Lightweight Objective Quality of Voice Estimation Through Machine Learning

Daniel Riordan

Institute of Technology, Tralee, Co. Kerry, Ireland.

Follow this and additional works at: https://sword.cit.ie/allthe

Part of the Artificial Intelligence and Robotics Commons, and the Systems and Communications Commons

Recommended Citation


This Master Thesis is brought to you for free and open access by the Dissertations and Theses at SWORD - South West Open Research Deposit. It has been accepted for inclusion in Theses by an authorized administrator of SWORD - South West Open Research Deposit. For more information, please contact sword@cit.ie.
Lightweight Objective Quality of Voice Estimation Through Machine Learning

by

Daniel E. Riordan

Supervisor: Dr. Pat Doody

A Thesis Submitted to the Institute of Technology Tralee in Fulfilment of the Requirements for the Master of Science Degree of Higher Education and Education Awards Council, April 2008
Lightweight Objective Quality of Voice Estimation Through Machine Learning

Daniel Riordan

Abstract

Communication systems are undergoing constant and rapid innovation, both at the design stage and in the field. This in turn has led to an increasing need for fast, efficient, portable and economic methods for the testing of these systems. For voice carrying communication systems the quality of the transmitted voice that the system produces is a large factor in the overall performance rating of the system. This measure is known as the ‘Quality of Voice’ (QoV) and can be evaluated either subjectively or objectively.

Speech quality is a complex subjective phenomenon that can be best quantified by subjective testing. A subject QoV measurement requires a ‘listener’ to rate a sample of speech produced by the system. To achieve accurate results an average rating, or Mean Opinion Score (MOS), must be found from a large panel of listeners. This results in subjective QoV testing being a highly expensive and time consuming process to conduct.

Objective methods of QoV estimation attempt to predict the results a panel of listeners would produce when presented with a given sample of speech. Objective QoV estimation techniques comprises both Intrusive and Non-Intrusive methods.

Non-intrusive QoV estimation methods involve an automated algorithm taking account of various speech impairment factors based on the operational parameters of the system under test. These parameters are then used predict the distortion levels introduced to the speech during transmission by the system and thereby an estimate of the QoV capabilities of the system can be made.

Intrusive QoV estimation methods involve a comparison between an original speech sample and a resulting speech sample which has been degraded by transmission through the system. By performing a distance measure between the original and degraded speech samples, an estimation of the QoV capabilities of the system can be made.

This project aims to create an objective method for the estimation of the voice transmission capabilities of a system using Artificial Intelligence (A.I.) techniques. Algorithms for both Intrusive and Non-intrusive objective QoV estimation through machine learning will be investigated during this thesis. It is hoped that the application of A.I. techniques to objective QoV estimation algorithms will improve the efficiency and economy of communication system testing.
Acknowledgements

I would like to acknowledge the help and guidance provided to me by my supervisor, Dr. Pat Doody. While allowing me great freedom within my research to pursue the avenues I saw fit, his assistance, advice and encouragement proved invaluable. Pat’s expertise in technical writing has also been of great benefit to me and the quality of this thesis.

Special thanks also go all of my colleagues in the Science and Computing Department at I.T. Tralee, especially those within the Wireless and Mobility Research Group, whose advice, technical knowledge and insights have helped me throughout my research.

I must also thank my family and friends for their constant support throughout my Masters research. I could not have made it through without them.
# Table of Contents

Abstract......................................................................................................................ii
Acknowledgements...................................................................................................iii
Table of Contents......................................................................................................iv
List of Figures............................................................................................................x
List of Tables...........................................................................................................xiii

1 Introduction........................................................................................................1
   1.1 Project Motivation 1
   1.2 Aims of Project 3
   1.3 Thesis Structure 3

2 The Human Auditory System...........................................................................5
   2.1 Introduction 5
   2.2 The Anatomy of the Human Auditory System 6
       2.2.1 The Outer Ear 6
       2.2.2 The Middle Ear 7
4.2.2 Subtractive Distortion
4.2.3 Packet-Loss
4.2.4 Jitter

4.3 Subjective Methods of QoV Measurement
4.3.1 Absolute Category Rating (ACR)
4.3.2 Degraded Category Rating (DCR)
4.3.3 Comparison Category Rating (CCR)

4.4 Intrusive Objective QoV Estimation Methods
4.4.1 Bark Spectral Distortion (BSD)
4.4.2 Measuring Normalised Blocks (MNB)
4.4.3 Perceptual Speech Quality Measure (PSQM)
4.4.4 Perceptual Analysis/Measurement System (PAMS)
4.4.5 Radial Basis A.N.N. Intrusive QoV Estimation

4.5 Perceptual Evaluation of Speech Quality (PESQ)
4.5.1 Level Align and Equalisation
4.5.2 Short Term FFT
4.5.3 Delay Estimation and Compensation
4.5.4 Frequency Warping
4.5.5 Loudness Density Mapping
4.5.6 Disturbance Density Computation
4.5.7 Cognitive Modelling

4.6 Non-Intrusive Objective Methods of QoV Estimation
4.6.1 The E-Model
4.6.2 A.N.N. Based Non-Intrusive Objective QoV Estimation

4.7 Summary

5 Modelling of the Human Auditory System

Through Machine Learning

5.1 Introduction
5.2 Methodology
5.2.1 Training / Testing
5.2.2 Analysis of Results
5.3 An A.N.N. Implementation of the Perceptual Loudness Measure (Phon)
5.3.1 Motivation for A.N.N. Implementation .......................... 80
5.3.2 Development of A.N.N. Architecture ............................ 81
5.3.3 Training/Testing ..................................................... 81
5.3.4 Results ............................................................... 82
5.4 An A.N.N. Implementation of the Perceptual Loudness Measure (Sone) .......................................................... 86
5.4.1 Motivation for A.N.N. Implementation .......................... 86
5.4.2 Development of A.N.N. Architecture ............................ 86
5.4.3 Training/Testing ..................................................... 88
5.4.4 Results ............................................................... 89
5.5 An A.N.N. For the Frequency to Critical Band Rate Conversion .......................................................... 91
5.5.1 Motivation for A.N.N. Implementation .......................... 91
5.5.2 Development of A.N.N. Architecture ............................ 92
5.5.3 Training/Testing ..................................................... 92
5.5.4 Results ............................................................... 94
5.6 An All-In-One A.N.N. Pure-Tone Perception Model ................. 96
5.6.1 Motivation for A.N.N. Implementation .......................... 96
5.6.2 Development of A.N.N. Architecture ............................ 96
5.6.3 Training/Testing ..................................................... 96
5.6.4 Results ............................................................... 97
5.6 Conclusions .................................................................. 99

6 Intrusive Objective Quality of Voice Estimation

Through Machine Learning ..................................................... 100

6.1 Introduction ............................................................ 100
6.2 Requirements ........................................................... 101
6.3 Pre-Processing Algorithm ............................................. 102
6.3.1 Formatting of Inputs ............................................... 102
6.3.2 IRS Filtering ......................................................... 104
6.3.3 Division of Samples into 32ms Frames .......................... 105
6.3.4 Time to Frequency Domain Conversion ......................... 106
6.3.5 Utterance Identification ........................................... 108
6.3.6 Delay Estimation and Compensation .......................... 112
6.3.7 16-bit Linear PCM to Sound Pressure Level 122
6.3.8 Conversion to Perceptual Domain 124
6.3.9 Disturbance Distance Calculation 125
6.3.10 Cognitive Modelling 127
6.4 Perceptual Mapping of Distance Measure to QoV Score 131
6.4.1 A.N.N. Architecture Development 132
6.4.2 Training Data Generation 133
6.4.3 Training/Testing 134
6.4.4 Results 135
6.5 Conclusions 137

7 Non-Intrusive Objective Quality of Voice Estimation
Through Machine Learning ..........................................................138
7.1 Introduction 138
7.2 Requirements 139
7.3 Evaluation of System Operation Parameters with regard to
QoV Estimation 140
7.3.1 Codec Type 140
7.3.2 Speaker Gender 142
7.3.3 Unconditional Loss Probability (ULP) 142
7.3.4 Conditional Loss Probability (CLP) 143
7.3.5 ULP and CLP 144
7.4 Training Set Generation 147
7.5 VoIP Network Simulator 148
7.5.1 Codec Types 149
7.5.2 Packet-Loss Simulation 149
7.6 A.N.N. Architecture 152
7.7 Training/Testing 152
7.8 Results 154
7.9 Conclusions 157

8 Summary and Conclusions ......................................................... 158
8.1 Recommendations for Further Research 160
List of Figures

2.1 The Outer Ear ................................................................. 7
2.2 The Middle Ear ............................................................. 8
2.3 The Inner Ear ............................................................... 9
2.4 Critical-Band Rate vs. Frequency ............................... 12
2.5 The Equal Loudness Contours ................................. 13
2.6 Simultaneous Masking .................................................. 16
2.7 Temporal Masking ....................................................... 17
3.1 The Main Features of a Biological Neuron ............... 21
3.2 An Artificial Neuron Model ................................. 25
3.3 The Unit Step Function ............................................... 28
3.4 The Sigmoid Function .................................................. 29
3.5 The Radial Based Function ........................................... 30
3.6 Recurrent Neural Network ................................. 31
3.7 A Feed-Forward A.N.N. ............................................. 32
3.8 Over-Fitting ............................................................. 39
6.9 Results of Utterance Splitting/Matching to Eliminate All Delay ......................................................... 120
6.10 Example of Misalignment .................................................................................................................... 121
6.11 Conversion from Linear PCM to Decibels Measure ............................................................................. 123
6.12 Conversion of Spectral Power and Frequency to Perceived Pitch and Loudness ................................ 125
6.13 Disturbance Distance ......................................................................................................................... 127
6.14 Conversion from Disturbance Distance to Perceived Distortion ....................................................... 130
6.15 Overview of the Operation of Decision Making A.N.N. ................................................................... 131
6.16 Sensitivity About The Mean .............................................................................................................. 133
6.17 M.S.E. Vs. Number of Neurons in Hidden Layer .............................................................................. 135
6.18 PESQ Generated Target Values vs. A.N.N. Based Algorithm QoV Scores ........................................ 136
6.19 Validation Set Target Values vs. A.N.N. Output ................................................................................ 137

7.1 Comparison of Codecs Using the QoV Measure .................................................................................. 141
7.2 ULP vs. QoV Score ............................................................................................................................... 143
7.3 CLP vs. QoV Score ............................................................................................................................... 143
7.4 ULP vs. CLP vs. QoV Score ................................................................................................................ 144
7.5 Sensitivity About The Mean ................................................................................................................. 146
7.6 Overview of A.N.N. for Non-Intrusive Objective QoV Estimation ................................................... 146
7.7 VoIP Network Simulator ...................................................................................................................... 148
7.8 2-State Gilbert Model .......................................................................................................................... 150
7.9 & 7.10 M.S.E. Vs. Number of Neurons in Hidden Layer ....................................................................... 153
7.11 Non-Intrusive Objective QoV Estimation A.N.N. Training Procedure ............................................ 153
7.12 A.N.N. QoV Estimations vs. PESQ Generated Target Values ......................................................... 155
7.13 A.N.N. QoV Estimation Vs. PESQ QoV Estimation ............................................................................ 156
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>ACR Opinion Scale</td>
<td>49</td>
</tr>
<tr>
<td>4.2</td>
<td>DCR Scale of Degradation</td>
<td>49</td>
</tr>
<tr>
<td>4.3</td>
<td>CCR Scale</td>
<td>50</td>
</tr>
<tr>
<td>4.4</td>
<td>Provisional Guideline for R-Value to QoV Score Mapping</td>
<td>67</td>
</tr>
<tr>
<td>5.1</td>
<td>Results from Training of A.N.N. Model of the Perceptual Loudness Conversion (Phon)</td>
<td>83</td>
</tr>
<tr>
<td>5.2</td>
<td>Results from Training of A.N.N. Model of the Perceptual Pitch Conversion (Bark)</td>
<td>93</td>
</tr>
<tr>
<td>6.1</td>
<td>Array Containing Utterance Information</td>
<td>112</td>
</tr>
<tr>
<td>6.2</td>
<td>Utterance Match Array</td>
<td>118</td>
</tr>
<tr>
<td>6.3</td>
<td>Utterance Match Array</td>
<td>122</td>
</tr>
<tr>
<td>6.4</td>
<td>Results of Training A.N.N. for the Mapping of Perceived Distortion Measure to QoV Score</td>
<td>135</td>
</tr>
</tbody>
</table>
6.5 Validation testing results of A.N.N. for Mapping of Perceived Distortion Measure to QoV Score ......................................................... 136

7.1 Results from Training of Non-Intrusive Objective QoV Estimation A.N.N. (Using CLP Measure) ..................................................... 154

7.2 Results from Validation Testing of Non-Intrusive Objective QoV Estimation A.N.N. (Using CLP Measure) ..................................................... 154

7.3 Results from Training of Non-Intrusive Objective QoV Estimation A.N.N. (Without CLP Measure) ..................................................... 155

7.4 Results from Validation Testing of Non-Intrusive Objective QoV Estimation A.N.N. (Without CLP Measure) ..................................................... 156

A.1 Subdivision of the Audible Frequency Range into Critical Bands . A-2

A.2 Look-Up Table for the Estimation of the Perceptual Loudness Measure ................................................................. A-3

A.3 Results of Testing of A.N.N. Implementation of the Conversion from Frequency and S.P.L to Perceived Loudness in Sone . A-4

A.4 Results of Testing of A.N.N. All-In-One Pure-Tone Perception Model ................................................................. A-4
Chapter 1

Introduction

1.1 Project Motivation

Speech quality is an important factor in determining the quality of service (QoS) provided by a speech based communication network. The development of any speech-based communication system will require Quality of Voice (QoV) testing at some stage during the process. As such systems have become more complex and sophisticated the need for a reliable and efficient method for the measurement of QoV has become of greater importance.

QoV measurement is a highly complex evaluation based on an array of physical, perceptual and subjective evaluations. Its rating is based upon the comfort or ease of
listening perceived by the human auditory system. While QoV is largely a subjective measure, the modelling and evaluation of the factors which affect the perceived ease of listening facilitated the employment of objective QoV measurements leading to a more economical testing process.

Traditional methods of QoV estimation include testing by subjective means. A panel of listeners is required to perform this subjective test accurately. The panel listens to a series of speech samples and grades them according to the perceived speech quality based upon pre-defined rating scales.

In a modern industrial setting subjective QoV estimation is highly inefficient, as described in section 4.3 of chapter 4. Therefore the integration of subjective QoV evaluation methods into communication system development would prove to be very time-inefficient and expensive. This has forced industry to turn to objective means of QoV testing. Objective methods for QoV estimation attempt to simulate the results subjective methods would generate.

Early QoV estimation methods operated using analytical measures such as Signal-to-Noise ratio (SNR) and Mean Square Error (MSE) to estimate QoV (Sun 2004, p.33). More modern methods have tried simulating the manner in which the human auditory system perceives speech and distortions. Others methods have attempted to map easily identifiable distortion sources, such as codec type and propagation delay, to associated QoV ratings.

With the advent of mobile communication systems, there is now a requirement for a portable, lightweight and economically viable QoV estimation algorithm. Today 'in the field' and 'on site' testing is required during both the communication system development and maintenance. For testing in the field to be practical QoV estimation algorithms must have the ability to operate on portable devices with limited resources. To allow these systems to be implemented successfully, the cost of algorithm development must be kept to a minimum.

As described later in this thesis, speaker gender and voice characteristics play an important role in objective QoV estimation. Therefore, to accommodate different user
types, QoV estimation algorithms may need to be fine-tuned for each implantation. This requires that a QoV estimation system or algorithm be easily reconfigurable.

1.2 Aims of Project

The main aim of this thesis is the application of Artificial Intelligence (A.I.) techniques to the problem of objective QoV estimation. It is expected that the application of A.I. will lead to a more lightweight QoV Estimation algorithm than existing algorithms. It is also hoped that this will allow objective QoV estimation algorithms the ability to operate on portable systems with limited hardware resources.

The use of A.I. techniques may also prove to aid a more lightweight development stage for QoV estimation algorithms. It is proposed that through the use of AI techniques, certain processing steps of existing QoV estimation algorithms may be improved upon or even eliminated.

Once a usable and efficient objective QoV estimation algorithm is developed it is envisaged that it will be easily adapted to accommodate use with different speech patterns. A.I. techniques such as Artificial Neural Networks (A.N.N.s) are easily adapted and retrained. This may lead to easily reconfigurable objective QoV estimation algorithm capable of being quickly and efficiently reconfigured for each implementation.

1.3 Thesis Structure

The main goal of this thesis is to develop an algorithm to objectively estimate the QoV capabilities of a communication system. It is proposed that advantages with respect to the development, reconfigurability and mobility of the algorithm can be achieved through the application of A.I. techniques. The branch of A.I. to be used in this thesis is Artificial Neural Networks (A.N.N.s). A.N.N.s will be employed in the development of both Intrusive and Non-Intrusive QoV estimation algorithms.
The operation and modelling of the human auditory system is described in Chapter 2. Chapter 3 presents an overview of A.N.N.s and their ability to approximate the operation of biological neural networks is demonstrated. Chapter 4 is dedicated to presenting the background of QoV estimation in communication networks. QoV estimation and many of the established methods for its implementation are described.

Chapter 5 describes the development and testing of an A.N.N. model of the human auditory system. This is a vital component of any intrusive objective QoV estimation algorithm while also being suitable for many other implementations such as speech recognition. Chapter 6 presents the development and analysis of an intrusive objective QoV estimation algorithm incorporating machine learning. Chapter 7 presents the development and analysis of a non-intrusive QoV estimation algorithm implemented entirely with A.I. techniques. Chapter 8 will summarise the accomplishments of this thesis and suggest further work to be carried out.
Chapter 2

The Human Auditory System

2.1 Introduction

The human auditory system is the sensory system which governs the perception of audio signals presented to the human ear. An understanding of the operation of the auditory system is vital to the study of Quality of Voice estimation. This understanding is required to allow the manner in which sound is perceived to be modelled. This model will allow the perception of distortions by a system user to be discerned.
Section 2.2 of this chapter will describe the anatomy of the human auditory system. Section 2.3 will discuss the manner in which the auditory system perceives sound and how this can be modelled mathematically.

2.2 The Anatomy of the Human Auditory System

The human auditory system consists of a number of sections, including the outer ear, the middle ear, inner ear and the auditory cortex of the brain. The outer, middle and inner ear act mainly as a receptor for audio information. After processing by the ear, the received audio information is then transmitted to the brain's primary auditory cortex for cognitive processing.

2.2.1 The Outer Ear

The outer ear is a fleshy cartilage composed of the auricle, also known as the pinna, the ear lobe and the external auditory canal. A detailed diagram of these features is shown in Figure 2.1 (adapted from ‘Human Ear Diagram’ No Date [n.d.]). When an audio signal encounters the outer ear it is collected and channelled into the external auditory canal by the auricle. The shape of the outer ear, along with lesser effects from the head, shoulders and torso, has a filtering effect upon the signal. These features combine to attenuate frequency components of the signal which are outside of the range 3 kHz to 12 kHz while also giving a slight boost to certain frequencies within this range. It is this frequency range, between 3kHz and 12kHz, that has been found to be vital for the perception of speech.

The reflection of sound in the pinna also adds directional information to the sound that enters the external auditory canal. The reflection of audio waves from the pinna can undergo destructive interference which results in a notch filtering of the audio signal. This notch is often referred to as the pinna notch. The frequency at which this notch is present is directly related to the angle at which the audio signal strikes the pinna. The presence of this notch allows the transmission of directional information to
the middle & inner ear and subsequently the auditory cortex of the brain. (Chiras 2002, p. 289)

![Ear Diagram](image)

**Figure 2.1: The Outer Ear**

Once the signal has entered the external auditory canal, it is then directed towards the tympanic membrane, better known as the eardrum. The eardrum marks the beginning of the middle ear. (Chiras 2002, p.289)

### 2.2.2 The Middle Ear

The Middle ear contains the eardrum, the Eustachian tube and three minute bones, the malleus (hammer), the incus (anvil) and the stapes (stirrup). These bones are collectively known as the ossicles. Figure 2.2 presents a diagram of the layout of these features (Adapted from ‘Middle Ear’ n.d.).

When an audio signal travels down the external auditory canal and hits the eardrum it causes the eardrum to vibrate. The malleus presses against the ear drum. Therefore, when the eardrum vibrates the malleus is also set in vibration. The connected incus is then caused to tremor, in turn causing the stapes to strike against
the oval window. The Oval window marks the divide between the middle and inner ear. The net result is an amplification of the audio signal being perceived. This amplification allows the signal to pass through the extracellular fluid in the cochlea of the inner ear (Sears & Winwood 1982, p346).

![Diagram of the Middle Ear](image)

**Figure 2.2: The Middle Ear**

The Eustachian tubes primary function is to equalise the pressure between the middle ear and the outer ear. When a pressure imbalance between the outer and middle ear occurs the ear becomes very susceptible to damage. This change in pressure can usually be attributed to a change in atmospheric pressure. The Eustachian tube is usually closed, but when a pressure imbalance between the middle and outer ear occurs the tube can open. This allows a passage of air in or out of the otherwise isolated middle ear cavity. The result is an equalisation of pressure between the middle and outer ear. (Chiras 2002, p.290)
2.2.3 The Inner Ear

The Inner ear contains the cochlea and the vestibular apparatus. The cochlea is a snail shaped sensory organ which contains the auditory systems receptors for hearing. The vestibular apparatus is the organ responsible for the senses of balance and position. This organ is sometimes referred to as the Semicircular Canals. These features are shown in Figure 2.3 (Adapted form 'Inner Ear' n.d.).

The cochlea is made up of three cavities, the scala vestibule, scala tympani and scala media, each filed with an extracellular fluid. The scala media contains the organ of Corti. The organ of Corti is the auditory systems audio receptor.

The cochlea is responsible for division of an audio signal into its composite frequency/intensity components and subsequent transmission to the primary auditory cortex of the brain. It is the shape of the basilar membrane together with the length of the hair cells along the length of the organ of Corti that perform this transformation (Chiras 2002, p291).

![Figure 2.3: The Inner Ear](image-url)
The basilar membrane is narrow and rigid where it meets the oval window and grows wider and more flexible as it progresses to its apex. This structure is tuned specifically to allow maximum vibrations along the cochlea at specific frequencies.

At the base of the cochlea, where it meets the oval window, maximum vibration occurs when a high frequency signal (~20 kHz) is perceived. This is facilitated by the narrow width of the membrane at this point. Closer to the apex the maximum amount of vibration occurs when a low frequency signals (~20 Hz) are being perceived. This is due to greater width of the membrane near the apex. These vibrations cause the hair cells in the area affected by the maximum vibration to vibrate. The vibration of the hairs cells cause impulses to be sent to the auditory cortex of the brain. (Hartmann 1997, p.6 & Chiras 2002, p.292)

2.2.4 The Primary Auditory Cortex

The primary auditory cortex is the area of the brain which has proved to be responsible for the cognitive perception of sound. Transmission of the audio signal, as perceived by the ear, to the auditory cortex occurs via the cochlear nerve, the brain stem and the Medial Geniculate Body. The audio signal undergoes a small amount of processing at each stage of this transmission. Finally the signal is delivered to each side of the primary cortex of the auditory cortex.

It has been found that the neurons contained in the primary auditory cortex are tonotopically organised. This means that certain neurons are stimulated when a certain frequency is presented to the ear. (Hartmann 1997, p.6)

2.3 The Perception of Sound by the Human Auditory System

Sound is loosely defined as vibrations which travel through the medium of air (although any medium or combination of media will suffice) as longitudinal waves and are perceived by the human ear. There are two main analytical parameters that
define the characteristic of a sound, the Sound Pressure Level (SPL) and the frequency components of the longitudinal waveform.

Similarly, there are two main characteristics which define a sound as perceived by the auditory system, pitch (measured in Bark) and perceived loudness (measured in Phon or Sone). An otologically normal person is a person who has a fully functioning auditory system, free from impairments. For such a person, the magnitude of these vibrations that can be perceived is generally accepted to be those with a SPL of greater than 20μPa or 0 dB. This is known as the Absolute Hearing Threshold (AHT). This value is actually the AHT for a signal of frequency 1 kHz. The AHT is known to vary with the frequency of signal being perceived. (Hartmann 1997, p.32)

For a similarly otologically normal person, the frequencies of vibrations which can be perceived are those within the range of 20 Hz to 20 kHz and of sufficient SPL. This detectable frequency range generally deteriorates with the age of the listener. This frequency range may also be adversely affected by overexposure to loud sounds causing hearing damage. (Hartmann 1997, p.7)

2.3.1 Pitch

Critical-Band Rate is a perceptual measure, usually quantified in Bark, of the perceived pitch of an audio signal. This measure is directly related to the frequency of the sound being perceived. The conversion from frequency to perceived pitch is often referred to as ‘frequency-warping’. The critical-band rate is a sub-division of the audible frequency range into ‘critical bands’. These ‘critical bands’ are more closely related to the manner in which the mechanics of the basilar membrane of the human inner ear operate. (Fastl & Zwicker 2007, p.158)

The conversion from frequency to pitch was originally presented by Zwicker (Zwicker, 1961) in table format. This table is presented in Appendix A as Table A.1. Zwicker's table documents the Critical-Band number along with their corresponding center frequency, maximum cut-off frequency and bandwidth. (Hartmann 1997, p.252) A plot of the relationship between Frequency and Critical Band Rate outlined by Zwicker is shown in Figure 2.4.
Since the first publication of this table in 1961 the conversion from frequency to Critical-Band Rate has been modelled using many function approximations of the data. Resulting equations and algorithms have been proposed by (Tromov, 1971), (Fourcin, 1977) and (Zwicker & Terhardt, 1980). The current, most widely used and accepted method for this conversion is outlined by Traunmuller (Traunmuller, 1990). Traunmuller’s equation for the conversion from frequency to Critical-Band Rate is

\[ z' = \frac{26.81 f}{1960 + f} - 0.53 \]

If \( z < 2 \), \( z' = z + 0.15(2 - z) \)

If \( z > 2 \), \( z' = z + 0.22(z - 20.1) \)

Else \( z = z' \) \hspace{1cm} (2.1)

where \( z \) is the critical-band rate (Bark) and \( f \) is the frequency (Hz).
2.3.2 Loudness

The Perceptual Loudness Measure is a psychoacoustic measure correlating to the physical intensity of an audio signal. Perceived Loudness is usually measured in the units Phon or Sone. As well as being sensitive to the SPL of the signal being observed, the perceived loudness of a signal is also highly dependent on the frequency components of the signal. This has led to the creation of the ‘Equal Loudness Curves’, shown in Figure 2.5 (Hartmann 1997, p.203).

![Equal Loudness Curves](image)

**Figure 2.5: The Equal Loudness Contours (I.S.O. 2003)**

The ‘Equal-Loudness Contours’ depict the sound pressure levels (SPL) which are required to ensure a perceived constant ‘loudness’ over the audible frequency band. As it can be seen from the contours of Figure 2.5, for a perceived loudness of 10 Phons at 1000 Hz an SPL of 10dB is required. To maintain a perceived loudness of 10 Phons at 50Hz an SPL of approximately 55dB is required.
The ‘Equal-Loudness’ contours were initially devised by Fletcher and Munson in 1933. The contours were derived using subjective measures, involving a panel of test subjects. Each listener was presented with a pure tone of 1 kHz of certain intensity and then a second pure tone of a different frequency. Both tones are presented to the listener via a set of headphones. The intensity of the second tone was then varied until the listener perceived the 2 tones to be of equal loudness. The intensity of the second tone was then said to be on the same ‘Equal-Loudness Contour’ as that of the 1 kHz signal. The mean of the results obtained from the various test subjects were found to obtain the final contours (Fletcher & Munson 1933).

This experiment was repeated in 1956 by Robinson and Dadson, who found their results to differ greatly from those of Fletcher and Munson (Robinson & Dadson, 1956). Robinson and Dodson’s results were accepted as the International Standardisation Organizations (I.S.O.) official standard until replaced by the current standard in 2003. The current standard is comprises the work of a large number of researchers worldwide (I.S.O. 2003).

\[ L_N = \left( 40 \cdot \log B_f \right) + 94 \]  
(2.2)

where

\[ B_f = \left[ 0.4 \times 10^{ \frac{L_f + L_U - 9}{10} } \right]^{q_f} - \left[ 0.4 \times 10^{ \frac{T_f + L_U - 9}{10} } \right]^{q_f} + 0.005135 \]

The current I.S.O. standard is documented in I.S.O. 226:2003. This document gives information on the conditions under which the subjective testing for the definition of the curves took place. The derived equations which may be used for the conversion of sound intensity data to perceptual loudness data are also included. These consist of equations for the conversion from frequency and SPL to perceptual Loudness (in Phon) and visa versa and are given here as Eq. 2.2 and Eq. 2.3 respectively. These equations are accompanied by a ‘look-up’ table which is required...
to implement these equations. This ‘look-up’ table can be found in Appendix A of this thesis labelled Table A.2 (I.S.O. 2003).

In Eq. 2.2, \( L_n \) is the perceived loudness level in Phon, \( T_f \) is the threshold of hearing, \( a_f \) is the exponent for loudness perception, \( L_u \) is a magnitude of the linear transfer function normalized at 1000 Hz and \( L_p \) is SPL. The three factors \( T_f, a_f \) and \( L_u \) each have values determined by the 29 frequencies specified in the lookup table.

\[
SPL = ((10 \cdot a_f) \cdot \log_{10} (A_f)) - L_u + 94 \quad (2.3)
\]

where,

\[
A_f = 4.47 \times 10^{-3} \times (10^{(0.025 \cdot L_n)} - 1.15) + (0.4 \times 10^{(((T_f + L_u)/10)-9)}) a_f
\]

and all symbols represent the same factors as in Eq. 2.2.

The Sone scale of perceived loudness is very similar to the Phon scale. In fact it is a direct translation of the calculated Phon value. In certain instances, the Sone scale can be a more useful measure than the Phon measure.

The Sone unit of perceived loudness is analogous to the manner in which the human auditory system perceives a change in loudness. In the Phon scale of perceived loudness, a doubling of the perceived loudness is associated with a rise of 10 Phon (Fastl & Zwicker 2007, p. 207). Using the Sone scale, the perceived loudness of two different signals would be in ratio to the resulting positions on the Sone scale. In other words, a perceived doubling of the loudness of a signal would result in a doubling of the units of the perceptual loudness measure on the Sone scale.

\[
S = \begin{cases} 
2^{\left(\frac{l}{40}\right)/10}, & \text{if } l \geq 40 \\
\left(\frac{l}{40}\right)^{2.642}, & \text{otherwise}
\end{cases} \quad (2.4)
\]
The equation for the conversion from the Phon scale to the Sone scale is shown in Eq. 2.4, where $S$ is the resulting perceived loudness in Sone and $I$ is the loudness level in Phon. (Bladon & Lindblom 1981)

2.3.3 Auditory Masking

Masking is a psycho-acoustical phenomenon which results in the non-perception of certain features of an audio signal. The masking effect can be divided into two separate types: 'Simultaneous' masking and 'Temporal' masking.

