase lag across the same frequency span. Consequently, these two individual transfer functions do not affect the phase of the overall measured transfer function given by (3.19). Additionally, they each increase the magnitude by a fixed amount that is uniform across the frequency span and do not alter the dynamics of the transfer function. As a result, the magnitude and phase lag of the measured data is representative of the dynamics and phase lag due to the speaker dynamics and acoustic propagation only.

In order to separate the last two individual transfer functions, the acoustic propagation transfer function and the speaker transfer function, the speaker must be modeled. The standard low frequency model of a speaker with a diaphragm vibrating in its rigid piston mode (Clark and Lane, 1998) consists of a complex pair of poles at the mechanical resonance of the system. In the low frequency region, the diaphragm vibrates as a rigid piston and this model is accurate. However, as the speaker drive frequency increases, the diaphragm starts to vibrate with more mechanical modes and the rigid piston behavior is no longer observed. As a first approximation, the standard speaker model is used and the higher-order modes of diaphragm vibration are ignored. This model predicts a speaker transfer function that has a bode plot similar to the response illustrated in Figure 3.9.

This bode plot forms the basis upon which the acoustic propagation formulation is evaluated. Once the dynamics of the formulated acoustics are mathematically removed from the measured data, the resulting data should be representative of the speaker dynamics alone and exhibit a bode plot similar to Figure 3.9. The bode plot illustrates the

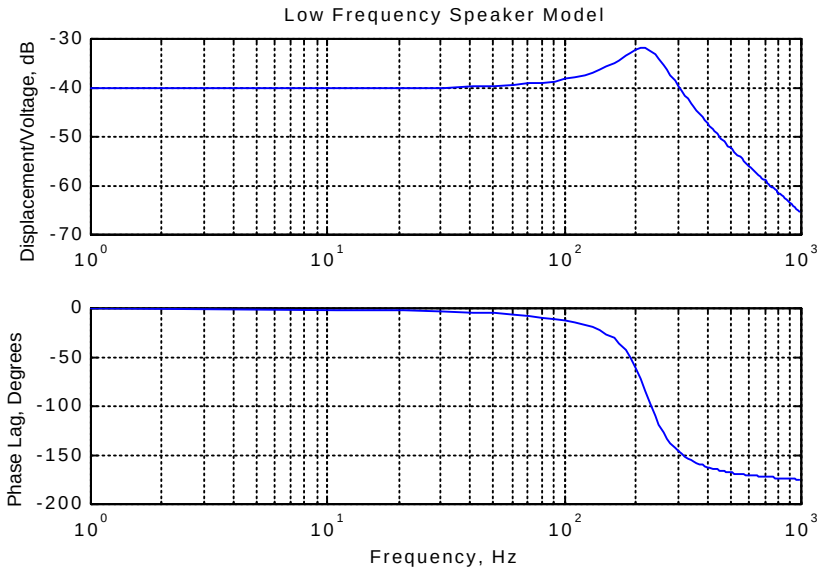

Figure 3.9 Bode plot of low frequency model speaker dynamics. The transfer function represents (voltage)/(diaphragm displacement). The mechanical resonance of the speaker is at 225 Hz.

fact that the diaphragm displacement is in phase with the supplied voltage at very low frequencies and as the mechanical resonance of the system is surpassed the displacement lags the input voltage by 180°.

In constructing the prototype plant as illustrated in Figure 3.8, a 2 inch diameter speaker was chosen. The mechanical resonance of the speaker was observed at 225 Hz. Therefore, the bode plot expected as a result of mathematically removing the acoustic transfer function from the measured data should have a magnitude peak at 225 Hz. At this natural frequency, the phase response should pass through -90° and then, asymptotically approach -180°.

3.7 Experimental verification of rigid diaphragm acoustics

Applying the theory discussed in the previous section, data was taken representing the overall system's transfer function as given by (3.19) and the modeled acoustic propagation transfer function was mathematically removed. The resulting data, which should have represented only the dynamics of the speaker, did not match the bode plot of

Figure 3.9. As a result, it was concluded that the acoustic formulations and speaker model were not accurate in describing the physical system.

The measured magnitude response of the prototype system is plotted in Figure 3.10. The figure demonstrates the (1/r) dependence of the pressure field on-axis in the far field. As the microphone was moved further away from the speaker diaphragm, the acoustic pressure magnitude at the error microphone decreased accordingly. The key information in the figure is contained in the frequencies at which the magnitude peaks. The peak at 225 Hz corresponds to the mechanical resonance of the speaker. At 4450 Hz, the magnitude peaks again as a result of a higher-order breakup mode of the diaphragm. This higher-order mode of the diaphragm was not accounted for in the speaker model and will affect the accuracy of the model as will be demonstrated in the following paragraphs.

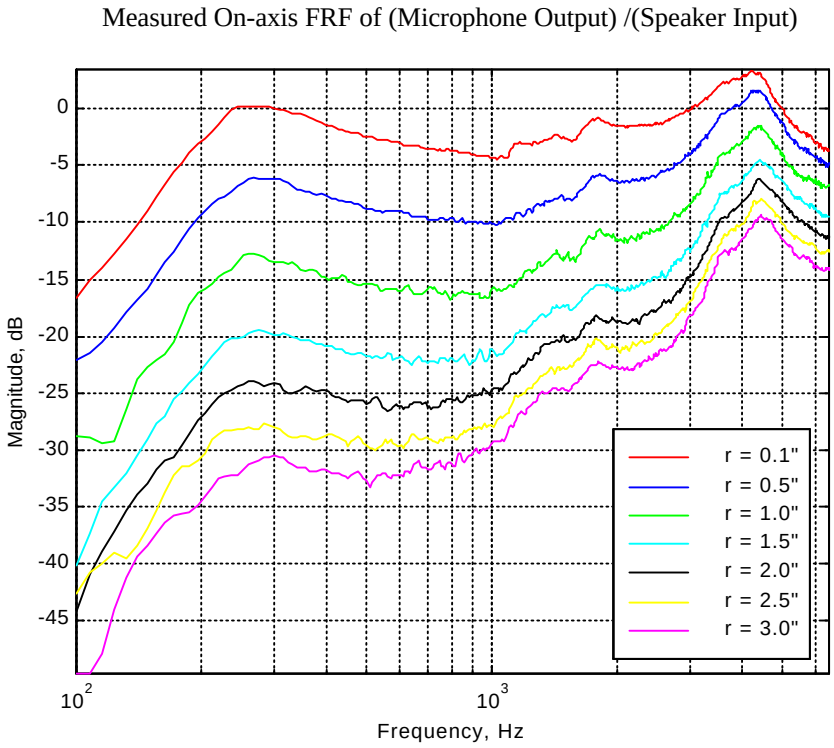

Figure 3.10 Measured on-axis magnitude response of a 1 inch radius speaker. Magnitude represents (microphone output)/(speaker input).

The phase response of the prototype system is plotted in Figure 3.11. The figure is quite interesting when the dependence on (r) is examined. In the low frequency region, the

phase is approximately independent of the distance from the diaphragm. It is not until 1000 Hz that a deviation is seen between each of the data sets and a dependence on (r) is observed. The explanation lies in the fact that the acoustic wavelength is large at the lower frequencies, resulting in an acoustic propagation phase lag that is quite small. The system's phase plot serves as verification of the natural frequency of the mechanical resonance of the speaker because it passes through -90° at 225 Hz.

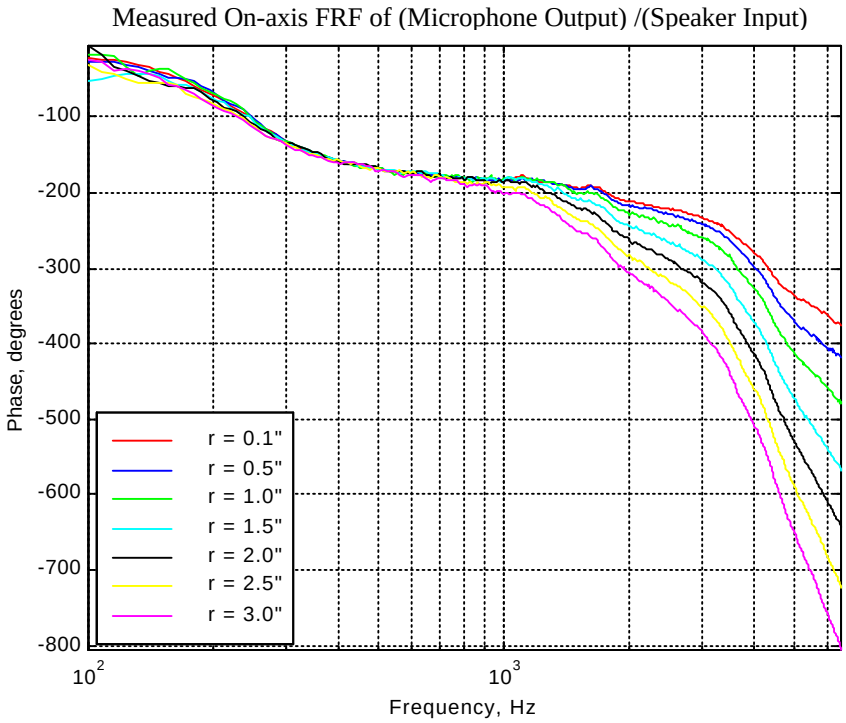

Figure 3.11 Measured on-axis phase response of a 1 inch radius speaker. Phase lag represents (microphone output)/(speaker input).

As previously discussed, the removal of the acoustic phase lag as calculated by (3.15) from the measured data should yield a bode plot of only the speaker dynamics. If the resultant bode plot does match the predicted speaker dynamics, then it can safely be concluded that the speaker model and acoustic phase lag formulation are accurate. If the resultant bode plot does not match the predicted speaker dynamics, the speaker model and acoustic phase lag formulation will have to be reformulated. The resultant bode plot should be identical for each data set as the only parameter that varies among the data sets

is the distance (r). This distance from the speaker diaphragm to the error microphone is only present in the acoustic dynamics which are being removed from the data sets.

Before mathematically removing the acoustic phase lag from the measured phase lag of the prototype system, the vector representation of the speaker diaphragm motion must be formulated. This is accomplished in the same manner as discussed previously and illustrated in Figure 3.5. In the low frequency range, the diaphragm is referenced as having a positive displacement at its center ($r = 0$). This signifies that the acoustic pressure is in phase with diaphragm displacement (D(t)) at the face of the piston. Finally, a time reference of ($t = 0$) is enforced.

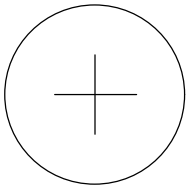

Figure 3.12 Speaker diaphragm displacement at t = 0. The positive sign refers to an outward excursion.

The vector representation that results for the motion of the speaker diaphragm is presented by (3.20). Referring back to section 3.5.3 and (3.15), this diaphragm motion yields ($\phi_{cone} = 0°$). The result is (3.21) which formulates the acoustic propagation phase lag referenced to diaphragm displacement.

$$D(t) = Ue^{i\omega t + 0°} \tag{3.20}$$

$$\phi(r, k, a) = \tan^{-1}\left[\frac{-\sin(kr) + \sin(k\sqrt{r^2 + a^2})}{\cos(kr) - \cos(k\sqrt{r^2 + a^2})}\right] - 90° \tag{3.21}$$

Figure 3.13 illustrates the results of removing the acoustic propagation phase lag predicted by (3.21) from the measured transfer function data. Assuming the speaker model and acoustic model were correct, this data should match that of the speaker

model's bode plot. The plot, however, does not match. In the low frequency region, each data set passes through -90° at 225 Hz as predicted by the model. The data sets are also independent of the distance from the diaphragm to the error microphone (r) in this low frequency region. These two observations validate the speaker model and acoustic model in the low frequency region. At higher frequencies, however, the data sets separate, displaying a dependence on (r). Additionally, the data sets do not asymptotically approach -180°. Instead, as the microphone is moved farther from the diaphragm, the data exhibit decreased phase lag. The acoustic phase lag being removed

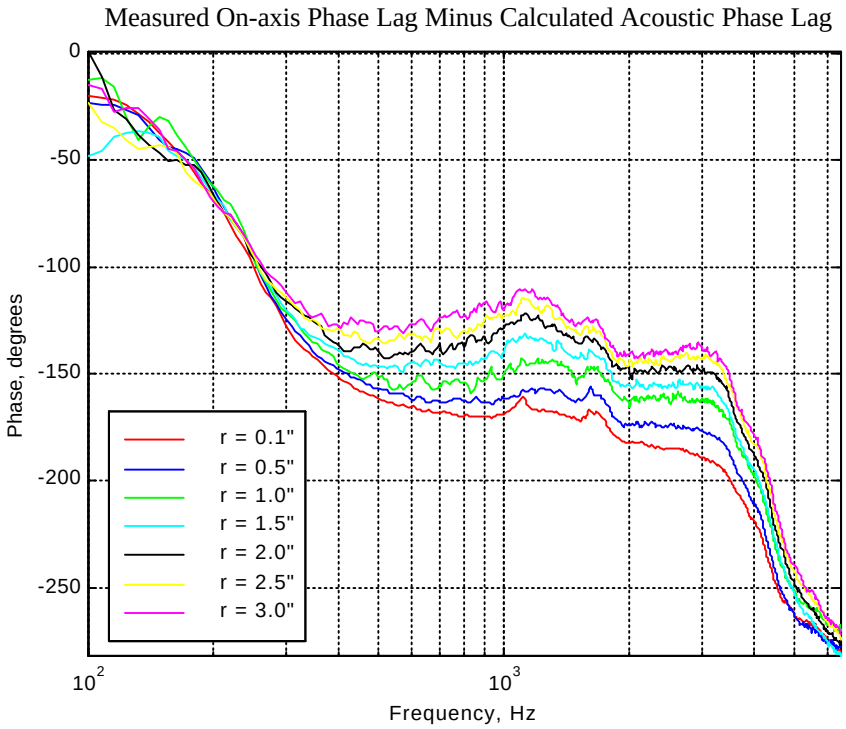

Figure 3.13 The resulting phase response after the predicted acoustic propagation phase lag has been mathematically removed from the experimentally measured data. Acoustic phase lag predicted by rigid diaphragm formulation,(3.11).

from the original data is too large at these higher frequencies. This combination of a dependence on (r) and a phase lag that does not approach -180° invalidates the speaker model and acoustic model in the higher frequencies. As a result, the speaker model and acoustic model must both be improved if high frequency performance is to be predicted.

3.8 Diaphragm dynamics - first breakup mode

The previous section did not support the practice of analyzing the speaker diaphragm as a rigid piston. Instead, it supported the notion that the higher-order breakup modes of the diaphragm interact with the mechanical resonance of the speaker. The system transfer function magnitude data displayed two distinct peaks: one at 225 Hz and one at 4450 Hz. The peak at 4450 Hz is the natural frequency for the first breakup mode, or second mode. Inclusion of the dynamics due to this second mode of diaphragm vibration, provided the necessary accuracy in the speaker model and acoustic model.

To incorporate the second mode of vibration, the diaphragm is modeled as a circular membrane. Circular membranes exhibit two kinds of modes beyond their rigid body mode, Figure 3.14. One of these modes is a radial mode in which nodes appear at radial lines. The second type of mode is a circumferential mode in which nodal circles appear.

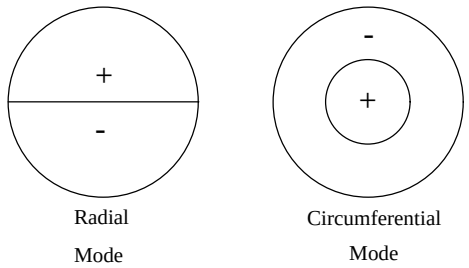

Figure 3.14 Modes of a circular membrane. The positive signs refer to an outward excursion and the negative sign refers to an inward excursion.

With the aid of time-averaged holography (Frankhort, 1978), analyses have been done to examine the breakup modes of conical loudspeaker diaphragms. Computerized analysis (Nomoto and Suzuki, 1982) has also aided some analyses of this phenomena. Today, the use of laser transducers makes viewing the vibration patterns on a diaphragm a simple matter. The vibration pattern on the diaphragm at 4450 Hz will clarify which of the two circular membrane modes characterizes the first breakup mode.

An Ometron VPI Scanning Laser Doppler Vibrometer was used to scan the surface of the speaker diaphragm driven at a single frequency. With the speaker suspended by strings, so as to remove the dynamics of a rigid support, a rectangular velocity scan was taken that encompassed the round speaker face. The laser data acquisition system provided a visual display of the velocity pattern, Figure 3.15 and Figure 3.16, on the speaker diaphragm referenced to the speaker input voltage. Each of the figures contains three individual plots. The first plot represents the real part of the velocity response. The middle plot represents the phase response of the diaphragm's velocity relative to the speaker input voltage. This middle plot contains data taken on the metal support of the speaker that is not related to the dynamics of the diaphragm. To alleviate any confusion, the third plot contains the phase response of only the diaphragm.

Figure 3.15 illustrates the results when the speaker was driven at 225 Hz corresponding to its mechanical resonance. As expected, the magnitude response was rather large in amplitude and uniform across the face of the diaphragm. The phase was also constant across the face of the diaphragm as the velocity and speaker input voltage were in phase. This figure supports the practice of modeling the diaphragm as a rigid piston at low frequencies.

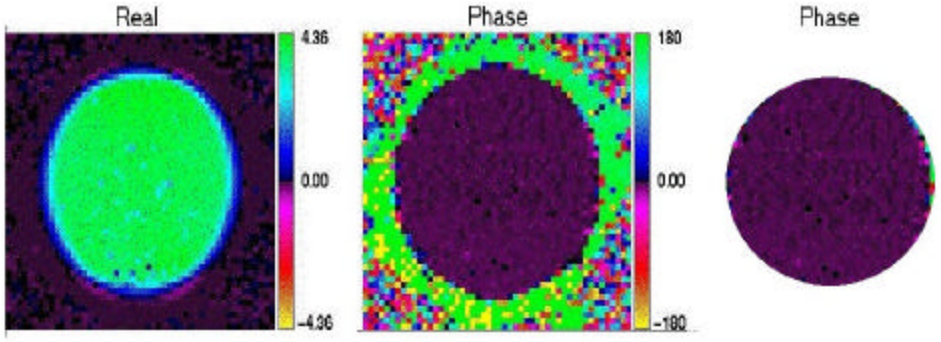

Figure 3.15 The real part of the diaphragm velocity and the phase response of a 1 inch radius speaker. The phase is in degrees and the real response is unitless. Speaker excited at mechanical resonance, f = 225 Hz.

Figure 3.16 illustrates the results when the speaker was driven at 4450 Hz in an effort to describe the vibration pattern of the diaphragm in its first breakup mode. The previous work done in exploring breakup modes of speaker diaphragms suggested that the flexible outer edge of the diaphragm, the surround, starts to vibrate in anti-phase with the central portion of the diaphragm to create the first breakup mode. The results in Figure 3.16 are in full agreement with this theory. As evidenced by the opposite signs, the real data suggest that the surround is vibrating in an opposite direction from the central portion of the diaphragm. The phase plot provides further collaboration as it demonstrates that the surround is 180° out of phase with the central portion.

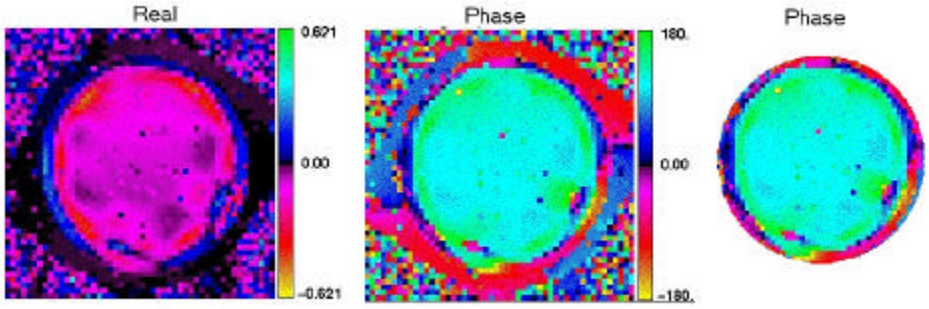

Figure 3.16 The real part of the diaphragm velocity and the phase response of a 1 inch radius speaker. The phase is in degrees and the real response is unitless. Speaker excited at first breakup frequency, f = 4450 Hz.

From these two laser scans it is clear that the first beakup mode of the diaphragm is a circumferential mode, Figure 3.17. The data illustrates that the amplitude of the first mode is 4. It also demonstrates that the amplitudes of the surround and central portion of the diaphragm in mode two are equal and opposite at 0.14. These relative amplitudes are necassery when analyzing the influence that each mode has at a particular frequency on the acoustic field created by the speaker diaphragm.

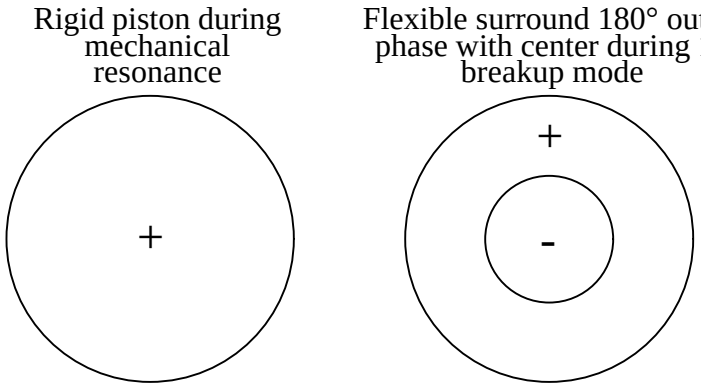

Figure 3.17 Graphical representation of the mode shapes of a speaker diaphragm. Positive sign represents an outward excursion and negative sign represents an inward excursion.

3.9 Acoustics based on first breakup mode

The developments of the previous section necessitated that the acoustic analysis include the first breakup mode of the diaphragm. Once the acoustic propagation phase lag from the diaphragm's second mode was formulated, it was mathematically removed from the measured transfer function data in an effort to isolate the speaker dynamics again. However, the resultant data did not match the bode plot as predicted by the speaker model in the low frequencies.

Formulating the pressure radiation due to the second diaphragm mode begins with (3.9), which is shown again for reference.

$$P(r) = \frac{i\rho ckU}{2\pi}\int_0^a \frac{e^{i(\omega t - k\sqrt{\sigma^2+r^2})}}{\sqrt{\sigma^2+r^2}} 2\pi\sigma d\sigma \qquad (3.9)$$

The only difference from the rigid piston analysis is that the diaphragm velocity ($Ue^{i\omega t}$) is now a function of cone radius (a) for the breakup mode. Over the center of the diaphragm, the velocity is inward ($U_{in}e^{i\omega t+\phi_{in}}$) with ($\phi_{in}$=-180°). Over the flexible surround, the velocity is outward ($U_{out}e^{i\omega t+\phi_{out}}$) with ($\phi_{out}$ = 0°). This necessitates splitting the surface integral up into two parts. The first integral is taken from the center of the

diaphragm to the inner edge of the surround and the second integral is taken from the inner edge of the surround (a_{in}) to the outer edge of the surround (a_{out}).

$$P(r) = \frac{i\rho\omega}{2\pi}\left[\int_0^{a_{in}} U_{in} e^{i\omega t+\phi_{in}} \frac{e^{-ik\sqrt{r^2+a^2}}}{\sqrt{r^2+a^2}} 2\pi\sigma d\sigma + \int_{a_{in}}^{a_{out}} U_{out} e^{i\omega t+\phi_{out}} \frac{e^{-ik\sqrt{r^2+a^2}}}{\sqrt{r^2+a^2}} 2\pi\sigma d\sigma\right] \qquad (3.22)$$

Solution of this two part integral is achieved in exactly the same manner as the first mode's solution. Integration and simplification produces and expression for the sound pressure at a point along the axis a distance (r) away from the diaphragm.

$$P(r) = \rho c e^{i\omega t}\left[U_{in} e^{\phi_{in}}\left(e^{-ikr} - e^{-ik\sqrt{r^2+a_{in}^2}}\right) + U_{out} e^{\phi_{out}}\left(e^{-ik\sqrt{r^2+a_{in}^2}} - e^{-ik\sqrt{r^2+a_{out}^2}}\right)\right] \qquad (3.23)$$

The equation demonstrates that there are two plane waves emanating from the central portion of the diaphragm and two plane waves emanating from the surround, creating a total of four plane waves that interact. The phase lag is found by re-writing (3.23) as a complex vector and finding the angle of the vector. The result is (3.24) which has 90° subtracted to transform the phase relative to diaphragm velocity into phase relative to diaphragm displacement.

$$\begin{aligned}
\text{real} &= U_{in}\left(\cos(kr) - \cos\left(k\sqrt{r^2+a_{in}^2}\right)\right) + \\
&\quad U_{out}\left(\cos(k\sqrt{r^2+a_{in}^2}) - \cos\left(k\sqrt{r^2+a_{out}^2}\right)\right) \\
\text{imaginary} &= U_{in}\left(\sin\left(k\sqrt{r^2+a_{in}^2}\right) - \sin(kr)\right) + \\
&\quad U_{out}\left(\sin(k\sqrt{r^2+a_{out}^2}) - \sin\left(k\sqrt{r^2+a_{in}^2}\right)\right)
\end{aligned} \qquad (3.24)$$

$$\phi(r,a,k) = \tan^{-1}\left[\frac{\text{imaginary}}{\text{real}}\right] - 90° + \phi_{in} + \phi_{out}$$

Figure 3.18 compares the acoustic phase response of the mechanical resonance and the diaphragm's second mode. The breakup mode yields less of an acoustic phase lag than the mechanical resonance. This is due to the complex interaction of the four plane traveling waves. This reduced acoustic phase lag is expected from the previous analysis in which the acoustic phase lag due to the mechanical resonance was mathematically removed from the measured data. In Figure 3.13, the data sets with the acoustic propagation phase lag removed separated at higher frequencies. As the error microphone was moved further from the diaphragm, the data sets drifted above the -180° line more and more. The acoustic propagation model was predicting too much phase lag. As the diaphragm's second mode's acoustic phase indicates, the phase lag should be less at these higher frequencies. Therefore, when the acoustic phase of this mode is removed at the higher frequencies it is expected that the data sets will shift downward toward -180°.

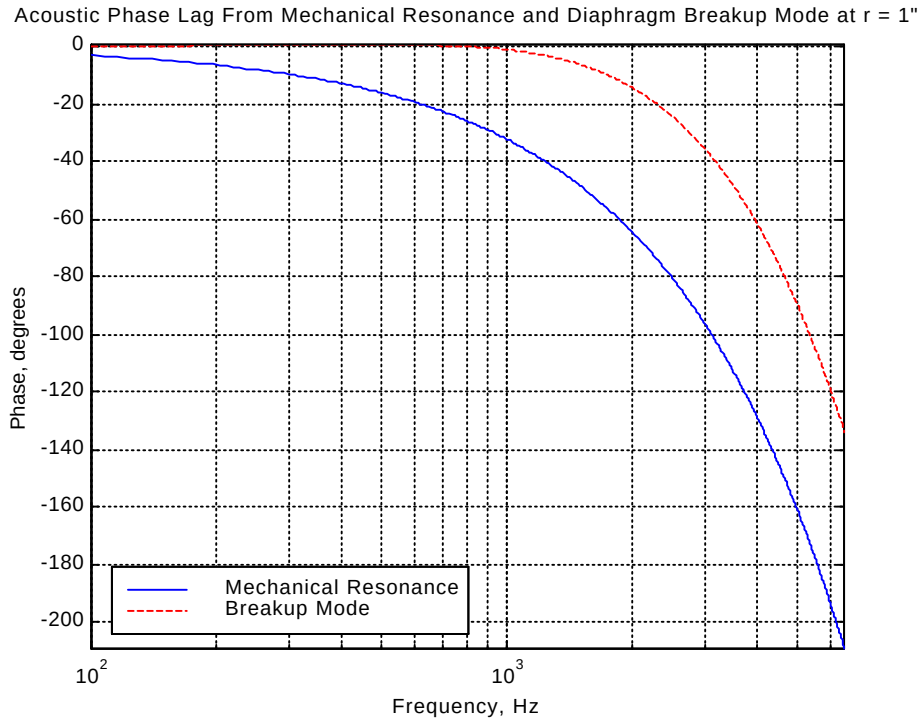

Figure 3.18 Acoustic propagation phase lag caused by each mode of the diaphragm.

The formulation for the acoustic propagation phase lag due to the diaphragm's second mode alone was removed from the transfer function that was measured in the laboratory. Removing only this second mode's acoustic phase lag and ignoring the fact that there is a

mechanical resonance yields Figure 3.19. If only this mode were present, this plot should contain nothing but accurate speaker dynamics. All of the data sets should lie on top of each other and the low frequencies should resemble the bode plot of the low frequency speaker dynamics model. As the figure illustrates, the data does not satisfy either requirement. In the lower frequencies, the data sets tend to separate, and at the higher frequencies the data sets tend to come together. However, this is expected. Only the phase lag due to the higher-order mode was removed from the data sets, so the error at the low frequencies should be large. As the higher frequencies are reached, the error should diminish, as it does. This provides qualitative support that the acoustic propagation phase lag formulation due to the second mode of the diaphragm is correct.

