Signal Subspace Speech

Enhancement with Adaptive Noise Estimation

Barry Commins

Submitted for the degree of M. Eng. Sc.
to the
National University of Ireland, Galway

September 2005

Department of Electronic and Computer Engineering, National University of Ireland, Galway

Supervisor: Mr. Fergal O’Malley
Abstract

Communication can be greatly hindered by noise. Speech enhancement seeks to eliminate noise in a variety of environments, the most prominent of which are telecommunications applications.

After over thirty years of research throughout the world, no perfect solution exists to this problem. The objective of this thesis is to develop a novel speech enhancement algorithm which offers superior noise reduction over current methods.

All speech enhancement systems suffer from distortion or residual noise due to imperfect noise removal. Some variations are more promising than others. One such method is signal subspace speech enhancement. However, this algorithm can only update the noise estimate when speech is absent, and suffers degradation in performance in many different noise types.

The system designed in this thesis takes the subspace method as its basis and develops a robust and accurate noise estimation algorithm that can update the noise estimate throughout the signal, not just in speech absence. Results show the new algorithm is an improvement over the other systems tested.
Acknowledgements

I would like to thank my supervisor Mr. Fergal O’Malley and all the staff of the electronic engineering department for all of their help throughout my time here. It was much appreciated.

Thanks also to everyone who helped out with testing the project, particularly Dr. Mohan Krishna Ranganathan and Shaun Porter for their advice.
Further thanks must go to all the engineering postgraduate students, especially Tom Bennett, Siora Coll, Frank Callaly, John Maher and Ailish O’Halloran, for keeping things interesting. Never a dull moment.

I would also like to sincerely thank my parents Damian and Denise, my brothers Mark and Niall and my sister Nicola for everything they have done for me and all the encouragement I needed.

Finally, a big thank you to Angela Dwane for her love and support and for telling me to work any time it seemed easier not to.

Thank you all,

Barry
Declaration

This thesis is presented in fulfilment of the requirements for the M. Eng. Sc. Degree. It is entirely my own work and has not been submitted to any other University or higher education institution, or for any other academic award in this University.

Signature: ______________________________

Barry Commins (September 2005)
# Table of Contents

Abstract.........................................................................................................................i  
Acknowledgements ..................................................................................................... ii  
Declaration.................................................................................................................... iii  

Chapter 1: An Introduction to Speech Enhancement..................................................1  
1.1 Applications of Speech Enhancement.................................................................2  
1.1.1 Telecommunications .....................................................................................2  
1.1.2 Automatic Speech Recognition ....................................................................3  
1.1.3 Electronic Hearing Aids .............................................................................3  
1.1.4 Audio Recording Restoration.....................................................................3  
1.2 Organisation of Thesis .....................................................................................4  

Chapter 2: Frequency Domain Speech Enhancement...............................................5  
2.1 Theoretical Aspects...........................................................................................5  
2.1.1 Additive Noise ............................................................................................5  
2.1.2 Discrete Fourier Transform .......................................................................6  
2.1.3 Windowing ....................................................................................................7  
2.2 Spectral Subtraction.........................................................................................8  
2.3 Drawbacks of Spectral Subtraction...................................................................9  
2.3.1 Musical Noise .............................................................................................9  
2.3.2 Distortion .....................................................................................................10  
2.4 Improvements to Spectral Subtraction............................................................11  
2.4.1 Wiener Filtering.........................................................................................11  
2.4.2 Spectral Subtraction with over-subtraction and spectral floor ...............12  
2.4.3 Spectral Subtraction with MMSE ..............................................................12  
2.4.4 Other methods............................................................................................13  
2.5 Perceptually motivated Spectral Subtraction...............................................13  
2.6 Summary .........................................................................................................17  

Chapter 3: Noise Estimation .....................................................................................18  
3.1 Two-Channel Method .....................................................................................18  
3.2 Voice Activity Detection.................................................................................19  
3.3 Implicit Methods Overview..............................................................................21  
3.3.1 Computationally Efficient Speech Enhancement by Spectral Minima Tracking in Subbands ..........................................................21  
3.3.2 Noise Power Spectral Density Based on Optimal Smoothing and Minimum Statistics ..........................................................23  
3.3.3 Subband Noise Estimation for Speech Enhancement using a Perceptual Wiener Filter ...............................................................................24  
3.4 Summary .........................................................................................................27
## Chapter 4: Signal Subspace Speech Enhancement ..............................................28

4.1 Subspace Theory ..........................................................................................28
4.1.1 Signal Model ..........................................................................................29
4.1.2 Noise Model ..........................................................................................30
4.1.3 Signal and Noise Subspaces ...................................................................30
4.2 Linear Signal Estimators ...........................................................................33
4.2.1 Time Domain Constrained Estimator ...................................................33
4.2.2 Spectral Domain Constrained Estimator ..............................................36
4.3 Pre-whitening of additive coloured noise ..................................................38
4.4 Rank Estimation .........................................................................................39
4.5 Noise Estimation .........................................................................................40
4.6 Summary ......................................................................................................40

## Chapter 5: Adaptive Noise Estimation for Subspace Speech Enhancement .................................................................41

5.1 Improved Subspace Speech Enhancement for Coloured Noise ...............42
5.2 Adaptive Noise Estimation for Subspace Speech Enhancement ...............45
5.3 Speech Enhancement using the Subspace Adaptive Noise Estimation Algorithm .............................................................................................................49
5.3.1 Time Domain Constraint Linear Estimator for Coloured Noise ..........49
5.3.2 Spectral Domain Constraint Linear Estimator for Coloured Noise .......50
5.4 System Implementation .............................................................................52
5.5 Summary .....................................................................................................52

## Chapter 6: Testing and Results ......................................................................54

6.1 Test Data ......................................................................................................54
6.2 Informal Testing ..........................................................................................55
6.2.1 Performance in White Noise .................................................................55
6.2.2 Performance in different Noise Types ..................................................57
6.2.2.1 Pink Noise .....................................................................................57
6.2.2.2 F16 Cockpit Noise ..........................................................................58
6.2.2.3 Jet Engine Noise .............................................................................60
6.2.2.4 Car Interior Noise ...........................................................................61
6.2.3 Informal Testing Conclusions ...............................................................62
6.3 Objective Testing .......................................................................................63
6.3.1 PESQ Algorithm Description ..............................................................63
6.3.2 Test Description ....................................................................................64
6.3.2.1 Spectral Subtraction Implementation ..............................................65
6.3.2.2 Perceptual Wiener Filter with Adaptive Noise Estimation Implementation ...............................................................65
6.3.2.3 Subspace Implementation ..............................................................65
6.3.3 PESQ Results .......................................................................................66
6.3.3.1 White Noise PESQ Scores ..............................................................67
6.3.3.2 Pink Noise PESQ Scores .................................................................69
6.3.3.3 F16 Cockpit Noise PESQ Scores ....................................................71
6.3.3.4 Jet Engine Noise PESQ Scores ......................................................72
Chapter 1: An Introduction to Speech Enhancement

Whenever speech is recorded by a microphone, unwanted noise is also recorded. This noise depends on the environment and can range from anything such as computer fan noise, car engine noise to factory floor noise. The goal of any speech enhancement system is to suppress or completely remove the unwanted noise while maintaining the quality and/or intelligibility of the speech. This has been an ongoing area of research since it was first proposed in 1979 by Boll in [1].

![Figure 1.1 Basic overview of additive noise](image1)

**Figure 1.1** Basic overview of additive noise

![Figure 1.2 Basic overview of a speech enhancement system](image2)

**Figure 1.2** Basic overview of a speech enhancement system
1.1 Applications of Speech Enhancement

Speech enhancement has several real world applications including telecommunications, electronic hearing aids and automatic speech recognition software. These are all important consumer applications, as improving the quality and intelligibility of speech vastly improves a users listening experience.

1.1.1 Telecommunications

One of the most common uses of speech enhancement systems is found in the area of telecommunications, and more specifically mobile or cellular telephony. Since the majority of mobile phone conversations take place in noisy environments, for example within automobiles, streets or public places, noise will inevitably be picked up along with the speech, making the conversation disturbing for the listener. A speech enhancement algorithm incorporated into the digital signal processor (DSP) chip on the phone could reduce the amount of noise and allow the conversation to take place more easily.

![Figure 1.3 Example of a Speech Enhancement System Application](image)

**Figure 1.3** Example of a Speech Enhancement System Application
1.1.2 Automatic Speech Recognition

Automatic Speech Recognition (ASR) has been a focus of research since the 1950s [2]. It is the process of using software to recognise words in spoken input and displaying the words as text, or allowing computer users to control programs by means of spoken commands. High intelligibility is the key requirement of these applications, as the systems must be able to distinguish between similar sounding words. Speech enhancement in this case is used as a front end to the ASR system to remove unwanted line noise or hiss from the input microphone before the ASR software attempts to recognise words.

1.1.3 Electronic Hearing Aids

Hearing aids consist of a microphone and amplifier with some DSP hardware. Like any other microphone, this is susceptible to picking up unwanted noise along with the speech. A good, robust speech enhancement algorithm programmed on the DSP chip can improve the users listening experience. In this situation the speech must be natural sounding and clear to reduce listener fatigue. A demonstration of the principle can be found at [3].

1.1.4 Audio Recording Restoration

Removal of static, or hiss, from recorded audio is another useful application for speech enhancement. The problem arises through use of poor quality recording equipment or a poor recording environment and is detrimental to the sound of the recording. An example of this application can be found in [4]. More specific examples of audio recording restoration concern the cases of older recordings such as analogue tapes, vinyl records, etc., which may have been distorted by noise due to the quality of the recording process. A more modern example involves the case of a recording using the internal microphone of an MP3 player, where the close proximity of a hard disk causes noise to become an issue.
Speech enhancement works extremely well in some of these situations, particularly where the noise is at a high Signal-to-Noise Ratio (SNR) and is confined to a narrow frequency range.

### 1.2 Organisation of Thesis

This thesis is divided into seven chapters, including this introduction to speech enhancement. Chapter 2 presents the theory of Spectral Subtraction, Discrete Fourier Transform (DFT) based speech enhancement, as proposed by Boll [1], which is the basis of the majority of speech enhancement algorithms. It then outlines some of the drawbacks of spectral subtraction and some more modern improvements on this system, including the Wiener filter.

Chapter 3 is dedicated to noise estimation methods, including 2-channel methods, voice activity detectors and adaptive noise estimators.

Chapter 4 describes Signal Subspace speech enhancement, proposed by Ephraim and Van Trees [5] in 1995. This uses a signal dependent transform and decomposes the signal into speech and noise subspaces.

Chapter 5 presents details of a novel adaptive noise estimation algorithm for signal subspace based speech enhancement, which compensates for some of the difficulties the regular signal subspace method has in dealing with noise that varies over time and with different types of coloured noise.

In chapter 6 comparative results between the new signal subspace with adaptive noise estimation algorithm and other methods are presented. These include subjective listening tests and objective benchmarking.

Finally, chapter 7 draws a number of conclusions and summarises the main findings of the research.
Chapter 2: Frequency Domain Speech Enhancement

The majority of speech enhancement systems use some variation on the frequency domain analysis techniques [1, 6-10]. In simplest terms this involves transforming noisy speech to the frequency domain using the Discrete Fourier transform (or Fast Fourier Transform) and subtracting an estimate of the noise spectrum from the noisy spectrum, leaving an approximation of the spectrum of the clean speech, which is then converted back to the time domain using an inverse Fourier transform. Problems arise due to incorrect approximation of the noise spectrum. These problems will be discussed in greater detail later in this thesis.

2.1 Theoretical Aspects

2.1.1 Additive Noise

Most speech enhancement systems assume that noise is additive and uncorrelated with speech. This is formulated using the following expression:

\[ y(t) = x(t) + w(t) \]  \hspace{1cm} (2.1)

where \( y \) is the noisy signal, \( x \) is the clean speech signal and \( w \) is the additive noise signal, all dependent on time \( t \).

In the frequency domain, this translates as

\[ Y(\omega) = X(\omega) + W(\omega) \]  \hspace{1cm} (2.2)

where \( Y \) is the noisy signal, \( X \) is the clean speech signal, and \( W \) is the additive noise, all as a function of frequency \( \omega \).
2.1.2 Discrete Fourier Transform

The Discrete Fourier Transform (DFT) converts signals from the time domain to the frequency domain. Since noise is normally confined to a relatively narrow frequency range, this transformation allows a noise reduction algorithm to identify which frequencies contain noise and which do not. The frequencies which contain noise are then removed or attenuated.

![Clean Speech in Time Domain](image1)

![Noisy Speech in Time Domain](image2)

Figure 2.1 Clean and noisy speech samples in the time and frequency domains

From the diagram above it is easy to see the importance of the DFT. In the time domain there is no clear distinction between the speech and the noise, but in the frequency domain the difference is more obvious. The noise frequencies are circled above to show the difference between the clean and noisy signals.

The Fourier transform is formulated as follows:

\[
X(e^{j\omega}) = \sum_{t=0}^{L-1} x(t) e^{-j\omega t} \tag{2.3}
\]

where L is the window length.
The inverse Fourier transform is given by:

\[ x(t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega t} d\omega \]  \hspace{1cm} (2.4)

### 2.1.3 Windowing

Speech is quasi-stationary meaning it is possible to analyse over short segments, thus the signal must be buffered into frames, usually in the region of 10–30 milliseconds. These frames are windowed to allow identical reconstruction when no noise reduction takes place, i.e. no distortion or discontinuity occurs as a result of the windowing process. A Hanning window is chosen as being appropriate to the task, and is given by:

\[ w(k+1) = 0.5 \left(1 - \cos \left(2\pi \frac{k}{n-1}\right)\right), \quad k = 0, \ldots, n-1 \]  \hspace{1cm} (2.5)

These frames are 50% overlapped to avoid end-point discontinuity.

![Figure 2.2 Speech samples buffered with overlapping Hanning windows](image-url)
2.2 Spectral Subtraction

The earliest and most commonly used method of speech enhancement is magnitude spectral subtraction, [1]. As discussed earlier, speech and noise are assumed to be additive and uncorrelated, therefore, if an estimate of the noise spectrum can be found for a particular frame of a noisy speech signal, then an estimate of the clean speech signal can be calculated by subtracting it from the noisy signal as described by the following expression:

$$\hat{X}(\omega) = Y(\omega) - \hat{W}(\omega)$$  \hspace{1cm} (2.6)

$\hat{X}(\omega)$ is the estimate of the clean frequency spectrum for a given frame, $Y(\omega)$ is the noisy spectrum for that frame, and $\hat{W}(\omega)$ is the noise spectrum estimate. An estimate of the clean speech is recovered by applying the inverse DFT to $\hat{X}(\omega)$, to give $\hat{x}(t)$. Since the human ear is relatively insensitive to phase, the phase angle of the noisy signal can be used when reconstructing the speech.

This is quite a computationally simple system, however noise estimation is a non-trivial problem, still without an optimal solution after almost thirty years of research. There are many different methods of noise estimation, some of which will be discussed in the next chapter.
2.3 Drawbacks of Spectral Subtraction

Spectral Subtraction may be fairly straight-forward to implement, and although reducing the noise significantly, it has some severe drawbacks. It is clear that the effectiveness of spectral subtraction is heavily dependant on accurate noise estimation, which is a difficult task to achieve in most conditions. When the noise estimate is less than perfect, two major problems occur, musical noise and distortion. These are discussed in the following sections.

2.3.1 Musical Noise

Musical noise occurs when random short sinusoids, which are tone-like in sound, are created due to flaws in the noise estimate, making the noise removal imperfect, [6].

\[ e = \hat{X}(\omega) - X(\omega) \] (2.7)

\( e \) is the error, the difference between the clean estimate \( \hat{X}(\omega) \) and the actual clean signal \( X(\omega) \).

Musical noise artefacts are randomly distributed over time and frequency because some, but not all, of the frequency components are removed from the noisy signal. Perceptually they are very annoying to the listener, due to their randomness and unnatural quality. Studies show than many listeners find musical noise more disturbing than the original noisy signal. Since the ultimate goal of speech enhancement is to provide good quality speech for human listeners, this is a severe flaw.
In figure 2.4 it can be seen that while spectral subtraction enhances the speech, it is at the cost of random, perceptually annoying, musical noise. Comparing the sections circled in figure 2.4 (a), (b) and (c) it is evident that the enhanced signal (c) contains random frequencies that are not present in the clean signal (a). This is an example of musical noise.

### 2.3.2 Distortion

The second problem created by the speech enhancement system is speech distortion, also examined in [6]. This happens, again due to imperfections in the noise estimation process, when speech components are incorrectly attenuated or completely removed. This reduces the naturalness and intelligibility of the speech, and is again annoying to the listener.

As spectral subtraction gives reasonable quality speech with a good level of noise reduction, it has been the goal of speech enhancement researchers over the past two decades to find accurate noise estimation techniques and to minimise these errors [12-14]. Since it is not possible to completely eliminate both, researchers have tried to reduce distortion, while keeping the residual noise below a certain level, as an acceptable trade-off between the two. It has
been proven that listeners can tolerate a low level of noise [10], provided that it is of the same spectrum as the original additive noise, as this is less irritating than random musical noise.