Simultaneous masking, sometimes known as frequency masking, occurs when two separate audio events occur simultaneously and one is 'masked' by the other. When masking occurs, one of the events is not perceived by the listener due to the simultaneous presence of the other. The factors which dictate whether a sound will mask another sound are the relative frequencies and amplitudes of the two signals. Figure 2.6 depicts the conditions under which simultaneous masking may occur for two specific pure-tone signals. The Masker (represented with a thick blue line) is in this case a 400Hz pure-tone signal with an SPL of 90dB. The green line represents any potential maskee. Any signal that occurs under this line will be masked by the masker. The Absolute Threshold of Hearing (ATH) is shown in red.

![Figure 2.6: Simultaneous Masking](image-url)
As can be seen from Figure 2.6, the masker must be of greater amplitude than the maskee and both signals must be relatively close on the frequency scale for masking to occur. The relative amplitudes and frequency ranges involved in the masking effect vary upon a great many factors, including signal composition, noise-like characteristics and centre frequencies. (Fastl & Zwicker 2007, p.61)

Temporal Masking is similar to simultaneous masking but is concerned with the temporal domain as opposed to the frequency domain. In the event of temporal masking, a tone will be masked by another tone if it occurs immediately before (pre-masking) or immediately after (post-masking) a tone of higher amplitude. (Fastl & Zwicker 2007, pp.82-84).

