AFIT/GAM/ENC/98M-01

NEURAL NETWORK MODELING OF THE HEAD-RELATED TRANSFER FUNCTION

THESIS
Damion Reinhardt
Second Lieutenant, USAF

AFIT/GAM/ENC/98M-01

Approved for public release; distribution unlimited

19980409037

AFIT/GAM/ENC/98M-01

NEURAL NETWORK MODELING OF THE HEAD-RELATED TRANSFER FUNCTION

THESIS

Presented to the Faculty of the Graduate School of Engineering
of the Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Mathematical Statistics

Damion Reinhardt, B.S. Physics and Mathematics
Second Lieutenant, USAF

March, 1998

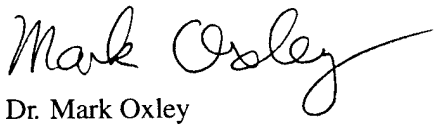
Approved for public release; distribution unlimited

AFIT/GAM/ENC/98M-01

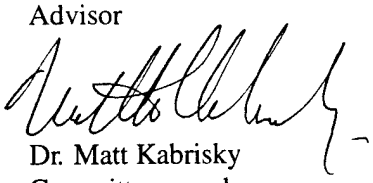
NEURAL NETWORK MODELING OF THE HEAD-RELATED TRANSFER FUNCTION

Damion Reinhardt
Second Lieutenant, USAF

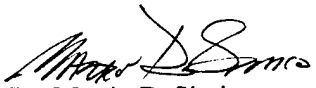
Approved:



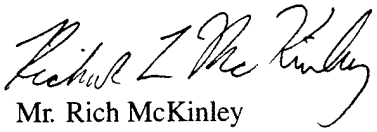
Dr. Mark Oxley
Advisor



Dr. Matt Kabrisky
Committee member



Dr. Martin DeSimio
Committee member



Mr. Rich McKinley
Committee member

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U. S. Government.

Acknowledgements

I extend my heartfelt gratitude to the folks in the AFIT math department, especially Dr. Alan Lair, Janet Daniel, and Capt Sam Gardner, who were especially kind and helpful to me, thereby making my transition from undergrad to graduate student as painless as possible. Special thanks to my advisor, Dr. Mark Oxley, for his support and expertise.

Thanks to Dr. Marty DeSimio and Mr. Rich McKinley of my thesis committee for their time and comments, and for the unique roles they played in getting me into this research. Also, thanks to Bob Bolia and Dennis Allen, who were valuable sources of information on the nature of the data with which this thesis was accomplished.

At the risk of some frivolity, I extend my gratitude to the GNU project for Emacs and a whole slew of other GPL programs which I used in the creation of this work. At the risk of extreme frivolity, I would like to thank Pepsi Corporation for both Mountain Dew and Taco Bell (without which I could not have survived the long nights in the lab) and for their keeping the free market alive by preventing McDonald's and Coca-Cola from overrunning the world.

Most importantly (and least frivolously) I would like to thank my lovely wife Laura for her incredible patience and steadfast love throughout this thesis effort. She stood by my side most literally and metaphorically during all circumstances, to the point of sleeping on the floor of the SIPL during my extended nights there.

Damion Reinhardt

Table of Contents

List of Figures

List of Tables

Abstract

Air Force interest in directional audio research is stimulated by the numerous potential cockpit applications of virtual audio, such as target identification, navigation, and communication. To this end, Air Force Research Laboratory maintains the Auditory Localization Sphere, a geodesic sphere with 272 vertex-mounted loudspeakers, with which they measure the direction dependent spectral changes induced upon sounds by the human head and pinnae, modeling these as a filter function known as the Head-Related Transfer Function (HRTF).

The HRTF is a remarkably complex function of azimuth, elevation, and frequency. This thesis examines approximation of the HRTF using Artificial Neural Networks (ANNs) and in doing so attempts to overcome the dual problems of compact data representation and interpolation. The data are divided into subsets upon which ANNs are trained, and the resultant networks are blended together using a number of techniques to produce a continuous model of the HRTF.

Two tessellation schemes are employed, the first based upon spatially clustered "caps" and the other composed of data points clustered about small circles parallel to the sagittal plane. Two neural network architectures are implemented: a multilayer perceptron and a radial basis function network.

The results of the research suggest that ANNs can successfully model and interpolate the entire human HRTF. The two tessellation methods appear to perform nearly equally well, as do all of the blending methods. By contrast, the radial basis function networks appear to have outperformed the perceptrons, yielding average error values of 5% to 10% deviation between modeled and sampled data.

NEURAL NETWORK MODELING OF THE HEAD-RELATED TRANSFER FUNCTION

I. Introduction

1.1 Background

The rarity with which we take note of our natural ability to perceive the direction of a sound source belies the incredible usefulness of such a faculty, especially in situations in which audio cues stimulate head movement for gathering visual data (6) (29). For example, when someone speaks at a conference table, directional audio cues tell us where to direct our attention; likewise, when traffic noises warn us of impending peril, our ability to spatially localize sound allows us to look immediately in the direction of danger. In fact, it is theorized that the evolutionary development of binaural and spatial hearing was originally driven by the advantage given to those creatures who could better utilize directional sound cues to more quickly respond to changes in the environment (41).

Since survival of the fittest is a grim reality of aerial combat operations, the Air Force is seeking to provide their pilots and crews with the advantages of spatial hearing. The implementation of effective synthetic directional audio cueing in the cockpit would greatly simplify a number of routine and combat flight tasks, such as target acquisition and identification, threat avoidance, navigation, and inter/intracrew communication (25) (36). For example, an aircraft operator can locate and identify a target faster when presented with a sound signal which appears to be coming from the direction of the target (25). Similarly, the task of navigation could be aided by the presentation of a sound signal which appears to emanate from the desired direction of flight. Finally, crew members would likely have less difficulty

distinguishing various communications if those verbal signals were presented with directional cues (25).

To further Air Force research in this area, the Air Force Research Laboratory (AFRL) maintains an anechoic chamber for audio research, in which the Auditory Localization Sphere (ALS), a large geodesic spherical speaker array, is utilized for the measurement of the direction dependent spectral and phasic changes induced upon sound by the human head and pinnae. The spectral changes are modeled as a filter function known as the Head-Related Transfer Function (HRTF).

The HRTF is a continuous function of sound frequency and spatial position, the aforementioned empirical sampling of which yields large quantities of discretized data; various functional models and interpolants have been proposed and implemented to overcome the resultant problems of data storage and interpolation. This thesis presents new methods based upon artificial neural networks for the construction of functional models which return HRTF dB gain for any arbitrary frequency and position. The network models are tested against sampled HRTF data, and the variability and errors in the network models are analyzed.

1.2 Definitions

This section supplies definitions for the key terms which will be utilized in this thesis. While the most widely agreed upon usages are presented, common variances and synonyms are noted. The definitions listed are paraphrases of the common usages found throughout the literature (6) (15) (25) (26) (45).

Pinna(e) denotes the cartilaginous portions of the outer ear(s), also referred to as the *auricle(s)* In this thesis, only human pinnae are used.

Monaural means involving only one ear, having a single sound signal

Binaural denotes a process involving both ears.

(Mid)sagittal plane is the planar surface which longitudinally divides the body into equal left and right halves. Also referred to as the *median plane.*

Interaural axis describes a line intersecting both of the eardrums.

Auditory localization is the ability to acoustically determine the location of a sound source; also known as *spatialization.*

Auditory localization cues are the aspects of a sound signal which allow for localization.

Interaural intensity difference (IID) refers to the difference in amplitude of sounds from one ear to another, caused primarily by the acoustical effects of the pinnae. Also referred to as the *interaural level difference* (ILD).

Interaural time delay (ITD) is a broadband estimate of the length of time transpired between the arrival of a sound at one eardrum and its arrival at the other. Also called the *interaural time difference* and *interaural phase difference* (IPD).

Duplex theory of hearing was expounded by Lord Raleigh in 1907, and focused on the two binaural auditory localization cues of ITD and broadband ILD as the primary means of human auditory localization.

Cone of confusion denotes a conic surface extending outward from the side of the head, axially coincident with the interaural axis, upon which sounds produce the same value for the ITD. In the case of sounds equidistant from both ears, the conic shape geometrically degenerates into the sagittal plane. For sounds directly to the left or right of a listener, the cone degenerates into the interaural axis, allowing for exact spatialization from the ITD alone. In actuality, the locus of points of a constant difference in distance from the eardrums is hyperboloidal; however, outside of the head it is well approximated by a cone.

Localization reversals occur as a result of the cone of confusion whenever a listener mistakes the spatial location of a source sound by perceiving it as reflected into an opposing hemisphere. The most common of these is the front to back reversal,

in which the listener perceives a sound with origin in the front hemisphere (in front of the head) as having originated in the rear hemisphere (behind the head). Back to front and, more rarely, up/down confusions have also been reported (36).

Virtual audio is sound which has been synthetically processed so as to create the illusion of spatial location, thereby allowing for sound localization. It is usually presented over earphones, although this is not necessarily always the case.

Head-related transfer function (HRTF) denotes the functional model of the spectral transform performed upon sound as a result of the acoustical effects of the head and pinnae. A function of spatial location and frequency, the HRTF specifies the gain/attenuation imparted upon a sound. The *directional transfer function* (DTF) is the HRTF at a fixed azimuth and elevation.

Head-related impulse response (HRIR) is the time-domain counterpart to the HRTF, consisting of the impulse responses which, when Fourier transformed, give rise to the HRTF.

Minimum audible angle is the smallest difference in a sound source's angular position detectable by the human ear. Also called the *minimum audible difference* (MAD).

Artificial Neural Network (ANN) describes a plurality of connected units which individually process numeric data and pass it along to other units. Typically, the network has some sort of training rule whereby it "learns" the proper weights of the connections given specific examples.

Nearest neighbor(s) denotes the closest speaker(s) on the ALS to a particular spatial position.

Tessellation is the process of partitioning a set (in this case, the 272-point ALS data sets) into non-overlapping (disjoint) subsets.

1.3 Problem

The human HRTF is a highly convoluted continuous function of spatial location and frequency, the measurement of which results in discrete sample points of an underlying continuous space. The complexity of the function necessitates sampling on a fine mesh grid, that is, taking samples at relatively closely spaced spatial locations and many different audible frequencies, in order to obtain a reasonably accurate model of the underlying transform. This sampling yields vast amounts of data which must be effectively interpolated in both location and frequency to determine the amplitude gain upon particular frequencies received from specific spatial locations. Consequently, the issues of compact data storage and interpolation are two of the most beguiling problems in current 3-D audio research (12) (20) (42).

1.4 Research Objectives

This thesis describes the implementation of a functional model of the entire human HRTF using artificial neural networks (ANNs), which are particularly well suited to the problem. Several research objectives were necessary for the completion of such a task:

- Implement several simple, computationally efficient HRTF interpolants.
- Tessellate the ALS data sets into disjoint subsets suitable for neural network training.
- Create, optimize, and train neural network approximations for the HRTF data on the tessellated subsets.
- Implement a physically sensical, mathematically sound method for blending the resultant values of the neural networks to determine the HRTF on regions between the tessellated subsets. It is desirable that the method employed result in a continuous HRTF over all azimuths and elevations.

- Test the final ANN models against the actual HRTF at interpolated data points, and analyze the results.

With these objectives accomplished, the level of validity and usefulness of ANN models of the HRTF should be well established.

1.5 Scope

This thesis develops ANN models for the approximation of the entire human HRTF. The HRTF and ITD data in this thesis were provided by the AFRL, sampled from the ALS (24).

1.6 Approach

1.6.1 Artificial Neural Networks. To overcome previously described difficulties associated with the HRTF, researchers have attempted to derive simplified functional models from the measured data (12). In typical functional HRTF models, the input parameters of spatial location and frequency result in an output of HRTF gain. Ideally, the output of the model matches the actual HRTF gain at the given spatial location and frequency.

The artificial neural networks (ANNs) implemented in this thesis effort are but a few of many possible functional models of the HRTF; they have never previously been used to model the entire human HRTF. The specific implementations of the ANNs will be delineated in Chapter 3.

1.6.2 Validity Checking. Validation of the model may be easily accomplished by comparing the model HRTFs against actual sampled data; however, while comparing the ANNs against the data upon which they were trained provides some validation of the model, such a method reveals nothing of the true interpolative ability of the ANN. This problem may be overcome by either leaving out one or more of the 272 speaker positions from the training set, or by taking some new data points in

between the regularly spaced points on the sphere. As the former approach results in overly large spacings, the latter approach is used in this effort.

1.7 Thesis Outline

Chapter I is a brief introduction to this thesis, providing the background of USAF 3-D audio research and the nature and importance of the HRTF, thus leading to the problem of interpolation and compact representation through functional modeling. It then describes the proposed solution to the problem, and delineates the approach taken to achieve said solution.

Chapter II consists of a literature review and detailed explanation of many topics mentioned in the introductory chapter. The history of spatial hearing research is reviewed, and the literature describing the mathematical structure of ANNs and their practical implementation is also reviewed. Finally, the specifics of the AFRL ALS and its measured data sets are explained, as well as the conventions used in referring to those sets.

Chapter III presents the methodologies employed in this thesis. A number of simple interpolants are described and implemented for use in the HRTF problem, established methods which may also be used as comparative gauges for the ANN models. Finally, the specific implementations of the ANNs are detailed for the various types of networks and metrics employed.

Chapter IV exposits the results of this research effort, evaluating both the simple interpolants and the ANNs using the two metrics. Finally, Chapter V includes a summary, conclusions, and recommendations for further research.

II. Background

2.1 Introduction

This chapter presents the background for this thesis effort, treating several topics of relevance. The development of our understanding of binaural and spatial hearing is briefly recounted, up to the current research into functional HRTF modeling. Following this treatment, the various attempts at functional modeling of the HRTF are described. Next, the theory of neural networks is overviewed, as are the structure and application of the particular architectures used herein. Finally, the specifics of the AFRL ALS are presented, along with details of the data sets used in this thesis, and their associated notational conventions.

2.2 Binaural and Spatial Hearing Overview

The following is an overview of the development of spatial hearing research, intended to motivate a proper understanding of the HRTF and its relevance to audio research.

2.2.1 Duplex Theory. While Fechner made the very first steps in localization research in the 1860's (16), Lord Raleigh is generally recognized as the founder of binaural and spatial hearing research. In 1907 he expounded his duplex theory of hearing (32), which emphasized the importance of two primary binaural localization cues: low frequency interaural time delays (ITDs) and high frequency interaural level differences (ILDs). From that time until recently, most research in binaural hearing has been focused on refining and expounding duplex theory. The two primary binaural cues have been thoroughly investigated, revealing a great deal about their nature, usage, and salience.

Raleigh's original conception of ITD as a broadband estimation of the phase shift across all frequencies has been confirmed as true in an approximate sense (6)

(45). On the other hand, his formulation of the ILD as an overall volume difference has been modified to include the complex direction and frequency dependent gains (monaural cues) produced by the pinnae. Accordingly, the cueing information of the ILD is now thought to reside in the multiplicity of ILDs for the various frequency bands present in a source sound, rather than in an overall level difference (45). The pinna-induced monaural spectral transformations which give rise to these ILDs are contained within the well-known head-related transfer function (HRTF); their importance as a primary localization cue has prompted extensive research into the determination and functional modeling of the HRTF.

It has been experimentally confirmed that the relative salience of the ITD cue generally decreases with increasing frequency (36) (44). By contrast, the relative salience of both the ILD and monaural spectral cues tends to *increase* with frequency (44) (45). Also, it is generally agreed that the ITD is the more dominant localization cue (21) (43) (44), and that it provides the principal azimuthal cue. For a given ITD, there exists a roughly conic (actually hyperboloidal) locus of points from which the source sound may have originated, this is the so called "cone of confusion." It is widely accepted that the ITD's obfuscation in this region is resolved by the ILD cue (26) (36) (45); however, the true psychophysical relations between these primary interaural cues and other, less significant monaural cues is only beginning to be understood (45).

2.2.2 Modern Additions to Duplex Theory. While the duplex theory of localization has held sway for some time, recent developments indicate that additional significant localization cues exist. For example, monaural spectral cues may allow for a listener to determine the location of a familiar or broadband smooth sound with the use of only one ear (36) (45). Also, it has recently been shown that the ITD cue may be useful in the localization of high frequency sounds having certain waveform shapes (19). Additionally, many authors have commented on the significance of head movement to the resolution of ambiguities in other cues (39) (40) (46). In current

theoretical development the two cues originally presented in the duplex theory remain principal, however, the other cues are considered perceptually significant. It is noteworthy that many of these new cues are HRTF dependent, as is the ILD.

2.2.3 Conclusions. While a unified theory of sound localization is still inchoative, it is clear that the spectral transforms of the HRTF are cardinal to audio spatialization. The effective modeling of the HRTF will not only allow for the production of synthetic virtual audio, but will also aid our understanding of the true nature of spatial hearing.

2.3 HRTF Modeling and Synthesis

Functional modeling of the free-field-to-eardrum transform deepens understanding of the true nature of the transform while facilitating the artificial synthesis of spatial audio. The first steps in modeling some of the effects of the outer ear were made in the late 1960's by Batteau, who modeled the pinna as a two delay and sum acoustic coupler, with one delay based on source elevation and another corresponding to source azimuth (2) (3). Later, Shaw determined the frequency response of a simplified pinna via direct measurement (34). While both of these physically motivated models increased the understanding of the nature of the outer ear transform, neither was sufficient for approximation of the actual HRTF.

After these initial attempts at modeling the physical mechanisms and effects of the pinnae, researchers began to investigate the HRTF, attempting to directly represent the transform characteristics of the outer ear by means of an analytic expression (11). The first such attempt was reported in 1986 by Genuit, who derived a relationship between filter parameters and pinna morphology using methods of classical acoustics, in order to construct a 16 time-delayed channel filter bank model of the HRTFs (18). His approach eliminated the necessity for empirical recording

of the HRTF for any given pinna shape, however, it has not been demonstrated to generate a reasonably close approximation to the actual HRTF (12).

Later, as advances in computing technology lifted computational barriers which had previously prevented artificial synthesis of virtual audio, research focused more upon the empirical measurement of the HRIR/HRTF and the derivation of functional models based directly upon the resultant data. In 1992, Chen proposed a functional model for the HRTF based on principles of beamforming (10) (11). The beamforming model used a spatially arrayed weighted sum of input data, and is similar to ANN modeling in its use of a deterministic weighting function.