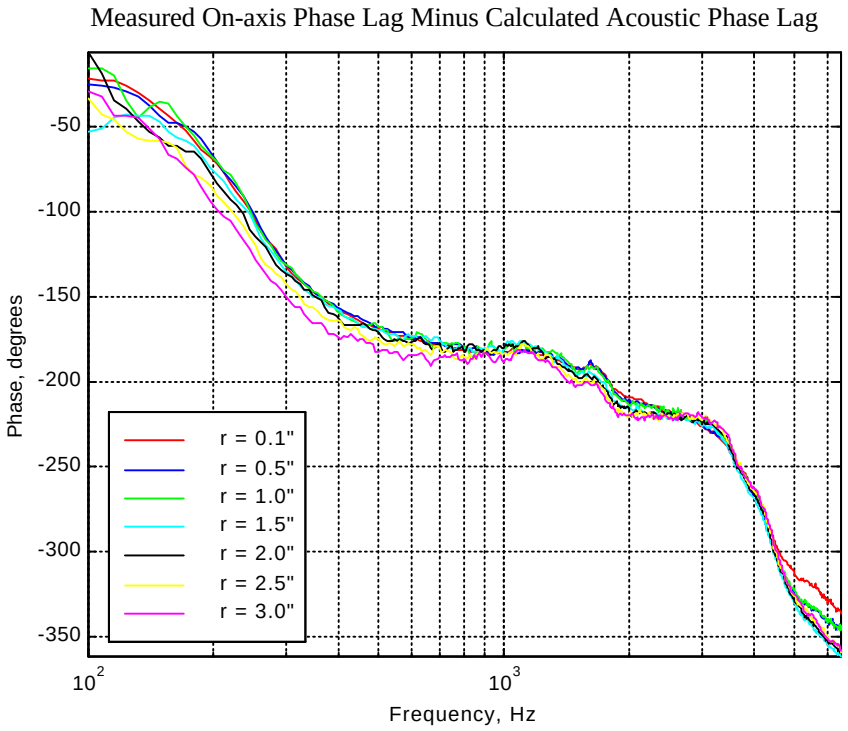

Figure 3.19 The resulting phase response after the predicted acoustic propagation phase lag has been mathematically removed from the experimentally measured data. Acoustic phase lag predicted by breakup mode formulation,(3.24).

3.10 Acoustics based on rigid diaphragm and first breakup mode

It has been demonstrated that both the mechanical resonance, or mechanical mode, and the diaphragm's first breakup mode affect the acoustic propagation phase lag in an ANR system. To complete the analysis, the phase lag due to each mode must be combined into a total phase lag. A modal analysis of the diaphragm yields the proper modal contributions of the mechanical resonance and the diaphragm breakup mode at each frequency. By mathematically removing the combined acoustic radiation effects of the speakers mechanical resonance and the diaphragm's first breakup mode, the measured data finally matched the predicted bode plot for a speaker, providing proof of the accuracy of the acoustic propagation phase lag formulation.

A modal analysis attempts to weight the contribution, or influence, of each mode at a particular frequency. Analyzing across a frequency spectrum results in a modal contribution curve for each mode. This curve is used to determine the combined phase lag at each frequency by taking the modal contribution of mode one (Mc_1) and multiplying by the phase of mode one (ϕ_1), then taking the modal contribution of mode two (Mc_2) and multiplying by the phase lag of mode two (ϕ_2). The combined phase lag at a particular frequency is the summation of these two multiplications, (3.25).

$$\phi_{combined}(f) = Mc_1(f)\phi_1(f) + Mc_2(f)\phi_2(f) \qquad (3.25)$$

The determination of the modal contribution for each mode requires examining the effect of each mode across the frequency spectrum. At low frequencies, the mechanical mode should dominate, and at high frequencies the breakup mode should dominate. A good starting point is to assume that the first mode has a decaying exponential modal contribution curve that is unity for all frequencies prior to mechanical resonance, 225 Hz, and zero by 2500 Hz. On the contrary, it is assumed that the diaphragm's breakup mode has an increasing exponential modal contribution curve that is zero prior to 500 Hz and unity by 1000 Hz and thereafter. The two modal contributions curves are shown in

Figure 3.20. The use of this curve to determine the phase lag due to the interaction of the two modes yields Figure 3.21.

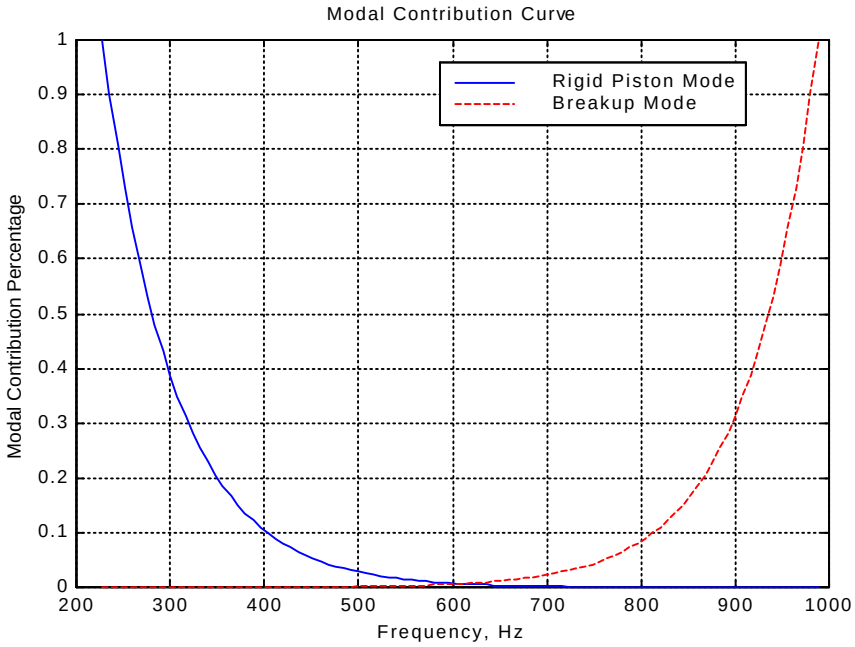

Figure 3.20 Modal contribution of rigid piston mode and 1st breakup mode of diaphragm.

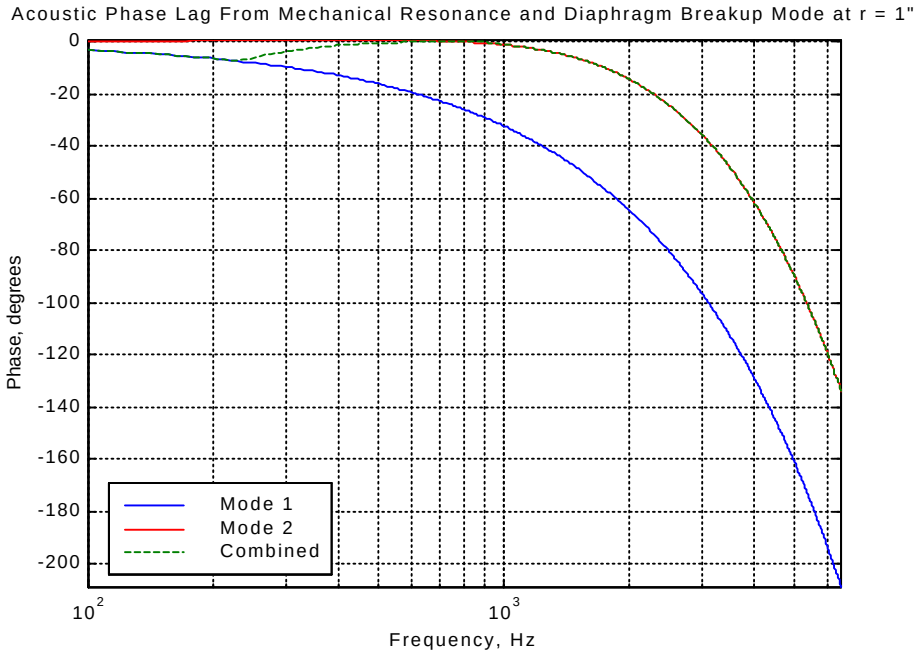

Figure 3.21 Phase lag of rigid piston mode and 1st breakup mode of diaphragm.

Figure 3.21 demonstrates the overall effect caused by the combination of both modes. At low frequencies, the phase lag is highly dependant on the mechanical mode, and towards the high frequencies the phase lag is highly dependant on the breakup mode.

In order to test the accuracy of the assumed modal contribution curves, the combined phase lag is mathematically removed from the measured data, resulting in Figure 3.22. The speaker dynamics have finally been isolated. Each of the data sets comes together, showing no dependence on the distance from the speaker diaphragm to the error microphone. In addition, the phase crosses -90° at 225 Hz and levels off at -180°, matching the speaker model's bode plot. This provides conclusive evidence that the correct modes of the speaker diaphragm have been identified and that the resulting acoustic propagation phase lag formulation is correct.

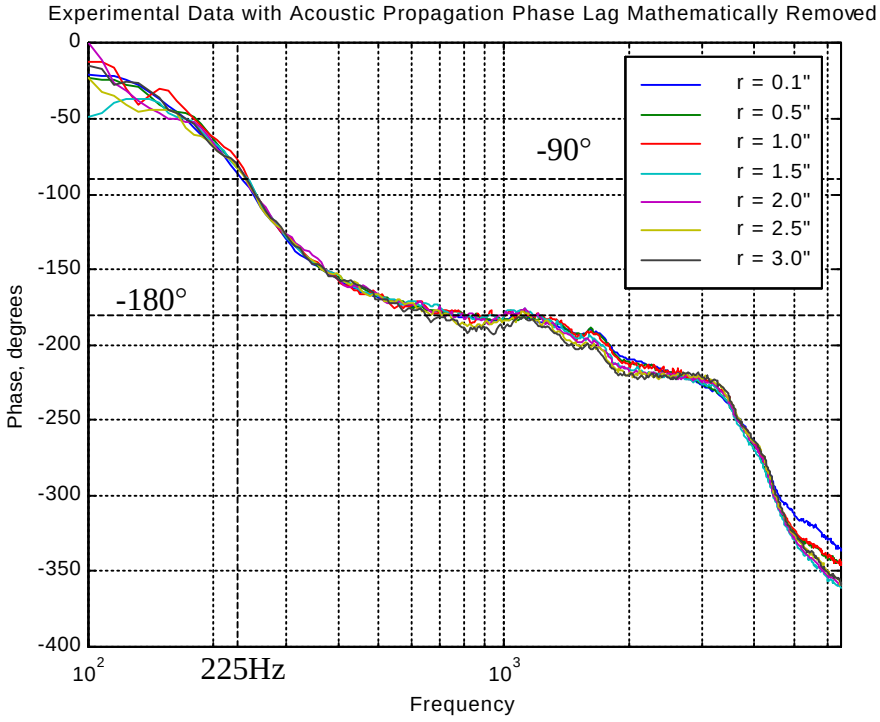

Figure 3.22 The resulting phase response after the predicted acoustic propagation phase lag has been mathematically removed from the experimentally measured data. Acoustic phase lag predicted by rigid piston mode and breakup mode combined.

Figure 3.21 and Figure 3.22 illustrate that the acoustic propagation phase lag in an ANR headset does not limit performance as much as the phase lag introduced by the speaker. The acoustic propagation contributes less than 40° of phase lag for frequencies below 3000 Hz for a propagation distance of 1 inch. The speaker, however, introduces 180° of phase lag by 1000 Hz and rolls off at an extremely fast rate as higher frequencies are observed. The acoustic propagation is responsible for only a minimal phase lag and improvements to the system should concentrate on speaker design, not the separation distance between the speaker and the error microphone.

3.11 Conclusion

The acoustic propagation delay from the control speaker to the error microphone is not the major contributor of phase lag in an ANR headset. With an accurate description of the vibration pattern on the speaker diaphragm at its first breakup frequency, the acoustic propagation phase lag was calculated and compared to the phase lag induced by the mechanical resonance of the speaker and its diaphragm. For a speaker radius of 1 inch and a microphone-speaker separation of 1 inch, the acoustic propagation phase lag at 1000 Hz was less than 10,° while the speaker created 180° of phase lag. Therefore, in order to decrease the phase lag in an ANR headset and improve system performance, the dynamic response of the speaker diaphragm should be improved.

4 CHAPTER: DIGITAL CONTROL SYSTEM

4.1 Introduction

Digital control implementation introduces complexity that does not accompany analog control. While the actual digital controller may be much more versatile in programming various control laws, more hardware is required. Once designed and built, the classic analog controller can stand alone without signal conditioning filters. The digital controller allows the versatility of reprogramming a control law without the need to rebuild the controller itself, but it usually must be accompanied by an anti-aliasing filter and may need to be accompanied by a smoothing filter. The introduction of these two filters affects the frequency response of the system and in many cases may limit the performance of the system.

Figure 4.1 Block diagram of digital control system.

The typical hardware for a digital controller is shown in Figure 4.1. The input signal (e) first passes through an anti-aliasing filter which shapes the signal spectrum, conditioning it for accurate sampling. Next, the A/D converter samples the conditioned signal and outputs a discretized approximation that becomes the input to the controller. Once again filtering the signal, the controller shapes the spectrum of the signal according to the control law. This controlled digital signal is then converted into an analog signal by the D/A converter. Finally, the output signal (u) is obtained by shaping the spectrum of the D/A output through a smoothing filter.

Properly designing a digital controller includes much more than formulating a control law. The proper control law must be accompanied by effective anti-aliasing and smoothing filters. In addition to these designs, A/D and D/A converters must be specified. Failure to adequately design any one of these components will lead to a

control system that is not optimal. Therefore, it is imperative to understand each of the listed components and the effects each has on the frequency response of the overall system.

This chapter presents the background information needed to understand each of the components in a digital control system and to begin designing a digital ANR headset. After presenting this information, criteria are derived that aid in the design of minimal phase anti-aliasing and smoothing filters for an ANR headset. In addition, MATLAB codes are developed that can be used as aids in this design process.

4.2 Anti-aliasing filter

An anti-aliasing filter is a low-pass filter used to condition the input signal for accurate sampling. Proper conditioning of the signal is mandated by the A/D converter which has an associated sampling frequency (f_s). This sampling frequency determines the Nyquist frequency (f_{nyq}).

$$f_{nyq} = \frac{f_s}{2} \tag{4.1}$$

Frequencies above the Nyquist frequency are not represented properly in the digital representation of an analog signal. Instead, they are aliased to frequencies below the Nyquist frequency. As a consequence, the sampled signal will only have a frequency spectrum from 0-f_{nyq} Hz. The anti-aliasing filter is used to help remove or minimize the amplitude of these higher frequencies that would otherwise be aliased back into the lower frequency region.

The phenomena of aliasing during the sampling process can create a problem for a digital control system. Typically, a specific range of frequencies is targeted where control performance is desired. If frequencies are allowed to alias, they may appear in this frequency range of interest. Their appearance creates an input spectrum to the controller

that is not the same as the true analog input spectrum. If this occurs, the controller is subjected to an altered open-loop disturbance spectrum and poor performance may result. Particular care must be taken to shield the frequency range of interest from aliasing effects.

Aliasing can be avoided by tailoring the frequency response characteristics of the anti-aliasing filter. Aliasing effects are a magnitude problem, and the filters magnitude response can be altered through the addition of poles and zeros to the transfer function. Each pole or zero that is added to the filter affects the roll-off rate, the pass-band and the stop-band characteristics. Equally important, they each affect the phase characteristics of the filter. It is desirable to have a brick wall filter with the cutoff frequency set at the Nyquist frequency. In addition, it is desirable to have a 0 dB pass-band followed by a significantly reduced stop-band. These criteria yield a filter that eliminates frequencies above the Nyquist frequency and does not alter the frequencies below the Nyquist where accurate sampling occurs. However, building a filter with such characteristics requires a large number of poles, each of which contributes phase lag to the filters frequency response characteristics. This conflict between the need of poles in the magnitude response and the damaging phase response resulting from these poles creates a precarious position for a designer.

In a digital ANR headset, the phase lag limits the bandwidth, performance region, of the system. The inclusion of an anti-aliasing filter necessarily adds phase lag to the headset and limits the performance. Therefore, if an anti-aliasing filter is necessary, the phase lag must be kept minimal without sacrificing it's desired attenuating characteristics.

4.3 A/D converter

There are different types of A/D converters on the market today, such as successive approximation converters and sigma delta converters. Each type uses a slightly different technique to perform the operation of sampling an analog signal. However, there are

characteristics that help define any converter: the number of bits, the maximum dynamic range, and the conversion time.

The conversion time is a time delay that is a linear function of frequency. The time delay, however, does not create a phase lag in the closed-loop response. Instead, the conversion time consumes a portion of the sampling period, limiting the number of calculations that can be carried out between samples. As a result, the conversion time limits the size of the controller that can be implemented. Therefore, a fast conversion time is required to maximize the computational abilities of the controller.

The number of bits (n) and the dynamic range (ΔV) of the converter are two closely related properties that don't affect the phase lag of a digital control system, but do have an impact on performance. The voltage resolution (ε) of a converter is defined in terms of both of these quantities.

$$\varepsilon = \frac{\Delta V}{2^n} \tag{4.2}$$

This voltage resolution represents the smallest magnitude that can be detected by the converter. This phenomena, referred to as quantization, can have a significant impact on the performance of a system if the signals that are being processed have very large variations in amplitude. These large variations lead to a big dynamic range and ultimately to a resolution that is large. As a result, small magnitudes are difficult to represent accurately.

4.4 Controller

Control theory is a very broad topic that is presented in detail in many textbooks. The scope of this chapter is not to present a theory of controls, rather to present background

information on digital control hardware as applied to an ANR headset. Therefore, the user is left to find information on designing a good control filter.

4.5 D/A converter

The D/A performs the inverse operation of the A/D. It produces an analog signal out of a digital signal received from the controller. The most widely used method for conversion is a zero order hold (zoh). This method produces a piece-wise continuous step function as an output. If this method is employed, the D/A process creates a few main areas of concern for a digital control system.

The first concern created by the D/A is the fact that the process introduces imposters. From a theoretical standpoint, the process is represented in two parts as seen in Figure 4.2. First, the input signal (u) is converted to a series of impulses (u*) through a mathematical process known as impulse modulation. Second, the series of impulses is filtered by the zoh transfer function to produce the final output (u_{held}).

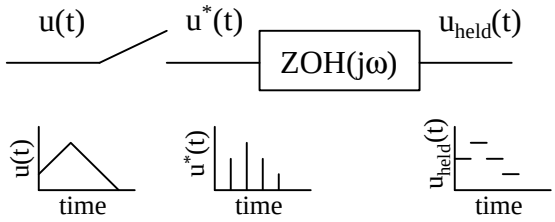

Figure 4.2 Block diagram of a D/A converter.

Mathematically, the two operations are represented by equation (4.3).

$$u_{held}(t) = \overbrace{\left[\frac{1}{T}\sum_{n=-\infty}^{\infty} u\left(j\omega - \frac{j2\pi n}{T}\right)\right]}^{\text{impulse modulation}} \overbrace{\left[\frac{1-e^{-j\omega T}}{j\omega}\right]}^{\text{zoh}} \qquad n = \ldots -2, -1, 0, 1, 2 \ldots \tag{4.3}$$

This equation demonstrates that over the frequency range (ω) the output signal (u_{held}) is a function of the input signal (u) and the sampling period (T). Further analysis reveals that if the input signal is a pure sine wave, the resulting output signal will be a combination of sine waves at well defined frequencies. The sine waves that appear at frequencies other than the input signal frequency are referred to as imposters because they are unwanted and appear merely as a result of the D/A process. These imposters appear at frequencies that can be described through the impulse modulation part of the process. Assuming the input is a sine wave, the input frequency content is a single amplitude at the input frequency (ω_o). This means that (u) is zero at every frequency (ω) other than (ω_o). The result is an expression for the pulse train (u*) representing the impulse modulated signal.

$$u^*(j\omega) = \begin{cases} 0 & \text{if} \quad u\left(j\omega - \dfrac{j2\pi n}{T}\right) \neq u(j\omega_o) \\ \dfrac{1}{T}u(j\omega_o) & \text{if} \quad u\left(j\omega - \dfrac{j2\pi n}{T}\right) = u(j\omega_o) \end{cases} \qquad (4.4)$$

$$n = \ldots -2, -1, 0, 1, 2 \ldots$$

Analyzing (4.4) results in an expression for the frequencies (ω_i) at which the impulse modulated signal will have a non-zero amplitude.

$$\omega_i = \omega_o + \frac{2\pi n}{T} \qquad n = \ldots -2, -1, 0, 1, 2 \ldots \qquad (4.5)$$

Because (n) can be positive or negative, mathematically the resulting frequencies (ω_i) may also be positive or negative. But, the negative frequencies alias about the zero frequency and appear as positive frequencies. The result is expressed mathematically by (4.6).

$$\omega_i = \begin{cases} \omega_o & \text{for} \quad n = 0 \\ n\omega_s \pm \omega_o & \text{for} \quad n > 0 \end{cases} \qquad n = 0, 1, 2, \ldots \qquad (4.6)$$

Therefore, the imposters will appear at frequencies that correspond to integer multiples (n) of the sample frequency (ω_s) plus and minus the input frequency (ω_o).

The result of the entire D/A process is illustrated in Figure 4.3 in which a sine wave is sampled and sent directly to the D/A with no control filter. The process has already created two problems. First, a single sine wave has been represented by an infinite number of sine waves. Second, an additional source of phase lag has been introduced by the zoh filter.

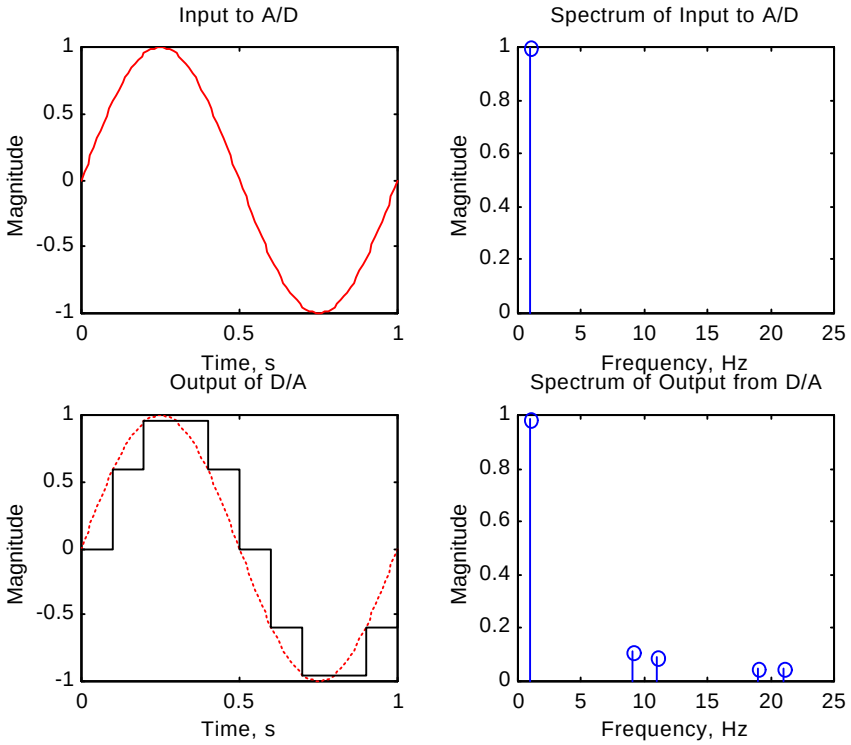

Figure 4.3 Time trace and frequency content of a signal after passing through a digital control system. The input frequency is 1 Hz and the sampling frequency is 10 Hz. No anti-aliasing filter, control filter, or smoothing filter was applied.

The third point of concern created by the D/A is the time delay due to the settling time. The converter has a time constant which governs how fast the output can track, or settle, to a voltage. As discussed previously, the A/D has a conversion time associated with it that consumes a portion of the time between samples. The settling time of the D/A

consumes more of this time period. Although the settling time is generally smaller than that of the A/D, it is still an additional source of limitation on the size of the controller.

Finally, the D/A converter suffers from the same limitations as the A/D in terms of quantization. This quantization determines the accuracy with which the converter can track the output of the controller. Again, this quantization plays a significant role when the output signals have very large variations in amplitude.

4.6 Smoothing filter

The imperfections of the D/A cause the need for a smoothing filter. This filter is used in an attempt to eliminate, or at least reduce, the magnitude of the imposters included in the composition of the piece-wise continuous step function (u_{held}) that is output by the D/A. By sufficiently decreasing the amplitude of these imposters, the step function can be forced to resemble the sine wave that the D/A is approximating. Because these imposters appear at frequencies greater than the input frequency, the smoothing filter is low-pass in design.

The response characteristics of the smoothing filter can be controlled in the same manner as the anti-aliasing filter. Through addition of poles and zeros to the filter's transfer function, the response can be tailored to meet desired specifications. Once again, it is desirable to have a 0 dB pass-band followed by a significantly reduced stop-band. For this application, however, the roll-off of the filter starts just above the frequency of the signal output by the controller. Therefore, the signal frequency is unmodified while the imposters are reduced. The requirement to reduce the high frequencies immediately above the signal frequency leads to the same problem as in the case of the anti-aliasing filter. A very steep role-off rate is essential which necessitates many poles. The addition of these poles is significant when analyzing the phase lag. A steep roll-off rate yields significant reduction of imposters, but, in terms of overall system response, the poles increase the phase lag and limit the bandwidth of the system.

In many mechanical applications, a smoothing filter is not necessary. If the sampling rate is significantly higher than the frequency range of desired performance and if the plant is low-pass in nature, the imposters introduced by the D/A process may be insignificant. This eliminates the additional phase lag created by the filter. However, this is not always possible.

In acoustic applications, such as an ANR headset, these imposters are significant. The audible range of the human ear is typically defined as 20-20,000 Hz[i]. In addition, speakers typically have responses that extend into the 20 kHz region. So, even at high frequencies, the speaker is capable of producing sound and the ear is capable of perceiving this sound. If the imposters are not reduced to a level below the minimum voltage capable of powering the speaker, the listener may hear these imposters and the controller would actually be contributing noise to the listener that would not be present with an analog controller. Therefore, the imposters must be reduced below the minimum voltage capable of powering the control speaker.

4.7 Testing/simulating a single frequency input to a digital system

To illustrate the theoretical developments thus far in the chapter, MATLAB code[ii] was written. The code tracks a sine wave as it passes through a digital control system. In reality, inputs will not usually be one sinusoid, rather they will be broad spectrums containing many sinusoids. However, analyzing one wave at a time clearly demonstrates what happens to a signal in a digital control system. In order to validate the code, an experimental study of the system was performed and compared to the output of the code. The goal of the analysis was to show that theoretical results from the code and experimental results matched. Therefore, the anti-aliasing filter, smoothing filter, and

[i] Bies et al. (1996)

[ii] ONEFREQ.M is included in APPENDIX B: MATLAB CODES

control filter were not designed optimally. The code should work for an optimal design as well as for a poor design.

It was important that the hardware characteristics exhibited in the lab matched the characteristics programmed into the MATLAB code. To accomplish this, a method was developed that defined the components of an active circuit that would yield a desired analytical transfer function and consequently a desired frf. This work has been included in APPENDIX A**.**

The system that was analyzed was constructed from available equipment and custom built filters. Transfer functions were arbitrarily chosen for the anti-aliasing filter, the smoothing filter and the control filter. Then, using the method in APPENDIX A, the circuits were constructed that would produce the desired transfer functions. The control was implemented through the use of a DSPACE board which had a zoh A/D converter. The system and its characteristics are depicted in Figure 4.4 and listed in Table 4.1.

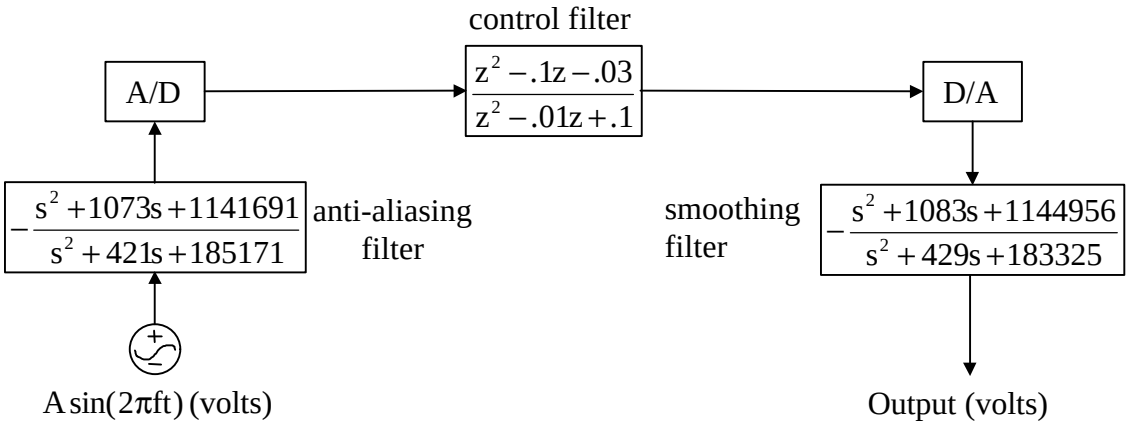

Figure 4.4 Block diagram of system used to test digital sampling theory

Table 4.1 Characteristics of system used to test digital sampling theory

Input Signal		$A\sin(2\pi ft)\,V$
A/D Converter	conversion time	$32\mu s$
	bits	12
	dynamic range	20V
Sampling Rate		1024 Hz
Anti-Aliasing Filter		$AA(s) = -\frac{s^2 + 1073s + 1141691}{s^2 + 421s + 185171}$
Controller		$C(s) = \frac{z^2 - .1z - .03}{z^2 - .01z + .1}$
D/A Converter	settling time	$6\mu s$
	bits	12
	dynamic range	20
Smoothing Filter		$SM(s) = -\frac{s^2 + 1083s + 1144956}{s^2 + 429s + 183325}$

4.7.1 Simulation with no aliasing

To illustrate what happens to a signal as it passes through a digital control system, a sinusoidal input was first studied. A low frequency ($f = 200\,Hz$) was chosen because it was below the Nyquist frequency ($fnyq = 512\,Hz$). This eliminated aliasing effects. The amplitude was arbitrarily chosen to be ($A = .707\,V$).

Figure 4.6 verifies that the anti-aliasing filter was not designed optimally. It amplified low frequencies and it passed high frequencies with only a slight attenuation. The filter provided no preventative measures against aliasing. Furthermore, this poorly designed filter did not have a 0 dB pass-band. It actually attenuated the input frequency slightly at 200 Hz. Again, this was not a concern as this was merely a test to verify the simulation and illustrate what happens to signals as they pass through a digital control system.

