12-31-2006

Use of principal component analysis with linear predictive features in developing a blind SNR estimation system

Matthew James Marbach
Rowan University

Use of Principal Component Analysis with Linear Predictive Features in Developing a Blind SNR Estimation System

by

Matthew James Marbach

A Thesis Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Department: Electrical and Computer Engineering
Major: Engineering (Electrical Engineering)

Approved: Members of the Committee

______________________________ ______________________________
In Charge of Major Work

______________________________ ______________________________
For the Major Department

____
For the College

Rowan University
Glassboro, New Jersey
2006

ABSTRACT

Matthew Marbach
USE OF PRINCIPAL COMPONENT ANALYSIS WITH LINEAR PREDICTIVE FEATURES IN DEVELOPING A BLIND SNR ESTIMATION SYSTEM
2005/06
Dr. Ravi Ramachandran
Master of Science in Electrical Engineering

Signal-to-noise ratio is an important concept in electrical communications, as it is a measurable ratio between a given transmitted signal and the inherent background noise of a transmission channel. Currently signal-to-noise ratio testing is primarily performed by using an intrusive method of comparing a corrupted signal to the original signal and giving it a score based on the comparison. However, this technique is inefficient and often impossible for practical use because it requires the original signal for comparison. A speech signal's characteristics and properties could be used to develop a non-intrusive method for determining SNR, or a method that does not require the presence of the original clean signal.

In this thesis, several extracted features were investigated to determine whether a neural network trained with data from corrupt speech signals could accurately estimate the SNR of a speech signal. A MultiLayer Perceptron (MLP) was trained on extracted features for each decibel level from 0dB to 30dB, in an attempt to create 'expert classifiers' for each SNR level. This type of architecture would then have 31 independent classifiers operating together to accurately estimate the signal-to-noise ratio of an unknown speech signal. Principal component analysis was also implemented to reduce dimensionality and increase class discrimination. The performance of several neural network classifier structures is examined, as well as analyzing the overall results to determine the optimal feature for estimating signal-to-noise ratio of an unknown speech signal. Decision-level fusion was the final procedure which combined the outputs of several classifier systems in an effort to reduce the estimation error.

ACKNOWLEDGEMENTS

I would like to thank my parents and my entire family for their love and support throughout my college years. Their hard work has provided me with opportunities I thought may never be possible. I certainly couldn't have accomplished this much over the last 5 years without the love and support from my girlfriend, Kate Evangelista.

Additionally, I would like to thank my peers in the Graduate Student Office for their determination over the past several years. In particular, I'd like to thank Russell Ondusko, III for his continued work on this project.

I would also like to thank the faculty of the Electrical & Computer Engineering Department at Rowan University, as they have provided me with an abundant amount of knowledge to further my education and skills in the engineering field. I would also like to thank Dr. Rusu for his assistance and as a member of my Thesis Defense Committee. I would also like to thank both of my Graduate Advisors, Dr. Ravi Ramachandran and Dr. Linda Head, for their guidance and assistance over the past few years.

This work was supported by the U.S. Air Force Research Laboratory, Rome NY, under contracts FA8750-05-C-0029 and F30602-03-C-0067.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 – INTRODUCTION

1.1 Measures of Speech Quality

Speech signals are subject to the addition of noise during multiple stages of their transmission. Quantization of the signal for digital transmission and noise in the channel of transmission are two notable sources for the degradation of an analog signal. Recent research has been done to develop techniques to calculate a confidence metric to accompany the decision of the speaker identity [1, 2]. Prior to recent implementations there were two primary methods of determining the quality of a speech signal. The first method is a quantitative assessment of the quality of the speech signal known as the Mean Opinion Score (MOS). This technique requires an individual to listen to the signal in question, and give it a vote on their opinion of the quality of the signal. Usually a discrete scale is used with labels such as excellent, good, fair, poor, and bad and assigned integers from 5 to 1. One downfall to this approach is that it relies on the opinion of a human, which may develop accuracy issues and force testing to become expensive and inconvenient.

The second method of determining the quality of a speech signal uses *a priori* knowledge and directly compares the original clean signal to the degraded output signal. While an accurate calculation, this intrusive approach is not always feasible as the original signal may not be known. The motivation of this thesis is aimed at developing a non-intrusive method of determining the quality of a speech signal without need for human interaction. An SNR estimation system will be devised using linear predictive features to train a multilayer perceptron classifier system.

1.2 Objectives of Thesis

The main objectives of this thesis are:

1. *To investigate seven speech signal features used primarily in speaker recognition system and their possible contribution to a signal-to-noise ratio estimation scheme.*
2. *To implement a neural network classifier approach to estimating signal-to-noise ratio.*
3. *To investigate the effect of classifier parameters in determining optimal neural network architecture.*
4. *To study the effects of the extracted speech features when the original speech signal has been corrupted with one of three different types of noise; Additive White Gaussian Noise, Pink Noise, and Telephone Channel Noise.*
5. *To create and train a multiple classifier system robust to all three types of noise.*
6. *To implement several fusion techniques in determining the optimal features to create a more robust estimation system.*

1.3 Expected Contributions

This thesis details the steps taken in constructing a robust, non-intrusive approach to estimating signal-to-noise ratio of a speech signal. The overall goal of the proposed approach is to achieve an average estimation error of less than 3dB through the use of the linear predictive speech features.

1.4 Focus and Organization

The focus of this thesis looks at the construction of a neural network classifier system trained on linear predictive features extracted from speech signals which have been corrupted with known levels of additive noise. The thesis is divided into the following chapters:

Chapter 1 is an introduction to the proposed approach in developing an SNR estimation system.

Chapter 2 is a literature review of previous methods used to estimate signal-to-noise ratio. In addition, this chapter also provides background information on the following: seven extracted features, two clustering algorithms, dimensionality reduction, and the multilayer perceptron.