2.4 Improvements to Spectral Subtraction

In practice, spectral subtraction is imperfect, due to inaccurate estimation of the noise spectrum. Many improvements to the basic idea have been suggested over the years. A few of the more important concepts will be detailed briefly in this section.

2.4.1 Wiener Filtering

The Wiener filter for speech enhancement was suggested as an improvement to spectral subtraction by Lim and Oppenheim in December 1979, [6]. Rather than direct subtraction, a wiener gain function is calculated, and then multiplied by the noisy spectrum to attenuate noise component frequencies.

\[ X(\omega) = G(\omega)Y(\omega) \]  

(2.8)

where \( G(\omega) \) is the wiener filter gain coefficient for a given frequency \( \omega \).

\[ G(\omega) = \frac{|Y(\omega)|^2 - |\hat{Y}(\omega)|^2}{|Y(\omega)|^2} \]  

(2.9)

\( G(\omega) \) attenuates each frequency component by a certain amount depending on the power of the noise at the frequency.

If \( |\hat{Y}(\omega)|^2 = 0 \) then \( G(\omega) = 1 \) and no attenuation takes place, i.e. there is no noise component at the frequency \( \omega \), whereas if \( |\hat{Y}(\omega)|^2 = |Y(\omega)|^2 \), then \( G(\omega) = 0 \) and the frequency component is completely nulled. All other values of \( G(\omega) \) scale the power of the signal by an appropriate amount.
2.4.2 Spectral Subtraction with over-subtraction and spectral floor

Another useful variation on spectral subtraction was proposed by Berouti et al [7]. The improvement suggested was to introduce an over-subtraction factor $\alpha$ to control the level of noise subtraction, and a spectral floor factor, $\beta$ to maintain the spectrum of the residual noise and prevent perceptually annoying musical noise artefacts.

$$|\hat{X}(\omega)|^2 = |Y(\omega)|^2 - \alpha |\hat{W}(\omega)|^2$$

(2.10)

$$|\hat{X}(\omega)| = \begin{cases} |X(\omega)|^2, & \text{if } |X(\omega)|^2 > \beta |W(\omega)|^2 \\ \beta |W(\omega)|^2, & \text{else} \end{cases}$$

(2.11)

This aims to reduce musical noise at the cost of retaining a low level of noise in the correct spectrum.

2.4.3 Spectral Subtraction with MMSE

Ephraim and Malah [8], introduced the minimum mean-square error short-term spectral amplitude estimator in 1984. It calculates a gain function based on the a priori and a posteriori Signal to Noise Ratios (SNRs).

$$X(\omega) = H(\omega)Y(\omega)$$

(2.12)

$$H(\omega) = \frac{\sqrt{\pi}}{2} \frac{1}{\lambda N} \frac{\gamma_f}{1 + \gamma_f} F \left[ \gamma_f, \frac{\gamma_f}{1 + \gamma_f} \right]$$

(2.13)

where $\gamma_f$ is the a priori SNR, which is calculated as:

$$\gamma_f(\omega) = 0.98 \left( \frac{|\hat{X}(\omega)|^2}{|\hat{W}(\omega)|^2} \right) + (1 - 0.98)P(\gamma_{N,f} - 1)$$

(2.14)
\( i \) is the frame index with \( P(x) = \begin{cases} x & x \geq 0 \\ 0 & \text{else} \end{cases} \)

\( \gamma_n \) is the a posteriori SNR and \( F \) is given by

\[
F(x) = e^{\frac{x}{2}} \left[ (1+x)I_0\left(\frac{x}{2}\right) + I_1\left(\frac{x}{2}\right) \right] \tag{2.15}
\]

Where \( I_0 \) and \( I_1 \) are zero and first order Bessel functions, respectively. This performs non-linear smoothing when the SNR is low, i.e. in non-speech frames, and reduces speech distortion due to averaging. However it does not work well when the SNR of the speech frames is low.

### 2.4.4 Other methods

There have been countless other modifications and suggested improvements on spectral subtraction including Non-Linear Spectral Subtraction [9]. One interesting currently active area of research is spectral subtraction based on perceptual properties of speech, which was first conceived by Virag [10] in 1999. This method is a large step forward from previous methods, as it incorporates knowledge of human sound perception and introduces the masking threshold. It will be dealt with more thoroughly in the next section.

### 2.5 Perceptually motivated Spectral Subtraction

Natalie Virag [10] was the first to explore the concept of using properties of the human auditory system in speech enhancement. When two signals are close in time or frequency, one is rendered completely or partially inaudible by the other. This is known as auditory masking.
In the diagram above, the signal S1 completely masks S3 and partially masks S2, as shown by the masking curve due to S1.

The system aims to calculate the noise masking threshold and use this property to control the level of noise subtraction, allowing a trade off between residual noise and speech distortion. Firstly, the signal must be decomposed into critical bands, corresponding to the frequency selectivity of the human ear, using an auditory filterbank, as shown in table 2.1 and figure 2.6.

When a tone (e.g. voice or music) and a noise signal are both present at the same time, the masking level remains constant until the noise exceeds a certain frequency. The bandwidth at which the masking level is constant is called a critical band, or a *bark*.
## Table 2.1 Critical Band numbers and their corresponding centre frequencies

<table>
<thead>
<tr>
<th>Band Number</th>
<th>Frequency (Hz)</th>
<th>Band Number</th>
<th>Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>50</td>
<td>13</td>
<td>1,735</td>
</tr>
<tr>
<td>1</td>
<td>95</td>
<td>14</td>
<td>1,970</td>
</tr>
<tr>
<td>2</td>
<td>140</td>
<td>15</td>
<td>2,340</td>
</tr>
<tr>
<td>3</td>
<td>235</td>
<td>16</td>
<td>2,720</td>
</tr>
<tr>
<td>4</td>
<td>330</td>
<td>17</td>
<td>3,280</td>
</tr>
<tr>
<td>5</td>
<td>420</td>
<td>18</td>
<td>3,840</td>
</tr>
<tr>
<td>6</td>
<td>560</td>
<td>19</td>
<td>4,690</td>
</tr>
<tr>
<td>7</td>
<td>660</td>
<td>20</td>
<td>5,440</td>
</tr>
<tr>
<td>8</td>
<td>800</td>
<td>21</td>
<td>6,375</td>
</tr>
<tr>
<td>9</td>
<td>940</td>
<td>22</td>
<td>7,690</td>
</tr>
<tr>
<td>10</td>
<td>1,125</td>
<td>23</td>
<td>9,375</td>
</tr>
<tr>
<td>11</td>
<td>1,265</td>
<td>24</td>
<td>11,625</td>
</tr>
<tr>
<td>12</td>
<td>1,500</td>
<td>25</td>
<td>15,375</td>
</tr>
</tbody>
</table>

The masking threshold is calculated for each critical band, then they are combined to find the masking curve. Auditory coding has been used very successfully in the popular MPEG standard [11] to compress audio by discarding inaudible information, i.e. everything below the masking threshold.
The overall response of the critical band filter is similar to that of an all-pass filter, so the complete signal can be reconstructed by simply adding each band’s output together.

The gain function in Virag’s systems is calculated as follows:

\[
G(\omega) = \begin{cases} 
1 - \alpha \left( \frac{\hat{W}(\omega)}{Y(\omega)} \right)^{\gamma_1} \left( \frac{\hat{W}(\omega)}{Y(\omega)} \right)^{\gamma_2} < \frac{1}{\alpha + \beta} \\
\beta \left( \frac{\hat{W}(\omega)}{Y(\omega)} \right)^{\gamma_2} \quad \text{else}
\end{cases}
\]  

(2.16)

where the over-subtraction factor \(\alpha\) and the spectral floor parameter \(\beta\) are functions of the masking threshold \(T(\omega)\). \(\gamma_1\) and \(\gamma_2\) control the smoothness of the update of \(G(\omega)\).
This method shows reduced musical noise at the cost of incomplete noise removal, but as always is heavily reliant on accurate noise estimation.

2.6 Summary

This chapter has examined an overview of frequency domain speech enhancement algorithms, based around the Fourier transform. Most of these involve variation on the spectral subtraction method. The problems common to these methods, namely musical noise and speech distortion have also been discussed. It has been seen that all of these methods require accurate noise information to be successful. The next chapter looks at the problem of noise estimation and outlines several methods of attempting to produce a good noise estimate.
Chapter 3: Noise Estimation

As mentioned in the previous chapter, accurate estimation of noise statistics is crucial to any successful speech enhancement system. There are several distinct approaches to finding a good noise estimate, some of which are more useful in different situations than others. A number of these methods are examined in the following sections.

3.1 Two-Channel Method

A two-channel method, also known as adaptive noise cancellation [12, 13], involves the use of two recording devices as shown in figure 3.1. One microphone is used to record the speech plus noise while a second microphone is placed in a position where it can pick up noise only. The input from the latter microphone is used to calculate the spectrum of the noise in the noisy signal, and one of the algorithms described in chapter 2 can be used to reduce the noise. This only works in a stationary environment, where the noise source is known, for example where the noise from a car’s engine might interfere with mobile phone conversation. A microphone attached to a hands-free mobile phone docking station could be place appropriately to pick-up car engine noise, but not speech.

In practice it is problematic to find a suitable location for the second microphone, due to the difficulty of placing the microphone in a location where speech cannot be heard, but noise can. The level of input noise detected
by the second microphone may be different from the actual noise picked up by the mobile phone’s microphone due to their relative distances from the noise source.

Two-channel methods are thus only suitable in very limited conditions, since the noise source must be stationary with respect to the speaker. This makes it impossible in most cases, since noise can come from many sources, and many conversations take place in changing environments. For these reasons only single channel methods are considered in detail here.

In most situations only the noisy signal is available, so a means of determining noise information without a reference noise signal is necessary. There are two main options available: Voice Activity Detection (VAD) and implicit methods, and numerous variations of both methods exist.

3.2 Voice Activity Detection

Voice activity detectors aim to calculate which frames of a noisy signal contain speech and which do not. Noise statistics are estimated and updated every time a frame is judged not to contain speech, i.e. noise-only frames.

![Feature Extraction](image)

**Figure 3.2** Voice Activity Detection overview

One example of a VAD involves calculating the number of times the signal amplitude crosses the x axis (i.e. the amplitude is zero) in a frame. This is known as the Zero Crossing Rate (ZCR). Non-speech frames have a lower average ZCR than noisy speech frames, since they contain less signal information. Therefore if the ZCR for a given frame is below a certain threshold value, \( \delta \), it is determined to be a noise-only frame. Otherwise, it is determined that the frame contains speech as well as noise.
Rabiner’s VAD algorithm [14] uses the ZCR along with the short-term energy of the noisy signal to determine the presence (or absence) of speech in each frame. If the noisy signal energy in frame \( m \), \( Y(m) \), rises above the average estimated noise energy \( \hat{W}(m) \), then it is likely that frame \( m \) contains speech plus noise. Otherwise it is a noise-only frame.

\[
D(m) = \begin{cases} 
1 & \text{ZCR}(m) > \delta \text{ and } Y(m) < \hat{W} \\
0 & \text{else}
\end{cases} 
\]  \hspace{1cm} (3.1)

Where \( D(m) \) is the VAD output, with 1 representing a speech frame and 0 representing a noise-only frame.

However, there are difficulties common to all VADs [17]. Firstly, there is the situation where the noisy signal contains mostly speech with very few speech pauses, meaning the noise updates are few and far between. In that time the noise may have varied sufficiently to make the estimate inaccurate. This produces errors in the enhanced speech, such as musical noise and distortion, as discussed earlier. Even theoretical VADs which perfectly decide between noise and speech frames can produce poor results if the speech pauses are too infrequent or if the noise changes too rapidly.

Secondly, most VADs have difficulty distinguishing correctly between noise and speech at low SNRs, This results in the estimated noise spectrum containing speech components which become incorrectly attenuated, resulting in loss of speech information or distortion.
3.3 Implicit Methods Overview

Implicit noise estimation methods are those which do not require detection of speech or noise frames. Sometimes they will operate in a manner similar to VADs, but without explicitly deciding which frames contain only noise. Often they will have an adaptive update parameter which controls the level of noise update when speech is present or absent. Therefore the noise estimate is continually updated throughout the signal, and not limited to regions where no speech is present, allowing much more frequent updating.

Several different methods exist including [15, 16, 17, 19, and 20]. Some of which will be examined here.

3.3.1 Computationally Efficient Speech Enhancement by Spectral Minima Tracking in Subbands

Doblinger, [16] was amongst the first to implement a noise estimation algorithm that did not rely on voice activity detection. This technique is based on tracking the minima of the noise power in each subband.

The algorithm can be summarised as follows:

- The noisy signal power in the \( k \)th subband at frame \( m \) is denoted by \( Y_i(m) \)
- The noisy signal power estimate is smoothed using

\[
\hat{Y}_i(m) = \alpha \hat{Y}_i(m-1) + (1-\alpha)Y_i(m) \tag{3.2}
\]

Where \( \hat{Y}_i(m) \) is the averaged noisy signal power, and \( \alpha \) is the smoothing parameter.
• The noise power estimate in frame \( m \) is calculated by:

\[
\hat{W}_k(m) = \begin{cases} 
\gamma \hat{W}_k(m-1) + \frac{1-\gamma}{1-\beta}(\hat{Y}_k(m) - \beta \hat{Y}_k(m-1)) & \hat{W}_k(m-1) < \hat{Y}_k(m) \\
\hat{W}_k(m-1) & \text{else}
\end{cases}
\]  

(3.3)

Where \( \hat{W}_k \) represent the noise power estimate.

The parameters \( \alpha, \beta \) and \( \gamma \) were experimentally chosen as \( 0.7 \leq \alpha \leq 0.9 \), \( \beta = 0.96 \) and the noise smoothing parameter \( \gamma \) was chosen very close to 1, \( \gamma = 0.998 \).

![Figure 3.3 Noise estimation in subband 17 using Doblinger’s algorithm.](image)

The major drawback in Doblinger’s algorithm is evident in figure 3.3. The algorithm cannot distinguish between an increase in noise power and an increase in speech power, resulting in an overestimation of noise power during speech frames. For example in the region between frame 190 and frame 230
above the noise power estimate, $\hat{W}$ has followed the speech power. Hence distortion occurs in the estimated clean signal, due to this noise over-estimate.

### 3.3.2 Noise Power Spectral Density Based on Optimal Smoothing and Minimum Statistics

In [17], Martin proposed a noise estimation technique involving the optimally smoothed noisy signal power spectral density estimate and the analysis of the statistics of spectral minima. It is based on the observation that the power level of the noisy speech signal often decays to the power level of the noise. Hence, by tracking the minimum of the noisy speech spectrum, an estimate of the noise spectrum can be calculated.

The key improvement here over Doblinger’s algorithm [16] is that it does not use a fixed smoothing parameter as $\alpha$ above, but a time varying updating parameter, $\alpha(m)$ in each frame, $m$,

$$\alpha(m) = \frac{\hat{Y}(m-1)^2}{\hat{W}(m)^2}$$  \hspace{1cm} (3.4)

Where $\hat{Y}(m-1)^2$ is the smoothed noisy signal power estimate in frame $m-1$ and $\hat{W}(m)^2$ is the noise power estimate in the frame $m$.

The algorithm requires a multitude of different equations to determine the update parameter and will not be summarised here, but can be found in detail in [17].

The algorithm produces a noise estimate of quite good accuracy, but is relatively slow to compensate for an increase in noise power, taking up to 30 frames to update sufficiently [18]. However the algorithm offers significant improvement over Doblinger’s algorithm in many situations, as it
distinguishes well between an increase in noise power and an increase in speech power.

### 3.3.3 Subband Noise Estimation for Speech Enhancement using a Perceptual Wiener Filter

Lin, Holmes and Ambikairajah published details of another method for adaptively estimating noise in subbands [19] in 2003. Similar to Martin’s algorithm [17], it uses a time varying update to smooth the noise estimate and prevent over-estimation in speech frames.

The algorithm may be formulated as follows:

\[
\sigma^2_y(p) = \frac{1}{N} \sum_{n=0}^{N-1} Y(pN + n),
\]

(3.5)

where \( \sigma^2_y \) is the averaged noisy signal power at frame \( p \), for frame size \( N \).

The subband index, \( i \), will be dropped in subsequent formulae for simplicity.

The noise variance \( \sigma^2_w(p) \) in frame \( p \) is estimated using the one pole smoothing filter

\[
\sigma^2_w(p) = \alpha(p)\sigma^2_w(p-1) + (1 - \alpha(p))\sigma^2_y(p)
\]

(3.6)

The smoothing parameter \( \alpha(p) \), which controls the level of noise update in each frame, is found by:

\[
\alpha(p) = 1 - \min \left\{ 1, \left( \frac{\sigma^2_y(p)}{\sigma^2_w(p-1)} \right)^{-Q} \right\}
\]

(3.7)

Where \( Q \) is an integer and \( \sigma^2_w(p-1) \) is the average of the noise estimate over the previous 5 to 10 frames, i.e.,

\[
\sigma^2_w(p-1) = \frac{1}{10} \sum_{k=1}^{10} \sigma^2_w(p-k)
\]

(3.8)
The min() operation in equation 3.7 stops the possibility of negative updates in the case where the noisy signal power is less than the noise power average.

