USING HELICOPTER NOISE TO PREVENT BROWNOUT CRASHES: AN ACOUSTIC ALTIMETER

A Thesis
Presented to
The Academic Faculty

by

Joseph S. Freedman

In Partial Fulfillment
of the Requirements for the Degree of
Masters of Science in Mechanical Engineering in the
Woodruff School of Mechanical Engineering

Georgia Institute of Technology

August 2010

USING HELICOPTER NOISE TO PREVENT BROWNOUT CRASHES: AN ACOUSTIC ALTIMETER

Approved by:

Dr. Peter Rogers, Advisor
Mechanical Engineering
Georgia Institute of Technology

Dr. Ken Cunefare
Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Mr. Michael Gray
Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: June 4, 2010

ACKNOWLEDGEMENTS

While it is an impossible task for me to thank everyone who became involved with this thesis over its entire course, it is still necessary to try. None of this work would have been possible without the guidance and support from my advisor, Peter Rogers. His knowledge and foresight were invaluable.

I am also immensely grateful for help that I have received from other faculty, staff, and students during my tenure at Tech. The atmosphere of cooperation fostered by the Woodruff School has let me do so much more with this work that I would have been able to do alone. On this note, I owe a big thanks to Ken Cunefare, who provided me with so much of the assistance and equipment that was needed for these experiments. Mike Gray has been able to talk out many of the technical and organizational sticking points with me. Also, Jim Martin was able to point me in the right direction when I was completely off base on more occasions than I would like to remember.

When building of the microphone array and its mount, I used services of the Machine Shop and Electronics Shop. The talent (and patience) of the people whom I worked with – namely John Graham, Kyle French, and Kevin Johnson – was wonderful. Also, this project would be nowhere near what it is today without the help of Tim Hsu, Peter Cameron, Jamie Wilson, and many others.

Last but certainly not least, I need to acknowledge how supportive my family has been through my entire time at Georgia Tech. The loving encouragement of my father, mother, two sisters and brother has meant the world to me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

α	Attenuation Coefficient
a	Particle radius
a_0	Doppler Distortion Constant
BW	Bandwidth (frequently normalized to 1)
c	Speed of sound
c_{pf}	Specific heat capacity of the fluid
c_{pp}	Specific heat capacity of the particle
cm	Centimeters
δ	Dirac delta impulse
dB	Decibel
η	Mass loading
$\mathcal{F}, \mathcal{F}^{-1}$	Fourier Transform, Inverse Fourier Transform
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
γ	Specific heat ratio
h	Helicopter height
Hz	Hertz (1/sec)
i	$\sqrt{(-1)}$
k	Sample index in discrete frequency domain
κ_f	Thermal Conductivity of the fluid
ln	Natural logarithm
m	Meters
n	Sample number in discrete time domain

N	Total number of samples
P_0	Spectrum of rotor noise signal
ρ_f	Density of the fluid
ρ_p	Density of the particulate material
r	Radial coordinate (in cylindrical coordinate system)
S/N	Signal to Noise Ratio
$s(t)$	Noise signal created by helicopter
$S(\omega)$	Frequency Spectrum of noise signal
$S(\Omega)$	Scale Spectrum of noise signal
t	Continuous time variable
τ_d	Dynamic (Translational) Relaxation time
τ_t	Thermal Relaxation time
τ	Dummy time variable
θ	Angle (in radians)
ν_f	Viscosity of the fluid
v_x	Wind velocity in the x direction
ω	Continuous frequency variable
Ω	Mellin-scale Transform independent variable
$x(t)$	Random broadband signal
$x[n]$	Discretized random broadband signal
$X(\omega)$	Fourier Transform of x(t)
$X[k]$	Fourier Transform of x[n]
$\hat{x}(t)$	Cepstrum of x(t)

SUMMARY

Brownout is a phenomenon that occurs in aviation when high dust and sand levels are stirred up by the rotor downwash of landing helicopters. The sand/dust levels reduce the pilot's visibility to nearly zero, which can lead to crashes, injuries, and fatalities. This thesis explores one possible method of preventing brownout crashes using the noise generated by the helicopter rotor as an altimeter.

The hypothesis explored here is that the helicopter's height and velocity, and also obstacle locations with respect to the helicopter, can be determined by monitoring the differences between rotor noise incident and reflected to the helicopter, provided adequate bandwidth and signal to noise ratio. Heights can be determined by measuring the time required for the noise from the rotor to reflect from the ground. Distances to smaller obstacles, such as vehicles and people, can be sensed by different interference patterns in the acoustic signal. The velocity can be determined by measuring small amounts of Doppler distortion.

To detect the height, a signal processing technique known as the cepstrum is employed. Then to detect the velocity, an algorithm is developed which measures the amount of Doppler distortion. A transform called the Mellin-Scale Transform is at the core of the algorithm. Algorithms associated with these two methods are tested for robustness and accuracy. Once the algorithms are fully developed and optimized for this application, a microphone array is built to confirm the simulations. The array is tested in a hemianechoic chamber and outside in Georgia Tech's Burger Bowl. Height and obstacle detection are determined to be feasible with the existing array. However, velocity detection and surface mapping are not successfully accomplished.

CHAPTER 1

MOTIVATION FOR STUDY

DARPA has defined brownout as "intense, blinding dust clouds stirred up by the aircraft's main-rotor down-wash during near ground flight [which can] cause helicopter pilots to suddenly lose all visual cues (Fuller, Tavakkoli et al. 1988)." This problem has cost the US military more than $100 million each year in helicopter damages between 2000 and 2005 (Sabbagh 2006). It has been estimated that as many as three out of every four helicopter accidents in the current conflicts in Iraq and Afghanistan are caused by brownout (Brower 2004). This rise to a prominent cause of helicopter accidents is rather recent, and probably results from the synergy of geography, climate, and environment of areas where the US military is now engaged.

The thick dust of brownouts presents numerous problems. In addition to the pilot's loss of all visual cues during brownout, the dust that has been the cause of recent brownout events has been so thick that the army's modern altimeter systems are rendered useless. Further, radar signals and radio signals to GPS do not reliably penetrate these dust clouds (Sabbagh 2006).

Current methods for overcoming brownout disorientation are crude and not always effective. The most common method is to treat the soil surrounding the landing area before the helicopter lands: for instance, a substance called Envirotac II™ can be sprayed before landing to stabilize the soil. This requires spraying toxic chemicals over a wide area on a regular basis. Another solution is to cover the soil before the helicopter lands. Mobi-Mat™ does this using a large polyester mesh that can be spread out across a landing area; the mat acts as a rug to hold down loose particles. These solutions require

advanced knowledge of the landing site, are invasive to the environment (Brower 2004), and are bad for special operations that require speedy and untraceable deployment.

There is a need for a passive method for brownout crash prevention. This thesis addresses this need by proposing and evaluating a prototype of an acoustic altimeter that does not add any chemicals, materials, or signals to the environment. This altimeter uses the signal that is already present, coming from the aircraft's rotor. This altimeter will ultimately be able to determine both the height and velocity of the helicopter, as well as the locations of any obstacles in the area.

CHAPTER 2

GENERAL APPROACH

The objective of this thesis is to design and build a prototype of an acoustic altimeter that can determine the helicopter's height and speed during landing. The altimeter would use the noise created by the helicopter rotor as a type of sonar signal to locate the ground and also warn of any obstacles in the vicinity. The use of the helicopter's own noise as a probe signal has many potential advantages: (1) it is very loud; (2) has frequency content across the entire spectrum; (3) it is already there—nothing would have to be added to the system.

It is well known that a plane wave train reflecting off a flat, smooth, rigid surface will set up a pattern of sound that is periodic in the frequency domain (Pierce 1994). If the plane wave is normal to a rigid surface, then a spatial periodicity is created by the interference between the incident and reflected waves. The basis for the spatial periodicity is given for the time and frequency domains as:

Time Domain: $$x(t) = \left(\delta(t) + \delta\left(t - \frac{2h}{c}\right)\right) * s(t) \quad (1)$$

Frequency Domain: $$X(\omega) = \left(2 + 2\cos\left(\frac{2\omega h}{c}\right)\right) S(\omega) \quad (2)$$

In these equations, $s(t)$ and $S(\omega)$ represent the rotor noise signal without interference by the ground in the time and frequency domains, respectively. The quantities $x(t)$ and $X(\omega)$ represent the observed time and frequency signals that result from the helicopter noise reflecting off the ground; δ(t) represents an infinite impulse at time t; h represents the height of the helicopter; the symbol * indicates convolution; and c represents the speed of sound. The second δ impulse is delayed by an amount of $2h/c$, the length of time that it takes the sound wave to travel to the ground and back. This

relationship shows that the sum of the reflected and incident signals depends (in part) on the height of the source. Since the height is in the argument of a cosine function, it sets up the spatial periodicity. The method developed in this study will rely on that dependence on height to determine the height of the source of the helicopter. Equation 1 can be manipulated to change the height into a more easily observed quantity.

In the derivation of velocity and height detection algorithms, $\delta(t)$ is treated as an infinite impulse in the continuous domain and $\delta[n]$ as a unit impulse in the discrete domain. This allows an analytical form to be reached and interpreted more clearly.

The manipulation that will be used in this work is the cepstrum technique, an algorithm that can detect ground interference by means of the spectrum of the logarithm of the spectrum. This technique will allow the height to be determined using the location of peaks in the cepstrum domain.

Once the details have been fully derived and explained, the techniques under discussion are simulated under a variety of circumstances, with increasing complexity. These simulations lead into experimental work in which a microphone array is simulated, designed, and built. With the array built, the first objective is to test its abilities to measure height effectively. The next level of complexity is to use the array to detect obstacles in the landing area. Each obstacle will reflect sound, and it is hypothesized that these reflections will be detectable in the cepstrum domain.

After the height and obstacles have been detected, the array's ability to determine velocity will be measured. Two methods for velocity measurement will be tested. The first involves calculating the height at different times as the helicopter descends. The difference in height of multiple measurements can be divided by the difference in time

between the measurements to calculate the height. The second method of velocity calculation is more involved, but only requires a single measurement. The second method takes advantage of the small amount of Doppler distortion in the reflected signal that is not present in the incident signal. A transformation known as the Mellin-Scale transform will be used to recast the Doppler distortion as phase information. Once the Doppler distortion is converted into phase information, the amount of Doppler distortion is determined using Fourier Transforms. The velocity can be deduced from the amount of Doppler distortion.

Figure 1 is a cartoon diagram showing how the proposed altimeter would measure height and velocity. A two-sided microphone array is mounted beneath the helicopter to separate the downward- and upward-moving sound. Each side of the microphone array will consist of unidirectional electret condenser microphones selected for their durability, cost, and directivity. Figure 1 shows where the spatial periodicity might be observed in the case of the hovering helicopter, and it illustrates which part of the signal is the incident ($P_{incident}$) and which is reflected ($P_{reflected}$).

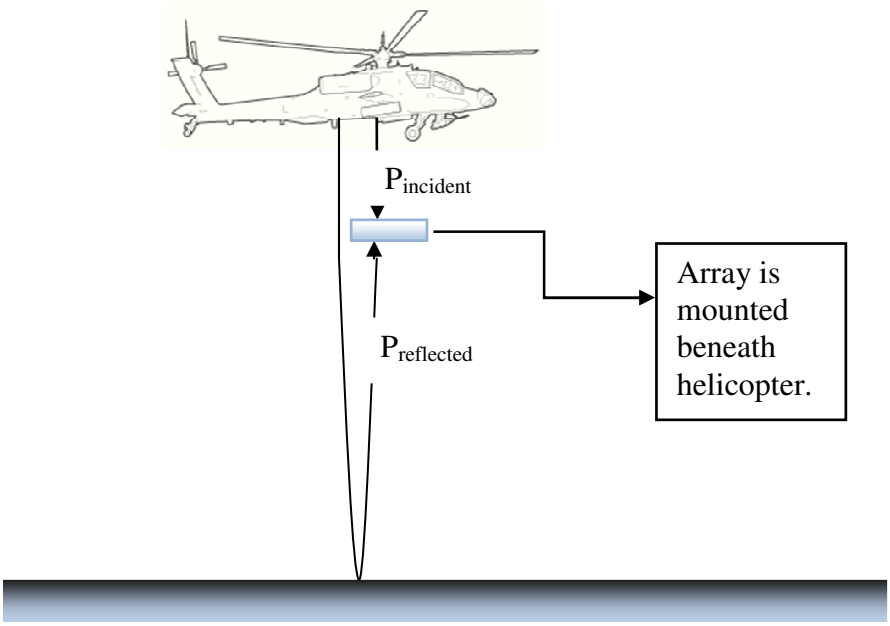

Figure 1 – Cartoon diagram illustrating incident and reflected paths

Figure 2 shows the output of a microphone array that would be placed as shown in Figure 1. The incident and reflected pressure signals will combine to form a periodic pattern in the frequency domain. Figure 2 shows three plots: the first is the incident spectrum, the second is the reflected spectrum, and the final is the sum of the two. The periodic nature of the total signal is clearly visible in the right plot. The array will be tested first while suspended above the floor in a hemi-anechoic chamber, and second while suspended outdoors in an open field.

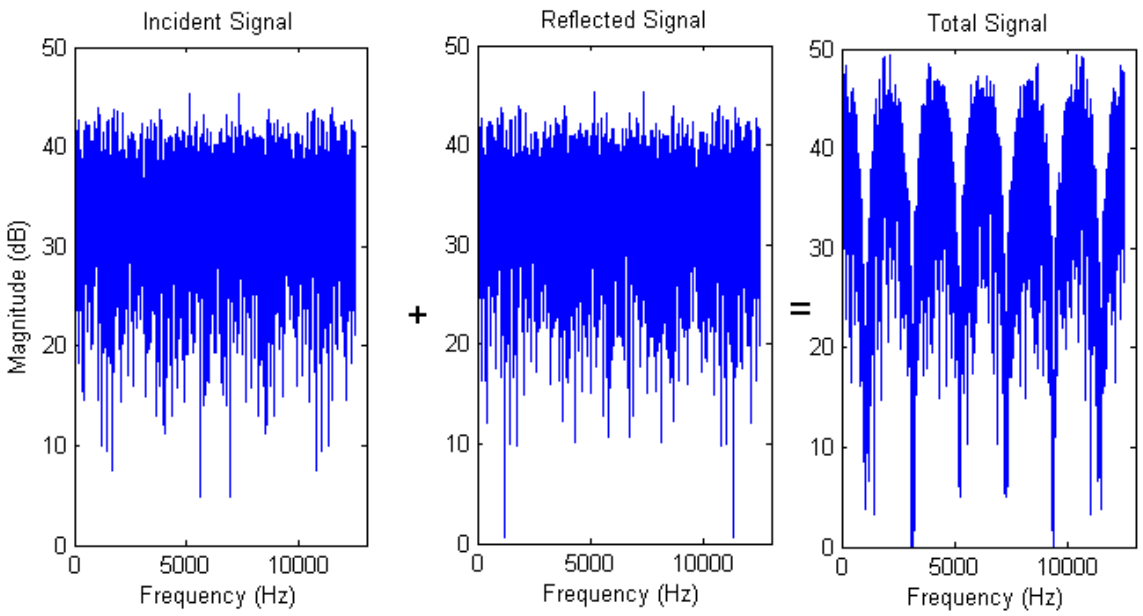

Figure 2 – Result of adding the incident signal to the reflected signal

The following two sections will introduce background on brownout and the proposed theory for determining the helicopter's height and velocity. Then, a means of testing this theory experimentally will be presented.

2.1. Overview of Brownout

Brownout is caused by the helicopter's (or other rotorcraft's) downwash blowing around loose and dry sand in a toroidal pattern when the rotorcraft is close to the ground.

Figure 3 shows an Osprey at the onset of brownout (Harrington 2007). The blades of the helicopter push air down and then outward as it moves to the ground, blowing around the loose particles on the surface. Once the air has moved outward, it circulates back through the blades creating the toroidal flow and obscuring the line of sight.

Figure 3 - Osprey in brownout

The primary physical conditions that dictate the occurrence and severity of brownout are the soil composition and the helicopter's landing angle, rotor size, and approach velocity. Loose soil, where a wide variety of particle sizes is present, significantly increases the likelihood of brownout (Ryerson, Haehnel et al. 2005). These conditions are commonplace in both Iraq and Afghanistan.

2.2. Height Detection Background

The foremost concern of the helicopter pilot is the helicopter's height above the ground. Using the cepstrum, the helicopter's height can be determined by monitoring periodic behavior in the interference patterns made by the rotor noise and its reflection as the helicopter approaches the ground.

2.2.1. Introducing the Cepstrum

The cepstrum is an algorithm in digital signal processing first developed in the 1970s to help with dereverberation, speech recognition, and echo arrival time calculations

(Childers, Skinner et al. 1977). To explain the cepstrum algorithm, a basic overview with a flow chart will be introduced and the mathematical details will follow.

The fundamental idea of this algorithm is to find the inverse of the spectrum of the log of the spectrum (hence the anagram *cepstrum*). The flow chart in Figure 4 shows the steps of the algorithm. The following is a more formal explanation of the algorithm.

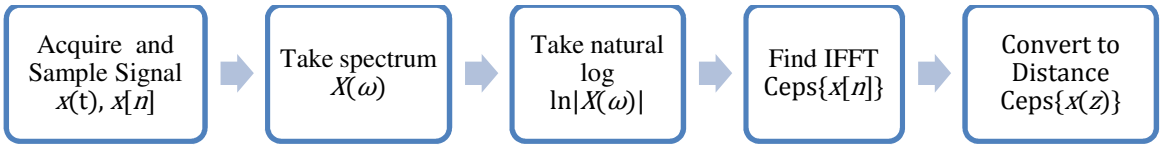

Figure 4 - Summary of Cepstrum Algorithm

The motives for each step are omitted until the algorithm is applied to the special case of a signal with its echo.

Suppose $x(\mathrm{t})$ is a band-limited continuous time signal. After being sampled, the resulting signal is $x[n]$. Using the Fast Fourier Transform (FFT), $x[n]$ can be converted to the frequency domain and becomes $X[k]$. This transformation from the time domain to the frequency domain is

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-i\frac{2\pi}{N}kn} \tag{3}$$

Here, k represents the frequency value and N the sample length.

Then, Equation (4) calculates the cepstrum by finding the Inverse Fast Fourier Transform (IFFT) of the natural logarithm of the absolute value of $X[k]$ in Equation (2). The "hat" above the x indicates that the variable is now in the cepstrum domain. The cepstrum is thus

$$\hat{x} = \frac{1}{N} \sum_{k=0}^{N-1} \ln(|X[k]|^2) e^{-i\frac{2\pi}{N}kn} \tag{4}$$

The hat notation will be used throughout this thesis to indicate when a variable is in the cepstrum domain.

To facilitate the description of signals once they are inside the cepstrum domain, there is an entire set of anagrams that are based on their frequency domain counterpart. As mentioned previously, cepstrum is an anagram of spectrum. The cepstrum's independent variable is called the quefrency (anagram of frequency) and dependent variable is called the gamnitude (anagram of magnitude).

The primary application for the cepstrum is to signals having periodic nature in the frequency domain. In the case of a descending helicopter, the frequency periodicity is formed by rotor noise as the incident rotor noise signal interferes with the reflected rotor noise signal. The cepstrum can be used to calculate the helicopter's height with this interference pattern, as detailed in the next section.

2.2.2. Application of the Cepstrum to Height Detection

Now that the cepstrum algorithm has been explained in a more general context, it can be applied to the specific case of a helicopter hovering above a flat rigid surface. Suppose a hovering helicopter sends down a noise signal $s(\mathrm{t})$ that is reflected by a rigid boundary with no propagation losses. The resulting signal at the array is

$$x(\mathrm{t}) = \left(\delta(\mathrm{t}) - \delta\left(\mathrm{t} - \frac{2h}{c}\right)\right) * s(t) = s(\mathrm{t}) + s\left(\mathrm{t} - \frac{2h}{c}\right) \tag{5}$$

Equation (5) is then converted into the frequency domain using a Fourier Transform. The result of this transformation is

$$X(\omega) = S(\omega) + e^{-\mathrm{i}\frac{2h}{c}\omega} S(\omega) \tag{6}$$

Capital letters such as X and S indicate that the signal is in the frequency domain rather than the time domain. Equation (5) is important because it demonstrates that the observed signal X is a function of the helicopter height.

Following the algorithm for the cepstrum, the natural log of the magnitude squared is now carried out in Equation (7), (8), and (9). This becomes:

$$\ln|\mathrm{X}|^2 = \ln\left|S(\omega) + e^{-\mathrm{i}\frac{2h}{c}\omega}S(\omega)\right|^2 \tag{7}$$

$$= \ln\left(|S(\omega)|^2\left(2 + 2\cos\left(\frac{2h}{c}\omega\right)\right)\right) \tag{8}$$

$$= \ln|S(\omega)|^2 + \ln\left(1 + \cos\left(\frac{2h}{c}\omega\right)\right) + \ln 2 \tag{9}$$

Using the properties of the logarithm, the observed signal has been converted into three distinct terms in Equation (9). The first term, $\ln|S(\omega)|^2$, describes the dereverberated signal. The second term, $\ln\left(1 + \cos\left(\frac{2h}{c}\omega\right)\right)$, describes the height of the helicopter and its echo, and is independent of the signal's content. The final term, $\ln 2$, is a constant.

Proceeding with the last stage of the algorithm, the inverse Fourier Transform ($\mathcal{F}^{-1}$) of Equation (9) is calculated. Using the linear property of the inverse Fourier Transform, it operates on each term separately

$$\hat{x}(\mathrm{t}) = \mathcal{F}^{-1}(\ln|S(\omega)|^2) + \mathcal{F}^{-1}\left(\ln\left(1 + \cos\left(\frac{2h}{c}\omega\right)\right)\right) + \mathcal{F}^{-1}\big(\ln(2)\big) \tag{10}$$

To make the transformation less unwieldy, Equation (10) is broken into two pieces, $\hat{x}_1(\mathrm{t})$ and $\hat{x}_2(\mathrm{t})$:

$$\hat{x}(\mathrm{t}) = \hat{x}_1(\mathrm{t}) + \hat{x}_2(\mathrm{t}) \tag{11}$$

$$\hat{x}_1(\mathrm{t}) = \mathcal{F}^{-1}\left(\ln\left(1+\cos\left(\frac{2h}{c}\omega\right)\right)\right) \tag{12}$$

$$\hat{x}_2(\mathrm{t}) = \mathcal{F}^{-1}(\ln|S(\omega)|^2) + \mathcal{F}^{-1}(\ln 2) \tag{13}$$

The focus now shifts to Equation (12), because it is the only portion of the algorithm with height dependence.

In order to use the cepstrum to calculate the height, Equation (12) can be expanded using its Taylor Series. The relevant Taylor series is

$$\ln\left(1+\cos\left(\frac{2h}{c}\omega\right)\right) = \sum_{m=1}^{\infty}(-1)^{m+1}\frac{cos^m\left(\frac{2h}{c}\omega\right)}{m} \tag{14}$$

Substituting Equation (14) into (13) yields

$$\hat{x}_1(\mathrm{t}) = \sum_{m=1}^{\infty}(-1)^{m+1} * \mathcal{F}^{-1}\left(cos^m\left(\frac{2h}{c}\omega\right)\right) \tag{15}$$

where m is an index of summation. Equation (15) allows for an analytical interpretation of the results in the cepstrum domain, and can be developed by recognizing that $cos^m\left(\frac{2h}{c}\omega\right)$ is equivalent to a weighted sum of $cos\left(\frac{2h}{c}\omega\right)$ and its first *m* harmonics. This trigonometric identity is

$$\cos^m\left(\frac{2h}{c}\omega\right) = \sum_{k=0}^{m} A_k \cos\left(k\left(\frac{2h}{c}\right)\omega\right) \tag{16}$$

A_k is the weighting of each harmonic and *k* is the summation index. Proceeding with the derivation, substituting the trigonometric identity in Equation (16) into Equation (15) yields

$$\hat{x}_1(\mathrm{t}) = \sum_{m=1}^{\infty}\frac{(-1)^{m+1}}{m}\mathcal{F}^{-1}\sum_{k=0}^{m}\left(A_k cos\left(k\frac{2h}{c}\omega\right)\right) \tag{17}$$

When the inverse Fourier Transform is applied in Equation (17), the expression becomes

$$\hat{x}_1(\mathrm{t}) = \sum_{m=1}^{\infty} \frac{(-1)^{m+1}}{m} \sum_{k=0}^{m} A_k \left(\delta\left(\mathrm{t} + m\frac{2h}{c}\right) + \delta\left(\mathrm{t} - m\frac{2h}{c}\right) \right) \tag{18}$$

Equation (18) yields a sum of impulses at the echo arrival time of the helicopter's rotor noise (and integer multiples of this echo arrival time). The author will refer to multiple overlapping impulses as a "spike" for the remainder of this work.

At this point it would be helpful formulate $\hat{x}_1$ as a function of helicopter height rather than as echo arrival time, because helicopter height determination is the ultimate purpose of the altimeter. To convert from $\hat{x}_1(\mathrm{t})$ to $\hat{x}_1(z)$, the relationship that $z = \mathrm{ct}/2$ is used. The helicopter's height, *z*, is one-half the amount of time it takes for its rotor noise to travel to the ground and back, multiplied by the speed of sound. Substituting this into Equation (18), the equation becomes

$$\hat{x}_1(z) = \hat{x}_1\left(\frac{\mathrm{c}}{2}\mathrm{t}\right) = \sum_{m=1}^{\infty} \frac{(-1)^{m+1}}{m} \sum_{k=0}^{m} A_k \left(\delta(z + mh) + \delta(z - mh) \right) \tag{19}$$

While the literature has explored in considerable detail the exact amplitudes of the spike heights that result from the weighting A_k (Fjell 1976), only two of these observations are relevant to the present discussion. The first is that the weightings decay by a factor of 1/*m* and alternate in sign with each successive spike, and the second is that impulses overlap to form spikes, making it possible to simplify the double summation into a single sum.

