Robust Sound Localization: An Application of an Auditory Perception System for a Humanoid Robot

by

Robert Eiichi Irie

S.B., Harvard University (1993)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1995

Signature of Author . . . .
Department of Electrical Engineering and Computer Science
May 24, 1995

Certified by . . . .
Rodney A. Brooks
Professor, Department of Electrical Engineering & Computer Science
Thesis Supervisor

Accepted by . . . .
Frederic R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

Robust Sound Localization: An Application of an Auditory Perception System for a Humanoid Robot

by

Robert Eiichi Irie

Submitted to the Department of Electrical Engineering and Computer Science
on May 24, 1995, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Localizing sounds with different frequency and time domain characteristics in a dynamic listening environment is a challenging task that has not been explored in the field of robotics as much as other perceptual tasks. This thesis presents an integrated auditory system for a humanoid robot, currently under development, that will, among other things, learn to localize normal, everyday sounds in a realistic environment. The hardware and software has been designed and developed to take full advantage of the features and capabilities of the humanoid robot of which it will be an integral component. Sounds with different frequency components and time domain characteristics have to be localized using different cues; a neural network is also presented that has been developed off-line to learn to integrate the various auditory cues, using primarily visual data to perform self-supervised training.

Thesis Supervisor: Rodney A. Brooks
Title: Professor, Department of Electrical Engineering & Computer Science

Acknowledgments

I would like to thank Professor Rodney A. Brooks for giving me the opportunity and freedom to explore my areas of interest in robotics. He has been a constant source of inspiration and education.

Many thanks go to the other members of the Cog Group (Cindy, Matt M., Yoky, Scaz, and Matt W.) for their valued comments, helpful suggestions, and overall support.

I would also like to thank the Office of Naval Research for the financial independence that the ONR Fellowship has given me. At a time when research funding is scarce, having an external source of funding for three years is absolutely a dream for any graduate student.

Finally, my deepest appreciation goes to my family, my parents and sister, who have always been there for me.

This research was supported in part by the Office of Naval Research.

Preface

This thesis describes work that is part of a larger, ongoing project, and should be viewed in that context. In any major robotics undertaking, it is first necessary to build the hardware and low-level software components before exploring the more interesting aspects of artificial intelligence.

Contents

List of Figures

Chapter 1

Introduction

> *In no other field of science... does a stimulus produce so many different sensations as in the area of directional hearing.*[1]

1.1 Motivation

In the robotics and artificial intelligence fields, the most popular sensory modality to be incorporated in systems is vision; until very recently hearing has not played much of a role in the intelligent systems research. Few attempts have been made to incorporate sound processing in a self-contained robot.[2] However, sound provides a rich source of information: many animals rely on localization and other auditory perceptual tasks to survive; speech and hearing are the primary means of communication for human beings. In some ways audition on a robot is more subtle and difficult than vision. Unlike the eyes, the ears do not directly receive spatial information from the surroundings. The auditory system thus relies much more heavily on the processing of raw sensory data to extract acoustic cues and indirectly derive spatial information.

Despite these complexities, it is important to take advantage of the complementary nature of auditory and visual information to extract features and information from the surrounding environment that would be difficult or impossible from either modality alone (Gamble & Rainton 1994). To this end, it is crucial to have an integrated system that can tightly couple different sensory modalities, like audition and vision. Work presented in this thesis is part of a larger ongoing project, the Humanoid Robot Group at the MIT Artificial Intelligence Laboratory, that seeks to explore and take advantage of such tight couplings of sensors and motors to achieve human-like behavior.

This thesis presents a scalable, general-purpose auditory system for a humanoid robot that will be able to perform a wide variety of auditory perception tasks. The humanoid robot, Cog, approximates a human being from the waist up, with corresponding structures and sensors, such as video cameras for eyes, mechanical arms and hands, and a plastic shell for a skin. Modularity was a key design goal for the audi-

[1](von Békésy 1960)

[2]There is a large body of research in speech processing, usually on regular computer systems.

tory system, in order to facilitate the close integration of the various sensors on Cog. Scalable computational power was another requirement, to allow complex on-board and real-time signal processing of sensory data. An application is also presented as an example of the system's functional capabilities; simple audio and visual localization have been directly implemented on the system, while more complex features have been designed and developed off-line.

One of the fundamental auditory perception tasks is the localization of the sources of sounds, and much psychoacoustic research has been performed on human beings and animals to isolate the individual cues of sound localization. Very simple single source localization based on multiple cues, including vision, has been implemented on the auditory system, as a validation of the system and the underlying signal processing architecture. More interesting and complex localization techniques have been developed off-line using a standard mathematics package, and preliminary results are also presented. A neural network learns how to localize normal sounds in a realistic listening environment by integrating visual and audio cues. The key idea is that visual motion detection is used for self-supervised training of the network.

1.2 Scope and Contents

It is important to note the scope of the thesis; a major portion is the hardware and software design of the general purpose auditory system. It is not the intention of this thesis to explore all the intricacies and subtleties of sound localization, yet. The actual implementation of localization on the system is currently crude and simple, but is meant more as a verification of the auditory system's functionality; it forms a signal processing foundation for future work in auditory perception and is a first step towards more complex integration and perception. Work performed off-line with neural networks gives an indication of what can and will be performed by the system.

Chapter 2 contains a brief introduction to auditory perception and neural networks. Some recent relevant work in both fields is also presented.

Chapter 3 covers the design and implementation of the actual auditory perception system, including the issues that affected the overall design of the hardware and low-level software. discussed.

Chapter 4 describes the signal processing architecture of the auditory system implementation of simple two-dimensional sound localization using two audio cues and vision. Work that forms the basis of future, more complex, perceptual tasks is also described in depth.

Chapter 5 presents the results of the off-line neural network development and training.

Chapter 2

Background

By indirections find directions out[1]

2.1 Auditory Perception

2.1.1 Terms

In describing sound localization, several terms need to be defined. The head of the listener can be thought of as a sphere, with three planes that intersect at a point in the center (refer to Figure 2-1). As an approximation, the center can also be thought of as the midpoint of the line segment joining the two ears. Localization studies often consider only the horizontal plane, called the azimuthal plane. *Azimuthal angle* will be defined in this paper to be the angle in the azimuthal plane, with $0°$ corresponding to directly in front of the head. The medial plane is the vertical plane that is used to describe elevation information.

With two openings, sounds can only be localized on a single plane. Human beings can localize on two planes due to the effects of the outer ear structures called pinnae, that frequency filter incoming signals depending on elevation, and shadows high frequency sounds, which is used in front-back determination.

The basic configuration of the auditory system described in this thesis includes a pair of microphones and no external structures. To accomplish sound localization in more than one plane, the system can be easily expanded with an additional pair of microphones.

2.1.2 Localization techniques

Sound localization for human beings is primarily a binaural phenomenon, and cues are usually based on differences between the inputs to the two ears. Two cues that have been found to play the most dominant role in the direct sound field are interaural time (ITD) and intensity (IID) differences. Binaural cues are effective, with some overlap,

[1]Hamlet, Act II, Scene I

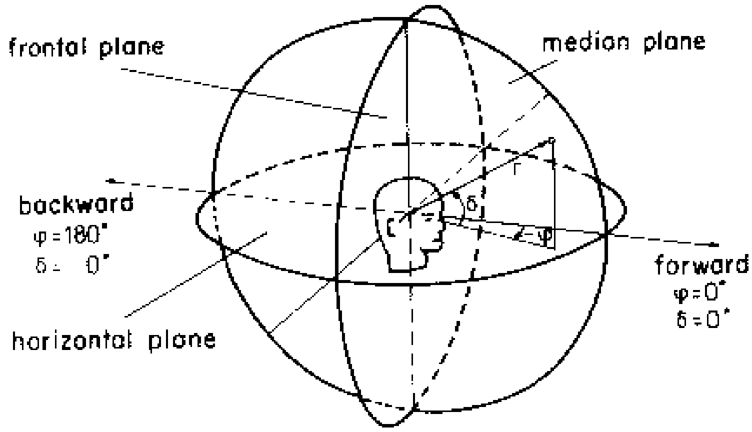


Figure 2-1: Planes used to describe localization. [Blauert, p.14]

for different sounds and situations. Robust localization is achieved when a system can combine information gathered from different cues to localize different sounds.

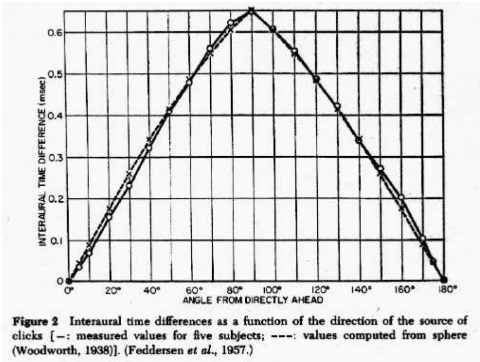
ITDs arise from the fact that the two ears are located a finite distance apart; sound impinging on the near ear takes some time to reach the far one. Delays range from 0 sec. for a source directly in front of the head to about 700 μsec. for an azimuthal angle of ± 90 degrees. For low frequency signals, below about 1.5KHz, the ITD can be measured as a phase delay in the left and right channel waveforms. For higher frequencies, the resolution of the ears is not fine enough to distinguish the phase difference. In this case, the onset time difference of the signal envelopes at the two ears provides a form of ITD (Burgess 1992). Using some geometry, an approximate expression for interaural time difference can be derived[2]. Figure 2-2 shows a close correspondence between the approximate model and actual ITD. Note however that with an adaptive system such as the one presented here, there is no need to specify head-specific head parameters such as diameter, microphone position, etc.

High frequency sounds, those having wavelengths that are comparable to the width of the head, are "shadowed" by the head; the intensity of sound entering the far ear is diminished with respect to the sound entering the near ear. The interaural intensity difference can not be modeled as easily as ITDs. The head shadow effect can cause IIDs of up to 20dB, and the effect is very frequency dependent as shown in Figure 2-3 (Mills 1972).

Head motions remove ambiguities from localization that may occur from using ITDs and IIDs alone. Although it has been shown that, with fully developed auditory systems, we can localize sounds without head motion (Blauert 1983), it is also known that newborn human infants orient their heads to the general direction of sounds (Muir & Field 1979).[3] In addition, other experiments have concluded that localization

[2]If we model the head as a sphere of radius 8.75cm, $\Delta t_{\mu sec} = 255(\theta + sin\theta)$, where θ is the azimuthal angle (Mills 1972)

[3]Muir et. al. admit however that it can not be concluded that newborns actually localize sound or possess a spatial map.



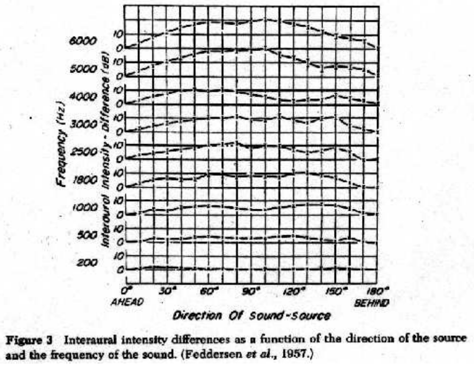


Figure 2-2: ITD Functional Form (Mills 1972)

Figure 2-3: IID Functional Form (Mills 1972)

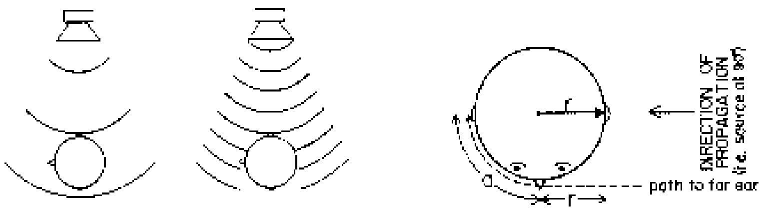


Figure 2-4: Interaural Time and Intensity Differences. [Durrant, p. 251]

resolution in human adults is greatest in the area directly in front of the head, so it makes sense to orient the head towards the sound source for better localization (Mills 1972).

Vision serves as a significant non-acoustic localization cue. In many cases auditory cues are ignored if they conflict with visual cues. When watching television or a movie, we perceive speech to be coming from the mouths of people instead of from speakers. Similarly, a ventriloquist's dummy appears to be actually speaking if its mouth moves synchronously with the speech. It is because of this phenomenon that vision was chosen to be the reference for training the neural network.

Other localization cues include reflections off of shoulders and the upper body and pinnae shadowing, though the exact mechanisms of these cues have not been studied as thoroughly as the ones described above, and will not be explicitly utilized at first in this research project.

2.1.3 Advantages of an Adaptive Learning System

The necessity of a learning component in auditory localization is obvious for several reasons. Each individual organism has different sized and shaped heads and bodies; moreover, as the individual matures, the size and shape of the body changes. Thus, localization cues, which are affected by the size and shape of the body, would be different for each individual at different ages, precluding any sort of neural encoding of size/shape information. It has also been shown that localization of unfamiliar sounds is worse than for familiar ones, so the auditory system clearly adapts to new sounds during the life of the organism (Bose 1994).

With respect to the humanoid robot, there has been a deliberate decision not to "hard-code" or store models of the auditory system. As discussed in Section 2.1.2, psychoacoustics researchers have modeled approximately the functions of ITD and (to a lesser extent) IID cues. Rather than using these models explicitly, there is biological motivation to learn the functional maps adaptively so that calibration and head specific parameters are unnecessary.

2.1.4 Auditory-Visual Integration

There is biological evidence that vision plays a major role in the development or "training" of sound localization, and it is this biological basis that provides the inspiration for much of the project. The underlying assumption is that there is a corresponding motion associated with most normal and "interesting" sounds the robot is likely to hear.[4]

Investigations with owls have determined that owls that have had one of their ears occluded since infancy could not, after reaching maturity and having the ear plugs removed, correct their auditory localization errors without visual input. With the plugs removed and vision fully restored, the owls could "relearn" how to localize sounds correctly. If, however, vision was restored but subjected to a constant error using prisms, the owls would adjust their localization such that localization errors match the induced visual error. Vision therefore provides the spatial reference for "fine-tuning" auditory localization (Knudsen & Knudsen 1985).

Auditory-visual integration is important not only for localization, but other perceptual tasks. Speech perception also benefits from visual input; isolated word recognition in a noisy environment improved significantly when normal hearing subjects were able to see the speakers as well as hear the speech (Yuhas, Jr., Sejnowski & Jenkins 1990). This is not surprising, since even those who have impaired hearing can learn to "lip read" and thus perceive speech mostly or solely from vision.

It should be noted that, once the neural network has been trained, the auditory system can direct the eyes to "interesting" objects that are not initially in the visual field. This system can be used to initiate head movements based soley on audio stimuli. This will aid in future work in object recognition, as Cog will be able to make assumptions on what sort of object it is looking for or trying to identify by the nature and direction of the sound the object makes.

[4]This assumption is even more valid when one considers that Cog is to behave like a human infant; infants are often subjected to rattles, exaggerated motions accompanying sounds, etc.

2.1.5 Listening Environment

It is important to consider the problem of localization in a realistic setting.[5] Many researchers, when studying sound localization, work in either anechoic chambers or approximate direct sound fields to simplify the processing or experiment that is performed. While this makes localization much easier, it is not realistic, as we normally live and interact in closed spaces that give rise to echoes and reverberations from reflections off of walls and objects. A very popular test sound in psychoacoustic research, the continuous tone or sinusoid, is ironically one of the most difficult to localize in reverberant fields. What is desirable, therefore, is to have the sound localization system handle sounds in both direct and reverberant fields, and adapt techniques that will be optimal for each.

A realistic listening environment includes naturalistic stimuli and both direct and reverberant sound fields. The listener is said to be in the *direct sound field* if the sound source is located sufficiently close that the first arrival of the sound dominates the signal entering the ears; subsequent echoes due to the reflection of the original sound off of walls and other objects are negligible. In the *reverberant field*, the listener is far enough from the source that the sound heard by the listener is due primarily to repeated reflections; localization becomes difficult since the localization cues of the initial direct sound are soon corrupted by reflected sounds that arrive from all directions.[6] The major cue that must be used in the reverberant field is the onset time difference of the signal envelopes (Bose 1994). Since this cue disappears after the start of the signal, continuous tones can not be accurately localized in reverberant fields, while clicks and other transients, with sharp onset time differences, can be localized quite well. This is fortunate, since normal everyday sounds, including speech, are rarely continuous pure tones, but complex, transient signals.

2.1.6 Related Work

Researchers at the ATR Human Information Processing Laboratories have started to work on a head/eye/ear system that can learn a spatial mapping between auditory and visual stimuli. They too make the assumption that acoustic and visual signals that occur roughly at the same time (temporally correlated) are from the same source (spatially coincident). They use somewhat of an artifical setting, working with a computer controlled speaker/light array; a light turns on at the same time its corresponding speaker emits a sound. The system makes an association between the motor commands necessary to saccade to the light (center its image in the visual field) and the left and right power spectra of the ears (Gamble & Rainton 1994).

