1996

Estimating azimuth from speech in a natural auditory environment

Nathaniel Dow Henderson
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Henderson, Nathaniel Dow, "Estimating azimuth from speech in a natural auditory environment" (1996).
Master's Theses. 1307.
DOI: <https://doi.org/10.31979/etd.2b9p-3jf3>
https://scholarworks.sjsu.edu/etd_theses/1307

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

ESTIMATING AZIMUTH FROM SPEECH IN A NATURAL AUDITORY ENVIRONMENT

A Thesis

Presented to

**The Faculty of the Department of Electrical Engineering
San José State University**

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Nathaniel Dow Henderson

August 1996

UMI Number: 1381418

**Copyright 1996 by
Henderson, Nathaniel Dow**

All rights reserved.

**UMI Microform 1381418
Copyright 1996, by UMI Company. All rights reserved.**

**This microform edition is protected against unauthorized
copying under Title 17, United States Code.**

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

© 1996

Nathaniel Dow Henderson

ALL RIGHTS RESERVED

APPROVED FOR THE DEPARTMENT OF
ELECTRICAL ENGINEERING

Dr. Richard O. Duda

Dr. Essam A. Marouf

Dr. Brian Holmes (Department of Physics)

APPROVED FOR THE UNIVERSITY

ABSTRACT

ESTIMATING AZIMUTH FROM SPEECH IN A NATURAL AUDITORY ENVIRONMENT

by Nathaniel D. Henderson

This thesis investigates and develops a system that enables computers to estimate the azimuth of a sound source. The system was developed with express intent of estimating the azimuth from speech in a reverberant environment. The system's design is based on a human auditory model.

The system calculates the interaural intensity difference (IID), interaural time difference (ITD), and interaural envelope delay (IED) from the output of a cochlear model. Fuzzy sets are defined to extract non-reverberant localization cues from any present sound sources. These fuzzy sets account for onsets, coincidence over time and frequency, and temporal integration. The fuzzy weighted IID, ITD, and IED calculations are combined, and a maximum-likelihood criterion is used to produce a probability map of the perceived azimuth for the system.

Experiments with 60 samples of speech and other sounds showed that the system performs with an error rate equivalent to that of humans.

Acknowledgments

I am greatly indebted to Professor Richard Duda for his guidance, enthusiasm, and patience. None of this thesis would have been possible without his support. I am truly fortunate to have had the opportunity and privilege of working with such a great scientist and teacher.

Many thanks are also due to Phil Brown, my partner in crime, whose debugging ability and extensive knowledge of 3D audio greatly assisted me in my work. He was also instrumental in helping me with the test recordings. He is "the source" of the phoneme tests.

I would like to thank Malcom Slaney of Interval Research Corporation for allowing me to use his gammatone cochlear model, and Richard Lyon of Apple Computer for the loan of KEMAR.

I would also like to thank the members of my thesis committee for taking time out of their busy schedules to review my work.

This work was financially supported by the Interactive Systems Program of the National Science Foundation under grant number IRI-9402246, with additional equipment support from Apple Computer, Inc.

Finally and above all, I wish to thank my wife Carmen and my daughter Ana, who have showered me with love, support and happiness. This thesis is dedicated to them.

Table of Contents

1 Introduction

1.1 Why Sound Localization?	1
1.2 Localization Cues	2
1.3 The Head-Related Transfer Function (HRTF)	7
1.4 Simple Localization Systems	9
1.5 Goals in Sound Localization	12

2 System Design

2.1 Confronting Limitations in Sound Localization	15
2.1.1 The Precedence Effect	15
2.1.2 Coincidence in Frequency and Time	17
2.1.3 Temporal Integration	18
2.2 System Overview	20
2.3 System Components	24
2.3.1 The Cochlea	24
2.3.2 Low-Frequency Localization: ITD	32
2.3.3 High-Frequency Localization: IED	44
2.3.4 High-Frequency Localization: IID	46
2.3.5 Combining Probability Maps	52

3 Testing and Evaluation

3.1 Walking Through a Single Test	58
3.2 Word Tests	63
3.3 Azimuth Tests	68
3.4 Test with Two Sources	72
3.5 Reflection Suppression	75
3.6 The Frassen Effect	77

4 Conclusions	80
----------------------	-----------

References	83
-------------------	-----------

Appendix A - The Code

A.1 cochlea	85
A.2 find_peaks	87
A.3 localizeITD	89
A.4 ITD_maps_to_prob	97
A.5 Notes for localizeIED	98
A.6 localizeIID	98
A.7 combine_probs.	104
A.8 Filter Coefficients	106

Chapter 1

Introduction

1.1 Why Sound Localization?

Our ability to localize a sound source is one of the most important sources of information we have to understand the objects and events in our environment. When an event produces a sound we are able to localize this event in space and time. This process is so automatic that we often take our sound-localization ability for granted, but if we were unable to localize sounds our perception of events in our environment would be greatly compromised. Our ability to localize sound is also connected with our ability to suppress echoes [1]. This ability to suppress echoes greatly enhances our understanding of sound sources. Therefore sound-localization in reverberant environments is a fundamental part of human hearing.

Since sound-localization is essential to the hearing process, it has inspired research for centuries [6]. The study of sound localization crosses many traditional academic fields; biologists, psychologists, and engineers all have interest in this field. Biologists wish to understand the physical mechanisms in the neural paths from the ear to the brain, and in particular how these mechanisms allow us to localize sounds. Given that we do possess this ability, psychologists would like to know exactly how we use this information to perceive our environment, since this perception will modify our behavior. Engineers are

very interested in building systems that can hear and understand as well as humans. Engineers would also like to build hearing aids that could overcome any biological deficiencies we may have in our hearing. To build better hearing aids and computers that hear, more research needs to be performed in the area of sound localization. Fortunately, much research has already been done in the area of sound localization, and today we have a good understanding of the cues that allow us to localize sounds.

As engineers attempt to build intelligent machines that can "hear," a central issue is the machines' ability to localize sounds. Since we have an understanding of sound-localization cues in humans, we wish to develop systems that can localize sounds based on these cues. It is important to note that systems that localize sound could be developed with no regard to the sound-localization cues used by humans. While these systems could localize sounds very well, it would be very difficult to incorporate them in an over-all hearing system that attempts to emulate human hearing and understanding. Due to these considerations and a desire to more fully understand human hearing, the sound-localization system developed in this thesis is based on a human hearing model.

1.2 Localization Cues

The basic cues that humans use to localize sound sources are: interaural time difference (ITD), interaural envelope difference (IED), interaural intensity

difference (IID), monaural spectral cues, and echo analysis. Each of these cues is limited in one way or another. Due to this fact, we combine all of them to estimate the location of a sound source.

The interaural time difference (ITD) is the difference in the times at which a sound arrives at each ear. Since our ears are in different locations in space, a sound will reach each ear at a different time, unless the sound source lies in the median plane. (The median plane refers to all points that are equidistant from each ear.) By measuring this time difference, we are able to localize a sound source at a particular azimuth angle. The main limitation of the ITD is that there are many points in space that give rise to the same ITD (see Figure 1). The collection of these points are referred to as a *cone of confusion*. Another limitation of the ITD is the fact that signals above ~1 kHz have wavelengths that are smaller than the distance between the ears. So for narrow-band high-frequency sounds it's impossible for humans to measure time-correlation accurately between the two ears. Therefore the ITD is an accurate sound-localization cue only for frequencies below 1 kHz. Nevertheless if a high-frequency sound source has a low-frequency envelope, like a modulated signal, this envelope can be used to extract time-correlation information for localization. In this case the timing difference is called the IED (interaural envelope difference) [3].

Although the ITD and IED can only localize sound within a particular cone of points, the interaural intensity difference (IID) varies around such a cone, giving the IID an important role in sound localization [4].

Figure 1: The cone of confusion. A sound-source placed anywhere on the cone will produce the same ITD measurements.

As long as the sound source is not located in the median plane, one ear will be closer to the source than the other. Therefore the intensity of the sound in the ear closer to the source will be greater. Even more important, different frequencies propagate around our heads and bodies in different ways. Exactly how the IID varies with frequency depends on the position of the sound source relative the head and body of the listener.

Figure 2 shows how IID information can be used to distinguish location on the cone of confusion. In the horizontal plane there are two points that intersect any particular cone of confusion, one in the front and the other in the back. A sound source at azimuth 50° or 130° will experience the same time delay, but Figure 2 clearly shows different IID values for these two locations. Therefore, the IID provides an important localization cue to help us distinguish front from back.

Figure 2: IID at azimuth = 50° and 130°, elevation = 0°

It has been shown that we use IID information to produce neural-IID maps to which we can compare incoming sound signals [5]. If an incoming sound matches one of our neural-IID maps, we will have localized the sound in the horizontal and vertical planes. An important exception arises for sound sources in the median plane. In the median plane there is no significant IID, ITD, or IED. So how do we localize sounds in the median plane? This localization is achieved with monaural spectral cues and head motion [6].

When a sound source is in the median plane, we naturally have some confusion about its location, since the ITD, IED and IID are of no use. One of the easiest ways we can avoid this confusion is to simply move our head so that the sound

source is no longer in the median plane, and thus allowing us to use ITD, IED, and IID information once again. Nevertheless, even if we do not move our heads we are able to localize in the median plane due to the fact that the shapes of our outer ears (pinnae) create certain pinnae "echoes" that have a different frequency response depending on the position of the sound source [6]. Humans combine these monaural cues with the ITD, IED, and IID to enable us to locate sound sources in two dimensions.

Our ability to determine range of a sound source is the least understood of our sound localization abilities. There are a variety of mechanisms that are believed to play a important role in the determination of range. Two of these basic mechanisms are the following: 1) We estimate range by combining a sound's intensity along with our familiarity of the sound. For example we can be very familiar with the sound of a voice and if that familiarity tells us that the person is speaking loudly and yet our ears sense a low intensity of sound we know that person is far from us. 2) We can use an echo analysis of our auditory environment to estimate range. For example, if we hear a sound along with a great deal of echo we know that the sound source is far away. Since the first mechanism to estimate range is very qualitative it is difficult to study let alone to model. The second mechanism could be incorporated into a sound localizing system, but a great deal of research in echo analysis of real auditory environments is needed. The lack of understanding in our ability to estimate range of a sound source signifies that this hearing mechanism will probably be the last to be incorporated into a sound localizing system.

1.3 The Head-Related Transfer Function (HRTF)

Mathematically, both the binaural and monaural cues can be described by the *Head-Related Transfer Function* (HRTF). This is a transfer function that operates on the sound source to produce the signal received by the ear. The HRTF depends on frequency and the three dimensional position of the sound source relative to the ear. If $X(\omega)$ is the Fourier transform of the sound source, then the signals received by the right and left ear $Y_R(\omega)$ and $Y_L(\omega)$ are given by:

$$Y_R(\omega) = H_R(\omega, \theta, \phi, r)X(\omega)$$

$$Y_L(\omega) = H_L(\omega, \theta, \phi, r)X(\omega)$$

where

ω = frequency

θ = azimuth angle to source

ϕ = vertical angle to source

r = range to source

$H_R(\omega, \theta, \phi, r)$ = HRTF for right ear.

$H_L(\omega, \theta, \phi, r)$ = HRTF for left ear.

By taking the ratio of these two equations we obtain an interaural head-related transfer function which doesn't depend on the spectrum of the sound source.

$$\frac{Y_R(\omega)}{Y_L(\omega)} = \frac{H_R(\omega, \theta, \phi, r)}{H_L(\omega, \theta, \phi, r)} = H(\omega, \theta, \phi, r)$$

The magnitude of $H(\omega, \theta, \phi, r)$ (measured in dB) is simply the IID previously described, and the phase of $H(\omega, \theta, \phi, r)$ can be used to measure the ITD and the IED. Although the ITD and IED can be used successfully to locate the azimuth of the sound source, we must remember that there are always two positions in the horizontal plane that have the same ITD and IED. In Figure 1 sound sources on the cone of confusion have the same timing differences since the distance from each ear to any source is the same.

The binaural information given by $H(\omega, \theta, \phi, r)$ can be used successfully to estimate both azimuth and vertical location when the source is away from the median plane [7]. However, due to symmetry $H(\omega, \theta, \phi, r) \approx 1$ in the median plane, and therefore we must consider monaural cues to localize the vertical position when the azimuth is small. The monaural transfer function is given by

$$Y_M(\omega) = H_M(\omega, \theta, \phi, r)X(\omega)$$

where $Y_M = Y_L$ (or Y_R) and $H_M = H_L$ (or H_R).

By combining monaural and binaural cues it is possible to successfully locate sounds in two dimensions [8].

Although the HRTF will vary with range, it has yet to be determined exactly how $H(\omega, \theta, \phi, r)$ depends on range. Therefore until a great deal more of research is conducted on range localization it will be difficult to include this parameter in any sound localizing system. Fortunately, $H(\omega, \theta, \phi, r)$ is not sensitive to range

beyond about one meter, and we will restrict our sources from coming too close to the head.

An estimate of the sound source location can be obtained by comparing the measured HRTF data with tables of HRTF information for each azimuth and elevation. The problem of sound localization can therefore be considered as the problem of estimating the HRTF from incoming sound signals.

1.4 Simple Localization Systems

Let's consider again an input sound signal $X(\omega)$ that we wish to localize. The ratio of the signals that are received at each ear is given by

$$\frac{Y_r(\omega)}{Y_l(\omega)} = \frac{H_r(\omega, \theta, \phi)X(\omega)}{H_l(\omega, \theta, \phi)X(\omega)} = H(\omega, \theta, \phi)$$

where the dependency on range has been removed since we only wish to consider localizing in azimuth and elevation. If the input $X(\omega)$ is the transform of an impulse then $H(\omega, \theta, \phi)$ is the transform of the impulse response. By measuring the impulse response at different azimuths and elevations a table of HRTF's can be produced. A system that records signals at the two ears can take a ratio of their spectra and compare this ratio with the table of HRTF data. It is reasonable to expect that the closest match will correspond to the direction of the sound source. In other words a simple sound-localization system can be described as a system that does the following: a) records input signals at two ears, b) performs a spectrum analysis, c) extracts ITD, IED and IID information,

d) compares ITD, IED and IID extracted to tabulated HRTF data to produce a directional probability map. Such a system is shown in Figure 3.

Figure 3: A Simple Sound-Localization System

The preceding method has been used in many sound-localization systems (see Duda[4], Martin[9], or Nandy and Ben-Arie[10]) . All of these sound-localization systems have similar limitations, which the authors dutifully note, and they make suggestions on how these limitations can be overcome.

The first limitation is that these systems are limited to sound sources that are wide-band in nature. This is due to the fact that they need accurate IID information. To obtain accurate localization by comparing a measured IID to a

table of IID information it's clear that the input must be wide-band (see Figure 2). If many frequencies are absent in a sound signal the IID information extracted becomes very limited in usefulness.

Secondly, sound-localization systems designed to date assume operation in an anechoic environment. If a simple localization system were to operate in a reverberant environment it would get localization cues from all the virtual sources created by echoes. These echoes would greatly degrade the localization abilities of the system.

Finally, these simple systems assume that there is only one sound source to localize. Multiple sources, like echoes, confuse simple sound-localization systems.

Nevertheless, humans can localize on multiple, narrow-band sounds in a reverberant environment. Obviously, simply measuring the IID, IED, and ITD is not sufficient to achieve localization abilities comparable with humans.

Therefore to build a system that emulates human sound-localization ability other aspects of the human auditory system should be modeled to improve system performance.

1.5 Goals In Sound Localization

The main goal of this thesis is to design a sound-localization system that

confronts the limitations in present day sound-localization systems as mentioned in the previous section. Since accurate sound-localization has a powerful application as a front-end to speech recognition systems, the main goal of the design is to localize on speech. Also since it's desired to model the human hearing processes, a human auditory model is used for spectral analysis.