The parameter $Q$ controls the way in which $\alpha(p)$ changes with
$$\left( \frac{\hat{\sigma}^2_Y(p)}{\sigma^2_w(p-1)} \right),$$
and is normally chosen in the range $3 \leq Q \leq 5$. In general larger values of $Q$ produce larger values of $\alpha(p)$, but slower noise updates, while smaller values of $Q$ give faster noise updates at the risk of possible over-estimation of noise in speech frames.

The noise estimate is further smoothed to give the final noise estimate by

$$\hat{\sigma}^2_{w,\text{final}}(p) = \alpha_{\text{final}}\hat{\sigma}^2_{w,\text{final}}(p-1) + (1-\alpha_{\text{final}})\hat{\sigma}^2_w(p) \quad (3.9)$$

Where $\alpha_{\text{final}}$ is chosen in the range $0.5 \leq \alpha_{\text{final}} \leq 0.95$, with $\alpha_{\text{final}} = 0.5$ giving the quickest update.

The operation of the algorithm can be explained as follows:

- If speech is absent in frame $p$, the new noise power estimate $\hat{\sigma}^2_w(p)$ will be very close to the average noise estimate $\bar{\sigma}^2_w(p-1)$, so $\alpha(p) \approx 0$ and the noise estimate is updated according to equations 3.6 and 3.9.

- Conversely if speech is present in frame $p$, $\hat{\sigma}^2_Y(p)$ is much larger than $\bar{\sigma}^2_w(p-1)$, so $\alpha(p) \approx 1$ and the noise update slows down greatly, almost stopping, so as to prevent over-estimation in speech frames.

Speech information may be present in some subbands, but not in others, so the noise estimation must be carried out in each subband. Even if a frame contains
speech, the noise estimate can be updated for the subbands which are speech-free, i.e. noise-only frequencies.

The time-varying update parameter $\alpha(p)$ in this case produces a much more accurate noise estimate than Doblinger’s [16] fixed update parameter algorithm.

![Figure 3.4](image)

**Figure 3.4** Noise estimation in subband 17 using the adaptive noise estimation algorithm of Lin, Holmes and Ambikairajah.

Comparing figure 3.3 and figure 3.4, it can be seen that the algorithm in [19] does not suffer from noise over-estimation when speech power increases to the same extent as the algorithm in [16], remaining much closer to the true noise power. Comparing for example the noise estimate in frames 190 to 230 in both figures it can be seen that the noise estimate in figure 3.4 is very close to the actual noise power, while in figure 3.3 the noise power has been greatly over-estimated for the same region.

Lin et al [19] suggested a perceptual wiener filter to make use of their adaptive noise estimation algorithm, with the gain factor $G_i(p)$ calculated in frame $p$, subband $i$ by:
\[ G_i(p) = \frac{\hat{\sigma}_x^2(p)}{\hat{\sigma}_x^2(p) + \mu \max\left\{ (\hat{\sigma}_w^2(p) - \eta T_i(p)), 0 \right\}} \tag{3.10} \]

with \( \hat{\sigma}_x^2(p) = \max\left\{ \hat{\sigma}_y^2(p) - \hat{\sigma}_{\text{W,final}}^2(p), 0 \right\} \tag{3.11} \)

\( T_i(p) \) is the masking threshold in frame \( p \) calculated using the MPEG simultaneous masking model of [11]. The arbitrary parameters \( \mu \) and \( \eta \) allow a degree of freedom in the solution. For \( \eta = 0 \), the generalised wiener filter solution is obtained.

The purpose of using this masking model is to keep the musical noise below the masking threshold, and to reduce speech distortion. However performance is still limited by the wiener filter, which requires a perfect noise estimate to completely eliminate musical noise and distortion. It is however a significant improvement over the speech enhancement systems discussed in chapter 2.

### 3.4 Summary

This chapter has examined the problem of noise estimation, and discussed several possible solutions. Two-channel speech enhancement systems and voice activity detectors were briefly described before several implicit methods of noise estimation were discussed in detail. The next chapter introduces the signal subspace method of speech enhancement, which is a recently developed method and does not involve the use of the discrete Fourier transform.
Chapter 4: Signal Subspace Speech Enhancement

Ephraim and Van Trees [5] introduced the Signal Subspace approach for speech enhancement in 1995. The approach involves the use of a signal dependant transform to decompose a noisy signal into two separate subspaces, the signal plus noise subspace, and the noise-only subspace. The transform employed to perform this operation is the Karhuenen-Loeve transform (KLT). This theory assumes that speech can only span the signal plus noise subspace, for simplicity called the signal subspace, while noise can span the entire Euclidean space. Only the signal subspace is used when estimating the clean signal. The KLT components which represent the noise only subspace are nulled, while the components which represent the noisy signal are modified by a gain function. The enhanced signal is determined from the inverse KLT of the altered components. The aim here is to improve the quality, while minimising any loss in intelligibility.

![Figure 4.1 Block diagram of Subspace speech enhancement system](image)

4.1 Subspace Theory

The theory behind subspace speech enhancement is now presented. The signal subspace theory assumes a linear model of speech, as described in the following section.
4.1.1 Signal Model

The linear model of a clean signal assumes that any clean signal, \( x \) is composed of a linear combination of \( M \) basis vectors, such that

\[
x = \sum_{m=1}^{M} s_m V_m, \quad M \leq K
\]

(4.1)

where \( K \) is the dimension of the vector space.

In general \( \{s_1, ..., s_m\} \) are zero mean complex random variables and \( V_1, ..., V_m \) are \( K \)-dimensional complex basis vectors, which are assumed linearly independent. If \( M = K \) this signal representation is always possible, however for speech signals such a representation is also possible for \( M < K \). This model is consistent with the commonly used damped complex sinusoid model of speech \[21\]

\[
V_m = \left( 1, \rho_m^1 e^{j \omega_1}, ..., \rho_m^{K-1} e^{j \omega(K-1)} \right)^T
\]

(4.2)

This model can be written as

\[
x = Vs
\]

(4.3)

where \( V = [V_1, ..., V_m] \) is a \( K \times M \) matrix whose rank is \( M \), and \( s = (s_1, ..., s_M)^T \).

When \( M < K \), the set of all possible signal vectors \( \{x\} \) lie in a subspace of the Euclidean space \( \mathbb{R}^K \), spanned by the columns of \( K \), the signal subspace. When \( M = K \), the signal subspace spans the entire \( \mathbb{R}^K \) space, however this is not relevant to speech, as the speech signal is almost always contained in a subspace \( \mathbb{R}^M \) of \( \mathbb{R}^K \).
The covariance matrix of $x$ is given by:

$$R_x = \mathbb{E}\{xx^\#\} = VRV^\#$$  \hspace{1cm} (4.4)

$(\cdot)^\#$ denotes the vector conjugate transpose and $R_x$ denotes the covariance matrix of the vector $s$ in equation 4.3, which is assumed positive definite. The rank of $R_x$ is $M$, and the matrix has $K - M$ zero eigenvalues, since speech is contained in only a subspace of the entire Euclidean space.

### 4.1.2 Noise Model

Let $w$ denote a K-dimensional vector of the noise process, which is assumed zero mean, additive and uncorrelated with the speech signal. Ephraim and Van Trees [5], assume here that the noise is white, and propose applying a pre-whitening transformation $Rw^{-1/2}$ to $w$ in cases where noise may not be white. This is not optimal, as will be seen in section 4.3. For now we will proceed under the assumption that the noise is either white or can be whitened by the above transformation.

Therefore:

$$R_w = \mathbb{E}\{ww^\#\} = \sigma_w^2 I$$  \hspace{1cm} (4.5)

In this case, the rank of $R_w$ is $K$ as the noise spans the entire Euclidean space. Thus noise exists in the signal subspace as well as the noise-only subspace. We will next describe how the KLT can be used to decompose the noisy signal into a signal and a noise subspace.

### 4.1.3 Signal and Noise Subspaces

Let $y$ denote a K-dimensional vector of the noisy signal.

$$y = Vs + w$$  \hspace{1cm} (4.6)
The covariance matrix of $y$ is given by:

$$R_y = E\left\{yy^\theta\right\} = VR\theta V^\theta + R_w$$  \hspace{1cm} (4.7)

Let $R_y = U\Delta_y U^\theta$ be the eigendecomposition of $R_y$. $U = [u_1,...,u_K]$ is an orthonormal matrix of eigenvectors $\{U_k \in \mathbb{R}^K\}$ of $R_y$ and $\Delta_y = \text{diag}(\lambda_y(1),...,\lambda_y(K))$ denotes a diagonal matrix of eigenvalues of $R_y$. Since the noise is assumed white the eigenvectors of $R_y$ are also the eigenvectors of $R_x$ and $R_w$, and all eigenvalues of $R_w$ equal $\sigma_w^2$.  

Since the rank of $R_x$ is $M$, the matrix $R_x$ has $M$ positive eigenvalues and $K - M$ zero eigenvalues. Assuming that the $M$ positive eigenvalues of $R_x$ are $\{\lambda_x(1),...,\lambda_x(M)\}$ and the corresponding $M$ eigenvalues of $R_x$ are $\{u_1,...,u_M\}$. For convenience we assume that $\{\lambda_x(1),...,\lambda_x(M)\}$ are given in descending order. The eigenvalues of $R_y$, $\Delta_y$ are given by:

$$\lambda_y(k) = \begin{cases} 
\lambda_x(k) + \sigma_w^2 & \text{for } k = 1,...,M \\
\sigma_w^2 & \text{for } k = M + 1,...,K 
\end{cases}$$  \hspace{1cm} (4.8)

Thus the eigendecomposition of $R_y$ is given by

$$R_y = U\Delta_y U^\theta$$  \hspace{1cm} (4.9)

$$\Delta_y = \text{diag}\left[\Delta_y, \sigma_w^2 I\right]$$  \hspace{1cm} (4.10)

$$\Delta_y, 1 = \text{diag}\left(\lambda_x(1),...,\lambda_x(M)\right)$$  \hspace{1cm} (4.11)

And the eigendecomposition of $R_x$ is given by

$$R_x = U\Delta_x U^\theta$$  \hspace{1cm} (4.12)

\footnote{This is not strictly a valid assumption for non-white noise, or even noise pre-whitened by the pre-whitening transformations. The reasons for this are given in section 5.1}
\[ \Delta_r = \text{diag}[\Delta_r, 0I] \] \hspace{2cm} (4.13)

\[ \Delta_r = \text{diag}(\lambda_r(1), \ldots, \lambda_r(M)) = \Delta_r - \sigma^2 \] \hspace{2cm} (4.14)

The eigenvalues of \( \Delta_r \) and their corresponding eigenvectors are referred to as the principal eigenvalues and eigenvectors of \( R_r \), respectively.

Let \( U = [U_1, U_2] \), where \( U_1 \) denotes the \( K \times M \) matrix of principal eigenvectors of \( R_r \), i.e.

\[ U_1 = \{ u_k : \lambda_r(k) > \sigma^2 \} \] \hspace{2cm} (4.15)

Since \( U \) is orthonormal

\[ I = U_1 U_1^* + U_2 U_2^* \] \hspace{2cm} (4.16)

The matrix \( U_1 U_1^* \) is idempotent and Hermitian. Hence, it is the orthogonal projector onto the subspace spanned by the columns of \( U_1 \), but \( \text{span}(U_1) = \text{span}(V) \). Therefore \( U_1 U_1^* \) is the orthogonal projector onto the signal subspace. The complementary orthogonal subspace is spanned by the columns of \( U_2 \) and it constitutes the noise-only subspace. The matrix \( U_2 U_2^* \) is the orthogonal projector onto that subspace.

Thus, a noisy vector, \( y \) can be decomposed as

\[ y = U_1 U_1^* y + U_2 U_2^* y \] \hspace{2cm} (4.17)

where \( U_1 U_1^* \) is the projection of \( y \) onto the signal subspace and \( U_2 U_2^* \) is the projection of \( y \) onto the noise subspace. The coefficient vectors of the two
projections $U_1^y$ and $U_2^y$, respectively, are obtained from $U^z$ which is the KLT of $y$

$$
\text{cov}(U^y) = \text{diag}[\Delta_{c,1} + \sigma^2_\delta I, \sigma^2_\delta I]
$$

(4.18)

$\text{cov}(U_2^y) = \sigma^2_w I$, and hence contains no signal information and may be nulled when estimating the clean signal. $\sigma^2_w$, the noise estimate, is traditionally found using a VAD.

### 4.2 Linear Signal Estimators

Ephraim and Van Trees also suggested linear signal estimators, or gain functions, which minimise the signal distortion while constraining the energy and spectrum of the residual noise, since both cannot be totally eliminated at the same time [5].

The first estimator maintains the energy of the residual noise in the entire frame below a certain threshold. This allows time domain constraint (TDC) on the residual noise.

The second guarantees that the energy of the residual noise in each spectral component is kept below a given threshold. It is designed for noise shaping using spectral domain constraints (SDC).

#### 4.2.1 Time Domain Constrained Estimator

Let $\hat{x} = Hy$ be a linear estimator of the clean signal $x$ where $H$ is a $K \times K$ matrix. The residual signal, i.e. the error is given by

$$
r = \hat{x} - x = (H - I)x + Hw = r_y + r_w
$$

(4.19)

where $r_y = (H - I)x$ represents the signal distortion, and $r_w = Hw$ represents the residual noise.
\[ \mathcal{E}_r = \text{tr} \{ r_r r_r^* \} = \text{tr} \left\{ (H - I) R_s (H - I)^* \right\} \] (4.20)

is the energy of the residual noise vector \( r_r \). Similarly,

\[ \mathcal{E}_w = \text{tr} \{ r_w r_w^* \} = \sigma_w^2 \text{tr} \{ HH^* \} \] (4.21)

is the energy of the residual noise vector \( r_w \). The linear estimator with TDC on the residual noise is obtained from

\[
\min_{\varepsilon} \mathcal{E}_r, \text{ subject to } \frac{1}{K} \mathcal{E}_r \leq \alpha \sigma_w^2 \] (4.22)

where \( 0 \leq \alpha \leq 1 \). The estimator derived in this way minimises the signal distortion over all linear filters which result in the permissible residual noise level \( \alpha \sigma_w^2 \). The value of \( \alpha \) is restricted to \( 0 \leq \alpha \leq 1 \), since \( \alpha \) cannot take negative values.

After some further calculations, which are omitted here, Ephraim and Van Trees deduce that the timed domain constraint filter is given by:

\[
H_{\text{TDC}} = \frac{R_s}{R_s + \mu \sigma_w^2 I} \] (4.23)

This results in a Wiener filter with adjustable input noise level \( \mu \sigma_w^2 \).

The optimal linear estimator can be rewritten as:

\[
H_{\text{TDC}} = U \begin{pmatrix} G_\mu & 0 \\ 0 & 0 \end{pmatrix} U^* \] (4.24)

\[
G_\mu = \frac{\Delta_r}{(\Delta_r + \mu \sigma_w^2)} \] (4.25)
Hence the signal estimate $\hat{x}_{\text{msc}} = H_{\text{msc}} y$ is obtained by applying the KLT to the noisy signal, modifying the components of the KLT, $U^H x$, by a gain function, and then applying inverse KLT to the modified components. The gain function nulls the components which lie in the noise-only subspace, as well as attenuating the noise in the signal plus noise subspace. Hence:

$$H_{\text{msc}} = U_i G_{\mu} U_{i}^H$$ 
(4.26)

For this estimator the coefficients of the projection of the noisy signal onto the signal subspace, $U_{i}^H y$, are calculated, then these coefficients are modified by the gain function, $G_{\mu}$. The modified components are then used to reconstruct the signal in the signal subspace.

The linear estimator in equation 4.26 can be explicitly written as:

$$H_{\text{msc}} = \sum_{m=1}^{M} g_{\mu}(m) u_{\text{null}m} u_{\text{null}m}^H$$ 
(4.27)

where $g_{\mu}(m)$ denotes the $m^{th}$ diagonal element of $G_{\mu}$ given by

$$g_{\mu}(m) = \frac{\lambda_{\mu}(m)}{\lambda_{\mu}(m) + \mu \sigma_w^2}$$ 
(4.28)
4.2.2 Spectral Domain Constrained Estimator

The spectral domain constrained linear signal estimator minimises the signal distortion subject to constraints on the spectrum of the residual noise. The spectrum can be made similar to that of the speech, and thus the residual noise can be masked by the speech, without the production of perceptually disturbing musical noise artefacts. The $k^{th}$ spectral component of the residual noise is given by $\mu_k^\# r_w$. For $k = 1,...,M$, we require that the energy in $\mu_k^\# r_w$ be smaller than or equal to $\alpha_k \sigma_w^2$, where $0 < \alpha_k \leq 1$. For $k = M+1,...,K$, we require that the energy in $\mu_k^\# r_w$ be zero, since the signal energy in the noise subspace is zero. Hence the filter is designed by:

$$\min_\varepsilon \varepsilon^2 \text{ subject to } \begin{cases} E\left\{\left|\mu_k^\# r_w\right|^2\right\} \leq \alpha_k \sigma_w^2, & k = 1,...,M \\ E\left\{\left|\mu_k^\# r_w\right|^2\right\} = 0, & k = M+1,...,K \end{cases} \quad (4.29)$$

This is optimised using a similar procedure to the TDC estimator, while taking into account that $H$ can now have complex entries. $H$ must satisfy the following gradient matrix equation

$$HR_\varepsilon + \sigma_w^2 LH - R_\varepsilon = 0 \quad (4.30)$$

where $L = U \Delta u U^\#$

$$\Delta u = \text{diag}(\mu_1,...,\mu_K)$ is a diagonal matrix of Lagrange multipliers.

