HOW NONIDEAL MICROPHONES AFFECT DIRECTIONAL IMPULSE RESPONSE MEASUREMENTS IN ROOM ACOUSTICS

by

J. James Esplin

A capstone project report submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Bachelor of Science

Physics 492R

Department of Physics and Astronomy

Brigham Young University

April 2010

BRIGHAM YOUNG UNIVERSITY

DEPARTMENT APPROVAL

of a capstone project report submitted by

J. James Esplin

This capstone report has been reviewed by the research advisor, research coordinator, and department chair and has been found to be satisfactory.

Date	Brian E. Anderson, Advisor
Date	Eric G. Hintz, Research Coordinator
Date	Scott D. Sommerfeldt, Chair

ABSTRACT

HOW NONIDEAL MICROPHONES AFFECT IN ROOM ACOUSTICS

J. James Esplin

Department of Physics and Astronomy

Bachelor of Science

Reduction of undesirable reflections in a room (by treating the offending reflecting surfaces) may only be accomplished if the locations of the offending surfaces are determined. Several measurement techniques exist to identify these surfaces, including the Polar Energy Time Curve (Polar ETC) method, which requires six cardioid impulse response measurements along each Cartesian axis. The purpose of the current study is to quantify the angular estimation error introduced into the Polar ETC due to non-ideal microphone directivities, imprecise microphone positioning, and different signal processing techniques. In this paper, it is shown that errors may be minimized with the correct choice of microphone type but also with additionally applied bandpass filtering. Additionally, errors due to positioning and signal processing variations have a relatively large effect on the overall performance of the Polar ETC.

ACKNOWLEDGMENTS

I'd like to acknowledge the following people

- Dr. Brian E. Anderson and Dr. Timothy W. Leishman, my faculty advisors
- The BYU Acoustics Research Group, for their support and help
- My wife, Jessica, and my daughter, Kathryn
- The BYU Endowment fund for its support of my research

Contents

List of Figures

Chapter 1

Introduction

A common goal in architectural acoustics is to make an environment as pleasing as possible for listeners. In some environments, reflections from walls reinforce the direct sound, resulting in pleasing sensations of spaciousness, intimacy, clarity, and warmth. However, other environments can suffer from reflections that result in perceptible echoes and incorrect localization which annoy and/or distract the listener from the desired source signal(s). In these latter instances it is desirable to reduce or eliminate undesirable room reflections. However reducing undesirable reflections, by treating the offending reflecting surfaces, may only be accomplished if the locations of the offending surfaces are determined. This is the primary reason that reflection localization methods are used.

A commonly used method to locate an offending surface is to analyze an impulse response of the room that contains the undesirable echo. Using that impulse response, one can determine the time delay between the direct source arrival and the arrival of the offending reflection. This time delay corresponds to the additional distance the reflection traveled compared to the direct source. It is possible to guess the location of the offending surface using this extra distance together with a knowledge of the room dimensions. However, the method just described is time-consuming and can be very inaccurate, especially in the case of simultaneous reflection arrivals. This is why automated reflection localization methods were developed.

There are several automatic localization methods currently in use. Some use spherical beamforming [1], directional microphones [2], or correlation methods with spherical arrays [3] and tetrahedral probes [4–6] to identify reflecting surfaces. The spherical beamforming method, which consists of a 32-microphone spherical array, is costly and requires very precise construction. The directional microphone method has, to the author's knowledge, only been presented theoretically. The correlation method with a spherical array is also costly, requiring fifteen microphones. The correlation methods with tetrahedral probes have also only been presented theoretically. The method most commonly used today is the Polar Energy-Time Curve (Polar ETC) [7]. It uses a series of six sequential measurements, each measured at the same location, with a cardioid microphone oriented along each of the six Cartesian axes. From these measurements, the difference in the so called "energy-time curve" along an axis is used to identify the location of reflecting surfaces. The merits of the energy-time curve (ETC) will be discussed later in the paper.

The Polar ETC assumes three things: first, the microphone maintains a cardioid directivity pattern over its entire frequency band (when no cardioid microphone does so); second, all measurements are taken at the same point in space (which is difficult to maintain as the microphone is rotated); third, a certain, flawed signal processing technique is used (the aforementioned ETC). To the authors' knowledge, a study quantifying potential errors in the Polar ETC itself or those introduced by violating any of the above assumptions has not been done. The purpose of this paper is to quantify the error introduced by non-cardioid microphones, inconsistent microphone positioning, and by a more physically realistic signal processing techniques. These errors indicate the confidence level in results obtained with the Polar ETC.

This paper will show that a range of cardioid family microphones can provide acceptable results (as long as the microphone is not omnidirectional or bi-directional), though the measurements made with one of these microphones should be bandpass filtered to minimize errors. Additionally, errors in microphone positioning and signal processing variations have a relatively large effect on the overall performance of the Polar ETC. This implies that although microphone directivity is unimportant, microphone positioning and post-processing techniques are important for successful implementation of the Polar ETC.

Chapter 2

Theoretical Predictions

2.1 Directivity Theory

To predict the effect of microphone variations directivity variations on the Polar ETC, one needs to first understand the Polar ETC itself. The Polar ETC is based on the so called ETC [8–11]. This curve is essentially the envelope of the impulse response and is computed by taking the magnitude of the Hilbert transform of the impulse response. Originally it was erroneously interpreted as a method to obtain a value proportional to either the instantaneous sound intensity or the instantaneous energy density. Becker used the energy-time curve for the Polar ETC under this assumption. It has been shown that the ETC is based upon an acausal operation and does not in general accurately represent energy flow as Becker asserted [12, 13]. Regardless, the form of the equations developed by Becker, [Eqs. (2.1) and (2.2)], are correct because they are based upon the standard Cartesian to spherical coordinate transformation.

The localization of reflecting surfaces in the Polar ETC is governed by the following two equations,

$$\phi_M = \tan^{-1}\left(\frac{E_{+y} - E_{-y}}{E_{+x} - E_{-x}}\right), \tag{2.1}$$

and

$$\theta_M = \sin^{-1}\left(\frac{E_{+z} - E_{-z}}{E_{TOT}}\right), \tag{2.2}$$

where ϕ represents the azimuthal angle measured from the +x axis, θ represents the elevation angle measured from the x-y plane, E represents the "energy" measured along the axis identified by the subscript, and E_{TOT} represents the total "energy." ϕ_M and θ_m represent the measured (estimated) angle for an incoming reflection, while ϕ and θ, used later on, represent the actual angle of the same reflection [14].

We first assume that the components (interpreted here as instantaneous potential energies) in Eqs. (2.1)–(2.2) are proportional to the pressure squared, and therefore the square of the following directivity function

$$H(\theta) = A + B\cos(\theta), \tag{2.3}$$

subject to these constraints

$$A \geq 0, B \geq 0, A + B = 1.$$

The analysis presented in this section employs squared directivity functions. Note however that D'Antonio *et al.* (of which Becker was a coauthor) [14] assumed that these components were proportional to the directivity function, which may in fact be correct for the ETC formulation. Thornock presented an analysis [15] similar to the one presented in this section [up to Eq. (2.12)] based upon the assumption by D'Antonio *et al.*

Equation (2.3) defines the directivity of any microphone in the cardioid family. If $A = 1$ and $B = 0$, this implies an omnidirectional directivity (where it is equally sensitive to sound incident from all directions). If $A = B$, this implies a true cardioid directivity (where it is more sensitive to sound incident upon the front of the microphone than sound incident from behind it). If $B = 1$ and $A = 0$, this implies a bidirectional or figure-8 directivity (the microphone is equally sensitive to incident sound from the front or behind it, but not sensitive to sound incident from its sides).

The Cartesian energetic components then become

$$E_{+x} = E_0(A + B\cos\phi\cos\theta)^2, \tag{2.4}$$

$$E_{-x} = E_0(A - B\cos\phi\cos\theta)^2, \tag{2.5}$$

$$E_{+y} = E_0(A + B\sin\phi\cos\theta)^2, \tag{2.6}$$

$$E_{-y} = E_0(A - B\sin\phi\cos\theta)^2, \tag{2.7}$$

$$E_{+z} = E_0(A + B\sin\theta)^2, \tag{2.8}$$

$$E_{-z} = E_0(A - B\sin\theta)^2, \text{ and} \tag{2.9}$$

$$\begin{aligned} E_{TOT} &= \sqrt{(E_{+x} - E_{-x})^2 + (E_{+y} - E_{-y})^2 + (E_{+z} - E_{-z})} \\ &= 4ABE_0. \end{aligned} \tag{2.10}$$

Substitution of Eqs. (2.4)–(2.10) into Eqs. (2.1)–(2.2) yields

$$\begin{aligned} \phi_M &= \tan^{-1}\left(\frac{E_0[A + B\sin\phi\cos\theta]^2 - E_0[A - B\sin\phi\cos\theta]^2}{E_0[A + B\cos\phi\cos\theta]^2 - E_0[A - B\cos\phi\cos\theta]^2}\right) \\ &= \tan^{-1}\left(\frac{4AB\sin\phi\cos\theta}{4AB\cos\phi\cos\theta}\right) \\ &= \phi \end{aligned} \tag{2.11}$$

$$\begin{aligned} \theta_M &= \sin^{-1}\left(\frac{E_0[A + B\sin\theta]^2 - E_0[A - B\sin\theta]^2}{4ABE_0}\right) \\ &= \sin^{-1}\left(\frac{4AB\sin\theta}{4AB}\right) \\ &= \theta \end{aligned} \tag{2.12}$$

This means that as long as the microphone is a member of the cardioid family, microphone directivity should have no effect at all on the Polar ETC, contingent upon two conditions. First, an omnidirectional microphone must not be used (B must not equal to zero). This is because when $B = 0$, Eqs. (2.1) and (2.2) become indeterminate. Second, a figure-8 microphone must not be used (A must not equal zero) for exactly the same reason. Thus, theoretically, even when the microphone is practically omnidirectional or bidirectional (*i.e.* $A = 0.99$ and $B = 0.01$ or $A = 0.01$ and $B = 0.99$) the analysis given here holds for a noise-free signal. However, when noise is added to the signal, the range of acceptable values for A are limited.

Figure 2.1 shows how much angular estimation error results as A is varied from 0 to 1 (assuming $A + B = 1$) when random noise, with an amplitude one percent of the actual energy, is introduced into the analysis. It should be noted here that the general trend of Fig. 2.1 is approximately independent of the amplitude of the random noise (assuming the noise is still a fraction of the actual energy). As illustrated by Fig. 2.1, the errors are relatively small when A is between X and Y (assuming a tolerance of one degree error from the optimal value for A).