Speech is by nature a narrow-band signal with most of its energy content below a few kHz. There are spurious moments during the vowel sounds in speech in which the glottal pulses produce formant frequencies of up to 4 kHz, but even with energy content up to 4 kHz speech is still basically a narrow-band low-frequency signal. Due to this fact, the most important sound localization cue for speech is the ITD. The main problem with basing the system design on the ITD is that the timing difference can only localize on a particular cone of confusion (see Figure 1) . The IID cue is only reliable for signals above ~ 1.5 kHz, so some (though not much) IID information can be extracted from speech. Due to these difficulties, and the fact that most speech we listen to is in the horizontal plane, the design presented in this thesis attempts to localize only in the horizontal plane. This effectively reduces the problem from deciding where the sound source is on the cone of confusion to a problem of deciding between two points on the cone of confusion: front or back. Timing calculations locate a single sound sources at two points in the horizontal plane, leaving front-back confusion due to the fact that there are two points in the horizontal plane that correspond to the intersection of the cone of confusion and the horizontal plane.

Humans also experience front-back confusion [11].

Figure 4 shows the coordinate system that is used throughout this thesis. As an example of front-back confusion, a sound source placed at 40° will produce a timing delay between the right and left ears exactly the same as a sound placed at 140° . Since speech does have some frequency content above 1.5 kHz, it is hoped that there will be enough IID information to distinguish between front and back, even though the IID itself experiences a good deal of front-back confusion [7].

Figure 4: Coordinate System for Azimuth

Although speech localization potentially has many applications, it should be noted that if we design a system that can localize speech in a reverberant environment, the most important limitations of localization are addressed.

Therefore the design presented in this thesis is also applicable to the localization of many other sources of sound.

Also, one of the most difficult problems facing any auditory processing system is dealing with reverberation. In the case of sound localization, reverberation can be thought of as an infinite number of virtual sound sources. Therefore if a sound-localizing system is placed in a reverberant environment it will locate a single sound source in many different directions. So the problem of locating a sound-source in a reverberant environment becomes a problem of deciding which ITD, IED and IID measurements correspond to the source and which ones correspond to the reverberation.

In conclusion, the goal of this thesis is to design a sound-localization system based on a human auditory model that can locate multiple sound sources in the horizontal plane. The sound sources can have arbitrary frequency content, and the system should be able to operate in a noisy reverberant environment. The systems performance, since it is based on a human auditory model, should strive to perform as well as people do under similar circumstances.

Chapter 2

System Design

2.1 Confronting Limitations In Sound Localization

The main principle applied in designing the system has been to use a human auditory model as a starting point and then to engineer further solutions from the results of direct observation of its response. We know that humans do not experience the limitations in sound localization as do the simple systems described in Chapter 1. Therefore, adding more aspects of the human auditory system to a sound localization system should remove some of the previous limitations.

2.1.1 The Precedence Effect

The first aspect of the human auditory system that was modeled in this thesis was that of the precedence effect [1]. The precedence effect refers to the fact that when a listener perceives a sequence of sounds as a single auditory event, the location of that event is primarily determined by the directional cues associated with the signal that arrives first. By modeling the precedence effect we hoped that the problem of reverberation would be mostly solved. This turned out to be the case for short-duration wide-band signals (e.g., snaps, clicks or claps). Unfortunately it did not solve all the problems associated with reverberation.

The main distinction that can be made between the sound source and its early reflections is that the sound from the source will arrive before the reflections. Early reflections will typically arrive at between 3 ms and 10 ms after the source. So one can distinguish the sound source from reflections by monitoring the sound input into the system. When a sudden increase of acoustic energy is noted it's safe to assume that the first 3 ms of the signal corresponds to a sound source and that it is free from interference from reflections. Since this 3-ms portion of sound is free from reflections it is very useful in determining the source's location.

Modeling of the precedence effect essentially solves the problem of dealing with reflections as long as the sound source has a very quick onset. An onset is quick if the maximum acoustic energy is reached at about 1 to 3 ms after the beginning of the sound. Clicks, claps, knocks, and similar impulsive sounds can be localized in a reverberant environment as long as there is onset analysis incorporated in the localizing system.

Onset analysis was incorporated in this system by assigning the signal input a fuzzy weight "onsetness" at regular time intervals. If an onset occurred at a particular time the signal would be assigned large "onsetness," whereas if there was no onset at this particular time the signal would be assigned small "onsetness." Complete details of this fuzzy weight and its use are given in Section 2.3.2 . Another novel approach to modeling the precedence effect has been developed by Huang and Ohnishi [18]. Their approach was to deconvolve

the room impulse response and the signal. They have had success with their model, however we did not use their model since it made prior assumptions about the acoustic environment.

Unfortunately speech signals can take as long as 50 ms to build up enough energy to enable any acoustic measurements to be taken. Therefore any recording of speech made in a natural auditory environment is going to be mixed with reverberation. Accordingly onset weighting alone is no longer sufficient when attempting to localize speech in an reverberant environment, and other methods must be used to localize.

2.1.2 Coincidence in Frequency and Time

There is no way in which reflections can be avoided from sound sources that have an onset duration greater than 10 ms. When the ITD, IED, and IID measurements are made on such a signal, the cues will be noisy and point in all directions. Nonetheless we observed that coincidence in localization cues across various frequencies was a strong determining factor in locating a sound source. There is also biological evidence that humans use coincidence in frequency for localization [12]. More importantly, it was observed that coincidence in localization cues over time (a period of about 5 ms to 20 ms) was a very strong determining factor in locating a sound source.

Probably the best way to understand these factors of time and frequency coincidence is through example. In a typical experiment, we recorded a spoken

word binaurally in a reverberant environment. The signals were broken up into about 70 frequency bands by band-pass filtering. Now since this recording was made in a reverberant environment the ITD, IED and IID cues at any particular time during a vowel segment seemed erratic and random. Nonetheless, if the localization measurements for 5 or 6 of these bands coincided we observed that it was highly likely that these measurements corresponded to the source signal. These 5 or 6 channels were even more likely to be correct if they were in adjacent channels. This type of coincidence analysis was even more effective across time. That is, if any one channel produced consistent results for more than 10 ms we observed that it was highly likely that those results corresponded to a sound source.

The localization factors of coincidence in time and frequency are incorporated in the design by defining a set of fuzzy weights that gives preference to ITD, IED and IID measurements that coincide in frequency and time. By incorporating coincidence fuzzy weights into the system design the system was able to localize on speech signals.

2.1.3 Temporal Integration

During a typical segment of speech, there will be a few times when the localization cues are good, and many times when the localization cues are poor. To better emulate human localization ability some form of temporal integration must be incorporated into the system design. Temporal integration being

defined as the weighting of localization cues over time in such a way that good localization cues are remembered. In other words if good localization cues are available during a 50 ms portion of sound, these cues influence the cues that follow.

One of the best examples of temporal integration is known as the Frassen effect [12]. The Frassen effect is an auditory illusion that humans experience in a reverberant environment when the sound source is a very narrow-band signal (a sine wave). Sounds are played over two loudspeakers. The sound from the first speaker has a very quick onset and is quickly attenuated. At the same time the second speaker begins to play the same signal with a very long onset. While the signal in the second speaker is being increased in volume the signal in the first speaker becomes silent. Since standing waves are set up in a reverberant environment there are no good localization cues after the initial onset. When asked seconds after the initial onset where the sound is coming from, people who experience the Frassen effect will say it's coming from the first speaker even though the first speaker is silent and the second speaker is playing at full volume. This is due to the fact that the sharp onset produced by the first speaker produces strong localization cues. The standing waves in the room produce contradictory information, but people remember the best localization cues to which they have been exposed.

To incorporate temporal integration into the system design a confidence factor is produced so that important localization cues can be remembered. The system design uses the previously mentioned fuzzy weights to produce a confidence

factor every 50 ms. The confidence factor can be thought of as a fuzzy weight which has a strong influence on the directional probability maps produced over time.

In conclusion the goals of this thesis were achieved by adding fuzzy-weights to the simple localization system for the following factors:

- 1) Onsets
- 2) Coincidence over frequency
- 3) Coincidence over time
- 4) Temporal integration

A detailed description of each of these fuzzy weights is given in the following sections.

2.2 System Overview

This section describes the main components of the system. The system was implemented in MATLAB™. It requires almost an hour to process about one second of audio data. A future goal might be to implement the system on specialized hardware to enable it to run in real time. Accordingly the programs were written with such a real-time implementation in mind. In particular, instead of taking recordings of a few seconds in length and processing them in their entirety, the system buffers 50 ms of input and processes these buffers in a looping operation. Therefore many final and initial conditions must be saved

during the execution of any particular loop. Although this adds significant complexity to the programs, it allows for much easier portability to hardware. The actual localization system was broken up into several different modules, such as the modules for the ITD, IED and IID localization cues. This simplified development and allowed for observation of the localization abilities for each localization cue separately. An integrating module produces a final directional probability map which combines the probability maps produced by the ITD, IED and IID modules. Figure 5 shows a block diagram of the sound localization system designed in this thesis. Detailed descriptions of each of the components of the system are given in Section 2.3. The following overview may help place the detailed descriptions in perspective.

The input to the system is a binaural recording made through an acoustic mannequin called KEMAR. KEMAR's plastic head and rubber pinnae produce a human-like HRTF. Inside KEMAR's ears there are two microphones which are used to make the recordings. The recordings are 16-bit digital recordings with a sampling frequency of 44.1 kHz. A microphone pre-amplifier and a signal amplifier are used in the recordings to adjust the signal levels. Once any recording is made it is processed in 50 ms segments.

The right and left signals are first processed by a model of the cochlea. This essentially performs a spectrum analysis on the signals. The cochlear model

Figure 5: Block Diagram of Overall System

breaks the signal up into 69 frequency channels, with the channel's center frequencies ranging from 45 Hz to 20 kHz. The channel outputs are rectified and low-pass filtered to provide estimates of the short-term energy in each channel.

Next the ITD information is extracted by cross-correlating the peaks of the signals in each of the channels that have center frequencies below 1 kHz. The IID information is extracted by computing the average energy over 1 ms of signal and taking the ratio of the right to left signals. The system calculates the IID only for bands with center frequencies above 1.9 kHz. The IED is extracted in almost the exact same way as the ITD. Since the cochlear model includes a low-pass filter with a cutoff frequency of 1 kHz following the band-pass filters, the cochlear model essentially extracts the envelope of the higher frequency signals instead of the signals themselves. Time-difference extraction is the same for the ITD as the IED. Also an onset analysis is performed on the output of the cochlea to determine the fuzzy amount of onset that any particular peak may have. If a portion of a signal has a strong onset, the ITD, IED and IID measurements made at that time will be given more credence.

Next a coincidence analysis is performed on all the ITD, IED and IID measurements and fuzzy weights are assigned to each measurement. The set of coincidence fuzzy weights and onset fuzzy weights are combined to produce an overall confidence weight to be used to determine directional probabilities. Directional probabilities are calculated using a maximum likelihood criteria with the weighted ITD, IED and IID data. One directional probability map is produced for every 50 ms of data, and the probability map is given an overall confidence

factor. It is this confidence factor that helps integrate the directional probabilities over time. If a particular 50-ms time period has poor localization cues its probability map will be remembered with less emphasis, whereas a probability map with a very high confidence level will dominate the other probability maps over time.

Once each of the ITD, IED, and IID programs has run, their probabilities are combined to produce a single directional probability map. The combination basically averages the three probability maps, although some weighting is used both to take in account the energy of the signal, and to correct for biases observed in the system.

The following sections describe in detail the operation of each particular block of the system.

2.3 System Components

2.3.1 The Cochlea

The human cochlea acts as a "front end" for human hearing. The cochlea filters incoming acoustic signals by dividing the signal into its corresponding frequency channels. The frequency channels of the cochlea can be approximated as constant-Q filters as described in the Patterson-Holdsworth gammatone model [13]. For center frequencies down to 350 Hz, the system uses a MATLAB

implementation of the Patterson-Holdsworth model developed by Malcom Slaney [14]. At lower frequencies and high sampling rates, Slaney's implementation encounters numerical problems. These problems were dealt with by designing cascaded gammatone filters for the lower frequencies. This was necessary to keep the sampling rate the same for each frequency band. It may seem odd to insist on keeping the sampling rate at a constant rate especially for very low-frequency signals; however ITD calculations were greatly simplified by having each frequency band sampled at the same rate.

In general a 10-degree shift in azimuth corresponds to about a 0.1-ms delay in arrival time between the two ears. This time delay is measured in numbers of samples by the system, and at 44.1 kHz each sample delay corresponds to approximately 2 degrees in azimuth. Since humans can distinguish a 2-degree movement of sound source location it seems logical that a sampling rate this high be used to measure the ITD. In order to use a lower sampling rate for the low-frequency signal an anti-aliasing filter would be needed and this filter would change the phase and add a delay to the signal, thus making it more difficult to compare the signal with the signals from other channels with different sampling rates. The output of the lower frequency channels would also have to be up-sampled by interpolation in order to get clear ITD resolution. Therefore for these reasons the sampling rate throughout the ITD and IED calculations remains a constant 44.1 kHz.

As mentioned there are numerical problems caused by having narrow-band band-pass filters at very low frequencies compared to the sampling frequency.

This is due to the fact that the poles are very near the real axis in the z-plane. In keeping with Slaney's gammatone implementation the system uses 4 sets of conjugate poles for each band-pass filter. When these poles are expanded, the expansion is inaccurate even using double precision floating-point numbers. This problem was solved by cascading the filters and by adapting the pole's positions with the use of an LMS procedure. After adaptation the cascaded filters' response was within 1% error of the Patterson-Holdsworth model. The resulting filter coefficients are given in Appendix A.

The Patterson-Holdsworth gammatone model defines the equivalent rectangular bandwidth (ERB) as

$$\text{ERB} = \frac{f_c}{9.26} + 24.7$$

where f_c is the center frequency in Hz.

Thus for frequencies above 2 kHz this model is essentially constant-Q with a $Q \approx 9.26$ for frequencies above 2 kHz. At lower frequencies the filters approach a constant bandwidth (see Figure 6).

By using 69 gammatone filters with their center frequencies equally spaced, the highest amount of attenuation any signal could experience was 1.5 dB. (The magnitude response of adjacent channels intersects at 1.5 dB of attenuation). After passing the signal through the gammatone filters the array of 69 signals for each ear is half-wave rectified and low-pass filtered in keeping with a biological model of the cochlea. The low-pass filter is a single-pole filter with time constant

$\tau = 1$ ms. This essentially means that the output from the higher frequency bands is the envelope of the signal at that frequency instead of the signal itself. Figure 7 shows the block diagram of the cochlear model used by the system.

Figure 6: Center Frequency vs. Bandwidth for Gammatone Filters

Probably the most important aspect of the cochlea not modeled is automatic gain control (AGC). An effective AGC is probably the best method of measuring onsets, nonetheless the system design was simplified greatly by not including an AGC portion in the cochlear model. However this means that certain care had to be made when setting recording levels while recording input. Obviously

saturation had to be avoided, and signal-to-noise ratios were generally kept at approximately 20 dB for most tests.

Figure 7: Block Diagram of Cochlear Model

One of the best ways to gain insight into the function of the cochlear model is to observe the output of the model given different inputs.

Figure 8 and Figure 9 show input signals, 3 channel outputs from the gammatone filters, and 3 channel outputs of the cochlear model. These channels correspond to center frequencies of 220 Hz, 830 Hz, 2.3 kHz. The final

outputs of the cochlear model in these figures also have their "peaks" marked with "o's." This peak marking is used to extract timing differences between the right and left signals. Figure 8 uses 50 ms of the vocal sound "a" in "sat" spoken in a reverberant room at 50° azimuth as input. Figure 9 uses 50 ms of the consonant sound "s" in the same utterance.

Figure 8: Output from voiced reverberant speech: "a" at 50° azimuth

Figure 9: Output from unvoiced reverberant speech: "s" at 50° azimuth

2.3.2 Low-Frequency Localization: ITD

The localization program using the ITD is described by the block diagram in Figure 5. But how is the ITD information extracted? The first step is to recognize that there are clear timing cues only for frequencies below 1 kHz, therefore the ITD localizer operates only on the channels from the cochlear model that have center frequencies below a 1 kHz. This corresponds to a total of 22 channels. Next some form of analysis is needed to estimate the delay between the right and left ears. Many different types of analysis were attempted until suitable results were obtained.