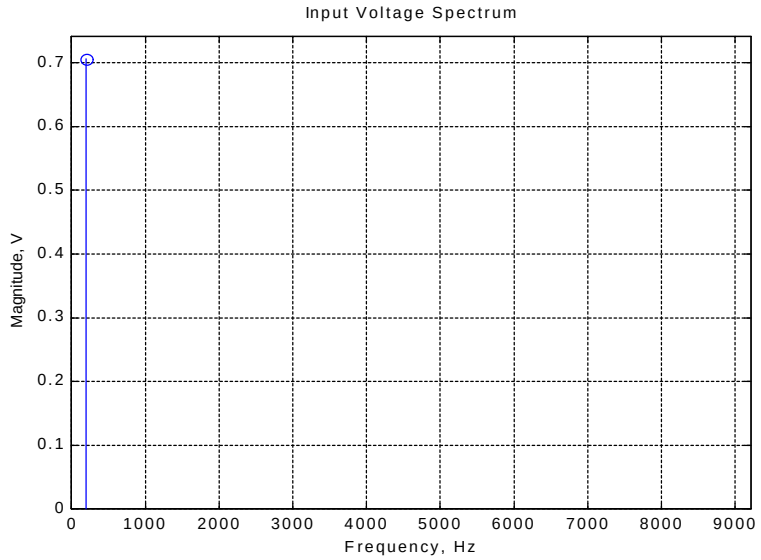

Figure 4.5 Frequency spectrum of the signal input to the test system.

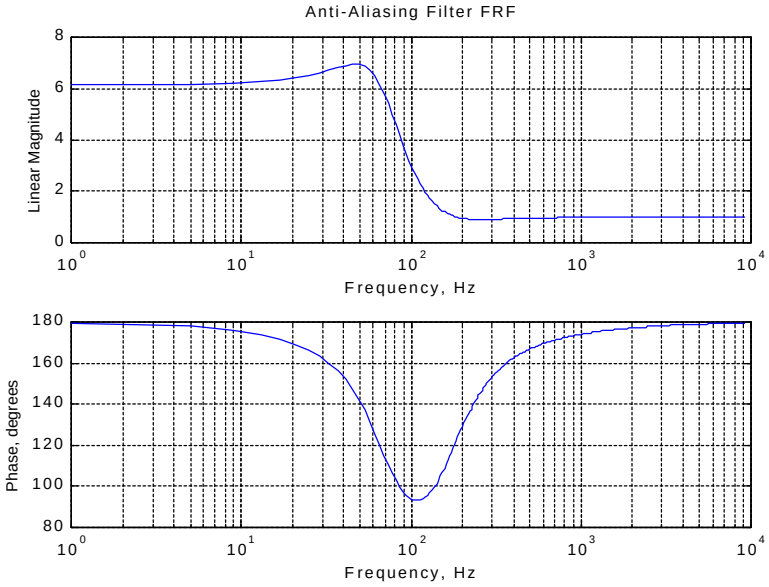

Figure 4.6 Frequency response characteristics of the anti-aliasing filter.

Figure 4.8 illustrates that, as predicted, the input signal was not aliased. The digital representation of the analog signal is accurate in this case. The only error that the sampling process has introduced is quantization error. Due to the large amplitude of the input signal chosen and the small resolution, this error is not significant.

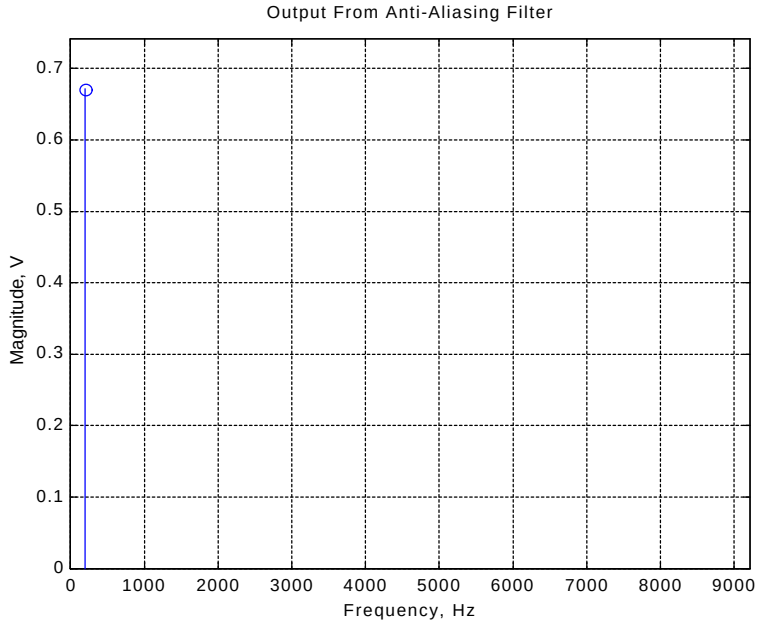

Figure 4.7 Frequency spectrum of the signal after passing through the anti-aliasing filter.

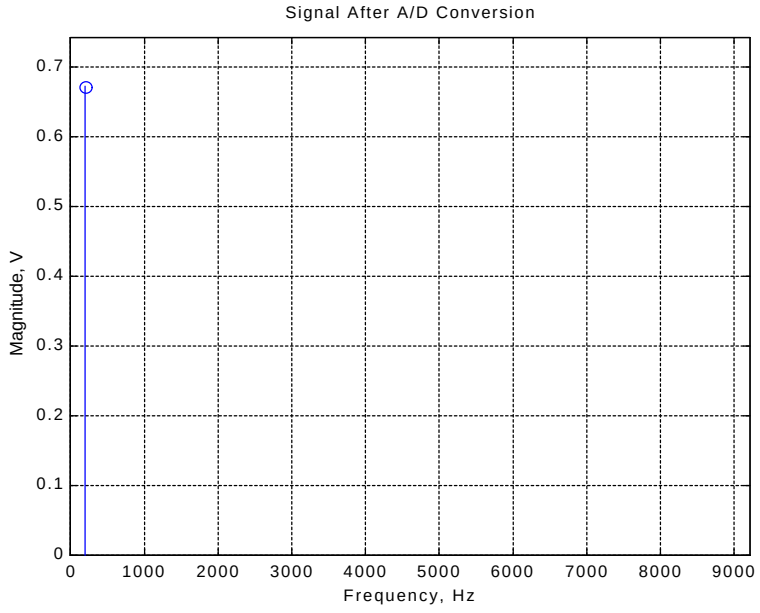

Figure 4.8 Frequency spectrum of the signal after being sampled.

A key concept of digital controls is depicted in Figure 4.9. The frequency characteristics of the digital controller are cyclic. The response is defined between 0 Hz and the sampling frequency. Then, this response is repeated during every succeeding interval corresponding to a multiple of the sampling frequency.

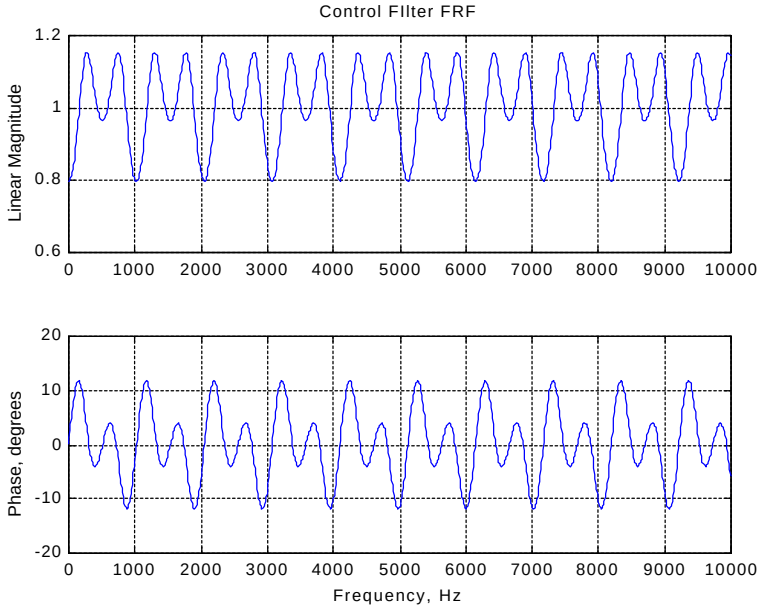

Figure 4.9 Frequency response characteristics of the controller.

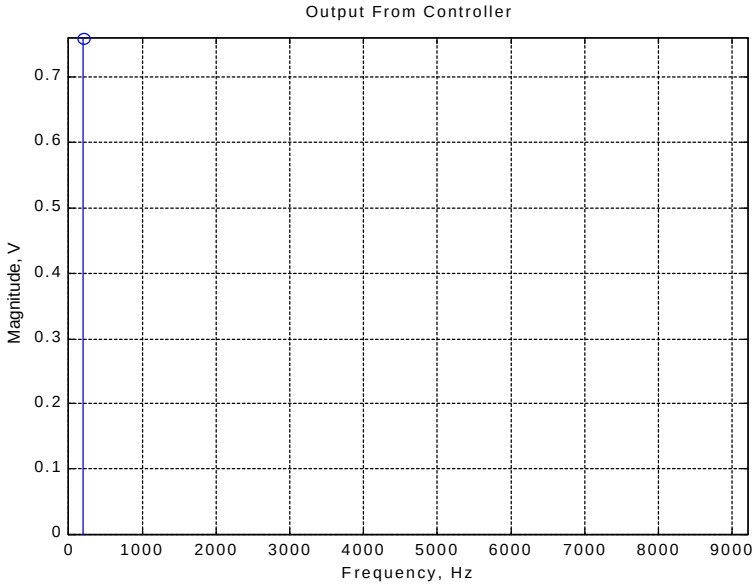

Figure 4.10 Frequency spectrum of the signal after passing through the controller.

Verification of equation (4.5), impulse modulation, is provided by Figure 4.11. The figure clearly shows that the single frequency input signal has been converted into an infinite number of sine waves at discrete frequencies. These high amplitude imposters are then attenuated by the zoh filter, Figure 4.12. It should be noted that the zoh frf vanishes at discrete frequencies corresponding to integer multiples of the sampling frequency (fs = 1024 Hz).

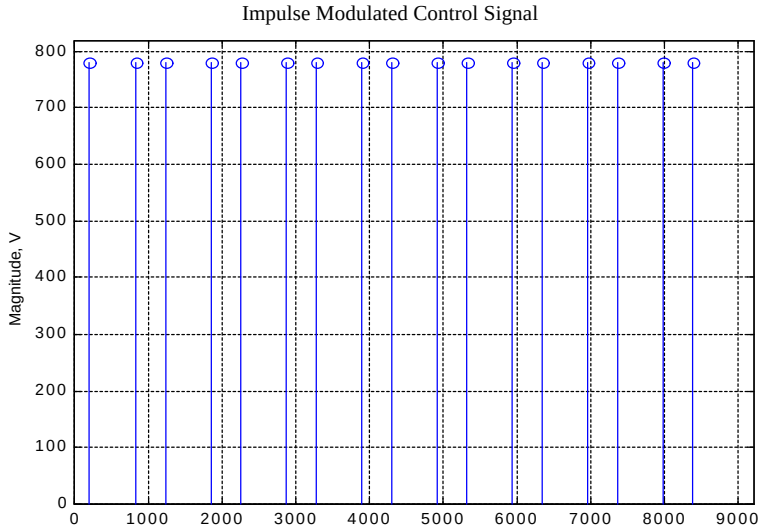

Figure 4.11 Frequency spectrum of the signal after impulse modulation.

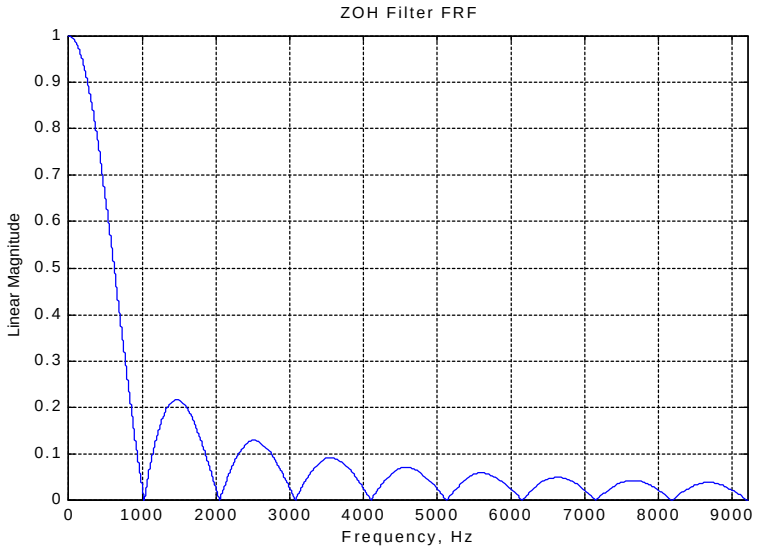

Figure 4.12 Frequency response characteristics of the zoh filter.

The zoh filter accomplishes attenuating all of the imposters, but not to a level that is acceptable in acoustic applications. The minimum voltage capable of powering the speaker is a few millivolts and, as Figure 4.13 demonstrates, the imposters are all at levels greater than a few millivolts. A low-pass smoothing filter is needed for further attenuation of the imposters.

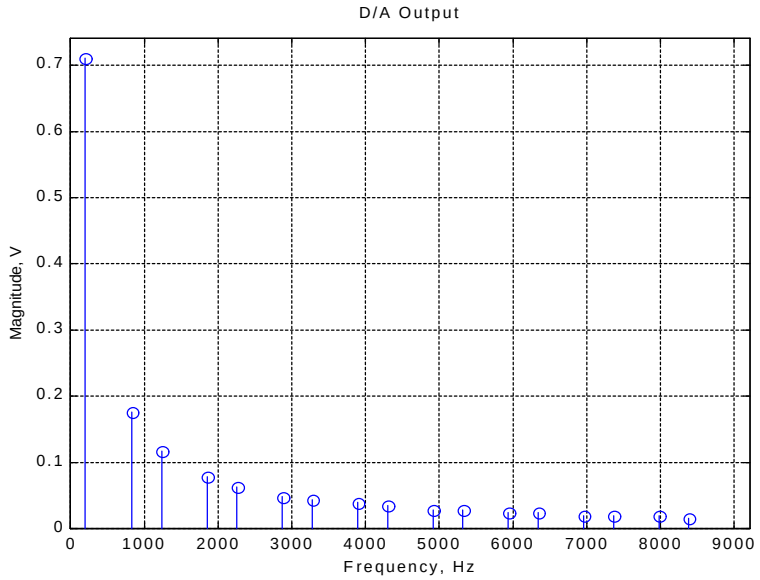

Figure 4.13 Frequency spectrum of the signal output by the D/A.

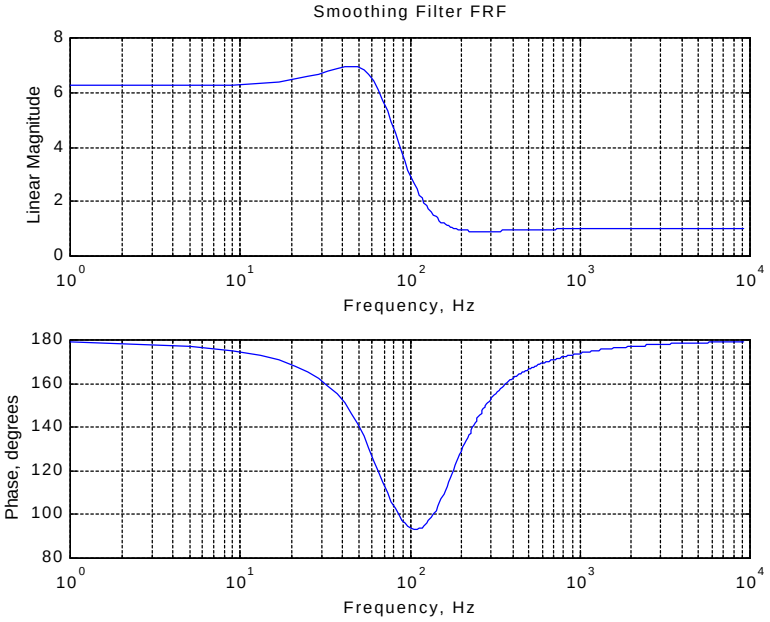

Figure 4.14 Frequency response characteristics of the smoothing filter.

Figure 4.15 supports the observation that the smoothing filter was not designed optimally. The imposters were not reduced sufficiently enough to bring them below the minimum voltage capable of powering the speaker. If this were an acoustic system, the D/A would have introduced all of these imposter signals into the original disturbance field, creating a louder disturbance.

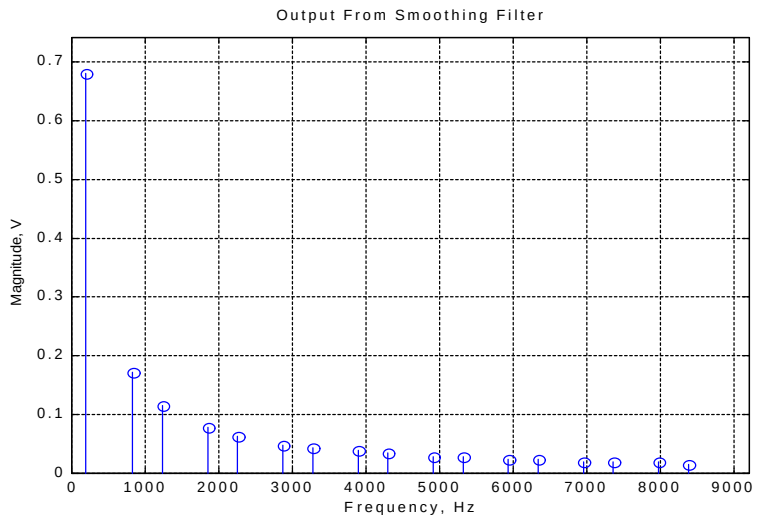

Figure 4.15 Frequency spectrum of the signal after passing through the smoothing filter.

4.7.2 Experimental verification of simulation

Figure 4.5 through Figure 4.15 match the theory very well, but do they match reality? The answer was obtained by comparing the frequency and magnitude of each imposter as predicted by the simulation with the values measured experimentally. Table 4.2 provides this comparison for the first six imposters.

The simulation results compare favorably to the experimental results. The frequencies predicted by the simulation match the frequencies observed in the lab exactly as the imposters appear at discrete frequencies predicted by (4.5). On the other hand, the magnitudes do not match exactly. Positively, the magnitudes of the imposters as predicted by the simulation are all within 9 mV of what was measured experimentally. Negatively, a discrepancy is seen between the input signal frequency as predicted by the

simulation and as measured. The simulation predicts an amplitude that is 28 mV greater at 200 Hz than what was measured. The discrepancy is explained by the lack of accuracy in the filter transfer functions that were entered into the simulation. The transfer functions all matched the measured transfer functions of the hardware closely, but not exactly. This error between the actual filter characteristics and the characteristics entered into MATLAB is greatest in a region of transition for the filter. In this particular case, the transfer functions of the anti-aliasing filter and the smoothing filter are approximately flat above 300 Hz. Therefore, at higher frequencies the error is small. Hence, the imposter magnitudes should match the simulation more closely than the low frequency sine wave of interest.

Table 4.2 Comparison of imposters between simulation and experiment. *Note that 200 Hz is the input frequency and not an imposter.

Simulation Input = .707sin(2π200t)		Experimental Input = .711sin(2π200t)	
Imposter Frequency (Hz)	Imposter Magnitude (V)	Imposter Magnitude (V)	Imposter Frequency (Hz)
200*	0.681	0.653	200*
824	0.173	0.164	824
1224	0.116	0.111	1224
1848	0.078	0.074	1848
2248	0.063	0.061	2248
2872	0.049	0.048	2872
3272	0.044	0.042	3272

From these results, it was concluded that the simulation does predict reality well for the scenario that was tested. But, this scenario did not involve aliasing. So, as a final check of the code and theory, a test was performed that involved aliasing.

4.7.3 Experimental verification with aliasing

The final test to verify the code and theory was run with the exact same hardware as the previous test. In this case, however, the input signal was different. An amplitude of $(A = 1\,V)$ was applied at a frequency of $(f = 824\,Hz)$. This meant that the input frequency should alias back to 200 Hz.

Table 4.3 Comparison of imposters between simulation and experiment.

Simulation Input = 1sin(2π824t)		Experimental Input = 1.001sin(2π824t)	
Imposter Frequency (Hz)	Imposter Magnitude (V)	Imposter Magnitude (V)	Imposter Frequency (Hz)
200	0.993	0.951	200
824	0.250	0.248	824
1224	0.170	0.162	1224
1848	0.112	0.108	1848
2248	0.093	0.089	2248
2872	0.073	0.069	2872
3272	0.063	0.061	3272

The results in Table 4.3 follow the same trend as the previous results. Due to the incapability of the actual filter to reproduce an exact transfer function, the low frequency results are slightly less than they should be. The higher frequency results agree to within a few millivolts. This supports the accuracy of the code and theory.

4.8 Simulating the multi-frequency case

In the previous section, the case in which the input was a single sine wave was studied. In the case of an ANR headset, however, the sound field will always be a spectrum containing many frequencies. Therefore, the previously discussed code would not be

applicable. But, by utilizing the knowledge gained from analyzing the simple single frequency input case, code[i] was written that is capable of processing broad spectrum inputs. The theory applied to this code does not differ from the theory in the single frequency code. The only difference is that it processes many single frequencies at one time. This code can be used as a final step in the design process to predict what will happen open-loop if a certain digital control system is implemented and a certain input disturbance field is present.

4.9 Designing minimal phase filters

Now that the theory of a digital control system has been tested, a digital ANR headset design procedure is developed. It was demonstrated that the anti-aliasing filter and smoothing filter each contribute phase lag to a digital system that can be controlled by tailoring the transfer functions of the filters. These components, therefore, are the most logical candidates to focus on when attempting to minimize the phase lag in a digital ANR headset. The knowledge of how aliasing and the creation of imposters affect the output signals frequency spectrum forms the basis of the design procedure which minimizes the amount of phase lag introduced by the digital control hardware. The procedure formulates minimal phase filters that eliminate aliasing and reduce imposters to a voltage level below the minimum voltage capable of powering the speaker.

4.9.1 Designing a minimal phase anti-aliasing filter

There are three main characteristics that must be described to fully define an anti-aliasing filter: a frequency range for the pass-band, the gain across the pass-band, and the roll-off rate. Analysis of what happens to a signal as it is sampled yields design criteria that define each of these three characteristics. The criteria result in a minimal phase filter that

[i] FRF.M.M is included in APPENDIX B: MATLAB CODES

eliminates aliasing, but does not affect the frequency region where system performance is desired.

The pass-band of a minimal phase anti-aliasing filter is defined by choosing a frequency range for control performance, the control region. For instance, control might be desired for the frequency range $0\text{-}1000\,\text{Hz}$. This range of frequencies where system performance is desired, $0\text{-}1000\,\text{Hz}$, defines the pass-band of the filter. Over this frequency range, the anti-aliasing filter should not alter the input spectrum. Altering of the input spectrum in this control region would yield an input to the controller that does not accurately represent the analog input signal.

The pass-band gain for a minimal phase anti-aliasing filter is 0 dB. This characteristic, again, stems from the necessity for the anti-aliasing filter to not alter the input signal in the control region. A $0\,\text{dB}$ pass band across the frequencies where control performance is desired subjects the controller to an accurate representation of the analog input signal across this same frequency range.

The third and final design characteristic, the roll-off rate, of a minimal phase anti-aliasing filter is formulated based on the process of aliasing. To avoid altering the control region, frequencies must not be allowed to alias back into this frequency range. This is accomplished by ensuring that the spectrum input to the A/D converter does not contain signal content with magnitude greater than the resolution of the A/D converter at frequencies that will alias into this range. Defining the highest frequency in the control region as (x), the lowest frequency that aliases into the desired control region is defined as $(f_s - x)$. Figure 4.16 demonstrates this concept and illustrates that if the magnitude of the input spectrum at frequencies greater than $(f_s - x)$ is below the A/D resolution, no aliasing will occur into the control region.

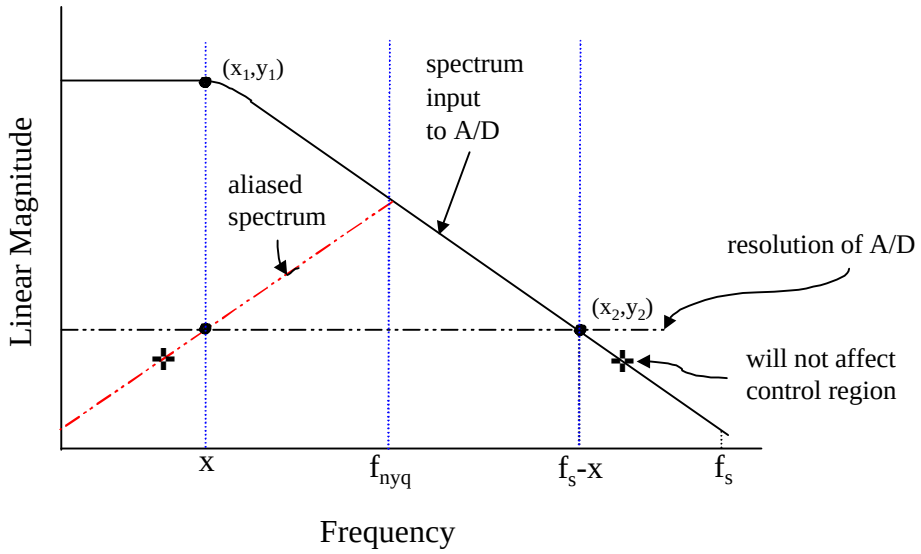

Figure 4.16 Aliasing during the A/D process. Signal content at $(f_s\text{-}x)$ aliases back to (x). Signal content with magnitude less than resolution of A/D is ignored.

The pass-band, 0-x Hz, has already been defined as having zero gain and zero slope, so all that remains to mathematically describe the minimal phase anti-aliasing filter is an equation describing the roll-off starting at the frequency (x). The minimal slope of the filter (M_{AA}) is defined by utilizing two points on the magnitude response curve. The 'rise' of the slope is given by the difference in magnitude (y_1) at the first frequency (x_1) and the magnitude (y_2) at the second frequency (x_2) while the 'run' is determined by the difference in the two frequencies.

$$M_{AA} = \frac{\text{rise}}{\text{run}} = \frac{y_2 - y_1}{x_2 - x_1} \tag{4.7}$$

The first frequency (x_1) is the highest frequency in the control region, (x). At this frequency, the magnitude (y_1) of the filter should be 0 dB, or 1 on a linear scale. The second frequency (x_2) has been defined as the first frequency that aliases back into the control region, $(f_s - x)$. At this frequency, the magnitude (y_2) of the filter must be determined.

The magnitude of the output signal from the filter is simply the linear input ($V_i(f)$) multiplied by the linear magnitude response ($AA(f)$) of the filter at each frequency. Additionally, the magnitude of the output signal from the filter at $(f_s - x)$ must be less than the resolution of the A/D converter (ε_{ad}). Therefore, (4.8) and (4.9) define the magnitude response (y_2) of the anti-aliasing filter at $(f_s - x)$.

$$\varepsilon_{ad} = AA(f_s - x)V_i(f_s - x) \tag{4.8}$$

$$y_2 = AA(f_s - x) = \frac{\varepsilon_{ad}}{V_i(f_s - x)} \tag{4.9}$$

Substitution for (x_1), (x_2), (y_1), and (y_2) into (4.7) results in the expression for the slope of the anti-aliasing filter.

$$M_{AA} = \frac{\left[\frac{\varepsilon_{ad}}{V_i(f_s - x)} - \frac{1}{1}\right]}{(f_s - x) - x} \quad \left(\frac{\text{Lin. Mag.}}{\text{Frequency}}\right) \tag{4.10}$$

Typically, filter roll-off rates are defined in units of $(\text{dB}/\text{decade})$. Through some simplification, (4.10) is written in such a manner that yields a slope in this common form.

$$M_{AA} = \frac{20\log_{10}(\varepsilon_{ad}) - 20\log_{10}(V_i(f_s - x))}{\log_{10}(f_s - x) - \log_{10}(x)} \quad \left(\frac{\text{dB}}{\text{dec.}}\right) \tag{4.11}$$

The slope in (4.11) and the described pass-band define a filter response that does not alter the control region and that ensures the input spectrum at $(f_s - x)$ will not alias back into the control region. When the input signal is low-pass in nature, this design procedure is effective. However, this criteria is not stringent enough for input spectrums that are not low-pass in nature. If the input spectrum at frequencies higher than $(f_s - x)$ has a greater magnitude than the spectrum at $(f_s - x)$, care must be taken to ensure that the magnitudes

at these higher frequencies are attenuated to a level below the resolution of the A/D. This is accomplished by scanning the frequencies above ($f_s - x$) that will alias into the control region and checking the output of the calculated filter. If a frequency is found that has an output signal with a greater magnitude than (ε_{ad}), the slope must be increased sufficiently to attenuate the signal below (ε_{ad}). Increasing the slope will not adversely affect the lower frequencies that created the basis for the first minimal slope. In fact, the increased slope will attenuate these lower frequencies even more.

This last piece of information complicates matters as it can not be quantified and added to the design equations. Instead, code must be written to ensure that the last condition is met. This reduces the formulation of the anti-aliasing filter slope to an initial calculation. The formulation provides an initial design that must be checked for accuracy. Therefore, the magnitude response characteristics of the minimal phase anti-aliasing filter are designed appropriately by applying the following procedure.

1) Define a frequency range of desired control 0-x Hz.
2) Define the filter's pass-band as 0-x Hz with a gain of 0 dB.
3) Define the roll-off rate with the cut-off frequency set at (x):

$$M_{AA} = \frac{20\log_{10}(\varepsilon_{ad}) - 20\log_{10}(V_i(f_s - x))}{\log_{10}(f_s - x) - \log_{10}(x)} \left(\frac{dB}{dec.}\right)$$

4) If necessary, increase the slope of the filter such that higher frequencies that would alias into the control range are attenuated below the resolution of the A/D converter.