The $A_k$values have been changed to $\tilde{A}_m$ to indicate that the actual values of the weighting have changed as a consequence of combining the summands. The simplified result is

$$\hat{x}_1(\mathrm{z}) = \sum_{m=1}^{\infty} \frac{\tilde{A}_m(-1)^{m+1}}{m}\big(\delta(t+mh) + \delta(t-mh)\big) \tag{20}$$

Returning to Equation (11), the relation $\hat{x}(\mathrm{t}) = \hat{x}_1(\mathrm{t}) + \hat{x}_2(\mathrm{t})$ can be applied. It now takes a few steps to simplify $\hat{x}_2$ and to convert it to a function of helicopter height rather than echo arrival time. From Equation (13) it is known that

$$\hat{x}_2(\mathrm{t}) = \mathcal{F}^{-1}(\ln|S(\omega)|^2) + \mathcal{F}^{-1}(\ln 2) \tag{21}$$

This expression can be simplified by recognizing that the first term is the cepstrum of the dereverberated signal, and that the second term is located at t=0. These simplifications yield

$$\hat{x}_2(\mathrm{t}) = \hat{\mathrm{s}}(\mathrm{t}) + (\ln 2)\,\delta(\mathrm{t}) \tag{22}$$

In Equation (22), $\hat{\mathrm{s}}(\mathrm{t})$ has now become the cepstrum of the signal in the absence of ground reflection.

Equation (22) can be converted to distance using the same substitution of $z = \mathrm{ct}/2$ as has been done in Equation (18), yielding

$$\hat{x}_2(z) = \hat{x}_2\left(\frac{\mathrm{ct}}{2}\right) = \hat{\mathrm{s}}\left(\frac{\mathrm{ct}}{2}\right) + (\ln 2)\,\delta\left(\frac{\mathrm{ct}}{2}\right) = \hat{\mathrm{s}}(z) + (\ln 2)\,\delta(z) \tag{23}$$

When Equation $\hat{x}_1$ and $\hat{x}_2$ are combined, Equation (24) is the result. Equation (24) consists of two portions: The first portion represents $\hat{x}_1(z)$ and is a spike train that alternates in sign and decays by 1/*m* with each successive impulse. The second portion represents $\hat{x}_2(z)$ and is the cepstrum of the reflection-free quantity $\hat{\mathrm{s}}(\mathrm{t})$. Combining $\hat{x}_1(z)$ and $\hat{x}_2(z)$ yields

$$\hat{x}(\mathrm{z}) = \sum_{m=1}^{\infty} \frac{\tilde{A}_m(-1)^{m+1}}{m}\big(\delta(z+mh) + \delta(z-mh)\big) + \ln 2\,\delta(z) + \hat{\mathrm{s}}(z) \tag{24}$$

In Equation (24), the input signal has been completely converted into the cepstrum domain. The change in domain is accompanied by a set of descriptive terms to describe the independent and dependent variables. The independent variable z can be referred to as the quefrency and the dependent variable $\hat{x}(z)$ can be referred to as gamnitude. Quefrency is an anagram of frequency, and gamnitude in the cepstrum domain is an anagram of magnitude in the frequency domain.

Figure 5 shows an example of a typical cepstrum of a simulated helicopter hovering at 1 meter (m), demonstrating all the features discussed in the present derivation of the analytical form of the helicopter's height. It shows the multiple spikes decaying in amplitude at integer multiples of the helicopter height along the quefrency axis. There is also a noise threshold resulting from the cepstrum of the incident helicopter signal ($\hat{s}(z)$). The figure does not show the alternating signs arising from $(-1)^{m+1}$ because the figure shows the absolute value of the final cepstrum. By taking the absolute value, this quantity (sometimes referred to as the power cepstrum in the literature) looks at the total power at each quefrency.

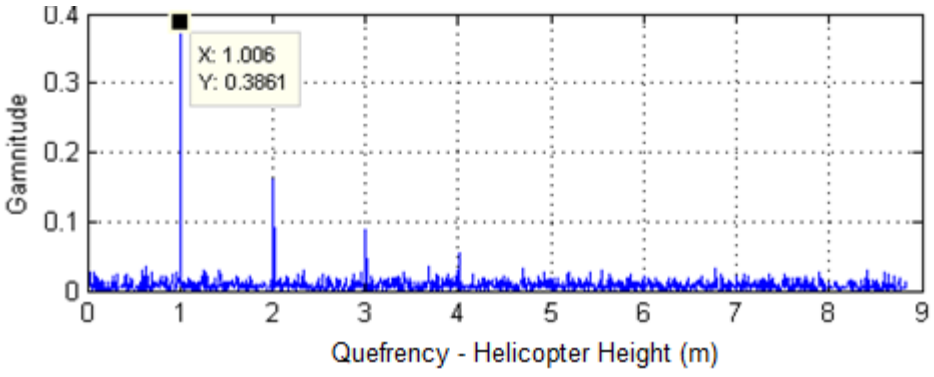

Figure 5 - Cepstrum of Helicopter at 1m

2.3. Compound Height Detection

This section considers the case in which the helicopter's noise is reflected from more than one barrier. In such cases, each reflection will generate its echo arrival, and thus its unique set of spikes in the cepstrum domain. Figure 6 illustrates the path lengths that will yield spikes in the cepstrum domain.

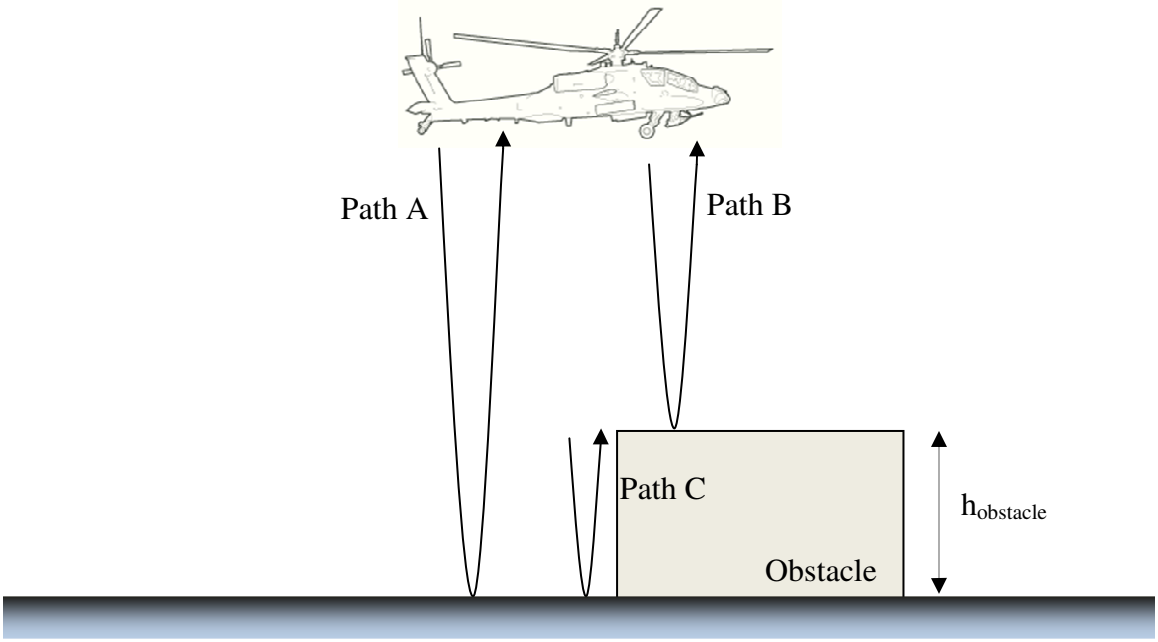

Figure 6 - Multiple echoes leads to multiple spikes in the cepstrum domain

Both Path A and B have a cepstrum spike set because they correspond to echo arrivals. Path C generates a spike set because the echo from Path A (Echo A) is a time delayed version of the echo from Path B (Echo B). To the cepstrum algorithm, the difference in time between the arrival of Echo A and Echo B appears as a third echo, and yields a third spike third spike set in the cepstrum domain.

As the helicopter descends, which spikes correspond to real echo arrivals and which spikes correspond to fixed distances can be determined, because Path C will remain fixed, whereas Path A and Path B decrease.

Figure 7 shows the results of a simulation of a helicopter descending toward the ground. In the top chart, the helicopter is 9 m above the ground, as represented by spike A, and 8 m above an obstacle, as represented by the spike B. In the middle frame, spikes A and B indicate that the helicopter is 7 m above the ground and 6 m above the obstacle. In the final frame, the helicopter is 5 m above the ground but 4 m above the obstacle. In all frames, Path C remains fixed. Another important observation that arises from

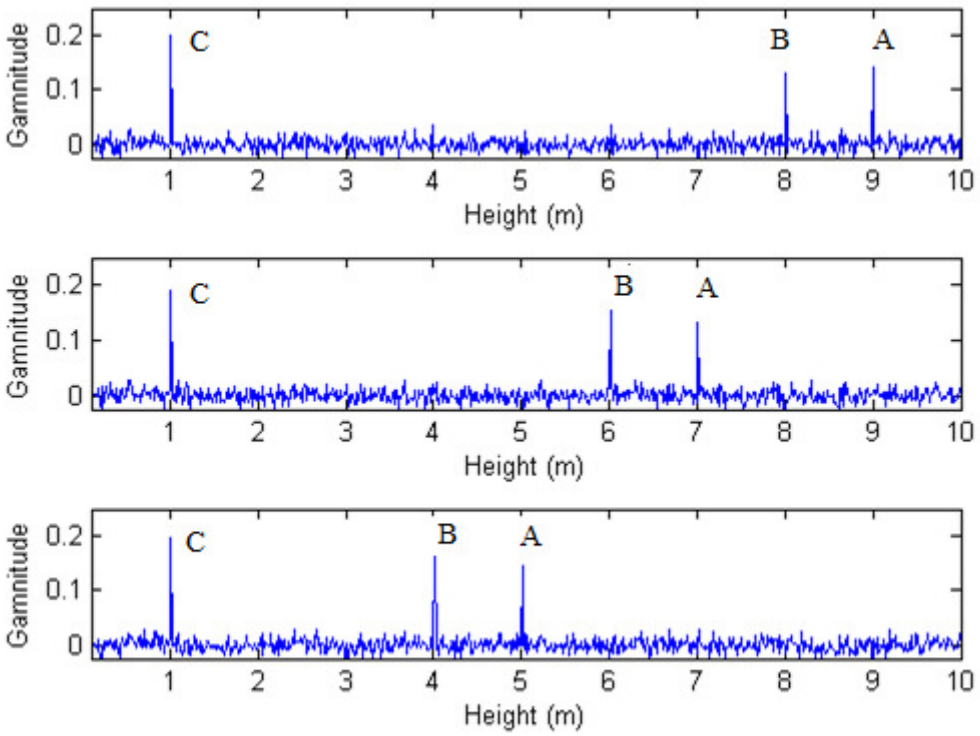

Figure 7 - Heights of helicopter as it descends

Figure 7 is the disappearance of higher order spikes, which were visible in both the earlier simulation and the derivation of the helicopter height. The spikes become obscured when additional noise is present or when an obstacle is added. Only a small amount of additional noise is required to obscure the extraneous spikes, which enables detection of height based on the main spike alone. The reason for this disappearance is unclear, but it has been noted (also without explanation) in the literature (Fjell 1976). One

possible explanation is that, because the smaller spikes are caused by higher order terms, they are more easily covered up by the background noise.

2.4. Velocity Detection

This section explains how a helicopter's descent velocity can be determined with the system proposed in this thesis. There are two methods that can be used to detect the velocity. The first and most basic method of velocity detection is to find the difference in the helicopter's height at different times. The velocity is simply equal to change in height divided by change in time.

If the helicopter descends with a sufficiently rapid vertical velocity, then velocity can be detected by compressions in frequency content in the reflected signal relative to the incident signal.

Doppler distortion is an acoustic phenomenon that occurs when the source is in motion with respect to the receiver. In this case, the source is the image of the helicopter's rotor, and the receiver is the observer on the helicopter. Figure 8 illustrates how the descending helicopter will receive the reflected signal from its image.
The Doppler distortion doubles in this case because the source and receiver are approaching each other at the same speed. The Doppler distortion will be determined as

$$\omega_{shifted} = \omega_0 \left(1 + 2\frac{v_{rec}}{c}\right) \tag{25}$$

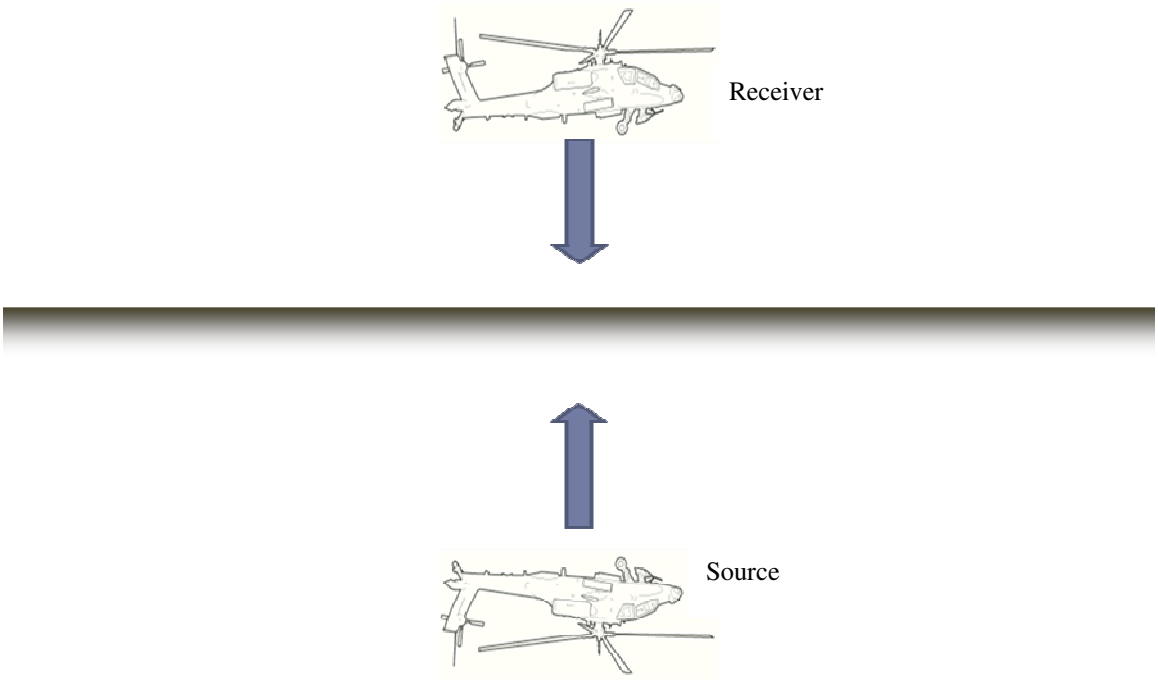

Figure 8 - The helicopter approaching the ground is an image problem

The Doppler distortion can be used to calculate the velocity of the helicopter as it approaches the ground because there will be a difference in the frequency content of the incident signal and the reflected signal. To extract the differences in frequency content between the incident and reflected signal, a new algorithm will be derived in the following. First, a general overview will presented, and next, a flow chart will be used to illustrate the steps of the algorithm. Then, the details of the algorithm will be explained fully.

Since the objective of the algorithm is to detect the velocity of the helicopter by comparing the differences between the two signals (incident and reflected), the spectra of the two signals are added. Once added, the Mellin-Scale transform is used to extract the scale shift of the data (Mellin-Scale transforms are defined in Appendix A). These transforms are analogous to Fourier transforms. Fourier transforms treat frequency as a physical quantity, where Mellin-Scale transforms treat “scale” as a physical quantity.

After the Mellin-scale transform operates on the data, the natural log and IFFT are performed to turn the time scale data into a spike that can be detected.

The Mellin-Scale transform operates on time scaling in a way similar to the Fourier transform operation on time shifts. A time delay can be transformed into the frequency domain as

$$x(t) = s\left(t - \frac{2h}{c}\right) \xrightarrow{time/freq} X(\omega) = S(\omega)e^{-i\omega\frac{2h}{c}} \tag{26}$$

Here, $x(t)$ is an observed time signal, and $X(\omega)$ is the representation of $x(t)$ in the frequency domain.

The Mellin-Scale transform can yield a similar result, given a signal which has been scaled. A scaled signal is transformed from the time domain into the Mellin-Scale domain as

$$x(t) = s(2v_0 t) \xrightarrow{time/scale} X(\Omega) = S(\Omega)e^{-(i\Omega)\ln(2v_0)} \tag{27}$$

where $x(t)$ is again an observed time signal, but a_0 is the scaling coefficient. $X(\Omega)$ is the representation of $x(t)$ in the Mellin-scale domain.

Figure 9 is a flow chart that summarizes the algorithm. This algorithm will be referred to as the Doppler method throughout this thesis.

Figure 9 – Doppler method for velocity detection

Now that the overarching idea and motivation behind the Doppler velocity detection method has been explained, the theory can be explained in somewhat more detail and with mathematical rigor.

At the core of the algorithm is the use of the Mellin-Scale transform to extract scale information from a function. In the following pages, the Mellin-Scale transform is defined, and explained, and its scale-extracting quality is derived.

The Mellin-Scale transform is defined as

$$F(\Omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(\mathrm{t})\, \mathrm{t}^{-i\Omega - 1/2}\, d\mathrm{t} \tag{28}$$

where:

$\mathrm{t} = \text{Independent Time Domain variable,}$

$f(\mathrm{t}) = \text{Function of independent time variable,}$

$\Omega = \text{Independent Mellin} - \text{scale variable,}$

$\mathrm{F}(\Omega) = \text{Function of independent Mellin} - \text{scale variable}$

Consider a function compressed by a factor a_0 operated on by the Mellin-Scale transform

$$G(\Omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(a_0 \mathrm{t})\, \mathrm{t}^{-i\Omega - 1/2}\, d\mathrm{t} \tag{29}$$

Substituting $\tau = a_0 \mathrm{t}$, Equation (29) becomes

$$G(\Omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(\tau) \left(\frac{\tau}{a_0}\right)^{-i\Omega - 1/2} d\left(\frac{\tau}{a_0}\right) \tag{30}$$

By differentiating the differential term and regrouping terms, Equation (30) becomes

$$G(\Omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(\tau) \tau^{-i\Omega - 1/2}\, d\tau \left(\frac{a_0^{i\Omega + 1/2}}{a_0}\right) \tag{31}$$

In one final manipulation, the terms with Ω are put into a complex exponential. The remaining $\sqrt{a_0}$ is grouped with the $1/\sqrt{2\pi}$ to clarify what is constant and what is not; then

$$G(\Omega) = \sqrt{\frac{a_0}{2\pi}} F(\Omega) e^{\ln(a_0) i\Omega} \tag{32}$$

Now, the result of the time scaling $g(t) = f(a_0 \mathrm{t})$ can be compared to the well-known result of the Fourier transform of a time-delayed signal, $h(\mathrm{t}) = f(\mathrm{t} - \mathrm{t}_0)$, or

$$H(\omega) = \sqrt{\frac{1}{2\pi}} F(\omega) e^{-i\mathrm{t}_0 \omega} \tag{33}$$

The similarity between Equations (32) and (33) is the basis for the velocity detection algorithm. It has already been shown that existing techniques can detect height by determining the time delay between the incident and reflected signals. These techniques can be modified to detect the amount of Doppler distortion between incident and reflected signals.

An algorithm is now derived that can take this property of the Mellin-Scale transform and apply it to a helicopter attempting to land in brownout conditions. The input to the algorithm is two separate signals received by opposite sides of the microphone array. The front of the array detects the incident signal x_{inc}, and the back of the array detects the reflected signal x_{ref}. The term $s(t)$ represents the unaltered rotor noise signal and is the same as x_{inc}. The incident signal is

$$x_{inc} = s(\mathrm{t}) \tag{34}$$

The expression for the reflected sound also accounts for the Doppler shift and the time delay of the rotor signal. The reflected signal is

$$x_{ref} = \sqrt{a_0} s\big(a_0(\mathrm{t} - \mathrm{t}_0)\big) \tag{35}$$

After finding the power spectrum of x_{inc} and x_{ref}, the two terms are summed to find

$$X_{\mathcal{F}}(\omega) = |\mathcal{F}(x_{inc})|^2 + |\mathcal{F}(x_{ref})|^2 = |S(\omega)|^2 + |S(\omega/a_0)|^2 \quad (36)$$

It is assumed that a_0 is close to 1 because the helicopter will not be traveling very fast toward the ground relative to the speed of sound. Equation (36) does not yield the spectrum of the total waveform. Rather, it sums the incident spectrum and reflected spectrum separately. By keeping these summations separate, cross terms involving both Doppler-distorted and undistorted terms are avoided.

Next, the Mellin-Scale transform is applied to quantify the Doppler shift of the reflected waveform with respect to the incident signal. When the transform is applied, the equation becomes

$$\mathcal{M}\{X_{\mathcal{F}}(\omega)\} = X_{\mathcal{MF}}(\Omega) = \mathcal{M}\{|S(\omega)|^2 + |S(\omega/a_0)|^2\} \quad (37)$$

Because the Mellin-Scale Transform is a linear operator, it can operate on the two terms separately

$$X_{\mathcal{MF}}(\Omega) = \mathcal{M}\{|S(\omega)|^2\} + \mathcal{M}\{|S(\omega/a_0)|^2\} \quad (38)$$

Using Mellin-scale identity in Equation (32), $X_{\mathcal{MF}}(\Omega)$ becomes

$$X_{\mathcal{MF}}(\Omega) = \mathcal{M}\{|S(\omega)|^2\} + \mathcal{M}\{|S(\omega)|^2\} * \sqrt{a_0}\, e^{\ln(a)(-i\Omega)} \quad (39)$$

This can be simplified by rewriting $\mathcal{M}\{|S(\omega)|^2\}$ as $S(\Omega)$ in the equation

$$X_{\mathcal{MF}}(\Omega) = S(\Omega) + S(\Omega) * e^{\ln(a)(-i\Omega)}\sqrt{a_0} \quad (40)$$

With this expression, the amount of Doppler distortion (a_0) can be extracted using Fourier transforms, similar to the way the amount of time delay is calculated using the cepstrum. To do this, the logarithm of the square of Equation (40) is taken to separate the Mellin-Scale spectrum from the distortion terms. The following equations carry out that

manipulation, as it was carried out in the earlier section when deriving the cepstrum. The natural logarithm of $X_{\mathcal{MF}}(\Omega)$ becomes:

$$\ln\left(X_{\mathcal{MF}}(\Omega)\right) = \ln|S(\Omega)|^2 + \ln\left|1 + \sqrt{a_0}e^{\ln(a_0)(-i\Omega)}\right|^2 \tag{41}$$

$$= \ln|S(\Omega)|^2 + \ln\left(1 + a_0 + 2\sqrt{a_0}\cos(\Omega \ln a_0)\right) \tag{42}$$

$$= \ln|S(\Omega)|^2 + \ln(1 + a_0) + \ln\left(1 + \frac{2\sqrt{a_0}}{1+a_0}\cos(\Omega \ln a_0)\right) \tag{43}$$

This equation can be rewritten using the Taylor expansion

$$\ln(1 + \alpha\cos\theta) = \sum_{m=1}^{\infty}(-1)^{m+1} * \frac{(\alpha\cos\theta)^{\mathrm{m}}}{m} \tag{44}$$

To apply this Taylor expansion correctly, $\alpha = \frac{2\sqrt{a_0}}{1+a_0}$ and $\theta = \Omega \ln a_0$. Using these substitutions, and carrying out the Taylor Expansion, it is found that

$$\ln\left(X_{\mathcal{MF}}(\Omega)\right) = \ln|S(\Omega)|^2 + \ln(1 + a_0) + \frac{2\sqrt{a_0}}{1 + a_0}\cos(\Omega \ln a_0) \tag{45}$$

Taking the inverse Fourier transform (with Ω as the frequency variable) will yield a spike at $\ln a_0$ in this new domain

$$X_{\mathcal{F}^{-1}\mathcal{MF}}(\tau) = \mathcal{F}^{-1}\left\{\ln\left(|S(\Omega)|^2(1 + a_0)\right)\right\} + \frac{2\sqrt{a_0}}{1 + a_0}\delta(\tau \pm \ln a_0) \tag{46}$$

Now the amount of Doppler distortion (and thus the velocity) can be evaluated by locating the spikes in $X_{\mathcal{F}^{-1}\mathcal{MF}}$.

Figure 10 summarizes the previous algorithm derivation step by step.

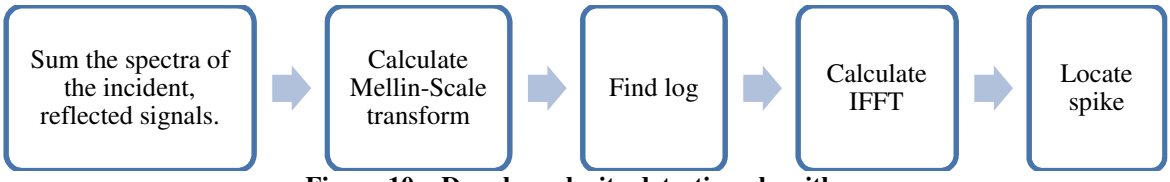

Figure 10 – Doppler velocity detection algorithm

The simulations used to study this algorithm will be similar in structure to the simulations testing the cepstrum algorithm. The algorithm will be tested in three different situations, as follows.