Chapter 3 is an explanation of the proposed approach in developing an SNR estimation system. This includes step-by-step details for selecting both the training size and the specific construction of the neural network architecture. Each feature is examined and evaluated in great depth to determine its contribution to estimating SNR. A performance measure is also derived to accurately evaluate the comparison between various different structures of classifiers.

Chapter 4 presents the results from each step of the process, with figures and tables included to indicate their significance.

Chapter 5 summarizes and draws conclusions from the results.

Chapter 2 – Background

The complications of additive noise plague every communication system, whether analog or digital. Often, the noise can be extremely detrimental in transmitting a decipherable message over a communication channel. Prior to extracting information such as speaker identification, appropriate procedures should be taken to determine the quality of the speech signal which could diminish confidence in the overall identification system.

Noise estimation has been a thoroughly studied method of signal processing for many years, with varying levels of success among the many different methods. Several of the researched methods rely on power spectral estimates of the noisy signal in comparison with the probability of local signal presence to estimate the noise present within the signal [3]. In addition to researching spectral estimators, our approach to this problem requires investigating established features used for speaker recognition as well as a suitable classifier. The idea of using established speech features with a neural network to estimate signal-to-noise ratio of a signal has not been previously implemented.

This chapter provides a short literature review on some of the standard methods and derivations, as well as important terminology for understanding the proposed approach for estimating signal-to-noise ratio.

2.1 SNR Estimation Techniques

Over the years, there have been a number of proposed SNR estimation techniques proposed by the signal processing community. The majority of the traditional estimation methods rely on either spectrum or amplitude modulation analysis, with only a few basing the estimation using a cepstrum calculation [5].

Rangachari et al. proposed a noise estimation algorithm for use in a highly nonstationary environment. The method which was implemented by Rangachari included computing the noise estimate by averaging past spectral values of noisy speech using a time and frequency dependent smoothing factor. This smoothing factor was adjusted based on the probability of signal presence with a subband. Signal presence was computed using a comparison between the noisy speech and the local minimum power spectrums [3].

Cohen and Berdugo proposed a minima controlled recursive averaging (MCRA) approach to noise estimation by averaging past power spectral values and using an adjustable smoothing parameter to estimate the noise of a speech signal. Each frame of the utterance has its local energy calculated to determine the probability of speech signal presence within that frame. The presence of speech in the subbands is calculated by the ratio of the local energy of the noisy speech and its local minimum energy within a specified time window [4].

Hirsch et al. describes an approach that is based on a statistical analysis of the spectral energy envelope. Their method involved generating histograms of energy values for different frequency bands. They were able to extract information about the signal from these histograms, including a low energy and a high energy mode. The low energy mode related to the presence of speech pause frames and possible noise. The high energy mode corresponded to the presence of speech frames with the possibility of noise. An estimate of the signal-to-noise ratio was devised from these two histograms [6].

Krom developed an SNR estimation method based on the cepstrum feature. This technique filters the magnitude spectrum of a fundamental frequency adaptive cepstrum comb-liftering algorithm. The algorithm is designed to use the level difference in a certain frequency band between the original, unfiltered spectrum and the filtered spectrum as the SNR of the signal [5].

2.2 Features of Interest

As previously mentioned, the proposed method of estimating signal-to-noise ratio involves extracting established features from the speech signal. The methods in which the features are extracted, for any pattern recognition problem, is extremely significant to the overall performance of the classifier system [12]. The seven features which will be examined are: LP cepstrum (CEP), ACW cepstrum, PFL cepstrum, log area ratio (LAR), reflection coefficients (REFL), line spectral frequency (LSF), and the variance of the cepstrum (CEP_VAR) [7, 8, 9].

A linear predictive (LP) analysis is used for the feature extraction, and since a speech sample is a weighted linear combination of 'p' previous samples, it will result in set of weights a_k [7, 10]. The equation relating to this step is: $s(n)=\sum_{k=1}^{p} a_k s(n-k)+e(n)$ where s(n) is the speech signal, and e(n) is the error of the LP residual.

The calculated weights relate to the coefficients of a nonrecursive filter $A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k} = \prod_{k=1}^{p} (1 - f_k z^{-1})$ where f_k for $1 \leq k \leq p$ represent the zeros of A(z). Minimization of the mean squared error over N samples using the autocorrelation approach will result in determining the coefficients a_k using a system of linear equations. A(z) is guaranteed to be minimum phase and the magnitude spectrum of $\frac{1}{A(z)}$ describes the spectral envelope of the speech signal [7].

An autoregressive moving average (ARMA) analysis of speech can be used to derive a pole-zero transfer function U(z)/V(z). The coefficients of both U(z) and V(z) (u_k and v_k respectively) are computed starting with a minimum phase V(z) obtained through LP analysis. Hence, V(z) is equal to A(z). The impulse response of 1/V(z) is h(n), which is discretized to N samples. The associated error e(n) will be s(n)-h(n)*u(n), where u(n) is the finite impulse response of U(z). By once again minimizing the mean squared error, the coefficients of U(z) can be determined by a system of linear equations. The minimum phase characteristic is not inherent, but can be made so by reflecting the zeros of U(z) outside the unit circle to lie inside. For the pole-zero based cepstrum features (the ACW and PFL cepstrum), we start with V(z) = A(z) and introduce a simple modification of A(z) to get a minimum phase numerator U(z). This differentiates the ACW and PFL approaches from classical ARMA modeling, and has been shown to work for speaker identification. It is our intention to see how this works for SNR estimation [7].