4.9.2 Designing a minimal phase smoothing filter

As in the design of the anti-aliasing filter, there are three main characteristics that must be described to fully define a smoothing filter: a frequency range for the pass-band, the gain across the pass-band, and the roll-off rate. Analysis of where imposters are introduced as a digital signal is passed through a D/A converter yields design criteria that define each

of these three characteristics. The criteria result in a minimal phase filter that eliminates the negative effects of imposters, but does not affect the frequency region where system performance is desired.

The pass-band and pass-band gain of a minimal phase smoothing filter are defined in an identical fashion to that of the anti-aliasing filter. The smoothing filter should not alter the magnitude of the signal output by the controller at frequencies in the control region. Altering the magnitudes at these frequencies, would effectively create an altered control law. Therefore, the pass-band is defined as 0-x Hz with a gain of 0 dB.

The roll-off rate of the smoothing filter is defined by choosing two points on the frequency response plot and formulating the slope.

$$M_{sm} = \frac{\text{rise}}{\text{run}} = \frac{y_2 - y_1}{x_2 - x_1} \tag{4.12}$$

The first magnitude (y_1) and frequency (x_1) used in defining the smoothing filter slope are exactly the same as in the anti-aliasing filter's development. The linear magnitude response of the filter is unity below the highest frequency (x) in the control region.

$$y_1 = 1 \text{ and } x_1 = x \tag{4.13}$$

The second magnitude (y_2) and frequency (x_2) require more attention. The sampling process creates a digital signal that has a frequency spectrum from 0-f_{nyq} Hz. This has significant implications for the smoothing filter. The D/A converter approximates the digital spectrum, but in the process introduces imposters which appear at frequencies higher than the Nyquist frequency. The smoothing filter must eliminate these imposters. Eliminating these imposters is accomplished by reducing their magnitudes below the minimum power voltage of the speaker. This voltage level (V_{spk}) defines the second point used in the smoothing filter's slope formulation, (y_2) and (x_2). It is required that the output signal level at the Nyquist frequency $V_o(f_{nyq})$ be at most (V_{spk}) volts, thus

enforcing the rule that frequencies above the Nyquist, which appear solely due to the D/A process, are reduced to an insignificant level.

In order to define what the output signal from the smoothing filter will be, an assumption must be made. It is assumed that the control system includes a well designed anti-aliasing filter that effectively eliminates any aliasing. Operating under this assumption and defining the linear magnitude response of the controller (C(f)), zoh filter (ZOH(f)), and smoothing filter (SM(f)) respectively, leads to the expression for the output of the smoothing filter ($V_o(f_{nyq})$) at the Nyquist frequency.

$$V_o\left(f_{nyq}\right) = \frac{1}{T} V_i(f_{nyq}) AA(f_{nyq}) C(f_{nyq}) ZOH(f_{nyq}) SM(f_{nyq}) \tag{4.14}$$

Applying the criteria that $V_o(f_{nyq})$ should be equal to or less than the defined minimal voltage (V_{spk}) results in the expression for the smoothing filter's linear magnitude response (y_2) at the Nyquist frequency (x_2).

$$SM(f_{nyq}) = \frac{V_{spk} T}{V_i(f_{nyq}) AA(f_{nyq}) C(f_{nyq}) ZOH(f_{nyq})} \tag{4.15}$$

Substituting (y_1), (x_1), (y_2), and (x_2) into the general slope formula results in the expression for the minimal phase smoothing filter's slope (M_{sm}).

$$M_{sm} = \frac{\left[\frac{V_{spk} T}{V_i(f_{nyq}) AA(f_{nyq}) C(f_{nyq}) ZOH(f_{nyq})} - \frac{1}{1}\right]}{(f_{nyq} - x)} \quad \left(\frac{\text{Lin. Mag.}}{\text{frequency}}\right) \tag{4.16}$$

Converting the units of this expression and applying the fact that ($ZOH(f_{nyq}) = 2/f_s\pi$) yields the final equation representing the slope of the minimal phase smoothing filter that is required to eliminate imposters above the Nyquist frequency without altering the control region.

$$M_{sm} = \frac{20\log_{10}(\pi V_{spk}) - 20\log_{10}[2V_i(f_{nyq})AA(f_{nyq})C(f_{nyq})]}{\log_{10}(f_{nyq}) - \log_{10}(x)} \quad \left(\frac{dB}{dec.}\right) \qquad (4.17)$$

As in the case of the anti-aliasing filter, this is the minimal slope and it is based upon the fact that the input spectrum to the smoothing filter is low-pass in nature. Therefore, higher frequency imposters must be checked to ensure their reduction below the level (V_{spk}). If they are not reduced to this level, the slope must be increased accordingly to achieve this attenuation. The lower frequency imposters will once again not be adversely affected. They will be attenuated even more than required. Therefore, the magnitude frequency response characteristics of the minimal phase smoothing filter are designed by applying the following steps.

1) Define a frequency range of desired control 0-x Hz.
2) Define the filter's pass-band as 0-x Hz with a gain of 0 dB.
3) Define the roll-off rate with the cut-off frequency set at (x):

$$M_{sm} = \frac{20\log_{10}(\pi V_{spk}) - 20\log_{10}[2V_i(f_{nyq})AA(f_{nyq})C(f_{nyq})]}{\log_{10}(f_{nyq}) - \log_{10}(x)} \quad \left(\frac{dB}{dec.}\right)$$

4) If necessary, increase the slope of the filter such that higher frequency imposters are attenuated below the minimum drive voltage defined as (V_{spk}).

4.9.3 MATLAB code to calculate filter characteristics

The previous design procedures have been implemented via MATLAB code*. The code calculates the anti-aliasing and smoothing filter slopes as defined by (4.11) and (4.17). As input, the program requires an input spectrum, a control frequency range, a sampling rate, the characteristics of the A/D and D/A converter, and a control filter transfer function. The program returns the minimum slope of an anti-aliasing filter that

* DES.M is included in APPENDIX B: MATLAB CODES

eliminates aliasing and the minimum slope of a smoothing filter that attenuates imposters below the minimum speaker power voltage..

4.9.4 Phase considerations of minimal phase filters

The magnitude responses of the anti-aliasing and smoothing filters have been designed such that minimal slopes were generated. These minimal slopes inherently minimize the phase lag contribution of the filters. A shallower roll-off rate requires fewer poles than a steep roll-off rate.

4.10 Conclusion

This chapter provided background information on the components of a digital control system. After studying each component, it was determined that the anti-aliasing and smoothing filters were the most logical candidates to focus on when minimizing the phase lag of a digital control system. The information on aliasing and imposters was applied to the process of designing anti-aliasing and smoothing filters which add the least amount of phase lag to a system.

To aid in this design process, three MATLAB codes were written. The first code is a learning tool that demonstrates what happens to a single frequency input signal as it passes through a digital control system. A second code builds upon the information used in the first code and demonstrates what happens to an entire frequency spectrum as it passes through a digital control system. Finally, the third code designs the magnitude response characteristics of minimal phase anti-aliasing and smoothing filters for a given input spectrum. Additionally, the code simulates the open-loop results of a digital control system design.

5 CHAPTER: DESIGN & PERFORMANCE OF AN ANR HEADSET

5.1 Introduction

The previous chapters provided background information necessary to understanding the basic principles of operation of a digital ANR headset. They also gave some guidelines to be used in the design of such a system. In order to test the information and strategies discussed, a plant was constructed that represented one half of a headset. In reality, a headset will have a dual channel configuration that controls the noise at each ear separately, thus analyses may be performed on one of the two channels and applied to the second channel.

Testing of the prototype single channel headset was carried out on four different hardware configurations. A digital controller was designed and closed-loop system performance was measured for only a minimal phase anti-aliasing filter, only a minimal phase smoothing filter, and for both a minimal phase anti-aliasing filter and a minimal phase smoothing filter. The final test consisted of comparing closed-loop performance with only a minimal phase smoothing filter and with only an elliptic smoothing filter.

5.2 Single channel headset - plant identification

The prototype single channel headset that was constructed consisted of an audio amplifier, a control speaker, an error microphone, and an amplifier for the microphone signal. Due to the complexities of the speaker that were presented earlier, modeling the plant was not practical. For this reason, the system's frequency response was determined experimentally and is shown in Figure 5.1.

The figure demonstrates that the plant is not ideal for low frequency controller design. First, the plant is not low-pass in nature over the frequencies illustrated. The response increases in magnitude up into the 15,000 Hz region. Second, due to the large number of

higher frequency diaphragm resonances, the phase roll-off is extreme. The combination of these two factors indicates that control of the low frequency region will be difficult to achieve without significant spill-over, amplification in the closed-loop frf at frequencies outside the bandwidth of desired control. In addition to these poor characteristics, the figure demonstrates that positive feedback should be implemented for low frequency control applications. The plant's phase lies between 0° and -180° below 1000 Hz and the positive feedback phase cross-overs are at 0° and -360°.

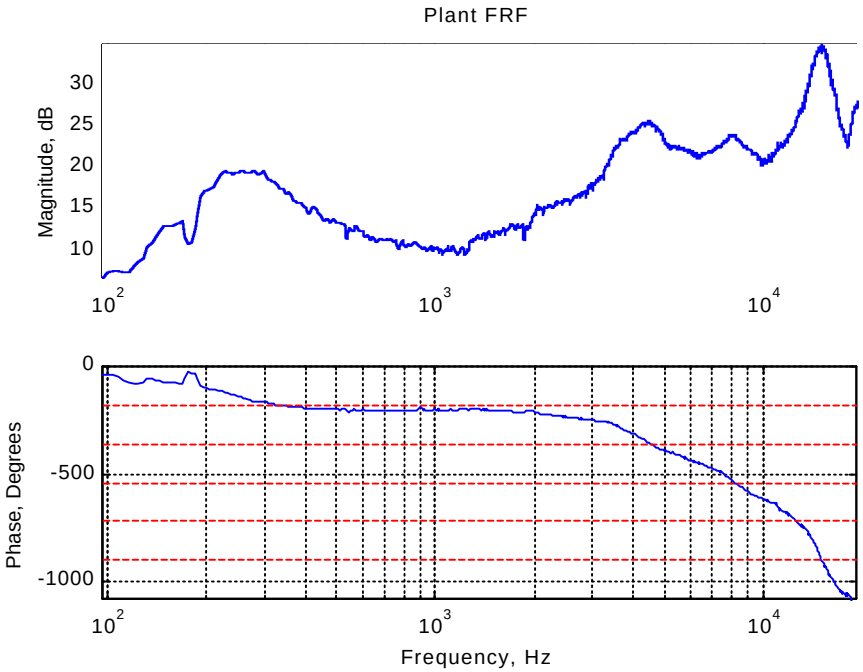

Figure 5.1 Prototype plant open-loop frequency response. The dotted lines represent phase cross-overs where stability is determined (±180n n = 0,1,2,3...).

5.3 Disturbance noise field

The design method formulated in the previous chapter required knowledge of the disturbance noise field. A disturbance speaker was driven with a 700 Hz sine wave creating a tonal noise field in testing the first three hardware configurations: only a minimal phase anti-aliasing filter, only a minimal phase smoothing filter, and both a minimal phase anti-aliasing filter and a minimal phase smoothing filter. The tonal

disturbance was chosen because of its simplicity and clarity in illustrating system performance. The combination of the disturbance tone at 700 Hz with low level ambient background noise resulted in the noise field depicted in Figure 5.2.

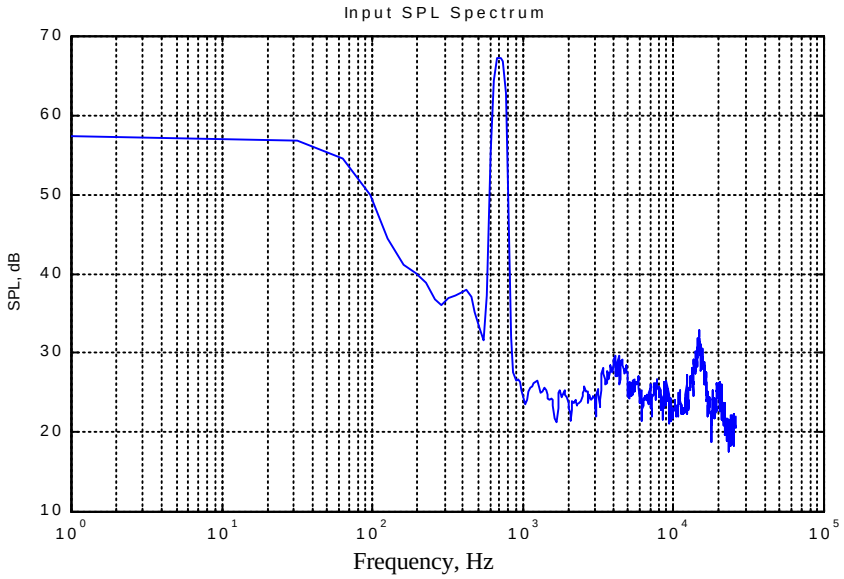

Figure 5.2 Sound pressure levels in the disturbance noise field for the first three tests.

5.4 Signal levels entering the A/D and supplied to the control speaker

The resolution of the A/D converter, power limitations of the control speaker, and saturation voltages of the anti-aliasing and smoothing filters created the necessity to analyze the signal levels throughout the system. The 12-bit A/D converter had a 20 V dynamic range resulting in a 5 mV resolution. Ambient noise fields in the lab, however, only induced microphone signal levels in the micro-volt range. In order to overcome this mismatch, an amplifier was used to boost the error microphone's signal into the 1-5 V range. By increasing the level of the input to the controller, the output from the controller and the supply voltage to the control speaker were also increased. Thus, it was necessary to check the voltage level at the output from the D/A to ensure that the control speaker would not be excited with signal levels greater than the rated speaker power. Finally, it was ensured that the signal levels at the inputs to the anti-aliasing and smoothing filters would not saturate these devices.

In addition to hardware, the disturbance noise spectrum influenced the values of the gains. For the disturbance noise field in Figure 5.2, the dc signal level of the microphone is approximately 20dB(SPL) greater than the signal level at 300 Hz. This high signal level in the low frequency range limits the amount of gain that can be used in the system. If the tone in Figure 5.2 were not present and control of higher frequency noise was desired, the important high frequency disturbance signals would be amplified 20dB less than they could be and performance would be adversely affected.

5.5 Characterization of the DSP board

The final hardware characterization that was carried out was on the DSP board being used to sample the signals, implement the control law, and DAC out the control signal. Like any piece of hardware, the board had certain limitations on how fast it could operate and how much computation could be done between sample times. A maximum sample rate of approximately 17 kHz was achievable with a control law consisting of an 8^{th} order numerator (4 complex zeros) and an 8^{th} order denominator (4 complex poles). If a larger controller was desired, the sample rate had to be reduced. This had an important impact on the designs carried out. The controllers were not optimal. In each case, however, they provided the best control and least amount of spill-over possible, while using no more than an 8^{th} order numerator and an 8^{th} order denominator.

5.6 Closed-loop performance and evaluation criteria of the headset

Having identified the plant and the disturbance field, the last step before designing the control law was to define the performance goals and evaluation criteria. The primary objective was to control the region from 0-1000 Hz, but more specifically to reject the tone at 700 Hz. In addition, it was desirable to limit the amount of spill-over experienced by regions next to the control bandwidth. However, in this case, with the limited size of the controller available, certain levels of spill-over were sacrificed to achieve good low frequency control. As a means to evaluate these performance goals, three metrics were

introduced: analog comparison, qualitative acoustic masking, and quantitative acoustic bandpower measurements.

5.6.1 Comparison to an analog controller with the same control law

The overall, bottom-line evaluation of a particular design can be done rather simplistically in theory. Analog headsets have been built and form the basis for comparison at this stage in the development of digital ANR technology. If a headset can be built digitally that performs as well as its analog counterpart, the versatility of the digital system will most certainly lead to it's replacing the analog headset. Therefore, the designs were evaluated on how well they matched the predicted performance of an analog controller with the same control law. This provided a means for assessing how much of an impact the anti-aliasing and smoothing filters had on performance.

5.6.2 Acoustic masking

In addition to control performance, each test case was qualitatively evaluated on acoustic performance by introducing masking. The premise for acoustic masking is derived from the idea that a human's ear sometimes can't distinguish a signal at one frequency when a noise, a masker, is present at a different frequency. The line-busy hypothesis attributes this masking concept to the notion that relevant neurons are too busy responding to the masker to respond properly to the signal.

The masking level is defined as the signal level at which neurons are capable of responding adequately to the signal minus the masker's level at the same point. The masking level at a given frequency depends on the relative relationship between the masking frequency, or frequencies, and the signal frequency. Due to asymmetries in the physical composition of the ear, masking does not occur equally in the forward direction, low frequencies masking higher frequencies, and the backward direction, higher

frequencies masking lower frequencies. Figure 5.3, taken from Boff et. al. (1986), illustrates this point for a narrow band masker and a tone signal.

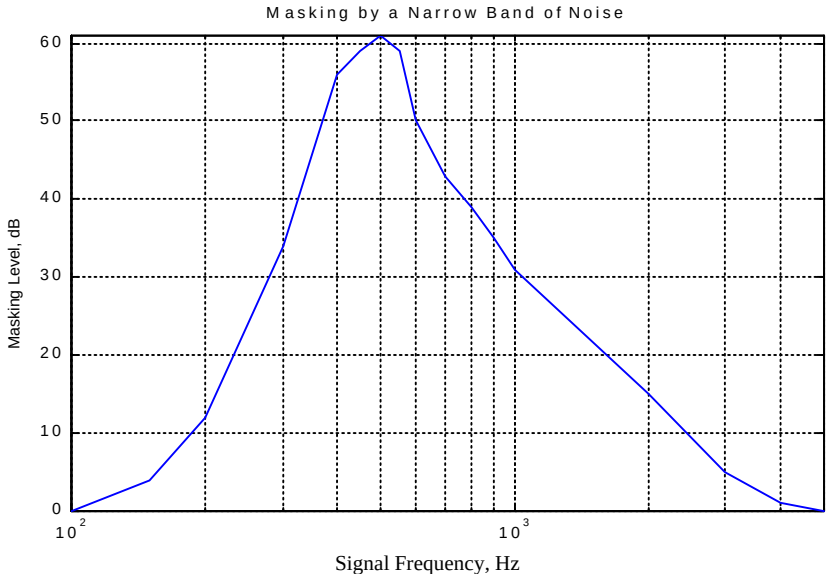

Figure 5.3 Masking level of a narrow band of noise 90Hz wide centered at 410Hz. Note that masking of lower frequencies by higher frequencies is not as effective as masking of higher frequencies by lower frequencies.

According to the figure, if the signal is present at 100 Hz there is no masking effect. The narrow band of noise centered around 410 Hz does not affect what the ear hears at 100 Hz . The steep slope of the curve below 410 Hz suggests that as long as the tone is below the narrow band of noise, masking will be ineffective. However, the gradual slope on the higher frequencies suggest that the narrow band of noise significantly masks forward. The tone has to be increased to approximately 5000 Hz before the masking effects of the low frequency noise become negligible.

This concept of masking is important in the evaluation of an ANR headset. This information illustrates the concept that spill-over regions in a design affect the listeners ability to distinguish signals at other frequencies. In all of the tests cases in this thesis, and in most practical cases, low frequency regions are controlled. This bodes well for an ANR headset as masking of the lower frequencies by the higher frequency spill-over is not effective. However, the spill-over in these regions is still a concern.

5.6.3 Acoustic bandpower

Finally, to quantitatively evaluate the performance of each design, acoustic bandpower was utilized. Bandpower is a measure of how much acoustic energy is present over a given range of frequencies. By measuring the bandpower in the disturbance noise field and then the bandpower in the controlled noise field, a measure of how much the acoustic energy was increased or decreased was formulated. An optimal design would result in a decreased bandpower in the control region and no change over the rest of the spectrum.

5.7 Control design procedure - a final MATLAB code

Having characterized all of the hardware, the plant, performance criteria, and the necessary gains in the system, the final step was to design the control law and necessary filters for each test case. This was accomplished through use of the codes discussed in the previous chapter with the addition of one final code.

Code[i] was developed that would apply a control transfer function and utilize plant open-loop frequency response data to plot the open-loop and closed-loop system frequency response. Through this code, the controller may be heuristically designed by shaping the closed-loop response of the system. This code, however, was written for the design of an analog controller. In reference to the design performance section, it was desired to design an analog controller and then achieve equal performance through a digital implementation. Therefore, all designs were done using this analog code, and the resulting controllers were converted to a digital difference equation. This allowed for the analog closed-loop performance to be predicted and plotted verses the closed-loop performance measured with the digital controller implemented.

[i] COMP.M is attached in APPENDIX B: MATLAB CODES

5.8 Three design test cases

The difference between the analog controller and the digital controller is the presence of the anti-aliasing and smoothing filters. To illustrate the effects of anti-aliasing and smoothing filters on closed-loop performance, three test cases were studied. The first test case consisted of implementing only a minimal phase anti-aliasing filter, and the second test employed only a minimal phase smoothing filter. For both cases, the same controller was used. The third test combined the use of both minimal phase filters and therefore, required an altered control law to accommodate the additional phase lag. In all three tests cases, the maximum sampling rate achievable was 17024 Hz, resulting in a Nyquist frequency of 8512 Hz.

5.8.1 Testing a minimal phase anti-aliasing filter

The first test case implemented only a minimal phase anti-aliasing filter in the system. It was determined that a single real pole filter would eliminate aliasing of the tonal noise disturbance field, Figure 5.2. The theoretical output from the anti-aliasing filter is shown in Figure 5.4. Frequencies above (f_s-x) alias , but are below the resolution of the A/D.

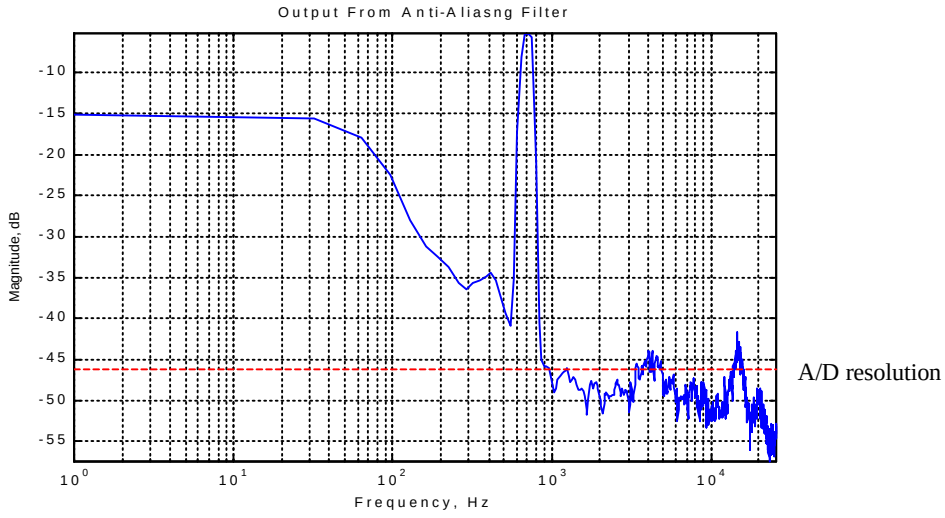

Figure 5.4 Predicted output of minimal phase anti-aliasing filter subjected to tone noise field in Figure 5.2.

The performance of the system, Figure 5.5 and Figure 5.6, at low frequencies was excellent. The predicted analog and measured digital reduction in the tone at 700 Hz was 11 dB(SPL). In addition, the measured spill-over region from 1000 - 4000 Hz behaved identical to the predicted analog spill-over. The weakness of this single filter digital technique occurred at the higher frequencies. The noise levels experienced by the microphone were increased for all frequencies above the Nyquist frequency (8512 Hz). This deviated from the prediction that there should be no spill-over at these high frequencies with the analog controller. The increase occurred because the output from the D/A converter consists of controlled signals from 0 Hz up to the Nyquist frequency. Above the Nyquist frequency, the output consists of imposters, and the magnitudes of these imposters are greatest at integer multiples of the sampling frequency when the control filter is low-pass. This is evident in the response as the sound pressure levels (SPLs) near 17024 Hz are all increased dramatically.

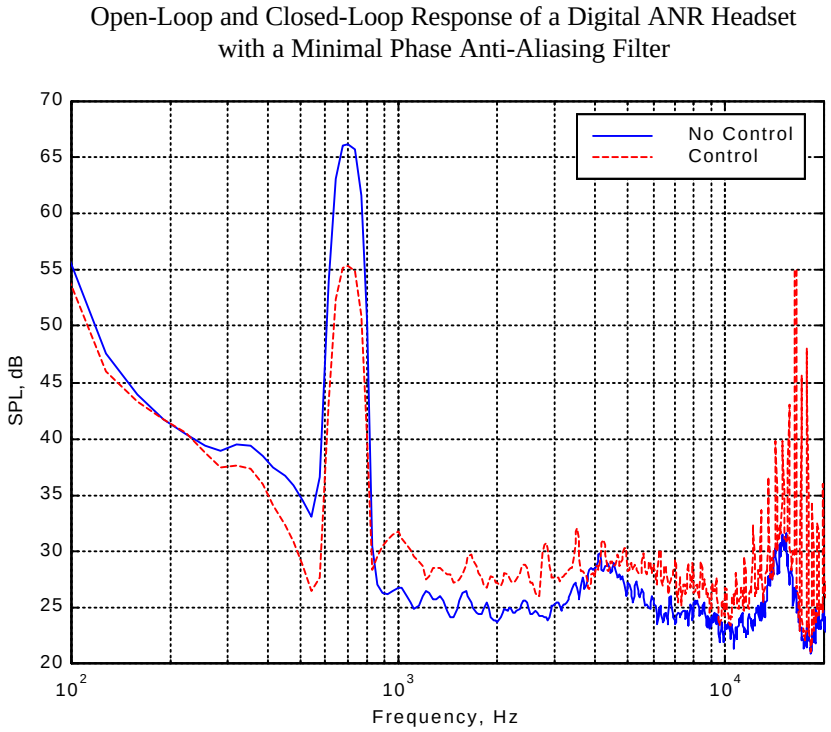

Figure 5.5 Open and closed-loop SPLs at the microphone when only a minimal phase anti-aliasing filter was implemented.

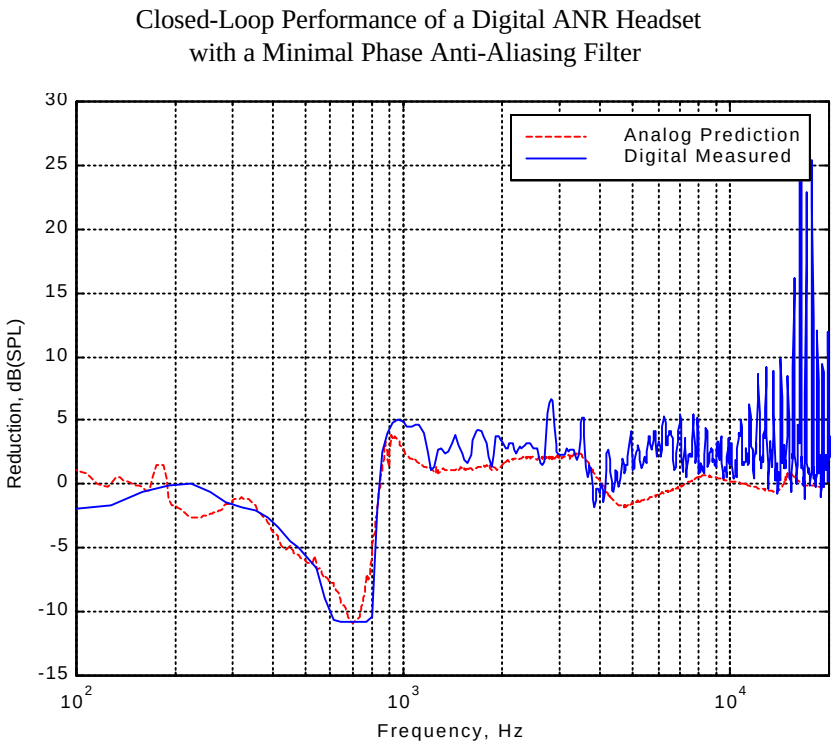

Figure 5.6 Closed-loop reduction in SPLs measured with the digital controller and predicted by the equivalent analog controller.

As discussed in Chapter 4, for many applications the introduction of the high frequency imposters does not hinder performance. For instance, an extremely large mass and actuator could be used on the roof of a skyscraper to dampen the oscillations of the structure due to wind forces. The actuator for such a large mass would be physically incapable of moving the mass very quickly. Therefore, the actuator would be impervious to high frequency imposters present in the output from a D/A converter. For such an application, a smoothing filter would not be necessary, and the results from using the anti-aliasing filter alone would be sufficient. However, acoustic applications create a need for these imposters to be removed. These high frequencies create a louder sound field for the listener. In addition, they tend to create a buzzing sound in the control speaker's acoustic field. Therefore, the minimal phase anti-aliasing filter alone is not capable of achieving good control performance in an acoustic application such as an ANR headset.

5.8.2 Testing a minimal phase smoothing filter

A minimal phase anti-aliasing filter alone was not sufficient for acoustic applications, but the results of the first test case suggest that a minimal phase smoothing filter might perform well. The smoothing filter will eliminate the imposters that decreased system performance with only a minimal phase anti-aliasing filter. Using the same control law as in the first test case, it was determined that a single pole smoothing filter would be sufficient for the noise field presented to the system, Figure 5.2. Figure 5.7 demonstrates that there should be no imposters introduced with a level greater than the minimum voltage, 5 mV (-46 dB), capable of powering the audio amplifier on the control speaker.