Another related effort is the Anthropomorphic Auditory Robot developed at Waseda University, Japan. They have developed a neural network that performs front/back determination of pulsed sounds, without visual input, and uses differences

[5]A detailed discussion of acoustic theory is beyond the scope of this thesis, and interested parties may refer to dedicated texts on acoustics. One such text is (Beranek 1970).

[6]It has been shown that the human auditory system suppresses these later arriving signals somewhat when determining directionality.

in both onset time and power spectra. Results were promising, although the experiments were performed in an anechoic room (Takanishi, Masukawa, Mori & Ogawa 1993).

2.2 Neural Networks

2.2.1 Introduction

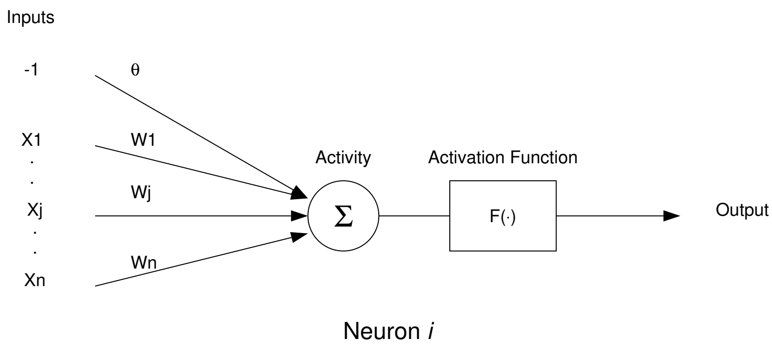


Figure 2-5: Perceptron

A comprehensive introduction to neural networks is beyond the scope of this thesis.[7]

While there are several different neural network architectures and methodologies, the feed-forward multi-layer perceptron (MLP) remains the most widely used and succesful architecture. The basic computational unit of a MLP is a perceptron, shown in Figure 2-5, that performs a nonlinear trasformation of its inputs to compute an output. The output of neuron i, given N inputs $x_1 \cdots x_N$, is expressed mathematically as :

$$y_i = F(\sum_{j=0}^{N} w_{ij} x_j - \theta_i)$$

where w are the *weights* associated with each input, θ is the *threshold*,[8] and $F(\cdot)$ is the *activation function*, a quantizer that limits the response of the perceptron.

As the name implies, a MLP consists of several layers of perceptrons, with usually full connectivity between layers, but none among perceptrons within a layer. MLPs are popular due to their relatively simple architecture and the existence of the back-propagation learning algorithm, which allows straightforward and efficient updating of the weights in the neural network. Section 4.3 goes into more detail about the specific multi-layer perceptron used to localize sound.

There is a large body of research dealing with applications of neural networks to signal processing. Neural networks are ideally suited for signal processing and

[7](Haykin 1994) provides an excellent introduction to the entire neural networks field and is highly recommended.

[8]To simplify notation, the threshold is usually considered to be just another weight with a fixed input of -1.

especially for audio/visual processing for several reasons: they are nonlinear computational units that can learn an input-output mapping by generalizing and adapting to changes in the input; they can perform data reduction by extracting features from a higher order input space; they are biologically inspired.[9] Currently, only biological auditory systems have completely and effectively solved complex perceptual tasks, like sound localization, discrimination, and recognition in a noisy environment (Morgan & Scofield 1991); by performing computation in a similar manner to biological systems, it is hoped that artificial systems can achieve a similar level of performance.

It is also natural to try to exploit a property of neural networks that make them "universal approximators." It can be mathematically proven that certain neural networks are able to approximate any function, given enough input/output pairs and training. It has been discussed above that psychoacoustic research has revealed that the ITDs and IIDs take some functional form, though the relative interaction and combination of these (and other) cues are far from being well understood. Neural networks are ideal not only for determining the functional forms, but for combining them into one consistent system.

While it might be said that localization is a "simple" task that has been already solved, it is the author's firm belief that *robust* localization, integrated with vision and able to handle a variety of sounds in a realistic listening environment with noise is a far from simple task. In a noise free environment, localization of a limited set of samples would be relatively easy. Indeed, most of what we have gathered from psychoacoustic research come from only artificial settings, like anechoic chambers and test "clicks." Applying standard neural networks architectures in novel ways will hopefully allow "simple" tasks such as localization to perform well in realistic, and therefore complex, listening environments.

2.2.2 Related Work

There have been recent attempts to specifically apply neural networks in the integration of vision and audition. (Yuhas et al. 1990) explore the use of neural networks to improve speech perception, specifically the recognition of isolated vowels. One MLP was trained to estimate the spectral characteristics of the corresponding acoustic signals from visual images of the speaker's mouth. An alternative MLP was trained to directly recognize vowels from the visual signals. Performances of both neural networks were similar to human performance.

[9] Inspiration does not necessarily mean duplication; biological neural systems are significantly more complex. However, both systems approach a problem by exploiting the benefits of parallelism and high connectivity.

Chapter 3

System Design and Implementation

It's just a matter of hardware and software now ...[1]

3.1 Setup

3.1.1 Cog

The humanoid robot, Cog, on which this project is based is an ambitious effort lead by Professors Rodney Brooks and Lynn Stein at the MIT Artificial Intelligence Laboratory to understand human cognition by embodying intelligence in a physical manifestation (Brooks & Stein 1994). This belief, that cognition must be rooted in a physical embodiment and can not usefully be relegated to simulation, is a notion firmly believed by all members of the group, including the author. One of the unique features of this auditory system, the author feels, is that it is running on a human-like robot, with vision and dextrous upper body and arm motion abilities. Using such a system at once simplifies and complicates the task of sound localization. Compared to static setups that most researchers use, having a mobile head that can orient itself with three degrees of freedom makes the system much more dynamic, as the robot can orient the microphones in such a way to maximize the sensitivity of localization and remove ambiguities. Having vision capabilities introduces non-acoustic cues for sound localization, and is, as will be discussed, vital for the robot to learn how to localize. Complexities arise from the fact that not only will the robot be generating its own noise, from its motors and manipulators, but it is currently located in a very noisy environment where reverberations will likely be dominant (refer to Figure 3-1). The system described in this thesis is flexible enough to overcome the difficulties involved in using a humanoid robot.

The robot itself, shown in Figure 3-2, is still under development.[2] This robot

[1]Anonymous colleague, 1992 Undergraduate Group Engineering Design Project

[2]Refer to papers by other members of the group for a more complete description of the other subsystems (Ferrell, Scassellati & Binnard 1995, Marjanović 1995, Williamson 1995, Matsuoka 1995).

is built from the waist up, and is currently bolted to an immovable stand; as we learn more about the issues involved in embodied intelligence, more robots will be built incorporating the lessons we have learned. The upper torso and head assembly are complete, with each having three degrees of freedom. The head houses an active vision system consisting of four cameras, mounted in pairs and having two DOF each. Each "eye" consists of a pair of cameras, one having a wide angle view and the other a narrow view; this simulates the fovea and wide-angle vision of a human eye. The visual input for the auditory system will be primarily from the wide angle cameras. Currently in development is a six degree of freedom arm and a lightweight grasping hand. The entire robot will be enclosed in a plastic shell that will serve as the "skin" of the robot. The microphones will be mounted directed on the plastic head casing.

The "brain" of the robot is an off-board, large-scale MIMD parallel computer, referred to as $\pi\beta$, with a Motorola 68332 micro-controller[3] board at each node. The board has local memory that contains L, a multitasking Lisp language, and user programs. L will be the primary language for high level processing, and is described in Section 3.2.6. The current backplane supports sixteen nodes, loosely coupled and in a configurable but fixed topology network, with communication between nodes and sensory hardware accomplished through the use of dual-ported static RAMs, which provide independent, asynchronous access to the same memory range through two ports. $\pi\beta$'s features—modularity, total asynchrony, no global control or shared memory, scalability—were chosen to make the entire system have some degree of biological relevance (Kapogiannis 1994). A new backplane is currently being designed for better performance and reliability.

Section 3.2 contains a more detailed description of the actual hardware and software of the auditory system.

3.1.2 Styrofoam Head System

While the main body of Cog is under development, a simpler head system has been constructed (See Figure 3-3). A Styrofoam head mounted on a hobby servo, found in radio control models, has one degree of freedom (pan) and houses two electret condenser microphones.[4] A single CCD camera is "mounted" on approximately the center of the forehead. Instead of using the multi-processing backplane, a smaller version, referred to as the μ-Cog, with support for two nodes is used. While the size and material of the two heads are not identical, using an adaptive learning system like a neural network allows the auditory system to adapt to different configurations.

[3]The MC68332 is a 32-bit micro-controller running at 16.78 MHz, with built-in timer and serial subsystems.

[4]The hobby servo, not used in this particular application yet, is controlled by a pulse width modulated (PWM) signal from a parallel processing node. Using the servo to perform some form of tracking based on sound localization is a logical next step.

Figure 3-1: Cog Setup

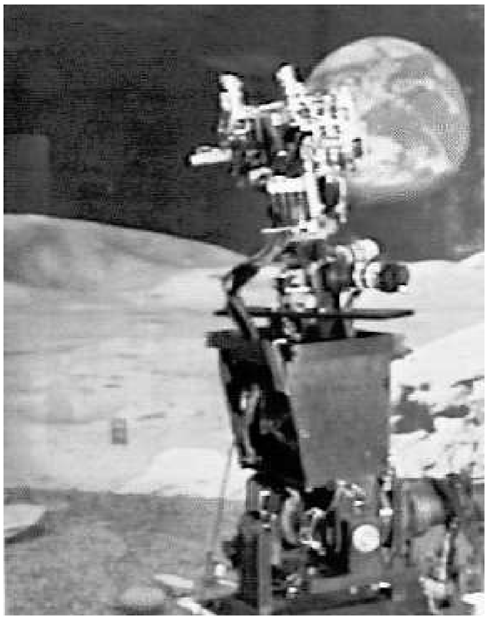


Figure 3-2: Cog

3.2 Design and Implementation

A major portion of the project consists of developing a general purpose auditory system for Cog with enough flexibility to be used in wide variety of applications, including but not limited to localization. The system has been designed to take full advantage of the parallelism of Cog's architecture, and to be as scalable as possible; multiple microphones and speakers can be easily interfaced to Cog. Multiple DSP boards may also be connected together in various configurations. Sound localization was chosen as a natural first task to accomplish, as it is one of the most fundamental auditory perception tasks, and makes full use of the modularity of this system. At first, all localization will be referenced on a plane in the frontal region of the azimuthal

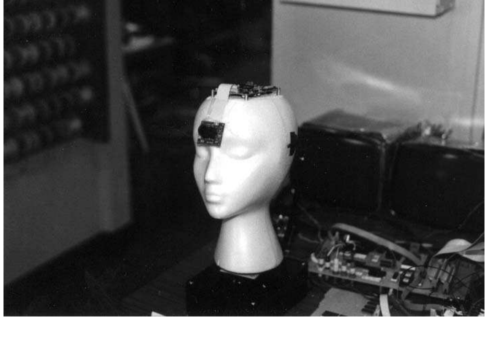


Figure 3-3: Styrofoam Head Setup

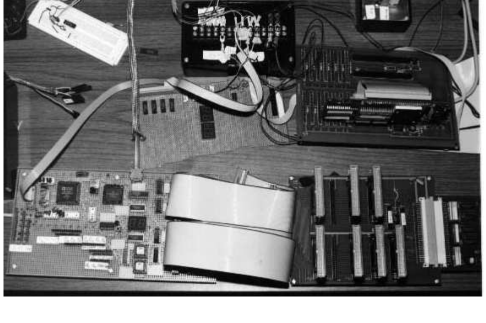


Figure 3-4: Hardware Setup

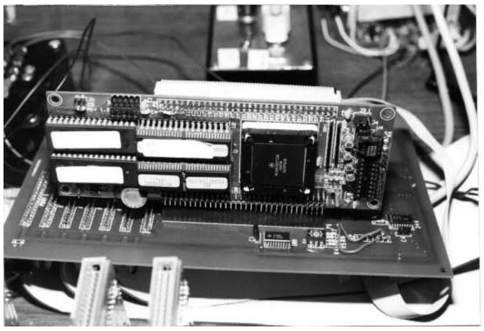


Figure 3-5: μ-Cog Setup

plane.[5] An overall system diagram is shown in Figure 3-6.

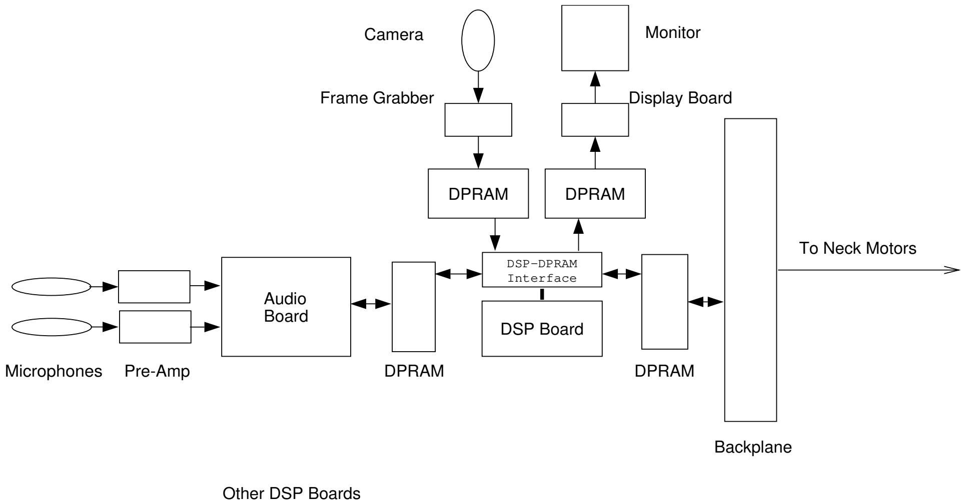


Figure 3-6: Overall System Diagram

3.2.1 Dual Ported RAM

A key component in this system is the dual-ported static RAM (DPRAM), which is used for communication among most of the components of the humanoid robot, including the parallel processing nodes, the sensory (audio and visual) hardware, and a high performance DSP board. Two devices connected to the two ports of a DPRAM can access the same memory range simultaneously. DPRAMs used in this system, and in Cog in general, have 8K of 16-bit words. A crude form of handshaking is available using the DPRAM's interrupt mechanism; devices on either of the ports can write to distinct memory locations to generate an interrupt on the other port. It is through this mechanism that most components of the auditory system and Cog in general can perform synchronized communication.

[5] As the system develops, we will add more microphones and localize fully on two planes.

Using DPRAMs provide a basis for designing modular system components, and makes the high level of integration of the components, necessary to perform intelligent, human-like tasks, possible. Sound localization is one excellent example of such a task.

3.2.2 Microphone and Pre-amplifer

Figure 3-7: BT1759 Microphone

While the auditory system is designed to work with any standard microphone, the Knowles BT1759 electret condenser (pressure) microphone has been selected for this particular application due to its small size[6] and high sensitivity. It has a sensitivity of -60$\pm$3 dB re $1\frac{V}{\mu bar}$ at 1KHz, with a frequency response roll-off of around 10KHz (Kno 1973).

The pre-amplifier circuit, adapted from the sample circuit in the data sheets ((AD 1994)), is a simple inverting amplifier configuration that serves to level-shift and amplify the microphone output to "industry standard" MIC input levels that the codec expects. The op-amp is also configured to low-pass filter (first order) the analog input, for anti-aliasing purposes. (See Section 3.2.3 for more information)

Appendix A.1 contains the schematic for the preamplifier.

3.2.3 Audio Board

A custom audio board has been designed with the same modular philosophy as with the rest of the robot; both raw and processed sound data is communicated via DPRAMs, so that the board can be directly interfaced not only to the backplane, but to a DSP board for faster computation via a DSP-DPRAM interface board that was also designed and built. The board contains a codec,[7] the Analog Devices AD1848K, which allows simultaneous stereo audio recording and playback. Data is transferred to and from the codec on an 8-bit data bus. Sampling and playback rates of up to 48KHz on both channels is possible, with up to 16-bit resolution. Sound data

[6] It was designed for hearing aid applications

[7] Codec stands for *co*mpressor-*dec*ompressor, and has come to mean a combined ADC/DAC, usually with a high degree of programmability.