Figure 2.7 (adapted from ‘Temporal Masking’ 1997) documents the temporal domain conditions under which temporal masking will occur. As shown in this diagram, all instances of sound which occur under the line black line will be masked either by pre-masking, simultaneous masking or post-masking.

```
<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>Pre-Masking</th>
<th>Simultaneous Masking</th>
<th>Post-Masking</th>
</tr>
</thead>
<tbody>
<tr>
<td>-60</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-40</td>
<td>50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-20</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>90</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 2.7: Temporal Masking

2.4 Summary

This chapter has outlined the anatomy and operation of the human auditory system in the perception of sound. Each section of the ear and the primary auditory cortex has been described. Each section’s role in the perception of sound has also been
described. Analytical approximations of the manner in which sound is perceived are also presented. These include the measures of perceived pitch and perceived loudness. Masking effects which occur during the perception of sound are also discussed.

In the field of intrusive objective QoV estimation, the ability to model the perception of sound is vital. By using the methods described here to discern the perceived loudness and pitch, QoV estimation algorithms can accurately measure the perceived distortion present in an audio sample. For a true measure of the perceived distortion many modern QoV estimation algorithms also incorporate models of the masking effects described.
3.1 Introduction

A biological neural network is an interconnection of processing elements (neurons) responsible for the processing of information in the nervous systems of animals. Each connection between neurons has a certain strength, or 'weight', which may be strengthened or weakened. It is the strengthening and weakening of these connections that allow the neural network to 'learn' and thus perform processing operations. It is the use of neural networks that allow animals to perform various tasks with ease which have proved excessively difficult to achieve by computational means. (Freeman & Skapura 1991, p. 2)
An Artificial Neural Network (A.N.N.) is a computational method which is modeled on biological neural networks. An A.N.N. consists of an interconnection of processing elements (artificial neurons) which each carry out a simple computational operation. The neurons are interconnected by weighted connections similar to the connections in biological neural networks. The weights of each connection are updatable during the ‘training’ process. It is this ability that allows the A.N.N. to learn functions and processes in a similar manner to biological neural networks.

This chapter will present a broad overview of A.N.N.s and their origins in the biological neural network. It will begin by presenting a description of the operation of biological neural networks. The history of A.N.N. development is detailed in section 3.4 while section 3.5 outlines the operation of a simple neuron. In section 3.6 A.N.N. architectures are dealt with, followed by a summary of the training algorithms used in A.N.N. development in section 3.7. The uses of A.N.N.s are presented in Section 3.8 and the chapter is summarized in section 3.9.

3.2 Biological Neural Networks

Biological Neurons are a type of cell found in the nervous system of animals. Their main function is to process and transmit information with which they are presented. It is composed of a main cell body, an axon and a large number of dendrites. An illustration of these features is shown in Figure 3.1.

The main body of the neuron contains the nucleus where the processing of the information takes place. When the collective inputs to the neuron exceed a predefined level over a short time period the neuron will ‘fire’. This predefined level is known as the threshold of the neuron. When a neuron ‘fires’ the neuron body will emit an electrical impulse of between 70 and 100 millivolts to the Axon (Freeman & Skapura 1991, p. 9).

The Axon is a long tube-like structure through which information is transmitted from the neuron to other neurons. It is surrounded by a thin layer of insulation called

20
the Myelin Sheath which is interrupted at regular intervals by the Nodes of Ranvier. It is the polarization & depolarization of these nodes that allow the propagation of the information through the Axon. This is needed as the Axon itself is made of poorly conducting nerve fiber. At the end of the Axon are terminals which act as the connectors to the Dendrites and Axons of other neurons. (Mehrotra, Mohan & Ranka 1997, p. 8)

Dendrites are hair-like fibers that form a fine mesh around the neuron. These fibers form connections, called synapses, with other neurons. Through these synapses information from other neurons and sensory organs enter the neuron for processing.

Synapses can occur between either Dendrites and Axons, Axons and Axons or Dendrites and Dendrites. In general, a narrow gap is left between the two connectors. This gap is known as the Synaptic Cleft. Neurotransmitters diffuse across this gap, passing information from one neuron to the next. The neuron from which the information originated is called the Presynaptic cell. The receiving neuron is known as the Postsynaptic cell. (Freeman & Skapura 1991, pp. 11-12)

![Figure 3.1: The Main Features of a Biological Neuron](image)

When an impulse is created in the body of the neuron, it propagates along the Axon to the Axon Terminal. This change in potential in the Axon Terminal causes the release of the cells neurotransmitters into the synaptic cleft. The neurotransmitter then diffuses across the gap to the postsynaptic dendrites. The Dendrites of the receiving neuron will then deliver the impulse to the main body of the receiving cell for processing. The magnitude of the impulse received by the postsynaptic cell is
dependent on the ‘strength’ of the connection between the two neurons in question. It is the alteration of these ‘strengths’ that occurs during the learning process. By enhancing certain connections and deteriorating others, patterns and functions can be learned and remembered by a biological neural network. (Freeman & Skapura 1991, p. 15-16)

3.3 Artificial Neural Networks

Artificial Neural Networks (A.N.N.s) are a branch of the inductive machine learning subfield of Artificial Intelligence (A.I.) techniques. A.N.N.s are based upon the behavior, structure and architecture of biological neural networks. For this reason A.N.N.s are very suited to the modeling of biological functions which have traditionally proved to be extremely difficult for other computing methods to model.

Their advantages over traditional processing techniques include their ability to “learn” from pre-existing training material. An A.N.N. generally learns in much the same way as biological neural networks learn. When presented with training material the connection strengths within the A.N.N. are either strengthened or weakened until the desired associations are made.

Many A.N.N. architectures and training algorithms have been developed to date, each having specific advantages and disadvantages. The algorithms relevant to this thesis are described in Sections 3.6 and 3.7.

3.4 History of the Artificial Neural Network

3.4.1 The Threshold Logic Unit

Artificial Neural Networks began with the mathematical model of a neuron built by McCulloch & Pitts in 1943. The model operates as a Threshold Logic Unit (T.L.U.). The inputs to the neuron are summed and if they exceed a threshold value, the output of the neuron is logic 1. Otherwise, the output is logic 0. (McCulloch & Pitts 1943)
It wasn’t until some years later that Hebb outlined an algorithm by which A.N.N.s could ‘learn’ (Hebb 1949). He proposed that when two interconnected neurons are activated simultaneously the connection between the neurons is strengthened according to Eq. 3.1 where \( \Delta \omega \) represents the change in the weight associating the neurons, \( X \) is the input to the system, \( Y \) is the output of the system and \( \eta \) is a scaling factor used to regulate the rate at which the weights are modified.

\[
\Delta \omega = \eta \times X \times Y \quad (3.1)
\]

3.4.2 The Perceptron

The ‘Perceptron’ was developed by Frank Rosenblatt in the 1958. The Perceptron is a simple processing element which models a single neuron. A Perceptron is used to demonstrate the operation of a artificial neuron in Figure 3.2. It is mostly used as a classifier but is limited to the classification of linearly separable sets of data (Minsky & Papert, 1969). The Perceptron is trained by ‘example’ using a Hebbian-based learning algorithm. The weights are initially assigned at random and the perceptron is tested with the available training data. If a misclassification occurs the weights of the perceptron are updated according to the expression below (Eq. 3.2).

\[
w_{i,k} = w_{i,k-1} + \eta (d_i - y_i) x_i \quad (3.2)
\]

In Eq. 3.2 \( w_{i,k} \) represents the new set of weights generated, \( w_{i,k-1} \) represents the preceding set of weights, \( X_i \) is the input applied, \( d_i \) is the desired output and \( y_i \) is the actual output. \( \eta \) is the learning rate which is used to control the size of the step taken in each iteration. This process is repeated until all entries in the training set are classified correctly or until a predefined percentage of correct classification is achieved. (Fausett 1994, p. 59)
3.4.3 Adalines

Around the same time, the Adaline (Adaptive Linear Neuron) was developed by Prof. Bernard Widrow and Ted Hoff. It operates as a simple single processing unit similar to the perceptron, but differs greatly in the way it is trained. The Adaline is based around the Widrow-Hoff Least Mean Square adaptive training algorithm (Widrow & Hoff 1960).

While the Perceptron learns by attempting to reduce the number of misclassifications directly, the Adaline aims to reduce the mean square error (M.S.E.) of the classifications. This is carried out using a Gradient Decent algorithm (Eq. 3.3). A quadratic representation of the error with respect to the weights is generated. The desired change in the weight values ($\Delta W$) is then said to be a negative multiple of the change in error with respect to the change in weights ($\frac{\partial E}{\partial \omega}$).

$$\Delta w_k = \eta \left(d_i - \sum_j w_j x_j\right) x_k$$  \hspace{1cm} (3.3)

3.4.4 Madalines

The Madaline Neural Network architecture can be viewed as a collection of Adalines arranged in the form of a multi-layer neural net. It was originally developed by Widrow & Hoff (Widrow & Hoff 1960) along with the Madaline Rule 1 (MRI) algorithm for training the Madaline. The MRI algorithm only alters the weights which connect the input nodes to the hidden nodes. The connection weights of the output nodes are set when the net is created. In 1987, this learning algorithm was improved to allow alteration of the output weights of the net. The revised algorithm is called the Madaline Rule 2 (MRII). (Fausett 1994, p. 91)

3.4.5 Backward Error Propagation

The 1970’s brought the “Quiet Years” of Neural Network development. With Minsky and Papert (Minsky & Papert, 1969) proving the limitation of the perceptron,
The development of the Backpropagation of Errors optimization algorithm independently by Werbos in 1974, Parker in 1985, LeCun and Rumelhart et al. in 1986 allowed multilayered networks to be trained efficiently and reliably. This revived popular interest in the field of neural networks in the 1980's which has continued to the present day. (Fausett 1994, p. 25)

3.5 The Artificial Neuron

Figure 3.2 shows the layout of a common Artificial Neuron Model. The composite components shown are the input nodes, \(X_0...n\), the input weights, \(\omega_0...n\), the Summing Function, the Activation Function and the output node, \(Y\). All of these features are described below, with a detailed mathematic model of the Artificial Neuron being presented.

Figure 3.2: An Artificial Neuron Model
3.5.1 Inputs and Weights

The input nodes of the Artificial Neuron take the input values required by the neuron to make its ‘decision’. Depending upon the type of network being implemented and the neurons position in the network, the inputs supplied may be inputs to the A.N.N., the outputs from other neurons or a combination of both. When the inputs are applied to the input nodes, they are transported to the summing section via the weighted connections.

The weighted connections transfer the data received at the input nodes to the Summing Function. Each connection has a weight value associated with it. This value corresponds to a gain operation on the value being transmitted along it. These weighted connection model the connection strength between two biological neurons.

The connection between neurons can be both excitatory and inhibitory. The weight values in Artificial Neurons generally vary in the range -1 to +1 but are also sometimes limited to the range 0 to 1. Negative weights generate inhibitory inputs to the summing function while positive weights provide excitatory inputs. (Mehrotra, Mohan & Ranka, 1997, p. 44)

The larger the magnitude of the weight value, the greater the connection between the neurons in question. During the training process, these weights are altered to either strengthen or weaken the connection between the neuron and other neurons. This allows the A.N.N. to learn various associations and relationships between inputs and thereby produce a usable output result.

3.5.2 Summation Function

The Summing Function takes as its inputs the weighted version of the inputs to the Artificial Neuron. It then sums all of these values to discern the total ‘excitation’ received by the neuron. This excitation value may originate from either the inputs to the neural network or from other neurons with which it is associated. The result is the value $u$, which is marked on Figure 3.2. Eq. 3.4 presents a mathematical
representation of the operations required to generate $U$ from the values of the inputs, $X$, and the weight values, $\omega$.

$$U = (X_0 \times \omega_0) + (X_1 \times \omega_1) + (X_2 \times \omega_2) + \ldots + (X_n \times \omega_n) \quad (3.4)$$

3.5.3. The Activation Function and Output

The Activation Function is a function that is used to calculate the output of the neuron based on the sum of the weighted inputs to the neuron. There are many types of functions that may be used as an activation function, such as a Step function, a Sigmoid, a Radial function, a Tanh function or a Clipped Linear function. Step and Sigmoid functions are the most commonly used activation functions. Linear transforms are also commonly used, mainly within the output neurons of a network for function approximation problems.

The value resulting form the Activation Function is the output of the neuron. This output value is then either transmitted to the output of the A.N.N. or to other neurons which will use this value as an input value. This depends upon the architecture of the A.N.N. and the neurons position within that architecture.

The threshold value of zero is usually incorporated into the activation function equation. In order to modify this threshold value during the training process a constant input of magnitude 1 is often employed as one of the inputs to each neuron. This is usually assigned to the input $X_0$ of the neuron. The weight of this connection, $\omega_0$, can then be altered in the same manner as the other weights in the network by the training algorithm. This is known as a bias. The result is a highly flexible threshold value that can be altered during training. (Mehrotra, Mohan & Ranka 1997, p. 44)

3.5.3.1 Identity / Linear Function

The Identity function does not perform an operation of the values passed to it from the summing function. It simply presents the value passed to it by the summing function at the output of the neuron. This function is often used in the output neurons of an A.N.N. used as an approximation of a function. By using the Identity Function
as the activation function the output of the neuron make take on any value and is not limited to the range of -1 to 1 as with many other activation functions.

### 3.5.3.2 Step Function.

The step function, also known as the Heaviside function or the Linear Threshold function was the first activation function used in A.N.N.. When the input to the function is less than zero the output is zero. When the input is greater than zero the output is one. The Perceptron is a single artificial neuron with a Step Activation Function and is shown in Figure 3.2.

\[
y(u) = \begin{cases} 
0 & u \leq 0 \\
1 & u \geq 0 
\end{cases} \quad (3.5)
\]

Eq. 3.5 presents a mathematical representation of the step function while a graphical representation is shown in Figure 3.3. The derivative of this function is the Dirac Delta function. This results in infinity at zero, but zero everywhere else. Today, this function is rarely used as many modern training algorithms required the activation function to be differentiable. To allow an A.N.N. to approximate non-linear functions a non-linear activation function is required. (Fausett 1994, p. 17)

![Figure 3.3: The Unit Step Function](image)

### 3.5.3.3 Sigmoid Function

The Sigmoid function is a non-linear function, bounded between the values zero and one. When \( u \) is much less than zero the output will be almost zero. When \( u \) is much greater than zero the output will be almost one. When the input is zero, the output is 0.5. A graphical representation of the sigmoid function is given in Figure 3.4.
The sigmoid is generally preferred over the Step function as its derivative, \( y(1-y) \), can be easily computed. Being a non-linear function the use of a sigmoid function allows an A.N.N. to approximate a non-linear function. Eq. 3.6 gives a mathematical description of the Sigmoid function, while its derivative is given in Eq. 3.7. (Fausett 1994, p. 18)

\[
y(u) = \frac{1}{1 + e^{-u}} \quad (3.6)
\]

\[
\frac{dy}{du} = \frac{e^{-u}}{(1 + e^{-u})^2} = y(1-y) \quad (3.7)
\]

Figure 3.4: The Sigmoid Function

3.5.3.4 Radial Basis Function

A Radial Basis function operates slightly differently from the two previous functions, in that it does not use a threshold. A radial basis function calculates its output based on the distance of the input vector from a predefined vector. The direction of this distance is irrelevant to the resulting output. Only the magnitude of the distance is computed.

The most common form of radial basis function used in A.N.N.s is the Gaussian function. Gaussian functions take the form shown in Eq. 3.8 and Figure 3.5. Radial Based activation functions are often used when the function to be approximated is circular in form. (Mehrotra, Mohan & Ranka 1997, p. 14-16)
In Eq. 3.8, $y$ is the output value, $u$ is the input, $i$ is the expected value and $A$ and $\beta$ are constants. $A$ controls the magnitude of the radial function which $\beta$ control the width of the function.

### 3.6 Architectures of Artificial Neural Networks

A.N.N. architecture is the arrangement of neurons into layers and the patterns of the interconnection of those layers and the neurons within the layers. Neural networks are often separated into single layer and multi-layer architectures.

Single-layer networks such as Perceptrons and T.L.U.s, usually comprise of an input layer, a single layer of connections and an output layer. The input layer of a neural net does not perform any computation and therefore is rarely counted when determining the number of layers in a net. Single-layer nets are often used for pattern classification problems when the output of each output neuron represents a specific class of input pattern. Minsky and Papert proved that single layer neural networks can only be used effectively in problems that are linearly separable. They showed that for more complex problems more complex multi-layer nets need to be used. (Minsky & Papert 1969)
Multi-layer networks contain an input layer, any number of hidden layers and an output layer. In multi-layer networks it is common to have a layer of connections between each successive layer; however connection of any individual neuron or layer of neurons to any other is possible.

Recurrent A.N.N. architectures are those networks in which a closed loop feedback is present. Connections leading from any neuron to any other neuron in the network are permitted. This network architecture is often used to deal with problems of a sequential nature. (Mehrotra, Mohan & Ranka 1997, pp. 18-20) An example of a recurrent A.N.N. is shown in Figure 3.6. Note the connection from the uppermost hidden neuron which feeds information back to the input neuron \( X_i \) in the previous layer.

Layered network architectures allow neurons only to be connected to neurons of the same or subsequence layers. No intra layer connection is allowed within the input layer. This architecture insures that no closed loop feedback occurs in the network. Acyclic networks are a form of layered network in which no connection between neurons of the same layer is permitted. Only connections from a neuron to any neurons of a subsequent layer are permitted. A special case of the Acyclic Network
architecture is the Feed-Forward network architecture. (Mehrotra, Mohan & Ranka 1997, pp. 18-20)

Feed-Forward A.N.N.s are the most popular form of A.N.N., with the term ‘A.N.N.’ often being used to describe only Feed-Forward type networks (Mehrotra, Mohan & Ranka 1997, p. 20). In this architecture the flow of the signal is always forward through the network towards the output neurons. Connections leading from a neuron to neurons in the same or previous network layers are prohibited in this architecture. An example of a Feed-Forward A.N.N. is shown in Figure 3.7.

![Feed-Forward A.N.N.](image)

**Figure 3.7: A Feed-Forward A.N.N.**

### 3.7 Training Algorithms

The purpose of a training algorithm is to optimise a neural network so that the network will perform in the manner desired by the user. Upon creation of an A.N.N. the weight values of the network are often assigned at random. The A.N.N. must then be optimised using a training algorithm to perform a useful computational function. This is achieved by altering the weights according to a predefined set of rules (Mehrotra, Mohan & Ranka 1997, pp. 22-23).
Learning algorithms can be divided into two wide-ranging types, Supervised Learning algorithms and Unsupervised Learning algorithms.

3.7.1 Supervised Learning

Supervised Learning algorithms are very similar to function approximation algorithms. The A.N.N is provided with a training set from which to learn. Each training set consists of an input vector with a corresponding target output vector. The inputs are presented at the inputs nodes of the network and the resulting output is logged. The difference between this A.N.N. outputs and the target output vector contained in the training vector is said to be the error vector. The supervised learning algorithm then performs some form of optimisation algorithm in order to minimise this error. Depending upon the type of training algorithm being used and for what purpose the A.N.N. will be used, either the Mean Squared Error value (described in Section 5.2.2) or the number of misclassifications is minimised. This involves a measured alteration of the weights of the connections within the network.

Most training algorithms are repeated for a number of iterations until some termination criterion is met. This criterion is often a predefined number of iterations, a goal M.S.E. or number of misclassifications or a minimum reduction of error per iteration.

3.7.1.1 Backward Error Propagation

The Back-Propagation Algorithm is a supervised learning algorithm predominantly used with Feed-Forward multilayer networks. Back-Propagation uses a gradient decent method to minimize the Mean Squared Error (M.S.E.) of the A.N.N.. This algorithm operates by altering the weights of the A.N.N. to minimise the M.S.E. of the output. Each respective weight in the network is altered in proportion to the negative of the change in the error measure with respect to the change of the weight in question.

To achieve this, the Back-propagation algorithm propagates the error signal backwards from the output layer to each of the weights which contributed to the error.
Each contributing weight is altered based on the contribution it made in producing the output error. A brief overview of the algorithm is as follows:

1. Present input set to the input nodes of the network.
2. Evaluate the Error by comparing the resulting output to the desired output.
3. Calculate \( \frac{\partial E}{\partial W_{i,j}} \), the change in error with respect to each individual weight.
4. Adjust each weight based upon their influence upon the error and the gradient of the error surface.
5. Repeat 1-4 until the training algorithm termination criterion are met.


3.7.1.2 Back-Propagation with Momentum

Often a momentum weighting factor is incorporated into the Back-propagation training algorithm. Here it uses information derived during the previous iteration to determine the next movement of the algorithm. Both the current gradient of the error surface, \( G_k \), and the direction of movement during the last iteration of the algorithm, \( D_{k-1} \), are combined to generate the direction of the next step, \( D_k \). The calculation of the new direction is shown in Eq. 3.9 where \( y \) is used to scale the influence of the previous iterations direction.

\[
D_k = -G_k + yD_{k-1} \quad (3.9)
\]

This method adds a momentum factor to the direction along the error surface in which the algorithm is searching for the minimum. The momentum factor acts to remove a feature called Over-stitching. This is caused when the algorithm over-shoots a minimum error value. Often the algorithm will tend to jump over and back across the valley which contains the error minimum. The momentum factor helps to cancel-out oscillation in the algorithm and cause it to progress more smoothly and efficiently towards the error minimum. (Reed & Marks 1998, p. 166)
3.7.1.3 Newton's Method

Newton's method is a popular iterative algorithm which is used to find the zeros of a function. The algorithm can also be used to find the local maxima and minima of a function by finding the zeros of the derivative of the function.

In the field of A.N.N., Newton’s Method is used to find minima of the error function of the A.N.N.. For this to be possible the error function must be continuous and differentiable.

Newton’s method operates by approximating the error surface with a quadratic equation. The quadratic equation is generated using a Taylor series expansion truncated at the second order term. The local minimum of the error surface approximation is then found. This leads to an expression for the change in weight required to achieve this minimum. (Reed & Marks 1998, p. 170)

\[ \Delta \omega_{i,k} = -\eta H^{-1}g \] (3.10)

Eq. 3.10 presents the equation used to calculate \( \Delta \omega \), the change required for each weight for the algorithm to approach the error surface minimum. \( H \) represents the Hessian matrix composed of the second derivatives of the error function with respect to each individual weight. \( g \) is the gradient of the error surface at the current position of the algorithm, while \( \eta \) is a scaling factor to control the step size of the algorithm.

For the calculation of the change in weight (\( \Delta \omega \)) to be executed, the Hessian matrix must be computed and inverted. This is a very computationally intensive operation if the neural network is large. This has led to other 2nd order gradient decent methods which use an approximation of the Hessian matrix to perform the calculation of \( \Delta \omega \).

The initialisation of the weight values of the A.N.N. is very important to the overall accuracy and convergence of Newton's optimisation algorithm. Newton's method will only converge when the Hessian is positive definite. If the initial weights values 'guess' is very inaccurate the method will generally converge to a local
minimum as opposed to the global minimum. This will lead to a poor A.N.N. performance. (Reed & Marks 1998, p. 170)

3.7.1.4 Quasi-Newton Methods

Quasi-Newton Methods, as the name suggests, are very similar to Newton’s method. One of the main drawbacks of Newton’s method is the necessity to generate and invert the Hessian Matrix. The calculation of the Hessian proves to be very labour intensive in larger problems. Quasi-Newton methods remove this necessity by approximating the Hessian matrix. This is achieved using information from the first order gradient of the error function and is carried out by an iterative process.

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) is the most common of the Quasi-Newton methods. It bases its iterative approximation algorithm upon the previous Hessian approximation, step taken, gradient and the difference between the current and previous gradient. (Broyden 1970, Flethcer 1970, Goldfarb 1970 & Shanno 1970)

Quasi-Newton methods require the ability to store the full Hessian matrix and therefore may still prove impractical for larger problems.

3.7.1.5 Gauss-Newton Method,

The generation of the Hessian matrix required when using Newton’s optimisation method is often found to be too computationally intensive to be of practical use. The Gauss-Newton Method uses an Outer-Product approximation of the Hessian matrix to reduce the computational burden. This approximates the Hessian using mean values of the outer products of the 1st order gradient vector, \(g\).

\[
H = 2(-P + Q) \quad (3.11)
\]

where,

\[
P = (gg^T)
\]

and

\[
Q = (d - y) \frac{\partial^2 y}{\partial \omega_i \partial \omega_j}
\]
The exact calculation of the Hessian matrix can be written as Eq. 3.11, where \( d \) represents the desired output value, \( y \) represents the actual output, \( w \) represents the weight values and \( g \) represents the 1\(^{\text{st}}\) order gradient vector of the error surface.

The Outer-Product Hessian approximation assumes that when the number of training points is large and the \((d-y)\) has a zero-mean, \( Q \) will have a mean of zero and a very small variance value. This means that the \( P \) term dominates the Hessian, \( H \), and \( Q \) can be ignored. This will usually occur near a minimum of the error surface. This reduces the calculation of the Hessian to \( H = -2P \). The resulting weight update, \( \Delta w \), for the Gauss-Newton optimisation method is shown in Eq. 3.12. (Reed & Marks 1998, p. 130)

\[
\Delta w = -\eta \frac{g}{P} \quad (3.12)
\]

3.7.1.6 Levenberg-Marquardt

Both 1\(^{\text{st}}\) Order Gradient Decent and 2\(^{\text{nd}}\) Order Gradient Decent methods have their drawbacks. 1\(^{\text{st}}\) Order methods have the ability to converge towards the global error minimum from everywhere on error surface but generally progress very slowly. 2\(^{\text{nd}}\) order methods converge much more quickly but when only when near the error minimum. 2\(^{\text{nd}}\) order methods often diverge from the error minimum when a poor initial guess is made. These drawbacks resulted in demand for a training algorithm which could provide a compromise between the reliability of 1\(^{\text{st}}\) order methods with the efficiency of 2\(^{\text{nd}}\) order methods.

The Levenberg-Marquardt method employs both 1\(^{\text{st}}\) and 2\(^{\text{nd}}\) order methods to provide an efficient and reliably error minimisation algorithm. The 1\(^{\text{st}}\) order Gradient decent method is predominantly used following the initial guess. Once the 1\(^{\text{st}}\) order method converges close to the minimum of the error surface, the algorithm efficiently switches to Newton’s 2\(^{\text{nd}}\) order method. This method then completes the convergence to the minimum of the error surface.

\[
w_{k+1} = w_k - (H + \lambda I)^{-1} g \quad (3.13)
\]
Eq. 3.13 shows the algorithm used to switch between the Gradient decent method and Newton's method once the Levenberg-Marquardt algorithm approaches the error minimum. \( \omega \) represents the weight values, \( H \) represents the Hessian matrix (the second derivative of the Error function with respect to the weights), \( I \) represents the identity matrix and \( g \) represents the direction of steepest decent. Finally \( \lambda \) is the value which is used to provide the compromise between the Gradient decent method and Newton's method. This value is computed from the eigenvalues of the Hessian matrix. When the error is quite distant from the minimum, \( \lambda \) is large. This results in the combination of \( \lambda \) and \( I \) ensuring that \( (H+\lambda I) \) is positive. This in effect causes the 1\textsuperscript{st} order method to be used for the error minimisation initially. Once the error minimum is approached, the value of \( \lambda \) falls to almost 0. This efficiently switches the algorithm to Newton's method by reducing the influence of \( I \) in Eq. 3.13. (Levenberg 1944 & Marquardt 1963)

While the exact calculation of the Hessian matrix may be used in this optimisation method, the Outer-Product Approximation used by the Gauss-Newton method is often used.

3.7.2 Unsupervised Learning

While Supervised Training methods require a teacher to train an A.N.N. there is no teacher required for Unsupervised Learning algorithms. The training set supplied to facilitate Unsupervised Learning contains only the input data to be presented to the A.N.N.. There are no target values supplied for the network. During Unsupervised Learning the system attempts to detect and adapt to regularities and patterns within the training data. The types of patterns detected by the A.N.N. can often be specified by the A.N.N. architecture used.

Unsupervised Training is often favoured over Supervised Training in applications where labelled input data and associated target values are unavailable or difficult to acquire. Unsupervised Training is commonly used for a process known as 'clustering'. Here similar input vectors will trigger activation of a specific output. This is undertaken by the A.N.N. by measuring the magnitude of the differences
between the samples provided and associating a specific output with certain input patterns. (Reed & Marks 1998, pp. 11-12)

Auto-Associative networks are a subclass of Unsupervised Learning used to generate a mirror of the input at the output. This can be beneficially used as an encoder if an appropriate “bottleneck” is introduced within the A.N.N.. This is often created by employing a small number of neurons in a hidden layer of the network. This results in a scaled-down representation of the inputs to be passed through small hidden layer. The network will remove any data from the input which is not essential in reproducing the input values at output. (Morabito & Coccorese 1996)

3.7.3 Over-Fitting & Generalisation

Generalization is the ability of an A.N.N. to perform well when presented with unseen data based upon what it has learnt during the training process. One of the major problems which occur during the training of A.N.N. is the ‘over-fitting’ of training material. When ‘over-fitting’ occurs, the A.N.N. has memorised the requirements of the training data. It has exactly learned the input-output values in the training set, but performs poorly when presented with unseen data. (Bishop 1995, p.11)

![Figure 3.8: Over-Fitting](image)

Over-fitting of the training data may occur when the network has been excessively trained. If a suitable large A.N.N. is trained repeatedly until its M.S.E. with the target values is it a minimum, it may have memorised the input output relationship. This may result in a poor performance on unseen data. (Mehrotra, Mohan & Ranka 1997, p. 84)
An example of this is shown in Figure 3.8. Here it can be seen that at 1000 training epochs the M.S.E. of the A.N.N. output with respect to the target values of the training data is still decreasing, but the M.S.E. of the network when tested with the validation data has begun to rise.

Over-fitting can be avoided by limiting the amount of training iterations of the training algorithm. By dividing up the training set into a training set and a validation set the generalization of the A.N.N. can be monitored. The A.N.N. is trained with the training set. Once trained, the A.N.N. is presented with the validation set. The M.S.E. of the resulting output is monitored. This operation is known as Cross-Validation. Often many A.N.N. of varying architectures are trained to solve a single problem. By implementing the Cross-Validation training technique with each A.N.N., the A.N.N. with the best generalisation can be identified. This is often not the A.N.N. with the best performance on the training data. (Reed & Marks 1998, p. 257)

Another method to ensure better generalisation is to limit the degrees of freedom (number of adjustable parameters) of the A.N.N.. This is done by limiting the number of neurons in the A.N.N.. An optimally sized A.N.N. will be unable to memorize the data due to its lack of flexibility. In this way the A.N.N. is forced to generalise the relationship between the input and target values of the training set. (Bishop 1995, p. 11)

3.8 Implementations of Artificial Neural Networks

As outlined earlier, A.N.N.s are adept at the approximation of functions through the observation of the behaviour of that function. Therefore, A.N.N.s have been applied to many fields where a wealth of data is available to accommodate the training processes described earlier. Such areas include; Signal Processing, Pattern Recognition, Speech Generation, Data Clustering and Time Series Prediction.

It has also been noted earlier in this thesis that A.N.N.s have been shown to provide efficient modelling techniques on operations which have traditionally proved very difficult for other computer based methods. Many biological functions such as Auditory
and Vision Systems, fall into this category. This has made A.N.N.s a popular method for the modelling of such systems.

A common problem of machine vision systems is the recognition of handwritten figures. Due to the large variations present in handwriting types, figure positions, sizes and orientations, this problem has proved difficult for standard computational methods. This problem has been overcome by a number of A.N.N. architectures. A Feed-Forward A.N.N.s trained using the Backpropagation algorithm is a well known example of an A.N.N. capable of solving this problem. (Fausett 1994, p. 9)

Similar to the problem of handwritten figure recognition is the problem of speaker-independent speech recognition. Again the varying parameters of speech from speaker to speaker make recognition difficult for traditional computing methods. Again Feed-Forward A.N.N.s have proved to have the ability to overcome these problems, as have Recurrent A.N.N.s. (Fausett 1994, p. 10)

3.9 Summary and Conclusions

The purpose of this Chapter has been to present an overview of Artificial Neural Networks. The basis, history and operation of A.N.N.s have been presented along with methods used in the development, training and implementation.

It has been shown that A.N.N.s have been used effectively in the modelling of biological functions such as speech recognition. This suggests that A.N.N.s may have the ability to assist in estimating the Quality of Voice of a speech sample by identifying distortions present in a sample.

The Levenberg-Marquardt A.N.N. training algorithm has been identified as an excellent candidate for training A.N.N.s. The ability to switch between 1st order and 2nd order error minimisation methods makes the Levenberg-Marquardt algorithm an efficient and reliable method for the supervised training of A.N.N.s. Therefore, the Levenberg-Marquardt algorithm is used to train the A.N.N.s employed in this thesis.
Chapter 4

Quality of Voice Measurement and Estimation

4.1 Introduction

The measurement of speech quality is vital in the development of modern communication systems. Communication systems may suffer from many different types of distortions and delays which each affect the quality of voice (QoV) differently. This has made QoV a vital component in the overall estimation of the Quality of Service
(QoS) provided by communication systems. QoV measurement can be undertaken using two main methods which are Subjective analysis and Objective estimation.

Subjective QoV analysis methods involve a panel of listeners generating a Mean Opinion Score (MOS) for a given speech sample. Subjective methods of QoV estimation are the most accurate methods available but also have many disadvantages. They are very expensive to execute, time consuming and impractical for repeated and long term system analysis. Overall, they lack the efficiency required in the development field. With Voice over Internet Protocol (VoIP) and GSM communication systems becoming more and more widely used, a real-time and economical method for QoV analysis is required.

This has led to the development of objective methods of QoV estimation. These involve automated analysis of communication systems or speech samples representative of those communication systems. The goal of an objective QoV estimation system is to generate a QoV score equivalent to that produced by subjective means. In general, two forms of objective QoV estimation are used which are Intrusive and Non-Intrusive methods.

Intrusive methods of QoV estimation have access to both the input and output of the system under test. This consists of a recorded segment of speech to be transmitted by the system and a recording of the resulting speech which arrives at the receiving end of the system. These will henceforth be referred to the ‘Original Speech Sample’ and the ‘Degraded Speech Sample’ in this thesis.

Intrusive QoV estimation methods operate by making a comparison between the original speech sample and the degraded speech sample. A prediction of the QoV capabilities of the system is then made based upon the differences observed.

Non-intrusive methods of objective QoV estimation have access only to the operational parameters of the system under test. They predict the systems QoV performance based upon a prediction of the interaction of system parameters such as
codecs used, transmission distance and packet loss probabilities. Also, factors such as speaker gender and nationality may also be incorporated.

Intrusive objective QoV estimation methods generally prove to be more accurate than non-intrusive methods on individual system performance (Sun 2004, p. 28). Intrusive methods can be less suitable for use in the development of communication systems as a speech sample pass-through is required. For accurate results this requires a fully implemented system. Non-intrusive methods only require an outline of the systems operational parameters to predict a usable QoV score. This makes non-intrusive methods far more suitable at the development stage of communication systems.

The International Telecommunication Union (ITU) controls the standardization of communication system evaluation through their Telecommunication standardization Sector (ITU-T).

Section 4.2 of this chapter defines the term ‘Quality of Voice’ as communication system analysis. This section also outlines the main factors which degrade the QoV capabilities of a system. The methodologies behind subjective QoV estimation will be presented in Section 4.3. Section 4.4 describes some historic intrusive objective methods of QoV estimation, while section 4.5 presents PESQ, the current ITU QoV estimation algorithm standard. Section 4.6 describes non-intrusive QoV methods.

4.2 Speech Quality Impairment Factors

In the field of telecommunications, ‘Quality of Voice’ is a perceptual measure of the one-way performance of a speech based communication system. It is a rating of the tonality and clarity of the speech-audio reproduced by the system at the receiving end. This is often referred to as the ‘mouth-to-ear’ transmission quality. Some of the main factors which cause the degradation of QoV in modern communication systems are
described in this section. These include additive and subtractive distortions, packet-loss and variable delay.

It must be noted here that many factors which are present in a communication system (for instance constant delays, 'dropped calls' and acoustic echo heard by the speaker) do not affect the QoV rating of a system. These factors come under other heading such as 'Conversational Quality' and 'Quality of Service'. The QoV rating is based only on the one-way performance of the system and quantifies the listening experience of the user on the receiving end of the system. While there is a strong correlation, the QoV measure does not quantify the intelligibility of speech.

4.2.1 Additive Distortion

Additive Distortion involves the adding of a frequency component or multiple frequency components to the speech signal. This type of distortion can be attributed to such features as electrical interference, inadequacies of the compression algorithm and clipping of the signal due to overloading of the system.

When the distorted speech signal is perceived by the listener, the added frequency components are usually easily distinguished from those frequencies associated with speech by the cognitive processes of the auditory cortex. This leads to the perception of two separate 'sounds', the speech and the added frequency components. In this way, additive distortion can be easily detected by the human auditory system. It is generally regarded as quite objectionable as regards the QoV. (ITU-T 2001)

4.2.2 Subtractive Distortion

Subtractive distortion is the removal of a frequency component or frequency components from the signal during transmission. Subtractive distortion is perceived by the listener in a different manner than additive distortion. When a segment of speech which has suffered from subtractive distortion is perceived by the listener it can not be
decomposed in the same way as additive distortion. This results in subtractive distortion being perceived as having a less severe effect on the QoV than the additive variety. (ITU-T 2001)

Subtractive distortion often occurs during the implementation of a codecs. Many speech codecs apply a band-pass filter to the speech. Those frequency components of the signal deemed to be outside the range (generally 300 – 3.4kHz) vital for speech intelligibility are often discarded. This benefits the system by reducing the amount of the signal which needs to be coded and transmitted which in turn reduced the system bandwidth required. Limited system bandwidth may also result in subtractive distortion.

4.2.3. Packet-loss

In many communication systems, the continuous stream of digitised speech data is divided into packets. Each packet may generally contain from 10ms to 50ms of speech data. These packets are then transmitted individually and in sequence. These systems are referred to as packet-based systems, of which VoIP and GSM systems are most popular.

If one of these packets is lost, the result is a gap in the received speech equating to the duration represented by the packet. Packet-loss is most often caused by congestion within the network. At the systems gateway, packets may be dropped due to late arrival. If the packet arrives outside the time span set by the system buffer it is discarded. Within the network, if the router being used becomes congested, receiving more packets than it can successfully send, packet-loss will occur. Transmission errors, though uncommon in fixed networks, also cause packets to be dropped. (Clark, 2001)