Two years later, Millhouse created an ANN approximation for the HRTF for the horizon circle, that is, all azimuths at zero elevation. Millhouse's approximation met with limited success, however, computational limitations prevented the development of an ANN model for the full HRTF for all spatial locations.

Finally, in 1995 Chen *et al.* developed yet another functional model for the HRTF, known as the spatial feature extraction and regularization (SFER) model, based upon a weighed combination of eigentransfer functions generated from the Karhunen-Loeve expansion. The most successful model to date, the SFER reports typical errors of around one percent deviation between measured and modeled HRTFs for most spatial locations (12).

2.4 Artificial Neural Networks

The concept of an artificial neural network is a very broad one, with several debatable and imprecise definitions clinging to it. In the wide sense, it is simply a mapping of a transformation in terms of interconnected neuronal unit transforms. The reader who is unfamiliar with such constructs may find comprehensive yet comprehensible introductions to the topic in both Bishop's and Rogers' works (5) (33).

Both of the networks considered herein are multi-layer, feed-forward networks, which, as a consequence of the latter property, may be represented as explicit, analytical functions of their inputs and weights.

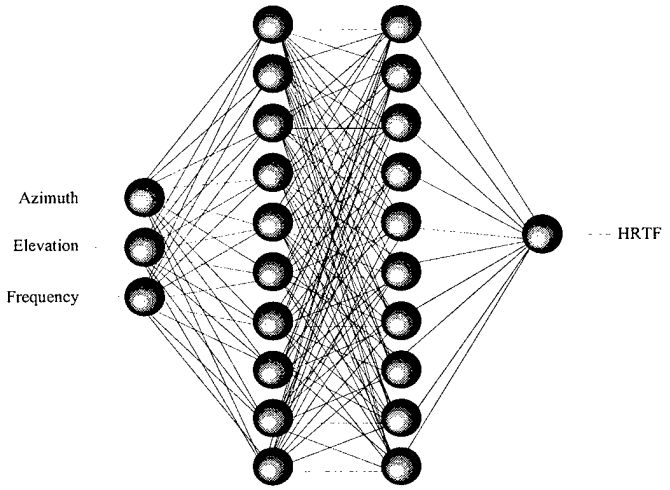


Figure 1. Multilayer Perceptron (MLP) Network Architecture, 3:10:10:1

2.4.1 Multilayer Perceptron. The first of the neural network architectures considered is the multilayer perceptron (MLP), an example of which is displayed in Figure 1. The network in the figure has three inputs, two hidden layers of ten nodes each, and a single output; it is hence referred to as an MLP 3:10:10:1 in the conventional network shorthand notation.

An MLP with sigmoidal activation functions and linear output functions consisting of only one hidden layer and one output layer, while simple in design, has nonetheless been mathematically proven capable of approximating any arbitrary continuous functional mapping, given enough hidden nodes (13). Of course, a perceptron consisting of 3-layers (two hidden and one output, like that in Figure 1) is also capable of such a feat (22). This latter design is one of the two employed herein for HRTF approximation, the specifics of which are detailed in the next chapter.

2.4.2 Radial Basis Function Network. The other network architecture of interest is the radial basis function network, depicted in Figure 2 with three inputs, 10

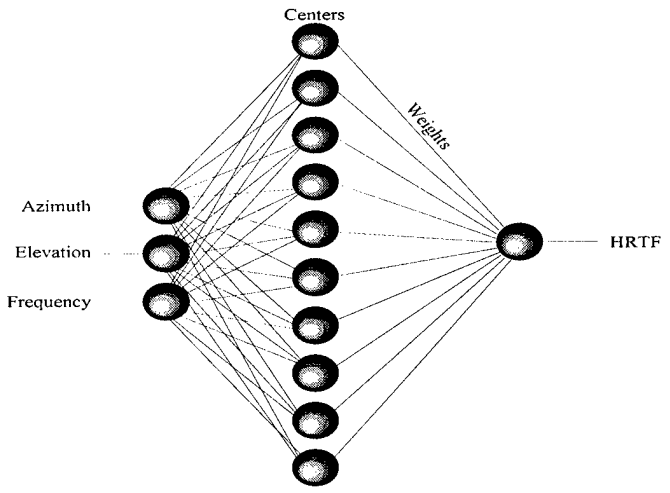


Figure 2. Radial Basis Function Network Architecture, 3:10:1

hidden nodes, and a single output. Rooted in the methods of exact interpolation (7) (28) (31), the RBF network consists of a number (typically Gaussian) basis functions which act as hidden nodes and are transformed by linear activation functions in the output layer. The centers and widths of the basis functions are determined during training so as to minimize the given error function, typically the L2 norm (5) (14). Again, the specifics of the networks utilized herein are deferred to the upcoming chapter on methodology.

2.5 The AFRL ALS and Associated Data Sets

The empirical measurement of the HRTF and other 3-D audio localization cues have been pursued of late by various persons and organizations. Some of the earliest attempts were made by Plenge, who recorded binaural sounds using two microphones mounted upon a manikin, and by Butler and Belendiuk, who did the same with human subjects (8) (30). Later, organizations such as the MIT Media Lab began collecting relatively dense data sets encompassing all azimuths and elevations.

Air Force interest in 3-D audio simulation for cockpit applications has prompted military research in this field; the Biocommunications division of AFRL's Human Effectiveness Directorate maintains an anechoic chamber at Wright-Patterson Air

Force Base for acoustical measurement. This chamber originally contained a system for measuring HRTF data consisting of a simple horizontal ring with mounted speakers, which has since been superseded by the Auditory Localization Sphere (ALS), a geodesic sphere with 272 roughly uniformly spaced, vertex mounted loudspeakers fully enclosing a platform for acoustical measurement. The ALS schematic is shown in Figure 3.

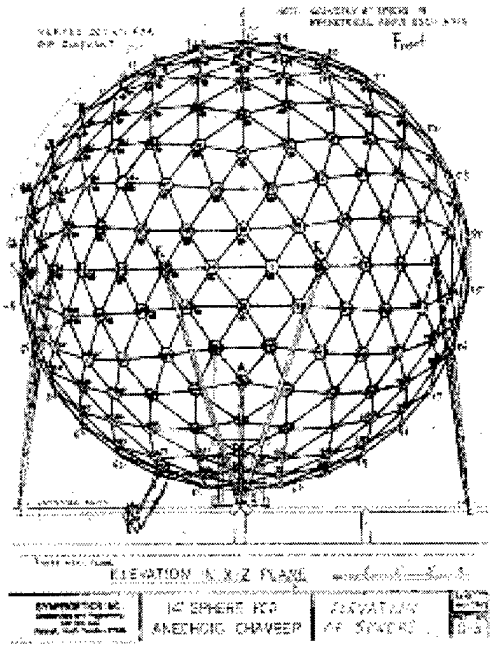


Figure 3. The AFRL's Auditory Localization Sphere.

The AFRL's conventions for reference to the spatial positions of the speakers on the ALS are quite naturally based upon a spherical coordinate system of azimuth, elevation, and range, with an origin at the midpoint between the subject's ears. Azimuth refers to the amount of angular separation from the front half sagittal plane to the vertical plane containing the sound source and the origin, measured in the counterclockwise direction as seen from above, as shown in Figure 4.

Elevation describes the angular separation from the sound source to the horizon, as shown in Figure 5. For the purposes of this thesis, in accordance with the

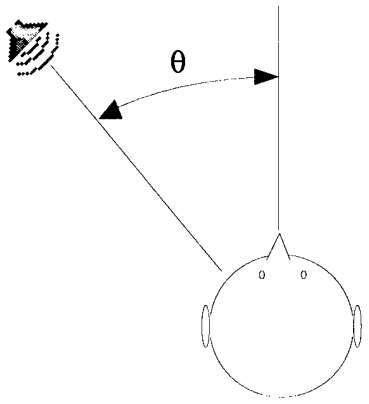


Figure 4. Azimuth, θ of sound source to directly in front of face

conventions used in the AFRL data sets, both azimuth and elevation are measured in degrees.

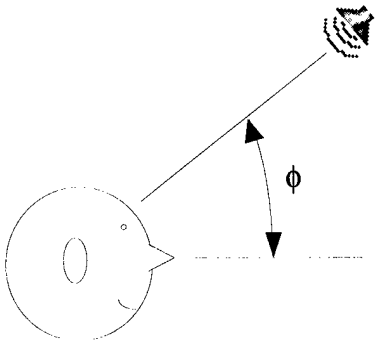


Figure 5. Elevation, ϕ measured from the ground plane to a sound source

Using an Knowles Electronics Manikin for Acoustical Research (KEMAR) or a similar device, AFRL researchers record microphone responses in the free field (no manikin present) and at the manikin eardrum positions for 104 test frequencies and 272 speaker locations. From these two data sets, one may determine the spectral gains induced upon the tested frequencies, thereby deriving the HRTF.

The data in this thesis were recorded from the right hand side of the head, using pinna model DB-066 mounted on a DB-4004 model KEMAR mannikin. Examples

of the data files used in this research are found in Appendix A, and a list of the ALS speaker positions and designation numbers is found in Appendix B.

2.6 Conclusion

While the ITD is clearly the most dominant spatialization cue, it requires a bare minimum of modeling and analysis compared to that necessitated by the HRTF, which induces most of the other perceptually significant localization cues. Functional modeling of the HRTF simultaneously alleviates the dual problems of data storage and interpolation of the HRTF, while also providing some insight into its underlying physical nature.

This thesis research investigates the interpolation and modeling of the HRTF over all azimuths and elevations using artificial neural networks. It is, in some sense, a continuation and refinement of initial HRTF modeling efforts on the zero elevation horizon circle performed by Millhouse (27).

III. Methodology

3.1 Introduction

The previous chapter overviewed earlier attempts at modeling and interpolation of the HRTF. Examined herein are techniques and analyses employed for interpolation and modeling within this thesis effort. Several simple interpolants are described, preliminary data analyses are detailed, and implementations of ANN models are laid forth.

3.2 Simple Interpolants

A number of relatively straightforward spatial interpolants were implemented for use with the ALS HRTF data sets, both as comparisons for testing the relative efficiency of the neural network models, and as techniques for recombining ANN models trained on subsets of the 272 spatial locations. These interpolants are detailed below.

3.2.1 Nearest Point Rounding. Several 3-D audio research efforts avoid the complexities of the interpolation problem altogether by simply utilizing the closest measured data point to the desired location (17) (37). This method, while seemingly an oversimplification, has some validity in that the sheer complexity of the HRTF tends to invalidate interpolative blending methods; that is, a particular DTF cannot necessarily be accurately constructed from a combination of surrounding DTFs. Also, in a data set with a maximum pointwise separation close to or less than the minimum audible difference, simply crossfading filter coefficients from one measured data location to the next without interpolation produces little or no audible effect. However, as the human minimum audible angle may be as low as 1 to 2 degrees, and as the AFRL ALF data sets have an average pointwise angular separation of approximately 13.3 degrees, virtual audio synthesis from a single 272 point ALS data

set may require a more sophisticated approach than simply utilizing the nearest data point. Nonetheless, this method is useful for its comparative value.

3.2.2 Weighted Averages. A moment's reflection reveals that the above mentioned method of rounding is algebraically equivalent to a weighted average (albeit a rather crude one) in which surrounding HRTF data points are assigned weights of either 1 or 0. More sophisticated weighted averages have found use in various virtual audio applications requiring HRTF interpolation. Such averaging schemes have the advantages of straightforward, rapid computation, consequently, they have gained a foothold in some real time convolution applications, such as Rick Bidlack's Virtual Sonic Space (4).

Two weighted averages are implemented in this thesis: the first based upon an arbitrary number of nearest ALS speakers to the spatial position of a desired virtual sound source location; the other based upon only three nearest neighbors forming a roughly equilateral triangle enclosing such a location.

3.2.2.1 Simple Weighted Average of Nearest Neighbors. As the HRTF as an underlying physical phenomenon varies with spatial position, it is logical to interpolate the HRTF at a fixed point using the collected data nearest to that position, and to weight the data points closer to the interpolated point more heavily than more distant data. Perhaps the most straightforward technique based on this line of reasoning is to choose an arbitrary number n of measured nearest neighbor data points $\{nn_i | i = 1...n\}$ and assign the data weights $\{scale_i | i = 1...n\}$ inversely proportional to their respective distances $d(X, nn_i)$ from the interpolation position X.

$$scale_i(x) = \frac{1}{d(X, nn_i) \sum_{i=1}^{n} \frac{1}{d(X, nn_i)}} \tag{1}$$

The code implementing the above interpolant is found in `interp_nn.m` in Appendix D.

It is obvious that the inclusion of fewer nearest neighbors results in decreased computational overhead, which naturally raises the question of a minimum sufficient number of neighboring data points necessary for effective interpolation.

The answer to this question is found at the root of interpolation theory, in the formulation of the approximation problem itself. Given a function f in a metric space (A, ρ) can there be found a function F defined on the fixed space A which is close to the function f?

From this, it is clear that given two adjacent speaker points on the ALS, one may construct a function for the value of the HRTF in terms of position along the arc connecting those two points; however, this function will only be useful in interpolating the value of the HRTF for positions along that arc. Extending the space by including an additional speaker adjacent to both of the original speakers yields the construction of an interpolant function valid over a triangular region on the surface of the sphere.

It is clear from a visual inspection of the ALS in figure 3 that, for any arbitrary fixed azimuth and elevation, the location thus described must fall either within or on the border of a roughly equilateral triangle composed of three adjacent speaker positions. For descriptive purposes, the triangle encompassing a given spatial position is referred to herein as the nearest neighbor triangle (NNT) for that position, as the triangle is composed of the three closest neighboring speaker positions.

Thus, it is physically sensical that the three vertices of an NNT comprise a minimum number of necessary points for interpolation of any arbitrary location on the sphere. The above interpolation scheme has been shown to yield reasonable results using only three speaker positions in weighting, as in Figure 6.

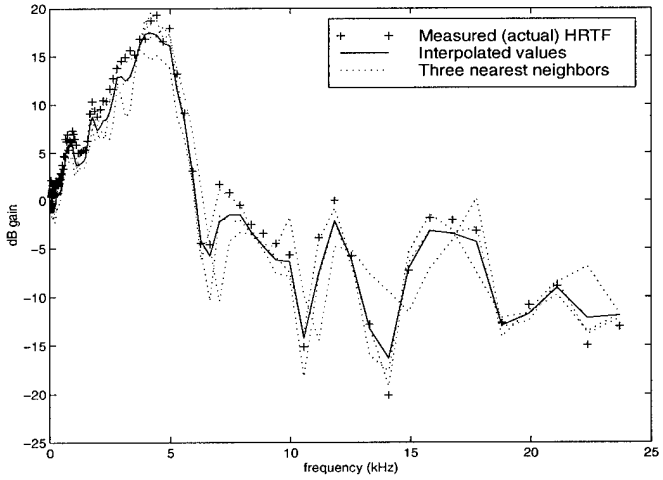


Figure 6. Nearest Neighbors Weighted Average Technique for Azimuth = 97.5, Elevation = -44.71.

While the above method yields apparently appealing results, it has several drawbacks. There is a problem with overflow at positions too close to the measured points, nearest point rounding must be employed for such locations. Also, the algorithm fails to result in a continuous surface over the entire ALS. Clearly, a more sophisticated method of weighted averaging is required.

3.2.2.2 Three Nearest Neighbors Triangle Weighted Average. The problems inherent in the previous method may be averted by formulating an interpolant based on a fixed number of nearby speaker positions; for the reasons described above it is logical to choose three.

A satisfactory weighted average within a particular NNT ought to have certain desirable properties: (1) an inversely proportional relationship between the weighting assigned to a measured location and the distance from that position , (2) convergence to measured values at the triangle vertices, and (3) a weighting of zero for a measured vertex location whenever the new spatial position to be interpolated lies directly upon the triangle edge opposite that vertex. The first two conditions are desirable for any interpolants, while the final condition is unique to the problem of interpolating within

triangular regions, and is necessary to insure a continuous interpolated surface over the entire sphere of the HRTF.

For a triangle with vertices a, b, and c, a desired interpolation position x, and a distance function $d(\cdot,\cdot)$, the proposed scaling coefficients

$$scale_a(x) = \frac{d(b,x) + d(c,x) - d(b,c)}{d(a,b) + d(a,c) - d(b,c)}\,, \tag{2}$$

$$scale_b(x) = \frac{d(a,x) + d(c,x) - d(a,c)}{d(a,b) + d(b,c) - d(a,c)}\,, \tag{3}$$

$$scale_c(x) = \frac{d(a,x) + d(b,x) - d(a,b)}{d(a,c) + d(b,c) - d(a,b)}\,, \tag{4}$$

fulfill all of the requirements outlined above. The inverse proportionality of distance and weighting is evident, and the latter properties are readily demonstrable. The case where the interpolant point x is coincident with vertex a illustrates the second requirement.

$$scale_a(a) = \frac{d(b,a) + d(c,a) - d(b,c)}{d(a,b) + d(a,c) - d(b,c)} = 1$$

$$scale_b(a) = \frac{d(a,a) + d(c,a) - d(a,c)}{d(a,b) + d(b,c) - d(a,c)} = 0$$

$$scale_c(a) = \frac{d(a,a) + d(b,a) - d(a,b)}{d(a,c) + d(b,c) - d(a,b)} = 0$$

Thus, at any vertex, this weighted average scheme will simply return the measured HRTF data from that point. Also, note that as the weighting functions are rational functions of polynomials, this scheme will result in a continuous surface on its tri-

angular region of definition which converges to measured HRTF values as the new interpolated position approaches NNT verticies.

The final property is also easily demonstrated. Suppose that the new interpolated position x lies on the secant line connecting points b and c. Then the sum of the distances between points b and x and points c and x must equal the distance $d(b,c)$, and thus

$$scale_a(x) = \frac{d(b,x) + d(c,x) - d(b,c)}{d(a,b) + d(a,c) - d(b,c)} = \frac{d(b,c) - d(b,c)}{d(a,b) + d(a,c) - d(b,c)} = 0$$

which is the desired result.

This method has also been shown to yield reasonable results, as shown in Figure 7.