It should be stressed here that, in general, cardioid microphones are inherently band limited in terms of the range of frequencies at which they maintain a true cardioid directivity pattern. This is due to the fact that the phase shift, introduced between the omnidirectional and dipole components of a cardioid microphone, is inherently frequency dependent. At very low frequencies, the microphone will behave as an omnidirectional microphone. At very high frequencies, the microphone will behave as a bidirectional (figure-8) microphone. Figure 2.2 shows a surface plot of the directivity of the Shure SM81 microphone as a function of angle and frequency. Subplot (a) of the figure displays the measured directivity while subplot (b) displays the departure of the measured directivity of the microphone, $D_{measured}$, versus a frequency independent expression for a true cardioid directivity ($D_{measured} - (.5 + .5 * \cos\theta)$) . Subplot (b) clearly shows the departure of the microphone at low and high frequencies from a cardioid directivity pattern.

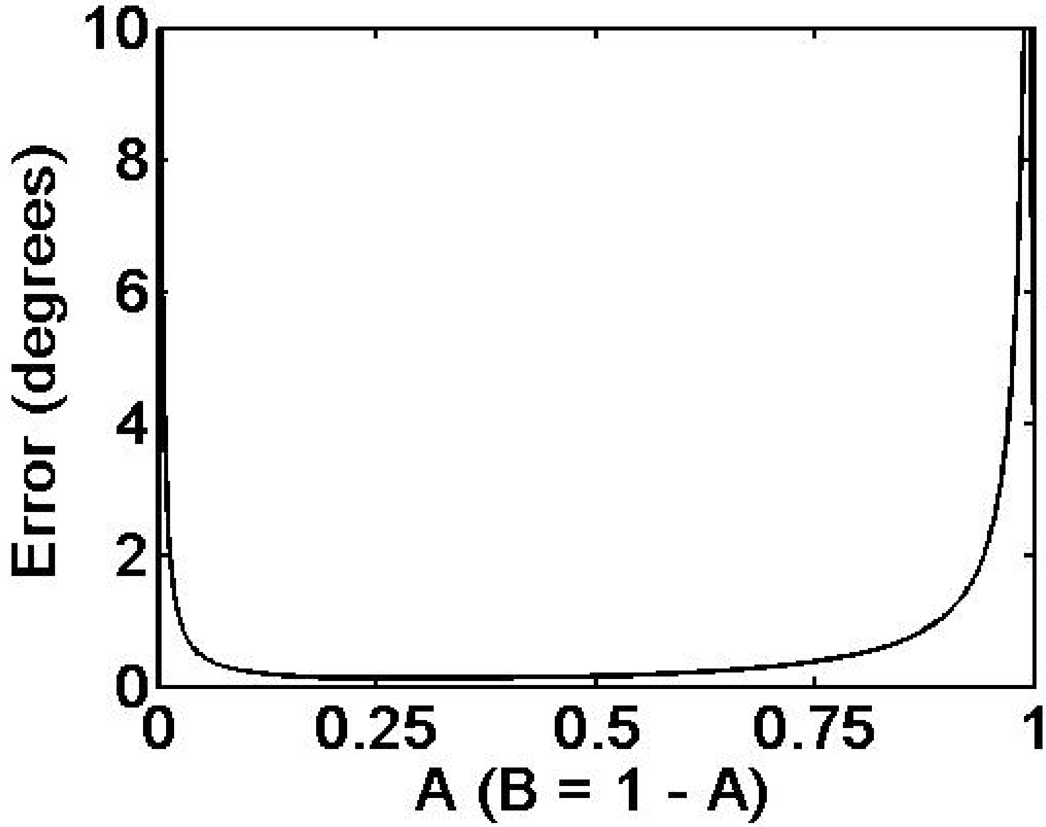


Figure 2.1 Simulated angular error as a function of A, which varies from 0 (omnidirectional directivity) to 1 (bidirectional directivity).

We did not attempt to theoretically or numerically predict the effects of inconsistent microphone positioning due to the unclear relationship between spatial variances and the ETC.

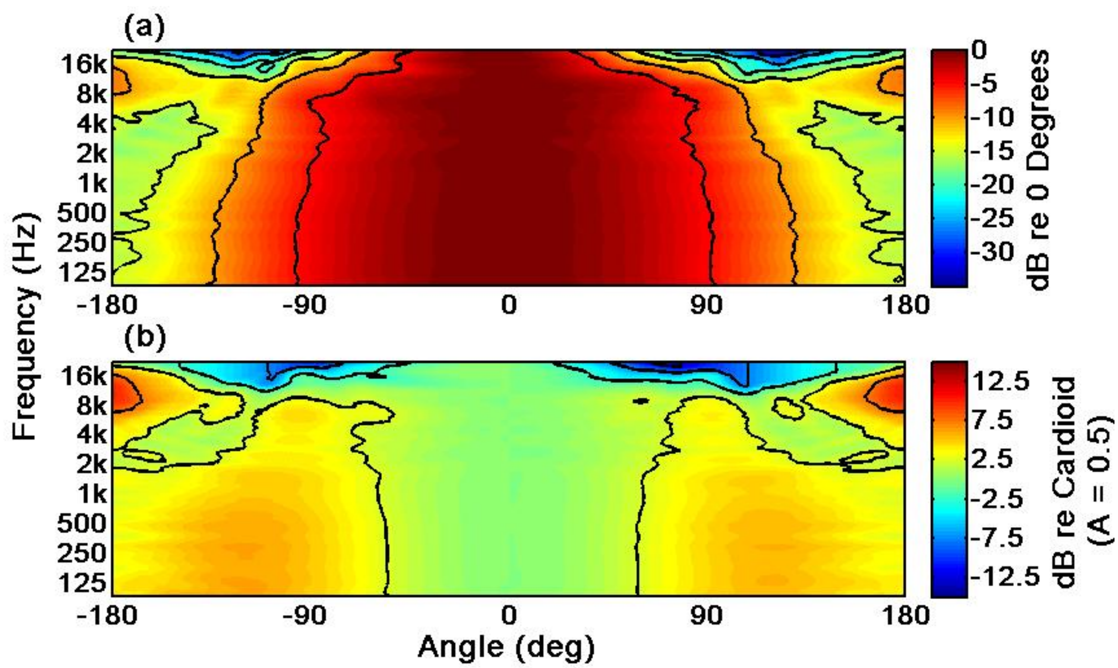


Figure 2.2 (a) Directivity of a Shure SM81 Microphone as a function of angle and frequency. (b) Difference between the directivity in (a) and a frequency-independent ideal cardioid. The color represents magnitude and the contours are at the locations labeled on the color scale.

Chapter 3

Experimental Methods

3.1 Experimental Setup

In order to experimentally quantify the confidence level of Polar ETC measurements due to the issues described in the introduction, a variable acoustics chamber, a Tannoy dual-concentric loudspeaker, and two different microphones are used. The first microphone is an AKG 414 microphone. This microphone has a variable polar pattern—it can shift between omnidirectional(OD) ($A = 1$), subcardioid(SC) ($A = 0.75$), cardioid(C) ($A = 0.5$), hypercardioid(HC) ($A = 0.25$) and figure-8(F8) ($A = 0$) polar patterns. The second microphone is a Shure SM81 cardioid microphone. Both are pictured in Fig. 3.1.

We first describe a set of Polar ETC experiments with reflection surfaces at known angles to quantify the accuracy of the estimated angles of incidence. In order to take sequential rotation measurements about a point in space with a single microphone, a microphone positioner, able to rotate over a full 4π steradians, is used. An altazimuth-mounted laser pointer is placed at the location of the microphone and, with a set of plane mirrors, is used to accurately determine the incidence angles for reflecting

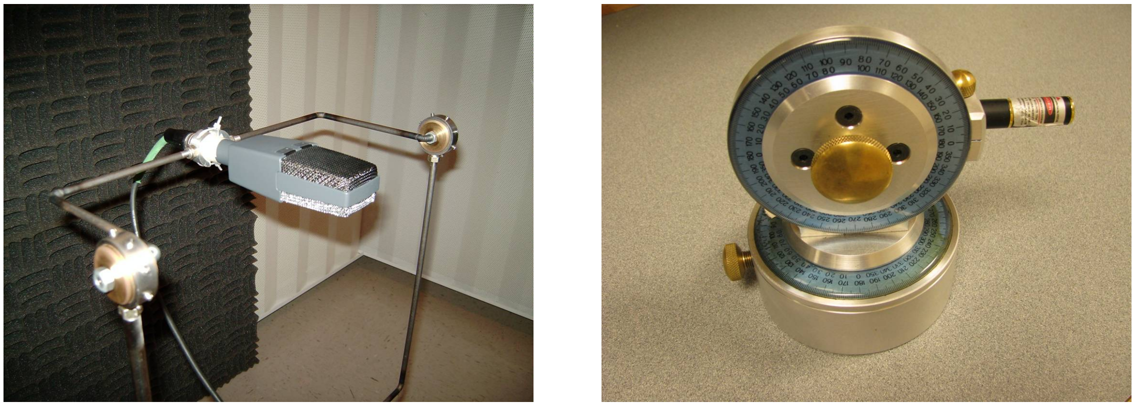
(a) AKG 414

(b) Shure SM81

Figure 3.1 Pictures of all microphones used. (a) The AKG 414 is a microphone with a variable polar pattern. (b) The Shure SM81 is a cardioid microphone.

surfaces in each experiment. Both of these apparatus are pictured in Fig. 3.2.

Two separate experiments are done. The first involves a microphone and a loudspeaker, placed in a variable acoustics chamber configured as a hemi-anechoic chamber (hemi-anechoic above approximately 100 Hz). The direct sound arrives from an angle (ϕ, θ) of (0,0) (measured in degrees) while a reflection is incident from (0,-43.5). The second experiment uses the same microphone and speaker placement as the previous setup, but the microphone is rotated -45^o in ϕ relative to the speaker, thus shifting the microphone's coordinate system. In this case, the direct sound arrives at an angle of (45,0) while a reflection is incident from (45,-43.5). Both of the above experiments are done with the Shure SM81 microphone and all five settings of the AKG 414 microphone.



(a) (b)

Figure 3.2 Photographs of some experimental apparatus used. (a) Rotational microphone positioner. This allows rotation of a microphone about a specific point. (b) Altazimuth-mounted laser pointer. This allows determination of the actual angle of arrival, in conjunction with a set of plane mirrors.