1. Extraneous noise: Random noise will be added to both the incident and reflected signal.
2. Front to back leakage: Some amount of the incident signal will be leaked to the reflected signal to evaluate the directivity requirements of the array.
3. Bandwidth limitations: The only signal available will be the noise created by the rotor and the engine. This is a signal with finite bandwidth. The amount of bandwidth needed in order to obtain a reliable measurement will be determined.

Further results and details of these simulations are found in the Simulations section.

2.5. Experimental Testing

To test the algorithms discussed here, a microphone array was designed to allow for separate detection of the incident and reflected signals. The separation of the signals will allow for the amplification of one channel as it becomes attenuated through propagation losses.

The array uses multiple microphones to narrow the beamwidth. This narrower beamwidth allows side rejection of noise coming in from the side that could degrade the quality of the spectrum. Each of the individual microphones in the array has a cardioid directivity that improves the backside rejection.

The array was tested in three different environments: an anechoic chamber, a hemi-anechoic chamber, and outside in Georgia Tech's Burger Bowl (a relatively flat athletic field with a grass surface). Each testing location had a special purpose. Tests

were conducted in the anechoic chamber to test the array's frequency sensitivity and directivity. The hemianechoic chamber was used to simulate an ideally quiet environment with a perfectly reflecting and flat ground. The setup was tested outdoors to find the maximum height at which the array was effective, and to apply more realistic test conditions.

CHAPTER 3

SIMULATIONS

Four simulations were conducted: height detection, obstacle detection, velocity detection, and brownout distortion. The height detection simulation tested the effectiveness of the cepstrum routine. The obstacle detection simulation studied the ability of the cepstrum algorithm to locate obstacles. The velocity detection section evaluates two different ways to calculate velocity and to determine under which circumstances a given method was preferable. The brownout distortion section examines how the swirling brownout dust cloud affected the rotor's noise signal.

3.1. Probability of Detection

To determine the reliability and robustness of the height and velocity detection algorithms, a "probability of detection" quantity is defined here. The probability of detection (POD) is determined by recalculating an algorithm (the cepstrum, for example) several thousand times. In each trial, the detected quantity was determined to be the maximum value within a certain valid region. The detection was said to be accurate if it fell within ± 1 data point of the correct value. In all other cases, the detection was said to be inaccurate. The POD is the number of accurate detections divided by the total number of trials.

By forcing the algorithm to detect a single successful detection, the algorithm becomes robust to redundant spikes from a single reflection as seen in Equation (24). This robustness occurs because the algorithm detects the highest maximum (within a certain region) for each trial. While there are multiple spikes in the region, the highest

amplitude spike will be the first. This spike, which also corresponds to the correct height, is highest because the spikes decay by 1/*m* as quefrency increases. Equation 24 shows in detail the way that the spikes decay by 1/*m* with each subsequent spike. The behavior can be seen in Figure 5. One drawback of looking at the POD is that receiver operating characteristic curves cannot be generated because the threshold effectively changes with each trial.

3.2. Height Detection

3.2.1. Basic Height Detection

The purpose of the basic height detection tests was to verify the abilities of the cepstrum algorithm. The algorithm was tested by taking the cepstrum of a random white noise signal added to a delayed version of itself.

Figure 11 shows the results from one of these simulations. In this case, the sampling rate was set to 25 kHz, the sampling length to 0.15 seconds, and the helicopter height was set to 1 m. These settings were based on the upper limits for the data acquisition board in the experimental setup.

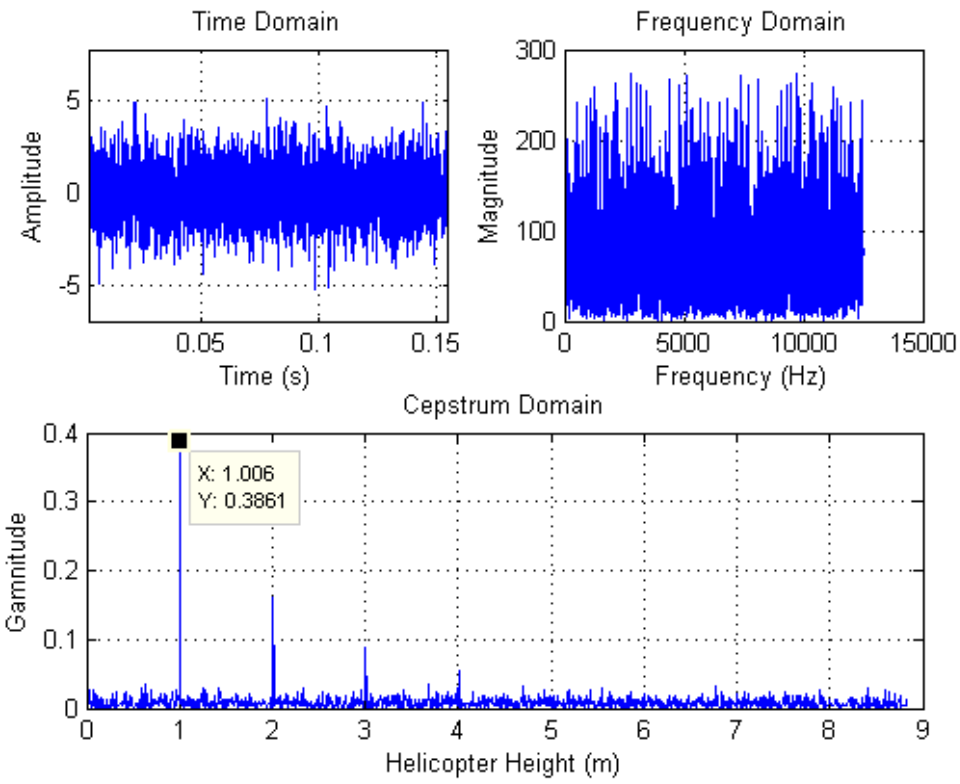

Figure 11 – Echoed white noise echo and the absolute value of its cepstrum

The time domain graph (top left) shows noise generated with MATLAB's randn() function. The frequency domain graph (top right) shows the results of a power spectrum found using only the absolute value of the FFT of the input signal. The cepstrum domain graph (bottom) shows the decaying spikes at integer multiples of the helicopter height. Since the algorithm selects only the tallest spike, the height is accurately detected at 1 m. The other spikes are ignored. In the cepstrum domain graph, there is a clear but small noise floor. This noise floor becomes closer to the spike height as external noise is added, as will be explored in the following section.

3.2.2. Adding Extraneous Random Noise

The previous results discuss the spike's height in the case where there is no noise. However, in real world situations, there will be a significant amount of extraneous noise,

for instance, sound from a nearby helicopter. To test how the noise would affect the cepstrum algorithm, extraneous and uncorrelated random white noise was added to the process.

Figure 12 shows the signal-to-noise ratio (SNR) in the time domain plotted against the height's POD. The "signal" here is the sound from the rotor blades. The noise is random white noise that is not related to the signal. The SNR in the time domain is shown in decibels, and is found by

$$SNR = 20\log_{10}\left(RMS_{signal}/RMS_{noise}\right) \tag{47}$$

The blue dots represent individual trials. The green line shows the detection threshold of 90% at a signal to noise ratio of -6 dB.

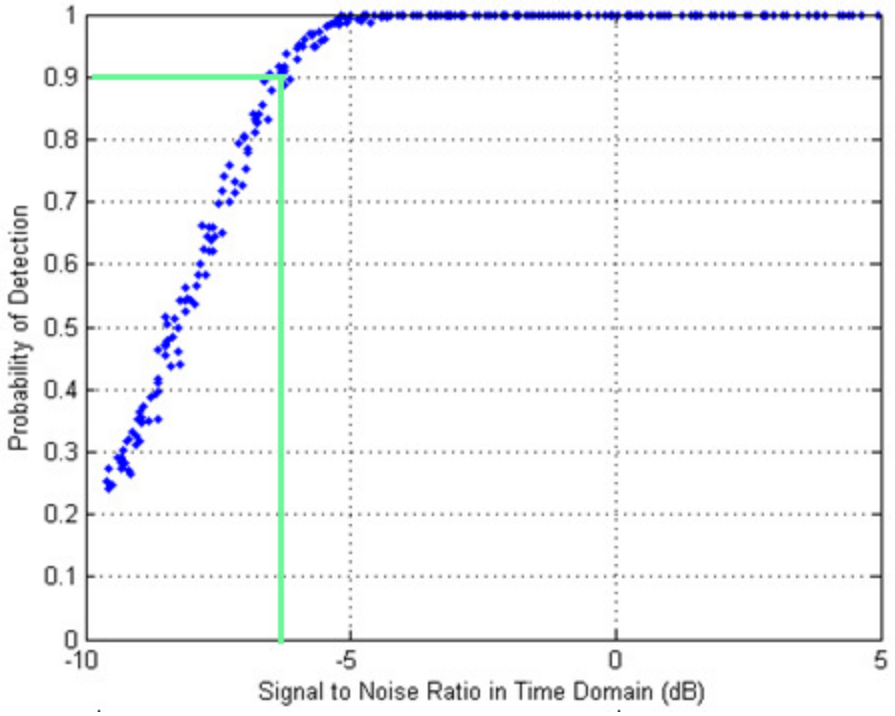

Figure 12 – Signal-to-noise ratios in the time domain affects the likelihood of detection

The most important trend here is that the cepstrum only requires a -6 dB signal to noise ratio to detect height reliably (> 90% of the time). Once the noise becomes more

than 6 dB louder than the signal, the cepstrum begins to become unreliable. This is significant because it suggests that a high POD is possible in the application.

In reality, the SNR will probably be much better than -6 dB because the helicopter noise is loud compared to the surroundings. The biggest contributor to noise will probably be helicopter noise that gets rejected from one side of the array and then accepted on its return path.

3.2.3. Bandwidth Limitations

Limiting the bandwidth can affect the signal-to-noise ratio and the precision of the height measurement in the cepstrum domain. However, there is no preferred bandwidth region.

Figure 13 shows results from signals that have been passed through a low pass, a high pass, and a band pass filter. The signal is for a helicopter that hovers approximately 1 meter above the ground with no extraneous noise. The SNR of the cepstrum domain is determined by

$$SNR_{ceps} = 20\log_{10}\left(Spike\ Gamnitude/RMS_{ceps}\right) \tag{48}$$

The figure demonstrates that the SNR of each signal depends on the amount of bandwidth present and not the location of the bandwidth. As long as there is more than 1 kHz of bandwidth available, the spike height will be above the noise floor, which was determined to be approximately 10 dB, using the cepstrum's RMS as a reference value. These results are also noteworthy because they do not vary more than 4 dB, even when the height of the helicopter changes as much as 7 m.

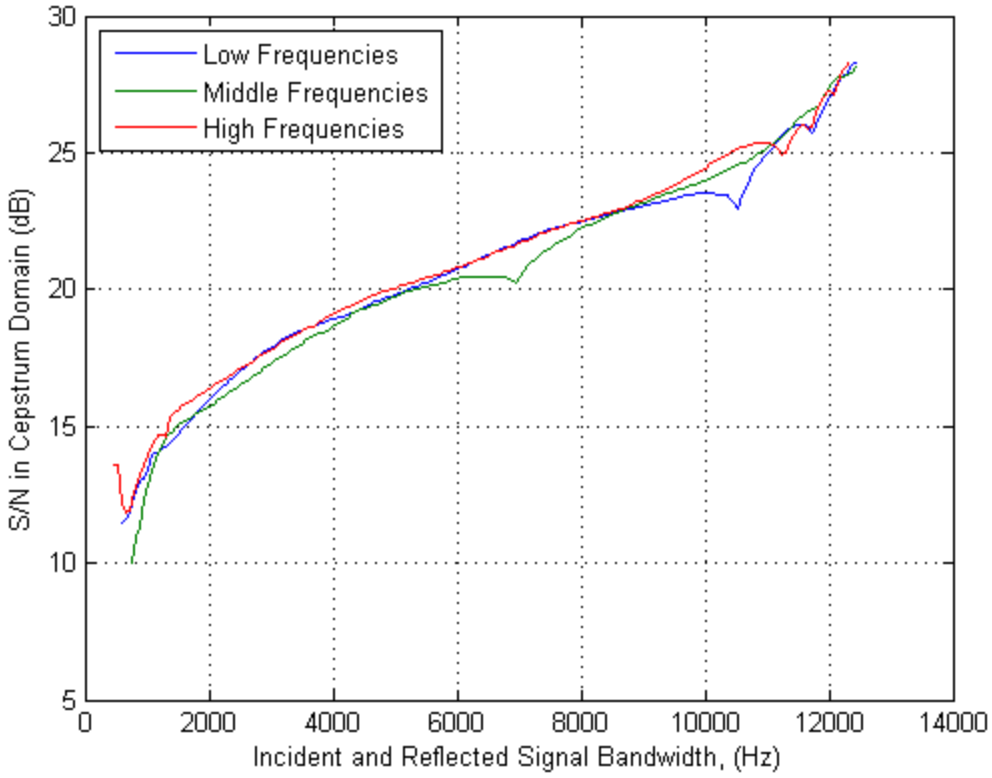

Figure 13 - S/N Ratio depends on amount of bandwidth

Figure 14 shows how limiting the bandwidth can degrade and smooth the cepstrum spike. This cepstrum smoothing is a logical consequence of the time-bandwidth product, which says that narrowly defined pulses in the frequency domain correspond to spread-out signals in the time domain. As the amount of bandwidth becomes smaller, the signal in the cepstrum domain is spread out.

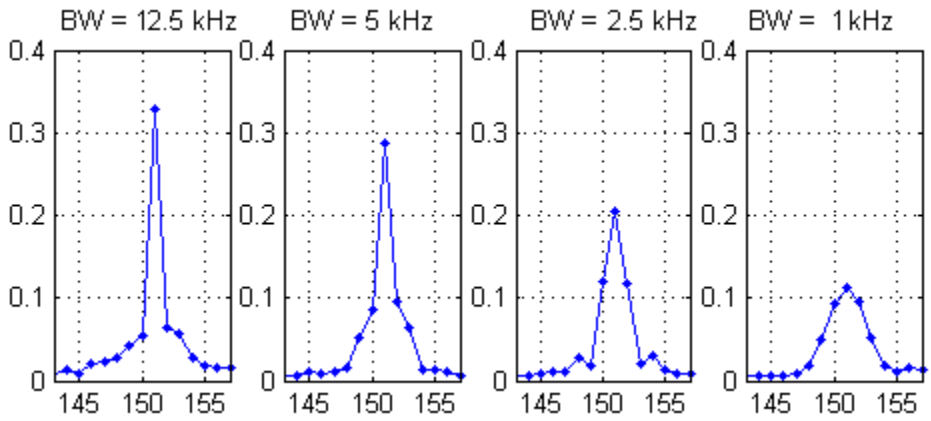

Figure 14 –Effects of Limiting Bandwidth on Spike Strength and Shape

3.3. Velocity Detection

Two techniques to detect velocity were studied for this thesis. The first uses the height detection methods discussed previously, and then finds the change in height divided by change in time. Since $v = \Delta h/\Delta t$, the velocity can be determined. The second method is a novel method that looks at the Doppler distortion between the downward incident wave and the upward reflecting wave. This distortion in signal can be determined using an algorithm similar to the cepstrum algorithm.

3.3.1. Measuring Velocity at Slow Speeds

The success of the cepstrum algorithm for detecting height makes it a logical starting point from which to calculate the velocity. Velocity can be calculated by using difference techniques, i.e., $\Delta h/\Delta t$. However, this technique is less reliable when the helicopter ascends or descends too quickly, as will now be shown, because Doppler distortion makes the height impossible to measure.

Figure 15 shows the probability of detecting the helicopter's height as it descends at a constant rate. The Doppler distortion was simulated by resampling the incident noise

signal and then applying the original time basis to the resampled signal. At descents faster than 1 m/s (about 2 knots), the cepstrum can no longer reliably measure the height or calculate velocities. The graph shows that the successful detection rate falls off precipitously around 1 m/s.

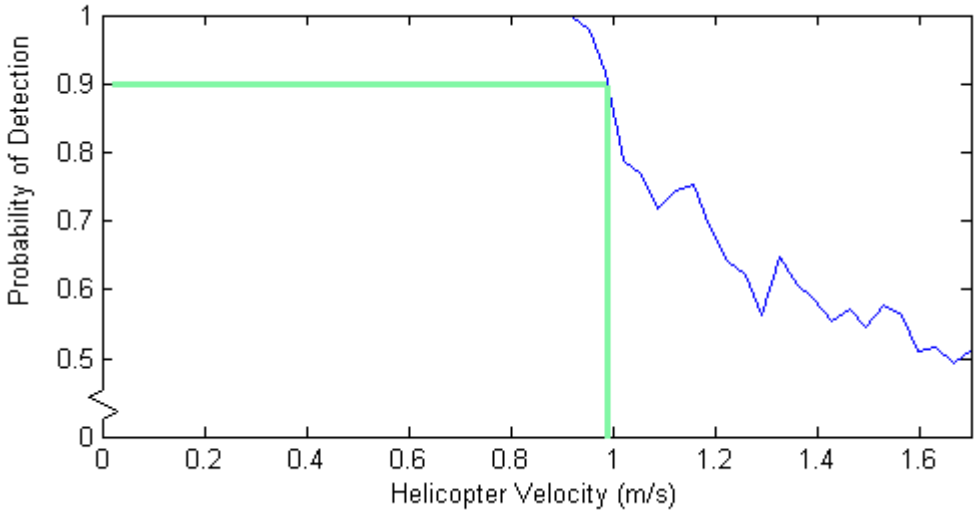

Figure 15 – Cepstrum-based Height Detection (difference method) POD

3.3.2. Measuring Velocity at Mid-range Speeds

While the cepstrum method works for very small velocities, the Doppler method works for the medium speed velocities occurring in the helicopter's descent. This speed range falls between 1 m/s and 6 m/s. Descents faster than 6 m/s close to the ground are extremely dangerous and beyond the scope of the work in this thesis (Wald 2000).

Figure 16 shows an evaluation of the accuracy of the Doppler velocity detection method. The simulations show that the algorithm is sensitive and reasonably accurate in this medium velocity region, but that the algorithm becomes less reliable at very slow velocities. At velocities much slower than 1 m/s, quantization effects become a problem. However, velocity can be calculated using the difference method at these slower speeds.

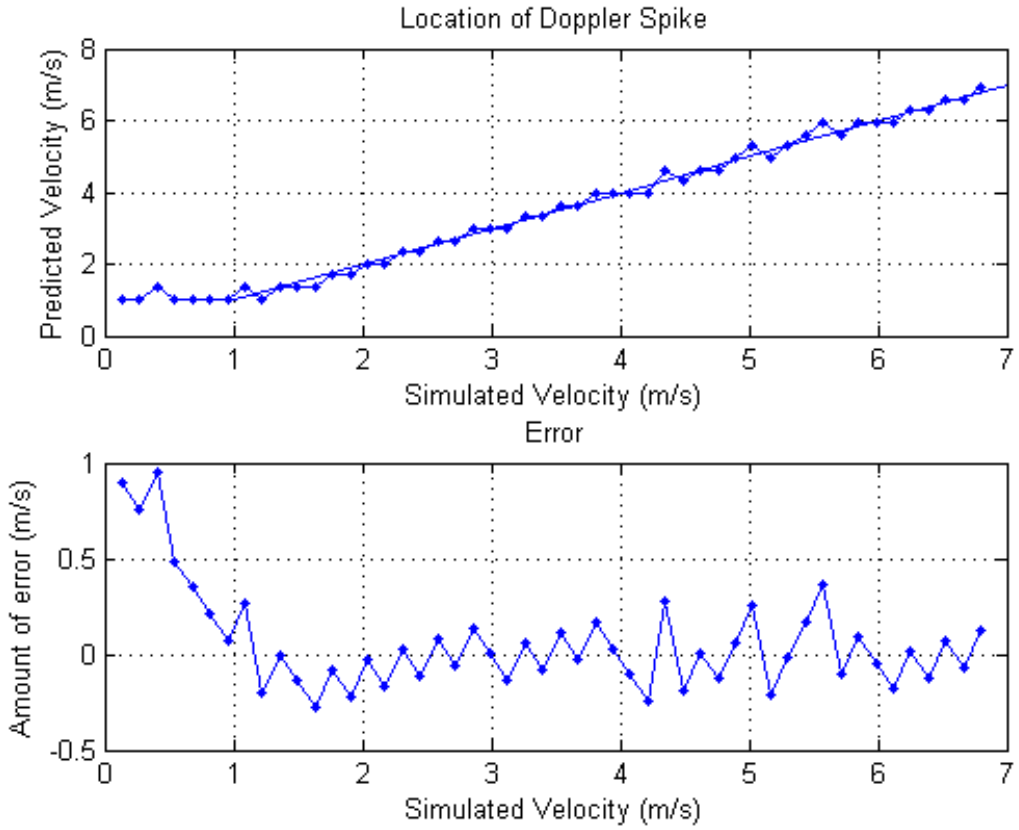

Figure 16 – Doppler Method Velocity Detection Accuracy

3.3.3. Extraneous Noise and the Doppler Method

One way to test the Doppler method is to add extraneous noise and examine how reliably the algorithm determines velocity. Simulations show that the algorithm can detect velocities reliably as long as the SNR of the acquired data is greater than 9 dB. Below 9 dB, it becomes easy to lose the Doppler spike in the noise. Figure 17 shows how POD is affected by signal to noise ratio in the time domain. Each trial in this simulation had a sampling frequency of 25 kHz and used 0.15 seconds of data. Individual trials are represented by blue dots. The green line shows that the POD is 90% when the SNR is 9 dB.

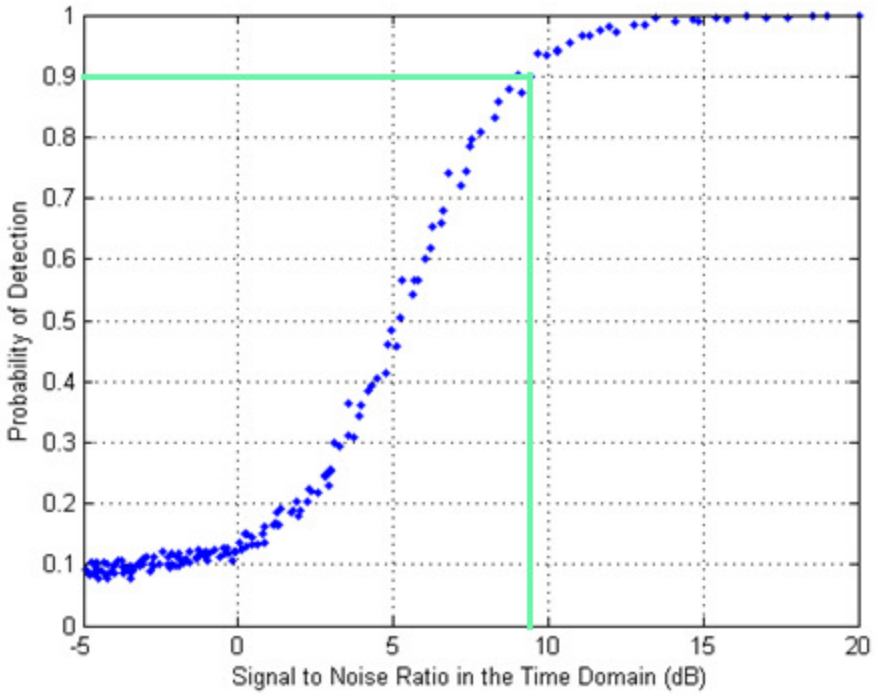

Figure 17 – Doppler Method POD with Added Noise

3.3.4. Bandwidth Limitations and the Doppler Method

Not surprisingly, limiting the bandwidth negatively affects the reliability of the Doppler method. Figure 18 shows the POD given the amount of bandwidth. Each trial in this simulation had a sampling frequency of 25 kHz and used 0.15 seconds of data. The POD drops substantially when there is less than 8 kHz of bandwidth.

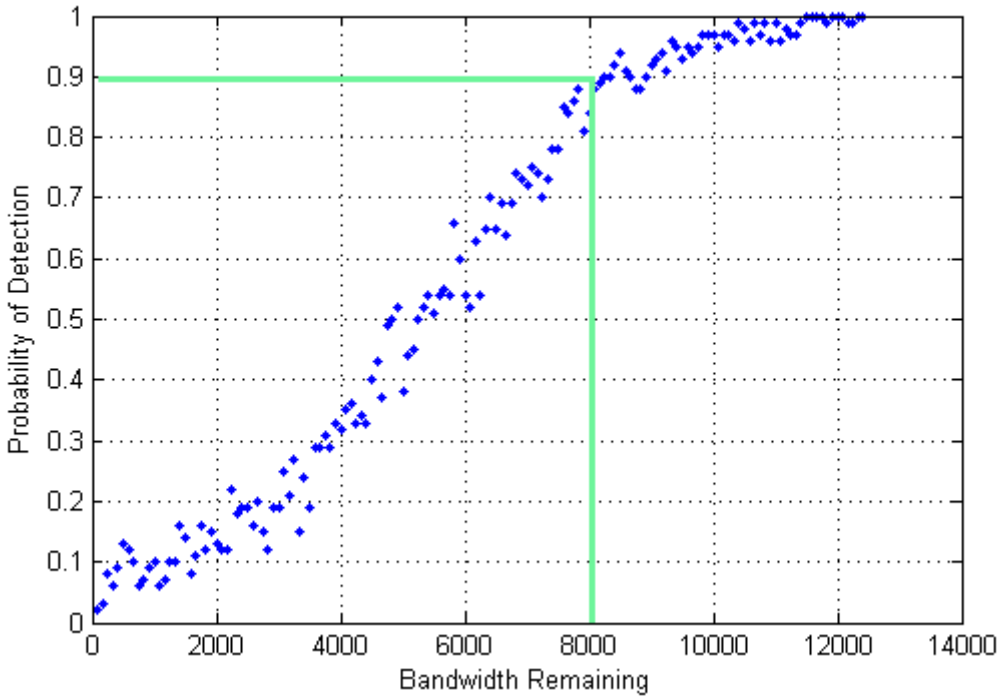

Figure 18 – Effects of Bandwidth Limitations on Doppler Method

3.3.5. Front-to-Back Leakage and the Doppler Method

Front–to-back leakage describes a condition that occurs when a signal that is received on one side of the array is also received on the reverse side. Front refers to the side of the array facing the incident signal and back refers to the opposite side.