Applying the eigendecomposition from equation 4.12 of $R_\varepsilon$ to equation 4.30 we obtain

$$(I - Q)\Delta_\varepsilon - \sigma_w^2 \Delta u Q = 0 \quad (4.32)$$
where $Q = U^H H U$. A possible solution to equation 4.32 is obtained when $Q$ is diagonal with elements given by

$$q_{uk} = \begin{cases} \frac{\lambda_u(k)}{\lambda_u(k) + \sigma_w^2 \mu_k} & k = 1, \ldots, M \\ 0 & k = M + 1, \ldots, K \end{cases} \quad (4.33)$$

For this $Q$ we have

$$E\left[|\mu_k \# r_w|^2\right] = \begin{cases} \sigma_w^2 q_{uk}^2 & k = 1, \ldots, M \\ 0 & k = M + 1, \ldots, K \end{cases} \quad (4.34)$$

If the non-zero constraints in equation 4.29 are satisfied with equality, then $\sigma_w^2 q_{uk}^2 = \alpha_k \sigma_w^2$ implies that

$$q_{uk} = \alpha_k^{\frac{1}{2}}, \quad k = 1, \ldots, M \quad (4.35)$$

and

$$\mu_k = \frac{\lambda_u(k)}{\sigma_w^2} \left(1 - \left(\frac{1}{\alpha_k}\right)^{\frac{1}{2}}\right), \quad k = 1, \ldots, M \quad (4.36)$$

It can be concluded that the desired $H$ is given by

$$H = UQU^H$$

$$Q = \text{diag}(q_1, \ldots, q_M)$$

$$q_{uk} = \begin{cases} \alpha_k^{\frac{1}{2}} & k = 1, \ldots, M \\ 0 & k = M + 1, \ldots, K \end{cases} \quad (4.37)$$
The choice of $\alpha_s$ completely specifies the gain of the estimator. The input noise is assumed white (or pre-whitened) with spectrum $\sigma_w^2$ and the non-zero spectrum of the output residual noise is $\alpha_s\sigma_w^2$.

A possible choice for $\alpha_s$ is:

$$\alpha_s = \left( \frac{\lambda_s(k)}{\lambda_s(k) + \sigma_w^2} \right)^\gamma$$  \hspace{1cm} (4.38)

where $\gamma \geq 1$ is an experimentally determined constant. This choice makes the spectrum of the residual noise look similar to that of the clean signal. The value of $\gamma$ controls the suppression level of the noise as well as the resulting signal distortion. When $\gamma$ increases, the permitted residual noise level decreases and the signal distortion level increases.

An alternative choice of $\alpha_s$ which results in a more aggressive noise suppression gain function is given by:

$$\alpha_s = \exp\left\{ -\nu \sigma_w^2 / \lambda_s(k) \right\}$$  \hspace{1cm} (4.39)

where $\nu \geq 1$ is an experimentally chosen constant which controls the level of noise suppression as well as the resulting signal distortion.

A value of $\nu = 2$ results in the generalised wiener gain function. The generalised wiener gain function with $\nu = 5$ was found particularly useful in speech enhancement [5].

### 4.3 Pre-whitening of additive coloured noise

The above subspace principles require that the additive noise is white and that the variance $\sigma_w^2$ is equal in each channel, i.e. $R_w = E\{ww^T\} = \sigma_w^2 I$. This is simply not the case for many real world, or coloured noise, signals. Ephraim and Van Trees [5] suggest a pre-whitening filter $\sqrt[\frac{1}{2}]{R_w}$ to deal with this problem.
The noisy signal is pre-multiplied by $R_w^{-\frac{1}{2}}$ to give:

$$\tilde{y} = R_w^{-\frac{1}{2}} y = R_w^{-\frac{1}{2}} x + R_w^{-\frac{1}{2}} w = \tilde{x} + \tilde{w} \quad (4.40)$$

The filter $\tilde{H}$ can be derived using the linear estimators described in section 4.2.

$$H = R_w^{-\frac{1}{2}} \tilde{H} R_w^{-\frac{1}{2}} \quad (4.41)$$

However these filters are not optimal as they are derived from the values

$$\tilde{e}_e^2 = \text{tr} \left( R_w^{-\frac{1}{2}} E \{ e_e e_e^H \} R_w^{-\frac{1}{2}} \right) \quad \text{and} \quad \tilde{e}_w^2 = \text{tr} \left( R_w^{-\frac{1}{2}} E \{ e_w e_w^H \} R_w^{-\frac{1}{2}} \right) \quad \text{instead of}$$

$$e_e^2 \quad \text{and} \quad e_w^2.$$ 

The performance of the subspace system has been shown to degrade in coloured noise, even when a pre-whitening filter is used.

### 4.4 Rank Estimation

The estimation of the rank order, $M$ is an important part of the subspace theory, as it controls how much of the Euclidean space is noise-only and can be nulled.

Three estimators were tested by Klein and Kabal [22], The theoretical estimator which nulls all eigenvectors with negative eigenvalues, the Minimum Description Length estimator and Merhav’s estimator [23]. They deduced that the MDL method and Merhav’s method, while more complex, did not yield significantly better quality speech, so only the simple theoretical estimator is considered in this thesis.

For $R_v = R_v - \sigma_v^2 I$, the order is chosen as the number of strictly positive eigenvalues of $R_v$, i.e.

$$M = \# \{ k \in \mathbb{N} : \lambda_k > 0 \} \quad (4.42)$$
4.5 Noise Estimation

Like all speech enhancement systems, the signal subspace speech enhancement algorithm is entirely dependant on the noise estimate, $\sigma_w^2$. In any current signal subspace algorithm $\sigma_w^2$ is found from eigendecomposition of the noise covariance $R_w$, i.e. $R_w = U\Delta w U^H$, with all values of $\Delta w$ being equal to $\sigma_w^2$, under the white noise assumption. To find this estimate $R_w$ must first be found. So far, this has always been done using a voice activity detector, similar to those discussed in chapter 3.

In the next chapter a method of estimating noise for signal subspace based speech enhancement that does not require any explicit voice activity detection and which make no assumptions about noise being white or pre-whitened is proposed.

4.6 Summary

In this chapter the signal subspace theory and its application to speech enhancement have been described in detail, and the filters which minimise distortion, while keeping the residual noise below some threshold have been derived. Some of the drawbacks involved have also been mentioned, particularly in the case of non-white noise. The use of a VAD to estimate the noise covariance matrix is also a limiting factor in the performance of the algorithm, to which no solution has so far been proposed. In the next chapter a novel approach to signal subspace speech enhancement is detailed. The paradigm proposed can update the noise estimate through the entire signal, requires no explicit voice activity detection and makes no assumption about noise being white.
Chapter 5: Adaptive Noise Estimation for Subspace Speech Enhancement

A novel approach to signal subspace based speech enhancement, which deals with the issue of coloured noise in a more realistic way than the traditional subspace method is presented in this chapter. As mentioned previously no explicit voice activity detection is required, and the noise estimate is updated throughout the signal. The approach, including the adaptation to the subspace method for coloured noise (i.e. real-world noise signals), the eigenvalue tracking method for noise estimation, as well as its actual implementation in software are described in the following sections.

Figure 5.1 Block diagram of complete Subspace Speech Enhancement with Adaptive Noise Estimation algorithm

Figure 5.1 shows an overview of the complete system. The noisy signal is firstly framed into overlapping blocks of \( K \) samples, then several of these are used to calculate the noisy covariance matrix. Eigendecomposition is then performed on the noisy covariance matrix. Following this the noise eigenvalues for the current frame is calculated using the algorithm described in this chapter. The linear estimator needed to estimate the clean signal is calculated before being applied to the noisy signal blocks, to produce an estimate of the clean signal for the current frame. The linear estimators are derived in sections 5.3.1 and 5.3.2. Finally, each frame is windowed and overlapped with the previous frame to produce the enhanced signal.
5.1 Improved Subspace Speech Enhancement for Coloured Noise

The subspace approach of Ephraim and Malah [5], as described in the previous chapter, assumes that the noise variance $\sigma_w^2$ is equal in each eigenvector. This is not true in the case of coloured or real-world noise types, and can easily be observed by looking at the noise variances for a noisy signal over several of the eigenvectors.

![Figure 5.2 Noise Eigenvalues for six different eigenvectors of a noisy sample](image)

In the above diagram noise eigenvalues (noise variance) for six different eigenvectors of a noisy signal are shown. The noise is pink noise at a signal to noise ratio of 5dB. It is obvious from the diagram that the noise variance is not equal for every eigenvector, and therefore $\lambda_w \neq \sigma_w^2$ for all values of $i$.

Ephraim and Malah [5] suggest a pre-whitening filter $R_w^{-\frac{1}{2}}$ to overcome this, and while it makes the eigenvalues closer in each eigenvector, it still does not make them all equal, as can be seen in figure 5.3. The other problem associated with pre-whitening is that the linear estimators are not optimal, as mentioned in the previous chapter.
From this stage forward, the noise in each eigenvector will not be assumed to have an equal variance, and each eigenvector will be treated separately when estimating noise statistics. This approach was used by Gazor and Razayee in [24] to develop a coloured noise subspace based method which uses a voice activity detector to estimate the noise.

It has previously been seen that the eigendecomposition of the noise covariance matrix $R_y$, is given by $R_y = U\Delta v U^\dagger$

In equation 4.8, $\Delta v = \text{diag}(\lambda_\nu(1),...,\lambda_\nu(K))$ was given as

$$\lambda_\nu(k) = \begin{cases} \lambda_\nu(k) + \sigma_w^2 & \text{for } k = 1,\ldots,M \\ \sigma_w^2 & \text{for } k = M + 1,\ldots,K \end{cases}$$

But since equal noise variances are no longer assumed, but eigenvalues for uncorrelated additive noise are themselves additive we now have:

$$\Delta v = \Delta v + \Delta w$$

(5.1)
\[ \Delta_v = \text{diag}(\lambda_v(1), \ldots, \lambda_v(K)) \] is now composed of eigenvalues given by

\[ \lambda_v(k) = \begin{cases} \lambda_v(k) + \lambda_w(k) & \text{for } k = 1, \ldots, M \\ \lambda_w(k) & \text{for } k = M + 1, \ldots, K \end{cases} \quad (5.2) \]

This can be written as

\[ \lambda_v(n) = \begin{cases} \lambda_v(n) + \lambda_w(n) & \text{for } i = 1, \ldots, M \\ \lambda_w(n) & \text{for } i = M + 1, \ldots, K \end{cases} \quad (5.3) \]

for clarity, where \( i \) is the eigenvector number, and \( n \) is the frame number, since speech and noise eigenvector change slowly from frame to frame.

The system implemented by Gazor and Razayee [24] shows good improvement over the original subspace method of Ephraim and Malah [5], due to this more appropriate coloured noise handling, but is still dependant on VAD performance. A novel approach to noise estimation in eigenvectors which does not require explicit voice activity detection is described in the following section.
5.2 Adaptive Noise Estimation for Subspace Speech Enhancement

While there have been several methods of noise estimation in the frequency domain, employing spectral subtraction and Fast Fourier Transform methods, as discussed in chapters 2 and 3, thus far no such method have been implemented in the subspace domain. Every current subspace method uses a voice activity detector to obtain a noise estimate, and therefore can only update the noise estimate during speech pauses. This leads to inaccurate noise estimates when the noise is non-stationary and also results in the associated problems of musical noise and distortion. Here an adaptive noise estimation algorithm for subspace speech enhancement is proposed. It is similar to that of Lin et al [19] (as described in section 3.3.3) in the technique of tracking noise statistics, but has been modified for use in the subspace domain rather than the frequency domain.

It is proposed to track the eigenvalues of the (non pre-whitened) noisy covariance matrix \( R_n = U \Delta(n) U^T \), given by \( \Delta(n) = \text{diag}(\lambda_v(n), \ldots, \lambda_v(n)) \) where

\[
\lambda_v(n) = \begin{cases} 
\lambda_v(n) + \lambda_w(n) & \text{for } i = 1, \ldots, M \\
\lambda_w(n) & \text{for } i = M + 1, \ldots, K
\end{cases}
\]

where \( i \) is the eigenvector number, and \( n \) is the frame number, as seen in the previous section.

Since \( \Delta_v(n) \) is a diagonal matrix, it only contains one eigenvalue for each eigenvector in any frame, and thus does not require calculation of an average value, unlike in equation 3.5. \( \lambda_v(n) \) represents the power of the signal for a given eigenvector.

The noise power, \( \hat{\lambda}_w(n) \) in eigenvector \( i \), frame \( n \) is estimated using:

\[
\hat{\lambda}_w(n) = \alpha(n) \hat{\lambda}_w(n-1) + (1-\alpha(n)) \lambda_v(n)
\]  

which represents a single-pole smoothing filter.
The update parameter $\alpha(n)$ is given by:

$$
\alpha(n) = 1 - \min \left\{ 1, \left( \frac{\hat{\lambda}_y(n)}{\bar{\lambda}_w(n-1)} \right)^Q \right\}
$$

(5.5)

where $Q$ is an integer and $\bar{\lambda}_w(n-1)$ represents the average noise estimate over the previous 10-20 frames (due to shorter frame length for subspace method), i.e.

$$
\bar{\lambda}_w(n-1) = \frac{1}{20} \sum_{p=1}^{20} \hat{\lambda}_w(n-p)
$$

(5.6)

The parameter $Q$ controls the rate of change of $\left( \frac{\hat{\lambda}_y(n)}{\bar{\lambda}_w(n-1)} \right)$ with respect to the noisy signal power, reducing the rate of update in eigenvectors which contain speech. It has been experimentally been chosen as $Q = 5$. This value helps prevent over-estimation of the noise eigenvalues, at the cost of reducing the rate of noise update.

The estimate is further smoothed by:

$$
\hat{\lambda}_{w, \text{final}}(n) = \alpha_{\text{final}} \hat{\lambda}_{w, \text{final}}(n)(n-1) + (1 - \alpha_{\text{final}}) \hat{\lambda}_w(n)
$$

(5.7)

$\alpha_{\text{final}}$ has been experimentally chosen as $0.75 \leq \alpha_{\text{final}} \leq 0.99$, due to the short frame size of the subspace method. This is because the covariance matrix of the noisy signal is chosen as a $K \times K$, with $K$ usually chosen as 40. It takes roughly six subspace frames to represent the same amount of time (~10 milliseconds) as one frame in the frequency domain. Thus a larger value of $\alpha_{\text{final}}$ is necessary to avoid spurious noise updates.

The algorithm works by tracking the minimum values of $\hat{\lambda}_y(n)$. This is acceptable as eigenvalues for noise are lower than those for noisy speech. The parameter $\alpha(n)$, which controls the level of noise update is crucial, and works as follows:
• If the eigenvalue for the given eigenvector is much higher than the average noise eigenvalue, then speech is most likely present. The \[ \left( \frac{\lambda_v(n)}{\lambda_w(n-1)} \right)^Q \] term becomes very small and \( \alpha(n) \approx 1 \), therefore the noise update slows down to almost zero, and the previous value of the noise eigenvalue is again used as the current estimate.

• If the value of \( \lambda_v(n) \) is close to \( \lambda_w(n-1) \), then speech is most likely absent and \( \alpha(n) \approx 0 \), and the noise eigenvalue estimate is updated using the current noisy eigenvalue.

The noise estimate in the noise-only subspace can be updated constantly since no speech is present.

\[ \text{Figure 5.4 (a)} \] Speech corrupted with pink noise at an SNR of 2dB
Figure 5.4 (b) Noise estimation in eigenvector 10.

Figure 5.4 (c) The same signal in eigenvector 20

Figure 5.4 shows a noisy speech file corrupted with pink noise at a signal to noise ratio of about 2dB. It can be seen from figure 5.4 (b) and (c) that the noise estimate (represented by the solid thick black line) is very close to the actual noise eigenvalues (solid blue line). The noisy signal eigenvalues are also shown (dashed red line).
5.3 Speech Enhancement using the Subspace Adaptive Noise Estimation Algorithm

With an algorithm to estimate noise in the subspace domain available, which requires no voice activity detection and makes no assumptions about the noise being white, it may be applied to subspace speech enhancement.