3.2 Directivity Error

Two methods are used to quantify the error due to nonideal microphone directivity. First, measurements taken with the variable polar patterns of the AKG 414 are compared directly. These measurements employ a bandwidth up to approximately 30 kHz (the loudspeaker only responds up to 30 kHz), using a 192 kHz sampling frequency. Polar ETC measurements with each different AKG microphone configuration are then compared to determine whether the trend presented in Chap. 2 is correct. Second, a fixed high-pass filter (with a cutoff frequency of 300 Hz) and a variable cutoff frequency low-pass filter are used with the Shure SM81 microphone Polar ETC measurement signals to also determine agreement with the trend in Chap. 2. The fixed high-pass filter cutoff frequency excluded the omnidirectional and part of the subcardioid directivity behavior from the measurement. As we lowered the low-pass cutoff frequency, we gradually eliminated the figure-8 and hypercardioid directivity behaviors and thus restricted the Polar ETC results to a microphone possessing a true cardioid directivity. We thus expect the angular error to start high, decrease, and reach a minimum as we reduce the low-pass filter cutoff frequency, as predicted in the theory presented in Chap. 2. One tradeoff that is inherent in this type of analysis is that the temporal resolution of the arrival times of reflections decreases as we eliminate the higher frequency information.

3.3 Spatial Error

As stated in Chap. 1, the Polar ETC requires that each of the six Cartesian axes measurements are made at the same point in space. In order to quantify the errors associated with microphone positioning inaccuracies, the arrival of the direct sound is compared among each of the six measurements from one trial (i.e. x, y, z). Ideally, if there was no error in positioning the direct sound would arrive simultaneously for each of the six measurements. Figure 3.3 displays sample received signals recorded with each of the six configurations. Subplot (a) shows the unaligned impulse responses. Subplot (b) shows the unaligned ETCs for each impulse response. One can determine that the direct sound does not simultaneously arrive in each signal, which implies errors in microphone positioning. These positioning errors may be due to physical inconsistencies in microphone placement between measurements or introduced through a shift in the microphone's acoustic center. The acoustic center is a function of frequency and is different for each microphone. Because of this latter fact it is difficult to quantify the microphone's acoustic center, implying that this error is unavoidable and microphone specific.

In order to correct for these misalignments, a time alignment of the six measurements is performed. One measurement was taken as a baseline and cross-correlations were found between the baseline signal and the other five signals. These cross-

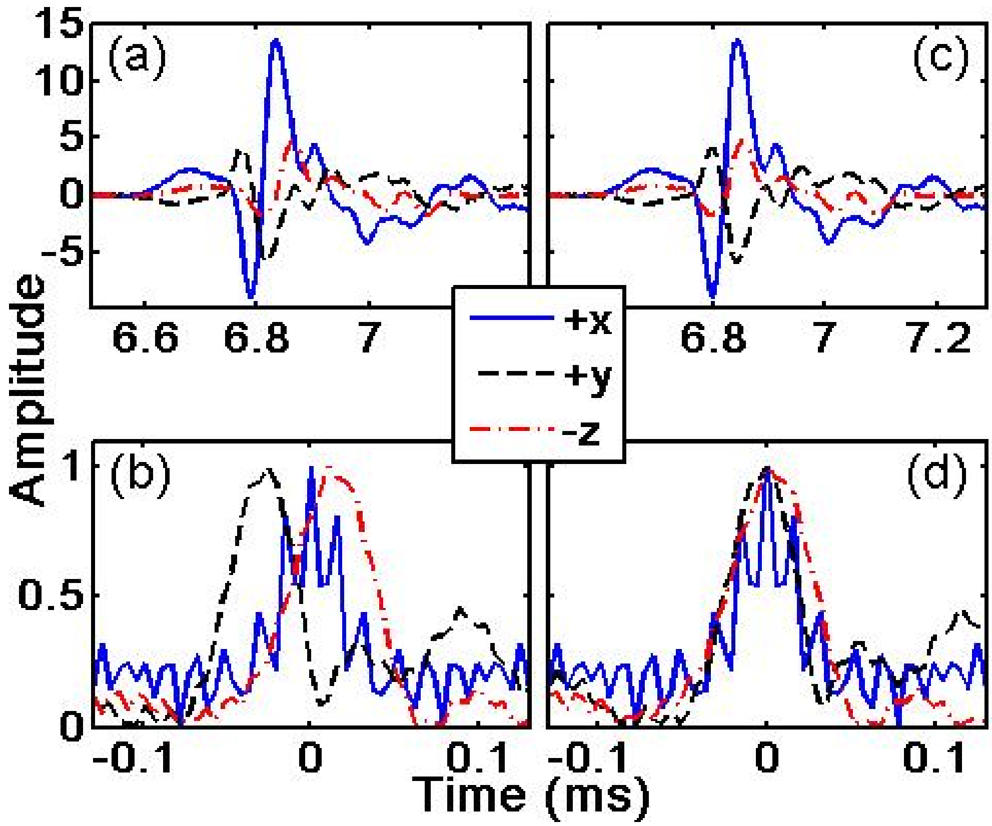


Figure 3.3 Example of the time-alignment process using three measurements from one trial. (a) Shows the unaligned impulse responses. (b) Shows the cross correlations for each impulse response in (a). (c) Shows the aligned impulse responses while (d) shows their minimized cross correlations. Notice how the measurements were aligned using the energy-time curve instead of the impulse response—this is to correct for phase mismatches, which can be seen in (a) and (c).

correlations correspond to the time delay between the baseline signal and the other signal. By minimizing these cross-correlations, the time delay between the two measurements is eliminated. Subplot (c) shows the aligned impulse responses while subplot (d) shows the minimized cross correlations. It is anticipated that this time alignment reduces the amount of error introduced through inconsistent microphone positioning. The angular error in the Polar ETC incidence angle estimation from the set of time-aligned signals is then compared to the errors resulting from the same set of signals without a time alignment. If the angular errors are similar, then the misalignments are inconsequential. If they are not, this means that the Polar ETC is very sensitive to inconsistent microphone positioning.

3.4 Signal Processing Error

As mentioned in Chap. 2, the Polar ETC is based on the so called ETC [8–11]. This curve is essentially the envelope of the impulse response. It was originally developed as a method to obtain an instantaneous value proportional to the instantaneous energy density. It has been shown that the ETC is based upon an acausal operation and therefore does not accurately represent energy flow [12,13]. For this reason, we explore the use of the squared impulse response (SIR), instead of the ETC, since the SIR is proportional to the instantaneous potential energy (an easily measurable, physically based quantity). An example of the SIR, along with the IR and ETC, are displayed in Figure 3.4. We term this new method the Polar SIR. By comparing the angular errors produced by both of these methods we can determine how sensitive the Polar ETC is to a more physically-based signal processing method, the Polar SIR.

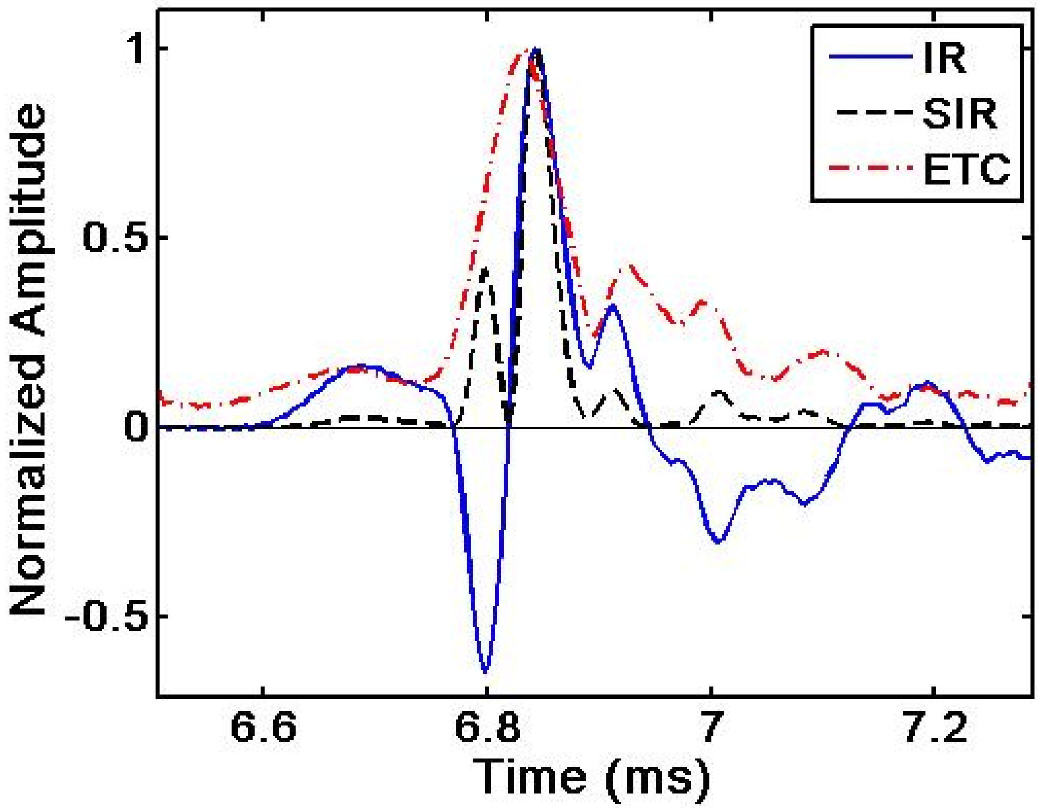


Figure 3.4 Comparison of the energy-time curve and the squared impulse response for the impulse response shown. Amplitudes have been normalized for all three curves.

Chapter 4

Results

In each of the previously described sets of experiments the angular error is quantified three different ways using circular statistics, as illustrated in Fig. 4.1 [16]. The first type of error is the angular error between the estimated location associated with the peak of a given sound arrival and the actual location associated with that same sound arrival. The second type of error is the angular error of the average estimated location of the peak of a sound arrival and its three neighboring time samples on either side of the peak (seven time samples in all), compared to the actual sound arrival location. This type of error corrects for a peak value that may be unusually large due to noise. Finally, the third type of error is the circular standard deviation of the aforementioned seven points to determine the angular scatter of the estimation.

