Audio Processing and Loudness Estimation Algorithms with iOS Simulations

by

Girish Kalyanasundaram

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved September 2013 by the
Graduate Supervisory Committee:

Andreas Spanias, Chair
Cihan Tepedelenlioglu
Visar Berisha

ARIZONA STATE UNIVERSITY

December 2013

ABSTRACT

The processing power and storage capacity of portable devices have improved considerably over the past decade. This has motivated the implementation of sophisticated audio and other signal processing algorithms on such mobile devices. Of particular interest in this thesis is audio/speech processing based on perceptual criteria. Specifically, estimation of parameters from human auditory models, such as auditory patterns and loudness, involves computationally intensive operations which can strain device resources. Hence, strategies for implementing computationally efficient human auditory models for loudness estimation have been studied in this thesis. Existing algorithms for reducing computations in auditory pattern and loudness estimation have been examined and improved algorithms have been proposed to overcome limitations of these methods. In addition, real-time applications such as perceptual loudness estimation and loudness equalization using auditory models have also been implemented. A software implementation of loudness estimation on iOS devices is also reported in this thesis.

In addition to the loudness estimation algorithms and software, in this thesis project we also created new illustrations of speech and audio processing concepts for research and education. As a result, a new suite of speech/audio DSP functions was developed and integrated as part of the award-winning educational iOS App 'iJDSP." These functions are described in detail in this thesis. Several enhancements in the architecture of the application have also been introduced for providing the supporting framework for speech/audio processing. Frame-by-frame processing and visualization functionalities have been developed to facilitate speech/audio processing. In addition, facilities for easy sound recording, processing and audio rendering have also been

developed to provide students, practitioners and researchers with an enriched DSP simulation tool. Simulations and assessments have been also developed for use in classes and training of practitioners and students.

DEDICATION

To my parents.

ACKNOWLEDGMENTS

This thesis would not have been possible without the support of a number of people. It is my pleasure to start by heartily thanking my advisor, Prof. Andreas Spanias, for providing me with the wonderful opportunity of being a Research Assistant with his supervision and pursuing a thesis for my Master's degree. I thank him for his support and for providing me with all the resources for fuelling a fruitful Masters program. I also thank Prof. Cihan Tepedelenlioglu and Dr. Visar Berisha for agreeing to be part of my defense committee.

In particular, I would like to thank Dr. Tepedelenlioglu for his support for the first six months of my assistantship on the PV array project. My interactions with him were great learning experiences during the initial phases of my assistantship. I also thank 'Paceco Corporation' for providing the financial support during my involvement in the initial stages of my assistantship. And, I thank Henry Braun and Venkatachalam Krishnan for all their support during this project.

Dr. Harish Krishnamoorthi played a particularly important role by providing me with all the required materials and software during the initial stage of my work on auditory models. The lengthy discussions with him provided me with crucial insight to work towards the initial breakthroughs in the research.

I would like to specially thank Dr. Mahesh Banavar , Dr. Jayaraman Thiagarajan and Dr. Karthikeyan Ramamurthy for all those stimulating discussions, timely help, suggestions and continued motivation which were vital for every quantum of my progress. I also thank Deepta Rajan, Sai Zhang, Shuang Hu, Suhas Ranganath, Mohit

Shah, Steven Sandoval, Brandon Mechtley, Dr. Alex Fink, Bob Santucci, Xue Zhang, Prasanna Sattigeri and Huan Song for all their support.

The best is always reserved for the last. I express my deepest and warmest thanks to my parents for placing their faith in me. Their contributions are beyond the scope of this thesis. I thank all friends and relatives for their support. And I finally thank the grace of the divine for everything.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Algorithms of particular interest in the audio engineering community involve the understanding of properties of the human auditory system and exploiting phenomena pertaining to human hearing in designing audio processing solutions. For instance, the notion of loudness, which is a highly non-linear and subjective phenomenon, is very much dependent on the manner in which the human ear processes sounds [1]. Hence, processing signals based on a reliable measure of loudness requires the incorporation of mathematical models that characterize the notion of loudness as perceived by humans, which in its core, involves modeling the human ear. Such systems can be developed for instance, to automatically control the volume in televisions or computers or mobile phones and tablets in real-time with minimal user involvement. As is described later in this chapter, the involvement of auditory models leads to rising computational complexity. In such scenarios, it is wise to improve the computationally efficiencies of the algorithms while ensuring that the end goal of achieving the desired performance is not compromised with. The work presented in this thesis explores avenues for improving the estimation of loudness using human auditory models.

With the rapid proliferation of portable devices with ever increasing processing power and memory capacity, the need to develop applications exploiting these platforms is rising in parallel, motivated by increasing benchmarks for as well as expectations of user experience. The design of efficient low-complexity algorithms for audio and speech processing applications is of particular importance in this context. For instance, modern mobile phones and tablets have speech recognizing capabilities based on existing state of

the art speech recognition algorithms. This technology is used for giving voice commands to the device and performing functions such as internet search by converting speech to text using standard algorithms. A number of mobile applications have emerged, which make use of signal processing algorithms for novel audio processing techniques. For instance, “SoundHound” is an app that performs content-based music recognition by accepting a sample music input from the microphone [2]. The app has the capability to identify a song even if the user simply hums or sings the tune of the song. Similarly, audio processing techniques combined with sophisticated user interfaces have resulted in a number of apps on mobile devices for music composition, recording and mixing such as GarageBand [3] and Music Studio [4].

The ability to perform advanced DSP techniques with intensive computation can be challenging on a mobile platform. The requirement to deliver better services in mobile devices is an important motivation for developing efficient algorithms to reduce the number of computations so that the algorithms run fast and consume less power in the processor, hence reducing load on the battery life, which is crucial. In other scenarios, algorithms exist which are computationally demanding to the extent that it is impossible to implement them in real-time in most modern computers and portable devices. It would doubtlessly be beneficial to reduce the complexity of such algorithms and render them implementable in real-time on both desktop computers and mobile devices.

The ability to implement sophisticated DSP algorithms on mobile devices can also be exploited in education. Digital signal processing (DSP) techniques are strongly motivated by many popular speech and audio processing applications in which they are used. Hence, illustration of these DSP techniques shown along with their underlying

motivation would strongly benefit students specializing in this field. Advanced signal processing concepts such as time-frequency representation of signals, concepts related to speech coding such as linear predictive coding and line spectral pairs, etc. are widely used in existing DSP systems. Effective illustration of these concepts generally requires the use of examples and visualizations in order to help students better understand them. In addition, the presence of interactivity in such educational applications would provide students with an enriching learning experience.

This thesis is comprised of two parts. In the first part, efficient algorithms are reported which reduce the computational complexity of human auditory models-based loudness estimation. The second part of this thesis discusses the incorporation of audio DSP algorithms into iJDSP [5], an interactive educational mobile application for DSP education on the iOS platform. The use of interactive user interfaces in iOS devices, exploiting features such as touch-screen technology in visually illustrating rich graphical examples of fundamental speech/audio processing concepts would aid in better student understanding of these concepts. Elaborated in the section below are the motivations behind developing efficient algorithms and focusing on their educational value on mobile platforms.

1.1 Human Auditory Models Based Processing and Loudness Estimation

Perceptual models characterize and quantify the relation between auditory stimuli and associated hearing sensations. Specific phenomena can be observed experimentally through the presentation of auditory stimuli to human subjects, and subsequently modeled as functions of a set of parameters characterizing the stimuli. Since psychoacoustic phenomena are cognitive inferences produced in the brain upon reception of electrical

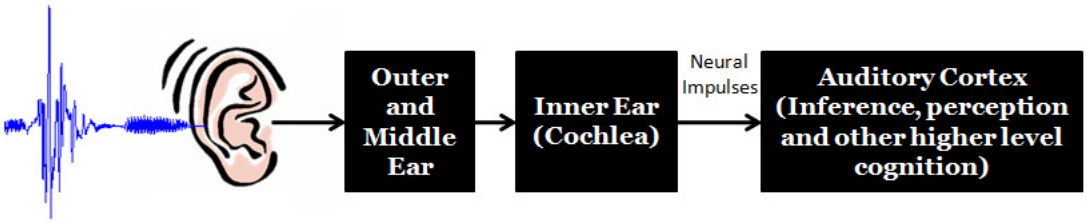


Figure 1.1: Overview of auditory processing. Sound is converted by the human auditory system to neural impulses, which are transmitted to the auditory cortex in the brain for higher level inferences.

signals generated by the human auditory system as responses to auditory inputs (Figure 1.1) [6], it is not possible to directly measure the physiological activity in the brain representing these phenomena. Hence, in experiments designed to study such phenomena, listeners record their responses to subjective questions after hearing test stimuli. Such experiments are referred to as psychoacoustic experiments. Based on the results of psychoacoustic experiments, perceptual models are developed.

Several perceptual models have been developed for the purpose of characterizing specific sensations [1,7,8,9,10,11,12]. For instance, models predicting the phenomenon of auditory masking in signals have been extensively used in speech and audio coding applications [13]. Algorithms used in such audio coders encode the signal with as less bits as possible while ensuring transparent signal quality, i.e., ensuring that the decoded audio is perceptually indistinguishable from the original audio. This is achieved by ensuring that the power of the quantization noise introduced during the encoding process is maintained below certain threshold levels so that they cannot be perceived by the ear. These threshold levels, referred to as “masking thresholds”, are computed by underlying perceptual models they are employed by the algorithms. These masking thresholds represent the important property of the ear to mask certain components of the sound.

Other applications of auditory models include speech enhancement and speech quality assessment metrics [13].

An important application involving extensive use of human auditory models, which is dealt with in this thesis, is the estimation of loudness. Loudness is the measure of perceived intensity of sound. It is a psychological phenomenon. Different methods and units have been proposed to quantify loudness [14]. The Equal Loudness Contours (ELC) reported by Fletcher in [1] was among the earliest attempts to characterize the non-uniform sensitivity of the human auditory system to frequencies in the spectrum. From these contours, the A, B and C Weighting Curves were derived to weight the spectra of signals according to the human auditory system's sensitivity to them and obtain a measure of loudness from them (see ANSI S 1.4-1983 [15]).

More sophisticated models model the human ear as a bank of a large number of highly selective bandpass filters [16,17,18,19]. Some of the popular models for these auditory filters are the Gammatone filters [20], the Gammachirp filters [21], the dual resonance nonlinear filter (DRNL) [22], and the rounded exponential filters [23]. The energy of the signal within each filter band gives knowledge of the spectrum of the signal in the auditory system, which is used to compute the perceived loudness per filter band (called the auditory pattern), and the total loudness. The Moore and Glasberg Model has shown to perform well with a variety of auditory inputs, giving accurate measures of loudness [19]. But a major drawback of these models, including the Moore and Glasberg Model, is the computational load in calculating the loudness. The filtering of signals through the bank of filters modeling the ear is computationally expensive and uninformed reduction of computational complexity can lead to erroneous estimates of loudness.

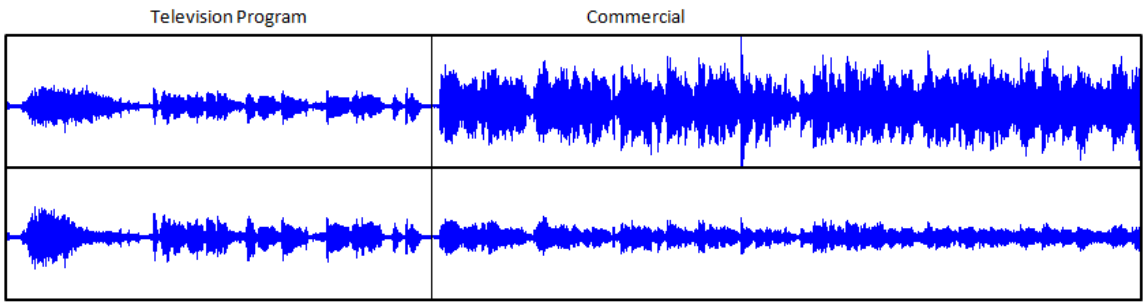


Figure 1.2: The top plot showing the drastic change in intensity level at the onset of a commercial break. The bottom plot shows the signal corrected for uniformity in the intensity.

Hence, the estimation procedure is to be analyzed further to explore the possibility of deriving computationally efficient approximations of the algorithm by exploiting redundancies.

Low complexity loudness estimation procedures exploiting the properties of the auditory filters have been proposed in [24,25]. This method reduces complexity by choosing a subset (also referred to as 'pruning') of the bandpass filters modeling the ear and estimating the output of the rest. The choice of the subset of filters depends on the signal's spectral content. In this thesis, methods to better choose the subset of filters for improved error performance are explored.

1.2. Computation Pruning for Efficient Loudness Estimation

In [25], a computation pruning mechanism was proposed for fast auditory pattern estimation by pruning the set of filters in the auditory filter bank. The algorithm achieves huge computational savings and results in reasonable accuracy of loudness estimates for a variety of sounds. But the accuracy drops with tonal sounds such as music from instruments like the flute, and synthetic sounds with sharp spectral peaks. The

performance of the pruning scheme can be significantly improved with minimal addition to the computational complexity. In this thesis, an enhanced pruning scheme is proposed which is aimed at reducing limitations of the above mentioned pruning scheme.

1.3. Automatic Control of Perceptual Loudness

Automatic loudness control is an immediate application of a loudness estimation algorithm. Human auditory models can be used to control the perceived loudness of a stream of audio such that the perceived tonal balance of the signal is preserved but the loudness is maintained at a constant level. Such algorithms can be of particular use in television, where sudden increases of decreases in the loudness during transitions of programs and at the onset or end of commercial advertisements (Figure 1.2). Other applications would be in media players where consecutive tracks in a playlist are mastered at different volume levels. This can cause one song to be much louder than the other and otherwise require manual volume adjustment. Another application of volume control can be in telephony during sudden increase in the noise or the loudness of the speaker. A demonstrative real-time loudness control system using the Moore and Glasberg model is presented in this thesis. This system is implemented in Simulink.

1.4. Interactivity in Speech/Audio DSP Education

Digital signal processing techniques are strongly motivated by many popular speech and audio processing applications in which they are used. Hence, the illustration of these DSP techniques shown along with their underlying motivation would strongly benefit students specializing in this field. Advanced signal processing concepts such as time-frequency representation of signals, concepts related to speech coding such as linear predictive coding and line spectral pairs, etc. are widely used in existing DSP systems.

Effective illustration of these concepts generally requires the use of examples and visualizations in order to help students better understand them.

Interactive speech and audio processing visualizations for education were developed as part of J-DSP, an interactive web-based Java applet for performing DSP laboratories online on desktop PCs [26,27,28]. J-DSP offers a block diagram based approach to the creation of DSP simulation setups. Users can easily set up simulations by defining signal processing operations as a network of DSP functional blocks with editable parameters. Through specific blocks, the application provides interactive visualizations of various properties of the designed system and its outputs in the form of plots.

In recent years, mobile devices have been identified as powerful platforms for educating students and distance learners. For instance, in a recent study on using the iPad in primary school classrooms, it was found that iPads are effective due to their mobility and that they enhance student engagement [29]. Studies indicate that mobile tools have several advantages in teaching a broad range of subjects, from the arts, to language and literature, to the sciences [30,31,32]. With greater technology outreach, it has become possible for students to get easy access to educational software on mobile platforms. This enables them to reinforce the lessons learnt in a classroom, and even perform homework assignments.

MOGCLASS, a mobile app for music education through networked mobile devices was used successfully in collaborative classroom learning for children [33]. Other examples of existing educational tools on mobile platforms include StarWalk from Vito Technology Inc. for astronomy [34], the HP 12C Financial Calculator for business [35], and Spectrogram for music [36].

Mobile education paradigms are also being extended to higher education [37]. Touch Surgery is an educational app designed for iPhones/iPads, which simulates surgical operations on the screen of the mobile device, with students using finger gestures on the touch screen to interactively perform the surgery in the simulator [38]. Mobile apps are also available for Digital Signal Processing education on mobile devices. For instance, MATLAB Mobile, which is the mobile version of MATLAB, can be used for DSP simulations, which performs simulations through a conventional command line interface [39]. However, it does not provide a sophisticated GUI and relies heavily on a stable internet connection, as it performs all intensive computations remotely on the MathWorks Cloud or a remote computer [40].

Conventionally, students are taught Digital Signal Processing (DSP) with the description of systems as block diagrams. A system can be represented as a black box taking a set of inputs, processing those inputs, and producing a set of outputs. This simple representation of a system enables construction of systems from blocks of simpler systems by connecting them suitably. Software such as Simulink and LabVIEW use such a graphical approach for system design, in contrast to software such as MATLAB, Mathematica and GNU Octave, which contain a rich set of functions for simulating DSP systems, but provide a command line interface and run scripts to simulate systems [41,42,43,44,45].

An important consideration in educational software for mobile platforms is the effective utilization of the touch-screen features of devices to provide a hands-on and immersive learning experience. In particular, touch-screen capabilities can be of immense advantage for graphical programming, as users can simply place functional blocks on the

Figure 1.3: The iJDSP interface, as seen on an iPhone.

screen with their fingers and make connections between them. This paradigm of learning DSP is more intuitive than typing scripts to build a system. Graphical display also presents the system in a pictorial manner, enabling easy comprehension of the system.

iJDSP is an educational mobile application tool developed for iOS devices, providing graphical, interactive, informative and visually appealing illustrations of concepts of digital signal processing (DSP) to undergraduate and graduate students (Figure 1.3). iJDSP provides an engaging GUI and supports a block diagram based approach to DSP system design to create DSP simulations in a highly interactive user interface, exploiting the touch-screen features of mobile devices such as the iPad or the

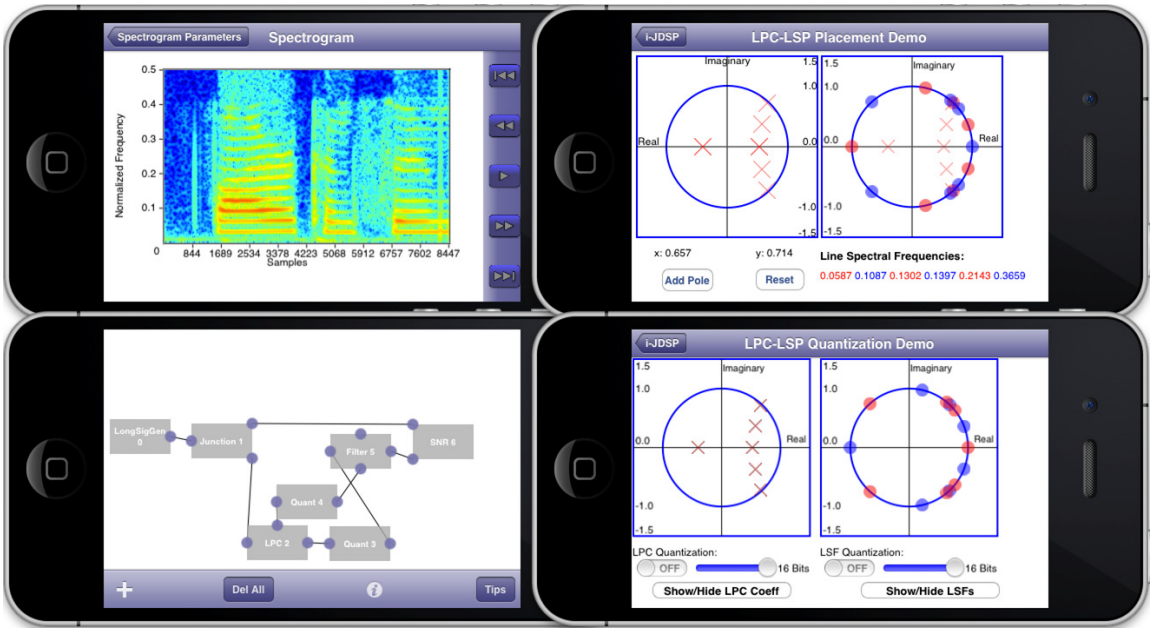


Figure 1.4: Sample screenshots showcasing the suite of speech and audio processing functions created as part of iJDSP.

iPhone. It is computationally light and does not require an internet connection to perform simulations. The application's interface is native to Apple mobile devices. Hence, users of these devices can familiarize themselves and navigate through the application with ease.

A rich suite of functions is available in iJDSP to build and simulate DSP systems, including the Fast Fourier Transform (FFT), filter design, pole-zero placement, frequency response, and speech acquisition/processing functions. Furthermore, a graphical interface that allows users to import and process data acquired by wireless sensors has been developed in iJDSP. The user interface and the color scheme of the software have been designed with end-user engagement as the goal.

The second component reported in this thesis is a set of functions in iJDSP which illustrate basic techniques to represent, analyze and process speech and audio signals,

such as linear predictive coding and line spectral pair representations which are popularly used for speech coding in telephony, the spectrogram, which is a widely used time-frequency representation for the analysis of time-varying spectra of speech and audio signals, and phenomena of psychoacoustics such as masking and perception of loudness (see Figure 1.4). These functions would then be suitably used by graduate students specializing in DSP for understanding concepts behind processing speech and audio. To that end, exercises have been formulated, appropriately using these blocks to create simulation setups that require the students to accomplish specific tasks in iJDSP, which are meant to reinforce concepts taught earlier in a lecture session.

1.5. Contributions

The contributions in thesis can be grouped into two parts. The first part explores the use of the Moore and Glasberg Model in estimating perceptual loudness and its application in real-time loudness control. The following are the key contributions in the first part of the thesis:

- Proposed a pruning mechanism for the Moore and Glasberg model-based loudness estimation algorithm.
- Performed algorithm complexity analyses to highlight advantages of the proposed pruning schemes.
- Implemented real-time automatic loudness control using the Moore and Glasberg model.

The second part of this thesis focuses on expanding the capabilities of iJDSP to incorporate functionality to illustrate some key speech and audio processing concepts to students. The following were achieved in the second part of the thesis:

- Expanded the functionality of iJDSP to allow processing long signals such as speech and audio, and allow frame-by-frame processing and visualization of plots.
- Developed a suite of functions demonstrating certain fundamental concepts related to signal analysis, and speech and audio processing concepts, with interactive user interfaces exploiting the multi-touch features of the iOS devices.
- Created laboratory exercises for illustrating specific speech and audio DSP concepts.

1.6. Organization of Thesis

This thesis is organized in the following manner, with a description of the proposed loudness estimation algorithms and the exploration of real-time loudness control covered by the first few chapters, followed by the description of the functionalities introduced in iJDSP and enhancements made to the software in the rest of the chapters. Chapter 2 in this thesis gives a brief introduction to the functioning of the human auditory system and essential concepts related to psychoacoustics. Chapter 3 elaborates on loudness estimation, giving an overview of different loudness estimation methods proposed in the literature and elaborated upon the Moore and Glasberg model, which is the model adopted for the research reported in this thesis. Computation pruning schemes to speed up loudness estimation are discussed in Chapter 4 along with simulation results. In Chapter 5, the implementation of real-time loudness control using the Moore and Glasberg model is described. Chapter 6 expands upon the enhancements added to iJDSP, the mobile educational app, and the set of speech and audio processing

functions developed as part of the software. The software assessments conducted for iJDSP to evaluate its qualitative aspects as an educational application for speech and audio processing are also presented in Chapter 6. Chapter 7 summarizes, draws conclusions and discusses the scope for future work.

Chapter 2

PSYCHOACOUSTICS AND THE HUMAN AUDITORY SYSTEM

Psychoacoustics is the study of the psychological phenomena pertaining to the perceptions of sound. The human auditory system, which receives auditory stimuli from the ear, processes it through different stages in the organ system and produces neural impulses. These neural impulses are transmitted to the brain through a bundle of nerve fibers. All auditory perceptual phenomena are the results of the interpretation of these nerve impulses by the brain. An overview of the functioning of the human auditory system and the principles of psychoacoustics is presented in this chapter.

2.1. The Human Auditory System

A diagram of the human auditory system structure is shown in Figure 2.1 [46]. The auditory system could be divided into three main stages, namely the outer ear, middle ear and the inner ear, which are described below.

2.1.1. The Outer Ear

The outer ear consists of the pinna and the external auditory canal, which collect the sound waves and transmitting it to the ear drum, which is located at the end of the auditory canal. The auditory canal is a narrow tube about 2 centimeters long, which has a resonant frequency at about 4 kHz. This results in a higher sensitivity of the ear to frequencies around 4 kHz [47]. However, due to the same reason, the ear has a higher susceptibility to pain and damage due to high intensity sounds at these frequencies.

2.1.2. The Middle Ear

The outer ear receives sound pressure through the oscillation of particles in the air. On the other hand, the inner ear medium is made up of a salt-like fluid material and

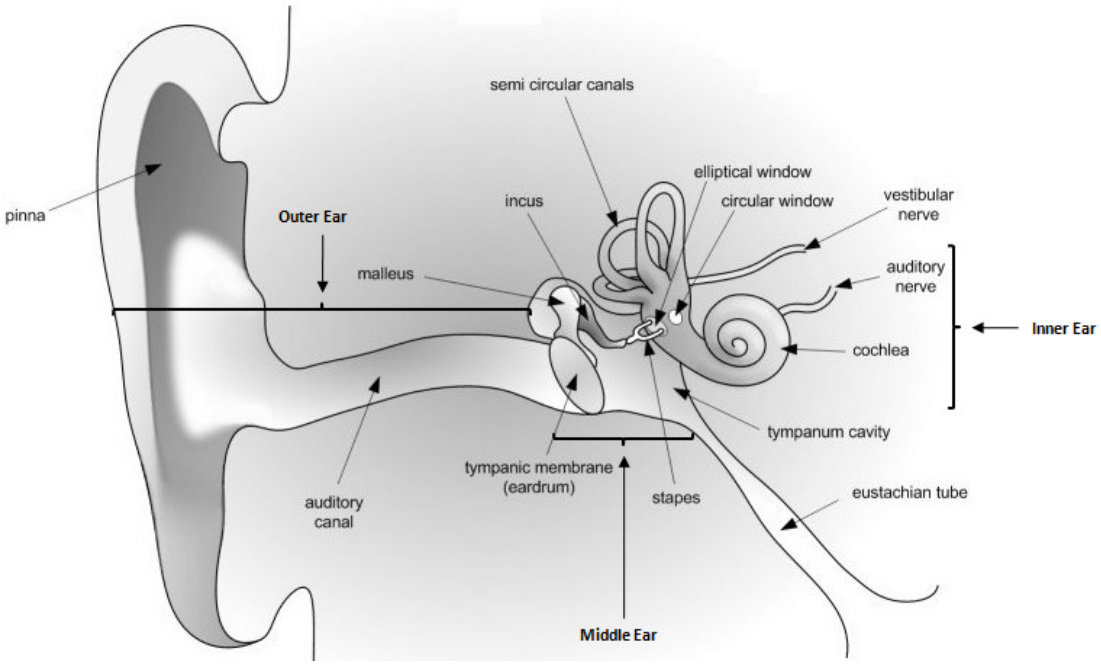


Figure 2.1: Structure of the human auditory system, broadly divided into three parts, viz., the outer ear, middle ear and inner ear [46].

the basilar membrane is contained in it. To excite the membrane, the energy in the air received at the ear drum from the outer ear must be effectively transmitted to the fluid medium in the inner ear. This is the function of the middle ear. The air vibrations are a result of particles oscillating with small forces but with large displacements.

But in the inner ear, the particles in the medium would have to oscillate with large forces but and small displacements. To prevent energy losses during the transfer, impedance matching is required. The middle ear employs a mechanical system to achieve this. The malleus, which is a hammer-like bone structure, is firmly attached to the eardrum. The malleus is connected to the incus and the incus is connected to the stapes [47]. The stapes footplate, along with a membrane called the oval window, is connected to the inner ear. The malleus, incus and the stapes are made of hard bones, and collectively perform the function of impedance matching and effective energy transfer

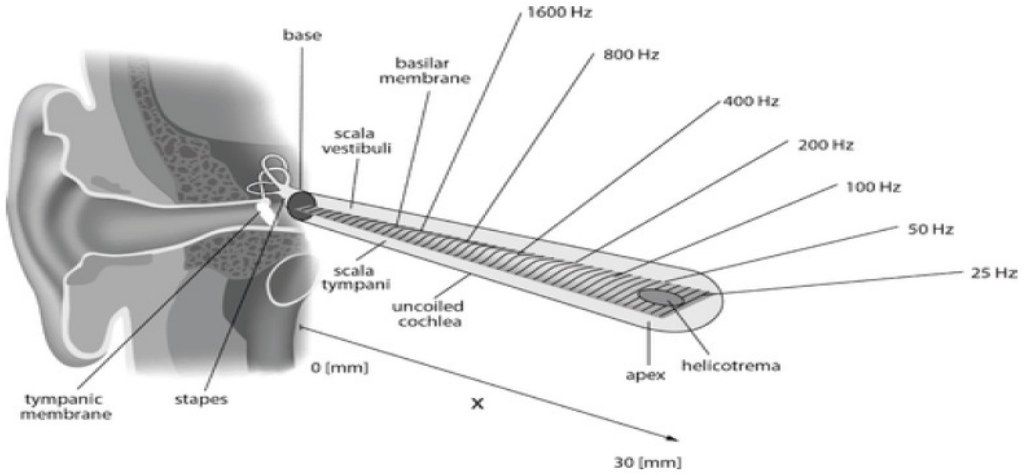


Figure 2.2: Structure of the inner ear, shown with the cochlea unwound, revealing the basilar membrane. Each point on the membrane is sensitive to a narrow band of frequencies [48].

from the outer ear to the inner ear. The best impedance match is achieved at around 1 kHz.

2.1.3. Inner Ear

The inner ear, which is otherwise known as the cochlea, is a snail-shaped spiral structure wound two and a half times around itself [47]. The cochlea processes the incoming vibrations and it is the part of the ear responsible for creating the electrical signals, which are transmitted to the brain and hence, result in the perception of sound. The cochlear structure is shown in Figure 2.2 when it is unwound [48]. The cochlea consists of a region called the scala vestibuli, which contains a fluid different from that in scala media and scala tympani. The scala vestibuli and scala media are separated by the thin Reissner's membrane. The scala vestibuli is located at the beginning of the cochlea, at the end of the oval window. The stapes transfers the vibrations to the fluid regions, which excite the basilar membrane. The basilar membrane runs along the length of the

cochlea, beginning at the base and ending at the apex. The basilar membrane is narrow at the base but about thrice as wide at the apex. The membrane separates the scala media from the scala tympani and supports the Corti. Each point in the basilar membrane is tuned to a particular frequency. The base of the membrane is tuned to the higher frequencies and the lower frequencies are tuned towards the apex of the membrane. Hence, the basilar membrane responds faster to higher frequencies than lower frequencies, as the lower frequencies take longer before traveling down to their resonance point on the basilar membrane. This results in faster sensation for higher frequencies than lower frequencies. It was initially assumed by Helmholtz in 1940 that the basilar membrane consists of a set of dense but discrete locations, which are tuned to specific frequencies [49]. However, von Bekesy later discovered that the membrane consists of a continuum of resonators [50].

When the stapes transfers the vibrations to the fluids in the inner ear, the fluid vibrations trigger basilar membrane vibrations. Sensory hair cells contained in the Corti transform the mechanical vibrations of the basilar membrane to electrical signals. The sensory cells in the Corti are present along the length of the membrane. A number of these hair cells are present in the Corti, and all electrical impulses are transmitted in a bundle of nerve fibers called the auditory nerve to the auditory cortex. The spatial relationship of the individual nerve fibers is preserved in the cortex, which results in the perception of frequencies as they are.

2.2. Psychoacoustics

The field of psychoacoustics deals with measurement and modeling of phenomena related to the perception of sounds. Due to lack of methods to physiologically

measure the hearing sensations produced in the brain, indirect techniques are adopted to measure sensations related to specific phenomena, such as auditory masking and the sense of loudness. Experiments designed to measure these phenomena involve the presentation of an auditory stimulus to a test human subject, and recording a subjective response of the subject to the stimulus. Such experiments are referred to as psychoacoustic or psychophysical experiments. For instance, in experiments for studying loudness perception, a listener can be asked to rank a set of sounds on a relative scale of loudness with respect to a reference sound. Essential principles of psychoacoustics and the techniques used to measure them are described below.

2.2.1. The Absolute Threshold of Hearing

The most fundamental aspect of human hearing is the threshold of hearing sensation. The Absolute Threshold of Hearing (ATH) is defined as the smallest intensity level that is just audible in a quiet surrounding. The strength of an auditory stimulus, which is essentially a sound wave, is measured in the unit of "decibels of Sound Pressure Level" or dB SPL. The sound pressure level β of a stimulus is defined as follows [6].

$$\beta \text{ (dB SPL)} = 10 \log_{10} \left(I / I_0 \right) = 20 \log_{10} \left(p / p_0 \right) \tag{2.1}$$

Here, I is the intensity level of the sound expressed in watts/meter^2 and p is the pressure of the sound in Newton/meter^2 (Pascal). Intensity and pressure of a sound are related by the following equation.

$$I = \left(\frac{p^2}{4} \right) 10^{-10} \tag{2.2}$$

This equation is valid only at one temperature and pressure. But the corrections in the equations for varying temperatures and pressures are negligible for most practical

acoustic experiments and hence, the equation can be assumed valid. The intensity level I_0 is 10^{-12} watts/m^2, or correspondingly, the pressure p_0 is equal to 20 μPa.

Common methods for measuring the ATH involve evaluation of the pressure levels of pure tones at which listeners find them to be just audible. Hence, the ATH (which is evaluated for each frequency) represents hearing thresholds only for tones, and not for signals with multiple tones or with complex spectra. The ATH curve can vary depending on the experimental method employed to measure the sound pressure levels. The main types of the absolute threshold curve are the minimum audible pressure (MAP) and the minimum audible field (MAF). The MAP curve is estimated by measuring the sound pressure level in the ear canal at a point close to the eardrum using a small microphone that is inserted into the ear. Hence, the MAP curve represents the absolute hearing threshold for a single ear. On the other hand, the MAF curve is estimated by measuring the sound pressure level of a tone at the center of the listener's head in the absence of the head, i.e., in free field. The sound is presented in such cases in an anechoic chamber through a loudspeaker. The MAF curve thus, represents the binaural hearing threshold. Monaural thresholds are about 2 dB above binaural thresholds.