2.2.1 LP Cepstrum (CEP)

The transfer function P(z)= U(z)/V(z) can be expressed as:

$$P(z) = \frac{U(z)}{V(z)} = \frac{\prod_{k=1}^{u}(1-u_k z^{-1})}{\prod_{k=1}^{v}(1-v_k z^{-1})}$$

If P(z) is minimum phase, the cepstrum can be computed from a recursion algorithm based on the polynomial coefficients, or by considering the polynomial roots u_k and v_k from:

$$c_p(n) = \frac{1}{n}\sum_{k=1}^{v} v_k{}^{n} - \frac{1}{n}\sum_{k=1}^{u} u_k{}^{n}$$

The LP cepstrum is computed using a recursive relation involving the predictor coefficients, given below: [11]

$$c_{LP}(n) = a_n + \sum_{i=1}^{n-1}(\frac{i}{n})c_{LP}(i)a_{n-1}$$

2.2.2 Postfilter Cepstrum (PFL)

The original concept of the postfilter cepstrum was to enhance noisy speech. The theory behind the postfilter cepstrum is that more noise can be tolerated in the spectral peaks than in the spectral valleys. The postfilter cepstrum is merely a weighted form of the LP cepstrum, and can be computed by either method shown below:

$$H_{pf}(z) = \frac{A(z/\beta)}{A(z/\alpha)} \quad \text{where} \quad 0 < \beta < \alpha \le 1$$

$$PFL_n = CEP_n(\alpha - \beta^n)$$
$$\alpha = 1.0$$
$$\beta = 0.9$$

When the spectrum of $H_{pf}(z)$ is taken, it can be seen that it emphasizes the spectral peaks. If A(z) is minimum phase, then $H_{pf}(z)$ is also minimum phase. The cepstrum of $H_{pf}(z)$ is used as the feature vector [11]. The postfilter cepstrum is said to be merely a weighting or liftering of the LP cepstrum, and is very robust to channel and noise effects.

2.2.3 Adaptive Component Weighting (ACW)

For the ACW cepstrum, a partial fraction expansion of 1/A(z) is completed resulting in:

$$\frac{1}{A(z)} = \sum_{k=1}^{p} \frac{\lim_{z\to f_k}[(1-f_k z^{-1})/A(z)]}{1-f_k z^{-1}} = \sum_{k=1}^{p} \frac{r_k}{1-f_k z^{-1}}$$

The residues r_k associated with the above equation will show considerable deviation when the speech is corrupted. With this knowledge, the ACW cepstrum can be computed by setting $r_k = 1$ for every k, leaving the transfer function in the form:

$$\frac{N(z)}{A(z)} = \sum_{k=1}^{p} \frac{1}{1-f_k z^{-1}} = \frac{1}{A(z)} \sum_{k=1}^{p} \prod_{i=1\neq k}^{p} (1-f_i z^{-1})$$

$$\frac{N(z)}{A(z)} = p \frac{1-\sum_{k=1}^{p-1} b_k z^{-k}}{1-\sum_{k=1}^{p-1} a_k z^{-k}}$$

The adaptive component weighting feature is calculated by computing the cepstrum of N(z)/A(z) by a recursion based on the polynomial coefficients [11]. The ACW cepstrum is robust to noise and hence, we expect it will not perform as well as features that are not as robust to noise. However, both the PFL and ACW features will be implemented in the proposed approach to gain an understanding as to how well noise robust features are able to discriminate between SNR levels.

2.2.4 Reflection Coefficients (REFL)

Using the Levinson Durbin Algorithm as our linear predictive analysis technique permits easy extraction of the reflection coefficients: $A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}$. It was shown that the polynomial above obtained by linear prediction analysis could be obtained from the recursion [11]:

$$\begin{aligned} &A^{(0)}(z) = 1 \\ &A^{(i)}(z) = A^{(i-1)}(z) - k_i z^{-1} A^{(i-1)}(z^{-1}) \\ &A(z) = A^{(p)}(z) \end{aligned}$$

where the above parameters k_i are normally called the PARCOR reflection coefficients.

2.2.5 Log Area Ratios (LAR)

From the above obtained reflection coefficients it is possible to derive the next feature known as the log area ratios. The equation for this derivation is

$$g_i = \log[\frac{A_{i+1}}{A_i}] = \log[\frac{1-k_i}{1+k_i}] \quad \text{where } 1 \le i \le p \text{ [8].}$$

2.2.6 Line Spectral Frequency (LSF)

If A(z) represents an all pole LP filter, the symmetric and anti-symmetric polynomials P(z) and Q(z) may be computed as:

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

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

p=order of linear prediction

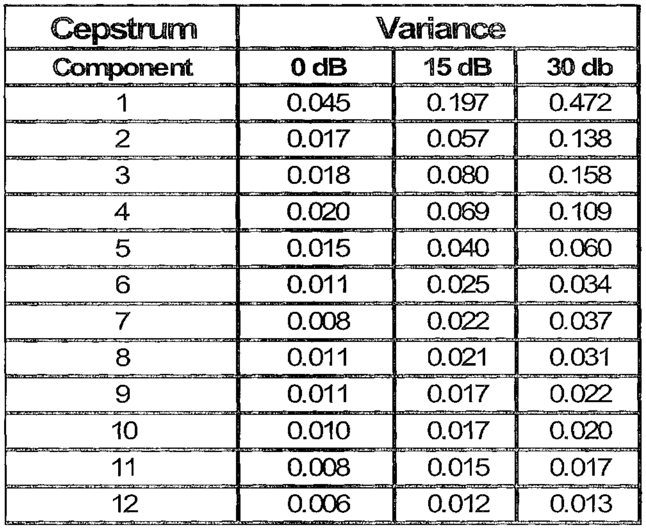
The LSF features which will be used are the angles of the roots of P(z) and Q(z) that lie between 0 and 180 degrees (not inclusive) [12].