The first method attempted was simply to cross-correlate the right and left signals. The main difficulty with this method is that the data windows dominated the cross correlation. Many different window types were applied, but in every case the cross-correlation was between the windows and not the signals. The window problem was augmented by echoes and noise. After careful examination of the data we observed that the peaks in the signal contained the correct delay sometimes, while at other times they did not. The only method that seemed to solve the problem was some form of peak detection. This method of timing extraction is also observed in biology [5].

What seemed to be the easiest method of peak extraction, taking the derivative, did not work due to the amount of noise in the signals. Using the derivative method many "non-peaks" were extracted. So instead of trying to fine tune this more elegant method a brute force solution was applied that worked very well.

This brute force method consisted of essentially the following algorithm. Observe successive samples as they increase in value until they begin decreasing in value. If the successive increasing values exceed a user defined threshold, and if successive decreasing values span the same threshold, a "peak" has been found. The peak-extraction program recognized only positive peaks - local maxima. The program was very successful at peak extraction even under very noisy conditions since the minimum peak size to be extracted could be defined by the user. For all tests run on the system the threshold was set at 0.5×10^{-4} . (The signal was represented by numbers with 16-bit precision, normalized to values between -1 and +1).

Figure 8 shows the output from the cochlear model for a few channels for voiced speech. The fact that the gammatone filters are very narrow-band at low frequency is evident in the channel ringing. The peaks extracted by the "find peaks" program are marked with "o's." (The channel ringing is difficult to see due to the large number of "o's" placed on Figure 8).

Once the peak positions are known calculating the ITD is simply a matter of measuring the difference in time between the right and left channel peaks. Some difficulties arise due to the fact that a peak may be large enough to be recognized on the right ear's signal but not large enough on the left. In other words, care must be taken to make sure a difference is only taken between corresponding peaks. We can ensure this is achieved by noting that the maximum delay between right and left sides is 45 samples (@ 44.1 kHz). If there are no corresponding peaks within a 45 sample window, the peak-

extraction program has recognized a peak that exists on one side only.

At this point the system has produced a matrix of ITD values. The rows of this matrix correspond to the cochlear channels, and the columns to the ITD measurements made in the 50-ms cycle. Since accurate ITD measurements depend on the signal, the number of columns in the ITD matrix can vary from 0 to 50. (50 ms x 1 kHz = 50).

When recordings are made in noisy reverberant environments the ITD values are essentially random numbers between -45 and 45 samples. Figure 10 shows the number of samples delayed for different azimuth angles using the HRTF data. Delay times vary linearly with azimuth, and the maximum number of sample shift is about 45 samples. A least-squares best fit for the envelope delay of the impulse response produces the following equation which is used in estimating the azimuth once the timing information is extracted.

$$\text{azimuth} = 2.4(\# \text{ of samples delayed}) + 1$$

where $-90^\circ \leq \text{azimuth} \leq 90^\circ$

(Since the ITD cannot distinguish front from back the source might also be located at $180^\circ - \text{azimuth}$.) To extract the ITD measurements that correspond to the source all the fuzzy weights must first be calculated. The first fuzzy weight to be determined is that of "onsetness", since this can be directly calculated from the output of the cochlear model. Since there is an ITD measurement for each peak in any channel signal, an amount of onsetness is also assigned to each peak.

Figure 10: Envelope Delay of Impulse Response

There was no underlying theory used in the development of the onset fuzzy membership function. We defined "onsetness" based on observation and plausibility. The following method was used to determine "onsetness."

A ratio is taken between the level at a peak with the level of the preceding peak. If this ratio is greater than one the level has increased, and this ratio is converted to dB. This is the value of "onsetness" corresponding to the peak. If the ratio were less than or equal to one, "onsetness" is simply given a value of zero. The

amount of time between peaks in a channel also influences values of "onsetness." For example, if a peak were detected in a channel that had not detected a peak in a "long time," then this peak would signify activation of this channel. This transition from quiet channel to active channel should modify "onsetness" accordingly. The following is the algorithm for calculating "onsetness."

If $\text{peak}_2 > \text{peak}_1$

$$\text{onset}_{\text{mag}} = 20 \log_{10} \left(\frac{\text{peak}_2}{\text{peak}_1} \right);$$

else

$$\text{onset}_{\text{mag}} = 0;$$

end

$$\text{onset}_{\text{time}} = 20 \log_{10} (t \cdot \text{cf});$$

$$\text{onsetness} = \frac{(\text{onset}_{\text{time}} + \text{onset}_{\text{mag}})}{2}$$

t = time between peak_2 and peak_1

cf = center frequency of the channel in question

$\text{peak}_1 \text{ or } 2$ = signal level at the time a peak was detected. It is assumed that peak_1 that was detected first.

By multiplying the time between peaks with the center frequency of the channel, the time between peaks is normalized. The closest that two peaks could be in time in a 1 kHz channel would be 1 ms, but the closest that two peaks could be in a 10 kHz channel would be 0.1ms. Multiplying these times by the center frequencies of the channel normalizes how quick onsets occur. It should be

noted that since the peak extraction program was very sensitive, peaks were almost always found at intervals that corresponded to center frequency of the channel. Therefore $\text{onset}_{\text{time}}$ was not a strong determining factor in the calculation of "onsetness."

At this point in the program there are two matrices of equal size that contain ITD measurements and onset weights. The onset weights by themselves are not sufficient to extract the correct ITD measurements, and coincidence analysis is also necessary. First the cross-channel, or frequency coincidence is calculated. This is achieved by summing across the columns of the onset matrix, and at the same time calculating the number of channels that have an onset value greater than zero. These two numbers are added and combined with the original onset value for a given column, provided that the onset value is greater than zero. Thus the onset matrix is altered giving greater weights to those onsets that happen at the same time as other onsets.

One of the great advantages of working with the peaks of the signals instead of the signals themselves is that the problem due to the non-linear phase response of the channels in the cochlear model is avoided. (The gammatone filters in the cochlear model are IIR filters.) A column in the onset matrix represents corresponding onsets in the different channels, even though these onsets are displaced in time by the non-linear phase response of the gammatone filters.

The onset matrix undergoes one last alteration before it is used to help

calculate the directional probabilities: coincidence in time must be calculated.

Now since a weighting matrix, onset, had already been constructed it was easiest just to add the new coincidence in time information to the weights in that matrix. Coincidence over time was calculated by observing the ITD matrix. If adjacent ITD values in any given row were the same it was a strong indication that the values corresponded to a sound source. Based on observation, the following coincidence membership function was defined for adjacent ITD values:

D = absolute difference in adjacent ITD measurements (samples)
(D is always positive)

C_t = coincidence over time membership

```

If  $D < 5$ 
     $C_t = -\frac{2}{5} D + 20$ 
else if  $D < 7$ 
     $C_t = -9 D + 63$ 
else
     $C_t = 0$ 
end
end

```

Coincidence weights were also given to ITD measurements that were 2 and 3 columns away. Since in this case the coincidence was further away in time, the coincidence weights were less than the previous definition. If coincidence was observed two or three columns away the weights were defined as

D_2 = absolute difference in ITD measurements 2 and 3 columns away
(D_2 is always positive)

```

If  $D_2 < 3$ 
     $C_t = -\frac{2}{3} D_2 + 10$ 
else if  $D_2 < 5$ 
     $C_t = -4 D_2 + 20$ 
else

```

Ct = 0
end
end

Figure 11 shows a plot of both these fuzzy membership functions. Again, a value of C_t is calculated for every element in the ITD matrix, and the values of C_t

Figure 11: ITD Coincidence-over-time membership functions

are added to the corresponding elements in the onset matrix. This leaves the onset matrix with a combination of fuzzy weights for each ITD measurement.

Finally a weighted average of ITD information can be calculated, although it is important to note that these averages are over the different sample delay possibilities. In other words a sum is performed for each of the 83 possible delay values from -41 to 41. This is done by summing all the fuzzy weights corresponding to a particular delay. In this way multiple sources are easily discernible. The entire sum of fuzzy weights is used as a measure of confidence for the corresponding 50-ms block of data.

The last task for the program was to translate the number of samples delayed into an estimate of azimuth position. This was achieved by using the linear relation shown in Figure 10. The final azimuth estimates were also normalized to sum to one so as to represent directional probabilities.

It was observed that 50-ms blocks of data that had a small amount of energy produced low confidence factors, e.g., during periods of unvoiced speech, while periods during high signal energy (periods of voiced speech) produced high confidence factors. This can be seen by comparing Figure 8 and Figure 9. Figure 8 shows the peaks extracted during voiced speech, and it is clear that the right signal precedes the left. (The position of the speaker was 50° azimuth.) Whereas no clear timing tendency can be observed in Figure 9.

Figure 12 and Figure 13 show plots of the fuzzy weights during voiced and unvoiced speech for two channels (220 Hz and 830 Hz). It should be noted that although `onset_mag` looks small on these plots, without `onset_mag` there could be no coincidence over frequency or coincidence over time. It should also be noted that these plots are not over time, but are plots over the column indices of the ITD measurements. The amount of valid ITD measurements (columns) can vary for any 50-ms section. The largest Final Weights shown on Figure 13 correspond to the ITD measurements made at the onset of the voiced speech "a." Also note that many more valid ITD measurements were made in the 830 Hz channel during the voiced speech (36), as compared to the same channel during the unvoiced speech (24).

This ITD system was very accurate at extracting the timing delays from speech in a reverberant environment. Nevertheless, timing data provides two sound source positions in the horizontal plane for every sound source (see Figure 1). Therefore some IID information must be extracted to distinguish front from back. However, we describe the IED module first because it is so similar to the ITD module.

Figure 12: Fuzzy weights for unvoiced reverberant speech: "s" at 50° azimuth

Figure 13: Fuzzy weights for voiced reverberant speech: "a" at 50° azimuth

2.3.2 High-Frequency Localization: IED

The program that localized on IED information is essentially the same as the ITD program. The main difference is that the IED program operates on channels with center frequencies above 1.9 kHz, thereby extracting timing information from the envelope of the signal instead of the signal itself. For both the IID and the IED the upper 38 channels of the cochlear model were used to estimate azimuth.

It may seem odd that 22 channels were used for the ITD and 38 for the IID and IED while the entire cochlear model has 69 channels not 60. This was done for a variety of reasons. First the localization cues for signals between 1 kHz and 2 kHz are poor at best, therefore it did not warrant the extra amount of computing time for little benefit. While the cochlear model used for sound localization research could have simply omitted the 9 unused channels, it was desired to have a general cochlear model that could be used in any type of hearing research. As an analytical tool, a program named "cochlea-gram" using the cochlear model was written to observe images of the output of the cochlea. The program was inspired by the work of Lyon and Mead [15]. Accordingly, a 69-channel cochlear model was designed for completeness, even though only 60 of the channels were actually used for the localization programs in this thesis.

As mentioned, the IED program is essentially the same as the ITD program. The only significant difference is that it operates on the higher frequency channels, and therefore it should probably also called an ITD measurement.

Nevertheless, the peaks extracted in the high-frequency channels are a good representation of the signal's envelopes (see Figure 8). Since the ITD program estimates azimuth using peaks, no substantial change to the program is needed to estimate azimuth with high-frequency channels. The main problem with using the ITD program to measure the IED is that the entire fuzzy weighting scheme presented in the last section assumed that the peaks extracted were signal peaks. In the IED case, the peaks represent the envelope of the signal and not the peaks of the envelope. The IED program should run the peak extraction program twice in order to extract the peaks of the envelope. Unfortunately, this method of running the peak extraction program twice was discovered after many tests had already been run on the system, and it was decided to keep the IED program as it was.

It should be mentioned that the original development of the IED program used many different filters in an attempt to extract the envelope from high-frequency signals. None of these filters worked. Although these filters extracted the envelope of the high-frequency signals, they also smoothed over the delicate timing shifts between the right and left channels. In desperation, the ITD program which worked so well was applied to the high-frequency channels, and this worked surprisingly well. It was not discovered until most of the testing was finished, that the reason why the ITD program worked well with high-frequency signals was that the peak extraction program was in effect extracting the envelope. The fact that the ITD program can also operate as an IED program demonstrates the robustness of the ITD program.

2.3.3 High-Frequency Localization: IID

The IID is potentially the most powerful localization cue, because it can localize in both azimuth and elevation [4,7]. Unfortunately, to extract IID information there must be significant energy in frequencies above 2 kHz. This is rarely the case with speech signals. It was hoped that the little IID information that could be extracted from speech could at least help distinguish the front-back confusion inherent in the ITD calculations. No previous work had attempted to extract IID information in a noisy reverberant environment, therefore many different extraction methods were attempted. None of these methods was very successful. The following discussion describes these attempts, and the method that was finally used.

Again in an attempt to distinguish source from reflections a great deal of thought was put in on the role of onsets. If an onset could be quantitatively described, it could be used as an accurate method of deciding a point in time when an IID measurement would be accurate. An attempt was made to measure the onsets by taking a difference between the signal average over 10 ms and the signal delayed by 10 ms. This method was based on the work by Shahwan and Duda [16]. It was very successful at finding the onsets of glottal pulses in the upper-bands, but when IID measurements were made at these times, the results did not correspond to the HRTF data. The measurements were about 30-40% in error and always towards the median plane. This large amount of error forced

abandonment of this method, even though it was very successful at locating onsets.

The fact that the previous method always made mistakes toward the median plane reconfirmed an observation made during some preliminary work done at San José State University in the summer of 1995. At that time a real-time sound localizer was built that used only an IID measurement to localize. Since the system operated in real-time, clicks of short duration were easily separated from their corresponding reflections. When the system localized using the reflection as the input, it always localized the sound in the median plane. Many experiments were made with different sounds and different orientations in the room, but reflections always continued to be localized in the median plane. This observation led to the following rule of thumb for reverberant environments: *The IID of reflections is always zero.*

When the present system attempted to localize during times of glottal-pulse onsets, the signals at the two ears consisted of source plus reflections. Using the previous rule of thumb it is obvious why the errors always tend toward the median plane. Using this fact, many attempts were made at introducing an offset that would cancel this error factor due to reflections. Unfortunately all linear attempts to modify the HRTF data to account for room acoustics failed to produce the desired results. Nevertheless, it is the author's belief that accurate IID measurements will only come from an HRTF that can self-modify to accommodate the changes in room acoustics. Since developing a self-

modifying HRTF is well beyond the scope of this thesis, a practical ad hoc method was used to obtain useful IID information.

The solution was to use the maximum absolute-valued IID for each channel during a 50-ms cycle. This maximum method proved to be a simple method for compensating for the effects of reflections on the IID. The IID was calculated by taking a running average of the channel signals over 1 ms, taking the ratio of the right side to the left, and then converting this ratio to a dB scale.

sig_r = averaged signal level from the right side

sig_l = averaged signal level from the left side

For each channel

$$\text{IID} = 20\log_{10}\left(\frac{\text{sig}_r}{\text{sig}_l}\right)$$

Figure 14 shows the IID for three channels during two different segments of speech. The segments of speech are the "s" in "sat" and the "a" in "sat." For every azimuth one can define a target IID from anechoic measurements of the HRTF. The word was spoken at azimuth 50°, and the dashed lines in Figure 14 mark the target IID for 50° azimuth. Figure 14 also shows that the maximum absolute-valued IID is close to the target value. The 4.0 kHz channel on Figure 14 is very important since if the sound were coming from 130° (its reversal point on the cone of confusion) the IID would be 3 dB not 12 dB as shown (see Figure 2).