This mix of incident and reflected signals is undesirable because the derivation of the velocity detection algorithm requires the two signals to be separate. In the derivation of the velocity detection algorithm, the separation made an analytical solution possible. To determine if the separation is necessary, the POD was calculated. To calculate the POD curve, the helicopter signal was again assumed to be 0.15 seconds of white noise generated with Matlab's randn() function. The helicopter was given a constant velocity of approximately 3 m/s downward. The sampling rate is 25 kHz. In each trial:

$$\text{Incident Signal} = x[n] + \alpha * x[1.02(n - 250)]$$

$$\text{Reflected Signal} = x[1.02(n - 250)] + \alpha * x[n]$$

where α represents the fraction of signal leaked to the opposite side. There were 1000 trials per data point and 100 data points.

The results of this simulation in Figure 19 show that although this separation is not a requirement, it is helpful. A 2 dB reduction between the front and back of the array will give a POD of 90%.

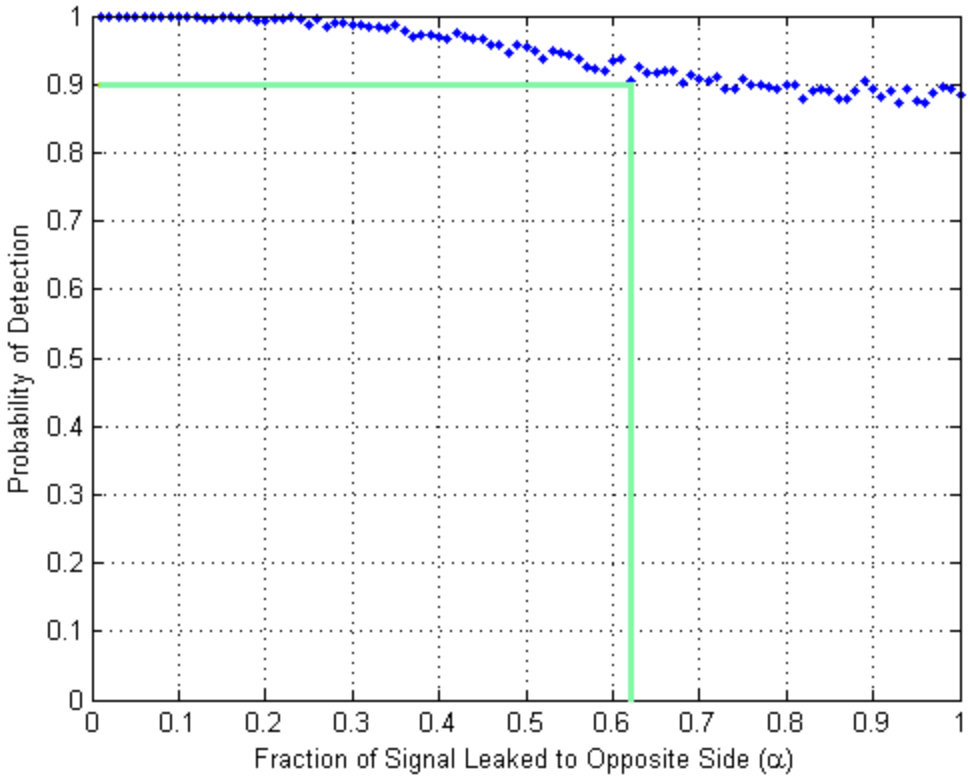

Figure 19 – Effects of Front-to-back leakage on Doppler Method

Leakage does not affect the POD as much as bandwidth and extraneous noise do. However, separating the signals will allow for more flexibility in post-processing. For instance, there may be times when the helicopter is high enough that the reflected signal has been attenuated significantly because of geometrical spreading. If the array can detect

the reflected signal separately from the incident signal, it can amplify the reflected signal to match the incident signal.

CHAPTER 4

DESIGN AND TESTING

This section discusses the array design, gives an overview of the experimental procedure, and then presents results from each type of testing (height, obstacle detection, and velocity).

4.1. Array Design

An array of microphones was designed and tested to meet the specifications of the objective. Based on the simulations, there were three main requirements:

1. A wide frequency response. For the cepstrum, simulations showed that 1 kHz was sufficient bandwidth. The Doppler velocity detection algorithm required closer to 8 kHz of bandwidth.
2. Separation of upward and downward signals. The POD in the Doppler velocity detection algorithm was improved when there was at least 2 dB in front to back rejection.
3. Strong directivity. There could be additional spikes in the cepstrum domain if the array picks up signals arriving from clutter. Figure 20 demonstrates an example of a clutter path that could be mistaken for the actual height.

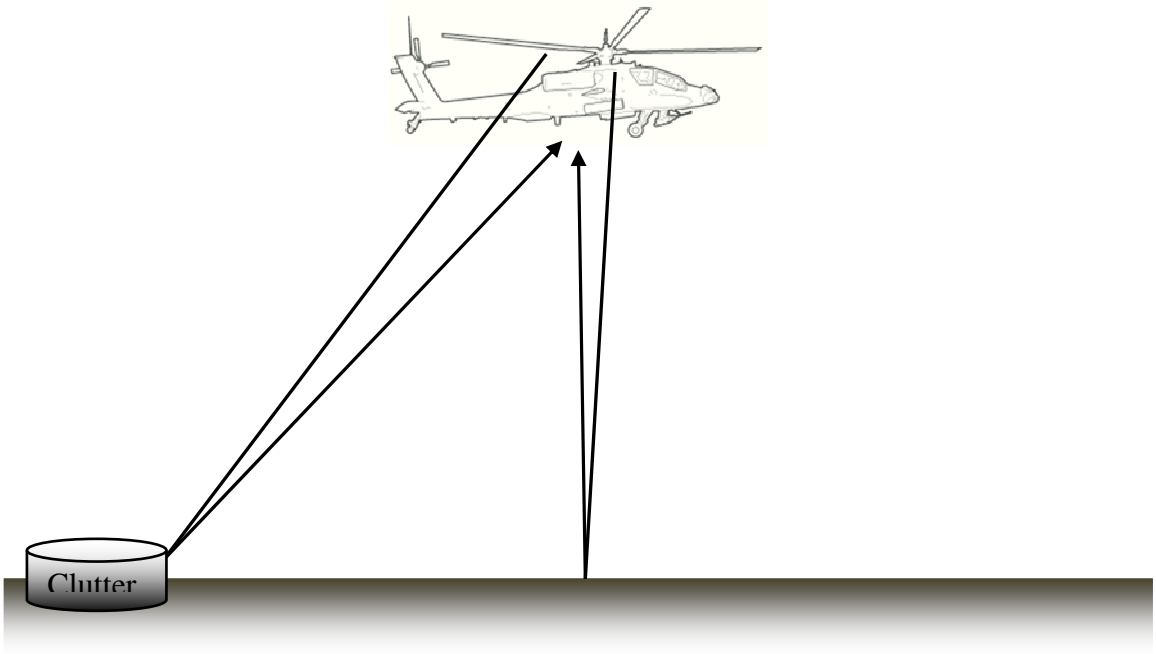

Figure 20 – Clutter can cause errors in height estimates

To prevent clutter reflections, the microphone array was designed to be directional. The array was designed with multiple microphones, with each microphone having a cardioid directivity pattern.

The array consisted of two medium density fiberboard panels, each with four microphones placed equidistant from the center of the circular panel (Figure 21). Medium density fiberboard was used for its strength and easy machining properties. The two panels were connected such that one faced the sound source and one faced the ground. The amplifiers and data acquisition equipment were located between the two panels. Eight microphones were used because this was the maximum that could be sampled with the data acquisition system available.

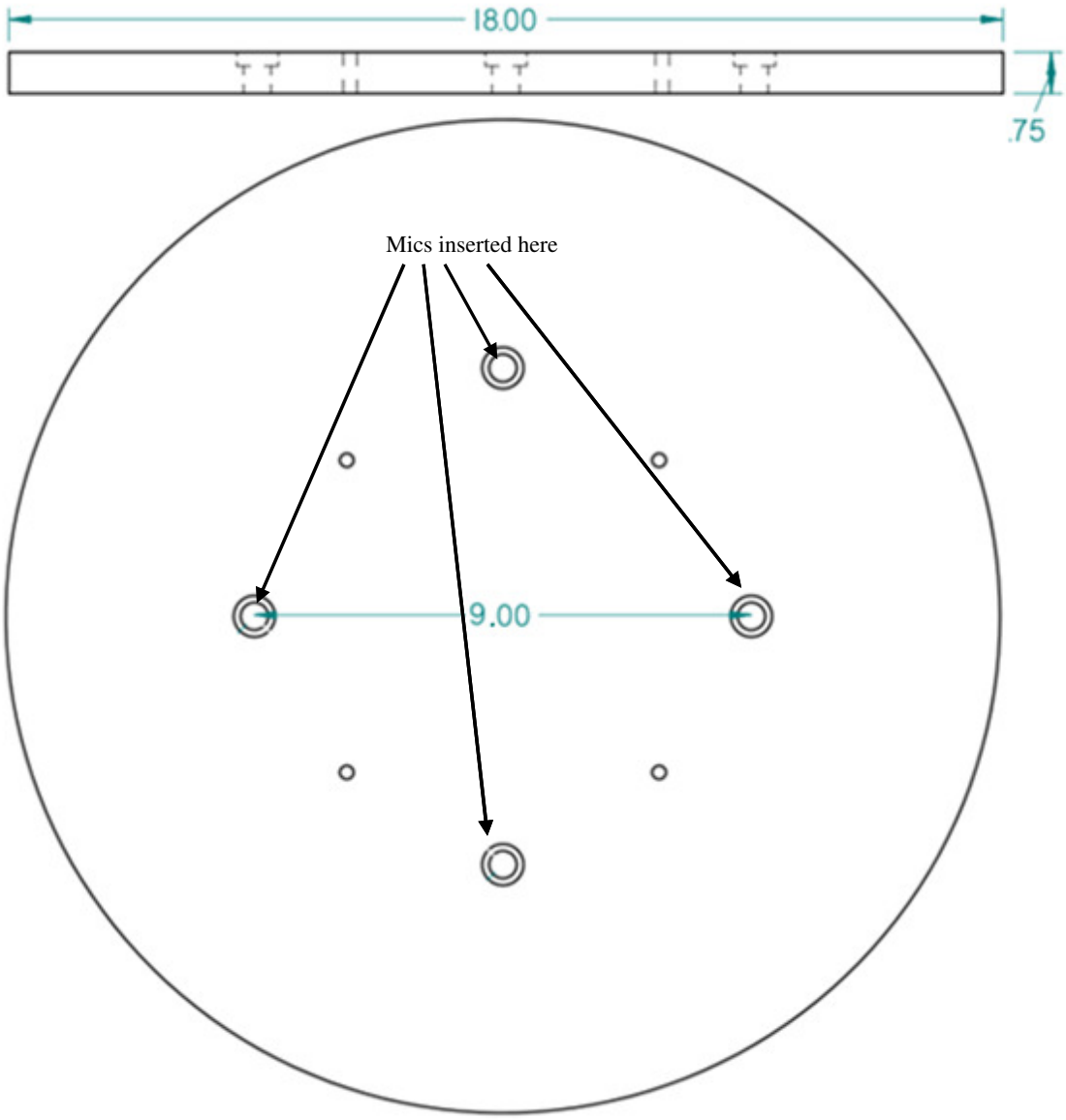

Figure 21 – A single panel; dimensions are in inches

Panasonic's unidirectional back electret condenser cartridge microphones (Panasonic WM-55A103) were chosen for their flat frequency response, low cost and durability. Each of the microphones was connected to its own amplifier and had its own channel on the data acquisition board. Each amplifier had a fixed gain of 35 dB and a built-in anti-aliasing filter. The filter was a fourth-order Butterworth filter with a cutoff frequency of 12,500 Hz. The filter was implemented using a modified Sallen-Key topology (Horowitz and Hill 1980). A full schematic is included in Appendix C.

The data acquisition board used was a Measurement Computing USB1608-FS with a maximum burst sampling rate of 25000 Hz on each of the eight channels. Figure 22 (left) shows a photograph of the data acquisition system and a photograph of the electret microphones (right). The microphones are 5 mm in height and 9.7 mm in diameter.

Figure 22 - Data Acquisition System (left) and Microphone used (right)

Figure 23 shows the results of a simulation of the directivity of a single panel of four cardioid microphones. Multiple microphones minimized the central beam width and side lobe strength. The diffraction grating lobes, which start around 3 kHz, were minimized by the directional nature of the microphone and the baffle.

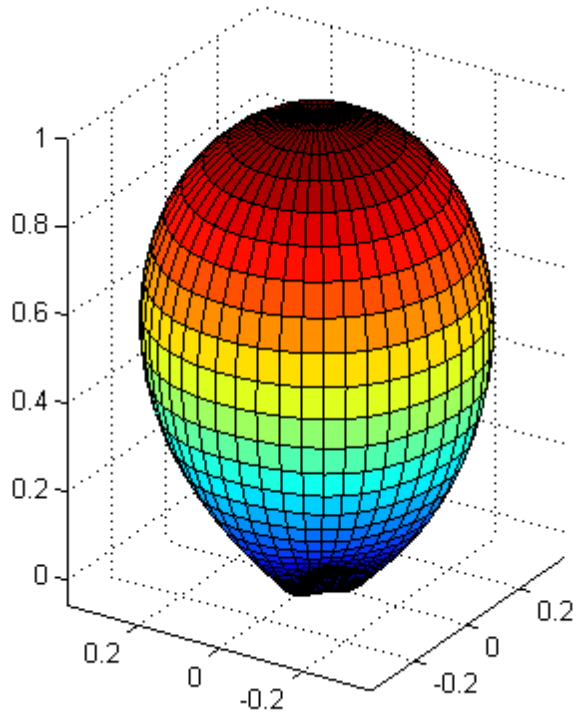

Figure 23 - Simulated directivity of array at a frequency of 1 kHz

Figure 24 and Figure 25 show the top and side views of the completed array.

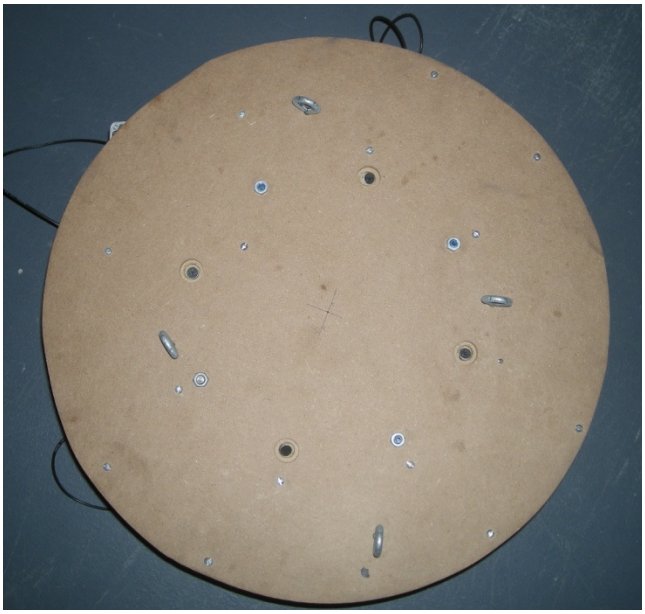

Figure 24 - Photo of array (top view)

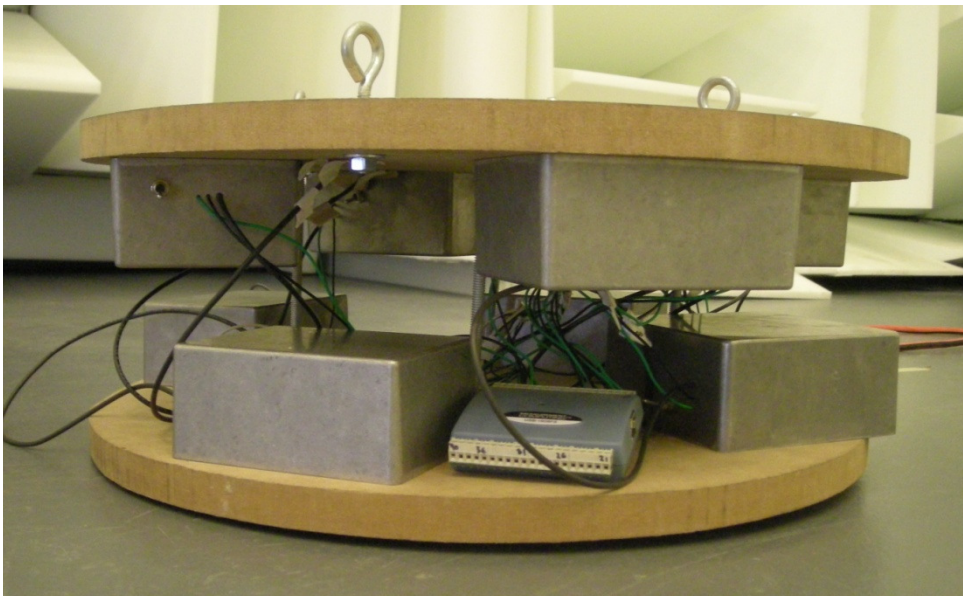

Figure 25 - Photo of array (side view)

Figure 26 illustrates the entire system, from sound generation to the data analysis. A Mackie HR824 studio speaker was used to generate white noise to approximate a flat spectrum.

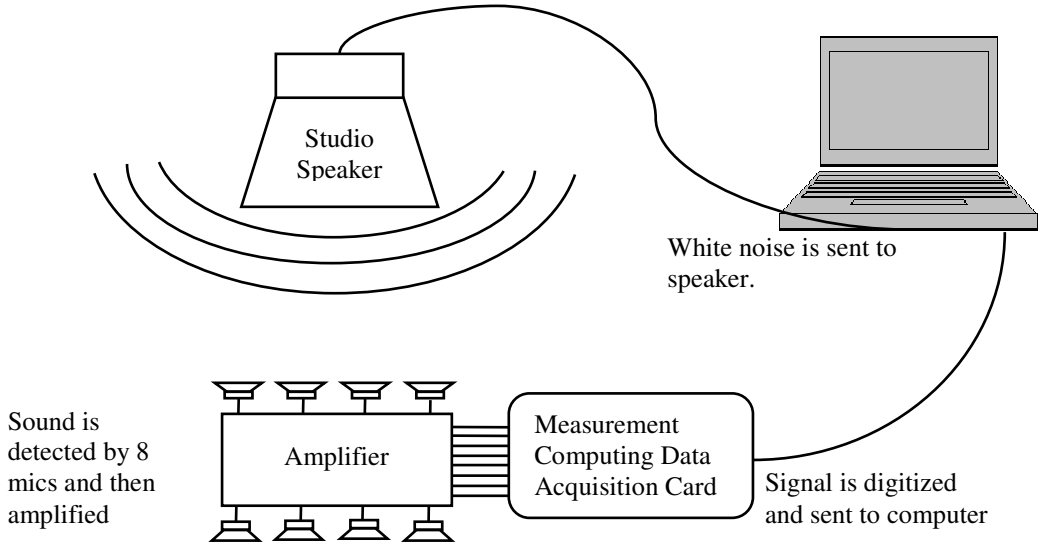

Figure 26 - Block diagram of experimental setup

The first tests were performed to find the properties of the array. The array's frequency response, its front-to-back sensitivity, and its directivity were tested inside the anechoic chamber using the Mackie speaker.

The anechoic chamber was used to characterize the array's frequency response and directivity. The Mackie HR-824 speaker was placed on one side of the anechoic chamber and the array on the other side. A white noise signal was sent out from the speaker and was recorded using the data acquisition system on the microphone array.

Figure 27 shows the results from the directivity tests of the microphone array. The blue and red dots refer to data taken when the front side of the array was pointed at the speaker, the blue dots show the signal received by the front of the array, and the red dots show the signal received by the back side. The green dots indicate data taken when the array's side was facing the speaker and the microphones were laid out in a diamond formation (as opposed to a square formation). The figure demonstrates that the amount of front-to-back leakage of the array remains small; there is an approximate 30 dB difference between the front and back of the array. It also shows that the side rejection worked well—approximately a 25 dB difference. The combination of the array design and microphone choice made the array very directional.

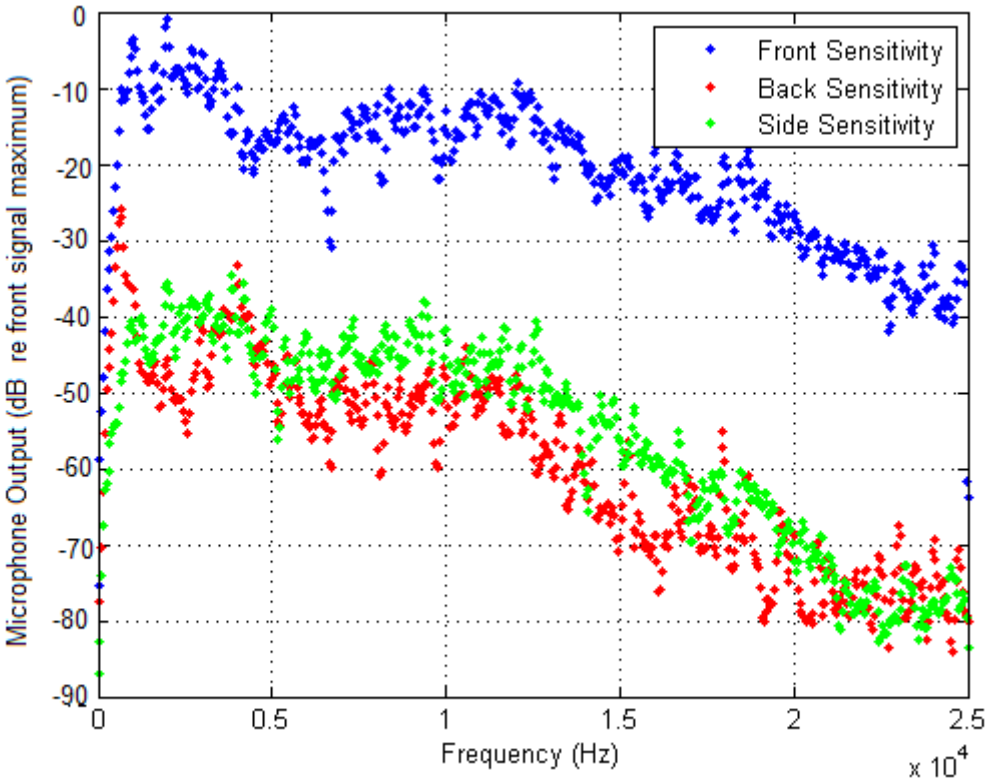

Figure 27 - Array's front to back sensitivity

The high front-to-back ratio is likely caused by two different effects: The first is that the cardioid microphones had a lower sensitivity on the back side than on the front side, and the second is that the baffle blocked the signal from reaching the microphones in the back.

4.2. Overview of Experimental Setup

After the microphone array setup had been tested, the hemianechoic chamber was set up to test the height detection methods. A speaker was suspended from the ceiling and the array suspended beneath the speaker. The height was varied by moving the speaker and array with a pulley system.

Figure 28 shows a schematic diagram of the setup in the hemianechoic chamber. Experiments were set up similarly outdoors, except the cables were replaced by a cherry

picker that could raise and lower the array. Sound was sent out of the speaker and detected by the array. The height detection was done as post-processing.