2.2.7 Variance of Cepstrum (CEP_VAR)

The seventh and final feature is computed directly from the LP cepstrum, as it is the variance of the 12 dimensional LP cepstrum feature. The motivation behind using this feature lies with the initial calculation of the LP cepstrum. It has already been said that a speech sample is a weighted linear combination of 'p' previous samples with the resulting equation: $s(n) = \sum_{k=1}^{p} a_k s(n-k) + e(n)$

When the speech sample is corrupted with noise, the sample is harder to predict. The resulting a_k weights will then decrease towards zero which ultimately decreases the cepstrum coefficients. The decreased cepstrum coefficient values will not vary as much as those extracted from a clean speech sample which makes for an impressive, yet simple, SNR estimation feature. It has been previously established that additive noise reduces the variance of the cepstrum [13, 14, 15]. Table 1 below represents preliminary tests on our speech samples indicating this phenomenon was present for all three types of additive noise.

Table 1: Component Analysis for LP Cepstrum



Cepstrum	Variance		
Component	0 dB	15 dB	30 db
1	0.045	0.197	0.472
2	0.017	0.057	0.138
3	0.018	0.080	0.158
4	0.020	0.069	0.109
5	0.015	0.040	0.060
6	0.011	0.025	0.034
7	0.008	0.022	0.037
8	0.011	0.021	0.031
9	0.011	0.017	0.022
10	0.010	0.017	0.020
11	0.008	0.015	0.017
12	0.006	0.012	0.013

2.3 Clustering Algorithms

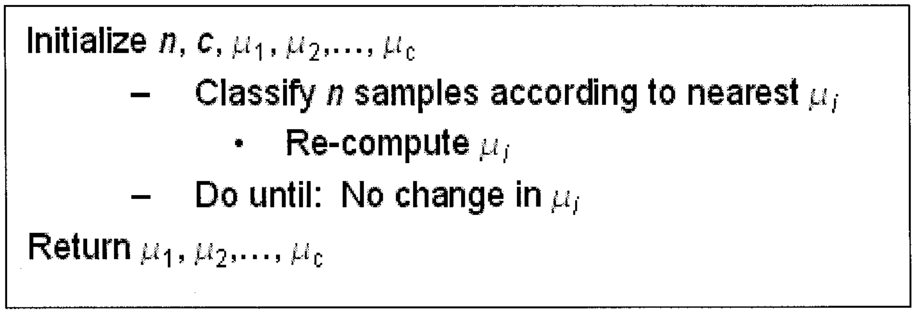
Speech coding often requires the use of clustering algorithms to accurately represent the dataset with a limited number of values, as using raw data may be computationally intensive. When extracting linear predictive features for each frame of an utterance, the resulting dataset can produce redundant information. To reduce the computation time in training a classifier with such information, a clustering algorithm is utilized to create a codebook (summary) which still conveys a great deal of information about the original data.

2.3.1 Linde-Buzo-Gray Algorithm

The Linde-Buzo-Gray Algorithm is a data clustering algorithm most often used for developing codebooks for speech coding. The process in which the codebooks are created is relatively straightforward and requires: a training set, the desired codebook size, and a distance measure. First, the original training set $T=\{x_1, x_2, x_3 \ldots x_M\}$ having M total number vectors is gathered, as well as the desired codebook size N. Additionally, a distance measure $d(x,y)$ is necessary to calculate the distance between each individual training point and the codebook. The initial codebook of size 1 is generated by taking the mean of the data. Once that is completed, the mean value is then split into two points by perturbing the original point by a small value. This continues until the distortion between the training set and the generated codebook is less than a certain threshold, or the desired codebook size has been reached [16].

2.3.2 K-means Clustering

Similar to the LBG Algorithm, the K-means algorithm is also used to cluster data to a specified 'k' number of centroids. The algorithm initially partitions the data into 'k' number of random sets, which then have their respective centroid calculated. Once the initial centroids for each set have been determined, each point in the data set is then reclassified based upon the squared Euclidean distance to the closest centroid. This process continues until the centroids remain constant [17]. Figure 1 below represents pseudo-code for clustering a dataset using the k-means algorithm.



```
Initialize $n$, $c$, $\mu_1, \mu_2, \ldots, \mu_c$
        – Classify $n$ samples according to nearest $\mu_i$
                • Re-compute $\mu_i$
        – Do until: No change in $\mu_i$
Return $\mu_1, \mu_2, \ldots, \mu_c$
```

Figure 1: Pseudo-code for K-means Algorithm [17].

The LBG and k-means algorithm are often used interchangeably, as they are both considered to be popular compression techniques. It was necessary to use both algorithms for this approach due to the structure of the training data inputted into the multilayer perceptron classifier.

2.4 Multilayer Perceptron (MLP)

The Multi-Layer Perceptron (MLP) is a network that uses several layers of interconnected neurons, or perceptrons.. The basic concept of a single perceptron is that it computes a single output from multiple inputs by forming a linear combination according to its input weights, and then putting the output through a nonlinear activation function [18]. A single-layer perceptron is generally only implemented when the boundaries between classes can be drawn linearly, which is not the case for the speech features used in this approach. The training datasets used for the classifier system will consist of the extracted features from each corrupted training signal. Instead, a multilayer perceptron structure will be used, which consists of an input layer, an output layer, and also a hidden layer, or a number of layers [18, 17]. Figure 2 below illustrates the basic architecture of a multilayer perceptron, including the three layers: input layer, hidden layer, and output layer. The weights connecting each node will be computed using the back-propagation algorithm.