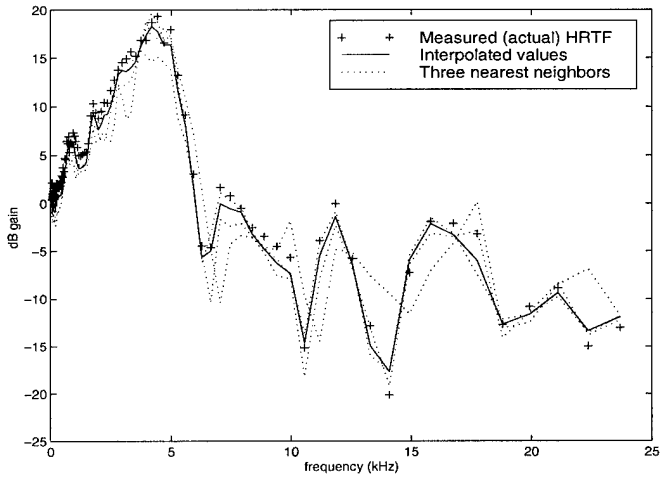


Figure 7. Three Nearest Neighbors Weighted Average Technique for Azimuth = 97.5, Elevation = -44.71.

As mentioned above, the true underlying HRTF is a continuous function of frequency and position, and thus it is desirable that spatial interpolant models should produce an HRTF that is continuous over all azimuths and elevations. It may be shown that the HRTF generated by the interpolant procedure just described pos-

sesses this desirable property. The weighted average employed is merely a linear combination of continuous rational functions over a given triangular region of interest, and thus the interpolated HRTF is clearly continuous on that region. Additionally, this method is formulated specifically to degenerate along triangle borders into a weighted average of only the two endpoints of that triangle side on which the interpolant position falls, the scaling coefficients of which are dependent only upon the interpolated position's distance from the two endpoints. This property ensures that any two adjacent triangles will converge to the same values at their common border, thereby ensuring continuity over the entire ALS.

The code for this interpolant as implemented for this research is found in `interp_tri.m` and `triweightavg.m` in Appendix D.

3.2.3 Piecewise linear. A very popular method for simple interpolation in the HRTF problem (20), the piecewise linear method boasts all of the desirable features of those previously described, with the possible drawback of additional computation involved in obtaining the solution of a line or a plane.

For the purposes of this thesis, the piecewise linear method was employed by simply solving for the plane described by the azimuthal, elevational, and HRTF gain values of the three nearest neighbors, and then using the solution of said plane to determine the HRTF values at the interpolated azimuth and elevation. The code for this interpolant is found in `interp_tri.m` and `tripiecelin.m` in Appendix D. The interpolant yields very similar results as the previous two methods, as shown in Figure 8.

As in the previous method, the HRTF produced by the piecewise linear spline is continuous on each individual triangular region, and the HRTF values of adjoining triangles again converge to common values along triangle boundaries. Thus, this method also yields an HRTF that is continuous on the entire sphere.

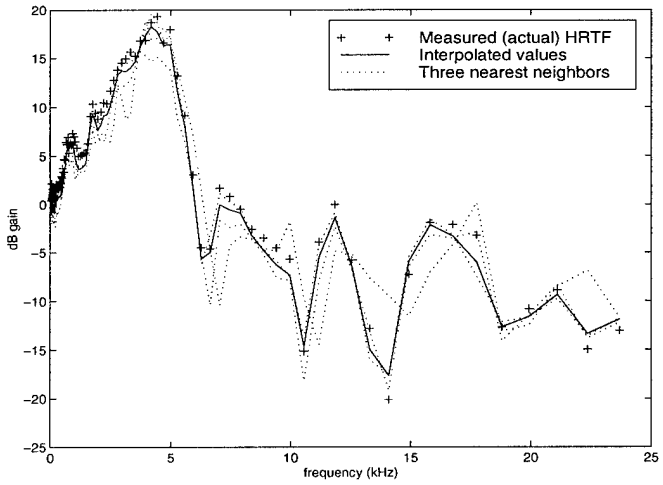


Figure 8. Piecewise Linear Technique for Azimuth = 97.5, Elevation = -44.71.

3.2.4 Comparison of the Various Simple Interpolants. It is clear from the similarities between Figures 6, 7, and 8 that all of the above methods yield very similar results. This is especially the case for points well interior to a fixed NNT, as the first method tends to diverge at points close to the triangle vertices, as mentioned earlier. Also, all of these simple methods appear to yield results fairly close to the actual underlying HRTF, as depicted in Figure 9.

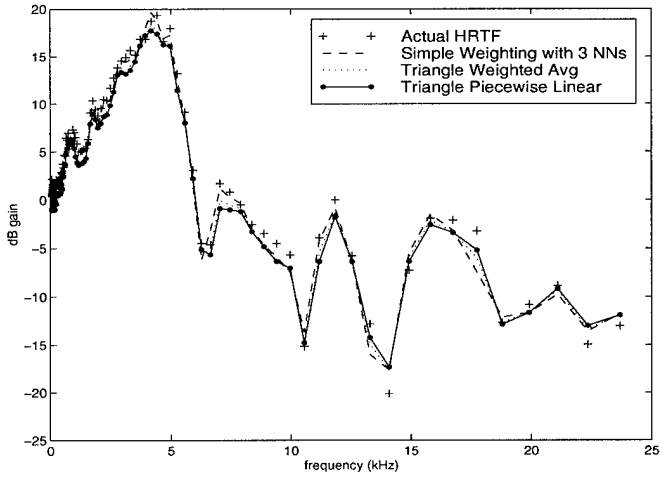


Figure 9. Comparison of Simple Interpolants at Azimuth = 97.5, Elevation = -44.71.

Thus, these simple interpolants should be good for measuring the performace of the ANN models, as well as combining the results of neighboring ANN networks, which will be described below.

3.3 Implementation of ANN models

The realization of the neural networks in this research may be considered a three step process. The first problem was to intelligently partition the ALS into small enough subsets to allow useful network approximation, the next was to construct and train neural network architectures upon those subsets, and the final step was to recombine the results of the networks. These processes are delineated in detail in the following sections.

3.3.1 Tessellation. In theory, the implementation of a neural network approximation for the human HRTF is a straightforward procedure. Simply choose a robust network architecture and then train the network on all 272 data points until the desired accuracy is achieved. No doubt this approach will become computationally feasible at some point in the future; however, at this point it is impractical even for advanced hardware. For example, a Sun Ultra 1 platform running a Matlab Neural Network Toolbox implementation of a 20 node radial basis function network on the data from merely 40 speaker locations takes the better part of a week to train, and often results in out-of-memory errors as the process pushes over 700-800 Megabytes of RAM. MLPs fare even worse, generally taking far longer to train, and, in the case of the Levenberg-Marquardt approximation, possesing even more of a tendency towards memory inflation.

The computational difficulties involved in training an ANN on the entire HRTF may be greatly alleviated by simply dividing the data set into several subsets which are more readily trainable. The question of how to best divide the speakers into the subsets is herein referred to as the HRTF tessellation problem.

Desirable solutions to the tessellation problem should minimize the HRTF variance over the resultant subsets, as greater variance in a training set generally results in longer training time and less faithful approximation of data (5). A few possible schemes seem promising. The most readily obvious approach is to group the speakers together into "caps" of clustered nearest neighbors (the term is drawn from an analogy to the Earth's polar ice caps). Minimizing the average spatial separation of the subsets should help to minimize HRTF variance, as the HRTF varies with spatial position.

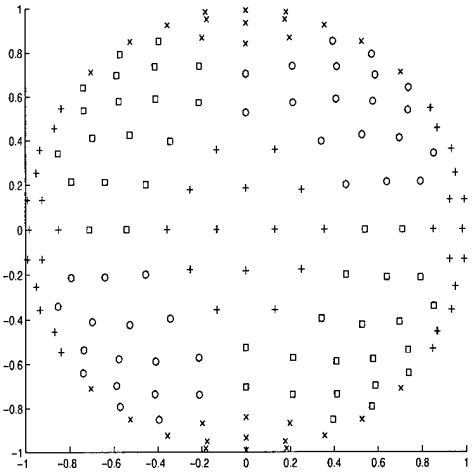


Figure 10. Tessellation of ALS Based on Nearest Neighbor Clusters.

There are various ways of choosing the clusters, perhaps the most straightforward of which is to choose a number of well-distanced cap centers and assign each of the 272 speakers to the closest center. An example of such a tessellation scheme is depicted in Figure 10. Note that there are no lines displayed between the cap regions; while visually helpful and appealing, it would be mathematically invalid to display such lines as we have not yet provided a method by which to define the borders between cap regions. Such methods will be described in later sections on recombination.

In the tessellation scheme shown in Figure 10, the cap centers are chosen as the points of unit distance from the sphere center along the Cartesian axes, and the

points in each of the octants that are of unit distance from the sphere center while equidistant from the Cartesian axes. That is, the six face-centered points of a unit cube and the eight vertices of a cube of diameter $2\sqrt{3}$.

There are, to be certain, many other possibilities for tessellation based upon an arrangement of well spaced, roughly equidistant cap centers. For example, one could use the vertices of any of several uniform polyhedra, such as an icosahedron, a dodecahedron, a snub cube, or a small rhombihexahedron. Alternatively, one could use a generalized numerical algorithm for spacing an arbitrary number of cap centers around the surface of the sphere. Treating the cap centers as repulsive bodies of equal negative charge, as in valence shell electron pair repulsion (VSEPR) modeling, yields desirable results. Such an algorithm has been constructed for use within this thesis, the code for this method as well as the program used to tessellate the sphere into caps are included in Appendix D under the names `repulsion.m` and `sph2caps.m`, respectively.

A rather different approach to the tessellation would be to group the speakers into "rings" of roughly equal ITD. Such an approach may be valid in that the IID cue has been noted to vary primarily in scale rather than shape (15) in such regions. One such approach is depicted in Figure 11.

Choosing the width and number of such rings is at least as difficult a task as that of determining the caps, in that it, too, presents a number of possibilities. In this thesis, the regions were built upon a number of evenly spaced small circles of equal ITD. Each of the 272 speakers is assigned to the closest of a generated set of ITD circles, just as the speakers were clustered to the closest cap centers. The number of circles chosen is variable, and is optimized based upon the evenness of resultant cluster sizes. Again, the code for this algorithm is provided in Appendix D, under `sph2circs.m`.

As mentioned above, the key to a successful tessellation is the minimization of HRTF variance in each of the subsets. For the purposes of comparison, we utilize a

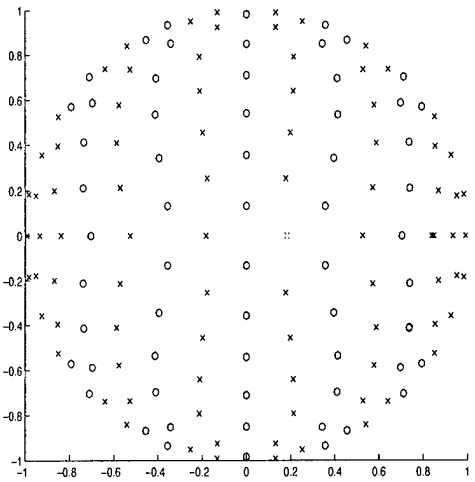


Figure 11. Tessellation Scheme Based on Regions of Similar ITD.

formulation of summed variance over a region. For each tested frequency, there is a single HRTF dB gain value at each unique spatial location. The variance of this set is computed for each of the 104 test frequencies, and the resultant 104 variances are summed into a comparative test statistic, referred to here as the frequency summed variance.

For a number of different tessellation schemes, the frequency summed HRTF variance has been computed for comparison, the results are shown in Table 1.

Table 1. Comparison of Variance for Tessellation Schemes (dB^2)

Number of Clusters in Tessellation	Avg. Variance of HRTF in caps layout	Avg. Variance of HRTF in ITD circles
9	923.8695	1063.3490
11	810.7745	818.2701
13	819.2882	772.6308

Thus, it seems one may reasonably expect the caps-based networks to perform about as well as those based on ITD circles – an expectation confirmed in testing and reported in the next chapter. Other statistics of interest include the variance of the entire 272-point ALS, that of the 24 speaker points on the zero-elevation horizon circle, and that of the 24 speakers coincident with the sagittal plane, the values

of which are 2075, 2413, and 1282 dB^2, respectively. These results imply that the horizon circle is an especially difficult region on which to train ANNs, while the sagittal plane and the other aforementioned tessellated regions are more conducive to such training.

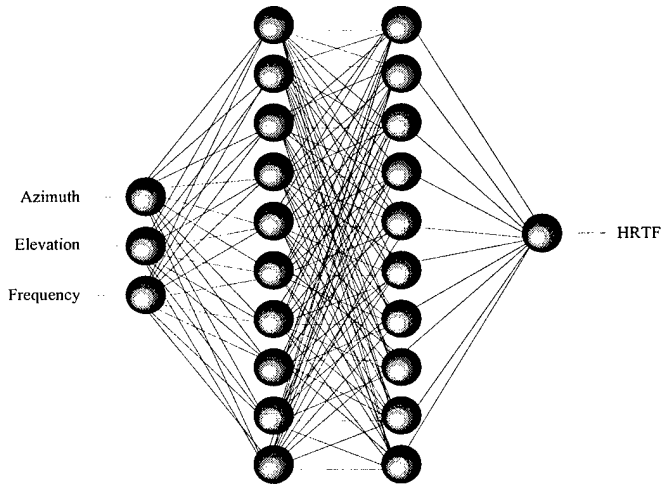


Figure 12. Multilayer Perceptron (MLP) Network Architecture for HRTF Approximation

3.3.2 Multilayer Perceptron. The MLPs utilized for HRTF modeling were constructed and trained with three inputs: azimuth, elevation, and frequency; and a single output of HRTF dB gain, as shown in Figure 12. However, the networks used were not the 3:10:10:1 shown here for illustrative simplification, but rather 3:50:50:1; the large number of hidden nodes necessitated by the complex shape of the HRTF. The activation functions of the hidden nodes were sigmoidal, based on the hyperbolic tangent; the output nodes were transformed with linear activation functions.

Preliminary backpropagation training of the several MLPs was accomplished using both ordinary gradient descent and the Levenberg Marquardt (LM) approximation (14); they produced similar results. However, for the actual MLP training ordinary gradient descent with momentum was chosen over the ordinarily faster LM algorithm due to implementation problems explained in the following chapter. The

models thus implemented were based upon modified code from the Neural Network Toolbox, the full listing of which may be found in Appendix D.

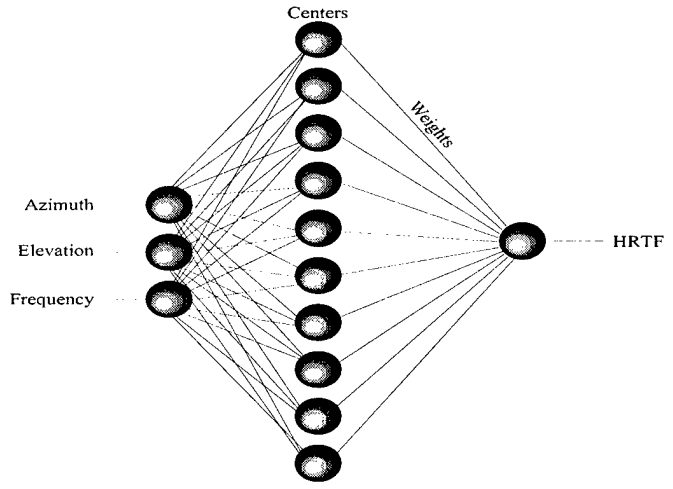


Figure 13. Radial Basis Function Network Architecture for HRTF Approximation

3.3.3 Radial Basis Function Network. Like the MLPs, the RBF networks were designed to input azimuth, elevation, and frequency, and output HRTF dB gain. Again, the actual networks utilized were larger than the one provided here for illustration; 3:20:1 rather than the 3:10:1 network setup shown in Figure 13. Just as with the MLP, the size of the hidden layer was empirically, subjectively determined so as to provide reasonable approximation with a minimum of hidden nodes.

The training of the RBF networks followed the orthogonal least squares learning algorithm first presented by Chen et al. (9), which is now a commonly used radial basis function update rule. Again, the network was based largely upon code from Matlab's Neural Network Toolbox, especially `solverb.m`, and `simurb.m`. The modified versions of these as well as the other code required to implement the RBF nets are included in Appendix D for reference.

3.3.4 Recombination. In order to calculate the HRTF from the network models at any arbitrary azimuth and elevation, the neural networks trained upon

subsets of the ALS's 272 speaker set must be somehow recombined, due to their lack of overlap.

To facilitate discussion of this process, certain terms must be specified. For our purposes, the *exterior speakers* of a particular subset are those whose adjacent speakers are not all contained within the subset, and the *boundary* of such a subset is defined simply as the collection of line segments joining adjacent exterior speakers. To be considered inside of, or interior to, such a group of speakers, a spatial location must be within the closed, connected region outlined by the subset boundary, which is equivalent to requiring the position to be within an NNT composed of speakers within the region.

It seems reasonable to assume that it is valid to simulate the HRTFs of spatial locations interior to a particular network using the weights and biases produced by that network. The problem of recombination, then, reduces to the question of how to simulate HRTFs for those regions between network boundaries, interior to none of the tessellated subsets. One possibility is to modify the tessellation schemes to allow for overlap, however, this results in larger networks which are notably more difficult to train.

If it is assumed that the ANN simulated HRTFs are invalid on regions exterior to their training data, then one must seek to somehow combine the results of surrounding networks for points betwixt disjoint speaker subsets. This could be accomplished by simulating ANN approximations at each of the three surrounding speaker positions and then using any of the four simple interpolants described at the beginning of this chapter to combine the resultant HRTFs. Of course, using the method of nearest point rounding for inter-network blending would result in discontinuities in the regions between network borders. The method of weighting an arbitrary number of nearest neighbors in weighted average would result in discontinuities around the 272 speaker positions due to its defaulting to nearest neighbor rounding near those regions. Finally, the latter two of the four simple interpolant

methods, the NNT based weighted average and piecewise linear spline, would both produce a continuous HRTF over the entire spherical coordinate system. This is a very desirable result, from a physical standpoint, as the actual head related transfer function is considered to be continuous.

Of course, it is not entirely reasonable to presume that ANN simulated HRTFs are prefectly valid within the training region, and yet somehow become completely invalid the moment they venture outside of it. It is more likely that they become less reliable approximators in a gradual fashion as they are simulated further from center of the training set. If this is truly the case, then it may be reasonable to use the azimuth and elevation of the interpolant position as inputs to the ANN for interpolant positions within boundary regions. If this technique is valid, a number of alternative methods for network recombination present themselves.