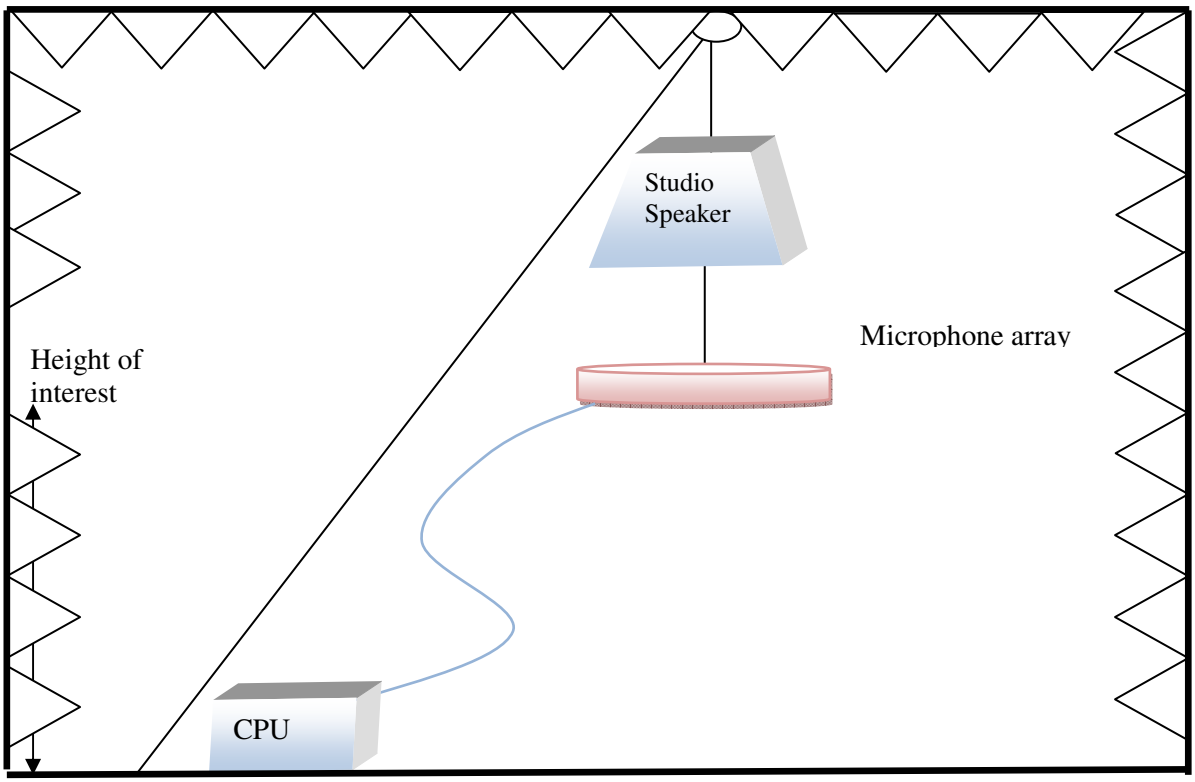

Figure 28 - Schematic of Experimental Setup in Hemianechoic Chamber

The speaker and array were heavy, and required pulleys to raise and lower the setup in a controlled fashion. Pulleys were connected to the ceiling hook directly above the array and the floor hook nearest to the array. A cable passed through these pulleys to a third hook on the opposite side of the chamber from the other floor hook. The array's height was changed incrementally by inserting a chain between the hook and the end of the cable. The true height above the floor was calculated by taking measurements with a tape measure. As with the directivity testing, a Mackie HR-824 studio speaker was used as the noise source in hemianechoic chambers.

4.3. Height Detection Experiments

4.3.1. Basic Height Detection

In the first set of trials, the speaker and the array were moved up and down to verify that the algorithm was effective. The distance between the speaker and the array was not held constant in these trials; the height of the array was the variable of interest.

Figure 29 shows a photograph of the microphone array suspended beneath the speaker in the hemianechoic chamber. The array was hung below the speaker. The speaker was hung suspended from the ceiling. By varying the lengths of the chain, the array could be raised and lowered, and the speaker could be raised and lowered.

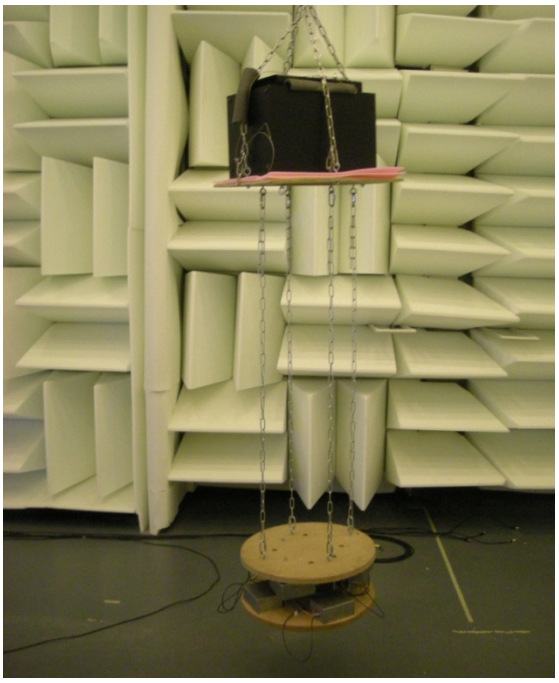

Figure 29 - Array hanging in hemianechoic chamber

Figure 30 shows the height measured with a tape measure plotted against the height measured using the microphone array.

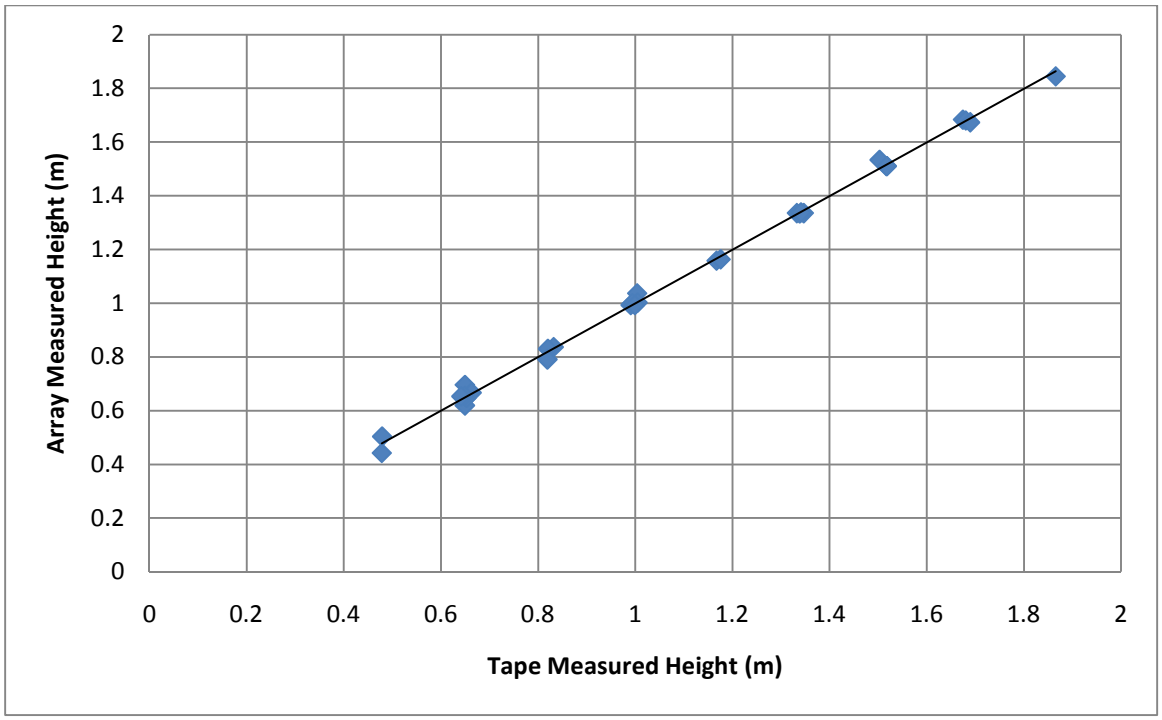

Figure 30 - Cepstrum calculated heights

The values are so close to the tape-measured values that a graph of the error is helpful to evaluate the results. Figure 31 shows the error in the height measurements plotted as a function of height. The error bounds were determined from the time/bandwidth limitations. Only 90% of the bandwidth was available because of the poor low frequency response of the electret condenser microphone. A typical frequency response of the electrets microphone is shown in Appendix C. The frequency response of the entire array is shown in Figure 27.

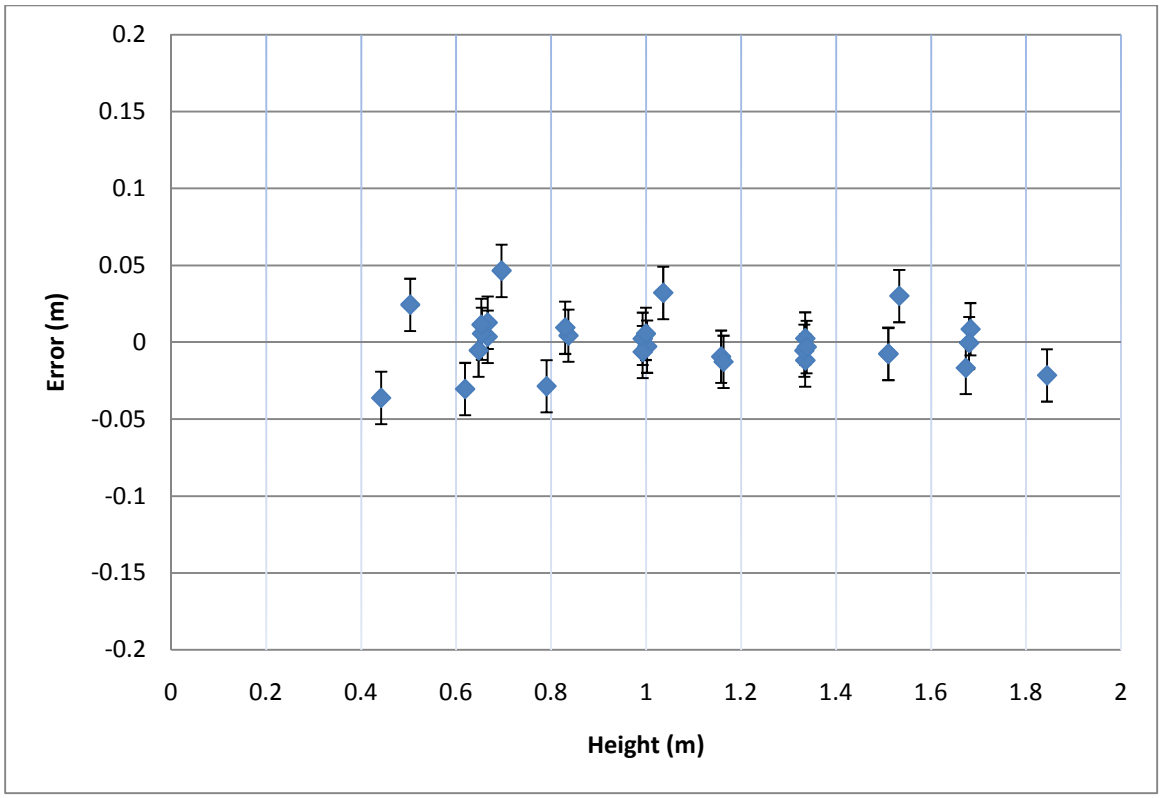

Figure 31 – Error in height data

The uncertainty was derived using the time bandwidth product as:

$$\text{Time} * \text{Bandwidth} = \Delta t * .9 * f_{nyq} = 1$$
$$\Delta h_{bw} = c\Delta t = 340/(.9 * 12500) \quad (49)$$
$$\Delta h_{bw} = \pm 1.4\ cm$$

The results of both Equation 49 and Figure 31s show that the cepstrum technique is a very powerful and accurate method. The error estimates do not include an uncertainty that accounts for variations in the speed of sound because sound speed measurements were not taken.

4.3.2. Optimal Distance between Array and Noise Source

In an actual helicopter brownout situation, the microphone array will be fixed to the helicopter, so that the noise source and array will have to move parallel to one another, unlike in the previous tests. There will most probably be an optimal distance at

which the error in height detection is minimized. In a helicopter, this ideal distance will be determined mainly by the geometry of the respective rotor, engine, and fuselage locations.

There are no rotors and fuselages in this experimental setup, therefore calculating the optimal distance becomes less complex. The problem is one-dimensional when the ideal distance between speaker and array will minimize error and false alarms. If the array is too close, a reflection can be detected between the array and the speaker itself. If the array is too far from the speaker, then the signal has already spread out and information has been lost. This optimal distance is relevant only to this experimental setup, and cannot be applied to an actual helicopter.

Figure 32 shows the results when the ideal distance was determined by minimizing the error. The array and speaker were moved separately to study each different array–speaker distance individually. Solid lines connect the measurements when the speaker was kept at a constant height and the array was moved.

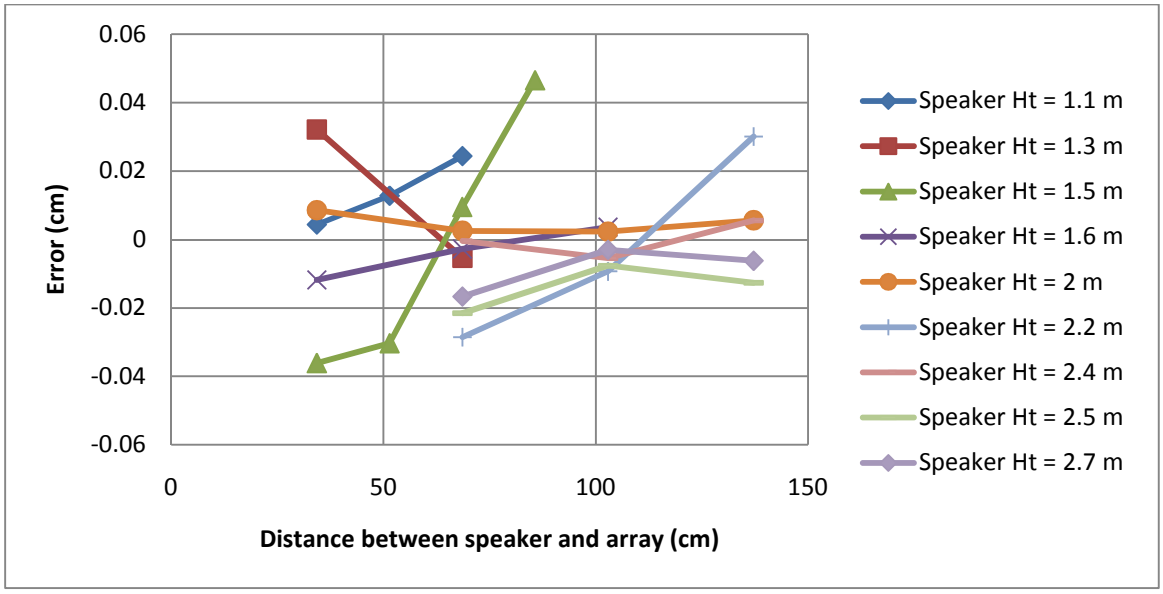

Figure 32 - Error when Speaker-to-Array distance varies at different heights

The results indicate that a distance of 1 m between the noise source and microphone array produced the least error. When the array was any closer to the speaker, the true height became obscured by reflections between the array and the speaker. When the distance was greater, the cepstrum spike height became smaller and more difficult to detect. This result (optimal distance around 1m) is specific only to this experimental setup. If this height detection technique were to be used aboard a helicopter, the geometry of the helicopter would determine the placement of the microphone array.

4.3.3. Outdoor Testing

Doing the experiments outdoors allowed the array to be tested at greater heights than in a hemianechoic chamber. It also simulated a more realistic environment because of a higher ambient noise level and a weaker ground reflection. The Georgia Tech facilities group graciously provided equipment for the experiments. They provided a cherry picker to suspend the speakers/array above the field and two cherry picker operators to help with the experiments.

The outdoor experiments required more equipment and more people to run each test. Two people operated the truck and crane, one person was responsible for triggering the data acquisition of the array, and another person recorded array heights and background noise levels during testing. All tests were done in the Burger Bowl field of Georgia Tech, which is a reasonably flat athletic field with a grass surface. The Georgia Tech Research Institute provided a gasoline-powered electric generator for testing.

Figure 33 shows the setup of the cherry picker and microphone array.

The data gathered in outdoor simulations was similar to the data gathered in the hemianechoic chamber. The crane attached to the lift moved the array to different heights. At each height, a measurement was taken using a tape measure.

Figure 33 - Outdoor experimental setup

A crane attached to the basket of the lift elevated and lowered the microphone array and speaker. The speaker and array were both connected to the same laptop computer as in Figure 28. At times, the truck's bed was visible in the cepstrum domain. These extra reflections are noted in Appendix F.

Figure 34 shows the results from these basic height tests. The height measurements in Figure 34 were accurate, but their accuracy decreased with height.

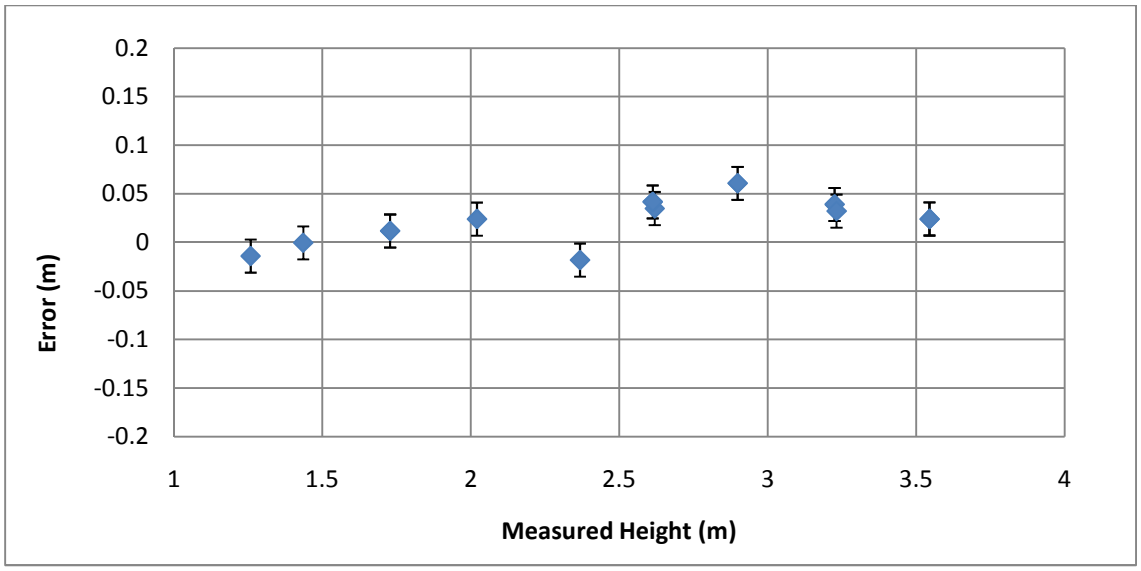

Figure 34 - Accuracy of outdoor height testing

Figure 35 shows the trend in spike height, which also decreased as the height increased. The spike height was measured in decibels, with the root mean square of the signal as the reference value.

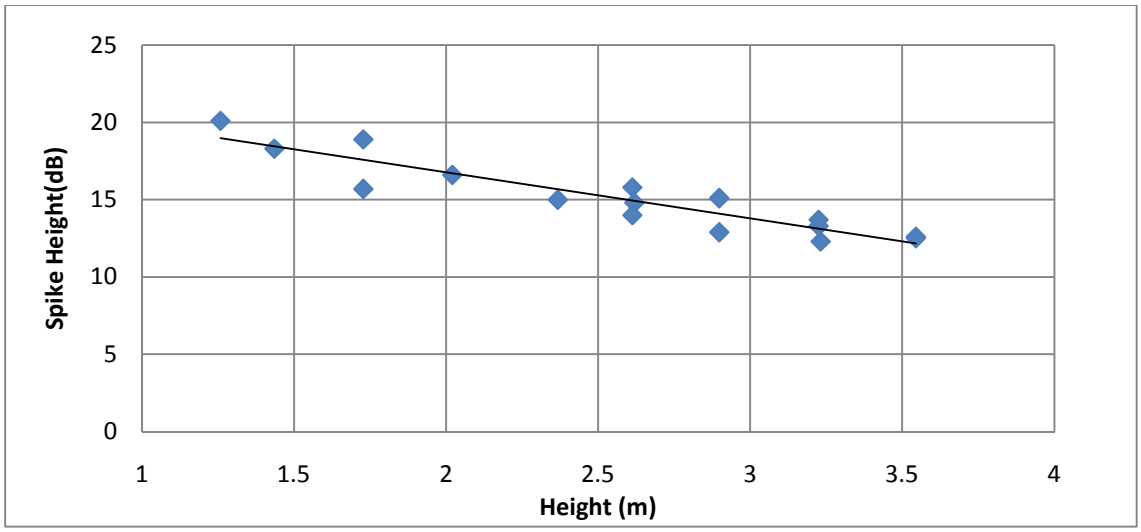

Figure 35 – Cepstrum spike of outdoor height testing

Signal strength also dropped off noticeably with as height increased. At heights beyond 3.6 meters, the SNR drops to a point where the relevant cepstrum spike is no larger than

its surrounding spikes caused by random noise. Because real helicopters are so much louder than the sources used here, this is not likely to be a problem in practice.

Figure 36 shows the raw cepstrum data from a trial where reflections from both the truck bed (1.177 m from the array) and the ground (2.197 m from the array) are visible.

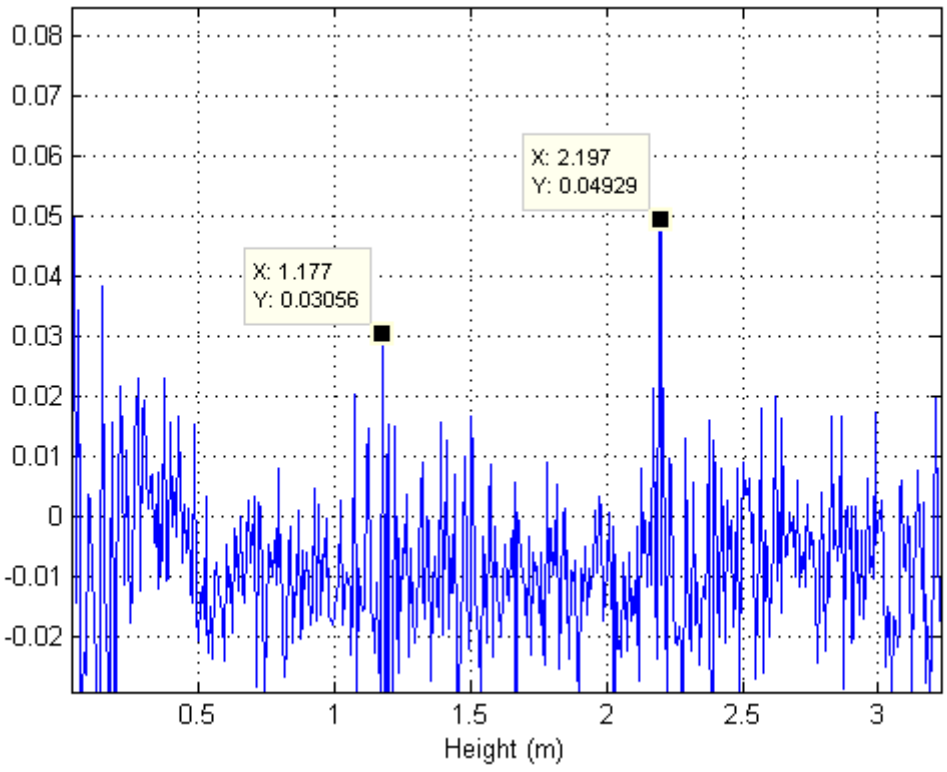

Figure 36 – Raw cepstrum data from a successful detection in outdoor height testing

Figure 37 shows cepstrum data after the spike is buried by the noise. The data was taken when the array was 3.7 m above ground level.

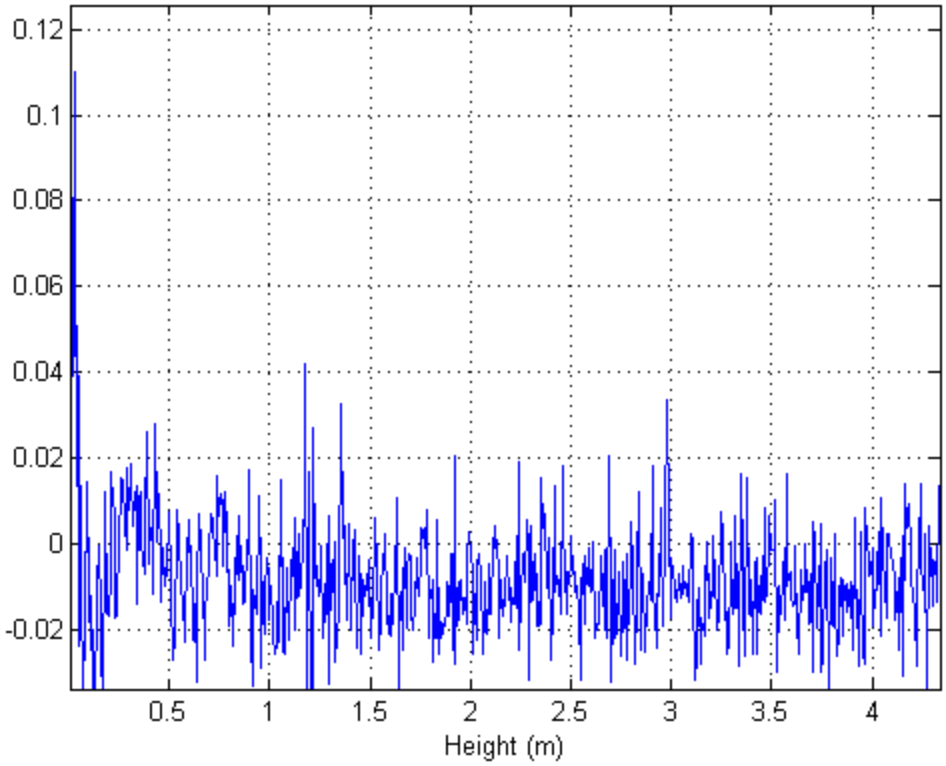

Figure 37 – Raw cepstrum data after height spike has fallen beneath noise threshold

4.4. Obstacle Detection Experiments

In an actual helicopter landing situation, complex terrain might require navigation around obstacles. Those obstacles should appear as smaller reflections in the cepstrum domain. To test the sensitivity of the cepstrum algorithm to obstacles in the landing area, the array was raised and lowered in the hemianechoic chamber in the presence of obstacles.

The main obstacle used (during indoor testing) was a flatbed jack. The bed of the jack provided a rigid surface large enough to reflect a substantial portion of the sound energy. The cepstrum algorithm under that condition was more sensitive than expected. It was very easy to measure the arrival time of extra echoes from sources other than the ground. The algorithm was actually so sensitive that it was sometimes difficult to

differentiate among echoes from the ground, the echo from the obstacle (the jack), and the scattered echoes from the support beneath the jack. The relevant spikes were successfully identified by raising and lowering the obstacle incrementally. The scattered echoes dispersed randomly, but the spikes resulting from the jack and the ground were clearly distinguishable because as the array moved upward, clear spikes appeared at heights corresponding to the distances to the ground and jack.