Due to the transient nature of the packet-based systems, packet-loss is often ‘bursty’ in nature. When an impairment occurs within the system, it will often affect a number of successive packets before it is remedied. This results in a burst of packets being dropped. If the burst of packet-loss is of the correct duration it may result in the omission of an entire word or syllable. This may not result in the degradation of the QoV score as the
listener will not be aware that the word or syllable ever existed. Also, packet-loss during silence is imperceptible.

Due to its ‘bursty’ nature, Packet-Loss is often described using two parameters; Unconditional Loss Probability (ULP) and Conditional Loss Probability (CLP). ULP is the overall probability that any individual packet will be lost. CLP is the probability that a specific packet will be lost based upon whether the previous packet was lost. By combining these two parameters a useful model of packet-loss within a communication system can be generated. (Sanneck 2000, p. 24)

4.2.4 Jitter

Delay variation or ‘Jitter’ in a communication system refers to the variation of the transmission delay experienced by the speech signal. While the presence of a constant a delay in a communication network has very little effect on the QoV, Jitter plays a major role. Jitter in a speech signal is caused by variations in the delay of the different portions of the signal. It is very prominent in VoIP based systems. In these systems queuing delays commonly occur during transmission. Re-routing of the signal path during periods of continuous transmission is also a common feature of VoIP systems. This results in the constant variation of the propagation distance and hence the propagation delay encountered by the information being transmitted. (Sanneck 2000, p. 22 )

The use of Jitter Buffers in more modern communication networks has proved to almost eliminate Jitter. Jitter Buffers operate by buffering the incoming packets and releasing them at a constant rate. Jitter buffers discard any packet that arrives at the buffer outside a specified time frame. In this way Jitter is replaced by a constant delay and packet-loss. (Clark 2001)
4.3 Subjective Methods of QoV Measurement

Originally, the QoV over a communication system was analysed by subjective means. Subjective methods of QoV measurement involve a panel of multiple listeners rating the quality of a speech sample. Each listener of the panel must be considered to be otologically normal, that is their auditory system should be in full working order and encompass the expected human auditory range of 20 Hz to 20 kHz. Each listener is then presented with a speech sample or a number of speech samples that are representative of the system under test. These speech samples should also follow the guidelines set out by ITU-T P.800, according to which rating system is in use in the test. Depending upon the rating system in use, the listener may or may not be provided with the original speech sample as well as the degraded sample. (ITU-T 1996(b))

The rating of speech samples can be carried out using one of three main QoV rating systems which are Absolute Category Rating (ACR), Comparison Category Rating (CCR) and Degraded Category Rating (DCR). Generally, the mean value of the opinion scores generated by the subjective listening tests is then calculated. This gives a more accurate overview of the actual QoV capabilities of the system. This mean value is known as the Mean Opinion Score (MOS).

4.3.1 Absolute Category Rating (ACR)

ACR is the most popular subjective method and is the method recommended by the Telecommunications Standardization Sector of the International Telecommunication Union’s (ITU-T) in recommendation ITU-T P.800 ‘Methods for Subjective Determination of Transmission Quality’. ITU-T P.800 also recommends that the subjects rate the degraded speech sample according to the opinion scale shown in Table 4.1.

It is stated in P.800 that subjects should not listen to the original speech sample. Only the degraded speech sample should be presented to the listener for ACR evaluation. P.800 also goes into great detail in specifying the manner in which the listening tests
should be carried out, specifying the method of recording the samples, the environment in which they should be carried out and lists requirement for the listeners involved in the testing. (ITU-T 1996(b))

<table>
<thead>
<tr>
<th>QUALITY OF SPEECH</th>
<th>RATING</th>
<th>SUBJECTIVE PERCEIVED DISTORTION LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsatisfactory</td>
<td>1</td>
<td>Very annoying and objectionable.</td>
</tr>
<tr>
<td>Poor</td>
<td>2</td>
<td>Annoying but not objectionable.</td>
</tr>
<tr>
<td>Fair</td>
<td>3</td>
<td>Perceptible and slightly annoying.</td>
</tr>
<tr>
<td>Good</td>
<td>4</td>
<td>Barely Perceptible, but not a hindrance.</td>
</tr>
<tr>
<td>Excellent</td>
<td>5</td>
<td>Imperceptible.</td>
</tr>
</tbody>
</table>

Table 4.1: ACR Opinion Scale

4.3.2 Degraded category Rating (DCR)

The Degraded Category Rating (DCR) is also outlined in ITU-T P.800 and provides the listener with two speech samples, the original and the degraded speech samples. The listener then rates the level of degradation caused by the system according to the scale outlined in Table 4.2. The mean of the resulting DCR scores is known as the Degraded Mean Opinion Score (DMOS).

<table>
<thead>
<tr>
<th>RATING</th>
<th>Degradation</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Degradation is inaudible</td>
</tr>
<tr>
<td>4</td>
<td>Degradation is audible but not annoying</td>
</tr>
<tr>
<td>3</td>
<td>Degradation is slightly annoying</td>
</tr>
<tr>
<td>2</td>
<td>Degradation is annoying</td>
</tr>
<tr>
<td>1</td>
<td>Degradation is very annoying</td>
</tr>
</tbody>
</table>

Table 4.2: DCR Scale of degradation
DCR is often used to distinguish between good quality signals as the sensitivity of the ACR measure tends to decrease as the quality of the speech samples increases. (ITU-T 1996(b))

4.3.3 Comparison Category Rating (CCR)

Comparison Category Rating (CCR) is the third subjective QoV measurement method outlined by ITU-T P.800. Like DCR, CCR specifies that the subjects listen to both the original and degraded speech samples. Unlike DCR, the order of the degraded and non-degraded samples is chosen at random and the listener is not aware of the order in which they are played. The listener then rates the quality of the second sample compared to the quality of the first, according to the scale shown in Table 4.3. The scores generated are then averaged to produce the Comparative Mean Opinion Score (CMOS).

<table>
<thead>
<tr>
<th>3</th>
<th>Much Better</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Better</td>
</tr>
<tr>
<td>1</td>
<td>Slightly Better</td>
</tr>
<tr>
<td>0</td>
<td>About the Same</td>
</tr>
<tr>
<td>-1</td>
<td>Slightly Worse</td>
</tr>
<tr>
<td>-2</td>
<td>Worse</td>
</tr>
<tr>
<td>-3</td>
<td>Much Worse</td>
</tr>
</tbody>
</table>

Table 4.3: CCR Scale

This method is especially useful in situations where the input was affected by background noise. In such situations it may be possible that the transmission of the input over the system under test may have improved the QoV. (ITU-T 1996(b))
4.4 Intrusive Objective QoV Estimation Methods

The main objective of intrusive objective QoV estimation methods is to generate a QoV score which corresponds to that which would be generated by subjective testing methods. As mentioned earlier, these QoV estimation methods are provided with both the original and degraded speech sample. Taking these speech samples as input, the algorithm then attempts to estimate the QoV score of the system used to generate the samples. This QoV score is a prediction of the MOS QoV score that would be generated by a subjective means using one or both of these speech samples.

This method of QoV estimation was initially undertaken by deriving such features as the signal to noise ratio (SNR), frequency response and system delays from the available speech samples. These methods were found to give poor predictions of the actual QoV capabilities of modern communication systems [Sun 2004, p.33]. More modern intrusive objective QoV estimation methods predict QoV scores by creating perceptual representations of the available speech samples. This is done by carrying out auditory transforms of the samples which mimic the perception of sound by the Human Auditory System. These transforms have been described in detail in Section 2.3.

These speech samples may also undergo level equalisation and delay compensation. A distance measure is then performed on the transformed signals to locate instances of distortions in the samples. Using the distance measure along with any other comparative measures which may have been made, a QoV score prediction is then generated. In the prediction of the QoV score, cognitive measures which model the perception of distortions are also used. A block diagram showing the basic features of an Intrusive QoV estimation algorithm is shown in Figure 3.1.

QoV estimation algorithms which incorporate perceptual modelling include, the Bark Spectral Distortion (BSD) measure (Wang, Sekey & Gersho 1992), Measuring Normalising Blocks (MNB) (Voran 1999), Perceptual Speech Quality Measure (PSQM)
(Beerends & Steimerdink 1994) and Perceptual Analysis/Measurement System (PAMS) (Rix Reynolds & Hollier 1999).

![Diagram of A Modern Intrusive Objective QoV Estimation Algorithm](image)

**Figure 4.1: A Modern Intrusive Objective QoV Estimation Algorithm**

At present, the ITU-T has not standardised any method which incorporates the use of A.N.N.s in the field of intrusive objective QoV estimation. The ITU-R, the ITU Radio Communications Sector, has standardised a method of the estimation of audio quality over radio communication systems which incorporates the use of A.N.N.s. Here the A.N.N. is used to account for the cognitive operations of the human auditory system. (Sun 2004, p.42)

Methods have also been presented that used A.N.N.s to achieve the perceptual mapping functions of an intrusive objective QoV estimation algorithm. (Meky & Saadawi 1997)

**4.4.1 Bark Spectral Distortion (BSD)**

Bark Spectral Distortion, presented in 1992, was one of the first QoV estimation algorithms to incorporate perceptual measures. BSD generates a series of power spectral vectors to represent each 10ms section of both the original and degraded speech samples.
The power spectral densities are then mapped to a perceptual loudness magnitude for each of 15 critical frequency bands.

The Bark Spectral Distortion is computed by finding the mean squared Euclidean distance between the corresponding pitch-power densities of the original and degraded samples. The BSD measure is then translated into a measure which corresponds to the more widely used MOS speech quality rating (Wang, Sekey & Gersho 1992).

The Modified Bark Spectral Distortion (MBSD) algorithm is an advancement of the previously established BSD algorithm. This new algorithm employs a cognitive model which allows for the effects of simultaneous masking. Any distortion which is detected, but lies below a predefined masking threshold will be discarded. Therefore it will not be used in the calculation of the bark spectral distortion. (Yang, Dixon & Yantorno 1997)

Enhanced Modified Bark Spectral Distortion (EMBSD) is yet another advancement on the BSD measure. Building upon the cognitive model used in MBSD, EMBSD incorporates factors to model the effects of post-masking. The effects of post-masking have been described in Section 2.3.3. With this in place, any distortion that would not be perceived due to the effects of post-masking will not be incorporated into the calculation of the predicted QoV score. (Yang 1999)

4.4.2 Measuring Normalised Blocks (MNB)

While many other methods create a distance measure based on a frequency domain representation of the temporal speech data, the MNB uses both ‘time measuring normalising blocks’ (TMNB) and ‘frequency measuring normalising blocks’ (FMNB). A TMNB identifies the discrepancies on the frequency scale over a given time interval. Distortions are identified by integrating the frequency scale of both the original and degraded speech samples over a time interval and comparing the results. Similarly, a FMNB identifies discrepancies in the temporal field by integrating over time intervals for a given frequency scale.
When these measured normalised blocks are then weighted and summed they create an estimation of the ‘Auditory Distance’ (AD) of the speech samples presented. The AD measure is then fed into a sigmoid type function which generates a value ranging from zero to one. This result, referred to as L(AD), has a high rate of correlation with the MOS speech quality rating given by subjective means. (Voran 1999)

4.4.3 Perceptual Speech Quality Measure (PSQM)

The Perceptual Speech Quality Measure (PSQM) was created in PTT Research in the Netherlands in 1994. It was adopted as the ITU-T standard for intrusive objective QoV estimation of a speech based communication system from 1996 until its replacement by PESQ in 2001. (ITU-T 1996(d))

PSQM operates by computing the difference between the representative versions of the original and the degraded speech samples. Both samples are split into frames of 32ms duration (256 samples each for 8 kHz sample-rate) and each frame is then mapped to the perceptual loudness domain.

The amplitudes of the components of the degraded sample is scaled using a non-linear scaling factor. This factor is generated through comparison of the loudness levels of the degraded sample with the original speech sample. This scaling factor compensates for short term gain variations in the system under test.

Through comparison of the loudness components of the original and the scaled degraded speech samples, PSQM then computes the Noise Disturbance (ND) vector. The mean of the ND vector over each frame is then computed. This gives an estimation of the overall distortion present in the degraded speech sample. This mean value is then mapped to a MOS in the range 1 to 5 (Beerends & Stemerdink 1994).
PSQM is quoted as having a correlation with subjective results of between 97% and 99% (Beerends & Stemerdink 1994). But it was found to perform poorly when presented with speech samples which had been degraded by transmission through a packet-based system such as VoIP. To compensate PSQM+ was devised. PSQM+ had much the same cognitive measures as PSQM, but an extra scaling factor was inserted which ensures that the distortion level report is in proportion to the level of temporal clipping present in the sample. (ITU-T 1997)

4.4.4 Perceptual Analysis/Measurement System (PAMS)

With the growing popularity of packet-based communication system such as GSM and VoIP has come a greater need for QoV estimation tools to meet there requirements. The most prominent feature of packet-based networks from a QoV perspective is the presence of variations in the transmission delay times from packet to packet. This is caused by features of the network architecture such as variable buffering, re-routing of transmission path and queuing delays. It is common that the delay time may be varying during instances of speech causing Jitter. This has a large effect on the QoV capabilities of the system. PAMS was the first intrusive objective quality of speech estimation algorithm to include variable delay compensation in the estimation of the QoV score.

Before applying the perceptual mapping and cognitive processes present like many of its predecessors, PAMS carries out a histogram-based variable time delay estimation and compensation. This is described in greater detail in Section 4.4.5 as the same system is used in the PESQ QoV estimation algorithm. (Rix & Hollier 2000)

Gain equalisation is also carried out on each 32ms frame. The compensated 32ms frame vectors are then mapped to the perceptual domains of Pitch and Loudness. The distortion measure between the original and the degraded speech samples is then calculated. Cognitive features are incorporated into this measure to allow for auditory masking. The resulting perceptual distance measure is then mapped to a QoV score which
4.4.5 Radial Basis A.N.N. Intrusive QoV Estimation

An intrusive objective method for the prediction of the QoV capabilities of a communication system which incorporates the use of A.N.N.s was presented by Meky and Saadawi in 1997. (Meky & Saadawi 1997) The method uses a Radial Basis A.N.N. to account for the cognitive features of human auditory system in the estimation of a QoV score.

This QoV estimation algorithm operates in much the same manner in the pre-processing stage as many of those already described. Framed perceptual representations of equalised versions of the original and degraded speech samples are created in the Pitch-Loudness domain. One major difference is the use of a fixed number of frames, which is independent of the duration of the speech samples. This number is set to 211 overlapping frames each with durations in the range 20ms to 30ms. (Meky & Saadawi 1997)

These perceptual Power Spectral Density arrays are they used to generate Linear Prediction Filter Coefficients (LPC) representative of each frame of each the speech sample. These coefficients are then used recursively to compute the perceptual Cepstral Coefficients of each of the 211 frames of each sample. Through a comparison of the Cepstral Coefficients of each speech sample the Cepstral Distance vector is computed. An A.N.N. is then used to map the Cepstral Distance vector to an estimation of the QoV score. (Meky & Saadawi 1997)

The A.N.N. was then trained in a supervised manner with training data generated using the algorithm outlined above. Each training vector consisted of an input vector of 211 values. This number corresponds to the fixed number of frames used pre-processing of each speech sample. The target output of each training vector is a subjectively derived MOS score generated by subjective means. (Rix, Reynolds & Hollier 1999)
QoV MOS score. 240 training vectors were prepared using 24 speech files for each of 10 different codecs.

![Diagram](attachment:image.png)

**Figure 4.2: Radial Basis A.N.N. based Intrusive Objective QoV Estimation**

Testing of this system was performed by the use of speech samples coded with “16 other coders” The resulting QoV predictions of this intrusive objective QoV estimation algorithm proved to be very accurate, with a maximum error of 0.33 over a range of 1 – 5 correspond to the ACR subjective MOS rating. (Meky & Saadawi 1997)
4.5 Perceptual Evaluation of Speech Quality (PESQ)

PESQ is the current intrusive objective QoV estimation algorithm recommended by the ITU-T. Its capabilities and an outline of the algorithm are presented in ITU-T P.862 (ITU-T 2001).

Being an intrusive objective QoV estimation algorithm, PESQ takes as input an original and a degraded speech sample. The output is an estimation of the QoV score in the range of -0.5-4.5. In the majority of cases the score will be in the range 1-4.5, which represents a prediction of the QoV score that would be generated if subjective testing was used. Figure 4.3 shows a block diagram of the main steps in the PESQ algorithm. A detailed description of the operation of each of the blocks is presented in the following subsections. (ITU-T 2001)

Figure 4.3: The PESQ QoV Estimation Algorithm

Throughout the PESQ algorithm, a number of constants are introduced. While the implementation and usefulness of these constants is clearly described in ITU-T 2001 and certain other publications referenced here, the exact origin is often omitted from these documents.
4.5.1. Level Align and Equalisation

The overall gain of the system is then determined by comparing the power densities of the filtered versions of the two speech samples. Both samples are then scaled to have a listening level of 79dB. This corresponds to the 'ear reference point' set out by ITU-T P.830.

As it is assumed listening tests are carried out using a handset similar to a telephone handset, an Intermediate Reference System (IRS) receive characteristic is applied to each speech sample. The IRS receive characteristic models the frequency response of a standard telephone handset. The characteristic is set out by ITU-T recommendation P.830. (ITU-T 1996(c). This filter is applied to a frequency domain representation of each speech sample. The samples are then return to the time domain with an inverse FFT. (ITU-T 2001)

4.5.2 Short Term FFT

To model the time-frequency mapping performed by the cochlea of the human ear, a short term Fast Fourier Transform (FFT) is performed on both samples. The frame size used in PESQ is 32ms with a 50% overlap between frames. Each frame is filtered through a hamming window in the temporal domain before time-frequency mapping.

The result of this operation is the creation of an array of 32ms blocks for both the original and degraded speech samples. Each block is a frequency-power density representation of a specific 32ms frame of the speech sample.
4.5.3 Delay Estimation and Compensation

The PESQ algorithm incorporates a delay estimation and compensation algorithm which accounts for both constant and variable delay. This is required as the later steps of the algorithm assume that the degraded signal is perfectly aligned with the original signal in the time domain.

Unlike many of its predecessors, PESQ allows for variable delay in the system under test. A cross-correlation method of delay estimation is used to eliminate any crude delay caused by the system. A subsequent cross-correlation/histogram based method is then implemented to account for variable delays in the system.

The delay compensation is carried out by initially identifying the instances of speech in the samples provided. Instance of speech, known as an utterances, are located using a Voice Activity Detector (VAD). An utterance is defined as an instance of speech of 300ms duration containing no period of silence greater than 200ms. The constant delay observed between the utterances of the original and degraded system is identified first. This is carried out using a crude envelope-based delay estimation procedure incorporating a fine time-alignment histogram-based method. Once the constant delay has been identified and removed, a technique called ‘Utterance Splitting’ is used to account for any variations in delay which have occurred during an utterance interval. (Rix et al. 2002)
Utterance Splitting involves the bisection of individual utterances and re-alignment of the two resulting utterance segments with corresponding segments in the other speech sample. If this re-alignment results in a better correlation value, then a significant delay has occurred during the utterance. This procedure is repeated each time a delay variation is identified to check for subsequent variations. (ITU-T 2001) This procedure is presented in block diagram format in Figure 4.4.

4.5.4 Frequency Warping

The frequency scale of each 32ms frame is warped to the perceived pitch (Bark) scale using approximations of Zwickers Critical Bands. This transform is outlined in Chapter 2 Section 2.3.1. The mapped values are then binned into their associated Bark Band and summed to give an array of pitch power densities. (Beerends et al. 2002)

4.5.4.1 Compensation for System Transfer Function

A compensation factor is generated for each bark band to counteract the effects of filtering in the system under test. This is carried out by comparing the ratio of the power spectral densities of the original and degraded signal. From this comparison a compensation factor is generated for each Bark Band. The magnitude of this factor is limited to 20dB. The compensation factor vector is then applied to the original signal to create an ‘inversely filtered pitch power density’. (Beerends et al. 2002)

4.5.4.2 Compensation for Short-term gain variations

Short-term gain variations are accounted for by comparing the pitch power densities in much the same manner as the previous step. In this instance it is carried out frame by frame as opposed to for the entire speech samples. This is to account for short-term variations in the system transfer function. The appropriate alterations are derived based upon the ratios of the pitch power densities of the original and degraded speech samples. The alterations are then applied to the degraded speech sample, but are bounded to the range 0.0003 to 5. (ITU-T 2001)
4.5.5 Loudness Density Mapping

The power density for each Bark Band of each frame is mapped to the perceptual loudness domain (Sone scale). This is done using the algorithm outlined in I.S.O. 2003:226. (ISO 2003) The result is 2 arrays of 32ms frames representing the pitch-loudness densities for each 32ms frames of the speech samples. (Rix et al. 2002)

4.5.6 Disturbance Density Computation

The raw disturbance density array is then computed. This is the signed distance between the original and degraded representative speech samples. The array is populated by computing the distance between the loudness densities of the corresponding time-frequency cells in the representations of the speech samples. A positive disturbance density represents additive distortion. A negative disturbance density represents a speech component being lost by the system under test, known as subtractive distortion. (ITU-T 2001)

4.5.7 Cognitive Modelling
degraded loudness densities. This is computed for each time-frequency cell (Bark Band) and then multiplied by 0.25. Masking is then modelled using the following algorithm:

\[
\text{If } \text{raw} > \text{mask value}, \\
\text{Raw} = \text{Raw} - \text{Mask} \\
\text{Elseif } |\text{raw}| < \text{mask}_\text{value} \\
\text{Raw} = 0 \\
\text{Elseif } \text{Raw} < -\text{mask}_\text{value} \\
\text{Raw} = \text{Raw} + \text{mask}
\]

The result of this algorithm is the creation of a dead zone in the perception of distortion. If the raw distortion value is of a lower magnitude than the masking value then it is said that the distortion is imperceptible by the human auditory system. Therefore the raw value is set to zero. (ITU-T 2001)

### 4.5.7.2 Asymmetry Factor

Psycho-acoustically, subtractive distortion is less intrusive to the QoV than additive distortion. To account for this an Asymmetry Factor (AF) is computed. The Asymmetry Factor is equal to the ratio of the pitch power densities of the degraded sample to the original sample raised to the power of 1.2.

\[
AF = |\text{Max}(\text{PPD}_{\text{orig}}, \text{PPD}_{\text{deg}})|^{1.2}
\]

\[
\text{If } AF < 3 \\
AF = 0 \\
\text{Elseif } AF > 12 \\
AF = 12
\]

The asymmetrically weighted disturbance (DA) is then created by multiplying the Raw Disturbance (D) value by AF. This results in only additive distortion being measured by this factor. (Rix et al. 2002)
4.5.7.3 Aggregate of Disturbance Densities over the Frequency Scale for each Frame

The disturbance density and the asymmetrical disturbance density are aggregated over the frequency scale to create the Frame Disturbance, $FD$, and the Asymmetrical Frame Disturbance, $AFD$. The calculation of these values involves a scaling factor to allow variations of the width of the Bark Bands.

Special emphasis is given to frames with a low power density in the original speech sample. This is performed by adding a constant to the power of the original frame and raising the result to the power of $-0.004$. If the speech samples are greater than 16s in duration another scaling factor is introduced to reduce emphasis from distortions which occur towards the beginning of the speech sample. This factor is used to model the effect of short-term memory in subjective listening tests. The Frame Disturbance value is limited to 45. (ITU-T 2001)

4.5.7.4 Re-alignment of ‘Bad Intervals’

Once transformed to the psycho-acoustical domain misalignment of the delay estimation and compensation algorithm become apparent. Misaligned frames often generate extremely high Frame Distortion values. When the Frame Disturbance value is found to be greater than 30, a frame alignment algorithm is said to have failed and a ‘bad frame’ has occurred. A bad interval is said to have occurred if two bad frames are detected within 4 frames of each other. (Rix et al. 2002)

When a bad interval is identified it needs to be re-aligned in order to achieve a more accurate QoV estimation from the algorithm. The new delay value is found by finding the absolute maximum cross-correlation between the original and degraded signals and the frame is re-aligned according to this re-evaluated delay. If this new alignment results in a small measure of Frame Disturbance, this new measure replaces the old measure. (Rix et al. 2001)
4.5.7.5 Aggregation of Disturbance values and PESQ Score Calculation

The aggregation of the Frame Disturbance and the Asymmetrical Frame Disturbance vectors over the duration of the speech samples is carried out in two phases.

Firstly Split-Second aggregation is carried out. The Frame Disturbance and the Asymmetrical Frame Disturbance vectors are each aggregated over intervals of 20 frames. Each 20 frame interval overlaps 50% with its surrounding frames. Then the split-second aggregated Frame Disturbance and the Asymmetrical Frame Disturbance vectors are aggregated over the duration of active speech in the samples. (ITU-T 2001)

Each aggregation is performed using a normalising equation, shown here as Eq. 4.1. (Rix et al. 2001) The factor $p$ in Eq. 4.1 is incorporated to account for the varying relevance of distortion in the Split-Second aggregation compared with whole sample aggregation. For Split-Second aggregation, $p$ is set to 6, while for whole sample aggregation $p$ is set to 2. This gives increased emphasis to distortions occurring over a Split-Second interval. (ITU-T 2001)

The PESQ score is calculated using the average Frame Disturbance value and average Asymmetrical Frame Disturbance value. These factors are combined using the equation shown as Eq. 4.2. (Rix et al. 2001)

\[
PESQ\text{MOS} = 4.5 - (0.1 \times F\text{D}) - (0.309 \times A\text{FD})
\] (4.2)

The result, PESQ\text{MOS}, is bound to the range of -0.5 to 4.5 but is generally in the range 1 to 4.5. (ITU-T 2001)
4.6 Non-Intrusive Objective Methods of QoV Estimation

Non-intrusive objective QoV estimation methods attempt to estimate the QoV capabilities of a system without needing access to speech samples representative of the system in operation. The QoV estimation is generated using system parameters such as components used in the system, signal to noise ratio (SNR), frequency response and packet-loss probabilities.

While generally not being as successful as intrusive QoV estimation methods (Sun 2004, p33), non-intrusive methods have one main advantage. They can be implemented at the design stage, since a speech sample which has been passed through the system is not required.

The most widely used form of non-intrusive QoV estimation is the 'E-model', which is outlined in ITU-T standard G.107 which defines it as "a computational model for use in transmission planning". Other methods of non-intrusive objective QoV estimation involving A.N.N.s have also being developed but none have been standardised by the ITU to date [Mohamed, Cervantes & Afifi 2000 and Sun 2003].

4.6.1 The E-model

The E-model is a computational model used for QoV estimation. The E-model can be used as a planning tool to estimate the QoV over a communication system (specifically 3.1 kHz telephony) at the design stage. The QoV estimation is achieved by combining the effects of the various components of the system and the system parameters. The output of the system is the transmission rating factor, $R$, which is in the range 0 to 100. The equation used for the computation of the $R$ factor is given in Eq. 4.3. This $R$ factor can then be mapped to an associated QoV MOS style score. (ITU, 2003)

$$R = Ro - Is - Id - (Ie - eff) + A \quad (4.3)$$
**Ro** is the measure of the SNR present in the system. This takes into account noise sources such as circuit noise and noise from the environment in which the system is operating.

**Is** is the simultaneous impairments which occur during transmission of an audio signal. These include quantisation distortion, un-optimised side-tones and poor system gain.

**Id** is the delay impairment factor. This accounts for all delays which are caused by the system, such as constant system delay and talker echo.

**le-eff** is the equipment impairment factor which takes account of the effects of the equipment used in the network such as codecs. This factor has been updated from previous 'E-models' to include systems where the event of packet loss needs to be accounted for.

**A** is the advantage or expectation factor. This is based on the expected performance to which the system is designed and expected to achieve.

As can be seen from Eq. 4.3, the E-model assumes that the factors which degrade QoV over a communication network are additive in nature. ITU-T G107 states that “This feature has not been checked to a satisfying extent”. (ITU-T 2003)

The resulting **R** factor can be converted to a MOS score using the algorithm shown below as Eq. 4.4. This algorithm was derived from the guidelines in Table 4.4.

<table>
<thead>
<tr>
<th>R-Value (Lower Limit)</th>
<th>MOS</th>
<th>User Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>4.34</td>
<td>Very Satisfied</td>
</tr>
<tr>
<td>80</td>
<td>4.03</td>
<td>Satisfied</td>
</tr>
<tr>
<td>70</td>
<td>3.60</td>
<td>Some Users Dissatisfied</td>
</tr>
<tr>
<td>60</td>
<td>3.10</td>
<td>Many Users dissatisfied</td>
</tr>
<tr>
<td>50</td>
<td>2.58</td>
<td>Nearly All Users Dissatisfied</td>
</tr>
</tbody>
</table>

Table 4.4: Provisional Guidelines for R-value to QoV Score Mapping
For $R < 0: MOS = 1$

For $0 < R < 100: MOS = 1 + 0.035R + R(R - 60)(100 - R)7 \cdot 10^{-6}$ \quad (4.4)

For $R > 100: MOS = 4.5$

While the E-model is very useful for the planning of networks, it also has significant draw-backs. It can only be used in predefined network conditions for which model parameters has been deduced from subjective testing. For example, a newly developed codec will not have an associated le-eff parameter. This would mean that the system could not be tested using the E-model or the codec must be approximated by a similar pre-existing codec.

4.6.2 A.N.N. Based Non-Intrusive Objective QoV Estimation

A non-intrusive QoV estimation method incorporating the use of A.N.N.s was put forward in a Ph.D. thesis by L. Sun. in 2004. The A.N.N. took as its inputs parameters; the conditional and unconditional packet loss probabilities, Codec type used and speaker gender. From these inputs the A.N.N. can generate an estimation of the QoV score of the system under test. This QoV estimation methodology was designed to work with packet based communication system. (Sun 2004)

The A.N.N. based QoV estimation was based on the results generated by PESQ when presented with pre-prepared speech samples. Due to the logistical issues involved in the creation of sufficient speech samples from actual network implementations, a network simulator was developed. The simulator developed incorporates a codec implementation and a packet-loss simulator. These two factors were deemed to be the predominant causes of QoV degradation within a packet-based communication system.

The system simulation system began by using a specified (packet based) speech codec to encode the speech data.
A Gilbert model was used to simulate the packet loss commonly experienced during transmission over a packet-based system. This model is required as both conditional and unconditional packet loss probabilities are features of the network. In a Gilbert model the packet loss probabilities can be semi-specified. Due to the nature of the model the specified probability values are not generated exactly in the resulting speech sample. The probability value entered into the model is called a 'seed'. The resulting packet loss percentage is generally quite close to the seed value when a large quantity of packets is involved. The actual resulting packet loss must be measured to give accurate results. The actual figure used in this A.N.N. implementation is not the overall packet loss percentage but the percentage of packet loss which occurs during speech. This measure was found by Sun to give more accurate QoV estimation results.

The codec speech packet files resulting from the packet-loss simulator were decoded using the appropriate decoder.

Speech samples were generated for three codec types; AMR (4.75Kb/s), G.729 and G.723.1 (6.3Kb/s). Seeds of 0, 10, 20, 30 and 40% for unconditional loss probabilities and 10, 50 and 90% for conditional loss probabilities were entered into the Gilbert model to simulate different network conditions. Both a male and a female speaker were used to generate the original speech samples. (Sun 2004, p. 97)

The original and degraded speech samples were then presented to the PESQ QoV estimation algorithm which generated a QoV score.

The A.N.N. was trained in a supervised manner with the speaker gender, Packet loss probabilities and codec type as inputs of the training vector. The QoV score generated by PESQ which corresponds to these input parameters was used as the target value for each training vector. This novel method of using a standardised intrusive QoV estimation algorithm to generate training data for the creation of non-intrusive methods was established by Sun. (Sun 2004, p.74)
A single hidden layer Feed-Forward A.N.N. containing four input nodes, five hidden nodes and a single output node, as shown in Figure 4.6, was used. A standard Back-Propagation algorithm was used to train the A.N.N.. Sun generated a total of 362 training vectors (each consisting of four input values and a corresponding QoV score output value). 253 of these vectors were used in the training set for the A.N.N.. The remaining 109 were used as a validation set in a Cross-validation based testing process. (Sun 2004, p. 95)

After training, the A.N.N. achieved a correlation coefficient of 0.9667 with the target values presented with the training data, with an average error of 0.12. A correlation coefficient of 0.952 and an average error of 0.15 were achieved with the validation set. (Sun 2004, p. 99)
4.7 Summary

This chapter has presented a background to Quality of Voice estimation in communication systems. The motivation behind QoV estimation has been outlined along with the factors that cause QoV impairments in speech based systems. The two main areas of QoV estimation, subjective and objective estimation, have also been presented. A selection of methods and algorithms used for QoV estimation in each area are also described.

It has been established that the application of A.N.N.s to the field of objective QoV estimation is a viable concept. A non-intrusive method of QoV estimation through machine learning will be developed modelled upon the system developed by L. Sun. An intrusive method of QoV estimation through machine learning will also be developed modelled upon the current ITU-T standard, PESQ.
Chapter 5

Modeling of the Human Auditory System Through Machine Learning

5.1 Introduction

The modeling of the perception of sound by the human auditory system is vital in fields such as speech recognition, speech synthesis and audio quality measurement. In QoV estimation algorithms, such a model can be used to assess the effects of distortions upon the QoV capabilities of the system under test. As discussed in Chapter 2, the perception of sound is governed by two main perceptual measures; the Perceptual Loudness measure (Phon or Sone Scale) and the Perceived Pitch (Critical-Band Rate or Bark Scale) of an audio signal.
The perceived loudness of an audio signal presented to the ear is influenced by both the signal's frequency and sound pressure level (S.P.L). The current method for the conversion from frequency and S.P.L. to perceptual loudness is outlined in ISO 226:2003 – ‘Acoustics – Normal equal-loudness-level-contours’ (ISO 2003). This conversion involves a complex calculation which incorporates three 29 entry look-up tables. This conversion is described in greater detail in Section 2.3.2 of this thesis.

The conversion from frequency to pitch was originally presented by Zwicker (Zwicker 1960) in table format. Zwicker's table documents the Critical Frequency Bands along with their corresponding center frequency, maximum cut-off frequency and bandwidth. The current most widely used method for this conversion is a function approximation of Zwicker's data created by Traunmüller. (Traunmüller 1990)

![Figure 5.1: An A.N.N. Model of Pure-Tone Perception](image)

The existing measures for modeling the auditory system's perception of pure-tone audio signals are attempting to model the behavior of the ear (and to a certain extent the filtering effects of the head and torso) in conjunction with the primary auditory cortex. The primary auditory cortex is a biological neural network. This suggests it is beneficial to model the conversion from the analytical measures of frequency and S.P.L to loudness and pitch using Artificial Neural Networks (A.N.N.s). A.N.N.s are regarded as a good candidate for the estimation of this perceptual mapping function as their structure is based upon biological Neural Networks. Also, their behavior both during and after the training process, has also been found to mimic that of biological Neural Networks (Freeman & Skapura 1991, p. 2).
This Chapter describes the development and testing of a system which will use A.N.N. to model both features of sound perception mentioned above. It will also be investigated if a single A.N.N. model can be used to model both of these aspects of sound perception simultaneously, as in Figure 5.1.

5.2 Methodology

The Matlab® prototyping environment is used to develop, train and test A.N.N.s in this thesis. The Matlab® prototyping environment is an interactive technical computing environment and programming language specialising in the areas of algorithm development, data analysis and numeric computation. A large number of Matlab add-on products are available which contain user interfaces and functions for specialised technical areas. For the development of A.N.N.s the Matlab® Neural Network Toolbox was used. The Neural Network Toolbox contains a dedicated user interface (UI) for the creation, development, training and testing of A.N.N.s.

5.2.1 Training / Testing

The Matlab® Neural Network Toolbox contains 14 algorithms which may be used to train A.N.N.s. The Levenberg-Marquardt backpropagation algorithm was chosen to train the A.N.N. in this thesis. The reasons for this choice are outlined in Section 3.7.1.

The Levenberg-Marquardt algorithm is a least-squares error minimization technique used in the supervised training of A.N.N.s. It is noted as having an appropriate trade-off between efficiency and accuracy which would be suitable for function approximation problems with randomised initial weights. The Levenberg-Marquardt algorithm is described in more detail in Section 3.7.1.

Throughout this thesis the number of nodes in the hidden layer of each A.N.N. was decided based on the network performance during successive training/testing iterations. The number of hidden nodes was varied with each configuration being
tested for ten training sessions. Each of these training sessions begin with the randomisation of the network weight values and consists of 1000 training epochs. These initial training sessions are limited to 1000 epochs as this quantity was found during testing to give a suitable indication of the performance of the A.N.N. when fully trained.

Testing to determine the number of epochs required for preliminary training of the test A.N.N.s was carried out using 1000 of the training vectors used to train the ANN in Section 5.3. The A.N.N.s each contained a single hidden layer with the number of neurons in the hidden layer varying from 2 to 60. Each ANN configuration was fully trained using the Levenberg-Marquardt training algorithm. Intermediate MSE values were noted after 100, 500 and 1000 training epochs. The MSE value of the fully trained ANN was also noted.

A sample of the results of this testing is shown in Figure 5.2 where the MSE after 100, 500 and 1000 training epochs is plotted against the final MSE of the fully trained ANN. The red line in each plot indicates the ideal relationship if the intermediate MSE value exactly matched the MSE of the fully trained ANN.

It can be seen from these plots that on completion of 1000 training epochs the MSE error fits more closely to the MSE of the fully trained ANN compared with that after 100 and 500 epochs.
For more complex function approximation problems using larger A.N.N.s the number of training epochs may need to be increased. The resulting M.S.E. at the end of each training session is noted and the associated network weights logged.

For each configuration, the A.N.N. with lowest M.S.E after the ten training sessions is taken as the best initial approximation of the function. These ‘best’ A.N.N.s are then trained to the maximum amount of epochs as defined by the stop conditions of the Levenberg-Marquardt backpropagation algorithm. These stop conditions were defined as a goal error of 0 and a minimum error gradient of 1e-10.

5.2.2 Analysis of Results

Once fully trained each A.N.N. configuration is tested using both seen and unseen input data. The measures of Maximum Error, M.S.E., Standard deviation and Pearson’s Correlation with respect to the expected results are then used to evaluate the performance of each A.N.N.. The validity of each Pearson’s Correlation value is established by Null Hypothesis testing.

Testing is also carried out to determine the level of generalisation achieved by each A.N.N.. This is achieved by analysing the performance of each A.N.N. configuration when presented with unseen data. This process has been described in greater detail in Section 3.7.3.

5.2.2.1 Maximum Error Measurement

The maximum error value is a measure of the largest deviation of the network outputs from the targets values. It is generated by firstly subtracting each of the target network output values of the training / validation set from the corresponding network output values. This generates a series of signed error values. The magnitude of each signed error value is then found and the maximum magnitude value is then taken to be the maximum error value.

5.2.2.2 Mean Squared Error

The M.S.E. is a measure of the expected squared error produced by the system under test. It is generated by finding the signed errors values as above. The square of
each of these values is then found. The M.S.E. is then equal to the mean value of the series of squared error values. The calculation of the M.S.E. is shown in Eq. 5.1.

\[
M.S.E. = \frac{\sum_{n=1}^{N} (\text{Signed Error})^2}{N} \quad (5.1)
\]

5.2.2.3 Standard Deviation

The Standard Deviation of a series of values is a measure of the spread of the values about the mean. The calculation of the standard deviation, \( \sigma \), of a series is shown in Eq. 5.2 where \( N \) is the number of values in the series \( X \) and \( X_m \) is the mean value of the series.

\[
\sigma = \sqrt{\frac{1}{N} \left( \sum_{i=1}^{N} x_i - x_m \right)^2} \quad (5.2)
\]

For this application the Standard deviation measure is used to evaluate the dispersion of the error values of the network under test.

5.2.2.4 Pearson’s Cross-Correlation

Pearson’s Cross-Correlation is a method for measuring the similarity of two separate signals or data series. The result of Pearson’s Cross-Correlation testing, the correlation coefficient, \( r \), is in the range -1 to +1. The greater the magnitude of this result, the stronger the linear relationship between the signals under test is. A positive \( r \) value indicates a positive linear relationship between the data series. A negative \( r \) value indicates a negative (or inverse) linear relationship.