4.1 Directivity Error

Figure 4.2 displays the three types of angular error, averaged over each different experimental setup for both the direct sound arrival and first reflection arrival, for the measurements taken with the AKG 414 in its five polar pattern settings (OD,

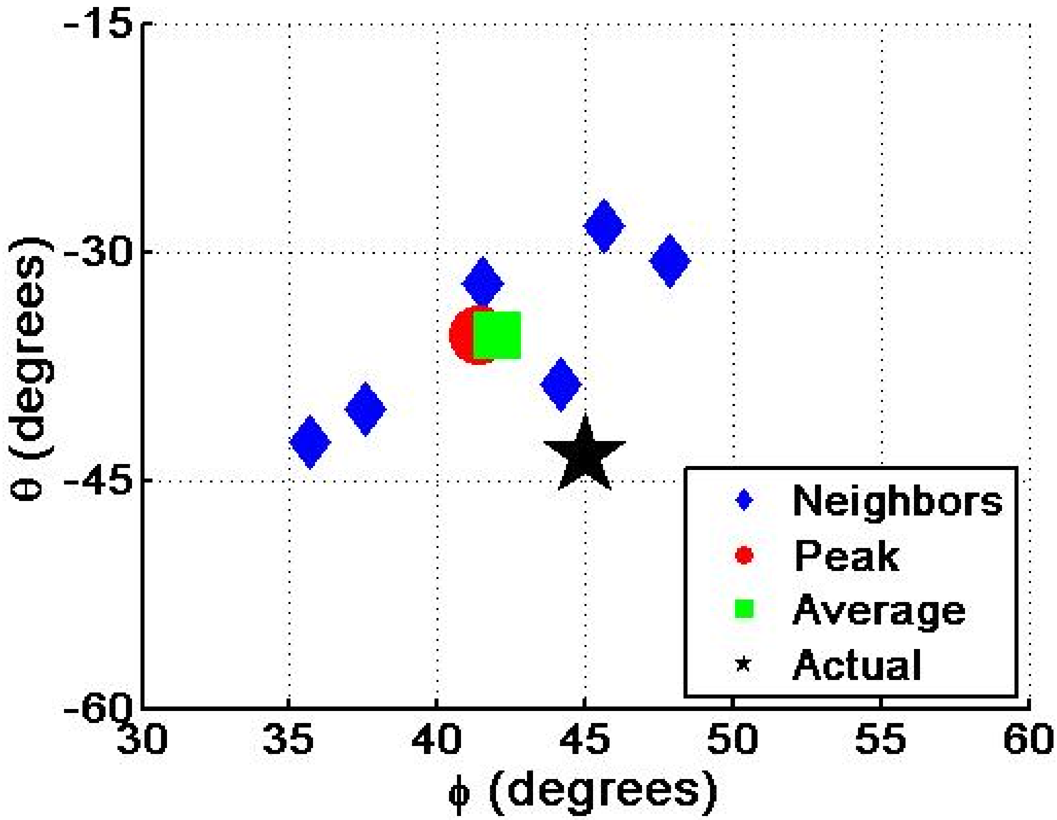


Figure 4.1 Example illustration showing both the estimated and actual image source locations. The star is the actual source direction, the circle is the peak source direction estimation, and the square is the angular average of the peak direction estimation and the six other neighboring estimations (represented by diamonds).

SC, C, HC, and F8). We see that the Polar ETC follows the trend predicted in Chap. 2. As long as the microphone is not nearly OD or F8, the angular errors remain relatively low. This implies that, for optimal results, one should band limit Polar ETC measurements to eliminate the OD and F8 behaviors at low and high frequencies respectively.

We now look at the results of our progressive filtering study as we filtered out the low frequency information and steadily filtered out the high frequency information (done with measurements taken with the Shure SM81). In Fig. 4.3 we again see the

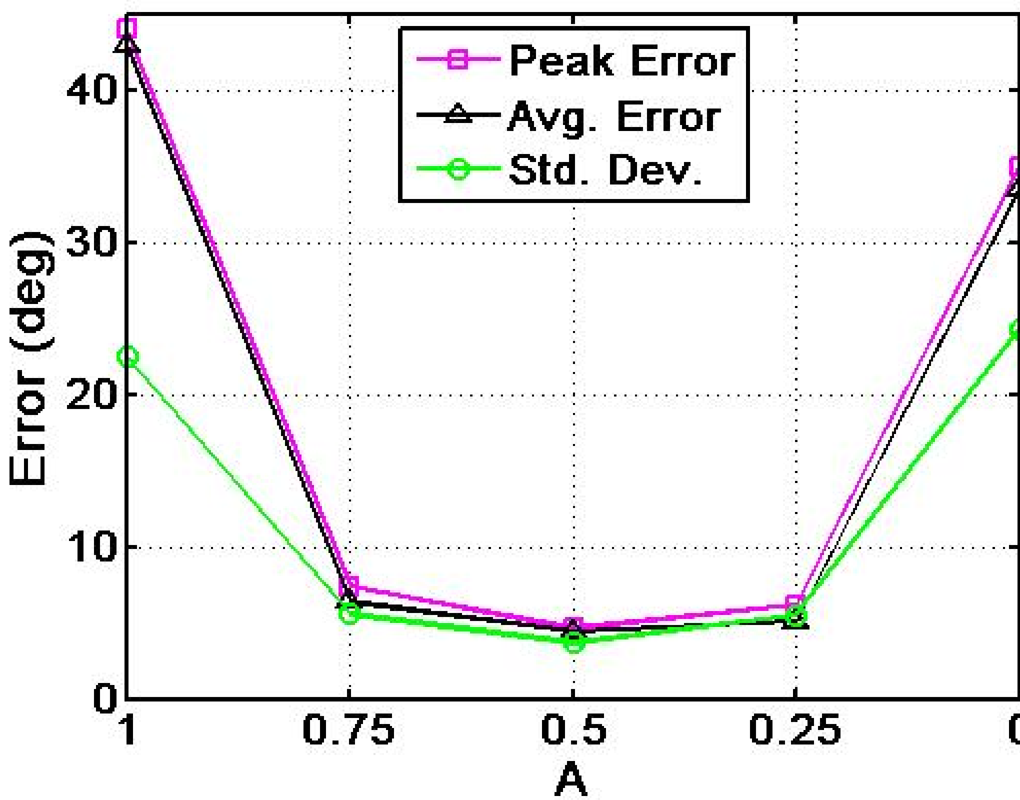


Figure 4.2 Angular errors in the Polar ETC as a function of microphone directivity. Peak error, average error, and standard deviation are shown as explained in Chap. 4.

same trend we saw in Fig. 4.2. As the low-pass filter cutoff frequency is decreased, the angular estimation error decreases quickly and remains constant over a large frequency range. The reason for the large errors on the low end of this plot is not likely due to the omnidirectional response; rather, the frequency bandwidth becomes too narrow. This causes the measurements to contain insufficient high-frequency information for temporal estimation of arrival times.

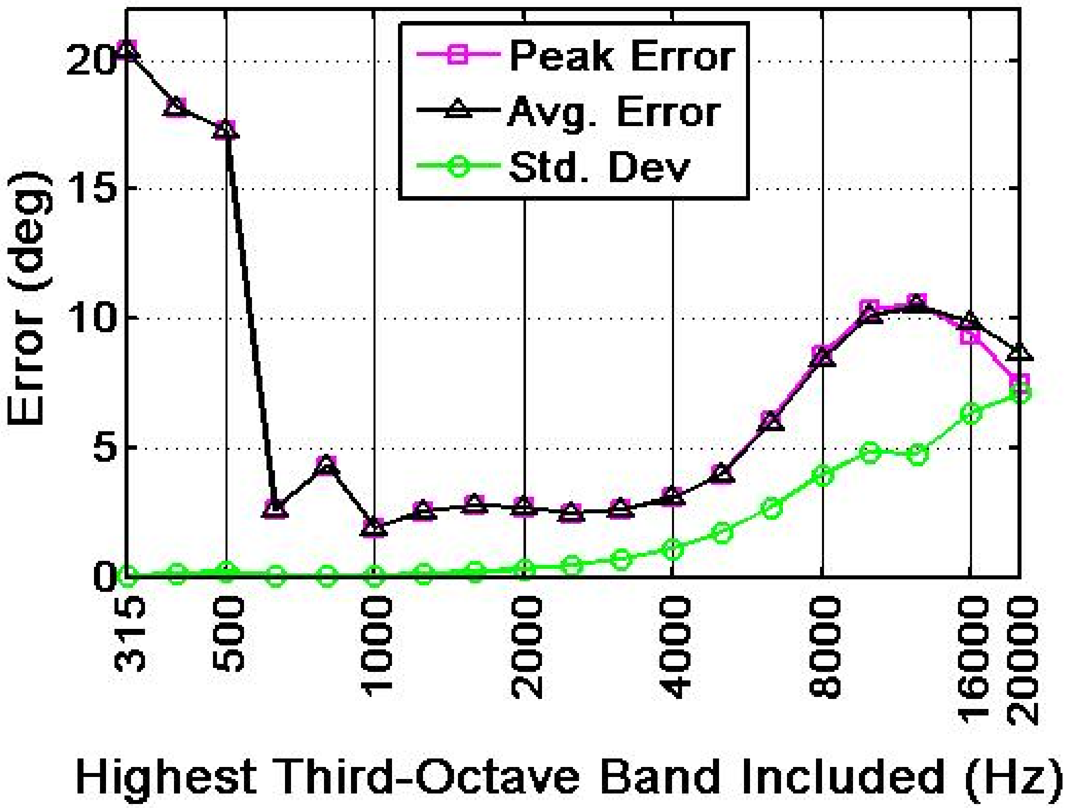


Figure 4.3 Angular error as a function of varying low-pass filter cutoff frequency, thereby controlling the directivity. A constant high-pass low frequency cutoff at 300 Hz (eliminating OD and SC directivities) is also used. Peak error, average error, and standard deviation are shown as explained in Chap. 4.

4.2 Spatial Error

Figure 4.4 compares the average errors when signals are not time-aligned versus when they are time-aligned. In Fig. 4.4 we see that the errors in positioning can increase the angular error by 35% or more. This implies that inconsistencies in microphone positioning due to either imperfect measurements or shifts in the acoustic center can have a significant impact on the accuracy of the Polar ETC. To remedy this, the Polar ETC should implement automated time-alignment methods into their process similar to the one used for this study. However, care should still be exercised to minimize physical microphone positioning inconsistencies. The authors are unsure at the present time why the errors of the HC directivity increase, instead of decrease, as the time-alignment is applied.