Figure 38 shows some of the heights detected as compared to the actual heights. The three different shapes indicate the three different heights that appeared as spikes during testing.

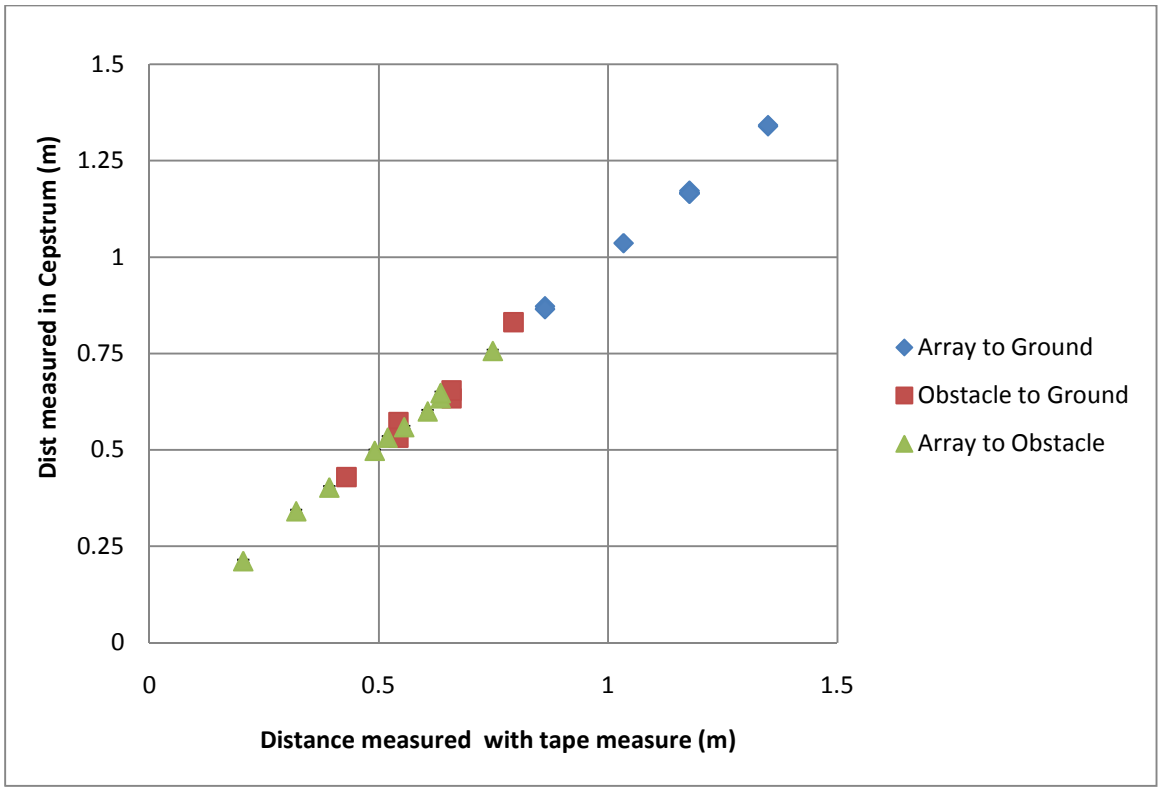

Figure 38 - Results from obstacle detection tests

Figure 39 shows the error between the measured value and the cepstrum-calculated value. The error bounds in Figure 39 were found using the time bandwidth techniques discussed previously. Most of the data error lies within the error bounds

imposed by the bandwidth limitations.

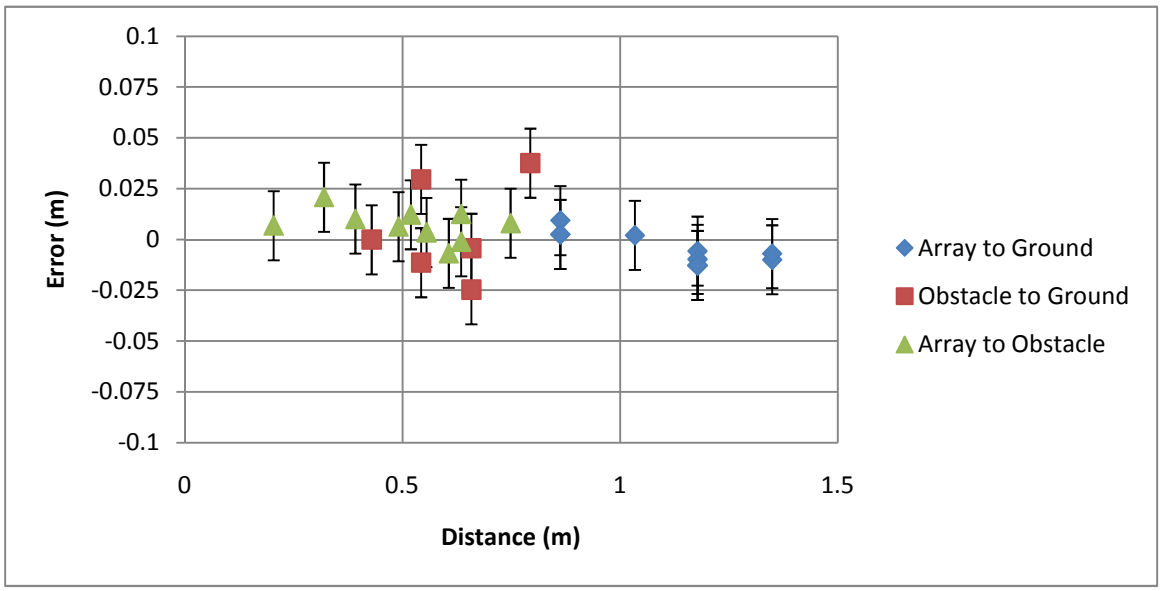

Figure 39 - Error from different cepstrum spikes

At times, there was an ambiguity in the source of a reflection. This would make it difficult to land in an environment where there are an unknown number of obstacles (e.g., multiple trucks, people, boulders). Work needs to be done to enable the system to differentiate among obstacles. This could be accomplished with a larger array of microphones having the ability to map the ground using beam-steering techniques. In this first generation prototype, the beam was too wide to steer in any particular direction.

4.5. Velocity Detection Experiments

Velocity testing was performed in two settings: the hemianechoic chambers where the indoor height testing was performed, and the Burger bowl field where the outdoor height testing was performed.

4.5.1. Indoor Velocity Testing

The indoor velocity tests required the array to be moved upwards at a constant velocity for a given amount of time. The operator started the array's data acquisition at

the point when the setup was in motion and stopped the data acquisition before the motion ceased. Both the height change and the time elapsed during the pull were recorded; therefore, the average velocity was simple to calculate ($v = \Delta h/\Delta t$).

The Doppler velocity detection algorithm did not yield accurate results, and the difference method was not able to recognize the height of the cepstrum while it was in motion. Two possible reasons for the failure are (1) insufficient bandwidth and (2) unsteady velocity. The microphones were not sensitive to frequencies below 100 Hz at the outset of the experiment. In addition, they deteriorated over the course of the experimental process and lost sensitivity above 9 kHz, probably due to rough handling of the array. With only 8.5 kHz of useable bandwidth remaining, the array was operating close to the threshold of acceptable bandwidth for a POD of 90% for the Doppler method.

Oversampling is one method that could prevent this problem in future experiments, because a faster sampling rate would give more bandwidth, thereby improving the POD. In all these trials, 0.15 sec of data is recorded. As an example, the signal only need be 0.012 s in duration if the helicopter is 2 m in altitude because that is the amount of time required for the signal to travel to the ground and back.

Table 1 shows a sample of the results of the indoor velocity measurements, demonstrating that the array did not reliably measure the velocity.

Table 1 - Indoor Velocity Measurements

Trial	Estimated Velocity	Actual Velocity
1	0.9 m/s	1.0 m/s
2	-	0.6
3	-	0.5
4	0.4	0.8
5	2.1	0.7

4.5.2. Outdoor Velocity Testing

For the velocity tests, a crane attached to a cherry picker pulled the array up and down at a predetermined velocity. The crane's motor contributed a large amount of extraneous noise to the experiment. Height measurements were taken at the beginning and end of the run to compare the actual velocity to the velocity predicted by the array.

No Doppler distortion was measured when the array was raised and lowered by the cherry picker. This was probably caused by the noise contribution of the motor moving the array. The SNR was -9 dB where the signal was the white noise being generated by the speaker and the noise was the sound created by the crane's motor.

CHAPTER 5

ATTENUATION, WIND SHEAR, & SURFACE ROUGHNESS

One final concern that must be addressed for the proposed solution to brownout is whether environmental factors (sands, swirling winds, terrain) can cause acoustical problems. The three most likely issues to arise are:

1. Sand from the brownout may attenuate the acoustic signals
2. Wind shear may distort the acoustic signal
3. Rough terrain may obscure clarity in the reflected signal.

The discussion in the following section explores of the mechanisms behind these issues Subsequently, calculations demonstrate that these are not problems which would affect the results of the height and velocity detection.

5.1. Acoustic Attenuation from Sand

Approaching the first issue, the amount that a signal will be attenuated as it propagates through a sandstorm needs to be determined. Attenuation through fluid media has been well formulated in terms of relaxation times, which refer to the amount of time it takes for a substance to adjust from a non-stable to a stable state. Short relaxation times mean that the substance is able to adapt quickly to change, whereas long relaxation times indicate that a substance is resistant to changes in state (Temkin 2005).

Attenuation coefficients can be derived from relaxation times, and transmission loss can then be derived from the attenuation coefficients. The transmission loss, which is the quantity of interest, determines how much of the signal is lost as it propagates through

a sand storm. The decrease in signal strength will make the height and velocity more difficult to determine.

The first step toward deriving the transmission loss and consequent change in signal strength is to understand the attenuation and relaxation processes taking place inside the brownout cloud. The brownout cloud can be classified as a dilute aerosol, having two main forms of attenuation: translational and thermal. Translational attenuation relates the back and forth motion of the particles to the motion of the fluid. The two motions are not always in perfect phase with one another, which can lead to fluid dragging on the aerosol particles. This drag leads to the loss of energy. This situation is described by the dynamic relaxation time (τ_d), which is defined as

$$\boldsymbol{\tau_d = \left(\frac{2\rho_p}{9\rho_f}\right)\frac{a^2}{\nu_f}} \tag{50}$$

In this equation, $\boldsymbol{a}$ refers to the radius of the particles in the suspension, and ρ_f and ρ_p refer to the densities of the fluid and the aerosol. The attenuation due to the translation of the particles with respect to the fluid is

$$\alpha_{trans} = \left(\frac{\eta}{2}\,\frac{\omega\tau_d}{1+\omega^2\tau_d^2}\right)\frac{\omega}{c} \tag{51}$$

Here η is the mass loading (mass of aerosol per unit volume of fluid) and τ_d is the dynamic relaxation time (Temkin 2005, p274).

The other major contributor to attenuation and dispersion in dilute aerosols is thermal attenuation. The thermal relaxation time is defined as

$$\boldsymbol{\tau_t = \left(\frac{2}{3}\,\frac{\rho_p c_{pp}}{\rho_f c_{pf}}\right)\frac{a^2}{2\kappa_f}} \tag{52}$$

where c_{pp} and c_{pf} refer to the specific heat capacities of the fluid and the particulate aerosol. The thermal attenuation coefficient follows a form similar to the translational attenuation coefficient:

$$\alpha_{therm} = \left(\frac{\eta}{2}\,\frac{\omega\tau_t}{1+\omega^2\tau_t^2}\,\frac{c_{pp}}{c_{pf}}\,(\gamma-1)\right)\frac{\omega}{c} \tag{53}$$

Here, τ_t refers to the thermal relaxation time, and c_{pp}/c_{pf} refers to the ratio of specific heats in the aerosol and the fluid (Temkin 2005 p250). In the brownout scenario, thermal attenuation dominates at lower frequencies and translational attenuation dominates at higher frequencies.

The next step to determine whether brownout affects the propagation of sound is to combine each different type of attenuation and then determine the signal transmission loss. The total attenuation coefficient α_{tot} can be determined by adding the coefficients together, $\alpha_{tot} = \alpha_{trans} + \alpha_{therm}$.

The transmission loss TL, can then be determined by calculating the drop in sound pressure level after the signal travels a given distance: $TL = 10 * \log_{10}\left(e^{-\alpha_{tot} * dist}\right)$.

Figure 40 shows the transmission loss for typical conditions in a high mass loading situation (something that could be occur in brownout conditions).

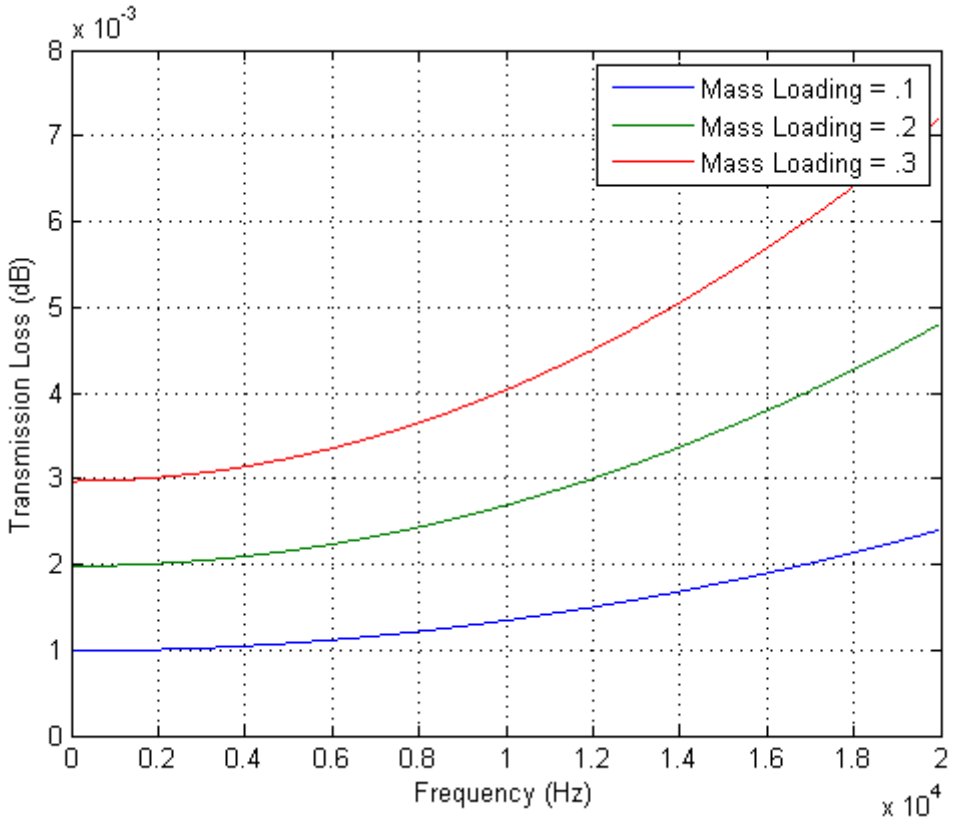

Figure 40 – Transmission loss from brownout when helicopter is at 5 m

Figure 40 shows that a sandstorm will give transmission losses that are on the order of .001 to .01 dB. They will not significantly affect the amplitude of the rotor noise. The constants used for this calculation were collected from (Temkin 2005) and (Rodgers 1968).

5.2. Wind Shear in Brownout

If the wind shear of the rotor wash is steep enough, refraction of the acoustic signal by the wind shear could negatively affect height and velocity detection. The following analysis makes a rough approximation to determine if wind shear will distort the signal enough to be problematic, or even noticeable.

To determine whether this refraction caused by wind shear will present problems in the calculation, a linear wind gradient model is used to describe the rotor wash of the helicopter and the acoustic effects of that gradient are analyzed.

The first assumption needed to describe the wind gradient is that the rotor blades must be in subsonic motion. As a consequence of this subsonic motion, the analysis can rely on linear acoustics and avoid nonlinear, high Mach number effects. This assumption is valid when the helicopter is hovering or landing, and not when it is traveling forward at high speeds.

With this assumption, the simplest possible wind gradient can be studied to determine its effects on the wind flow in brownout: the linear wind gradient. Assuming that the wind velocities drop off linearly with distance from the rotor blades, much can be learned about the problem.

For this discussion, let z represent the direction orthogonal to the rotor and represent the distance from the center of the rotor in the plane of the rotor. Place the origin is at the center of the rotor. As z decreases, the wind velocity decreases. In the radial direction, the wind increases as r increases until the end of the rotor is reached. After that, the wind drops off quickly.

An analytical description of the sound propagation in linear wind gradients can be derived from the wave equation for moving media (Pierce), where $\bar{v}$ corresponds to the velocity of the fluid and p corresponds to the pressure as a function of time

$$\nabla^2 p = \frac{1}{c^2}\left(\frac{\partial}{\partial t} + \bar{v}\cdot\nabla\right)^2 p \tag{54}$$

The result is that the ray from the helicopter blade will bend towards the slower wind speeds with a constant radius of curvature $r_{c.}$, where

$$r_c \approx c/\frac{dv}{dz} \tag{55}$$

Calculations reveal that sound propagation inside the brownout cloud will not cause inaccuracy because of the extremely small velocity gradient relative to the speed of sound. Using this linear gradient refraction equation for ray bending, it can be shown exactly how small an effect the wind shear will actually have. If the winds drop from 100 m/s at the rotor to 0 m/s on the ground over a distance of 5 m, dv/dz will be 20. The speed of sound is about 340 m/s, therefore the radius of curvature of a ray in this type of extreme gradient is 17 m. The curvature adds only 0.3% to the path length of the ray traveling to the ground and back; this amount is inconsequential. The same applies to the attenuation from the swirling sand. In both cases, the distance scales being studied are too small for these effects to affect the accuracy of the calculation.

5.3. Surface Roughness

Up to this point, all descriptions of the helicopter's height and velocity determination have been based on a model in which the noise reflects off of a smooth, flat surface. This assumption held true in the hemianechoic chamber testing, because any surface features in the floor were significantly smaller than the wavelengths detectable by the array. The assumption is less valid in the Burger Bowl setting, where clumps of grass and dirt are common. Despite these clumps of grass, the height was detected relatively easily, meaning that the algorithm is robust enough to tolerate some amount of terrain roughness.

However, the terrain could alter the signal by reducing the reflection amplitude. This could happen through several mechanisms, including sound transmission into the

ground and scattering of sound in other directions. These mechanisms were captured by a "reflection coefficient" in the ensuring simulation: The reflection coefficient represents the fraction of the incident signal that is sent directly back to the microphone array. Simulations were run using similar settings to the height detection tests in the Height Detection section. In these tests, the helicopter height was fixed at 4 meters from the ground.

Figure 41 shows the results of the simulation reflection coefficient simulation. It shows that as long as the reflected wave is at least 6% as strong as the incident wave, the cepstrum peak can be identified reliably.

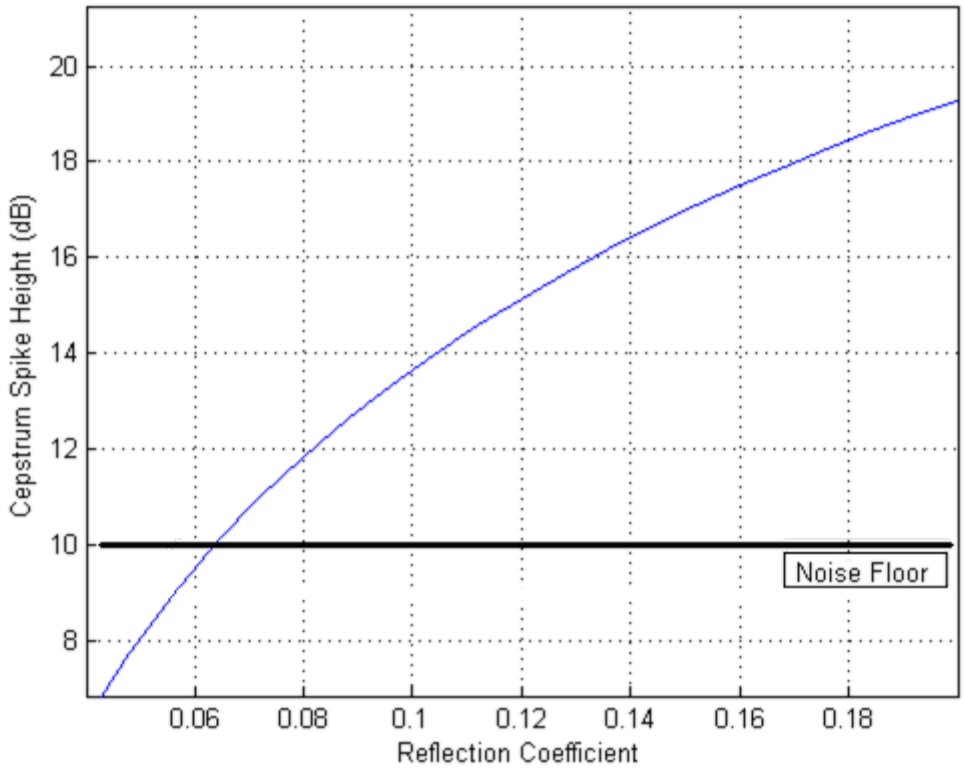

Figure 41 - Attenuation of reflected signal

Now, suppose there is a rough surface that reflects the signal back as a Gaussian pulse. For this simulation, the height is set at 4 m and the reflection coefficient is 1. All other settings are similar to those in the previous simulation.

Figure 42 shows how wide the pulse can be before the cepstrum algorithm becomes invalid. The width of the pulse is determined by the terrain's roughness. The pulse can be detected reliably as long as it is less than 30 cm wide.

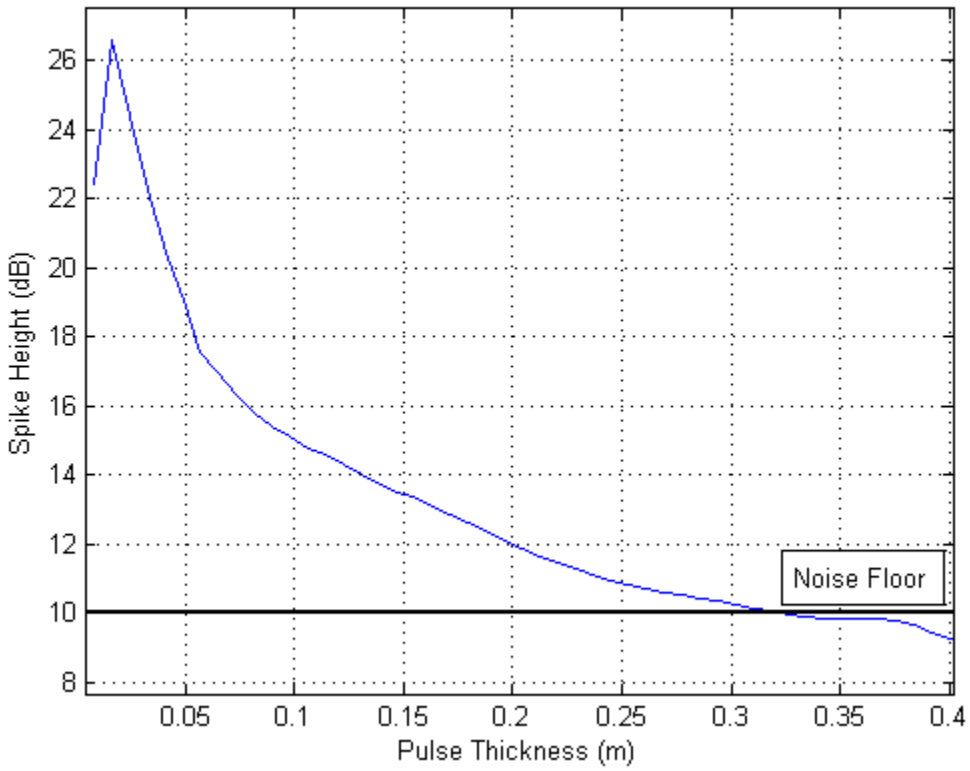

Figure 42 – Effects of Rough Terrain on Spike Height

Estimates for the roughness of homogenous terrain vary widely, but most estimates specify that terrain tends to have no more than 5 cm of roughness occurring regularly (Wierninga 1993). This is well within the minimum requirement for detection. Therefore, the roughness of the terrain is not likely to become problematic. Furthermore, if future iterations of the array increase the number of microphones, the beamwidth will shrink;

the smaller beamwidth will allow for less terrain variation within the array's beam. In the case of non-homogenous terrain, obstacles which reflect a pulse wider than 30 cm will not be detected if the beamwidth is this wide.

CHAPTER 6

CONCLUSIONS

The height and velocity detection techniques derived in this thesis are methods which take advantage of the large noise created by helicopters rather than trying to overcome the noise with an additional signal. Height is detected by measuring echo arrival time from the ground and velocity is detected by measuring the associated Doppler distortion. Algorithms to determine height have been published in the literature, but the Doppler velocity detection algorithm has not. This is an area where more research is necessary.

The simulations have shown that both the cepstrum and the Doppler velocity-detection algorithm are methods powerful enough to detect height and velocity accurately, but the simulations showed that an array with a broad frequency response is crucial. For the height detection, at least 1 kHz of bandwidth is required, and for the velocity detection, at least 8 kHz.

An array was constructed to test the results of the simulations and theory. The array accurately measured heights up to 3.5 meters and was able to detect obstacles effectively. The velocity detection experiments were not as successful as the height detection experiments, probably because of the limited bandwidth available in the present experimental setup.

The successes and failures in the development of the first prototype of this acoustic altimeter show that it is a promising idea, but requires further work. For instance, work is needed to make the velocity detection algorithm more robust and to incorporate more existing techniques that take advantage of the periodic and impulsive

nature of helicopter noise. Further, the array needs to be scaled up in size to allow for surface mapping and beam steering.