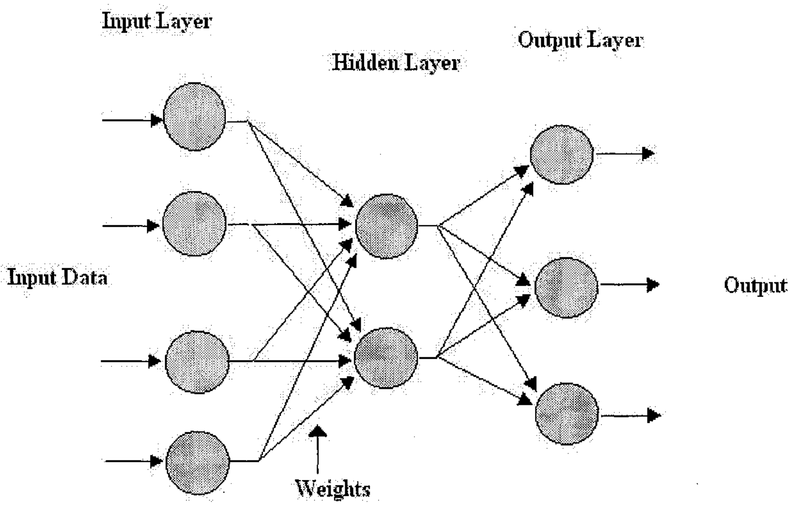


Figure 2: Multilayer Perceptron Network

One of the major goals in designing a neural network classifier is to create one that is not too simplistic that it can not explain the differences between the classes, and also one that is not too complex that it overfits the data. Choosing the correct number of hidden layer nodes when creating an MLP is impossible without conducting empirical trials. All of the scenarios in this thesis were run several times over, with a varying number of hidden layer nodes. If too many hidden layer nodes are selected, overfitting of the data is an all too common occurrence, resulting in a poorly designed network [17]. Figure 3 below illustrates a classifier overfitting the dataset. The dashed line is the ideal class boundary while the red line depicts the actual class boundary which the network has drawn. This phenomenon generally yields poor classification due to the network overfitting the training dataset.

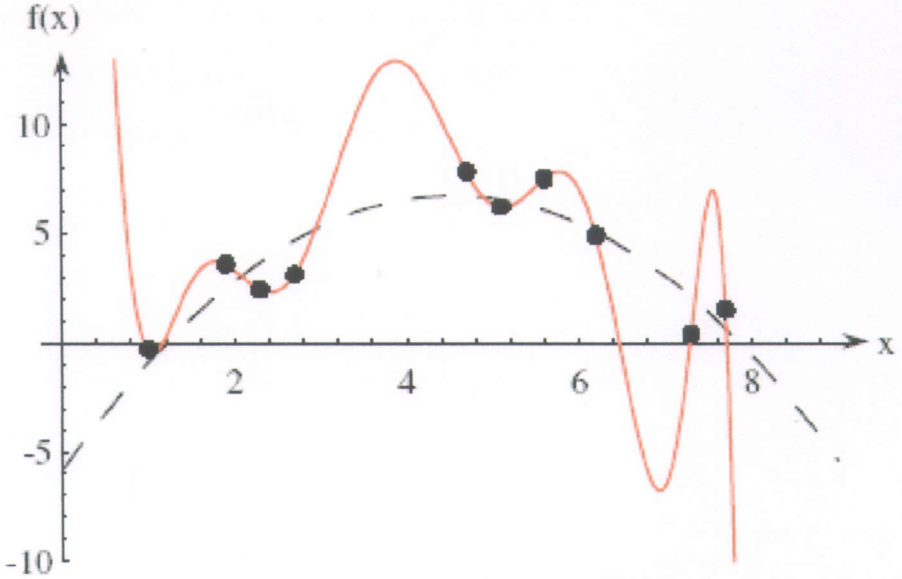


Figure 3: Overfitting of Training Data [17]

2.5 Neural Network Implementation with Speaker Recognition

With many of these seven speech features occurring primarily in speaker recognition systems, classification with a multilayer perceptron isn't an unheard of implementation for these features.

In [19], Farrell et al. use extracted cepstrum coefficients for classification with a neural network for a closed-set speaker identification problem. Rather than creating the neural network with just one type of architecture, the number of hidden layer nodes was altered among several trials in hopes of increasing classification performance. The MLP created with 16 hidden nodes outperformed those created with 32 and 64, as the more complicated systems were likely overfitting the training data. It can be seen in Table 2 that the cepstral coefficients are a suitable feature for use in a speaker identification system, with classification performance reaching over 90% [19].

Table 2: Closed-Set Speakers Identification Performance using MLP [19]

Hidden Nodes	5 Speakers	10 Speakers	20 Speakers
16	96%	90%	90%
32	96%	90%	82%
64	88%	94%	85%

Table 2 depicts the variation in performance which may occur when a classifier is overtrained for a particular problem. An increase in the number of hidden layer nodes does not always guarantee an increase in classification performance.

2.6 Dimensionality Reduction

Rather than using all of the dimensions of the features with the multilayer perceptron, a common component analysis method has been examined to best represent the data in the fewest dimensions necessary. The technique known as Principal Component Analysis (PCA), or the Karhunen-Loéve Transform, is quite common in image processing and other areas of signal processing where dimensionality can play a huge role in the discrimination of the data. The complexity of a classifier that discriminates among different classes increases as the dimensionality of the data increases. The *curse of dimensionality* states that high-dimensional problems have the potential to be much more complicated than low-dimensional ones, and those complications are harder to discern. PCA has been implemented to reduce the dimensionality of the training data, which may ultimately reduce the complexity of the classifier necessary to discern between classes. [18, 20].

2.6.1 Principal Component Analysis (PCA)

The method behind the PCA transformation for our 12-dimensional features is as follows: the m principal chosen axes, where $1 \leq m \leq 12$, are orthonormal axes projected into a new feature space where the retained variance is greatest [18, 20].

In reducing the dimensionality of a given data set, a few calculations must be completed before transforming the data to a new feature space. First, the mean vector and global covariance matrix must be computed from the entire data set. The eigenvectors of the global covariance matrix are then calculated and sorted in descending order by their respective eigenvalues. The transformation can then be computed from $Y = A * X$, where A contains the ordered eigenvectors of the covariance matrix and X is the original matrix. The dimensionality of the transformed vector Y can be reduced by eliminating some of the eigenvectors of A which have small eigenvalues. This would result in a decreased number of dimensions in a new feature space [18, 20, 21].