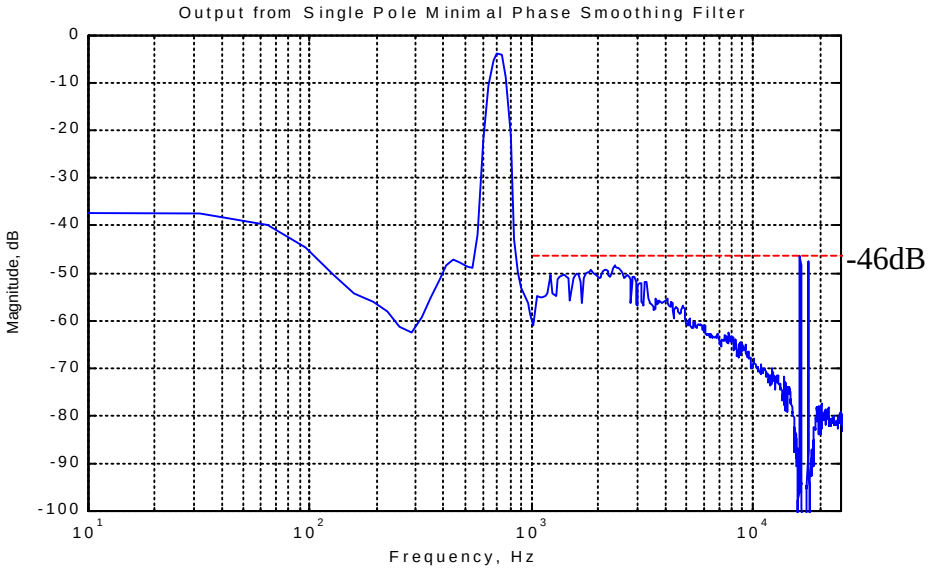

Figure 5.7 Predicted output from a single pole minimal phase smoothing filter subjected to noise field in Figure 5.2.

Figure 5.8 and Figure 5.9 both illustrate the performance of the controller with only the minimal phase smoothing filter being implemented. A quick glance reveals that the imposters were successfully attenuated, and the tonal reduction at 700Hz was 11 dB(SPL). Additionally, the higher frequency range of the open-loop and closed-loop SPLs are equivalent, indicating that no acoustic energy was added. However, performance was diminished in the low frequency range. The predicted analog

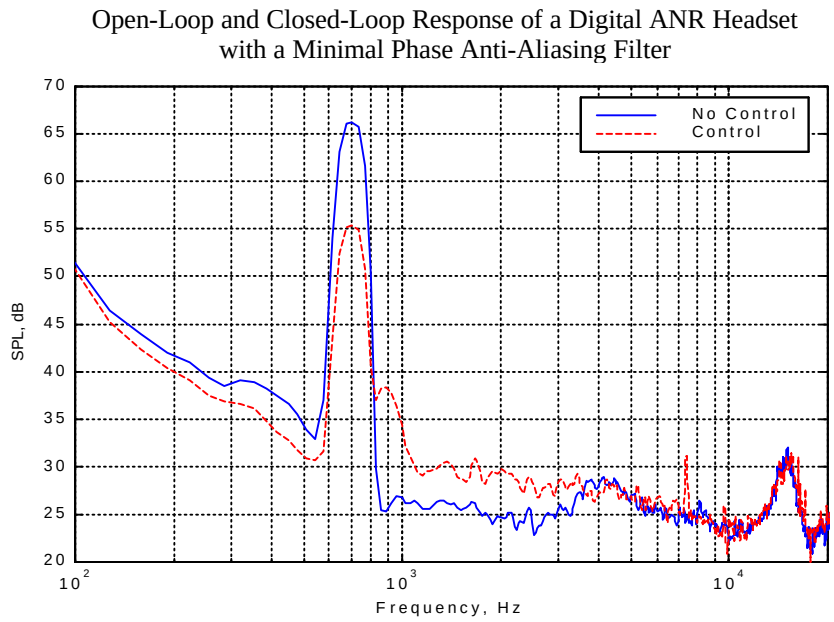

Figure 5.8 Open and closed-loop SPLs at the microphone when only a minimal phase smoothing filter was implemented.

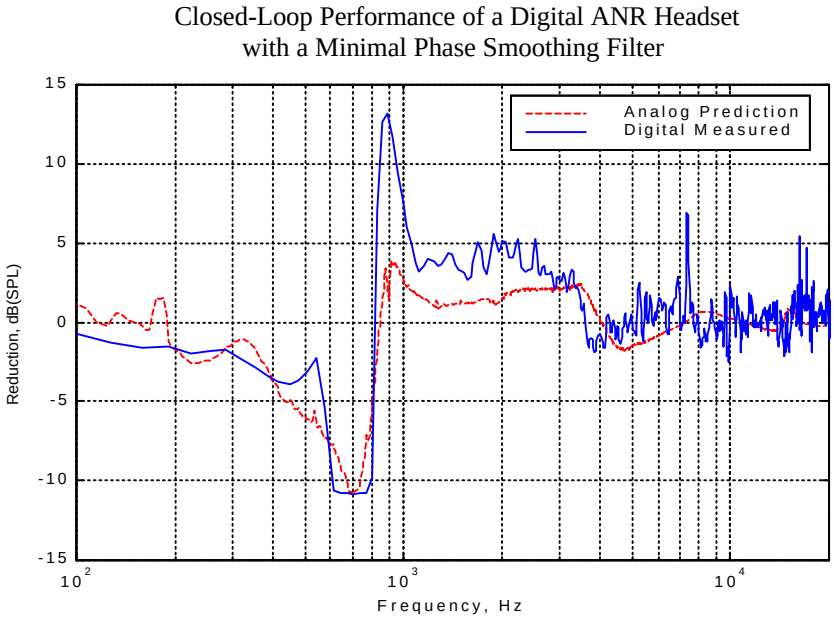

Figure 5.9 Closed-loop reduction in SPLs measured with the digital controller and predicted by the equivalent analog controller.

controller had 4 dB (SPL) of spill-over in the region 900-1000Hz, but the digital controller contributed a peak value of 13dB(SPL). This is a drastic increase in spill-over. To determine the source of this additional spill-over, the high frequency range in the open-loop spectrum was analyzed. For a sampling rate of 17024 Hz, signal content at

16124 Hz aliases to 900 Hz. The open-loop SPL at 16124 Hz in Figure 5.8 is 32 dB(SPL) and the level at 900Hz is 26 dB(SPL). Once the aliasing occurs, the pressure at 900 Hz is a combination of both these pressure levels, resulting in an effective pressure of 35 dB(SPL). This is an increase of 9 dB(SPL) in the open-loop pressure level at 900 Hz. In addition, the control law predicts 4 dB(SPL) of spill-over will be added to the open-loop pressure level. The net effect is 13 dB(SPL) of combined spill-over and aliasing. This effect carries over everywhere in the pressure spectrum below the Nyquist frequency. So, the minimal phase smoothing filter alone achieved excellent acoustic control performance in the high frequency range and in the low frequency range, $0-800$ Hz. The implementation, however, suffered in the mid-frequency spill-over range.

The concept of aliasing the open-loop SPL that the controller is subjected to is very important. It was demonstrated that aliased frequencies create a louder sound field when the loop is closed. However, this is a direct result of the disturbance noise field. The SPLs near the sampling frequency in Figure 5.2 are actually greater than some of the low frequency SPLs. In a vast majority of cases though, high frequency SPLs are significantly lower than the low frequency SPLs . At a certain low frequency SPL, the minimal addition of an aliased signal will not adversely affect performance. For instance, if the SPL at 16124Hz had been 14 dB(SPL) and the SPL at 900Hz remained at 26dB(SPL) the increase in SPL at 900Hz would have only been 2dB(SPL). This is an interesting point. If the noise field at the high frequencies is low enough, the anti-aliasing filter is not needed. The minimal phase smoothing filter will be sufficient.

5.8.3 Testing minimal phase anti-aliasing and smoothing filters

The first two test cases revealed the inherent weakness in designing a system with only a minimal phase anti-aliasing filter and illustrated that a minimal phase smoothing filter may be sufficient in some applications. The third test implemented both filters in an attempt to increase system performance. This design necessitated the reformulation of

the control law, however. The increased phase lag due to the combination of the two filters would have driven the previous controller unstable.

Figure 5.10 and Figure 5.11 demonstrate the positive performance results when both minimal phase filters were implemented. The spill-over region for the predicted analog controller and designed digital controller both match, indicating that no aliasing occurred. In addition, there were no imposters introduced into the noise field. The combination of both filters eliminated both aliasing and imposter impurities in the noise field.

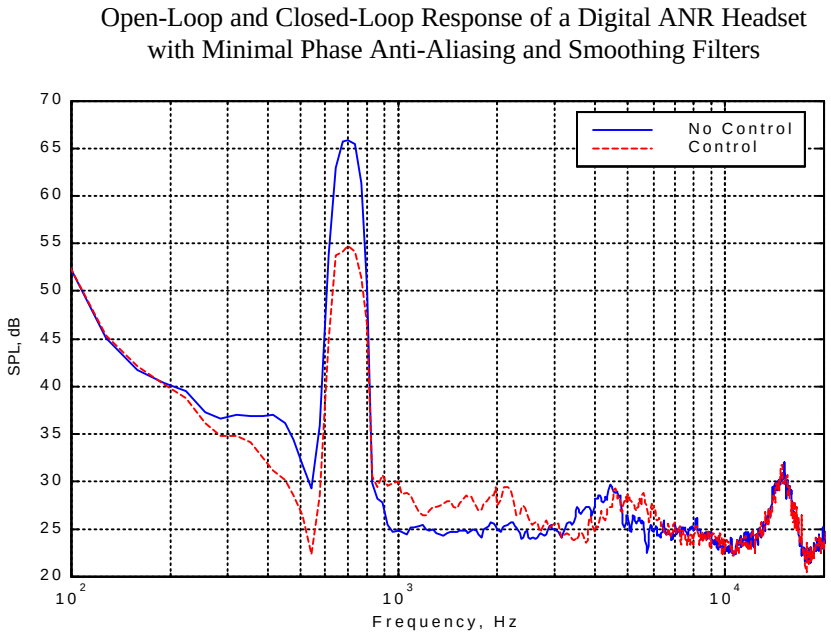

Figure 5.10 Open and closed-loop SPLs at the microphone when both a minimal phase anti-aliasing and smoothing filter were implemented.

These figures demonstrate the feasibility of building a digital ANR headset that includes an anti-aliasing filter and a smoothing filter. The key in designing the last controller was in thinking ahead and using the filters as an advantage. As previously discussed, the plant was not ideal because it didn't roll-off until very high frequencies. As a result, to achieve low frequency control with minimal spill-over the plant had to be rolled-off through the controller in the analog design. In the case of the digital counterpart, the aliasing filter and smoothing filter were each used to achieve this roll-off and simultaneously perform their primary functions. The filters did not deprive the system of

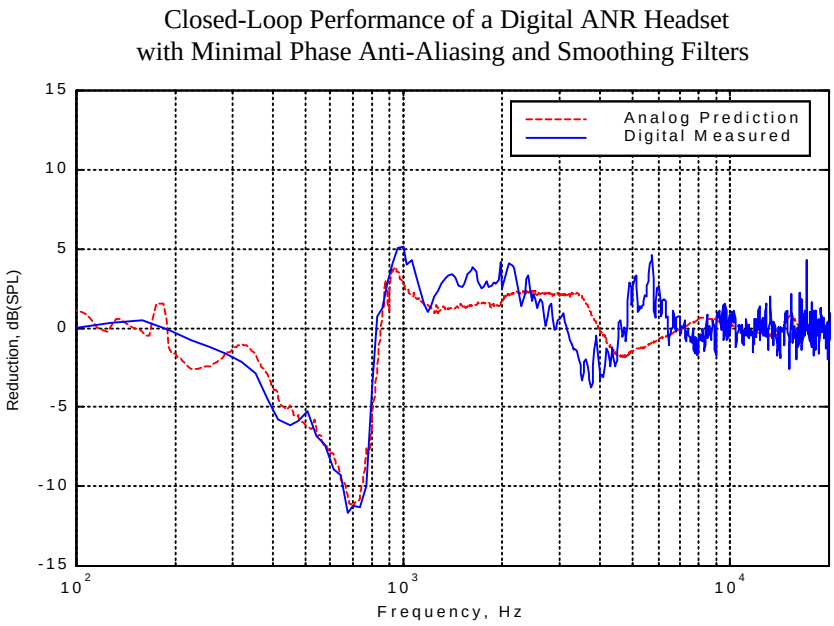

Figure 5.11 Closed-loop reduction in SPLs measured with the digital controller and predicted by the equivalent analog controller

performance because the roll-off was necessary with either design. Until speaker diaphragm dynamics are improved such that a plant like this does roll-off, the digital design should be comparable to the analog design.

5.8.4 Performance evaluation of the first three test cases

Thus far, the analysis of each of the configurations was based on a qualitative comparison between predicted closed-loop analog performance and measured closed-loop digital performance. A qualitative discussion of masking and quantitative measurements of acoustic bandpower provide the final evaluation tools for the first three design cases.

Table 5.1 includes the bandpower measurements for each of the test cases. The bandpower was measured in three separate regions which organically arose from the design criteria. The first region was 0-1000 Hz in which the control was designed to perform. The second region was 1000-8512 Hz which formed the rest of the frequencies that did not alias when sampled. The final region was 8512- 20,000 Hz which consumed

the rest of the ear's audible range . As a last measure of comparison, the SPL at the tone disturbance frequency of 700 Hz was tabulated.

The table illustrates that the anti-aliasing filter alone performed adequately at the low frequencies as it reduced the bandpower for frequencies below 1000 Hz and only slightly increased the bandpower in the mid-frequency region. Even though masking backwards is ineffective, this helps limit the possibility that spill-over will impact low frequency performance. In the highest region, the bandpower increased dramatically due to the addition of imposters to the open-loop disturbance field. Lastly, the tonal suppression at 700 Hz was not affected by not having the smoothing filter.

Table 5.1 The reduction in bandpower as perceived by the microphone in dB(SPL).

Frequency Range	0 - 1000 Hz	1000 - 8512 Hz	8512 - 20k Hz	at 700 Hz
Anti-Aliasing Filter Only	5.4	-2.4	-11.7	10.7
Smoothing Filter Only	6.9	-1.5	-0.6	10.8
Smoothing and Anti-Aliasing	7.5	-0.7	-0.2	12.1

The smoothing filter, on the other hand, performed very well in each region of concern. The reduction in bandpower below 1000 Hz was 1.5 dB(SPL) greater than for the anti-aliasing filter alone. Also, the increase in the high frequency range was negligible. Again, the tonal suppression was not affected by not having the anti-aliasing filter.

Finally, the design including both filters performed well. The tonal suppression was an excellent 12 dB(SPL) and the low frequency bandpower reduction was better than either of the first two test cases. High frequency performance was also excellent as the increase in bandpower was negligible.

Based on these results, it was concluded that the worst performance occurred when only the anti-aliasing filter was implemented. Additionally, the performance of the system with only a minimal phase smoothing filter nearly equaled the performance with both filters. This indicates that for some applications an ANR headset may be implemented without an anti-aliasing filter.

5.9 Elliptic vs. minimal phase smoothing filter

In testing the theory that only a smoothing filter is necessary, a comparison was done. Traditionally, smoothing filters are high-order filters that have very steep roll-off rates. With increased roll-off rates, the cut-off frequency can be set very high to allow for a greater pass-band. The final test case examined the performance of an elliptic smoothing filter and compared it to the new minimal phase filter design in light of the fact that an anti-aliasing filter may not be necessary for some applications.

For this last test case, the performance goals were slightly altered. Previously all cases targeted a pass-band of 0-1000 Hz, but this case was designed for a pass-band of $0 - 2000\ \text{Hz}$. This broader range increased the roll-off rate required of the minimal phase smoothing filter. In addition, the disturbance noise field was altered. A pink noise field was used instead of a tonal disturbance. This type of disturbance was tested because it rolls off at a rate of 3 dB per decade. So, the higher frequency disturbance content will have a reduced magnitude in comparison to the low frequency disturbance content. This is precisely the situation where using only a smoothing filter is plausible. Finally, the controller was altered such that introduction of either of the two smoothing filters would not drive the system unstable.

5.9.1 Testing an elliptic smoothing filter

In testing the digital ANR headset with an elliptic smoothing filter, a 6-zero, 8-pole, low-pass filter was used. This filter had a very steep roll-off, but sacrificed phase lag. In

order to achieve the necessary reduction of imposters, the filters cut-off frequency was set to 9000 Hz, 500 Hz above the Nyquist. For 'weaker' ambient noise fields this could be set even higher to alleviate some of the phase introduction of the filter at low frequencies. Set at this cut-off frequency, the filter added 30° of phase lag at 1000 Hz which affected the spill-over in the closed-loop response. The increased phase roll-off of a filter can cause a system's phase to shoot straight downward at an extremely high rate. This results in more phase cross-overs in a smaller frequency range. Consequently, the spill-over region may increase, may shift and more regions may be added.

The use of the elliptic filter, Figure 5.12 and Figure 5.13, illustrated the concept of shifting and adding spill-over. The controller was predicted to introduce 5 dB(SPL) of spill-over at 1000 Hz and 3-4 dB(SPL) across the frequency range 1000-2000 Hz. The elliptic filter's additional phase caused the spill-over region to change. Instead of peaking at 1000 Hz, the spill-over increased from 900 Hz until its is maximum near 1700 Hz.

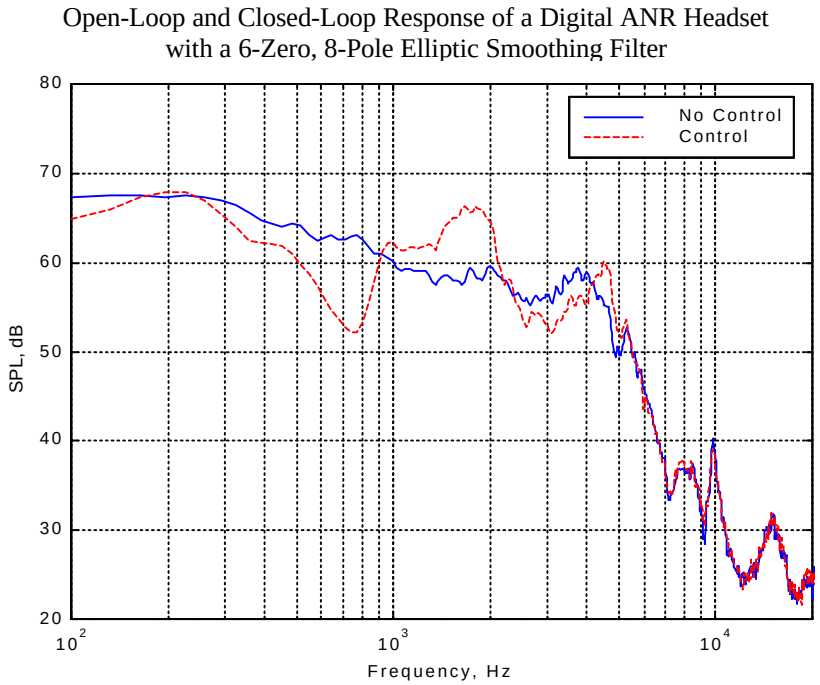

Figure 5.12 Open and closed-loop SPLs at the microphone when only an elliptic smoothing filter was implemented.

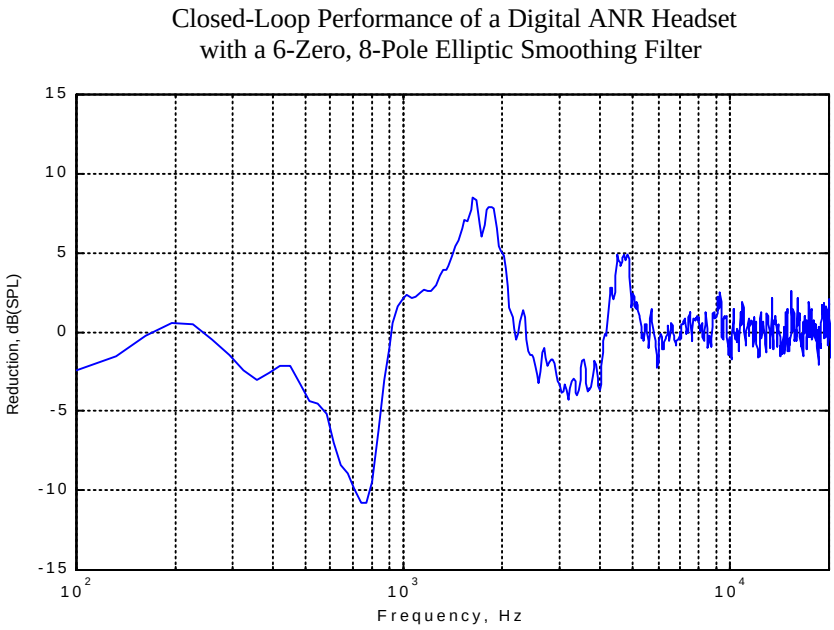

Figure 5.13 Closed-loop reduction in SPLs measured with the digital controller and predicted by the equivalent analog controller

The peak value was also well above 5dB(SPL). Additionally, the filter introduced a new spill-over region up near 5000 Hz. The reason for the increased spill-over near 2000 Hz and the new region near 5000 Hz was increased phase, but more importantly, the roll-off of the minimal phase filter was not present to attenuate the plant dynamics in these higher frequency regions. This elliptic filter strictly added phase at low frequencies with no magnitude consequences. For this plant, the magnitude roll-off is necessary. So, the elliptic filter was successfully implemented, but the additional spill-over limited system performance.

5.9.2 Testing a minimal phase smoothing filter

The minimal phase smoothing filter required to attenuate the imposters for the pink noise disturbance spectrum consisted of a single pair of complex poles with a corner frequency set at 2000 Hz. This complex pair of poles added approximately 45° of phase at 1000 Hz, but also attenuated all frequencies above 2000 Hz. Therefore, higher frequency regions of spill-over were not introduced.

Performance of the minimal phase smoothing filter, Figure 5.14 and Figure 5.15, was better than the elliptic filter. The spill-over region peaked around 1000Hz and leveled out until 2000 Hz, where it decreased to approximately 0 dB(SPL) as was predicted with just the controller. The slightly higher spill-over level at 1000 Hz introduced by the minimal phase filter as opposed to the elliptical filter was due to phase characteristics.

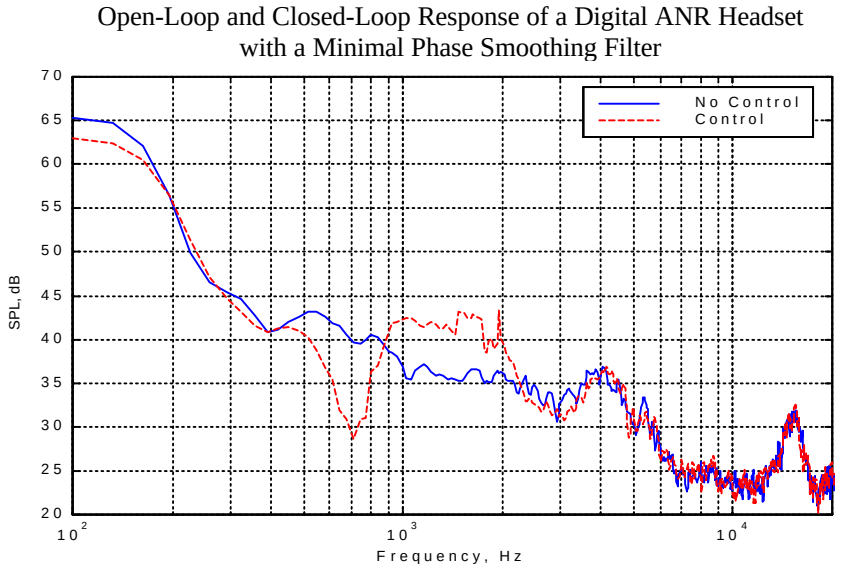

Figure 5.14 Open and closed-loop SPLs at the microphone when only a minimal phase smoothing filter was implemented.

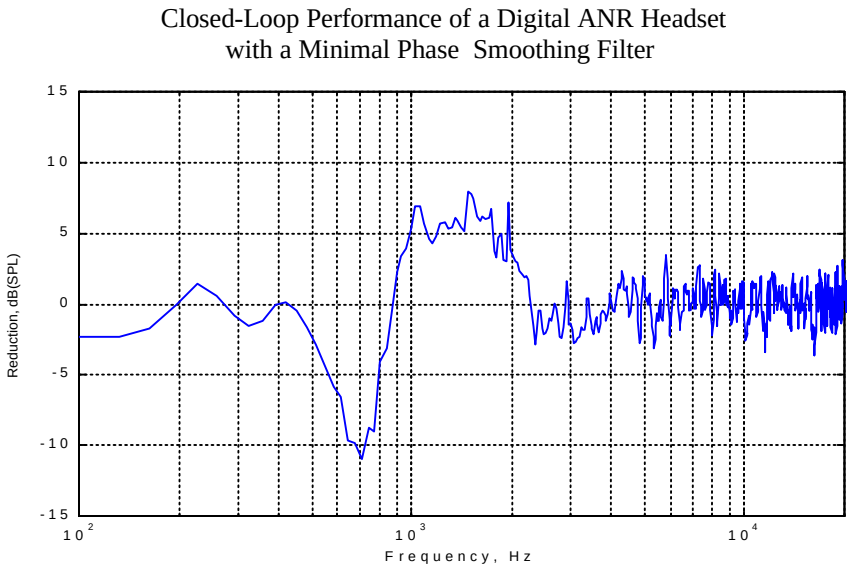

Figure 5.15 Closed-loop reduction in SPLs measured with the digital controller and predicted by the equivalent analog controller

The minimal phase filter added 20° more phase lag at 1000 Hz than the elliptical filter. However, due to the roll-off, there was no additional spill-over region added by the minimal phase filter at the higher frequencies. For this particular plant, the minimal phase smoothing filter out performed the elliptic filter.

5.9.3 Performance evaluation of elliptic and minimal phase smoothing filter

As in the previous tests cases, the designs including only smoothing filters were evaluated based on their acoustic performance. Table 5.2 shows the bandpower measurements for both designs and illustrates that the concept of eliminating the anti-aliasing filter was plausible.

Table 5.2 The reduction in bandpower as perceived by the microphone in dB(SPL).

Frequency Range	0 - 1000 Hz	1000 - 2000 Hz	2000 - 8512 Hz	8512 - 20k Hz
Elliptic Filter	2.8	-5.0	0.2	-0.4
Minimal Phase Filter	2.5	-4.3	1.0	0.0

The designs were very comparable in performance based on these bandpower measurements. In each of the regions, the designs had approximately equal reduction or increase in bandpower. However, the minimal phase filter had a slight advantage due to the fact that the closed-loop response did not have as much spill-over as the elliptic filter's closed-loop response. The elliptic filter achieved some control and some spill-over in the higher frequency regions, resulting in negligible increases in bandpower, but the portion of the spectrum that was increased may result in a louder sound field for an individual using the headset.

5.10 Conclusion

Using the strategies presented in Chapter 4, it is possible to design a digital ANR system consisting of an anti-aliasing filter, control filter, and a smoothing filter. Although using both filters is possible, in some applications it may be beneficial to eliminate the anti-aliasing filter and its additional phase lag. In these applications, a smoothing filter is capable of effectively meeting the anti-aliasing requirements, but an anti-aliasing filter would not be capable of eliminating imposters as required. In cases where the high frequency noise content is at a significantly lower SPL than the low frequency noise content, this strategy may be employed to enhance the overall performance of the system. Furthermore, if this technique is used, it is beneficial to use a minimal phase filter design before a high cut-off, fast roll-off filter. High-order filters add phase at low frequencies without attenuating the plant in the mid-frequency range, while the minimal phase filter achieves both simultaneously. This roll-off helps to eliminate spill-over regions.

6 CHAPTER: CONCLUSIONS AND FUTURE WORK

6.1 Final summary

This thesis has attempted to fill in some of the void in the literature relating to ANR headsets and, more specifically, literature on digital ANR headsets. After systematically characterizing each component of a headset, it was concluded that the anti-aliasing and smoothing filters were optimal candidates to reduce the phase lag in a digital implementation. Subsequently, a design procedure was developed that resulted in minimal phase anti-aliasing and smoothing filters. Through use of a prototype single channel headset, these minimal phase filters were tested and evaluated. The results of the testing provided support for eliminating the anti-aliasing filter and designing with only a smoothing filter.

During the exploration of the acoustics and speaker dynamics encountered in a headset, it was determined that the phase lag introduced by acoustic propagation from the control speaker to the error microphone is not of primary concern. For a speaker-microphone separation distance of $1\,\text{inch}$, the acoustic phase lag was less than $10°$ at 1000 Hz. The poor dynamic response of the control speaker introduced the most phase lag. Through use of laser vibrometry, it was demonstrated that at high frequencies of excitation the speaker diaphragm tends to break away from its low frequency rigid piston mode, and the central portion begins to vibrate $180°$ out of phase with the surround. An acoustic model of the sound propagation due to this second mode was derived and mathematically removed from test data to isolate the phase response of the speaker. The resulting speaker phase lag reached $180°$ by 1000 Hz and decreased very rapidly as higher excitation frequencies were approached. This phase lag from the speaker dynamics overshadows the acoustic propagation phase lag.

In order to fully understand the control aspect of the headset, each component of a digital sampling system was characterized. Based upon the knowledge of how an input frequency spectrum would be shaped while passing through such a system, a set of design parameters were developed that formulate anti-aliasing and smoothing filters that contribute minimal phase. This minimal phase filter design procedure was implemented in MATLAB code to aid in the design process.