APPENDIX A – SCALE TRANSFORMS

The scale transform is mentioned conceptually in the body of the work, but it is not defined and explained in detail. This section is written to inform the more interested reader. The scale transform is defined as

$$F(\Omega) = \frac{1}{\sqrt{2\pi}} \int_0^\infty f(t)\, t^{-i\Omega - 1/2}\, dt \tag{56}$$

Its inverse is

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^\infty F(\Omega)\, t^{i\Omega - 1/2}\, d\Omega \tag{57}$$

where $f(t)$ is a function in time, and $F(\Omega)$ is the transformed function of scale, and t and Ω are the independent variables in the time and scale domain, respectively. The $1/\sqrt{2\pi}$ factor is added in order to conserve energy between the scale and time domains (Cohen 1993). The scale transform is useful in this project because it can extract Doppler distortion similar to the way that a Fourier transform can extract time delays in the frequency domain.

APPENDIX B – HELICOPTER NOISE

To determine a helicopter's height and velocity using the noise from its rotors, an understanding of the spectral content created by the helicopter is paramount. The noise from a helicopter in takeoff and landing can be broken down into the component s of the physical mechanism causing noise. Air displacement leads to thickness and high-speed impulsive noise. The forces that the rotor applies to the air create loading and force noise. Blade vortex interaction is caused by the interaction of the blade and the preceding blade's vortex. Each of these sources of noise has a unique directivity pattern and frequency characteristics (Schmitz 1995).

The dominant noise source of a hovering rotorcraft comes from displacement of air from the rotating blade, known as thickness noise. As the blade moves toward the observer, the front of the blade creates a high pressure zone and the rear of the blade creates a low pressure zone. This creates a pressure disturbance that propagates outward, but primarily in the plane of the rotor disk. This signal is impulsive and highly directional. As the helicopter's rotor reaches higher Mach numbers, thickness noise is also referred to as high-speed impulsive noise.

Loading noise is the second major contributor to rotor noise. This noise, also known as force noise, can be resolved into three components: steady force noise, unsteady force noise, and broadband noise. Each part has a unique generating mechanism.

Steady force noise refers to the noise created by forces on the rotor disk (such as lift, drag, and thrust) that the blade encounters periodically with each revolution. These forces can be modeled as if they were dipoles strewn across a non-rotating radiating disk.

Unsteady force noise is caused by the variation in loading that a rotor blade experiences in each revolution. The primary cause of variation in rotor disk loading arises from the change in blade pitch as the blade rotates. The difference in force applied by the blade generates an unsteady (but periodic, because it happens with each revolution) acoustic disturbance (Schmitz 1995). These types of variations are unsteady in that they are not constant forces on the blade as it moves through its entire revolution. However, they do have a periodic nature because they are encountered with each full turn of the rotor disk.

Unsteady force noise is caused by unstable and irregular aerodynamic events that occur when turbulences and vortices are pulled through the rotor disk; these events occur randomly. A typical "event" might consist of a turbulent pocket of air being pulled through rotor blades. All these different types of loading/force noise have spectral content over a large amount of the audible frequency range and have a downward directivity (Johnson 1980).

Blade vortex interaction is another important source of noise. The interaction of the tips of the helicopter blades with the vortices created by the previous passage of rotor blades can be so loud as to dominate all other types of noise. Blade vortex interaction becomes most noticeable during descending flight and non-axial rotor disk rotation. Each time the helicopter blade strikes a vortex, the rotor blade forms a sudden high frequency impulse that radiates away from the helicopter.

Engine noise is also a significant contributor to noise, although its frequency content directivity characteristics vary significantly from helicopter to helicopter. Table 2 summarizes the different types of helicopter noise.

Table 2 - Summary of Helicopter Noise Sources

Noise Type	Cause	Sound Content	Directivity
Thickness (High Speed Impulsive at high Mach Number)	Displacement of air as blades rotate	Impulsive	In plane of rotor disk
Steady Force (aka Loading)	Downward force that blades apply	Low frequency	Downward
Unsteady Force	Variable downward force that blades apply	Mid to high frequency	Downward
Blade Vortex Interaction (BVI)	Blades interacting with their own wake	Impulsive	45 degrees downward
Engine noise	Mechanical noise created by the engine	Mid frequency	All directions

Figure 43 summarizes the preceding paragraphs and table in terms of a graphic describing the different types of noise and their directivity.

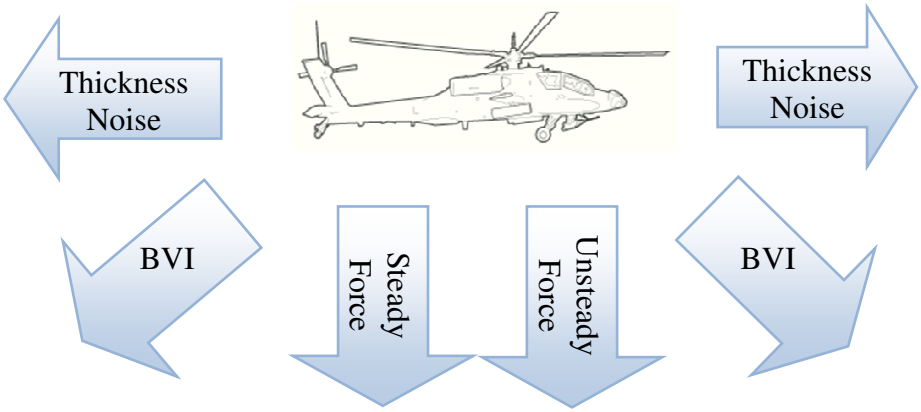

Figure 43 - Illustration of noise directivity

Figure 44 shows the noise spectrum from a Seanight helicopter. Helicopter spectra vary widely in shape and magnitude, but the Seanight spectrum does demonstrate that a helicopter spectrum can be very broadband.

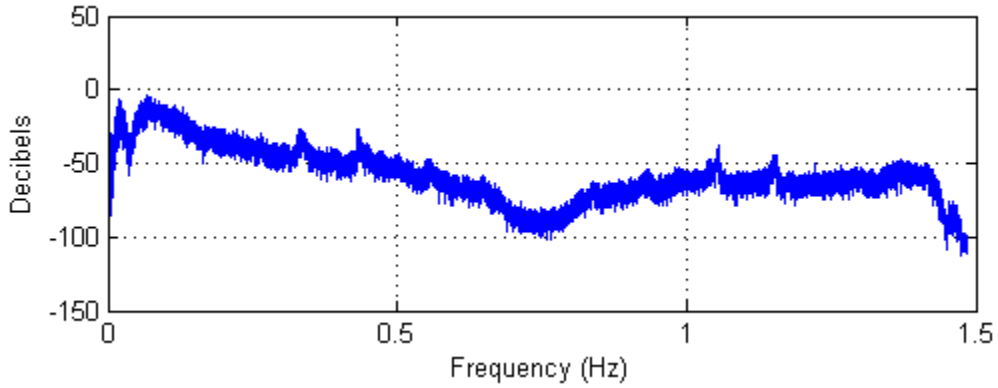

Figure 44 - Helicopter Spectrum of a Seanight Helicopter

Using Different Types of Helicopter Noise

The knowledge of the different types of helicopter noise can aid in simultaneous detection of height and velocity. In previous studies, when the height and velocity were calculated, simulations and experiments used white noise free from any type of impulse or periodicity arising from the engine noise or blade passage. This was done based on the assumption that the array would be placed directly beneath the helicopter where the blade passage noise is nearly zero (Schmitz 1995 p97).

This is not necessarily the case. If the array is placed closer to the cockpit, then some amount of impulsive noise will be present. This impulsive noise will probably improve the cepstrum's sensitivity to motion, because the presence of impulses will set up a spike train in the frequency domain.

These spikes can aid in the process of velocity detection. Instead of assuming that the spectrum is a single broadband problem, the Doppler distortions of individual frequencies can be analyzed. Well-developed algorithms, such as matched filtering, can be used determine the velocity.

APPENDIX C – AMPLIFIER CIRCUITS

The amplifier circuits used in the microphone array were fourth-order Butterworth antialiasing filters implemented using a modified Sallen and Key topology. Figure 45 shows the design of the filters. The first op-amp is for the amplification stage. The next two are the low-pass filters. The final stage is an impedance converter.

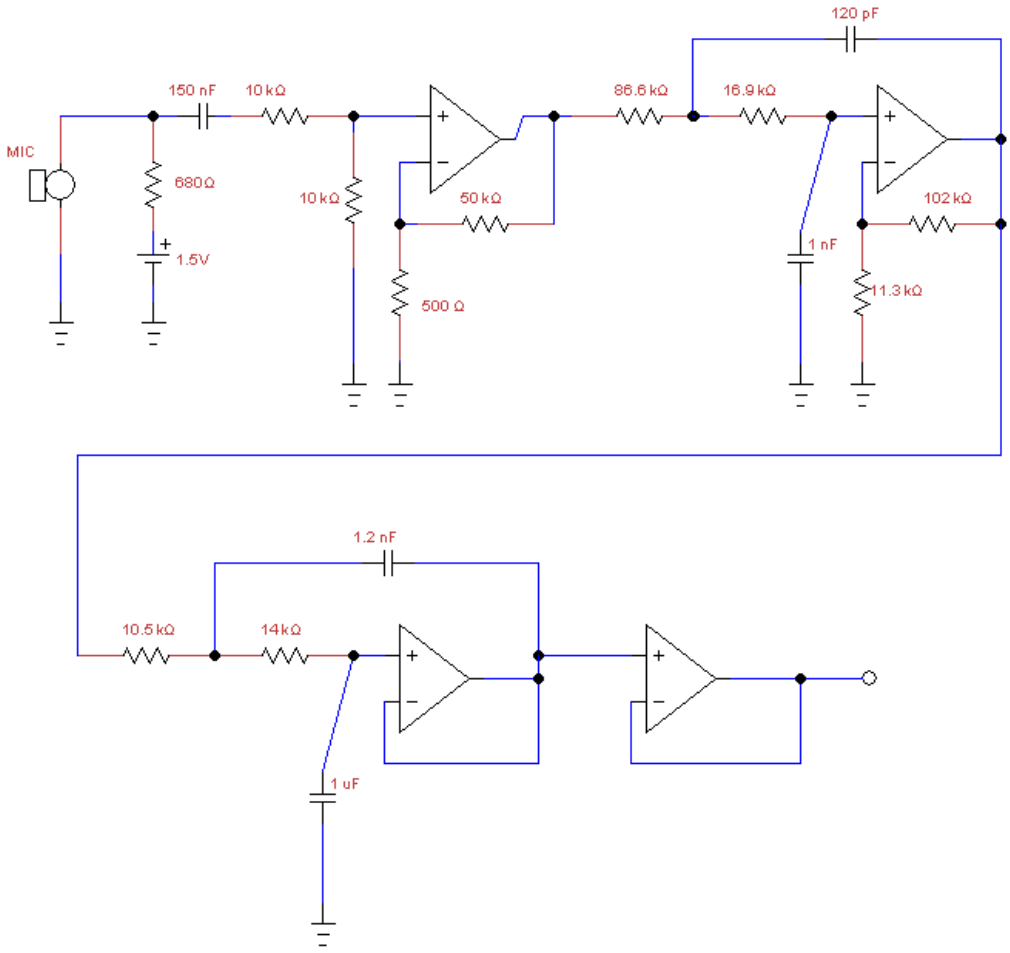

Figure 45 - Amplifier Circuit

Figure 46 shows the layout of the amplifier's power source.

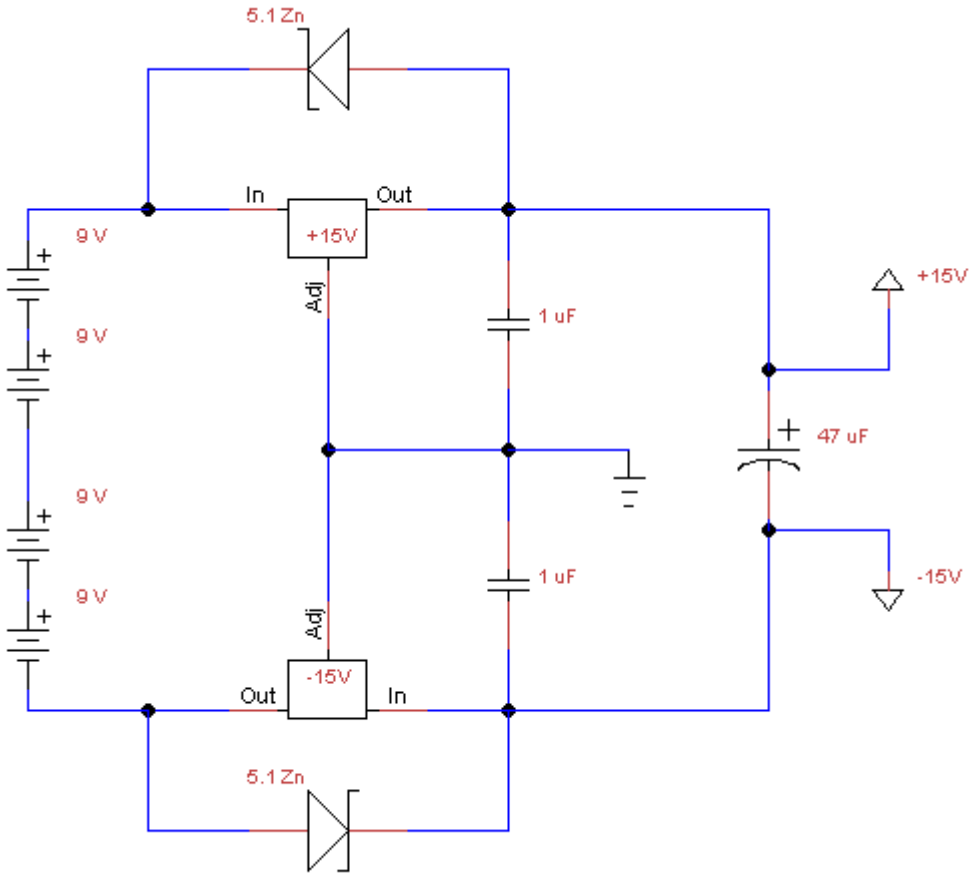

Figure 46 – Power supply of microphone and amplifier

The following figure shows the frequency response of the front and back of the electrets microphone used in this thesis. The chart is given in units relative to the microphone's frontal response at 1 kHz. The chart was given on Panasonic's data sheet for the WM-55A103.

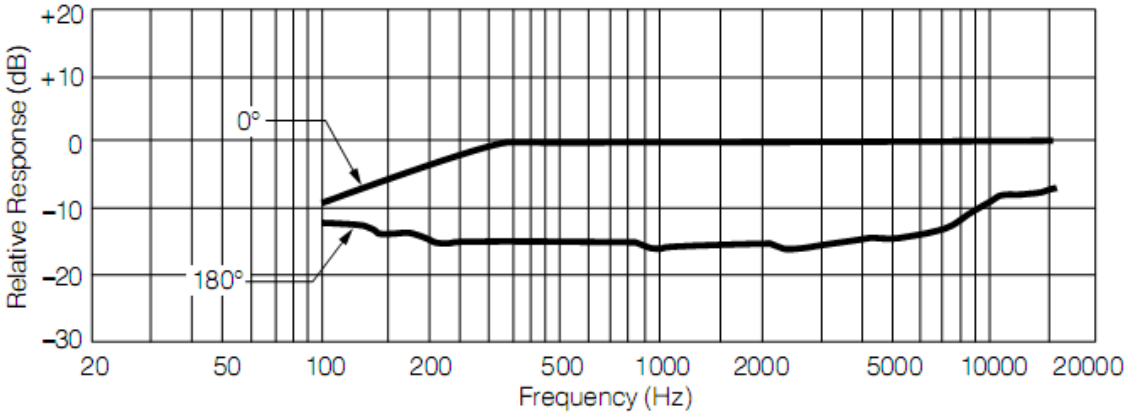

Figure 47 - Front and Back Frequency Response of WM-55A103

APPENDIX D – MATLAB CODE

Note that all code is written for MATLAB 7.4. The Fast Mellin Transform Toolbox v1.2, developed by Sena and Rochesso, was used extensively in velocity detection functions.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function basic_cepstrum
% Displays a basic cepstrum of white noise and returns SNR.

delay = 150;
fs = 25000;
len = .15*fs;
c = 340;

r = randn(len,1);
x = [r; randn(delay,1)];
y = [randn(delay,1); r];
signal = x+y;

subplot(2,2,1);
plot((1:length(signal))/fs, signal);
title('Time Domain')
xlabel('Time (s)');
ylabel('Amplitude');

subplot(2,2,2);
spectr = abs(fft(signal));
absc = linspace(0, fs/2, length(spectr));
plot(absc, spectr);
title('Frequency Domain')
xlabel('Frequency (Hz)');
ylabel('Magnitude');

subplot(2,2,3:4);
time = (1:length(signal))/(fs);
ceps = abs(ifft(log(abs(fft(signal)))));
plot(time(2:floor(end/3)), ceps(2:floor(end/3)));
title('Cepstrum Domain');
xlabel('Echo arrival time (s)');
ylabel('Gamnitude');
```

```
function cepstrum_add_whitenoise
% Check the effects of adding random white noise to the cepstrum
algorithm.
% Plots the SNR

delay = 150;
fs = 25000;
len = .15*fs;
c = 340;
r = randn(len,1);
x = [r; randn(delay,1)];
y = [randn(delay,1); r];
rms_signal = sqrt(mean((x+y).^2));

for ii = 1:300
    noise = ii/100*rms_signal*randn(size(x));
    rms_noise(ii) = sqrt(mean(noise.^2));
    signal = x+y+noise;
    snr_time(ii) = 20*log10(rms_signal/rms_noise(ii));
    ceps = abs(ifft(log(abs(fft(signal)))));
    pk(ii) = ceps(delay+1);
    rms_ceps(ii) = sqrt(mean(ceps(2:end).^2));
    snr_ceps(ii) = 20*log10(pk(ii)/rms_ceps(ii));
end
 plot(snr_time, snr_ceps, '.'); grid on;
title('Signal to Noise Ratio');
xlabel('S/N Ratio of the Time Domain (dB)');
ylabel('S/N Ratio of the Cepstrum Domain');
```

```
function cepstrum_dopplershift
% Displays the results of the cepstrum after a Doppler distortion.
% Just displays a basic cepstrum of white noise and returns SNR.
delay = 150;
fs = 25000;
len = .15*fs;
c = 340;
% hit = zeros(50,1);
pkval = zeros(50,1);
pkloc = zeros(50,1);
stretch_factor = zeros(50,1);
snr_ceps = zeros(50,1);
for jj = 1:100
r = randn(len,1);
for ii = 1:50
    x = [r; randn(delay,1)];
    y = [randn(delay,1); r];
    rms_signal = sqrt(mean((x+y).^2));
    y_stretched = resample(y, ii+19999, 20000);
    y_stretched = y_stretched(1:length(y));
    signal = x+y_stretched;

    ceps = ifft(reallog(abs(fft(signal))));
    [pkval(ii), pkloc(ii)] = max(ceps(101:200));
    rms_ceps = sqrt(mean(ceps(2:end-1).^2));
    snr_ceps(ii) = (pkval(ii)/rms_ceps) + snr_ceps(ii);
    stretch_factor(ii) = (ii+20000)/20000;
end
hit = hit + ((pkloc>=51)&(pkloc<=55));
disp([jj])
end
v = (stretch_factor-1)*c*2;
plot(v, hit/max(hit), 'o');
xlabel('Helicopter Velocity (m/s)');
ylabel('Probability of Detection');
```

```
function cepstrum_different_bw_regions
% Prints charts showing how SNR changes as bandwidth changes.
clear all;
delay = 150;
fs = 25000;
len = .15*fs;
J = 100; % Number of iterations
I = 200;% Resolution

hit_lp = zeros(I,1);
hit_bp = zeros(I,1);
hit_hp = zeros(I,1);
spike_height_lp = zeros(I,J);
spike_height_bp = zeros(I,J);
spike_height_hp = zeros(I,J);
%% LP Filter - Both reflection and incident
for jj = 1:J
    fprintf('LP Filter, Iteration %g\n',jj);
    r = randn(len,1);
    x = [r; randn(delay,1)];
    y = [randn(delay,1); r];
    z = x+y;
    rms_signal = sqrt(mean(z.^2));
    for ii = 1:I
        freq_limits_lp(ii) = (ii+9)/210;
        [b, a] = butter(3, freq_limits_lp(ii), 'low');
        signal = filter(b, a, x + y);
        ceps = abs(ifft(log(abs(fft(signal)))));
        [pkval, pkloc] = max(ceps(100:200));
        rms_ceps = sqrt(mean(ceps(2:end).^2));
        snr_lp(ii) = 20*log10(pkval/rms_ceps);
        hit_lp(ii) = hit_lp(ii) + ((pkloc>=50)&&(pkloc<=52));
    end
end
%% BP Filter - Both reflection and incident
for jj = 1:J
    fprintf('BP Filter, Iteration %g\n',jj);
    r = randn(len,1);
    x = [r; randn(delay,1)];
    y = [randn(delay,1); r];
    z = x+y;
    rms_signal = sqrt(mean(z.^2));
    for ii = 1:(I-10)
        freq_limits_bp(ii, :) = [ii/(2*I+10), (2*I-ii)/(2*I)];
        bandwidth_bp = freq_limits_bp(:,2) - freq_limits_bp(:,1);
        [b, a] = butter(3, freq_limits_bp(ii,:));
        signal = filter(b, a, x + y);
        ceps = abs(ifft(log(abs(fft(signal)))));
        [pkval, pkloc] = max(ceps(100:200));
        rms_ceps = sqrt(mean(ceps(2:end).^2));
        snr_bp(ii) = 20*log10(pkval/rms_ceps);
        hit_bp(ii) = hit_bp(ii) + ((pkloc>=50)&&(pkloc<=52));
    end
end

%% High Pass Filter - Both reflection and incident
```

```
for jj = 1:J
    fprintf('HP Filter, Iteration %g\n',jj);
    r = randn(len,1);
    x = [r; randn(delay,1)];
    y = [randn(delay,1); r];
    z = x+y;
    rms_signal = sqrt(mean(z.^2));
    for ii = 1:I
        freq_limits_hp(ii) = (ii+2)/210;
        [b, a] = butter(3, freq_limits_hp(ii), 'high');
        signal = filter(b, a, x + y);
        ceps = abs(ifft(log(abs(fft(signal)))));
        [pkval, pkloc] = max(ceps(100:200));
        rms_ceps = sqrt(mean(ceps(2:end).^2));
        snr_hp(ii) = 20*log10(pkval/rms_ceps);
        hit_hp(ii) = hit_hp(ii) + ((pkloc>=50)&&(pkloc<=52));
    end
end

%% Plot
plot(freq_limits_lp*fs/2, snr_lp, bandwidth_bp*fs/2, snr_bp, (1-
freq_limits_hp)*fs/2, snr_hp);
legend('Low Frequencies', 'Middle Frequencies', 'High Frequencies',
'location', 'NW');
xlabel('Incident and Reflected Signal Bandwidth, (Hz)');
ylabel('S/N in Cepstrum Domain (dB)');
grid on;
function cepstrum_helicopter_descending
% Multiple Echoes
% Illustrate effects of multiple echoes as helicopter descends.
fs = 25000;
c = 340;
signal = randn(5000,1);
x = (1:3750)*c/fs;

delay1 = floor(9*fs/c);
obst1 = floor(8*fs/c);
signal1a = signal(1:3750);
signal1b = signal(1+delay1:3750+delay1);
signal1c = signal(1+obst1:3750+obst1);
signal1 = signal1a + signal1b + signal1c;
subplot(3,1,1);
plot(x, real(ifft(log(abs(fft(signal1))))));
axis([.1 10 -.025 .25])
xlabel('Height (m)');
ylabel('Gamnitude');

delay2 = floor(7*fs/c);
obst2 = floor(6*fs/c);
signal2a = signal(1:3750);
signal2b = signal(1+delay2:3750+delay2);
signal2c = signal(1+obst2:3750+obst2);
signal2 = signal2a + signal2b + signal2c;
subplot(3,1,2);
signal2(1) = 0;
plot(x, real(ifft(log(abs(fft(signal2))))));
```

```
xlabel('Height (m)');
axis([.1 10 -.025 .25])
ylabel('Gamnitude');
delay3 = floor(5*fs/c);
obst3 = floor(4*fs/c);
signal3a = signal(1:3750);
signal3b = signal(1+delay3:3750+delay3);
signal3c = signal(1+obst3:3750+obst3);
signal3 = signal3a + signal3b + signal3c;
subplot(3,1,3);
signal3(1) = 0;
plot(x, real(ifft(log(abs(fft(signal3))))));
xlabel('Height (m)');
axis([.1 10 -.025 .25]);
ylabel('Gamnitude');
```

```
function cepstrum_spike_changes_with_bw
% See what filtering does to spike in cepstrum domain
r = randn(len,1);
fcr = ([1 .8 .6 .4 .2]);
for ii = 1:5
    for jj = 1:200
        x = [r; randn(delay,1)];
        y = [randn(delay,1); r];
        if fcr(ii) ~=1
            [b, a] = butter(4, fcr(ii), 'low');
            signal = filter(b, a, x + y);
        else
        signal = x + y;
        end
        ceps(:,jj) = abs(ifft(log(abs(fft(signal)))));
    end
    subplot(1, 5, ii);
    plot(mean(ceps,2), '.-');
    title(['BW = ', num2str(fcr(ii))]);
    axis([143 157 0 .4]); grid off;
end
```