Perhaps the easiest way to accomplish recombination is to once again make use of nearest neighbor considerations. One may simply simulate the value of the HRTF at the desired interpolant position using the weights and biases from the nearest ANN subset. Note that this is not equivalent to the above method of nearest point rounding; while both techniques use the weights and biases from the nearest ANN subregion, this approach takes the inputs of azimuth and elevation from the interpolant position rather than the nearest data position. Of course, the obvious difficulty with this scheme is the formation of step discontinuities along the borders between tessellated subsets.

A more elegant approach would be to formulate an algorithm by which the simulated results of the trained ANNs may be combined using windowed weighting functions, similar to the lapped orthogonal transforms now used to reduce edge effects in digital image and sound encoding (1) (23).

The practical implementation of these windows on the ANNs may be greatly simplified by the assumption that the window for a particular network should be of unitary weight inside the region of network training data, and zero inside regions on

which other networks were trained. There remains only the formulation of window values in the regions between the tessellated networks.

As previously mentioned, the ALS naturally decomposes into roughly equilateral triangles of adjacent speaker points, called nearest neighbor triangles (NNTs). If the windowed weighting functions could be adequately defined for any arbitrary position within any NNT positioned between the trained networks, then windows in all of the boundary regions would be defined. In combination with the aforementioned suppositions, such a scheme would suffice to define the windows over the entire ALS.

For an NNT joining two networks which have been tessellated as described in the previous section, one of the three speakers, call it speaker 1, belongs to one of the two networks, call it network 1, whereas the other two speakers, which may be referred to as speakers 2a and 2b, are assigned to the other network, here designated network 2. Consider the line segment joining the point A at speaker 1 with the point B along the spar between speakers 2a and 2b, constrained to pass through the desired interpolant position X, as shown in Figure 14.

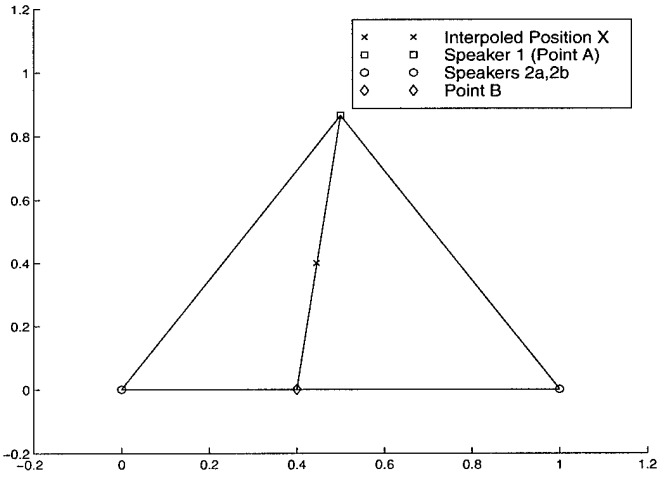


Figure 14. NNT with two networks

For any interpolant position on the ALS, the line segment described above exists and may be used to ascribe a continuous, univariate windowed weighting scheme to the ANN model. An example of such a weighting scheme was provided

by Suter and Oxley (38). Utilized herein, this technique relies upon trigonometric basis functions defined on the overlap region so that the squares of their values sum to unity at any position in that region. For a variable x representing the distance $d(A, X)$ from point A to the interpolant position X and an overlap region of width 2ϵ, the functions

$$\mathrm{w}_1(x) = \begin{cases} 1 & 0 < x < \frac{d(A,B)-2\epsilon}{2} \\ \cos\left(\frac{x-[d(A,B)\frac{(1-2\epsilon)}{2}]}{d(A,B)(2\epsilon)}\right) & \frac{d(A,B)-2\epsilon}{2} < x < \frac{d(A,B)+2\epsilon}{2} \\ 0 & \frac{d(A,B)-2\epsilon}{2} < x < d(A,B) \end{cases}$$

$$\mathrm{w}_2(x) = \begin{cases} 1 & 0 < x < \frac{d(A,B)-2\epsilon}{2} \\ \sin\left(\frac{x-[d(A,B)\frac{(1-2\epsilon)}{2}]}{d(A,B)(2\epsilon)}\right) & \frac{d(A,B)-2\epsilon}{2} < x < \frac{d(A,B)+2\epsilon}{2} \\ 0 & \frac{d(A,B)-2\epsilon}{2} < x < d(A,B) \end{cases}$$

fulfill the general requirements for amplitude normalized windows (38). These windows are depicted in Figure 15.

After defining the windows, a single continuous function $h(x)$ is produced by using the window function weights $w_1(x), w_2(x)$ to combine HRTF values $h_1(x), h_2(x)$, in quadrature sum.

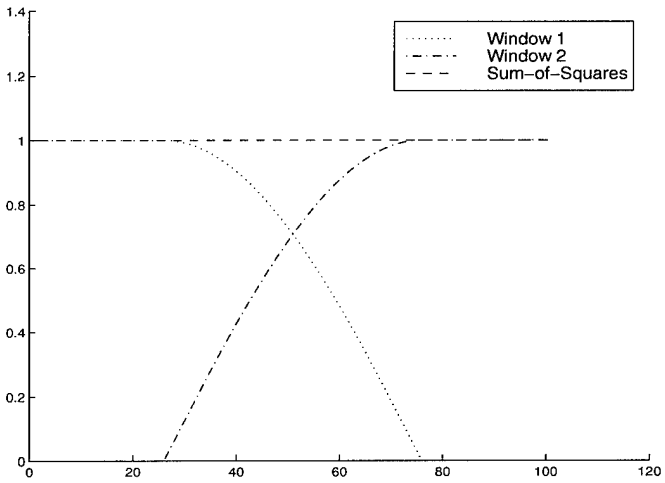


Figure 15. 1-D Overlapped Windows with Weighted Orthonormal Bases

$$f(x) = \sqrt{w_1(x)f_1(x) + w_2(x)f_2(x)}$$

Of course, this is not the only possible blending method. Simple affine functions may also be used. Define the window functions alternatively as

$$\mathrm{w}_1(x) = \begin{cases} 1 & 0 < x < \frac{d(A,B)-2\epsilon}{2} \\ \text{1-x} & \frac{d(A,B)-2\epsilon}{2} < x < \frac{d(A,B)+2\epsilon}{2} \\ 0 & \frac{d(A,B)-2\epsilon}{2} < x < d(A,B) \end{cases}$$

$$w_2(x) = \begin{cases} 1 & 0 < x < \frac{d(A,B)-2\epsilon}{2} \\ x & \frac{d(A,B)-2\epsilon}{2} < x < \frac{d(A,B)+2\epsilon}{2} \\ 0 & \frac{d(A,B)-2\epsilon}{2} < x < d(A,B) \end{cases}$$

to be used in conjunction with an ordinary sum

$$f(x) = w_1(x)f_1(x) + w_2(x)f_2(x)$$

Such window functions are shown in Figure 16. They were implemented alongside the orthonormal windows to determine whether there is a significant advantage to the former, energy conservative method.

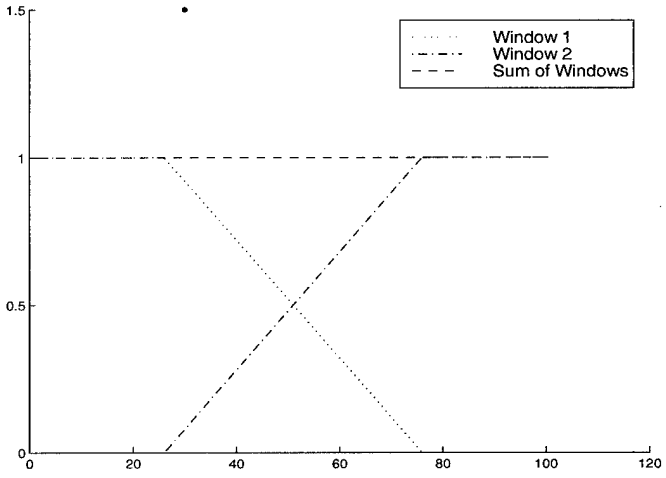


Figure 16. 1-D Overlapped Windows with Linear Bases

Since both ANN architectures are based upon successive layers of continuous function transforms of continuous input variables, it is clear that the outputs of the ANNs will vary continuously with the inputs. Since the windowed transforms are

continuous over the entire ALS, it is clear that this windowing method applied to the ANNs will yield an HRTF free of spatial discontinuities.

The borders formed by this weighting scheme are shown in Figures 17 and 18, where the lines represent the midpoints at which the weights are equal for both transform functions.

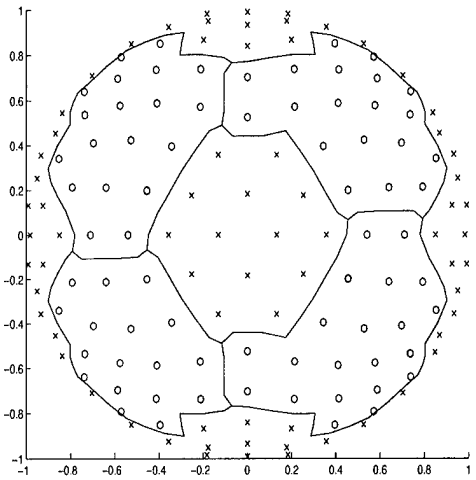


Figure 17. Similar ILD regions: caps

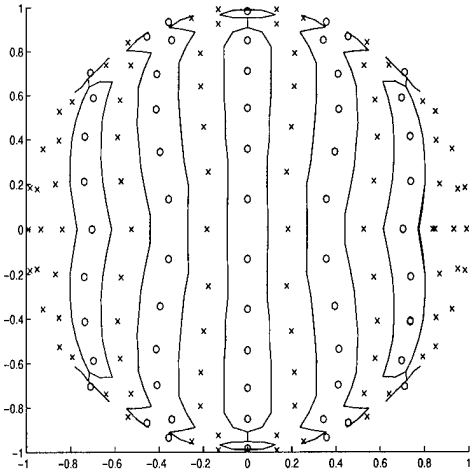


Figure 18. Similar ITD regions: small circles

This weighted scheme as well as the latter two of the four simple interpolants described above were successfully employed for the blending together of the ANN HRTFs into a single, continuous HRTF. The results are given in the next chapter.

3.4 Conclusion

The tessellation of the ALS data sets, subsequent building and training of both MLP and RBF neural networks upon the resultant regions, and the recombination techniques for the networks have been detailed, along with alternative simple interpolants for comparison and validity checking. The results of these methodologies are presented in the following chapter.

IV. Data Analysis

4.1 Introduction

The interpolants and ANN models implemented in this work were described in the former chapter, this chapter delineates the testing and analysis of those methods. A description of the interpolant testing process is given, followed by results and comments for each method.

All computational processing for this thesis was accomplished in the Matlab 5.1 environment on the AFIT's Signal Information Processing Laboratory (SIPL) Sun Ultra platforms. Digital signal processing for 3-D audio synthesis was performed using Entropic's Signal Processing System (ESPS) on a Sun SPARCstation 5 equipped with an Ariel board. No products of hegemonic software vendors were used in this research.

4.2 Validation

In Millhouse's precursor to this effort, model verification was accomplished via human testing and statistical analysis. While such a testing method has the advantage of perceptually proving model results, it suffers from several possible sources of error, including human and experimental error, infidelities in the digital signal processing required to simulate the binaural signals from the HRTFs, and a loss in spatialization ability due to the use of non-individualized HRTFs. The lattermost difficulties are due to the fact the human subjects generally localize poorly on HRTFs generated from pinnae other than their own; we are not skilled at listening with other people's ears (35). All of these problems tend to obfuscate the experimental results, making it difficult to accurately determine model validity.

An alternative validation of interpolation and modeling techniques may be achieved by comparing model results against actual data for specific spatial locations (12). While comparing the model against its training data may provide some

indication of model accuracy, for the comparison of modeled to sampled data to be a valid indicator of interpolative ability it must be performed against measured data points not included in the original training set (5) (33). This necessitates either a removal of sample points from the ALS data set, or else a measurement of additional HRTF data points for interpolant testing. As the ALS has a nearly uniform 12-14 degree angular separation between data points, the latter technique is desirable, as the former would require interpolation over extremely spatially distanced data points, yielding poor results.

Accordingly, additional data was obtained from the ALS for the testing of ANNs and other interpolant methods. This was achieved by simply rotating the KEMAR mannikin and running another test set for all of the 272 speaker positions and 104 test frequencies. The mannikin was rotated by 2.5, 5.0, and 7.5 degrees, resulting in three full data sets in addition to the original.

Testing was accomplished by evaluating the HRTF at each of the measured data locations and test frequencies using the interpolant method or ANN model and comparing the results against the empirically determined values at those points. For every test frequency at a fixed spatial position, the error between measured and simulated values is computed, and then averaged over all 272 spatial positions and 104 frequencies to determine the average squared error between the interpolant and the sample data for fixed positions and frequencies. Each method of interpolation/modeling has four such statistics, one corresponding to each of the four test sets: 0.0, 2.5, 5.0, and 7.5 degrees rotation. The results are compiled and presented below.

4.3 Simple Interpolants

The performance of the four simple interpolants is shown in Table 2. The first column of zero values validates the earlier claim that each of these methods is constructed so as to match the value of the HRTF exactly at measured spatial

Table 2. Simple Interpolant Results, Average SSE (dB^2)

Interpolation Method Employed	Avg. SSE at 0 deg.	Avg. SSE 2.5 deg.	Avg. SSE 5.0 deg.	Avg. SSE 7.5 deg.
Simple Rounding	0.0000	0.5158	1.0023	8.6880
NN Weighted Avg.	0.0000	0.6586	1.1741	5.8702
3NN Weighted Avg.	0.0000	0.4767	0.8462	6.1548
Piecewise Linear	0.0000	0.5236	1.0301	6.0167

locations. It is interesting to note that no one of these methods clearly outperforms the others. It appears that simple rounding does relatively better closer to the sample points, whereas the nearest neighbor weighted average performs better in regions further from these points; the latter two methods appear to do about equally well. All of these results confirm the observations on these methods made in the previous chapter.

Table 2 is useful as a baseline by which the various other results will be judged, as it depicts several of the types of interpolant schemes which have hitherto been in common use in 3-D audio research, judged for closeness based on the most widely used norm.

4.4 MLP Results

The multilayer perceptrons were slow to train and produced relatively poor results. Unfortunately, the incredible amount of RAM necessary to store the Jacobian in the LM approximation induced an inordinate amount of page-swapping, which slowed down the training considerably, forcing the use of gradient descent training with momentum. Training the MLPs on the entire 272-point data set for a single tessellation method required several days runtime.

Table 3 provides the results of the multilayer perceptrons trained using the L2 norm. Note the average sum of squares errors tends to decrease with the number of subsets (caps or circles) used in tessellation. This is exactly as one might expect, as it is easier to achieve accurate neural network representation on smaller subsets

Table 3. MLP Model Results, No Blending, Average SSE (dB^2)

Tessellation Scheme Employed	Avg. SSE 0 deg.	Avg. SSE 2.5 deg.	Avg. SSE 5.0 deg.	Avg. SSE 7.5 deg.
12 caps (repl)	14.6029	15.0025	14.4786	15.097
14 caps (cube)	11.5352	12.7250	12.0751	13.1042
11 ITD circles	17.8477	18.0848	17.1805	19.4359
13 ITD circles	11.3696	11.1340	10.0316	13.9271
14 ITD circles	11.6327	10.2412	10.9340	11.4058

(5). As mentioned in Chapter III, the disadvantage in creating more subregions is the loss of effective data compression.

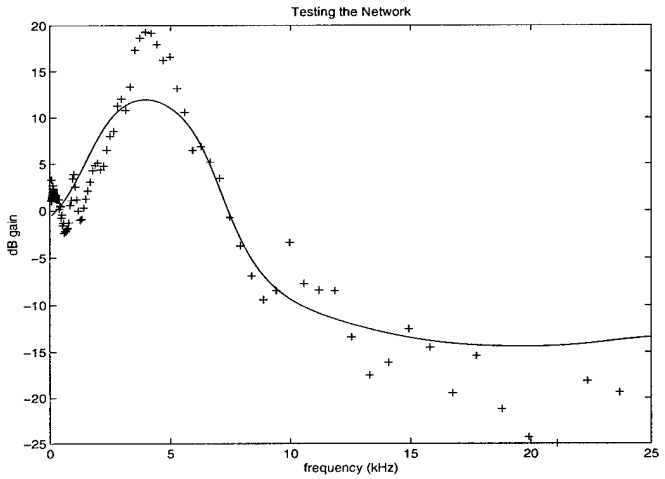


Figure 19. MLP Simulation at Azimuth=64.09, Elevation=0.0

Figure 19 shows a representative sample of MLP model results for a particular DTF at azimuth zero and elevation 79.43, simulated from a network trained upon the 24-speakers in the sagittal plane. The average SSE on this DTF is 10.9456, and thus it is representative of the various MLP models. The tendency of the MLP to smooth the data is evident.

4.5 RBF Results

The implementation and results of the RBF networks were more encouraging than those of the MLPs. RBF nets trained rapidly, taking less than two hours to

encapsulate all of the tessellated regions in the data set. Networks were constructed for the entire ALS using both of the proposed tessellation schemes and simulated with both of the proposed windowed blending methods mentioned in the previous chapter. Results and associated comments follow.

Table 4. RBF Model Results, No Blending, Average SSE (dB^2)

Tessellation Scheme Employed	Avg. SSE 0 deg.	Avg. SSE 2.5 deg.	Avg. SSE 5.0 deg.	Avg. SSE 7.5 deg.
12 caps (repl)	6.3068	6.6972	6.6062	10.1045
14 caps (cube)	5.5670	5.8657	5.8788	9.5089
11 ITD circles	6.7467	7.1230	7.0594	10.6650
13 ITD circles	6.5994	6.9708	6.9329	10.5470
14 ITD circles	6.5314	6.9021	6.8587	10.6406

The results for the RBF without any blending are shown in Table 4. Here again the SSE tends to decrease with more subdivisions in the tessellation. It appears that the two tessellation methods performed very nearly equally, with perhaps a slight advantage in the caps-based scheme.