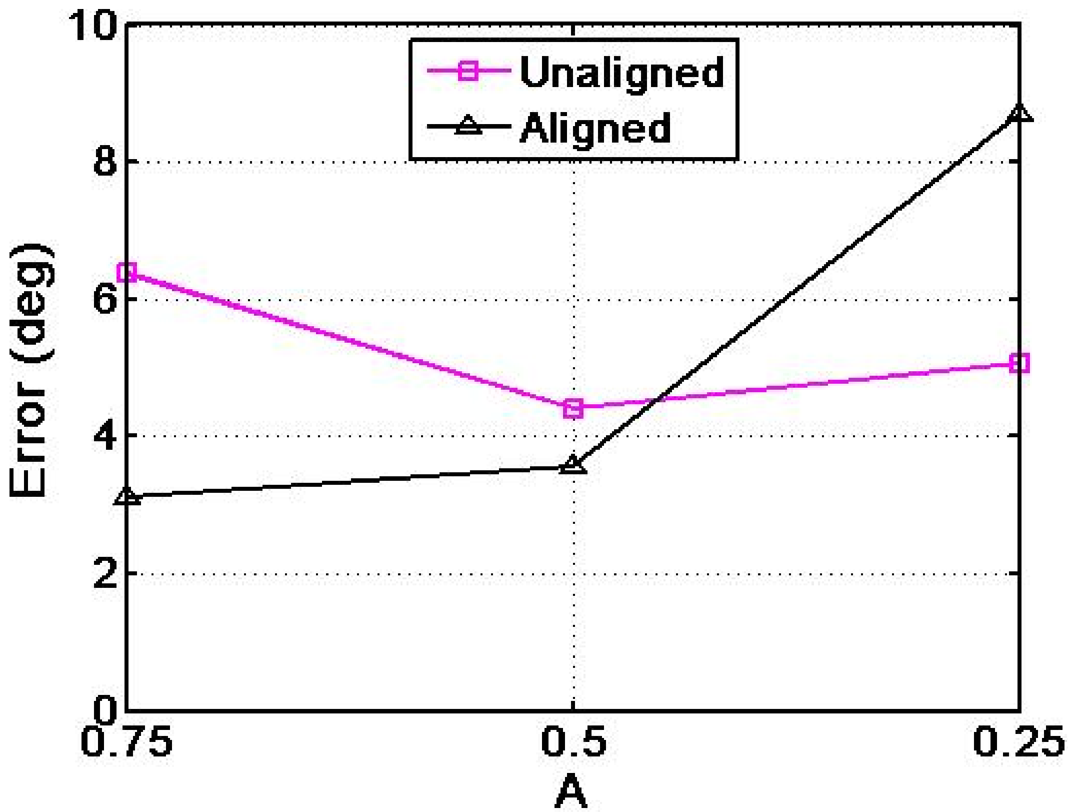


Figure 4.4 Average error for unaligned and aligned signals as a function of microphone directivity. Average error is explained in Chap. 4.

4.3 Signal Processing Error

Figure 4.5 compares the average errors when using the Polar SIR versus the Polar ETC. In Fig. 4.5 we see that the Polar SIR performs poorly in comparison to the Polar ETC. This shows that the equations that govern the Polar ETC work much better with the ETC than the SIR despite the flaws of the ETC. One possible reason for this, referring to Fig. 3.4, is that the peak of the ETC, notwithstanding its lack of physical meaning, comes closer to the correct peak in actual energy flow than that of the SIR, especially in cases where a single reflection is composed of both a positive and a negative peak. Another reason has to do with the "smoothness" of the ETC compared with the SIR. Because the ETC is "smoother" than the SIR, it will be less sensitive to random noise that the SIR. Regardless, further research needs to be done to investigate these and other possible reasons for this result.

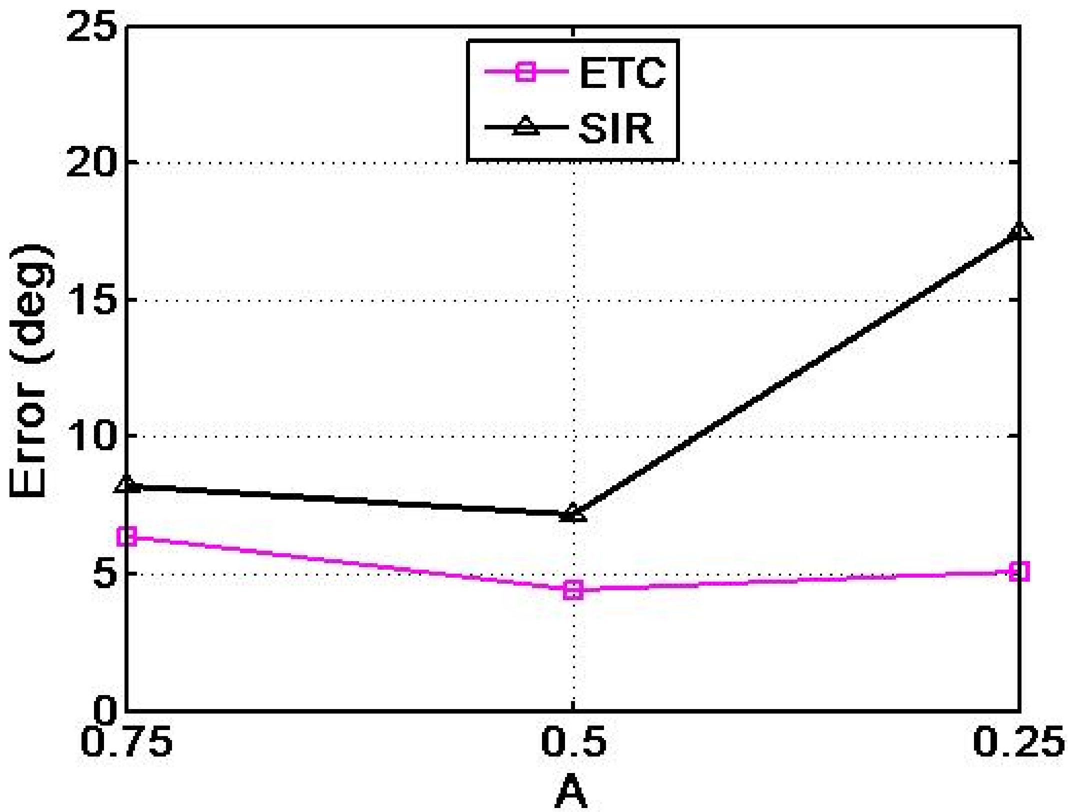


Figure 4.5 Average angular error for the Polar SIR and the Polar ETC as a function of microphone directivity. Average error is explained in Chap. 4.

Chapter 5

Conclusions

Theoretical developments in this paper have shown that the Polar ETC may employ many different types of cardioid family microphones as long as the microphone directivity is not nearly omnidirectional or nearly figure-8.

A set of experiments has been conducted to verify the theoretical findings. Based on the data presented, it can be concluded that the Polar ETC is not sensitive to errors in microphone directivity as long as the polar pattern does not approximate an omnidirectional or figure-8 pattern. One should therefore consider employing a band-pass filter to filter out the omnidirectional and figure-8 behaviors in the low and high frequency portions of a standard cardioid microphone response. Additionally, the "so-called" average error estimation of the incidence angle was always slightly lower than the peak error in Fig. 4.2. This suggests that Polar ETC processing methods should include neighboring points to the peak in the incidence angle estimation for higher accuracy.

The Polar ETC is very sensitive to slight errors in microphone positioning (whether by physical positioning errors or deviations in the acoustic center versus frequency). Small errors in positioning can result in an increase in errors by 35%.

The Polar ETC is also rather sensitive to signal processing variations. It is better to use the ETC, despite its flaws, rather than using the SIR. As stated earlier, further research needs to be done to investigate the reasons behind this last issue. The authors propose the idea that the ETC is less sensitive to digital sampling errors in peak detection that the SIR is (since the ETC peaks are broader in time), hence the smaller errors in the ETC versus the SIR.

Bibliography

[1] B. Gover, "Directional Measurement of Airborne Sound Transmission Paths Using a Spherical Microphone Array," Journal of the Audio Engineering Society **53,** 787–795 (2005).

[2] F. Massa and D. Massa, "Directional Energy Receiving Systems for use in the Indication of the Direction of Arrival of the Received Signal,", US Patent 5729507, 1988.

[3] C. Noël, V. Planeaua, and D. Habault, "A New Temporal Method for the Identification of Source Directions in a Reverberant Hall," Journal of Sound and Vibration **296,** 518–538 (2006).

[4] Y. Yamasaki and T. Itow, "Measurement of Spatial Information in Sound Fields by Closely Located Four Point Microphone Method," Journal of the Acoustical Society of Japan **10,** 101–110 (1989).

[5] K. Sekiguchi, "Analysis of Sound Fields on Spatial Information Using a 4-Channel Microphone System Based on a Regular Tetrahedron Peak Point Method," Applied Acoustics **37,** 305–323 (1992).

[6] C. Choi, "Measurement of Early Reflections in a Room with Five Microphone System," IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences **E86-A,** 3283–3287 (2003).

[7] F. Becker, "The Polar Energy Time Curve," In *6th International Conference of the Audio Engineering Society*, pp. 54–57 (Audio Engineering Society, 1988).

[8] R. Heyser, "Determination of Loudspeaker Arrival Times, Part I," Journal of the Audio Engineering Society **19,** 734–743 (1971).

[9] R. Heyser, "Determination of Loudspeaker Arrival Times, Part II," Journal of the Audio Engineering Society **19,** 829–834 (1971).

[10] R. Heyser, "Determination of Loudspeaker Arrival Times, Part III," Journal of the Audio Engineering Society **19,** 902–905 (1971).

[11] R. Heyser, "Instantaneous Intensity," In *81st Convention*, (Audio Engineering Society, 1986).

[12] D. Keele, "The Analytic Impulse and the Energy-Time Curve: The Debate Continues," In *93rd Convention*, (Audio Engineering Society, 1992).

[13] J. Vanderkooy and S. Lipshitz, "Uses and Abuses of the Energy-Time Curve," Journal of the Audio Engineering Society **38,** 819–836 (1990).

[14] P. D'Antonio, J. Konnert, F. Becker, and C. Bilello, "Sound Intensity and Interaural Cross-Correlation Measurements Using Time-Delay Spectrometry," Journal of the Audio Engineering Society **37,** 659–673 (1989).

[15] B. T. Thornock, Master's thesis, Brigham Young University, 2009.