```
function fmf_basic
% Check accuracy and S/N of FMF algorithm.  This is called Doppler
% velocity detection method in body of the work.
% clear all
path(path, './MellinCode');
tic;
snr = zeros(100,1);
pkval = zeros(100,1);
pkloc = zeros(100,1);
for ii = 1:100
    x0 = randn(3500,1);
    y0 = resample(x0,5000+ii,5000);
    N = min([length(x0), length(y0)]);
    x1 = x0(1:N);
    y1 = y0(1:N);
    x2 = [x1; randn(250,1)];
    y2 = [randn(250,1); y1];
    x3 = abs(fft(x2));
    y3 = abs(fft(y2));
    signal = ifft(reallog(abs(FMT(x3 + y3))));
    signal(1:floor(3+(ii/10))) = 0;
    [pkval(ii), pkloc(ii)] = max(signal(1:ceil(end/10)));
    noise = sqrt(mean(signal(1:ceil(end/10)).^2));
    snr(ii) = 20*log10(pkval(ii)/noise);
    fprintf('Iteration # %g\n',ii);
end
v = 2*340*(5:100)/5000;
subplot(2,1,1);
plot(v, pkloc(5:100), '.-');
title('Location of Doppler Spike');
xlabel('Doppler Shift (m/s)');
subplot(2,1,2);
plot(v, (pkloc-p(2))/p(1) - v, '.-'); hold on;
title('Error'); grid on;
xlabel('Simulated Velocity (m/s)');
ylabel('Amount of error (m/s)');
```

```
function fmf_add_whitenoise
% Calculate robustness of FMF algorithm to noise

path(path, './MellinCode');
tic;
snr_time = zeros(300,1);
pkval = zeros(300,1);
pkloc = zeros(300,1);
hit = zeros(300,1);

for jj = 1:500;

x0 = randn(3500,1);

for ii = 1:300
    y0 = resample(x0,5000+100,5000);
    N = min([length(x0), length(y0)]);
    x1 = x0(1:N);
    y1 = y0(1:N);
    rms_signal = sqrt(mean((x1 + y1).^2));
    noise = ii/200*rms_signal*randn(length(x0)+250,1);
    rms_noise = sqrt(mean(noise.^2));

    x2 = [x1; randn(250,1)]+noise;
    y2 = [randn(250,1); y1]+noise;
    x3 = abs(fft(x2));
    y3 = abs(fft(y2));
    signal = ifft(reallog(abs(FMT(x3 + y3))));
    [pkval(ii), pkloc(ii)] = max(signal(30:70));
    snr_time(ii) = 20*log10(rms_signal/rms_noise);
end
hit = hit + ((pkloc>=12)&(pkloc<=14));

disp(jj)
end

plot(snr_time, hit/max(hit),'.');
xlabel('Signal to Noise Ratio in the Time Domain (dB)');
ylabel('Probability of Detection');
axis([-5 20 0 1])
toc
```

```
function fmf_leakage
% Determine the effects of front to back leakage for the FMF algorithm.

path(path, './MellinCode');
snr = zeros(100,1);
pkval = zeros(100,1);
pkloc = zeros(100,1);
miss = zeros(100,1);
hit = zeros(100,1);

for jj = 1:1000
fprintf('Trial #%g\n',jj);

for ii = 1:100
    x0 = randn(3500,1);
    y0 = resample(x0,5000+100,5000);
    N = min([length(x0), length(y0)]);
    x1 = x0(1:N);
    y1 = y0(1:N);
    x2 = [x1; randn(250,1)];
    y2 = [randn(250,1); y1];
    x2_leaked = x2 + ii/100*y2;
    y2_leaked = y2 + ii/100*x2;
    x3 = abs(fft(x2_leaked));
    y3 = abs(fft(y2_leaked));
    signal = ifft(reallog(abs(FMT(x3 + y3))));
    signal(1:floor(3+(1/10))) = 0;
    [pkval(ii), pkloc(ii)] = max(signal(35:45));
    rms_noise = sqrt(mean(signal(1:ceil(end/10)).^2));
    snr(ii) = 20*log10(pkval(ii)/rms_noise);
end

for ii = 1:100
    if pkloc(ii)>=7 && pkloc(ii)<=9
        hit(ii) = hit(ii) + 1;
    else
        miss(ii) = miss(ii) + 1;
    end
end

end

fraction_leaked = (1:100)/100;
plot(fraction_leaked, hit/max(hit));
xlabel('Fraction of Signal Leaked to Opposite Side');
ylabel('Probability of Detection');
axis([0 1 0 1])
```

```
function fmf_limitfreq
% Limit the frequency content and check its effects on the FMF S/N

close all;
clc;
disp('Begin Mellin Bandwidth Limitations');

path(path, './MellinCode');
snr = zeros(149,1);
pkval = zeros(149,1);
pkloc = zeros(149,1);
hit = zeros(149,1);

for jj = 1:100
    fprintf('Trial #%g\n',jj);
    x0 = randn(3500,1);
    y0 = resample(x0,5000+50,5000);
    N = min([length(x0), length(y0)]);
    for ii = 1:149
        freq_limits = (150-ii)/150;
        [b,a] = butter(3, freq_limits);
        x1 = filter(b, a, x0(1:N));
        y1 = filter(b,a, y0(1:N));
        x2 = [x1; randn(250,1)];
        y2 = [randn(250,1); y1];
        x3 = abs(fft(x2));
        y3 = abs(fft(y2));
        signal = ifft(reallog(abs(FMT(x3 + y3))));
        signal(1:11) = 0;
        [pkval(ii), pkloc(ii)] = max(signal(1:40));
        rms_noise = sqrt(mean(signal(12:ceil(end/10)).^2));
        snr(ii) = 20*log10(pkval(ii)/rms_noise);
        freq_limit(ii) = freq_limits;
    end
    hit = hit + (pkloc>=21 & pkloc<=23);
end

plot(freq_limit*12500, hit/max(hit));
xlabel('Bandwidth Remaining');
ylabel('Probability of Detection');
save 'mellin_limit_bandwidth_1000trials';
```

```
function front_to_back_postprocessing
% Compare the frequency sensitivity of the back of the array to the
% frequency sensitivty of the front of the array when in the fully
% anechoic chamber.  Used to post-process data taken on 5/2/08.

clear all;
load('frontside workspace.mat');
sideways = open('sideways workspace.mat');

for ii = 1:8
    x_nodc(:,ii) = x(:,ii) - mean(x(:,ii));
    X_nodc(:,ii) = 10*log10(pwelch(x_nodc(:,ii)));
    sideway_x(:,ii) = sideways.x(:,ii) - mean(sideways.x(:,ii));
    X_sideway_nodc(:,ii) = 10*log10(pwelch(sideway_x(:,ii)));
end

X_front = mean(X_nodc(:,1:4),2);
X_back = mean(X_nodc(:,5:8),2);
X_side = mean(X_sideway_nodc(:,1:8),2);

X_front1 = zeros(size(X_front));
X_front1(1) = X_front(1);
X_front1(end) = X_front(end);
X_front1(end-1) = X_front(end-1);

X_back1 = zeros(size(X_back));
X_back1(1) = X_back(1);
X_back1(end-1) = X_back(end-1);

X_side1 = zeros(size(X_side));
X_side1(1) = X_side(1);
X_side1(end) = X_side(end);
X_side1(end-1) = X_side(end-1);

% Smooth out frequency response so it is easier to read
for ii = 2:(length(X_nodc)-2)
    X_front1(ii) = X_front(ii-1)/2 + X_front(ii) + X_front(ii+1)/2 +
X_front(ii+2)/3;
    X_back1(ii) = X_back(ii-1)/2 + X_back(ii) + X_back(ii+1)/2 +
X_back(ii+2)/3;
    X_side1(ii) = X_side(ii-1)/2 + X_side(ii) + X_side(ii+1)/2 +
X_side(ii+2)/3;
end

freq = linspace(0, 25000, length(X_front));
plot(freq, X_front1, 'b.', freq, X_back1, 'r.', freq, X_side1, 'g.');
legend('Front Sensitivity', 'Back Sensitivity', 'Side Sensitivity');
title('Difference in sensitivity between front and back of microphone
array');
xlabel('Frequency (Hz)');
ylabel('Microphone Output (dB)'); grid on;
```

```
function typical_data_acquisition(speakerlinks, arraylinks, pausetime)
% Calculate the height of the entire array, with one trial and all
% microphones.

% DAq settings:
fs = 25000;
n = fs*.15;
ai = analoginput('mcc',1);
chans = addchannel(ai, [0:7]);
set(ai, 'SampleRate', fs)
set(ai, 'SamplesPerTrigger', n)
beep; pause(pausetime);
% Acquire data
start(ai);
x = getdata(ai); % Each channel is in a separate column.

% Analyze data
figure(1); % Waveforms
for m = 1:8
    subplot(2, 4, m)
    x1(:, m) = x(:,m) - mean(x(:,m));
    time = (0:(length(x1)-1))/fs;
    plot(time, x1(:, m));
    title(['Channel',num2str(m)]);
end
saveas(gcf,  [num2str(speakerlinks), '_speakerlinks,',
num2str(arraylinks), '_arraylinks,waveform']);

figure(3); % Height detection
x1(:, 5:8) = 2*x1(:,5:8);
x3 = sum(x1, 2);
x4 = real(ifft(reallog(abs(fft(x3)))));
h = linspace(0,500,500)/fs*340*.5;
plot(h, x4(1:500));
title([num2str(speakerlinks), ' speakerlinks,', num2str(arraylinks), '
arraylinks,height'])
saveas(gcf,  [num2str(speakerlinks), '_speakerlinks,',
num2str(arraylinks), '_arraylinks,height']);

save([num2str(speakerlinks), '_speakerlinks,', num2str(arraylinks),
'_arraylinks,workspace.mat']);
```

```
function sand_attenuation
% Evaluate how much attenuation the sand from the brownout causes using
% two phase relaxation model.

nu_m1 = .1; %Mass loading 1
nu_m2 = .2; %Mass loading 2
nu_m3 = .3; %Mass loading 3
a = 2e-4;    % Radius of Particle
gamma = 1.4; % Heat capacity ratio of fluid (cp/cv)
cpf = 1000; %Heat capacity of fluid (air)
cpp = .7; %Heat capacity of particle (silica)
f = linspace(1,1000); % Frequency
w = 2*pi*f; % Angular Frequency
c = 340;    % Speed of sound
k = .024; % Thermal conductivity (air)
rho_f = 1.2; % Density of air
rho_p = 1550; % Density of sand
vf = 1.78e-5;   % Viscosity of air

td = 2*rho_p*a^2/(9*rho_f*vf);
tt = 2*rho_p*cpp/(3*rho_f*cpf)*a^2/(2*k);


a_th = .5*nu_m1*(gamma-1)*(cpp/cpf)*w*tt./(1+w.^2*tt^2).*w/c;
a_tr = .5*nu_m1*w*td./(1+w.^2*td^2).*w/c;
a1 = a_th + a_tr;
plot(f, a1);
hold on;

a_th = .5*nu_m2*(gamma-1)*(cpp/cpf)*w*tt./(1+w.^2*tt^2).*w/c;
a_tr = .5*nu_m2*w*td./(1+w.^2*td^2).*w/c;
a2 = a_th + a_tr;
plot(f, a2, 'r');

a_th = .5*nu_m3*(gamma-1)*(cpp/cpf)*w*tt./(1+w.^2*tt^2).*w/c;
a_tr = .5*nu_m3*w*td./(1+w.^2*td^2).*w/c;
a3 = a_th + a_tr;

plot(f, a3, 'g');
xlabel('Frequency (Hz)')
ylabel('Attenuation Coeff (\alpha)');
title('p(x, t) = p_0e^-^\alpha^xf(t)')
legend('Mass Loading = .1','Mass Loading = .2','Mass Loading = .3')
grid on;
```

```
function array_directivity
% Calculate array directivity

f = 1000;
d = 4.5/sqrt(2)*.0254; % Distance microphones are from center
h = 4*.0254/2; % Height from half way to array
k = 2*pi*f/340;
a = .01;
% Location of mics (x,y,z). Suppose array axis is on z axis.
loc_mic1 = [d*cos(0), d*sin(0), h];
loc_mic2 = [d*cos(pi/2), d*sin(pi/2), h];
loc_mic3 = [d*cos(pi), d*sin(pi), h];
loc_mic4 = [d*cos(3*pi/2), d*sin(3*pi/2), h];
% Sum up and attenuate for distance to find total directivity
r = 100;
phi = linspace(0, pi, 50);
theta = linspace(0, 2*pi, 50);
p = zeros(length(phi));

for ii = 1:length(phi)
    for jj = 1:length(theta)
        loc_obs = [r*sin(theta(jj))*sin(phi(ii)), ...
            r*cos(theta(jj))*sin(phi(ii)), ...
            r*cos(phi(ii))];
        dist1 = norm(loc_obs - loc_mic1);
        dist2 = norm(loc_obs - loc_mic2);
        dist3 = norm(loc_obs - loc_mic3);
        dist4 = norm(loc_obs - loc_mic4);
        s1 = 1 + cos(phi(ii)); s2 = s1; s3 = s1; s4 = s1;
        s = (2*besselj(1,k*a*sin(phi(ii)))./(k*a*sin(phi(ii)))).^2;
        p_cardioid(ii, jj) = s1*exp(i*k*dist1) + s2*exp(i*k*dist2) +...
            s3*exp(i*k*dist3) + s4*exp(i*k*dist4);
        p_baffle(ii,jj) = s*exp(i*k*dist1) + s*exp(i*k*dist2) +...
            s*exp(i*k*dist3) + s*exp(i*k*dist4);
    end
end

[THETA, PHI] = meshgrid(theta, phi);
RHO_cardioid = abs(p_cardioid)/max(max(abs(p_cardioid)));
X_cardioid = RHO_cardioid.*sin(PHI).*sin(THETA);
Y_cardioid = RHO_cardioid.*sin(PHI).*cos(THETA);
Z_cardioid = RHO_cardioid.*cos(PHI);

RHO_baffle = abs(p_baffle)/max(max(abs(p_baffle)));
X_baffle = RHO_baffle.*sin(PHI).*sin(THETA);
Y_baffle = RHO_baffle.*sin(PHI).*cos(THETA);
Z_baffle = RHO_baffle.*cos(PHI);

surf(X_cardioid,Y_cardioid,Z_cardioid);
title(['Array''s directivity @ f=', num2str(f), ' Hz']); axis equal;
figure;
subplot(2,1,2);
surf(X_baffle,Y_baffle,Z_baffle); axis equal;
```

APPENDIX E – MICROPHONES

Based on the design needs of the microphone array for this project, unidirectional electret microphones were selected for their directivity, flat frequency response, and durability. The following is a brief review of the inner workings of an electret microphone and the basic notion behind cardioid directivity and its implications.

Condenser Microphone

A condenser microphone measures pressure by turning the microphone diaphragm into one plate of a capacitor. As the diaphragm oscillates, changes in the electrical signal correspond to changes in acoustic pressure at the diaphragm. In most condenser microphones, the capacitor must be charged externally. In some microphones, the capacitor is charged with an electret. An electret is a permanently charged electrical material named after a magnet because of their similarities. Using electrets is a tradeoff: These microphones can be sturdier and less expensive, but some fidelity is lost in the process.

Cardioid Directivity

In many applications, a unidirectional rather than an omnidirectional microphone may be desirable. One common way to achieve a unidirectional microphone is to use a combination of two microphones, one with a monopole directivity pattern and one with a dipole directivity pattern. The directivity of the combination is the sum of the two:

$$D(\theta) = D_{mono} + D_{dip} \tag{58}$$

A monopole has a constant directivity in all directions, and a dipole's directivity varies as a cosine. When these components are evaluated, the directivity pattern for a cardioid is

$$D(\theta) = 1 - \cos(\theta) \tag{59}$$

Figure 48 shows the directivity pattern derived from Equation (59) (Weisstein 2008).

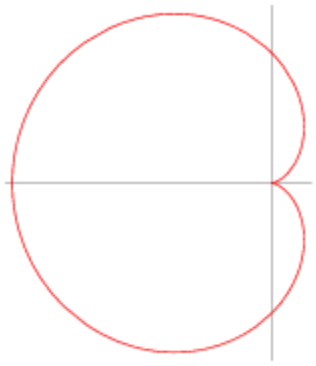

Figure 48 - Cardioid plot

The cardioid pattern can also be achieved by inserting vents in the microphone capsule, which add a delayed version of the signal because they are away from the central diaphragm. The array built in this thesis used vented microphones.

APPENDIX F – RESULTS FROM EXPERIMENT

"Speaker Links" refers to number of chain links between the steel cable (which is connected only to the speaker) and the floor hook. "Array link" refers to the number of chain links between the array and the speaker platform. Each chain link was approximately 1.3" in length.

Table 3 - Results from Indoor Experiments

Speaker Link/Array Link	Measured Height (m)	Cepstrum Predicted Height (m)	Error (cm)
30/10	0.832048438	0.8364	-0.43516
30/15	0.653653125	0.6664	-1.27469
30/20	0.478829688	0.5032	-2.43703
35/10	1.003895313	1.036	3.210469
35/20	0.652660938	0.6473	-0.53609
40/10	0.47814933	0.442	3.614933
40/15	0.649145759	0.6188	3.034576
40/20	0.820142188	0.8296	-0.94578
40/25	0.649145759	0.6956	-4.64542
45/10	1.346795313	1.335	-1.17953
45/20	1.004802455	1.002	-0.28025
45/30	0.662809598	0.6664	0.35904
55/10	1.674472321	1.683	0.852768
55/20	1.332479464	1.335	0.252054
55/30	0.990486607	0.9928	0.231339
55/40	0.64849375	0.6541	0.560625
60/20	0.818951563	0.7904	-2.85516
60/30	1.167343326	1.158	-0.93433

Table 3 Continued

Speaker Link/Array Link	Measured Height (m)	Cepstrum Predicted Height (m)	Error (cm)
60/40	1.502937277	1.533	3.006272
65/20	1.680453795	1.68	0.045379
65/30	1.338460938	1.333	0.546094
65/40	0.993973438	0.9995	-0.55266
70/20	1.865510938	1.844	2.151094
70/30	1.51765	1.51	0.765
70/40	1.175657143	1.163	1.265714
75/20	1.689695313	1.673	1.669531
75/30	1.341040625	1.338	0.304062
75/40	0.999047768	0.9928	0.624777

The following are the data from the obstacle-detection testing. Most of the spikes found in the cepstrum domain were caused by obvious reflections, which are categorized in this chart.

Table 4 - Results from Obstacle Experiments

Array to Ground			
Tape Measured Height (m)	Cepstrum Height (m)	Error (m)	% error
0.862806	0.8653	0.002494	0.288195
0.862806	0.8721	0.009294	1.065675
0.862806	0.8653	0.002494	0.288195
1.033992	1.036	0.002008	0.193855
1.177727	1.172	-0.00573	-0.48861
1.177727	1.168	-0.00973	-0.83275
1.177727	1.165	-0.01273	-1.09241

Table 4 Continued

1.177727	1.165	-0.01273	-1.09241
1.177727	1.165	-0.01273	-1.09241
1.348978	1.339	-0.00998	-0.74519
1.348978	1.342	-0.00698	-0.51998
Obstacle to Ground			
Tape Measured Height (m)	Cepstrum Height (m)	Error (m)	% error
0.542925	0.5725	0.029575	5.165939
0.658416	0.6337	-0.02472	-3.90021
0.429419	0.4293	-0.00012	-0.02766
0.542925	0.5315	-0.01143	-2.14958
0.658416	0.6541	-0.00432	-0.65978
0.79375	0.8313	0.03755	4.517022
Array to Obstacle			
Tape Measured Height (m)	Cepstrum Height (m)	Error (m)	% error
0.319881	0.3407	0.020819	6.110581
0.204391	0.2112	0.006809	3.224136
0.606425	0.5996	-0.00682	-1.13826
0.491067	0.4974	0.006333	1.273288
0.634802	0.6337	-0.0011	-0.17383
0.748308	0.7563	0.007992	1.056748
0.634802	0.6473	0.012498	1.930857
0.519311	0.5315	0.012189	2.293333
0.391914	0.402	0.010086	2.50894
0.555228	0.5587	0.003472	0.62142
Array to Speaker			
Height (m)	Cepstrum Height (m)		
1.19695	1.186		

Table 4 Continued

Double Spikes		
Height (m)	Cepstrum Height (m)	
	0.8517	
	0.2248	
	0.5175	
	0.9198	
	0.5792	

However, not all data could be categorized successfully. The location of the spike is in the right column of the next chart, and the dimensions that accompanied it are to its left.

Table 5 - Uncategorized Results from Obstacle Experiments

Array to Ground (m)	Ground to Obst (m)	Array to Obst (m)	Ceps Height (m)
0.86280625	0	0.86280625	0.7631
0.86280625	0	0.86280625	1.622
0.86280625	0	0.86280625	2.385
0.86280625	0.542925	0.31988125	0.3747
0.86280625	0.65842	0.20438625	0.3271
0.86280625	0.65842	0.20438625	0.4361
1.033991667	1.033991667	0	1.383
1.033991667	1.033991667	0	1.962
1.033991667	1.033991667	0	2.896
1.033991667	0.427566667	0.606425	0.62
1.033991667	0.427566667	0.606425	1.117
1.033991667	0.542925	0.491066667	0.6677
1.033991667	0.542925	0.491066667	0.8994
1.033991667	0.542925	0.491066667	1.301
1.033991667	0.65841	0.375581667	0.402
1.033991667	0.65841	0.375581667	0.8994
1.177726563	1.177726563	0	2.235
1.177726563	0.42941875	0.748307813	1.39

Table 5 Continued

Array to Ground (m)	Ground to Obst (m)	Array to Obst (m)	Ceps Height (m)
1.177726563	0.658415625	0.519310938	0.9335
1.177726563	0.658415625	0.519310938	1.349
1.177726563	0.7858125	0.391914063	0.6844
1.177726563	0.7858125	0.391914063	0.9812
1.177726563	0.7858125	0.391914063	1.758
1.348978125	0	1.348978125	2.576
1.348978125	0.42783125	0.921146875	1.015
1.348978125	0.42783125	0.921146875	1.438
1.348978125	0.42783125	0.921146875	2.358
1.348978125	0.42783125	0.921146875	2.78
1.348978125	0.542925	0.806053125	0.9062
1.348978125	0.542925	0.806053125	1.44
1.348978125	0.658415625	0.6905625	1.444
1.348978125	0.658415625	0.6905625	1.775
1.348978125	0.658415625	0.6905625	1.806
1.348978125	0.658415625	0.6905625	2.003
1.348978125	0.79375	0.555228125	1.029
1.348978125	0.79375	0.555228125	1.479

Outdoor Experiment Results

Table 6 - Outdoor Experimental Results

Height (m)	Error	SNR
1.25863	-0.01403	20.1
1.43552	-0.00042	18.3
1.72806	0.01184	18.8979
1.728	0.0119	15.7
2.02061	0.02409	16.6
2.36758	-0.01808	15
2.61931	0.03499	14.8
2.6125	0.0418	14
2.6125	0.0418	15.8
2.89825	0.06085	12.9
2.89825	0.06085	15.1
3.22481	0.03909	13.3
3.22481	0.03909	13.7
3.2316	0.0323	12.3
3.54457	0.02413	12.6
3.54457	0.02413	12.5

BIBLIOGRAPHY

Brower, J. M. (2004). Preventing Brownout. Special Operations Technology. **2**.

Childers, D. G., D. P. Skinner, et al. (1977). "The Cepstrum: A Guide to Processing." **65**(10).

Cohen, L. (1993). "The Scale Representation." IEEE Transactions on Signal Processing **41**(12).

Colby, S. (2005). Rotor Spin. Rotor & Wing. Rockville, MD, Access Intelligence.

Fjell, P. O. (1976). "Use of the Cepstrum Method forArrival Times Extraction of Overlapping Signals Due to Multipath Conditions in Shallow Water." **59**(1).

Fuller, C., S. Tavakkoli, et al. (1988). "Application of the Complex Cepstrum to Locate Acoustic Sources Near Reflective Surfaces." **26**(8).

Harrington, W. (2007). Helicopter brownout. http://www.eglin.af.mil/news/story.asp?id=123052402.

Horowitz, P. and W. Hill (1980). The Art of Electronics, Cambridge University Press.

Johnson, W. (1980). Helicopter Theory. New York, Dover.

Pierce, A. D. (1994). Acoustics: An Introduction to Its Physical Principles and Applications. Melville, NY, Acoustical Society of America.

Rodgers, S. J. (1968). Evaluation of the Dust Cloud Generated by Helicopter Rotor Downwash. U. A. A. Laboratories. Fort Eustis, VA**:** 136.

Ryerson, C. C., R. B. Haehnel, et al. (2005). Visibility Enhancement in Rotorwash Clouds. 43rd AIAA Aerospace Sciences Meeting and Exhibit. Reno, Nevada.

Sabbagh, L. (2006). Flying Blind in Iraq: US Helicopters Navigate Real Desert Storms. Popular Mechanics.

Schmitz, F. H. (1995). Rotor Noise. Aeroacoustics of Flight Vehicles. H. Hubbard. Woodbury, NY, Acoustical Society of America. **1:** 65-150.

Temkin, S. (2005). Suspension Acoustics. New York, Cambridge University Press.

Wald, M. L. (2000). Pilot's Rapid Descent Cited in Osprey Crash Fatal to 19. New York Times. New York City.

Weisstein, E. W. (2008). "Cardioid." Mathworld - A Wolfram Web Resource Retrieved 7/15/2008, from http://mathworld.wolfram.com/Cardioid.html.

Wierninga, J. (1993). "Representative Roughness Parameters for Homogenous Terrain." Boundary Layer Meteorology **63**: 323-363.