\[
r = \frac{\sum_{i=1}^{n} [(x(i) - m_x) * (y(i) - m_y)]}{\sqrt{\sum_{i=1}^{n} (x(i) - m_x)^2} \sqrt{\sum_{i=1}^{n} (y(i) - m_y)^2}} \quad (5.3)
\]
In Eq. 5.3, \( x \) is defined as the original sample, \( y \) is defined as the degraded sample and \( m_x \) and \( m_y \) are their respective means. Both samples, \( x \) and \( y \), must be of equal length, \( n \).

Figure 5.3 presents two highly correlated data series of A.N.N. QoV score and PESQ QoV score. When Pearson's Cross-Correlation is used to measure the similarity of these two series an \( r \) value of 0.9597 is produced. This indicates that the two data series have a similarity of 95.97%.

### 5.2.2.5 Null Hypothesis Testing (P-Values)

Null Hypothesis testing is a common method used to test the statistical significance of the correlation coefficient of two data series. This testing method operates on a 'proof by contradiction' basis. By proving that there is a very low (or zero) level of confidence in the Null Hypothesis, the Alternative Hypothesis can be proved.

The null hypothesis used here, \( H_0 \), is that the two data series, A.N.N. QoV score and PESQ QoV score, being tested are independent of one another. An alternative Hypothesis, \( H_a \), states that the data series in question have a high level of dependence on one another.

![Figure 5.3: Comparison of PESQ and A.N.N. Generated QoV Scores](image)
It is hoped that the two data series under test have a strong level of agreement. This requires the use of a one-tailed Null Hypothesis test, which will result in no association between the data series if the variables are not in agreement. (Reilly 1997)

A P-Value is the probability of the Null Hypothesis being correct. Therefore a low P-Value supports the claim that the Alternative Hypothesis is correct. Generally a significance level, \( \alpha \), is assigned and then if the P-Value is less than \((1-\alpha)\) the Null Hypothesis is rejected in favour of the Alternative Hypothesis. (Doody 1999)

When Null Hypothesis testing is carried out on the Pearson’s Cross-Correlation of the data series shown in Figure 5.3 a P-Value of 2.31e-11 is generated for the correlation value of 95.97%. Such a low P-value suggests that the Pearson’s Correlation value is statistically significant.
5.3 An A.N.N. Implementation of the Perceptual Loudness Measure (Phon)

5.3.1 Motivation for A.N.N. Implementation

ISO 226:2003 “specifies combinations of sound pressure levels and frequencies of pure continuous tones which are perceived as equally loud by human listeners” (International Standards Organisation, 2003). These ‘Equal Loudness Contours’ were derived using subjective listening tests carried out on a panel of listeners. Each listener, an otologically normal person aged from 18 to 25, was presented with pure tones of varied intensity and frequency in a binaural fashion, that is, to both ears simultaneously. The source of the sound was placed directly in front of the listener.

From the results of these tests a set of equal-loudness curves were generated. From these curves, different combination of S.P.L. (dB) and frequency (Hz), which are perceived to be of equal loudness, can be identified. From the discrete data generated by subjective testing, an algorithm for the calculation of the perceived loudness (Phon) of an S.P.L – frequency combination was derived using the least squares method. The resulting algorithm to calculate the loudness level, $L_N$, given the frequency, $f$, and the S.P.L., $L_p$, of an audio signal is shown in Eq. 5.4 & Eq. 5.5. These equations use three, 29 entry, look-up tables, namely $\alpha_f$, $L_{u}$, and $T_f$ to perform the calculation outlined in Eq. 5.4 and Eq. 5.5. These look-up tables are included in Appendix A, Table A.2 (ISO 2003)

$$L_N = [(40\log_{10} B_f) + 94]phon \quad (5.4)$$

Where

$$B_f = \left[ 0.4 \times 10^{\frac{L_p + L_u}{10}} \right]^{\gamma_f} - \left[ 0.4 \times 10^{\frac{T_f + L_u}{10}} \right]^{\gamma_f} + 0.00513 \quad (5.5)$$
The algorithm outlined in ISO 226:2003 for the calculation of the perceived loudness can only be implemented accurately for 29 discrete values on the frequency scale. Therefore the frequency components of all audio signals need to be approximated by one of the specified 29 frequencies outlined by ISO 266:2003. This can result in a 'digitization' of such features as uniform tones rising steadily in the frequency domain.

The algorithm outlined in ISO 226:2003 is attempting to model the behavior of the biological function of the perception of audio loudness. The following sections outlines a method which models this biological function using A.N.N. techniques.

5.3.2 Development of A.N.N. Architecture

A two layer feed-forward architecture was used for this development. Two nodes are required in the input layer to take the values of frequency and S.P.L. of the audio signal. A single node is used in the output layer to accommodate the output of the Loudness Level in Phon. The number of nodes implemented in the hidden layer was decided during the training process based on the performance of networks during the training/testing process as described in Section 5.2.1.

A tan–sigmoid activation function is used in the nodes of the hidden layer. This allows for the use of efficient backpropagation based training algorithms. A linear output activation function is used in the output layer node. The linear output function is required to allow the output of the A.N.N. to take on values outside of the range 0 to + 1. This is required as the desired output of the network will be in the range 0 to 90 of the Phon Scale.

5.3.3 Training / Testing

The data used in the training of the A.N.N.s was generated from the Eq 5.4 and 5.5. These equations were implemented for all 29 specified frequencies at each SBL level from 1dB to 90dB (those specified to be accurately catered for by the equations), which resulted in 2581 training vectors. Each training vector contained a frequency value (Hz) and an S.P.L. level (dB) as the input values. A corresponding perceptual
loudness level (Phon), calculated by the Eq 5.4 and Eq. 5.5, was included in the training vectors as the target output value.

Eight A.N.N. architectures were tested with the number of hidden nodes varied from 10 to 60. Each configuration was examined for suitability using the methodology described in Section 5.2. It can be seen from Table 5.1 that the error values for each A.N.N. configuration increases as the number of neurons in the hidden layer decreases. A plot of this data is shown in Figure 5.3. It was deemed from the data that an A.N.N. with less than 10 neurons in the hidden layer is insufficient to accomplish the desired function approximation.

The generalisation achieved by each A.N.N. configuration was tested by presenting each network with unseen data. The unseen data consisted of frequency values other than those included in the look-up table associated with Eq. 5.4 and Eq. 5.5. As these equations only account for those 29 frequencies specified in the ISO standard, no target results can be generated for this unseen data. The results generated by the A.N.N. are therefore examined objectively. A smooth continuous output is necessary if the A.N.N. has achieved good generalisation, while an erratic or discontinuous output will indicate an A.N.N. with poor generalisation.

5.3.4 Results

Table 5.1 shows the results achieved during the investigation of the performance of A.N.N.s in modeling the Perceptual Loudness Conversion (Phon). It can be seen that a number of A.N.N.s have proved to be suitable for the estimation of the Perceptual
Loudness measure in the Phon scale. The suitability of a network is indicated by a high correlation and low error value. A network comprising of 40 nodes in the hidden layer was developed and trained to provide a M.S.E. of 0.000958, with a standard deviation of 0.031, a maximum error of 0.2674 and a Pearson’s Cross Correlation value of 1 with the expected values generated by the method outlined by the ISO.

<table>
<thead>
<tr>
<th>No. Nodes in Hidden Layer</th>
<th>10</th>
<th>20</th>
<th>25</th>
<th>28</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total No. of Weights</td>
<td>41</td>
<td>81</td>
<td>101</td>
<td>113</td>
<td>121</td>
<td>161</td>
<td>201</td>
<td>241</td>
</tr>
<tr>
<td>Min M.S.E. of 10 1000 epoch Training Sessions</td>
<td>0.743</td>
<td>0.0168</td>
<td>0.01887</td>
<td>0.002901</td>
<td>0.003737</td>
<td>0.006259</td>
<td>0.001862</td>
<td>0.00607</td>
</tr>
<tr>
<td>Max error of Net with min MSE</td>
<td>5.7625</td>
<td>1.225</td>
<td>0.351</td>
<td>0.4469</td>
<td>0.6096</td>
<td>0.6866</td>
<td>0.3915</td>
<td>0.6653</td>
</tr>
<tr>
<td>Best MSE result received</td>
<td>0.742</td>
<td>0.01395</td>
<td>0.01418</td>
<td>0.00257</td>
<td>0.00256</td>
<td>0.000958</td>
<td>0.00113</td>
<td>0.001878</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.8441</td>
<td>0.1182</td>
<td>0.1191</td>
<td>0.0507</td>
<td>0.0506</td>
<td>0.031</td>
<td>0.0336</td>
<td>0.0433</td>
</tr>
<tr>
<td>Max error</td>
<td>5.7407</td>
<td>1.1682</td>
<td>0.5793</td>
<td>0.4328</td>
<td>0.4576</td>
<td>0.2674</td>
<td>0.3825</td>
<td>0.3467</td>
</tr>
<tr>
<td>Cross Correlation</td>
<td>0.9995</td>
<td>0.99999</td>
<td>0.99999</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>P-Value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.1: Results from Training of A.N.N. Model of the Perceptual Loudness Conversion (Phon)

As the number of neurons in the A.N.N. increase so does the amount of space required to store the network on a system and the number of calculation required to execute the A.N.N.. Therefore, it is beneficial that the number of neurons is kept to a minimum. A trade off between the number of neurons and the accuracy of the results is made. An A.N.N. comprising of 28 neurons in the hidden layer, with a M.S.E. of 0.00257, a standard deviation 0.0507 and a correlation of 1 with expected values is chosen for use as the ideal network in the estimation of the perceptual loudness measure according to the trade off explained above.

Of course, where greater accuracy is needed an increase in the number of neurons may be made. If the situation requires a smaller A.N.N. with a shorter execution time, an A.N.N. with fewer neurons may be used at the expense of accuracy.
Figure 5.5: Results from Testing of Perceptual Loudness (Phon) Mapping
A.N.N.

Figure 5.5 depicts a comparison of the performance of the A.N.N. method developed here and the method described by Eq. 5.4 and Eq. 5.5. Both methods were implemented with a constant S.P.L. of 80dB and a frequency value varying form 20Hz to 12500Hz. The equation based method was presented with those frequency values associated with the look-up table. The A.N.N. was presented with frequency values rising from 20Hz to 12500Hz in increments of 1 Hz. The resulting estimation of the perceived Loudness from both methods is plotted in the figure. From this figure it can be seen that the A.N.N. shows a very high level of correlation with the values generated by the method outlined in the ISO standard.

It was also found that the A.N.N. method produced a continuous curve when presented with a constant SPL and varying frequency. This is in contrast to the discontinuous nature of the results generated by the method outlines by the I.S.O. standard.
Figure 5.6: Close-Up of Figure 5.5 (1050Hz – 1930Hz)

Figure 5.6 highlights the digitisation effects introduced by the implementation outlined by ISO 226:2003 (labeled 'DSP output'). This effect has been overcome by the A.N.N. method of perceptual loudness evaluation (implemented with 28 neurons in the hidden layer). An excellent level of generalisation is shown by the A.N.N. configuration as shown by the smooth continuous curve shown in Figures 5.5 and 5.6. (Riordan & Doody 2007(a))
output layer to accommodate the output of the Loudness Level in Sone. The number of nodes in each of the hidden layers was decided during the training process based upon the performance of network implementations during the training/testing process.

5.4.3 Training/Testing

The data used in the training of the A.N.N.s, was generated by calculating the Phon values from the equations provided in I.S.O. 226:2003 and then converting these to Sone with Eq. 5.3. This resulted in 2581 training vectors, each containing a frequency value (Hz), an S.P.L. level (dB) as inputs and a corresponding perceptual loudness level (Sone) as the target value. This training set facilitates supervised training methods.

Twelve A.N.N.s were developed with the number of hidden nodes varied from 15 to 40 for the first hidden layer and from 2 to 5 for the second hidden layer. Each configuration was examined for suitability using the methodology described in section 5.2. As can be seen in Section 5.4.4, the error values for each A.N.N. configuration increase as the number of neurons decreases. Based upon these results, shown in Figure 5.7, it was decided that an A.N.N. with less than 15 neurons in the first hidden layer would be insufficient to accomplish this function approximation.

![Figure 5.7: M.S.E. Vs. Number of Neurons in First Hidden Layer](image)

Each A.N.N. was also examined for its generalisation capabilities in the same manner as described in Section 5.3.3.
The conversion from the Phon measure to the Sone measure presented in Eq. 5.6 was found to be a C0 continuous function as shown in Eq. 5.7 and Eq. 5.8 where both methods give a result of 1 when \( L = 40 \) and the limit as \( L \) goes to 40 respectively.

Eq. 5.9 shows the at \( L = 40 \), the first derivative of Eq. 5.7, \( \frac{dS}{dL} \), is equal to 0.069315. Eq. 5.10 shows that at the limit as \( L \) goes to 40 of the first derivative of Eq. 5.8, \( \frac{dS}{dL} \), is equal to 0.06605. These values are not equal and therefore the function is not C1 continuous.

To approximate a non-continuous function efficiently, an A.N.N. with at least 2 hidden layers is required. For this reason a 3-layer Feed-Forward A.N.N. architecture was implemented. “...a three-layer network with threshold activation functions could represent an arbitrary decision boundary to arbitrary accuracy” (Bishop 1995, p. 128).
output layer to accommodate the output of the Loudness Level in Sone. The number of nodes in each of the hidden layers was decided during the training process based upon the performance of network implementations during the training/testing process.

5.4.3 Training/Testing

The data used in the training of the A.N.N.s, was generated by calculating the Phon values from the equations provided in I.S.O. 226:2003 and then converting these to Sone with Eq. 5.3. This resulted in 2581 training vectors, each containing a frequency value (Hz), an S.P.L. level (dB) as inputs and a corresponding perceptual loudness level (Sone) as the target value. This training set facilitates supervised training methods.

Twelve A.N.N.s were developed with the number of hidden nodes varied from 15 to 40 for the first hidden layer and from 2 to 5 for the second hidden layer. Each configuration was examined for suitability using the methodology described in section 5.2. As can be seen in Section 5.4.4, the error values for each A.N.N. configuration increase as the number of neurons decreases. Based upon these results, shown in Figure 5.7, it was decided that an A.N.N. with less than 15 neurons in the first hidden layer would be insufficient to accomplish this function approximation.

![Figure 5.7: M.S.E. Vs. Number of Neurons in First Hidden Layer](image)

Each A.N.N. was also examined for its generalisation capabilities in the same manner as described in Section 5.3.3.
5.4.4 Results

The table of results populated during the investigation of the performance of the various A.N.N. configurations can be found in Appendix A, Table A.3. From this table it can be seen that a number of A.N.N.s suitable for the estimation of the perceptual loudness measure in the Sone scale were created. A network comprising of 30 nodes in the first hidden layer and 5 in the second was designed which was trained to provide a M.S.E. of 0.0000678, with a standard deviation of 0.0082 and maximum individual error of 0.0826.

Figure 5.8: Results from Testing Perceptual Loudness (Sone) Mapping A.N.N.

Figure 5.8 depicts the performance of the A.N.N. that comprised of 30 neurons in the first hidden layer and 5 in the second. This A.N.N. was presented with a constant S.P.L. of 80dB and a frequency value varying form 20Hz to 12500Hz in increments of 1 Hz. For reference the results generated by presenting Eq. 5.4, 5.5 and 5.6 with the same input S.P.L value and those frequencies present in the associated look-up table are also shown in the figure, labeled ‘DSP Output’. A high degree of correlation between the A.N.N. based method and the equation based method is again shown. The digitisation effect of the equation based method is still present while the A.N.N. produces a highly continuous curve.
Figure 5.9: Close-Up of Figure 5.8 (1-2kHz)  
Figure 5.10: Close-Up of Figure 5.8 (2) (6-13kHz)

Figure 5.9 and Figure 5.10 are magnified versions of Figure 5.8 and shown in greater detail the digitization effect which has been overcome by the use of A.N.N.s. The smooth continuous curves shown in Figures 5.8, 5.9 and 5.10 depict the output of a network with a high level of generalisation.

As stated earlier, as the number of neurons in the A.N.N. increase so does the amount of space required to store the network on a system and the number of calculations required to execute the A.N.N.. Therefore, it is beneficial that the number of neurons is kept to a minimum.

A suitable trade-off between network size and performance is the network containing 20 neurons in the first hidden layer and 1 in the second. This network yields a M.S.E. of approximately 0.023 and a Standard Deviation of 0.048. Upon investigation it was also found to produce a smooth continuous curve when tested as above. This demonstrates that the A.N.N. possesses a high degree of generalisation. The actual choice of A.N.N. from those presented will be application specific and will be dependent on such features as accuracy required and system requirements.
5.5 An A.N.N. For the Frequency to Critical Band Rate Conversion

5.5.1 Motivation for A.N.N. Implementation

The conversion from frequency to pitch was originally presented by Zwicker in table format (Zwicker, E. 1960). Zwicker's table documents the Critical-Band number along with the corresponding center frequency, maximum cut-off frequency and bandwidth.

Since the first publication of this table in 1961, many function approximations of the data, with varying degrees of accuracy, have been presented (Fourcin. et al 1977), (Schroeder, Atal & Hall 1979) and (Zwicker & Terhardt 1980). The current most widely used and accepted method for this conversion is outlined by Traunmuller in his paper 'Analytical expressions for the Tonotopic Sensory Scale' (Traunmuller 1990).

From Zwicker's table outlining the limits of the Critical-Bands, only the Bark value at the specific frequencies listed can be discerned accurately. The Bark values of all other frequencies values are no-more than educated estimations. While this is generally acceptable in the field of speech processing, there is room for improvement. These improvements may be of use when accurate representations of the perceived pitch are required such as models of the cognitive aspects of sound perception.

Traunmuller's equation for the conversion from the frequency scale to the Bark Scale is a function approximation of the information presented in Zwicker's critical-band rate table. This function approximation equation is shown in Eq. 5.11. "The values calculated in this way agree with the table for $f > 100\text{Hz}$ to within $\pm 0.05\text{Bark}$" (Traunmuller 1990). This error measurement can only be taken from the frequency values present on the table. Errors associated with frequencies not listed on the table are unknown. Thus the values generated by this equation which are between Zwicker's values are, again, an extrapolation.
\[ z' = \frac{26.81 f}{1960 + f} - 0.53 \]  \hspace{2cm} (5.11)

\[
\begin{align*}
\text{If } z' &< 2, \quad z = z' + 0.15(2 - z) \\
\text{If } z' &> 20.1, \quad z = z' + 0.11(z - 20.1) \\
\text{Else } &z = z'
\end{align*}
\]

where \( z \) is the Critical-Band Rate in Bark and \( f \) is the frequency to be converted to Critical-Band Rate.

Both Traunmuller and Zwicker, along with many others, are attempting to model the behavior of a fundamentally biological function. Therefore it is logical to attempt to model this conversion using A.I. techniques.

5.5.2 Development of A.N.N. Architecture

As with the A.N.N. for the estimation of perceived loudness, a two layer feedforward architecture was used for this A.N.N.. A single node was required in the input layer to take the frequency value of the audio signal. A single node was used in the output layer to accommodate the output of the perceived pitch in Bark. A tan-sigmoid function was used in the nodes of the hidden layer and a linear function in the output layer node. The number of nodes in the hidden layer was decided during the training process based the performance of the networks during the training/testing process.

5.5.3 Training / Testing

The data used here in the training of the A.N.N.s was taken directly from Zwickers table of Critical-Band limits. This supplied an input of 25 input frequency values for the network with 25 corresponding output values. This allows for a supervised training algorithm to be used.
Seven A.N.N.s were developed with the number of hidden nodes varied from 2 to 20. Each configuration was examined for suitability using the methodology described in Section 5.2. As can be seen from the Section 5.5.4, the error values for each A.N.N. configuration increase as the number of neurons decreases. Based upon these results, shown in Figure 5.11, an A.N.N. with less than 2 neurons in the hidden layer would be insufficient to accomplish this function approximation.

![Figure 5.11: M.S.E. Vs. Number of Neurons in Hidden Layer](image)

Each trained A.N.N. configuration was also tested for instances of over-fitting and performance on unseen data. This is done by presenting each A.N.N. with frequency values not present in the data set and ensuring the result is consistent with known values. The unseen data consisted of frequency values other than those values listed in Zwicker's table of Critical-Band limits.

<table>
<thead>
<tr>
<th>Neurons in Hidden Layer</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total No. of Weights</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>22</td>
<td>32</td>
<td>42</td>
</tr>
<tr>
<td>Min M.S.E. of 10 1000 epoch Training Sessions</td>
<td>0.00255753</td>
<td>0.000724497</td>
<td>0.00094247</td>
<td>0.00072493</td>
<td>0.0004452</td>
<td>0.000315</td>
<td>0.000156</td>
</tr>
<tr>
<td>Max error of Net with min MSE</td>
<td>0.1003</td>
<td>0.0622</td>
<td>0.0593</td>
<td>0.0617</td>
<td>0.0407</td>
<td>0.046</td>
<td>0.0372</td>
</tr>
<tr>
<td>Best MSE result received</td>
<td>0.000889353</td>
<td>0.000722532</td>
<td>0.00090446</td>
<td>0.000416</td>
<td>0.0004452</td>
<td>0.000146</td>
<td>4.12E-05</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.0304</td>
<td>0.0274</td>
<td>0.0308</td>
<td>0.0208</td>
<td>0.0215</td>
<td>0.0123</td>
<td>0.0066</td>
</tr>
<tr>
<td>Max error</td>
<td>0.0577</td>
<td>0.0618</td>
<td>0.057</td>
<td>0.0554</td>
<td>0.0407</td>
<td>0.0363</td>
<td>0.0194</td>
</tr>
<tr>
<td>Cross Correlation</td>
<td>0.99999</td>
<td>0.99999</td>
<td>0.99999</td>
<td>0.99999</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>P-Value</td>
<td>2.49E-56</td>
<td>2.29E-57</td>
<td>3.03E-56</td>
<td>1.23E-60</td>
<td>8.72E-60</td>
<td>2.39E-65</td>
<td>2.57E-67</td>
</tr>
</tbody>
</table>

Table 5.2: Results from Training of A.N.N. Model of the Perceived Pitch Conversion
5.5.4 Results

Table 5.3 shows the results achieved during the investigation of the performance of the A.N.N. configurations. It is clear from these results that the A.N.N. with 20 neurons in the hidden layer has the best performance on both seen and unseen data. With a Pearson's correlation value of 1 and an extremely low P-value this network is an excellent system for the conversion of frequency to the perceived pitch.

Another network comprising of 10 nodes in the hidden layer is a suitable candidate for use in applications where network size must be kept to a minimum. This A.N.N. was trained to provide a M.S.E. of 0.000445, with a standard deviation of 0.0215 and maximum individual error of magnitude 0.0407. A plot of the output of this A.N.N. when presented with an input of frequency values ranging from 20 – 15500 Hz in increments of 1Hz, is shown in Figure 5.12. It can be seen to compare very well with a plot of the data listed in Zwicker's table. The curve generated by this A.N.N. is an extremely smooth continuous curve.

![Figure 5.12: Results of Testing Perceived Pitch (Bark) Mapping A.N.N.](image)

When tested for generalisation, networks with higher numbers of hidden nodes demonstrated symptoms of over-fitting. In an attempt to match the training data more closely, the values generated at instances between the training points became more non-uniform, as demonstrated in Figures 5.13 & 5.14. This results in a poor performance of the A.N.N. on unseen input data. The plots in Figures 5.13 and 5.14
show the continuous output from the 15 neuron network, when presented with a continuous input varying from 20 – 15500 Hz. Large variations can be seen in the region 100 – 1000 Hz, even though all of the expected outputs of the 25 entry training set have been met to within ±0.0363. Over-fitting was found on all A.N.N.s created for this function approximation problem with more than 10 neurons in the hidden layer, included the 20 neuron network mentioned earlier.

For a neural Network to perform uniformly for unknown inputs, the A.N.N. will need to generalise well. A degrading of the generalisation will cause over-fitting. Over-fitting of a neural network occurs when the flexibility, or degree of freedom, of the network is too great. The degrees of freedom of an A.N.N. can be controlled by a process called ‘structural stabilization’. This involves limiting the amount of changeable factors (neurons or weights) in a network. The fewer the changeable factors in the network the less likely it is that over-fitting will occur. Therefore the fewer neurons contained in a network, the less over-fitting is likely to occur. (Bishop1995, p.332 and Riordan & Doody 2007(b))
5.6 An ALL-In-One A.N.N. Pure-Tone Perception Model

5.6.1 Motivation for A.N.N.

The three previous sections of this chapter have shown that A.N.N. can be used to model the individual features of the human auditory system. This section will present the development of a single A.N.N. with the ability to generate both the Perceived Loudness and Pitch of a audio signal simultaneously.

5.6.2 Development of A.N.N. Architecture

This A.N.N. is designed to generate both the Perceived Pitch and Loudness. A minimum of three layers will be required in the network due to the non-linear characteristic of the conversion from frequency and S.P.L. to the Sone Scale of Perceived Loudness shown in Section 5.4.2. A network with two hidden layers and an output layer was implemented. Two neurons were required in the input layer to take the frequency and S.P.L. values of the audio signal. Two nodes were also required in the output layer to accommodate the output of the perceived pitch in Bark and Perceived Loudness in Sone. A tan-sigmoid function was used in the nodes of the hidden layers and a linear function in the output layer node. The number of nodes in the hidden layers was decided during the training process.

5.6.3 Training / Testing

A training set of 2581 vectors was compiled from the data used to train the A.N.N.s described in the Sections 5.3.3, 5.4.3 and 5.5.3. Each training vector contains two input values, frequency and S.P.L., and two corresponding target values, the pitch in Bark and the loudness in Sone.

Twelve A.N.N.s were developed with the number of hidden nodes varied from 5 to 40 for the first hidden layer and from 2 to 5 for the second hidden layer. Each configuration was examined for suitability using the methodology described in Section 5.2. As can be seen from Section 5.6.4, the error values for each A.N.N. configuration increase as the number of neurons decreases. Based upon these results,
shown in Figure 5.15, it was deemed that an A.N.N. with less than 5 neurons in the first hidden layer would be insufficient to accomplish this function approximation.

![Figure 5.15: M.S.E. Vs. Number of Neurons in First Hidden Layer](image)

Testing was also carried out to determine the level of generalisation achieved by each A.N.N. on both the estimations of Perceived Loudness and Pitch.

5.6.4. Results

Table A.4 outlines the performance of the A.N.N.s during training. From analysis of these results the network containing 30 neurons in the first hidden layer and 5 neurons in the second was decided upon as the best choice. For the estimation of perceived Pitch, this A.N.N produces a M.S.E. of less-than 0.0000075 and standard deviation of the 0.0083 with values obtained from Zwickers table. Similarly good results of 0.0000746 and 0.0.419 are produced for the estimation of Perceived Loudness in the Sone scale when compared with results outlined in ISO:226:2003.

Upon further investigation it seems over-fitting has occurred with this A.N.N. Figures 5.16 shows the estimation of Perceived Loudness resulting from inputs of 60 dB S.P.L. and a frequency varying from 20Hz to 12500Hz. Figure 5.17 is a magnified version of Figure 5.16 which highlights the irregularities which are not supported by the training data.
Investigations into other suitable A.N.N.s showed that the A.N.N. with 20 nodes in the first hidden layer and 2 nodes in the second is the best performing A.N.N. which does not suffer from over-fitting. Investigations showed that this network possesses a high level of generalisation. Figure 5.18 and 5.19 show the estimations of Perceived Loudness and Pitch, respectively, produced for the same input values mentioned above, for the A.N.N. containing 20 nodes in the hidden layer.
5.7 Conclusions

The results which have been presented here clearly show that the conversion from the analytical measures of an audio signal, frequency and S.P.L, to Perceived Loudness and Critical-Band Rate (or Bark) can be implemented using an A.N.N.. It has also been shown that the use of A.I. techniques have advantages over existing and accepted methods.

The use of A.N.N.s in the estimation of perceived loudness has been shown to eliminate the need to approximate the frequency value of the signal to one of 29 specified frequencies. The values generated for frequencies between those specified are generated purely by the A.N.N. and cannot be validated without subjective testing. Some validation can be inferred by the fact that A.N.N.s have been noted to posses very similarly characteristics to that of biological neural networks and be adept at modeling biological functions.

Similarly, this implementation of the frequency to Critical-Band Rate conversion through A.N.N.s is shown bridge the gap between the 25 specified critical band values specified by Zwicker. While this has been done in the past by many function approximation attempts, an A.N.N. approach has proved to be as suitable as other methods employed.
Chapter 6

Intrusive Objective Quality of Voice Estimation Through Machine Learning

6.1 Introduction

As outlined in the Chapter 4, Intrusive Objective QoV Estimation is vital in the fields of communication system development and testing. It facilitates an automated testing procedure that objectively predicts how the QoV will be perceived by the system user. This alleviates the need for costly subjective testing.
Intrusive QoV estimation algorithms generate a QoV score by comparing an original speech sample to a degraded speech sample. The degraded speech sample is a result of passing the original sample through the system under test. By evaluating the distortions present in the degraded sample with regard to the original sample a QoV score can be generated.

Chapter 4 outlines some of the many Intrusive Objective QoV estimation algorithms which have been developed to date. The pre-processing section of the intrusive objective QoV estimation algorithm developed here will take inspiration from these already established methods, with PESQ being the most influential (Section 4.5 Chapter 4). This pre-processing stage will provide the A.N.N. with the parameters needed to accurately predict a QoV score.

The first section of this chapter will outline the requirements of this QoV estimation algorithm. The next section will describe the development and testing of the algorithms pre-processing stage. This is used to extract the perceived difference between the two speech samples provided. The development and testing of the A.N.N.-based mapping section is then presented in Section 6.4. Finally the resulting Intrusive Objective QoV estimation algorithm developed is evaluated and conclusions made.

6.2 Requirements

The Intrusive QoV estimation algorithm developed in this thesis will use an A.N.N to estimate the QoV capabilities of the system under test. It will take as its inputs two speech samples, one original sample and one degraded sample. The A.N.N. portion will take as its inputs the various parameters which describe the perceived difference between the original and degraded speech samples. The single output of this QoV estimator will be a QoV score which will correspond to the ACR Mean Opinion Score described in Chapter 4.
6.3 Pre-Processing Algorithm

The pre-processing algorithm of this Intrusive QoV estimation tool incorporates the functionality of blocks 1 to 10 shown in Figure 6.1. The pre-processing section will take as its inputs an original and a degraded speech sample. The output of the pre-processing section consists of a number of measures representative of the perceived distortions present in the degraded speech sample. These measures are presented to the A.N.N. and mapped to a QoV score.

6.3.1 Formatting of Inputs

The original speech samples used as inputs for this QoV estimation system must conform to the standards set out by the ITU in P. 830. Each sample should have a duration of approximately 8 seconds and contain two unrelated sentences. Each sentence should have a duration of approximately 2 to 3 seconds. The samples should be recorded in a room with low reverberation time and low noise level. Speakers should speak the sentences fluently but not in a dramatic fashion and at a constant comfortable level. Care must be taken to ensure that this level is kept below the overload level of the recording equipment by approximately 20 to 30 dB. Both male and female samples are used as a communication system’s QoV capabilities can be gender specific (ITU 1996(c)).
The degraded speech samples should be generated by the transmission of the original speech samples over the communication system under test. Due to the difficulties in generating a large number of speech samples from varying communication networks, a speech samples database entitled “ITU-T Series P: Supplement 23” (ITU-T 1998) was employed. This database contains more than 5000 samples of original, pre-processed and degraded speech in English, French, Japanese and Italian. Each sample is in the form of a 16-bit linear PCM (binary) file with a ‘low-byte first’ format and conforms with the standards for speech samples for use in QoV estimation laid out in ITU-T P.830.

In addition to this database, degraded speech samples which were generated using a VoIP network simulator are also use. The development of this VoIP network simulator and the generation of these speech samples is described in Section 7.5.

In the interest of robustness, some processing is performed on the input speech samples to ensure their format will comply with that expected by this algorithm. The system is designed to accept two monaural speech samples of equal duration in 16-bit linear PCM format.

Firstly the number of channels present in each file is checked. As the system only accepts monaural samples, if a stereo recording is presented, it is converted to monaural. This is carried out by averaging each corresponding bits of the two audio channels and placing the results in a new file. This new file is a monaural representation of the stereo file. The algorithm then checks the length of each speech sample. If the samples are of unequal duration the shorter sample is padded with silence of an appropriate length. Equal sample lengths are required to perform whole sample level alignment described in Section 6.3.2. The addition of silence to the sample will have little effect upon the result as silent intervals will not contribute to the QoV score in this system.
6.3.2 IRS Filtering

The Intermediate Reference System (IRS) is the transmission characteristics recommended by the ITU-T. The IRS receive characteristic is a model of the frequency response of a contemporary analogue telephone receiver. A 300-3400 Hz band-pass filter is added to the IRS receive characteristic as this system is designed for use with speech band telephony. This new characteristic is known as the Modified IRS receive characteristic. A more accurate representation of the speech information perceived by the system user can be generated by applying this to the speech samples. The implementation of the Modified IRS receive characteristic is presented in ITU-T P.830 Annex D (ITU-T 1996(c)). A plot of the Modified IRS receive characteristic is shown in Figure 6.2.

For this project the Modified IRS receive characteristic has been implemented with an 8th order Infinite Impulse Response (IIR) digital filter. This filter is designed in accordance with the guidelines set out by the ITU-T documentation and is applied to both speech samples. The application of this IIR filter results in speech samples with a characteristic more representative of the actual audio presented to the user of the communication system under test. (ITU-T, 1996(c))

![Figure 6.2: The Modified IRS Receive Characteristic](image-url)
6.3.3 Division of Samples into 32ms Frames

In order to mimic the operation of the cochlea of the inner ear (Section 2.2.3 Chapter 2) a short-term Time-Domain to Frequency-Domain transform must be performed. To accomplish this each speech sample must be divided into a number of frames over which to perform the short-term Fourier Transform. The duration of each frame is set to 32ms. This corresponds to a time-scale which is often referred to as the sampling-rate of the ear. A similar frame duration is used in previously established Intrusive QoV estimation algorithms (Rix et al. 2002).

The length of each speech sample was measured in the formatting section described in Section 6.3.1. The number of samples per frame is computed based upon the duration of each frame multiplied by the sampling frequency of the speech sample. The number of frames per speech sample is found by dividing the length of the speech sample by the length of each block.

Two matrixes are then created, one to represent the original speech sample and one to represent the degraded speech sample. The number of rows of each matrix is equal to the number of samples per frame. The number of columns is equal to the number of frames per speech sample.

Each matrix is then populated sequentially with the 16-bit integer values of their associated speech samples. The result is two arrays, containing a series of 32ms frames, which collectively represent the original and the degraded speech samples.

If the number of blocks resulting from the calculation described above is not an integer, it is rounded up to closest integer. This may lead to the final block of the series being incomplete because there are not enough values in the sample to populate it. When this occurs, the sample is padded with zeroes. These will represent silence in the algorithm and will have little bearing of the result of the algorithm as silence is ignored at a later step of the algorithm.
6.3.4 Time to Frequency Domain Conversion

The cochlea of the human ear performs an operation highly analogous to a short-term Fourier Transform (Section 2.2.3). It decomposes a received audio signal into its frequency components and assigns an amplitude value to each. This information is then conveyed to the auditory cortex.

![Time Domain Representation](image1)

![Frequency Domain Representation](image2)

Figure 6.3: The Time Domain to Frequency Domain Conversion

The time to frequency domain conversion is carried out here using a short-term Fast Fourier Transform (FFT). The FFT algorithm used is outlined in Eq. 6.1, where $x$ is an array of integers of length $N$. The FFT is performed on each 32ms frame of each sample. The results of each FFT, $X$, is then transformed to the power spectral domain, $P$, using Eq. 6.2., where $X$ is multiplied by its complex conjugate. This provides an accurate representation of the power of each frame at specific points along the frequency scale. Figure 6.3 shows the conversion resulting from the method described. The top plot in this figure is a time-domain representation of a 32ms frame. The lower plot is the corresponding frequency-domain representation.
The Frequency-domain and Spectral Power conversions described here generate a power spectral density array, \( P \), of length, \( N \). This is equal to the length of the input data of the 32ms Frame. Only half of the values \( P \) are of use. With the occurrence of aliasing, the first \( N/2 \) values of array are repeated (in reverse order) in the second \( N/2 \) values. The \( N/2 \) useful data values correspond to those frequency values from zero to half the sampling rate of input data.

In order to perform certain perceptual auditory steps later in the algorithm it is required that each Spectral Power value in the array is associated with a frequency value.

\[
X[k] = \sum_{j=0}^{N/2-1} x[2j]e^{-i(2\pi/(N/2))kj} + e^{-i(2\pi/N)k} \sum_{j=0}^{N/2-1} x[2j + 1]e^{-j(2\pi/(N/2))kj} \quad (6.1)
\]

\[
P = X . \overline{X} \quad (6.2)
\]

The frequency values associated with each value in the Spectral Power array are distributed linearly and can therefore be calculated easily. This calculation is outlined in the pseudo code above where an array entitled ‘Freq_values’ is created to hold the frequency values which correspond to the Spectral Power values generated by the FFT and Spectral Power operations. The resulting ‘Freq_values’ array was also used in the generation of the plot in Figure 6.3.
6.3.5 Utterance Identification

For the PESQ QoV estimation algorithm, an ‘Utterance’ is defined as “a continuous section of speech of at least 300ms duration, containing no silent period longer than 200ms” (Rix et al. 2002). In this section Utterances are identified using a Voice Activity Detector (V.A.D) operating upon the ‘framed’ versions of the speech samples. The V.A.D. uses a threshold feature to allow for noise in the signal.

Initially each 32ms frame is identified as containing either speech or silence. The threshold is set by analysing the power levels of those frames at the beginning of the original speech sample. These are assumed to contain silence. The minimum mean power of the first 5 32ms frames of the original speech sample is said to be the measure of silence. The threshold is then set to this value raised by three orders of magnitude. This level of magnification allows for non-uniform noise in the speech signal and was found, by subjective means, to give optimum results to the VAD algorithm. Once the threshold is set, any packet with a power rating above this level is identified as containing speech.

The instances of speech detected by the VAD must then be assigned to utterances. Using the VAD, the mean power levels of the series of frames of each sample are scanned from the start through to the end of the sample. When a frame is defined as containing speech the start point of a ‘Provisional Utterance’ is declared. With the packetisation described in Section 6.3.3, a duration of 300ms corresponds to approximately nine 32ms frames and a duration of 200ms corresponds to six 32ms frames. Therefore, the algorithm continues to scan until it finds a series of 6 consecutive frames which are defined by the VAD as silence. The frame preceding the first frame of this series of silence is marked as the end of the ‘Provisional Utterance’. If the duration of this ‘Provisional Utterance’ is greater than 9 32ms frames the ‘Provisional Utterance’ is relabeled as an Utterance. If it is of duration less than 9 frames, it is discarded. This operation is shown in pseudo-code on the next page.
If $\text{ave}_\text{power}(k) > \text{Threshold}$

$\text{Provisional}_\text{Start}_\text{utt} = k$

Elseif $\text{ave}_\text{power}(k : k+5) < \text{Threshold}$

$\text{Provisional}_\text{End}_\text{utt} = k - 1$

end

IF ( $\text{Provisional}_\text{End}_\text{utt} - \text{Provisional}_\text{Start}_\text{utt}$ ) > 9

$\text{Utt}_\text{list}(i,1) = \text{Provisional}_\text{Start}_\text{utt}$

$\text{Utt}_\text{list}(i,2) = \text{Provisional}_\text{End}_\text{utt}$

$i = i++$

End

Figure 6.4 shows the resulting Utterance Identification of an 8 second speech sample. The uppermost plot in the figure shows the mean power of each 32ms frames versus that frames corresponding location on the time-scale. The resulting identified Utterances are shown in the lower plot. Those sections of the sample identified as an Utterance are shown in red.

Figure 6.4: Utterances Identified
The ITU-T recommendation on speech sample preparation state that a sample should contain two sentences (ITU-T 1996(c)). Therefore, two Utterances should be detected in the original speech sample presented to the QoV estimation algorithm. If more than or less than two Utterances are detected in the original sample, either the sample has been poorly prepared or an error has occurred. As this tool is designed to be used with ITU-T prepared speech sample, it is assumed that the samples have been prepared correctly. Therefore an error has occurred in the Utterance Identification system.

6.3.5.1 Incorporation of Adaptive Threshold Level

An obvious error that may occur is the inaccurate calculation of the noise threshold level. This can usually be attributed to a false assumption of the presence of silence at the beginning of the sample. This leads to a threshold level that is too high. This causes peaks within an Utterance to get marked as individual Utterances while low levels of speech within an utterance are labeled noise not constituting an utterance. An instance of this error is shown in Figure 6.5.

![Figure 6.5: Utterances Misidentified](image)

Similarly, if the noise level between the two sentences of the sample does not fall below the threshold, both Utterances being classified as a single utterance. In this case the threshold has been set too low and needs to be raised.

To overcome these error types, the threshold level has been made adaptive. Here it is assumed that the initial threshold level ‘guess’ is close to the desired threshold level. For
instance, if only a single Utterance is identified, it is not assumed that the original threshold level is greater than only a single very high peak in the frame powers. If this was true it would mean that the initial threshold level 'guess' was inaccurate by many orders of magnitude. It is assumed that the threshold level is of the correct order of magnitude but slightly below the ideal noise threshold level. When this occurs, the threshold level is raised according to the pseudo-code below and the Utterance Identification algorithm is repeated.