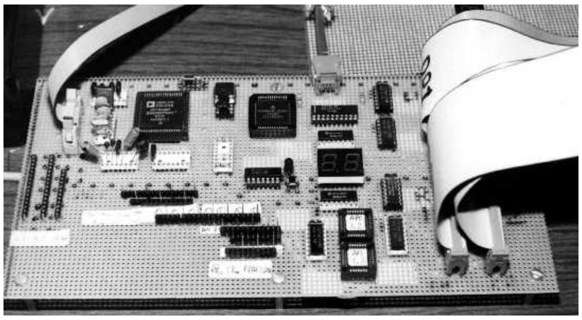


Figure 3-8: Audio Board

can be stored in a variety of standard formats, from uncompressed linear PCM to compressed μ-law and A-law encodings. The codec is highly programmable, with 16 registers specifying every aspect of data acquisition and conversion. The board has been designed to allow on-the-fly changes of all acquisition/conversion parameters, including sampling rate, data resolution, and internal input amplifier gain control. The built-in analog to digital converters (ADC) include linear phase (decimation) low-pass filters with a 3-dB point at around the Nyquist frequency.[8] Only a simple single pole external low pass filter is necessary to insure anti-aliasing, simplifying the microphone pre-amplifier circuit greatly (AD 1994). Any standard microphone can be interfaced to the system; electret condenser microphones were chosen for their small size and power requirements.

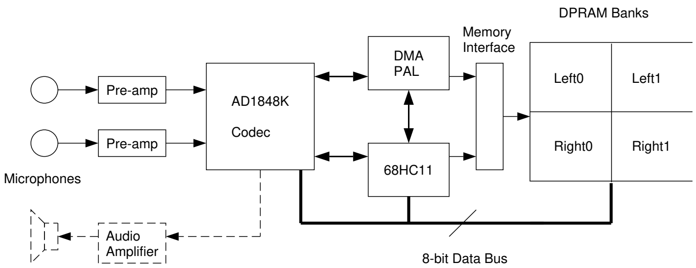


Figure 3-9: Audio Board:System Diagram

Figure 3-9 is a system diagram indicating the various components of the audio board and their interactions. The audio board has been designed to be very simple to program and use. The codec itself demands a relatively complex interface of direct and indirect registers. For optimal performance Direct Memory Access (DMA) is necessary, and has specific timing requirements that must be met for lossless data transfer. The problem is compounded when simultaneous data capture and playback (digital to analog conversion) is desired. To achieve a modular system, and remove the burden of exact timing and register manipulations, it was decided to add a Motorola 68HC11 microcontroller, on the same 8-bit data bus as the codec and DPRAMS, to

[8]The Nyquist Frequency is one-half the sampling frequency and is the highest frequency that can still be accurately represented by the discrete time series without aliasing

handle all low-level details of codec interfacing, so the audio board's only external interface will be through DPRAMs. The DPRAMs themselves have a very simple interrupt mechanism that is a lot easier to interface. As a result, any device (backplane, DSP board, etc), referred to as the *host*, that is connected to the audio board through DPRAMs can receive stereo audio data and send mono data simultaneously without having to worry about codec register accessing and the intricacies of DMA. Customizing acquisition/conversion parameters is a mere matter of the host writing the parameters to one of the DPRAMS and generating an interrupt.

The codec has two modes of operation, programmed I/O (PIO) for register accesses and DMA for sound data transfer. The 68HC11 handles the switching of the these two modes as necessary; parameter updates and status information requests by the host are read by the 6811 and processed using PIO. For actual sound transfer, the 6811 switches the codec into DMA mode, and controls Programmable Array Logic (PAL) chips, implementing a finite-state machine (FSM), that perform Direct Memory Access (DMA) between the codec and multiple banks of DPRAMs. Note that the 6811 itself does not transfer any sound data, as its 2MHz operating clock and .5 μs instruction cycle time would be a bottleneck.[9]

The bank of four DPRAMs serves as a double buffer for seamless data transfer. Stereo audio signals can not only be continuously captured by the codec writing to alternate Leftx/Rightx banks a bank at a time, but a mono signal can be output through the codec to a speaker simultaneously. Appendix A.2 contains selected schematics of the audio board, information concerning the codec from the data sheets, and the state diagram implemented by the PALs.

3.2.4 DSP Development System

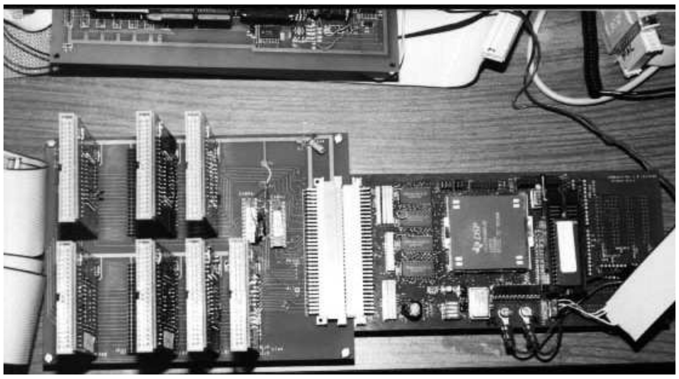


Figure 3-10: DSP Board and DPRAM Interface

TI TMS320C40 DSP Board

It was soon determined that at the current state of development, the parallel processing nodes needed to be augmented with a fast processor for applications requiring

[9] DMA transfers are a necessity, especially with simultaneous stereo 16-bit recording and playback at 48kHz, which would require data transfer rates of close to 400kHz.

significant computational power, such as signal processing. While the nodes running L routines are sufficient for low data throughput applications, and even for simple visual processing of frames, sound processing is much more dependent on complex computations such as convolution and FFT's.[10] Such tasks can be accomplished extremely quickly and efficiently with a digital signal processor.

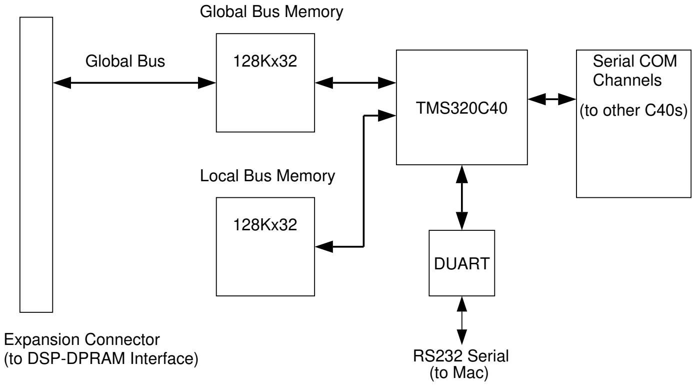


Figure 3-11: DSP System Diagram

The Texas Instruments TMS320C40 (C40) Digital Signal Processor was chosen for this task. It is capable of performing about 200 million operations per second, and has a powerful instruction set optimized for a variety of signal processing tasks, from time domain correlation and statistical analyses, to frequency domain Fast Fourier Transform computations. The C40 also has several features that make it ideal for connecting multiple DSP boards together for parallel processing; six DMA serial communications channels are specifically designed to allow 20-Mbytes/s bidirectional asynchronous transfer between multiple C40s. Two identical external data and address buses[11] are ideal for shared memory configurations (TI 1993).

A DSP board, shown in Figure 3-10, was independently developed at the AI Lab and is another key part of the system. A system diagram is given in Figure 3-11. The board adds 64K of static memory, divided equally on the local and global buses. C40 programs are downloaded serially via the DUART. The connector brings out the global bus for external expansion.

DSP-DPRAM Interface

The DSP-DPRAM interface board connects to the DSP board through the expansion connector, and allows modular connections between the DSP board and any other device in Cog, from the parallel processing nodes to the audio and vision boards.

[10]Since visual processing consists of taking "snapshots" of the image in time, many tasks are still possible on a slower processor by using a number of techniques, from sub-sampling the image, working with fewer frames per second, etc. We do not have as much luxury in sound processing, where the two dimensions of time and intensity are much more tightly coupled. There is no notion of "snapshots" and data must be taken continously. Unlike vision, audition usually can not rely solely on time-domain techniques, but requires in addition frequency domain (FFT) analyses.

[11]The C40 has a "harvard" architecture, with separate buses for data and addresses.

As shown by the block diagram in Figure 3-12, the DSP DPRAM interface board is a very simple circuit that interfaces up to 9 DPRAMs to the DSP.

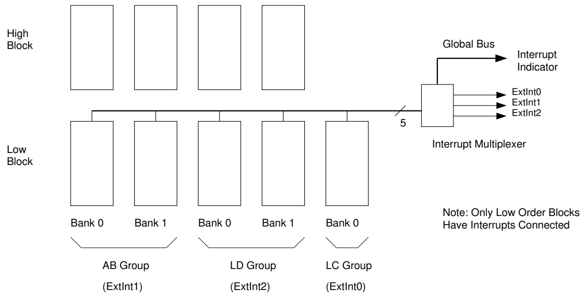


Figure 3-12: DSP-DPRAM Interface

Before describing the DSP system in detail, some terms should be defined. Each individual 8K word DPRAM is defined to be a *block*. To increase data throughput, and since the data bus of the C40 is 32-bits wide, two (16-bit wide) DPRAM blocks are combined to form one *bank*; the C40 accesses the DPRAMs one bank at a time. (It therefore makes sense to speak of high-order and low-order blocks in a bank.) Due to a limitation of the C40 that is discussed below, banks are organized into *groups*.

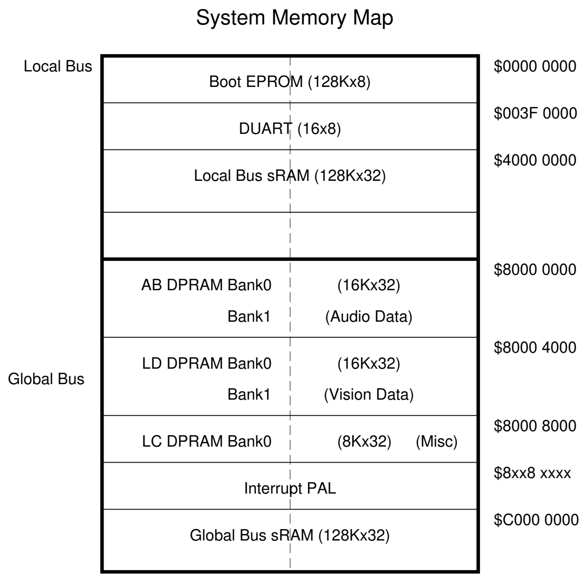


Figure 3-13: DSP System Memory Map

Interrupt Handling The C40 has relatively few external interrupt lines[12], so a scheme for multiplexing interrupts was devised. The DPRAM banks were organized into three *groups*, each sharing a single external interrupt. A PAL multiplexes the

[12]The C40 itself has four; the DSP board described here reserves one of the external interrupts, leaving only three.

interrupts and passes them onto the C40 and keeps track of which particular bank of a group caused the interrupt. The PAL is mapped to an address range on the C40 memory map (See Figure 3-13) so that when the C40 accesses this range, the PAL outputs the particular bank that caused the interrupt onto the data bus. Multiple interrupts, from different groups, that occur simultaneously are also supported.

3.2.5 Vision System

The video camera subsystem has been developed by other members of the Cog group. Two boards, the frame grabber and display board, have been built with DPRAM interfaces that allow the capture and display of video data. The frame grabber board uses standard video chips to convert normal NTSC video signals to 8-bit grey-scale values with a screen resolution of 128 by 128 pixels. Video data is written to the DPRAMs at a rate of 30 frames per second, with the end of frame signaled by an interrupt. The display board takes 8-bit grey scale values and converts them into a standard (black and white) NTSC signal.

The camera used is an inexpensive color CCD camera with a field of view of around 70 degrees, although any camera with an NTSC output would be sufficient.[13]

3.2.6 System Software

Most of the low-level signal processing routines and system software are written in assembly language and C for the C40. Presently, high-level processes such as the neural network have been implemented off-line in Matlab[14], and will shortly be ported to the DSP in C. Eventually, when a more stable version of the backplane is designed and built, high-level processes will be implemented in *L*, a downwardly-compatible multi-tasking subset of Common Lisp written by Professor Rodney Brooks, for the parallel processing nodes. *L* provides a multitasking lisp environment for the development of "brain models," where the nature and organization of processing will be influenced by actual biological brains. The goal is not to build a model of an actual brain, but to take inspiration from the modular structure of brains (Brooks & Stein 1994).

[13]While the camera output is color, the frame grabbers output grayscale values.

[14]Matlab is a registered trademark of The Mathworks Inc. and is an easy to use mathematics software package.

Chapter 4

Application

Use the right tool for the right job.[1]

4.1 Procedure

It was originally planned that the neural network to perform sound localization will be implemented on Cog itself, using the parallel processor nodes. Due to several factors, it became prudent to develop on a separate setup, the temporary Styrofoam head system described in Section 3.1.2, and mostly on the DSP system for tighter integration of audio and vision. When the new backplane is completed, and other structures of Cog built, the auditory hardware can be easily mounted on the robot itself, and the high-level software ported to L.

Even in this separate setup, it became much easier to perform initial signal processing and algorithm prototyping of cue extractors using Matlab. Audio and visual data was collected using the integrated auditory system, and then transferred to a Sun workstation on which Matlab was running. Since the DSP C compiler also resides on the Suns, this was not as inconvenient as originally thought. Figure 4-1 shows the actual working setup. The Macintosh is running MCL, a version of Common Lisp, that communicates with L on the μ-Cog via a serial port. Binary data received by the μ-Cog via DPRAMs can be saved into a file on the Macintosh through this link. The Macintosh is also running a terminal program that communicates with the DSP board via another serial port; DSP programs are downloaded to the DSP using this link, and text output from the DSP can be saved to a file on the Mac.

Development of the neural network was also based primarily in Matlab, using parts of the Neural Network Toolbox. Matlab has a C-like language, so the developed neural network should be easily ported to the DSP.[2] There are some implementation issues such as synchronization that must be addressed; see Section 4.4.

Initial experiments involved using the auditory system to collect raw sound data and processed visual data; these were then passed to the μ-Cog which then saved them to the Mac. They were then transferred to the Sun workstations and processed

[1]Old engineering maxim.

[2]Porting the neural network eventually to L should not be much more difficult.

using Matlab. Section 4.4 describes the implementation of an actual architecture to handle sound processing on the DSP board itself.

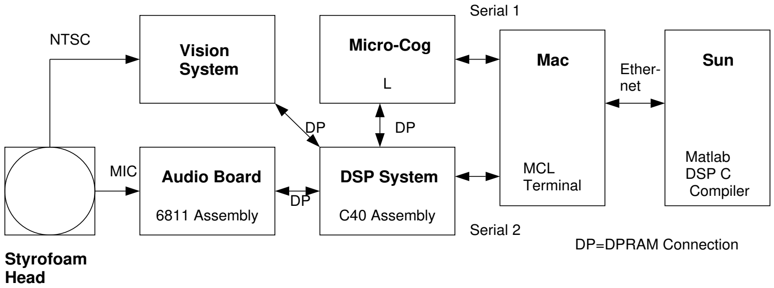


Figure 4-1: Overview of Development System

4.2 Cue Extractors

Very rarely are neural networks fed raw signal inputs; some form of pre-processing is usually performed to reduce the data into a more manageable form while retaining the important features and characteristics.

The *cue* extractors for the input layer can be divided into two broad categories, time domain and frequency domain functions. Most extractors will use short-time block signal processing, in which the continuous input audio stream is divided into short time blocks and any processing is performed on a block at a time. Each extractor will give its own estimate of the source of the sound. Naturally, not all of the extractors will give meaningful estimates for every sound. It is up to the neural network to learn the relevance of each cue in each situation.

Discrete time representations of sound signals are usually analyzed within short-time intervals. Depending on the application, analysis frames of 5-25 ms. of the signal are selected during which the signal is assumed to be quasi-stationary, or has slowly changing properties. For sound localization, a good balance between a short enough window to catch differences in cues and a long enough window to obtain meaningful information seems to occur with a window length of about 10ms.[3]

Most of the short-time processing techniques, as well as the short-time Fourier representation discussed in Section 4.2.2, can be expressed mathematically in the form

$$Q_n = \sum_{m=-\infty}^{\infty} T[x(m)]w(n-m)$$

where $x(m)$ is the signal, $w(n)$ is the analysis window sequence, and $T[\cdot]$ is some linear or nonlinear transformation function. $w(n)$ is usually finite in length and selects a short sequence of the transformed signal around the the time corresponding to the sample index n. Q_n can therefore be interpreted as a sequence of locally weighted average values of the sequence $T[x(m)]$ (Rabiner & Schafer 1978). The choice of $w(n)$

[3]Like much research in sound and speech processing, this value has been empirically determined.

not only determines the interval over which the signal is evaluated, but by how much each data point within the frame should be counted; a typical sequence called the *rectangular* or *boxcar* window is defined as

$$w(n) = \begin{array}{ll} 1, & 0 \le n \le N-1 \\ 0, & otherwise \end{array}$$

The boxcar window weighs each data point within the window equally.[4]

Figure 4-2 shows a block diagram of the sound pre-processing that was performed with Matlab. Only "interesting" portions (ie. those with actual non-background signals) were selected to be pre-processed. These portions were then segmented into equal-length frames using a boxcar window. After some experiments a suitable configuration of sampling rate and window size was determined. Data was sampled at 22KHz with 8-bit resolution. The window length was set to 128 bytes (approximately 8 ms.) with no overlap. A more detailed discussion concerning the various design issues of the acquisition and short-time segmentation is presented in Section 5.4.1.