The absolute threshold of hearing is defined at each frequency f (in Hz), and is given by the following equation [49]. The equation was obtained through fitting data from psychophysical experiments.

$$\text{ATH(dB SPL)} = 3.64\left(\frac{f}{1000}\right)^{-0.8} - 6.5e^{-0.6\left(\frac{f}{1000}-3.3\right)^2} + 10^{-3}\left(\frac{f}{1000}\right)^4 \qquad (2.3)$$

Figure 2.3 shows the absolute threshold of hearing curve. This curve represents the hearing thresholds for a person with normal hearing ability. It can be notices that the

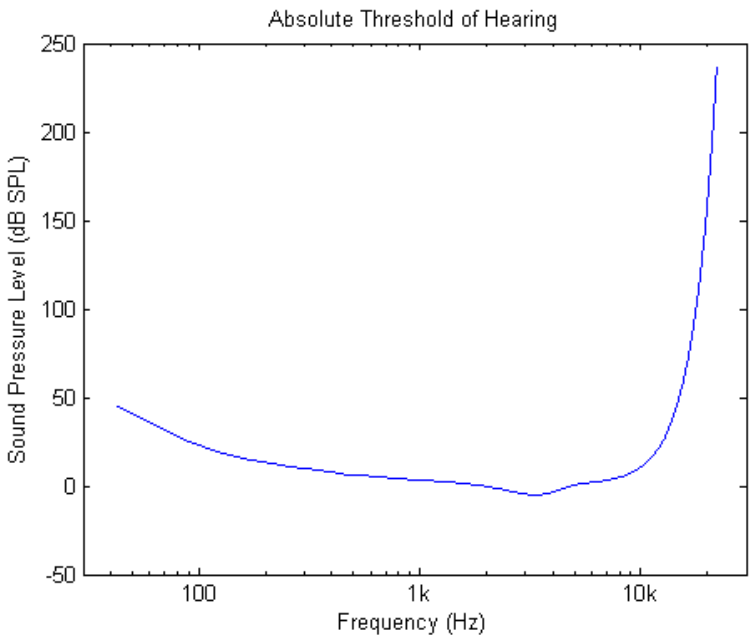


Figure 2.3: The absolute threshold of hearing (ATH) curve for humans, showing the sensitivity of the ear to tones at different frequencies.

sensitivity of the ear is quite low at the lower and the higher frequencies, but sharply increases towards the mid-range frequencies. This non-uniform nature of sensitivity to different frequency components is a result of the properties of the outer and middle ears, which transmit mid frequencies with lesser attenuation than the lower and higher frequencies. The outer ear's resonance around 4 kHz causes a particularly noticeable drop in the threshold around the same frequencies.

The absolute threshold of hearing is exploited in many applications, particularly in audio encoding. Algorithms such as the MPEG - 1 Layer 3 encoding scheme ensure that the quantization noise of the encoded audio is contained below the absolute threshold of hearing to render it imperceptible.

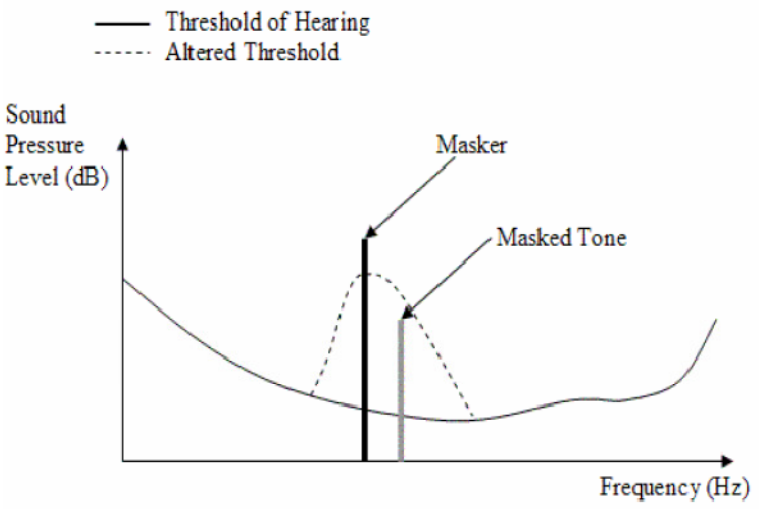


Figure 2.4: Masking phenomenon illustrated by the interaction between two closely spaced tones, one of which is stronger than the other and tends to mask the weaker tone [51].

2.2.2. Auditory Masking

An important auditory phenomenon observed in everyday life is masking of one sound by another. Masking refers to the phenomenon where one sound is rendered inaudible due to the presence another. For instance, when a loud interfering noise is present, the audibility of speech is reduced, and sometimes the speech is completely inaudible. Masking can be partial or complete. When the intensity of the masking sound is increased, the audibility of the sound of interest gradually reduces until a level beyond which the sound becomes completely inaudible. The sound pressure level of an auditory stimulus (referred to as the 'maskee') at which it is just audible in the presence of a masking sound (also called the 'masker') is referred to as the masking threshold of the sound. In the absence of any masker, the masking threshold of a pure tone is the absolute threshold of hearing, which is the threshold in quiet. Also, the threshold of the tone

remains at the threshold in quiet when the frequency of the tone and the frequencies in the masker are widely separated.

Masking is heavily exploited in several state of the art audio encoders such as the MP3 algorithm, which ensure that the quantization noise introduced during the encoding process are maintained under the masking thresholds in various frequency bands so that the quantization noise is inaudible. Due to its importance, masking has been widely studied in the field of psychoacoustics. The broad classification of different kinds of masking scenarios are described below.

Depending on the instant of occurrence of the masker and the maskee, masking can be broadly classified under the following two categories, which will be elaborated upon below.

- Simultaneous masking (or Spectral masking)
- Non-simultaneous masking (or Temporal masking)

Simultaneous Masking

Simultaneous masking, or spectral masking occurs when a masker and maskee occur simultaneously. In such a scenario, the extent to which is masking occurs depends on the intensity level of the maskee and masker and also on the frequency components present in the masker and maskee. An example is shown in Figure 2.4, where two closely spaced tones are present [51]. The stronger tone is the masker. The dotted line in the plot shows that the absolute hearing threshold (which is the black solid curve) is raised significantly at and around the masker tone. This modified threshold of hearing is known as the Masked Threshold. An interesting observation in the plot is that the masked threshold is also defined at the same frequency as the masker. The masking threshold at

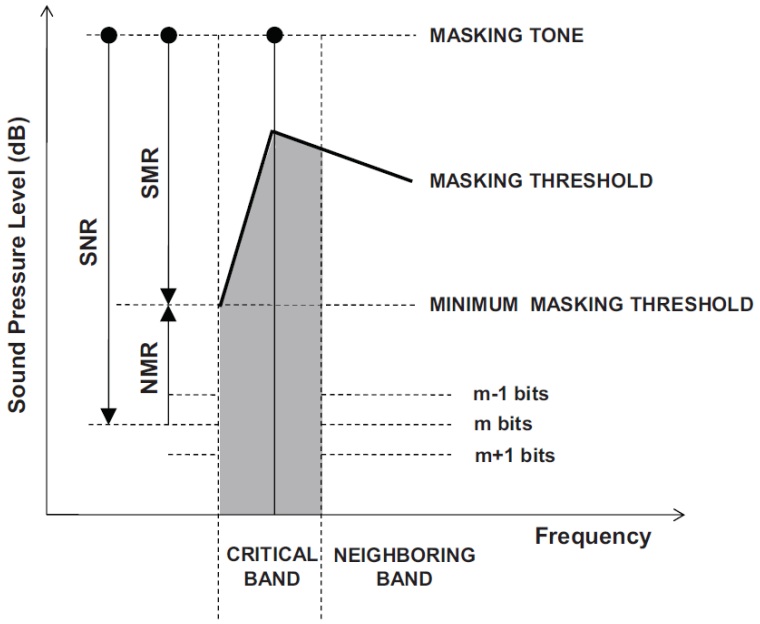


Figure 2.5: General structures of the masking pattern produced by a tone. The masking pattern resembles the response of the ear to the tone produced as vibrations on the basilar membrane. The masking threshold characterizes the pattern, and any spectral component below the threshold is inaudible [52].

the masker tone's frequency indicates that if the second (masked) tone were at the same frequency as the masker, then the intensities of the tones would superpose. In this case, second tone's intensity level needs to be higher than the masking threshold at that frequency to perceive an intensity change at that frequency.

Consider the example shown in Figure 2.5 [52]. If a masking tone is presented, then the excitation it produces in the basilar membrane creates a masking pattern, whose masking threshold is shown by the solid black line with the gray shaded region below it representing the critical bandwidth around the tone. When the ear senses a tone at a certain frequency, it significantly affects the audibility at a narrow band of frequencies around the tone, whose bandwidth is referred to as the critical bandwidth. There is also a predictable reduction in audibility in band neighboring the critical band. This

phenomenon is known as the *spread of masking*, and is exploited in audio encoders. The notion of critical bands will be described in detail in the next section.

The masking capability of a sound is measured by its Signal-to-Masker Ratio or Signal-to-Mask Ratio (SMR). The SMR is the ratio of the masker signal power to the minimum masking threshold power. Higher the masking ability of a signal, higher is its minimum masked threshold. Simultaneous masking scenarios can be classified into four kinds, as described below.

1. Tone Masking Tone (TMT) –

In this case, as the name suggests, both the masker and the maskee are tones. The measurement of masked thresholds of tones is quite straightforward, except when the masker and the maskee are closely spaced in frequency, in which case the thresholds are difficult to measure because of the occurrence of beats [57]. The beats indicate the presence of an additional frequency apart from the masker, and can sometimes render an audible maskee detectable. In this case, the listener is not actually detecting the maskee but the beat tones. A method to avoid this problem is to keep the masker and maskee at a 90 degree phase difference [47]. The minimum SMR in TMT scenarios is usually about 15 dB.

2. Noise Masking Tone (NMT) –

In this scenario, a stronger noise tends to mask a tone. In most psychoacoustic experiments the minimum SMR is usually -5 to 5 dB [13]. This indicates that noise is a better masker than a tone.

3. Tone Masking Noise (TMN) –

In the scenario where a tone tends to mask a noise, the tone is usually required to have sufficient intensity to produce enough excitation in the basilar membrane by itself, if the frequencies in the noise are to be masked. In most experiments to study this phenomenon, the noise is narrowband, with its center frequency at the masker tone, and the bandwidth of the noise is confined to one critical band. The minimum SMR usually observed in this case is from 20-30 dB [13].

4. Noise Masking Noise (NMN) –

In this case, a narrowband noise masks another narrowband noise. In this scenario, it is difficult to study properties such as the minimum SMR due to the complex interactions between the phases of the spectra of the masker and the maskee [13]. This is because different phase difference between components in the two signals can lead to different SMRs.

Non-simultaneous Masking

Non-simultaneous masking refers to the masking scenarios in which one sound tends to mask another even when both are presented in succession. This is also known as temporal masking. There are two kinds of temporal masking scenarios, viz., post-masking and pre-masking.

1. Post-masking –

Post-masking occurs when a masker sound is presented and immediately after it is turned off, the maskee occurs. Due to the gradual decay of the masker in time after it is switched off, the masker still produces some hearing sensations

which masks the subsequent stimulus. Usually, post-masking lasts for about 100-200 ms after the masker is removed.

2. Pre-masking –

Pre-masking, on the other hand, is the phenomenon where a sound, which is immediately followed by a stronger masking sound, is briefly masked even before the onset of the masker sound. This does not mean that the ear can anticipate future sounds, but can be attributed to the fact that a stimulus is not asserted instantly, but requires a small build-up time for its eventual onset. This build-up can produce some masking before the onset. Pre-masking is much shorter than post-masking, lasting usually only for about 20 ms.

2.2.3. Critical Bands

The auditory system can be modeled as a bank of overlapping frequency selective bandpass filters, otherwise known as auditory filters, with the bandwidth of a filter increasing with increasing center frequency. This model was suggested by Fletcher in [6] based on the results of psychophysical experiments to analyze the functioning of the auditory system.

In the experiments, the detection threshold of a pure tone was measured when it was presented to a listener in the presence of a narrowband noise with the same center frequency as the pure tone. The detection threshold of the tone was measured for varying bandwidths of the noise, while keeping the power spectral density of the noise constant (That is, the noise power increased linearly with increasing bandwidth). In the experiments, it was observed that the detection threshold of a tone increases as the bandwidth is gradually increased, but beyond a particular bandwidth, the threshold ceases

to increase. The detection threshold does not increase with further increase in the bandwidth of the noise, but instead remains constant. That is, the audibility of a tone depends on the noise power only within a certain bandwidth around that tone. In addition, it was observed that this bandwidth increased with increasing frequency of the pure tone. This bandwidth is termed as the critical bandwidth.

The basilar membrane acts as this bank of bandpass filters. Each point on the basilar membrane responds to a narrow band of frequencies around the center frequency corresponding to the location on the membrane. This narrow band of frequencies is the critical band for that center frequency. This experiment was repeated and the notion of critical bands confirmed by several other authors [14]. Zwicker and Fastl proposed an analytical expression of the critical bandwidth CB(f) as a function of the center frequency f, as given below.

$$\mathrm{CB}(f) = 25+75\left[1+1.4\left(\frac{f}{1000}\right)^2\right]^{0.69} \text{ Hz} \tag{2.4}$$

From the idea of critical bands, a scale was developed to represent frequencies in terms of distance units on the basilar membrane, in effect mapping the frequency scale in Hz onto distances along the basilar membrane. This scale is known as the critical band-rate scale. The scale is derived by stacking critical bandwidths such that the upper limit of one critical band corresponds to the lower limit of the next critical band [47]. The scale has the units “Bark”, coined in honor of Barkhausen, who introduced the “phon” units for loudness level measurement. The analytical expression mapping frequency to the critical band-rate ‘z’ is given by the following expression [47].

$$z(f)\ (\text{Barks}) = 13 \arctan\left(\frac{0.76f}{1000}\right) + 3.5 \arctan\left(\frac{f}{7500}\right)^2 \tag{2.5}$$

However, these experiments assumed the auditory filters to have a rectangular magnitude frequency response. More recent experiments estimated the shapes of auditory filters and discovered they are not rectangular.

Two important techniques have been used in the past to estimate auditory filter responses, which are briefly discussed here [14]. In one method, the filter shape around a center frequency is determined by presenting narrowband masker signals along with a tone at the center frequency. For a masker centered at a particular frequency, the intensity level of the masker required to just mask the signal is estimated. Performing this experiment for maskers at several frequencies produces a curve referred to as the Psychoacoustic Tuning Curve (PTC). This curve represents the masker levels required to produce a constant output from the filter, as a function of frequency. On the other hand, conventionally, in methods of estimating the response of a system (which is the auditory filter in this case), the response of the system is evaluated by keeping the input level constant over all frequencies. But if the system is linear, the PTC is an accurate measure of the auditory filter shape. Hence, it must be assumed that the auditory filters are linear in this experiment. Since the masker is a narrowband signal, the test tone can at times be sensed from auditory filters adjacent to the one intended to be excited. This is called "off-frequency" listening, and results in erroneous tuning curves [14].

The second method, namely the notched-noise method, overcomes this problem. In this method, the masker is a noise with a notch centered at the center frequency of the auditory filter to be studied. This ensures that the test tone at the center frequency is

sensed only through the corresponding auditory filter and no off-frequency responses are produced.

Through notched-noise experiments, Patterson suggested a "rounded exponential" filter shape for the auditory filters, which had a rounded-top for the pass-bands of the auditory filters and an exponential roll off in the stop bands. In this case, the critical bandwidth of the filter is equal to the effective bandwidth of the filter, which is also referred to as the Equivalent Rectangular Bandwidth (ERB). The ERB bandwidth of an auditory filter $ERB(f)$ as a function of its center frequency f in Hz is expressed according to the following expression [23].

$$ERB(f) = 24.7(4.37f/1000+1) \text{ Hz} \tag{2.6}$$

Since the ERB defined in equation (2.6) is derived from the actual shapes of auditory filters, it is a more accurate measure of the critical bandwidth than that defined in equation (2.4).

Similar to the critical band-rate scale, a scale was developed using ERB bandwidths, known as the ERB scale. The ERB number for any frequency is the number of ERB bandwidths that can be stacked under that frequency. The ERB number p of a frequency f in Hz is given by the following expression.

$$p \text{ (in ERB units)} = 21.4 \log_{10}\left(\frac{4.37f}{1000}+1\right) \tag{2.7}$$

Based on the above, in the subsequent chapter, loudness estimation methods are explained.

Chapter 3

HUMAN AUDITORY MODELS AND LOUDNESS ESTIMATION

Loudness is the intensity of sound as perceived by a listener. The human auditory system, upon reception of an auditory stimulus, produces neural electrical impulses, which are transmitted to the auditory cortex in the brain. The perception of loudness is inferred in the brain. Hence, it is a subjective phenomenon.

Loudness, as a quantity, is different from the measure of the sound pressure level in dB SPL. Through subjective experiments on human test subjects (also referred to as psychophysical experiments), it has been found that different signals produce different sensitivities in a human listener, because of which different sounds having the same sound pressure level can have different perceived loudness. Hence, quantifying this quantity requires incorporation of knowledge of the working the human auditory sensory system. Methods to quantify loudness are based on psychoacoustic models that mathematically characterize the properties of the human auditory system. Some of the important such models are discussed below.

3. 1. Loudness Level and the Equal Loudness Contours (ELC)

The early attempts to quantify loudness were based on psychophysical experiments on human test subjects, involving two techniques – magnitude production and magnitude estimation [14]. The magnitude production technique requires the test subjects to adjust the intensity level (dB SPL) of the test sound until its perceived loudness is equal to that of a reference sound. The reference is usually a 1 kHz sinusoid. In the magnitude estimation method, the subject is presented with a test sound of varying intensity levels, and is required to rank them according to their perceived loudness.

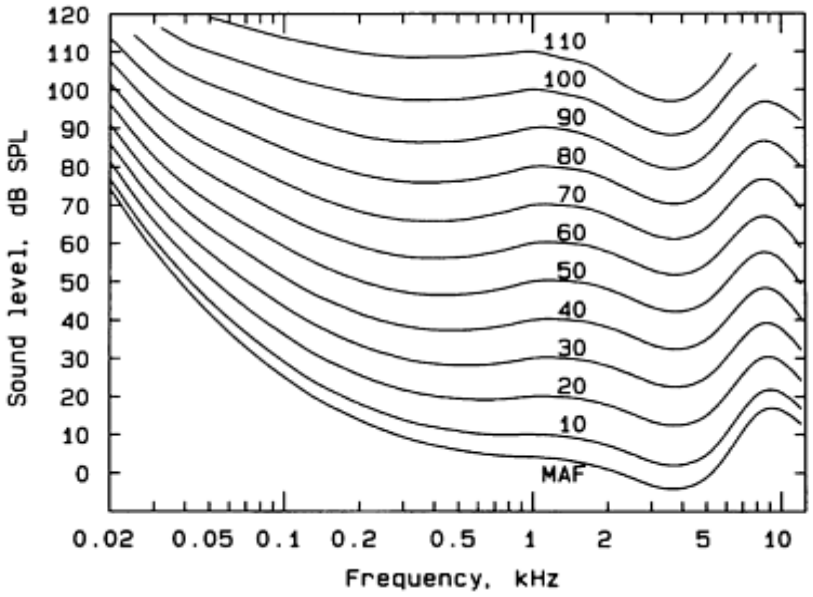


Figure 3.1: Equal Loudness Contours [19].

The magnitude production method requires the measurement of *loudness level*, which does not rely on an absolute scale of loudness, but involves judging how intense a 1 kHz tone must be to sound as loud as a test sound. From this technique, the 'Phon' scale of measurement of loudness level was derived. For instance, if a pure tone of certain intensity in dB SPL sounds as loud as a 1 kHz tone of intensity 40 dB SPL, then the given pure tone has a loudness level of 40 Phons.

In 1933, the magnitude production method was adopted by Fletcher and Munson to estimate the sensitivity of the ear to pure tones of different frequencies [1]. The intensities of pure tones of different frequencies were adjusted until they matched a reference 1 kHz tone of a fixed intensity level. The experiment was repeated for different intensities of the reference tone. The results of this experiment are the Equal Loudness Contours, shown in Figure 3.1. Each contour indicates the intensities of pure tones at different frequencies which have the same loudness level, measured in phons.

The lowest contour represents the absolute threshold of hearing, which is at a level of 3 phons. At lower frequencies, the ear is less sensitive at low loudness levels. It can be observed that at lower frequencies, the sensitivity of the ear to intensity increases as the loudness level increases. For instance, for a 100 Hz tone, the intensity at absolute threshold (3 phons) is 27 dB SPL, whereas at 90 phons, the intensity is 99 dB SPL. Thus, to increase the loudness level by 87 phons, the intensity of the 100 Hz tones needs to be raised by 72 dB SPL. That is, the growth of sensitivity is higher at the lower frequencies than middle of high frequencies. The contours flatten out with increase in the loudness levels.

Loudness Estimation Using Equal Loudness Contours

Equal Loudness Contours have been used in loudness level meters, which attempt to measure loudness of a signal using the ELC to account for the contribution of different frequency components to the perceived loudness. This is achieved by weighting the intensities of individual frequency components according to the shape of an appropriately chosen loudness contour.

The weighting networks of the loudness meters only roughly approximate the equal loudness contours. At low loudness levels, the lower frequencies contribute less to the perceived loudness. Hence, an approximate loudness contour is chosen and those weights are applied at lower intensities. Conventionally, the “A” weighting curve is used in these cases, which is derived from the 30-phon equal loudness contour. At moderate intensity levels, the “B” weighting curve, which is derived from the 70-phon contour, is used. At high intensity levels, the “C” weighting curve is used, which models the sensitivity of the ear at high intensities, where the loudness contours are flatter. Hence,

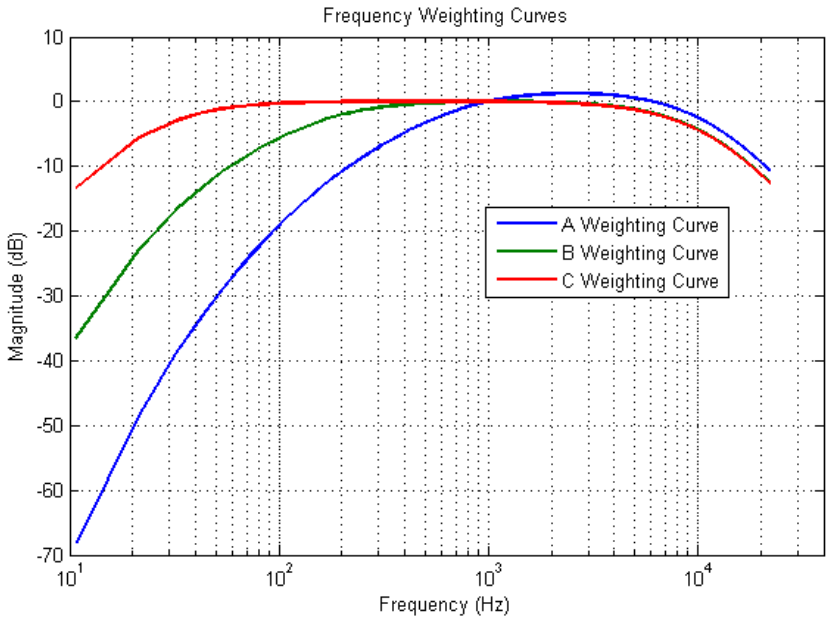


Figure 3.2: A, B, and C Weighting Curves as defined in ANSI S1.4-1983 standard [15].

the C weighting curve does not filter out the lower frequencies as aggressively as the A or B weighting curves. Measurements obtained from loudness meters are specified according to the weighting curves used. For instance, a meter produces a reading 40 dBA, when the A weighting curve was used and produced a weighted intensity of 40 dB. The A, B, and C Weighting Curves as defined in the ANSI S1.4-1983 standard [15] are shown in Figure 3.2.

The disadvantage of loudness meters is that the readings are only approximate, as they choose only an approximate weighting scheme. Moreover, the weighting schemes are reliable only with steady state sounds, and in the presence of transients, they do not produce readings true to the subjective perceptions of the loudness. They also do not perform well with complex sounds with energy spread over a wide range of frequencies. More importantly, the weighting curves provide only a measure of a physical intensity. They do not provide a measurement reflecting the psychological perception of loudness.

3.2. Steven's Law and the 'Sone' Scale for Loudness

An absolute scale for loudness is one where when the measure of loudness is scaled by a number 'x', the perceived loudness by a listener should also be scaled by the factor 'x', as mentioned in [6]. If the number is doubled, the perceived loudness should be doubled. Moreover, this should be consistent with experimental results performed on a sufficiently large group of human subjects. This, of course, is under the presumption that such scales do exist.

The development of an absolute scale of loudness by Stevens led to the 'Sone' scale, proposed in [53]. One sone is arbitrarily defined to be the *binaural* loudness of a 1 kHz pure sinusoid at a level of 40 dB SPL presented to a listener from a frontal angle of incidence in free field. The monaural loudness of a sound (the loudness perceived when presented to only one ear) is approximately half its binaural loudness. The following relationship was proposed by Stevens in [11] between the loudness level P in phons of a signal and its loudness L in sones.

$$L = 2^{(P-40)/10}, \quad \text{if } P \geq 40 \tag{3.1}$$

Steven's Law for loudness, proposed in [12] suggests a power law dependence of loudness L (measured in sones) on the signal's physical intensity I (in watts/meter^2), as expressed in equation (3.2).

$$L = kI^{0.3} \tag{3.2}$$

Here, k is a constant dependent on the signal spectrum. This relation suggests that mathematically, the intensity is compressed (through the fractional exponent) to obtain the loudness. For a ten-fold increase in intensity, the loudness is approximately doubled. This relationship does not hold for sounds with pressure levels below 40 dB SPL. The

power law relationship was confirmed by several experiments using a variety of techniques [14].

3.3. Loudness Estimation from Neural Excitations

Steven's Law is a good model to quantify the loudness of an auditory stimulus, which has been confirmed by a number of psychophysical experiments. However, it provides information only about the perceived loudness, which is but one of a number of psychophysical phenomena experienced from auditory stimuli. In addition, phenomena such as masking and the notion of critical bands is a consequence of the properties of the human auditory system. It is, hence, beneficial to develop models that characterize all the stages of the human auditory system in order to develop from them more fundamental metrics from which all psychophysical properties of auditory stimuli can be computed.

It is well known that sound received by the ear is processed by the auditory system to produce vibrations along the basilar membrane (referred to as the excitation pattern) in the cochlea, which are converted to neural impulses by the hair cells in the cochlea and transmitted by a bundle of nerve fibers in the auditory nerve to the auditory cortex in the brain, where the electrical signals are interpreted. Hence, several human auditory models have been developed which mimic the stages of the auditory system and obtain neural excitations of a given stimulus [16,18,19,21,22]. These auditory models are employed in loudness estimation under the idea that perceived loudness is proportional to the intensity of neural activity produced along the length of the cochlea. The neural impulses are also referred to as auditory patterns or auditory representations. The term "auditory pattern" was introduced by Fletcher in [6].

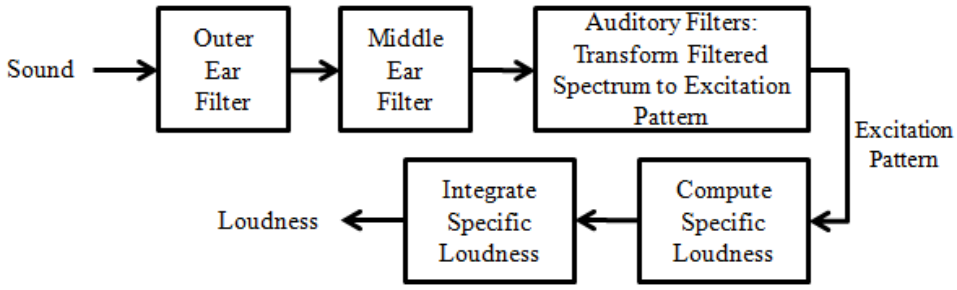


Figure 3.3: The basic structure of auditory representation based loudness estimation algorithms.

The general structure of auditory pattern based loudness estimation is shown in Figure 3.3. The input auditory stimulus is processed by filters that model the outer and middle ear transfer functions. The resulting signals are filtered by the bank of frequency selective critical bandwidth filters representing different locations on the basilar membrane in cochlea, and the excitations at these points are computed. Then, the loudness per critical band is calculated as a non-linear compression of the respective excitation. The so derived pattern is called partial loudness, or the specific loudness. The specific loudness represents the intensity of the neural impulses produced by the cochlear hair cells. The total loudness (or simply, loudness) is then computed as the integral of the specific loudness, by evaluating the area under the specific loudness curve.

Prior loudness estimation methods based on the notion of auditory patterns are described below. In 1933, Fletcher and Munson, apart from the Equal Loudness Contours, also developed a method to estimate loudness of a signal through auditory representations [1]. Their method consisted of a transformation of loudness level of each frequency component to a loudness measure at that frequency. The loudness measure for a frequency was assumed to be proportional to the rate of firing of neural impulses from the nerve endings at the corresponding location in the basilar membrane. The loudness

values for individual frequency components (which could also be thought of as the specific loudness) were then weighted according to a suitable scheme for measuring total loudness in a sound with multiple tones. These weights accounted for the effect of masking on the contributions of loudness of individual tones. The computed loudness agreed with experiments with signals containing well separated tones, but performed poorly when a large number of frequency components were introduced and more so when the signals had continuous spectra [7]. A key aspect of this method to be noted is that the masking phenomenon and the auditory system's filtering properties are applied in the model *after* the computation of the loudness (or nerve impulses) of individual spectral components. Hence, the onus of accounting for the properties of the auditory system (which acts on a stimulus before the production of neural impulses) rests on the weights, which are applied only after computing the neural impulses (or specific loudness measures).

An improved model was proposed by the same authors in [7], where a quantitative relationship between loudness and masking was found and implemented in estimating loudness of sounds containing closely spaced frequency components or a continuous spectrum. From the spectrum of the stimulus, a masking audiogram was obtained. The masking audiogram represents the intensity in dB by which the threshold in quiet at each frequency is raised because of the presence of nearby frequency components in the signal. The masking audiogram is assumed to be a measure of the neural excitations (i.e., the auditory pattern). Hence, the loudness is estimated by integrating the area under the audiogram. Though this method was found to perform satisfactorily for signals with continuous spectra, it did not work well with signals containing tone

complexes or narrowband spectral content. This is because the masking audiogram is difficult to obtain at regions close to a tone due to the occurrence of beats.

In [16], a generalized model applicable to signals with both tonal and continuous spectra was developed by Zwicker and Scharf. In this approach, the masking patterns were computed by filtering the stimulus signal through a bank of bandpass filters. These bandpass filters have critical bandwidths that mimic the filtering of signals along the basilar membrane in the human auditory system. Their passband and stopband characteristics and the transitions in their responses determine the extent of masking. This leads to a more accurate representation of the masking phenomenon, as opposed to the aforementioned method proposed in [7]. The masking pattern obtained through the bank of bandpass filters is converted to the excitation pattern, which represents the basilar membrane vibrations along its length. The excitation pattern is then converted to neural impulses which represent the specific loudness (or loudness per critical band), by a non-linear compression through a relation similar to Steven's Law. The total loudness of the stimulus is then obtained by integrating the specific loudness over all the critical bands.

In [17], Moore and Glasberg proposed several changes to the model reported in [16] by Zwicker and Scharf. An important modification is in the computation of the excitation pattern. Instead of calculating the excitation pattern after obtaining the masking pattern, the model proposed by Moore and Glasberg computed the excitation pattern directly from the signal power spectrum as the energy of the outputs of the bank of bandpass auditory filters modeling the basilar membrane.

More accurate representations of auditory filter responses are essential in accurate modeling of the masking phenomenon, and hence, accuracy of the derived excitation

pattern. The rounded exponential filters were used to represent the auditory filters in several models [21]. Gammatone filters were developed by [54,55] to model cochlear filter responses in cats. A Gammatone filter of center frequency f_c and bandwidth B has the following impulse response, where n is the order of the filter, φ is its phase offset and k is a scaling factor.

$$h(t) = kt^{n-1} \exp(-2\pi Bt) \cos(2\pi f_c t+\varphi)\, u(t) \tag{3.3}$$

The Gammatone filters were deduced by Schofield [56] to be a good representation of the responses of auditory filters to explain masking data from results of psychophysical experiments reported by [57]. At low orders (about 3-5), the Gammatone filters are very similar to rounded exponential filters, as reported in [20]. The Gammatone filters are symmetric, linear and independent of the input sound intensity level at the corresponding critical band. On the other hand, psychophysical experiments reported in [23,58,59,60] indicated that the auditory filter shapes were asymmetrical, non-linear and level-dependent. Several modifications were made to modify Gammatone filters to model these phenomena. In [61], the gammachirp filterbank was developed to address this issue. In another attempt, the dual resonance non-linear filter was developed [22].

In the papers authored by Moore and Glasberg in [17,60], the characteristics of level dependence and asymmetry in the rounded exponential filters were incorporated into the auditory model to represent the auditory filters. These auditory filters were used in the loudness estimation model proposed by them in [19], which is referred to as the Moore and Glasberg model. The model was found to perform well for sounds of different types of spectra, whether tonal or continuous. In the loudness estimation scheme

discussed henceforth, the Moore and Glasberg model has been employed. The following motivate this choice.

The model incorporates an accurate human auditory model, which captures the properties of each stage of processing in the human ear, and hence, produces reliable measures of basilar membrane excitation patterns and auditory patterns. The model estimates loudness accurately for both sounds with both tonal and continuous spectra. In addition, the loudness is estimated in sones, which is an absolute scale of loudness and quantizes perceived loudness in an intuitive manner. It also estimates loudness accurately for sounds with pressure levels below 40 dB SPL, unlike Steven's Law. In a nutshell, the Moore & Glasberg model analytically provides a reliable estimate of loudness given any signal with arbitrary spectral content, using linear auditory filters. Due to these reasons, the model was standardized by ANSI in 2005 as a new loudness standard [62]. The detailed mathematical expressions involved in the Moore and Glasberg model are described below.