Beside the maximum method, fuzzy weights were also used to determine the

Figure 14: IID values for 3 channels during voiced and unvoiced speech

confidence in each IID measurement. The first weight was based on the signal level in each channel (`ave_energy` in the MATLAB code), and the second weight measured the increase of channel level during an entire cycle. This second weight provided a simple method of measuring the amount of onset in each channel (`onset_weight` in the MATLAB code). We used another method of measuring "onsetness" by assuming that when a large onset occurs the IID will

differ substantially from the previous measures of the IID. This is calculated by taking the absolute value of the difference between the maximum IID measurement and the average IID measurement for any 50 ms cycle. All of these fuzzy weights are combined in the following way to produce a confidence of the IID measurement for each channel.

CC = channel confidence

maxIID = maximum absolute-valued IID measured during a 50-ms cycle (dB)

meanIID = average value of all IID measured during a 50-ms cycle (dB)

$$\text{onset_weight} = 20\log_{10}\left(\frac{\text{maximum signal level in channel}}{\text{minimum signal level in channel}}\right)$$

min_energy = -90 dB

(Each sample coming out of the cochlear model is assigned a minimum value of 3.16×10^{-5} if the sample's value is 3.16×10^{-5} or less. This is done in the IID program to avoid taking the log of zero). $-90 = 20\log_{10}(3.16 \times 10^{-5})$

ave_energy = average signal level (dB)

$$\text{CC} = |\text{maxIID} - \text{meanIID}| + \text{onset_weight} + e^{\left(\frac{\text{min_energy}}{\text{ave_energy}}\right)}$$

The average signal level in each channel tended to vary from -90 dB to -40 dB.

Instead of having a matrix of IID values for each 50 ms cycle as was the case for ITD and IED, there is simply an IID vector of length 38 (one for each channel) for every 50 ms cycle. This value corresponds to the largest IID measured for each channel, and each value in the IID vector has a corresponding channel confidence. Cross-channel coincidence was also given a weight by counting the number of channels during a 50-ms cycle that had a channel confidence

above 40. However, it is important to note that cross-channel coincidence was a factor in temporal integration and not for producing the 50-ms cycles' probability map (see Appendix A).

Once the IID measurements were made a maximum likelihood estimation was made for each channel regardless of its confidence level. A gaussian distribution was assumed with $\sigma = 5$ dB:

$$p(ch, az) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{[IID(ch) - IID_map(ch, az)]^2}{2\sigma^2}}$$

This equation was used to define the probability densities for each channel (ch), and in each azimuth (az). A weighted average over all the channels was used to produce a single probability map for every 50-ms cycle, the weights being the channel-confidence fuzzy weights. The IID_map contains the IID value for each channel and at every 10° increment in azimuth. This map was calculated by convolving a random signal with KEMAR's head-related impulse responses (HRIRs) at the various azimuth locations.

Temporal integration over various cycles was achieved by taking a weighted average of previous probability maps and the present one. The weighting was determined by an overall confidence factor which in turn was determined by the channel fuzzy weights and the cross-channel coincidence.

By itself, this method of localizing with the IID generally did a poor job of localizing speech and other sounds in a reverberant environment. The

directional probability maps this system produced all had a very large variances, with the system tending to favor the front over the back. Luckily this front bias was removed with an offset, and the IID system was successful at distinguishing front from back at least 80% of the time. Although this localizing system was poor at determining the sound source's location it was very helpful in removing front-back confusion. Since the ITD system could localize speech very well on a particular cone of confusion, the IID system's information was combined with that of the ITD to obtain localization in the entire horizontal plane.

2.3.4 Combining probabilities

At this point we have described three main modules in the overall localizing system. There is a program that localizes with the ITD cue, another with the IED cue, and finally a third that uses the IID cue. All of these programs operate on binaural sound signals in a looping operation (each loop or cycle takes 50 ms), so that they can be implemented in real time. At any given time each one of these programs contains an overall directional probability map. What remains is to combine these maps to produce one map.

For speech, most of the signal energy is below 2 kHz, therefore it is logical that the ITD localization should be given the most credence. By contrast, clicks and snaps have plenty of energy above 2 kHz, and the ITD program has little chance to measure coincidence over time. Therefore in the case of wide-band signals

the IED and IID should be given more credence. Using these basic ideas of energy content, the following method of combining probabilities was used.

The ITD and IID programs incorporate a measure of average energy per channel. This basic measure enables the system to be able to decide where most of the signal energy resides. Using this measure weighted averages of the ITD, IED and IID maps are calculated to produce a single probability map.

Unfortunately this simple method tended to weight very heavily the ITD measurements for speech signals. This is logical since most of the energy in a speech signal is below 2 kHz. Nevertheless, it was very important to incorporate more front-back information from the IID program in order to improve the systems localization ability.

A method to help clarify front-back confusion seemed to work well at first until a bias was discovered. This method summed the IID probabilities for the front, and divided that number by the sum of the IID probabilities for the back. If this ratio were greater than one, the system decided that the sound was in the front and arbitrarily doubled all the front probabilities. If the ratio were less than one it would double the back probabilities. This seemed to work until it was discovered that this method would almost always decide that the sound was in the front even when it was actually in the back.

The front-back determination was changed from an absolute decision to a membership in the fuzzy sets "frontness" and "backness." This membership

function was determined from some of the test data, and then applied to all of the results. The test data used was 16 of the recordings of the azimuth tests described in Section 2 of Chapter 3. These were recordings of speech made at the following locations: azimuth = 10°, 20°, 30°, 40°, 50°, 60°, 70°, 80°, 100°, 110°, 120°, 130°, 140°, 150°, 160°, and 170°. The ratio previously described was calculated in each case, and another ratio was calculated which consisted of the maximum IID probability in the front divided by the maximum IID probability in the back.

IID_prob_front = The set of probabilities that the IID program calculated for azimuths = 10°, 20°, 30°, 40°, 50°, 60°, 70°, 80°.

IID_prob_back = The set of probabilities that the IID program calculated for azimuths = 100°, 110°, 120°, 130°, 140°, 150°, 160°, 170°.

$$\text{Ratio}_1 = \frac{\text{sum}(\text{IID_prob_front})}{\text{sum}(\text{IID_prob_back})}$$

$$\text{Ratio}_2 = \frac{\text{max}(\text{IID_prob_front})}{\text{max}(\text{IID_prob_back})}$$

The sum function sums all the probabilities and the max function extracts the maximum probability. One would assume that the values for Ratio₁ and Ratio₂ would be greater than one if the sound source were on the front side of the horizontal plane, and less than one if the sound source were on the back side. Nevertheless, it was observed that the ratios were about 1.3 for sounds in the

Figure 15: Front-Back Ratios from IID Probabilities

front and 1.1 for sounds in the back. Figure 15 shows the sum of these ratios for test cases at azimuths between 10° and 170° . Figure 15 also shows that the IID ratios do distinguish front from back, but the decision boundary between front and back, using the sum of the two ratios, is not 2 as expected. The decision boundary observed is about 2.4 (shown as a dashed line in Figure 15). Using 2.4 as the decision boundary the IID probabilities misclassifies 3 of the 16 test cases or 19%. Although a slightly higher decision boundary would correctly classify more of the test cases, the objective was not to find the best fit for the data, but to extract a front-back pattern from the data. Actually, the 2.4 cutoff

ratio does a better job at classifying the independent test data as compared to classifying the training data.

The fuzzy membership functions defined in Figure 16 alter the overall probabilities by multiplying the probabilities in the front by "frontness," and multiplying the probabilities in the back by "backness." "Backness" is given a larger maximum value (3) as another method to compensate for the forward bias in the IID measurements (see Figure 16).

Figure 16: Fuzzy Membership Functions "Frontness" and "Backness"

The combination of offsetting the decision boundary (from 2 to 2.4) and assigning different maximum membership values helps alleviate the forward bias of the IID probabilities.

With the combination of the ITD, IED, and IID programs completed, the overall system can produce a directional probability map at any time based on the given signal inputs. The system was designed using a more complex looping operation so as to allow for real-time implementation. The only aspect of a real-time system not modeled in this thesis is the interval of time in which directional probabilities are forgotten. Preliminary work suggests that forgetting should be phased in after 2 to 5 seconds. Since the present system takes over an hour to process 1 second of data, no test recordings lasting this long were made, and there was no need to incorporate "forgetting" into the system.

Chapter 3

Testing and Evaluation

3.1 Walking Through a Single Test

It is very enlightening to observe how the different cues used by the localizing system affect overall probabilities. These same observations reconfirm some basic notions we have about which parts of speech give the clearest localization cues. Since unvoiced speech is wider band in nature, it should produce better IID and IED information as compared to ITD information. However, unvoiced speech contains very little energy and is more difficult to localize. During voiced speech, there is a large amount of energy in the signal and the system should improve its ability to localize.

The illustrative example used a recording of the word "sat" made at 50° at a distance of 2 meters. It should be noted that the placement of the source in all the tests was only within a $\pm 5^\circ$ azimuth. The speech was uttered by a male speaker in a noisy, reverberant environment (SNR = 15 dB, and SRR = 7 dB). The SNR was measured by observing the data input, and the SRR (Signal to Reflection Ratio) was estimated by measuring the impulse response of the room in which the recordings were made. The binaural recording was made through KEMAR, and the resulting right and left signals were processed by the localizing system. During the utterance of the "s" sound in the word "sat" the probability maps on Figure 17 were produced by the ITD, IID and IED programs.

ITD Probability Map: Confidence = 202

IED Probability Map: Confidence = 340

IID Probability Map: Confidence = 498

Figure 17: Probability maps produced during a 50 ms segment of the utterance of an "s" in the word "sat" at 50° azimuth

It should be noted that no real attempt was made to normalize the overall confidence factors that each program produces. It has been observed that the confidence factor for the ITD program generally ranges from 200 to 2000. If the confidence factor for the ITD program falls below 200 temporal integration ignores the map, whereas a confidence over 600 is a very good indication of accurate localization. The confidence factor for the IID program tends to vary over the same range, but the IED program produces confidence factors that range from about 20 to 1000. Therefore a confidence of 1000 in the IED program is approximately equivalent to a confidence of 2000 in the ITD or IID programs. This discrepancy is due to the fact that the same definition of "onsetness" was used in the ITD and IED programs. As a future development the measure of "onsetness" should be modified in the IED case to distinguish envelope "onsetness" from signal "onsetness."

Figure 17 shows that during the unvoiced speech of "s", both the IED and IID do a better job than the ITD at localizing the sound source. The ITD program clearly has no idea of the source's location. Although the IID program seems to clearly distinguish the front from back, it should be remembered that the IID program has a forward bias. In other words if the forward probabilities in the IID program are about 10% greater than the backward probabilities then it's more probable that the sound source is in the back. In Figure 17 the forward probabilities for the IID program are a little more than 20% greater than the backward probabilities. In this case the combine-probabilities program will only give a slight forward bias.

Figure 18: Probability maps produced during a 50 ms segment of the utterance of an "a" in the word "sat" at 50° azimuth.

Figure 18 shows the probability maps produced during voiced speech, the "a" in "sat." During this 50-ms segment there is considerably more energy in the lower frequencies. Also the periodic nature of the vowel sound helps the ITD program since its fuzzy weights give greater credence to ITD measurements that show coincidence over time. There is also improved confidence in the IID localization, but the wide variance of direction clearly demonstrate the weaknesses of the IID program.

Figure 19: Overall Probabilities for "sat," spoken at 50° azimuth

Finally, Figure 19 shows the combined probabilities for the entire recording of the word "sat". This overall probability map is shown with arrows and KEMAR's head to distinguish it from the probability maps produced every 50 ms. The length of the vectors correspond to the amount of probability in that direction. The arrow heads are added in the case that a probability is at least half the maximum value. In this test no clear front-back distinction was made, although the greatest probability is at 60°, only a 10° error from the actual location of 50°.

This example using the word "sat" was chosen because it produced one of the poorer results of all the phoneme tests. Even with the poorer performance in this example, the system produced accurate results.

3.2 Word Tests

The objective of the following set of tests was to determine if the system varied its performance depending on the word being spoken. Eighteen recordings were made all at the same location (azimuth = 50° ±5°, distance = 2m). The same speaker was used for all tests.

The eighteen test words were chosen to examine the localization ability of the system using all the English vowel phonemes (front, mid, back, and diphthongs). Most of the consonant phonemes were also mixed in with the tests. Table 1 shows the words used in this test and the ARPABET representation of the tested phoneme.

Figure 20 shows a gray-scale plot of the overall probabilities. The black line represents the actual location of the sound source, and the dotted lines represent the range or error that humans experience when trying to localize on speech in a reverberant environment. Humans experience an average error of between 12° and 17° when attempting to localize speech in a reverberant room [11].

<u>ARPABET</u>	<u>Test Word</u>
IY	se <u>a</u> t
IH	b <u>i</u> t
EY	l <u>a</u> te
EH	g <u>e</u> t
AE	s <u>a</u> t
AA	s <u>o</u> b
AH	b <u>y</u> t
AO	c <u>a</u> ught
OW	g <u>o</u> at
UH	l <u>oo</u> k
UW	b <u>oo</u> t
AX	<u>a</u> bout
IX	r <u>o</u> ses
ER	b <u>i</u> rd
AXR	b <u>u</u> tter
AW	f <u>r</u> own
AY	b <u>u</u> y
OY	t <u>o</u> y

Table 1: Words used for phoneme test

Figure 20 shows two whiter vertical bands. The first band corresponds to the sound source location and the second band corresponds to the front-back confusion, however it is clearly lesser in magnitude, and in no case was there a front-back reversal. Figure 21 shows the same results as a mesh plot.

Figure 20: Results of Phoneme Tests

The phoneme tests also indicate that there is no conclusive proof that any particular phoneme that is "better" for localization. It was previously thought that the phonemes that contained higher frequency content would perform better, but

the test with "seat" shows that this is not the case. Although another explanation could be that the formant energy in "seat" may be around 1 kHz, which is a frequency that the system ignores. If any conclusion can be made from the phoneme tests it would be that the unvoiced stops ("P", "T", "K") tend to help the system, although this conclusion is tentative.

If the highest probability is used as a method of deciding where the sound is coming from the 18 phoneme tests give the following results.

<u>Location</u>	<u>Number of Times Guessed</u>
30°	1 ("seat")
50°	3
60°	13
70°	1 ("bird")
Front-Back Reversals	None

Since the source actually was at 50°, the combination of the phoneme tests have an average error of 9.4°. This is significantly better than human performance (average error ~ 15° [11]), especially when one considers that the placement of the sound source had an error of $\pm 5^\circ$.

Since Figure 19 showed the overall probabilities of one of the poorer results of the phoneme tests, Figure 22 shows one of the better results. All the other results were in between these two extremes.

Figure 21: Mesh View of Results from Phoneme Tests

Figure 22: Overall Probabilities for "look," spoken at 50° azimuth

3.3 Azimuth Tests

Since no particular word tended to perform better than any other, the word "KEMAR" was arbitrarily chosen to be used in the azimuth tests. The word was spoken at 36 different azimuth locations in 10° increments. At 3 of the locations a female child spoke the word, and at 6 of the locations an adult female spoke the word, and at the rest of the locations an adult male spoke the word. There was no distinction produced by the system to the variations in speakers. In other words the system performed no better or worse for any particular speaker.

Tests were also made at a variety of distances ranging from 1.5 m to 5 m, but this again produced no measurable distinctions in system performance. It was thought that at greater distances the system would perform worse due to a smaller signal-to-reflection ratio (SRR), nevertheless the SRR ratio did not vary substantially from distances 1.5 to 5 meters. Actually, two tests at 5 meters outperformed tests at 2 meters, but in general there was no distinction between tests made at 2 meters and those made at 5 meters. This lack of distinction is probably due to the fact that all the tests were made in the same room where the SRR remains fairly uniform over the range 2 to 5 meters. Nevertheless, the SRR does have a large impact on system performance as can be seen in Section 3.4.

Due to the fact that range did not change system performance, all the following

Figure 23: Results of the Azimuth Tests

azimuth tests were done at a distance of 2 meters placing the speaker within an azimuth accuracy of $\pm 5^\circ$.