Localization cues were computed from the time and frequency domain measurements by subtraction, to result in the following: cues indicating a sound source in the left direction would be negative, cues indicating a sound source in the right direction would be positive.

4.2.1 Short-Time Time Domain Processing

Four time domain measures were chosen to obtain localization cues. A deliberate attempt was made to choose measures that were as simple as possible to compute, but could still provide useful information.

Phase Delay This is the dominant ITD cue for low-frequency and sustained signals. The delay of one channel with respect to the other can be computed by performing a cross-correlation of both channels (See Section 4.2.2).

Maximum Value The maximum positive value of each signal for each segment was determined. A localization cue based on the difference of the two maximum values in each segment is a form of IID.

Maximum Location The locations of the maximum values in each segment of each channel were also recorded. The difference in the locations is an ITD cue similar to phase delay, and is meant as an approximate measure of onset delay. Onset delay is a useful cue for high frequency sounds or complex transients.

$\sum$ **Magnitudes** Another simple to compute IID cue is the difference in the sum of magnitudes of the signals in each segment. This is an approximate measure of the short-time energy of each signal (Rabiner & Schafer 1978). As with the maximum value measure, the channel from the near microphone should have a greater energy (magnitude) content than that from the far microphone.

[4] Another popular window is the *Hamming* window, discussed in Section 5.4.1.

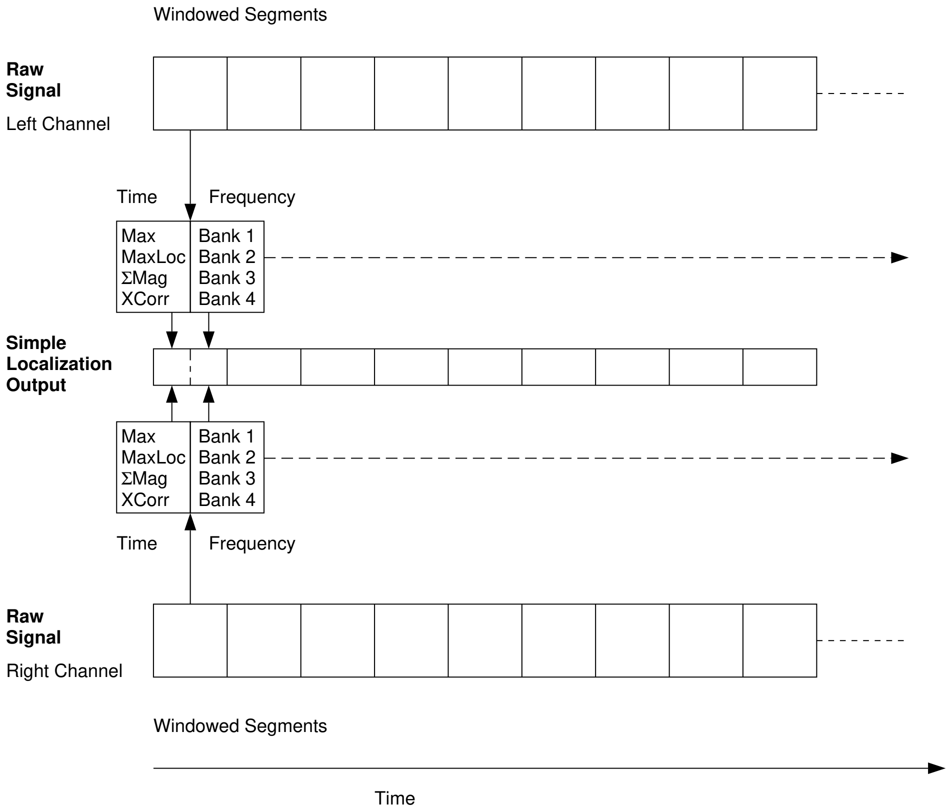


Figure 4-2: Localization Cues Block Diagram

4.2.2 Short-Time Frequency Domain Processing

Frequency domain cue extractors use the spectrum of the signal in their processing. There are several representations for the spectrum, but by far the most commonly used in digital systems is the Discrete Fourier Transform, having the mathematical form[5]

$$X[k] = \sum_{n=0}^{N-1} w(n)x(n)e^{\frac{-j2\pi nk}{N}}$$

As in short-time time domain processing, the boxcar window is often used for $w(n)$. Since an N-point discrete time series transforms to an $\frac{N}{2}$-point discrete Fourier Transform series (the other $\frac{N}{2}$ points of the DFT are symmetric copies and contain no additional information), the original time series is usually extended with zero-value samples to increase the resulting resolution of the DFT series (Beauchamp & Yuen 1979).

The development of the Fast Fourier Transforms (FFTs),[6] has improved the speed

[5]The DFT is the sampled version of the continuous short-time Fourier Transform (STFT), expressed as $X_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} w(n-m)x(m)e^{-j\omega m}$, within each window. ω is evaluated at N points around the complex unit circle, or $\omega = \frac{2\pi n}{N}$(Morgan & Scofield 1991).

[6]The term *FFT* is a misnomer; it is not a transform at all, but a collection of algorithms with which one obtains the Discrete Fourier Transform.

at which the DFT is computed from $O(N^2)$ operations to $O(NlogN)$ operations. Modern DSPs have been optimized mainly to perform Fourier transforms, and as a result spectral processing has finally become feasible in real-time intelligent systems. Even the implementation of time domain processing functions has benefited from the development of FFT techniques, as some analyses like correlation are now more efficiently computed by first transforming into the frequency domain, performing an equivalent operation, and transforming back into the time domain.

Correlation Analysis

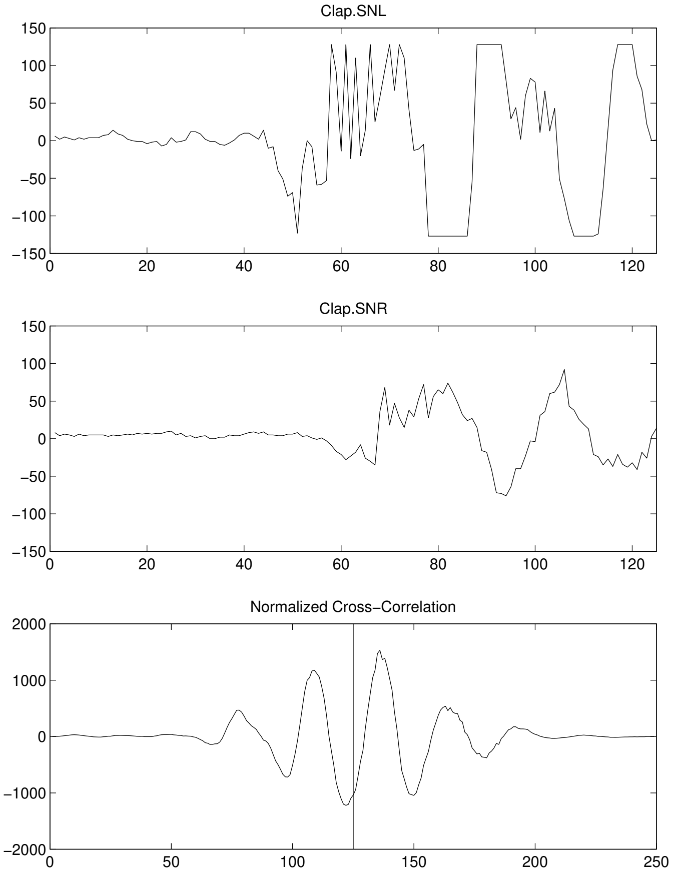


Figure 4-3: Cross correlation of beginning of clap

Correlation analysis has been used in a variety of fields from statistics, in determining the similarities or "correlation" between two signals, to control theory, in deriving an approximate impulse response function of the plant, to signal processing, in recovering signal from noise (Beauchamp & Yuen 1979). Another application of correlation analysis is to determine the phase difference between two identical time-shifted signals. While the left and right channel sound signals are not identical, they are similar enough to exploit this last application.

Discrete cross-correlation is an operator on two discrete time signals (assumed to have length N each) that can be expressed as

$$l[n] \otimes_\tau r[n] \equiv \sum_{m=0}^{N-1} l[m]r[m+\tau]$$

One interpretation of the above definition is that the two discrete time series $l[n]$ and $r[n]$ are multiplied together element-by-element, after one is shifted in time τ samples. The correlation will be large for some positive value of τ if the first signal, $l[n]$, leads the second, $r[n]$ in time, and for some negative value of τ if $l[n]$ lags $r[n]$. Figure 4-3 shows the cross correlation of the left and right channels of a clap performed to the left of the head (the left channel signal leads the right); note that the two signals are not identical, even with time shifting. Still, a peak to the right of the vertical line denoting $\tau = 0$ indicates that the left signal leads the right signal, which confirms visual inspection.

While it may seem straightforward to implement the correlation operator in the time domain, FFT techniques have proven to be more efficient by exploiting the discrete correlation theorem, stated as

$$l[n] \otimes_\tau r[n] \Longleftrightarrow L^*[k]R[k]$$

Thus, implementing efficient cross-correlation involves transforming the two discrete time series into their DFT representations, performing element-by-element multiplication of one DFT series with the complex conjugate of the other, and then inverse transforming the product back into the time domain (Press, Flannery, Teukolsky & Vetterling 1988).

Filterbank-Based Cues

One common representation obtained from signal spectrums consists of "banks" or groupings of passband filters, called *filterbanks* (Morgan & Scofield 1991). The center frequency of the filterbanks are usually spaced logarithmically, emphasizing the low frequency end of the spectrum, especially in speech processing. Due to the nature of IID cues, which are predominantly high frequency cues, it was thought that having equally spaced filterbanks would be more suitable.

Computing the DFT of the sound signals results in a spectrum with a bandwidth of $\frac{F_s}{2}$, where F_s is the sampling frequency. The range $[0, \frac{F_s}{2}]$ is divided into four equally spaced banks, and the sum of magnitudes in each bank are computed. These sums represent the average spectral density of the short-time signal at each frequency range.

4.2.3 Visual Processing

For this project, only very simple visual processing has been performed to train the network. Since we are assuming that the sound sources that will be used in the training phase of the network will have corresponding motion associated with them—door slamming, hands clapping, rattles shaking, etc.—visual processing consists of detecting the "centroid of motion," or the centroid of the difference between successive frames. This value will be used to derive a very coarse azimuthal angle, with only three values corresponding to "left," "center," and "right." This value will serve as a reference for training the neural network using error backpropagation (see Section 4.3.2).

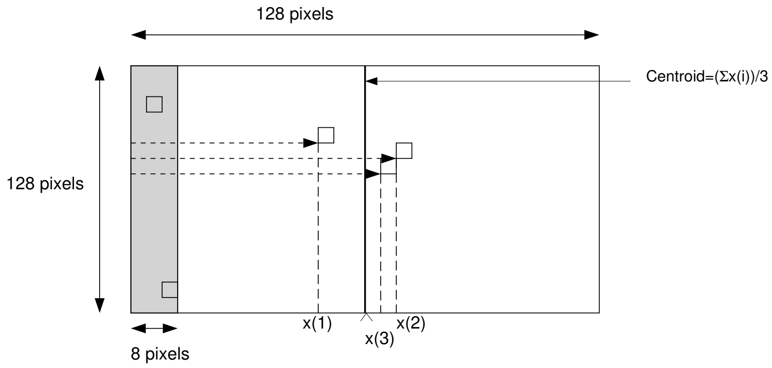


Figure 4-4: Motion pixel image (pixels in shaded block are ignored).

Figure 4-4 illustrates the concept of horizontal center of motion. After subtracting successive image frames, any pixel that had changed intensity between the frames would have an intensity value of the difference.[7] Large motions such as hand clapping in a particular part of the image will result in a number of motion pixels having nonzero values in the particular region. Since we are only interested in the azimuthal angle, averaging the *horizontal* components of all nonzero motion pixels will produce the mean, or centroid, of motion. We are admittedly making many assumptions about the nature of sound production and its associated motion. A more careful analysis will be performed in future work.

There is added complexity caused by noise from the (inexpensive) cameras. Large portions of the screen flicker constantly, adding noise to the successive subtraction operation. This was handled by a combination of averaging and thresholding, and is described in the Section 4.4. In addition, the pixels in the leftmost portion of the image were especially noisy, so were omitted from the processing.

High resolution is not necessary for this particular application. In fact, the numerical value of the centroid of motion is eventually converted into an abstract representation with only three distinct values (see Section 4.3.2).

4.3 Neural Networks

4.3.1 Design

The primary role of a neural network is to associate inputs, in this case binaural cues, with an output, the azimuthal angle. This association is stored in the network

[7]To avoid confusion, the term *image pixel* will refer to a pixel from the raw image. *Motion pixel* will refer to a pixel from the processed (successive subtraction) image.

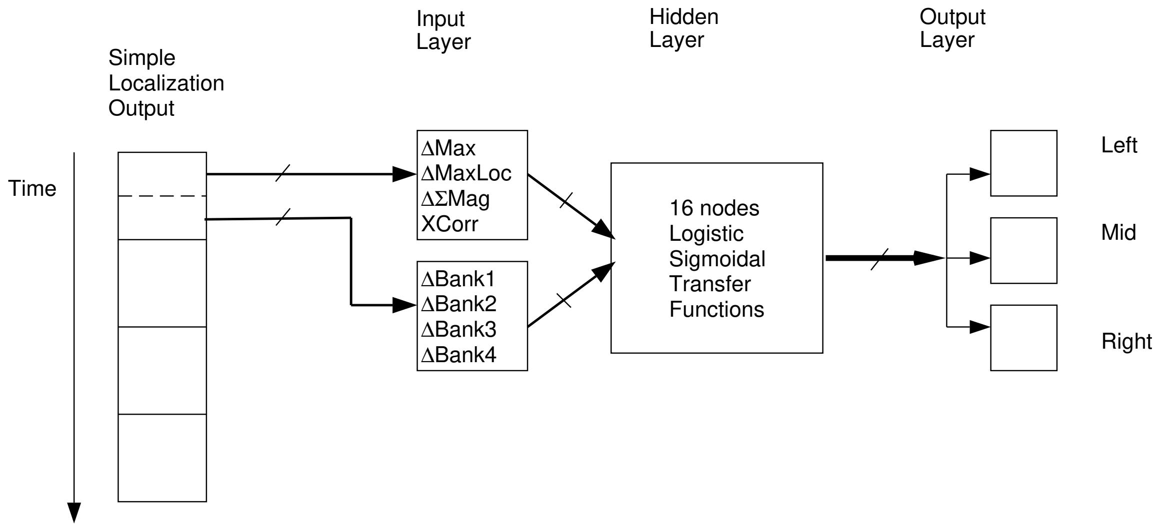


Figure 4-5: Neural Network Block Diagram

as the variable interconnection weights between nodes; these weights are a form of "memory." It is an established fact that we can localize familiar sounds better than unfamiliar sounds. This is especially true for sounds that produce ambiguous localization cues, since the auditory system learns to pay attention to only those cues that are in agreement and discard irrelevant or misleading cues (Blauert 1983). As stated earlier, vision plays a major role in this learning process, for both humans and the neural network described here; everyday sounds, including speech, are mostly transient in nature, and their production often involves some sort of motion. Taking advantage of both audio and visual features of such stimuli will improved the robustness of sound localization.

Figure 4-5 shows the architecture of the neural network that was implemented in Matlab. The input layer (8 nodes) is fed with the simple localization cues that were described above. The hidden layer (16 nodes) represents the associations and interactions of the various localization cues. The output layer (3 nodes) of the network will produce some form of "angle" on the azimuthal plane, that will be the best estimate of the location of the sound source. This output will be used in conjunction with the visual motion detector to generate an error signal for back-propagation learning.

The neural network implemented in this project was a standard feed-forward multi-layer perceptron with one hidden layer. Neural networks with more than two hidden layers are rarely necessary, and the bulk of neural networks research deals with MLPs with only one of two hidden layers (Haykin 1994). Standard MLPs can only learn static maps from input to output. This is not a severe limitation for the purposes of localization, as it has been determined that the cues, acting separately, have a fixed, functional form (see Section 2.1.2). More complex, time-dependent neural network architectures are also possible and are discussed in Section 5.4.2.

4.3.2 Training

Procedure

Visual feedback will play the major role in the training phase. As the objective of this project as well as the entire Cog project is to have the robot imitate a human infant as much as possible, the training methods have been chosen to reflect this.

Several different sounds were recorded with the auditory system, at different azimuthal angles with respect to the head. The signals were divided into two classes, *training* and *validation* signals. The training signals were selected as representatives of the types of naturalistic audio stimuli that the robot may hear. These signals were also chosen for their relatively high degree of associated motion. *Claps* were chosen as a typical short, complex transient signal, while the *spoken vowel*, "ahh," represents sustained, periodic signals.[8] These sounds will be used to successively train the neural network. Sounds of a door slamming from two different directions were recorded as validation signals, which will be used to test the neural network once the weights have been determined.[9]

Each training signal was recorded at three different locations within the visual field of the camera, denoted "left," "center," and "right," all with respect to the head, for a total of 6 training signals.[10] "Left" and "right" positions were at the very edges of the field of view, corresponding to about ± 35 degrees from the center; distances from the head were on the order of four feet. As mentioned above, exact distances and models were not recorded or used. This particular project is not interested in developing an accurate localization system yet, and for now, azimuthal angle will be expressed only in terms of representations of "left," "center," and "right." Performing only coarse localization is consistent with the overall project philosophy of accomplishing simple tasks in complex environments, at least in the beginning; more precise localization will be explored in the future.