3.4. The Moore and Glasberg Model for Loudness Estimation

The block diagram describing the Moore-Glasberg model is shown in Figure 3.4. The model imitates the characteristics of the outer and middle ear of the human ear, by filtering the input signal through a filter modeling the combined transformation characteristics of the outer and middle ear. The filtered auditory stimulus is subsequently processed by a bank of bandpass filters that model the response of the basilar membrane of the ear in the cochlea along its length. The energies of the signals produced by the bandpass filters comprise the excitation pattern. The excitation pattern is then compressed by a rule similar to Steven's Law to obtain the loudness pattern. The loudness pattern is

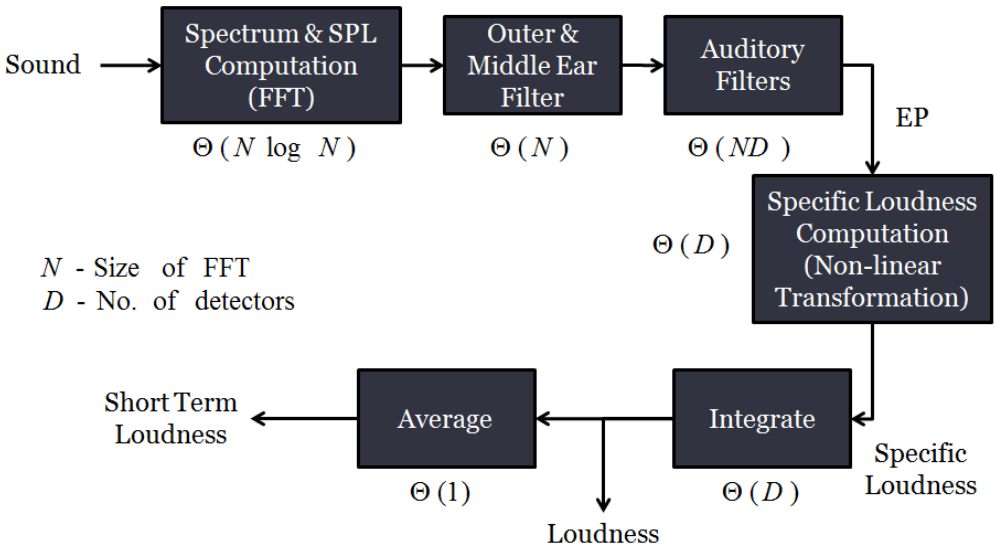


Figure 3.4: Block diagram representation of the Moore & Glasberg model.

then integrated to obtain the total loudness, or simply, loudness. This measure of loudness is also referred to as instantaneous loudness. An averaged measure of the instantaneous loudness, called the short-term loudness is also derived by smoothing the instantaneous loudness over time through an averaging window.

3.4.1. Outer and Middle Ear Transformation

The outer ear accepts the auditory stimulus and transforms it as it is transferred to the ear drum. The transfer function of the outer ear is defined as the ratio of sound pressure of the stimulus at the eardrum to the free-field sound pressure of the stimulus. The outer ear response used in the experiments is derived from stimuli incident from a frontal direction. Other angles of incidence would require correction factors in the response. The free-field sound pressure is the measured sound pressure at the position of the center of the listener's head when the listener is not present. The outer ear can be modeled as a linear filter, whose response is shown in Figure 3.5. As it can be observed,

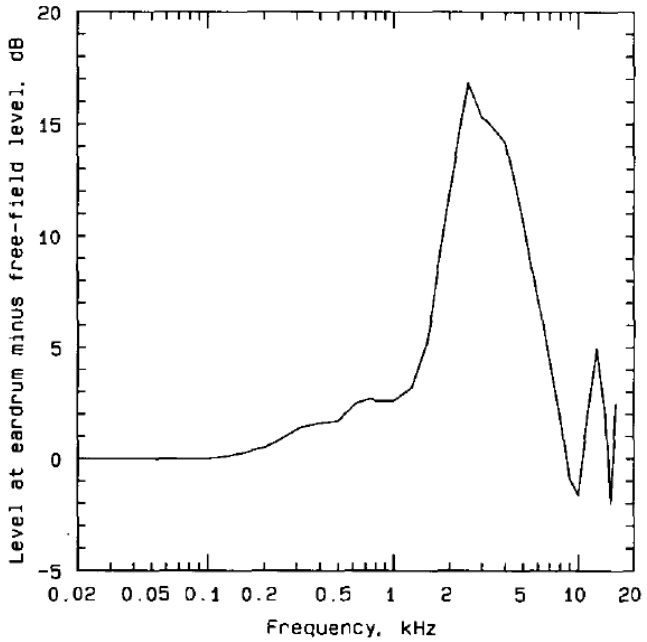


Figure 3.5: The outer ear filter response in the Moore & Glasberg model [19].

the resonance of the outer ear canal at about 4 kHz results in the sharp peak around the same frequency in the response.

The middle ear transformation provides an important contribution to the increase in the absolute threshold of hearing at lower frequencies, as suggested in [47]. The middle ear essentially attenuates the lower frequencies. The middle ear functions in this manner to prevent the amplification of the low level internal noise at the lower frequencies. These low frequency internal noises commonly arise from heart beats, pulse, and activities of muscles [47]. Hence, it is assumed in the Moore & Glasberg model that the middle ear has equal sensitivity to all frequencies above 500 Hz. And below 500 Hz, the response of the middle ear filter is roughly the inverted shape of the absolute threshold curve at the same frequencies.

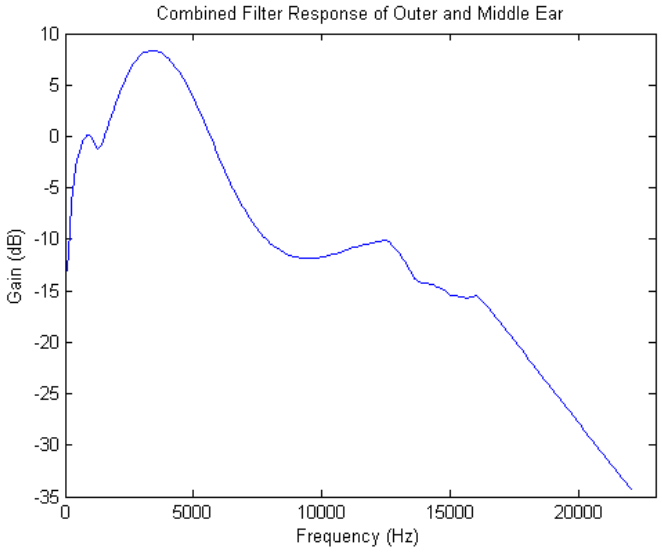


Figure 3.6: Combined magnitude response of the outer and middle ear.

In addition to the middle ear characteristics, an additional attenuation at lower frequencies was required to explain the observed binaural absolute threshold specified in ISO 389-7. This was modeled by introducing a gain factor at the cochlea (called the cochlear gain), which was lesser at lower frequencies [19]. Biologically, this cochlear amplification can be attributed to the outer hair cells, which evolved this trait to prevent the amplification of the low frequency internal noises in the inner ear.

The combined outer and middle ear filter's magnitude frequency response is shown in Figure 3.6. The input sound $x(n)$ with a power spectrum $S_x(\omega_i)$ (where $\omega_i=\exp\left(\frac{j2\pi f_i}{f_s}\right)$ when the sampling frequency is f_s) can be processed with the combined outer-middle ear filter. If the frequency response of the outer-middle ear filter is $M(\omega_i)$, then the output power spectrum of the filter is $S_x^c(\omega_i)=|M(\omega_i)|^2 S_x(\omega_i)$. The spectrum $S_x^c(\omega_i)$ reaches the inner ear and is also called the effective spectrum.

3.4.2. The Auditory Filters: Computing the Excitation Pattern

The basilar membrane receives the stimulating signal filtered by the outer and middle ear to produce mechanical vibrations. Each point on the membrane is tuned to a specific frequency and has a narrow bandwidth of response around that frequency. Hence, each location on the membrane acts as a detector of a particular frequency. The auditory filters comprise a bank of bandpass filters. Each filter represents the response of the basilar membrane at a specific location on the membrane. The auditory filter is modeled as a rounded exponential filter, and the rising and falling slopes of the auditory filter are dependent upon the intensity level of the signal at the corresponding frequency band.

The detector locations on the membrane are represented on an auditory scale measured by the Equivalent Rectangular Bandwidth (ERB) at each frequency. For a given center frequency *f*, the equivalent rectangular bandwidth is given by the expression below.

$$ERB(f) = 24.67\left(\frac{4.37f}{1000}+1\right) \tag{3.4}$$

The auditory filters are represented on an auditory scale derived from the center frequencies of the filters. This auditory scale represents the frequencies based on their ERB values. Each frequency is mapped to an “ERB number”, because of which it is also referred to as the ERB scale. The ERB number for a frequency represents the number of ERB bandwidths that can be fitted below the same frequency. The conversion of frequency to the ERB scale is through the following expression. Here, *f* is the frequency in Hz, which maps to *d* in the ERB scale.

$$d(\text{in ERB units}) = 21.4 \log_{10}\left(\frac{4.37f}{1000}+1\right) \tag{3.5}$$

Let D be the number of auditory filters are used to represent responses of discrete locations of the basilar membrane. Let $L_r=\{ d_k \mid |d_k-d_{k-1}| = 0.1,\ k = 1, 2 \ldots D\}$ be the set of detector locations equally spaced at a distance of 0.1 ERB units on the ERB scale. Each detector represents the center frequency of the corresponding auditory filter. The magnitude frequency response of the auditory filter at a detector location d_k is defined as:

$$W(k,i) = \left(1+p_{k,i}g_{k,i}\right)\exp\left(-p_{k,i}g_{k,i}\right),\ k = 1,\ldots D \text{ and } i = 1,\ldots N, \tag{3.6}$$

,where $p_{k,i}$ is the slope of the auditory filter corresponding to the detector d_k at frequency f_i and $g_{k,i}= | (f_i - f_{c_k}) / f_{c_k} |$ is the normalized deviation of the frequency component f_i from the center frequency f_{c_k} of the detector.

The auditory filter slope $p_{k,i}$ is dependent on the intensity level of the effective spectrum of the signal within the equivalent rectangular bandwidth around the center frequency of that detector. The intensity pattern, I(k), is the total intensity of the effective spectrum within one ERB around the center frequency of the detector d_k. It is computed from the following expression.

$$\mathrm{I}(k) = \sum_{i \in A_k} S_x^c(\omega_i),\ A_k = \{\ i \mid d_k - 0.5 < 21.4 \log_{10}\left(\frac{4.37f_i}{1000}+1\right) \le d_k + 0.5,\ i = 1,\ldots N \tag{3.7}$$

As known through experiments, an auditory filter has different slopes for the lower and upper skirts of the filter response. In the model [23], the slope of the lower skirt p_k^l is dependent on the corresponding intensity pattern value, but the slope of the upper skirt p_k^u is fixed. The parameters are given by the expressions below.

$$p_{k,i}^{l} = p_k^{51} - 0.38\left(\frac{p_k^{51}}{p_{1000}^{51}}\right)(\mathrm{I}(i)-51) \tag{3.8}$$

$$p_{k,i}^{u} = p_k^{51} \tag{3.9}$$

In the above equation, p_k^{51} is the value of $p_{k,i}$ at the corresponding detector location when the intensity I(i) is at a level of 51 dB. It can be computed as follows.

$$p_k^{51} = 4 f_{c_k} / ERB\left(f_{c_k}\right) \tag{3.10}$$

Thus, it can be seen that the slope of the lower skirt matches the auditory filter that is centered at a frequency of 1 kHz, when the effective spectrum of the auditory stimulus has an intensity of 51 dB at the same critical band. The slope $p_{k,i}$ chooses the lower skirt and the upper skirt according to the following equation.

$$p_{k,i} = \begin{cases} p_{k,i}^{l}, & g_{k,i} < 0 \\ p_{k,i}^{u}, & g_{k,i} \geq 0 \end{cases} . \tag{3.11}$$

The excitation pattern is thus, evaluated from the following expression.

$$E(k) = \sum_{i=1}^{D} W(k,i).S_x^c(\omega_i)\ ,\ k = 1,\ldots D \text{ and } i = 1,\ldots N \tag{3.12}$$

$$= \sum_{i=1}^{D} \left(1+p_{k,i}g_{k,i}\right)\exp\left(-p_{k,i}g_{k,i}\right),\ k = 1,\ldots D \text{ and } i = 1,\ldots N \tag{3.13}$$

3.4.3. Specific Loudness Pattern

The specific loudness pattern as mentioned earlier, represents the neural excitations generated by the hair cells, which convert the basilar membrane vibrations at each point along its length (which is the excitation pattern) to electrical impulses. The specific loudness, or partial loudness is a measure of the perceived loudness per ERB.

The specific loudness is computed from the excitation pattern as per the following expression.

$$S(k) = \mathrm{c}\left(\left(E(k)+A(k)\right)^{\propto}\text{-}A^{\propto}(k)\right) \text{ for } k = 1,\ldots D \tag{3.14}$$

The constants are chosen as $\mathrm{c} = 0.047$ and $\alpha = 0.2$. It can be observed that the specific loudness pattern is derived through a non-linear compression of the excitation pattern. $A(k)$ is a frequency dependent constant which is equal to twice the peak excitation pattern produced by a sinusoid at absolute threshold, which is denoted by E_{THRQ} (i.e., $A(k) = 2\,E_{THRQ}(k)$) [19]. It can be inferred from this expression that the specific loudness is greater than zero for any sound, even if below the absolute threshold of hearing. Hence, the total loudness, which would be derived by integrating the specific loudness over the ERB scale, will also be positive for any sound. At frequencies greater than or equal to 500 Hz, the value of E_{THRQ} is constant. For frequencies lesser than 500 Hz, the cochlear gain is reduce, hence, increasing the excitation E_{THRQ} at the corresponding frequencies. This can be modeled as a gain g for each frequency, relative to the gain at 500Hz and above (the gain at and above 500 Hz is constant), acting on the excitation pattern [19]. It is assumed that the product of g and E_{THRQ} is constant. The specific loudness pattern is then expressed as follows.

$$S(k) = \mathrm{c}\left(\left(gE(k)+A(k)\right)^{\propto}\text{-}A^{\propto}(k)\right) \text{ for } k = 1,\ldots D \tag{3.15}$$

The rate of decrease of specific loudness is higher when the stimulus is below absolute threshold, than what is predicted in equation (3.15). This is modeled by introducing an additional factor dependent on the excitation pattern strength. Hence, if $E(k) < E_{THRQ}(k)$, the following holds for the specific loudness pattern.

$$S(k) = \mathrm{c}\left(\frac{E(k)}{E(k)+E_{THRQ}(k)}\right)^{1.5}\left(\left(gE(k)+A(k)\right)^{\propto} - A^{\propto}(k)\right) \tag{3.16}$$

Similarly, when the intensity is higher than 100 dB, the rate of increase of specific loudness is higher, and is modeled by the following equation, which is valid when $E(k) > 10^{10}$.

$$S(k) = \mathrm{c}\left(\frac{E(k)}{1.04\times10^{6}}\right)^{0.5} \tag{3.17}$$

Hence, putting together equations (3.15), (3.16) and (3.17), the specific loudness function can be expressed as in equation (3.18). The constant 1.04×10^{6} is chosen to make $S(k)$ continuous at $E(k) = 10^{10}$.

$$S(k) = \begin{cases} \mathrm{c}\left(\left(gE(k)+A(k)\right)^{\propto} - A^{\propto}(k)\right) & , \; E(k) < E_{THRQ}(k) \\ \mathrm{c}\left(\dfrac{E(k)}{E(k)+E_{THRQ}(k)}\right)^{1.5}\left(\left(gE(k)+A(k)\right)^{\propto} - A^{\propto}(k)\right) & , \; E_{THRQ}(k) \le E(k) \le 10^{10} \\ \mathrm{c}\left(\dfrac{E(k)}{1.04\times10^{6}}\right)^{0.5} & , \; E(k) > 10^{10} \end{cases} \tag{3.18}$$

3.4.4. Total Loudness

The total loudness is computed by integrating the specific loudness pattern $S(k)$ over the ERB scale, or computing the area under the loudness pattern. While implementing the model with a discrete number of detectors, the computation of the area under the specific loudness pattern can be performed by evaluating the area of trapezia formed by successive points on the pattern along with the x – axis (which is the ERB scale). The loudness can then be computed using the following expression.

$$L = \sum_{k=1}^{D\text{-}1} \left[S(k)\delta_d + \frac{1}{2}\left(S(k+1) - S(k)\right)\delta_d \right] \tag{3.19}$$

$$L = \delta_d \left[\sum_{k=2}^{D\text{-}1} S(k) + \frac{1}{2}\left(S(1) + S(D)\right) \right] \tag{3.20}$$

The loudness computed in this manner quantifies the loudness perceived when a stimulus is presented to one ear (the monaural loudness). The binaural loudness can be computed by summing the monaural loudness of each ear.

3.4.5. Short-term and Long-term Loudness

The measure of loudness derive above is also referred to as the instantaneous loudness, as it is the loudness for a short segment of an auditory stimulus. This measure of loudness is constant only when the input sound has a steady spectrum over time. Signals in reality are time-varying in nature. Such sounds exhibit temporal masking, which results in fluctuating values of the instantaneous loudness. Hence, it is important to derive metrics of loudness that are steadier for time-varying sounds.

In [63], loudness estimation for time-varying sounds was performed by suitably capturing variations in the signal power spectrum to account for the temporal masking. The power spectrum was computed over segments of the signals windowed with different lengths, viz., 2, 4, 6, 8, 16, 32 and 64 milliseconds. Then, particular frequency components were selected from the obtained spectra to get the best trade-off time and frequency resolutions. The spectrum was updated every 1 ms, by shifting the windowing frame by 1 ms every time. The steady state spectrum hence derived was processed with the model described above and the instantaneous loudness was computed.

The short-term loudness is calculated by averaging the instantaneous loudness using a one-pole averaging filter. The long-term loudness is calculated by further averaging the short-term loudness using another one-pole filter. The short-term loudness smoothes the fluctuations in the instantaneous loudness, and the long-term loudness reflects the memory of loudness over time. The filter time constants are different for rising and falling loudness. This models the non-linearity of accumulation of loudness perception over time. Increasing loudness due to an attack rapidly accumulates, unlike reducing loudness, which is more gradual. If $L(n)$ denotes the instantaneous loudness of the n^{th} frame, then the short-term loudness $L_s(n)$ at the n^{th} frame is given by the following expression, where α_a and α_r are the attack and release parameters respectively.

$$L_s(n) = \begin{cases} \alpha_a L(n) + (1 - \alpha_a) L_s(n-1), & L(n) > L_s(n-1) \\ \alpha_r L(n) + (1 - \alpha_r) L_s(n-1), & L(n) \le L_s(n-1) \end{cases} \tag{3.21}$$

$$\alpha_a = 1 - e^{-\frac{T_i}{T_a}}, \; \alpha_r = 1 - e^{-\frac{T_i}{T_r}} \tag{3.22}$$

The value T_i denotes the time interval between successive frames. T_a and T_r are the attack and release time constants respectively. Similarly, the long-term loudness $L_l(n)$ can be computed from the following expression.

$$L_l(n) = \begin{cases} \alpha_{l_a} L_s(n) + (1 - \alpha_{l_a}) L_l(n-1), & L_s(n) > L_l(n-1) \\ \alpha_{l_r} L_s(n) + (1 - \alpha_{l_r}) L_l(n-1), & L_s(n) \le L_l(n-1) \end{cases} \tag{3.23}$$

3.5. Moore and Glasberg Model: Algorithm Complexity

The structure of the Moore and Glasberg Model is shown in Figure 3.4, where the complexity of each element in the algorithm is indicated. Enlisted below in detail are the complexities of the operations in the algorithm.

1. Given a frame of N samples of an input signal $x(n)$, the computation of the N-point FFT, and hence, the power spectrum of the signal $\{S_x(\omega_i)\}_{i=1}^{N}$ of the signal has a complexity of $\Theta(N\log N)$.
2. The effective power spectrum reaching the inner ear $S_x^c(\omega_i)$ is computed by filtering the spectrum $S_x(\omega_i)$ through the outer-middle ear filter $M(\omega_i)$. In the dB scale, this reduces to additions of the magnitudes of the signal power spectrum and he filter response, which has a complexity of $\Theta(N)$.
3. The next few steps are part of the auditory filtering process on effective spectrum.
    a. The determination of the intensity pattern $I(k)$ has a complexity of $\Theta(D)$.
    b. The subsequent computation of the auditory filter slopes p_k has a complexity of $\Theta(D)$.
    c. The computation of the auditory filter responses $\{W(k,i)\}_{k=1,i=1}^{D,N}$ has a complexity of $\Theta(ND)$.
    d. Then, the auditory filter operates on the effective spectrum to determine the excitation pattern $E(k)$, which has a complexity $\Theta(ND)$.
4. The computation of the specific loudness pattern $S(k)$ from the excitation pattern has a complexity of $\Theta(D)$.
5. The step of integrating the specific loudness pattern to estimate the total instantaneous loudness L has a complexity of $\Theta(D)$.

6. The final steps of computing the short-term and long-term loudness require a constant number of operations and hence, have a complexity of $\Theta(1)$.

It can be seen from the above analysis that the steps of computing the auditory filter responses and the filtering of the effective spectrum with the auditory filters has the highest complexity, of $\Theta(ND)$. This complexity can be reduced by pruning the number of spectral components to be computed and by pruning the number of detector locations required to accurately capture the excitation pattern shape.

Chapter 4

EFFICIENT LOUDNESS ESTIMATION – COMPUTATION PRUNING TECHNIQUES

The human auditory system, upon reception of a stimulus, produces neural excitations as described in Chapter 3. These neural excitations are transmitted to the auditory cortex where all higher level inferences pertaining to perception are made. Hence, in auditory patterns based perceptual models, excitation patterns can be viewed as the fundamental features describing a signal, from which perceptual metrics such as loudness can be derived. The excitation pattern, after non-linear compression, is integrated to obtain an estimate of loudness. Errors in the excitation pattern have a profound effect on the accuracy of the estimated loudness, because of accumulation of errors in the integration.

The excitation of a signal at a detector is computed as the signal energy at that detector. The computation of the excitation pattern is intensive, having a complexity of $\Theta(ND)$ when the FFT length is N and the number of detectors is D. Pruning the computations involved in evaluating the excitation pattern can be achieved by explicitly computing only a salient subset of points on the excitation pattern and estimating the rest of the points through interpolation.

Pruning can be achieved in two forms. In one case, the number of frequency components can be pruned to approximate the spectrum with only a few components such that the total loudness is preserved. That is, one can choose to retain a subset of frequencies $\{f_i\}_{i=1}^{N}$ for computing the excitation pattern. This is referred to as Frequency Pruning [25]. In the other case, the set of detectors $\{d_k\}_{k=1}^{D}$ can be pruned to choose only a

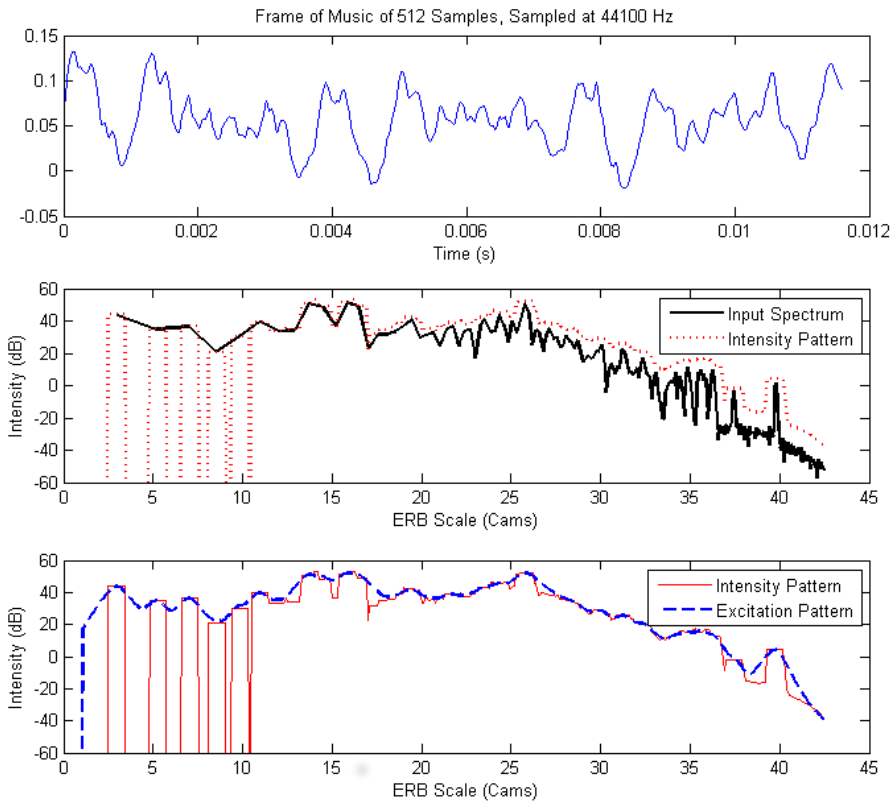


Figure 4.1: A frame of music sampled at 44.1 kHz (top). The intensity pattern along with the spectrum in the ERB scale (middle), and the intensity pattern along with the excitation pattern (bottom) are shown.

subset of detector locations for evaluating the excitation pattern $\{E(k)\}_{k=1}^{D}$. This approach is referred to as Detector Pruning [25]. This is synonymous to non-uniformly sampling the excitation pattern along the basilar membrane to capture its shape.

The intensity pattern, defined in equation (3.7), gives the energy per ERB. The intensity pattern can be used for determining the pruned frequency components and detector locations, as described below.

Frequency Pruning

Pruning the frequency components in the spectrum can be performed by using a quantity called the averaged intensity pattern. The average intensity pattern $Y(k)$ is

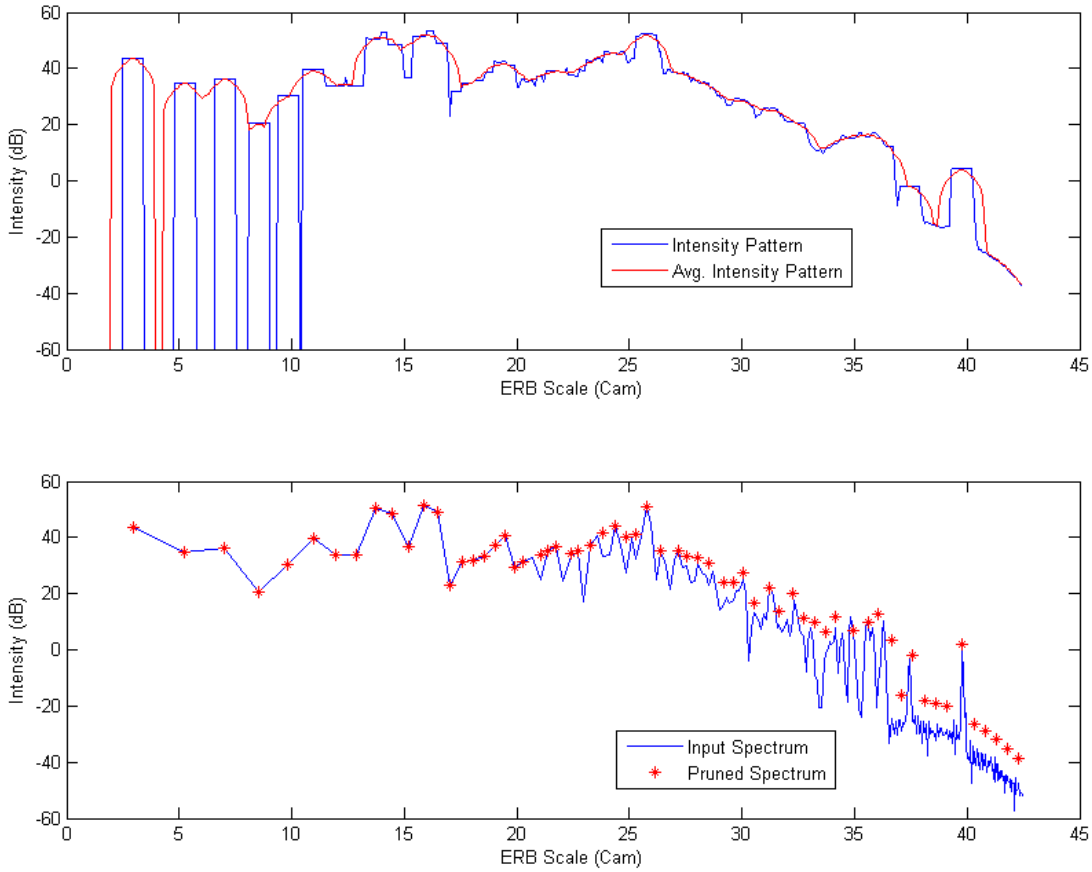


Figure 4.2: The intensity pattern shown with the average intensity pattern (top). The corresponding outer-middle ear spectrum and the pruned spectrum are shown in the bottom.

computed by filtering the intensity pattern, as show in equation (4.1). The average intensity pattern is a measure of the average intensity per ERB.

$$Y(k) = \frac{1}{11}\sum_{i=-5}^{5} \mathrm{I}(k\text{-}i) \tag{4.1}$$

This allows us to divide the spectrum into *tonal bands* and *non-tonal bands*. Tonal bands are ERBs in which only a dominant spectral peak is present. The intensity pattern in these bands is quite flat, with a sudden drop at the edge of the ERB around the tone. The tonal bands can be represented by just the dominant tone, ignoring the remaining components.

These tonal bands are identified as the locations of the maxima of the average intensity pattern $Y(k)$ (see Figure 4.2).

The portions of the spectrum which do not qualify as tonal bands are labeled as non-tonal bands. Each non-tonal band is further divided into smaller bins $B_{1:Q}$ of width 0.25 ERB units (Cam), where Q is the number of sub-bands in the non-tonal band. Each sub-band B_p is assumed to be approximately white. From this assumption, each sub-band B_p is represented by a single frequency component $\hat{S}_p$, which is equal to the total intensity within that band. If M_p is the indices of frequency components within B_p, then $\hat{S}_p$ is given by the following expression.

$$\hat{S}_p = \sum_{j \in M_p} S_x^c(\omega_j) \tag{4.2}$$

This method of dividing the spectrum into smaller bands and representing each band with a single equivalent spectral component is justified, as it preserves the energy within each critical band and consequently, preserves the auditory filter shapes and their responses. Spectral bins smaller than 0.25 ERB may also be chosen for non-tonal bands, but it would result in less efficient frequency pruning.

Detector Pruning

The excitation at a detector location is the energy of the signal filtered by the auditory filter at that detector location. Since the intensity pattern at a detector defined in equation (3.7) is the energy within the bandwidth of the detector, the intensity pattern would have some correlation with the excitation pattern. This is illustrated in the plot in Figure 4.1. It can be observed that for the given signal in Figure 4.1, the shape of the excitation pattern is to a significant extent, dictated by the intensity pattern. The peaks

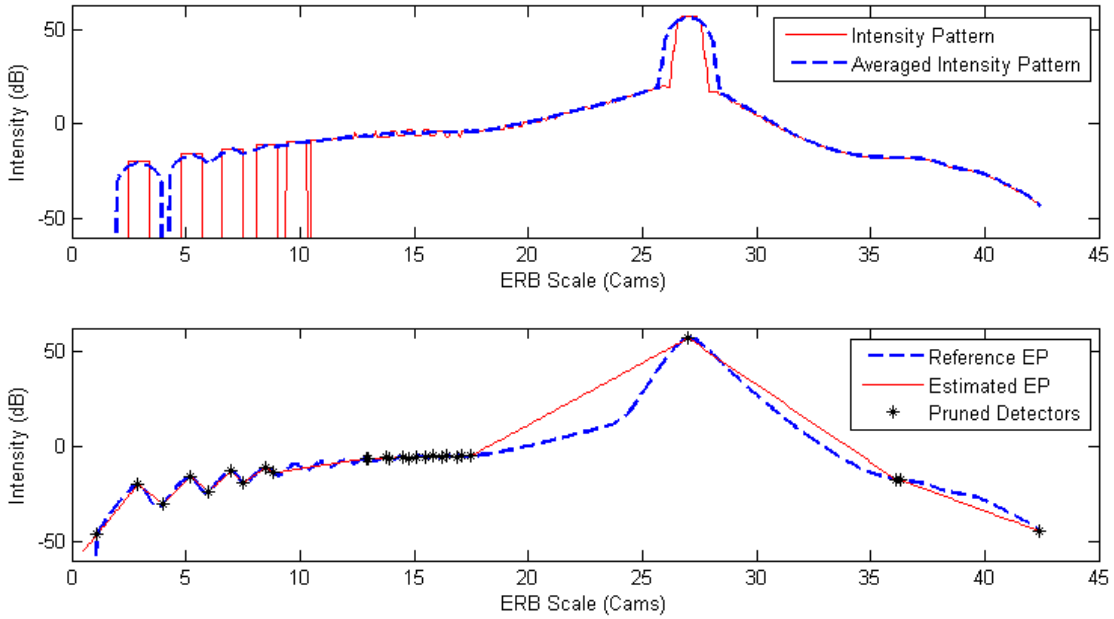


Figure 4.3: The intensity pattern and average intensity pattern (top) for a sinusoid of frequency 4 kHz sampled at a rate of 44.1 kHz. The reference excitation pattern of the sinusoid, the estimated excitation pattern and the pruned detector locations are shown (bottom).

and valleys of the excitation pattern largely follow the peaks and valleys in the intensity pattern.