```
If num_utts > 2
    Thres_level = Thres_level * 0.99
    ** Execute Algorithm **
Elseif num_utts < 2
    Thres_level = Thres_level * 1.01
    ** Execute Algorithm **
End
```

Similarly, if more than two Utterances are detected (as in Figure 6.5), the threshold level is assumed to have been set too high and needs to be lowered. The threshold is lowered according to the pseudo-code above and the Utterance Identification algorithm is executed again. This process is repeated until only two Utterances are detected by the algorithm.

As mentioned above, by using the ITU-T prepared speech samples, it is assured that two utterances will be present in each original speech sample. The same is not true for degraded speech samples. These samples may have undergone heavy distortion, packet-loss and variable delay. This may result in the sentences contained in the original sample being split into a number of Utterances as defined earlier.

To overcome this feature of degradation in this Utterance Identification algorithm, an intelligent feature to account for gaps in the degraded utterances has been incorporated. This feature is only used when a thorough variation of the threshold level has not yielded a result of two utterances.
6.3.5.2 Account of Gaps/ Pauses Within Speech

When more than two Utterances are detected after threshold variation has failed to yield a correct result, the largest 'gap' is said to be the genuine divide between the two ‘true’ Utterances. All other gaps are removed and said to have been the insertion of silence by the system under test. Later in this QOV estimation algorithm, these gaps may be reintroduced by the delay estimation and compensation algorithm. The result is two utterances being identified in the degraded sample which correspond to the Utterances identified in the original sample.

This QoV estimation algorithm will take into account only the distortions which occur during instances of speech in the samples. Distortions which occur during silence will not be taken into account as these have little effect to the QoV MOS achieved (Beerends et al. 2002). Once the Utterances are identified satisfactorily, an array logging the beginning and ending points of the identified Utterances in each sample is created. The points are referred to by their 32ms frame number. Table 6.1 shows the resulting array generated when the speech sample shown in Figure 6.5 was presented to the Utterance Identification algorithm developed here.

<table>
<thead>
<tr>
<th>Original Speech Sample</th>
<th>Utterance 1</th>
<th>Utterance 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utterance Start Frame</td>
<td>6</td>
<td>182</td>
</tr>
<tr>
<td>Utterance End Frame</td>
<td>85</td>
<td>243</td>
</tr>
</tbody>
</table>

Table 6.1: Array Containing Utterance Information

6.3.6 Delay Estimation and Compensation

The corresponding Utterances identified in the previous section have undoubtedly undergone some form of delay during the transmission process. Effects such as propagation delay and codec delay are unavoidable in a communication network, while other effects such as buffer jitter may also be present. Negative delays may also be present. Due to the nature of the recording of these speech samples, an utterance may
occur earlier in the degraded sample than its corresponding utterance in the original sample. This is often due to time discrepancy in the activation of the recorders.

The instances of speech in the degraded speech sample must be realigned correctly with their corresponding utterances in the original sample. This will allow the original and degraded speech to be compared directly at a later stage in the algorithm. Both constant delay and variable delay must be accounted for. Two different methods are used in this algorithm to account for these separate delays types. ‘Utterance Matching’ and ‘Utterance Splitting’ are used to account for constant and variable delay respectively. To account for both types of delay, both of these methods will be combined into a single algorithm called ‘Utterance Splitting/Matching’

Once the Utterances of the two speech samples are aligned satisfactorily, a new array of 32ms frames is created for each speech sample. These arrays contain only frames which are deemed to contain speech information. The array representing the degraded speech sample is ordered in the manner defined by the results of the delay estimation and compensation algorithm described here. The result of this Delay Estimation and Compensation will be two arrays of 32ms frames which will contain only speech information. Each 32ms frame in the original sample array will have a corresponding 32ms frame in the degraded array with which it can be compared with directly. This will facilitate the evaluation of the distortions present in the degraded sample.

6.3.6.1 Constant Delay Estimation and Compensation

The constant delay present between the Utterances of the two speech samples can be identified using a cross-correlation based method. Once the constant delay is identified, the two Utterance of the degraded sample can be aligned to their corresponding utterance in the original sample in the time domain.

It is assumed that the delay characteristic of the communication system under test is not such that it may invert the order of the utterances. During the development of the PESQ QoV estimation algorithm field trials for the measurement of delay variation in
both VoIP and PSTN/IP Gateway networks were carried out. It was found that VoIP based
network suffer the largest magnitude delay variations, but even these show a very
low occurrence of variations of more than 100ms. (Rix et al. 2002) This is far below the
time period required to be defined as a utterance.

To compare the likeness of two corresponding utterances (one from each speech
sample), both are entered into the cross-correlation algorithm. The utterances are entered
into the algorithm as a series of frequency domain representations of 32ms frames of each
sample. The cross-correlation operation is carried out on each pair of corresponding
frames (one from each utterance under test). Once each pair of frames has being tested
the result is a series of cross-correlation values. The mean of the magnitude of each of
these correlation values is then found to give a final correlation value. This correlation
coefficient represents the similarity of the two utterances.

The cross-correlation algorithm requires its input signals to be of equal length. If the
corresponding utterances are unequal in length, the end of the shorter utterance is padded
with zeros.

As this operation is being carried out for delay estimation purposes, the utterances
must be time shifted with respect to one another. All time shifting is carried out on the
degraded sample. The cross-correlation is first carried out with the degraded sample time-
shifted by -10 32ms frames. The resulting correlation value is then stored along with its
corresponding time-shift.

This process is repeated for all time shifts of the degraded signal from -9 to +10 32ms
frames. ±10 frames will allow the algorithm to account for delays of up to 320ms. The
time shift that results in the highest correlation result is deemed the best estimation of the
delay experienced by the utterance.

Figure 6.6 shows two representations of corresponding speech samples labeled
‘Original’ and ‘Degraded’. Both plots depict the mean power of each 32ms frame of each
speech sample presented along with the corresponding frame number. The Degraded speech sample is the result of transmission of the Original sample through the VoIP network simulator described in Section 7.5. It is clear from this figure that a constant delay, approximately equivalent to 5 32ms frames, has occurred during transmission.

![Figure 6.6: An Instance of Constant Delay](image)

By providing the Utterance Matching algorithm with the Utterance information found by the Utterance Identification algorithm, it is hoped that this constant delay can be identified and eliminated. The results are shown in Figure 6.7, where all frames deemed to contain silence have been removed as per the algorithm description.

As can be seen from Figure 6.7, the algorithm has accomplished its objectives very well. By examining the peaks and troughs present in both plots, it can be seen that the constant delay has been removed and the samples are now aligned correctly.
testing was carried out in a similar fashion for another 49 sets of corresponding original and degraded speech samples. Each result was checked subjectively by examining the plots of the resulting arrays, similar to that shown in Figure 6.7. In all but two instances the Utterance Matching algorithm was found to perform as designed. In the instances where the algorithm was deemed to have failed, the degraded speech sample was found to be extremely highly degraded. In such cases even perfectly aligned Utterances would generate a low correlation value. This makes it extremely difficult for the algorithm to discriminate between Utterance mismatches and correct alignments.

![Original Speech Sample](image)

![Degraded Speech Sample](image)

**Figure 6.7: Results of Utterance Matching to Eliminate Constant Delay**

Further testing of those speech sample pairs that caused the algorithm to fail found that a misaligned Utterance produced distortion measure results similar to that of a correctly aligned but highly distorted Utterance. For this reason the algorithms 4% fail rate was found to be acceptably, noting that fails only occurred during instances of heavy distortion.
6.3.6.2 Variable Delay Estimation

In the QoV estimation tool developed as part of this thesis, only variable delay occurring during instances of speech are dealt with. Delay variations during silence generally go unnoticed by the user. (Rix & Hollier 2000) If a delay variation occurs during speech, one of two distortion types can occur. If the delay time increases, the current packet arrives a short time later than the previous. Silence is inserted during the gap between these packets. The result of this is a distinct break in the speech. If the delay time decreases slightly it is most-often accounted for by the system buffer. If the decrease in delay is large enough, a packet may arrive before a packet that was transmitted before it. When this occurs, the order of the packets in question is reversed leading to audible distortion in the speech.

To locate variable delay during speech a 'trial and error' method incorporating Utterance Splitting is used. From the Utterance Matching described in the previous section, each utterance is divided into two equal parts. Now these two Utterance Segments are treated as separate Utterances. The steps described in the previous Utterance Matching section are repeated to match the newly created Utterances. In this way delay estimation is performed for each of the parts of the Utterance Segment.

If the resulting delay estimations of the two Utterance Segments are found to be equal (i.e. both may be found to have experienced a delay of +3 frames) then no delay variation is said to have occurred. This leads to a similar final correlation value as found previously with the un-split Utterance.

The Utterance Segments are only retained if the dissection and subsequent realignment of the Utterance has significantly increased the final correlation value. For this algorithm an increase of greater than 10% is said to be significant. This value provides a good compromise between efficiency and accuracy vital for lightweight implementation of the algorithm. If the increase is not greater than 10%, the two Utterance Segments are reassembled and the Utterance is returned to its original whole state.
Table 6.2: Utterance Match Array

When a variation in delay is identified the Utterance segment alignment information is stored in an array, as in Table 6.2. These Utterance Segments are then again split into two equal segments. The resulting segments are investigated for the presence of variable delay using the Utterance Splitting/Matching algorithm as described above. If no delay variation is found, then the latest Utterance split is discarded, leaving the overall Utterance in just two Segments. If a variation in delay is found, the four segments are maintained. Their respective alignment information is recorded and these segments are again split and investigated for further instances of variable delay.

This process of splitting and variable delay investigation is repeated until one of two stop conditions is met. The process will stop if no delay variations are found or the number of 32ms frames per Utterance Segment falls below a predefined threshold.

6.3.6.3 Error Limitation

When the number of blocks in the Utterance Segments is very low, mismatching of segments becomes very prominent. When this occurs it is possible that the algorithm could switch the order of the two similar sounds. For example, if the vowel sound “aaa” occurs twice in a single Utterance, the algorithm may mistakenly align the first instance of “aaa” in the degraded speech sample with the second instance in the original sample. The second instance in the degraded sample may then be aligned with the first “aaa” of the original sample. The result of a switch in the position would be an erroneous identification of variable delay. This may lead to major errors later in the algorithm when each 32ms frame of the original speech sample is compared with its corresponding frame in the degraded speech sample.
These error types would most-often occur when an utterance segment contains only the particular speech sound in question and no other speech information. To avoid such errors a threshold for the minimum size of an utterance segment was set to 9 frames. This segment size corresponds to the definition of minimum duration of an utterance as outlined earlier in this chapter. Subjective testing showed that this segment size eliminated the problems described.

![Original Speech Sample](image1)

![Degraded Speech Sample](image2)

**Figure 6.8: An Instance of both Constant and Variable Delay**

At this point the portions of the degraded speech sample containing speech are appropriately aligned to the corresponding sections of the original speech sample. All frames which do not contain speech data are discarded. Each 32ms frame of the original sample can now be compared directly to a corresponding 32ms frames which represents the degraded version of itself. This will allow the speech contained in the samples to be compared accurately. In this manner the distortions present in the degraded speech sample can be identified and evaluated.
Figure 6.8 shows two speech samples, one an original speech sample and the other its degraded equivalent. It is clear from the plot that the signal has been affected by both constant and variable delay during its transmission. The area indicated by the red marker is the result of a positive change in the delay time. The positive change in delay time is identifiable here by the insertion of silence into the Utterance. The Utterance data generated by the Utterance Identification algorithm was provided to the Utterance Splitting/Matching algorithm. The resulting Utterance alignment is shown in Figure 6.9.

Figure 6.9: Results of Utterance Splitting/Matching to Eliminate All Delay

It can be seen from Figure 6.9 that the Utterance Splitting/Matching algorithm has performed well and correctly re-aligned the Utterance segments. Again this can be seen from subjective examination of the peaks and troughs of the respective plots. Examination of the Utterance matching array, Table 6.2, shows that a constant delay corresponding to 2 32ms frames occurred for both Utterances of the sample. A positive delay variation of 4 frames was also identified during the first Utterance of the speech sample pair.
Similar testing was performed for 49 other Original/Degraded speech sample pairs. The results of each were tested by subjective means, as above, with only 3 instances deemed to have accounted for constant or variable delay in error. On the 3 occasions when the algorithm failed to successfully identify all delays correctly, it was found that the degraded speech sample was highly degraded. As before, these samples would have generated very low correlation values even when the correct alignment was identified. This makes it very difficult for the algorithm to identify a correct alignment as opposed to a misalignment.

![Original Speech Sample](image1)

![Degraded Speech Sample](image2)

**Figure 6.10: Example of Misalignment**

Further examination of those speech sample pairs which were misaligned by the algorithm showed that the misalignment had little effect on the end result of the distortion measures. Due to the heavy distortion within the degraded speech sample, the misaligned Utterance Segment proved to generate similar distortion magnitudes to those of Utterances Segments when 'manually' aligned correctly. An example of such a degraded speech sample is shown in Figure 6.10. Here much of the speech information has been
lost during transmission. The algorithm has attempted to align the information present but has failed. The resulting match list is given in Table 6.3

<table>
<thead>
<tr>
<th>Original Speech Sample</th>
<th>Utterance 1</th>
<th>Utterance 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utterance Start Frame</td>
<td>8</td>
<td>48</td>
</tr>
<tr>
<td>Utterance End Frame</td>
<td>45</td>
<td>85</td>
</tr>
<tr>
<td>Degraded Speech Sample</td>
<td>Seg 1</td>
<td>Seg 2</td>
</tr>
<tr>
<td>Utterance Start Frame</td>
<td>8</td>
<td>46</td>
</tr>
<tr>
<td>Utterance End Frame</td>
<td>45</td>
<td>85</td>
</tr>
</tbody>
</table>

Table 6.3: Utterance Match Array

Subjective evaluation proved that this utterance alignment proved to be in error, but the QoV estimation to be quite accurate with a score of 0.7292 compared with a value of 1.136 predicted by PESQ.

This QoV estimation algorithm will base its QoV score solely upon levels of distortion perceived. Values representing the identified delays will not be used in the generation of the QoV score. Therefore it is acceptable for the Utterance Splitting/Matching algorithm to fail during instances of heavy distortion.

6.3.7 16-Bit Linear PCM to Sound Pressure Level

The Decibel (dB) is a widely used logarithmic unit measure of a physical quantity. It is very commonly used in the fields of acoustics. In acoustics it can be used to quantify the magnitude of a signal relative to Absolute Threshold of Hearing (ATH). For application in this QoV estimation algorithm it is required to convert the Power of the audio signals, currently stored as PCM-based integers, to a Sound Pressure Level (S.P.L.) in dB. This is required for the accurate implementation of the perceptual loudness measure.

Digitally recorded speech is stored in integer format, most commonly 16-bit linear PCM. Those speech samples supplied by the ITU-T are of this format. The level at which the material is then presented to a listener is mainly governed by the user set parameters.
of the playback system. ITU-T P.830 recommends a playback level of 79dB (ITU-T, 1996(c)). Therefore the level of both signals is assumed to be 79dB and both signals are normalised to that level. (Beerends et al. 2002)

\[
S.P.L. = 10 \log_{10} \left( \frac{\text{INTEGER VALUE}}{\text{REFERENCE}} \right) \tag{6.3}
\]

![Frequency Domain Representation of 32ms Frame with Power Given in PCM Based Integer](image1)

![Frequency Domain Representation of 32ms Frame with Power Given in Decibels](image2)

**Figure 6.11: Conversion from Linear PCM to Decibels Measure**

From this the reference level is calculated by dividing the maximum integer value in the sample by \(10^{7.9}\). This reference level corresponds to the ATH. Once the ATH is calculated the conversion from Linear 16-bit PCM values to decibels is carried out for each frequency component of each 32ms frame using Eq. 6.3. An example of the
resulting conversion is shown in Figure 6.11, where the uppermost plot depicts a frequency domain representation of a 32ms frame of speech data with PCM-based power data. The lower plot presents a representation of the same 32ms frame with the spectral power data presented in decibels.

6.3.8 Conversion to Perceptual Domain

To give an accurate representation of the perceived distortions present in the degraded signal, a perceptual representation of the speech signal must be generated. Each 32ms frame of each sample currently contains speech information in the form of frequency (Hz) and corresponding spectral power (dB) data. The perceptual representation will describe how this information would be perceived by the human auditory system. This is accomplished by converting the analytical measure of frequency and spectral power to the measures of perceived pitch and loudness. These conversions are outlined in detail in Section 2.3, which deals with the operation of the human auditory system.

In Section 5.6 an ‘All-in-One’ A.N.N. model of the perception of sound by the human auditory system was developed. This A.N.N. model is used in this Intrusive QoV estimation system. The inputs supplied to the A.N.N. are the frequency (Hz) and S.P.L. (dB) value of each individual frequency component of each 32ms frame of both speech samples. From these inputs, the A.N.N. generates the perceived pitch and loudness. The resulting array contains the perceptual representation of each 32ms frame processed. The frequency scale of each frame is warped to perceived pitch (on the Bark scale). The S.P.L. in the dB scale is converted to the perceived loudness measure in Sone.
An example of the resulting conversion is shown in Figure 6.12. The uppermost plot is an analytical representation of the speech information contained in a single 32ms frame, while the lower plot is the perceptual-domain equivalent of the upper plot.

6.3.9 Disturbance Distance Calculation

The disturbance distance is a measure of the error margin between the original and degraded speech samples. The measure is computed from the perceptual representation of the speech samples which have been created in the preceding sections of this chapter.
The disturbance distance is calculated by a simple subtraction of the perceptual representations of the original speech sample from the degraded speech sample. The result is an error surface which describes the distortions which have occurred during the transmission of the audio signal. This subtraction is carried out upon 'Bark Bins' as opposed to each individual pitch component.

A Bark Bin is an amalgamation of the loudness levels in each Bark Band into a single Loudness level associated with that Bark Band. Each Loudness level falling within a specified Bark Band is summed and then the mean loudness within that Bark Band is found. (ITU-T, 2001)

This is carried out for each Bark Band in each 32ms frame for both samples. The loudness level for each Bark Band in each 32ms frame in the original sample is subtracted from the corresponding Bark Band loudness level in the degraded samples. The result is an error surface representative of the distortions present in the degraded samples.

\[ E(i,j) = Y_{\text{Bark\ Bin}(i,j)} - X_{\text{Bark\ bin}(i,j)} \]

The pseudo-code presented above outlines the operation which is performed for all \( i \) and \( j \) for each original, \( X \), and degraded, \( Y \), speech sample where \( i \) represents all pairs of corresponding 32ms frames and \( j \) represents each Bark Band of a 32ms frame. An example of the resulting error surface generated from a single pair of 32ms frames is shown in Figure 6.13
6.3.10 Cognitive Modeling

The cognitive modeling section of this algorithm accounts for the effects of simultaneous masking described in Chapter 2 and the variations of speech loudness levels throughout the samples.

To account for Simultaneous Masking an algorithm similar to that used in the PESQ QoV estimation algorithm (Section 4.5.7) is incorporated here. This is required to ensure that any distortion detected by this QoV estimation algorithm is of the nature that would actually be perceived by the listener. If it is found using this algorithm that a specific distortion would not be perceived by the listener, the measure of that distortion is set to zero. If the distortion is found to be of a nature that would be perceived by the listener,
the distortion is rescaled to account for the speech power in that section of the speech sample.

Firstly a Masking Factor is created for each individual bark band of each pair of corresponding 32ms frames. The Masking Factor, $M$, is the minimum loudness of two corresponding Bark Bands from the original and degraded speech samples raised to the power of 0.25. (Beerends et al. 2002)

$$M(i,j) = \left| \min \left( Y_{Bark\_Bin}(i,j), X_{Bark\_Bin}(i,j) \right) \right|^{0.25}$$

The masking factor of each specific Bark Band is then compared to the error measure for that Bark Band. If the magnitude of the distortion is found to be less than the masking factor this distortion is said to be unperceived by the listener. Therefore it will not be used in the estimation of the perceived QoV and is discarded. If the magnitude of the distortion is greater than that of the masking factor, it is retained but with its magnitude reduced by the masking factor. The algorithm is shown in pseudo-code below. (Beerends, J.G., A.P. Hekstra, A.W. Rix & P. Hollier, 2002)