One training signal out of the set of six was selected, processed, and presented to the network. The azimuthal angle to the sound source derived from the visual centroid of motion detector was taken as the reference or desired localization angle; the error signal was be computed by subtracting the output of the network, the estimated azimuthal angle, from the desired angle. This signal will then be propagated backwards through the network to adjust the interconnection weights. The presentation-backpropagation process was repeated, up to a maximum of 1000 epochs, until the error signal was sufficiently low (1% sum-squared error). Other training signals were then selected, until all six had been presented to the neural network. The entire procedure was repeated ten times to allow the neural network to assimilate all the training signals.

Each training signal, as well as the two validation signals, were then presented in

[8] This type of signal is more worthwhile to study than sustained sinusoids, as it is can also be used in speech experiments in the future. The associated motion is the mouth movements to produce the vowel sound.

[9] Conveniently, there are two doors located to the left and right of the auditory system.

[10] Due to the very coarse resolution of the chosen azimuthal angle representation, it was not necessary to worry about exact positions and angles.

succession to the neural network and the results (localization outputs) were recorded.

Parameters

The auditory system itself is very flexible in terms of the format of sound acquisition. Parameters such as sampling rate and sample resolution can be varied depending on the particular application. For sound localization, most parameters were empirically determined so as to produce the best performance from the cue extractors and neural network. It was determined that a sampling rate of 22KHz with 8-bit quantizing resolution, and a hidden layer consisting of 16 fully connected nodes were sufficient for adequate performance.[11]

Representation

Using a neural network for signal processing raises the question of at what level should the localization cues be represented. In other words, what form should the inputs and outputs of the neural network take? Research in both image and sound processing have explored both extremes, from taking raw data after minimal subsampling, to pre-processing the raw data and abstracting most of the signal characteristics to very high-level representations (Yuhas et al. 1990).

An intermediate approach was taken for this project, and some amount of pre-processing on the raw sound data was performed resulting in rough localization cues from the binaural differences of various signal properties. The inputs to the network are therefore not raw signals, nor representations of "left," "right," etc., but an intermediate representation. Since each localization cue had different ranges, initial, randomly chosen weights and thresholds took into account the different input ranges.

The form of the outputs turns localization into a classification problem; the three output nodes symbolically represent the three general directions a sound can come from, and have a range of [0,1]. This representation lends itself easily to the integration with the output of the visual centroid of motion detector. Azimuthal "angle" can therefore be determined by taking the maximum of the three output nodes.

4.4 Online Implementation

Having prototyped the design of the pre-processor and neural network off-line, as much functionality was implemented on the DSP as possible given time constraints. Much greater attention must be paid to issues of synchronization, memory, and processing time for implementations on the actual system than for an off-line system with (nearly) unlimited resources. Part of the reason for selecting simple localization cues was to ease the computational demands and the synchronization problems.

[11] A 22KHz sampling rate, with a Nyquist frequency of 11KHz, is fast enough to accurately model, without aliasing, most common sounds, including speech. The corresponding sampling delay of about 46 μs is also short enough to capture interaural time differences while still avoiding oversampling.

4.4.1 Visual Processing

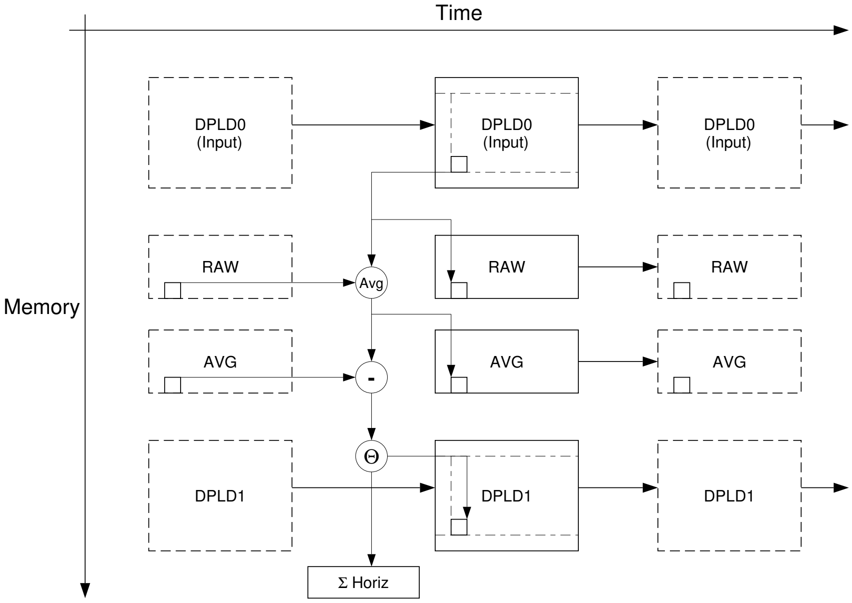


Figure 4-6: Visual Processing Block Diagram

As stated earlier, visual data was processed directly on the DSP from the start. This was to reduce the data throughput through the μ-Cog setup, and because visual processing was very straightforward to implement on the DSP system.

Figure 4-6 presents the overall processing algorithm. *DPLD0* refers to the actual DPRAM that receives the raw image data from the frame grabber. *DPLD1* is the output DPRAM, and is connected to a monitor via the display board. *RAW* and *AVG* refer to internal buffers that store the raw and average image pixel values, respectively, of the *previous* frame. This scheme is necessary to smooth the noisy raw images before processing (subtracting) them. Note that the entire 128 by 128 image is not used in the visual processing; a vertically centered horizontal strip, 32 pixels in width, was used, and the rest of the image was ignored. Even with averaging, the images were still noisy, resulting in spurious nonzero motion pixels. A thresholding function, Θ, was added, and only motion pixels with a value greater than the threshold[12] were actually output, to DPLD1 as well as the horizontal centroid of motion extractor.

4.4.2 Auditory Processing

Figure 4-7 shows a block diagram of the interrupt service routine (ISR) for collecting continuous streams of data from the audio board. When a bank in the AB group is full, an interrupt (ExtInt1) is signaled. The ISR determines which bank caused the

[12]This threshold was also empirically determined, and was set to 25.

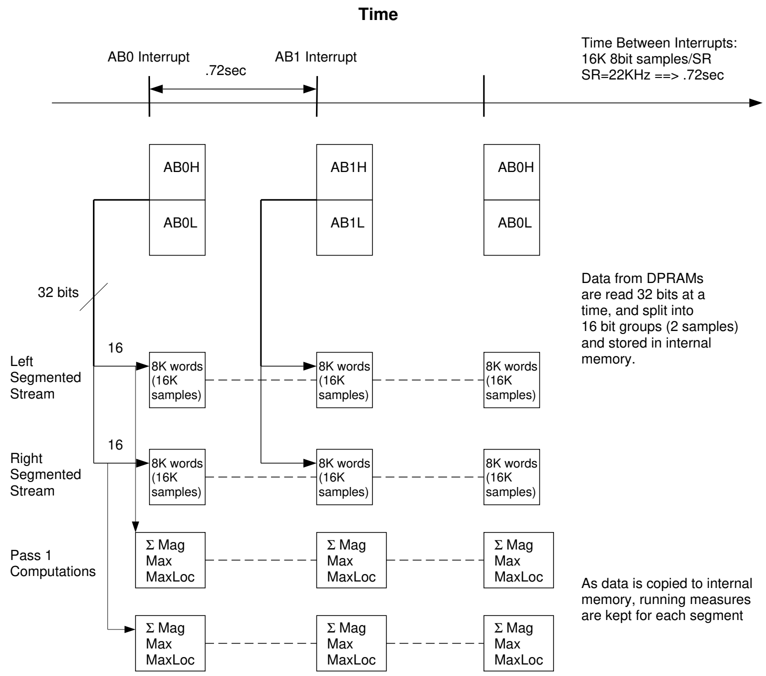


Figure 4-7: Auditory Processing on the DSP System

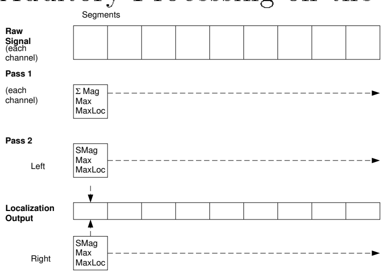


Figure 4-8: Cue Extraction
on the DSP System

interrupt, by examining the interrupt PAL.[13] Data is read 4 samples at a time (32 bits), and stored in the appropriate internal buffers. Reading and writing multiple samples at once increases data throughput and saves memory.[14]

The cue extraction routine (see Figure 4-8) that was implemented on the DSP was a simpler version of the processing performed in Matlab, due to the lack of a routine to compute FFTs.[15] Thus, only time domain signal processing was possible. Pass 1, performed by the ISR, keeps tracking of running measures of the maximum sample

[13]As shown in the DSP System memory map (Figure 3-13), the interrupt PAL is mapped to a range in the global bus. Reading the PAL returns which bank(s) have caused interrupt(s).

[14]All data types in the DSP C language are 32 bits wide; storing one sample per data type would take up twice as much space as this scheme, which packs 2 samples (16 bits total) into each internal memory location.

[15]We are currently in the process of porting a public-domain FFT routine to the C40.

and its location, and the sum of magnitudes while the samples are being transferred to internal memory. Pass 2 takes the updated segment measures of both channels and computes time domain cues based on the differences of the corresponding measures.

While the nature of the processing is very simple compared to that performed off-line, the important point to note is that an interrupt-driven architecture for obtaining raw audio data and processing segments of data was implemented; this architecture can be readily expanded to include the features described in Section 4.2.

4.4.3 Synchronization

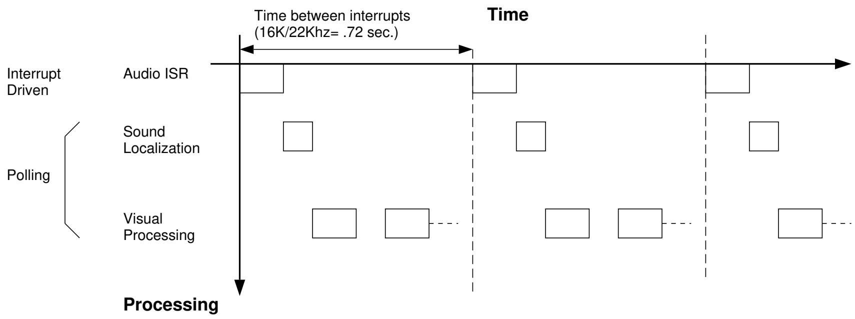


Figure 4-9: Process Synchronization

A combination of interrupt-driven and polling schemes was employed to integrate the auditory and visual processing. To ensure continuous sound processing, the DPRAMs receiving data from the audio board must be processed in a timely manner. Thus, low-level auditory processing routines were interrupt-driven, while less critical visual processing routines were implemented by polling the video DPRAM interrupt. Since it was not necessary to get frequent updates of the centroid of motion, not every video frame needed to be processed, and a fixed number of frames were dropped per second.[16]

[16]Satisfactory results were obtained by dropping as many as 20 out of 30 frames a second.

Chapter 5

Results and Discussion

5.1 Cue Extractions

5.1.1 Training Signals

Two types of signals were used to train the network, a short clapping sound and a voiced vowel, "ahh." Included in this section are representative figures of each of these sounds and their corresponding time and frequency domain cue extractions. Note the "noisiness" of each individual cue for any particular signal; no single cue is accurate in crude localization. What is necessary is to have a neural network "learn" which cue is correct in a particular situation. Figures of the remaining training and validation sounds are presented in Appendix B.

Clapping

Figure 5-1 shows time domain signals of a hand clap from the "left" direction. The left channel (corresponding to the left ear of the Styrofoam head) is shown on top and the right channel on the bottom. Note that it is evident even from visual inspection that the left channel slightly leads the right, and that in general, the magnitudes of the left are greater than the corresponding ones in the right. Time-domain (Figure 5-2) and frequency-domain (Figure 5-3) cue extractions generally agree. (A negative azimuthal angle corresponds to the "left" direction, with respect to the head.)

Spoken "ahh"

Figure 5-4 shows time domain signals of the voiced vowel, "ahh," coming from the "left" direction. Note that the signal has much lower magnitudes in general, as compared with those of the clap signal. Still, the left channel appears to have a higher magnitude content than the right. ITDs are difficult to ascertain from visual inspection, but are present, as shown in Figure 5-5. As expected, frequency domain IIDs shown in Figure 5-6 agree with the actual direction of the source.

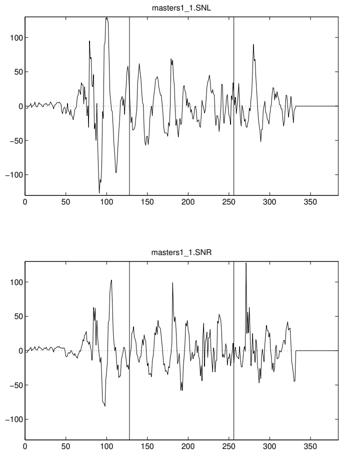


Figure 5-1: Left and Right channels of clap in "Left" direction.

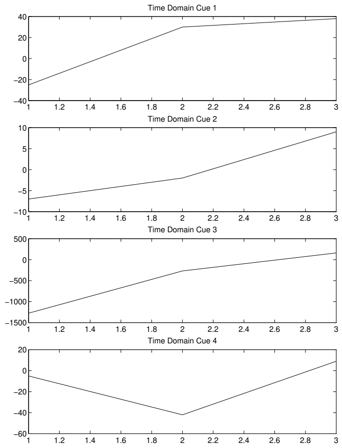


Figure 5-2: Time Domain Cues

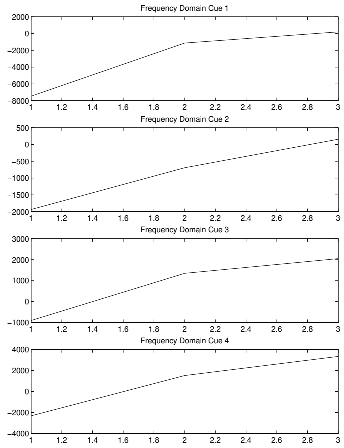


Figure 5-3: Frequency Domain Cues

Door Slam

As an application of the completed neural network, two additional test signals, from the slamming of doors located to the left and right of the head, were recorded and presented to the trained neural network to see how well it localizes a slightly different sound.

As can been seen from figure 5-7, the sound of a door slam is very similar to that of a hand clap; both are short transient signals. In general, both time and frequency domain cue extractors indicate a "left" direction; however, it also appears that both extractor outputs are noisier than those for the clap signal, which could cause the neural network some problems in determining localization angle.

5.2 Visual Input

Figure 5-10 shows the output of a sample run of the visual centroid of motion extractor. (Actual raw and processed images are given in Appendix B.4.) Large values correspond to a centroid in the left portion of the camera's visual field, while small values correspond to a centroid in the right portion. In this particular run, plateaus indicate a clapping motion at the particular direction, while slopes connecting plateaus indicate motion from the subject moving to a new location. The relatively flat plateaus eased the generation of desired responses for the neural network greatly; after manual extraction of interesting sound segments from the raw sound stream, it was discovered that the corresponding centroid of motion was uniformly constant. This was because the final output of the centroid of motion extractor was limited to the three "left," "right," and "center" directions. Thus, for one particular training segment, for example a clap from the left direction, the desired response for the neural network

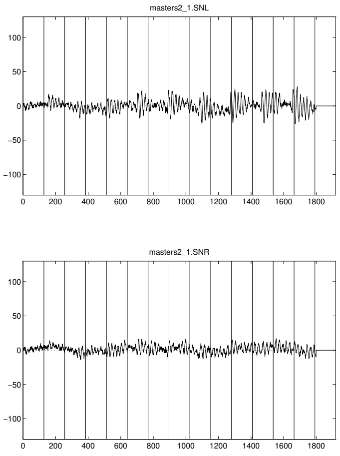


Figure 5-4: Left and Right channels of spoken "ahh" in "Left" direction.