In [25], detector pruning was achieved by choosing salient points based on the averaged intensity pattern. The detectors at the locations of the peaks and valleys of the averaged intensity pattern are chosen for explicit computation. If the reference set of detectors is $L_r = \{d_k | \, |d_k - d_{k-1}| = 0.1, k = 1, 2 \ldots D\}$, then the pruning scheme produces a smaller subset of detectors $L_e = \{d_k | \, \frac{\partial Y(k)}{\partial k} = 0, k = 1, 2 \ldots D\}$. The points on the excitation pattern are computed for the detectors in L_e. The rest of the points in the excitation pattern are computed through linear interpolation.

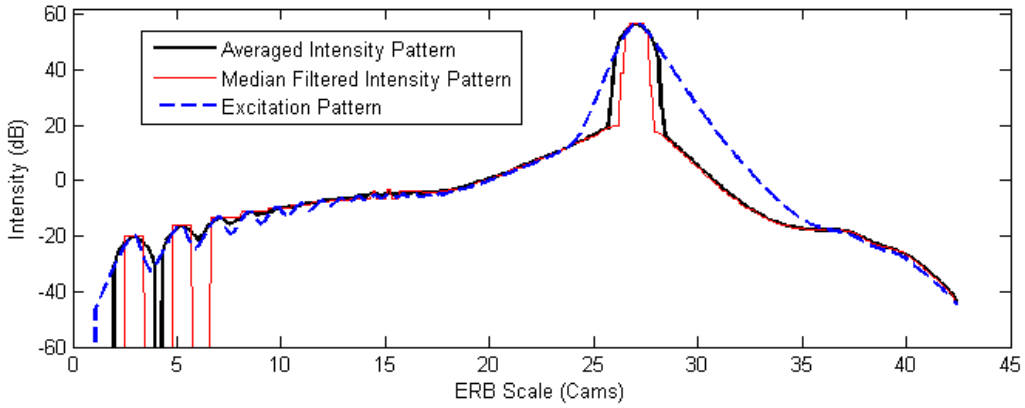


Figure 4.4: The averaged intensity pattern, median filtered intensity pattern and the excitation pattern of a frame of sinusoid of frequency 4 kHz.

In Figure 4.3, the top plot shows the intensity pattern and the averaged intensity pattern of a 4 kHz sinusoid. The bottom plot shows the reference excitation pattern, the pruned detector locations obtained by choosing the locations of maxima and minima (depicted as '*') [25], and the estimated excitation pattern as the interpolated curve. It can be seen that many detectors critical to accurately reproducing the original excitation pattern are not chosen. For the purposes of loudness estimation, the accumulation of errors during integration of specific loudness results in a significant error in the loudness estimate. Hence, attempts must be made to choose more detectors which better capture the shape of the excitation pattern curve.

4.1. Estimating the Excitation Pattern from the Intensity Pattern

The excitation at a detector location strongly depends on the energy of $S_x^c(\omega)$ within the bandwidth (i.e., the ERB) of the detector. It is higher when the magnitudes of frequency components of the signal in the ERB are higher. This can be observed in Figure 4.1, where rises and falls in the excitation pattern closely follow those of the intensity pattern. Moreover, it is observable that sharp transitions in the intensity pattern

correspond to steep transitions in the excitation pattern. Detector locations at to these transitions must also be chosen to accurately capture the shape of the excitation pattern.

To ensure retention of sharp transitions in the intensity pattern and yet effectively smoothen the pattern, median filtering is more effective than averaging. This is illustrated in the plots in Figure 4.4. The median filtered intensity pattern $Z(k)$ better captures the sharp rises and falls in the intensity pattern.

$$Z(k) = \text{median}(\{I(k\text{-}2)\ I(k\text{-}1)\ I(k)\ I(k\text{+}1)\ I(k\text{+}2)\}) \tag{4.3}$$

This is particularly useful when there are strong tonal components in the signal, such as sinusoids and music from single instruments. When the intensity pattern does not have sharp discontinuities, the filtered patterns are smoother and closely follow the excitation pattern.

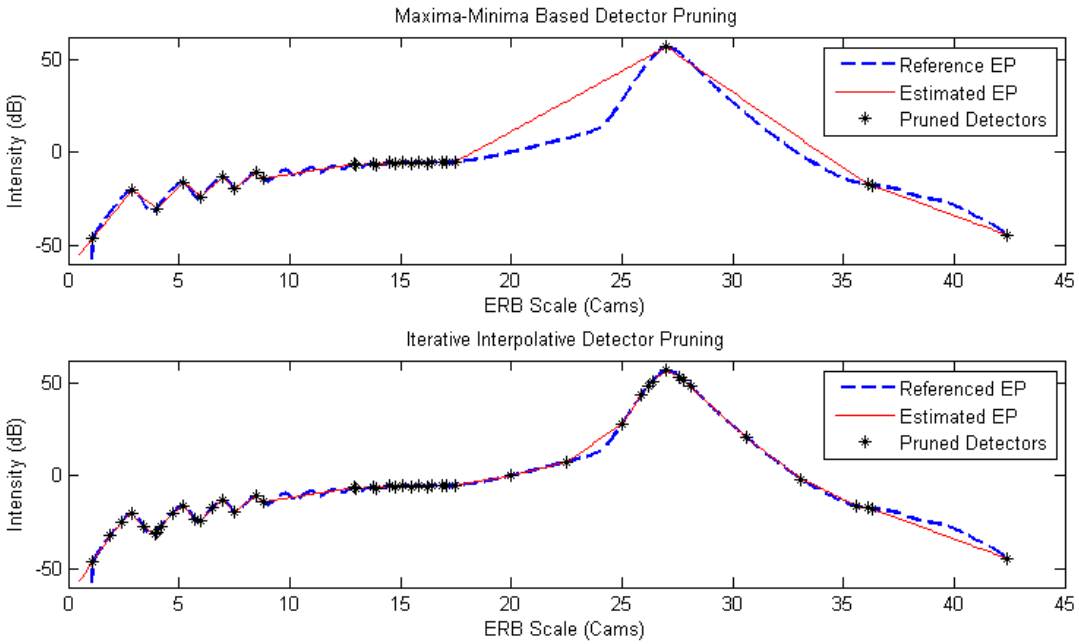


Figure 4.5: The comparison of the excitation pattern estimated through Approach 1 (top) and the proposed pruning method (bottom).

4.1.1. Interpolative Detector Pruning

In order to capture salient points in addition to the maxima and minima of the averaged intensity pattern $Y(k)$, the following method is adopted. The initial pruned set is chosen to be $L_e=\{d_k | \frac{\partial Y(k)}{\partial k} = 0$ or $\frac{\partial Z(k)}{\partial k} = 0, k = 1, 2 \ldots D\}$ and the pruned excitation pattern sequence E_e is computed. If the first difference of the excitations is high in any location with a large separation of pruned detectors at that location, then, more detectors are chosen in between these two detectors.

$$E_e = \{(d_k, E(k)) | \, d_k \in L_e, k = 1,2, \ldots D\} \tag{4.4}$$

For any two consecutive pairs $(d_m, E(m))$ and $(d_{m+n}, E(m+n+1)) \in E_e$, if $|E(m+n+1) - E(m)| > E_{thresh}$ and $|d_{m+n+1} - d_m| > d_{thresh}$, then the detectors $\{d_k \mid k = m+P, m+2P, \ldots, k < m+n+1\}$ are chosen and L_e is reassigned as shown in equation (4.5). The value of P in the experimental setup was chosen to be 25. E_{thresh} was chosen as 30 dB and d_{thresh} was chosen as 5.0. Z_{thresh} was chosen as 10.

$$\begin{aligned} L_e = \{d_k \mid \frac{\partial Y(k)}{\partial k} = 0 \text{ or } \frac{\partial Z(k)}{\partial k} > Z_{thresh}, k = 1, 2 \ldots D\} \\ \cup \{d_k \mid k = m+P, m+2P, \ldots, k < m+n+1\} \end{aligned} \tag{4.5}$$

An example is shown in Figure 4.5 for comparing the excitation patterns estimated from the original pruning method involving choosing only detectors at maxima and minima of the intensity pattern and the proposed interpolative pruning approach. For convenience, the original pruning approach is henceforth denoted “Pruning Approach I”, and the proposed scheme “Pruning Approach II”. It can be seen from the figure that Pruning Approach II produces an estimate of the excitation pattern which better resembles the reference pattern, when compared to that of Pruning Approach I. Capturing

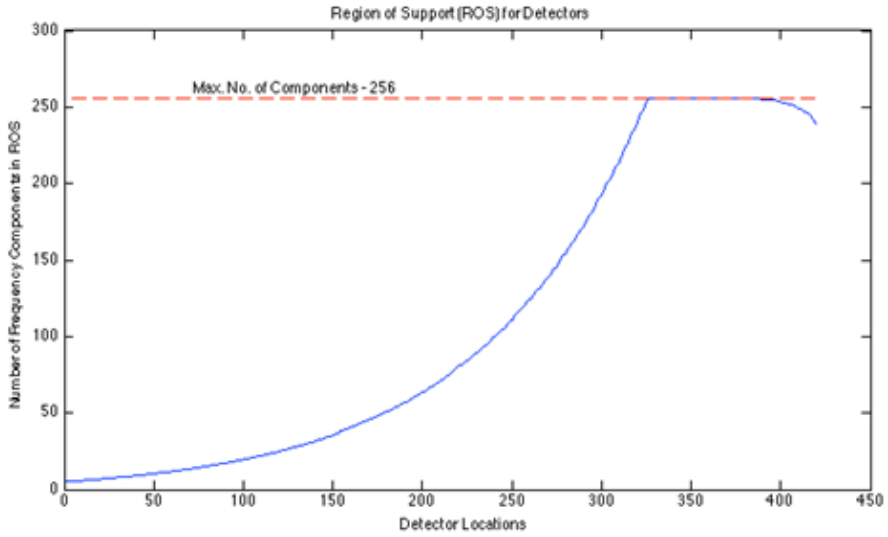


Figure 4.6: The region of support (ROS) of detectors in the current experimental setup.

the additional detectors is useful at sharp roll-offs in the excitation pattern. Such patterns can be commonly produced by tonal and synthetic sounds.

4.1.2. Exploiting Region of Support of Auditory Filters in Computation Pruning

The auditory filters, as already discussed, are frequency selective bandpass filters. Hence, by exploiting their limited regions of support, huge computational savings can be achieved. The regions of support for the filters $W(k,i)$ (denoted by $ROS(W(k,i))$) given an N-point FFT, are shown in Figure 4.6. It is seen that the region of support is small for the lower detector locations and gradually rises for detectors at higher center frequencies. Hence, choosing more detectors at lower center frequencies does not add significant computational complexity as opposed to choosing detectors at higher center frequencies.

4.2. Simulation and Results

In this section, results of simulations for evaluating the performances of the proposed pruning technique are presented and compared with the original pruning

Table 4.1: Categories of sounds in the Sound Quality Assessment Material (SQAM) database and the indices of their tracks **[64]**.

Sound Category	**Track Indices**
Alignment Signals	Tracks 1 and 2
Artificial Signals	Tracks 3 to 7
Single Instruments	Tracks 8 to 43
Vocal	Tracks 44 to 48
Speech	Tracks 49 to 54
Solo Instruments	Tracks 55 to 60
Vocal & Orchestra	Tracks 61 to 64
Orchestra	Tracks 65 to 68
Pop Music	Tracks 69 and 70

method. Performance metrics include accuracy of estimated loudness and the excitation pattern, and the empirical complexity of the algorithms. The experimental setup is described below, followed by discussion of the results.

4.2.1. Experimental Setup

In the experiments, signals from the Sound Quality Assessment Material (SQAM) database were used as auditory stimuli to examine the pruning algorithm's performance. The SQAM database consists of 70 sounds tracks which are categorized as described in Table 4.1 [64]. The SQAM database is comprised of variety of audio clips, which can be used to test the pruning scheme's performance with different types of spectra. The sounds in the SQAM database were recorded at a sampling frequency of 44.1 kHz in stereo mode. The experimental setup involved estimation of the monaural loudness of the signals, for which the signals were converted to single channel mode.

Frames of 512 samples (amounting to a duration of 11.61 ms) of the signals were provided as input to the loudness estimation algorithm. The spectra of frames were

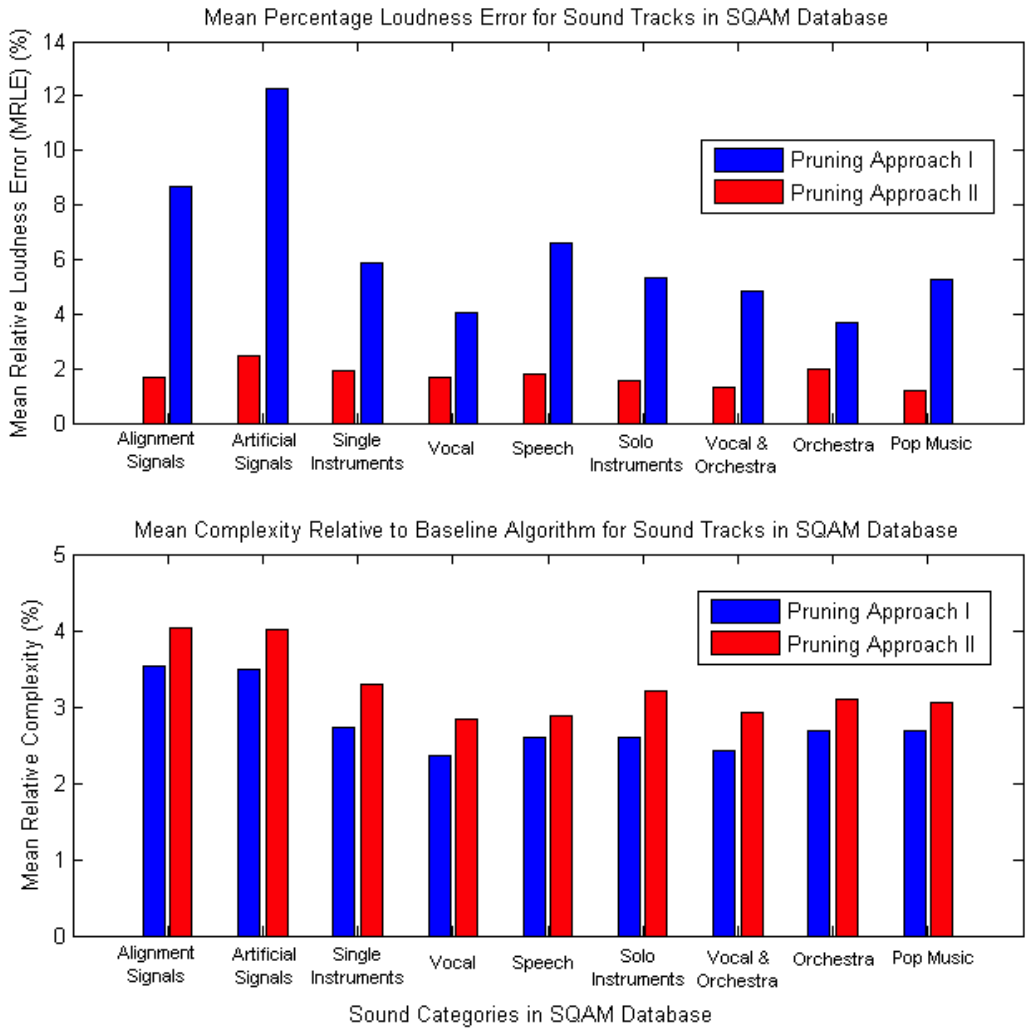


Figure 4.7: Comparison of MRLEs of Pruning Approaches I and II for sounds in the SQAM database (top). The corresponding complexities relative to the baseline algorithm are shown (bottom).

computed using a 512 point FFT (i.e., $N = 256$). The number of detectors in the reference set was chosen to be $D = 420$, uniformly spaced in the ERB scale.

4.2.3. Performance of Proposed Detector Pruning

The results of the proposed pruning technique are presented here. The loudness estimation performance of the proposed algorithms is measured in terms of their Mean

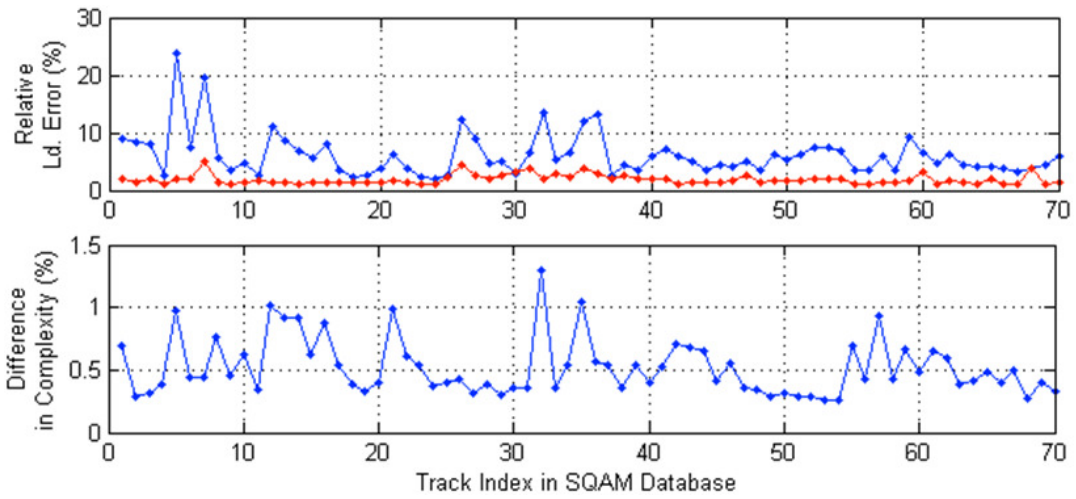


Figure 4.8: Comparison of MRLEs of Pruning Approaches I and II for individual sound tracks in the SQAM database (top). The corresponding complexities relative to the baseline algorithm are shown (bottom).

Absolute Loudness Errors (MALE) and Mean Relative Loudness Errors (MRLE) in the SQAM database.

$$\text{MALE} = \frac{1}{\text{P}}\sum_{i=1}^{\text{P}} |L_i - \hat{L}_i| \text{ ,for P frames in the signal} \tag{4.6}$$

$$\text{MRLE} = \frac{1}{\text{P}}\sum_{i=1}^{\text{P}} \frac{|L_i - \hat{L}_i|}{L_i} \text{ (or) } \frac{1}{\text{P}}\sum_{i=1}^{\text{P}} \frac{|L_i - \hat{L}_i|}{L_i} \times 100 \text{ \% (mean percentage error)} \tag{4.7}$$

$$\text{Max. ALE} = \max |L_i - \hat{L}_i| \text{ , } i = 1, \ldots \text{P} \tag{4.8}$$

The excitation pattern error is measured by the Mean Relative Excitation Error (MREE), as expressed in the equation below.

$$\text{MREE} = 20 \log_{10} \left\{ \frac{1}{\text{P}.D} \sum_{i=1}^{\text{P}} \sum_{k=1}^{D} \frac{|E_i(k) - \hat{E}_i(k)|}{E_i(k)} \right\} \tag{4.9}$$

The complexity of the pruning algorithm relative to the baseline approach is computed as:

Table 4.2: Maximum Loudness and Excitation Pattern Error performance comparison of Pruning Approach II with Pruning Approach I for categories of sounds in the SQAM database.

Sound Category	(Max. ALE) (Sones)		Mean EP Error (dB)	
	Pruning Approach I	Pruning Approach II	Pruning Approach I	Pruning Approach II
Alignment Signals	1.1852	0.2874	-9.995	-17.616
Artificial Signals	2.5716	0.4497	-9.912	-19.268
Single Instruments	2.0567	0.8380	-11.421	-17.650
Vocal	1.2124	0.8686	-12.098	-15.974
Speech	1.6845	0.8703	-11.973	-17.293
Solo Instruments	1.2751	0.5618	-11.823	-17.256
Vocal & Orchestra	2.1875	1.0169	-11.822	-16.979
Orchestra	2.1602	0.5590	-12.736	-17.594
Pop Music	1.3017	0.4277	-12.446	-17.990

$$\text{Relative Complexity } (C_r) = \frac{\sum_{k:d_k \in L_e} \left(\sum_{ROS(W(k,i))} 1 \right)}{\sum_{k=1}^{D} \left(\sum_{ROS(W(k,i))} 1 \right)} \times 100\ \% \qquad (4.10)$$

The complexity of the baseline algorithm is $O\left(\sum_{k=1}^{D} \left(\sum_{ROS(W(k,i))} 1 \right) \right)$ and that of the pruned algorithm is $O\left(\sum_{k:d_k \in L_e} \left(\sum_{ROS(W(k,i))} 1 \right) \right)$.

The top plot in Figure 4.7 shows the MRLE for Pruning Approach I and the Pruning Approach II for sound categories in the SQAM database. Shown in the lower plot are the corresponding complexities of the two schemes. It can be seen that the complexity of Pruning Approach II is about 0.5% more than that of Pruning Approach I. But significant reduction in the MRLE error percentage is achieved by this small addition in complexity.

The top plot in Figure 4.8 compares MRLEs of Pruning Approaches I and II for audio tracks in the SQAM database. The bottom plot shows the corresponding mean relative complexities of the two approaches. It can be seen that the Pruning Approach II significantly reduces the percentage errors in most sounds over the first approach. It can be seen that corresponding to significant percentage error reductions, significant increases in the complexity for can be seen Pruning Approach II. In most other cases, the complexity increase is not significant. Thus, the interpolative approach increases computations whenever required.

Table 4.2 shows the maximum loudness error of the two approaches and again evinces the improved accuracy of the interpolative pruning approach. Also, the excitation pattern error is reduced in Pruning Approach II.

Chapter 5

LOUDNESS CONTROL

5.1. Background

Control of perceptual loudness of audio must involve variation of the intensity of the signal over time such that the perceived spectral content of the signal is preserved as much as possible, while the perceived loudness of the signal over the duration of the signal is at a level desirable to the user. Also, it must be ensured that this is achieved without introducing any significantly perceptible audio artifact in the signal.

A common technique used to control the intensity of sound is the Automatic Gain Controller (AGC). The AGC modifies the incoming signal's power to a desired level by appropriately scaling it. If the AGC processes the incoming audio input frame-by-frame and modifies the signal power for every frame, while concatenatively resynthesizing the signal, it must be ensured that the variation of the output signal power across consecutive frames is smooth enough to avoid discontinuities at frame boundaries which may give rise to undesirable audible artifacts. For this purpose, the scaling factor for the signal power is limited by range compression or other such mechanisms.

While the AGC effectively controls the signal intensity, it is not tantamount to controlling the perceived loudness of the signal, as the loudness has a non-linear relationship with the intensity and a complicated dependence on the spectrum of the signal, both of which are governed by the properties of the human auditory system. Hence, the loudness control system is to be designed taking into account the characteristics of the human auditory system and hence, the resulting relationship between the perceived loudness of the signal and its *content*.

One method of designing this system is to establish a mapping from the parameters that characterize any signal to its perceptual loudness. Based on this mapping, the signal intensity (or signal power) can be scaled to produce an output signal of the desired loudness. This technique essentially gives the conventional AGC system knowledge about the effect of gain modification on the loudness of the signal. For the rest of the article, the terms intensity, power and RMS value will be used interchangeably to denote the signal power. The signal scaling factor may also be referred to as the gain.

Several conventional loudness control systems implement volume control through a wideband manual gain control with a bass boost or a treble boost to equalize the intensities of the respective frequencies according to the Equal Loudness Contours. Such control does not preserve the tonal balance at all listening levels. But using auditory models, more precise tonal balance can be achieved at all levels through sub-band gain control.

The Moore and Glasberg model itself is a mapping from the signal content (viz. the spectrum) to its loudness. But because of the non-linear transformation of the excitation patterns of individual bands to form the loudness pattern, the relationship between the signal intensity and the loudness is very complex. Thus, it is difficult to analytically derive a closed form solution and instead, iterative methods would have to be resorted to for estimating the target gain required to achieve a desired loudness.

Instead, in the approach described in this chapter, an approximate relation between the power and the loudness of a signal is derived by fitting functions to empirically obtained data from the Moore and Glasberg model. Since the Moore and Glasberg model represents the perceptual loudness of a signal as a function of the signal's

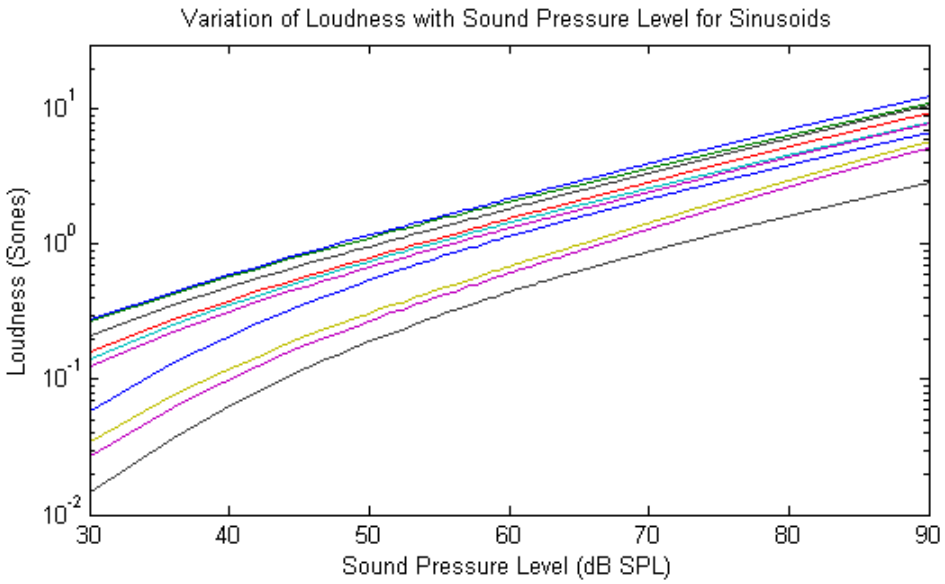


Figure 5.1: Loudness versus sound pressure level for a set of sinusoids.

spectrum, the attempt to express loudness as a function of the signal power would have to result in a parametric representation of the loudness in terms of the intensity, where the parameters would be functions of the signal's normalized spectral shape and independent of the intensity.

5.2. Loudness Based Gain Adaptation

The model described herein provides such an explicit mapping between a signal's intensity and its loudness, and the equation is a function of a single parameter which is dependent on the spectrum of the signal. As it will be shown, the model is another form of Steven's Law, which in this case, has been derived as an empirical approximation of the mapping provided by the Moore and Glasberg model.

By obtaining the loudness estimates of the Moore-Glasberg model for a signal of varying intensities, and performing the same for various types of signals, empirical mappings of signals to their loudness are obtained. For instance, when the Moore -

Glasberg model estimates the loudness of different sinusoids for varying gains, it gives us information about the variation of loudness with signal power for sinusoids and how the relationship changes with changing frequency. The plot in Figure 5.1 illustrates the obtained data from a set of such sinusoids.

In Figure 5.1, the topmost curve corresponds to a sinusoid of about 4 kHz. That is, for a fixed signal power, the sinusoid at 4 kHz has the highest loudness. This is in accordance with the equal loudness contours.

Similarly, narrowband noise signals were created with different frequencies and a bandwidth equaling one critical band around the respective center frequency. Their loudness values were estimated from varying gains by the Moore-Glasberg model and a plot of the same data is shown in Figure 5.2. Since these signals are noise, different frames of the signal would exhibit random fluctuations in the spectral shape. Hence, to get a good estimate of the loudness, averaging of the processed signal needs to be done at some stage in the model. Averaging of the excitation patterns can be performed before estimating the loudness of the signal. Since, the operations in the auditory model till the stage of the calculation of excitation pattern is linear it is logical to average the excitation pattern over multiple frames.

The excitation patterns are then non-linearly transformed as elaborated in the description of the Moore-Glasberg model. This would give a good estimate of the loudness. In another approach, the loudness of the signal at different frames can be averaged to get an estimate of the loudness. This is a reasonable approach if the loudness fluctuations of the frames can be considered insignificant compared to the mean level of the loudness. In the experiment of estimating loudness of narrowband noise signals, the

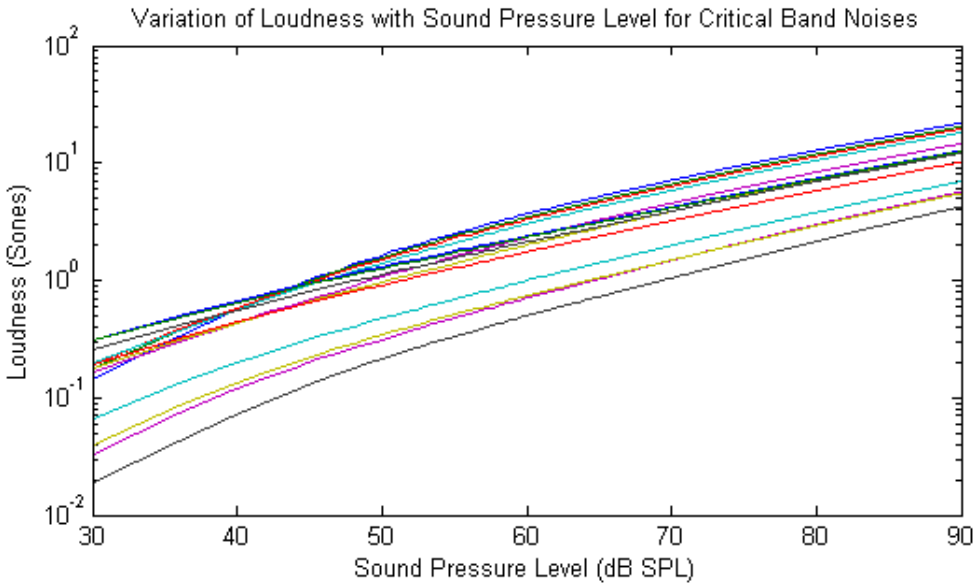


Figure 5.2: Loudness versus signal power for a set of critical band noises.

short term loudness is averaged over multiple frames to obtain an estimate the loudness for a fixed signal power.

It is evident from the Moore-Glasberg model that for a *fixed* spectrum, the loudness of a signal is a function of the signal power. On this premise, it is obvious that all signals whose spectra can be considered invariant over a given time period display a consistent behavior in the variation of loudness with signal power. It can be observed that as the signal power increases, the loudness is less sensitive to changes in the signal power. This is in consistency with Steven's Power Law, based on which the mapping from the excitation pattern to the loudness pattern is derived in the Moore-Glasberg model.

Steven's Law relates signal intensity to loudness by a power law, which in mathematical form is $L = qg^{\alpha}$, where g is the signal's power. α and q are parameters dependent on the signal. This equation can be modified as follows.

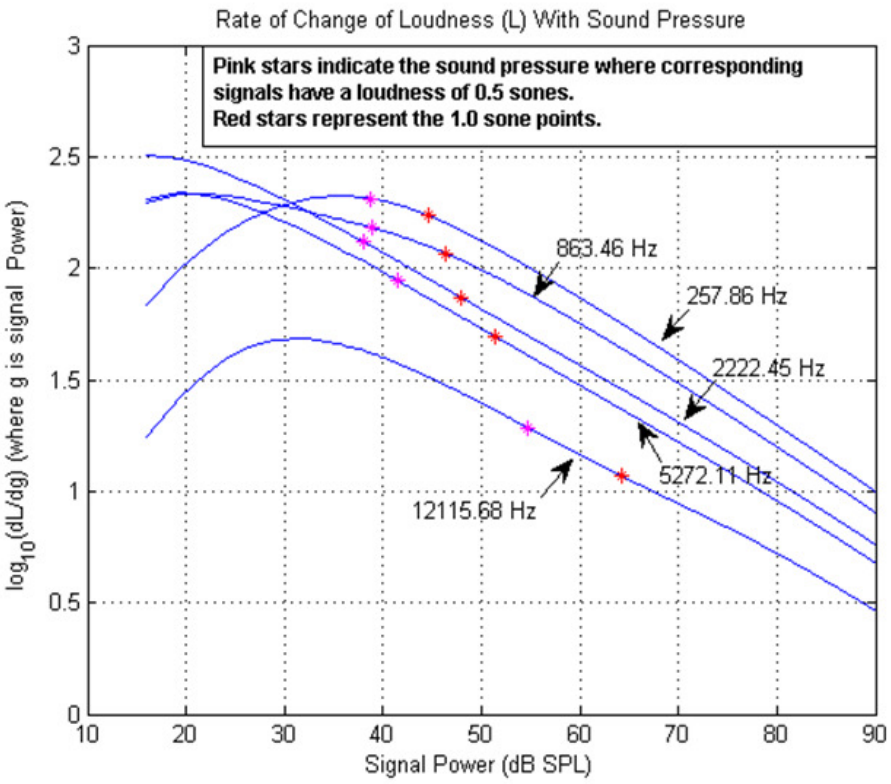


Figure 5.3: Rate of change of loudness with sound pressure for critical bandwidth noise signals, whose corresponding center frequencies are mentioned in the figure.

$$\log_{10}\frac{dL}{dg} = (\alpha\text{-}1)\log_{10} g + \log_{10}(\alpha q) \tag{5.1}$$

The logarithm of the derivative of loudness w.r.t. the signal power vs. the logarithm of the sound pressure is shown in Figure 5.3 for a set of critical-bandwidth noise signals. It is seen that beyond about 40 dB SPL, the rate of increase of loudness with gain reduces has the power increases.

It is observed that when loudness levels are higher than 0.5 sones, the curves for the set of signals are nearly parallel to each other. The behaviors of the curves are similar, except for their levels, which may be attributed to their spectral content. These inferences from the curves form the basis for the model derived here.

Each curve is modeled using a piecewise linear model, assuming all the curves to be approximately parallel to each other in the regions where loudness levels are significant enough. The range of the signal power is split so that each segment of the curve can be approximated to a straight line. Since all the curves are parallel to each other, the corresponding segments of all the curves in the piecewise linear model would have the *same* slope. These slopes can be easily determined by fitting the data in each piece-wise linear segment to a straight line. The intervals separating the line segments in the curves' model were chosen by visual inspection and the goodness of the fit was acceptable for all the signals testing the model.