```plaintext
if ( |E(i,j)| < |M(i,j)| )
    E(i,j) = 0;
elseif ( E(i,j) > M(i,j) )
    E(i,j) = E(i,j) - M(i,j)
elseif ( E(i,j) < -M(i,j) )
    E(i,j) = E(i,j) + M(i,j)
```

The error, $E$, is then divided up into two separate distortion measures, additive distortion, $\text{Add\_Dist}$, and subtractive distortion, $\text{Sub\_Dist}$.

For $I = 1:\text{Num\_Frames}$

IF $E(i,j) > 0$
Else

Add_Dist(i,j) = E(i,j)
Sub_Dist(i,j) = 0

End

Sub_Dist(i,j) = E(i,j)
Add_Dist(i,j) = 0

To facilitate the emphasis of the distortions which occur during frames with a low power density another method adopted from the PESQ QoV estimation algorithms is incorporated. This is required as distortions which occur during low-level but audible speech have a greater effect than those which occur during high-level speech. (Beerends et al. 2002)

The power of each 32ms frame is found by summing the power of each Bark Band in that frame. The power value is then warped by the addition of a constant and then scaled as shown below in the pseudo-code. This creates the 'Bark_power' measure which is then used to scale the distortion measure.

For ( j = 1:Num_Bark_bands )

Bark_Energy(i) = Bark_Energy(i) + X_Bark_Bin(i,j)

End

Bark_Power = (( Bark_Energy(j) + 10^-5 ) / 10^-7 ) ^-0.004

Perceived_Distortion(j) = Dist_Measure(j) * Bark_Power
Perceived_Add_Dist(j) = Add_Dist(j) * Bark_Power
Perceived_Sub_Dist(j) = Sub_Dist(j) * Bark_Power

Figure 6.14 demonstrates the results of the Cognitive modeling methods described. The uppermost plot presents the Disturbance Distance found in a single 32ms frame. The
lower plots shows the results of the conversion of this Disturbance Distance to the Perceived Disturbance Distance.

![Disturbance Distance](image)

![Perceived Disturbance Distance](image)

**Figure 6.14: Conversion from Disturbance Distance to Perceived Distortion**

The perceived distortion value of each individual 32ms frame is found by summing each distortion value of Bark Band of each frame.

The PESQ algorithm uses a complicated system of amalgamation and analysis to create suitable distortion measures which can be easily mapped to a QoV estimation in the range of -0.5 to 4.5. This process is described in detail in Section 4.6. The QoV estimation tool under development here will use an A.N.N. to perform the mapping of suitable distortion measures to a QoV estimation. The use of an A.N.N. in this stage will compensate for far less complex representations of the distortion measures than those required in the PESQ algorithm. Stages of PESQ such as dual disturbance distance aggregation stages and creation of an asymmetrical distortion measure are omitted with the aid of the A.N.N..

To generate values which will be representative of the distortion that occurred over the entire speech sample, the mean frame distortion is found. Also, the maximum frame
distortion values are found. Both of these values are generated for both Additive and Subtractive distortions.

The result is an output of four parameters; Mean_Additive_Distortion, Mean_Subtractive_Distortion, Max_Additive_Distortion and Max_Subtractive_Distortion. These four parameters will be the input for the A.N.N. described in Section 6.4.

6.4 Perceptual Mapping of Distance Measures to QoV Score

An A.N.N. is used to perform the transformation from the values generated in Section 6.3.9 to a QoV score in the range 1 to 5 as shown in Figure 6.15. The A.N.N. takes as its inputs four parameters that describe the distortions present in the degraded speech samples. It is hoped through the use of A.N.N.s the perceived effect of the measured distortions can be derived and an associated QoV score generated.

![Figure 6.15: Overview of the Operation of Decision Making A.N.N.](image-url)
6.4.1 A.N.N. Architecture Development

From analysis of the results of the Utterance Identification algorithm, instances of constant and variable delay can be identified and quantified. PESQ is the only intrusive objective QoV estimation algorithm standardised by the ITU that accounts for the presence of variable delay. This QoV estimation algorithm does not incorporate values representative of the delay variation into the calculation of the QoV score (ITU-T, 2001). Only values concerned with the distortion levels present in the degraded speech will be considered here for use in the calculation of the QoV score.

NeuroDimensions A.N.N. development environment, NeuroSolutions, was used to evaluate the relevance of the available inputs in the training set generated in Section 6.4.2. (NeuroDimension Inc. n.d.) This program is predominantly used in this work for its feature ‘Sensitivity about the Mean’. Once a suitable A.N.N. is developed and trained, this feature individually varies each input about its mean and logs the resulting variations in the output of the trained A.N.N.. The results of this provide a good estimation of which inputs are most relevant and which are least relevant in the production of the output of the A.N.N.. If an input is deemed to be significantly irrelevant it can be discarded as an input to the A.N.N..

An A.N.N. with four input nodes, 10 hidden layer nodes and a single output node was created in NeuroSolutions. This was then trained with 300 training vectors randomly selected from the set generated in Section 6.4.2 of this chapter. The resulting A.N.N. was then tested using the ‘Sensitivity About The Mean’ function. The Sensitivity about the Mean graph generated by NeuroSolutions is shown in Figure 6.16.
This graph shows that while the \textbf{Sub}_m (Mean Subtractive Distortion) input proved to have the greatest sensitivity, the output of the A.N.N. is sensitive to all four inputs. Therefore all four generated input parameters are used in the training and operation of the A.N.N..

A two layer feed-forward architecture was used for the A.N.N.. A tan-sigmoid function was used in the nodes of the hidden layer and a linear function in the output layer node accommodates the variations in the output value between 1 and 5. Four nodes were required in the input layer to take the values of \textbf{Mean Additive Distortion}, \textbf{Mean Subtractive Distortion}, \textbf{Max Additive Distortion} and \textbf{Max Subtractive Distortion}. The number of nodes in the hidden layer was decided during the training/testing methodology described in Section 5.2.1.

6.4.2 Training Data Generation

The training data for this application of A.N.N. was generated by applying the preprocessing algorithm outlined in Section 6.3. By supplying the preprocessing algorithm with a pair of corresponding original and degraded speech samples, the four required A.N.N. input values were generated.
The speech samples are taken from the ITU Series P Supplement 23 database of speech samples (ITU-T 1998). Degraded samples generated with the VoIP network simulator developed in Chapter 7 were also used.

The PESQ QoV estimation algorithm was used to generate accurate QoV scores for each of the sample sets used. This QoV score was used as the target value in each corresponding training vector. (Sun 2004, p. 74)

The resulting training set consists of 1176 Training vectors, each contain four inputs and the corresponding PESQ QoV score as the target value. 800 of these training vectors were generated by the packet-loss simulator detailed in Chapter 7. The remaining 376 training vectors were generated from the speech sample pairs provided by the ITU-T speech sample database (ITU-T 1998).

100 training vectors were chosen at random to be used as a validation set. The remaining 1076 training vectors were used in the training of the A.N.N.

6.4.3 Training / Testing

Seven A.N.N.s were developed with the number of hidden nodes varied from 3 to 15. Each configuration was examined for suitability using the methodology described in Section 5.2. As can be seen from Table 6.4, the error values for each A.N.N. configuration increase as the number of neurons decreases. Based upon the results shown in Figure 6.17, it was decided that an A.N.N. with less than 3 neurons in the hidden layer is insufficient to accomplish this function approximation.
Each trained A.N.N. configuration is also tested for its performance and for instances of over-fitting using the data contained in the validation set.

Table 6.4: Results of Training Procedure of A.N.N. Model for Mapping of Perceived Distortion Measure to QoV Score

<table>
<thead>
<tr>
<th>Neurons in Hidden Layer</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total No. of Weights</td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>75</td>
</tr>
<tr>
<td>Min M.S.E. of 10 1000 epoch Training Sessions</td>
<td>0.217434</td>
<td>0.192262</td>
<td>0.19364</td>
<td>0.160948</td>
<td>0.1765</td>
<td>0.162953</td>
<td>0.167729</td>
</tr>
<tr>
<td>Max error of Net with min MSE</td>
<td>1.3674</td>
<td>1.2852</td>
<td>1.2656</td>
<td>1.3092</td>
<td>1.1496</td>
<td>1.2986</td>
<td>1.2297</td>
</tr>
<tr>
<td>Best MSE result received</td>
<td>0.217267</td>
<td>0.191266</td>
<td>0.1979</td>
<td>0.162352</td>
<td>0.175861</td>
<td>0.162256</td>
<td>0.156359</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.4664</td>
<td>0.4376</td>
<td>0.4331</td>
<td>0.4273</td>
<td>0.4197</td>
<td>0.42351</td>
<td>0.3957</td>
</tr>
<tr>
<td>Max error</td>
<td>1.3674</td>
<td>1.2745</td>
<td>1.2440</td>
<td>1.2202</td>
<td>1.1494</td>
<td>1.3096</td>
<td>1.2244</td>
</tr>
<tr>
<td>Cross Correlation</td>
<td>0.7903</td>
<td>0.6162</td>
<td>0.6216</td>
<td>0.6276</td>
<td>0.6354</td>
<td>0.6483</td>
<td>0.6543</td>
</tr>
<tr>
<td>P Value</td>
<td>2.95E-154</td>
<td>4.62E-174</td>
<td>7.66E-177</td>
<td>1.78E-181</td>
<td>4.16E-187</td>
<td>1.30E-199</td>
<td>2.29E-205</td>
</tr>
</tbody>
</table>

6.4.4 Results

Table 6.4 presents the results of the performance of A.N.N.s with various numbers of neurons in the hidden layer. From the M.S.E. values it can be seen that a number of A.N.N.s have been created which closely match the training data. When tested with the training data, a network comprising of 15 nodes in the hidden layer was developed and trained to provide a M.S.E. of 0.15636, with a standard deviation of 0.3957 and cross...
correlation of more than 85%. Figure 6.18 shows a scatter plot of the results of the trained 15 neuron A.N.N. versus the expected results generated by the PESQ algorithm.

The results from testing carried out using the validation data set are shown in Table 6.5. The A.N.N. with 15 neurons in the hidden layer showed a relatively poor performance, with the cross correlation dropping to 80%. This show that this A.N.N. showed a poor level of generalisation of the function approximation. An A.N.N. with 8 neurons in the hidden layer generated the best results when tested with the validation data, with a M.S.E. of 0.1352, a standard deviation of 0.3701 and a cross-correlation of 84.26%. QoV scores generated by the 8 neuron A.N.N. for the validation data are plotted along with corresponding scores generated by the PESQ algorithm in Figure 6.19.

![Figure 6.18: PESQ Generated Target Values vs. A.N.N. Based Algorithm QoV Scores](image)

<table>
<thead>
<tr>
<th>Neurons in Hidden Layer</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>0.1925</td>
<td>0.1548</td>
<td>0.1412</td>
<td>0.1436</td>
<td>0.1352</td>
<td>0.1703</td>
<td>0.1624</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.4417</td>
<td>0.3961</td>
<td>0.3795</td>
<td>0.3616</td>
<td>0.3701</td>
<td>0.4156</td>
<td>0.4067</td>
</tr>
<tr>
<td>Max error</td>
<td>0.8748</td>
<td>1.0315</td>
<td>0.9922</td>
<td>1.0133</td>
<td>1.1158</td>
<td>1.0044</td>
<td>1.1464</td>
</tr>
<tr>
<td>Cross Correlation</td>
<td>0.7644</td>
<td>0.814</td>
<td>0.83239</td>
<td>0.8295</td>
<td>0.8426</td>
<td>0.7932</td>
<td>0.8043</td>
</tr>
<tr>
<td>P-Value</td>
<td>1.36E-14</td>
<td>1.07E-17</td>
<td>4.26E-19</td>
<td>7.23E-19</td>
<td>5.98E-20</td>
<td>2.73E-16</td>
<td>5.06E-17</td>
</tr>
</tbody>
</table>

Table 6.5: Validation Testing Results of A.N.N. for Mapping of Perceived Distortion Measure to QoV Score
6.5 Conclusions

This chapter has presented the development of an A.N.N. based intrusive objective QoV estimation algorithm. This algorithm is based upon existing algorithms such as PESQ. The developments of the pre-processing algorithm and the A.N.N.-based perceptual mapping have been described. The generation of the training and validation set for use in the development of the A.N.N. was also presented.

It has been shown that this approach has produced a usable QoV measure. Testing of this algorithm with unseen data has provided results which match the results of the ITU standardised method, PESQ, with a correlation value of more than 84% with a P-value of 5.98E-20.
Chapter 7

Non - Intrusive Quality of Voice
Estimation through Machine Learning

7.1 Introduction

Non-intrusive Quality of Voice estimation is vital in the field of Communication System Design. The ability to predict the QoV capabilities of the system before implementation in hardware greatly accelerates the development process. It also makes system development far more economical.

In the past, A.N.N.s have been used to objectively predict the QoV scores of packet-based communication systems (Sun 2004 & Mohamed, Cervantes & Afifi
In this chapter, the development of an A.N.N. based Non-Intrusive QoV Estimation system is presented. It will predict the QoV capabilities of a packet-based network based upon the operational parameters of the network.

With the rising popularity of the Internet, Voice over Internet Protocol (VoIP) networks have become the most widely used packet-based voice-carrying communication system. This QoV Estimation system is designed and developed to accommodate the impairment factors common in VoIP systems.

### 7.2 Requirements

This method operates as a standalone Non-intrusive QoV Estimation System, similar to that developed by L. Sun in her thesis “Speech Quality Prediction for Voice Over Internet Protocol Networks” (Sun 2004). This system uses an A.N.N. to estimate the performance of a VoIP communication system when presented with certain operating parameters of the VoIP network.

The operational parameters evaluated in this thesis are Conditional Packet-Loss Probability (CLP), Unconditional Packet-Loss Probability (ULP), Speaker Gender and Codec Type. The parameters chosen as viable candidates to model the QoV performance of the system they represent will be taken as the input to the A.N.N.. The output of the A.N.N. will represent an estimation of the QoV score that the system would yield under subjective testing.

For this project, this QoV estimation system is implemented with a VoIP network simulation, with packet-loss factors measured directly from this simulator. In real VoIP system implementations these factors can be discerned from the Real-Time Transport Protocol (RTP) header. RTP is the most common protocol used in time-constrained packet-based data transmission system such as VoIP. (Sun 2004, 16)
7.3 Evaluation of System Parameters with Regard to QoV Estimation

7.3.1 Codec Type

To evaluate the effects of various codec types on the QoV capabilities of a VoIP network a number of Speech samples were coded and decoded with a selection of Codecs. The Codecs examined were AMR, G726, G729a and G723.1.

The Adaptive Multi-rate Codec (AMR) is a codec widely used in GSM communications. It has been adopted as the standard GSM codec by 3GPP. AMR frames are of 20ms duration with a sampling frequency of 8000 kHz. This leads to a frame size of 160 samples. The codec may be used in 8 different modes, for bit rates of 4.75, 5.15, 5.9, 6.7, 7.4, 7.95, 10.2 and 12.2 Kbits/s. In this project only the 12.2 Kbit/s mode will be used. (Sun 2004, p. 20)

G.726 is an ITU-T codec standard, designed to provide 16, 24, 32 and 40 Kbits/s encoding. It was introduced in 1990 to replace both the G.721 and G.723 codecs. A frame size of 10ms is associated with the G.726 codec, with 8kHz sampling generating 80 samples per frame. (ITU 1990) It is the primary codec used in international ‘trunk calls’ due to its low bandwidth requirements and is also specified to be the codec used on Digital Enhanced Cordless Telecommunications (DECT) by the ETSI standard (ETSI, 2007).

G.729a is a fixed bit-rate encoder of 8 Kbits/s and frame length of 10ms. It is an advancement upon the G.729 codec which was developed to lessen the computational burden of the algorithm. This reduction in computation has been noted as reducing the QoV compared to the G.729 codec. Due to its low bandwidth requirements it is often the codec of choice for VoIP systems. (ITU 1996(a))

G.723.1 is a dual rate codec which has been standardised by the ITU-T. It is capable of encoding speech at 5.3 and 6.4 Kbits/s. With a frame length of 30ms and 8kHz sampling frequency, frames are of 240 samples each. Again, due to its low bandwidth requirements it is also often used in VoIP systems. (ITU 2006)
Software implementations of these codecs were implemented. 12 degraded speech samples were generated from each codec by these means. These degraded samples were then used to evaluate the performance of the Codecs. These 12 samples consisted of 6 samples of Male speech and 6 samples of Female speech.

To estimate the QoV score of each pair of original and degraded speech samples in an economic fashion, the PESQ QoV Estimation algorithm was used. The generated degraded speech samples and the corresponding original speech samples were presented to the PESQ QoV estimation algorithm resulting in 48 QoV scores in the range 1-4.5.

A plot of the QoV scores generated by the various codecs is shown in Figure 7.1. From this plot it is clear that the choice of codec has a large effect on the QoV performance of the network. Therefore, the codec type is supplied to the A.N.N. to help to achieve an accurate QoV prediction for a VoIP system.

![Figure 7.1: Comparison of Codecs using the QoV Measure](image-url)
7.3.2 Speaker Gender

To evaluate the effect of the speaker gender upon the perceived QoV over a VoIP network, the results of Section 7.3.1 is used. Figure 7.1 shows the performance of 12 instances of each of the four codecs tested. The first 6 samples of each codec are generated using female speakers, while the last 6 use male speakers.

The QoV of the G.723.1 codec seems to decrease when presented with samples containing male speech. When examined it was found that the mean QoV score for the female speech samples coded using G.723.1 is 3.305 while the mean score for the male speech samples coded using G.723.1 is 3.066. The other codecs seem to perform equally for both Male and Female speakers. To achieve a truly accurate QoV estimation from this A.N.N. based method, the speaker gender will be included as an input to the net.

7.3.3 Unconditional Loss Probability (ULP)

The unconditional loss probability (ULP) of a packet is a potentially promising measure for the estimation of QoV as it represents the percentage of the transmitted data which arrives at the receiver end. To estimate the effect this factor has upon the QoV, a series of 30 samples were degraded by passing them through a packet loss simulator described in Section 7.5. A different ULP and Conditional Loss Probability (CLP) value was seeded for each sample. The percentage of packets lost during the processing of each sample was noted. The CLP was not noted. The resulting QoV for each sample was then measured using the PESQ algorithm as before.

Figure 7.2 displays with a plot of the PESQ generated QoV score versus ULP. A definite trend is present in the plot, with the PESQ generated QoV score decreasing steadily with the rise in ULP. A correlation coefficient of 0.8961 with a p-value of 2.2E-11 was found between the two data sets. Therefore, ULP is a valid parameter for use in Non-Intrusive estimating the QoV of a VoIP system.
7.3.4 Conditional Loss Probability (CLP)

The effect of the conditional loss probability (CLP) on the QoV capabilities of VoIP system was evaluated in the same manner as ULP in Section 7.3.3. The CLP generated by the packet loss simulator for each of 30 samples was noted and again the associated QoV score was found using the PESQ algorithm.
Figure 7.3 presents the resulting plot of QoV score versus CLP. No significant correlation between the two factors can be seen. When examined a correlation coefficient of 0.047 with a p-value of 0.7942 was found. It was concluded that with such a low correlation value and high p-value, CLP (on its own) cannot be used to predict the QoV of a VoIP system.

7.3.5. ULP and CLP

It has been shown in the previous section that CLP alone offers little direct linear correlation with associated QoV scores. In Sun’s thesis (Sun 2004) CLP is stated to be a useful factor in the Non-Intrusive estimation of QoV and is used in her A.N.N. based approach. Therefore, further investigations into the effects of CLP on the QoV of speech are made. The factors ULP and CLP are often used collectively to model the Packet-loss characteristics of a system.

Figure 7.4: ULP vs. CLP vs. QoV Score
To further analyse the combined effects of ULP and CLP 50 degraded speech samples were generated using the Packet-loss simulator developed. An original sample containing speech from a male speaker was used to generate the entire set of degraded samples. Seed values for ULP and CLP were changed for the generation of each degraded speech sample. The ULP and CLP values of each sample were noted. The QoV score of each degraded sample was again estimated by the PESQ algorithm.

A three-dimensional plot of the ULP, CLP and associated PESQ generated QoV score is shown in Figure 7.4. From this plot it is difficult to draw conclusions whether the combination of ULP and CLP could produce a more relevant measure of QoV than simply ULP on its own.

In a final attempt to ascertain whether CLP should be included as an input to the A.N.N. for Non-Intrusive QoV estimation, the NeuroSolutions program was employed as in section 6.4.1. Using this program, an A.N.N. can be created that will map the ULP and CLP values to a QoV estimation. Once trained, the relevance of each input to computing the output of the A.N.N. can then be tested. The ULP, CLP and an associated PESQ QoV score were used to generate 50 training vectors to train an A.N.N.. The ULP and CLP values were used as inputs to the A.N.N. with the PESQ QoV estimation being used as the target output.

Using NeuroSolutions, an A.N.N. with a single hidden layer comprising of three processing elements was created. It was trained to match the training set target values to a M.S.E. of less-than 0.008265. To determine the contribution from the CLP value in creating this output value a 'Sensitivity About the Mean' test was carried out.
The resulting 'Sensitivity About the Mean' chart generated by NeuroSolutions is shown in Figure 7.5. This gives 'Sensitivity About the Mean' values of 0.060272 and 0.361 for CLP and ULP respectively. With CLP having a value approximately one sixth that of ULP it shows that ULP is the dominant factor of the two in the calculation of QoV score. That said, the CLP value can still be considered a relevant factor in the calculation. For this reason two A.N.N.s will be developed and trained. One will included the measure of CLP along with the other previously established measures while the other will omit the CLP input.

Figure 7.6: Overview of Possible A.N.N. for Non-intrusive Objective QoV Estimation
7.4 Training Set Generation

The training set consists of a series of vectors each containing either three or four (as appropriate) input values and a single output value. The input values used represent the four parameters of the system which were chosen in the previous section; Conditional Loss Probability, Unconditional Loss Probability, Speaker Gender and Codec Type. The target value of this training set is a QoV score in the range 1 – 5, corresponding to the Absolute Category Rating (ACR) for speech quality.

In the past, A.N.N. non-intrusive QoV estimation algorithms were trained using target data generated by subjective means (Meky & Saadawi 1997). L. Sun established a method by which the current ITU-T QoV estimation standard, PESQ could be used to generate the target QoV scores (Sun 2004, p. 74). This method will be used here for the generation of target values.

The PESQ score resulting from each pair of original and degraded speech samples is associated with the parameters of the network which generated the degraded speech sample. This creates a complete training vector of four inputs and a single target output.

Precisely tracking the packet-loss occurring over a fully implemented, representative VoIP network can be difficult and often imprecise. It may also be difficult to generate speech samples representative of certain VoIP system conditions with a single VoIP system. Multiple VoIP network are required to generate samples of a wide variety of network conditions.

To generate representative speech samples of a system with known packet loss characteristics a VoIP system simulation tool is used. With such a system simulating the parameters such as conditional and unconditional loss probabilities can be 'seeded'. The actual occurrences of packet-loss can then be measured precisely from the packet-loss simulation section of the simulator. This is described in greater detail in Section 7.5.2.
Computer based, VoIP Network simulation provides a time and cost efficient method for generating a large number of speech samples of varying quality. The simulation is used to represent a wide variety of different system conditions by adjusting the input parameters to the system. The development of the VoIP Network Simulator is detailed in Section 7.5.

In total, 800 training vectors were generated using the VoIP network simulator in conjunction with the PESQ QoV estimation algorithm. 87.5% (700) of these vectors were used in the training process, while the remaining 12.5% (100) were held as a validation set.

7.5 VoIP Network Simulator

A VoIP system can be modeled using an Encoder, a packet-loss simulator and a decoder. This system will not account for any delays or transmission based distortions of speech data by the VoIP system. The main QoV impairment factors present in a VoIP system are distortions due to codec compression algorithms, packet-loss and jitter. Jitter, that is variations in delay time, can be almost eliminated with the correct optimization of the systems buffers. Those packets which fall outside the buffer zone are most-often dropped and are therefore incorporated into this packet-loss based model. (Clark 2001) Only distortions due to the operation of the codec compression algorithms and packet-loss are modeled.
7.5.1 Codec Types

Software implementations of the four audio codecs described in Section 7.3.1 were used within the VoIP Network Simulator. The option of forgoing the use of a codec was incorporated into the simulator.

7.5.2 Packet-Loss Simulation

Packet loss in a VoIP system can be described as ‘bursty’ losses. This means that packets are generally lost in groups, or ‘bursts’, as opposed to randomly occurring individual packet losses. The bursty nature of packet-loss in VoIP networks is due to the transient nature of these networks. Events such as network congestion and buffer overflow are among the most common reasons for bursty losses. To model these bursty losses both unconditional and conditional packet loss probabilities of the system to be modeled are provided to the packet loss simulator.

Unconditional loss probability (ULP) is the mean probability of any single packet being dropped while being transmitted by the system. The conditional loss probability (CLP) is the probability a specific packet will be dropped, given that the previous packet was dropped.

Bolot found that Packet-Loss over Internet connection could be represented using Markovian-based loss models such as Gilbert Models (Clark 2001). In the packet-loss simulator developed here, a Two-State Gilbert model was used to simulate the unconditional and conditional packet loss probabilities of a VoIP system. This method has been widely used in the past for this purpose (Yajnik et al. 1999 & Sun 2004). It has proven to provide a good approximation of the head of the loss distribution curve for VoIP networks (Sannek 2000, p. 24).
Figure 7.8 depicts the 2-state Gilbert model used to implement ULP & CLP in this packet loss simulator. The state ‘Successful’ represents a successfully transmitted packet. The state ‘Dropped’ represents a dropped packet. $P_{ss}$ represents the probability that a packet will be successfully transmitted if the previous packet was successfully transmitted. $P_{sd}$ represents the probability that a packet will be dropped if the previous packet was successfully transmitted. $P_{ss}$ is equal to $1 - P_{sd}$. $P_{dd}$ represents the probability that a packet will be dropped if the previous packet was dropped. $P_{dd}$ is equal to the CLP. $P_{ds}$ represents the probability that a packet will be successfully transmitted if the previous packet was dropped. $P_{ds}$ is equal to $1 - CLP$.

\[
P_s = (P_{ss} \times P_s) + (P_{ds} \times P_d) \quad (7.1)
\]

\[
P_s + P_d = 1 \quad (7.2)
\]

Eq. 7.1 and Eq. 7.2 shown the relationships between the probabilities of packet-loss occurring. $P_s$ is the overall probability of a successful transmission and $P_d$ is the overall probability of the packet being dropped, also known as ULP.

Eq. 7.3 presents an expression for $P_{sd}$ in terms of ULP and CLP. This expression can be derived directly from Eq. 7.1 and Eq. 7.2, as shown. From the expression for $P_{sd}$ an expression for $P_{ss}$ in terms of ULP and CLP can be derived as in Eq. 7.4
\[ P_D = 1 - \left[ (P_{SS} \times P_S) + (P_{DS} \times P_D) \right] = \left[ (P_{SD} \times P_S) + (P_{DD} \times P_D) \right] = \left[ (P_{SD} \times (1 - P_D)) + (P_{DD} \times P_D) \right] \]

\[
P_D = \frac{P_{SD}}{(P_{SD} + 1 - P_{DD})} = ULP = \frac{P_{SD}}{(P_{SD} + 1 - CLP)}
\]

\[
\Rightarrow P_{SD} = \frac{ULP - (CLP \times ULP)}{1 - ULP} \tag{7.3}
\]

\[
P_{SS} = 1 - P_{SD} \tag{7.4}
\]

The Gilbert model developed here takes the values ‘Current State’, ‘ULP’ and ‘CLP’ as its inputs. The ‘Current State’ value identifies whether the previous packet was transmitted successfully or dropped.

The model operates by generating a random number based upon a normal Gaussian distribution. The probability values generated in Eq. 7.3 and Eq. 7.4 are then mapped to a normal Gaussian curve. Depending on the ‘Current State’ of the Gilbert model the random number is compared to the appropriate mapped probability value. Based upon this comparison the new state is assigned. For example, if the previous packet was dropped, the random number would be compared to the Gaussian equivalent of \( P_{DS} \), \( G_\_P_{DS} \). If the random number is less than \( G_\_P_{DS} \), then the new state is equal to ‘Successfully Transmitted’.

The packet-loss parameter generated will not match the values of ULP and CLP supplied to the packet-loss simulator exactly. This is due to limited number of packets being processed by the system and the use of random number generation. Therefore the values of ULP and CLP provided are referred to as the ‘Seed’ values. The true values of the packet-loss probabilities are measured directly from the packet-loss simulator by monitoring and recording the outputs of the Gilbert Model. The resulting ‘true’ values for ULP and CLP are labeled ‘CLP(real)’ and ‘ULP(real)’.

Packet loss during silence has very little effect on the QoV score of a speech sample since it is not perceived by the listener. (Sun 2004, p. 94) A more accurate measure of
the packet-loss probabilities that do affect the perceived QoV is required. A measure of the occurrences of the packets-loss during voiced speech provides a more relevant measure from which to ascertain the QoV. This measure is found by applying a Voice Activity Detection (VAD) system to the Packet-loss simulator. The VAD system tags each speech packet as containing either speech information or silence. Once each packet is tagged, only dropped voiced blocks are counted to generate two new probability measures; CLP(voiced) and ULP(voiced). These measures present the packet-loss probabilities which will affect the QoV of the speech sample.

### 7.6 A.N.N. Architecture

A Feed-forward A.N.N. architecture with a single hidden layer was chosen for this QoV estimation system. This architecture was chosen based on the highly successful results achieved by L. Sun while using a similar architecture (Sun 2004). A tan-sigmoid function is used as the activation function for the hidden layer neurons. This facilitates the use of a Back-Propagation training algorithm to train the A.N.N.s. A linear function is used as the activation function for the neurons of the output layer that facilitates an output that can vary in the range 0-5.

As discussed in Section 7.3.5 two separate non-intrusive QoV estimation A.N.N.s are developed. The first will take three inputs, namely ULP, Codec Type and Speaker Gender while the second will take an additional forth input of CLP. The output layer consists of a single node as corresponding to a QoV score. The number of nodes in the hidden layer is chosen on a trial and error method during successive training/testing iteration.

### 7.7 Training/Testing

Eight A.N.N.s were developed for each of the two network configurations used, with the number of hidden nodes being varied from 2 to 20. Each configuration was examined for suitability using the methodology described in Section 5.2. From Section
7.8, it can be seen that the error values for the A.N.N.s increase as the number of neurons decreases. Based upon these results, shown in Figures 7.9 (with CLP measure) and 7.10 (without CLP measure), it was deemed that an A.N.N. with less than 2 neurons in the hidden layer would be insufficient to accomplish either function approximation. An overview of the training process is shown in Figure 7.9 in block diagram format.

Figure 7.9 & 7.10: M.S.E. Vs. Number of Neurons in Hidden Layer

Each trained A.N.N. configuration is also tested for instances of over-fitting and its performance on the data contained in the validation set.

Figure 7.11: Non-Intrusive Objective QoV Estimation A.N.N. Training Procedure