Principal component analysis also assumes that the information of the dataset lies within the variance of the data. This dimensionality reduction technique does not take class information into account, so there is no guarantee that the separation between classes in the new feature space will be better than the original one [17].

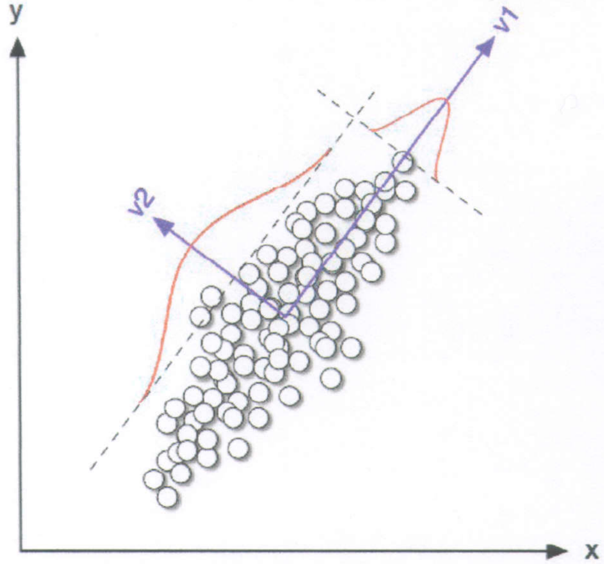


Figure 4: Principal Component Analysis [22].

Figure 4 illustrates the two principal component axes for a sample dataset. The new feature space is simply a rotated version of the original one, in which the principal component axis is chosen such that the maximum variability within the data can be seen. For this example, a plot of the new feature space would result in V_1 as the x-axis.

2.6.2 PCA Implementation with Speech Processing

When compared to image processing, the use of principal component analysis is somewhat rare and scattered among different fields within speech processing. Hu et al. [23] and Wang et al. [21] both applied PCA to vowel recognition with limited success, while Ding et al. investigated the use of PCA with the linear predictive cepstrum for a text-independent speaker recognition system using a VQ classifier [21]. Their results for those experiments can be seen in Table 3.

Table 3: Classification Performance of Speaker Recognition system with reduced dimensionality[21]

Dimensionality	6	12	24
LPC	87.31%	93.67%	96.54%
MFPC	98.63%	98.63%	98.85%
MFCC	90.54%	95.62%	98.63%

Aside from the LP cepstrum, two other features were also examined by Ding et al. [24] and their speaker recognition system: Mel-Frequency Principal Coefficient (MFPC), and Mel-Frequency Cepstral Coefficient (MFCC). While MFPC showed no loss in performance with a reduction in dimension, a sacrifice in classification was seen when reducing the dimensionality of the other two features [21]. Despite the linear predictive cepstrum feature not providing particularly good results for Ding et al. and their implemented PCA algorithm, it may still prove to be a useful feature in SNR estimation as the components of the feature will vary in a different manner than it would for a speaker recognition scenario. A system trained specifically on certain speakers may not have the same variance in the training data as a system trained towards different SNR levels.

CHAPTER 3 – APPROACH

This chapter has three major sections. Section 3.1 gives details of the process of extracting the features from the speech database. Section 3.2 describes the calculations to measure the performance of the classifier system. Section 3.3 explains the various neural network architectures which were created, as well as the construction of their respective training data.

3.1 Feature Extraction Process

An important asset to this project was the access and use of the TIMIT database which provided both the train and test datasets for this project [25]. The database is a collection of transcribed speech of American English speakers of different sexes and dialects. The portion of the corpus which was used is the New England dialect, with 38 speakers uttering 10 different sentences. Training and testing datasets were divided by using the first set of 5 sentences from the 38 speakers for training, and the second set of 5 sentences for testing the classifier.

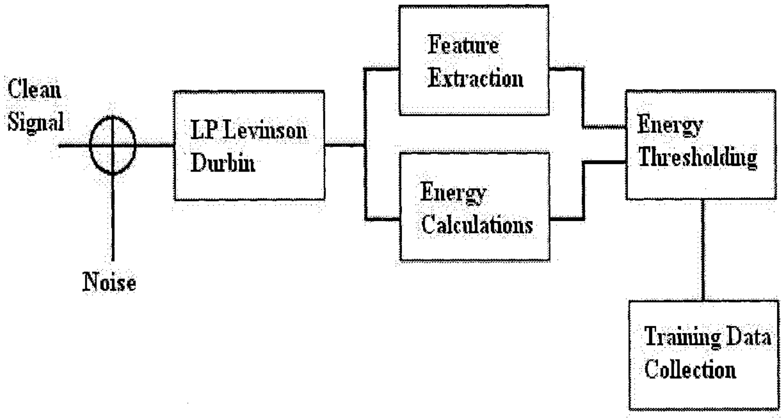


Figure 5: Block Diagram of Training Data Collection

After splitting the database into training and testing sets, the feature extraction is performed. First, a known level of noise is added to the training signal. The next step is the energy thresholding process where silent portions of the corrupted speech signal are removed. This step is imperative for the feature extraction process since the established speech features are for speech segments only. Features extracted on silent portions could corrupt the data and increase the difficulty for the classifier to discern between classes. The corrupted speech signal is partitioned into frames from which respective energy values are computed for each frame. The Levinson-Durbin algorithm is used to calculate the linear predictive (LP) 'a' vectors for each frame of the corrupted speech signal by solving the Toeplitz autocorrelation matrix. These 'a' vectors are then used to derive each speech feature. The 'a' vectors which correspond to the 'silent' portions of the signal are removed. From these extracted 'a' vectors, each of the features are then calculated. The test features are extracted in the same fashion but from the second portion of the database. Figure 5 illustrates a block diagram of the steps taken to extract the training data necessary for the classifier.