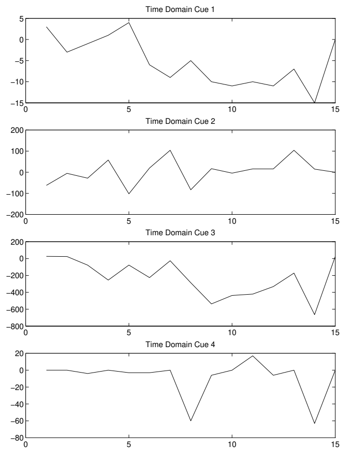


Figure 5-5: Time Domain Cues

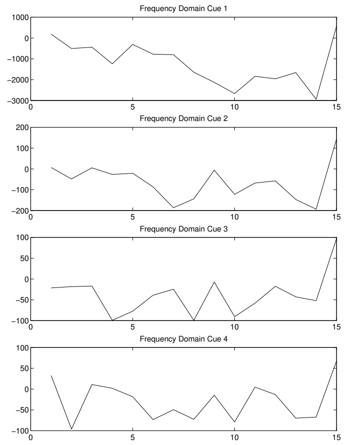


Figure 5-6: Frequency Domain Cues

was a constant "left." Noisy centroid of motion signals would have complicated the training of the network by producing false, spurious desired responses.

5.3 Neural Network Performance

After ten cycles of presenting and training the neural network with the entire training set of six signals, the resulting neural network was presented with the training set as well as the two validation signals to examine its performance in localization.[1]

5.3.1 Training Data

Figure 5-11 shows the localization angle output of the neural network for a clap originating from (top to bottom) the "left," "center," and "right" directions. (The output of the "left" classifier is denoted by a dashed line, the output of the "center" classifier by a dotted line, and the output of the "right" classifier by a solid line.) Note that the network localizes the left and right clap signals well, but has difficulty with the clap from the center direction. This is most likely because it is difficult for the cue extractors to output a zero value, indicating the center direction; the outputs of the cue extractors are usually positive or negative, and very rarely zero.

Figure 5-12 indicates an even worse performance of the network for the voiced "ahh" sounds. The network failed to localize the "ahh" from the left direction correctly.[2] The network response to the centered "ahh" is interesting; it appears

[1] When a fully on-line implementation of this neural network is completed, ongoing training of the network would be possible, instead of a fixed number of cycles.

[2] There appears to be a bias of localizing towards the left due to the nature of training; training

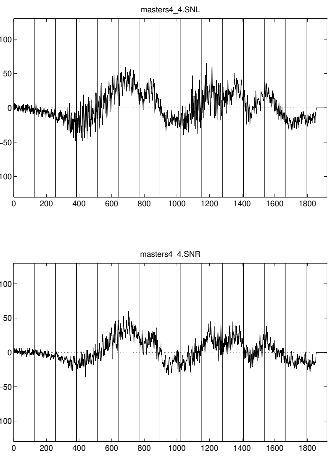


Figure 5-7: Left and Right channels of a door slam from "Left" direction.

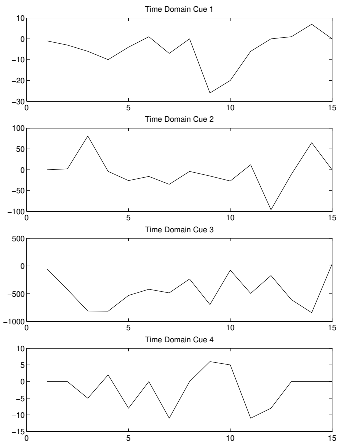


Figure 5-8: Time Domain Cues

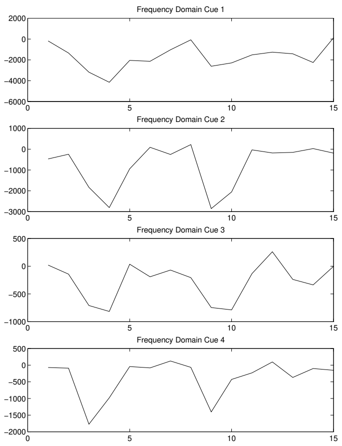


Figure 5-9: Frequency Domain Cues

that there is some competition between the left and right localization cues, which might be expected for a sound coming from the center.

5.3.2 Validation Data

Figure 5-13 also shows that the network performs poorly on the validation signals. This is puzzling, since the time domain signals for a door slamming appeared to be very similar to that of hands clapping. One explanation is that the network was overtrained, and became dependent on particular characteristics of the training set.

5.4 Discussion

5.4.1 Design Issues

A variety of neural networks, each with slightly different parameters, were trained. The results presented here are of the single neural network that seemed to have the best overall performance, and yet it still failed to perform adequately in the validation set.

There is much that can be improved in the immediate future concerning this simple backpropagation network. Of course, more training data of different types of sounds can be collected, and more cycles of the entire training set presentation can be performed. There is a danger of over-fitting the neural network, whereby particular

signals from the right were presented to the network last, and may have caused the bias.

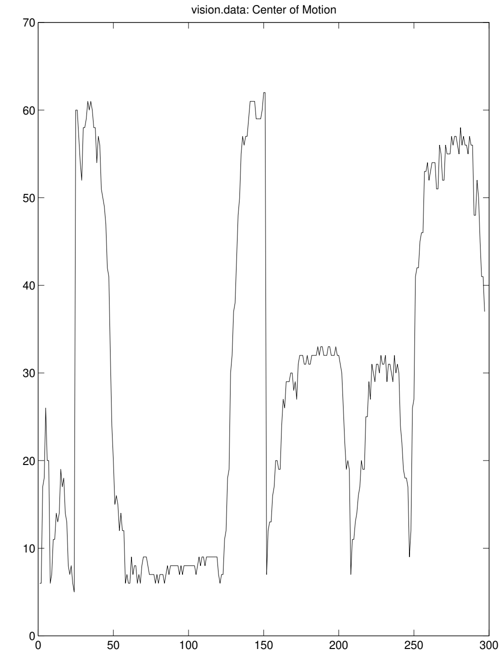


Figure 5-10: Visual Centroid of Motion Detection

characteristics found only in the training set will be learned. The use of a comprehensive validation set to check the progress of the training, ending when performance of the network on the training set exceeds the performance on the validation set, can prevent such over-fitting.

The pre-processing of the signals can also be done differently: different windows[3] and different length segments can be tried; overlapping segments will smooth the output of both the time domain and frequency domain cue extractors; a more rigorous filter bank method can be pursued.

5.4.2 Extensions

Even for a simple static neural network, there are a large number parameters that can all be changed to "tweak" the performance of the neural network. Unfortunately, there are usually no hard and fast rules or an easy function to compute the optimal variables. Like much in the field of neural networks, empirical study coupled with "rules of thumb" are the best means of finding such values. The problem is compounded since changes in one parameter invariably affect others, and so the search space is vast. An extension of this current work is to make many of the parameters, such as window length, type, filter bank division, sampling rate, etc. be *adaptive*, based on the type

[3]Another often used window is the *Hamming* window, defined as $w(n) = 0.54 - 0.46cos(\frac{2\pi n}{N-1})$ for $0 \leq n \leq N-1$ and 0 otherwise. A Hamming window weighs data points near the center of the analysis frame greater than those at either end.

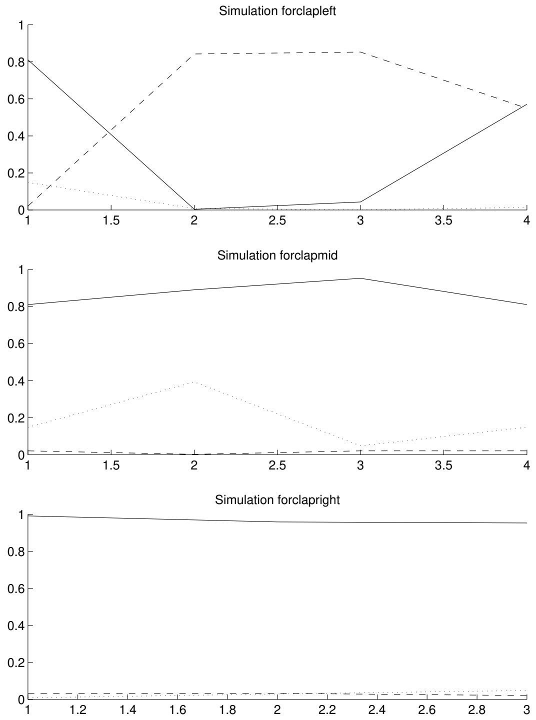


Figure 5-11: NN Output: Clap (from top: "Left," "Center," "Right")

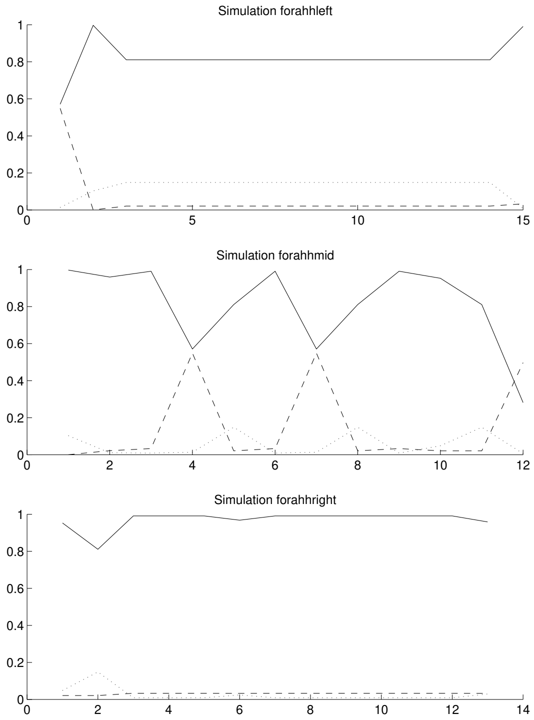


Figure 5-12: NN Output: Spoken "Ahh"

and time varying characteristics of the sound signal.[4]

A major change to the structure of the neural network will be to incorporate time dependence with a sound stream explicitly. Research in neural networks for speech processing have developed time delay neural networks (TDNN) that take into account the time varying changes of speech characteristics in a given utterance (Morgan & Scofield 1991). The TDNN is a MLP whose hidden and output nodes are replicated across time. In other words, the same weights are applied to a series of time-delayed inputs.[5] Training a TDNN is performed by a modified *temporal* backpropagation learning algorithm. TDNNs have been implemented that have better performance in recognizing isolated words than traditional hidden Markov models (Haykin 1994).

There is some biological justification for incorporating time dependence into a neural network for sound localization; it has been noted that a sound that has been already localized recently in the past is expected or predicted to remain in the same general location, helping the human auditory system in determining the present localization angle (Bregman 1990).

[4]Note that there are two separate time dependencies at work here—changes in the localization cues throughout the duration of a particular signal, and changes in the different signals that are heard.

[5]The TDNN architecture can be implemented on an existing MLP by representing each synapse of each neuron in the network as a finite impulse response (FIR) filter (Haykin 1994). In other words, each synapse has a finite memory of past inputs associated with it.

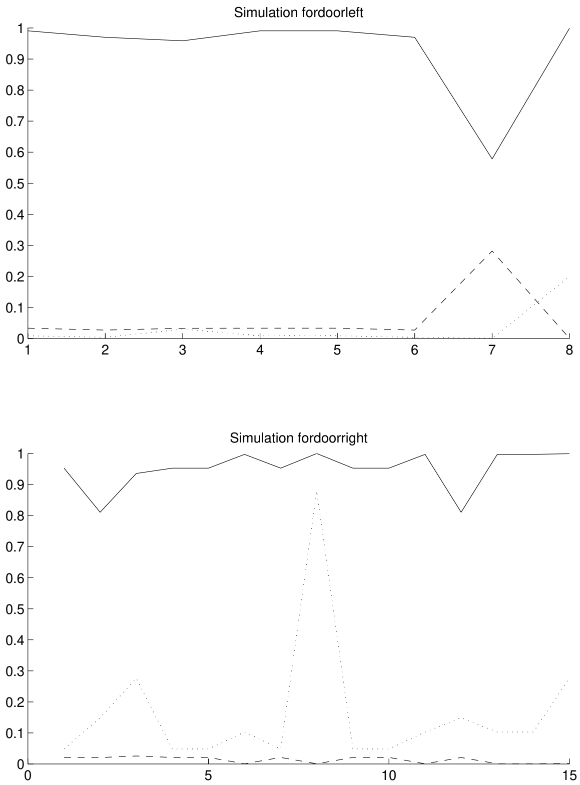


Figure 5-13: NN Output: Door Slam from "Left" (top) and "Right" (bottom)

Chapter 6

Conclusion

6.1 Future Work

With the completion of the hardware and low-level software of the auditory system, efforts can finally be concentrated on improving the performance of the neural network presented in this paper. Once sound localization based on audio-visual integration has been improved, more advanced tasks can be explored, including the characterization and recognition of sounds. For example, when Cog hears a familiar sound, it should be able to localize it and predict what the object producing the sound is, without having to actually see it. Other advanced auditory perception skills will also be studied, including multiple source discrimination (including the so-called "cocktail party effect") and eventually, speech understanding.

As more components of Cog become available, interesting intelligent behavior can be explored; with the completion of the arms and hands, it is conceivable to have Cog: hear but not see a toy rattle, move its head to the general direction of the sound so that it appears in its visual field, fine tune the localization using both visual and audio data, and attempt to grab or swat the rattle with its hand and arm. Just as human infants may find it difficult to grab the rattle on their very first try, Cog may make gross errors initially, but it could learn to better control its motions (hands, arms, and body) based on its senses (vision and hearing). The key point is that the inherent parallel architecture of its "brain" and the modularity of hardware will allow tight coupling of its sensors and effectors, and make such a complicated task possible. Cog is meant to be a testbed for artificial intelligence, specifically the closely coupled phenomena of embodiment and cognition; having a general purpose auditory system that provides robust sound localization adds an extra dimension to the perceptual capabilities of the humanoid robot.

6.2 Conclusion

This thesis presented an auditory system that has been designed and built for a humanoid robot. It also described the software and signal processing architecture that has been developed that will allow the robot to learn how to use a variety of techniques to localize sounds, and to react accordingly. A prototype of a neural network, developed off-line, to perform robust sound localization based on several binaural cues was also presented. As Cog develops and we learn more about embodied cognition, more advanced auditory and visual perception skills will be explored.