### Table 7.1: Results from Training of Non-Intrusive Objective QoV Estimation A.N.N. (Using CLP Measure)

#### 7.8 Results

Table 7.1 shows the results achieved during the investigation of the performance of various A.N.N. configurations which use the input of the CLP. It can be seen that a number A.N.N.s suitable for the non-intrusive objective estimation of the QoV measure were created. A network comprising of 20 nodes in the hidden layer was found to give the best performance on the training data. It provided a M.S.E. of 0.029, with a standard deviation of 0.1704 and a cross-correlation measure of 97.8% and a p-value of 0 with the target values of the training data set.

### Table 7.2: Results from Validation Testing of Non-intrusive Objective QoV Estimation A.N.N. (Using CLP Measure)

When tested with the validation set, the A.N.N. with 20 nodes in the hidden layer produced the highest correlation (0.9716) with the PESQ QoV estimations with a low P-value of 3.51E-63. A network comprising of 5 neurons in the hidden layer also proved to have an excellent performance. It provided the lowest maximum error value (0.5601) and also very acceptable mean error, cross-correlation and standard deviation results. These results are shown in Table 7.2. Figure 7.12 displays the output of this A.N.N. when presented with the validation set of input values versus the target output.
of the validation set. A high degree of correlation can be witnessed from this figure. This A.N.N. is an excellent option where reducing network size is important.

In Table 7.3 the performance of the A.N.N. configurations which did not use the CLP measure as an input are presented. It can be seen that a number A.N.N.s suitable for the non-intrusive objective estimation of the QoV measure were created. A network comprising of 20 nodes in the hidden layer was designed which was trained to provide a M.S.E. of 0.0495, with a standard deviation of 0.2145 and a cross-correlation measure of 96.49% with a p-value of 0. This was the best performing A.N.N. based upon results from the training data set which did not include the CLP measure.

![Figure 7.12: A.N.N. QoV Estimation vs. PESQ Generated Target Values](image)

<table>
<thead>
<tr>
<th>No. Nodes in Hidden layer</th>
<th>2</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total No. of Weights</td>
<td>8</td>
<td>16</td>
<td>20</td>
<td>24</td>
<td>32</td>
<td>40</td>
<td>60</td>
<td>80</td>
</tr>
<tr>
<td>Min M.S.E. of 10 1000 epoch Training Sessions</td>
<td>0.089165</td>
<td>708763</td>
<td>0.071825</td>
<td>0.071507</td>
<td>0.062787</td>
<td>0.066617</td>
<td>0.052758</td>
<td>0.48043</td>
</tr>
<tr>
<td>Max error of Net with min MSE</td>
<td>1.0179</td>
<td>0.9998</td>
<td>0.9804</td>
<td>1.001</td>
<td>0.8813</td>
<td>0.7945</td>
<td>0.8598</td>
<td>0.915</td>
</tr>
<tr>
<td>Best MSE result received</td>
<td>0.089164</td>
<td>0.070873</td>
<td>0.071755</td>
<td>0.071096</td>
<td>0.062536</td>
<td>0.066467</td>
<td>0.524179</td>
<td>0.045945</td>
</tr>
<tr>
<td>Max error</td>
<td>1.0179</td>
<td>0.9997</td>
<td>0.9794</td>
<td>1.0013</td>
<td>0.8932</td>
<td>0.7945</td>
<td>0.8511</td>
<td>0.9168</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.2988</td>
<td>0.2664</td>
<td>0.2681</td>
<td>0.2143</td>
<td>0.2503</td>
<td>0.258</td>
<td>0.2291</td>
<td>0.2145</td>
</tr>
<tr>
<td>Cross Corr.</td>
<td>0.9308</td>
<td>0.9454</td>
<td>0.9447</td>
<td>0.9452</td>
<td>0.952</td>
<td>0.9489</td>
<td>0.9599</td>
<td>0.9649</td>
</tr>
<tr>
<td>P Value</td>
<td>3.14E-307</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.3: Results from Training of Non-Intrusive Objective QoV Estimation
A.N.N. (Without CLP Measure)
The validation testing results from the A.N.N. configurations which do not use the CLP measure are shown in Table 7.4. The A.N.N. containing 15 neurons in the hidden layer proved to have the highest level of correlation with the expected results with a value of 0.9407 and a p-value of 8.35E-48. The network comprising of only 6 neurons in the hidden layer provided a comparably high correlation value of 0.9401 with a p-value of 3.35E-45 making this A.N.N. an excellent option where the nework size must be kept to a minimum.

<table>
<thead>
<tr>
<th>No. Nodes in Hidden layer</th>
<th>2</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Error</td>
<td>0.7807</td>
<td>0.6755</td>
<td>0.7143</td>
<td>0.6721</td>
<td>0.6799</td>
<td>0.7169</td>
<td>0.8471</td>
<td>0.7886</td>
</tr>
<tr>
<td>Mean Error</td>
<td>0.2573</td>
<td>0.2253</td>
<td>0.238</td>
<td>0.223</td>
<td>0.2261</td>
<td>0.2022</td>
<td>0.219</td>
<td>0.2412</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.3277</td>
<td>0.2793</td>
<td>0.2996</td>
<td>0.2767</td>
<td>0.2811</td>
<td>0.2895</td>
<td>0.2753</td>
<td>0.3055</td>
</tr>
<tr>
<td>Cross Correlation</td>
<td>0.9169</td>
<td>0.9393</td>
<td>0.9298</td>
<td>0.9401</td>
<td>0.938</td>
<td>0.9342</td>
<td>0.9407</td>
<td>0.9263</td>
</tr>
<tr>
<td>P-Values</td>
<td>7.09E41</td>
<td>2.62E47</td>
<td>2.51E44</td>
<td>3.35E45</td>
<td>6.86E47</td>
<td>1.18E45</td>
<td>8.35E48</td>
<td>2.44E43</td>
</tr>
</tbody>
</table>

Table 7.4: Results from Validation Testing of Non-Intrusive Objective QoV Estimation A.N.N. (Without CLP Measure)

Figure 7.13 presents a plot of the output of the A.N.N. containing 6 neurons in the hidden layer versus the expected values of the validation set.

Figure 7.13: A.N.N. QoV Estimation Vs. PESQ QoV Estimation
7.9 Conclusions

This chapter has presented an A.N.N. approach to non-intrusive objective QoV estimation. Two systems have been developed which have both proven to have the ability to estimate the QoV capabilities of a communication network. Firstly, an implementation of a system put forward by L. Sun was developed which required four inputs in order to predict a QoV measure. Secondly, an adapted system was developed which discarded one of these inputs while still maintaining the ability to predict a QoV measure. Both of these approaches have proved to be highly viable methods that show high levels of correlation with the established methods.

Training and testing of the A.N.N.s was carried out with target values generated by an intrusive QoV estimation algorithm (PESQ). This ensures that the resulting QoV scores generated by the A.N.N. implementation will have a high level of correlation with actual subjectively derived scores. The use of this algorithm has alleviated the necessity to carry out expensive and time-consuming subjective testing to generate the target QoV scores.

In the past non-intrusive QoV estimation methods using A.N.N. have used the four measures of Codec-Type, Speaker Gender, ULP and CLP as inputs to the network (Sun 2004). This chapter has shown that performance of the network has not been diminished by discarding the CLP measure as an input to the system. In fact a more efficient and lightweight non-intrusive A.N.N.-based QoV estimation system can be implemented with little cost with regard to accuracy of the system. This result may also have implications for other areas where ULP and CLP are used such as packet loss modeling.
Chapter 8

Discussion & Conclusions

The primary aim of this thesis was to develop an objective QoV estimation system which would incorporate the use of Machine Learning. As described in Chapter 4, objective QoV estimation algorithms have been successfully developed in the past and this project has taken inspiration from many of them. It was hoped that the application of A.N.N. techniques to the field of QoV estimation would improve both the mobility of the algorithms and reduce the development cost required.

During this project a number of specific areas were identified that warranted investigation. Firstly, Intrusive QoV Estimation was examined and a QoV estimation algorithm which incorporated the use of A.N.N.s was developed. The A.N.N. was used to map measures of the perceived distortion in a speech sample to a QoV score. The use of an A.N.N. in this capacity allowed for less sophisticated measures of distortion to be used to generate a QoV score. Features of the existing standardised intrusive QoV estimation algorithm, PESQ, such as dual disturbance distance aggregation stages and creation of an asymmetrical distortion measure are omitted in the
method developed. The absence of these features has been compensated for by the machine learning capabilities of the A.N.N. These techniques have been found to satisfactorily replace complex algorithmic distortion evaluation techniques and hence reduce the processing power required to operate the overall QoV estimation algorithm. This is of huge benefit to the communications industry when it comes to in-the-field QoS testing. When testing in-the-field logistics are a major issue and therefore equipment must be kept to a minimum. The implementation of a more lightweight QoV estimation algorithm will aid the reduction processing power required. This may allow the use of hand-held devices for QoV testing where before PC based equipment was necessary, making this algorithm more commercially viable than its predecessor.

During the development of the intrusive objective QoV estimation algorithm it was decided to investigate the modelling of the human auditory system using A.N.N.s. This was achieved successfully and the fully developed A.N.N. model of the human auditory system was incorporated successfully in the intrusive QoV estimation algorithm as shown in Chapter 6. The use of A.N.N.s in the modelling of the main features of the human perception of sound resulted in a continuous and reliable conversion from frequency and SPL to the perceptual measures of pitch and loudness. This has many benefits when it comes to the implementation of a system which makes use of these conversions. When using the accepted equations to perform these conversions, the value to be converted must be approximated by the nearest discrete input value outlined as part of these equations, as listed in Chapter 3. If the value which needs to be converted to the perceptual domain is not one of these discrete values these equations will not function. In such cases some processing must be carried out to approximate the value to be converted with one which is defined by the equation. The A.N.N. based method does not have the need to perform this extra processing as it can operate continuously. Therefore it has the ability to accept any value in the defined range and produce an accurate output value.

Finally, a system for Non-Intrusive QoV estimation was developed in Chapter 7. Through the use of A.N.N.s, two Non-Intrusive QoV Estimation systems were created and proved to operate with extremely low error rates. The first of these systems was an implementation of a previously developed system which used four analytical operational parameters of the system under test to estimate a QoV score.

Investigation into the importance of each of these parameters in the estimation of QoV led to an uncertainty over the role of the C.L.P. measure in this process. An alternative system was developed which uses only three operation parameters to estimate the QoV capabilities of the system, the
C.L.P. measure being omitted. It was found that the omission of the C.L.P measure caused only a slight reduction in the accuracy of the QoV estimation, compared with the previous system which included the C.L.P. measure. Of the parameters previously used to determine the QoV non-intrusively, CLP is arguably the most difficult to obtain as each packet being transmitted must be tracked throughout its transmission and reception during testing. The other parameters, ULP, speaker gender and codec type, can be found more easily. ULP can be found by transmitting a known amount of packets and then counting the number of packets received. Speaker gender can be specified by the system analyst and codec type is specified in the system design. By eliminating the CLP factor, this method not only reduces the amount of processing required to calculate the QoV score, but also the amount of processing required to provide the parameter values required for the QoV estimation.

Another interesting point which the omission of the CLP measure raises is related to the optimisation of VoIP systems. Many VoIP systems are optimised to perform at predefined CLP measures, based upon the perception that the CLP measure has a bearing on the perceived QoV. The results here show that the CLP measure has very little effect on the QoV capabilities of a system. This leads to the conclusion that the optimisation of a VoIP system to perform within a certain CLP range is of very little benefit to the resulting QoV capabilities of the system.

8.1 Recommendations for Further Research

There are a number of ways in which this thesis could be complimented by further research. One area would be the expansion of the A.N.N.-based non-intrusive objective QoV estimation algorithm described in Chapter 7. This algorithm was developed and tested with speech samples produced by native English speakers. Just as speaker gender plays a significant role in the perceived QoV capabilities of a system, so does speaker nationality and language spoken. This is due to different frequency range characteristics of different languages and accents. By identifying and assigning an appropriate identifier to speakers of specific nationalities, ethnic backgrounds or language spoken a more accurate QoV estimation algorithm may be created.

Another possible improvement to this algorithm would be to use RTP header files in the
estimation of variable delay and packet-loss in an objective QoV application. RTP is the predominant protocol in use in VoIP networks and the header files used contain much useful information. This header contains a 16-bit 'sequence number' which increments by 1 for each RTP packet sent. This value may be used to accurately detect packet-loss and identify variations in the transmission delay. (Schulzrinne et al. 1996)
Bibliography


164


International Telecommunications Union 2006, ‘G.732.1: Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s’ ITU-T Recommendation.


Livingstone, New York.

Ph.D. Thesis, University of Plymouth, Retrieved June 20, 2006, from
http://www.tech.plymouth.ac.uk/spmc/people/lfsun


Traunmuller, H 1990, ‘Analytical Expression for the Tonotopic Sensory Scale’, The
Journal of the Acoustical Society of America, Vol. 88, pp. 97-100

Wang, S, Sekey, A & Gersho, A, 1992, ‘an Objective Measure for Predicting Subjective
SAC-10, pp.819-829.

Werbos, P 1974, ‘Beyond Regression: New Tools For Prediction and Analysis in the

Record, Part 4, pp. 96-104.

of the Measuring Normalizing Block Technique’, IEEE Transactions on Speech and


Appendix A

Further Tables
<table>
<thead>
<tr>
<th>Centre Frequency (Hz)</th>
<th>Bandwidth (Hz)</th>
<th>Cut-off Frequency (Hz)</th>
<th>Critical Band Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>80</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>160</td>
<td>100</td>
<td>200</td>
<td>10</td>
</tr>
<tr>
<td>250</td>
<td>100</td>
<td>300</td>
<td>5</td>
</tr>
<tr>
<td>350</td>
<td>100</td>
<td>400</td>
<td>10</td>
</tr>
<tr>
<td>450</td>
<td>110</td>
<td>510</td>
<td>10</td>
</tr>
<tr>
<td>570</td>
<td>120</td>
<td>630</td>
<td>10</td>
</tr>
<tr>
<td>700</td>
<td>140</td>
<td>770</td>
<td>10</td>
</tr>
<tr>
<td>840</td>
<td>150</td>
<td>920</td>
<td>10</td>
</tr>
<tr>
<td>1000</td>
<td>160</td>
<td>1080</td>
<td>10</td>
</tr>
<tr>
<td>1170</td>
<td>190</td>
<td>1270</td>
<td>10</td>
</tr>
<tr>
<td>1370</td>
<td>210</td>
<td>1480</td>
<td>10</td>
</tr>
<tr>
<td>1600</td>
<td>240</td>
<td>1720</td>
<td>10</td>
</tr>
<tr>
<td>1850</td>
<td>280</td>
<td>2000</td>
<td>10</td>
</tr>
<tr>
<td>2150</td>
<td>320</td>
<td>2320</td>
<td>10</td>
</tr>
<tr>
<td>2500</td>
<td>380</td>
<td>2700</td>
<td>10</td>
</tr>
<tr>
<td>2900</td>
<td>450</td>
<td>3150</td>
<td>10</td>
</tr>
<tr>
<td>3400</td>
<td>550</td>
<td>3700</td>
<td>10</td>
</tr>
<tr>
<td>4000</td>
<td>700</td>
<td>4400</td>
<td>10</td>
</tr>
<tr>
<td>4800</td>
<td>900</td>
<td>5300</td>
<td>10</td>
</tr>
<tr>
<td>5800</td>
<td>1100</td>
<td>6400</td>
<td>10</td>
</tr>
<tr>
<td>7000</td>
<td>1300</td>
<td>7700</td>
<td>10</td>
</tr>
<tr>
<td>8500</td>
<td>1800</td>
<td>9500</td>
<td>10</td>
</tr>
<tr>
<td>10500</td>
<td>2500</td>
<td>12000</td>
<td>10</td>
</tr>
<tr>
<td>13500</td>
<td>3500</td>
<td>15500</td>
<td>10</td>
</tr>
</tbody>
</table>

Table A.1: Subdivision of the Audible Frequency Range into Critical Bands
(Zwicker 1961)
<table>
<thead>
<tr>
<th>Frequency, $f$ (Hz)</th>
<th>$\alpha_f$</th>
<th>$L_U$ (dB)</th>
<th>$T_f$ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.532</td>
<td>-31.6</td>
<td>78.5</td>
</tr>
<tr>
<td>25</td>
<td>0.506</td>
<td>-27.2</td>
<td>68.7</td>
</tr>
<tr>
<td>31.5</td>
<td>0.480</td>
<td>-23.0</td>
<td>59.5</td>
</tr>
<tr>
<td>40</td>
<td>0.455</td>
<td>-19.1</td>
<td>51.1</td>
</tr>
<tr>
<td>50</td>
<td>0.432</td>
<td>-15.9</td>
<td>44.0</td>
</tr>
<tr>
<td>63</td>
<td>0.409</td>
<td>-13.0</td>
<td>37.5</td>
</tr>
<tr>
<td>80</td>
<td>0.387</td>
<td>-10.3</td>
<td>31.5</td>
</tr>
<tr>
<td>100</td>
<td>0.367</td>
<td>-8.1</td>
<td>26.5</td>
</tr>
<tr>
<td>125</td>
<td>0.349</td>
<td>-6.2</td>
<td>22.1</td>
</tr>
<tr>
<td>160</td>
<td>0.330</td>
<td>-4.5</td>
<td>17.9</td>
</tr>
<tr>
<td>200</td>
<td>0.315</td>
<td>-3.1</td>
<td>14.4</td>
</tr>
<tr>
<td>250</td>
<td>0.301</td>
<td>-2.0</td>
<td>11.4</td>
</tr>
<tr>
<td>315</td>
<td>0.288</td>
<td>-1.1</td>
<td>8.6</td>
</tr>
<tr>
<td>400</td>
<td>0.276</td>
<td>-0.4</td>
<td>6.2</td>
</tr>
<tr>
<td>500</td>
<td>0.267</td>
<td>0.0</td>
<td>4.4</td>
</tr>
<tr>
<td>630</td>
<td>0.259</td>
<td>0.3</td>
<td>3.0</td>
</tr>
<tr>
<td>800</td>
<td>0.253</td>
<td>0.5</td>
<td>2.2</td>
</tr>
<tr>
<td>1000</td>
<td>0.250</td>
<td>0.0</td>
<td>2.4</td>
</tr>
<tr>
<td>1250</td>
<td>0.246</td>
<td>-2.7</td>
<td>3.5</td>
</tr>
<tr>
<td>1600</td>
<td>0.244</td>
<td>-4.1</td>
<td>1.7</td>
</tr>
<tr>
<td>2000</td>
<td>0.243</td>
<td>-1.0</td>
<td>-1.3</td>
</tr>
<tr>
<td>2500</td>
<td>0.243</td>
<td>1.7</td>
<td>-4.2</td>
</tr>
<tr>
<td>3150</td>
<td>0.243</td>
<td>2.5</td>
<td>-6.0</td>
</tr>
<tr>
<td>4000</td>
<td>0.242</td>
<td>1.2</td>
<td>-5.4</td>
</tr>
<tr>
<td>5000</td>
<td>0.242</td>
<td>-2.1</td>
<td>-1.5</td>
</tr>
<tr>
<td>6300</td>
<td>0.245</td>
<td>-7.1</td>
<td>6.0</td>
</tr>
<tr>
<td>8000</td>
<td>0.254</td>
<td>-11.2</td>
<td>12.6</td>
</tr>
<tr>
<td>10000</td>
<td>0.271</td>
<td>-10.7</td>
<td>13.9</td>
</tr>
<tr>
<td>12500</td>
<td>0.301</td>
<td>-3.1</td>
<td>12.3</td>
</tr>
</tbody>
</table>

Table A.2: Look-Up Table for the Estimation of the Perceptual Loudness Measure (ISO 2003)
Table A.3: Results of Testing of A.N. implementation of the conversion from Frequency and S.P.L. to Perceived Loudness in Space

<table>
<thead>
<tr>
<th>No. Nodes in Hidden layer</th>
<th>15</th>
<th>12</th>
<th>10</th>
<th>8</th>
<th>6</th>
<th>4</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. M.S.E. of 100 Epoch Training Sessions</td>
<td>0.524899</td>
<td>0.959792</td>
<td>0.247998</td>
<td>0.086954</td>
<td>0.055946</td>
<td>0.036277</td>
<td>0.019458</td>
</tr>
<tr>
<td>Max. ERR of A.N. with mini MSE</td>
<td>0.09128</td>
<td>2.5146</td>
<td>1.9877</td>
<td>1.3968</td>
<td>1.0974</td>
<td>1.0095</td>
<td>1.0009</td>
</tr>
<tr>
<td>Best MSE received</td>
<td>0.000673</td>
<td>0.0078</td>
<td>0.0099</td>
<td>0.0019</td>
<td>0.0019</td>
<td>0.0029</td>
<td>0.0037</td>
</tr>
<tr>
<td>Best Correlation Coefficient</td>
<td>0.9995</td>
<td>0.9999</td>
<td>0.9999</td>
<td>0.9999</td>
<td>0.9999</td>
<td>0.9999</td>
<td>0.9999</td>
</tr>
<tr>
<td>P-Value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table A.4: Results of Testing of A.N. All-In-One Time Perception Model

<table>
<thead>
<tr>
<th>No. Nodes in Hidden layer</th>
<th>15</th>
<th>12</th>
<th>10</th>
<th>8</th>
<th>6</th>
<th>4</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. M.S.E. of 100 Epoch Training Sessions</td>
<td>0.524899</td>
<td>0.959792</td>
<td>0.247998</td>
<td>0.086954</td>
<td>0.055946</td>
<td>0.036277</td>
<td>0.019458</td>
</tr>
<tr>
<td>Max. ERR of A.N. with mini MSE</td>
<td>0.09128</td>
<td>2.5146</td>
<td>1.9877</td>
<td>1.3968</td>
<td>1.0974</td>
<td>1.0095</td>
<td>1.0009</td>
</tr>
<tr>
<td>Best MSE received</td>
<td>0.000673</td>
<td>0.0078</td>
<td>0.0099</td>
<td>0.0019</td>
<td>0.0019</td>
<td>0.0029</td>
<td>0.0037</td>
</tr>
<tr>
<td>Best Correlation Coefficient</td>
<td>0.9995</td>
<td>0.9999</td>
<td>0.9999</td>
<td>0.9999</td>
<td>0.9999</td>
<td>0.9999</td>
<td>0.9999</td>
</tr>
<tr>
<td>P-Value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

A-4
Appendix B

Intrusive Objective QoV Estimation Pre-Processing Algorithm Matlab Code (M-files)
B.1 size_stereo_align.m

function [x2, y2] = size_stereo_align(x, y);

[nx, mx] = size(x);
[ny, my] = size(y);

if mx > 1 % X From stereo to Mono
    x_mono = zeros(nx, 1);
    for k = 1:nx
        x_mono(k) = (x(k, 1) + x(k, 2))/2;
    end
    xl = x_mono;
else
    xl = x;
end

if my > 1 % y From stereo to Mono
    y_mono = zeros(ny, 1);
    for k = 1:ny
        y_mono(k) = (y(k, 1) + y(k, 2))/2;
    end
    yl = y_mono;
else
    yl = y;
end

% If x is longer than y, Pad y with zeros
if nx > ny
    y2 = zeros(nx, 1);
    y2(1:ny) = yl;
else
    y2 = yl;
end

% If y is longer than x, pad x with zeros
if ny > nx
    x2 = zeros(ny, 1);
    x2(1:nx) = xl;
else
    x2 = xl;
end
B.2 irs_filter_multi_fs.m

function [xf] = irs_filter_multi_fs(x,fs);

if fs == 8000  % Calculation of Frequency Representations
    f = [0 100/(fs/2) 125/(fs/2) 160/(fs/2) 200/(fs/2) 250/(fs/2) 300/(fs/2) 315/(fs/2) 400/(fs/2) 500/(fs/2) 600/(fs/2) 630/(fs/2) 800/(fs/2) 1000/(fs/2) 1250/(fs/2) 1600/(fs/2) 2000/(fs/2) 2500/(fs/2) 3000/(fs/2) 3150/(fs/2) 3500/(fs/2) 4000/(fs/2)];
    m = [0 0.000676 0.0034 0.019 0.0168 0.0933 0.141 0.148 0.2 0.234 0.25 0.257 0.3236 0.4266 0.59 0.89 1.0333 1.35 1.585 1.622 0.933 0.4467];
end

if fs > 8000  % Calculation of Frequency Representations
    f = [0 100/(fs/2) 125/(fs/2) 160/(fs/2) 200/(fs/2) 250/(fs/2) 300/(fs/2) 315/(fs/2) 400/(fs/2) 500/(fs/2) 600/(fs/2) 630/(fs/2) 800/(fs/2) 1000/(fs/2) 1250/(fs/2) 1600/(fs/2) 2000/(fs/2) 2500/(fs/2) 3000/(fs/2) 3150/(fs/2) 3500/(fs/2) 4000/(fs/2) 5000/(fs/2) 1];
    m = [0 0.000676 0.0034 0.019 0.0168 0.0933 0.141 0.148 0.2 0.234 0.25 0.257 0.3236 0.4266 0.59 0.89 1.0333 1.35 1.585 1.622 0.933 0.4467 0.1259 0];
end

[b,a] = yulewalk(8,f,m);  % Generation of Filter Coefficients
xf = filter(b,a,x);        % Implementation of IRS filter

B.3 split_ya_dig.m

function [x_split] = split_ya_dig(x,fs);

s32 = ceil(fs*0.032);  %No. of samples contained in 32ms segments of audio at FS sampling
[N,M] = size(x);

if N == 1  %Check Size
    N = M;
end

x_split = zeros(s32,ceil(N/s32));  % Creation of Array of 32ms blocks
j = 1;
for k = 1:s32:N  % Sequential Population of Array
    for i2 = 1:1:s32
        if k+i2-1 <= N
            x_split(i2,j) = x(k+i2-1);
        end
    end
    j = j+1;
end

B.4 short_term_fft.m

function [fft_x, fft_y, freq_values] = short_term_fft(x_split, y_split, fs);

    [Nx,Mx] = size(x_split);
    [Ny,My] = size(y_split);
    fft_x = zeros(Nx,Mx);
    fft_y = zeros(Ny,My);

    for k = 1:Mx  %FFT's blocks from start block to end Block
        [x,px] = fft_array(x_split(1:Nx,k));
        fft_x(1:Nx,k) = px;
    end

    for k = 1:My
        % All of y_split is FFT'd - matched to fft_x
        [y,py] = fft_array(y_split(1:Ny,k));
        fft_y(1:Ny,k) = py;
    end

    freq_values = zeros(Nx/2+1,1);  % Freq_values contains the
    % frequency values that correspond to the
    % power densities int fft_x and

    freq_values = zeros(Nx/2+1,1);
    for k = 1:Nx/2+1
        freq_values(k) = [(fs/Nx)*(k-1)];
    end

B.5 fft_array.m

function [X, Pxx] = fft_array(y);
    [N,M] = size(y);
    if N == 1
        N = M;
    end
    X = zeros(N, 1);
    X_even = zeros(N, 1);
X_odd = zeros(N, 1);

for k = 1:N/2
    for j = 1:(N/2)
        X_even(k) = X_even(k) + (y((2*j)-1) * exp((-2*pi*i*(j-1)*(k-1))/(N/2)));
        X_odd(k) = X_odd(k) + (y(2*j) * exp((-2*pi*i*(j-1)*(k-1))/(N/2)));
    end
    X(k) = X_even(k) + (exp((-2*pi*i*(k-1))/N) * (X_odd(k)));
X(N-k) = X(k);
end
Pxx = X.* conj(X) / N; % Calculation of Power Spectral Density

B.6 utterance_find_limited.m

function [utterance_list] = utterance_find_limited(fft_x);

ave_power = mean(fft_x); %Mean value for each 32ms Frame
[n,m] = size(ave_power);

more_utts = 1; %Monitoring the number of Utterances detected
less_utts = 0;

%if mean(ave_power) > 1
%    base_level_orig = 1e-9;
%elseif min(ave_power) > 0
%    base_level_orig = min(ave_power) + eps; %Assuming sample starts
%else
%    base_level_orig = 5e-9;
%end

lev = 990;
num_utts = 0;

while num Utts ~= 2 % Continue this algorithm until 2 (and only 2) utterances are identified
    if num_utts < 2 % If incorrect number of utterances declared, alter threshold of VAD accordingly.
        less_utts = less_utts + 1; % flag mis-classification
        lev = lev+10;
        base_level = base_level_orig * lev;
    elseif num_utts > 2
        more_utts = more_utts + 1; % flag mis-classification
lev = lev - 10;
base_level = base_level_orig * lev;

end

% Over 'base_level' is considered speech

utterance_list = zeros(2,10);
j = 1;
start = 0;
end_point = 0;

% Utterance = 300ms of noise containing less than 200ms of silence
% 9 blocks of noise containing less than 6 blocks of silence

for k = 1:1:m
    if (ave_power(k) >= base_level) && (start == 0) && (k < m-6)
        % Flag start of utterance when threshold is broken
        start = k;
    end
    if (k < m-6) && (start > 0) && (ave_power(k) < base_level) &&
        (ave_power(k+1) < base_level) && (ave_power(k+2) < base_level) &&
        (ave_power(k+3) < base_level) && (ave_power(k+4) < base_level) &&
        (ave_power(k+5) < base_level)
        end_point = k;
    end
    if (k >= m-5) && (start > 0) && (ave_power(k) < base_level)
        end_point = k;
    end
    if (k == m) && (start > 0) && (end_point == 0)
        end_point = m;
    end
    if ((end_point - start) >= 9) && (start > 0) && (end_point >
        0)
        utterance_list(1,j) = start;
        utterance_list(2,j) = end_point;
        j = j+1;
        start = 0;
        end_point = 0;
    end
end

utts_start = (find(utterance_list(1,1:10)));
num_utts = length(utts_start);

if (num_utts > 2) && (more_utts > 1) && (less_utts > 1)
    utterance_list;
    % sort out large gaps in degraded samples
gap = zeros(1,num_utts - 1);

% remove gaps to create only 2 utterances
for j = 1:(num_utts-1)
    gap(j) = (utterance_list(1,j+1) - utterance_list(2,j)) -
end

utterance_list_temp = zeros(2,2);
[max_gap,max_gap_ind] = max(gap);
utterance_list_temp(1,1) = utterance_list(1,1);
utterance_list_temp(2,2) = utterance_list(2,num_utts);
utterance_list_temp(1,2) = utterance_list(1,max_gap_ind + 1);
utterance_list_temp(2,1) = utterance_list(2,max_gap_ind);
utterance_list = utterance_list_temp;
num_utts = 2;
end

B.7 Utterance_match_Controlled_Split.m

function [utt_match_list, utterance_list_x, fft_x_utts, fft_y_utts, gap, gap_pos, gap_neg] =
utterance_match_controlled_split(fft_x,fft_y,utterance_listx,utterance_listy);

rmax = 0.0001;
r = 0;
splits = 0;
lx = 100;
[nfx,mfx] = size(fft_x);
[j,k_size] = find(utterance_listx);  % From Utterance_find.m

num_frames = 0;
for k = 1:1:max(k_size)  %Counting number of 32ms frames of speech
    num_frames = num_frames + (utterance_listx(2,k) -
    utterance_listx(1,k)) + 1;
end

while rmax > (r*1.10) && lx > 10  % Continue Loop While the previous
iteration provided a 10% improvement in correlation

B-7
greater than 10 framesnum

splits = splits + 1;

% and the block size is still divided.

t = rmax;

n = (splits/2)*length(k_size);
utterance_list_x = zeros(2,n);
utterance_list_y = zeros(2,n);

for k = 1:1:n

%Splitting of original sample utterance into utterance blocks

lx = utterance_listx(2,ceil(k/splits)) - utterance_listx(1,ceil(k/splits));

if (rem(k-1,splits) == 0)
    utterance_list_x(1,k) = utterance_listx(1,ceil(k/splits));
    utterance_list_x(2,k) = utterance_list_x(1,k) + (floor(lx/splits));
elseif k > 1
    utterance_list_x(1,k) = utterance_list_x(2,k-1) + 1;
    utterance_list_x(2,k) = utterance_list_x(1,k) + (floor(lx/splits));
end
if (rem(k,splits) == 0)
    utterance_list_x(2,k) = utterance_listx(2,k/splits);
end

end

for k = 1:1:n

%Splitting of degraded sample utterance into utterance blocks

ly = utterance_listy(2,ceil(k/splits)) - utterance_listy(1,ceil(k/splits));

if (rem(k-1,splits) == 0)
    utterance_list_y(1,k) = utterance_listy(1,ceil(k/splits));
    utterance_list_y(2,k) = utterance_list_y(1,k) + (floor(ly/splits));
elseif k > 1
    utterance_list_y(1,k) = utterance_list_y(2,k-1) + 1;
    utterance_list_y(2,k) = utterance_list_y(1,k) + (floor(ly/splits));
end
if (rem(k,splits) == 0)
    utterance_list_y(2,k) = utterance_listy(2,k/splits);
end

end
% Matching of blocks. Split lists from above passed in.
[utt_match_list_temp,rmax] =
utterance_match(fft_x,fft_y,utterance_list_x,utterance_list_y,
splits)

% Assign if the cross_cor is greater than last time
% Else discard
if rmax > (r*l.05)
    utt_match_list = utt_match_list_temp;
else
    splits = splits - 1;
    n = (splits/2)*length(k_size);
end

% Populate matched Utterance arrays
fft_x_utts = zeros(nfx,num_frames); %array for original utterances
fft_y_utts = zeros(nfx,num_frames); %array for degraded utterances
x = 1;
for j = 1:max(k_size)
% Populating fft_x_utts - with all utterances only
    xl = utterance_listx(2,j)-utterance_listx(l,j);
    fft_x_utts(1:nfx,x:x+xl) =
    fft_x(1:nfx,utterance_listx(1,j):utterance_listx(2,j));
    x = x + xl + 1;
end
y = 1;
for j = 1:n
% Populating fft_y_utts - with all utterances only - Silence Discarded
    yl = utt_match_list(2,j)-utt_match_list(1,j);
    if (utt_match_list(2,j) > 0) & (utt_match_list(1,j) > 0) %Only populate if successfully matched
        fft_y_utts(1:nfx,y:y+yl) =
        fft_y(1:nfx,utt_match_list(1,j):utt_match_list(2,j));
    end
    y = y + yl + 1;
end
% Identifies variable delay magnitudes

B-9
gap = zeros(1,(splits-1)*2);
gap_pos = 0;
gap_neg = 0;
if splits > 1

    for j = 1:splits-1
        gap(j) = (utt_match_list(1,j+1) - utt_match_list(2,j)) - 1; % '-1' because ideal gap is 1
        %For Utterance 1
        if gap(j) > 0 % Logs positive gaps
            gap_pos = gap_pos + gap(j);
        elseif gap(j) < 0 % Logs negative gaps / missing speech frames
            gap_neg = gap_neg + gap(j);
        end

        gap(j+splits-1) = (utt_match_list(1,splits+j+1) - utt_match_list(2,splits+j)) - 1;
        %For Utterance 2
        if gap(j+splits-1) > 0 % Logs positive gaps
            gap_pos = gap_pos + gap(j+splits-1);
        elseif gap(j+splits-1) < 0 % Logs negative gaps / missing speech frames
            gap_neg = gap_neg + gap(j+splits-1);
        end
    end
end

B.8 to_db.m

function [fft_x_db,fft_y_db] = to_db(fft_x,fft_y);

%fft_x/y are N * M matrices -- N samples, M 32ms blocks
%max(N*M array) outputs 1*M array = max of each column

[N,M] = size(fft_x);
refx = (max(max(fft_x(11:110,1:M)))) / (10^7.9)); %calculation of reference level
refy = (max(max(fft_y(11:110,1:M)))) / (10^7.9)); % One ref each for whole samples
fft_x_db = zeros(size(fft_x));
fft_y_db = fft_x_db;

for n = 1:1:M
    for k = 1:1:((N/2)+1)
calcx = fft_x(k,n)/(refx+eps);
if calcx < eps %Avoid divide by zero
    calcx = eps;
end

fft_x_db(k,n) = 10*log10(calcx);
calcy = fft_y(k,n)/(refy+eps);
if calcy < eps %Avoid divide by zero
    calcy = eps;
end
fft_y_db(k,n) = 10*log10(calcy);

end

B.9 perceptual_mapping.m

function [bark_values, fft_x_sone, fft_y_sone] = perceptual_mapping(fft_x_db, fft_y_db, freq_values, net);

%Perceptual mapping carried out by A.N.N. model of the human auditory system

[M,N] = size(freq_values);
if N == 1
    N = M;
end

[mx,nx] = size(fft_x_db);
fft_x_sone = zeros(N,nx);
fft_y_sone = fft_x_sone;

for m = 1:nx
    input(2,1:N) = freq_values; %Create input array for A.N.N.
    input(1,1:N) = fft_x_db(1:N,m);
    [out_x] = sim(net,input); %Present Input to A.N.N. and store results

    input(1,1:N) = fft_y_db(1:N,m); %Create input array for A.N.N.
    [out_y] = sim(net,input); %Present Input to A.N.N. and store results

end
for i = 1:N
  if out_x(2,i) < 0
    out_x(2,i) = 0;
  end

  if out_y(2,i) < 0
    out_y(2,i) = 0;
  end
end

fft_x_sone(1:N,m) = out_x(2,1:N);
fft_y_sone(1:N,m) = out_y(2,1:N);

bark_values(1:N,1) = out_x(1,1:N);

B.10 binned BSD disturbance distance mask.m

function [add_max, sub_max, bsd_add_fin_m, bsd_sub_fin_m, bsd_fin_m, bsd_n_m, bsd_m] = binning_dist_dist(fft_x_sone, fft_y_sone, bark_values);

[N,M] = size(fft_x_sone);
num_bins = floor(max(bark_values));
bark_count = zeros(num_bins,1);
bsd_m = zeros(M,1);
bsd_m_add = zeros(M,1);
bsd_m_sub = zeros(M,1);
energy_bark = zeros(M,1);

for j = 1:1:M
  num_bins = floor(max(bark_values));
  bin_x = zeros(num_bins,1);
  bin_y = zeros(num_bins,1);

  for k = 1:1:N
    bark = floor(bark_values(k)); %Creation of Bark Bin
    if (bark >= 1)
      if j == 1 % Just do once
        bark_count(bark) = bark_count(bark) + 1;
      end

      %add everything in specific Bark band
      bin_x(bark) = bin_x(bark) + fft_x_sone(k,j);
      bin_y(bark) = bin_y(bark) + fft_y_sone(k,j);

    end
  end
end

% Mean for each Bark Band
bin_x_scale = bin_x./bark_count;
bin_y_scale = bin_y./bark_count;

for k = 1:1:num_bins

    % The masking section from PESQ
    R(j,k) = bin_y_scale(k)-bin_x_scale(k);
    M = (min(bin_y_scale(k),bin_x_scale(k))*0.25);

    R2 = 0; % confirm Algorithm
    R_add = 0;
    R_sub = 0;

    if R(j,k) > M % If distortion greater than speech level
        R_add = R(j,k)-M;
    end

    if abs(R(j,k)) <= abs(M) % If distortion of less mag than speech level
        R2 = 0;
        R_add = 0;
        R_sub = 0;
    end

    if R(j,k) < (-M) % If mag of subtractive distortion greater than speech level
        R_sub = R(j,k)+M;
    end

    % Asymmetry Section
    A = (bin_y_scale(k) / (bin_x_scale(k)+eps)) ^ 1.2;
    if A < 3
        A = 0;
    elseif A > 12
        A = 12;
    end

    diff(k) = R_add + R_sub;
    bsd_m(j) = bsd_m(j) + (diff(k)^2);
    bsd_m_add(j) = bsd_m_add(j) + (R_add^2);
    bsd_m_sub(j) = bsd_m_sub(j) + (R_sub^2);
    energy_bark(j) = energy_bark(j) + (bin_x_scale(k)^2);
end

B-13
bsd_n_m(j) = bsd_m(j) * (((energy_bark(j) + (10^5)/(10^7)))^-0.04);  %Bark energy equation

% compensates for varying Frame energy
bsd_n_m_add(j) = bsd_m_add(j) * (((energy_bark(j) + (10^5)/(10^7)))^-0.04);
bsd_n_m_sub(j) = bsd_m_sub(j) * (((energy_bark(j) + (10^5)/(10^7)))^-0.04);

end

bsd_fin_m = mean(bsd_n_m);
bsd_add_fin_m = mean(bsd_n_m_add);
add_max = max(bsd_n_m_add)
sub_max = max(bsd_n_m_sub);
bsd_sub_fin_m = mean(bsd_n_m_sub);