Table 5. RBF Model Results, Linear Blending, Average SSE (dB^2)

Tessellation Scheme Employed	Avg. SSE 0 deg.	Avg. SSE 2.5 deg.	Avg. SSE 5.0 deg.	Avg. SSE 7.5 deg.
12 caps (repl)	6.3068	6.6972	6.4939	9.7066
14 caps (cube)	5.5670	5.8657	5.7830	9.2554
11 ITD circles	6.7467	7.1230	6.8798	10.1376
13 ITD circles	6.5994	6.9708	6.7437	10.0074
14 ITD circles	6.5314	6.9021	6.8667	10.1304

Results for the RBF networks blended using linearly tapered windows with parameter ϵ equal to one-fourth of the total window width $d(A, B)$ are shown in Table 5. It appears that this blending technique provided only marginal improvement over merely simulating the HRTF from the nearest network without any blending. This result is probably not a repudiation of the blending technique, but rather a validation of the ANN models, demonstrating the truth of aforementioned conjecture that the networks perform well even at the spatial boundaries of their training sets.

Table 6. RBF Model Results, Trigonometric Blending, Average SSE (dB^2)

Tessellation Scheme Employed	Avg. SSE 0 deg.	Avg. SSE 2.5 deg.	Avg. SSE 5.0 deg.	Avg. SSE 7.5 deg.
12 caps (repl)	6.3068	6.6972	6.5266	9.8089
14 caps (cube)	5.5670	5.8657	5.8045	9.2901
11 ITD circles	6.7467	7.1230	6.9212	10.1947
13 ITD circles	6.5994	6.9708	6.7896	10.0692
14 ITD circles	6.5314	6.9021	6.8751	10.1978

The results of RBF networks blended with trigonometric windows forming lapped orthonormal transforms, again with $\epsilon = 0.25\, d(A, B)$, are shown in Table 6. Perhaps surprisingly, it appears that this method provided no significant advantage over linear blending.

Table 7. RBF Model Results, Triangle Weighted Average Blending, Average SSE (dB^2)

Tessellation Scheme Employed	Avg. SSE 0 deg.	Avg. SSE 2.5 deg.	Avg. SSE 5.0 deg.	Avg. SSE 7.5 deg.
12 caps (repl)	6.3068	6.5747	6.2366	9.1784
14 caps (cube)	5.4496	5.4790	5.4397	7.0723
11 ITD circles	6.7467	7.0594	6.7896	9.9532
13 ITD circles	6.5994	6.9019	6.6506	9.8227
14 ITD circles	6.5314	6.8175	6.5597	9.5874

Table 8. RBF Model Results, Piecewise Linear Blending, Average SSE (dB^2)

Tessellation Scheme Employed	Avg. SSE 0 deg.	Avg. SSE 2.5 deg.	Avg. SSE 5.0 deg.	Avg. SSE 7.5 deg.
12 caps (repl)	6.3068	6.5743	6.3768	9.4870
14 caps (cube)	5.4496	5.4830	5.3827	7.3254
11 ITD circles	6.7467	7.0134	6.7742	10.0462
13 ITD circles	6.5994	6.8538	6.6387	9.9107
14 ITD circles	6.5314	6.7914	6.6112	9.7223

Tables 7 and 8 give the results of the RBFs networks blended using simple interpolants. As described in the previous chapter, these methods use the ANNs weights and biases to simulate the HRTF at the surrounding three data points, rather

than at the desired interpolation position. Despite this change in methodology, the results from these blending methods do not appear significantly different than those from the others.

All of the tessellations and blendings of RBF networks yielded very promising results, with average SSEs in the range from 5 to 10, indicating that the average absolute deviation of the model was about 2-3 dB. These results may be graphically demonstrated by plotting DTFs generated from the RBF networks. In Figures 20 to 22, such results are given for networks trained on a 12-cap tessellation, tested at positions given by a 7.5 degree azimuthal rotation from the training set.

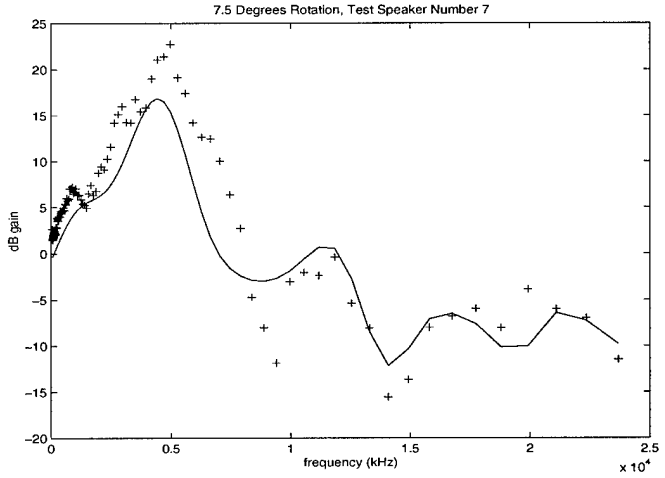


Figure 20. RBF Approximation at Azimuth = 105.09, Elevation = 0

Figure 20 shows an example of a atypically poor RBF approximation. The interpolated position here is midway in between training speakers 7 and 8, and the model yields an average DTF SSE of 12.3678. In is clear; however, that even this case the RBF yields reasonable results.

Figure 21 illustrates more typical RBF performance, with an average SSE of 6.0476. The interpolated position in this case was again on the horizon circle, nearly midway in between training data speaker positions 3 and 4. Note that the RBF approximation manages to avoid the excessive smoothing of the MLP, closely following the jagged, complex contours of the DTF.

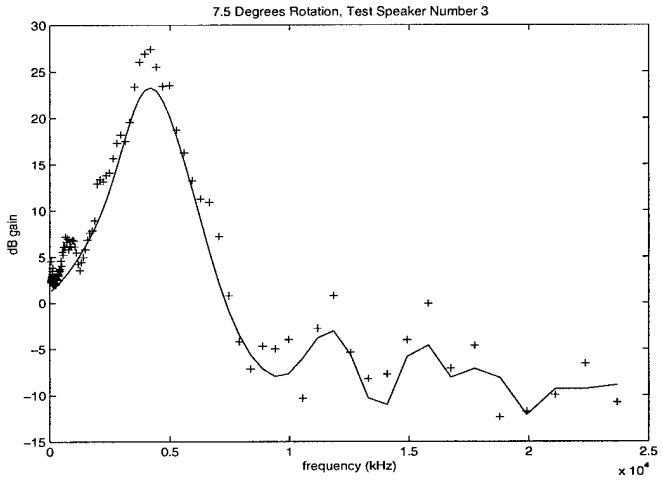


Figure 21. RBF Approximation at Azimuth = 52.2, Elevation = 0

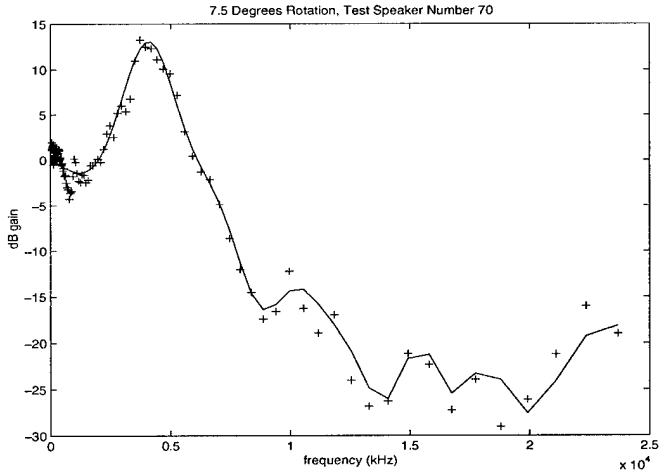


Figure 22. RBF Approximation at Azimuth = 277.5, Elevation = 57.25

Figure 22 depicts the RBF model performing very well, with an average DTF SSE of only 1.6807 between measured and modeled values.

4.6 Conclusion

The results of this research are encouraging. While the MLPs are not quite sufficient for the task of HRTF approximation, the RBF networks performed well, demonstrating the validity of both tessellation schemes. Since the average squared value of the full training set itself was 84.9964, the RBF model's average SSEs between 5-10 dB^2 correspond to average percent errors of around 5% to 11%, which are comparable to the results reported by Chen for his SFER model (12). Finally, it should be noted that average squared difference between ALS HRTFs for two different individuals is typically on the order of 20-30 dB^2, implying that the errors in RBF models are certainly tolerable for modeling non-individualized HRTFs.

V. Conclusions and Recommendations

5.1 Summary

Overall, the results of this research effort are quite satisfactory. All of the desired objectives were accomplished: the tessellation and recombination problems were solved, and both the ANN models and the computationally efficient simple interpolants were successfully implemented and tested.

The simple interpolants included nearest point rounding, two weighted averages, and a piecewise linear spline; the ANN models implemented were the RBF network and the MLP.

The tessellation problem was solved by creating caps of similar spatial position and circles of similar ITD, both of which performed similarly well. Recombination was accomplished via the simple interpolants as well as a novel application of variable overlapped windows to the HRTF.

Testing was achieved by comparing the results of the models and interpolants to measured data at positions both in between and coincident with those of the training data set.

5.2 Conclusions

The MLP ANNs performed relatively poorly, with sluggish training and overly smoothed approximate HRTF gain surfaces. By contrast, the RBF nets performed rather well, yielding results comparable to both the simple interpolants and Chen's SFER method. The RBF ANNs are clearly of practical value, as they could successfully compress an HRTF data set into a diminutive set of values which may be readily impressed into an integrated circuit.

5.3 Recommendations for Further Research

This thesis represents the first encapsulation of the entire human HRTF by neural network models, and as such leaves a number of parameters and processes which may be further varied and optimized, the most important of which are detailed below.

Firstly, the tessellation schemes may be further refined, to represent the HRTF as accurately as possible while decreasing the number of subregions and weights used, thereby increasing the compressive value of the network model. The most obvious step in this direction would be to optimize the shape of the tessellated subsets. Numerous other possibilities also present themselves, such as using overlapping, variably sized networks, with variable numbers of weights. Also the implementation of the lapped windows may be refined, particularly the choice of window basis functions, the shape of the windows, and amount of overlap used.

Secondly, the RBF nets themselves could be optimized, both in their mathematical structure and their algorithmic and coded implementations. For example, a more optimal choice of the number of basis functions and their widths may be systematically determined. A recurrent problem in this effort was the near halting of training due to a lack of available RAM. Accordingly, more memory efficient algorithms may be developed for network training, which should also aid in the maximizing the objective of effective HRTF data compression.

Finally, the model's testing procedures may also be improved upon. The data sets should be rotated in elevation in addition to azimuth, as the ALS HRTF sets are already relatively dense in azimuth. Also, human testing may be included, although this is not highly recommended without the use of individualized HRTFs for training and testing; the models will likely perform well enough to obfuscate any statistically significant performance differences from subject to subject using non-individualized HRTFs.

With subsequent improvements in these various aspects of neural network modeling of the HRTF, this research may progress from an academic exercise into a useful system component of various virtual audio displays, including those used in USAF cockpits.

Appendix A. Sample HRIR data files

The following are samples of the HRIR data files recorded from the AAMRL ALS, for speaker number one, at an azimuth of 10.57 degrees, with zero elevation.

file LSPKR12.1 (bamar data)	-11.585825
-29.623213	-10.878924
-28.161764	-6.166762
-28.245293	-6.785713
-28.330460	-7.491044
-27.459211	-2.951831
-28.528421	-3.311255
-27.661791	-2.523776
-26.998594	-0.880440
-27.922951	-0.946511
-27.267853	0.025304
-26.765709	0.343239
-27.609209	-0.382943
-25.582281	0.247662
-21.367271	-0.490228
-21.138245	0.279501
-21.062008	0.455494
-21.576046	0.480620
-22.159197	0.576885
-22.416086	0.474472
-21.412340	0.970449
-16.137003	2.364222
-15.999801	2.325866
-16.233906	2.644288
-16.934948	2.475843
-16.311758	0.076000
-10.561598	-3.354200
-10.850018	-1.583782

0.321021
1.371587
2.191915
3.315938
4.121714
1.710832
-2.623638
-4.746835
-3.493719
-1.371575
-0.130292
2.097157
3.283939
5.320154
7.376795
9.483487
11.211603
13.466881
17.123262
20.179850
19.910419
14.420902
12.468328
9.869209
9.521591
13.426007
11.272694
7.459867

3.935199
2.021008
-3.721013
-10.061999
-9.409718
-8.724374
-9.816716
-6.685171
-4.699739
-6.286468
-15.160823
-21.442904
-13.669736
-16.057402
-16.190948
-25.133207
-26.088293
-35.511532
-48.527409
-46.965027
-48.341373

	-8.331797
file SPEAK12A.1	-8.543134
(free field calibration data)	-9.120838
-32.397095	-8.304572
-26.520485	-3.355388
-26.740681	-4.109616
-26.972641	-5.009285
-26.280220	-0.514778
-27.544832	-0.773897
-26.892828	-0.205523
-26.522924	1.006315
-27.782087	1.178232
-27.514050	1.662371
-27.557976	2.035545
-29.081507	1.096707
-24.996857	1.784897
-18.894192	1.129538
-18.700518	1.961300
-18.672766	2.266683
-19.251520	3.288610
-19.919436	2.847599
-20.299879	2.772577
-19.178869	1.820756
-13.214403	1.502970
-13.099646	0.719653
-13.393995	-0.260097
-14.180647	-1.062705
-13.596003	-1.205330

-0.986382	2.709274
-0.459718	1.869489
0.630784	0.023707
1.693635	-1.052404
1.176250	-1.917290
0.282985	-1.280503
0.359488	0.580573
0.711930	2.627077
1.998464	2.970467
2.464921	4.961968
2.734841	5.420332
2.178993	0.697337
1.637267	-6.725608
1.482344	-9.653347
0.905720	-7.718470
0.955579	-10.685308
0.087083	-9.336636
-1.901385	-14.640713
-3.384275	-19.143776
-0.970028	-25.560707
1.222447	-35.530663
4.016621	-30.551491
4.335315	-30.796616
-0.797260	
-1.087929	
-1.874729	
-0.861694	
4.187546	

Appendix B. Speaker Locations for HRTF Measurements

The following is a list of the speaker designation numbers, locations and associated ITDs for each of the 272 speaker positions on the ALS. Azimuth and elevation are in degrees, and follow the conventions described in chapter 2. ITD is in microseconds. The first datum listed is the speaker designation number, the second is the azimuth, the third is elevation, and the fourth is interaural time delay. Azimuth is defined as the angular separation from the front half saggital plane to the vertical plane of the sound source, measured in the counterclockwise direction as seen from above, as shown in figure 4. Elevation is simply the angular separation from the sound source to the horizon, as shown in figure 5. Both of these are measured in degrees, while time delay is expressed in microseconds.

spkr#	azimuth	elev.	ITD	spkr#	azimuth	elev.	ITD
1	10.57	0.00	90	41	180.00	-20.91	0
2	31.72	0.00	260	42	180.00	-7.59	0
3	44.71	0.00	372	43	180.00	7.59	0
4	57.98	0.00	470	44	180.00	20.91	0
5	69.09	0.00	570	45	180.00	32.77	0
6	82.41	0.00	658	46	180.00	45.29	0
7	97.59	0.00	730	47	180.00	58.28	0
8	111.09	0.00	770	48	180.00	79.43	0
9	123.02	0.00	480	49	90.00	82.41	54
10	135.29	0.00	388	50	90.00	69.09	186
11	148.28	0.00	280	51	90.00	57.25	320
12	169.43	0.00	100	52	90.00	44.71	450
13	190.57	0.00	85	53	90.00	31.72	573
14	211.72	0.00	274	54	90.00	10.57	690
15	224.71	0.00	390	55	90.00	-10.57	648
16	236.98	0.00	480	56	90.00	-31.72	510
17	249.09	0.00	784	57	90.00	-44.71	428
18	262.41	0.00	744	58	90.00	-57.25	334
19	277.59	0.00	675	59	90.00	-69.09	210
20	290.91	0.00	586	60	90.00	-82.41	66
21	3 3.02	0.00	484	61	270.00	-82.41	112
22	315.25	0.00	380	62	270.00	-69.09	264
23	328.28	0.00	265	63	270.00	-57.25	396
24	349.43	0.00	90	64	270.00	-44.71	460
25	0.00	79.43	0	65	270.00	-31.72	556
26	0.00	58.28	0	66	270.00	-10.57	684
27	0.00	45.29	0	67	270.00	10.57	680
28	0.00	32.77	0	68	270.00	31.72	564
29	0.00	20.91	0	69	270.00	44.71	423
30	0.00	7.59	0	70	270.00	57.25	320
31	0.00	-7.59	0	71	270.00	69.09	180
32	0.00	-20.91	0	72	270.00	82.41	50
33	0.00	-32.77	0	73	20.30	67.60	80
34	0.00	-45.29	0	74	20.50	52.40	120
35	0.00	-58.28	0	75	16.00	39.80	125
36	0.00	-79.43	0	76	12.90	27.10	114
37	180.00	-79.43	0	77	10.60	14.70	100
38	180.00	-58.28	0	78	55.00	72.00	150
39	180.00	-45.29	0	79	40.90	58.40	282
40	180.00	-32.77	0	80	34.90	44.20	212

spkr#	azimuth	elev.	ITD	spkr#	azimuth	elev.	ITD
81	29.30	32.40	271	121	98.10	20.90	630
82	24.90	20.10	215	122	104.90	10.20	730
83	21.10	7.60	206	123	200.30	67.60	56
84	66.30	60.20	344	124	200.50	52.40	110
85	51.20	47.40	414	125	196.00	39.80	106
86	45.00	35.30	449	126	192.90	27.10	90
87	40.10	24.20	354	127	190.60	14.70	80
88	35.80	12.30	324	128	235.00	72.00	124
89	71.70	47.60	498	129	220.90	58.40	180
90	59.50	36.00	532	130	214.90	44.20	220
91	53.90	24.40	400	131	209.30	32.40	220
92	49.10	12.20	411	132	204.90	20.10	205
93	74.90	34.80	558	133	201.10	7.60	180
94	68.20	23.30	648	134	246.30	60.20	250
95	62.40	11.50	510	135	231.20	47.40	299
96	81.90	20.90	630	136	225.00	35.30	320
97	75.10	10.20	620	137	220.10	24.20	323
98	159.70	67.60	90	138	215.80	12.20	310
99	159.50	52.40	130	139	251.70	47.60	370
100	164.00	39.80	130	140	239.50	36.00	431
101	167.10	27.10	120	141	233.90	24.40	430
102	169.40	14.70	105	142	229.10	12.20	420
103	125.00	72.00	160	143	254.90	34.80	493
104	139.10	58.40	210	144	248.20	23.30	530
105	145.10	44.20	250	145	242.40	11.50	523
106	150.70	32.40	245	146	261.90	20.90	610
107	155.10	20.10	230	147	255.10	10.20	750
108	158.90	7.60	200	148	339.70	67.60	50
109	113.70	60.20	280	149	339.50	52.40	100
110	128.80	47.40	327	150	344.00	39.80	113
111	135.00	35.30	350	151	347.10	27.10	90
112	139.90	24.20	340	152	349.40	14.70	80
113	144.20	12.30	310	153	305.00	72.00	120
114	108.30	47.60	400	154	319.10	58.40	170
115	120.50	36.00	430	155	325.10	44.20	210
116	126.10	24.40	440	156	330.70	32.40	210
117	130.90	12.20	430	157	335.10	20.10	194
118	105.10	34.80	520	158	338.90	7.60	170
119	111.80	23.30	543	159	293.70	60.20	235
120	117.60	11.50	530	160	308.80	47.40	303