Consider any two consecutive straight lines on the curve with the following equations with p as the slope and q as the constant, where g is the signal power (RMS of the signal).

$$\log\left(\frac{dL}{dg}\right) = p_k \log(g) + q_k \tag{5.2}$$

$$\log\left(\frac{dL}{dg}\right) = p_{k+1} \log(g) + q_{k+1} \tag{5.3}$$

If they intersect at the RMS value g_t, then $p_k \log(g_t) + q_k = p_{k+1} \log(g_t) + q_{k+1}$. That is, $q_{k+1} = (p_k - p_{k+1}) \log(g_t) + q_k$. Hence, q_{k+1} is expressible as a function of q_k. In the same manner, the constant of one straight line can be represented in terms of the next straight line. Now, by splitting a given curve into m piece-wise linear segments, all the constants in all the individual straight line equations can be represented in terms of the first straight line's constant.

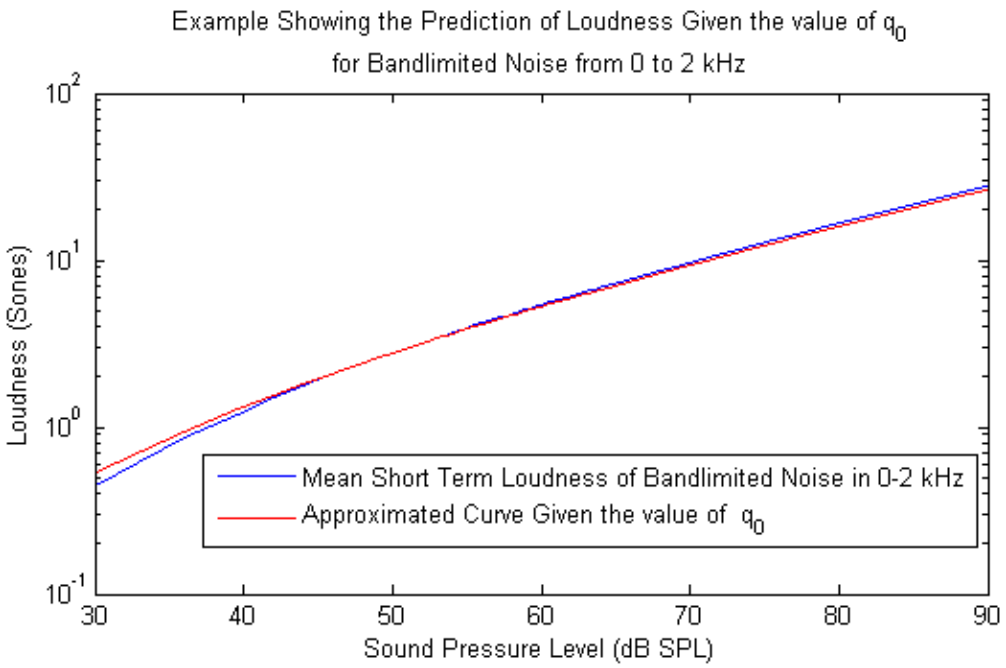


Figure 5.4: The comparison between the experimentally obtained short term loudness variation with frequency for the band-limited noise from 0-2kHz (the red curve) and the same curve predicted by the proposed parametric model mapping the signal intensity to loudness.

$$\log\left(\frac{dL}{dg}\right) = p_0 \log(g) + q_0 \text{ , if } g < g_0 \tag{5.4}$$

$$\log\left(\frac{dL}{dg}\right) = p_i \log(g) + q_i \text{ , if } g_{i-1} \le g \le g_i \text{ for } i = 1,2,\ldots,m\text{-}2 \tag{5.5}$$

$$\log\left(\frac{dL}{dg}\right) = p_{m-1} \log(g) + q_{m-1} \text{ , if } g \ge g_{m-2} \tag{5.6}$$

Now, $q_1, q_2, q_3, \ldots q_{m-1}$ can be represented in terms of q_0. Since the slopes p_i's are obtained through data fitting as mentioned above, the only parameter left to be determined is the constant of the first straight line, q_0. Hence, the relation between loudness and signal power derived from these equations is reduced to a parametric model relating the signal power to its loudness, with a single parameter q_0 that depends on the

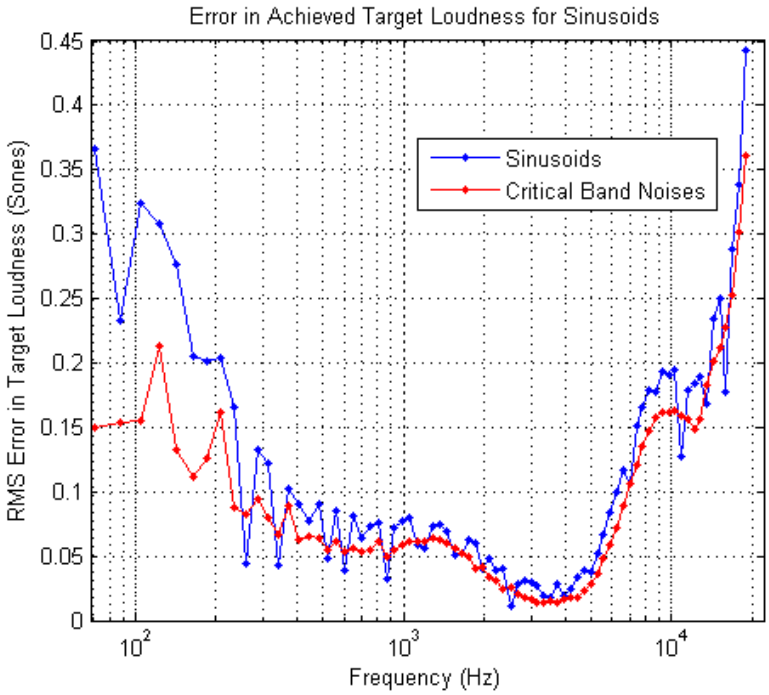


Figure 5.5: RMS error between achieved loudness and target loudness for sinusoids and narrow-band noise signals.

signal's content (spectrum). The above equations can be suitably integrated to determine the explicit relation between loudness and signal power, as shown below.

$$\frac{dL}{dg} = e^{q_0} g^{p_0} \text{ , if } g \le g_0 \tag{5.7}$$

$$L(g) = \int_0^g e^{q_0} v^{p_0} dv = \frac{e^{q_0} g^{p_0+1}}{p_0+1} \text{ , if } g \le g_0 \tag{5.8}$$

Similarly, the relation can be derived for other regions of g, as shown in the equation below.

$$L(g) = L(g_{i-1}) + \int_{g_{i-1}}^{g} e^{q_i} v^{p_i} dv \quad \text{, if } g_{i-1} \le g \le g_i \text{ for } i > 0 \tag{5.9}$$

It is thus possible to successfully separate the signal intensity and the spectral content in the equation into different variables. By achieving this, the signal-dependent

parameter in the equation can be estimated when the loudness of the signal and the intensity (or RMS value of the signal) are known. The loudness can be estimated online using the Moore-Glasberg model, and the signal intensity can also be easily computed. Hence, the parameter q_0 can be estimated online and can be used to estimate the required gain to achieve a desired perceptual loudness.

Shown in Figure 5.4 is the variation of the short term loudness with signal power for a sound clip containing bandlimited noise from 0-2kHz, which is the red curve. The blue curve is the variation of the short term loudness with power for the same signal predicted by the model derived above. It can be seen that the model predicts the loudness variation with good accuracy.

Upon setting a target loudness level, a set of sinusoids and narrowband noise signals with varying intensities were scaled to reach the target loudness, where the scaling factor was determined by the proposed model. This experiment was conducted was varying target loudness levels & varying intensities of the input signals, and the loudness of the scaled output signals were compared against the target loudness. The errors in the loudness are plotted in Figure 5.5.

5.3. Preserving the Tonal Balance

From the Equal Loudness Contours, it is known that by changing the intensity of a signal, the shape of the auditory pattern is distorted because the variation of sensitivity with intensity is different for different critical bands. This structure of the auditory pattern is also referred to as 'tonal balance'. A common example of tonal balance distortion is speech, which sounds *boomy* when the volume is raised to a large level. This is because the sensitivity of lower frequencies rapidly increases at high intensities and boosts the

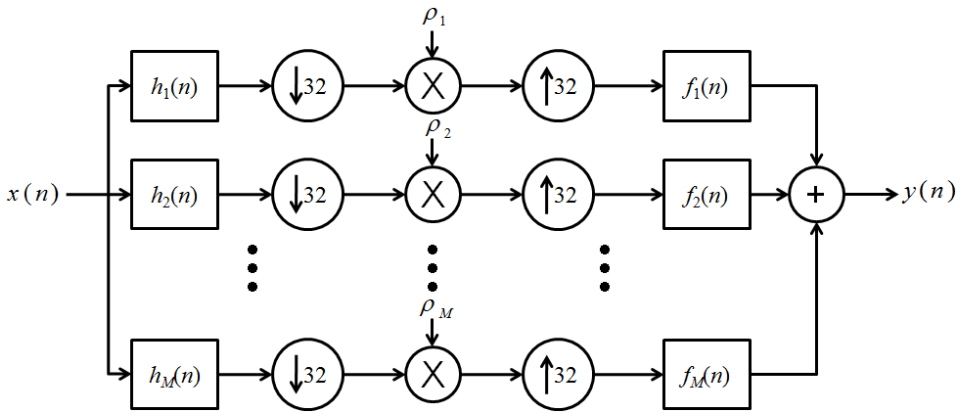


Figure 5.6: Sub-band gain control using an analysis-synthesis framework.

perception of lower frequencies. Hence, in addition to the wideband gain control described above, gain control for narrow frequency bands is required to preserve the tonal balance of the signal.

The narrowband gain control is implemented in the following manner, as illustrated in Figure 5.6. For narrowband gain control, the signal can be filtered by a bank of pseudo-QMF filters as part of an analysis-synthesis system. For an M-channel filter bank, the filters can have a tap length of $L=MK$, where K is the length of filters in the polyphase filter implementation of the filters. The filters are defined by the following equation.

$$h_k(n) = h(n)\cos\left(\frac{\left(k+\frac{1}{2}\right)\left(n-\frac{M-1}{2}\right)\pi}{N}+\varphi_k\right),\ k = 0,\ldots M-1 \tag{5.10}$$

The prototype lowpass filter $h(n)$ satisfies the following conditions.

$$\begin{aligned} |H(\omega)|^2 &= 0 \text{ for } |\omega| \geq \pi/M \\ |H(\omega)|^2+|H(\pi/M-\omega)|^2 &= 2 \text{ for } |\omega| < \pi/2M \end{aligned} \tag{5.11}$$

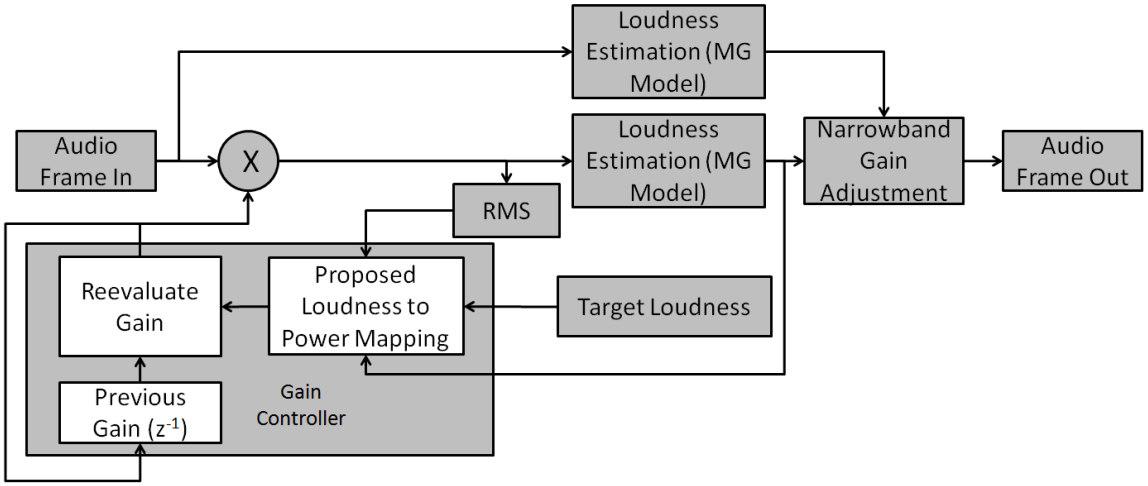


Figure 5.7: Block diagram of the loudness control system.

The phases are constrained by the equation φ_k-φ_{k-1}=(2p+1)π/2, where p is an integer. The corresponding synthesis filters are defined as $f_k(n)=h_k(L\text{-}1\text{-}n)$.

The gains in the individual sub-bands ρ_k are computed from the specific loudness within the individual sub-bands. If each individual sub-band of the original audio has the specific loudness $S_b(k)$, then by Steven's Law, its relation with the intensity g_k of that band is $S_b(k)=\tau_k g_k^{2\alpha}$. Similarly, if the wide-band gain controlled audio has a corresponding specific loudness $S_b'(k)$, then $S_b'(k)=\tau_k g_k'^{2\alpha}$. Assuming that the target intensity for preserving tonal balance is g_k^t, then,

$$g_k^t=\left(\frac{S_b(k)}{S_b'(k)}\cdot\frac{S_b'(k_0)}{S_b(k_0)}\right)^{\frac{1}{2\alpha}} g_k' . \tag{5.12}$$

Thus,

$$\rho_k=\left(\frac{S_b(k)}{S_b'(k)}\cdot\frac{S_b'(k_0)}{S_b(k_0)}\right)^{\frac{1}{2\alpha}} . \tag{5.13}$$

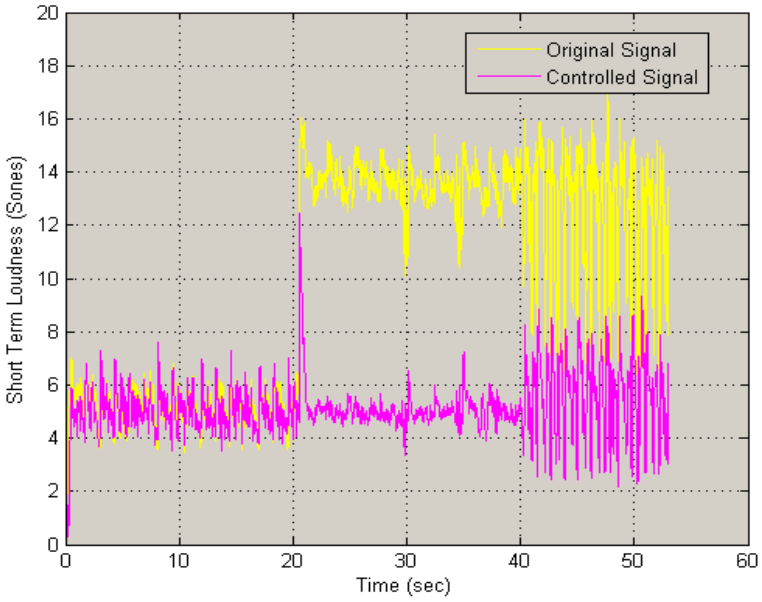


Figure 5.8: Loudness of a music file over time shown by the yellow graph is controlled by the loudness control system to produce an output with controlled loudness, which is plotted as the graph in magenta.

5.4. Loudness Control System Setup

The *loudness-aware* gain controller based on the mathematical model derived above is incorporated in the loudness control system shown in the setup in Figure 5.7. The incoming signal is processed frame-wise by the system. A 512-point frame of the signal is processed by the loudness estimation algorithm. The estimated loudness and the signal power are used to compute the required scaling factor to achieve the desired loudness level in the output.

This gain is then applied to the subsequent frame of the signal. This is a feedback system, where the current frame's loudness is measured and the signal is scaled accordingly in the consecutive frame. An assumption of quasi-stationarity of the incoming signal is made, which justifies the relevance of the *reactive* nature of the system in controlling the loudness.

As an example, a sound clip was passed to the loudness control system. As it can be seen in Figure 5.8, a portion of the original signal (plotted as the yellow graph) has a significantly higher loudness than the rest. The output of the loudness control system suppresses the increase in the loudness, successfully controlling it.

Chapter 6

SPEECH/AUDIO PROCESSING FUNCTIONALITY IN IJDSP

To be able to support speech and audio processing and visualization capabilities, iJDSP requires extensive enhancements in certain capabilities. One among these is the ability to process long signals, which are common in some audio processing schemes such as MP3 compression and psychoacoustic analysis. But in the conventionally designed functions of iJDSP, the maximum signal length permitted falls short of the requirements of such functions. For this purpose, a software framework was developed to create blocks which are capable to processing long signals, and yet are compatible with the conventional blocks in signal transmission, allowing easy creation of blocks that can smoothly interact with all other blocks without software bugs or exceptions, regardless of the signal handling capabilities of the blocks.

Another important capability is to perform frame-by-frame processing, which is a common technique adopted for speech or audio signals. The frame-by-frame processing ability was created for the relevant blocks to be actively used for speech and audio processing. In addition, blocks for data visualization such as plots and frequency response were also enhanced by augmenting them with frame-by-frame visualization capabilities. These enhancements will be described in detail in the following sections.

Presented in Section 2 this report are the aforementioned enhancements appended to the architecture of iJDSP and the expansion of its functionalities to perform more sophisticated simulations. Section 3 describes the functional blocks created in iJDSP to illustrate basic speech and audio processing concepts.

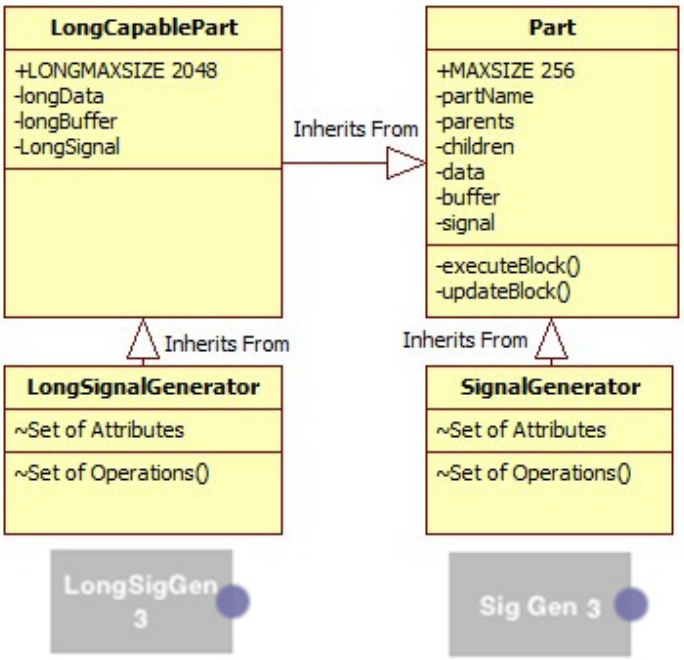


Figure 6.1: UML diagram describing the inheritance of the class 'Part' by 'LongCapablePart'. The 'SignalGenerator' block inherits from 'Part'. The 'LongSignalGenerator' block inherits from 'LongCapablePart'.

6.1. The iJDSP Architecture: Enhancements

This section discusses details of the supporting software framework developed for enhancing the capabilities of iJDSP. These software constructs have resulted in significant increase in the sophistication of the architecture, and an additional level of abstraction in the types of function blocks supported by iJDSP. This software design pattern is also recommended to be adopted by future developers involved in the iJDSP project, who are required to create new abstractions for function blocks for providing architecturally novel capabilities.

6.1.1. Framework for Blocks with Long Signal Processing Capabilities

iJDSP is built extensively using object-oriented programming practices using objective-C. Objective-C is a programming language created by augmenting C with

object-oriented programming primitives. In iJDSP, a class called 'Part' is defined, which is the skeleton class describing a function block. It defines basic *attributes* of the function blocks and defines *operations* to manage parent-and-child relationships between blocks in a block diagram, and signal flow through the blocks. In essence, it defines the core attributes and operations of a typical function block. A DSP block is created by inheriting the 'Part' function, and suitably defining additional attributes and operations specific to the particular block. The operations of the block define how the block acquires data from specific input pins, process them and to which pins it dispatches the desired outputs. The attributes of the block define the state of the block, which parameterize the signal processing operations performed by the block.

The essential attributes and operations of the 'Part' class are shown in the UML diagram in Figure 6.1. MAXSIZE denotes the maximum size of signal can be received as input, or that can be given as output, by a block built around the 'Part' class. For instance, the 'SignalGenerator' block, which creates signals and provides them as output through its output pin, inherits the 'Part' block. Hence, it can only provide output signals that can be up to 256 samples long. It is not advisable to simply increase the ability of the 'Part' class to be able to transmit longer signals because given the architecture of the application, it would result in excessive memory usage by every block, and reduce the number of blocks that can be used. Moreover, majority of the blocks do not require long signal transmitting capabilities. Hence, it would be wastage of resources.

To overcome these issues, a new class 'LongCapablePart' was created. This inherits 'Part', and customizes itself to handle long signals, while retaining the capabilities of connecting to 'Part' based blocks. Functions in 'Part' were suitably

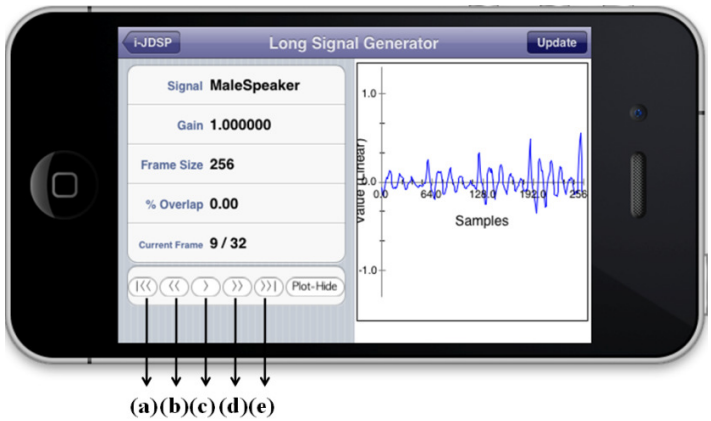


Figure 6.2: The interface for configuring the *long signal generator* block. Frames of the signal can be traversed using the playback buttons. A plot of the current frame of signal at the output pin is shown.

overridden in 'LongCapablePart' to achieve this. The blocks inheriting 'LongCapablePart' are capable of transmitting signal frames of size up to LONGMAXSIZE (2048) samples. Only blocks requiring this capability are allowed to inherit 'LongCapablePart'. For instance, the 'LongSignalGenerator' block, which generates frames of signals from a set of pre-defined speech and audio signals, can be used as a signal source for the speech and audio processing functions which would be described in the forthcoming sections. This block inherits 'LongCapablePart', hence, possessing the capability to transmit long frames.

6.1.2. Frame-by-Frame Processing and Visualization

Another major augmentation to the capabilities of iJDSP is the provision of plots, frequency response, and other such data visualization interfaces with the ability to view the data frame-by-frame, and traverse through all frames of the signal sources involved in the simulation setup.

The Long Signal Generator block configuration user interface is shown in Figure 6.2. The interface allows the user to configure parameters of the signal through the table on the left side of the screen. On the right, a plot of the current frame of signal is shown. The frames of the chosen long signal can be traversed using the playback buttons at the bottom of the table in the interface, indicated by the labels (a) – (e). As labeled in Figure 6.2, the buttons (a) and (e) respectively seek the first and last frames of the chosen signal. Buttons (b) and (d) respectively seek the previous and next frames of the signal. Button (c) traverses the signal frame-by-frame from the current frame to the last frame. The right half of the view displays the current frame of signal at the pin. The frame-wise traversal in the Long Signal Generator and such blocks with frame-wise capabilities need to be

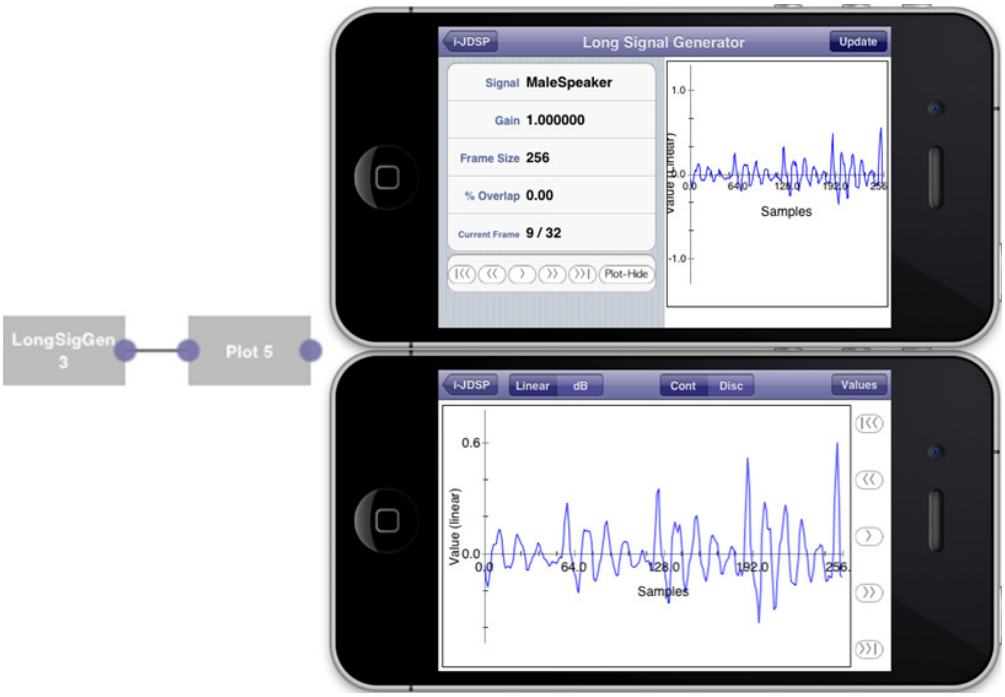


Figure 6.3: The block diagram in the figure shows signal from a Long Signal Generator block being fed to a Plot block. The top right picture shows the configuration GUI for the Long Signal Generator block. The bottom right screenshot shows the visualization interface for the Plot block.

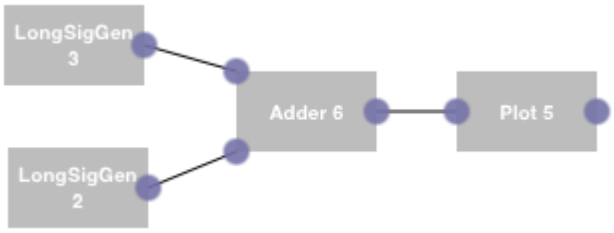


Figure 6.4: A block diagram where signals from two *Long Signal Generators* are added sample-wise and viewed in the *Plot* block.

made easier from a user-experience perspective, and GUI enhancements were introduced in many existing functionalities for this purpose.

As an example, a block diagram where a *Long Signal Generator* block is connected to a *Plot* block is shown in Figure 6.3. The Long Signal Generator has a visualization interface shown in the top right of Figure 6.3, which allows the user to traverse through the frames of the chosen signal through the playback buttons. In the Plot block's visualization interface, the graph of the frame of signal currently produced as output by the Long Signal Generator block is shown. It is convenient for the user to be able to traverse through the frames of the long signal from directly from within the Plot block's user interface, without having to access the Long Signal Generator block's configuration interface. For this purpose, the frame-wise traversal buttons were also added to the Plot block visualization interface, as shown in the bottom right screenshot in Figure 6.3. Using these buttons, one can directly control the Long Signal Generator block connected to the Plot block.

Similarly, this functionality has the sophistication of being able to handle multiple signal sources if they are present in the block diagram in the canvas. For instance, consider the block diagram shown in Figure 6.4. In this case, if a playback button in the Plot interface is pressed, then both the Long Signal Generators change their signal frames

accordingly. This is achieved programmatically by identifying all signal sources in the simulation setup that affect the blocks in the simulation setup, and then modifying the output frames of those signal sources.

6.1.3. Planned DSP Simulations

In order to illustrate specific concepts through elaborate block diagrams, certain complex block diagrams are stored as pre-defined simulation setups. Users can directly use these setups, without having to laboriously add all the individual blocks and establish all connections. For instance, the simulation setup for illustrating the concept of linear predictive coding, a popular speech coding technique, has been pre-stored in iJDSP. The simulation setup is shown in Figure 6.11, and will be discussed in detail in Section 3.3.

6.2. Developed DSP Blocks

This section describes some of the blocks developed in iJDSP for acquiring and rendering sounds, and for illustrating techniques of analyzing and processing speech/audio.

6.2.1. Signal Generation Functions

Long Signal Generator

In many applications, such as those involving long speech or audio signals, splitting the acquired signals into equally sized frames and processing them frame-by-frame is a common paradigm in digital systems. To illustrate processing of such signals, the *Long Signal Generator* function is supported in iJDSP. The *Long Signal Generator* provides a set of pre-defined long signals such as speech, music, and noise. The user is allowed to choose one from the provided set of signals. Figure 6.2 shows the interface of the *Long Signal Generator*.

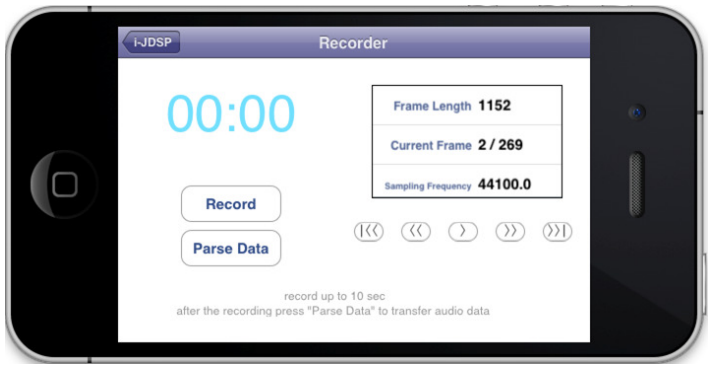


Figure 6.5: User interface for the *Sound Recorder* block.

The gain for the signal, the length of each frame and the overlap between adjacent frames can be configured by the user. The "current frame" indicates the index of the frame given as output in the format "*Current Frame / Total No. of Frames in the Signal*". The desired frame of the signal can be chosen from the playback buttons. The right half of the view displays the current frame of signal at the pin. The **Plot-Hide/Show** button allows the user to show or hide the plot view on the right half of the interface.

Sound Recorder

Apart from the signals provided in the *Long Signal Generator*, it is also beneficial for students to record speech/audio from microphones and process them frame-by-frame. The *Sound Recorder* function is provided for this purpose. Audio can be recorded from the microphone by the *Sound Recorder* as 16-bit PCM samples. The user interface of the sound recorder block is shown in Figure 6.5. The *Sound Recorder* has options to acquire sounds at three different sampling frequencies: 8000 Hz, 16000 Hz and 44100 Hz. The function's user interface has control buttons to start or stop recording, and playback control buttons to traverse through the frames of the signal. The length of the output frame can be configured by the user. The frame length, sampling frequency and the

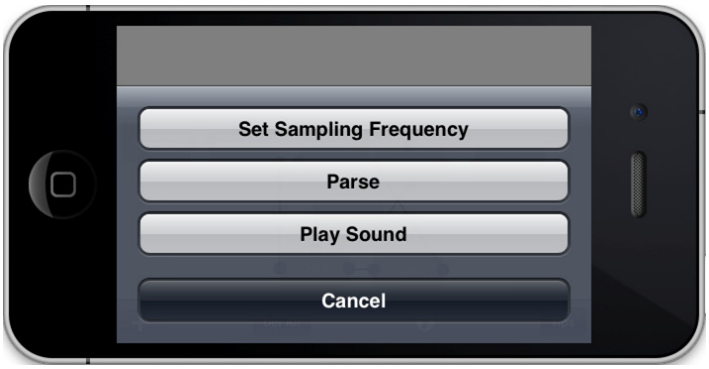


Figure 6.6: Options provided by the *Sound Player* block.

current frame index of the recorded signal being output are also displayed in the user interface. The recorded sound can be processed and played by the *Sound Player* block. The *Sound Player* block can be configured to play the received samples at any of the aforementioned sampling frequencies supported by the *Sound Recorder*.

Sound Player

The *Sound Player* function aggregates signal frames provided as input, and plays the resulting signal as audio. The sound player block can be configured by the user to play the audio at a specific sampling frequency. The options provided by the sound player for the sampling frequency are 8000 Hz, 16000 Hz and 44100 Hz. The sound player block can be connected to the output of a designed DSP block diagram in the canvas. The options provided by the sound player upon double tapping the block in the main canvas are 'Set Sampling Frequency', 'Parse' and 'Play Sound' (Figure 6.6).

The 'Sampling Frequency' option allows the user to configure the block's operating sampling frequency as any one of the aforementioned options. The 'Parse' button simulates the entire block diagram and enforces the Sound Player block to aggregate the signal frames provided as input to itself. This is achieved by the block by

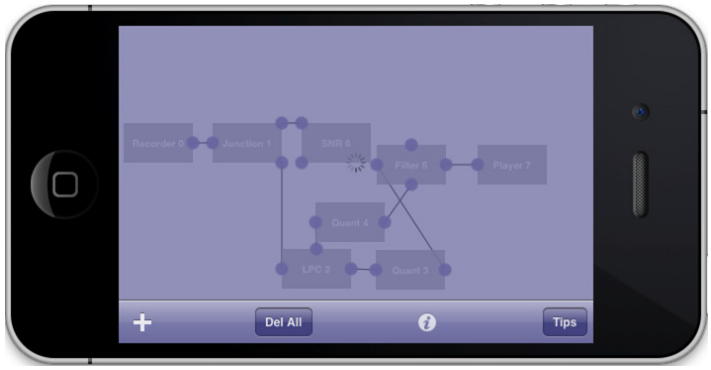


Figure 6.7: A rotating activity indicator with a translucent background is displayed while parsing through the signals generated by the sources from the Sound Player.

identifying all signal sources in the block diagram which are processed to provide signal input to the sound player, and then parsing through the signals frame-by-frame in all the signal sources. This mechanism also enforces concurrency of the system at the input of the Sound Player block. That is, the algorithm ensures that redundant execution of the block diagram does not result in intermediate results (between the completion of a single frame traversal for all the signal sources in the canvas) being incorrectly interpreted as valid frames to be aggregated by the Sound Player for rendering as audio. Hence, the user is always advised to use the Parse button at the Sound Player to aggregate signals for audio rendering.