[16] P. Berens, "CircStat: A MATLAB Toolbox for Circular Statistics," Journal of Statistical Software **31,** 1–21 (2009).

Appendix A

MATLAB Code

A.1 Time-Alignment

```
clc; close all; clear all;

set(0,'DefaultAxesFontName','Arial');
set(0,'DefaultAxesFontSize',16);
set(0,'DefaultAxesFontWeight','demi')
set(0,'DefaultAxesLineWidth',2);
set(0,'DefaultLineLineWidth',2);
set(0,'DefaultLineMarkersize',8);

% Load the unaligned data.
microphone='AKG 414\Subcardioid';
date='Oct 10 2009';
etc='';
exp='_Exp1';

Front=load(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Front',etc,exp]);
Front=Front.data(:,2);
Back=load(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Back',etc,exp]);
Back=Back.data(:,2);
Right=load(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Right',etc,exp]);
Right=Right.data(:,2);
Left=load(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Left',etc,exp]);
Left=Left.data(:,2);
Up=load(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Up',etc,exp]);
Up=Up.data(:,2);
```

```
Down=load(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Down',etc,exp]);
t=Down.data(:,1);
Down=Down.data(:,2);
clear data;

% Set the sampling frequency and determine the "omnidirectional" response
% at the measurement point.
FS=192000;
% W=1/3*(Front+Back+Left+Right+Up+Down);
W=abs(hilbert(1/3*(Front+Back+Left+Right+Up+Down)));

% Allow the user to determine the feature that determines how the unaligned
% data will be aligned.
disp('Select a range to examine and threshold')

% scsz=get(0,'ScreenSize');
figure(1)
% set(gcf,'Position',scsz);
plot(t,W)
title('Pressure Impulse Response')
xlabel('Time (s)')
ylabel('Magnitude')
ylim([1.3*min(W) 1.3*max(W)])
xlim([min(t) max(t)/8])

% time=ginput(2);
time=[t(1250) t(1400)];

tpremod=time(1):1/FS:time(2);
Originpremod=W(round(time(1)*FS):round(time(1)*FS)+length(tpremod)-1);

% Zoomind=input('Do you want to zoom in any more?  1/0 ');
Zoomind=0;

figure(1)
% set(gcf,'Position',[0,50,scsz(3),scsz(4)-150]);
plot(tpremod,Originpremod)
title('Zoomed Plot of the Origin IR')
xlabel('Time (s)')
ylabel('Magnitude')
ylim([1.1*min(Originpremod) 1.1*max(Originpremod)])
clc;

while Zoomind≠0
    time=ginput(2);
    tpremod=time(1):1/FS:time(2);
    Originpremod=W(round(time(1)*FS):round(time(1)*FS)+length(tpremod)-1);

    figure(1)
    set(gcf,'Position',[0,50,scsz(3),scsz(4)-150]);
```

```
    plot(tpremod,Originpremod)
    title('Zoomed Plot of the Origin IR')
    xlabel('Time (s)')
    ylabel('Magnitude')
    ylim([1.1*min(Originpremod) 1.1*max(Originpremod)])
    clc;
    Zoomind=input('Do you want to zoom in any more?  1/0 ');
end

t1ind=find(t≥time(1)-1/(2*FS)&t≤time(1)+1/(2*FS));
t2ind=find(t≥time(2)-1/(2*FS)&t≤time(2)+1/(2*FS));
timeind=t1ind:t2ind;

% Plot the unaligned impulse response and energy time curve.
figure(1)
% set(gcf,'Position',[0,50,scsz(3),scsz(4)-150]);
subplot(7,12,[1:6 13:18 25:30])
plot(t(timeind)*10^3,Front(timeind),'b-',...
    t(timeind)*10^3,Right(timeind),'k--',t(timeind)*10^3,Down(timeind),'r-.')
axis([min(t(timeind))*10^3 max(t(timeind))*10^3 ...
    1.1*min([min(Front(timeind)) min(Right(timeind)) min(Down(timeind))]) ...
    1.1*max([max(Front(timeind)) max(Right(timeind)) max(Down(timeind))])]);
text(t(timeind(1)+5)*10^3,...
    .9*max([max(Front(timeind)) max(Right(timeind)) max(Down(timeind))]),...
    '(a)','FontSize',18,'HorizontalAlignment','Left');
set(gca,'XTickLabel',{6.6 6.8 7 ''});
% title('Unaligned')
% xlabel('Impulse Response')

% figure
% set(gcf,'Position',[0,50,scsz(3),scsz(4)-150]);
% plot(t(timeind)*10^3,Front(timeind),'b-',...
%     t(timeind)*10^3,Right(timeind),'k--',t(timeind)*10^3,Up(timeind),'r-.')
% axis([min(t(timeind))*10^3 max(t(timeind))*10^3 ...
%     1.1*min([min(Front(timeind)) min(Right(timeind)) min(Up(timeind))]) ...
%     1.1*max([max(Front(timeind)) max(Right(timeind)) max(Up(timeind))])]);
% legend('+x','+y','+z')
% title('Unaligned Impulse Response')
% xlabel('Time (ms)')
% ylabel('Amplitude')

% Find the maximum value in the graph feature chosen above.

% [temp,Frind]=max(Front(timeind));
% [temp,Baind]=max(Back(timeind));
% [temp,Leind]=max(Left(timeind));
% [temp,Riind]=max(Right(timeind));
% [temp,Uind]=max(Up(timeind));
% [temp,Doind]=max(Down(timeind));

[temp,Frind]=max(abs(hilbert(Front(timeind))));
```

```
[temp,Baind]=max(abs(hilbert(Back(timeind))));
[temp,Leind]=max(abs(hilbert(Left(timeind))));
[temp,Riind]=max(abs(hilbert(Right(timeind))));
[temp,Uind]=max(abs(hilbert(Up(timeind))));
[temp,Doind]=max(abs(hilbert(Down(timeind))));
clear temp;

% Make the index above correspond to the index in the un-windowed file.
Frind=Frind+t1ind-1;
Baind=Baind+t1ind-1;
Leind=Leind+t1ind-1;
Riind=Riind+t1ind-1;
Uind=Uind+t1ind-1;
Doind=Doind+t1ind-1;

% Find the average peak value--this will be the new peak value for all
% measurements.
avgind=round(1/6*(Frind+Baind+Leind+Riind+Uind+Doind));

% Translate each measurement forward or backward based on whether its peak
% value is less or more than the average peak value.
if Frind-avgind<0
    Front(end+1:end-Frind+avgind)=0;
    for ind=length(Front):-1:avgind-Frind+1
        Front(ind)=Front(ind+Frind-avgind);
    end
    Front(end+Frind-avgind+1:end)=[];
else
    for ind=1:length(Front)-Frind+avgind
        Front(ind)=Front(ind+Frind-avgind);
    end
end

% Old way of time-alignment, based on peak value

% if Baind-avgind<0
%     Back(end+1:end-Baind+avgind)=0;
%     for ind=length(Back):-1:avgind-Baind+1
%         Back(ind)=Back(ind+Baind-avgind);
%     end
%     Back(end+Baind-avgind+1:end)=[];
% else
%     for ind=1:length(Back)-Baind+avgind
%         Back(ind)=Back(ind+Baind-avgind);
%     end
% end
% if Leind-avgind<0
%     Left(end+1:end-Leind+avgind)=0;
%     for ind=length(Left):-1:avgind-Leind+1
%         Left(ind)=Left(ind+Leind-avgind);
%     end
```

```
%     Left(end+Leind-avgind+1:end)=[];
% else
%     for ind=1:length(Left)-Leind+avgind
%         Left(ind)=Left(ind+Leind-avgind);
%     end
% end
% if Riind-avgind<0
%     Right(end+1:end-Riind+avgind)=0;
%     for ind=length(Right):-1:avgind-Riind+1
%         Right(ind)=Right(ind+Riind-avgind);
%     end
%     Right(end+Riind-avgind+1:end)=[];
% else
%     for ind=1:length(Right)-Riind+avgind
%         Right(ind)=Right(ind+Riind-avgind);
%     end
% end
% if Uind-avgind<0
%     Up(end+1:end-Uind+avgind)=0;
%     for ind=length(Up):-1:avgind-Uind+1
%         Up(ind)=Up(ind+Uind-avgind);
%     end
%     Up(end+Uind-avgind+1:end)=[];
% else
%     for ind=1:length(Up)-Uind+avgind
%         Up(ind)=Up(ind+Uind-avgind);
%     end
% end
% if Doind-avgind<0
%     Down(end+1:end-Doind+avgind)=0;
%     for ind=length(Down):-1:avgind-Doind+1
%         Down(ind)=Down(ind+Doind-avgind);
%     end
%     Down(end+Doind-avgind+1:end)=[];
% else
%     for ind=1:length(Down)-Doind+avgind
%         Down(ind)=Down(ind+Doind-avgind);
%     end
% end

% New way of time-alignment, based on cross-correlation
Frontfmod=fft(Front);
Backfmod=fft(Back);
Leftfmod=fft(Left);
Rightfmod=fft(Right);
Upfmod=fft(Up);
Downfmod=fft(Down);

Sff=abs(Frontfmod).^2;
Sfb=conj(Frontfmod).*Backfmod;
Sfl=conj(Frontfmod).*Leftfmod;
```

```
Sfr=conj(Frontfmod).*Rightfmod;
Sfu=conj(Frontfmod).*Upfmod;
Sfd=conj(Frontfmod).*Downfmod;

Rff=ifftshift(ifft(Sff));
Rfb=ifftshift(ifft(Sfb));
Rfl=ifftshift(ifft(Sfl));
Rfr=ifftshift(ifft(Sfr));
Rfu=ifftshift(ifft(Sfu));
Rfd=ifftshift(ifft(Sfd));

txcorr=-length(Rff)/2/FS:1/FS:(length(Rff)/2-1)/FS;

[temp,frind]=max(abs(Rff));
[temp,baind]=max(abs(Rfb));
[temp,leind]=max(abs(Rfl));
[temp,riind]=max(abs(Rfr));
[temp,uind]=max(abs(Rfu));
[temp,doind]=max(abs(Rfd));

% corrind=frind-round(length(timeind)/2):frind+round(length(timeind)/2);
corrind=frind-25:frind+25;

subplot(7,12,[49:54 61:66 73:78])
plot(txcorr(corrind)*10^3,abs(Rff(corrind))/max(abs(Rff(corrind))),'b-',...
    txcorr(corrind)*10^3,abs(Rfr(corrind))/max(abs(Rfr(corrind))),'k--',...
    txcorr(corrind)*10^3,abs(Rfd(corrind))/max(abs(Rfd(corrind))),'r-.')
% title('Energy Time Curve')
axis([min(txcorr((corrind)))*10^3 max(txcorr((corrind)))*10^3 0 1.1]);
text(txcorr(corrind(1)+2)*10^3,.975,'(b)','FontSize',18,'HorizontalAlignment','Left'

while baind≠frind
    if baind-frind<0
        Back(end+1:end-baind+frind)=0;
        for ind=length(Back):-1:frind-baind+1
            Back(ind)=Back(ind+baind-frind);
        end
        Back(end+baind-frind+1:end)=[];
    else
        for ind=1:length(Back)-baind+frind
            Back(ind)=Back(ind+baind-frind);
        end
    end
    Backfmod=fft(Back);
    Sfb=conj(Frontfmod).*Backfmod;
    Rfb=ifftshift(ifft(Sfb));
    [temp,baind]=max(abs(Rfb));
    baind-frind
end

while leind≠frind
```

```
    if leind-frind<0
        Left(end+1:end-leind+frind)=0;
        for ind=length(Left):-1:frind-leind+1
            Left(ind)=Left(ind+leind-frind);
        end
        Left(end+leind-frind+1:end)=[];
    else
        for ind=1:length(Left)-leind+frind
            Left(ind)=Left(ind+leind-frind);
        end
    end
    Leftfmod=fft(Left);
    Sfl=conj(Frontfmod).*Leftfmod;
    Rfl=ifftshift(ifft(Sfl));
    [temp,leind]=max(abs(Rfl));
    leind-frind
end

while riind~=frind
    if riind-frind<0
        Right(end+1:end-riind+frind)=0;
        for ind=length(Right):-1:frind-riind+1
            Right(ind)=Right(ind+riind-frind);
        end
        Right(end+riind-frind+1:end)=[];
    else
        for ind=1:length(Right)-riind+frind
            Right(ind)=Right(ind+riind-frind);
        end
    end
    Rightfmod=fft(Right);
    Sfr=conj(Frontfmod).*Rightfmod;
    Rfr=ifftshift(ifft(Sfr));
    [temp,riind]=max(abs(Rfr));
    riind-frind
end

while uind~=frind
    if uind-frind<0
        Up(end+1:end-uind+frind)=0;
        for ind=length(Up):-1:frind-uind+1
            Up(ind)=Up(ind+uind-frind);
        end
        Up(end+uind-frind+1:end)=[];
    else
        for ind=1:length(Up)-uind+frind
            Up(ind)=Up(ind+uind-frind);
        end
    end
    Upfmod=fft(Up);
    Sfu=conj(Frontfmod).*Upfmod;
```

```
    Rfu=ifftshift(ifft(Sfu));
    [temp,uind]=max(abs(Rfu));
    uind−frind
end

while doind≠frind
    if doind−frind<0
        Down(end+1:end−doind+frind)=0;
        for ind=length(Down):−1:frind−doind+1
            Down(ind)=Down(ind+doind−frind);
        end
        Down(end+doind−frind+1:end)=[];
    else
        for ind=1:length(Down)−doind+frind
            Down(ind)=Down(ind+doind−frind);
        end
    end
    Downfmod=fft(Down);
    Sfd=conj(Frontfmod).*Downfmod;
    Rfd=ifftshift(ifft(Sfd));
    [temp,doind]=max(abs(Rfd));
    doind−frind
end

Backfmod2=fft(Back);
Leftfmod2=fft(Left);
Rightfmod2=fft(Right);
Upfmod2=fft(Up);
Downfmod2=fft(Down);

Sfb2=Frontfmod.*conj(Backfmod);
Sfl2=Frontfmod.*conj(Leftfmod);
Sfr2=Frontfmod.*conj(Rightfmod);
Sfu2=Frontfmod.*conj(Upfmod);
Sfd2=Frontfmod.*conj(Downfmod);

Rfb2=ifftshift(ifft(Sfb));
Rfl2=ifftshift(ifft(Sfl));
Rfr2=ifftshift(ifft(Sfr));
Rfu2=ifftshift(ifft(Sfu));
Rfd2=ifftshift(ifft(Sfd));

figure(1)
% Plot the aligned impulse response and energy time curve.
subplot(7,12,[7:12 19:24 31:36])
plot(t(timeind)*10^3,Front(timeind),'b−',...
    t(timeind)*10^3,Right(timeind),'k−−',t(timeind)*10^3,Down(timeind),'r−.')
axis([min(t(timeind))*10^3 max(t(timeind))*10^3 ...
    1.1*min([min(Front(timeind)) min(Right(timeind)) min(Down(timeind))]) ...
    1.1*max([max(Front(timeind)) max(Right(timeind)) max(Down(timeind))])]);
text(t(timeind(end)−5)*10^3,...
```

```
    .9*max([max(Front(timeind)) max(Right(timeind)) max(Down(timeind))]),...
    '(c)','FontSize',18,'HorizontalAlignment','Right');
% title('Aligned')
% xlabel('Impulse Response')
set(gca,'XTickLabel',{'' 6.8 7 7.2},'YTickLabel',{});

subplot(7,12,[55:60 67:72 79:84])
plot(txcorr(corrind)*10^3,abs(Rff(corrind))/max(abs(Rff(corrind))),'b-',...
    txcorr(corrind)*10^3,abs(Rfr2(corrind))/max(abs(Rfr2(corrind))),'k--',...
    txcorr(corrind)*10^3,abs(Rfd2(corrind))/max(abs(Rfd2(corrind))),'r-.')
% title('Energy Time Curve')
axis([min(txcorr((corrind)))*10^3 max(txcorr((corrind)))*10^3 0 1.1]);
text(txcorr(corrind(end)-2)*10^3,.975,'(d)','FontSize',18,'HorizontalAlignment','Righ
% title('Energy Time Curve')
set(gca,'YTickLabel',{});
leg=legend('+x','+y','-z');
legplot=subplot(7,12,42:43);
pos=get(legplot,'position');
set(leg,'position',pos)
axis off
[ax1,h1]=suplabel('Time (ms)');
[ax2,h2]=suplabel('Amplitude','y');
% [ax4,h3]=suplabel('Alignment Process','t');
set(h1,'FontSize',18)
set(h2,'FontSize',18)

% figure
% % set(gcf,'Position',[0,50,scsz(3),scsz(4)-150]);
% plot(t(timeind)*10^3,Front(timeind),'b-',...
%     t(timeind)*10^3,Right(timeind),'k--',t(timeind)*10^3,Up(timeind),'r-.')
% axis([min(t(timeind))*10^3 max(t(timeind))*10^3 ...
%     1.1*min([min(Front(timeind)) min(Right(timeind)) min(Up(timeind))]) ...
%     1.1*max([max(Front(timeind)) max(Right(timeind)) max(Up(timeind))])]);
% legend('Front','Right','Up')
% title('Aligned Impulse Response')
% xlabel('Time (ms)')
% ylabel('Amplitude')

% clc;
abs(343/192000*[frind-frind baind-frind riind-frind ...
    leind-frind uind-frind doind-frind])

figure
% set(gcf,'Position',[0,50,scsz(3),scsz(4)-150]);
plot(t(timeind)*10^3,Front(timeind)/max(abs(Front(timeind))),'b-',...
    t(timeind)*10^3,Front(timeind).^2/max(Front(timeind).^2),'k--',...
    t(timeind)*10^3,abs(hilbert(Front(timeind)))/max(abs(hilbert(Front(timeind)))),'r
legend('IR','SIR','ETC')
hold on
plot(t(timeind)*10^3,zeros(1,length(t(timeind))),'k-','Linewidth',1)
hold off
```

```
xlim([min(t(timeind))*10^3 max(t(timeind))*10^3]);
ylim([1.1*min(Front(timeind))/max(abs(Front(timeind))) 1.1])
% title('Typical Impulse Response')
xlabel('Time (ms)','FontSize',18)
ylabel('Normalized Amplitude','FontSize',18)

% Save the aligned data files for future use.
data=[t,Front];
save(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Front',etc,exp,'_ETCAlignCC'],'data');
data=[t,Back];
save(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Back',etc,exp,'_ETCAlignCC'],'data');
data=[t,Left];
save(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Left',etc,exp,'_ETCAlignCC'],'data');
data=[t,Right];
save(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Right',etc,exp,'_ETCAlignCC'],'data');
data=[t,Up];
save(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Up',etc,exp,'_ETCAlignCC'],'data');
data=[t,Down];
save(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Down',etc,exp,'_ETCAlignCC'],'data');
```