Figure 23 shows a gray-scale plot of the results of the azimuth tests. Ideally there should be a bright diagonal line on the plot. The two fairly bright off diagonals are due to front-back confusion. There are also two notable darker spots when the system attempted to localize at 90° and -90° . This is due to the

fact that the system did not have HRTF data for these positions. The system simply used a linear combination of the impulse responses at 80° and 100°. These results imply that HRTF's do not vary linearly.

Again if the highest probability is used to decide where the sound is coming from the 36 azimuth tests give the following results.

<u>Performance</u>	<u>Number of Times</u>
Correct Azimuth - no error	11
10° error	12
20° error	5
30° error	2
Front-Back Reversals	6

It should be noted that since the system is based entirely upon binaural cues, it has no means to distinguish front from back in the median plane.

Correspondingly, 4 of the 6 front-back reversals were made within 20° of the median plane.

One of the difficulties of interpreting these results is deciding how to quantify front-back reversals. If a front-back reversal is made at 0°, the absolute error in angle judgment is 180°. By contrast, a front-back reversal at 50° has an absolute error in angle judgment of 80°. Both of these errors are basically equivalent, yet the angular errors vary greatly. Unfortunately nobody has proposed a fully satisfactory way to correctly quantify front-back errors.

In their paper *The Accuracy of Absolute Localization Judgments for Speech Stimuli*, [11] Gilkey and Anderson separate the problem of front-back reversals by separating their tests into front-back tests and right-left tests. In their right-left tests the subjects could not make any front-back errors. When the front-back tests were made, an absolute error was calculated and varied greatly for the subjects due to the problems just mentioned.

If a simple absolute angular error criterion is used on the results of the azimuth tests, the average error is 31.4° . A comparison to Gilkey and Anderson's right-left tests can be made if the front-back reversals are assigned an amount of error that corresponds to the absolute error before reversal. Using this method the average error of the system is 9.7° . If the front-back reversals are simply not included in the average, the error becomes 9.3° . The following table compares the results of human performance tests made by Gilkey and Anderson to that of the azimuth tests.

	Human RMS Error	System RMS Error
Right-Left Tests	12.3° to 17°	9.7°
Front-Back Tests	25.4° to 40.2°	31.4° .

Regardless of the method of error calculation, the system clearly performs as well as a human does when attempting to localize speech in a reverberant environment.

It should be noted that the tests that Gilkey and Anderson made with human subjects were very similar to the azimuth tests performed on this system. The human subjects in their tests attempted to localize single spoken words (without allowing head-motion), although they used recorded speech played over a loud-speaker whereas in our tests a human speaker was used. The most striking difference between Gilkey and Anderson's tests and ours was that theirs were made in an anechoic chamber. Since it is easier to localize sounds in an anechoic environment, we believe that the system outperforms a human.

3.4 Two Sources

People can localize multiple sources, even if they are active simultaneously, since the sources most likely contain different spectral contents. This fact is used by Marcus Bodden to achieve sound separation [17]. In the present case, different channels can produce different localization cues. Of course there is a limit to the number of sources that can be recognized due to the overlap of directional probabilities and the finite number of channels in the cochlea.

To adequately deal with multiple-sources the system should include some form of a clustering algorithm to estimate the number of sources present. If this information were available it would be much easier to combine probability maps with greater accuracy. Nevertheless, the localization cues should be more than sufficient to localize multiple sources if there are high SNR's and high SRR's.

The following tests were made to evaluate the system's ability to localize two sources. First two speakers in an anechoic room spoke two different words simultaneously at azimuths $-50^{\circ} \pm 5^{\circ}$ and $+50^{\circ} \pm 5^{\circ}$. The words spoken were "canceled" on the left and "apple" on the right. The distance from KEMAR in the anechoic room for both speakers was approximately 1 meter. For this recording the SNR was greater than 40 dB, and the SRR was at least that value if not more. The upper portion of Figure 24 shows the results of this test. Clearly, the system performs exceptionally in an anechoic environment even when two people are speaking simultaneously. The lower portion of Figure 24 shows the results of the same experiment in a reverberant room. Obviously system performance is greatly degraded in the reverberant room. The poor performance is amplified by the fact that the system does not use any clustering algorithms to estimate the number of sources present.

Figure 24: Two sources in an anechoic room (upper map) and a reverberant room (lower map)

3.5 Reflection Suppression

The next set of tests evaluated the system's ability to suppress reflections. Since the system has been able to operate in an reverberant environment it has proven that it can suppress reflections, but the question is to what extent?

In the following tests clicks were played over two loud-speakers placed at azimuths $-50^\circ \pm 5^\circ$ and $+50^\circ \pm 5^\circ$. The right speaker click preceded the left speaker click in each test. In the first test the right click preceded the left by 20 ms. In the second test the time difference was 50 ms, and in the third test the time lag was 70 ms.

Figure 25 shows the results of these three tests. For the first test, the left click was ignored by the system since the 20 ms delay mixed the sound with the room reverberation. I myself could only hear one source in the room when a 20 ms delay was used.

The second test showed the appearance of a sound source on the left although the system was clearly confused as to its location. With a 50 ms delay I could recognize the two sources in the room, but just as importantly I observed a strong increase in echo as well. An important observation in the second test is that the system produced a great deal of median plane localization. This is consistent with the previously mentioned rule of thumb: the IID of reflections is zero. This median plane localization due to strong reflections was not

Figure 25: Results of Echo Tests: Top 20 ms delay, Middle 50 ms delay, Bottom 70 ms delay. Click sources at -50° and $+50^\circ$

previously observed with the speech tests since speech does not contain a great deal of high-frequency energy. In the present tests, the system relies much more heavily on its IED and IID programs due the wide-band nature of clicks.

In the third test the clicks were separated by 70 ms. With this much delay the system clearly recognizes the two sources, albeit with a larger amount of error than previously observed in the speech tests.

The system clearly demonstrates the ability to suppress reflections that occur up to 20 ms after a sound source. However, these tests also show poorer localization performance by the system. This is due to the fact that clicks are localized with the IID and IED portions of the system, and these programs are not as accurate as the ITD program.

3.6 Frassen Effect

Since the system relies heavily on onsets for localization it should also experience the Frassen effect as humans do. Therefore the final test was to see if the system experienced the Frassen effect.

In the following test, two loud-speakers were placed at azimuths $-40^\circ \pm 5^\circ$ and $+40^\circ \pm 5^\circ$. The right speaker abruptly started playing a 900 Hz sine wave and in 10 ms was attenuated to silence. While the right speaker was being attenuated the left speaker began playing the same sine-wave in phase with the right

speaker. The left signal was increased at the same rate that the right signal was attenuated. Under these circumstances humans will localize the sound source at the right speaker, since due to the standing waves in the room the only localization cue on which humans can rely is that of the onset.

Since the system relies on the onset cue it should be fooled as to the location of the sound source for this test. Figure 26 shows the results. In the test the sound continued to come from the left speaker (marked source on Figure 26), while the sound coming from the right speaker had long fallen silent. The full length of the test was reduced to just 0.2 seconds due to the long computation time, but the onset speaker was active for only 10 ms.

The system does not localize on the onset of the sine wave as well as it localized speech. It should be noted that humans are also much poorer at localizing sine waves. Again, as in the reflection suppression tests, the system attempts to localize on the initial "click" caused by the onset of the sine wave. The reliance on the IID and IED programs again seem to cause poorer performance.

This is by no means a definitive test. In order to accurately test for the Frassen effect the system should be modified to include the cochlea channels from 1 kHz to 2 kHz. This is the frequency range that humans most notably experience the Frassen effect. Aside from the front-back reversal, the test results do show a tendency of the system to experience the Frassen effect.

Figure 26: Results from the Frassen Effect test

Chapter 4

Conclusions

The main goal of this thesis was to design a sound-localization system that could localize speech in the horizontal plane in a reverberant environment as well as a human. This goal was achieved by incorporating the effects of onsets, coincidence in time and frequency, and temporal integration using fuzzy sets. The tests performed on the system indicate that it does localize speech as well as a human. Consequently, the system design achieved its main goal.

Despite the system's success, there are many weaknesses in the design. The main weaknesses are in the IID and IED portions of the system. There are also performance issues that should be addressed before it is implemented in real-time.

During the entire design phase of this research a large amount of effort was devoted to ensuring that this system could eventually be implemented in real time. The code in its present form could easily be translated to work with a real-time system since all the processing is self contained in 50-ms loops. The main problem with the ITD program in its present form is that the peak detection needed for the cross-correlation calculation is computationally very intensive. This implies that a great deal of DSP hardware operating

in parallel would be required to implement the present system in real-time. To avoid the need for expensive hardware, different methods for peak extraction should be investigated.

There are two lines of research that I believe could improve the IID localization in a reverberant environment. The first would be to continue attempts at incorporating onset analysis to IID extraction. It should be remembered that glottal-pulse onsets can clearly be detected with the onset extraction method described by Duda and Shahwan [16]. Regrettably, I was unable to utilize this information to extract correct IID information, but I still believe that it may be possible.

The second line of research that should be pursued would be to study the effects of room-acoustics on the HRTF. Knowing how the HRTF varies in a reverberant environment would significantly help sound-localization research. This could extend the onset extraction method described by Huang and Onishi to allow the system to "learn" the impulse response of the auditory environment in which the system is located [18].

Both of these areas of research would naturally complement each other, and since the localization ability of the IID cue is so great, having more knowledge in either of these areas would enable the production of more accurate localization systems.

The basic problem with the present IED program is that the envelope of the

signal is represented as peaks. This problem might be eliminated by simply running the peak extraction program twice. The first run would operate on the signal and second run would operate on the output from the first run. I believe this method would greatly enhance the IED program. Unfortunately this realization came too late to incorporate it into the system.

Finally, it should be noted that the success of this system was mostly due to translating direct observation into fuzzy membership functions. I am confident that performance could be improved if the fuzzy membership functions are redefined using adaptive methods (i.e. artificial neural nets) with large sets of training data

In conclusion, I believe that the theoretical problem of extracting ITD information from speech in a reverberant environment has been solved. It has been solved by implementing a system of fuzzy weights to probabilistic calculations based on a human auditory model. For completeness, IID and IED programs were added to the system, but they are clearly not as successful at localizing in the reverberant environment.

References

- [1] P.M. Zurek, "The precedence effect and its possible role in the avoidance of interaural ambiguities," *J. Acoust. Soc. Am.*, Vol.67, No.3 pp. 952-964, 1980.
- [2] Zhenggin Shu, "A Model for Auditory Lateralization in Non-stationary Multi-source Environments," Ph.D. Thesis, University of British Columbia, 1995.
- [3] David, E.E., Jr., N. Guttman, and W.A. van Bergeijk "Binaural interaction of high-frequency complex stimuli," *J. Acoust. Soc. Am.*, 31:774-782, 1959.
- [4] R.O. Duda, "Estimating azimuth and elevation from the interaural head related transfer function," in R.H. Gilkey and T.B. Anderson, Eds., *Binaural and Spatial Hearing* (Lawrence Erlbaum Associates, Hillsdale, NJ, in press).
- [5] M. Konishi, "The Neural Algorithm for Sound Localization in the Owl," in *The Harvey Lectures*, Series 86, pp. 47-64 (Wiley-Liss Inc., NY, 1992).
- [6] J. Blauert, *Spatial Hearing* (M.I.T. Press, Cambridge, MA 1983).
- [7] C. Lim and R.O.Duda, "Estimating the Azimuth and Elevation of a Sound Source from the Output of a Cochlear Model," in *Proc. 28th Asilomar Conference on Signals, Systems, and Computers* (Asilomar, CA, 1994).
- [8] R.O. Duda and William Chau, "Combined Monaural and Binaural Localization of Sound Sources," in *Proc. 29th Asilomar Conference on Signals, Systems, and Computers* (Asilomar, CA, 1995).
- [9] K.D. Martin, "A Computational Model of Spatial Hearing," Master's Thesis, M.I.T., 1995.
- [10] D. Nandy and J. Ben-Arie, "Estimating the Azimuth of a Sound Source from the Binaural Spectral Amplitude," *IEEE Trans. Speech and Audio Processing*, Jan. 1996.
- [11] R.H. Gilkey and T.R. Anderson, "The Accuracy of Absolute Localization Judgments for Speech Stimuli," *J. Vestibular Research*, Vol 5, No. 6, pp. 487-497, 1995.
- [12] W. A. Yost, *Fundamentals of Hearing: An Introduction*, Academic Press, NY, 1994.

- [13] R.D. Patterson et al., "Complex sounds and auditory images," in Y. Cazals et al., Eds., *Auditory Physiology and Perception*, pp. 429-446 (Pergammon, Oxford, 1992).
- [14] M. Slaney, "An Efficient Implementation of the Patterson-Holdsworth Auditory Filter Bank," Technical Report No. 35, Advanced Technology Group, Apple Computer, Inc., Cupertino CA 1993.
- [15] R.F. Lyon and C. Mead, "An Analog Electronic Cochlea," *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. 36, pp. 1119-1134, 1988.
- [16] T. Shahwan and R.O. Duda, "Adjacent-Channel Inhibition in Acoustic Onset Detection," in *Proc. 28th Asilomar Conference on Signals, Systems, and Computers* (Asilomar, CA, 1994).
- [17] M. Bodden, "Binaural Modeling and Auditory Scene Analysis," *Proc. 1995 IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, Mohonk Mountain House, (New Paltz, NY, 1995.)
- [18] J. Huang and N. Ohnishi, "Robot With Ears: Sound Localization and Separation," *Proceedings of ICSPAT'95*, pp.1697-1792, Oct. 1995

Appendix A

The Code

This appendix includes all the MATLAB code necessary to run the sound localization system. To save space and clarify any confusion, the general programs are listed here and the variations needed to implement the exact system are clearly explained.

cochlea

The following m-file function implements the cochlear model. It was originally designed for 70 channels, but the filter for the 70th channel did not fit the desired model so it should never be used. In other words the forward and feedback matrices of coefficients are accurate to the 69th row. It should be noted that `cochlea_low` and `cochlea_high` (not `cochlea`) are the m-file functions actually called in the localization programs. Nevertheless, `cochlea_low` and `cochlea_high` are just variations of `cochlea`. The `cochlea_low` program just calculates the lower 22 channels, and `cochlea_high` calculates the upper 38 channels.

```
function [y , zf_a, zf_b] = cochlea(sig, zi_a, zi_b, forward, feedback);
% This function takes an input audio signal and filters
% it through a cochlear model consisting of a filter bank.
% There are 70 filters in the bank with center frequencies
% ranging from 45 Hz to 21 kHz. The coefficients used for these
% filters assumes a sampling rate of 44.1 kHz. The input signal
% should be 2205 samples long. This corresponds to 50 ms of signal
% sampled at 44.1 kHz. The signal may be shorter than 2205 samples,
% but if it is any longer an error will occur. If you wish to process
% more than 50 ms of signal, you may by calling cochlea in a loop
```



```

% and using initial, and final conditions.
% If no initial conditions are specified the function defaults to
% zero initial conditions.

load forward;
load feedback;

zfa = zeros(70,8);
zfb = zeros(70,1);

if( length(sig) > 2205)
    fprintf('Error: Signal too long must be less than 2205 samples. ');
    break;
end;
sigout = zeros(70,length(sig));

% Make low pass filter
fs = 44100;
tau = 0.001; % time constant
a2 = -exp((-1/(fs*tau))); % For low pass filter

% Filter through bandpass
for i = 1:70,
    if( i <= 10) % Lowest 23 channels 33 Hz to 1000 Hz
        % the lower end filters are cascaded.
        a = feedback(i,1:5);
        b = forward(i,1:3);
        [sigout(i,:), zfa(i,1:4)] = filter( b, a, sig, zi_a(i,1:4) );
        zia(i,:) = zfa(i,:);
        [sigout(i,:), zfa(i,1:4)] = filter(b,a, sigout(i,:), zi_a(i,1:4) );
    else
        if( i <= 22)
            a = feedback(i,:);
            b = forward(i,:);
            [sigout(i,:), zfa(i,:)] = filter(b,a, sig, zi_a(i,:));
        else
            a = feedback(i,:);
            b = forward(i,:);
            [sigout(i,:), zfa(i,:)] = filter(b,a, sig, zi_a(i,:));
        end;
    end;

% Half wave rectify

sigout(i,:) = max( sigout(i,:), 1e-15);