When the input signal sources host longer signals that require time for processing through an elaborate block diagram, for the entire duration of the parsing process, a rotating activity indicator graphic appears on the screen, as shown in Figure 6.7. The 'Play Sound' button renders the aggregated audio signal at the sampling frequency with which it was parsed. Hence, it must be ensured that the sound is always parsed with the appropriate sampling frequency in order to get the intended rendering while playing it.

6.2.2. Spectrogram

Time-frequency analysis involves the study of variations in the spectral composition of signals with time. The spectrogram is a popular time-frequency representation and is particularly useful for speech/audio processing algorithms. The spectrogram captures the temporal variations in the spectrum of a signal by computing the square of the magnitude of the FFT of the signal over successive windows of time. The spectrogram is elaborated upon in detail in [65].

The Short-time Fourier Transform (STFT) of a discrete-time signal $x(n)$ of a length of M samples is evaluated from the following expression, using a W-point window $w(n)$.

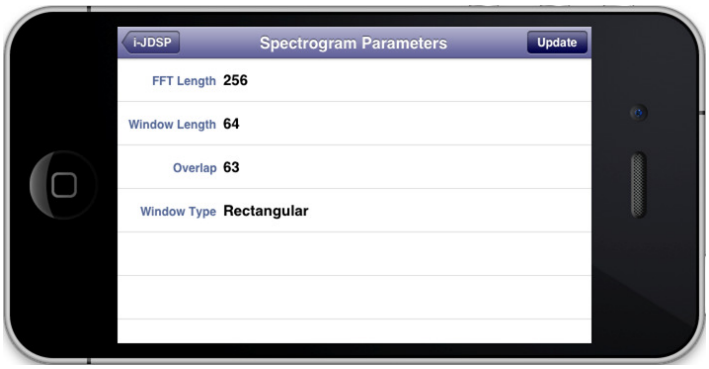
$$X(f,n_0) = \sum_{n=0}^{W-1} x\left(n - \left(n_0 - \left\lfloor \frac{W}{2} \right\rfloor\right)\right) w(n)\, e^{-j2\pi fn} \quad , -\pi \le f \le \pi \tag{6.1}$$

Implementing the above transform as an N-point DFT, the equation could be expressed as

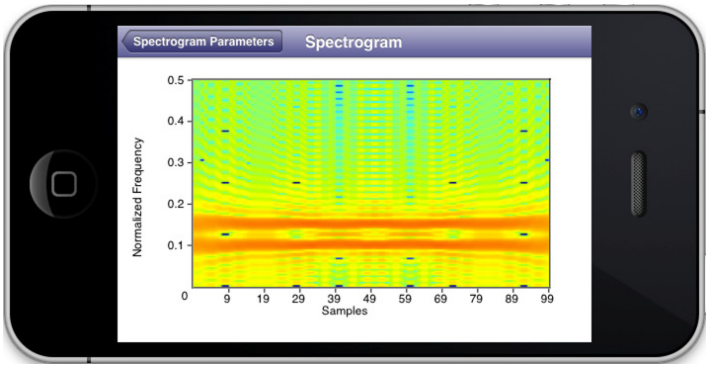
$$X(k,n_0) = \sum_{n=0}^{W-1} x\left(n - \left(n_0 - \left\lfloor \frac{W}{2} \right\rfloor\right)\right) w(n)\, e^{-\frac{j2\pi kn}{N}} , k = 0,1,\ldots N\text{-}1. \tag{6.2}$$

The spectrogram is evaluated from the signal as $S_x(k,n_0)=|X(k,n_0)|^2$. This is synonymous to computing the magnitude frequency response of the Short-time Fourier Transform of the signal.

The spectrogram's frequency resolution is higher when the window is longer and has a narrower main lobe. But longer windows reduce temporal resolution. For higher temporal resolution, smaller window lengths and higher overlap between successive windows in time are required. These trade-offs need to be carefully taken into consideration while choosing the right settings to compute the spectrogram.



(a)



(b)

Figure 6.8: (a) *Spectrogram* block detail view. (b) Spectrogram of a sum of two sinusoids, each of length of 100 samples and normalized frequencies 0.1 and 0.15 radians.

The *Spectrogram* block in iJDSP displays the spectrogram of the input signal. The block can be configured by the user to: (a) choose the desired size of the FFT (64, 128 or 256), (b) set the length of each frame of the signal for the FFT window (can be a maximum of 256), (c) set the number of overlapping samples between adjacent frames and (d) set the type of window to be applied. The spectrogram block detail view is shown in Figure 6.8(a). An example of the spectrogram of a sum of two sinusoids, each of length of 100 samples and normalized frequencies 0.1 and 0.15 radians is shown in Figure 6.8(b).

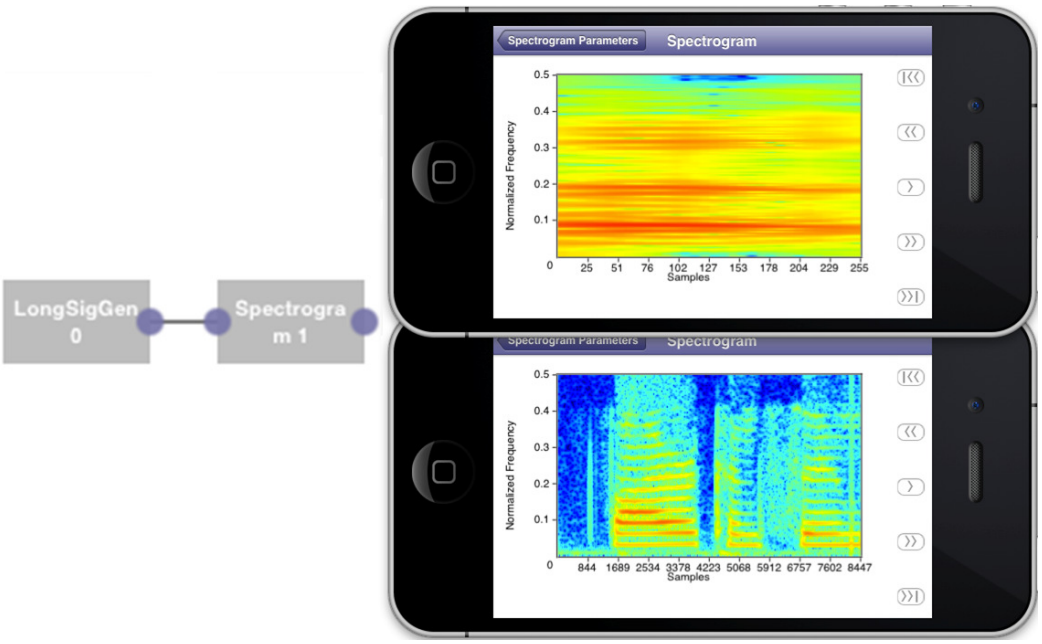


Figure 6.9: Spectrogram of speech clip of a female speaker generated by the *Long Signal Generator* block. The screenshot on the top shows the spectrogram of a single frame. The view on the bottom shows the spectrogram of the entire speech.

Spectrogram for Long Signals

The spectrogram of speech and audio signals is very useful to visualize how the spectral content of signals change over time. In particular, every phonetic sound produced in speech has characteristic spectral peaks called formant frequencies. The formant frequencies characterize the sound, and many speech recognition algorithms exploit this property of speech in some form. Over time, in a spectrogram, these formant frequencies are visible as clearly defined horizontal lines. An example is shown in Figure 6.9, where a *Long Signal Generator* block is connected to the *Spectrogram* block. In the Long Signal Generator, the 'FemaleSpeaker' signal is chosen. In the Spectrogram block, the spectrogram is defined to have an FFT length of 256, a Triangular window of length 200

and a window overlap of 192. The resulting spectrogram can be visualized as shown in Figure 6.9.

The playback buttons allow the user to traverse through the input signal and view the spectrogram for every frame of the input signal. Particularly useful is the '>>|' button, which when pressed, aggregates all the spectrograms over all the frames of the input signal and displays the spectrogram of the whole signal as shown in Figure 6.9. This feature of the Spectrogram block can be used with any signal source. It is to be noted that the Spectrogram block shows the spectrogram of at most 64 frames of an input signal at a time. This is done to avoid memory overruns in the app which may cause the app to crash.

For signals that have more than 64 frames (such as speech/audio recorded from the *sound recorder* block), the '>>|' must be repeatedly pressed to shows successive sets of 64 frames of the input signal.

6.2.3. Linear Predictive Coding (LPC)

Linear predictive coding (LPC) is a well-known approach used to represent speech signals, where speech is modeled as a time-varying system excited by a signal [66,67]. The speech is divided into small frames, and within each frame, the speech is assumed to be quasi-stationary. Hence, the time-varying system for a frame can be approximated as an LTI system, with the LPC coefficients representing the LTI system. The excitation signal is a train of impulses for voiced signals, and white noise for unvoiced signals. The system being excited is modeled as an all-pole filter $S(z)= 1/A(z)$. The filter coefficients of the denominator $A(z)=1+\sum_{i=1}^{M} a_i z^{-i}$ as well as the exciting signal are estimated by the LPC technique. The coefficients of the filter $\{a_i\}_{i=1}^{M}$ are called the

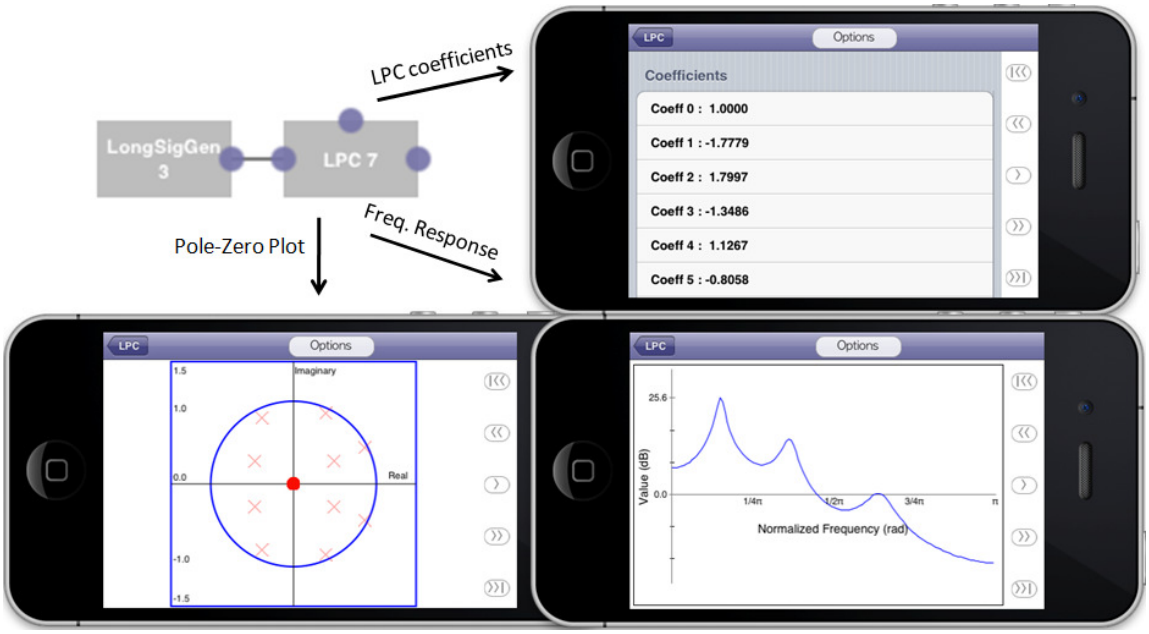


Figure 6.10: The LPC block computes the coefficients of the LPC filter and the residual. It gives as output the LPC coefficients at the top pin and the residual at the right pin.

LPC coefficients. The exciting signal is also referred to as the LPC residual. In iJDSP, the *LPC* function performs linear predictive coding for a given input signal and provides as output the LPC coefficients and the residual.

Shown in Figure 6.10 are some screenshots of the user interface of the LPC block. The LPC block acquires an input frame of signal, and computes the LPC filter coefficients and the residual using the Levinson-Durbin algorithm. The interface displays the LPC coefficients, the magnitude response of the LPC filter, the pole-zero plot of the same filter, and the LPC residual. The user can view any one of these views at any time by choosing them through the Options button, which opens up a pop-down menu listing these views. The playback buttons in the right side of the interface allow the user to control the signal source (the *Long Signal Generator* in this example) and traverse through all the input signal frames.

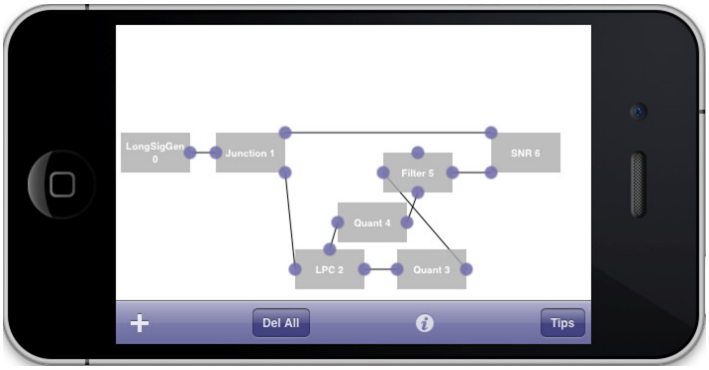


Figure 6.11: LPC Quantization and analysis-synthesis setup.

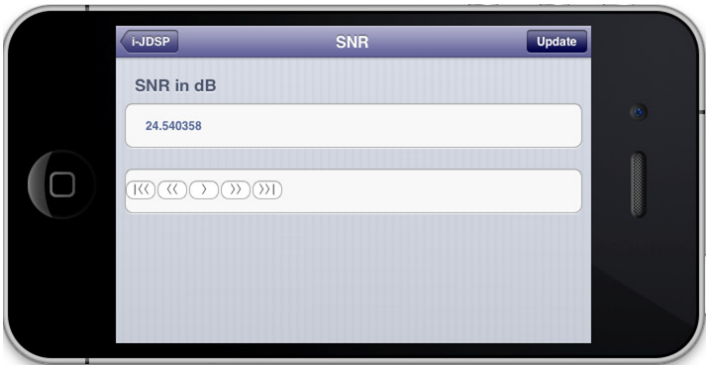


Figure 6.12: User interface of the SNR block. The SNR is displayed in decibels. The playback buttons allow traversal through the input frames to view the resulting SNR for each input frame.

The speech of any signal can be resynthesized from its LPC coefficients and the exciting residual signal. Hence, the resynthesized signal can be observed by filtering the residual signal with the LPC filter in a *Filter* block. It can be compared with the original signal by observing the SNR between the original signal and the signal resynthesized from the LPC coefficients. This allows us to formulate interesting exercises. One such exercise is described below.

Planned Exercise for Observing Quantization Effects on LPC

In iJDSP, a pre-planned simulation setup is provided for illustrating the variation of the Signal-to-Noise ratio of the resynthesized speech signal with respect to the original

speech signal. The simulation setup can be chosen from the "LPC Quantization Setup" provided in the menu of the available blocks accessed through the "+" button in the toolbar at the bottom of the main simulation workspace. The block diagram is automatically generated, as shown in Figure 6.11.

The *Quantizer* block quantizes any input according to the bit depth that is specified by the user. By setting a particular bit depth in a quantizer, the output SNR for the signal can be observed in the *SNR* block. The interface of the SNR block is shown in Figure 6.12. Playback buttons have been added to the interface to allow the user to view the SNR frame-by-frame for any input speech. The user can view the SNR for the current frame being processed, or also the SNR for the entire speech signal by pressing the ">>l" button.

6.2.4. Line Spectrum Pairs

Line Spectrum Pairs (LSPs) are derived as a pair of linear phase filters from the LPC filter coefficients [66]. If the LPC synthesis all-pole filter of order M is $F(z)= 1/A(z)$, then the line spectral pairs are the polynomials $P(z)$ and $Q(z)$.

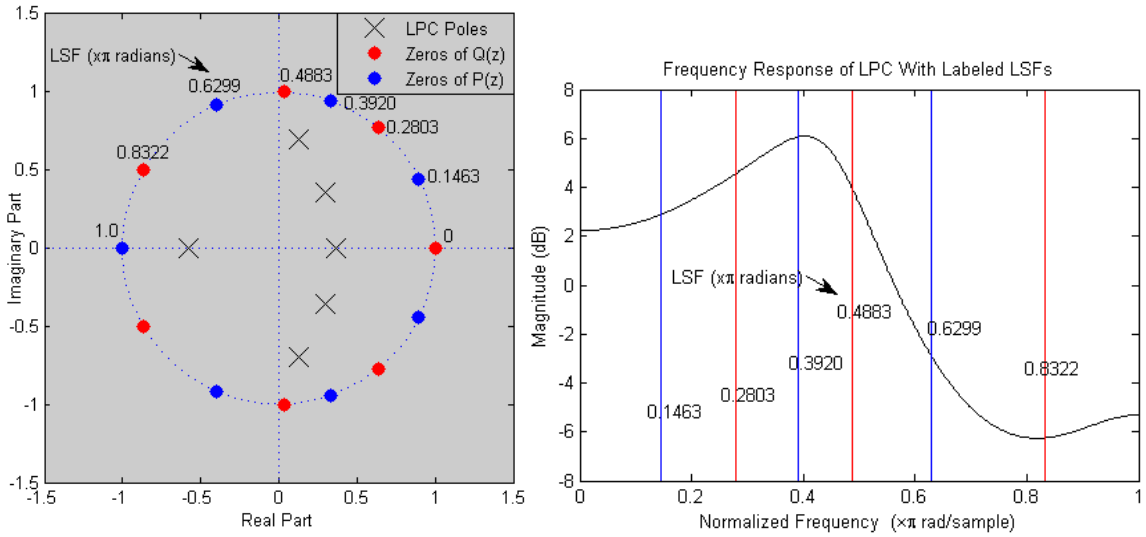
$$P(z) = A(z) + z^{-(M+1)}A(z^{-1}) \tag{6.3}$$

$$Q(z) = A(z) - z^{-(M+1)}A(z^{-1}) \tag{6.4}$$

It is easily deducible that

$$P(z) = z^{-(M+1)}\, P(z^{-1}),\;\; Q(z) = -z^{-(M+1)}\, Q(z^{-1}). \tag{6.5}$$

If M is even, then $P(z)$ is a linear phase filter of odd order with even symmetry. Hence, it has a zero at $z = -1$. Similarly, $Q(z)$ is a linear phase filter of odd order with odd symmetry. Hence, it has a zero at $z = 1$. The original LPC coefficients can be obtained from the line spectral pairs by adding them.

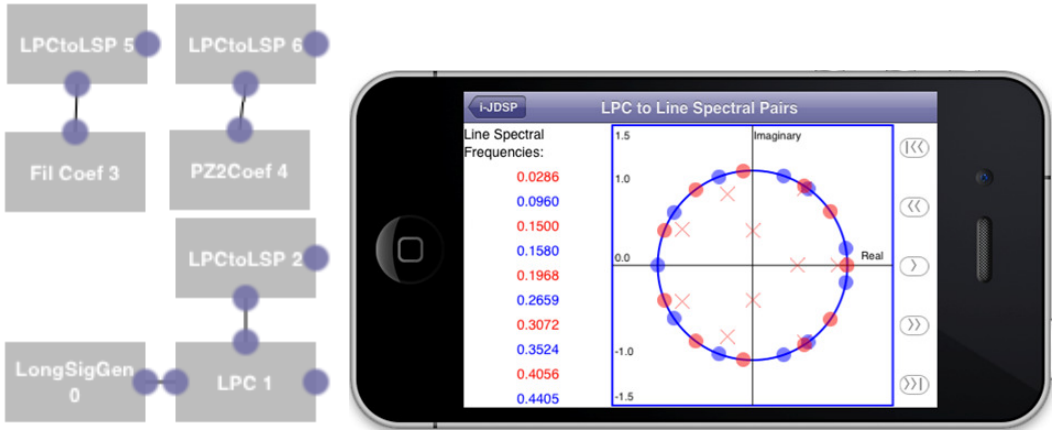


(a) (b)

Figure 6.13: (a) The pole – zero plot showing the poles of a stable filter, which represent the LPC synthesis filter (b) The frequency response of the LPC filter, with the LSF frequencies labeled on the plot.

Line spectrum pairs have the property that when the LPC filter is stable (that is, when the roots of $A(z)$ lie within the unit circle), their zeros lie on the unit circle. A detailed proof for this condition can be found in [68]. This allows the filters to be represented simply by the frequencies of their zeros. These frequencies are referred to as Line Spectral Frequencies (LSFs). Another interesting property of the LSFs is that when the all the roots of $A(z)$ are within the unit circle, the zeros of $P(z)$ and $Q(z)$ are interlaced on the unit circle, as shown in Figure 6.13 [68].

For transmission of speech signals, one can either quantize the LPC coefficients and transmit them along with the residual, or transmit quantized LSFs along with the residual. Upon quantizing the LPC coefficients, the LPC filter $1/A(z)$ is susceptible to becoming unstable [66]. On the other hand, quantizing the Line Spectral Frequencies maintains the line spectral pair zeros on the unit circle. Hence, as long as the quantization



(a) (b)

Figure 6.14: (a) The LPC-LSP block can accept a set of filter coefficients from a block and gives as output the LSF frequencies through its top pin. (b) The figure shows the visualization of the LPC-LSP block.

maintains the interlacing of the zeros of the LSPs, the stability of the filter is maintained. Additionally, the quantization error for a particular frequency affects the LPC filter's response only locally, i.e., only around that frequency. Hence, LSFs can be encoded at lower bit rates than LPC coefficients without introducing instability in the reconstructed LPC coefficients. Taking advantages of these properties, any modern audio encoders encode speech signals using Line Spectral Frequencies. In iJDSP, the line spectral pairs and its properties are illustrated by a set of blocks, which are described below.

The LPC-LSP Function

The LPC-LSP block computes the Line Spectrum Pairs and the Line Spectral Frequencies. In its visualization interface (Figure 6.14), it displays the LPC filter poles and the LSP zeros in a pole-zero plot, and also the values of the line spectral frequencies. On the right side of the screen, the playback buttons perform the function of traversing through the frames of the signal source in the simulation setup. The functions of the

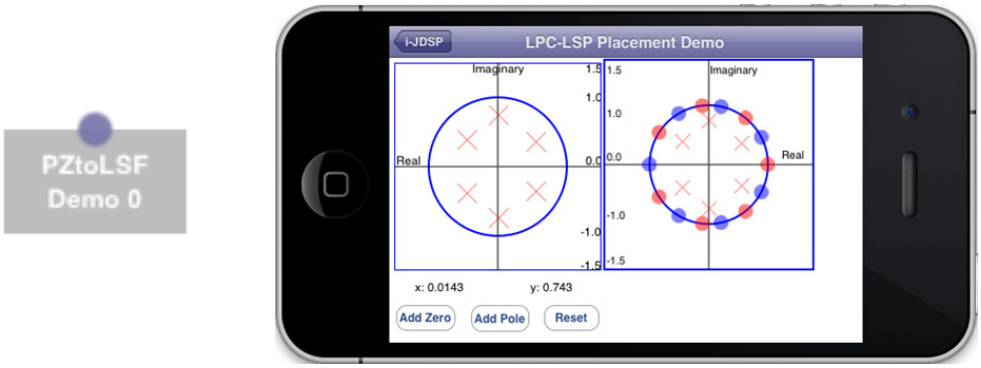


Figure 6.15: This figure shows a screenshot of the interface of the LPS-LSP Placement Demo block. The user can place poles on the z-plane on the left side to create an LPC synthesis filter. The corresponding pole-zero plot of the LSP filters is shown to its right.

buttons are similar to those in other blocks such as the Long Signal Generator (Figure 6.2), which are explained in Section 6.1.

LPC-LSP Placement Demo Block

The LPC-LSP Placement Demo block shows a demonstration of the variation of zeros of the LSP filters. Shown in Figure 6.15 is the interface of this demonstration block. The user can place poles on the z-plane on the left side of the screen to construct an LPC synthesis filter. The user can move the pairs of conjugate pairs of poles on the z-plane through the touch-screen interface. As the LPC poles are placed and moved on the z-plane, the corresponding LSP zeros displayed on the z-plane on the right side of the interface change dynamically in real-time.

This interface can be used to create test cases of configuration of LPC poles to better understand how the positions of the poles reflect in the LSFs. In particular, it can be shown through this demo that LPC poles outside the unit circle cause the LSP filters to lose the property of having interlaced zeros on the unit circle, and often cause the LSP

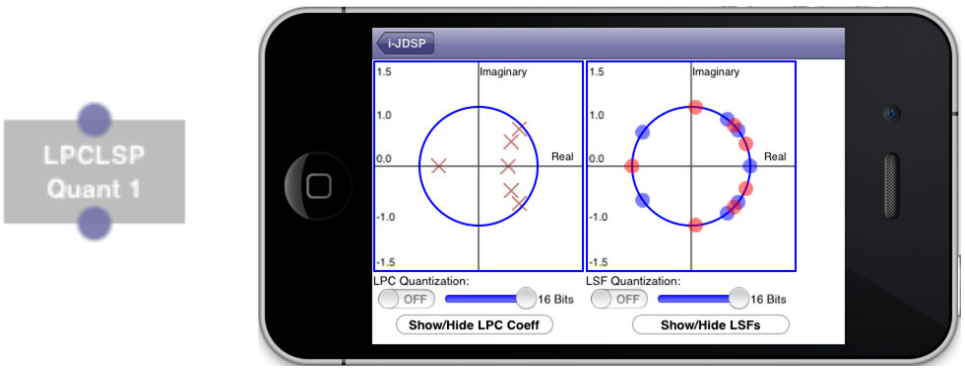


Figure 6.16: The LSP-LSP Quantization Demo block accepts as input LPC filter coefficients and computes the LSFs and reconstructs the LPC filter from the LSF. It compares the effect of quantizing LPC coefficients versus quantizing LSFs.

zeros corresponding to an unstable pole to move away from the unit circle. These two properties of LSP poles are also tests for instability of the corresponding LPC filter.

The LPC-LSP Quantization Demo Block

The LPC-LSP Quantization Demo block demonstrates the effect of quantizing the LPC poles or the LSF frequencies on the pole-zero locations of the original LPC filter and the LPC filter reconstructed from the quantized LSFs. The user interface of the block is shown in Figure 6.16. The block has an input pin at the bottom, which accepts a set of LPC filter coefficients and computes the Line Spectral Frequencies from them. On the z-plane on the left, it shows pole-zero plots – the poles colored black are the poles of the LPC filter when it is directly quantized. The poles colored red are the LPC poles when the LPC coefficients are converted to LSF frequencies, and the LPC filter is reconstructed after quantizing the LSFs. The z-plane on the right side shows the Line Spectral Frequencies after they are quantized.

Figure 6.17: The LPC-LSP Placement Demo block is used to create a test case of an LPC filter, which can be studied for quantization effects in the LPC-LSP Quantization Demo block.

The number of bits of quantization for the LPC and the LSF quantization can be chosen from the corresponding sliders, which allow the users to choose from 2 to 16 bits of quantization bit depth. The LPC and LSF quantization can also be turned ON or OFF from the corresponding switches provided. This allows the user to visually compare the pole-zero plots original LPC filter and the LPC filter reconstructed form the LSF frequencies.

A useful example that can be constructed is the creation of LPC filter pole-zero configurations which can potentially lead to unstable LPC filters if they are directly quantized. It can be shown that the reconstruction of the filters from LSFs is more resilient to instability. An example of such a setup is shown in Figure 6.17. The LPC-LSP Placement demo block is used to create an LPC filter. The constructed LPC filter coefficients are passed to the LPC-LSP Quantization Demo block, in which the user can observe the effect of quantization of the LPC coefficients and compare it with the filter reconstructed from the LSFs. In this particular example, the LPC filter, upon directs quantization to 4 bits, becomes unstable. But the LPC filter reconstructed from the LSFs remains stable.

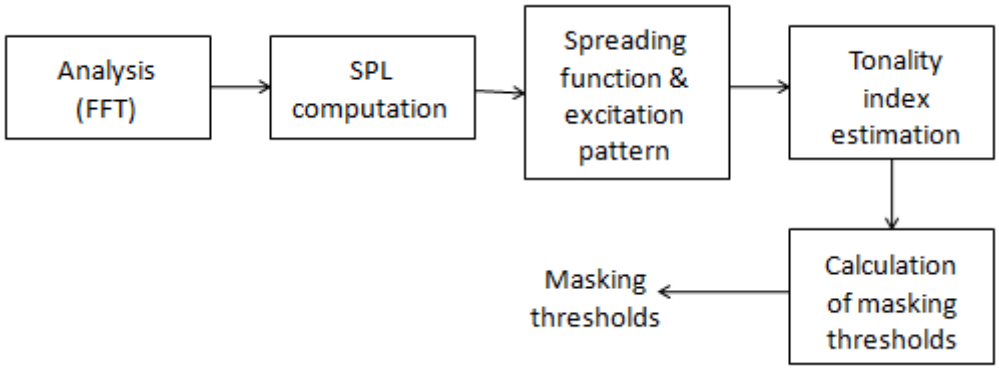


Figure 6.18: Block diagram for computing the masking thresholds in the MPEG I Layer 3 psychoacoustic model.

6.2.5. The Psychoacoustic Model

The Psychoacoustic Model II is the human auditory model used by the MPEG I Layer III audio encoder for exploiting the human ear's perceptual properties in improving the coding efficiency by neglecting those frequency components in the sound signal which cannot be perceived by the ear due to its spectral and temporal masking properties [66,52]. The basic outline of implementation of Psychoacoustic Model II for estimating the masking thresholds is shown in Figure 6.18. Note that the window switching techniques are not dealt with here for simplicity.

The psychoacoustic model used consists of a set of bands of frequencies, each with a response and bandwidth modeling the human auditory system's characteristics. The input signal is analyzed through this model and parameters derived from this analysis are used to set masking thresholds according to which frequency components in the signal extracted from the analysis filter bank would be discarded or retained for encoding.

The input frame is 1152 samples long. In the analysis section, a 1024 point FFT of the incoming signal frame is taken after it is windowed first using a 1024 point

Hanning window. Since the signal frame is too long for this analysis window, the frame is split into two frames, each with 576 samples. The frames are placed in a buffer such they align with the analysis window of the filter bank. This buffer is then processed using the Hanning window and then a 1024 point FFT is taken.

From the FFT, the energy *eb* of the signal is computed in each threshold partition of the psychoacoustic model.

$$eb(\mathrm{z}) = \sum_{f=\mathrm{bl}_z}^{\mathrm{bh}_z} \mathrm{R}^2(f) \quad \text{, for the z}^{\text{th}} \text{ critical band} \tag{6.6}$$

A measure of the signal's unpredictability for each threshold partition is derived by computing the predicted energy of the current frame from the two preceding frames and the actual energy of the current frame. Usually, tonal bands have more predictability than noise-like bands. The unpredictability is given by the following equation.

$$cw(f) = \frac{\left(\left(\mathrm{R_j}(f)\cos\phi_\mathrm{j}(f) - \hat{\mathrm{R}}_\mathrm{j}(f)\cos\hat{\phi}_\mathrm{j}(f)\right)^2 + \left(\mathrm{R_j}(f)\sin\phi_\mathrm{j}(f) - \hat{\mathrm{R}}_\mathrm{j}(f)\sin\hat{\phi}_\mathrm{j}(f)\right)^2\right)^{\frac{1}{2}}}{\mathrm{R_j}(f) - \left|\hat{\mathrm{R}}_\mathrm{j}(f)\right|} \tag{6.7}$$

Here, $\hat{\mathrm{R}}_\mathrm{j}(f) = 2\mathrm{R}_{\mathrm{j-1}}(f) - \mathrm{R}_{\mathrm{j-2}}(f)$ and $\hat{\phi}_\mathrm{j}(f) = 2\phi_{\mathrm{j-1}}(f) - \phi_{\mathrm{j-2}}(f)$. The weighted unpredictability in energy is

$$cb(z) = \sum_{f=\mathrm{bl}_z}^{\mathrm{bh}_z} \mathrm{R}^2(f)cw(f)\,. \tag{6.8}$$

The ability of a strong signal to mask weaker signals in the same threshold partition and also mask weak signals in neighboring partitions is an important property of the human ear, which is modeled by the psychoacoustic model by means of a spreading function to spread the energy of a threshold partition into its neighboring partitions. The

basilar excitation pattern is then computed by convolving the signal energy of the critical bands with the spreading function, denoted by *spf* (*i,j*) for the effect of the masker in the j^{th} band upon the i^{th} critical band, through the following expression.

$$ecb(z) = \sum_{b=1}^{z_{max}} eb(z_b) spf\,(z_b, zm) \tag{6.9}$$

Here, zm is the mean Bark value of the z^{th} band. The excitation pattern is then normalized to compensate for the increase in the energy due to the spreading function, to give the normalized pattern *enb*(z).

$$enb(z) = \frac{ecb(z)}{\sum_{b=0}^{z_{max}} spf\,(zm_b, zm)} \tag{6.10}$$

Also, the unpredictability in the threshold partitions is convolved with the spreading function to incorporate the spreading function into the unpredictability.

$$ctb(z) = \sum_{b=1}^{z_{max}} cb(z_b) spf(z_b, zm) \tag{6.11}$$

Here, zm is the mean Bark value of the z^{th} band. The masking characteristics of tonal and non-tonal bands are different and hence, it becomes essential to identify if a critical band has tonal components or otherwise. A parameter called 'tonality' is estimated for this purpose. The tonality of the signal in each threshold partition *tbb*(z) is computed from the spread unpredictability and the spread signal energy.

$$cbb(z) = \log\left(\frac{ctb(z)}{ecb(z)}\right) \quad , \quad tbb(z) = -0.299 - 0.43\,cbb(z) \quad , 0 < tbb(z) < 1 \tag{6.12}$$

The masking thresholds are then determined by providing an offset to the excitation pattern, where the value of the offset strongly depends on the nature of the

masker. The offset is obtained by weighting the maskers with the estimated tonality index.

$$O(\mathrm{z}) = 29tbb(\mathrm{z}) + 6\left(1 - tbb(\mathrm{z})\right) \tag{6.13}$$

The SNR for each threshold partition is computed as,

$$SNR(\mathrm{z}) = \max[\min val(\mathrm{z}), O(\mathrm{z})] \tag{6.14}$$

Transforming the *SNR* to the linear scale,

$$bc(\mathrm{z}) = 10^{-SNR(\mathrm{z})/10} \tag{6.15}$$

Then, the energy threshold for the basilar excitation pattern in each threshold partition is

$$nb(\mathrm{z}) = enb(\mathrm{z})bc(\mathrm{z}) \tag{6.16}$$

Then, the masking threshold for the given frame of signal is given by the following expression.

$$thr(\mathrm{z}) = \max[Tq(\mathrm{z}), \min[nb(\mathrm{z}), nb_{\mathrm{t-1}}(\mathrm{z}), nb_{\mathrm{t-2}}(\mathrm{z})]] \tag{6.17}$$

$$nb_{\mathrm{j-1}}(\mathrm{z}) = 2.nb(\mathrm{z}) \tag{6.18}$$

$$nb_{\mathrm{j-2}}(\mathrm{z}) = 16.nb(\mathrm{z}) \tag{6.19}$$

$Tq(\mathrm{z})$ is the absolute threshold of hearing, and $nb_{\mathrm{j-1}}(\mathrm{z})$ and $nb_{\mathrm{j-2}}(\mathrm{z})$ are the energy thresholds from the past two frames. The masking thresholds decide which frequency components can be discarded on perceptual grounds.