Appendix A

Schematic Diagrams

A.1 Microphone Pre-Amplifier

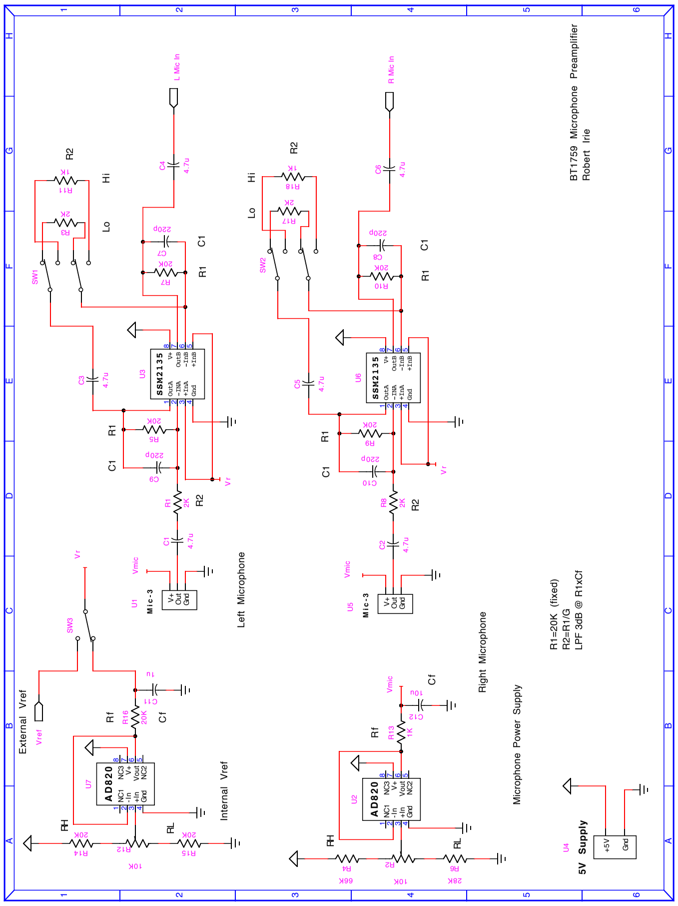


Figure A-1: BT1759 Microphone Preamplifier

A.2 Audio Board

A.2.1 Selected Schematics

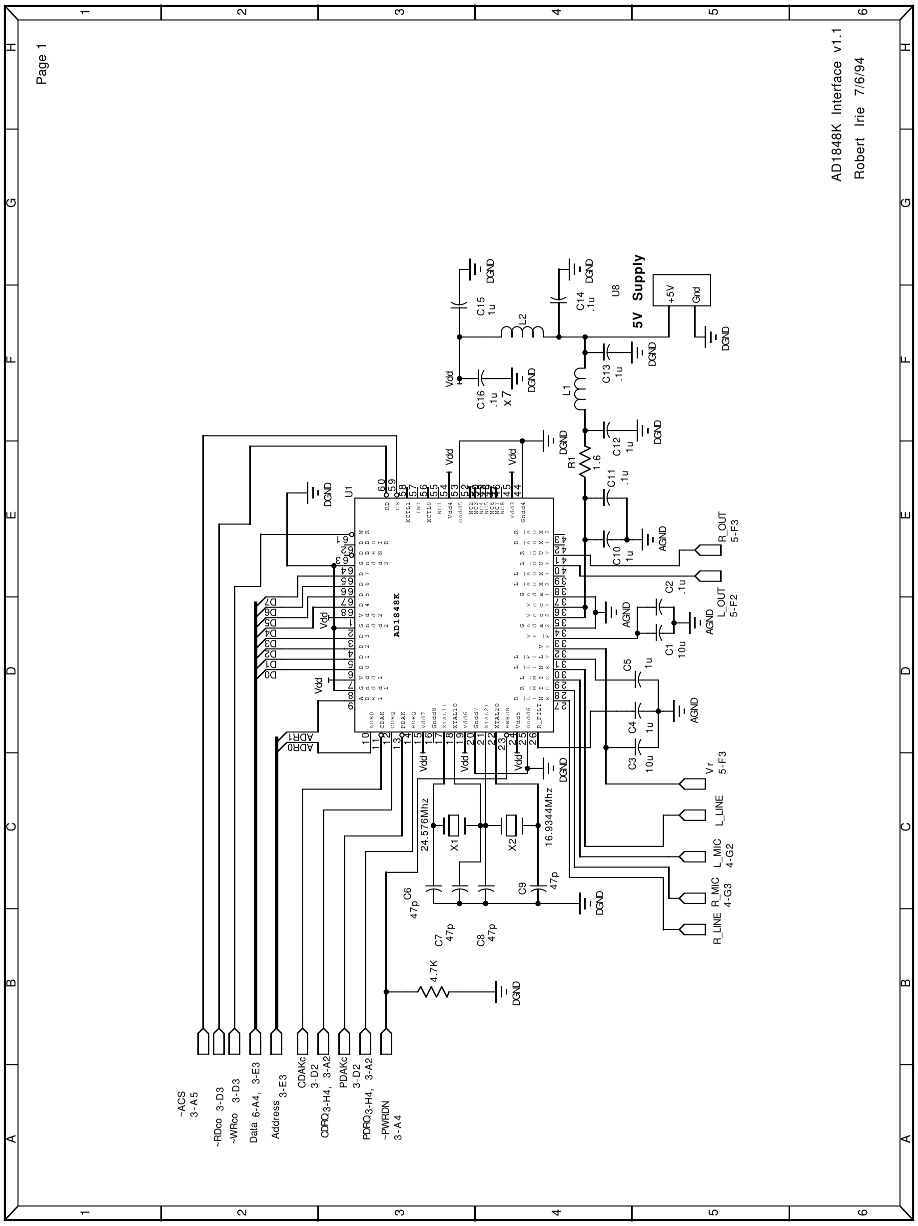


Figure A-2: Audio Board: Codec Interface

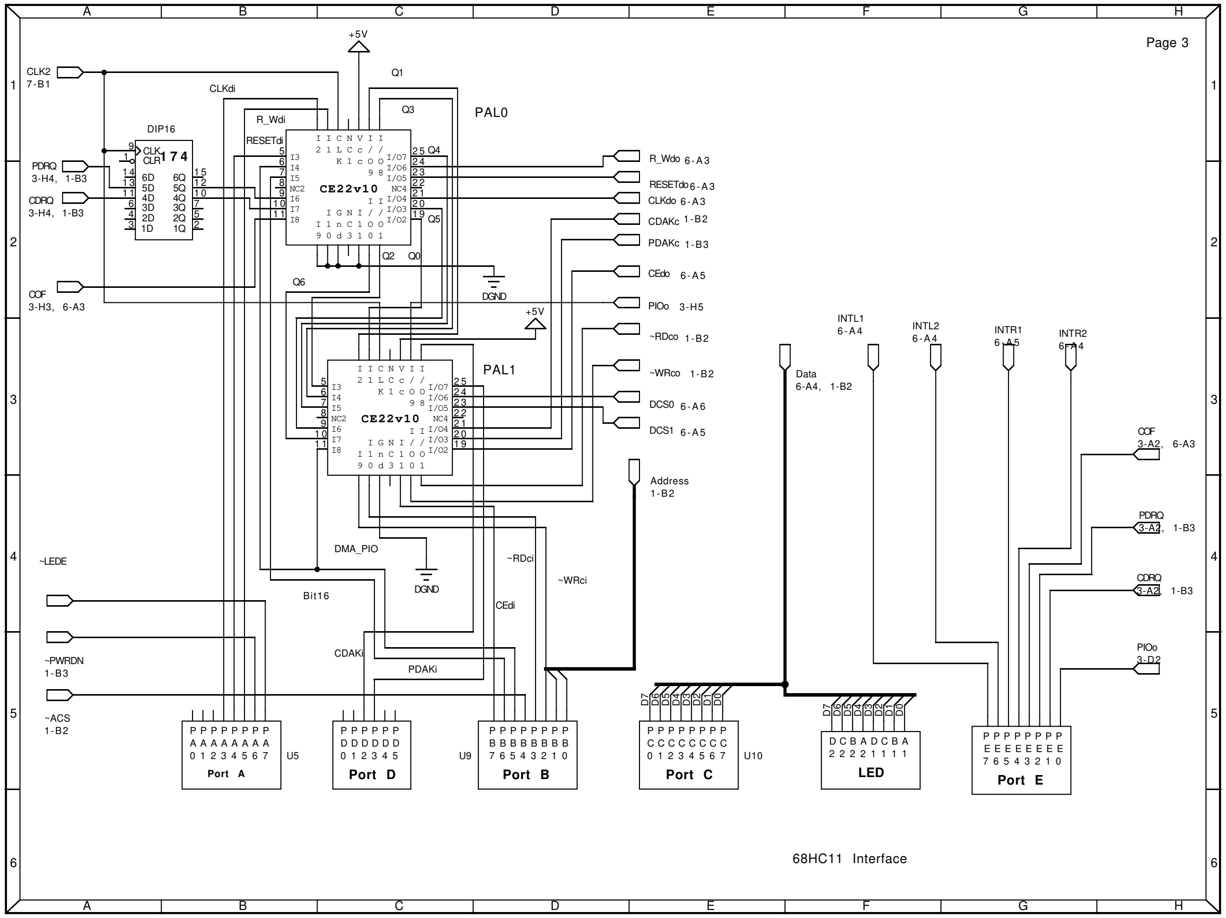


Figure A-3: Audio Board: DMA PAL Interface

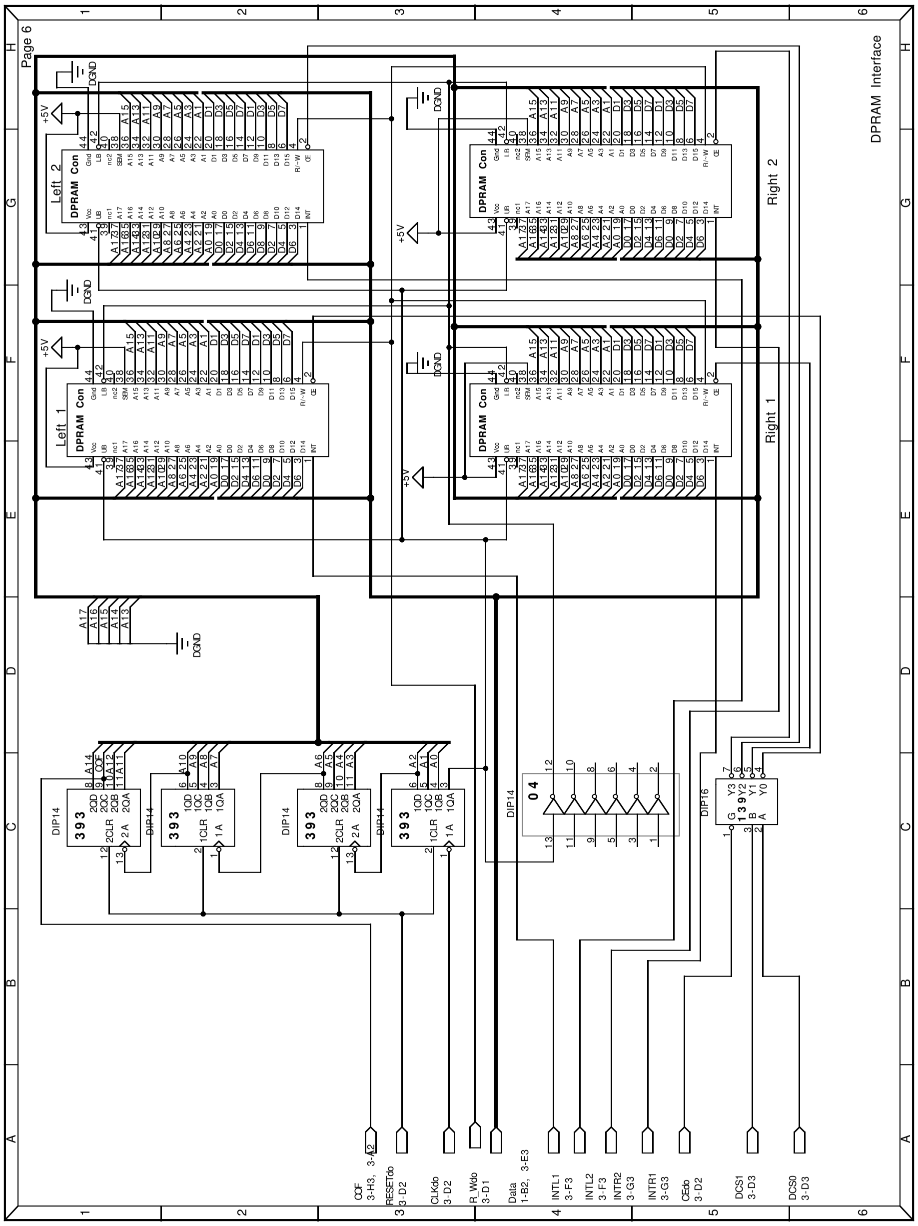


Figure A-4: Audio Board: DPRAM Interface

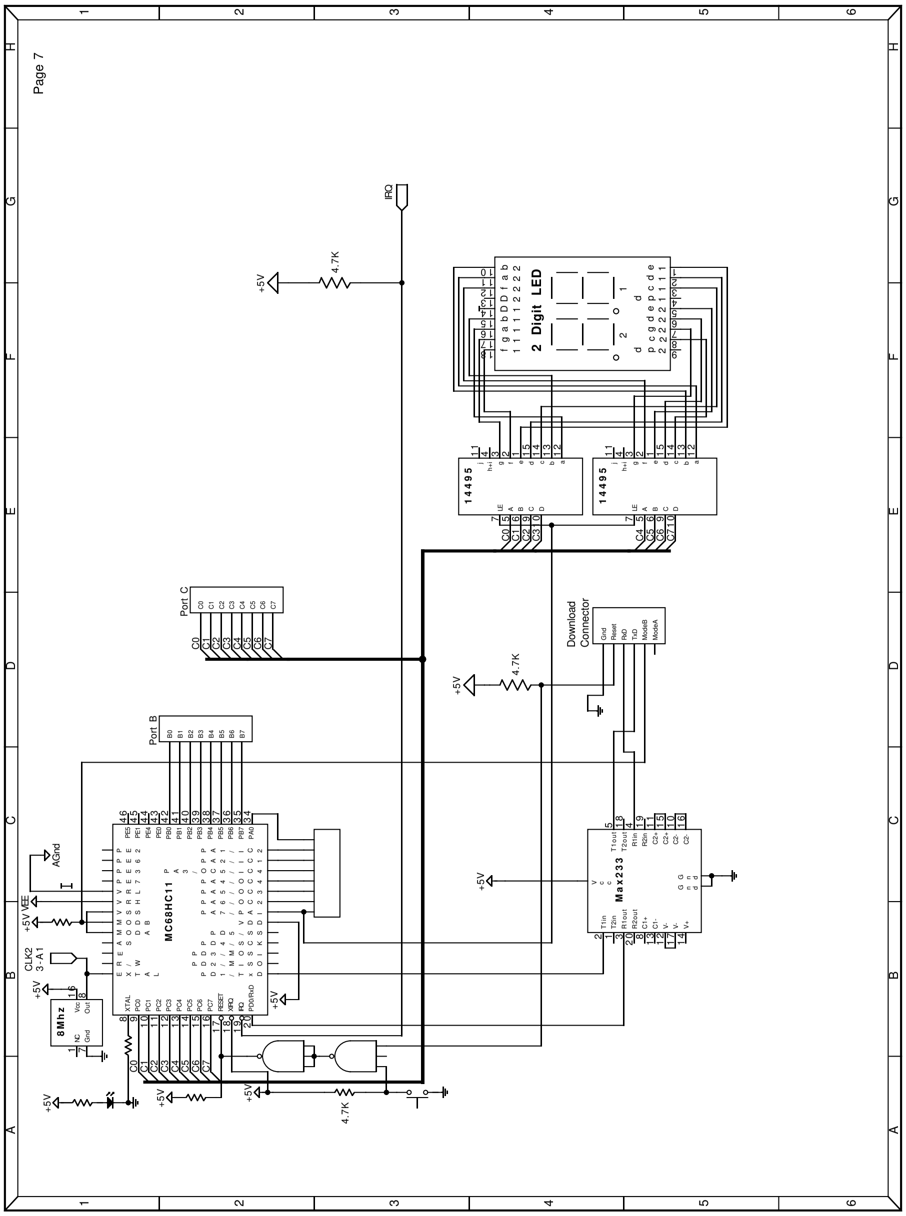


Figure A-5: Audio Board: 68HC11 Controller

A.2.2 Audio PAL State Diagram

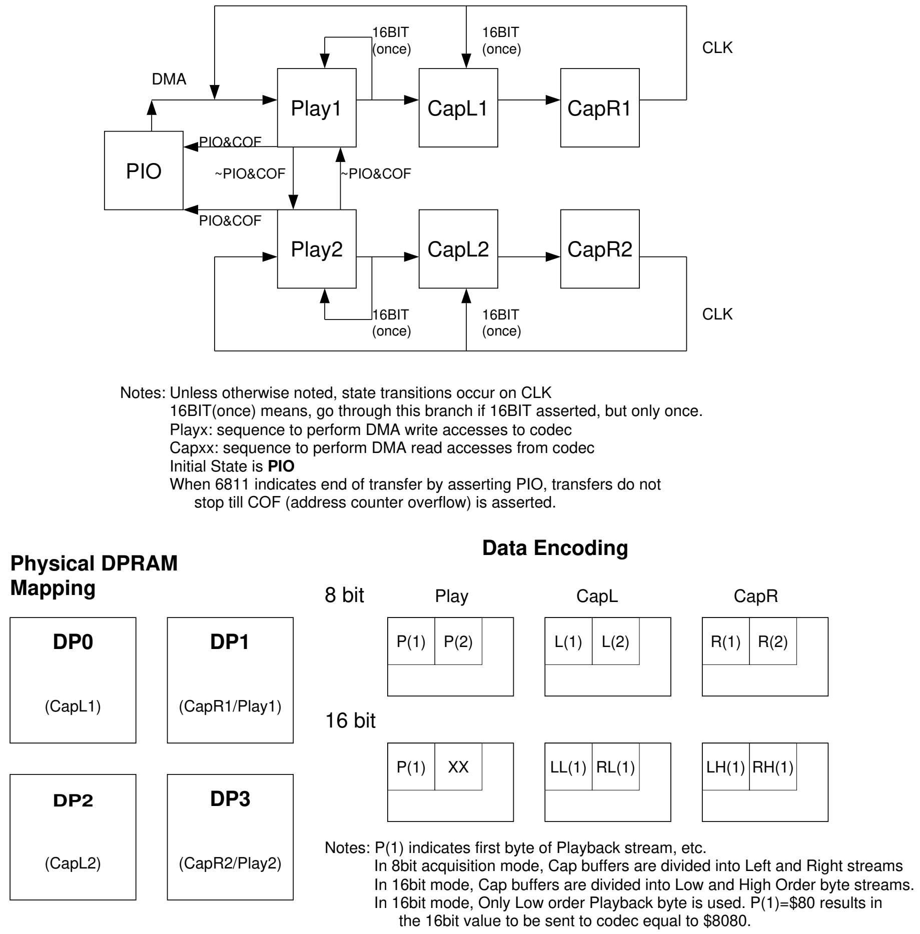


Figure A-6: Audio Board: FSM State Diagram

A.2.3 Codec Information

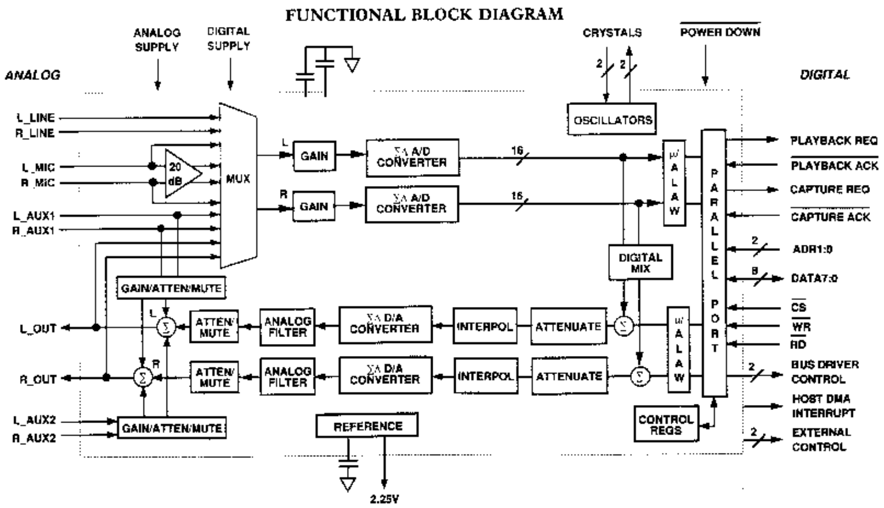


Figure A-7: Codec Block Diagram (AD 1994)