```

% Low pass filter

```
[sigout(i,:), zfb(i,1)] = filter( 1+ a2, [1, a2], sigout(i,:), zi_b(i,1));
```

```
end;
```

```
zf_a = zfa;
```

```
zf_b = zfb;
```

```
y = sigout;
```

find_peaks

The following code extracts peaks from the oscillating data. It is listed in the localization programs under the name `find_peaks_ITD`.

```
function y = find_peaks(sig, epsilon);  
% This function picks out the peaks in a 'wavy' data array.  
% Epsilon is the a measure of how sensitive the peak detector should  
% be. For example if you want to catch all peaks that have a height of  
% 0.1, epsilon should be set to that value.  
TRUE = 1;  
FALSE = 0;  
peaks = zeros(1,length(sig));  
high = [ 0 1 ];  
low = [ 0 1 ];  
peak_next_corner_h = [ 0 1];  
peak_next_corner_l = [ 0 1];  
last = 1;  
threshold = 1e-5;  
out = 0;  
for i = 1:length(sig),  
    present = sig(1,i);  
    if( i > 2)  
        previous = sig(1, i-1);  
    else  
        previous = 0;  
    end;  
    if ( abs(present) < threshold & abs(previous) > threshold )  
        last = i;  
    end;  
    if( abs(present) < threshold )
```

```

for in2 = last:min((last+30), length(sig)),
    if( abs(sig(1,in2)) > threshold )
        out = i;
        end;
    end;
    if( out ~= i)
        high = [ 0 i ];
    low = [ 0 i ];
        peak_next_corner_l = [FALSE i];
        peak_next_corner_h = [FALSE i];
    end;
end;

if( present > high(1))
    high = [present i];

    if(high(1) - low(1) > epsilon)
        peak_next_corner_h = [TRUE i];
    end;
end;
if( present < low(1))
    low = [present i];
    if(high(1) - low(1) > epsilon)
        peak_next_corner_l = [TRUE i];
    end;
end;

if( peak_next_corner_h(1) & abs(high(1) - present) > epsilon )
    peaks(1,high(2)) = 1;
    peak_next_corner_h(1) = FALSE;
    low = [present i];
end;
if( peak_next_corner_l(1) & abs(present - low(1)) > epsilon )
    peaks(1,low(2)) = -1;
    peak_next_corner_l(1) = FALSE;
    high = [present i];
end;

end;
index_positive_peaks = find( peaks == 1 );

y = index_positive_peaks;

```

localizeITD

This program implements the entire sound localization system using only the ITD cue for localization.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   File:          LocalizeITD.m
%
%   Description:   This m-file takes as its input a binaural audio signal.
%                   The user is prompted to give the name of the file that
%                   contains the binaural signal. This file should be
%                   have a sampling rate of 44.1kHz, and saved in an .aif
%                   format. The program estimates the azimuth position of
%                   of the sound source(s) using ITD.
%
%   Author:       Nathaniel Henderson
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;
TRUE = 1;
FALSE = 0;

% Read in sound file to be processed

sig_name = input('Enter the name of the ITD sound file to be processed: ', 's');
sig = readsound(sig_name);
fprintf([sprintf('\n Running sig %s\n', sig_name)]);
.

LEFT = 1;
RIGHT = 2;

% Note: The orientation MUST be checked because, readsound might
%       flip right and left depending on how the aif file was produced.

check_plot = TRUE;
if(check_plot),
    plot(sig(LEFT,:));

```

```

hold on;
plot(sig(RIGHT,:) + 1, 'r');
title('Check for correct orintation');
hold off;
grid on;
pause(1.0);
end;

fs          = 44100;          % Sampling frequency
time_step   = 0.05;          % Filter a 50 ms chunk
npts        = fs * time_step; % Number of points to be filtered each loop

% Time over which to filter
t1 = 1;
t2 = npts;
if ( t2 > length(sig(1,:)) )
    t2 = length(sig(1,:));
end;

% Initial conditions for cochlear bank
zia_l = zeros(22,8);
zib_l = zeros(22,1);
zia_r = zeros(22,8);
zib_r = zeros(22,1);

numchannels_low = 22; % Number of filters that will be used for ITD

load forward; % Feed-forward coefficients for filters.
load feedback; % Feedback coefficients for filters.
load center; % Center frequencies of filters in the bank.
load bandwidth; % Bandwidths of filters in the bank.

% Initialize last peak information for ITD
last_peak_i_left = zeros(numchannels_low,1);
last_peak_i_right = zeros(numchannels_low,1);
last_peak_v_left = zeros(numchannels_low,1) + 1e-5;
last_peak_v_right = zeros(numchannels_low,1) + 1e-5;

more_signal = TRUE;
loop_number = 1;
total_loops = ceil(length(sig(1,:))/npts);

epsilon = 0.5e-4; % Used for peak dectection. = -86 dB
max_delay = 41; % Corresponds to 1220 Hz.

ITD_maps = zeros( total_loops, 2*max_delay + 1);

```

```

use_previous_l = FALSE; % use previous peaks for ITD measurement
use_previous_r = FALSE; % use previous peaks for ITD measurement
err_con       = FALSE;

```

```

energy_low_r = zeros(numchannels_low, total_loops-1);
energy_low_l = zeros(numchannels_low, total_loops-1);

```

```

% Begin the looping operation
while( more_signal == TRUE),

```

```

    % Filter the signal with low end cochlear filter
    % There are 22 bandpass filters in cochlea_low
    % with center frequencies ranging from 45 Hz to 902 Hz

```

```

[out_left , zfa_l, zfb_l] = cochlea_low( sig( LEFT, t1:t2), zia_l, ...
                                         zib_l,forward,feedback);
[out_right, zfa_r, zfb_r] = cochlea_low( sig(RIGHT, t1:t2), zia_r, ...
                                         zib_r,forward,feedback);

```

```

zia_l = zfa_l;
zib_l = zfb_l;
zia_r = zfa_r;
zib_r = zfb_r;

```

```

% ITD
ITD          = zeros(numchannels_low, 60);
onset_fuzz   = zeros(numchannels_low, 60);
column_onset = zeros(1, 60);

```

```

for i = 1:numchannels_low,

```

```

    yl = find_peaks_ITD(out_left( i, :), epsilon);
    yr = find_peaks_ITD(out_right(i, :), epsilon);

```

```

    % Energies used for coming probs with IID and IED
    energy_low_r(i,loop_number) = mean(20*log10(out_right(i,:)));
    energy_low_l(i,loop_number) = mean(20*log10(out_left(i,:)));

```

```

    % Make unbiased ITD measurements from peaks
    index      = max( length(yl), length(yr) );
    shift_left = 0;
    shift_right = 0;

```

```

    for i2 = 1:index,

```

```

if( shift_right + i2 > length(yr) | shift_left + i2 > length(yi))
    break;
end;
while( abs(yr(i2 + shift_right) - yi(i2 + shift_left)) > max_delay )
    if( loop_number ~= 1 & i2 == 1) % Check if last values work
        if( (yr(1) - last_peak_i_left(i)) < max_delay)
            use_previous_l = TRUE;
            break;
        end;
        if( ( yi(1) - last_peak_i_right(i)) < max_delay)
            use_previous_r = TRUE;
            break;
        end;
    end;
    if( yr(i2 + shift_right) > yi(i2 + shift_left) ) % Make positive left shift
        shift_left = shift_left + 1;
        if( shift_left + i2 > length(yi) )
            break;
        end;
    else
        shift_right = shift_right + 1; % Make positive right shift
        if( shift_right + i2 > length(yr))
            break;
        end;
    end;
    if( shift_right > 10 | shift_left > 10)
        break;
    end;
end; % while

% Make ITD measurment
if( shift_right + i2 <= length(yr) & shift_left + i2 <= length(yi) & ...
    shift_right < 10 & shift_left < 10 )

    if(use_previous_l & i2 == 1)
        ITD(i,1) = yr(1) - last_peak_i_left(i) ;
        if( abs(ITD(i,1)) > max_delay)
            err_con = TRUE;
            fprintf('Error! L\n');
            break;
        end;
    else if(use_previous_r & i2 == 1)
        ITD(i,1) = -(yi(1) - last_peak_i_right(i));
        if( abs(ITD(i,1)) > max_delay)
            err_con = TRUE;
            fprintf('Error R\n');
            break;
        end;
    end;
end;

```

```

        end;
    else
        ITD(i,i2) = yr(i2 + shift_right) - yl(i2 + shift_left);
        if( abs(ITD(i,1)) > max_delay)
            err_con = TRUE;
            fprintf('Found it after 1st!\n');
            break;
        end;
    end;
end;
end;
% Negative ITD values mean that the signal is on the right side

use_previous_r = FALSE;
use_previous_l = FALSE;

% Fuzzy onset
if ( (i2 + shift_left) == 1),
    onset_time_fuzz_l = 20*log10(((yl(i2+ shift_left) - ...
                                   last_peak_i_left(i))/fs)*center(i));
    onset_mag_fuzz_l = 20*log10( out_left(i,yl(i2+ ...
                                   shift_left))/last_peak_v_left(i) );
else
    onset_time_fuzz_l = 20*log10(((yl(i2+ shift_left) - yl(i2+ ...
                                   shift_left - 1))/fs)*center(i));
    onset_mag_fuzz_l = 20*log10( out_left(i,yl(i2+ shift_left))/ ...
                                   out_left(i,yl(i2+ shift_left-1)));
end;
if ( (i2 + shift_right) == 1),
    onset_time_fuzz_r = 20*log10(((yr(i2+ shift_right) - ...
                                   last_peak_i_right(i))/fs)*center(i));
    onset_mag_fuzz_r = 20*log10( out_right(i,yr(i2+ ...
                                   shift_right))/last_peak_v_right(i) );
else
    onset_time_fuzz_r = 20*log10(((yr(i2+ shift_right) - yr(i2+ ...
                                   shift_right - 1))/fs)*center(i));
    onset_mag_fuzz_r = 20*log10( out_right(i,yr(i2+ shift_right))/ ...
                                   out_right(i,yr(i2+ shift_right-1)));
end;

onset_fuzz(i,i2) = ((onset_time_fuzz_r + onset_time_fuzz_l)/2 + ...
                    (onset_mag_fuzz_l + onset_mag_fuzz_r)/2);

if( onset_fuzz(i,i2) < 0),
    onset_fuzz(i,i2) = 0;
end;

end;

```



```

end; % for i2 = 1:index

    if( err_con) break; end;
    last_peak_i_right(i) = yr(length(yr)) - length(out_right);
    last_peak_v_right(i) = out_right(i, yr(length(yr)));
    last_peak_i_left(i)   = yl(length(yl)) - length(out_left);
    last_peak_v_left(i)   = out_left(i, yl(length(yl))); ;
    clear yl;
    clear yr;
end; %for i = 1:numchannels_low,

if( err_con) break; end;

% Give extra weight for cross-channel onsets
sum_onsets = sum(onset_fuzz);
sum_onsets = sum_onsets./ numchannels_low;

for i = 1:index % over columns
    row_count = 0;
    for i2 = 1:numchannels_low % over rows
        if( onset_fuzz(i2,i) > 0)
            row_count = row_count + 1;
        end;
    end;
    column_onset(1,i) = row_count;
end;

for i = 1:index,
    for i2 = 1:numchannels_low,
        if ( onset_fuzz(i2,i) > 0 )
            onset_fuzz(i2, i) = column_onset(1,i) + sum_onsets(1,i) + ...
                                onset_fuzz(i2, i) ;
        end;
    end;
end;

% Weight coincidence over time
for i2 = 1:numchannels_low,
    for i = 1:index
        if ( i == 1 )
            diff = abs(ITD(i2,i) - ITD(i2, i + 1) );
            if(diff <= 5)
                y = -(2/5)*diff + 20;
            else if( diff <= 7 )
                y = -(9)*diff + 63;
            else
                y = 0;
            end
        end
    end
end

```

```

        end;
    end;
    if ( onset_fuzz(i2,i) > 0 )
        onset_fuzz(i2, i) = onset_fuzz(i2, i) + y;
    end;
else
    y_sum = 0;
    for i3 = 1:2,
        if ( i3 == 1)
            diff = abs(ITD(i2,i) - ITD(i2, i - 1) );
        else
            diff = abs(ITD(i2,i) - ITD(i2, i + 1) );
        end;
        if(diff <= 3)
            y = -(2/3)*diff + 10;
        else if( diff <= 5 )
            y = -(4)*diff + 20;
        else
            y = 0;
        end;
    end;
    y_sum = y_sum + y;
end; % for( i3)
if ( onset_fuzz(i2,i) > 0 )
    onset_fuzz(i2, i) = onset_fuzz(i2, i) + y_sum;
end;
end; % if( i == 1)
end;
end;

timing = zeros(1, 2*max_delay + 1);
error_number = zeros(numchannels_low,50);
err_con = FALSE;
for i = 1:numchannels_low,
    for i2 = 1:50,
        number = ITD(i,i2);
        weight = onset_fuzz(i,i2);
        if(abs(number) > max_delay )
            error_number(i,i2) = number;
            err_con = TRUE;
            break;
        end;
        if( weight ~= 0 )
            timing(1,number + max_delay + 1) = timing(1,number + ...
                max_delay + 1) + weight;
        end;
    end;
end;

```

```

        if( err_con), break, end;
    end;

    if( err_con), break, end;
    ITD_maps(loop_number, :) = timing;
    clear timing;
    ITD_maps_to_prob_pl(ITD_maps(loop_number, :) / ...
                        sum(ITD_maps(loop_number, :)) );
    title(sprintf('ITD Confidence = %d', ...
                  round(max(ITD_maps(loop_number, :)))));
    pause(0.1);

% Set up for next time chunk
t1 = t2 + 1;
t2 = t2 + npts;
if ( t2 > length( sig(1,:)) )
    more_signal = FALSE;
end;
fprintf('Loop = %d\n',loop_number);
loop_number = loop_number + 1;

end; % while( more_signal == TRUE),

% Sum over ITD_maps with confidence levels over 200
itd = zeros(1,83);
for i = 1: loop_number-1,
    s = max(ITD_maps(i,:));
    if ( s >= 200)
        itd = itd + ITD_maps(i,:);
    end;
end;
itd = itd./ sum(itd);

% Calculate energy_low for combining maps with IID and IED
energy_low = 0;
if( total_loops == 1) total_loops = 2; end;
for i = 1:numchannels_low,
    energy_low = mean([energy_low_l(i,1:total_loops-1) ...
                      energy_low_r(i,1:total_loops-1)]) + energy_low;
end;
energy_low = energy_low/numchannels_low;

% Convnet ltd to probabilities
itd_probs = ITD_maps_to_prob(itd);

% Save overall itd probabilities for combining with IED and IID
save itd_probs itd_probs;

```

```
save energy_low energy_low;
```

ITD_maps_to_prob

The following code converts the sample delays in the ITD matrix into directional probabilities. A second version of it seen in `localizeITD` and `localizeIED`, `ITD_maps_to_prob_pl` is the same except that it also plots the resulting probabilities.