The linear estimators also assume white noise and must be modified to remove this assumption.

5.3.1 Time Domain Constraint Linear Estimator for Coloured Noise

The time domain constraint linear estimator given by equation 4.27

\[ H_{TDC} = \sum_{m=1}^{M} g_{\mu}(m) u_{m} u_{m}^{\#} \]

where (equation 4.28)

\[ g_{\mu}(m) = \frac{\hat{\lambda}_{\mu}(m)}{\hat{\lambda}_{\mu}(m) + \mu \sigma_{w}^{2}} \]

must be updated so as to cater for coloured noise. Therefore it becomes

\[ g_{\mu}(m) = \frac{\hat{\lambda}_{\mu}(m)}{\hat{\lambda}_{\mu}(m) + \mu \hat{\lambda}_{w}(m)} \tag{5.8} \]

This may also be written as

\[ g_{\mu}(m) = \frac{\hat{\lambda}_{\mu}(m) - \hat{\lambda}_{w}(m)}{\hat{\lambda}_{\mu}(m) - (1 + \mu) \hat{\lambda}_{w}(m)} \tag{5.9} \]
since $\hat{\lambda}_s(m)$ is unknown. $\lambda_s(m)$ is the clean speech eigenvalue in eigenvector $m$ and is the value we want to estimate. If $\mu = 1$, as is commonly used, then the equation becomes:

$$g_\mu(m) = \frac{\lambda_s(m) - \hat{\lambda}_s(m)}{\lambda_s(m)}$$  \hspace{1cm} (5.10)

### 5.3.2 Spectral Domain Constraint Linear Estimator for Coloured Noise

The linear SDC estimator was previously given as

$$q_{ks} = \begin{cases} \alpha_k^2 & k = 1, \ldots, M \\ 0 & k = M + 1, \ldots, K \end{cases}$$

where $\alpha_k^2$ was necessary due to the pre-whitening filter. As the pre-whitening filter is no longer used in this algorithm, this becomes:

$$q_{ks} = \begin{cases} \alpha_k & k = 1, \ldots, M \\ 0 & k = M + 1, \ldots, K \end{cases}$$  \hspace{1cm} (5.11)

$\alpha_k$ was given by

$$\alpha_k = \left( \frac{\lambda_s(k)}{\hat{\lambda}_s(k) + \sigma_w^2} \right)^\gamma,$$

assuming white noise. This now becomes

$$\alpha_k = \left( \frac{\lambda_s(k)}{\hat{\lambda}_s(k) + \hat{\lambda}_s(k)} \right)^\gamma$$  \hspace{1cm} (5.12)

or

$$\alpha_k = \left( \frac{\lambda_s(k) - \hat{\lambda}_s(k)}{\hat{\lambda}_s(k)} \right)^\gamma$$  \hspace{1cm} (5.13)
The alternative \( \alpha_k = \exp\left\{ -\nu \sigma_n^2 / \lambda_\nu(k) \right\} \) becomes

\[
\alpha_k = \exp\left\{ -\nu \hat{\lambda}_\nu(k) / \hat{\lambda}_\nu(k) \right\} \quad (5.14)
\]

Another advantage here over the normal subspace method is that the noise-only subspace is implicitly nulled using any of the linear estimators, but modified for the improved coloured noise handling assumptions, e.g.

\[
q_{ik} = \begin{cases} 
\frac{\hat{\lambda}_\nu(k)}{\hat{\lambda}_\nu(k) + \sigma_n^2 \mu_k} & k = 1, \ldots, M \\
0 & k = M + 1, \ldots, K 
\end{cases}
\]

which under the coloured noise handling improvement becomes:

\[
q_{ik} = \begin{cases} 
\frac{\hat{\lambda}_\nu(k)}{\hat{\lambda}_\nu(k) + \hat{\lambda}_\nu(k)} & k = 1, \ldots, M \\
0 & k = M + 1, \ldots, K 
\end{cases} \quad (5.15)
\]

This can also be represented by:

\[
q_{ik} = \begin{cases} 
\hat{\lambda}_\nu(k) - \hat{\lambda}_\nu(k) / \hat{\lambda}_\nu(k) & k = 1, \ldots, K 
\end{cases} \quad (5.16)
\]

For the noise-only subspace, i.e. for \( k = M + 1, \ldots, K \), \( \hat{\lambda}_\nu(k) = \hat{\lambda}_\nu(k) \), therefore, assuming an accurate noise update as provided by the above algorithm,

\[
q_{ik} = \begin{cases} 
\frac{\hat{\lambda}_\nu(k) - \hat{\lambda}_\nu(k)}{\hat{\lambda}_\nu(k)} & k = 1, \ldots, M \\
\frac{\hat{\lambda}_\nu(k) - \hat{\lambda}_\nu(k)}{\hat{\lambda}_\nu(k)} = 0 & k = M + 1, \ldots, K 
\end{cases} \quad (5.17)
\]

No explicit rank estimation is necessary, since an accurate noise estimate for noise-only eigenvectors will null these eigenvectors implicitly.
5.4 System Implementation

Several issues must be considered when implementing the algorithm in the *Matlab* programming environment. The choice of windowing length and type, selection of linear estimator and the various parameters of the noise estimation algorithm must all be considered carefully.

In implementing the subspace algorithm, the value of $K = 40$ was chosen, with a rectangular window and 50% overlap between frames. The noisy covariance matrix is calculated from 400 samples of the noisy signal, giving a $K \times K$ matrix, which was decomposed into component eigenvectors and eigenvalues using a *schur* decomposition. The enhanced vectors are hanning windowed and overlap-added to estimate the clean signal, as suggested in [5].

The alternative SDC estimator $\alpha_s = \exp\left\{-\nu\lambda_s(k)/\lambda_s(k)\right\}$, with $\nu = 5$, was experimentally chosen as the most successful linear signal estimator for the system.

The parameters $\alpha_{\text{final}}$ and $Q$ for the noise estimation algorithm were experimentally chosen as $\alpha_{\text{final}} = 0.95$ and $Q = 5$. These settings give the best trade off between a quick noise update and avoidance of noise over-estimation.

5.5 Summary

This chapter focused on alternative methods of dealing with coloured noise to the normal subspace method, removing the white noise assumption and the need for a pre-whitening filter. A novel method of estimating noise in each eigenvector using an adaptive algorithm was presented and discussed. Unlike previous subspace methods it requires no voice activity detection, no estimation of the noise covariance matrix and can update the noise estimate throughout the entire signal. It also requires no pre-whitening as it is designed to make uses of coloured noise subspace handling techniques. While the noise estimation algorithm is based on the adaptive noise estimation of Lin *et al*
[19], it has never been previously used in the subspace domain, having been implemented with a perceptual wiener filter in the frequency domain. The Signal Subspace adaptive noise estimation method does not track the noise power in critical bands unlike the frequency domain method, but follows the noise eigenvalues in each noisy covariance matrix frame. The linear estimators for coloured noise speech enhancement were derived before a complete system overview and some implementation issues were described. In the next chapter the performance of the algorithm is tested against several other speech enhancement schemes and the results are presented and analysed.
Chapter 6: Testing and Results

Evaluation of any speech enhancement system requires a series of subjective and objective tests to be conducted. This chapter describes the tests employed, the test data used, algorithms used for comparison and the justifications for these selections. The results for each test and an analysis of the results are also given. For reference the algorithm proposed in this thesis is referred to as the signal subspace with adaptive noise estimation algorithm (SSANE).

6.1 Test Data

All speech files used in the following tests were taken from the SpEAR database [25]. They are short speech samples (2-4 seconds each) sampled at 16 kHz. For reference the eight speech files used throughout the test were as follows:

<table>
<thead>
<tr>
<th>FILENAME</th>
<th>SENTENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigtips_16.wav</td>
<td>“Good service should be rewarded by big tips”</td>
</tr>
<tr>
<td>Butterscotch_16.wav</td>
<td>“Butterscotch fudge goes well with vanilla ice-cream”</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>“Our first necessity at the very outset of war is post-attack recognisance”</td>
</tr>
<tr>
<td>Peaches_16.wav</td>
<td>“The fifth jar contains big juicy peaches”</td>
</tr>
<tr>
<td>Porcupines_16.wav</td>
<td>“Porcupines resemble sea urchins”</td>
</tr>
<tr>
<td>Prison_16.wav</td>
<td>“The high security prison was surrounded by barbed wire”</td>
</tr>
<tr>
<td>Recognition_16.wav</td>
<td>“We apply auditory modelling in computer speech recognition”</td>
</tr>
<tr>
<td>Scholars_16.wav</td>
<td>“Biblical scholars argue history”</td>
</tr>
</tbody>
</table>

Table 6.1 SpEAR database speech filenames and their sentences

All noise samples were taken from the Noisex-92 database [26]. These were recorded at 19.98 kHz, and are resampled to 16 kHz in keeping with the sampling frequency of the speech files.

The noise was added to the speech files at a series of different signal to noise ratios. The SNR is defined as the ratio of the active speech level to the root mean square (RMS) level of the noise. This definition is taken from the ITU
standard for Methods for Objective and Subjective Assessment of Quality (ITU-T P.830) [27]

\[
SNR_{dB} = 20 \log_{10} \left( \frac{\text{Active Speech Level}}{\text{RMS value of noise}} \right)
\] (6.1)

6.2 Informal Testing

Informal testing can be carried out initially to give an overview of the performance of a system. Here spectrograms and informal listening tests were used to examine the quality of the enhanced speech compared to the noisy speech.

6.2.1 Performance in White Noise

White, or Gaussian, noise is randomly generated noise type with equal energy per octave.

Figure 6.1 Spectrograms of a (a) clean speech file, (b) speech corrupted by white noise at 10dB and (c) enhanced speech

Figure 6.1 (a) shows a spectrogram of the speech file “Good service should be rewarded by big tips”, while fig. 6.1 (b) shows the same file corrupted by white noise at an SNR of 10 dB. The noise can clearly be seen by comparing the first and second spectrograms. The third spectrogram (c), that of the
enhanced signal, shows that the noise has been completely removed. There is some incorrect attenuation of the noise-like frequencies of the speech, but listening tests show that this is almost imperceptible. The enhanced speech, produced by the SSANE algorithm, is of a good, natural-sounding quality and contains no audible noise whatsoever.

The algorithm was tested for all of the sentences in table 6.1 and for a variety of different SNRs. A summary of the characteristics of the enhanced speech is show in table 6.2

<table>
<thead>
<tr>
<th>SNR</th>
<th>Enhanced Speech Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Natural, clean speech with no audible noise</td>
</tr>
<tr>
<td>10</td>
<td>Natural, clean speech with no audible noise</td>
</tr>
<tr>
<td>5</td>
<td>Slight loss of naturalness, very high intelligibility</td>
</tr>
<tr>
<td>0</td>
<td>Slight distortion, very little noise, very high intelligibility</td>
</tr>
<tr>
<td>-5</td>
<td>Some distortion, a little noise, good intelligibility overall, but some speech loss</td>
</tr>
</tbody>
</table>

Table 6.2 Characteristics of Enhanced Speech from white noise at various SNRs

The SSANE algorithm is well suited to white noise, as it is an extension to the subspace method, which was designed with white noise in mind. It performs very well as low as an SNR of 0dB. However when there is a very high noise level (SNR of -5dB) some speech information is incorrectly attenuated for some samples, which produces a lower level of intelligibility than for higher SNRs.
6.2.2 Performance in different Noise Types

The traditional signal subspace approach works quite well for white noise, since it assumes white noise in its noise reduction model. It typically performs less well in other noise types. The characteristics of the speech enhanced by the SSANE algorithm for a variety of different noise types, each at various signal to noise ratios are examined in this section.

6.2.2.1 Pink Noise

Pink noise is defined as having equal energy per one third of an octave. It proves a more difficult test for most speech enhancement systems than white noise.

![Figure 6.2](image)

**Figure 6.2** Spectrograms of (a) a clean speech file, (b) speech corrupted by pink noise at 5dB and (c) enhanced speech

The spectrograms in figure 6.2 are for the sentence “porcupines resemble sea urchins”. The noise in the fig. 6.2 (b) is at an SNR of 5dB. It can be seen from the SSANE enhanced speech fig. 6.2 (c) that the noise has been fully removed, but a small amount of distortion in produced due to a small over-attenuation of some speech components. The speech is still of quite a high quality with very high intelligibility.
<table>
<thead>
<tr>
<th>SNR</th>
<th>Enhanced Speech Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Clear, natural speech with no audible noise</td>
</tr>
<tr>
<td>10</td>
<td>Slight loss of naturalness, no audible noise, very high</td>
</tr>
<tr>
<td></td>
<td>intelligibility</td>
</tr>
<tr>
<td>5</td>
<td>Slight distortion, very little noise, very high intelligibility</td>
</tr>
<tr>
<td>0</td>
<td>Some distortion, very little noise, very good intelligibility</td>
</tr>
<tr>
<td>-5</td>
<td>Significant distortion, some noise, reasonably good intelligibility</td>
</tr>
</tbody>
</table>

**Table 6.3** Characteristics of Enhanced Speech corrupted by pink noise at various SNRs

The algorithm performs very well in pink noise down as far as an SNR of 0dB, where there is a noticeable, but acceptable, amount of distortion. The level of intelligibility remains high throughout and the speech is generally of very good quality. At an SNR of minus 5 dB the distortion becomes very significant and the speech quality is adversely affected. However, intelligibility remains reasonably good.

### 6.2.2.2 F16 Cockpit Noise

The F16 cockpit noise is a real world noise sample, recorded at the co-pilot's seat in a two-seat F-16, travelling at a speed of 500 knots, and an altitude of 300-600 feet. It is considered to be quite a difficult test for any speech enhancement system, since it is quite obvious to a listener even at a high SNR.

Figure 6.3 shows spectrograms of the sentence “porcupines resemble sea urchins”, fig 6.3 (a), along with the same sentence corrupted with F16 noise at an SNR of 0dB in fig 6.3 (b), and the spectrogram of the SSANE enhanced speech in fig. 6.3 (c). It can be seen that while most of the noise has been removed, a small amount remains. There is also some distortion present. Most of the speech content however has been preserved and intelligibility remains good.
Figure 6.3 Spectrograms of a clean speech file (a), speech corrupted by F16 noise at 0dB (b) enhanced speech (c)

<table>
<thead>
<tr>
<th>SNR</th>
<th>Enhanced Speech Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Very slight unnaturalness, very low noise level, very high intelligibility</td>
</tr>
<tr>
<td>10</td>
<td>Slight distortion, low noise level, very high intelligibility</td>
</tr>
<tr>
<td>5</td>
<td>Some distortion, very little noise, very high intelligibility</td>
</tr>
<tr>
<td>0</td>
<td>Some distortion, some noise, good intelligibility</td>
</tr>
<tr>
<td>-5</td>
<td>Significant distortion, some noise, reasonable intelligibility</td>
</tr>
</tbody>
</table>

Table 6.4 Characteristics of Enhanced Speech from F16 noise at various SNRs

F16 noise does indeed prove to be a sterner test of the system and flaws occur in the enhanced speech even at quite high SNRs. The intelligibility is not adversely affected and for many listeners some distortion is considered a fair trade-off for a high level of noise removal.
6.2.2.3 Jet Engine Noise

Jet engine noise is another real-world signal and again is a difficult test for a speech enhancement system. According to the Noisex-92 documentation [26], the Buccaneer jet was moving at a speed of 190 knots, and an altitude of 1000 feet, with airbrakes out.

![Spectrograms](image)

**Figure 6.4** Spectrograms of (a) a clean speech file, (b) speech corrupted by jet engine noise at 15dB and (c) enhanced speech

From the spectrograms in figure 6.4, for the sentence “the first necessity, at the very outset of war is post attack reconnaissance”, it can be seen that the enhanced speech, figure 6.4 (c) contains virtually no noise and the speech reconstruction is almost perfect.
<table>
<thead>
<tr>
<th>SNR</th>
<th>Enhanced Speech Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Natural sounding speech, very low noise level, very high intelligibility</td>
</tr>
<tr>
<td>10</td>
<td>Slight distortion, very low noise level, very high intelligibility</td>
</tr>
<tr>
<td>5</td>
<td>Slight distortion, very low noise level, very high intelligibility</td>
</tr>
<tr>
<td>0</td>
<td>Some distortion, very little noise, very good intelligibility</td>
</tr>
<tr>
<td>-5</td>
<td>Significant distortion, fairly significant noise, reasonable intelligibility</td>
</tr>
</tbody>
</table>

Table 6.5 Characteristics of Enhanced Speech from jet engine noise at various SNRs

The system performs quite well under these noise conditions, with only a very slight amount of distortion as low as 5dB, which is a reasonably high noise level. The noise is almost completely removed, and what little residual noise is of a spectrum similar to that of the original noise, making it far less perceptually harmful than musical noise produced by many speech enhancement systems.