spkr#	azimuth	elev.	ITD	spkr#	azimuth	elev.	ITD
161	315.00	35.30	300	201	159.50	-52.40	102
162	319.90	24.20	303	202	159.70	-67.60	60
163	324.20	12.30	292	203	158.90	-7.60	190
164	288.30	47.60	360	204	155.10	-20.10	200
165	300.50	36.00	390	205	150.70	-32.40	200
166	306.10	24.40	404	206	145.10	-44.20	200
167	310.90	12.20	407	207	139.10	-58.40	170
168	285.10	34.80	470	208	125.00	-72.00	300
169	291.80	23.30	510	209	144.20	-12.30	370
170	297.60	11.50	518	210	139.90	-24.20	290
171	278.10	20.90	615	211	135.00	-35.30	280
172	284.90	10.20	628	212	128.80	-47.40	270
173	10.60	-14.70	80	213	113.70	-60.20	298
174	12.90	-27.10	90	214	130.90	-12.20	400
175	16.00	-39.80	80	215	126.10	-24.40	380
176	20.50	-52.40	90	216	120.50	-36.00	360
177	20.30	-67.60	60	217	108.30	-47.60	368
178	21.10	-7.60	172	218	117.60	-11.50	480
179	24.90	-20.10	190	219	111.80	-23.30	462
180	29.30	-32.40	2 5	220	105.10	-34.80	454
181	34.90	-44.20	180	221	104.90	-10.20	780
182	40.90	-58.40	158	222	98.10	-20.90	550
183	55.00	-72.00	370	223	190.60	-14.70	83
184	35.80	-12.30	280	224	192.90	-27.10	90
185	40.10	-24.20	290	225	196.00	-39.80	90
186	45.00	-35.30	270	226	200.50	-52.40	110
187	51.20	-47.40	250	227	200.30	-67.60	80
188	66.60	-60.20	266	228	201.10	-7.60	180
189	49.10	-12.20	386	229	204.90	-20.10	190
190	53.90	-24.20	394	230	209.30	-32.40	207
191	59.50	-36.00	340	231	214.90	-44.20	212
192	71.70	-47.60	540	232	220.90	-58.40	190
193	62.40	-11.50	490	233	235.00	-72.00	348
194	68.20	-23.20	470	234	215.80	-12.30	300
195	74.90	-34.80	458	235	220.10	-24.20	300
196	75.10	-10.20	590	236	225.00	-35.30	290
197	81.90	-20.90	556	237	231.20	-47.40	286
198	169.40	-14.70	90	238	246.60	-60.20	307
199	167.10	-27.10	92	239	229.10	-12.20	402
2	164.00	-39.80	100	240	233.90	-24.40	390

spkr#	azimuth	elev.	ITD
241	239.50	-36.00	372
242	251.70	-47.60	379
243	242.40	-11.50	490
244	248.20	-23.20	470
245	254.90	-34.80	470
246	255.10	-10.20	790
247	261.90	-20.90	576
248	349.40	-14.70	90
249	347.10	-27.10	100
250	344.00	-39.80	95
251	339.50	-52.40	100
252	339.70	-67.60	72
253	338.90	-7.60	180
254	335.10	-20.10	196
255	330.70	-32.40	208
256	325.10	-44.20	190
257	319.10	-58.40	170
258	305.00	-72.00	320
259	324.20	-12.30	296
260	319.90	-24.20	306
261	315.00	-35.30	288
262	3 8.80	-47.40	270
263	293.70	-60.20	418
264	310.90	-12.20	400
265	306.10	-24.40	390
266	300.50	-36.00	360
267	288.30	-47.60	560
268	297.60	-11.50	510
269	291.80	-23.30	510
270	285.10	-34.80	500
271	284.90	-10.20	610
272	278.10	-20.90	590

Appendix C. Adjacency Matrix for the AAMRL ALS

The adjacency matrix for the ALS is an enumeration of the speakers adjacent to every speaker on the sphere. The speaker under consideration is listed in the first column, and its five or six adjacent speakers are listed in the next six columns. Notice that for some speakers the sixth column is blank, indicating that the speaker in question is part of one of the few pentagonal structures on the sphere, and is thus adjacent to only five other speakers, as shown in figure 3.