```
function y = ITD_maps_to_prob(itd2);
% Convert ITD_maps into plots of probs
prob_az = zeros(1,36);
prob_az(10) = mean(itd2(1,1:7));
prob_az(11) = mean(itd2(1,8:11));
prob_az(9) = mean(itd2(1,8:11));
prob_az(12) = mean(itd2(1,12:15));
prob_az(8) = mean(itd2(1,12:15));
prob_az(13) = mean(itd2(1,16:19));
prob_az(7) = mean(itd2(1,16:19));
prob_az(14) = mean(itd2(1,20:23));
prob_az(6) = mean(itd2(1,20:23));
prob_az(15) = mean(itd2(1,24:27));
prob_az(5) = mean(itd2(1,24:27));
prob_az(16) = mean(itd2(1,28:31));
prob_az(4) = mean(itd2(1,28:31));
prob_az(17) = mean(itd2(1,32:35));
prob_az(3) = mean(itd2(1,32:35));
prob_az(18) = mean(itd2(1,36:39));
prob_az(2) = mean(itd2(1,36:39));
prob_az(19) = mean(itd2(1,40:44));
prob_az(1) = mean(itd2(1,40:44));
prob_az(20) = mean(itd2(1,45:48));
prob_az(36) = mean(itd2(1,45:48));
prob_az(21) = mean(itd2(1,49:52));
prob_az(35) = mean(itd2(1,49:52));
prob_az(22) = mean(itd2(1,53:56));
prob_az(34) = mean(itd2(1,53:56));
```

```

prob_az(23) = mean(itd2(1,57:60));
prob_az(33) = mean(itd2(1,57:60));
prob_az(24) = mean(itd2(1,61:64));
prob_az(32) = mean(itd2(1,61:64));
prob_az(25) = mean(itd2(1,65:68));
prob_az(31) = mean(itd2(1,65:68));
prob_az(26) = mean(itd2(1,69:72));
prob_az(30) = mean(itd2(1,69:72));
prob_az(27) = mean(itd2(1,73:76));
prob_az(29) = mean(itd2(1,73:76));
prob_az(28) = mean(itd2(1,77:83));

```

```

y = prob_az ./ sum(prob_az);

```

localizedED

The only substantial difference between the localizedED program and the localizedTD program is that localizedED uses cochlea_high and 38 channels instead of using cochlea_low with 22 channels.

localizedIID

This program implements the entire sound localization system using only the ITD cue for localization.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                 %
%   File:   localizedIID.m                                         %
%                                                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

TRUE   = 1;
FALSE  = 0;
LEFT   = 1;
RIGHT  = 2;

```

```

% Read in sound file to be processed
sig_name = input('Enter the name of the IID sound file to be ...

```

```

                                processed: ', 's');

sig = readsound(sig_name);

% Note: The orientation MUST be checked because, readsound will
%       flip right and left depending on how the aif file was
%       produced.

check_plot = TRUE;
if(check_plot),
    plot(sig(LEFT,:));
    hold on;
    plot(sig(RIGHT,:) + 1, 'r');
    title('Check for correct orintation');
    hold off;
    grid on;
    pause;
end;

fs      = 44100;           % Sampling frequency
time_step = 0.05;         % Filter a 100 ms chunk
npts    = fs * time_step; % Number of points to be filtered each loop

% Time over which to filter
t1 = 1;
t2 = npts;
if ( t2 > length(sig(1,:)) )
    t2 = length(sig(1,:));
end;

numchannels_high = 38; % Number of filters that will be used for IID

% Initial conditions for cochlear bank
zia_l = zeros(numchannels_high,8);
zib_l = zeros(numchannels_high,1);
zia_r = zeros(numchannels_high,8);
zib_r = zeros(numchannels_high,1);

load forward; % Feed-forward coefficients for filters.
load feedback; % Feedback coefficients for filters.
load center; % Center frequencies of filters in the bank.
load bandwidth; % Bandwidths of filters in the bank.

coin_param = 40; % Coincidence Parameter

```

```

energy_param = -90;    % Energy Parameter (dB)

load IID_noecho;        % Table of IID maps from Kemar impulse
                        % response made in an anechoic environment

% Impulse responses were only measured from one side of Kemar's
% head. Symmetry is assumed to complete the IID map

for i = 1:36,
    if (i <= 19) for
        IID_map(i,:) = IID_noecho(i,:);
    else
        IID_map(i,:) = -IID_noecho(38 - i, :);
    end;
end;

% theta is needed for for polar plot
for i = 1:36,
    if(i <= 10)
        theta(i) = 90 - (i-1)*10;
    else
        theta(i) = 360 - (i-10)*10;
    end;
end;
theta = theta*(2*pi/360);

more_signal    = TRUE;
loop_number    = 1;
total_loops    = fix(length(sig(1,:))/npts );
if( total_loops == 0) total_loops = 1; end;

out_left  = zeros(numchannels_high, npts);
out_right = zeros(numchannels_high, npts);

energy_high_l = zeros(numchannels_high, total_loops-1);
energy_high_r = zeros(numchannels_high, total_loops-1);

total_prob = zeros(1,36);
total_sum = 0;

% Begin the looping operation
while( more_signal == TRUE),

    % Filter the signal with high end cochlear filter
    % There are 38 bandpass filters in cochlea_low
    % with center frequencies ranging from 2000 to 20000 Hz

```

```
[out_left , zfa_l, zfb_l] = cochlea_high( sig( LEFT, t1:t2), zia_l, ...
                                         zib_l,forward,feedback);
[out_right, zfa_r, zfb_r] = cochlea_high( sig(RIGHT, t1:t2), zia_r, ...
                                         zib_r,forward,feedback);
```

```
zia_l = zfa_l;
zib_l = zfb_l;
zia_r = zfa_r;
zib_r = zfb_r;
```

```
coin_energy = 0;
```

```
for i = 1:numchannels_high,
```

```
    % Energies used for combining probs with ITD
    energy_high_r(i,loop_number) = mean(20*log10(out_right(i,:)));
    energy_high_l(i,loop_number) = mean(20*log10(out_left(i,:)));
```

```
    % Decimate by 44
```

```
    out_right_d(i,:) = decimate(out_right(i,:), 44, 'FIR');
    out_left_d(i,:) = decimate( out_left(i,:), 44, 'FIR');
```

```
    % Take logs
```

```
    out_right_d(i,:) = 20*log10( out_right_d(i,:) );
    out_left_d(i,:) = 20*log10( out_left_d(i,:) );
```

```
    % Set minimum value to energy_param ( -90 dB)
```

```
    for tt = 1:length(out_right_d(i,:)),
        if(out_right_d(i,tt) < energy_param)
            out_right_d(i,tt) = energy_param;
        end;
    end;
    for tt = 1:length(out_left_d(i,:)),
        if(out_left_d(i,tt) < energy_param)
            out_left_d(i,tt) = energy_param;
        end;
    end;
```

```
    IID_measured = out_right_d(i,:) - out_left_d(i,:);
    ave_energy_r = mean(out_right_d(i,:));
    ave_energy_l= mean(out_left_d(i,:));
```

```
    % Energy Onsets weights
```

```
    [ max_r index_maxr ] = max(out_right_d(i,:));
    [ min_r index_minr ] = min(out_right_d(i,:));
```



```

if( (index_maxr > index_minr) )
    onset_weight_r = abs( max_r - min_r);
else
    onset_weight_r = 0;
end;

[ max_l index_maxl ] = max(out_left_d(i,:));
[ min_l index_minl ] = min(out_left_d(i,:));
if( index_maxl > index_minl)
    onset_weight_l = abs( max_l - min_l);
else
    onset_weight_l = 0;
end;

% Choose IID calculate channel weights
if(max(IID_measured) >= abs(min(IID_measured)))
    IID_test(i) = max(IID_measured);

    channel_fuzzy_weights(i) = exp(energy_param./ ...
        ave_energy_r) + ...
        onset_weight_r + ...
        abs( max(IID_measured) - ...
            mean(IID_measured)));
else
    IID_test(i) = min( IID_measured );
    channel_fuzzy_weights(i) = exp(energy_param./ ...
        ave_energy_l) + ...
        onset_weight_l + ...
        abs( min(IID_measured) - ...
            mean(IID_measured)));
end;
if(channel_fuzzy_weights(i) > coin_param)
    coin_energy = coin_energy + 1;
end;

end; % looping through the channels

```

```

sigma = 5;
for az = 1:36,
    for chan = 1:numchannels_high,
        probs_IID(chan, az) = (1/sigma*sqrt(2*pi))*exp( -0.5 * ...
            ((IID_test(chan)-IID_map(az,chan))^2)/ ...
            sigma^2);
    end;
end;

```

```

    end;
end;
prob_map = zeros(1,36);
for chan = 1:numchannels_high
    prob_map = prob_map + (probs_IID(chan,:) * channel_fuzzy_weights(i));
end;

s = round(sum(channel_fuzzy_weights));
total_prob = prob_map*(s+ 20* coin_energy) + total_prob;
polar2([theta pi/2],[prob_map)/s prob_map(1)/s])
title(sprintf('IID Confidence = %d', ...
    round(sum(channel_fuzzy_weights))+10* coin_energy));
pause(0.1);

% Set up for next time chunk
t1 = t2 + 1;
t2 = t2 + npts;
if ( t2 > length( sig(1,:)) )
    more_signal = FALSE;
end;

fprintf('Loop = %d\n',loop_number);

loop_number = loop_number + 1;

end; % while( more_signal == TRUE),

iid_probs = total_prob ./ sum(total_prob );
polar2([theta pi/2],[total_prob ./ sum(total_prob )] ) total_prob(1)
./sum(total_prob ) ] );
energy_high = 0;
if(total_loops == 1)
    total_loops = 2;
end;
for i = 1:numchannels_high,
    energy_high = mean([energy_high_l(i,1:total_loops-1) ...
        energy_high_r(i,1:total_loops-1)]) ...
        + energy_high;
end;
energy_high = energy_high/numchannels_high;

save iid_probs iid_probs;
save energy_high energy_high;

```

combine_probs

The following program combines the probabilities from the ITD, IID and IED programs to produce a single probability map.

```
% Combine Probs
load itd_probs;
load ied_probs;
load iid_probs;
load energy_low;
load energy_high;

front_back_r = max(iid_probs(1,2:9)) / max(iid_probs(1,11:18)) ...
    + sum(iid_probs(1,2:9)) / sum(iid_probs(1,11:18)) ;

front_back_l = max(iid_probs(1,29:36)) / max(iid_probs(1,20:27)) ...
    + sum(iid_probs(1,29:36)) / sum(iid_probs(1,20:27)) ;
magic_num = 2.4;
% Note this magic_num is a poor way to combine probabilities

right_side = sum([ itd_probs(1,2:18) ied_probs(1,2:18) iid_probs(1,2:18) ]);
left_side = sum([ itd_probs(1,20:36) ied_probs(1,20:36) iid_probs(1,20:36) ]);

right_left_ratio = right_side/left_side; % Needed to detect different sources on
                                         % right and left sides on the head

% Work on right side
if( right_left_ratio > 0.7)
    if(front_back_r >= magic_num)
        front_r = (5/2)*(front_back_r - magic_num) + 1;
        if (front_r > 2) front_r = 2; end;
        back_r = 1;
    else
        front_r = 1;
        back_r = 10*(magic_num - front_back_r) + 1;
        if (back_r > 3) back_r = 3; end;
    end;

itd_probs(1,2:10) = front_r*itd_probs(1,2:10);
ied_probs(1,2:10) = front_r*ied_probs(1,2:10);
itd_probs(1,10:18) = back_r*itd_probs(1,10:18);
ied_probs(1,10:18) = back_r*ied_probs(1,10:18);
iid_probs(1,10:18) = back_r*iid_probs(1,10:18);
end;
```

```

% Work on left side
if( right_left_ratio < 1.3)
    if(front_back_l >= magic_num)
        front_l = (5/2)*(front_back_l - magic_num) + 1;
        if (front_l > 2) front_l = 2; end;
        back_l = 1;
    else
        front_l = 1;
        back_l = 10*(magic_num - front_back_l) + 1;
        if (back_l > 3) back_l = 3; end;
    end;

    itd_probs(1,28:36) = front_l*itd_probs(1,28:36);
    ied_probs(1,28:36) = front_l*ied_probs(1,28:36);

    itd_probs(1,20:28) = back_l*itd_probs(1,20:28);
    ied_probs(1,20:28) = back_l*ied_probs(1,20:28);
    iid_probs(1,20:28) = back_l*iid_probs(1,20:28);
end;

% Energy factor will be positive if there is more energy in the
% higher bands
energy_factor = energy_high - energy_low;
if(energy_factor < -1 )
    if(abs(energy_factor) > 6) energy_factor = -6; end;
    prob = (abs(energy_factor) .* 2 .* itd_probs ) + ied_probs + iid_probs;
end;
if(energy_factor > 1 )
    if(abs(energy_factor) > 6) energy_factor = 6; end;
    prob = 2 .* itd_probs + energy_factor .* (ied_probs + iid_probs);
end;
if( abs(energy_factor) <= 1 )
    prob = 2*itd_probs + ied_probs + iid_probs;
end;

prob = prob ./ sum(prob);
for i = 1:36,
    if(i <= 10)
        theta(i) = 90 - (i-1)*10;
    else
        theta(i) = 360 - (i-10)*10;
    end;
end;
theta = theta*(2*pi/360);
polar3(theta , prob, 'auto');

```

Filter coefficients

The following filter coefficients were used to implement the gammatone filters.

The first 10 filters were cascaded.

forward coefficients =

9.1745e-05	-1.8349e-04	9.1745e-05	0	0
9.9513e-05	-1.9903e-04	9.9513e-05	0	0
1.2042e-04	-2.4085e-04	1.2042e-04	0	0
1.3377e-04	-2.6755e-04	1.3377e-04	0	0
1.6792e-04	-3.3585e-04	1.6792e-04	0	0
2.1169e-04	-4.2337e-04	2.1169e-04	0	0
2.4148e-04	-4.8296e-04	2.4148e-04	0	0
2.5397e-04	-5.0795e-04	2.5397e-04	0	0
2.9321e-04	-5.8642e-04	2.9321e-04	0	0
3.4259e-04	-6.8518e-04	3.4259e-04	0	0
1.3296e-08	-5.2638e-08	7.7952e-08	-5.1178e-08	1.2568e-08
1.6940e-08	-6.7009e-08	9.9097e-08	-6.4934e-08	1.5906e-08
2.1582e-08	-8.5291e-08	1.2594e-07	-8.2338e-08	2.0111e-08
2.7493e-08	-1.0854e-07	1.5998e-07	-1.0433e-07	2.5399e-08
3.5020e-08	-1.3811e-07	2.0316e-07	-1.3211e-07	3.2037e-08
4.4605e-08	-1.7569e-07	2.5786e-07	-1.6713e-07	4.0352e-08
5.6807e-08	-2.2345e-07	3.2713e-07	-2.1123e-07	5.0741e-08
7.2341e-08	-2.8412e-07	4.1477e-07	-2.6667e-07	6.3687e-08
9.2113e-08	-3.6118e-07	5.2557e-07	-3.3626e-07	7.9764e-08
1.1728e-07	-4.5901e-07	6.6549e-07	-4.2340e-07	9.9656e-08
1.4930e-07	-5.8316e-07	8.4201e-07	-5.3228e-07	1.2416e-07
1.9003e-07	-7.4064e-07	1.0644e-06	-6.6795e-07	1.5418e-07
2.4186e-07	-9.4029e-07	1.3442e-06	-8.3647e-07	1.9074e-07
3.0777e-07	-1.1933e-06	1.6957e-06	-1.0450e-06	2.3494e-07
3.9159e-07	-1.5137e-06	2.1364e-06	-1.3021e-06	2.8790e-07
4.9816e-07	-1.9191e-06	2.6879e-06	-1.6174e-06	3.5064e-07
6.3362e-07	-2.4318e-06	3.3764e-06	-2.0018e-06	4.2398e-07
8.0578e-07	-3.0797e-06	4.2337e-06	-2.4672e-06	5.0819e-07
1.0245e-06	-3.8975e-06	5.2977e-06	-3.0261e-06	6.0267e-07
1.3024e-06	-4.9290e-06	6.6138e-06	-3.6905e-06	7.0529e-07
1.6553e-06	-6.2283e-06	8.2349e-06	-4.4703e-06	8.1161e-07
2.1033e-06	-7.8630e-06	1.0222e-05	-5.3711e-06	9.1359e-07
2.6719e-06	-9.9169e-06	1.2645e-05	-6.3901e-06	9.9799e-07
3.3934e-06	-1.2493e-05	1.5579e-05	-7.5109e-06	1.0440e-06
4.3086e-06	-1.5719e-05	1.9104e-05	-8.6945e-06	1.0204e-06
5.4691e-06	-1.9751e-05	2.3298e-05	-9.8686e-06	8.8137e-07
6.9400e-06	-2.4779e-05	2.8229e-05	-1.0910e-05	5.6181e-07
8.8037e-06	-3.1031e-05	3.3941e-05	-1.1625e-05	-2.8606e-08
1.1164e-05	-3.8786e-05	4.0431e-05	-1.1714e-05	-1.0121e-06
1.4152e-05	-4.8371e-05	4.7617e-05	-1.0741e-05	-2.5497e-06
1.7933e-05	-6.0174e-05	5.5292e-05	-8.0758e-06	-4.8471e-06
2.2714e-05	-7.4645e-05	6.3054e-05	-2.8396e-06	-8.1578e-06