In iJDSP, the Psychoacoustic Model block is supported to illustrate the psychoacoustic model used in MP3 encoders. The block accepts an input signal frame and computes its masking thresholds, and displays the plot of the masking thresholds versus frequency, along with the signal spectrum (Figure 6.19). The block provides to the user the option of passing signals recorded at either 16 kHz or 44.1 kHz. This option can

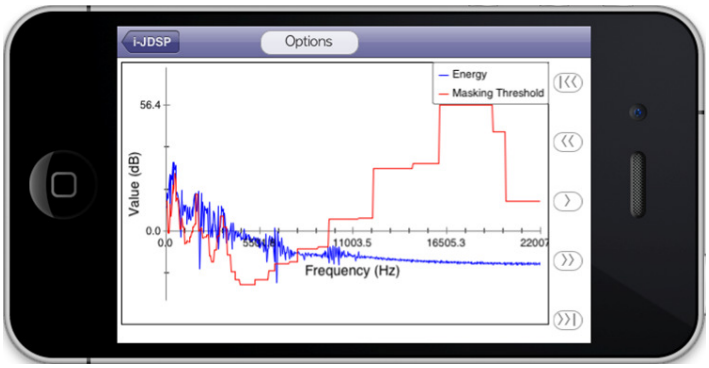


Figure 6.19: Psychoacoustic Model block interface showing the signal spectrum as the blue curve and the masking threshold for the frame as the red curve. The signal spectral components falling below the masking threshold are perceptually irrelevant, that is, they are masked and hence, inaudible to the listener.

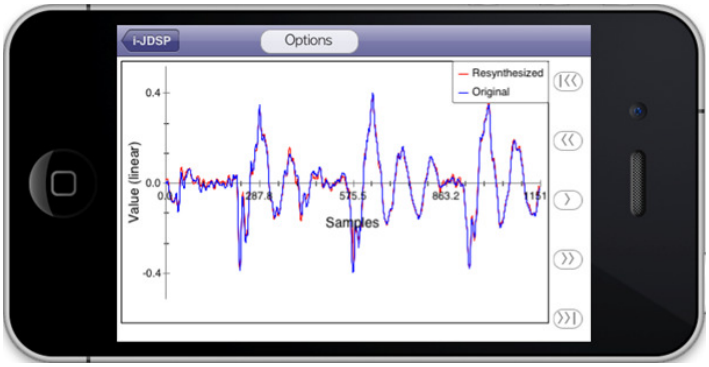


Figure 6.20: Psychoacoustic Model block interface showing the original frame of signal as the blue curve and the signal resynthesized after truncating the masked frequency components as the red curve.

be chosen while selecting the block in the main canvas for viewing the plots of the masking thresholds.

The block also performs a peak-picking on the signal frame by removing the spectral components that fall below the masking threshold in their respective threshold partitions, and resynthesizes the peak-picked signal to show its similarity to the original frame of signal. The plot of the original and resynthesized signal is shown in the block's user interface (Figure 6.20). The Options button at the top of the interface can be used to

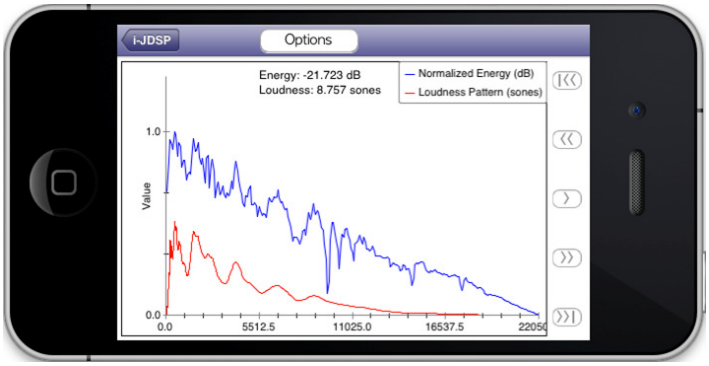


Figure 6.21: View showing the signal energy normalized in the dB scale as the blue graph and the specific loudness (or loudness pattern) of the signal as the red curve. The playback buttons on the right allow the user to traverse through all the frames of the input signal to view their respective loudness patterns.

switch between the different plots. The playback buttons on the right of the plot can be used to traverse through the frames of the input signal.

6.2.6. Loudness

The phenomenon of loudness and the methods for its quantification and estimation were described in Chapter 3. iJDSP provides an illustration of loudness estimation as part of the Psychoacoustic Model block. Given a frame of input signal, the block, along with the masking thresholds and the signal resynthesized after truncating the masked frequency components, also computes the loudness using the Moore and Glasberg model and displays the loudness pattern and the total loudness value of the signal frame. The loudness pattern plot can be viewed in the Psychoacoustic Model user interface by selecting the 'Loudness Pattern' from the pop down menu that appears on pressing the Options button.

6.2.7. System Identification Demonstration: LMS Demo

The estimation of characteristics of an unknown system is an important process in many DSP systems in a variety of applications. In one approach, a known signal $x(n)$ is

passed through the unknown system with impulse response $b(n)$ and its response $y(n)$ as shown in equation (6.20) is obtained. The estimate of the unknown system $\hat{b}(n)$ (which is assumed to be modeled as an L-order FIR filter) is obtained through the sequential Least Mean Square (LMS) algorithm [66]. The sequential LMS algorithm involves comparing the response of the original system $y(n)$ to the known input $x(n)$ with that of the estimated system $\hat{y}(n)$, and adapting the filter coefficients $\hat{b}(n)$ such that the error function in equation (6.22) is minimized. Minimizing this error is mathematically synonymous to minimizing the mean square error of the system response.

$$y(n) = \sum_{k=0}^{\infty} b(k)x(n-k) \tag{6.20}$$

$$\hat{y}(n) = \sum_{k=0}^{L} \hat{b}(k)x(n-k) \tag{6.21}$$

$$e(n) = y(n) - \hat{y}(n) \tag{6.22}$$

At each instant 'n', the filter taps are updated according to the steepest descent method in order to minimize the cost function given by the equation (6.23), where $E\{\}$ denotes the expectation operation when $e(n)$ is a random variable. The gradient descent method is in fact only stochastic, that is, the filter taps are adapted only based on the error at that instant, and not by averaging over all the past errors. The steepest descent algorithm minimizes the cost function by traveling against the gradient of the cost function evaluated at any instant. This requires the computation of partial derivatives of $C\big(e(n)\big)$ with respect to each filter tap of $\hat{b}(k)$, which give the filter tap update equations as given in (6.24). Here, $\hat{b}_n(k)$ denotes the filter taps at instant n and the taps are updated to $\hat{b}_{n+1}(k)$ before filtering at the $(n+1)^{\text{th}}$ iteration.

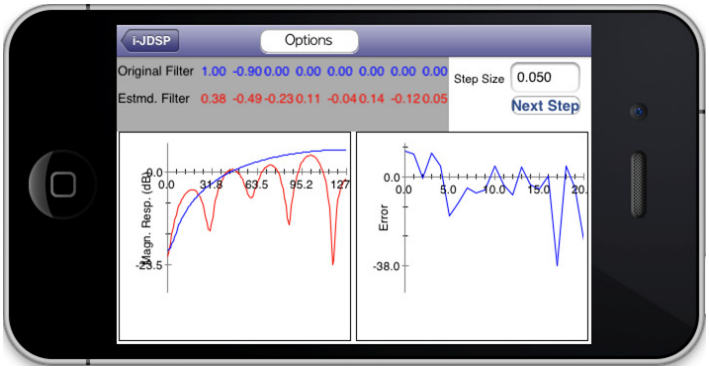


Figure 6.22: The visualization interface of the LMS Demo block.

$$C\big(e(n)\big) = E\{|e(n)|^2\} \tag{6.23}$$

$$\hat{b}_{n+1}(k) = \hat{b}_n(k) + \mu e(n)x(n-k) \quad , k=0,1,\ldots L \tag{6.24}$$

μ is the step size of the descent. The step size controls how fast the filter taps converge towards the filter taps of the unknown system. If the step size is high, the speed of convergence is faster but the filter taps fluctuations around the minimum error are higher, hence, giving a higher mean square error in steady state. On the other hand, a low step size results in a slow convergence rate, although the steady state mean square error is lower.

In iJDSP, the *LMS Demo* is a standalone demonstrative block provides an interface to visualize the demonstration of the LMS algorithm in system identification. The interface of the LMS Demo block is shown in Figure 6.22. The user interface allows the configuration of the input signal $x(n)$ and the unknown system $b(k)$ through the pop down menu which appears upon pressing the Options button, as described in Figure 6.23. The filter tap values of the original filter and the estimated filter are color-coded as blue and red respectively and listed in the view with a gray background. The portion of the screen with the gray view can be scrolled into in order to view the remaining tap values.

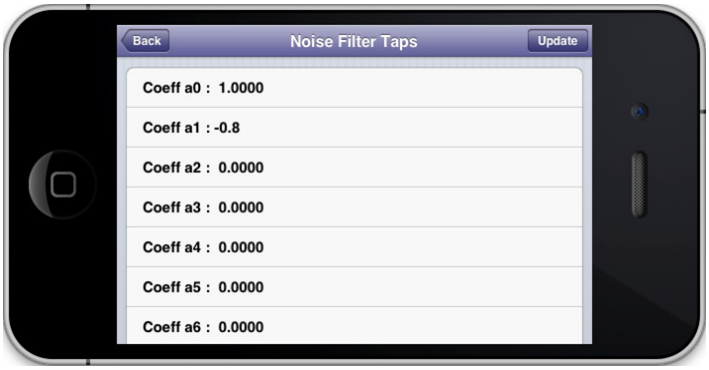


Figure 6.23: The interface for configuring the filter taps of the noise in the 'Custom-Filtered Noise' option in the *LMS Demo* block.

The step size is shown in an editable text field in the top right corner of the view. The magnitude frequency response of the original and the estimated system are shown in the plot on the left. The plot on the right shows the mean square error, until the last computed iteration. The system $b(k)$ is allowed to be configured as an FIR filter for simplicity of illustration of the concept in the user interface. The input signal is of a fixed length of 512 samples, and the estimate of the unknown system $\hat{b}(k)$ is represented as a 10^{th} order filter.

The available choices for the input signal are white noise, brown noise, or white noise filtered by a customizable filter that a user is allowed to configure. The customizable filter can have an order of up to 10, and be an FIR or IIR filter. These options allow the user to test the LMS algorithm with a variety of input signals and infer the effect of the input signal's characteristics on the accuracy of the estimated system when the LMS algorithm is implemented.

The filter representing the original filter can be configured only as an FIR filter, in order to ensure simplicity in representing convergence of error in the plots in the user interface. Students can configure the filter taps to observe the speed of convergence as

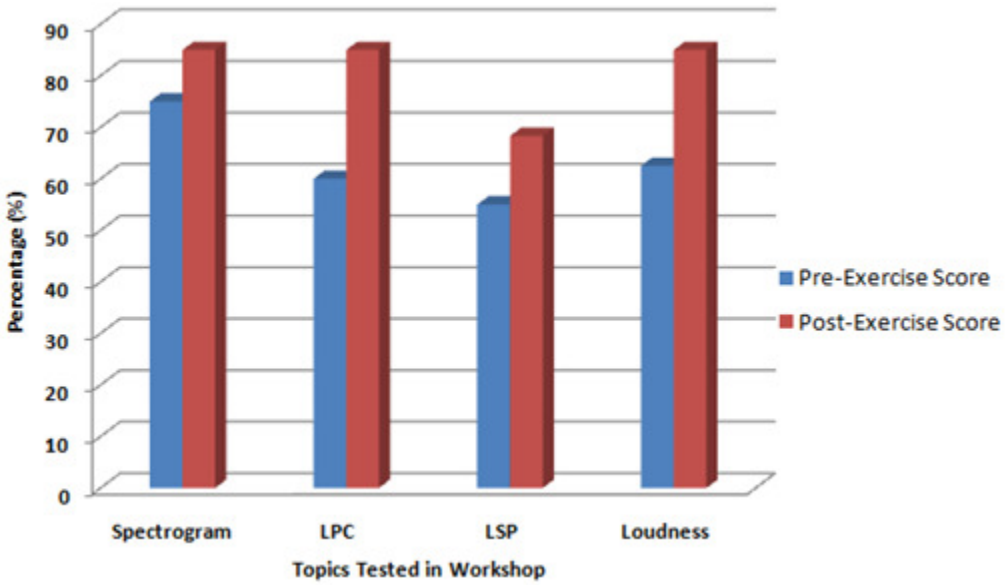


Figure 6.24: Pre- and post-assessment results to assess student performance before and after using iJDSP.

the complexity of the filter characteristics is varied. The step size can also be varied to observe the speed of convergence of the filter taps for different step sizes.

6.3. Assessments

The effectiveness of reinforcing relevant speech and audio processing concepts through illustrative simulations using the above described speech/audio DSP functionalities in iJDSP was evaluated through assessments conducted at ASU in Fall 2013. Graduate students specializing in Signal Processing and Communications were taught specific speech/audio DSP concepts, namely spectrograms and their properties, linear predictive coding for speech encoding, its motivation and application, line spectral pairs and their properties, and the concept of perceptual loudness with complementary exercises using iJDSP on the same concepts. The process adopted for the evaluation exercises comprised the following steps, (a) a pre-lecture questionnaire on the concepts involved in the exercise, (b) lecture on the pertinent signal processing concepts, (c) a

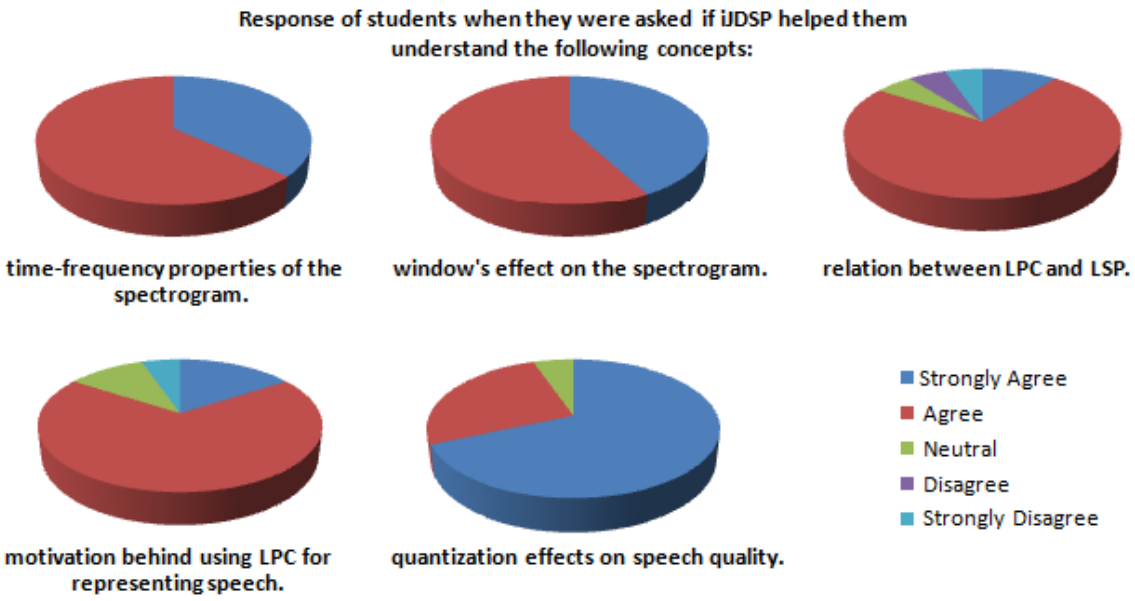


Figure 6.25: Response of students indicative of subjective opinions on effectiveness of iJDSP in understanding delivered speech/audio DSP concepts.

simulation exercise using iJDSP, and (d) a post-questionnaire to test the efficacy of iJDSP in improving student understanding of the concepts. In addition to the technical assessments, general assessments on the learning experience were solicited and student feedback about iJDSP was also collected. The simulation exercises are given in Appendix A. In this section, we present the results of these assessments.

6.3.1. Impact on Student Performance

The pre- and post-quizzes were aimed at understanding the effect of the iJDSP app in learning the speech/audio concepts delivered in the lecture. Fig. () illustrates the improvement in student understanding of the different concepts by using the iJDSP visualization tools. In each case, we show the average number of students that correctly answered the questions pertinent to a topic. For each topic, the improvement in the post-assessment, when compared to the pre-assessment, is shown in Figure 6.24. The percentage of students who understood spectrograms after the assessments were 85%,

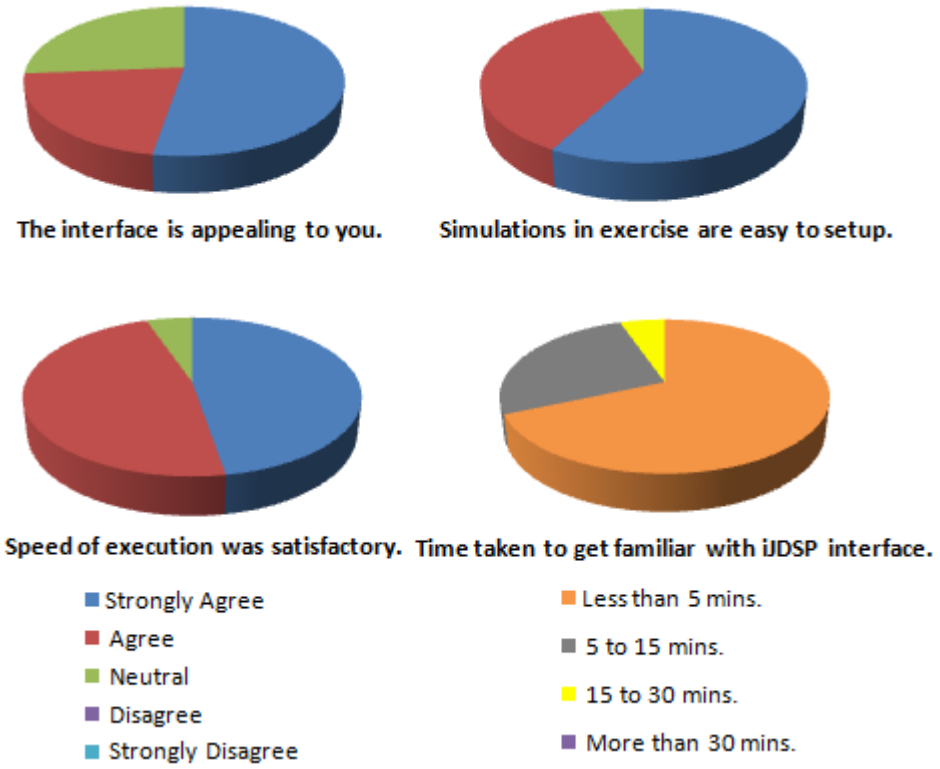


Figure 6.26: Response of students indicative of opinions on the quality of the iJDSP interface, ease of use, and responsiveness.

showing an improvement of about 10% over the pre-assessment quiz. Similarly, an improvement of 25% was seen with learning about linear predictive coding and its properties. Overall, by using iJDSP, more than 80% of the students understood all topics presented in the exercise as opposed to 63% with the lecture alone.

6.3.2. Technical Assessments

In the post assessment questionnaire, students provided subjective opinions on whether the speech processing modules in iJDSP were useful in improving understanding in each of the exercises. The responses were used to evaluate the usefulness of the developed iJDSP functions, in improving student understanding of the topics covered in the exercise. The students were asked to respond with one of the following options:

Strongly Agree, Agree, Neutral, Disagree, and Strongly Disagree. The results of the evaluation are shown in Figure 6.25. Almost unanimously, the students responded that iJDSP helped them understand the speech/audio DSP concepts taught in the assessment session.

6.3.3. Application Quality Feedback

In addition to the technical assessments, the app's quality was also assessed through a set of questions recording the user's opinion on the qualities of the app such as the fluidity of the interface, aesthetics and user-friendliness. As shown in Figure 6.26, it was found that more than 90% students got familiarized with the iJDSP user environment in less than 15 minutes. They also indicated that the interface is appealing and the simulations are easy to setup. The interface was also reported to be responsive, with the students indicating that the speed of the application's execution is satisfactory.

Chapter 7

CONCLUSIONS AND FUTURE WORK

Human auditory models based loudness estimation was explored in this thesis along with its application on real-time automatic loudness control. Methods to efficiently estimate loudness from the Moore and Glasberg model by pruning computations based on the signal properties were examined. Detector pruning and frequency pruning were employed to reduce the computations involved in evaluating the excitation pattern, which is the computational bottleneck in the algorithm. The existing detector pruning algorithm was improved to address its shortcomings in estimation of auditory patterns and loudness in the presence of sharp transitions in the excitation pattern. This significantly boosted the performance for sound tracks in the Sound Quality Assessments Materials (SQAM) database, with insignificant increase in the computation load. The examination of the improvement of the percentage error in the loudness with the computational error for individual sound tracks showed that higher increases in computations relative to the original detector pruning approach correspond to larger reductions in the percentage error. The Moore and Glasberg model was used to implement a perceptual loudness based real-time automatic loudness control system in Simulink. The loudness control system, which was developed as a perceptually controlled adaptive gain control system, involved both dynamic wide-band as well as dynamic narrow-band gain control mechanisms. The wide-band control's function is primarily to control the perceptual loudness in real-time while the narrow-band gain controller preserves the tonal balance of the output.

Given the ability to speedily estimate excitation and auditory patterns, they can be easily extracted from audio data and used as feature vectors for applications such as binaural source separation and direction finding, auditory scene analysis, objective sound quality measurements, audio coding, and perceptual bandwidth extension.

The application of sophisticated audio DSP algorithms in iOS devices was explored for creating interactive simulation environments in the educational mobile App iJDSP for illustrating key speech and audio DSP concepts. The system simulation architecture of iJDSP was enhanced to provide the necessary framework for facilitating speech and audio processing and visualization. Features such as the ability to handle long signals and frame-by-frame processing and visualization are important contributions in this regard. Using these developed functions, sample exercises were formulated to demonstrate specific DSP concepts such as the use of spectrograms in analyzing speech, Linear Predictive Coding (LPC), Line Spectrum Pairs (LSP) and the phenomenon of perceptual loudness.

The effectiveness of the software along with the developed software modules in enhancing student learning was examined by allowing students to perform exercises on speech and audio processing on iJDSP, and assessing their scores in quizzes before and after the use of iJDSP. The results indicate that the software helped increase student understanding of the concepts taught through the exercises. The students also gave positive responses about the user friendliness of the app, its speed of execution and overall appeal. Future work on improving the app can involve the utilization of larger screens on devices such as iPads to provide more comprehensive visualizations for better clarity of illustration.

REFERENCES

[1] H. Fletcher and W.A. Munson, "Loudness, its definition, measurement and calculation," *The Journal of the Acoustical Society of America*, vol. 5, no. 2, pp. 82-108, Oct 1933.

[2] SoundHound. [Online]. http://www.soundhound.com/index.php?action=s.about

[3] "GarageBand". [Online]. http://www.apple.com/apps/garageband/

[4] "Music Studio". [Online]. https://itunes.apple.com/US/app/id328608539?mt=8&ign-mpt=uo%3D4

[5] J. Liu, S. Hu, J.J. Thiagarajan, X. Zhang, S. Ranganath, M.K. Banavar, A. Spanias, "Interactive DSP laboratories on mobile phones and tablets," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

[6] H. Fletcher, "Auditory Patterns," *Reviews of Modern Physics*, vol. 12, no. 1, pp. 47-65, Jan. 1940.

[7] H. Fletcher and W.A. Munson, "Relation between loudness and masking," *The Journal of the Acoustical Society of America*, vol. 9, no. 1, pp. 1-10, July 1937.

[8] R. Hellman, "Asymmetry of masking between noise and tone," *Perception and Psychophysics*, vol. 11, no. 3, pp. 241-246, 1972.

[9] J.L. Hall, "Auditory psychophysics for coding applications," in *The Digital Signal Processing Handbook*, V. Madisetti and D.Williams, Ed. Boca Raton, FL: CRC Press, 1998, pp. 39.1-39.25.

[10] B.C.J. Moore, "Masking in the Human Auditory System," in *Collected Papers on Digital Audio Bit-Rate Reduction*, N. Gilchrist and C.Grewin, Ed., 1996, pp. 9-19.

[11] S.S. Stevens, "Procedure for calculating loudness: Mark vi," *The Journal of the Acoustical Society of America*, vol. 33, no. 11, pp. 1577-1585, Nov 1961.

[12] S.S. Stevens, "On the psychophysical law," *Psychological review*, vol. 64, no. 3, 1957.

[13] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 451-513, Apr. 2000.

[14] B.C.J. Moore, *An Introduction to the Psychology of Hearing*, 5th ed.: Academic Press, 2003.

[15] "American National Standard Specification for Sound Level Meters," ANSI S1.4-1983 (R2006).

[16] E. Zwicker and B. Scharf, "A model of loudness summation," *Psychological Review*, vol. 72, pp. 3-26, 1965.

[17] B.C.J. Moore and B.R. Glasberg, "A revision of Zwicker's loudness model," *Acustica - Acta Acustica*, vol. 82, pp. 335-345, 1996.

[18] T. Dau, D. Püschel, A. Kohlrausch, "A quantitative model of the effective signal processing in the auditory system. i. model structure," *The Journal of the Acoustical Society of America*, vol. 109, no. 5, pp. 2008-2022, 2001.

[19] B. C. J. Moore, B. R. Glasberg, and T. Baer, "A model for the prediction of thresholds, loudness, and partial loudness," *of Audio Engineering Society*, vol. 45, no. 4, pp. 224-240, Apr. 1997.

[20] R.D. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, M. Allerhand, "Complex sounds and auditory images," in *Auditory Physiology and Perception, Proceedings of the 9th International Symposium on Hearing*, L. Demany, and K. Horner Y. Cazals, Ed.: Oxford, 1992, pp. 429-446.

[21] T. Irino and R.D. Patterson, "A compressive gammachirp auditory filter for both physiological and psychophysical data," *The Journal of the Acoustical Society of America*, 2001.

[22] E. A. Lopez-Poveda and R. Meddis, "A human nonlinear cochlear filterbank," *The Journal of the Acoustical Society of America*, vol. 110, no. 6, pp. 3107-3118, 2001.

[23] B.R. Glasberg and B.C.J. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hearing Research*, 1990.

[24] H. Krishnamoorthi, V. Berisha and A. Spanias, "A low-complexity loudness estimation algorithm," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2008*, 2008, pp. 361-364.

[25] H. Krishnamoorthi, A. Spanias, and V. Berisha, "A frequency/detector pruning approach for loudness estimation," *IEEE Signal Processing Letters*, vol. 16, no. 11, pp. 997-1000, Nov. 2009.

[26] A. Spanias and V. Atti, "Interactive Online Undergraduate Laboratories Using J-DSP," *IEEE Transactions on Education*, vol. 48, no. 4, Nov. 2005.

[27] A. Clausen, A. Spanias, A. Xavier, M. Tampi, "A Java signal analysis tool for signal processing experiments," in *Proceedings of the IEEE International Conference on Acoustics Speech Signal Processing (ICASSP)*, May 1998, pp. 1849-1852.

[28] A. Clausen and A. Spanias, "An Internet-based computer laboratory for DSP courses," in *28th Annual Frontiers in Education*, 1998, pp. 206-210.

[29] S. Henderson and J. Yeow, "iPad in Education: A case study of iPad adoption and use in a primary school," in *the Proceedings of 45th Hawaii International Conference on System Sciences*, 2012.

[30] G. Engel, "Using Mobile Technology to Empower Student Learning," in *the Proceedings of 27th Annual Conference on Distance Teaching & Learning*, 2011.

[31] N. Ostashewski and D. Reid, "iPod, iPhone, and now iPad: The evolution of multimedia access in a mobile teaching context," in *the Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 2010.

[32] N. Ostashewski, D. Reid, M. Ostashewski, "Mobile Teaching and Learning Technologies: Ukrainian Dance Instruction in Canada," in *IADIS Mobile Learning 2009*, 2009.

[33] Y. Zhou, G. Percival, X. Wang, Y. Wang, S. Zhao, "MOGCLASS: evaluation of a collaborative system of mobile devices for classroom music education of young children," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 523-532.

[34] StarWalk. [Online]. http://vitotechnology.com/star-walk.html

[35] The HP 12C Financial Calculator. [Online]. www8.hp.com/us/en/products/smart-phones-handhelds-calculators/mobile-apps/app_details.html?app=tcm:245-799200&platform=tcm:245-799129

[36] Spectrogram. [Online]. http://spectrogramapp.com/

[37] L. Johnson, S. Adams, M. Cummins, "NMC horizon report: 2012 higher education edition," 2012.

[38] Touch Surgery. [Online]. http://www.touch-surgery.com/html/about.html

[39] MATLAB Mobile. [Online]. http://www.mathworks.com/mobile/

[40] MATLAB Mobile FAQs. [Online]. http://www.mathworks.com/mobile/faq.html

[41] Simulink. [Online]. http://www.mathworks.com/products/simulink/?s_tid=hp_fp_sl

[42] LabView. [Online]. http://www.ni.com/labview/whatis/

[43] MATLAB. [Online]. http://www.mathworks.com/products/matlab/?s_tid=hp_fp_ml

[44] Mathematica. [Online]. http://www.wolfram.com/mathematica/

[45] GNU Octave. [Online]. http://www.gnu.org/software/octave/about.html

[46] Connexions. [Online]. http://cnx.org/content/m32834/latest/

[47] H. Fastl and E. Zwicker, *Psychoacoustics: Facts and Models*, 3rd ed.: Springer, 2006.

[48] A. Kern, C. Heid, W.H. Steeb, N. Stoop, R. Stoop, "Biophysical parameters modification could overcome essential hearing gaps," *PLoS computational biology*, vol. 4, no. 8, Aug 2008. [Online]. http://www.ploscompbiol.org/article/info%3Adoi%2F10.1371%2Fjournal.pcbi.1000161

[49] H.L.F. Helmholtz, *On the Sensations of Tone as a Physiological Basis forthe Theory of Music*, 4th ed.: Longmans, Green and Co., 1912.

[50] G.V. Bekesy, *Experiments in Hearing*.: McGraw-Hill, 1960.

[51] H. Krishnamoorthi, "Incorporating Auditory Models in Speech/Audio Applications," PhD Dissertation, Dept. of Elec. Engg., ASU, Tempe AZ, 2011.

[52] J.J. Thiagarajan and A. Spanias, *Analysis of the Mpeg-1 Layer III (Mp3) Algorithm Using Matlab*, Synthesis Lectures on Algorithms and Software in Engineering ed.: Morgan & Claypool, 2011.

[53] S.S. Stevens, "A scale for the measurement of a psychological magnitude: loudness," *Psychological Review*, vol. 43, no. 5, pp. 405-416, Sep. 1936.

[54] E. de Boer, "Synthetic whole-nerve action potentials for the cat," *The Journal of the Acoustical Society of America*, vol. 58, no. 5, pp. 1030-1045, Nov. 1975.

[55] E. de Boer and H.R. de Jongh, "On cochlear encoding: Potentialities and limitations of the reverse-correlation technique," *The Journal of the Acoustical Society of America*, vol. 63, no. 1, pp. 115-135, Jan. 1978.

[56] D. Schofield, "Visualisations of speech based on a model of the peripheral auditory system," NASA STI/Recon Technical Report N, 1985.

[57] R.D. Patterson, "Auditory filter shapes derived with noise stimuli," *The Journal of the Acoustical Society of America*, 1976.

[58] R.A. Lutfi and R.D. Patterson, "On the growth of masking asymmetry with stimulus intensity," *The Journal of the Acoustical Society of America*, 1984.

[59] S. Rosen and R.J. Baker, "Characterising auditory filter nonlinearity," *Hearing research*, 1994.

[60] B.C.J. Moore and B.R. Glasberg, "Formulae describing frequency selectivity as a function of frequency and level, and their use in calculating excitation patterns," *Hearing Research*, 1987.

[61] T. Irino and R.D. Patterson, "A time-domain, level-dependent auditory filter: The gammachirp," *The Journal of the Acoustical Society of America*, 1997.

[62] R. P. Hellman, "Rationale for a new loudness standard," *The Journal of the Acoustical Society of America*, vol. 119, no. 5, pp. 3291-3291, May 2006.