1 77 30 31 173 178 83

2 88 3 184 178 83

3 92 4 189 184 2 88

4 95 5 193 189 3 92

5 95 97 6 196 193 4

6 97 54 7 55 196 5

7 54 122 8 221 55 6

8 120 9 218 221 7 122

9 10 214 218 8 120 117

10 113 11 209 214 9 117

11 108 203 209 10 113

12 43 42 198 203 108 102

13 127 43 42 223 228 133

14 133 228 234 15 138

15 14 234 16 142 138

16 15 239 243 17 145 142

17 16 147 18 246 243 145

18 147 67 19 66 246 17

19 67 172 20 271 66 18

20 172 170 21 268 271 19

21 170 167 22 264 268 20

22 167 163 23 259 264 21

23 163 158 253 259 22

24 152 30 31 248 253 158

25 148 73 153 72 49 78

26 149 148 73 74 27

27 26 74 75 28 150 149

28 150 27 29 76 75 151

29 152 151 28 76 77 30

30 29 77 1 31 24 152

31 1 173 32 248 24 30

32 31 173 174 33 249 248

33 32 174 175 34 250 249

34 33 175 176 35 251 250

35 34 176 177 252 251

36 183 60 61 177 252 258

37 202 208 227 61 233 60

38 39 201 202 227 226

39 38 201 200 40 225 226

40 41 199 200 39 225 224

41 40 224 223 42 199 198

42 41 198 223 12 13 43

43 12 42 13 102 44 127

44 102 43 127 126 45 101

45 100 101 44 126 125 46

46 100 45 125 124 47

47 46 99 98 123 124

48 49 72 128 123 98 103

49 48 103 50 78 25 72

50 109 51 84 78 49 103

51 109 114 52 89 84 50

52 51 114 118 53 93 89

53 52 118 121 96 93

54 121 122 7 6 97 96

55 6 7 221 222 197 196

56 197 222 220 57 195

57 56 220 217 58 192 195

58 57 217 213 59 188 192

59 58 213 208 60 183 188

60 59 183 36 61 37 208

61 36 37 60 233 258 62

62 233 61 258 238 263 63

63 242 238 62 263 267 64

64 245 242 63 267 270 65

65 64 270 272 247 245

66 18 19 271 272 247 246

67 146 171 172 19 18 147

68 171 146 143 168 69

69 70 164 168 68 143 139

70 69 71 159 164 139 134

71 70 72 153 159 134 128

72 49 25 153 71 128 48

73 78 79 74 26 148 25

74 79 73 26 27 75 80

75 76 28 27 74 80 81

76 75 81 82 77 29 28

77 76 82 83 1 30 29

78 50 84 79 73 25 49

79 84 85 80 74 73 78

80 79 85 86 81 75 74

81 80 86 87 82 76 75

82 81 87 88 83 77 76

83 82 88 2 178 1 77

84 78 50 51 89 85 79

85 79 84 89 90 86 80

86 80 85 90 91 87 81

87 81 86 91 92 88 82

88 87 92 3 2 82 83

89 84 51 52 93 90 85

90 85 89 93 94 91 86

91 90 94 95 92 87 86

92 91 95 4 3 88 87

93 89 53 96 94 90

94 90 93 96 97 95 91

95 94 97 5 4 92 91

96 93 53 121 54 97 94

97 96 54 6 5 95 94

98 104 99 47 123 48 103

99 46 100 105 104 98 47

100 99 46 45 101 106 105

101 44 102 107 106 45 100

102 44 43 12 108 107 101

103 98 109 104 49 50 48

104 98 99 105 110 109 103

105 99 100 106 111 110 104

106 100 101 107 112 111 105

107 106 101 102 108 113 112

108 102 12 203 11 113 107

109 103 104 110 114 50 51

110 104 105 111 114 115 109

111 105 106 110 112 115 116

112 111 106 107 113 117 116

113 107 108 11 10 117 112

114 109 110 115 118 52 51

115 111 116 119 118 114 110

116 119 120 117 112 111 115

117 116 120 9 10 113 112

118 115 119 121 53 52 114

119 116 120 122 121 118 115

120 117 9 8 122 119 116

121 119 122 54 96 53 118

122 120 8 7 54 121 119

123 129 124 47 98 48 128

124 129 130 125 46 47 123

125 45 46 124 130 131 126

126 44 45 125 131 132 127

127 43 44 126 132 133 13

128 71 72 48 123 129 134

129 124 123 128 134 135 130

130 125 124 129 135 136 131

131 125 130 136 137 132 126

132 127 126 131 137 138 133

133 13 127 132 138 14 228

134 129 128 71 70 139 135

135 130 129 134 139 140 136

136 135 140 141 137 131 130

137 131 136 141 142 138 132

138 133 132 137 142 15 14

139 135 134 70 69 143 140

140 136 135 139 143 144 141

141 137 136 140 144 145 142

142 138 137 141 145 16 15

143 68 146 144 69 139 140

144 143 146 147 145 141 140

145 147 17 16 142 141 144

146 147 67 171 68 143 144

147 146 67 18 17 145 144

148 25 73 26 153 154 149

149 148 26 27 150 155 154

150 149 27 28 151 156 155

151 150 28 29 152 157 156

152 151 29 30 24 158 157

153 71 72 25 148 154 159

154 153 148 149 155 160 159

155 154 149 150 156 161 160

156 155 150 151 157 162 161

157 156 151 152 158 163 162

158 157 152 24 253 23 163

159 71 153 154 160 164 70

160 159 154 155 161 165 164

161 160 155 156 162 166 165

162 161 156 157 163 167 166

163 162 157 158 23 22 167

164 70 159 160 165 168 69

165 164 160 161 166 169 168

166 165 161 162 167 170 169

167 166 162 163 22 21 170

168 69 164 165 169 171 68

169 168 165 166 170 172 171

170 169 166 167 21 20 172

171 68 168 169 172 67 146

172 171 169 170 20 19 67

173 1 178 179 174 32 31

174 173 179 180 175 33 32

175 174 180 181 176 34 33

176 175 181 182 177 35 34

177 176 182 183 36 252 35

178 83 2 184 179 173 1

179 178 184 185 180 174 173

180 185 186 181 175 174 179

181 180 186 187 182 176 175

182 181 187 188 183 177 176

183 60 59 188 182 177 36

184 3 189 185 179 178 2

185 184 189 190 186 180 179

186 185 190 191 187 181 180

187 186 191 192 188 182 181

188 192 58 59 183 182 187

189 4 193 190 185 184 3

190 189 193 194 191 186 185

191 190 194 195 192 187 186

192 191 195 57 58 188 187

193 4 5 196 194 190 189

194 193 196 197 195 191 190

195 194 197 56 57 192 191

196 5 6 55 197 194 193

197 196 55 222 56 195 194

198 12 203 204 199 41 42

199 198 204 205 200 40 41

200 40 199 205 206 201 39

201 38 39 200 206 207 202

202 38 227 201 207 208 37

203 108 12 198 204 209 11

204 198 203 209 210 205 199

205 199 204 210 211 206 200

206 200 205 211 212 207 201

207 201 206 212 213 208 202

208 37 202 207 213 60 59

209 11 203 204 210 214 10

210 209 214 215 211 205 204

211 205 210 215 216 212 206

212 207 206 211 216 217 213

213 208 207 212 217 58 59

214 209 10 9 218 215 210

215 210 214 218 219 216 211

216 212 211 215 219 220 217

217 58 213 212 216 220 57

218 214 9 8 221 219 215

219 216 215 218 221 222 220

220 217 216 219 222 56 57

221 219 218 8 7 55 222

222 55 221 219 220 56 197

223 42 41 224 229 228 13

224 41 40 225 230 229 223

225 224 40 39 231 226 230

226 227 232 231 225 39 38

227 226 38 37 232 233

228 223 13 133 14 234 229

229 223 224 230 235 234 228

230 224 225 231 236 235 229

231 225 226 232 237 236 230

232 227 238 237 231 226 233

233 37 61 62 238 232 227

234 14 15 239 235 229 228

235 229 234 239 240 236 230

236 230 235 240 241 237 231

237 231 236 241 242 238 232

238 233 62 63 242 237 232

239 15 16 243 240 235 234

240 235 239 243 244 241 236

241 236 240 244 245 242 237

242 245 64 63 238 237 241

243 16 17 246 244 240 239

244 243 246 247 245 241 240

245 244 247 65 64 242 241

246 17 18 66 247 244 243

247 246 66 272 65 245 244

248 24 31 32 249 254 253

249 248 32 33 250 255 254

250 249 33 34 251 256 255

251 250 34 35 252 257 256

252 36 177 35 251 257 258

253 158 24 248 254 259 23

254 253 248 249 255 260 259

255 254 249 250 256 260 261

256 255 250 251 257 262 261

257 256 251 252 258 263 262

258 257 252 263 36 61 62

259 23 253 254 260 264 22

260 259 254 255 261 265 264

261 260 255 256 262 266 265

262 261 256 257 263 267 266

263 62 63 267 262 257 258

264 21 22 259 260 265 268

265 264 260 261 266 269 268

266 265 261 262 267 270 269

267 270 266 262 263 63 64

268 20 21 264 265 269 271

269 268 265 266 270 272 271

270 269 266 267 64 65 272

271 20 268 269 272 66 19

272 271 269 270 65 247 66

Appendix D. Matlab code

```
function [newhrtf] = interp_nn(az,el,numpoints,tol,spkrmtrx,hrtfmtrx)

% function [newhrtf] = interp_nn(az,el,numpoints,spkrmtrx,hrtfmtrx)
%
% interpolates the closest three measured HRTF's on the AAMRL ALS to a new
% HRTF at a specified location, using one of many methods
%
% az:       the azimuth for the new, interpolated HRTF
% el:       the elevation for the new, interpolated HRTF
%
% 2Lt. Damion Reinhardt

if ~exist('tol');tol=1;end
if ~exist('spkrmtrx');load spkrmtrx;end
if ~exist('hrtfmtrx');load hrtfmtrx;end

nnspkrmtrx = nnpoints(az,el,numpoints);

if nnspkrmtrx(2,1)<tol;newhrtf = hrtfmtrx(nnspkrmtrx(1,1),:);return;end

for row=1:numpoints
  spkrlocs(row,:) = spkrmtrx(nnspkrmtrx(1,row),1:3);
  hrtfs(row,:) = hrtfmtrx(nnspkrmtrx(1,row),:);
end

for point=1:numpoints
  scalevec(point) = 1/diffangle(az,el,spkrlocs(point,2),spkrlocs(point,3));
end
```

```
scalevec = scalevec./sum(scalevec) % normalizing vector of scaling coeffs.

newhrtf = zeros(1,size(hrtfs,2)); % initializing the newhrtf at zero

% following loop adds in surrounding hrtfs using a weighting coefficients
% calculated above and stored in scalevec

for point=1:numpoints
  newhrtf = newhrtf + scalevec(point)*hrtfs(point,:);
end
```

```
function [newhrtf] = interp_tri(az,el,method,spkrmtrx,hrtfmtrx)

% function [newhrtf] = interp_tri(az,el,method,spkrmtrx,hrtfmtrx)
%
% interpolates the closest three measured HRTF's on the AAMRL ALS to a new
% HRTF at a specified location, using one of many methods
%
% az:       the azimuth for the new, interpolated HRTF
% el:       the elevation for the new, interpolated HRTF
% method:   the method used for interpolation, 'pl' is default
%           'pl' piecewise linear
%           'wa' weighted average
%
%
% 2Lt. Damion Reinhardt

if ~exist('method'),   method = 'wa'; end
if ~exist('spkrmtrx'), load spkrmtrx; end
%if ~exist('hrtfmtrx'), load hrtfmtrx; end

trispkrmtrx = whichtri(az,el);

if size(trispkrmtrx,2)==1,
  newhrtf=hrtfmtrx(trispkrmtrx(1,1),:);
  disp('exact location match induced bad tri in interp_tri');
  disp('returning exact value of hrtf at match point');
  return
```

```
end


for row=1:3
  spkrlocs(row,:) = spkrmtrx(trispkrmtrx(1,row),1:3);
  hrtfs(row,:) = hrtfmtrx(trispkrmtrx(1,row),:);
end

if (method=='pl')
  newhrtf = tripiecelin(hrtfs,spkrlocs,az,el);
elseif (method=='wa')
  newhrtf = triweightavg(hrtfs,spkrlocs,az,el);
elseif (method=='fftwa')
  newhrtf = fftweightavg(trihrtfmtrx,trilocmtrx,newloc)
else
  newhrtf = 'error - invalid method specified';
  return
end
```

```
function [newhrtf] = triweightavg(hrtfs,spkrlocs,az,el)

% function [newhrtf] = triweightavg(hrtfs,spkrlocs,newloc)
%
% interpolates three HRTF's at the vertices of a triangle on the ALS to a
% new HRTF at a specified location
%
% thrtfmtrx: a 3x272 row matrix of HRTF data; each row is an HRTF
%              consisting of dB gains across an array of measured frequencies
% spkrlocs:  a row matrix of the location of each HRTF in hrtfmtrx, expressed
%              in degrees azimuth and elevation, in respective columns
% az:
% el:
%
% 2Lt. Damion Reinhardt

numhrtfs = size(hrtfs,1);
numlocs = size(spkrlocs,1);

if ((numlocs==3)&(numhrtfs==3))

  dAB = diffangle(spkrlocs(1,2),spkrlocs(1,3),spkrlocs(2,2),spkrlocs(2,3));
  dBC = diffangle(spkrlocs(3,2),spkrlocs(3,3),spkrlocs(2,2),spkrlocs(2,3));
  dAC = diffangle(spkrlocs(1,2),spkrlocs(1,3),spkrlocs(3,2),spkrlocs(3,3));
  dAx = diffangle(az,el,spkrlocs(1,2),spkrlocs(1,3));
  dBx = diffangle(az,el,spkrlocs(2,2),spkrlocs(2,3));
  dCx = diffangle(az,el,spkrlocs(3,2),spkrlocs(3,3));
```

```
  scalemtrx(1) = (dBx+dCx-dBC)/(dAB+dAC-dBC);
  scalemtrx(2) = (dAx+dCx-dAC)/(dAB+dBC-dAC);
  scalemtrx(3) = (dAx+dBx-dAB)/(dAC+dBC-dAB);
  scalemtrx = scalemtrx./sum(scalemtrx)

  newhrtf = zeros(1,size(hrtfs,2));

  for point=1:3
    newhrtf = newhrtf + scalemtrx(point)*hrtfs(point,:);
  end

else newhrtf='error - the number of locations and hrtfs given do not match!'

end
```

```
function [newhrtf] = tripiecelin(hrtfs,spkrlocs,az,el)

% function [newhrtf] = triweightavg(hrtfs,spkrlocs,newloc)
%
% interpolates three HRTF's at the vertices of a triangle on the ALS to a
% new HRTF at a specified location
%
% hrtfmtrx: a 3x272 row matrix of HRTF data; each row is an HRTF
%              consisting of dB gains across an array of measured frequencies
% spkrlocs:  a 3x3 row matrix consisting of the speaker designation number,
%            azimuth, and elevation in each row.
%
% 2Lt. Damion Reinhardt

numhrtfs = size(hrtfs,1);
numlocs = size(spkrlocs,1);
if ~((numlocs==3)&(numhrtfs==3)),return,end

for freqnum=1:104
  tempmtrx = rref([spkrlocs(:,2:3),ones(3,1),hrtfs(:,freqnum)]);
  coeffvec = tempmtrx(:,4);
  newhrtf(freqnum) = [az el 1]*coeffvec;
end
```

```
function [tessmtrx] = sph2caps(numcaps)

% function [tessmtrx] = sph2caps(numcaps)
%
% parses the 272 point ALS into several caps

% 2Lt. Damion Reinhardt

clf;plotspkrs(1);hold on;
load spkrmtrx;

if ~exist('numcaps')

  % sets the six cap centers to the xyz axes, using azimuth and elevation
  % sets the next eight cap centers to those axes between the above xyz
  %
  %         [  az   el ]
  locmtrx = [   0    0 ;
              90    0 ;
      180    0 ;
      270    0 ;
        0  -90 ;
                0   90 ;
       45   45 ;
      135   45 ;
      225   45 ;
      315   45 ;
```

```
      45  -45 ;
     135  -45 ;
     225  -45 ;
     315  -45 ];

  % converts cap centers to cartesian coordinates
  for capnum=1:size(locmtrx,1)
    az = locmtrx(capnum,1);el = locmtrx(capnum,2);rho=1;
    [x,y,z] = az_el_deg2cart(az,el,rho);
    cartmtrx(capnum,1:3) = [x,y,z];
  end

else
  cartmtrx=repulsion2(numcaps);

  for i=1:size(cartmtrx,1)
    x=cartmtrx(i,1);y=cartmtrx(i,2);z=cartmtrx(i,3);
    [az,el,rho] = cart2azeldeg(x,y,z);
    locmtrx(i,:)=[az,el,rho];
  end

  cartmtrx
  locmtrx

end


if ~exist('numcaps')
```

```
  % plots the cap centers

  for capnum=1:6
    plot3([0 cartmtrx(capnum,1)],[0 cartmtrx(capnum,2)],[0
    cartmtrx(capnum,3)],'b'); %drawnow;
  end

  for capnum=7:14
    plot3([0 cartmtrx(capnum,1)],[0 cartmtrx(capnum,2)],[0
    cartmtrx(capnum,3)],'r'); %drawnow;
  end

end


% finds the caps by searching for nearest neighbors to the cap centers

tessmtrx = zeros(numcaps,30);

for spkr=1:272
  for capnum=1:size(cartmtrx,1)
    az_spkr = spkrmtrx(spkr,2);  el_spkr = spkrmtrx(spkr,3);
    az_cap = locmtrx(capnum,1);el_cap = locmtrx(capnum,2);rho=1;
    diffmtrx(capnum)=diffangle(az_cap,el_cap,az_spkr,el_spkr);
  end
  [sortdiffs,index] = sort(diffmtrx);wincap=index(1);
  tessmtrx(wincap,[size(find(tessmtrx(wincap,:)),2)+1])=spkr;
```

```
end
```

```
function [circzspkrs2] = sph2circs(numcircs,maxaz)

% sph2circs.m
%
% divides sphere into constant az cone bands
%

% 2Lt. Damion Reinhardt

if ~exist('spkrmtrx'),load spkrmtrx;end

% numcircs should be odd for sym
%maxaz = 75;numcircs=11;  % 70,9 ; 70,11 ;  80, 13 too
az = [-maxaz:(maxaz*2)/(numcircs+1):maxaz];
coneaz = az(2:size(az,2)-1);
circzspkrs2=zeros(size(coneaz,2),35);

for spkr=1:272
  for circnum = 1:size(coneaz,2)
    distmtrx(circnum) = spkrdist2coneplane(spkr,coneaz(circnum),spkrmtrx);
  end
  [sortdists,index] = sort(distmtrx);row=index(1);
  circzspkrs2(row,1+size(circzspkrs2(find(circzspkrs2(row,:))),2)) = spkr;
end

clf;plotspkrs(1,'w.','w');view(0,0);
for circnum = 1:size(coneaz,2)
  if mod(circnum,2)
```

```
      plotspkrs(circzspkrs2(circnum,:),'rx','n')
    else
      plotspkrs(circzspkrs2(circnum,:),'bo','n')
    end
  end
```

```
% repulsion.m
%
% creates spaced cap centers on the sphere
%
% 2Lt. Damion Reinhardt


clear;
numvecs = 6 ;

clf;plotspkrs(1);hold on;

for i=1:numvecs
  temp=rand(1,3);
  vecmtrx(i,1:3)=sqrt(temp/sum(temp));
  plot3([0,vecmtrx(i,1)],[0,vecmtrx(i,2)],[0,vecmtrx(i,3)]);
end

% take vector sum of repulsions and move each vector

for i=1:20
  vecorder=randperm(numvecs);
  for testvecnum=1:numvecs
    testvec = vecorder(testvecnum);
    othervecnums = vecorder(find([vecorder]~=testvec));
    othervecsum = zeros(1,3);
    for j=1:size(othervecnums,2)
      othervecsum =  othervecsum + [ vecmtrx(othervecnums(j),:) / ...
```

```
  diffangle_cartvecs(vecmtrx(testvec,:), ...
  vecmtrx(othervecnums(j),:))^2 ];

    end
    temp2 =  vecmtrx(testvec,:) - othervecsum/sum(abs(othervecsum));
    temp2 = temp2 / sum(abs(temp2));
    vecmtrx(testvec,:) =  [ ...
  sqrt(abs(temp2(1)))*sign(temp2(1)), ...
  sqrt(abs(temp2(2)))*sign(temp2(2)) , ...
  sqrt(abs(temp2(3)))*sign(temp2(3)) ];
  end

  clf;plot3(0,0,0);hold on;
  for i=1:numvecs
    plot3([0,vecmtrx(i,1)],[0,vecmtrx(i,2)],[0,vecmtrx(i,3)]);
  end
  drawnow;
end


%ack=vecmtrx(4,:);
%[az,el,rho] = cart2azeldeg(ack(1),ack(2),ack(3));
%view(az,el);

diffangle_cartvecs(vecmtrx(1,:),vecmtrx(2,:))
diffangle_cartvecs(vecmtrx(1,:),vecmtrx(3,:))
diffangle_cartvecs(vecmtrx(1,:),vecmtrx(4,:))
diffangle_cartvecs(vecmtrx(1,:),vecmtrx(5,:))
```

```
diffangle_cartvecs(vecmtrx(1,:),vecmtrx(6,:))
```

```
function [w1,b1,w2,b2,k,tr] = solverb(p,t,spkrs,unqstr,unqnum,dp,dp2)

% function [w1,b1,w2,b2,k,tr] = solverb(p,t,spkrs,unqstr,unqnum,dp,dp2)
%
% [W1,B1,W2,B2,TE,TR] = SOLVERB(P,T,DP)
%   P - RxQ matrix of Q input vectors.
%   T - SxQ matrix of Q target vectors.
%   DP - Design parameters (optional).
% Returns:
%   W1 - S1xR weight matrix for radial basis layer.
%   B1 - S1x1 bias vector for radial basis layer.
%   W2 - S2xS1 weight matrix for linear layer.
%   B2 - S2x1 bias vector for linear layer.
%   NR - the number of radial basis neurons used.
%   TR - training record: [row of errors]
%
% Design parameters are:
%   TP(1) - Iterations between updating display, default = 25.
%   TP(2) - Maximum number of neurons, default = # vectors in P.
%   TP(3) - Sum-squared error goal, default = 0.02.
%   TP(4) - Spread of radial basis functions, default = 1.0.
%
% Design parameters are:
%   TP2(1) - Iterations between updating line, default = 25.
%   TP2(2) - Circle number
%   TP2(3) - Number of nodes
```

```
%
%
% Missing parameters and NaN's are replaced with defaults.
%
% See also NNSOLVE, RADBASIS, SIMRB, SOLVERB.

% Mark Beale, 12-15-93
% Copyright (c) 1992-97 by The MathWorks, Inc.
%       $Revision: 1.3 $

if nargin < 5, error('Not enough input arguments'),end
homedir = '/home/trapper4/98m/dreinhar/thesis/';
datadir = 'matlab/anns/runs/';

% TRAINING PARAMETERS
if nargin == 5, dp = []; end
[r,q] = size(p);
dp = nndef(dp,[25 20 0.02 2.5e4/20]);
df = dp(1);
eg = dp(3);
b = sqrt(-log(.5))/dp(4);
[s2,q] = size(t);
mn = min(q,dp(2));


% MORE TRAINING PARAMETERS
if nargin < 7, dp2 = []; end
dp2 = nndef(dp2,[1 1 20 mn-1 10]);
```

```
dfl = dp2(1);
numnodes = dp2(3);
me = dp2(4);
%dfs = dp2(5);
dfs = mn-1;

% RADIAL BASIS LAYER OUDPUTS
P = radbas(dist(p',p)*b);
PP = sum(P.*P)';
d = t';
dd = sum(d.*d)';

% CALCULATE "ERRORS" ASSOCIATED WITH VECTORS
e = ((P' * d)' .^ 2) ./ (dd * PP');

% PICK VECTOR WITH MOST "ERROR"
pick = nnfmc(e);
used = [];
left = 1:q;
W = P(:,pick);
P(:,pick) = []; PP(pick,:) = [];
e(:,pick) = [];
used = [used left(pick)];
left(pick) = [];

% CALCULATE ACTUAL ERROR
w1 = p(:,used)';
a1 = radbas(dist(w1,p)*b);
```

```
[w2,b2] = solvelin(a1,t);
a2 = purelin(w2*a1,b2);
sse = sumsqr(t-a2);

% TRAINING RECORD
tr = zeros(1,mn);
tr(1) = sse;

% PLOTTING

% TRAINING
for k = 1:mn-1

  % CHECK ERROR
  if (sse < eg), break, end

  % CALCULATE "ERRORS" ASSOCIATED WITH VECTORS

  wj = W(:,k);

  %---- VECTOR CALCULATION

  a = wj' * P / (wj'*wj);
  P = P - wj * a;
  PP = sum(P.*P)';
%if any(any(PP == 0))
%  disp('PP has a 0')
%  keyboard
```

```
%end
  e = ((P' * d)' .^ 2) ./ (dd * PP');

  % PICK VECTOR WITH MOST "ERROR"
  pick = nnfmc(e);
  W = [W, P(:,pick)];
  P(:,pick) = []; PP(pick,:) = [];
  e(:,pick) = [];
  used = [used left(pick)];
  left(pick) = [];

  % CALCULATE ACTUAL ERROR
  w1 = p(:,used)';
  a1 = radbas(dist(w1,p)*b);
  [w2,b2] = solvelin(a1,t);
  a2 = purelin(w2*a1,b2);
  sse = sumsqr(t-a2);

  % TRAINING RECORD
  tr(k+1) = sse;

  % PLOTTING
  if rem(k,df) == 0
    disp('oop')
    k
    plot(p(3,1:104),t(1:104),'+k',p(3,1:104),a2(1:104),'r-');
    drawnow;
  end
```

```
end

[S1,R] = size(w1);
b1 = ones(S1,1)*b;

% TRAINING RECORD
tr = tr(1:(k+1));

% SAVING (added by DR)
eval(['save
',homedir,datadir,'hrtfrb_',unqstr,int2str(unqnum),'_',date,'_numnodes',
int2str(numnodes),'_epochs',int2str(k),' b1 w1 b2 w2 p t spkrs']);


% PLOTTING
  plot(p(3,1:104),t(1:104),'+k',p(3,1:104),a2(1:104),'r-');

% WARNINGS
if sse > eg
  disp(' ')
  disp('SOLVERB: Network error did not reach the error goal.')
  disp('  More neurons may be necessary, or try using a')
  disp('  wider or narrower spread constant.')
  disp(' ')
end
```

```
function [] = rbfhrtf(spkrs,unqstr,unqnum,hrtfmtrx)

% function [] = rbfhrtf(spkrs,unqstr,unqnum,hrtfmtrx)
%
% trains a radial basis function on the ALS HRTF data
%
% spkrs: a vector of speaker locations at which the rbfnet is to be trained
% rotation: azimuthal rotation
% freqsvec       1x104          832  double array
% hrtfmtrx     272x104       226304  double array
% spkrmtrx     272x4           8704  double array
%
%
% numnodes: number of hidden nodes - not used!
% numit: max number of iterations  - not used!

% 2Lt. Damion Reinhardt

if ~exist('unqstr'),unqstr = 'test';end;
if ~exist('unqnum'),unqnum = 42;end;
if ~exist('freqsvec'),load freqsvec;end;
if ~exist('spkrmtrx'),load spkrmtrx;end;
if ~exist('hrtfmtrx'),load hrtfmtrx;end;

% if ~exist('anntestvars')
%  load anntestvars;
% else eval(['load ',anntestvars]);
```

```
% end

% admin junk
echo off;
clf;

% loads hrtfs for the spkrs in the net

for spkrnum=1:size(spkrs,2)
  spkr = spkrs(spkrnum);
  spkrazel = spkrmtrx(spkr,2:3);
  for freqnum = 1:104
    rownum = freqnum+104*(spkrnum-1);
    spkrs_azelfreqmtrx(rownum,1:2) = spkrazel;
    spkrs_azelfreqmtrx(rownum,3) = freqsvec(freqnum); % not freqs 2kHz
  end
end

for spkrnum=1:size(spkrs,2)
  spkr = spkrs(spkrnum);
  for freqnum = 1:104
    rownum = freqnum+104*(spkrnum-1);
    spkrs_gainvec(rownum,1) = hrtfmtrx(spkr,freqnum);
  end
end

numpoints = size(spkrs_azelfreqmtrx,1);
```

```
p = spkrs_azelfreqmtrx(1:numpoints,1:3)';              % training data
t = spkrs_gainvec(1:numpoints,:)';                     % target values

% uncomment for the 'solverb' routine
[w1,b1,w2,b2] = solverb_hack(p,t,spkrs,unqstr,unqnum);

% uncomment for the 'solverbe' routine
%numnodes = size(p,2)
%z = (2.5e4)/numnodes
%[w1,b1,w2,b2] = solverbe(p,t,3);
%plot(p,t,'+');
```

```
% bp2hrtf.m
%
% trains a perceptron on the ALS HRTF data
% based loosely on DEMOP7 Classification with a two-layer perceptron
%
% spkrs: a vector of speaker locations at which the perceptron is to be
% trained
%
% 2Lt. Damion Reinhardt

disp('ack');
load freqsvec;load spkrmtrx;
load newhrtf_zero_right;hrtfmtrx=newhrtf_zero_right;
if ~exist('unqnum'),unqnum=69;end
if ~exist('unqstr'),unqstr='test';end
echo off;
clc;

%    extracting the data and defining the mlp learning problem
%    ---------------------------------------------------------

%    A matrix 'train' defines the input (column) vectors:
%    A matrix 'target' defines the categories with target (column) vectors.
%    A matrix 'testdata' defines the data against which to test the mlp.

for spkrnum=1:size(spkrs,2)
  spkr = spkrs(spkrnum);
  spkrazel = spkrmtrx(spkr,2:3);
```

```
  for freqnum = 1:104
    rownum = freqnum+104*(spkrnum-1);
    spkrs_azelfreqmtrx(rownum,1:2) = spkrazel;
    spkrs_azelfreqmtrx(rownum,3) = freqsvec(freqnum)/1000; % no freqs2kHz
  end
end

for spkrnum=1:size(spkrs,2)
  spkr = spkrs(spkrnum);
  for freqnum = 1:104
    rownum = freqnum+104*(spkrnum-1);
    spkrs_gainvec(rownum,1) = hrtfmtrx(spkr,freqnum);
  end
end

p = spkrs_azelfreqmtrx';                % training data
%p=p(3,:);                              % uncomment for unv freqs only fit
t = spkrs_gainvec';                     % target values
pinit(:,1) = min(p')';
pinit(:,2) = max(p')';

%    creating the perceptron
%    -----------------------

%    how many layers? how many neurons?  let's try 30?  and what to use for
%    the input and output functions?  hmm...

if ~exist('S1')
```

```
  S1 = 50;
end

if ~exist('S2')
  S2 = 50;
end

if ~exist('f1')
  f1 = 'tansig';
end

if ~exist('f2')
  f2 = 'tansig';
end

if ~exist('f3')
  f3 = 'purelin';
end

%    initff generates initial weights and biases for a feed forward network:
%    initff usage: [W1,B1,...] = INITFF(p,S1,'F1',...,Sn,'Fn')

if ~exist('w1')
  [w1,b1,w2,b2,w3,b3] = initff(pinit,S1,f1,S2,f2,t,f3);
end

%    training the network
%    ----------------------
```

```
%    TRAINBP trains a feed forward network using backpropagation

% Training parameters are:
%   TP(1) - Epochs between updating display, default = 25.
%   TP(2) - Maximum number of epochs to train, default = 1000.
%   TP(3) - Sum-squared error goal, default = 0.02.
%   TP(4) - Learning rate, 0.01.
%   TP(5) - Learning rate increase, default = 1.05.
%   TP(6) - Learning rate decrease, default = 0.7.
%   TP(7) - Momentum constant, default = 0.9.
%   TP(8) - Maximum error ratio, default = 1.04.

df = 10;
if ~exist('me'),me = 1000000;end
eg = 4000;
%grad_min = .001;
mu_init = 10000;
mu_inc = 100;
mu_dec = 1/mu_inc;

tp = [df me eg mu_init mu_inc mu_dec];
tp2 = [100];

%    training begins...please wait (this takes a while!)...
%    usage: [W1,B1,W2,B2,TE,TR] = TLM2(W1,B1,'F1',W2,B2,'F2',p,t)

%[w1,b1,w2,b2,w3,b3,te,tr] = ...
```

```
    %tbpx3_hack(w1,b1,f1,w2,b2,f2,w3,b3,f3,p,t,tpbpx,tpbpx2,spkrs);

% function [w1,b1,w2,b2,w3,b3,i,tr]=tbpx3(w1,b1,f1,w2,b2,f2,
% w3,b3,f3,p,t,tp,tp2,spkrs)
%
% [W1,B1,W2,B2,W3,B3,TE,TR] = TBPX3(W1,B2,F1,W1,B1,F2,W3,B3,F3,P,T,TP)
%   Wi - Weight matrix for the ith layer.
%   Bi - Bias vector for the ith layer.
%   Fi - Transfer function (string) for the ith layer.
%   P  - RxQ matrix of input vectors.
%   T  - SxQ matrix of target vectors.
%   TP - Training parameters (optional).
% Returns:
%   Wi - new weights.
%   Bi - new biases.
%   TE - the actual number of epochs trained.
%   TR - training record: [row of errors]
%
% Training parameters are:
%   TP(1) - Epochs between updating display, default = 25.
%   TP(2) - Maximum number of epochs to train, default = 1000.
%   TP(3) - Sum-squared error goal, default = 0.02.
%   TP(4) - Learning rate, 0.01.
%   TP(5) - Learning rate increase, default = 1.05.
%   TP(6) - Learning rate decrease, default = 0.7.
%   TP(7) - Momentum constant, default = 0.9.
%   TP(8) - Maximum error ratio, default = 1.04.
% Missing parameters and NaN's are replaced with defaults.
```

```
% Mark Beale, 1-31-92
% Revised 12-15-93, MB
% Copyright (c) 1992-97 by The MathWorks, Inc.
%       $Revision: 1.3 $

%if nargin < 11,error('Not enough arguments.');end

% TRAINING PARAMETERS
%if nargin == 11, tp = []; end
%if ~exist('tp'), tp = []; end
tp = nndef(tp,[25 1000 0.02 0.01 1.05 0.7 0.9 1.04]);
df = tp(1);
me = tp(2);
eg = tp(3);
lr = tp(4);
im = tp(5);
dm = tp(6);
mc = tp(7);
er = tp(8);
df1 = feval(f1,'delta');
df2 = feval(f2,'delta');
df3 = feval(f3,'delta');

dw1 = w1*0;
db1 = b1*0;
dw2 = w2*0;
db2 = b2*0;
```

```
dw3 = w3*0;
db3 = b3*0;
MC = 0;

% MORE PARAMETERS added by DR
%if ~exist('tp2'), tp2 = []; end
tp2 = nndef(tp2,[100]);
dfs = tp2(1);              % how often to save vars
%testspkr= spkrs(1);

% PRESENTATION PHASE
a1 = feval(f1,w1*p,b1);
a2 = feval(f2,w2*a1,b2);
a3 = feval(f3,w3*a2,b3);
e = t-a3;
SSE = sumsqr(e);

% TRAINING RECORD
tr = zeros(2,me+1);
tr(1:2,1) = [SSE; lr];

% PLOTTING FLAG
[r,q] = size(p);
[s,q] = size(t);
plottype = (max(r,s) == 1) & 0;

% PLOTTING
%newplot;
```

```
message = sprintf('TRAINBPX: %%g/%g epochs, lr = %%g, SSE = %%g.\n',me);
fprintf(message,0,lr,SSE)
%if plottype
  %h = plotfa(p,t,p,a3);
%else
  %h = plottr(tr(1:2,1),eg);
%end

% BACKPROPAGATION PHASE
d3 = feval(df3,a3,e);
d2 = feval(df2,a2,d3,w3);
d1 = feval(df1,a1,d2,w2);

for i=1:me

  % CHECK PHASE
  if SSE < eg, i=i-1; break, end

  % LEARNING PHASE
  [dw1,db1] = learnbpm(p,d1,lr,MC,dw1,db1);
  [dw2,db2] = learnbpm(a1,d2,lr,MC,dw2,db2);
  [dw3,db3] = learnbpm(a2,d3,lr,MC,dw3,db3);
  MC = mc;
  new_w1 = w1 + dw1; new_b1 = b1 + db1;
  new_w2 = w2 + dw2; new_b2 = b2 + db2;
  new_w3 = w3 + dw3; new_b3 = b3 + db3;

  % PRESENTATION PHASE
```

```
new_a1 = feval(f1,new_w1*p,new_b1);
new_a2 = feval(f2,new_w2*new_a1,new_b2);
new_a3 = feval(f3,new_w3*new_a2,new_b3);
new_e = t-new_a3;
new_SSE = sumsqr(new_e);

% MOMENTUM & ADAPTIVE LEARNING RATE PHASE
if new_SSE > SSE*er
  lr = lr * dm;
  MC = 0;
else
  if new_SSE < SSE
    lr = lr * im;
    end
  w1 = new_w1; b1 = new_b1; a1 = new_a1;
  w2 = new_w2; b2 = new_b2; a2 = new_a2;
  w3 = new_w3; b3 = new_b3; a3 = new_a3;
  e = new_e; SSE = new_SSE;

  % BACKPROPAGATION PHASE
  d3 = feval(df3,a3,e);
  d2 = feval(df2,a2,d3,w3);
  d1 = feval(df1,a1,d2,w2);
end

% TRAINING RECORD
tr(1:2,i+1) = [SSE; lr];
```

```
  % PLOTTING
  if rem(i,df) == 0
    fprintf(message,i,lr,SSE)
    %if plottype
      %delete(h);
      %h = plot(p,a3);
    %else
      %h = plottr(tr(1:2,1:(i+1)),eg,h);
    %end
  end

  % SAVING added by DR
  if rem(i,dfs) == 0
    eval(['save runs/bpx_',unqstr,int2str(unqnum),'_',date,'_S1_',int2str(S1),
    '_S2_',int2str(S2),' spkrs f1 f2 f3 b1 b2 b3 w1 w2 w3;']);
  end


end

% TRAINING RECORD
tr = tr(1:2,1:(i+1));

% PLOTTING
if rem(i,df) ~= 0
  fprintf(message,i,lr,SSE)
  if plottype
    delete(h);
```

```
    plot(p,a3);
  else
    plottr(tr,eg,h);
  end
end

% HRTF PLOTTING added by DR
figure;
comp_spkr_bp2;



% WARNINGS
if SSE > eg
  disp(' ')
  disp('TRAINBPX: Network error did not reach the error goal.')
  disp('  Further training may be necessary, or try different')
  disp('  initial weights and biases and/or more hidden neurons.')
  disp(' ')
end


%    ...and finishes.

%    PLOTTING THE ERROR CURVE
%    ========================

%    Here the errors are plotted with respect to training epochs:
```

```
%ploterr(tr);

%     If the hidden (first) layer prprocessed the origonal
%     non-linearly seperable input vectors into new linearly
%     seperable vectors, then the perceptron will have 0 error.

%     If the error never reached 0, it means a new preprocessing
%     layer should be created (perhaps with more neurons).  I.e.
%     try running this script again.

%
% echo on
% pause % Strike any key to use the classifier to find confusion matrix...
% clc
% [confmatrix,errorrate] = confmtrx(testdata,p,t,w1,b1,w2,b2,input,output)
% disp('End of DEMOP7')
```