2.8759e-05	-9.2298e-05	7.0213e-05	6.1672e-06	-1.2779e-05
3.6394e-05	-1.1371e-04	7.5663e-05	2.0540e-05	-1.9039e-05
4.6035e-05	-1.3949e-04	7.7711e-05	4.2343e-05	-2.7257e-05
5.8199e-05	-1.7027e-04	7.3850e-05	7.4158e-05	-3.7688e-05
7.3536e-05	-2.0668e-04	6.0476e-05	1.1909e-04	-5.0407e-05
9.2861e-05	-2.4919e-04	3.2535e-05	1.8064e-04	-6.5158e-05
1.1719e-04	-2.9808e-04	-1.6882e-05	2.6251e-04	-8.1119e-05
1.4780e-04	-3.5319e-04	-9.6998e-05	3.6799e-04	-9.6613e-05
1.8627e-04	-4.1369e-04	-2.1979e-04	4.9911e-04	-1.0877e-04
2.3459e-04	-4.7768e-04	-4.0033e-04	6.5515e-04	-1.1320e-04
2.9520e-04	-5.4165e-04	-6.5691e-04	8.3060e-04	-1.0382e-04
3.7117e-04	-5.9978e-04	-1.0106e-03	1.0123e-03	-7.3060e-05
4.6628e-04	-6.4294e-04	-1.4841e-03	1.1760e-03	-1.2697e-05
5.8520e-04	-6.5743e-04	-2.0989e-03	1.2827e-03	8.4170e-05
7.3373e-04	-6.2336e-04	-2.8709e-03	1.2758e-03	2.1973e-04
9.1899e-04	-5.1268e-04	-3.8030e-03	1.0807e-03	3.8646e-04
1.1498e-03	-2.8675e-04	-4.8739e-03	6.1001e-04	5.6121e-04
1.4368e-03	1.0623e-04	-6.0231e-03	-2.2274e-04	7.0000e-04
1.7932e-03	7.3405e-04	-7.1323e-03	-1.4752e-03	7.3743e-04
2.2351e-03	1.6829e-03	-8.0047e-03	-3.1327e-03	5.9602e-04
2.7821e-03	3.0587e-03	-8.3458e-03	-5.0498e-03	2.1162e-04
3.4581e-03	4.9868e-03	-7.7560e-03	-6.8886e-03	-4.2251e-04
4.2922e-03	7.6081e-03	-5.7462e-03	-8.0808e-03	-1.2006e-03
5.3201e-03	1.1070e-02	-1.7955e-03	-7.8587e-03	-1.8652e-03
6.5878e-03	1.5512e-02	4.5307e-03	-5.4166e-03	-2.0311e-03
8.1649e-03	2.1072e-02	1.3407e-02	-2.4382e-04	-1.3275e-03
1.0259e-02	2.8154e-02	2.4777e-02	7.4936e-03	3.4527e-04
1.4817e-02	4.1927e-02	4.2889e-02	1.8718e-02	2.9149e-03

feedback coefficients =

Columns 1 through 6

1.0000e+00	-3.9808e+00	5.9425e+00	-3.9426e+00	9.8094e-01	0
1.0000e+00	-3.9799e+00	5.9399e+00	-3.9402e+00	9.8015e-01	0
1.0000e+00	-3.9777e+00	5.9335e+00	-3.9340e+00	9.7818e-01	0
1.0000e+00	-3.9762e+00	5.9294e+00	-3.9302e+00	9.7700e-01	0
1.0000e+00	-3.9731e+00	5.9205e+00	-3.9216e+00	9.7425e-01	0
1.0000e+00	-3.9695e+00	5.9101e+00	-3.9118e+00	9.7112e-01	0
1.0000e+00	-3.9670e+00	5.9032e+00	-3.9054e+00	9.6916e-01	0
1.0000e+00	-3.9657e+00	5.8998e+00	-3.9025e+00	9.6838e-01	0
1.0000e+00	-3.9627e+00	5.8914e+00	-3.8948e+00	9.6604e-01	0
1.0000e+00	-3.9592e+00	5.8818e+00	-3.8859e+00	9.6332e-01	0
1.0000e+00	-7.9179e+00	2.7438e+01	-5.4352e+01	6.7314e+01	-5.3374e+01
1.0000e+00	-7.9113e+00	2.7394e+01	-5.4227e+01	6.7119e+01	-5.3192e+01
1.0000e+00	-7.9040e+00	2.7346e+01	-5.4092e+01	6.6909e+01	-5.2995e+01
1.0000e+00	-7.8959e+00	2.7294e+01	-5.3945e+01	6.6680e+01	-5.2783e+01
1.0000e+00	-7.8871e+00	2.7236e+01	-5.3785e+01	6.6433e+01	-5.2555e+01
1.0000e+00	-7.8775e+00	2.7173e+01	-5.3610e+01	6.6164e+01	-5.2308e+01
1.0000e+00	-7.8668e+00	2.7104e+01	-5.3420e+01	6.5873e+01	-5.2042e+01
1.0000e+00	-7.8551e+00	2.7029e+01	-5.3212e+01	6.5557e+01	-5.1755e+01

1.0000e+00	-7.8422e+00	2.6946e+01	-5.2985e+01	6.5214e+01	-5.1444e+01
1.0000e+00	-7.8279e+00	2.6855e+01	-5.2738e+01	6.4840e+01	-5.1109e+01
1.0000e+00	-7.8122e+00	2.6755e+01	-5.2467e+01	6.4435e+01	-5.0747e+01
1.0000e+00	-7.7948e+00	2.6645e+01	-5.2171e+01	6.3993e+01	-5.0355e+01
1.0000e+00	-7.7756e+00	2.6525e+01	-5.1847e+01	6.3513e+01	-4.9931e+01
1.0000e+00	-7.7543e+00	2.6392e+01	-5.1492e+01	6.2991e+01	-4.9473e+01
1.0000e+00	-7.7308e+00	2.6246e+01	-5.1104e+01	6.2422e+01	-4.8977e+01
1.0000e+00	-7.7048e+00	2.6085e+01	-5.0680e+01	6.1803e+01	-4.8441e+01
1.0000e+00	-7.6759e+00	2.5907e+01	-5.0215e+01	6.1129e+01	-4.7860e+01
1.0000e+00	-7.6439e+00	2.5712e+01	-4.9706e+01	6.0396e+01	-4.7233e+01
1.0000e+00	-7.6085e+00	2.5497e+01	-4.9148e+01	5.9598e+01	-4.6554e+01
1.0000e+00	-7.5691e+00	2.5260e+01	-4.8538e+01	5.8730e+01	-4.5820e+01
1.0000e+00	-7.5254e+00	2.4999e+01	-4.7871e+01	5.7787e+01	-4.5027e+01
1.0000e+00	-7.4769e+00	2.4712e+01	-4.7142e+01	5.6762e+01	-4.4171e+01
1.0000e+00	-7.4230e+00	2.4396e+01	-4.6345e+01	5.5649e+01	-4.3247e+01
1.0000e+00	-7.3632e+00	2.4048e+01	-4.5474e+01	5.4442e+01	-4.2251e+01
1.0000e+00	-7.2967e+00	2.3665e+01	-4.4526e+01	5.3136e+01	-4.1180e+01
1.0000e+00	-7.2229e+00	2.3245e+01	-4.3493e+01	5.1723e+01	-4.0028e+01
1.0000e+00	-7.1408e+00	2.2784e+01	-4.2370e+01	5.0199e+01	-3.8792e+01
1.0000e+00	-7.0496e+00	2.2278e+01	-4.1153e+01	4.8558e+01	-3.7469e+01
1.0000e+00	-6.9484e+00	2.1725e+01	-3.9834e+01	4.6796e+01	-3.6056e+01
1.0000e+00	-6.8359e+00	2.1122e+01	-3.8412e+01	4.4910e+01	-3.4551e+01
1.0000e+00	-6.7110e+00	2.0463e+01	-3.6881e+01	4.2897e+01	-3.2955e+01
1.0000e+00	-6.5725e+00	1.9748e+01	-3.5239e+01	4.0758e+01	-3.1267e+01
1.0000e+00	-6.4188e+00	1.8973e+01	-3.3487e+01	3.8497e+01	-2.9490e+01
1.0000e+00	-6.2486e+00	1.8136e+01	-3.1625e+01	3.6117e+01	-2.7629e+01
1.0000e+00	-6.0600e+00	1.7236e+01	-2.9658e+01	3.3630e+01	-2.5692e+01
1.0000e+00	-5.8515e+00	1.6274e+01	-2.7593e+01	3.1048e+01	-2.3688e+01
1.0000e+00	-5.6211e+00	1.5250e+01	-2.5440e+01	2.8389e+01	-2.1632e+01
1.0000e+00	-5.3669e+00	1.4168e+01	-2.3214e+01	2.5675e+01	-1.9540e+01
1.0000e+00	-5.0870e+00	1.3035e+01	-2.0935e+01	2.2936e+01	-1.7432e+01
1.0000e+00	-4.7792e+00	1.1858e+01	-1.8625e+01	2.0205e+01	-1.5331e+01
1.0000e+00	-4.4417e+00	1.0651e+01	-1.6312e+01	1.7520e+01	-1.3265e+01
1.0000e+00	-4.0724e+00	9.4301e+00	-1.4028e+01	1.4925e+01	-1.1260e+01
1.0000e+00	-3.6697e+00	8.2167e+00	-1.1804e+01	1.2465e+01	-9.3456e+00
1.0000e+00	-3.2318e+00	7.0375e+00	-9.6740e+00	1.0190e+01	-7.5475e+00
1.0000e+00	-2.7578e+00	5.9245e+00	-7.6658e+00	8.1475e+00	-5.8884e+00
1.0000e+00	-2.2468e+00	4.9152e+00	-5.8016e+00	6.3851e+00	-4.3833e+00
1.0000e+00	-1.6992e+00	4.0521e+00	-4.0908e+00	4.9466e+00	-3.0368e+00
1.0000e+00	-1.1157e+00	3.3813e+00	-2.5256e+00	3.8716e+00	-1.8402e+00
1.0000e+00	-4.9881e-01	2.9504e+00	-1.0766e+00	3.1948e+00	-7.6902e-01
1.0000e+00	1.4788e-01	2.8057e+00	3.1046e-01	2.9462e+00	2.1713e-01
1.0000e+00	8.1870e-01	2.9868e+00	1.7139e+00	3.1515e+00	1.1721e+00
1.0000e+00	1.5058e+00	3.5213e+00	3.2299e+00	3.8310e+00	2.1568e+00
1.0000e+00	2.1988e+00	4.4172e+00	4.9589e+00	4.9951e+00	3.2285e+00
1.0000e+00	2.8841e+00	5.6543e+00	6.9829e+00	6.6339e+00	4.4254e+00
1.0000e+00	3.5451e+00	7.1765e+00	9.3347e+00	8.6980e+00	5.7490e+00
1.0000e+00	4.1615e+00	8.8839e+00	1.1963e+01	1.1073e+01	7.1469e+00
1.0000e+00	4.7093e+00	1.0630e+01	1.4700e+01	1.3550e+01	8.5029e+00
1.0000e+00	5.1616e+00	1.2227e+01	1.7250e+01	1.5815e+01	9.6410e+00
1.0000e+00	5.4886e+00	1.3452e+01	1.9207e+01	1.7463e+01	1.0350e+01
1.0000e+00	5.6591e+00	1.4083e+01	2.0128e+01	1.8070e+01	1.0434e+01

Columns 7 through 9

0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
2.6460e+01	-7.4985e+00	9.3000e-01
2.6358e+01	-7.4667e+00	9.2578e-01
2.6248e+01	-7.4328e+00	9.2131e-01
2.6130e+01	-7.3966e+00	9.1659e-01
2.6004e+01	-7.3582e+00	9.1159e-01
2.5869e+01	-7.3172e+00	9.0632e-01
2.5724e+01	-7.2736e+00	9.0074e-01
2.5568e+01	-7.2271e+00	8.9485e-01
2.5401e+01	-7.1776e+00	8.8864e-01
2.5222e+01	-7.1249e+00	8.8208e-01
2.5030e+01	-7.0687e+00	8.7517e-01
2.4823e+01	-7.0089e+00	8.6788e-01
2.4601e+01	-6.9452e+00	8.6020e-01
2.4362e+01	-6.8773e+00	8.5211e-01
2.4106e+01	-6.8050e+00	8.4360e-01
2.3831e+01	-6.7280e+00	8.3465e-01
2.3535e+01	-6.6461e+00	8.2524e-01
2.3217e+01	-6.5588e+00	8.1535e-01
2.2876e+01	-6.4660e+00	8.0498e-01
2.2510e+01	-6.3672e+00	7.9410e-01
2.2117e+01	-6.2622e+00	7.8270e-01
2.1695e+01	-6.1505e+00	7.7076e-01
2.1244e+01	-6.0319e+00	7.5827e-01
2.0760e+01	-5.9058e+00	7.4522e-01
2.0242e+01	-5.7721e+00	7.3160e-01
1.9688e+01	-5.6303e+00	7.1740e-01
1.9098e+01	-5.4800e+00	7.0261e-01
1.8469e+01	-5.3210e+00	6.8722e-01
1.7800e+01	-5.1528e+00	6.7124e-01
1.7090e+01	-4.9752e+00	6.5467e-01
1.6339e+01	-4.7879e+00	6.3750e-01
1.5547e+01	-4.5908e+00	6.1975e-01
1.4714e+01	-4.3837e+00	6.0142e-01
1.3842e+01	-4.1666e+00	5.8255e-01
1.2935e+01	-3.9394e+00	5.6313e-01
1.1994e+01	-3.7024e+00	5.4321e-01
1.1027e+01	-3.4560e+00	5.2281e-01
1.0038e+01	-3.2006e+00	5.0196e-01
9.0375e+00	-2.9368e+00	4.8072e-01
8.0350e+00	-2.6657e+00	4.5914e-01
7.0430e+00	-2.3884e+00	4.3726e-01
6.0759e+00	-2.1062e+00	4.1514e-01

5.1502e+00	-1.8210e+00	3.9287e-01
4.2837e+00	-1.5348e+00	3.7051e-01
3.4956e+00	-1.2499e+00	3.4814e-01
2.8057e+00	-9.6900e-01	3.2584e-01
2.2331e+00	-6.9515e-01	3.0371e-01
1.7951e+00	-4.3157e-01	2.8183e-01
1.5053e+00	-1.8178e-01	2.6031e-01
1.3723e+00	5.0584e-02	2.3923e-01
1.3968e+00	2.6183e-01	2.1870e-01
1.5701e+00	4.4834e-01	1.9881e-01
1.8723e+00	6.0676e-01	1.7966e-01
2.2710e+00	7.3413e-01	1.6131e-01
2.7220e+00	8.2815e-01	1.4387e-01
3.1709e+00	8.8738e-01	1.2740e-01
3.5568e+00	9.1143e-01	1.1195e-01
3.8193e+00	9.0118e-01	9.7581e-02
3.9065e+00	8.5890e-01	8.4328e-02
3.7844e+00	7.8831e-01	7.2210e-02