Finally, this minimal order filter design procedure was implemented and tested on a single channel headset replica. Closed-loop testing in response to a tone disturbance at 700 Hz superimposed on low-level background noise consisted of three different hardware configurations: only a minimal phase anti-aliasing filter, only a minimal phase smoothing filter, both a minimal phase anti-aliasing filter and a minimal phase smoothing filter. Each of the configurations was evaluated based on acoustic masking, comparison to an analog controller, and acoustic bandpower measurements. The minimal phase smoothing filter outperformed the minimal phase anti-aliasing filter when each was implemented individually. In the region of control, $0-1000\,\text{Hz}$, the smoothing filter achieved $1.5\,\text{dB(SPL)}$ more reduction in acoustic bandpower power than the anti-aliasing filter. In addition, the anti-aliasing filter increased the acoustic power in the high frequency region, $8512-20000\,\text{Hz}$, by $11.1\,\text{dB(SPL)}$ more than the smoothing filter. Use of both filters only slightly improved performance over using only the smoothing filter. The decrease in acoustic energy in the region of control was slightly greater, $0.6\,\text{dB(SPL)}$, and the increase in acoustic energy in the high frequency region was slightly less, $0.4\,\text{dB(SPL)}$. As a result, it was concluded that the smoothing filter was necessary and sufficient for implementation of a digital ANR headset for some applications.

The order of the minimal phase anti-aliasing and smoothing filters implemented in the three test cases provided promising news for future designs. Each filter required only a single pole with a corner frequency set at the highest frequency in the region of control, 1000 Hz. Traditionally, high-order filters, such as elliptic filters, are used for these applications. These high-order filters are characterized by fast roll-off rates that allow the

corner frequency to be set at a higher frequency than the minimal phase filters designed in this thesis. These filters, however, contribute tremendous amounts of phase lag to the system. The Frequency Devices 9002 is a low-pass, 8-pole, 6-zero elliptic low-pass filter that contributes almost 30° of phase lag at 1000 Hz with the corner frequency set as high as the Nyquist frequency for these tests, ($f_{nyq} = 8512\,\text{Hz}$). The minimal phase filters each contribute 45° of phase lag at 1000 Hz, but the contribution levels off at -90°. The elliptic filter, however, has a phase lag that continually increases, 323° by 8512 Hz.

In order to illustrate the performance of a minimal phase smoothing filter, comparison was made to an elliptic smoothing filter's performance. The plant was subjected to a pink noise disturbance signal and a pass-band was designated from 0-2000 Hz. Using the same controller, each filter was tested. The elliptic filter's cutoff frequency was set as high as possible, 9000 Hz, without sacrificing performance, and the minimal filter required a pair of complex poles with a corner frequency of 2000 Hz. The performance of the two filters were nearly identical when acoustic bandpower reduction was analyzed, but the minimal phase filter had a more desirable closed-loop response. The elliptic filter caused more spill-over in high frequency ranges, driving the acoustic power higher, but also achieved better control over certain high frequency ranges driving the acoustic power down. A desirable flat response with very little spill-over, or control, in the high frequency range was observed with the minimal phase filter. Even though masking is not effective from higher frequencies to lower frequencies, this flat response eliminates the possibility that the closed-loop system will cause acoustic masking problems.

6.2 Future work

It has been demonstrated that a single channel digital ANR headset can be designed and implemented successfully. Using only a smoothing filter, the digital system's performance was satisfactory when applied to a tonal disturbance as well as a broadband pink noise disturbance spectrum. The designs, however, were not optimal. Poor control speaker dynamics impacted system performance as well as the limitations imposed by the

processing speed of the DSP board. Additionally, designs were done without considering the effects of masking until the evaluation stage. In order to improve digital ANR headset technology, certain topics need to be addressed further: speaker dynamics, acoustic masking, and improved control techniques.

6.2.1 Improving speaker dynamics

The poor dynamic response of the control speaker imposed a major phase lag limitation on closed-loop system performance. Breakup modes of the speaker diaphragm created a large phase lag that affected the amount of acceptable phase introduction by the controller, thereby limiting performance. Ideally, a speaker with only a mechanical resonance, rigid diaphragm, would be used. The peak in the magnitude response would allow for good control over a certain low frequency range and the phase would level out at -180°. In addition, with only one resonance, the magnitude response would roll-off at high frequencies, which would help to reduce closed-loop spill-over and avoid the possibility of masking low frequency performance. Design of such a speaker would drastically impact the performance of a headset.

Even with the poor dynamics of present speakers at high frequencies, the plant magnitude response may be altered to improve headset performance. Work has been done to help identify the breakup modes of diaphragms, and this knowledge can be used to actively re-shape the response. Use of modal filters may be implemented to negate the poor dynamics of these breakup modes and improve the magnitude response. By canceling these higher order modes, the speaker's response will appear to have only a mechanical resonance.

6.2.2 Examining acoustic masking

The concept of acoustic masking was introduced briefly in Chapter 5. This is an area that needs to be examined in great detail if success of a digital controller is to be realized for

certain applications. Testing needs to be done to determine exactly how much acoustic energy it takes in the regions adjacent to the control region to backward mask performance. Determination of these energy levels would help in the design of a minimal order filter. The design procedure that was developed was based on the fact that no energy should be added to the disturbance spectrum. This may be too stringent, however. Allowing for a slight increase in the frequency bands above the control region may not cause decreased performance in certain ANR headset applications. The decreased filter slopes that would accompany this design decision would allow greater flexibility in the controller design.

6.2.3 Applying adaptive control techniques

Very little attention was given to designing an optimal controller while this research was being completed. The tests were focused on demonstrating the effects of signal conditioning filters on the overall closed-loop system. However, the controller is as critical as any other component in a control system. This is actually one aspect that makes a digital controller so enticing. The use of a digital control law allows for greater flexibility than an analog control law.

A universal controller may be designed which adapts itself to a particular noise field. The ability of the controller to adapt itself in the face of different inputs would create better control. Instead of using a control law that operates satisfactorily over a wide range of disturbances, the adaptive controller would use an optimal control law in every situation. The incorporation of adaptive control techniques such as LMS should be examined. Combining the broadband capabilities of such adaptive methods with the tonal suppression of a fixed controller would enhance performance.

6.3 The end product: a digital ANR headset

In the future, all of this background information will be put to use and implemented in a digital ANR headset. As technological advances continue in the digital world, such a headset will become practical from a dollars and cents standpoint, as well as from a logistics standpoint. The inflexibility of the analog system will eventually be replaced by a more versatile digital system.

It is possible to envision the design of a headset that consists of many smoothing filters and many control laws from which the system will select depending upon the disturbance field. The system will be capable of sampling the existing disturbance and processing some simple code to calculate the necessary filter roll-off rates. Then, through some digital switches, the appropriate filters will be implemented. Finally, with the correct filters in place, the system will go through a routine in which it optimizes the control law to be used. Once all of the kinks are worked out, such a system should be relatively inexpensive and extremely versatile, making it a must for any factory, runway, or babysitter.

APPENDIX A: THE BIQUAD CIRCUIT

A.1 Introduction

When controls is first studied, the concept of a transfer function is introduced and it provides the foundation for the rest of the subject matter. The connection between the transfer function and analog circuitry is not always made by the student, or by the professor. However, this connection is imperative if someone is going to perform an analysis of a system, design a set of filters, and then implement his/her solution. The analysis task is well documented and the theory has been laid out. But, converting the results of the analysis from transfer functions into circuitry is not always straightforward.

In an effort to build filters with specific characteristics, a method for converting desired second-order transfer functions into analog circuitry was developed[i]. This method is based upon the biquad circuit shown in Figure A.1. The biquad circuit consists of three op-amps, and through manipulation of the input and feedback impedances the circuit's transfer function may be adjusted.

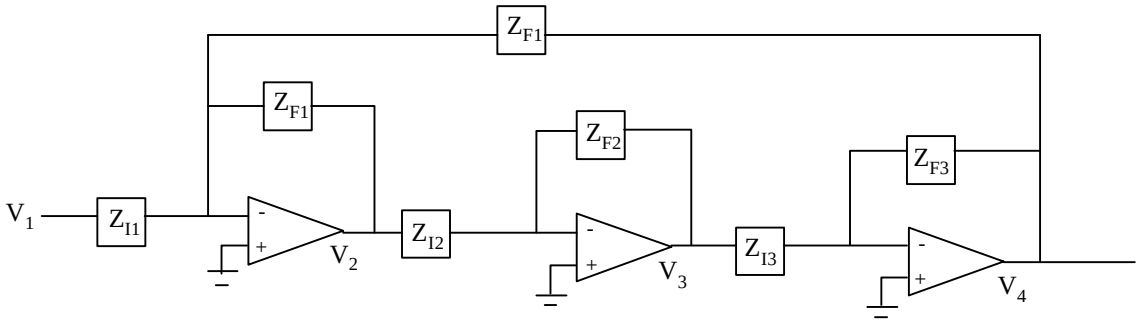

Figure A.1 Biquad circuit diagram.

[i] Other transfer functions are discussed by Van Valkenburg (1982)

The biquad circuit as pictured in Figure A.1 does not provide a means to create just any transfer function. It produces a second-order transfer function with poles and zeros defined by the impedances. Adjustment of the zeros of the circuit may be accomplished by adding op-amps in parallel with the existing three op-amp configuration. This procedure only affects the zeros, not the poles. If the desired transfer function is of an order greater than two, the circuit may be constructed by cascading biquad circuits in series.

A second-order system with two zeros can be described through damping values and natural frequencies. The zeros have an associated natural frequency (ω_{nz}) and damping ratio (ξ_z) that are independent from the natural frequency (ω_{np}) and damping ratio (ξ_p) of the poles.

$$TF(j\omega) = \frac{s^2 + 2\xi_z \omega_{nz} s + \omega_{nz}^2}{s^2 + 2\xi_p \omega_{np} s + \omega_{np}^2} \qquad (A.1)$$

Therefore, there are four characteristic values that define the system. Depending upon these values, the transfer function may take on a decidedly different appearance. Each of the four possible cases is shown in Table A.1, and each requires a slightly different circuit.

Case #	Characteristic Values
1	$\omega_{nz} > \omega_{np}$ and $\xi_z > \xi_p$
2	$\omega_{nz} > \omega_{np}$ and $\xi_p > \xi_z$
3	$\omega_{np} > \omega_{nz}$ and $\xi_p > \xi_z$
4	$\omega_{np} > \omega_{nz}$ and $\xi_z > \xi_p$

Table A.1 The four possible two-zero two-pole transfer function configurations.

In order to provide a quick and easy means of obtaining the resistor and capacitor values required to form the appropriate impedances, a MATLAB code[i] was developed. The code first uses the values of ω_{np}, ω_{nz}, ξ_z, and ξ_p to calculate which of the four cases is being implemented. Then, the code processes some design equations and produces the required resistor and capacitor values. The design equations that are used are developed in the following sections relating to each case.

The analysis procedure for each case is the same. First, a circuit is drawn, and the resulting transfer function is developed relating the input voltage and output voltage in terms of the resistors and capacitors of unspecified values, Table A.2. Then, this transfer function is compared to the nominal representation described in (A.1). Making the comparison between individual terms in the transfer functions yields four design equations for each situation that are expressed in terms of resistors and capacitors. In each case, there are more unknown capacitor and resistor values than equations[ii]. This results in a solution that is not unique. The results may be tuned by the designer until favorable component values are obtained that are common and easy to implement.

V_1	Input voltage
V_2,V_3,V_4	Intermediate voltages
V_5	Output voltage
R_1,R_2,R_3,R_4,R_5	Resistors
C_1,C_2	Capacitors

Table A.2 Notation used in the following sections

[i] See BUILDTF.M in APPENDIX B: MATLAB CODES

[ii] The 'free' capacitor values C_1 and C_2 along with the 'free' resistor value R_5 will be discussed following the formulation of each case.

A.2 Case 1: $\omega_{nz}>\omega_{np}$ and $\xi_z>\xi_p$

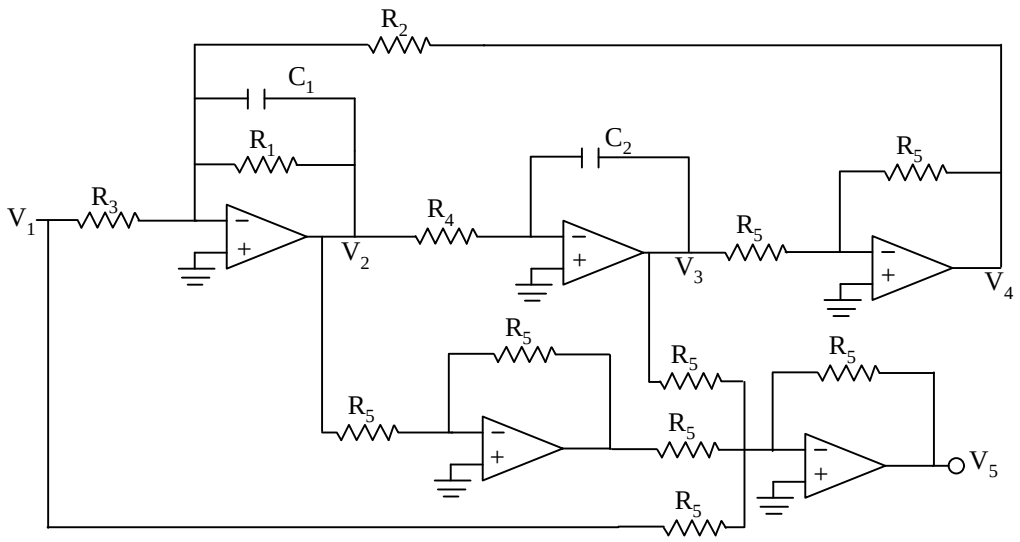

Figure A.2 Biquad circuit diagram for case 1.

The circuit transfer function is given by (A.2).

$$\frac{V_5}{V_1} = -\frac{s^2 + \left(\frac{1}{R_1C_1} + \frac{1}{R_3C_1}\right)s + \left(\frac{R_3 + R_2}{R_2R_3R_4C_1C_2}\right)}{s^2 + \left(\frac{1}{R_1C_1}\right)s + \left(\frac{1}{R_2R_4C_1C_2}\right)} \tag{A.2}$$

Comparing this transfer function to (A.1) yields four design equations.

$$R_1 = \frac{1}{2\rho_p\omega_{np}C_1} \tag{A.3}$$

$$R_2 = R_3\left(\frac{\omega_{nz}^2}{\omega_{np}^2} - 1\right) \tag{A.4}$$

$$R_3 = \frac{1}{2C_1\left(\rho_z\omega_{nz} - \rho_p\omega_{np}\right)} \tag{A.5}$$

$$R_4 = \frac{1}{R_2C_1C_2\omega_{np}^2} \tag{A.6}$$

A.3 Case 2: $\omega_{nz}>\omega_{np}$ and $\xi_p>\xi_z$

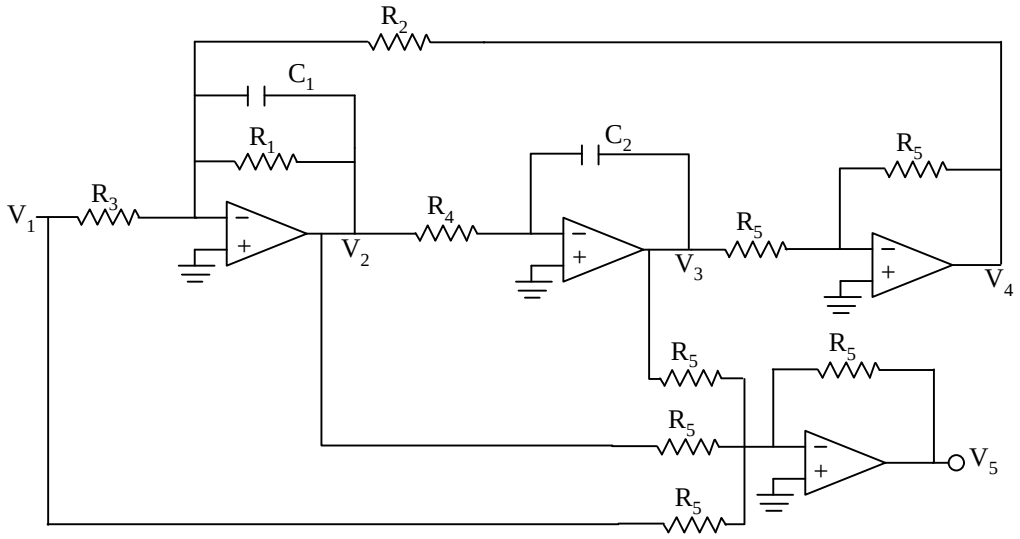

Figure A.3 Biquad circuit diagram for case 2.

The circuit transfer function is given by (A.7).

$$\frac{V_5}{V_1} = -\frac{s^2 + \left(\frac{1}{R_1C_1} - \frac{1}{R_3C_1}\right)s + \left(\frac{R_3 + R_2}{R_2R_3R_4C_1C_2}\right)}{s^2 + \left(\frac{1}{R_1C_1}\right)s + \left(\frac{1}{R_2R_4C_1C_2}\right)} \tag{A.7}$$

Comparing this transfer function to (A.1) yields four design equations.

$$R_1 = \frac{1}{2\rho_p\omega_{np}C_1} \tag{A.8}$$

$$R_2 = R_3\left(\frac{\omega_{nz}^2}{\omega_{np}^2} - 1\right) \tag{A.9}$$

$$R_3 = \frac{1}{2C_1\left(\rho_p\omega_{np} - \rho_z\omega_{nz}\right)} \tag{A.10}$$

$$R_4 = \frac{1}{R_2C_1C_2\omega_{np}^2} \tag{A.11}$$

A.4 Case 3: $\omega_{np}>\omega_{nz}$ and $\xi_p>\xi_z$

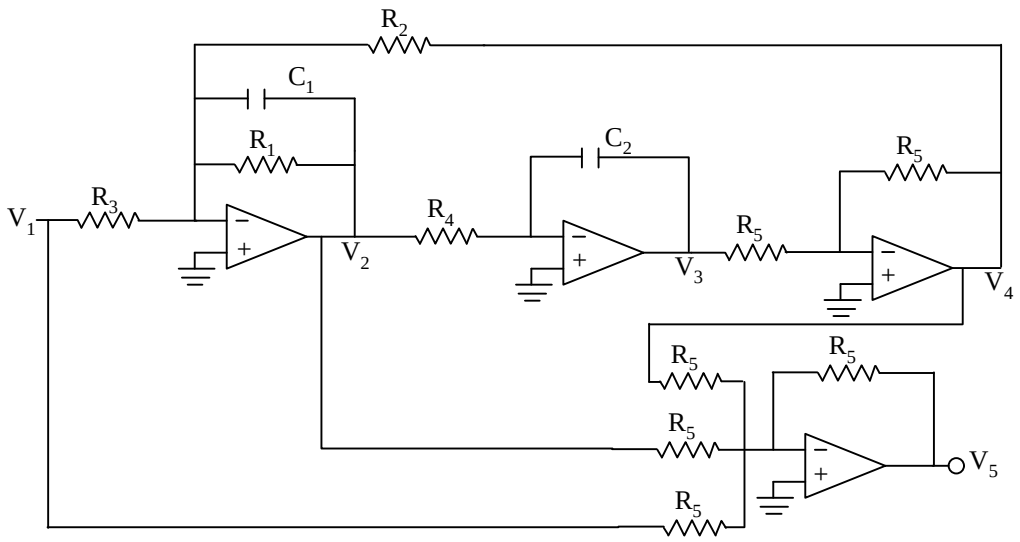

Figure A.4 Biquad circuit diagram for case 3.

The circuit transfer function is given by (A.12).

$$\frac{V_5}{V_1} = -\frac{s^2 + \left(\frac{1}{R_1C_1} - \frac{1}{R_3C_1}\right)s + \left(\frac{R_3 - R_2}{R_2R_3R_4C_1C_2}\right)}{s^2 + \left(\frac{1}{R_1C_1}\right)s + \left(\frac{1}{R_2R_4C_1C_2}\right)} \tag{A.12}$$

Comparing this transfer function to (A.1) yields four design equations.

$$R_1 = \frac{1}{2\rho_p\omega_{np}C_1} \tag{A.13}$$

$$R_2 = R_3\left(1 - \frac{\omega_{nz}^2}{\omega_{np}^2}\right) \tag{A.14}$$

$$R_3 = \frac{1}{2C_1\left(\rho_p\omega_{np} - \rho_z\omega_{nz}\right)} \tag{A.15}$$

$$R_4 = \frac{1}{R_2C_1C_2\omega_{np}^2} \tag{A.16}$$

A.5 Case 4: $\omega_{np}>\omega_{nz}$ and $\xi_z>\xi_p$

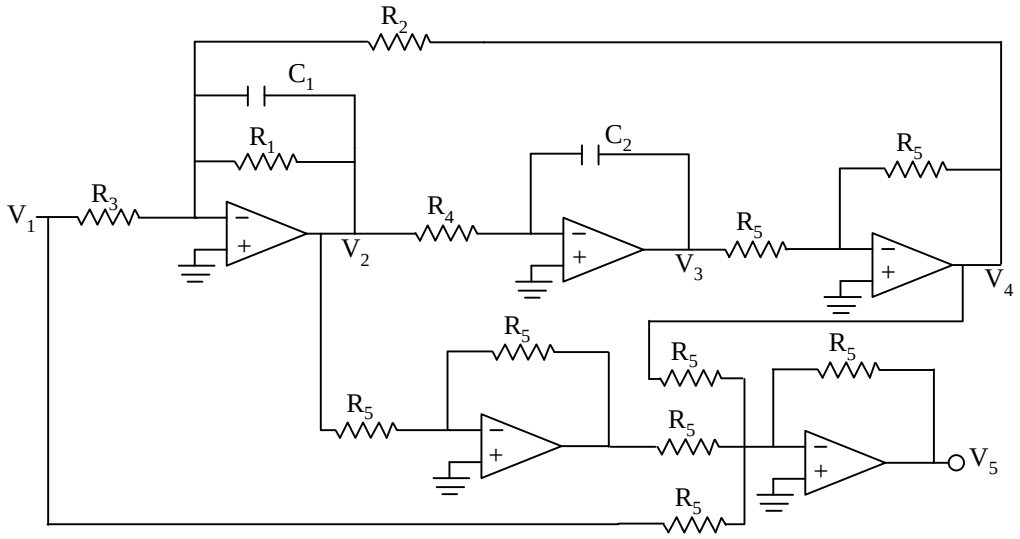

Figure A.5 Biquad circuit diagram for case 4.

The circuit transfer function is given by (A.17).

$$\frac{V_5}{V_1} = -\frac{s^2 + \left(\frac{1}{R_1C_1} + \frac{1}{R_3C_1}\right)s + \left(\frac{R_3 - R_2}{R_2R_3R_4C_1C_2}\right)}{s^2 + \left(\frac{1}{R_1C_1}\right)s + \left(\frac{1}{R_2R_4C_1C_2}\right)} \tag{A.17}$$

Comparing this transfer function to (A.1) yields four design equations.

$$R_1 = \frac{1}{2\rho_p\omega_{np}C_1} \tag{A.18}$$

$$R_2 = R_3\left(1 - \frac{\omega_{nz}^2}{\omega_{np}^2}\right) \tag{A.19}$$

$$R_3 = \frac{1}{2C_1\left(\rho_z\omega_{nz} - \rho_p\omega_{np}\right)} \tag{A.20}$$

$$R_4 = \frac{1}{R_2C_1C_2\omega_{np}^2} \tag{A.21}$$

A.6 'Free' capacitors and resistors in the biquad circuit

Some specific notes need to be made concerning the values of the resistors and capacitors that are used to design the previous circuits. First, in each case the values of the two capacitors are arbitrary. They provide the start point for each design. They should be chosen such that the resulting circuit has resistor values that are easy to implement, common values, and of moderate difference in magnitude. Second, the value of each resistor labeled R5 is arbitrary. Each R5 has to be the same, but they merely create a unity gain inverting op-amp. These should be chosen to be a standard moderate value. Finally, the circuits provided all yield negative transfer functions. This may be changed by simply adding an inverting op-amp to the output terminal.

7 APPENDIX B: MATLAB CODES

B.1 Tracking a single frequency signal as it passes through a digital system

ONEFREQ.M

```
% This tracks what happens to one single sine wave
% as it passes through the sampling, control, and
% D/A process.  The input frequency can only
% be four times the sampling rate max.

% Read in all the parameters
finput = eval(get(q2,'String'));
minput_lin = eval(get(qm2,'String'));
fs = eval(get(qfs2,'String'));
dt = 1/fs;
numaa = eval(get(q36b,'String'));
denaa = eval(get(q37b,'String'));
numcon = eval(get(q30,'String'));
dencon = eval(get(q32,'String'));
numsm = eval(get(q38b,'String'));
densm = eval(get(q39b,'String'));
ADlag = eval(get(qadlag2,'String'));
DAlag = eval(get(qdalag2,'String'));
ADdynrange = eval(get(q6,'String'));
DAdynrange = eval(get(q10,'String'));
if get(q8,'Value') == 1
   ADbits = 12;
elseif get(q8a,'Value') == 1
   ADbits = 16;
else
   ADbits = 20;
end
if get(q12,'Value') == 1
   DAbits = 12;
elseif get(q12a,'Value') == 1
   DAbits = 16;
else
   DAbits = 20;
end

% Run the signal through the anti-aliasing filter
minputdB = 20*log10(minput_lin);
AAout = freqs(numaa,denaa,[1 finput*2*pi]);
AAout_lin = abs(AAout(2))*minput_lin;
AAout_dB = 20*log10(AAout_lin);

% Determine where the frequency is in terms of
% aliasing regions.  Allow for the input to be up
% to 8 times the sampling frequency
if (finput >= 0  ) & (finput <= fs/2),
   fdig = finput; end
if (finput > fs/2) & (finput <= fs  ),
   fdig = fs/2-(finput-fs/2); end
if (finput > fs  ) & (finput <= 3*fs/2),
   fdig = finput-fs; end
if (finput > 3*fs/2) & (finput <= 2*fs),
   fdig = fs/2-(finput-3*fs/2); end
if (finput > 2*fs) & (finput <= 5*fs/2),
```

```
   fdig = finput-2*fs; end
if (finput > 5*fs/2) & (finput <= 3*fs  ),
   fdig = fs/2-(finput-5*fs/2); end
if (finput > 3*fs  ) & (finput <= 7*fs/2),
   fdig = finput-3*fs; end
if (finput > 7*fs/2) & (finput <= 4*fs),
   fdig = fs/2-(finput-7*fs/2); end
if (finput > 4*fs  ) & (finput <= 9*fs/2),
   fdig = finput-4*fs; end
if (finput > 9*fs/2) & (finput <= 5*fs),
   fdig = fs/2-(finput-9*fs/2); end
if (finput > 5*fs  ) & (finput <= 11*fs/2),
   fdig = finput-5*fs; end
if (finput > 11*fs/2) & (finput <= 6*fs),
   fdig = fs/2-(finput-11*fs/2); end
if (finput > 6*fs  ) & (finput <= 13*fs/2),
   fdig = finput-6*fs; end
if (finput > 13*fs/2) & (finput <= 7*fs),
   fdig = fs/2-(finput-13*fs/2); end
if (finput > 7*fs  ) & (finput <= 15*fs/2),
   fdig = finput-7*fs; end
if (finput > 15*fs/2) & (finput <= 8*fs),
   fdig = fs/2-(finput-15*fs/2); end
if (finput > 8*fs  ) & (finput <= 17*fs/2),
   fdig = finput-8*fs; end

% quantize the signal according to the A/D
Controlin_lin = quantize(AAout_lin,ADbits,ADdynrange);
Controlin_dB = 20*log10(Controlin_lin);

% Go through the controller
Controlout = freqz(numcon,dencon,[1 fdig*2*pi],fs);
Controlout_lin = abs(Controlout(2))*Controlin_lin;
Controlout_dB = 20*log10(Controlout_lin);

% D/A the signal
fzoh = [fdig fs-fdig fs+fdig 2*fs-fdig 2*fs+fdig...
      3*fs-fdig 3*fs+fdig 4*fs-fdig 4*fs+fdig...
      5*fs-fdig 5*fs+fdig 6*fs-fdig 6*fs+fdig...
      7*fs-fdig 7*fs+fdig 8*fs-fdig 8*fs+fdig];
zohout_lin                    =                  Controlout_lin*fs*abs((1-exp(-
i*(2*pi*fzoh)*(1/fs))./(i*(2*pi*fzoh)));

% Quantize the signal
zohout_lin = quantize(zohout_lin,DAbits,DAdynrange);
zohout_dB = 20*log10(zohout_lin);

% Run the signal through the smoothing filter
SMout = freqs(numsm,densm,fzoh*2*pi);
SMout_lin = abs(SMout).*zohout_lin;
SMout_dB = 20*log10(SMout_lin);

% Generate the plots specified by the user

if get(q15,'Value') == 1
   figure;
   stem(finput,minput_lin,'b')
   title('Input Voltage Spectrum')
   ylabel('Magnitude, V')
   xlabel('Frequency, Hz')
   axis([0 9*fs 0 1.05*minput_lin]);
   grid on;
end
```

```
if get (q22,'Value') == 1
   figure;
   [AAmag,AAp,AAw]=bode(numaa,denaa,[2*pi:2*pi*4:2*pi*9*fs]);
   subplot(211),semilogx(AAw/2/pi,AAmag,'b')
   title('Anti-Aliasing Filter FRF')
   ylabel('Linear Magnitude')
   xlabel('Frequency, Hz')
   grid on;
   subplot(212),semilogx(AAw/2/pi,AAp,'b')
   ylabel('Phase, degrees')
   xlabel('Frequency, Hz')
   grid on;
end
if get(q17,'Value') == 1
   figure;
   stem(finput,AAout_lin,'b')
   title('Output From Anti-Aliasing Filter')
   ylabel('Magnitude, V')
   xlabel('Frequency, Hz')
   axis([0 9*fs 0 1.05*minput_lin]);
   grid on;
end
if get(q18,'Value') == 1
   figure;
   stem(fdig,Controlin_lin,'b')
   title('Signal After A/D Conversion')
   axis([0 9*fs 0 1.05*minput_lin]);
   xlabel('Frequency, Hz')
   ylabel('Magnitude, V')
   grid on;
end
if get(q19,'Value') == 1
   figure;
   [Cmag,Cp,Cw]=dbode(numcon,dencon,1/fs,[2*pi:2*pi*4:2*pi*9*fs]);
   subplot(211),semilogx(Cw/2/pi,Cmag,'b')
   title('Control Filter FRF')
   ylabel('Linear Magnitude')
   xlabel('Frequency, Hz')
   grid on;
   subplot(212),semilogx(Cw/2/pi,Cp,'b')
   ylabel('Phase, degrees')
   xlabel('Frequency, Hz')
   grid on;
end
if get(q20,'Value') == 1
   figure;
   stem(fdig,Controlout_lin,'b')
   title('Output From Controller')
   axis([0 9*fs 0 max(Controlout_lin)]);
   xlabel('Frequency, Hz')
   ylabel('Magnitude, V')
   grid on;
end
if get(q20a,'Value') == 1
   figure;
   ampmod = [fs*Controlout_lin fs*Controlout_lin...
         fs*Controlout_lin fs*Controlout_lin...
         fs*Controlout_lin fs*Controlout_lin...
         fs*Controlout_lin fs*Controlout_lin...
         fs*Controlout_lin fs*Controlout_lin...
         fs*Controlout_lin fs*Controlout_lin...
         fs*Controlout_lin fs*Controlout_lin...
```

```
         fs*Controlout_lin];
   stem(fzoh,ampmod,'b')
   title('Amplitude Modulated Control Signal')
   axis([0 9*fs 0 1.05*ampmod(1)]);
   ylabel('Magnitude, V')
   grid on;
end
if get(q20b,'Value') == 1
   figure;
   f = 0:2:9*fs;
   zoh = (fs)*abs((1-exp(-i*(2*pi*f)/fs))./(i*(2*pi*f)));
   plot(f,zoh,'b')
   title('ZOH Filter FRF')
   axis([0 9*fs 0 1]);
   ylabel('Linear Magnitude')
   xlabel('Frequency, Hz')
   grid on;
end
if get(q21,'Value') == 1
   figure;
   stem(fzoh,zohout_lin,'b')
   title('D/A Output')
   ylabel('Magnitude, V')
   xlabel('Frequency, Hz')
   axis([0 9*fs 0 1.05*minput_lin]);
   grid on;
end
if get (q22,'Value') == 1
   figure;
   [SMmag,SMp,SMw]=bode(numsm,densm,[2*pi:2*pi*4:2*pi*9*fs]);
   subplot(211),semilogx(SMw/2/pi,SMmag,'b')
   title('Smoothing Filter FRF')
   ylabel('Linear Magnitude')
   xlabel('Frequency, Hz')
   grid on;
   subplot(212),semilogx(SMw/2/pi,SMp,'b')
   ylabel('Phase, degrees')
   xlabel('Frequency, Hz')
   grid on;
end
if get(q23,'Value') == 1
   figure;
   stem(fzoh,SMout_lin,'b')
   title('Output From Smoothing Filter')
   xlabel('Frequency, Hz')
   ylabel('Magnitude, V')
   axis([0 9*fs 0 1.05*minput_lin]);
   grid on;
end

% Calculate the phase loss in the system
pZOH = -(fdig*dt/2)*180/pi;
pAD = -(ADlag*fdig*360);
pDA = -(DAlag*fdig*360);
[mAA,pAA,fAA]=bode(numaa,denaa,[1 2*fdig*pi]);
[mC,pC,fC]=bode(numcon,dencon,[1 2*pi*fdig]);
[mSM,pSM,fSM]=bode(numsm,densm,[1 2*pi*fdig]);
[totalphase]=pAA(2)+pC(2)+pSM(2)+pZOH+pAD+pDA;

% Feedback the artifact frequencies and output amplitudes
fzohq = quantize(fzoh,1,2);
SMout_linq = quantize(SMout_lin,1,.02);
totalphaseq = quantize(totalphase,1,2);
```

```
set(q40b,'String',[num2str(fzohq(1)),',',...
      num2str(fzohq(2)),', ',num2str(fzohq(3))]);
set(q40bb,'String',[num2str(fzohq(4)),', ',...
      num2str(fzohq(5)),', ',num2str(fzohq(6))]);
set(q40bbb,'String',[num2str(fzohq(7)),', ',...
      num2str(fzohq(8)),', ',num2str(fzohq(9))]);
set(q40d,'String',[num2str(SMout_linq(1:3))]);
set(q40dd,'String',[num2str(SMout_linq(4:6))]);
set(q40ddd,'String',[num2str(SMout_linq(7:9))]);
set(q40ee,'String',num2str(totalphaseq));
```