3.2 Training Datasets

The construction of a neural network classifier relies on the training dataset more so than the network structure parameters. For a simple 2-class problem, a multilayer perceptron classifier is trained with data from both classes with output labels associated to each class. The network's task is to learn the training data associated with each class and to store the "knowledge" in the interconnected weights of the classifier.

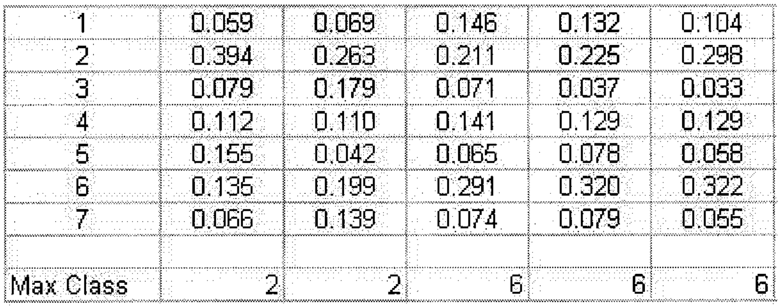
For this approach, the classifiers are trained with data corresponding to a specific SNR level. In the case of the single classifier system, one multilayer perceptron is trained with data corresponding to all SNR levels of interest, whether it is 5dB or 1dB spacing. The multiple classifier system has a collection of multilayer perceptrons, each trained with data corresponding to a specific SNR level and also data from all over SNR levels. Each MLP of the multiple classifier system is a 2-class classifier, trained with both 'data' and 'anti-data'. The 'data', or class 1, refers to the features extracted from speech signals corresponding to a specific SNR level. For example, an MLP trained for 16dB would be trained with features extracted from 16dB signals for its class 1. The second class would then consist of extracted data from all other SNR levels (0dB - 15dB & 17dB - 30dB). This second class will be known as the 'anti-data'.

The main motivational reason for using the LBG and k-means algorithms was to compress the data and anti-data to equal size matrices before inputting the data into a classifier for training. LBG is used to compress the data to a constant codebook size of 128. To eliminate any bias that may be shown by the classifier, the anti-data matrix will also be of size 128. The anti-data consists of compressed data from all other SNR levels, which is where k-means is used. Data from each of the remaining SNR levels must also be extracted and compressed to a codebook of size 4, so that the concatenation of these remaining SNR levels results in a matrix of size 128.

3.3 Network Output

After training the neural network structure with the appropriate datasets, the network is then ready to classify test signals. The test data is extracted from the test signals just as was done for the original training data, but there will be no compression of data for the test signals. After the features have been extracted, they are then sent through the classifier system for classification purposes.

Since each vector of the test data corresponds to a frame of speech, the network evaluates the test signal on a vector-by-vector basis to determine which class it most closely resembles. The test vector is then assigned an appropriate support value to each class by the classifier system. This value represents the support given to each class by the test vector, with the highest support value representing the class in which the test vector most closely resembles. The maximum support value showing which class the network classified the vector as is the only output which is of interest. This can also be thought of as a '1' for the class which received the most support, and a '0' for all other classes. This method is computed for each vector of the test signal, until all vectors of the test signal have been classified. Once all of the vectors of the test signal have been through the classifier system, the number of '1's for each class are totalled. This output depicts how much support each class received in comparison to the entire signal.



1	0.059	0.069	0.146	0.132	0.104
2	0.394	0.263	0.211	0.225	0.298
3	0.079	0.179	0.071	0.037	0.033
4	0.112	0.110	0.141	0.129	0.129
5	0.155	0.042	0.065	0.078	0.058
6	0.135	0.199	0.291	0.320	0.322
7	0.066	0.139	0.074	0.079	0.055
Max Class	2	2	6	6	6

Figure 6: Sample support values given by Single Classifier System on vector-by-vector basis

Figure 6 above illustrates a sample test signal's support values for the first 5 vectors of the signal, and the associated maximum class support for each vector. This particular example was depicted for the single classifier system, with 7 classes corresponding to SNR levels from 0dB to 30dB in 5dB increments.

3.4 Performance Measures

3.4.1 Hard & Soft Decision Scoring

From the network output and the associated support values given to each test vector, a performance measure is derived to determine the accuracy of the classifier in estimating the SNR of the test signal. The class which receives maximum support, or the most number of '1's, for that specific test data sample is determined as the hard decision estimation for that specific SNR test sentence.

To increase performance a soft decision approach was devised to estimate the true SNR level of the test signal. The devised method also used the network support given to each class from the classifier. The three SNR levels that received the highest support for an individual test signal were noted. A probability measure is calculated by dividing each SNR level's support by the total number of test vectors. The top three probabilities correspond to the top three classes which most closely resembled the test signal. The top three probabilities are then divided by the sum of the three. An estimated SNR value is then calculated by multiplying each SNR level's new probability with its respective SNR value. This calculation can be seen below;

$$\mathrm{SNR} = \sum_{j=1}^{N} \mathrm{Prob}(j)\mathrm{SNR}(j)$$

where N=3, and the top 3 probabilities are multiplied by their corresponding SNR level (0-30). This will give an approximate value for any given test signal. Once that is computed, and since each test signal has a known level of additive noise, the absolute error can be calculated by subtracting the estimated decibel level from the true SNR level.