[63] B.R. Glasberg and B.C.J. Moore, "A model of loudness applicable to time-varying sounds," *Journal of Audio Engineering Society*, vol. 50, no. 5, pp. 331-342, May 2002.

[64] *Sound Quality Assessment Material Recordings for Test Subjects*. Brussels: EBU Technical Centre, Sep. 2008.

[65] L. Cohen, *Time-Frequency Analysis*. New Jersey: Prentice Hall PTR, 1995.

[66] A. Spanias, *Digital Signal Processing: An Interactive Approach*.: Lulu, 2007.

[67] T. Painter and A. Spanias, *Audio Signal Processing and Coding*.: Wiley-Interscience, 2006.

[68] H. Schussler, "A stability theorem for discrete systems," *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1976.

[69] J. Liu, J.J. Thiagarajan, A.S. Spanias, K.N. Ramamurthy, S. Hu, S., M.K. Banavar, "iPhone/iPad based interactive laboratory for signal processing in mobile devices," in *American Society for Engineering Education. American Society for Engineering Education*, 2011.

[70] J. Liu, A. Spanias, M.K. Banavar, J.J. Thiagarajan, K.N. Ramamurthy, S. Hu, X. Zhang, "Work in progress—Interactive signal-processing labs and simulations on iOS devices," in *Frontiers in Education Conference (FIE), 2011*, 2011, pp. F2G-1.

[71] J. Liu, "Interactive Laboratory for Digital Signal Processing in iOS Devices," Master Thesis, Arizona State University, May 2011.

[72] H. Krishnamoorthi, V. Berisha and A. Spanias, "Method and system for determining an auditory pattern of an audio segment," Application US20110150229 A1, June 2011.

[73] V. Berisha and A. Spanias, "Split-band speech compression based on loudness estimation," Grant US8392198 B1, March 2013.

[74] K.N. Ramamurthy and A. Spanias, *MATLAB® Software for the Code Excited Linear Prediction Algorithm: The Federal Standard-1016*.: Morgan and Claypool Publishers, 2010.

APPENDIX A

SPEECH AND AUDIO PROCESSING ASSESSMENT EXERCISES USING iJDSP

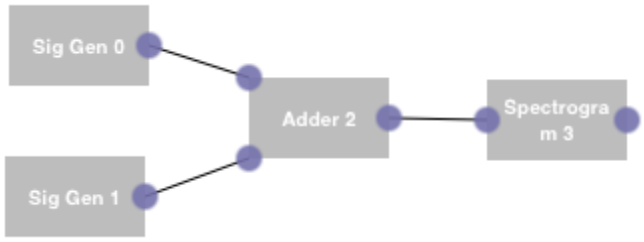


Figure A.1: Spectrogram simulation setup.

A.1 Spectrogram

The objective of this exercise is to visualize the spectrograms of signals, a popular time-frequency representation for speech and audio signals, and hence, understand the properties of spectrograms.

In this exercise, the focus is on examining and visualizing the spectrograms of signals in the *Signal Generator* and *long signal generator* functions. Set up the block diagram shown in Figure A.1 by connecting two *Signal Generator* blocks to an *Adder* and connecting it to a *Spectrogram* block.

Configure one signal generator to create a sinusoid of frequency 0.2π radians with a Pulsewidth of 256 samples. Configure the other signal generator to create a sinusoid of frequency 0.3π radians with a Pulsewidth of 128 samples. Choose the *spectrogram* block and open its parameter configuration window.

Choose: FFT Length = 256, Window size = 200 samples, Window overlap = 192 samples, Window Type: “Rectangular”. Press Update to view the spectrogram for the current frame.

Notice that the sinusoids spectra are visible as horizontal spectral lines.

Problem 1.1

With the rectangular window, one may be able to see the sidelobes around the harmonics. These are visible as horizontal red lines on either side of the thick red lines representing the input sinusoid tones. The sidelobes arise due to the spectral characteristics of the rectangular window.

Now, return to the spectrogram parameter configuration menu, choose a Triangular window and view the spectrogram for the entire signal.

The resolution of the harmonics in the spectrogram *improved*. (True/False)

The intensity of sidelobes *increased*. (True/False)

Now, choose the Blackman window, and view the spectrogram.

In comparison with the Triangular window, the spectral resolution of the sinusoids in the spectrogram *improved*. (True/False)

Problem 1.2

Now, choose the window length to be 64, change the overlap to 56, and view the spectrogram.

With change in window length, the spectral resolution of the sinusoids in the spectrogram has *improved*. (True/False)

The side lobes have *reduced*. (True/False)

Change the overlap to 32 and view the spectrogram again.

With reduced overlap, the temporal resolution has improved (True/False).

With reduced overlap, the spectral resolution is unchanged (True/False).

A.2 Linear Predictive Coding

The objective of this exercise is to understand the concepts underlying the representation of speech signals through the linear predictive coding technique. Linear Predictive Coding (LPC) is a popular technique for representing speech signals, where speech is modeled as a signal produced by exciting a system with either a train of impulse, or with white noise. The system being excited is modeled as an all-pole filter $S(z)= 1/A(z)$. The filter coefficients of the denominator $A(z) = 1 + \sum_{i=1}^{M} a_i z^{-i}$ as well as the exciting signal are estimated by the LPC technique, given the generated speech signal. The coefficients of the filter $\{a_i\}_{i=1}^{M}$ are called the LPC coefficients. The exciting signal is also referred to as the LPC residual. This representation is suitable for modeling the human speech production mechanism, where the vocal cords vibrate to produce the exciting signal. The vocal tract, which represents the system, resonates due to the excitation and hence, produces the sound. The filter $S(z)$ represents the response of the vocal tract and the residual represents the vocal cord vibrations.

In the menu of blocks in iJDSP, choose “LPC Quantization Setup 2”. It will

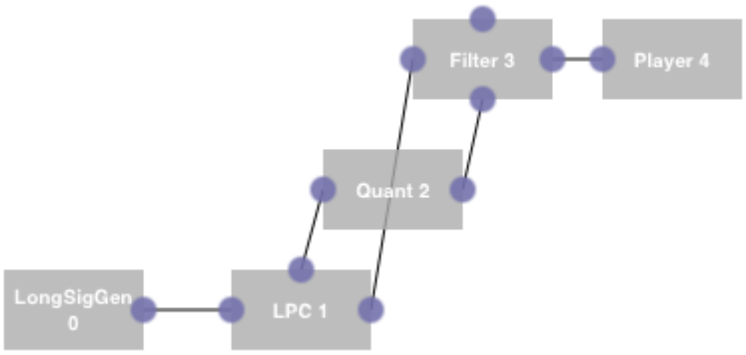


Figure A.2: Linear Predictive Coding analysis-synthesis setup.

automatically setup the simulation in Figure A.2.

Problem 2.1

Select the *long signal generator*, and choose the 10^{th} frame of the "Male Speaker" signal. You will find a frame of quasi-periodic speech plotted on the right. Due to the vibration of vocal cords, voiced sounds have a quasi-periodic nature, as opposed to unvoiced sounds such as /f/ or /s/, which are generated by mechanisms not involving the vocal cords. The chosen frame shows a voiced sound.

Return to the canvas and select the *LPC* block to view its details. Let the order remain as 10 and press **Update**. The LPC coefficients of the signal will be displayed. Press the **Options** button and choose 'Input Signal' to view the input signal frame. Measure the distance between successive periods of the signal. You can see the coordinates of the points by the pressing on the points, which will display the green cross hairs over the points and their coordinates on the lower left portion of the plot view.

The difference obtained is in the units of *samples*. Multiplying this difference with the sampling period (1/(8kHz)), the period is obtained in seconds.

This period is defined as the pitch period of the voiced speech and its reciprocal is defined as the pitch frequency or the fundamental frequency (in Hz). Compute the fundamental frequency.

Now, press the **Options** button and choose 'Residual Signal' to view the exciting signal. Find the time difference between the successive impulses in the residual, and compare it with the distance period of the signal. Are they similar?

In particular, voiced sounds such as vowels have a strong first formant. The effect of this can be observed in the time domain of the speech signal. Choose the 'Input Signal'

option again in the LPC user interface. Compute the time difference between successive rising zero-crossings of the signal.

Now, press the **Options** button and choose 'Frequency Response' to view the magnitude response of the LPC coefficients. Find the frequency of the first formant (denote it as ω_1) (the first peak in the response). Compute $\omega_1/2\pi$ to get the normalized frequency. Now, compute the reciprocal of the frequency, and compare it with the distance between the *rising (or falling)* zero crossings of the signal in time domain. Are they similar?

Problem 2.2

In Figure A.2, the LPC filter is excited by the LPC residual to resynthesize the speech signal. This part explores the effect of quantizing LPC coefficients on the resynthesized sound quality. Disconnect the Long Signal Generator from the LPC block and connect a Sound Recorder block to the LPC block. Record speech at a sampling rate of 8 kHz in the Sound Recorder. Currently, the quantizer has a bit precision of 6 bits. Double tap on the Sound Player block and press 'Parse' to process all frames of the recorded speech. Again, double tap on the Sound Player block and press 'Play Sound' to the play the resynthesized sound.

Do you notice any artifacts in the resynthesized sound? (Yes/No)

If you increase the number of bits in the quantizer block, parse the signal again at the Sound Player, and replay the Sound at the Sound Player, do you notice any reduction in the resynthesized sound quality? (Yes/No)

Thus, it can be concluded that increase in the number of bits for quantizing the LPC coefficients results in reducing artifacts in the resynthesized speech. (True/False)

A.3 Line Spectrum Pairs

Line Spectrum Pairs (LSPs) are polynomials derived as a pair of linear phase filters from the LPC filter coefficients. If the LPC synthesis all-pole filter of order M is $F(z)= 1/A(z)$, then the line spectrum pairs are:

$$P(z) = A(z) + z^{-(M+1)}A(z^{-1})$$

$$Q(z) = A(z) - z^{-(M+1)}A(z^{-1})$$

It is easily deducible that $P(z)= z^{-(M+1)}P(z^{-1})$ and $Q(z)= -z^{-(M+1)}\,Q(z^{-1})$. If M is even, then $P(z)$ is a linear phase filter of odd order with even symmetry. Hence, it has a zero at $z = -1$. Similarly, $Q(z)$ is a linear phase filter of odd order with odd symmetry. Hence, it has a zero at $z = 1$. The original LPC coefficients can be obtained from the line spectral pairs by adding them as

$$A(z) = \frac{1}{2}\big(P(z)+ Q(z)\big)$$

The polynomials $P(z)$ and $Q(z)$ representing the Line Spectrum Pairs have the property that when the LPC filter is minimum phase, i.e. the filter $1/A(z)$ is stable, their zeros lie *on* the unit circle. This allows the filters to be represented simply by the frequencies of their zeros. That is, given just the values of these frequencies, $A(z)$ can be reconstructed. These frequencies are referred to as Line Spectral Frequencies (LSFs). Another interesting property of the LSFs is that when the all the roots of $A(z)$ are within the unit circle, the zeros of $P(z)$ and $Q(z)$ are interlaced on the unit circle. These properties will be examined in the following problems.

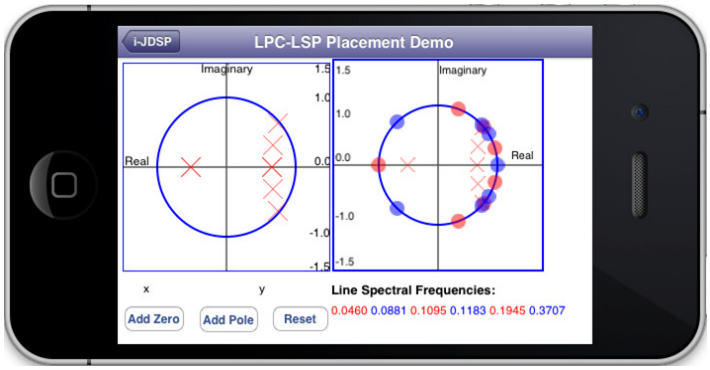


Figure A.3: A test case for the PZ to LSF Demo block.

Problem 3.1

Add the *PZ to LSF Demo* block to the simulation canvas. Double tapping on the block will open its interface. Here, on the z-plane on the left, you are allowed to create a test case of an LPC filter by placing and moving poles. Now, add a pair of poles by pressing the **Add Pole** button and place it within the unit circle. On the plot on the right, along with the poles you placed, the zeros of the line spectral pairs of the LPC filter are also shown. The LSPs are color coded, with the zeros of one filter being shown in blue and the other being shown in red. Notice that the zeros of the two filters alternate on the unit circle.

Now, when a pair of poles is moved out of the unit circle, notice what happens to that pair's corresponding LSFs and the locations of those LSP zeros on the z-plane. Note that the values of the corresponding LSFs are not in increasing order anymore, and the zeros do not alternate.

Problem 3.2

In this exercise, the effects of quantizing LPC coefficients on the stability of the LPC filter will be observed and compared with quantization of LSFs. Create a pole-zero

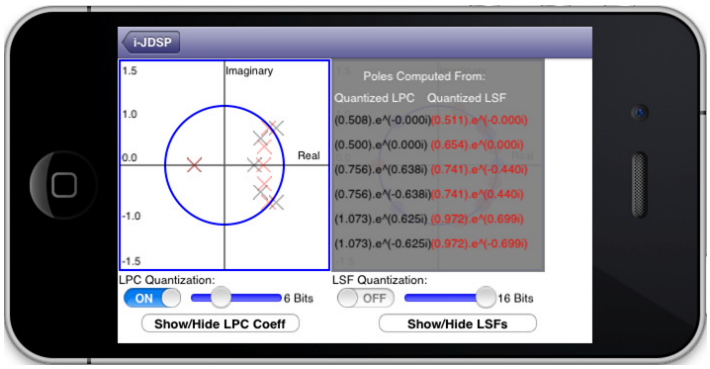


Figure A.4: The view of the *LPC-LSP Quantize Demo* block with LPC quantization enabled and the quantized LPC pole locations being listed next to the LPC PZ plot.

plot in the *PZ to LSF Demo* block similar to that shown in Figure A.3 (note that the created LPC filter's order *must* be even for this example).

Now, add the *LPC-LSP Quantization Demo* block to the canvas, and connect it to the *PZ to LSF Demo* block. This will pass the LPC filter coefficients generated by the *PZ to LSF Demo* block as input to the *LPC-LSP Quantization Demo* block. Then, double tap on the *LPC-LSP Quantization Demo* block to open its user interface.

Enable the LPC quantizer and gradually reduce the number of bits for LPC quantization. Press the **Show/Hide LPC Coeff** button to see the pole values for the quantized LPC filter and the LPC filter reconstructed from quantized LSFs. At low bit resolutions, poles of the quantized LPC filter will move out of the unit circle, hence, becoming unstable. But it can be seen that the poles of the LPC filter reconstructed from quantized LSFs have a lesser tendency to move out of the unit circle.

A.4 Loudness

The objective of this exercise is to understand the principle of loudness, which is the measure of perceived intensity. Loudness is a non-linear quantity, which depends on

the spectrum of the signal. The measure of loudness is different from the conventional scale of measurement of signal energy in dB. Two signals can have the same energy, but different perceived loudness.

Loudness is measured in 'sones'. 1 sone is the loudness of a 1 kHz sinusoid which has a sound pressure (or energy) of 40 dB. The conversion from energy E (dB) to loudness L in sones is defined by the following law, where k is a constant dependent on the signal spectrum.

$$L = k10^{0.03*E}$$

The human ear consists of many hair-like cells in the innermost part of the ear (called the cochlea), which convert incident sound waves into neural impulses (called excitations). Different hair cells respond to different parts of the spectrum. These cells act like a bank of selective bandpass filters.

The loudness is higher when there is higher neural excitation. The loudness pattern, which quantifies the perceived loudness at a given frequency, is derived from these neural excitations. The loudness pattern is integrated to obtain the (total) loudness of the signal.

Problem 4.1

Add a Signal Generator producing a sinusoid of frequency 0.02π rad, gain 0.1 and pulsewidth of 256 samples. Connect it to the Psychoacoustic Model block. Now, double tap on the psychoacoustic model block in the canvas and choose the 'Sampling Rate: 44100 Hz' option to view its interface. Choose 'Loudness Pattern' from the pop-down menu of the **Options** button. This shows the loudness pattern along the spectrum. The energy of the signal and the total loudness of the signal are displayed along with the plot

of the loudness pattern. Note down the energy and loudness values in the table given below. Now, change the frequency of the sinusoid to 0.1π and note down the new energy and loudness values. Similarly, note the values for a frequency of 0.2π.

Sinusoid Frequency	Energy (dB)	Loudness (Sones)
0.02π		
0.1π		
0.2π		

The loudness of a sinusoid gradually increases with increasing frequency up to a point and then decreases with further increase in frequency. (True/False)

Problem 4.2

Case I:

In the Signal Generator in the previous problem, set the frequency to 0.202π, set the gain to 0.017136 and note the energy and loudness values.

Energy (dB): _____ Loudness (Sones): _____

Case II:

Now, create the setup shown in Figure A.5. Set the following parameters in the blocks:

Sig Gen 0 - Gain = 0.01, Frequency = 0.04π, Pulsewidth = 256.

Sig Gen 1 - Gain = 0.01, Frequency = 0.044π, Pulsewidth = 256.

Sig Gen 2 - Gain = 0.01, Frequency = 0.048π, Pulsewidth = 256.

Now, note the energy and loudness of the cumulative signal in the Psychoacoustic Model interface for the Sampling Rate of 44100 Hz.

Energy (dB): _____ Loudness (Sones): _____

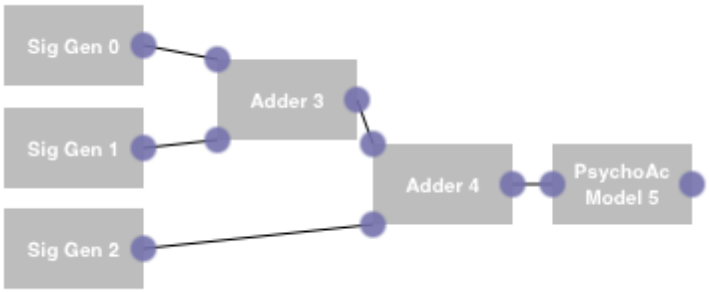


Figure A.5: Setup for loudness estimation.

Compare these values with the values previously noted for the sinusoid of frequency 0.202π.

In this example, the energies in the two cases are the same, but the presence of larger number of spectral components results in *higher* perceived loudness. (True/False)

APPENDIX B

LOUDNESS CONTROL SIMULINK DEMO

B.1 Introduction

This appendix describes the Simulink model implementing the Moore and Glasberg Model for loudness estimation as part of the real-time automatic loudness control system. Simulink models help elegantly describe the structure of a system as a block diagram. A complex system can be abstracted and represented as a simple network of subsystems, where each subsystem is a network of smaller less complex subsystems. Another advantage of simulating a system as a Simulink model is that real-time demonstration of processing, real-time interaction with the system and visualization of data is possible, which is an easy way of evaluating the feasibility of real-time implementation. In addition, fixed-point and floating point models can be generated from floating point models, which is beneficial for testing fixed-point implementations. It is also possible to generate embedded C code for target DSPs. The Moore and Glasberg model implemented as a Simulink model is shown in Figure B.1.

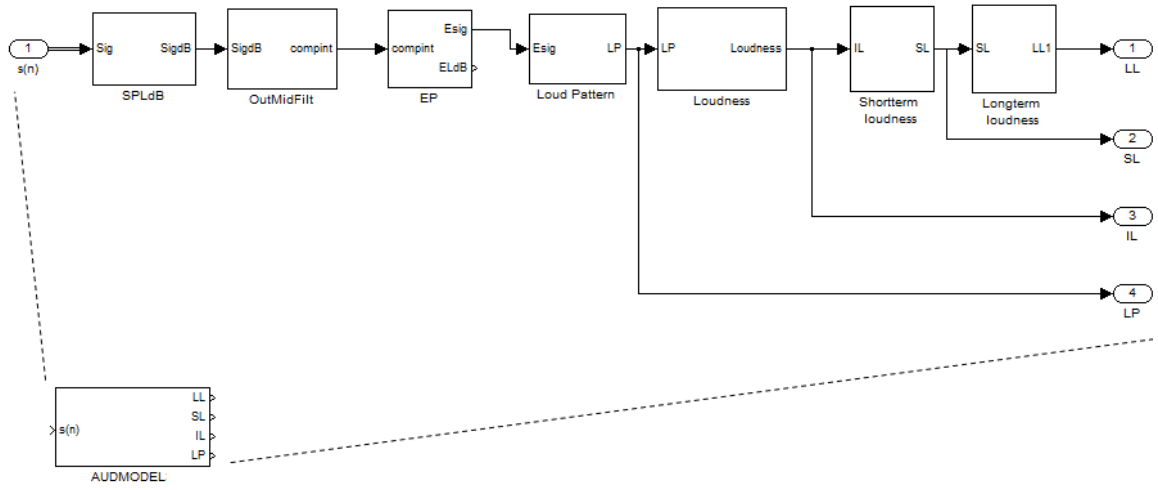


Figure B.1: The AUDMODEL block implements the Moore and Glasberg model for loudness estimation.

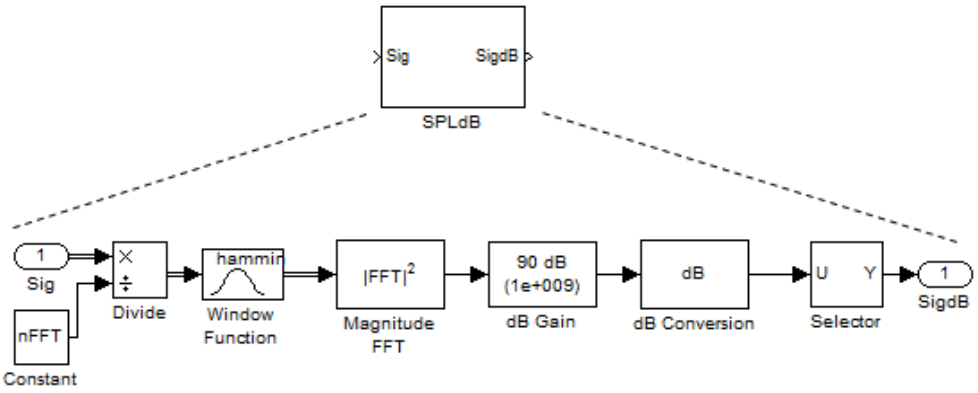


Figure B.2: Sound pressure level normalization Simulink model.

The model accepts a signal frame and computes the excitation pattern (Esig), loudness pattern (LP), the instantaneous loudness (IL), and the short-term (SL) and long-term (LL) loudness. The functional blocks comprising the AUDMODEL block are briefly described below.

B.1.1. Sound Pressure Level Normalization

The power spectrum of the input signal frame is computed after preprocessing using a Hamming window. The input signal is normalized to convert the full-scale decibel level to 0 dB. The spectrum is then normalized to a full-scale sound pressure level of 90 dB SPL, and the resulting power spectrum in dB is produced as output. The Selector block chooses only the first half of the FFT points as the signal is real. The DC component is not transferred as it is not in the range of human hearing.

B.1.2. Outer-Middle Ear Filtering

The outer and middle ear filter magnitude responses are modeled by this block in the 'OMEC' vector. The input signal is filtered by this block before being presented to the block modeling the auditory filters in the inner ear (which is referred to as the effective spectrum). The filtering is performed by adding the dB values of the magnitude

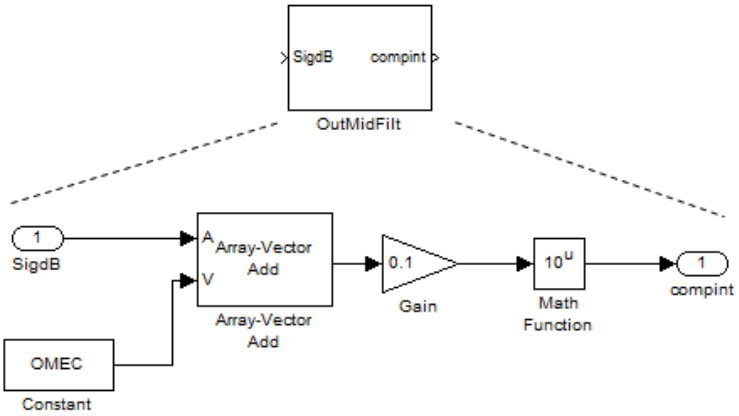


Figure B.3: Outer-middle ear filtering in the auditory model.

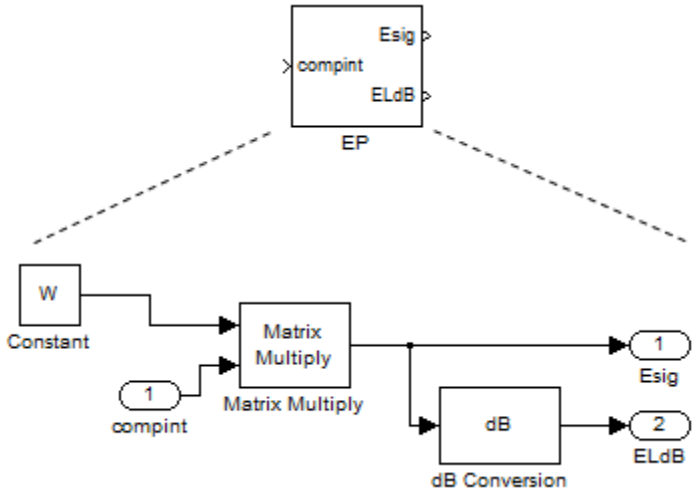


Figure B.4: Functional block for evaluating excitation pattern.

spectra of the signal and the filter and the response is recomputed in linear scale and produced as output.

B.1.3. Computing the Excitation Pattern

The excitation pattern is the computed as the energy of the signal at the output of each auditory filter in the filter bank. The energy at the output of each auditory filter is computed as the sum of the energies of the individual spectral components of the filter output spectrum. The excitation pattern is computed as a matrix multiplication. The

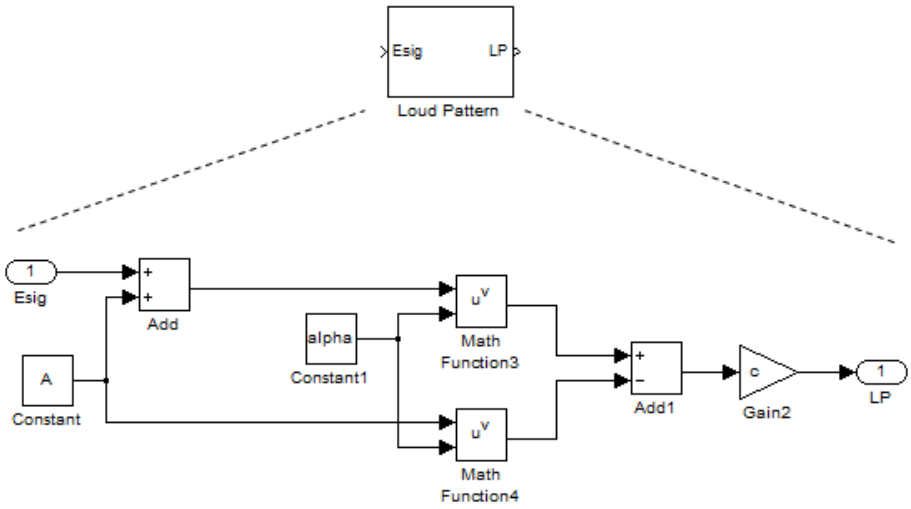


Figure B.5: Block diagram for evaluating the loudness pattern given an excitation pattern.

vector of the effective spectrum fed through the input pin is multiplied by the auditory filter magnitude responses pre-computed and stored in the matrix *W*. Each row in *W* contains the normalized response of the corresponding auditory filter. The auditory filters have equally spaced center frequencies in the auditory scale. When the FFT length is *N* and the number of detectors is *D*, the matrix *W* has the dimensions *D*×*N*. The excitation pattern is a *D*-dimensional vector.

B.1.5. Loudness Pattern Computation

The loudness pattern implements the equation (3.15) to compute the loudness pattern from the input excitation pattern. The numerical constants in the equation are c=0.047 and α=0.2.

B.1.6. Total Loudness

The specific loudness is integrated as shown in Figure B.6 to obtain the total loudness. The integration is performed according to equation (3.19). The integration is approximated for discrete loudness pattern by computing the area of trapeziums under the

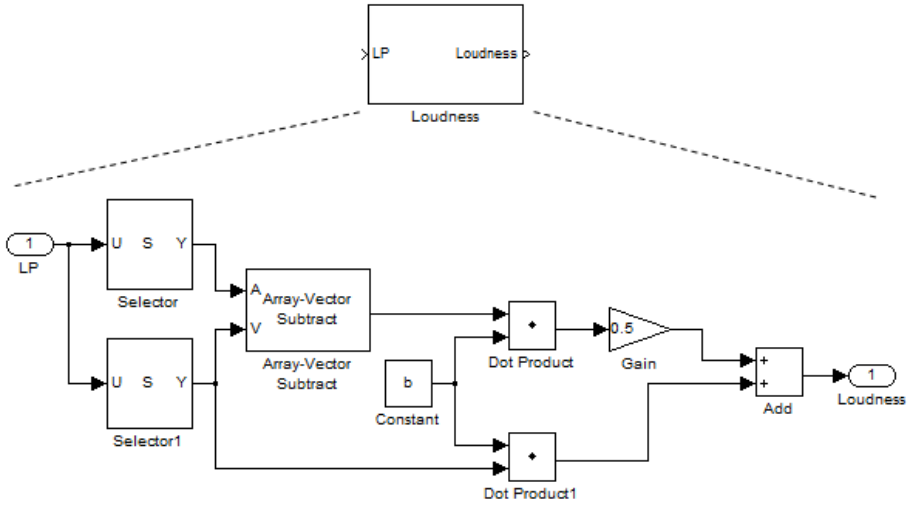


Figure B.6: Computing the total loudness from specific loudness.

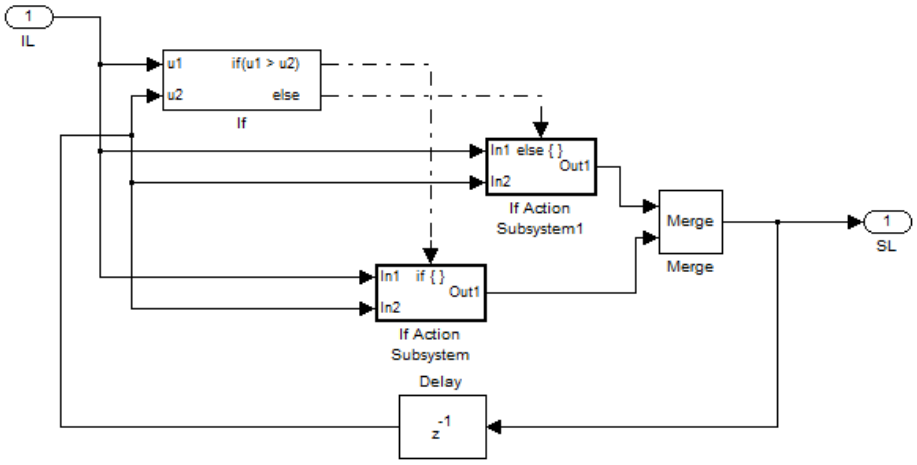


Figure B.7: Model for computing the short-term or long-term loudness.

curve. The constant ‘b’ is equal to the distance between consecutive detectors in the auditory scale.

B.1.7. Short-term and Long-term Loudness

The short-term loudness and long term loudness are computed by the Simulink sub-system shown in Figure B.7, according to equation (3.21). The time constants for the short-term and long-term loudness are defined in the ‘if Action Subsystems’.

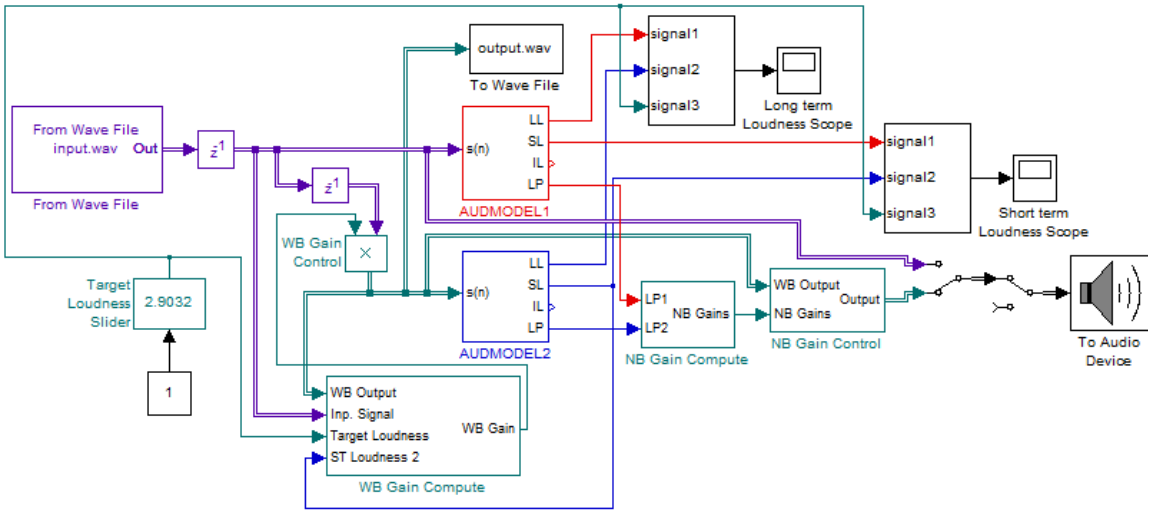


Figure B.8: A real-time loudness control system.

B.2. Real-time Loudness Control Using the Moore and Glasberg Model

The real-time loudness control system is a loudness-based adaptive gain control system. The system estimates the loudness of a frame of output signal, and adapts the value of the gain applied to it to maintain it at the target loudness, fixed by the Target Loudness Slider (see Figure B.8). The system applies a wide-band gain control mechanism to control the overall loudness through a single multiplicative gain that is derived from equation (5.9). Then, a narrowband gain control is applied to the resulting signal as defined in equation (5.13).