B.2 Tracking a signal as it passes through a digital system

FRF.M

```
% This mfile tracks an input spectrum through
% an ANR heaset.  The input spectrum must be
% from 0 to fs at least. Data must be a regularly
% spaced frequency intervals ans fn must fall on
% a sampled frequency. The file handles aliasing
% of frequencies upto 4 times the sample frequency.
% The GUI named desgui.m acts as a user interface.
% Date may be in the form of a sound field in
% SPLdB or in microphone voltage.

% load the data
directory = get(q26,'String');
filename = get(q28,'String');
load([directory filename '.txt']);
q = eval(filename);
freqdata = [1 q((2:length(q)),1)'];
L = (1*10^(-6))*eval(get(qL2,'String'));
   % voltage level that speaker won't make sound at
x = eval(get(q2,'String'));
   % this represents the maximum frequency of interest
dynrangeAD = eval(get(q6,'String'));
   % dynamic range of the A/D converter in volts
dynrangeDA = eval(get(q10,'String'));
   % dynamic range of the D/A converter in volts
micref = eval(get(q4,'String'));
   % voltage reference of microphone
   % RS = 9.44*10^(-8) BK = 8.8*10^(-7)
micgain = eval(get(qamp2,'String'));
   % linear amplification of microphone signal

% Check to see what type the data is (SPLdB or Volts)
if get(qvorSPLb,'Value') == 1;
   Vi = micgain*q(:,2)';
   SPLdatadB = 20*log10(Vi./micref);
   VidB = 20*log10(Vi);
else
   SPLdatadB = q(:,2)';
   Vi = micgain*micref*10.^(SPLdatadB/20);
   VidB = 20*log10(Vi);
end

% Read in A/D and D/A parameters
if get(q8,'Value') == 1
   bitsAD = 12;
elseif get(q8a,'Value') == 1
   bitsAD = 16;
```

```
else
   bitsAD = 20;
end
if get(q12,'Value') == 1
   DAbits = 12;
elseif get(q12a,'Value') == 1
   DAbits = 16;
else
   DAbits = 20;
end
ADresolution = dynrangeAD/(2^bitsAD);
   % below this the A/D won't register
ADreferencedB = 20*log10(ADresolution);

% Read in the sampling frequency
fs = eval(get(qfs2,'String'));
   % the sampling rate in hz
fn = fs/2;
   % Nyquist frequency in hz
fint = freqdata(9)-freqdata(8);
   % the spacing of the frequencies to be analyzed in hz
aliasf = fs-x;
   % highest allowable frequency to alias

% Split the input spectrum into frequency
%intervals of fint hz
freq0tox=[];, freqxtofn=[];, freqfntoaliasf=[];
freqaliasftofs=[];, freqabovefs=[];
for r = 1:length(freqdata),
     if freqdata(r) <= x
        freq0tox = [freq0tox freqdata(r)];
     elseif freqdata(r) <= fn
           freqxtofn = [freqxtofn freqdata(r)];
     elseif freqdata(r) <= aliasf
           freqfntoaliasf = [freqfntoaliasf freqdata(r)];
     elseif freqdata(r) <= fs
           freqaliasftofs = [freqaliasftofs freqdata(r)];
     else
           freqabovefs = [freqabovefs freqdata(r)];
     end
end
freq = [freq0tox freqxtofn freqfntoaliasf freqaliasftofs freqabovefs];

% Create some easy indexes to frequencies
freqalias = [freq0tox freqxtofn freqfntoaliasf];
fai = length(freqalias);
   % index on highest frequency allowed to alias (fs-x)
fxi= length(freq0tox);
   % index of frequency referred to as x
fni = length(freq0tox)+length(freqxtofn);
   % index of frequency fn
fsi = length([freqalias freqaliasftofs]);
   % index to fs
fendi = length(freq);
   % index to whole frequency spectrum
fslessf = (fai-1):-1:fni;
   % backwards index from aliasf to fn
alias = (fsi):-1:fni;
   % backwards index from fs to fni

% Characterize anti-aliasing filter
numaa = eval(get(q36b,'String'));
denaa = eval(get(q37b,'String'));
```

```
AAlin = freqs(numaa,denaa,freq*2*pi);
AAdB = 20*log10(AAlin);

% Calculate what happens to all of the frequencies
% as they pass through the AA filter
AAoutdB = VidB+AAdB;
AAoutlin = 10.^(AAoutdB/20);

% Add the effects of the A/D converter.
% This is where aliasing takes place.  Lets take
% into consideration frequencies upto 4*fs again.
% Anything in the regions done above should not
% alias into x but lets assume they could anyway
% and program for the case in which they would.
% This makes integration into other program without the design
% criteria a lot easier. First create indexes to the
% frequencies that define the different regions that alias
fsx1i=1;,fn2i=1;,fsx2i=1;,fs2i=1;,fsx3i=1;,fn3i=1;,fsx4i=1;
fs3i=1;,fsx5i=1;,fn4i=1;,fsx6i=1;,fs4i=1;,fsx7i=1;

for z = 1 : fendi,
   if freq(z) == fs+x
      fsx1i = z;
   elseif freq(z) == 3*fn
      fn2i = z;
   elseif freq(z) == 2*fs-x
      fsx2i = z;
   elseif freq(z) == 2*fs
      fs2i = z;
   elseif freq(z) == 2*fs+x
      fsx3i = z;
   elseif freq(z) == 5*fn
      fn3i = z;
   elseif freq(z) == 3*fs-x
      fsx4i = z;
   elseif freq(z) == 3*fs
      fs3i = z;
   elseif freq(z) == 3*fs+x
      fsx5i = z;
   elseif freq(z) == 7*fn
      fn4i = z;
   elseif freq(z) == 4*fs-x
      fsx6i = z;
   elseif freq(z) == 4*fs
      fs4i = z;
   elseif freq(z) == 4*fs+x
      fsx7i = z;
   end
end
% Now check to see if anything aliases into the x region
% put if statements around the counters to signal
% if the frequencies are not present.  Frequencies are not
% present when their counters are 1.  Also allow for the
% frequencies present to stop anywhere not just at an index.
Di0tox = AAoutlin(1:fxi);
if fai > 1
   counter=0;
   if fsi == 1
      fsi = fendi;
   end
   for z = fsi:-1:fai,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
```

```
         Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
      end
   end
end
if fsi > 1
   counter=1;
   if fsx1i == 1
      fsx1i = fendi;
   end
   for z = fsi+1:fsx1i,
       counter = 1+counter;
       if AAoutdB(z) > ADreferencedB
          Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
       end
    end
end
if fsx2i > 1
   counter=0;
   if fs2i == 1
      fs2i = fendi;
      counter = fsi-fai+2-length(fsx2i:fendi)
   end
   for z = fs2i:-1:fsx2i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
      end
   end
end
if fs2i > 1
   counter=1;
   if fsx3i == 1
      fsx3i = fendi;
   end
   for z = fs2i+1:fsx3i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
      end
   end
end
if fsx4i > 1
   counter=0;
   if fs3i == 1
      fs3i = fendi;
      counter = fsi-fai+2-length(fsx4i:fendi)
   end
   for z = fs3i:-1:fsx4i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
      end
   end
end
if fs3i > 1
   counter=1;
   if fsx5i == 1
      fsx5i = fendi;
   end
   for z = fs3i+1:fsx5i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
```

```
      end
   end
end
if fsx6i > 1
   counter=0;
   if fs4i == 1
      fs4i = fendi;
      counter = fsi-fai+2-length(fsx6i:fendi);
   end
   for z = fs4i:-1:fsx6i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
      end
   end
end
if fs4i > 1
   counter=1;
   if fsx7i == 1
      fsx7i = fendi;
   end
   for z = fs4i+1:fsx7i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
      end
   end
end
% check to see if anything aliases into the x to fn region
Dixtofn = AAoutlin((fxi+1):fni)+AAoutlin(fslessf);
if fni > 1
   counter=0;
   for z = fai-1:-1:fni,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Dixtofn(counter)=AAoutlin(z)+Dixtofn(counter);
      end
   end
end
if fsx1i > 1
   counter=0;
   if fn2i == 1
      fn2i = fendi;
   end
   for z = fsx1i+1:fn2i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Dixtofn(counter)=AAoutlin(z)+Dixtofn(counter);
      end
   end
end
if fn2i > 1
   counter=1;
   if fsx2i == 1
      fsx2i = fendi+1;
      counter = fai-fni-length(fn2i+1:fendi);
   end
   for z = fsx2i-1:-1:fn2i+1,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Dixtofn(counter)=AAoutlin(z)+Dixtofn(counter);
      end
   end
```

```
end
if fsx3i > 1
   counter=0;
   if fn3i == 1
      fn3i = fendi;
   end
   for z = fsx3i+1:fn3i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Dixtofn(counter)=AAoutlin(z)+Dixtofn(counter);
      end
   end
end
if fn3i > 1
   counter=1;
   if fsx4i == 1
      fsx4i = fendi+1;
      counter = fai-fni-length(fn3i+1:fendi);
   end
   for z = fsx4i-1:-1:fn3i+1,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Dixtofn(counter)=AAoutlin(z)+Dixtofn(counter);
      end
   end
end
if fsx5i > 1
   counter=0;
   if fn4i == 1
      fn4i = fendi;
   end
   for z = fsx5i+1:fn4i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Dixtofn(counter)=AAoutlin(z)+Dixtofn(counter);
      end
   end
end
if fn4i > 1
   counter=1;
   if fsx6i == 1
      fsx6i = fendi+1;
      counter = fai-fni-length(fn4i+1:fendi);
   end
   for z = fsx6i-1:-1:fn4i+1,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Dixtofn(counter)=AAoutlin(z)+Dixtofn(counter);
      end
   end
end
Di0toxdB = 20*log10(Di0tox);
DixtofndB = 20*log10(Dixtofn);
Dilin = [Di0tox Dixtofn];
DidB = 20*log10(Dilin);
freqdig = [freq0tox freqxtofn];


% Define the controller characteristics
numz = get(q30,'String');
denz = get(q32,'String');
[C] = freqz(numz,denz,2*pi*freqdig);
```

```
% Add the controller into the loop now
Dolin = C.*Dilin;
DodB = 20*log10(Dolin);

% Add the D/A to the loop
% First lets simply do the amplitude modulation
% half of the process
Dostar0tofn = Dolin*(fs);
freq0tofn = [freq0tox freqxtofn];
Dostarfntofs = Dolin(length(Dolin):-1:1)*fs;
freqfntofs = [fn freqfntoaliasf freqaliasftofs];
Dostarfsto3fn = Dostar0tofn;
freqfsto3fn = fs+[0:fint:fint*(length(freqfntofs)-1)];
Dostar3fnto2fs = Dostarfntofs;
freq3fnto2fs = 3*fn+[0:fint:fint*(length(freqfntofs)-1)];
Dostarlin = [Dostar0tofn Dostarfntofs Dostarfsto3fn Dostar3fnto2fs];
DostardB = 20*log10(Dostarlin);
freqDostar = [freq0tofn freqfntofs freqfsto3fn freq3fnto2fs];

% Now lets do the zoh half of the process
zoh = abs((1-exp(-i*(2*pi*freqDostar)*(1/fs))./(i*(2*pi*freqDostar)));
zohlin = Dostarlin.*zoh;
zohdB = 20*log10(zohlin);

% The final piece of the puzzle is the smoothing filter.
% Design criteria
%             1)0dB pass band from 0 to xHz.
%             2)output must be below floor of speaker by fn
%               so that we don't output frequencies that are
%               fn which appear only because of the D/A process.
% Characterize smoothing filter
numsm = eval(get(q38b,'String'));
densm = eval(get(q39b,'String'));
SM = freqs(numsm,densm,freqDostar*2*pi);
SMdB = 20*log10(SM);
VodB = SMdB+zohdB;
Volin = 10.^(VodB/20);

% Generate the plots specified by the user

if get(q15,'Value') == 1
   figure;
   semilogx(freq,VidB,'b')
   title('Input Voltage Spectrum')
   ylabel('Magnitude, dB')
   %axis tight;
   grid on;
end
if get(q15a,'Value') == 1
   figure;
   semilogx(freq,SPLdatadB,'b')
   title('Input SPL Spectrum')
   ylabel('SPL, dB')
   %axis tight;
   grid on;
end
if get(q16,'Value') == 1
   figure;
   semilogx(freq,AAdB,'b')
   title('Anti-Aliasng Filter FRF')
   ylabel('Magnitude, dB')
   %axis tight;
   grid on;
```

```
end
if get(q17,'Value') == 1
   figure;
   semilogx(freq,AAoutdB,'b',[aliasf aliasf],...
      [max(AAoutdB) ADreferencedB],'--r',[x x],...
      [max(AAoutdB) ADreferencedB],'--r',[fn fn],...
      [max(AAoutdB) ADreferencedB],'--r',[x aliasf],...
      [ADreferencedB ADreferencedB],'--r')
   title('Output From Anti-Aliasng Filter')
   ylabel('Magnitude, dB')
   xlabel(...
      ['Frequency, Hz','           Range of Interest = ',...
         num2str(x),'    Sampling Frequency = ',num2str(fs)])
   text(x,(ADreferencedB-15),'x');
   text(aliasf,(ADreferencedB-15),'(fs-x)')
   text(2*x,(ADreferencedB+10),'Res.')
   text(fn,(ADreferencedB-15),'fn')
   grid on;
end
if get(q18,'Value') == 1
   figure;
   semilogx(freq,AAoutdB,'g:',freqdig,DidB,'b',...
      [x aliasf],[ADreferencedB ADreferencedB],'--r')
   title('Signal Before and After A/D Conversion')
   xlabel(...
      ['Frequency, Hz','           Range of Interest = ',...
         num2str(x),'    Sampling Frequency = ',num2str(fs)])
   ylabel('Magnitude, dB')
   legend('Analog','Digital','A/D res.')
   grid on;
end
if get(q19,'Value') == 1
   figure;
   semilogx(freqdig,20*log10(C),'b')
   title('Control Filter FRF')
   ylabel('Magnitude, dB')
   grid on;
end
if get(q20,'Value') == 1
   figure;
   semilogx(freqdig,DodB,'b')
   title('Output From Controller')
   xlabel(...
      ['Frequency, Hz','           Range of Interest = ',...
         num2str(x),'    Sampling Frequency = ',num2str(fs)])
   ylabel('Magnitude, dB')
   grid on;
end
if get(q20a,'Value') == 1
   figure;
   semilogx(freqDostar,DostardB,'b')
   title('Amplitude Modulated Control Signal')
   ylabel('Magnitude, dB')
   grid on;
end
if get(q20b,'Value') == 1
   figure;
   semilogx(freqDostar,20*log10(zoh),'b')
   title('ZOH Filter FRF')
   ylabel('Magnitude')
   grid on;
end
if get(q21,'Value') == 1
```

```
   figure;
   semilogx(freqDostar,zohdB,'b')
   title('D/A Output')
   ylabel('Magnitude')
   xlabel(...
      ['Frequency, Hz','           Range of Interest = ',...
         num2str(x),'    Sampling Frequency = ',num2str(fs)])
   %axis tight;
   grid on;
end
if get (q22,'Value') == 1
   figure;
   semilogx(freqDostar,SMdB,'b')
   title('Smoothing Filter FRF')
   ylabel('Magnitude, dB')
   %%axis tight;
   grid on;
end
if get(q23,'Value') == 1
   figure;
   semilogx(freqDostar,VodB,'b',[x 2*fs],[20*log10(L) 20*log10(L)],'--r')
   title('Output From Smoothing Filter')
   xlabel(...
      ['Frequency, Hz','           Range of Interest = ',...
         num2str(x),'    Sampling Frequency = ',num2str(fs)])
   ylabel('Magnitude, dB')
   text(x,(20*log10(L)-10),'x');
   text(2*x,(20*log10(L)+10),'L')
   text(fn,(20*log10(L)-10),'fn')
   grid on;
end
```