6.2.2.4 Car Interior Noise

This recording was made at 120 km/h, in 4th gear, on an asphalt road, in rainy conditions in a Volvo car. Many speech enhancement systems perform very well with car interior noise due to the low-pass nature of this noise type.

In figure 6.5 (b) the spectrogram contains car interior noise at an SNR of minus 5dB, and even at this low SNR it is not clearly evident in the spectrogram. Nonetheless, the noise is obvious to the listener. The proposed system is extremely good at removing this type of noise, with excellent quality speech, very little noise and very high intelligibility even lower than minus 5dB.
Figure 6.5 Spectrograms of (a) a clean speech file, (b) speech corrupted by car interior noise at minus 5dB and (c) enhanced speech

<table>
<thead>
<tr>
<th>SNR</th>
<th>Enhanced Speech Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Clear, natural speech with no audible noise</td>
</tr>
<tr>
<td>10</td>
<td>Clear, natural speech with no audible noise</td>
</tr>
<tr>
<td>5</td>
<td>Slight loss of naturalness, no audible noise, very high intelligibility</td>
</tr>
<tr>
<td>0</td>
<td>Slight loss of naturalness, very little audible noise, very high intelligibility</td>
</tr>
<tr>
<td>-5</td>
<td>Slight loss of naturalness, a little noise, very high intelligibility</td>
</tr>
</tbody>
</table>

Table 6.6 Characteristics of Enhanced Speech from car interior noise at various SNRs

6.2.3 Informal Testing Conclusions

It can be seen from the informal testing that the algorithm performs well in most noise types even as far as very low SNRs, producing good quality speech with little or no noise. This shows that the algorithm is a very useful speech enhancement system. However the more significant tests are objective and subjective tests comparing the SSANE algorithm with other speech enhancement algorithms currently in use. These experiments are presented in the following sections.
6.3 Objective Testing

Any speech enhancement system should be tested against other comparable systems under fair conditions using experiments which give a useful and sensible measure of the success of the algorithm in producing speech of good quality. SNR improvement experiments were originally used in testing speech enhancement algorithms, but these tests do not take perceptual criteria into account. Thus there may be an overall reduction in noise throughout the signal, but musical noise artefacts can remain. This musical noise renders the speech more annoying to the listener than the noisy signal. SNR improvement tests have poor correlation with listening tests. For this reason the Perceptual Evaluation of Speech Quality (PESQ) system, under the ITU-T P.862 standard [28], has been used in several recent papers [29, 30, 31, 32] to test the proposed systems, and will be used here as the objective testing measurement technique.

6.3.1 PESQ Algorithm Description

According to [32], and references therein, PESQ provides a 93.5% correlation with subjective testing results, as opposed to 24% for simple SNR tests and 77% for segmental SNR tests, hence it is a logical choice of objective tests. Although the PESQ algorithm was not designed specifically for the task of evaluating speech enhancement systems, its design lends itself well to this purpose. Figure 6.6 shows the block diagram of the PESQ algorithm, which consists of several distinct stages.

![Figure 6.6 Block diagram of PESQ algorithm [33]](image-url)
Firstly, the clean reference speech file and the enhanced speech from the system under test are input into the system. Then the signals are pre-processed to level and time align the two signals, so that they are at the same listening volume and aligned in time to allow comparison between the same samples for both files.

Next a perceptual model is applied to map the two signals to a perceived loudness scale. The magnitude responses of the spectra are separated into forty-two groupings spaced in perceptual frequency. Short term energy fluctuations are smoothed and clipped over time.

The perceived distortion is calculated by measuring the absolute difference of the loudness between the clean and enhanced signals. The algorithm produces a symmetric and an asymmetric measure output at this point. These are used to decide whether the distortions are additions or deletions to the signal. Additions (musical noise) are much harsher on the human auditory system than deletions, and as such have a higher influence on the final PESQ score.

Finally, the system outputs a PESQ score for the pair of signals, somewhere in the range of -0.5 to 4.5, with higher values signifying better quality speech, a score of 4.5 being a perfect, flawless reconstruction.

6.3.2 Test Description

To evaluate the SSANE algorithm in a relevant context it must be examined against other contemporary algorithms, under a variety of conditions. For this purpose, the system will be compared with noisy signals, for reference, and with three other speech enhancement algorithms: spectral subtraction, wiener filtering with adaptive noise estimation (described in section 3.3.3) and regular signal subspace speech enhancement with a pre-whitening filter (chapter 4).

The reason for the choice of spectral subtraction is that it is the earliest and most widely known speech enhancement system. Perceptual wiener filtering
with adaptive noise estimation is used as the proposed system employs a modification of its noise estimation scheme, so it is a logical choice for comparison. The subspace method is also chosen as it forms a large basis for the algorithm presented in this thesis. Their implementations are outlined below.

All tests were performed with the test data as described in section 6.1, with SNRs chosen as -5, 0, 5, 10 and 15dB and using white, pink, F16 cockpit and jet engine noise samples as the additive noise types. This resulted in approximately 800 PESQ scores being calculated in total between each of the algorithms and the noisy speech samples.

### 6.3.2.1 Spectral Subtraction Implementation

The simple spectral subtraction of Boll [1], as described in section 2.2 was implemented here, using the actual noise spectrum as its noise estimate. The frame size was chosen as 256 samples and each frame was hanning windowed and 50% overlap added to reconstruct the signal.

### 6.3.2.2 Perceptual Wiener Filter with Adaptive Noise Estimation Implementation

The perceptual sub-band wiener filter with adaptive noise estimation (section 3.3.3) was implemented with a gammatone filter-bank as a front end, a frame size of 256 samples, $Q$ chosen as 5, $\alpha_{\text{final}}$ chosen as 0.7 and the masking model similar to the MPEG masking model [11] used to calculate the masking threshold.

### 6.3.2.3 Subspace Implementation

The subspace algorithm was implemented with a pre-whitening filter, as discussed in section 4.3, with $K$ chosen as 40 with the noisy covariance matrix calculated from 400 samples of the noisy speech. A perfect voice activity detector (possible due to the actual noise being available for these tests) was used to update the noise estimate in noise-only frames. In addition the signal
estimator in equation 4.39, with $v=5$ was also used to reconstruct the enhanced signal.

The proposed algorithm was implemented as discussed in section 5.4

**6.3.3 PESQ Results**

Each of the 8 speech files in table 6.1 was corrupted with different noise types at various SNRs as described above. The results are presented in graph form for each of the four implemented noise types, and their results are analysed.

Henceforth, for convenience, the algorithms will be shortened to SpecSub for Spectral Subtraction, Subspace for the regular subspace with pre-whitening method, WienerANE for the perceptual wiener filter with adaptive noise estimation and SSANE for the proposed signal subspace with adaptive noise estimation algorithm.

The level of performance increase is not always hugely obvious by a direct comparison of PESQ scores, so they are presented along with a delta PESQ (as used in [32]) chart in each case here.

Delta PESQ is found by calculating the percentage improvement over the reference noisy PESQ score:

$$\text{Delta PESQ} = \left( \frac{PESQ_{\text{sig}} - PESQ_{\text{ref}}}{PESQ_{\text{ref}}} \right) \times 100\%$$  \hspace{1cm} (6.2)

Where $PESQ_{\text{sig}}$ is the PESQ score of the signal being tested and $PESQ_{\text{ref}}$ is the PESQ score of the same signal without any enhancement, i.e. the noisy signal.
### 6.3.3.1 White Noise PESQ Scores

The first set of results show the average PESQ scores for speech corrupted with white noise at the aforementioned SNRs, and the delta PESQ score for the same signals.

![Graph of PESQ score for each of the test algorithms in white noise at various SNRs](image)

*Figure 6.7 (a)* Graph of PESQ score for each of the test algorithms in white noise at various SNRs

![Graph of delta PESQ for the same samples](image)

*Figure 6.7 (b)* Graph of delta PESQ for the same samples

It can be seen from the graphs in figure 6.7 that the proposed algorithm outperforms all of the other algorithms for white noise under all of the SNR values, with a delta PESQ score of 75% from -5dB up to 10dB and 60% for 15dB. The traditional subspace performs next best here, with improvements of 40% to over 60% in some cases, but is well below the SSANE results in every
case. Even though the subspace method was designed with white noise in mind, it is outperformed by the SSANE algorithm here. This is because the adaptive noise estimation algorithm allows the noise estimate to be updated throughout the signal, and not just in speech-free frames.

The WienerANE system performs less well than expected for this test with improvements of only 18% to 30%. Perhaps this is because subspace domain speech enhancement is more suited to this noise type than frequency domain speech enhancement, since musical noise is more often produced in the frequency domain.

Interestingly, for higher SNRs the SpecSub algorithms performs poorly in this situation, actually lowering the speech quality for the 10dB and 15dB cases. This is due to perceptually annoying musical noise production with this algorithm.
6.3.3.2 Pink Noise PESQ Scores

The tests were repeated with pink noise for the same SNRs and with the same speech files as the previous experiment.

![Figure 6.8 (a) Graph of PESQ score for each of the test algorithms in pink noise at various SNRs](image)

![Figure 6.8 (b) Graph of delta PESQ for the same samples](image)

The results again show that overall the SSANE algorithm performs better than the others in this situation with PESQ score improvements of 30% to 60%, with the exception of the subspace with pre-whitening algorithm, which...
performs slightly (5%) better at 15dB. This anomaly is surprising as the new algorithm is inherently more suited to coloured noise than the traditional subspace method. One possible reason for this is that the pre-whitening filter may work well for this noise type at this SNR. Regardless, the new system produces better scores for every other SNR against the regular subspace system, and for every single SNR against all the other algorithms.

Again, the spectral subtraction algorithm gives poor results, creating very noticeable musical noise. The perceptual wiener filter performs reasonably well through, providing an enhancement of up to 20%.
6.3.3.3 F16 Cockpit Noise PESQ Scores

F16 cockpit noise, as previously mentioned is a real-world noise type, and is a difficult test for any algorithm.

![Graph of PESQ score for each of the test algorithms in F16 cockpit noise at various SNRs](image)

**Figure 6.9 (a)** Graph of PESQ score for each of the test algorithms in F16 cockpit noise at various SNRs

![Graph of delta PESQ for the same samples](image)

**Figure 6.9 (b)** Graph of delta PESQ for the same samples

It can be seen that the SSANE algorithm gives the best results for this noise type also, giving average improvements of 20% to almost 40%, which is better than any of the others for all SNR values.
The subspace and spectral subtraction algorithms give similar results to each other here, with each producing better scores than the other for different SNRs.

### 6.3.3.4 Jet Engine Noise PESQ Scores

The fourth, and final, noise type evaluated is jet engine noise.

![Graph of PESQ score for each of the test algorithms in Jet engine noise at various SNRs](image)

**Figure 6.10 (a)** Graph of PESQ score for each of the test algorithms in Jet engine noise at various SNRs

![Graph of delta PESQ for the same samples](image)

**Figure 6.10 (b)** of delta PESQ for the same samples
The results are consistent with the previous tests in this case, with the proposed algorithm obtaining the highest scores throughout, with an even higher relative margin of improvement over the other algorithms than previous noise types.

The perceptual wiener filter performs next best in this situation, with the regular subspace method only giving a relatively small improvement over the noisy signal.

### 6.3.4 Objective Testing Conclusions

The objective tests show than the proposed signal subspace with adaptive noise estimation algorithm performs better than contemporary speech enhancement systems. Since the PESQ score represent a perceptual speech quality measure, it can be inferred that the new algorithm gives improved speech quality over the other systems. This was shown to be the case for all four tested noise types.

### 6.4 Subjective Testing

Speech enhancement must ultimately satisfy the goal of improving speech quality for the human listener, and as such subjective tests are necessary to evaluate the performance of any system. The conditions needed for formal testing are very stringent and time consuming, so informal A-B testing is used in this thesis to attempt to confirm the objective test results.

### 6.4.1 A-B Testing

A-B testing involves a group of people to listening to a number of pairs of speech files (labeled A and B in each case) and deciding which they think is better in each case. A choice of ‘undecided’ is also allowed where the listeners are unsure which sample is superior.
Four separate tests, each consisting of four different parts were set up. In each part of the four tests the SSANE algorithm was paired with either noisy speech, or speech enhanced with one of the other algorithms used in the objective tests. This was repeated with the four noise types used previously, each one chosen with a different SNR, to form sixteen pairs of speech files. The order of the files in each case was randomised to prevent bias.

Fourteen people took part in the tests. None of the listeners were previously familiar with the sentences used in the tests. The tests were chosen to provide a cross-section of different noises and SNRs to give an overview of the relative performance of the algorithm in listening tests.

### 6.4.2 A-B Testing Results

The percentage of preferred votes for each algorithm is shown in this section and the results are analysed.

#### 6.4.2.1 A-B Results for White Noise

The first set of tests used a speech file corrupted with white noise at 10dB. The file was enhanced with the SSANE algorithm, the spectral subtraction algorithm and the perceptual wiener filter with adaptive noise estimation.

<table>
<thead>
<tr>
<th>Test</th>
<th>Listener Preference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SSANE</td>
</tr>
<tr>
<td>SSANE / Noisy Signal</td>
<td>93%</td>
</tr>
<tr>
<td>SSANE / Spectral Subtraction</td>
<td>100%</td>
</tr>
<tr>
<td>SSANE / Subspace</td>
<td>100%</td>
</tr>
<tr>
<td>SSANE/ WienerANE</td>
<td>93%</td>
</tr>
</tbody>
</table>

*Table 6.7 Listener preferences for tests in white noise at 10dB*

These tests show an almost unanimous preference for the proposed algorithm over all the others. Unfortunately due to the relatively small number of
listeners, one vote accounts for a difference of 7%. It still remains evident though that the SSANE algorithm is strongly preferred over all the others.

6.4.2.2 A-B Results for Pink Noise

The second test used speech corrupted with pink noise at 5dB.

<table>
<thead>
<tr>
<th>Test</th>
<th>Listener Preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSANE / Noisy Signal</td>
<td>86% 14% 0%</td>
</tr>
<tr>
<td>SSANE / Spectral Subtraction</td>
<td>100% 0% 0%</td>
</tr>
<tr>
<td>SSANE / Subspace</td>
<td>86% 14% 0%</td>
</tr>
<tr>
<td>SSANE/ WienerANE</td>
<td>93% 0% 7%</td>
</tr>
</tbody>
</table>

Table 6.8 Listener preferences for tests in pink noise at 5dB

This test again shows a strong preference for the SSANE algorithm, which is in keeping with the subjective tests. The level of preference for the SSANE algorithm is perhaps a little higher than expected against the subspace algorithm given that the PESQ scores were quite similar for these two algorithms at 5dB. This may be accounted for by the low number of listeners.

6.4.2.3 A-B Results for F16 Noise

In the third test the speech was corrupted with F16 cockpit noise at a very low SNR of 0dB.

<table>
<thead>
<tr>
<th>Test</th>
<th>Listener Preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSANE / Noisy Signal</td>
<td>36% 64% 0%</td>
</tr>
<tr>
<td>SSANE / Spectral Subtraction</td>
<td>58% 21% 21%</td>
</tr>
<tr>
<td>SSANE / Subspace</td>
<td>58% 7% 35%</td>
</tr>
<tr>
<td>SSANE/ WienerANE</td>
<td>58% 14% 28%</td>
</tr>
</tbody>
</table>

Table 6.9 Listener preferences for tests in f16 cockpit noise at 0dB
Listeners showed a preference for the noisy signal over the enhanced signal in this situation. The intelligibility of the enhanced speech is reduced for this noise type at this low SNR and thus listeners have chosen the noisy signal over the enhanced one. The SSANE algorithm is still preferred to the other speech enhancement algorithms for this test.

### 6.4.2.4 A-B Results for Jet Engine Noise

Finally, the speech was corrupted with jet engine noise at 15dB.

<table>
<thead>
<tr>
<th>Test</th>
<th>Listener Preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSANE / Noisy Signal</td>
<td>100% 0% 0%</td>
</tr>
<tr>
<td>SSANE / Spectral Subtraction</td>
<td>93% 7% 0%</td>
</tr>
<tr>
<td>SSANE / Subspace</td>
<td>71% 7% 22%</td>
</tr>
<tr>
<td>SSANE/ WienerANE</td>
<td>71% 22% 7%</td>
</tr>
</tbody>
</table>

*Table 6.10* Listener preferences for tests in jet engine noise at 15dB

It is evident that the proposed algorithm gives the best performance, but there is a weaker preference for it over the subspace and perceptual wiener filter algorithms in this case. This is due to the fact that all three of these algorithms give good quality speech without much noise at this SNR. The small amount of musical noise or distortion in the other two account for the preference of the SSANE with most listeners, but is unnoticed by some.
6.4.2.5 Overall A-B Testing Results

Combining the results from the four A-B listening tests gives an overall listening test result.