A.2 Polar ETC

```
clear all; close all; clc;

% Define the speed of sound in the fluid.
c=343;

% The coordinate system used here is rotated 90 degrees from that
% defined in Becker's patent and in the D'Antonio paper (i.e. front is +X)
microphone='AKG 414\Hypercardioid';
date='Oct 10 2009';
etc='';
freq='';
exp='_Exp2';
align='_ETCAlign';

% First, load the data files that will be used in the algorithm.
Front=load(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Front',etc,freq,exp,align]);
Front=Front.data(:,2);
Back=load(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Back',etc,freq,exp,align]);
Back=Back.data(:,2);
Right=load(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Right',etc,freq,exp,align]);
```

```
Right=Right.data(:,2);
Left=load(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Left',etc,freq,exp,align]);
Left=Left.data(:,2);
Up=load(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Up',etc,freq,exp,align]);
Up=Up.data(:,2);
Down=load(['Z:\Dr. Leishman\James Esplin\Experimental Data\',...
    date,'\',microphone,'\Down',etc,freq,exp,align]);
t=Down.data(:,1);
Down=Down.data(:,2);
clear data microphone date etc exp align freq;
Filename='Shure SM81 Filtered Exp1 Direct Sound';

% Define where the reflection of interest should be located.
Allocphi=45;
Allocthe=-43.5;

% Define the sampling frequency of the data.
FS=192000;

% Truncate the data so as to make processing faster.
Front=Front(1:2^17);
Back=Back(1:2^17);
Left=Left(1:2^17);
Right=Right(1:2^17);
Up=Up(1:2^17);
Down=Down(1:2^17);
t=t(1:2^17);

% Construct the sum of the measurements to obtain an omnidirectional
% response at the measurement point.
% W=(1/3*(Front+Back+Left+Right+Up+Down)).^2;
W=abs(hilbert(1/3*(Front+Back+Left+Right+Up+Down)));
% W=real(W);

% Construct both the ETC and SIR of all data.
Fsq=abs(Front).^2;
Bsq=abs(Back).^2;
Lsq=abs(Left).^2;
Rsq=abs(Right).^2;
Usq=abs(Up).^2;
Dsq=abs(Down).^2;

FETC=abs(hilbert(Front));
BETC=abs(hilbert(Back));
LETC=abs(hilbert(Left));
RETC=abs(hilbert(Right));
UETC=abs(hilbert(Up));
DETC=abs(hilbert(Down));