B.3 Calculating the anti-aliasing filter slope and smoothing filter slope

DES.M

```
% des.m
% This mfile tracks an input spectrum open-loop through an ANR heaset.
% The input spectrum must be from 0 to fs at least. Data must be at regularly
% spaced frequency intervals and fn must fall on a sampled frequency.
% The file handles aliasing of frequencies upto 4 times the sample frequency.
% The GUI named desgui.m acts as a user interface.
% Date may be in the form of a sound field in SPLdB or in microphone voltage.

% clear the slopes on the gui
set(qaa,'String',' ');
set(qsm,'String',' ');

% load the data
directory = get(q26,'String');
filename = get(q28,'String');
load([directory filename '.txt']);
q = eval(filename);
freqdata = [1 q((2:length(q)),1)'];
L = (1*10^(-3))*eval(get(qL2,'String'));
% voltage level that speaker will not produce sound at
x = eval(get(q2,'String'));
% this represents the maximum frequency of interest
dynrangeAD = eval(get(q6,'String'));
% dynamic range of the A/D converter in volts
```

```
dynrangeDA = eval(get(q10,'String'));
% dynamic range of the D/A converter in volts
micref = eval(get(q4,'String'));
% voltage reference of microphone  RS = 9.44*10^(-8) BK = 8.8*10^(-7)
micgain = eval(get(qamp2,'String'));
% linear amplification of microphone signal

% Check to see what type the data is (SPLdB or Volts)
if get(qvorSPLb,'Value') == 1;
   Vi = micgain*q(:,2)';
   SPLdatadB = 20*log10(Vi./micref);
   VidB = 20*log10(Vi);
else
   SPLdatadB = q(:,2)';
   Vi = micgain*micref*10.^(SPLdatadB/20);
   VidB = 20*log10(Vi);
end

% Read in A/D and D/A parameters
if get(q8,'Value') == 1
   bitsAD = 12;
elseif get(q8a,'Value') == 1
   bitsAD = 16;
else
   bitsAD = 20;
end
if get(q12,'Value') == 1
   DAbits = 12;
elseif get(q12a,'Value') == 1
   DAbits = 16;
else
   DAbits = 20;
end
ADresolution = dynrangeAD/(2^bitsAD);     % below this the A/D won't register
ADreferencedB = 20*log10(ADresolution);

% Read in the sampling frequency
fs = eval(get(qfs2,'String'));          % the sampling rate in hz
fn = fs/2;                              % Nyquist frequency in hz
fint = freqdata(9)-freqdata(8);
% the spacing of the frequencies to be analyzed in hz
aliasf = fs-x;                          % highest allowable frequency to alias

% Split the input spectrum into frequency intervals of fint hz
freq0tox=[];,     freqxtofn=[];,     freqfntoaliasf=[];,     freqaliasftofs=[];,
freqabovefs=[];
for r = 1:length(freqdata),
     if freqdata(r) <= x
        freq0tox = [freq0tox freqdata(r)];
     elseif freqdata(r) <= fn
           freqxtofn = [freqxtofn freqdata(r)];
     elseif freqdata(r) <= aliasf
           freqfntoaliasf = [freqfntoaliasf freqdata(r)];
     elseif freqdata(r) <= fs
           freqaliasftofs = [freqaliasftofs freqdata(r)];
     else
           freqabovefs = [freqabovefs freqdata(r)];
     end
end
freq = [freq0tox freqxtofn freqfntoaliasf freqaliasftofs freqabovefs];

% Create some easy indexes to frequencies
freqalias = [freq0tox freqxtofn freqfntoaliasf];
```

```
fai = length(freqalias);
% index on highest frequency allowed to alias (fs-x)
fxi= length(freq0tox);
% index of frequency referred to as x
fni = length(freq0tox)+length(freqxtofn);     % index of frequency fn
fsi = length([freqalias freqaliasftofs]);     % index to fs
fendi = length(freq);
% index to whole frequency spectrum
fslessf = (fai-1):-1:fni;
% backwards index from aliasf to fn
alias = (fsi):-1:fni;                         % backwards index from fs to fni

% Calculate the slope of the anti-aliasing filter.
% Criteria:  1)0dB pass band from 0 to x Hz.
%            2)Output from filter must be below ADreferencedB by fs-x to avoid
aliasing
Maa = (VidB(fai)-ADreferencedB)/(log10(x)-log10(aliasf));
Baa = -Maa*log10(x);
if Maa > 0
   Maa = 0;
   Baa = 0;
end

% Define the characteristics of the AA filter at the frequencies of interest
AAdB0tox = zeros(1,length(freq0tox));
AAdBabovex = Maa.*log10(freq(fxi+1:fendi))+Baa;
AAdB = [AAdB0tox AAdBabovex];

% Calculate what happens to all of the frequencies as they pass through the AA
filter
AAoutdB = VidB+AAdB;
AAoutlin = 10.^(AAoutdB/20);

% Lets check all the frequencies above falias to see if they truly are reduced
% below the bit level and won't alias into our region.  Need to do this because
% spectrum might not be low pass in nature.
% First check region from falias to fs
zz = 10;
while zz > 1
   zz = 10;
   for z = (fai+1) : fsi,
      if AAoutdB(z) > ADreferencedB
         diff = AAoutdB(z)-ADreferencedB;
         Maa = (VidB(fai)+diff-ADreferencedB)/(log10(x)-log10(freq(fai)));
         Baa = -Maa*log10(x);
         AAdBabovex = Maa.*log10(freq(fxi+1:fendi))+Baa;
         AAdB = [AAdB0tox AAdBabovex];
         AAoutdB = VidB+AAdB;
         AAoutlin = 10.^(AAoutdB/20);
         zz=100;
      end
   end
   if zz < 50
      zz = 0;
   end
end

% Now lets check the frequencies above fs.  We only need to check in certain
region.
% A)fs to fs+x   B)2fs-x to 2fs+x  C)3fs-x to 3fs+x    D)4fs-x to 4fs+x
% More regions cab be added if frequencies above 4*fs are important
zz = 10;
while zz > 1
```

```
   zz = 10;
   for z = fsi+1 : fendi,
      if freq(z) < fs+x
         if AAoutdB(z) > ADreferencedB
            diff = AAoutdB(z)-ADreferencedB;
            Maa = (VidB(fai)+diff-ADreferencedB)/(log10(x)-log10(freq(fai)));
            Baa = -Maa*log10(x);
            AAdBabovex = Maa.*log10(freq(fxi+1:fendi))+Baa;
            AAdB = [AAdB0tox AAdBabovex];
            AAoutdB = VidB+AAdB;
            AAoutlin = 10.^(AAoutdB/20);
            zz = 100;
         end
      end
      if freq(z) > 2*fs-x
         if freq(z) < 2*fs+x
            if AAoutdB(z) > ADreferencedB
               diff = AAoutdB(z)-ADreferencedB;
               Maa          =          (VidB(fai)+diff-ADreferencedB)/(log10(x)-
log10(freq(fai)));
               Baa = -Maa*log10(x);
               AAdBabovex = Maa.*log10(freq(fxi+1:fendi))+Baa;
               AAdB = [AAdB0tox AAdBabovex];
               AAoutdB = VidB+AAdB;
               AAoutlin = 10.^(AAoutdB/20);
               zz = 100;
            end
         end
      end
      if freq(z) > 3*fs-x
         if freq(z) < 3*fs+x
            if AAoutdB(z) > ADreferencedB
               diff = AAoutdB(z)-ADreferencedB;
               Maa          =          (VidB(fai)+diff-ADreferencedB)/(log10(x)-
log10(freq(fai)));
               Baa = -Maa*log10(x);
               AAdBabovex = Maa.*log10(freq(fxi+1:fendi))+Baa;
               AAdB = [AAdB0tox AAdBabovex];
               AAoutdB = VidB+AAdB;
               AAoutlin = 10.^(AAoutdB/20);
               zz = 100;
            end
         end
      end
      if freq(z) > 4*fs-x
         if freq(z) < 4*fs+x
            if AAoutdB(z) > ADreferencedB
               diff = AAoutdB(z)-ADreferencedB;
               Maa          =          (VidB(fai)+diff-ADreferencedB)/(log10(x)-
log10(freq(fai)));
               Baa = -Maa*log10(x);
               AAdBabovex = Maa.*log10(freq(fxi+1:fendi))+Baa;
               AAdB = [AAdB0tox AAdBabovex];
               AAoutdB = VidB+AAdB;
               AAoutlin = 10.^(AAoutdB/20);
               zz = 1;
            end
         end
      end
   end
   if zz < 50
      zz = 0;
   end
```

```
end

% This next line is if you want to design the smoothing filter in the absence
% of an anti-aliasing filter only
aa_yes_no = get(q_aa3,'Value');
if aa_yes_no == 1
 AAdB=AAdB*0;
 AAoutdB = VidB+AAdB;
 AAoutlin = 10.^(AAoutdB/20);
 Maa = 0;
end

% Add the effects of the A/D converter.  This is where aliasing takes place.
Lets take
% into consideration frequencies upto 4*fs again.  Anything in the regions done
above
% should not alias into x but lets assume they could anyway and program for the
case
% in which they would.  This makes integration into other program without the
design
% criteria a lot easier.
% First create indexes to the frequencies that define the different regions
that alias
fsx1i=1;,fn2i=1;,fsx2i=1;,fs2i=1;,fsx3i=1;,fn3i=1;,fsx4i=1;
fs3i=1;,fsx5i=1;,fn4i=1;,fsx6i=1;,fs4i=1;,fsx7i=1;%

for z = 1 : fendi,
  if freq(z) == fs+x
      fsx1i = z;
   elseif freq(z) == 3*fn
      fn2i = z;
   elseif freq(z) == 2*fs-x
      fsx2i = z;
   elseif freq(z) == 2*fs
      fs2i = z;
   elseif freq(z) == 2*fs+x
      fsx3i = z;
   elseif freq(z) == 5*fn
      fn3i = z;
   elseif freq(z) == 3*fs-x
      fsx4i = z;
   elseif freq(z) == 3*fs
      fs3i = z;
   elseif freq(z) == 3*fs+x
      fsx5i = z;
   elseif freq(z) == 7*fn
      fn4i = z;
   elseif freq(z) == 4*fs-x
      fsx6i = z;
   elseif freq(z) == 4*fs
      fs4i = z;
   elseif freq(z) == 4*fs+x
      fsx7i = z;
   end
end
% Now check to see if anything aliases into the x region
% put if statements around the counters to signal
% if the frequencies are not present.  Frequencies are not
% present when their counters are 1.  Also allow for the frequencies
% present to stop anywhere not just at an index.
Di0tox = AAoutlin(1:fxi);
if fai > 1
   counter=0;
```

```
   if fsi == 1
      fsi = fendi;
   end
   for z = fsi:-1:fai,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
      end
   end
end
if fsi > 1
   counter=1;
   if fsx1i == 1
      fsx1i = fendi;
   end
   for z = fsi+1:fsx1i,
       counter = 1+counter;
       if AAoutdB(z) > ADreferencedB
          Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
       end
    end
end
if fsx2i > 1
   counter=0;
   if fs2i == 1
      fs2i = fendi;
      counter = fsi-fai+2-length(fsx2i:fendi)
   end
   for z = fs2i:-1:fsx2i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
      end
   end
end
if fs2i > 1
   counter=1;
   if fsx3i == 1
      fsx3i = fendi;
   end
   for z = fs2i+1:fsx3i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
      end
   end
end
if fsx4i > 1
   counter=0;
   if fs3i == 1
      fs3i = fendi;
      counter = fsi-fai+2-length(fsx4i:fendi)
   end
   for z = fs3i:-1:fsx4i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
      end
   end
end
if fs3i > 1
   counter=1;
   if fsx5i == 1
```

```
      fsx5i = fendi;
   end
   for z = fs3i+1:fsx5i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
      end
   end
end
if fsx6i > 1
   counter=0;
   if fs4i == 1
      fs4i = fendi;
      counter = fsi-fai+2-length(fsx6i:fendi);
   end
   for z = fs4i:-1:fsx6i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
      end
   end
end
if fs4i > 1
   counter=1;
   if fsx7i == 1
      fsx7i = fendi;
   end
   for z = fs4i+1:fsx7i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Di0tox(counter)=AAoutlin(z)+Di0tox(counter);
      end
   end
end
% Now check to see if anything aliases into the x to fn region
Dixtofn = AAoutlin((fxi+1):fni)+AAoutlin(fslessf);
if fni > 1
   counter=0;
   for z = fai-1:-1:fni,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Dixtofn(counter)=AAoutlin(z)+Dixtofn(counter);
      end
   end
end
if fsx1i > 1
   counter=0;
   if fn2i == 1
      fn2i = fendi;
   end
   for z = fsx1i+1:fn2i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Dixtofn(counter)=AAoutlin(z)+Dixtofn(counter);
      end
   end
end
if fn2i > 1
   counter=1;
   if fsx2i == 1
      fsx2i = fendi+1;
      counter = fai-fni-length(fn2i+1:fendi);
   end
```

```
   for z = fsx2i-1:-1:fn2i+1,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Dixtofn(counter)=AAoutlin(z)+Dixtofn(counter);
      end
   end
end
if fsx3i > 1
   counter=0;
   if fn3i == 1
      fn3i = fendi;
   end
   for z = fsx3i+1:fn3i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Dixtofn(counter)=AAoutlin(z)+Dixtofn(counter);
      end
   end
end
if fn3i > 1
   counter=1;
   if fsx4i == 1
      fsx4i = fendi+1;
      counter = fai-fni-length(fn3i+1:fendi);
   end
   for z = fsx4i-1:-1:fn3i+1,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Dixtofn(counter)=AAoutlin(z)+Dixtofn(counter);
      end
   end
end
if fsx5i > 1
   counter=0;
   if fn4i == 1
      fn4i = fendi;
   end
   for z = fsx5i+1:fn4i,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Dixtofn(counter)=AAoutlin(z)+Dixtofn(counter);
      end
   end
end
if fn4i > 1
   counter=1;
   if fsx6i == 1
      fsx6i = fendi+1;
      counter = fai-fni-length(fn4i+1:fendi);
   end
   for z = fsx6i-1:-1:fn4i+1,
      counter = 1+counter;
      if AAoutdB(z) > ADreferencedB
         Dixtofn(counter)=AAoutlin(z)+Dixtofn(counter);
      end
   end
end
Di0toxdB = 20*log10(Di0tox);
DixtofndB = 20*log10(Dixtofn);
Dilin = [Di0tox Dixtofn];
DidB = 20*log10(Dilin);
freqdig = [freq0tox freqxtofn];
```

```
% Define the controller characteristics
numz = eval(get(q30,'String'));
denz = eval(get(q32,'String'));
%[C] = freqz(numz,denz,2*pi*freqdig);
[C,qp,qpqp] = dbode(numz,denz,1/fs,2*pi*freqdig);

% Add the controller into the loop now
Dolin = C'.*Dilin;
DodB = 20*log10(Dolin);

% Add the D/A to the loop
% First lets simply do the amplitude modulation half of the process
Dostar0tofn = Dolin*(fs);
freq0tofn = [freq0tox freqxtofn];
Dostarfntofs = Dolin(length(Dolin):-1:1)*fs;
freqfntofs = [fn freqfntoaliasf freqaliasftofs];
Dostarfsto3fn = Dostar0tofn;
freqfsto3fn = fs+[0:fint:fint*(length(freqfntofs)-1)];
Dostar3fnto2fs = Dostarfntofs;
freq3fnto2fs = 3*fn+[0:fint:fint*(length(freqfntofs)-1)];
Dostarlin = [Dostar0tofn Dostarfntofs Dostarfsto3fn Dostar3fnto2fs];
DostardB = 20*log10(Dostarlin);
freqDostar = [freq0tofn freqfntofs freqfsto3fn freq3fnto2fs];

% Now lets do the zoh half of the process
zoh = abs((1-exp(-i*(2*pi*freqDostar)*(1/fs)))./(i*(2*pi*freqDostar)));
zohlin = Dostarlin.*zoh;
zohdB = 20*log10(zohlin);

% The final piece of the puzzle is the smoothing filter.
% Design criteria  1)0dB pass band from 0 to xHz.
%                  2)output must be below floor of speaker by fn or
%                    below the bit level of the A/D board
%                    so that we don't output frequencies that are
%                    fn which appear only because of the D/A process.
if ADresolution > L
   L = ADresolution;
end
Msm = (-20*log10(L/zohlin(fni)))/(log10(x)-log10(fn));
Bsm = -Msm*log10(x);
if Msm > 0
   Msm = 0;
   Bsm = 0;
end
SMdB0tox = zeros(1,length(freq0tox));
freqsmxto2fs = freqDostar((fxi+1):length(freqDostar));
SMdBxto2fs = Msm.*log10(freqsmxto2fs)+Bsm;
SMdB = [SMdB0tox SMdBxto2fs];
VodB = SMdB+zohdB;
Volin = 10.^(VodB/20);
orig_Msm = Msm;

% Lets check to see that the higher frequencies don't cause us any trouble
% Maybe the first design criteria knocked down the amplitude at fn but
% if the spectrum is highpass the higher frequencies may not be knocked down
% below the floor of the speaker.  If they cause problems then slope must be
% increased and all the frequencies checked again with the new design
LrefdB = 20*log10(L);          %speaker floor in dB
zz = 10;
while zz > 1
   zz = 10;
   for z = fni : length(freqDostar),
```

```
      if VodB(z) > LrefdB
         %diff = (VodB(z)-LrefdB);
         %Msm = ((orig_Msm)+diff)/(log10(x)-log10(fn));
         diff  = (VodB(z)-LrefdB)/(log10(freqDostar(z))-log10(x));
         Msm = Msm - diff;
         Bsm = -Msm*log10(x);
         SMdBxto2fs = Msm.*log10(freqsmxto2fs)+Bsm;
         SMdB = [SMdB0tox SMdBxto2fs];
         VodB = SMdB+zohdB;
         Volin = 10.^(VodB/20);
         zz=100;
         z = fni;
      end
   end
   if zz < 50
      zz = 0;
   end
end



% Generate the plots specified by the user

if get(q15,'Value') == 1
   figure;
   semilogx(freq,VidB,'b')
   title('Input Voltage Spectrum')
   ylabel('Magnitude, dB')
   %axis tight;
   grid on;
end
if get(q15a,'Value') == 1
   figure;
   semilogx(freq,SPLdatadB,'b')
   title('Input SPL Spectrum')
   ylabel('SPL, dB')
   %axis tight;
   grid on;
end
if get(q16,'Value') == 1
   figure;
   semilogx(freq,AAdB,'b')
   title(['Anti-Aliasng Filter FRF      Slope = ',num2str(Maa)])
   ylabel('Magnitude, dB')
   %axis tight;
   grid on;
end
if get(q17,'Value') == 1
   figure;
   semilogx(freq,AAoutdB,'b',[aliasf  aliasf],[max(AAoutdB)  ADreferencedB],'--
r',[x x],...
   [max(AAoutdB)  ADreferencedB],'--r',[fn  fn],[max(AAoutdB)  ADreferencedB],'--
r',[x aliasf],...
   [ADreferencedB ADreferencedB],'--r')
   title('Output From Anti-Aliasng Filter')
   ylabel('Magnitude, dB')
   xlabel(['Frequency, Hz','               Range of Interest = ',num2str(x),'
Sampling Frequency = ',num2str(fs)])
   text(x,(ADreferencedB-15),'x');
   text(aliasf,(ADreferencedB-15),'(fs-x)')
   text(2*x,(ADreferencedB+10),'Res.')
   text(fn,(ADreferencedB-15),'fn')
   %axis tight;
```

```
   grid on;
end
if get(q18,'Value') == 1
   figure;
   semilogx(freq,AAoutdB,'g:',freqdig,DidB,'b',[x         aliasf],[ADreferencedB
ADreferencedB],'--r')
   title('Signal Before and After A/D Conversion')
   xlabel(['Frequency,  Hz','               Range  of  Interest  =  ',num2str(x),'
Sampling Frequency = ',num2str(fs)])
   ylabel('Magnitude, dB')
   legend('Analog','Digital','A/D res.')
   %%axis tight;
   grid on;
end
if get(q19,'Value') == 1
   figure;
   semilogx(freqdig,20*log10(abs(C)),'b')
   title('Control Filter FRF')
   ylabel('Magnitude, dB')
   %axis tight;
   grid on;
end
if get(q20,'Value') == 1
   figure;
   semilogx(freqdig,DodB,'b')
   title('Output From Controller')
   xlabel(['Frequency,  Hz','               Range  of  Interest  =  ',num2str(x),'
Sampling Frequency = ',num2str(fs)])
   ylabel('Magnitude, dB')
   %axis tight;
   grid on;
end
if get(q20a,'Value') == 1
   figure;
   semilogx(freqDostar,DostardB,'b')
   title('Amplitude Modulated Control Signal')
   ylabel('Magnitude, dB')
   %axis tight;
   grid on;
end
if get(q20b,'Value') == 1
   figure;
   semilogx(freqDostar,20*log10(zoh),'b')
   title('ZOH Filter FRF')
   ylabel('Magnitude')
   %axis([min(freqDostar) max(freqDostar) -200 10+max(20*log10(zoh))])
   %axis tight;
   grid on;
end
if get(q21,'Value') == 1
   figure;
   semilogx(freqDostar,zohdB,'b')
   title('D/A Output')
   ylabel('Magnitude')
   xlabel(['Frequency,  Hz','               Range  of  Interest  =  ',num2str(x),'
Sampling Frequency = ',num2str(fs)])
   %axis tight;
   grid on;
end
if get (q22,'Value') == 1
   figure;
   semilogx(freqDostar,SMdB,'b')
   title(['Smoothing Filter FRF      Slope = ',num2str(Msm)])
```

```
   ylabel('Magnitude, dB')
   %%axis tight;
   grid on;
end
if get(q23,'Value') == 1
   figure;
   semilogx(freqDostar,VodB,'b',[x 2*fs],[20*log10(L) 20*log10(L)],'--r')
   title('Output From Smoothing Filter')
   xlabel(['Frequency, Hz','                Range of Interest = ',num2str(x),'
Sampling Frequency = ',num2str(fs)])
   ylabel('Magnitude, dB')
   text(x,(20*log10(L)-10),'x');
   text(2*x,(20*log10(L)+10),'L')
   text(fn,(20*log10(L)-10),'fn')
   %axis([min(freqDostar) max(freqDostar) -200 10+max(VodB)])
   %axis tight;
   grid on;
end

% feed the slopes back to the gui
set(qaa,'String',num2str(Maa));
set(qsm,'String',num2str(Msm));



% Calculate the phase loss in the system.  This is done on a per frequency
basis from 0Hz
% up to fn.  That is because everything above fn in the output is a result of
the D/A process
% and really has no correlation to the input spectrum in terms of phase
%ADlag = .000032;
%DAlag = .000006;
%pzoh = -(freqdig*1/2/fs)*180/pi;
%pAD = -(ADlag*freqdig*360);
%pDA = -(DAlag*freqdig*360);
%[msm,psm,wsm]=bode(numsm,densm,freq*2*pi);
%[maa,paa,waa]=bode(numaa,denaa,freq*2*pi);
%[mc,pc,fc]=dbode(numcondig,dencondig,dt,freq*2*pi);
%[totalphase]=paa+pc+psm+pzoh'+pAD'+pDA';
```

B.4 Designing a circuit to create a desired transfer function

BUILDTF.M

```
% buildtf.m
% This file constructs a circuit that produces a desired transfer % function.
The transfer function must be a set of complex
% conjugate zeros over a set of complex conjugate poles.
% The circuit used is the four op-amp biquad circuit.

clear all;,  close all;
zetaz = input('Damping Ratio of Zero -> ');
zetap = input('Damping Ratio of Pole -> ');
wnz = input('Natural Frequency of Zero (rad/s)-> ');
wnp = input('Natural Frequency of Pole (rad/s)-> ');
C1=1e-6;
C2=1e-6;

if wnz > wnp
```

```
      if (zetaz*wnz)/(zetap*wnp) > 1
            case = 1;
            R3 = 1/2/C1/(zetaz*wnz-zetap*wnp);
            R1 = 1/2/zetap/wnp/C1;
            R2 = R3*((wnz/wnp)^2-1);
            R4 = 1/C2/C1/R2/(wnp^2);
            num= [1 (1/R1/C1)+(1/R3/C1) (R3+R2)/R2/R3/R4/C1/C2];
            den= [1 1/R1/C1 1/R2/R4/C1/C2];
      else
            case = 2;
            R3 = 1/2/C1/(zetap*wnp-zetaz*wnz);
            R1 = 1/2/zetap/wnp/C1;
            R2 = R3*((wnz/wnp)^2-1);
            R4 = 1/C2/C1/R2/(wnp^2);
            num= [1 (1/R1/C1)-(1/R3/C1) (R3+R2)/R2/R3/R4/C1/C2];
            den= [1 1/R1/C1 1/R2/R4/C1/C2];
      end
else
      if (zetap*wnp)/(zetaz*wnz) > 1
            case = 3;
            R3 = 1/2/C1/(zetap*wnp-zetaz*wnz);
            R1 = 1/2/zetap/wnp/C1;
            R2 = R3*(1-(wnz/wnp)^2);
            R4 = 1/C2/C1/R2/(wnp^2);
            num= [1 (1/R1/C1)-(1/R3/C1) (R3-R2)/R2/R3/R4/C1/C2];
            den= [1 1/R1/C1 1/R2/R4/C1/C2];
      else
            case = 4;
            R3 = 1/2/C1/(zetaz*wnz-zetap*wnp);
            R1 = 1/2/zetap/wnp/C1;
            R2 = R3*(1-(wnz/wnp)^2);
            R4 = 1/C2/C1/R2/(wnp^2);
            num= [1 (1/R1/C1)+(1/R3/C1) (R3-R2)/R2/R3/R4/C1/C2];
            den= [1 1/R1/C1 1/R2/R4/C1/C2];
      end
end

'R1 R2 R3 R4 C1 C2 -> all resistances in Ohms and capacitances in uF'
[R1 R2 R3 R4 C1/1e-6 C2/1e-6]'

% Rebuild the TF from the resistances

[mag,phase,freq]=bode(-num,den);
subplot(211),semilogx(freq/2/pi,20*log10(mag));
grid on;,   title('Bode Plot')
ylabel('Magnitude, dB')
subplot(212),semilogx(freq/2/pi,phase);
grid on;,    ylabel('Phase, degrees')
xlabel('Frequency, Hz')
'case'
```

B.5 Designing a compensator by shaping the closed-loop

BUILDTF.M

```
%comp.m
%this takes a plant and a compensator and plots the frequency
%response characteristics of the closed-loop system
```

```
directory = get(file2,'String');
filename = get(file4,'String');
load([directory filename '.txt']);
frfdata = eval(filename);

freq = frfdata(:,1);
plantreal = frfdata(:,2);
plantimag = frfdata(:,3);

plantmag = 20*log10(sqrt(plantreal.^2+plantimag.^2));
plantH = plantreal + i*plantimag;
plantphase = unwrap(angle(plantH))*180/pi;

line180 = [180 180];
line360 = [360 360];
line540 = [540 540];

figure;
subplot(211),semilogx(freq,plantmag,[min(freq)+1 max(freq)],[0 0],'--r');
title('Plant FRF');
ylabel('Magnitude, dB');
axis([min(freq) max(freq) min(plantmag) max(plantmag)]);
grid on;
subplot(212),semilogx(freq,plantphase...
                      ,[min(freq)+1 max(freq)],line180,'--r'...
                      ,[min(freq)+1 max(freq)],-line180,'--r'...
                      ,[min(freq)+1 max(freq)],line360,'--r'...
                      ,[min(freq)+1 max(freq)],-line360,'--r'...
                      ,[min(freq)+1 max(freq)],line540,'--r'...
                      ,[min(freq)+1 max(freq)],-line540','--r');
axis([min(freq) max(freq) -540 540]);
ylabel('Phase, Degrees');
xlabel('Frequency, Hz');
grid on;

%-------------------------------------------------------------------------
% Define the compensator
w1p = eval(get(poles3a,'String'));, w1z = eval(get(zeros3a,'String'));
z1p = eval(get(poles3b,'String'));, z1z = eval(get(zeros3b,'String'));
w2p = eval(get(poles4a,'String'));, w2z = eval(get(zeros4a,'String'));
z2p = eval(get(poles4b,'String'));, z2z = eval(get(zeros4b,'String'));
w3p = eval(get(poles5a,'String'));, w3z = eval(get(zeros5a,'String'));
z3p = eval(get(poles5b,'String'));, z3z = eval(get(zeros5b,'String'));
w4p = eval(get(poles6a,'String'));, w4z = eval(get(zeros6a,'String'));
z4p = eval(get(poles6b,'String'));, z4z = eval(get(zeros6b,'String'));
w5p = eval(get(poles7a,'String'));, w5z = eval(get(zeros7a,'String'));
z5p = eval(get(poles7b,'String'));, z5z = eval(get(zeros7b,'String'));

f1p = eval(get(poles9a,'String'));, f1z = eval(get(zeros9a,'String'));
f2p = eval(get(poles10a,'String'));, f2z = eval(get(zeros10a,'String'));
f3p = eval(get(poles11a,'String'));, f3z = eval(get(zeros11a,'String'));
f4p = eval(get(poles12a,'String'));, f4z = eval(get(zeros12a,'String'));
f5p = eval(get(poles13a,'String'));, f5z = eval(get(zeros13a,'String'));

gain = eval(get(gainb,'String'));

comppoles = [];
compzeros = [];
if (w1p >= 0), comppoles = [placep(w1p,z1p)]; end
if (w2p >= 0), comppoles = [comppoles placep(w2p,z2p)]; end
if (w3p >= 0), comppoles = [comppoles placep(w3p,z3p)]; end
if (w4p >= 0), comppoles = [comppoles placep(w4p,z4p)]; end
```

```
if (w5p >= 0), comppoles = [comppoles placep(w5p,z5p)]; end
if (w1z >= 0), compzeros = [placep(w1z,z1z)]; end
if (w2z >= 0), compzeros = [compzeros placep(w2z,z2z)]; end
if (w3z >= 0), compzeros = [compzeros placep(w3z,z3z)]; end
if (w4z >= 0), compzeros = [compzeros placep(w4z,z4z)]; end
if (w5z >= 0), compzeros = [compzeros placep(w5z,z5z)]; end
realpoles = [];
realzeros = [];
if (f1p >= 0), realpoles = [f1p]; end
if (f2p >= 0), realpoles = [realpoles f2p]; end
if (f3p >= 0), realpoles = [realpoles f3p]; end
if (f4p >= 0), realpoles = [realpoles f4p]; end
if (f5p >= 0), realpoles = [realpoles f5p]; end
if (f1z >= 0), realzeros = [f1z]; end
if (f2z >= 0), realzeros = [realzeros f2z]; end
if (f3z >= 0), realzeros = [realzeros f3z]; end
if (f4z >= 0), realzeros = [realzeros f4z]; end
if (f5z >= 0), realzeros = [realzeros f5z]; end

zeroes = [realzeros compzeros]';
poles = [realpoles comppoles]';
[compnum,compden] = zp2tf(zeroes,poles,gain);
[compmag,compphase,compw] = bode(compnum,compden,freq*2*pi);
compmag = 20*log10(compmag);

figure;
subplot(211),semilogx(freq,compmag);
ylabel('Magnitude, dB');
title('Compensator FRF');
axis([min(freq) max(freq) min(compmag) max(compmag)+1]);
grid on;
subplot(212),semilogx(freq,compphase);
ylabel('Phase, Degrees');
xlabel('Frequency, Hz');
axis([min(freq) max(freq) min(compphase) max(compphase)+1]);
grid on;

%-------------------------------------------------------------------------
%Determine the open-loop system

oloop_mag = compmag+plantmag;
oloop_phase = plantphase + compphase;

figure;
subplot(211),semilogx(freq,oloop_mag);
ylabel('Magnitude, dB');
title('Open-loop FRF');
axis([min(freq) max(freq) min(oloop_mag) max(oloop_mag)]);
grid on;
subplot(212),semilogx(freq,oloop_phase...
                      ,[min(freq)+1 max(freq)],line180,'--r'...
                      ,[min(freq)+1 max(freq)],-line180,'--r'...
                      ,[min(freq)+1 max(freq)],line360,'--r'...
                      ,[min(freq)+1 max(freq)],-line360,'--r'...
                      ,[min(freq)+1 max(freq)],line540,'--r'...
                      ,[min(freq)+1 max(freq)],-line540','--r');
ylabel('Phase, Degrees');
xlabel('Frequency, Hz');
axis([min(freq) max(freq) min(oloop_phase) max(oloop_phase)+1]);
grid on;

%-------------------------------------------------------------------------
%Close the loop on the system
```

```
fb = get(feedback1a,'Value');
cloop_real = oloop_mag.*cos(oloop_phase*(pi/180));
cloop_imag = oloop_mag.*sin(oloop_phase*(pi/180));
cloop_complex =  cloop_real + cloop_imag*i;

if fb == 1
   cloop_mag = 20*log10(abs(1./(1 + cloop_complex.*cloop_complex)));
   figure;
   semilogx(freq,cloop_mag)
   title('Positive Feedback Close Loop Performance');
   xlabel('Frequency, Hz')
   ylabel('Magnitude');
   grid on;

end

if fb == 0
   cloop_mag = 20*log10(abs(1./(1 - cloop_complex.*cloop_complex)));
   figure;
   semilogx(freq,cloop_mag)
   title('Negative Feedback Close Loop Performance');
   xlabel('Frequency, Hz')
   ylabel('Magnitude');
   grid on;
end
```

REFERENCES

Allie, M.C., L.J. Eriksson and C.D. Bremigan. 1988. "Active Noise Control Using Adaptive Digital Signal Processing," *IEEE*. CH2561, pp. 2594-2597, 2598-2601

Beranek, Leo Leroy. 1954. *Acoustcis.* New York, NY: McGraw Hill.

Bies, David and Colin Hansen. 1996. *Engineering Noise Control.* Second Edition. New York: E and FN Spon.

Binks, L.A., D.J. Henwood and M.A. Jones. 1991. "Finite Element Methods Applied to the Analysis of High-Fidelity Loudspeaker Transducers," *Computers and Structures.* Vol. 44, No. 4, pp. 765-772

Boff, Kenneth, Lloyd Kaufman and James Thomas. 1986. *Handbook of Perception and Human Performance. Volume 1, Sensory Processes and Perception*. New York: John Wiley and Sons.

Bose, Amar G. and John Carter. "Headphoning," U.S. Patent No. 4,455,675 (1984 June 19)

Casali, John G. and Gary S. Robinson. 1994. "Narrow-Band Digital Active Noise Reduction in a Siren-Cancelling Headset: Real-Ear and Acoustical Manikin Insertion Loss," *Noise Control Engineering Journal*. Vol. 42, No. 3, May-June, pp. 101-115

Clark, R. L. and Steven Lane. 1998. "Improving Loudspeaker Performance for Active Noise Control Applications."

Coppens, Alan, Austin Frey, Lawrence Kinsler and James Sanders. 1982. *Fundamentals of Acoustics.* Third Edition. New York: John Wiley and Sons.

Costin, Mark H. and Donald R. Elzinga. 1989. "Active Reduction of Low-Frequency Tire Impact Noise Using Digital Feedback Control," *IEEE Control Systems Magazine*. August, pp. 3-6

Dorey, A.P., S.F. Pelc, R.D. Rawlinson and P.D. Wheeler. 1978. "The Development and Testing of an Active Noise Reduction System for use in Ear Defenders," *Proc. Inter-Noise 78.* pp. 977-982

Elliott, S.J. and P.A. Nelson. 1992 *Active Control of Sound*. London, England. Academic Press LTD.

Frankort, F.J.M. 1978. "Vibration Patterns and Radiation Behavior of Loudspeaker Cones," *J. Audio Eng. Soc.* Vol. 26, No. 9, September, pp. 609-622

Franklin, Powell and Workman. 1990. *Digital Control of Dynamic Systems*. Menlo Park, California: Addison-Wesley Publishing Company.

Geddes, Earl, James Porter and Yifan Tang. 1987. "A Boundary-Element Approach to Finite Element Radiation Problems," *J. Audio Eng. Soc.* Vol. 35, No. 4, April, pp. 211-229

Gogate, Sachin and Clark Radcliffe. 1992. "Identification and Modeling of Speaker Dynamics for Acoustic Control Applications," *Active Control of Noise and Vibration ASME*. Vol. 38, pp. 295 -300

Goodfellow, Elizabeth. 1994. "A Prototype Active Noise Reduction In-Ear Hearing Protector," *Applied Acoustics*. Vol. 42. pp. 299-312

Kaiser, Arie and Ad Leeuwestein. 1988. "Calculation of the Sound Radiation of a Nonrigid Loudspeaker Diaphragm Using the Finite-Element Method," *J. Audio Eng. Soc.* Vol. 36, No. 7/8, July/August, pp. 539-551

Leitch, R.R. and M.O. Tokhi. 1991. "Noise Control Systems," *IEEE Proceedings -D*. Vol. 138, No. 5, September, pp. 421-430

Leug, Paul. "Process of Silencing Sound Oscillations," U.S. Patent No. 2,043,416 (1936 June 09).

Mazzola, Claude. 1993. *Active Sound Absorption*. Mamaroneck, NY: Namlak.

Nomoto, Isami and Kiyoaki Suzuki. 1982. "Computerized Analysis and Observation of the Vibration Modes of a Loudspeaker Cone," *J. Audio Eng. Soc.* Vol. 30, No. 3, March, pp. 98-106

Pierce, Allan D. 1989 *Acoustics*. Woodbury, New York: Acoustical Society of America.

Rayleigh, John William. 1929. *The Theory of Sound*. New York, NY: MacMillan.

Struck, Christopher J. 1990. "Investigation of the Nonrigid Behavior of a Loudspeaker Diaphragm Using Modal Analysis," *J. Audio Eng. Soc.* Vol. 38, No. 9, September, pp. 667-674

Van Valkenburg, M. E. 1982. *Analog Filter Design.* New York, NY: Holt Reinhart and Winston.

VITA

Daniel Christopher Clatterbuck was born September 9, 1973 in Boonton, New Jersey and spent his formative years in Randolph enjoying school and participating in athletics. In February of 1991 he found his soul mate, Gretchen Ponzi. Little did the two know, but they would spend the next 7 years sharing their hopes and dreams from across state borders. In June of 1992, they graduated from Randolph High School. Dan's college life brought him to Blacksburg where he began his study of Mechanical Engineering at Virginia Tech, while Gretchen chose to attend the University of Delaware 6 hours north of Virginia Tech. As an undergraduate, Dan enjoyed his friends, schoolwork, and athletics. Playing on the Men's Varsity Soccer Team provided him with a unique experience that would shape his character forever. The teamwork and dedication it took to play on the field is something that permeates his off the field personality. Summers were spent at home catching up on lost time with Gretchen and working as an intern for The John Dusenbery Company. In May of 1996, Dan received his Bachelor of Science, Cum Laude, and Gretchen graduated with her Bachelor of Arts in Math Education. The separation, however, was not over. Upon graduation, Dan enrolled in the Master's program at Virginia Tech, while Gretchen found a teaching position at Haverford High School in Pennsylvania. Under the tutelage of Will Saunders, Dan studied acoustics and control theory. He also was a GTA for the department for his two years as a graduate student, and this proved to be a learning experience for the students as well as Dan. Upon completion of his degree in May of 1998, Dan and Gretchen plan to be married and relocate to Michigan where Dan will begin work at Ford Motor Company in the Global Test Operations division. The separation will finally end when the two join hands June 27, 1998.