<table>
<thead>
<tr>
<th>Test</th>
<th>Listener Preference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SSANE</td>
</tr>
<tr>
<td>SSANE / Noisy Signal</td>
<td>79%</td>
</tr>
<tr>
<td>SSANE / Spectral Subtraction</td>
<td>88%</td>
</tr>
<tr>
<td>SSANE / Subspace</td>
<td>80%</td>
</tr>
<tr>
<td>SSANE/ WienerANE</td>
<td>79%</td>
</tr>
</tbody>
</table>

*Table 6.11 Overall listener preferences*

It can be clearly seen that the proposed SSANE algorithm is strongly preferred over all of the other methods, which is consistent with the objective PESQ score, suggesting that it is a powerful algorithm, which could have useful applications in speech processing.

6.5 Summary

This chapter presented a series of tests for the new Signal Subspace with Adaptive Noise Estimation system. Firstly, the informal listening tests described the characteristics of speech enhanced from noisy speech signal at numerous signal to noise ratios and under various noise types. Objective testing was carried out using the Perceptual Evaluation of Speech Quality test, and the results for the new algorithm were compared with those of a number of other contemporary speech enhancement methods. Finally subjective listening test results were presented and examined. The SSANE algorithm was shown to outperform all of the others in the majority of tests.
Chapter 7: Conclusion

7.1 Conclusion

This thesis has focused on the design, implementation and testing of an adaptive noise estimation algorithm for signal subspace speech enhancement. This is a novel approach to the subspace method [5] which traditionally uses voice activity detection to estimate the noise in a signal. The proposed method requires no voice activity detection and thus can update the noise estimate throughout the signal instead of being limited to silence intervals. This allows a more accurate noise estimate to be produced and improves the quality of the enhanced speech.

Objective and subjective tests were carried out to evaluate the success of the proposed algorithm. The results were compared with those of contemporary speech enhancement systems and were shown to outperform these systems for the majority of situations. The proposed algorithm was shown to produce good quality speech in most noise types even at low signal to noise ratios.

The proposed system has potential applications in cellular telephony, audio archive restoration and automatic speech recognition. All of these applications are heavily reliant on accurate and robust noise estimation to provide high quality enhanced speech. Thus the proposed method is an ideal speech enhancement algorithm for these situations.

7.2 Future Work

Recent developments is subspace based speech enhancement, such as Klein and Kabal’s perceptual post filter [22], and the work of Jabloun and Champagne in [34] have involved the exploitation of auditory masking properties. The algorithm in this paper does not make use of these properties but they could be incorporated relatively easily. This could potentially result in a further increase in system performance.
The subspace method is also rather computationally complex. Future work should also focus on the reduction of this complexity. The discrete cosine transform was been proposed as an alternative to the computationally complex KLT transform, and single value decomposition is another option for reducing complexity.

This will be significant as speech enhancement algorithms require in real-time implementation for some applications, with more efficient algorithms allowing less power consumption and processor usage.
Appendices

Appendix A: Matlab Code for SSANE Algorithm
Appendix B: PESQ Scores
Appendix A: Matlab Code for SSANE Algorithm

A1: SSANE_Script.m

% Test Script For Signal Subspace with Adaptive Noise Estimation program

% Load Audio Files

speech_file = wavread(’C:\Noisy_speech_files\SpEARdatabase\Timit\necessity_16.wav’);
noise_file = wavread(’white.wav’);

% resample noise file from 20kHz to 16kHz
noise_file = resample(noise_file,16,20);

% add noise to speech file at specified SNR
[noisy,noise] = noisy_speech(speech_file,noise_file,10,16000);

% Normalise noisy speech
noisy = noisy*100;

% Call SSANE cleaning algorithm

disp(’’)
disp(’-----Starting Enhancement Algorithm-----’)
disp(’’);

enhanced = ssane_enhance(noisy,16000);

% Re-normalise speech and noisy speech
enhanced = enhanced/100;
noisy = noisy/100;

% Display Results

figure(1);
x=spectrogram(speech_file, 256, 2, 1);
y=spectrogram(noisy, 256, 2, 1);
z=spectrogram(enhanced, 256, 2, 1);
subplot(3,1,1)
imagesc(x)
subplot(3,1,2)
imagesc(y)
subplot(3,1,3)
imagesc(z)
colormap(1-gray);
% Declare Global Variables

global ssane_frame_length;
global ssane_overlap;
global ssane_total_size;
global ssane_num_frames;

% Specify Covariance matrix size and amount of overlap

ssane_overlap = 20;
ssane_frame_length =20;
ssane_total_size = ssane_overlap + ssane_frame_length; % K=40

% Calculate Signal Length

signal_length = length(noisy);

% initialise clean signal array

enhanced=zeros(signal_length,1);

% Calculate Number of Frames

ssane_num_frames = ceil(signal_length/ssane_frame_length);

% Specify Autocorrelation Window Size

window_size = 400;

% Split noisy signal into equal length blocks

[blocks] = buffer(noisy,ssane_total_size,ssane_overlap);

Ew_prev = ones(ssane_total_size,ssane_total_size);
Ey_prev = ones(ssane_total_size,ssane_total_size);
Ew_avg = ones(ssane_total_size,ssane_total_size);

% Run loop until all frames have been processed
for i = 1:ssane_num_frames-2

% Define Window For Covariance Estimate
block_start = (i-1)*ssane_frame_length + 1;
block_end = min(block_start + ssane_total_size - 1,signal_length);
N_start = max((block_start - (window_size-ssane_total_size)/2),1);
N_end = min((block_end + (window_size-
ssane_total_size)/2),signal_length);

% Estimate Noisy Covariance Matrix
current_signal_window = noisy(N_start:N_end);
occurrent_signal_covariance =
cov_estimate(current_signal_window,ssane_total_size);

% Estimate Clean Eigenvectors
[U,E] = schur(noisy_signal_covariance);

E = real(diag(E));

%initialise eigenvector estimates to the values of the first frame
if i==2
Ew_prev = E; %previous noise eigenvalue estimate
Ey_prev = E.*1.0001; %Previous value of noisy eigenvalue
Ew_avg = E; %average noise eigenvalue estimate
end

% call noise estimation function, which also calculates the denoising
% gain
[G, Ew_cur, Ey_cur, avg] = find_gain(E, Ew_prev, Ey_prev, Ew_avg, ssane_total_size);

Ew_avg = avg;       % new average value
Ew_prev = Ew_cur;   % new previous value is set as current value
Ey_prev1 = Ey_cur;  % new previous value is set as current value

% reorder eigenvectors and eigenvectors into descending order
[holder,index] = sortrows([G transpose(U)],1);
G = flipud(holder(:,1));
U = fliplr(transpose(holder(:,2:end)));

% Estimate Order of Speech

m = estimate_m(G,1);

% Create Speech Enhancement Filter for the Current Frame based on Calculated Gain
filt = U*(diag([G(1:m, 1) ; zeros(ssane_total_size - (m),1)])*U';

% Hanning Window and Overlap-Add Frames
temp_block(:,1)=(filt*blocks(:,i)).*hanning(40);
enhanced(block_start:block_end)=enhanced(block_start:block_end) + temp_block(:,1);

disp(['Frame ' num2str(i) ' of ' num2str(ssane_num_frames)])
end;

function signal_covariance = cov_estimate(data,M)

if (length(data) < M)
data = [data ; zeros(M - length(data),1)];
end;

X = corrmtx(data,M-1,'covariance');
signal_covariance = (X'*X);

% End of cov_estimate

% Theoretical Estimate

function [order] = estimate_m(G,noise_var)
order = length(find(G>0.0));

% End of estimate_m
A3: find_gain.m

function [G, Ew_cur, Ey_cur, Ew_avg] = find_gain(Ey, Ew_prev, Ey_prev, Ew_avg, matrix_size)

% FIND_GAIN finds the noise suppression gain for the current frame after calculating the noise estimate for the current frame using the noise estimation algorithm

%FIND_GAIN returns the, gain G, the current noise eigenvalue estimate %Ew_cur, the current noisy eigenvalue Ey_cur and the average noise %eigenvalue estimate over a specified number of frames Ew_avg

Q=5;
alpha_final = 0.95;
sample_frames = 15;
alpha = zeros(matrix_size,1);
G = zeros(matrix_size, 1);
Ew_cur = zeros(matrix_size, 1);
Ew = zeros(matrix_size,1);
Ey_cur = zeros(matrix_size, 1);

for i=1:matrix_size
alpha(i,1) = 1 - min(1, (Ey(i,1)/Ew_avg(i,1))^(-Q));
Ew(i,1) = alpha(i,1)*Ew_prev(i,1) + (1 - alpha(i,1))*Ey(i,1);
Ew_cur(i,1) = alpha_final*Ew_prev(i,1) + (1 - alpha_final)*Ew(i,1);
Ew_avg(i,1) = ((sample_frames -1)/sample_frames)*Ew_avg(i,1) + (1 - ((sample_frames -1)/sample_frames))*Ew(i,1);
Ew_avg(i,1) = max(1, Ew_avg(i,1));
Ey_cur(i,1) = Ey(i,1);
G(i,1) = (exp(-5*Ew_cur(i,1)/Ey_cur(i,1)));
G(i,1) = max(real(G(i,1)), 0.0);
end
Appendix B: PESQ Scores

B1: White Noise

SSANE:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>1.803</td>
<td>2.045</td>
<td>2.474</td>
<td>2.894</td>
<td>3.258</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.786</td>
<td>2.168</td>
<td>2.549</td>
<td>3.083</td>
<td>3.013</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.757</td>
<td>2.168</td>
<td>2.469</td>
<td>3.058</td>
<td>3.032</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>2.057</td>
<td>2.346</td>
<td>2.611</td>
<td>2.845</td>
<td>3.042</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.793</td>
<td>2.151</td>
<td>2.695</td>
<td>3.022</td>
<td>3.147</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.622</td>
<td>2.097</td>
<td>2.684</td>
<td>2.991</td>
<td>3.448</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.793</td>
<td>2.203</td>
<td>2.577</td>
<td>2.882</td>
<td>3.002</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>1.869</td>
<td>2.117</td>
<td>2.451</td>
<td>2.894</td>
<td>3.32</td>
</tr>
</tbody>
</table>

TOTAL PESQ SCORE  | 14.48| 17.295| 20.51 | 23.669| 25.262|

AVERAGE PESQ SCORE | 1.81 | 2.161875 | 2.56375 | 2.958625 | 3.15775 |

Noisy:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>0.894</td>
<td>1.241</td>
<td>1.67</td>
<td>2.112</td>
<td>2.519</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>0.951</td>
<td>1.329</td>
<td>1.772</td>
<td>2.21</td>
<td>2.608</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>0.968</td>
<td>1.336</td>
<td>1.785</td>
<td>2.209</td>
<td>2.616</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.586</td>
<td>1.88</td>
<td>2.176</td>
<td>2.444</td>
<td>2.659</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>0.986</td>
<td>1.367</td>
<td>1.831</td>
<td>2.261</td>
<td>2.603</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.527</td>
<td>1.788</td>
<td>2.084</td>
<td>2.386</td>
<td>2.701</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.079</td>
<td>1.401</td>
<td>1.757</td>
<td>2.097</td>
<td>2.423</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>0.612</td>
<td>1.013</td>
<td>1.505</td>
<td>1.969</td>
<td>2.365</td>
</tr>
</tbody>
</table>

TOTAL PESQ SCORE | 8.603| 11.355| 14.58 | 17.688| 20.494|

AVERAGE PESQ SCORE | 1.075375 | 1.419375 | 1.8225 | 2.211 | 2.56175 |
### Spectral Subtraction:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>1.143</td>
<td>1.5</td>
<td>1.827</td>
<td>2.05</td>
<td>2.193</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.162</td>
<td>1.46</td>
<td>1.757</td>
<td>2.005</td>
<td>2.151</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.264</td>
<td>1.571</td>
<td>1.789</td>
<td>1.935</td>
<td>2.034</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.568</td>
<td>1.693</td>
<td>1.762</td>
<td>1.819</td>
<td>1.858</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.247</td>
<td>1.628</td>
<td>1.923</td>
<td>2.084</td>
<td>2.182</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.605</td>
<td>1.793</td>
<td>1.98</td>
<td>2.15</td>
<td>2.255</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.406</td>
<td>1.655</td>
<td>1.859</td>
<td>2.021</td>
<td>2.147</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>0.976</td>
<td>1.328</td>
<td>1.627</td>
<td>1.81</td>
<td>1.918</td>
</tr>
</tbody>
</table>

**TOTAL PESQ SCORE**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10.371</td>
</tr>
<tr>
<td></td>
<td>12.628</td>
</tr>
<tr>
<td></td>
<td>14.524</td>
</tr>
<tr>
<td></td>
<td>15.874</td>
</tr>
<tr>
<td></td>
<td>16.738</td>
</tr>
</tbody>
</table>

**AVERAGE PESQ SCORE**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.296375</td>
</tr>
<tr>
<td></td>
<td>1.5785</td>
</tr>
<tr>
<td></td>
<td>1.8155</td>
</tr>
<tr>
<td></td>
<td>1.98425</td>
</tr>
<tr>
<td></td>
<td>2.09225</td>
</tr>
</tbody>
</table>

### Subspace:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>1.609</td>
<td>2.013</td>
<td>2.412</td>
<td>2.653</td>
<td>2.909</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.69</td>
<td>2.061</td>
<td>2.35</td>
<td>2.794</td>
<td>3.258</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.469</td>
<td>2.14</td>
<td>2.396</td>
<td>2.796</td>
<td>2.848</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>2.107</td>
<td>2.201</td>
<td>2.389</td>
<td>2.739</td>
<td>3.073</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.538</td>
<td>1.965</td>
<td>2.38</td>
<td>2.597</td>
<td>2.784</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.542</td>
<td>2.109</td>
<td>2.593</td>
<td>2.87</td>
<td>3.204</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.478</td>
<td>1.891</td>
<td>2.268</td>
<td>2.466</td>
<td>2.735</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>1.319</td>
<td>1.916</td>
<td>2.182</td>
<td>2.431</td>
<td>2.684</td>
</tr>
</tbody>
</table>

**TOTAL PESQ SCORE**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12.752</td>
</tr>
<tr>
<td></td>
<td>16.296</td>
</tr>
<tr>
<td></td>
<td>18.97</td>
</tr>
<tr>
<td></td>
<td>21.346</td>
</tr>
<tr>
<td></td>
<td>23.495</td>
</tr>
</tbody>
</table>

**AVERAGE PESQ SCORE**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.594</td>
</tr>
<tr>
<td></td>
<td>2.037</td>
</tr>
<tr>
<td></td>
<td>2.37125</td>
</tr>
<tr>
<td></td>
<td>2.66825</td>
</tr>
<tr>
<td></td>
<td>2.936875</td>
</tr>
</tbody>
</table>
WienerANE:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>1.112</td>
<td>1.567</td>
<td>2.01</td>
<td>2.398</td>
<td>2.743</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.229</td>
<td>1.559</td>
<td>1.994</td>
<td>2.42</td>
<td>2.918</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.246</td>
<td>1.64</td>
<td>1.999</td>
<td>2.429</td>
<td>2.936</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.099</td>
<td>1.895</td>
<td>2.233</td>
<td>2.581</td>
<td>2.863</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.329</td>
<td>1.754</td>
<td>2.166</td>
<td>2.54</td>
<td>2.94</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.506</td>
<td>1.795</td>
<td>2.183</td>
<td>2.568</td>
<td>2.859</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.296</td>
<td>1.647</td>
<td>1.993</td>
<td>2.34</td>
<td>2.785</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>1.023</td>
<td>1.384</td>
<td>1.851</td>
<td>2.313</td>
<td>2.671</td>
</tr>
</tbody>
</table>

**TOTAL PESQ SCORE**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

**AVERAGE PESQ SCORE**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.23</td>
<td>1.655125</td>
<td>2.053625</td>
<td>2.448625</td>
<td>2.839375</td>
</tr>
</tbody>
</table>
B2: Pink Noise

SSANE:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>1.659</td>
<td>2.005</td>
<td>2.423</td>
<td>2.711</td>
<td>3.031</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.759</td>
<td>2.08</td>
<td>2.407</td>
<td>2.626</td>
<td>2.954</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.801</td>
<td>2.128</td>
<td>2.461</td>
<td>2.689</td>
<td>2.975</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.871</td>
<td>2.321</td>
<td>2.478</td>
<td>2.585</td>
<td>2.807</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.756</td>
<td>2.061</td>
<td>2.33</td>
<td>2.546</td>
<td>2.75</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.699</td>
<td>2.055</td>
<td>2.446</td>
<td>2.856</td>
<td>3.108</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.75</td>
<td>2.099</td>
<td>2.474</td>
<td>2.727</td>
<td>2.878</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>1.464</td>
<td>1.929</td>
<td>2.279</td>
<td>2.623</td>
<td>2.883</td>
</tr>
</tbody>
</table>