```

```
clear Front Back Up Down Left Right;

% Define the "energy density" E from the D'Antonio paper to be able to use
% the direction cosines u,v and w and therefore extract the directional
% information.

E=1/2*sqrt((Fsq-Bsq).^2+(Rsq-Lsq).^2+(Usq-Dsq).^2);
ETC=1/2*sqrt((FETC-BETC).^2+(RETC-LETC).^2+(UETC-DETC).^2);

% Pick the specified peak of interest to examine.
format long e
disp('Select a range to examine and threshold')

plot(t,W)
title('Pressure Impulse Response')
xlabel('Time (s)')
ylabel('Magnitude')
xlim([min(t) max(t)/8])
[time,amp]=ginput(2);
tpremod=time(1):1/FS:time(2);

%Make the two vectors the same length
Originpremod=W(round(time(1)*FS):round(time(1)*FS)+length(tpremod)-1);
close all; clc;
clear amp; %clear time;

% Find the index of the peak chosen previously.
threshind=find(W==max(Originpremod));

u=zeros(length(W),1);
v=zeros(length(W),1);
w=zeros(length(W),1);
theta=370*ones(length(W),1);
phi=370*ones(length(W),1);
timeend=zeros(length(W),1);

% Calculate the direction of arrival for the peak value and 3 indices on
% either side of it.
for l=threshind-3:threshind+3
    if l~=threshind

        % Find the direction cosines using the energy time curve and
        % squared impulse responses since energy is proportional to
        % pressure squared.

%           u(l)=(Rsq(l)-Lsq(l))/(2*E(l));
%           v(l)=(Fsq(l)-Bsq(l))/(2*E(l));
%           w(l)=(Usq(l)-Dsq(l))/(2*E(l));

        u(l)=(RETC(l)-LETC(l))/(2*ETC(l));
        v(l)=(FETC(l)-BETC(l))/(2*ETC(l));
```

```
        w(l)=(UETC(l)-DETC(l))/(2*ETC(l));

        %Find the theta, phi and time values for the reflections.  Display
        %them so that one can locate the peak on the graph using time and
        %then find the offending surface using theta and phi where theta is
        %the elevation from -90 to +90 degrees and phi ranges from -180 to
        %180 degrees, as is the physics convention

        theta(l)=asind(w(l)/sqrt(u(l).^2+v(l).^2+w(l).^2));
        phi(l)=180/pi*atan2(-u(l),v(l));

        timeend(l)=l/FS;

    end
    if l==threshind

%          upeak=(Rsq(l)-Lsq(l))/(2*E(l));
%          vpeak=(Fsq(l)-Bsq(l))/(2*E(l));
%          wpeak=(Usq(l)-Dsq(l))/(2*E(l));

        upeak=(RETC(l)-LETC(l))/(2*ETC(l));
        vpeak=(FETC(l)-BETC(l))/(2*ETC(l));
        wpeak=(UETC(l)-DETC(l))/(2*ETC(l));

        thetapeak=asind(wpeak/sqrt(upeak^2+vpeak^2+wpeak^2));
        phipeak=180/pi*atan2(-upeak,vpeak);

    end
end

% Remove all the entries in phi and theta that correspond to samples that
% are not not in the interval of interest.
indextheta=find(theta==370);
indexphi=find(phi==370);
theta(indextheta)=[];
phi(indexphi)=[];

format short

% Find the angular estimation error for the peak value and the average
% value as well as the standard deviation of the seven points.
pkerror=acosd(sind(90-abs(phipeak-Allocphi))*...
    sind(90-abs(thetapeak-Allocthe)));
avgsrc=[circ_mean([phi;phipeak]*pi/180) ...
    circ_mean([theta;thetapeak]*pi/180)]*180/pi;
avgerror=acosd(sind(90-abs(avgsrc(1)-Allocphi))*...
    sind(90-abs(avgsrc(2)-Allocthe)));
stdsrc=[circ_std([phi*pi/180;phipeak*pi/180]) ...
    circ_std([theta*pi/180;thetapeak*pi/180])]*180/pi;
stderror=acosd(sind(90-stdsrc(1))*sind(90-stdsrc(2)));
Error=[pkerror avgerror stderror]
```

```

% Display a scatter plot with the actual source location, peak source
% location, average source location and all source locations.
close all;
figure(1)
scatter(phi,theta,25,'b.')
hold on;
scatter(phipeak,thetapeak,50,'r');
hold on;
scatter(avgsrc(1),avgsrc(2),50,'gs');
hold on;
scatter(Allocphi,Allocthe,100,'kp')
hold off;
title('Image Source Locations');
xlabel('\phi (degrees)');
ylabel('\theta (degrees)');
legend('All Sources','Peak Source','Average Source','Actual Source',...
    'location','NorthWest')
xlim([-180 180])
ylim([-90 90])
set(gcf,'position',[5 35 1275 915])
```

A.3 Circular Statistics

This code is courtesy of [16].

A.3.1 Angular Mean

```
function [mu ul ll] = circ_mean(alpha, w, dim)
%
% mu = circ_mean(alpha, w)
%   Computes the mean direction for circular data.
%
%   Input:
%     alpha sample of angles in radians
%     [w        weightings in case of binned angle data]
%     [dim  compute along this dimension, default is 1]
%
%     If dim argument is specified, all other optional arguments can be
%     left empty: circ_mean(alpha, [], dim)
%
%   Output:
%     mu        mean direction
%     ul    upper 95% confidence limit
%     ll    lower 95% confidence limit
%
% PHB 7/6/2008
%
% References:
%   Statistical analysis of circular data, N. I. Fisher
```

```
%   Topics in circular statistics, S. R. Jammalamadaka et al.
%   Biostatistical Analysis, J. H. Zar
%
% Circular Statistics Toolbox for Matlab

% By Philipp Berens, 2009
% berens@tuebingen.mpg.de - www.kyb.mpg.de/¬berens/circStat.html

if nargin < 3
  dim = 1;
end

if nargin < 2 || isempty(w)
  % if no specific weighting has been specified
  % assume no binning has taken place
    w = ones(size(alpha));
else
  if size(w,2) ≠ size(alpha,2) || size(w,1) ≠ size(alpha,1)
    error('Input dimensions do not match');
  end
end

% compute weighted sum of cos and sin of angles
r = sum(w.*exp(1i*alpha),dim);

% obtain mean by
mu = angle(r);

% confidence limits if desired
if nargout > 1
  t = circ_confmean(alpha,0.05,w);
  ul = mu + t;
  ll = mu - t;
end
```

A.3.2 Mean Resultant Vector Length

```
function r = circ_r(alpha, w, d, dim)
% r = circ_r(alpha, w, d)
%   Computes mean resultant vector length for circular data.
%
%   Input:
%     alpha sample of angles in radians
%     [w        number of incidences in case of binned angle data]
%     [d    spacing of bin centers for binned data, if supplied
%           correction factor is used to correct for bias in
%           estimation of r, in radians (!)]
%     [dim  compute along this dimension, default is 1]
%
%     If dim argument is specified, all other optional arguments can be
%     left empty: circ_r(alpha, [], [], dim)
%
```

```
%   Output:
%     r     mean resultant length
%
% PHB 7/6/2008
%
% References:
%   Statistical analysis of circular data, N.I. Fisher
%   Topics in circular statistics, S.R. Jammalamadaka et al.
%   Biostatistical Analysis, J. H. Zar
%
% Circular Statistics Toolbox for Matlab

% By Philipp Berens, 2009
% berens@tuebingen.mpg.de — www.kyb.mpg.de/¬berens/circStat.html

if nargin < 4
  dim = 1;
end

if nargin < 2 || isempty(w)
  % if no specific weighting has been specified
  % assume no binning has taken place
    w = ones(size(alpha));
else
  if size(w,2) ≠ size(alpha,2) || size(w,1) ≠ size(alpha,1)
    error('Input dimensions do not match');
  end
end

if nargin < 3 || isempty(d)
  % per default do not apply correct for binned data
  d = 0;
end

% compute weighted sum of cos and sin of angles
r = sum(w.*exp(1i*alpha),dim);

% obtain length
r = abs(r)./sum(w,dim);

% for data with known spacing, apply correction factor to correct for bias
% in the estimation of r (see Zar, p. 601, equ. 26.16)
if d ≠ 0
  c = d/2/sin(d/2);
  r = c*r;
end
```

A.3.3 Circular Standard Deviation

```
function [s s0] = circ_std(alpha, w, d, dim)
% s = circ_std(alpha, w, d, dim)
%   Computes circular standard deviation for circular data
```

```
%   (equ. 26.20, Zar).
%
%   Input:
%     alpha sample of angles in radians
%     [w         weightings in case of binned angle data]
%     [d    spacing of bin centers for binned data, if supplied
%           correction factor is used to correct for bias in
%           estimation of r]
%     [dim  compute along this dimension, default is 1]
%
%     If dim argument is specified, all other optional arguments can be
%     left empty: circ_std(alpha, [], [], dim)
%
%   Output:
%     s     angular deviation
%     s0    circular standard deviation
%
% PHB 6/7/2008
%
% References:
%   Biostatistical Analysis, J. H. Zar
%
% Circular Statistics Toolbox for Matlab

% By Philipp Berens, 2009
% berens@tuebingen.mpg.de - www.kyb.mpg.de/¬berens/circStat.html

if nargin < 4
  dim = 1;
end

if nargin < 3 || isempty(d)
  % per default do not apply correct for binned data
  d = 0;
end

if nargin < 2 || isempty(w)
  % if no specific weighting has been specified
  % assume no binning has taken place
    w = ones(size(alpha));
else
  if size(w,2) ≠ size(alpha,2) || size(w,1) ≠ size(alpha,1)
    error('Input dimensions do not match');
  end
end

% compute mean resultant vector length
r = circ_r(alpha,w,d,dim);

s = sqrt(2*(1-r));       % 26.20
s0 = sqrt(-2*log(r));     % 26.21
```

Index