Bibliography

1. Akanasu, A.N. and F.E. Wadas. "On Lapped Orthogonal Transforms," *IEEE Transactions on Signal Processing*, *40*:439–443 (February 1992).
2. Batteau, D.W. "The Role of the Pinna in Human Localization," *Proceedings of the Royal Society of London B*, *168*:158–180 (1967).
3. Batteau, D.W. *Listening with the Naked Ear: The Neuropsychology of Spatially Oriented Behavior*. Homewood, IL: Dorsey, 1968.
4. Bidlack, Rick, "Virtual Sonic Space." Freeware, published via internet at ftp://ftp.accessone.com/pub/misc/release/.
5. Bishop, Christopher M. *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press, 1995.
6. Blauert, Jens. *Spatial Hearing*. The MIT Press, 1983.
7. Broomhead, D.S. and D. Lowe. "Multivariable Functional Interpolation and Adaptive Networks," *Complex Systems* (1988).
8. Butler, R.A. and K. Belendiuk. "Spectral Cues Used in the Localization of Sound in the Median Sagittal Plane," *Journal of the Acoustical Society of America*, *61*:1264–1269 (1977).
9. Chen, S., et al. "Orthogonal Least Squares Learning Algorithm for Radial Basis Functions," *IEEE Transactions on Neural Networks*, *2*(2):302–309 (March 1991).
10. Chen, Jiashu. *Auditory Space Modeling and Virtual Auditory Environment Simulation*. Ph. D. thesis, University of Wisconsin, Madison, 1992.
11. Chen, Jiashu, et al. "External Ear Transfer Function Modeling: A Beamforming Approach," *Journal of the Acoustical Society of America*, *92*(4):1933–1944 (October 1992).
12. Chen, Jiashu, et al. "A Spatial Feature Extraction and Regularization Model for the Head Related Transfer Function," *Journal of the Acoustical Society of America*, *97*(1):439–452 (January 1995).
13. Cybenko, G. "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals, and Systems* (1989).
14. Demuth, Howard G. and Mark Beale. *Neural Network Toolbox User's Guide*. The Mathworks, Inc., 1994.
15. Duda, Richard O. "Elevation Dependence of the Interaural Transfer Function." *Binaural and Spatial Hearing in Real and Virtual Environments* edited by Robert H. Gilkey and Timothy A. Anderson, chapter 3, 1–23, Wright-Patterson AFB, OH: Lawrence Erlbaum Associates, 1997.

16. Fechner, G.T. *Elemente der Psychophysics.* Leipzig: Breitkopk und Hartel, 1860.

17. Gardner, William G. *Transaural 3-D Audio.* Technical Report 342, 20 Ames Street, Room E15-401B, Cambridge MA 02139: MIT Media Lab, July 1995.

18. Genuit, Klaus. "A Description of the Human Outer Ear Transfer Function by Elements of Communication Theory." *Proceedings of the 12th International Congress on Acoustics.* B6–8. 1986.

19. Henning, G.B. "Detectability of Interaural Delay in High-Frequency Complex Waveforms," *Journal of the Acoustical Society of America, 55*:84–90 (1974).

20. Jot, J.M., V. Larcher and O. Warusfel. "Digital Signal Processing Issues in the Context of Binaural and Transaural Stereophony," *Proceedings of the Audio Engineering Socitey* (1997).

21. Kistler, D.J. and Frederic L. Wightman. "A Model of Head-Related Transfer Functions Based of Principal Components Analysis and Minimum Phase Reconstruction," *Journal of the Acoustical Society of America, 91*:1637–1647 (March 1992).

22. Lapedes, A. and R. Farber. "How Neural Nets Work." *Neural Information Processing Systems* American Institute of Physics, 1988.

23. Malvar, H.S. and D.H. Staelin. "The LOT: Transform Coding Without Blocking Effects," *IEEE Transactions on Acoustics, Speech, and Signal Processing, 37*:553–559 (April 1989).

24. McKinley, Richard L. *Concept and Design of an Auditory Localization Cue Synthesizer.* MS thesis, AFIT/GE/ENG/88D-29, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, 1988.

25. McKinley, Richard L. and Mark A. Ericson. "Flight Demonstration of a 3-D Auditory Display." *Binaural and Spatial Hearing in Real and Virtual Environments* edited by Robert H. Gilkey and Timothy A. Anderson, chapter 31, 683–699, Wright-Patterson AFB, OH: Lawrence Erlbaum Associates, 1997.

26. Middlebrooks, John C. "Spectral Shape Cues for Sound Localization." *Binaural and Spatial Hearing in Real and Virtual Environments* edited by Robert H. Gilkey and Timothy A. Anderson, chapter 4, 683–699, Wright-Patterson AFB, OH: Lawrence Erlbaum Associates, 1997.

27. Millhouse, John K. *Head Related Transfer Function Approximation Using Neural Networks.* MS thesis, AFIT/GE/ENG/94D-21, School of Engineering, Air Force Institiute of Technology (AU), Wright Patterson AFB, OH, December 1994.

28. Moody, J. and C.J. Darken. "Fast Learning in Networks of Locally-Tuned Processing Units," *Neural Computation*, 281–294 (1989).

29. Perrott, D.R., "Auditory Psychomotor Coordination." Presentation Paper at the Sound Localization by Human Observers Symposium.

30. Plenge, G. "On the Difference Between Localization and Lateralization," *Journal of the Acoustical Society of America*, *56*:944–951 (1974).

31. Powell, M.J.D. "Radial Basis Functions for Multivariable Interpolation: A Review." *Algorithms for Approximation* Clarendon Press, 1987.

32. Raleigh, Lord (J.W. Strutt, 3rd Baron of Raleigh). "On Our Percepion of Sound Direction," *Philosophy Magazine*, *13*:214–232 (1907).

33. Rogers, Steven K., et al. *An Introduction to Biological and Artificial Neural Networks*. Wright-Patterson AFB OH, 45433: Air force Institute of Technology (AU), October 1990.

34. Shaw, E.A.G. "Physical Models of the External Ear." *Proceedings of the 8th International Congress on Acoustics*. 206. 1974.

35. Shaw, Edgar A.G. "Acoustical Features of the Human External Ear." *Binaural and Spatial Hearing in Real and Virtual Environments* edited by Robert H. Gilkey and Timothy A. Anderson, chapter 2, 611–663, Wright-Patterson AFB, OH: Lawrence Erlbaum Associates, 1997.

36. Shinn-Cunningham, Barbara., et al. "Auditory Displays." *Binaural and Spatial Hearing in Real and Virtual Environments* edited by Robert H. Gilkey and Timothy A. Anderson, chapter 29, 611–663, Wright-Patterson AFB, OH: Lawrence Erlbaum Associates, 1997.

37. Smith, Brian A. *Binaural Room Simulation*. MS thesis, AFIT/GAM/ENG/93D-1, School of Engineering, Air Force Institiute of Technology (AU), Wright Patterson AFB, OH, December 1993.

38. Suter, Bruce W. and Mark E. Oxley. "On Variable Overlapped Windows and Weighted Orthogonal Transforms," *IEEE Transactions on Signal Processing* (1994).

39. Thurlow, W. R., et al. "Head Movements During Sound Localization," *Journal of the Acoustical Society of America*, *42*:489–493 (1967).

40. Thurlow, W. and P.S. Runge. "Effects of Induced Head Movements on Localization of Direction of Sound Sources," *Journal of the Acoustical Society of America*, *42*:480–488 (1967).

41. Webster, Douglas B. "The Evolutionary Biology of Hearing." *The Evolutionary Biology of Hearing* edited by Douglas B. Webster, Springer-Verlag, 1992.

42. Wightman, Frederic L., Doris J. Kistler and M. Arrua. "Perceptual Consequences of Engineering Compromises in Synthesis of Virtual Auditory Objects," *Journal of the Acoustical Society of America*, *92*:2332 (1992).

43. Wightman, Frederic L. and Doris J. Kistler. "Headphone Stimulation of Free-Field Listening II: Psychophysical Validation," *Journal of the Acoustical Society of America*, *85*:868–878 (1989).

44. Wightman, Frederic L. and Doris J. Kistler. "The Dominant Role of Low-Frequency Interaural Time Differences in Sound Localization," *Journal of the Acoustical Society of America*, *91*:1648–1661 (1992).

45. Wightman, Frederic L. and Doris J. Kistler. "Factors Affecting the Relative Salience of Sound Localization Cues." *Binaural and Spatial Hearing in Real and Virtual Environments* edited by Robert H. Gilkey and Timothy A. Anderson, chapter 1, 1–23, Wright-Patterson AFB, OH: Lawrence Erlbaum Associates, 1997.

46. Wightman, Frederic L. et al. "Reassesment of the Role of Head Movements in Human Sound Localization," *Journal of the Acoustical Society of America*, *95*:3003–3004 (1994).

Vita

Damion Reinhardt was born on June 21, 1974 in Blue Island, Illinois. He graduated from Edmond Memorial High School of Edmond, Oklahoma, and subsequently attended the U.S. Air Force Academy. He graduated in May 1996 with a Bachelor of Science degree in physics and mathematics. That summer, he entered the Masters Program in the Graduate School of Engineering, Air Force Instituted of Technology, Wright-Patterson AFB, OH.

He is married to Laura A. Reinhardt (Stewart), of Edmond, Oklahoma.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1998	Master's Thesis

4. TITLE AND SUBTITLE
NEURAL NETWORK MODELING OF THE HEAD-RELATED TRANSFER FUNCTION

5. FUNDING NUMBERS

6. AUTHOR(S)
Damion Reinhardt, 2Lt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GAM/ENC/98M-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFRL/HEC
Attn: Mr. Rich McKinley
Bldg. 441, 2610 Seventh St.
Wright-Patterson Air Force Base, OH 45433-7901

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Battlefield synthesis of 3-D audio may require the interpolation and compression of head-related transfer function (HRTF) data. This thesis is an implementation of a functional model of the HRTF using artificial neural networks (ANNs), the model provides both compression and interpolation.

14. SUBJECT TERMS
3-D sound, spatial audio, HRTF, Head-related transfer function, neural networks, tessellation, lapped orthonormal transforms, interpolation, MLP, RBF

15. NUMBER OF PAGES
116

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. 239.18
Designed using Perform Pro, WHS/DIOR, Oct 94