AVERAGE PESQ SCORE | 1.719875 | 2.08475 | 2.41225 | 2.670375 | 2.92325 |

Noisy:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>0.95</td>
<td>1.299</td>
<td>1.779</td>
<td>2.189</td>
<td>2.571</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.068</td>
<td>1.45</td>
<td>1.904</td>
<td>2.283</td>
<td>2.659</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.082</td>
<td>1.456</td>
<td>1.895</td>
<td>2.291</td>
<td>2.673</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.635</td>
<td>1.91</td>
<td>2.247</td>
<td>2.441</td>
<td>2.671</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.111</td>
<td>1.482</td>
<td>1.915</td>
<td>2.287</td>
<td>2.628</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.466</td>
<td>1.77</td>
<td>2.111</td>
<td>2.422</td>
<td>2.751</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.121</td>
<td>1.471</td>
<td>1.818</td>
<td>2.187</td>
<td>2.507</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>0.663</td>
<td>1.088</td>
<td>1.601</td>
<td>1.994</td>
<td>2.375</td>
</tr>
</tbody>
</table>

TOTAL PESQ SCORE | 9.096 | 11.926 | 15.27 | 18.094 | 20.835 |

AVERAGE PESQ SCORE | 1.137 | 1.49075 | 1.90875 | 2.26175 | 2.604375 |
### Spectral Subtraction:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>0.867</td>
<td>1.202</td>
<td>1.468</td>
<td>1.591</td>
<td>1.663</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.119</td>
<td>1.256</td>
<td>1.434</td>
<td>1.563</td>
<td>1.661</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.205</td>
<td>1.521</td>
<td>1.807</td>
<td>1.984</td>
<td>2.082</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.358</td>
<td>1.545</td>
<td>1.728</td>
<td>1.829</td>
<td>1.886</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.076</td>
<td>1.38</td>
<td>1.542</td>
<td>1.615</td>
<td>1.627</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.562</td>
<td>1.735</td>
<td>1.862</td>
<td>1.962</td>
<td>2.017</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.115</td>
<td>1.291</td>
<td>1.454</td>
<td>1.6</td>
<td>1.722</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>0.604</td>
<td>0.928</td>
<td>1.206</td>
<td>1.383</td>
<td>1.454</td>
</tr>
</tbody>
</table>


**AVERAGE PESQ SCORE**: 1.11325 1.35725 1.562625 1.690875 1.764

### Subspace:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>1.479</td>
<td>2.133</td>
<td>2.591</td>
<td>2.913</td>
<td>3.153</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.19</td>
<td>1.751</td>
<td>2.227</td>
<td>2.622</td>
<td>3.155</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.677</td>
<td>1.953</td>
<td>2.351</td>
<td>2.728</td>
<td>3.072</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.651</td>
<td>2.011</td>
<td>2.468</td>
<td>2.797</td>
<td>2.836</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.355</td>
<td>2.038</td>
<td>2.363</td>
<td>2.792</td>
<td>3.104</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.41</td>
<td>1.812</td>
<td>2.16</td>
<td>2.409</td>
<td>2.74</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>1.148</td>
<td>1.744</td>
<td>2.112</td>
<td>2.274</td>
<td>2.55</td>
</tr>
</tbody>
</table>

**TOTAL PESQ SCORE**: 11.044 15.265 18.679 21.227 23.73

**AVERAGE PESQ SCORE**: 1.3805 1.908125 2.334875 2.653375 2.96625
**WienerANE:**

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>1.138</td>
<td>1.461</td>
<td>2.053</td>
<td>2.343</td>
<td>2.772</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.096</td>
<td>1.498</td>
<td>2.052</td>
<td>2.353</td>
<td>2.834</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.191</td>
<td>1.571</td>
<td>1.96</td>
<td>2.342</td>
<td>2.815</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.058</td>
<td>1.881</td>
<td>2.188</td>
<td>2.525</td>
<td>2.784</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.275</td>
<td>1.707</td>
<td>2.111</td>
<td>2.422</td>
<td>2.806</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.409</td>
<td>1.747</td>
<td>2.137</td>
<td>2.508</td>
<td>2.9</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.277</td>
<td>1.692</td>
<td>2.044</td>
<td>2.326</td>
<td>2.688</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>0.92</td>
<td>1.454</td>
<td>1.947</td>
<td>2.31</td>
<td>2.619</td>
</tr>
</tbody>
</table>

**TOTAL PESQ SCORE**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

**AVERAGE PESQ SCORE**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.1705</td>
<td>1.626375</td>
<td>2.0615</td>
<td>2.391125</td>
<td>2.77725</td>
</tr>
</tbody>
</table>
## B3: F16 Noise

### SSANE:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>1.341</td>
<td>1.834</td>
<td>2.281</td>
<td>2.626</td>
<td>2.916</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.39</td>
<td>1.897</td>
<td>2.203</td>
<td>2.558</td>
<td>2.851</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.372</td>
<td>1.928</td>
<td>2.341</td>
<td>2.599</td>
<td>2.811</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.885</td>
<td>2.154</td>
<td>2.369</td>
<td>2.586</td>
<td>2.745</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.537</td>
<td>1.954</td>
<td>2.157</td>
<td>2.449</td>
<td>2.722</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.444</td>
<td>1.854</td>
<td>2.37</td>
<td>2.739</td>
<td>3.073</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.243</td>
<td>1.915</td>
<td>2.212</td>
<td>2.528</td>
<td>2.857</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>1.224</td>
<td>1.758</td>
<td>2.097</td>
<td>2.451</td>
<td>2.774</td>
</tr>
</tbody>
</table>

**TOTAL PESQ SCORE**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11.436</td>
<td>15.294</td>
<td>18.03</td>
<td>20.536</td>
<td>22.749</td>
</tr>
</tbody>
</table>

**AVERAGE PESQ SCORE**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.4295</td>
</tr>
</tbody>
</table>

### Noisy:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>0.942</td>
<td>1.305</td>
<td>1.746</td>
<td>2.173</td>
<td>2.539</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.252</td>
<td>1.572</td>
<td>1.963</td>
<td>2.34</td>
<td>2.712</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.087</td>
<td>1.435</td>
<td>1.873</td>
<td>2.437</td>
<td>2.313</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.667</td>
<td>1.954</td>
<td>2.243</td>
<td>2.481</td>
<td>2.701</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.26</td>
<td>1.57</td>
<td>1.955</td>
<td>2.306</td>
<td>2.611</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.559</td>
<td>1.848</td>
<td>2.17</td>
<td>2.483</td>
<td>2.788</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.253</td>
<td>1.556</td>
<td>1.902</td>
<td>2.218</td>
<td>2.545</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>0.683</td>
<td>1.073</td>
<td>1.558</td>
<td>1.993</td>
<td>2.376</td>
</tr>
</tbody>
</table>

**TOTAL PESQ SCORE**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9.703</td>
</tr>
</tbody>
</table>

**AVERAGE PESQ SCORE**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.212875</td>
</tr>
</tbody>
</table>

92
Spectral Subtraction:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>1.189</td>
<td>1.548</td>
<td>1.845</td>
<td>2.028</td>
<td>2.131</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.282</td>
<td>1.556</td>
<td>1.756</td>
<td>1.853</td>
<td>1.916</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.28</td>
<td>1.569</td>
<td>1.786</td>
<td>1.909</td>
<td>1.999</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.348</td>
<td>1.551</td>
<td>1.736</td>
<td>1.875</td>
<td>1.963</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.183</td>
<td>1.486</td>
<td>1.757</td>
<td>1.914</td>
<td>1.985</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.466</td>
<td>1.574</td>
<td>1.67</td>
<td>1.743</td>
<td>1.791</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.163</td>
<td>1.366</td>
<td>1.581</td>
<td>1.756</td>
<td>1.869</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>0.823</td>
<td>1.09</td>
<td>1.282</td>
<td>1.396</td>
<td>1.461</td>
</tr>
</tbody>
</table>

**TOTAL PESQ SCORE**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9.734</td>
<td>11.74</td>
<td>13.413</td>
<td>14.474</td>
<td>15.115</td>
</tr>
</tbody>
</table>

**AVERAGE PESQ SCORE**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.21675</td>
<td>1.4675</td>
<td>1.676625</td>
<td>1.80925</td>
<td>1.889375</td>
</tr>
</tbody>
</table>

Subspace:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>1.369</td>
<td>1.995</td>
<td>2.411</td>
<td>2.733</td>
<td>2.9</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.394</td>
<td>1.835</td>
<td>2.218</td>
<td>2.473</td>
<td>2.953</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.288</td>
<td>1.903</td>
<td>2.224</td>
<td>2.58</td>
<td>2.759</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.409</td>
<td>2.047</td>
<td>2.186</td>
<td>2.112</td>
<td>2.705</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.282</td>
<td>1.695</td>
<td>1.972</td>
<td>2.159</td>
<td>2.327</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.172</td>
<td>1.726</td>
<td>2.297</td>
<td>2.521</td>
<td>2.734</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.165</td>
<td>1.541</td>
<td>2.082</td>
<td>2.503</td>
<td>2.742</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>0.972</td>
<td>1.303</td>
<td>1.644</td>
<td>2.07</td>
<td>2.319</td>
</tr>
</tbody>
</table>

**TOTAL PESQ SCORE**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

**AVERAGE PESQ SCORE**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.256375</td>
<td>1.755625</td>
<td>2.12925</td>
<td>2.393875</td>
<td>2.679875</td>
</tr>
</tbody>
</table>
WienerANE:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>1.018</td>
<td>1.423</td>
<td>1.912</td>
<td>2.363</td>
<td>2.803</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.211</td>
<td>1.517</td>
<td>1.928</td>
<td>2.27</td>
<td>2.702</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.174</td>
<td>1.559</td>
<td>1.905</td>
<td>2.242</td>
<td>2.694</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.318</td>
<td>1.864</td>
<td>2.19</td>
<td>2.442</td>
<td>2.812</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.434</td>
<td>1.76</td>
<td>2.125</td>
<td>2.429</td>
<td>2.767</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.4</td>
<td>1.738</td>
<td>2.043</td>
<td>2.431</td>
<td>2.787</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.415</td>
<td>1.694</td>
<td>2.022</td>
<td>2.37</td>
<td>2.74</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>1.045</td>
<td>1.463</td>
<td>1.853</td>
<td>2.254</td>
<td>2.546</td>
</tr>
</tbody>
</table>

**TOTAL PESQ SCORE**


**AVERAGE PESQ SCORE**

|                  | 1.251875| 1.62725| 1.99725| 2.350125| 2.731375|
## B4: Jet Engine Noise

### SSANE:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>1.261</td>
<td>1.799</td>
<td>2.136</td>
<td>2.468</td>
<td>2.881</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.354</td>
<td>1.81</td>
<td>2.061</td>
<td>2.462</td>
<td>2.813</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.39</td>
<td>1.936</td>
<td>2.291</td>
<td>2.592</td>
<td>2.844</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.776</td>
<td>2.043</td>
<td>2.217</td>
<td>2.458</td>
<td>2.697</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.46</td>
<td>2.023</td>
<td>2.152</td>
<td>2.365</td>
<td>2.639</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.504</td>
<td>1.877</td>
<td>2.281</td>
<td>2.726</td>
<td>3.13</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.552</td>
<td>1.943</td>
<td>2.351</td>
<td>2.628</td>
<td>2.776</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>1.37</td>
<td>1.583</td>
<td>2.139</td>
<td>2.43</td>
<td>2.827</td>
</tr>
</tbody>
</table>

**TOTAL PESQ SCORE** 11.667 15.014 17.628 20.129 22.607

**AVERAGE PESQ SCORE** 1.458375 1.87675 2.2035 2.516125 2.825875

### Noisy:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>0.839</td>
<td>1.15</td>
<td>1.579</td>
<td>2.02</td>
<td>2.414</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.093</td>
<td>1.387</td>
<td>1.771</td>
<td>2.163</td>
<td>2.532</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>0.974</td>
<td>1.249</td>
<td>1.676</td>
<td>2.116</td>
<td>2.516</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.566</td>
<td>1.826</td>
<td>2.089</td>
<td>2.344</td>
<td>2.58</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.116</td>
<td>1.404</td>
<td>1.801</td>
<td>2.182</td>
<td>2.517</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.52</td>
<td>1.763</td>
<td>2.049</td>
<td>2.371</td>
<td>2.686</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.076</td>
<td>1.346</td>
<td>1.7</td>
<td>2.054</td>
<td>2.372</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>0.565</td>
<td>0.89</td>
<td>1.353</td>
<td>1.811</td>
<td>2.234</td>
</tr>
</tbody>
</table>

**TOTAL PESQ SCORE** 8.749 11.015 14.018 17.061 19.851

**AVERAGE PESQ SCORE** 1.093625 1.376875 1.75225 2.132625 2.481375
Spectral Subtraction:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>1.213</td>
<td>1.554</td>
<td>1.862</td>
<td>2.066</td>
<td>2.193</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.296</td>
<td>1.568</td>
<td>1.818</td>
<td>2.016</td>
<td>2.154</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.126</td>
<td>1.455</td>
<td>1.774</td>
<td>1.965</td>
<td>2.059</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.628</td>
<td>1.835</td>
<td>2.005</td>
<td>2.141</td>
<td>2.273</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.084</td>
<td>1.467</td>
<td>1.845</td>
<td>2.096</td>
<td>2.246</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.643</td>
<td>1.863</td>
<td>2.059</td>
<td>2.227</td>
<td>2.327</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.213</td>
<td>1.452</td>
<td>1.701</td>
<td>1.893</td>
<td>2.027</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>0.761</td>
<td>1.138</td>
<td>1.474</td>
<td>1.71</td>
<td>1.864</td>
</tr>
</tbody>
</table>

**TOTAL PESQ SCORE**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

**AVERAGE PESQ SCORE**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.2455</td>
<td>1.5415</td>
<td>1.81725</td>
<td>2.01425</td>
<td>2.142875</td>
</tr>
</tbody>
</table>

Subspace:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>1.093</td>
<td>1.693</td>
<td>2.145</td>
<td>2.41</td>
<td>2.756</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>0.764</td>
<td>1.523</td>
<td>2.192</td>
<td>2.581</td>
<td>2.823</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>0.973</td>
<td>1.7</td>
<td>1.964</td>
<td>2.258</td>
<td>2.554</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.332</td>
<td>1.643</td>
<td>1.992</td>
<td>2.302</td>
<td>2.726</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.235</td>
<td>1.666</td>
<td>2.098</td>
<td>2.279</td>
<td>2.487</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.18</td>
<td>1.639</td>
<td>1.951</td>
<td>2.213</td>
<td>2.6</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>0.804</td>
<td>1.518</td>
<td>1.837</td>
<td>2.231</td>
<td>2.444</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>0.638</td>
<td>1.194</td>
<td>1.781</td>
<td>1.965</td>
<td>2.176</td>
</tr>
</tbody>
</table>

**TOTAL PESQ SCORE**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8.019</td>
<td>12.576</td>
<td>15.96</td>
<td>18.239</td>
<td>20.566</td>
</tr>
</tbody>
</table>

**AVERAGE PESQ SCORE**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.002375</td>
<td>1.572</td>
<td>1.995</td>
<td>2.279875</td>
<td>2.57075</td>
</tr>
</tbody>
</table>
WienerANE:

<table>
<thead>
<tr>
<th>FILE</th>
<th>-5dB</th>
<th>0dB</th>
<th>5dB</th>
<th>10dB</th>
<th>15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigtips_16.wav</td>
<td>1.02</td>
<td>1.386</td>
<td>1.847</td>
<td>2.266</td>
<td>2.638</td>
</tr>
<tr>
<td>butterscotch_16.wav</td>
<td>1.214</td>
<td>1.476</td>
<td>1.826</td>
<td>2.258</td>
<td>2.641</td>
</tr>
<tr>
<td>Necessity_16.wav</td>
<td>1.07</td>
<td>1.43</td>
<td>1.883</td>
<td>2.281</td>
<td>2.757</td>
</tr>
<tr>
<td>peaches_16.wav</td>
<td>1.04</td>
<td>2.02</td>
<td>2.22</td>
<td>2.444</td>
<td>2.709</td>
</tr>
<tr>
<td>porcupines_16.wav</td>
<td>1.323</td>
<td>1.663</td>
<td>2.06</td>
<td>2.33</td>
<td>2.663</td>
</tr>
<tr>
<td>prison_16.wav</td>
<td>1.437</td>
<td>1.7</td>
<td>2.107</td>
<td>2.452</td>
<td>2.812</td>
</tr>
<tr>
<td>recognition_16.wav</td>
<td>1.245</td>
<td>1.499</td>
<td>1.89</td>
<td>2.281</td>
<td>2.637</td>
</tr>
<tr>
<td>scholars_16.wav</td>
<td>0.996</td>
<td>1.401</td>
<td>1.873</td>
<td>2.266</td>
<td>2.603</td>
</tr>
</tbody>
</table>

**TOTAL PESQ SCORE**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9.345</td>
<td>12.575</td>
<td>15.706</td>
<td>18.578</td>
<td>21.46</td>
</tr>
</tbody>
</table>

**AVERAGE PESQ SCORE**

|                  | 1.168125| 1.571875| 1.96325| 2.32225| 2.6825|
References


http://www.ieee.org/organizations/history_center/sloan/ASSR/assr_index.html

http://www.ind.rwth-aachen.de/research/cochlear/audiodemo.html


[33] Opticom website PESQ description: http://www.opticom.de/technology/pesq.html