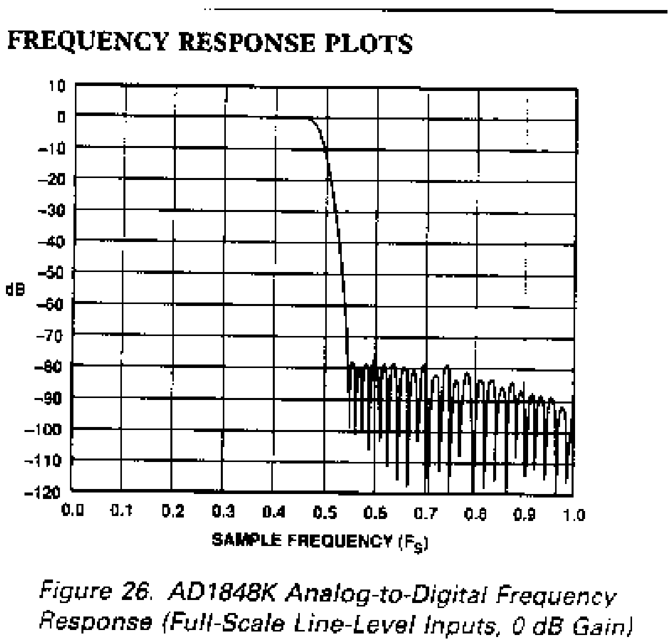


Figure A-8: Frequency Response of ADC (AD 1994)

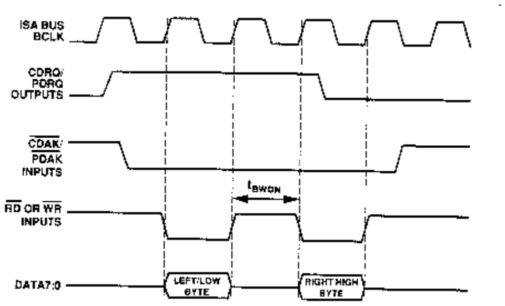


Figure 15. AD1848K 8-Bit Stereo or 16-Bit Mono DMA Cycle

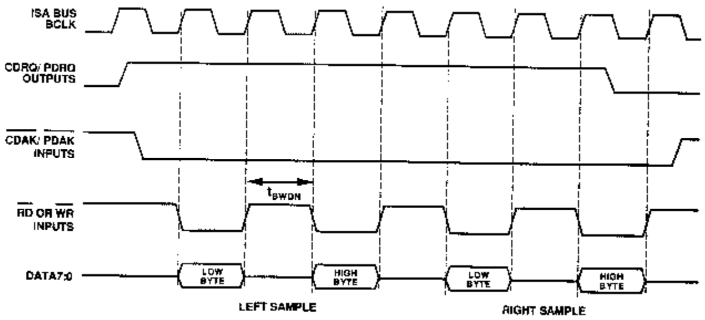


Figure 16. AD1848K 16-Bit Stereo DMA Cycle

Figure A-9: Timing Diagram for DMA accesses (AD 1994)

Appendix B

Training Data

B.1 Clap

B.1.1 "Center" Direction

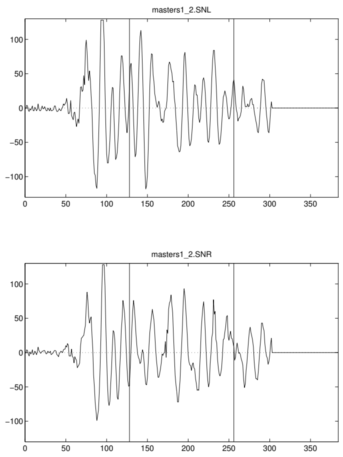


Figure B-1: Clap: "Center"

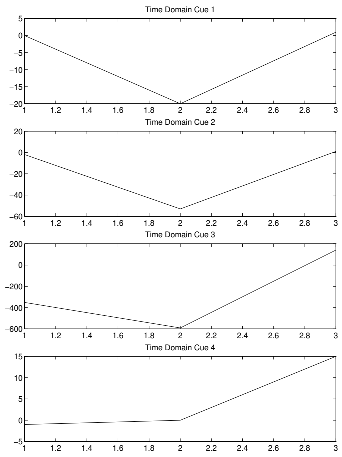


Figure B-2: Time Domain Cues

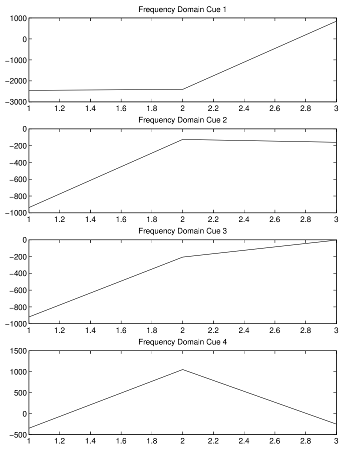


Figure B-3: Frequency Domain Cues

B.1.2 "Right" Direction

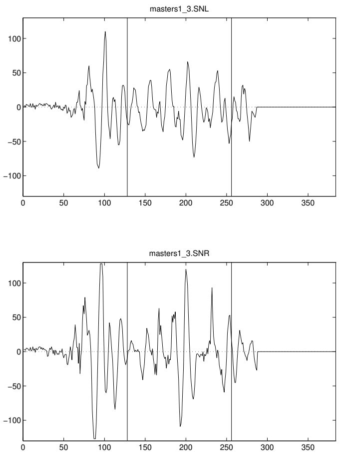


Figure B-4: Clap: "Right"

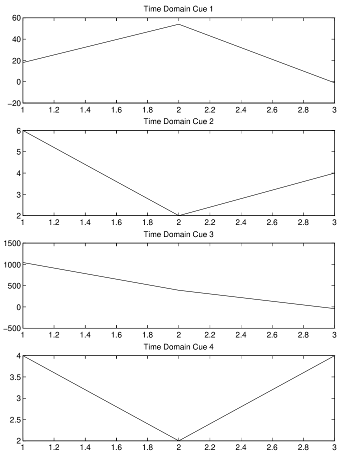


Figure B-5: Time Domain Cues

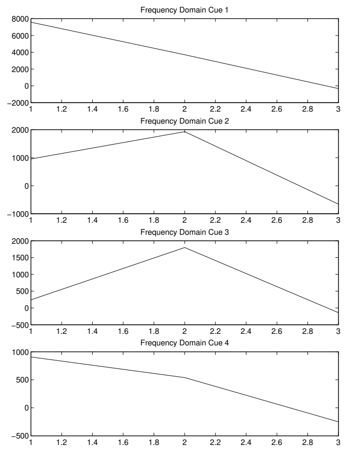


Figure B-6: Frequency Domain Cues

B.2 Spoken "ahh"

B.2.1 "Center" Direction

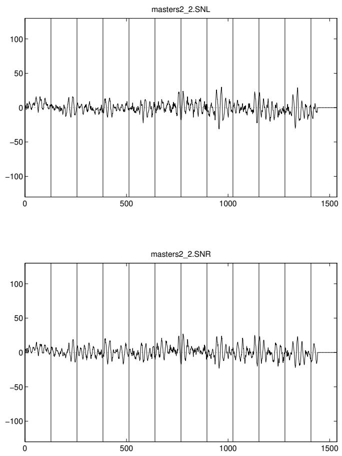


Figure B-7: "ahh": "Center"

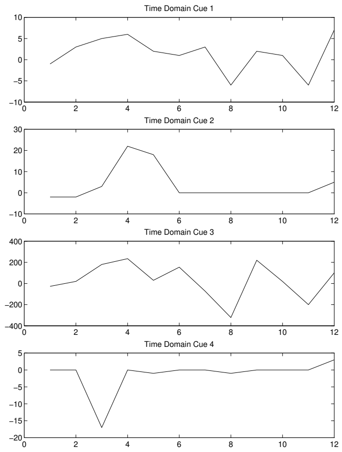


Figure B-8: Time Domain Cues

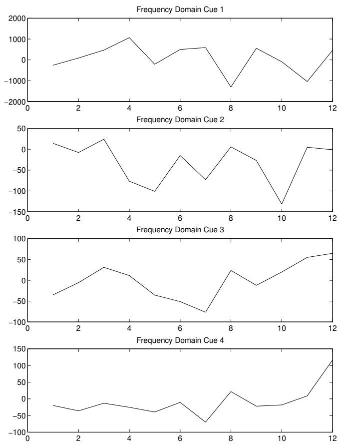


Figure B-9: Frequency Domain Cues

B.2.2 "Right" Direction

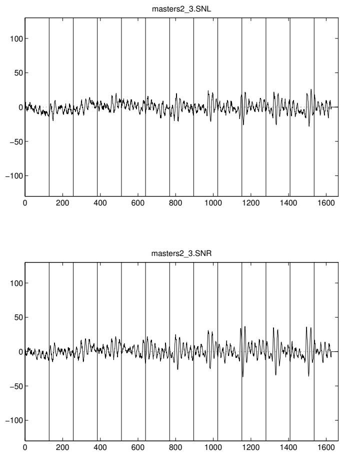


Figure B-10: "ahh": "Right"

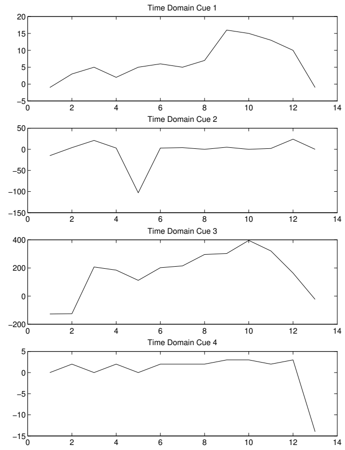


Figure B-11: Time Domain Cues

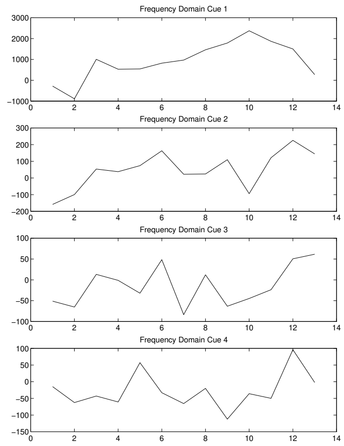


Figure B-12: Frequency Domain Cues

B.3 Door Slam from Right direction

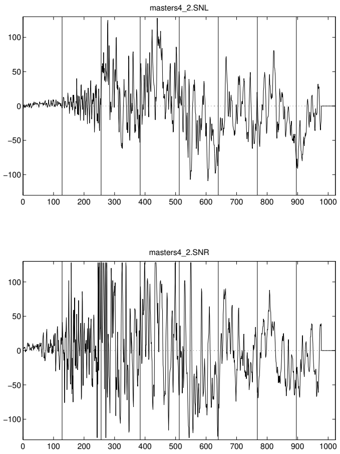


Figure B-13: Door slam: "Right"

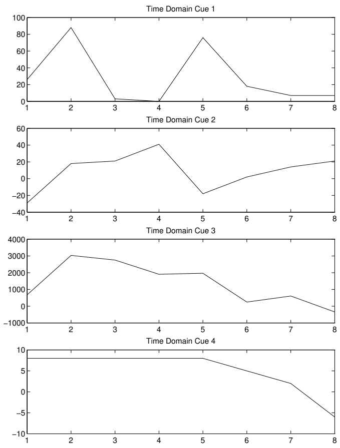


Figure B-14: Time Domain Cues

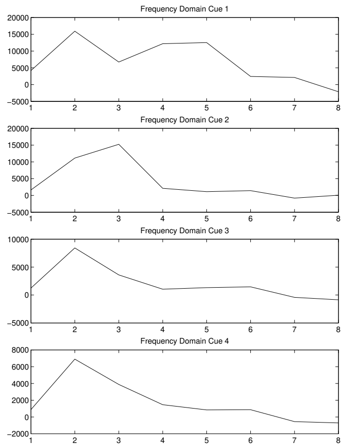
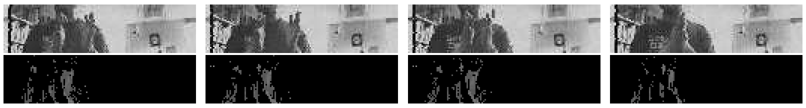


Figure B-15: Frequency Domain Cues

B.4 Visual Processing



The figures on the top row are raw visual images. Beneath them are processed, "motion images." Note the lack of any noise outside the desired clapping motion.

Bibliography

AD (1994), *Analog Devices AD1848K Parallel-Port 16-Bit SoundPort Stereo Codec*, rev. 0 edn.

Beauchamp, K. & Yuen, C. (1979), *Digital Methods for Signal Analysis*, George Allen and Unwin.

Beranek, L. (1970), *Acoustics*, MIT.

Blauert, J. (1983), *Spatial Hearing*, The MIT Press, Cambridge, MA.

Bose, A. (1994), 'Acoustics', 6.312 lectures. MIT.

Bregman, A. S. (1990), *Auditory Scene Analysis*, MIT Press.

Brooks, R. & Stein, L. A. (1994), 'Building Brains for Bodies', *Autonomous Robots* **1:1**, 7–25.

Burgess, D. A. (1992), Techniques for Low Cost Spatial Audio, *in* 'Fifth Annual Symposium on User Interface Software and Technology', ACM, Monterey. (UIST '92).

Durrant, J. D. & Lovrinic, J. H. (1984), *Bases of Hearing Science*, second edn, Williams & Wilkins, Baltimore.

Ferrell, C., Scassellati, B. & Binnard, M. (1995), A Robot for Natural Human-Machine Interaction, Submitted to International Joint Conference on Artificial Intelligence.

Gamble, E. & Rainton, D. (1994), Learning to Localize Sounds Using Vision, ATR Human Information Processing Laboratories, Kyoto Japan.

Haykin, S. (1994), *Neural Networks—A Comprehensive Foundation*, Macmillan College Publishing Co., New York.

Kapogiannis, E. (1994), Design of a Large Scale MIMD Computer, EE Master's Thesis, MIT, Cambridge,MA.

Kno (1973), *BT-1759 Performance Specification.*

Knudsen, E. I. & Knudsen, P. F. (1985), 'Vision Guides the Adjustment of Auditory Localization in Young Barn Owls', *Science* **230**, 545–548.

Marjanović, M. (1995), Learning Maps Between Sensorimotor Systems on a Humanoid Robot, Master's thesis, Massachusetts Institute of Technology.

Matsuoka, Y. (1995), Embodiment and manipulation process for a humanoid hand, Master's thesis, Massachusetts Institute of Technology.

Mills, A. W. (1972), Auditory Localization, *in* J. V. Tobias, ed., 'Foundations of Modern Auditory Theory', Vol. II, Academic Press, New York, pp. 303–348.

Morgan, D. P. & Scofield, C. L. (1991), *Neural Networks and Speech Processing*, Kluwer Academic Publishers.

Muir, D. & Field, J. (1979), 'Newborn Infants Orient to Sounds', *Child Development* **50**, 431–436.

Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T. (1988), *Numerical Recipes in C; The Art of Scientific Computing*, first edn, Cambridge University Press.

Rabiner, L. & Schafer, R. (1978), *Digital Processing of Speech Signals*, Prentice Hall.

Takanishi, A., Masukawa, S., Mori, Y. & Ogawa, T. (1993), Study on Anthropomorphic Auditory Robot—Continuous Localization of a Sound Source in Horizontal Plane, *in* 'Eleventh Japan Robot Society Arts and Science Lecture Series', Japan Robot Society. (in Japanese).

TI (1993), *Texas Instruments TMS320C4x User's Guide*.

von Békésy, G. (1960), *Experiments in Hearing*, McGraw-Hill, New York.

Williamson, M. (1995), Series Elastic Actuators, Master's thesis, Massachusetts Institute of Technology.

Yuhas, B. P., Jr., M. H. G., Sejnowski, T. J. & Jenkins, R. E. (1990), 'Neural Network Models of Sensory Integration for Improved Vowel Recognition', *Proceedings of the IEEE* **78**(10), 1658–1668.