While the above method would work for a single MLP classifier, the decision making for a multiple classifier system is quite different. For a multiple classifier system, the extracted test data must be passed through each of the trained MLPs to determine support of the test signal in comparison to each SNR level. As previously described, a multilayer perceptron will output a '1' or '0' for each frame of the test data to show whether it more closely resembles the data or anti-data for that particular SNR. After any given test signal has been passed through all of the MLPs, the number of '1's outputted by each MLP is stored and compared to make the hard and soft decisions. The MLP which outputted the most '1's for a particular test signal is chosen as the decibel level for the hard decision. The outputs from all MLPs are also ranked in order of support, to determine the three highest decibel levels which showed the greatest support to that particular test signal. From this point, the soft decision method is carried out by using the total number of '1's for each of the three winners, divided by the total number of vectors in that signal. This will give an accurate probability as to the classifier system's estimate of the SNR of the test signal, and will be used for the soft decision estimation as previously described for the single classifier system.

During the training process of a multilayer perceptron, there may be certain scenarios or conflicting training data which may prevent a classifier from training properly. This will result in a classifier outputting incorrect values when presented with appropriate test data. If this occurs for a single MLP within a multiple classifier system, often the classifier's output may be insignificant in estimating the SNR of a test signal. The outlying MLP may also introduce significant anomalies in which the estimation of an speech signal may result in an absolute error of 20dB. One particular method of deemphasizing a single classifier's output is to take an average of the classifier outputs with its neighbour's estimation when determining the decibel level of the signal. Table 4 illustrates the implementation of such a scoring scheme on a sample test signal and the associated classifier outputs.

The second row of Table 4 represents the associated output for each classifier trained on a specific SNR level. The classifier outputs result in a hard decision of 3dB, but the soft decision will be skewed in an unfavourable manner. The MLP corresponding to 27dB outputted 95 frames in which the test signal matched feature data for 27dB. All of the classifiers trained on SNR levels surrounding 27dB resulted in just several matching frames, leading to assume that the 27dB was not trained properly. A method to suppress this outlying score is to take the average of the respective SNR level score with its 2 neighboring MLP scores. This method results in an averaged score representative of the surrounding classifiers, as well as the classifier of interest. The new score for 27dB is now de-emphasized in such a manner that it will not affect the soft decision scoring method. Instead 3dB, 4dB, and 2dB are the three SNR levels used to calculate the soft decision estimate.

Table 4: Sample calculation for Averaging MLP outputs to reduce outlying classifiers

SNR - MLP	0 dB	1 dB	2 dB	3 dB	4 dB	5 dB	…	25 dB	26 dB	27 dB	28 dB	29 dB	30 dB
MLP Score	65	78	89	100	94	84	…	5	2	95	4	6	3
New Score	72	77	89	94	93	78	…	0	34	34	35	4.3	4.5

Due to the training algorithm of the neural network and the random initialization of the interconnected weights, the neural network will never draw the exact same decision boundaries between classes twice. To gain a generalization of the performance of the classifier system, each of the implemented trials were performed a total of five times. Confidence intervals can then be calculated [26]:

$$(X - 1.96\frac{\sigma}{\sqrt{n}} \leq \mu \leq X + 1.96\frac{\sigma}{\sqrt{n}})$$

where μ is the real average performance, X is the calculated mean of our performance tests, n is the number of samples, σ is the standard deviation of the data, and 1.96 is the area under the Gaussian curve encompassing 95% of the curve.

3.4.2 Overall Average Absolute Error (OAAE)

The hard and soft decision scoring schemes generate estimates from the classifier system on each specific utterance. Since the true SNR of the test signal is known, the estimation error is easily calculated by computing the absolute value of the difference between predicted SNR and the true SNR. This value is known as the Absolute Error (AE). This absolute error value is computed for test signals at each SNR level, and then divided by the total number of test signals to compute an Average Absolute Error (AAE) corresponding to each SNR level. The Average Absolute Error (AAE) illustrates the error one might expect if a test signal with a known level of additive noise is passed through the classifier system. This value is different for each SNR level, and represents the classifier system's ability to classify test signals at a specific SNR level. This performance measure can be taken one step further by computing the overall average of the AAE values, allowing a classifier system's performance to be summarized with a single value, the Overall Average Absolute Error (OAAE).

3.5 Neural Network Architecture

The MLP networks are constructed in a supervised learning environment. This means that a set of training data with desired outputs will be fed into the network, and the network must learn to model the data appropriately. This section addresses the various architectures which were implemented in training and testing the extracted features.

3.5.1 Single Classifier System

3.5.1.1 5 dB Class Increments

For the first implementation of the neural network classifier system, a single multilayer perceptron was created with seven different classes, corresponding to SNR levels from 0dB to 30dB in 5dB increments. The use of the LBG algorithm created training data sizes ranging from 32 to 256 vectors, in an attempt to determine the optimal data conditions for a single MLP classifier with seven classes.

3.5.1.2 1 dB Class Increments

The same procedure from the 5dB experiments was again performed with data extracted from the training data in 1dB increments. This created a single classifier with training data from 31 different classes. While only the cepstrum feature was used for these tests, the main motivation was to determine the appropriate size of the training data vector. Training data sizes varied from 16 to 512 vectors in factors of 2.

3.5.2 Multiple Classifier System

Instead of relying on a single classifier to estimate the SNR of a speech signal, an approach has been devised that would utilize a group of 'expert' classifiers trained on specific SNR levels. Keeping with the 1dB increment spacing, this method would create 31 individual classifiers which would each be trained on data extracted from a particular SNR level.

The MLP is able to better discern among classes when it is trained with both data and anti-data. For example, if one were to train an MLP with the intention of recognizing test features belonging to an SNR of 16dB, then the MLP should be trained with data belonging to 16dB as well as data belonging to other classes (anti-data). When creating the MLP, the training data is fed into the classifier along with its respective class information. The MLPs created for this particular architecture would each be 2 class classifiers and ultimately discerning whether the test data from each frame belongs to class 1, or class 2. Class 1 represents the decibel level of interest while class 2 represents the anti-data, which consists of data from all other decibel levels. A diagram of this approach can be seen in Figure 7. It depicts the input of features to the classifiers and that the output of each classifier is utilized in determining the overall SNR estimation [19]

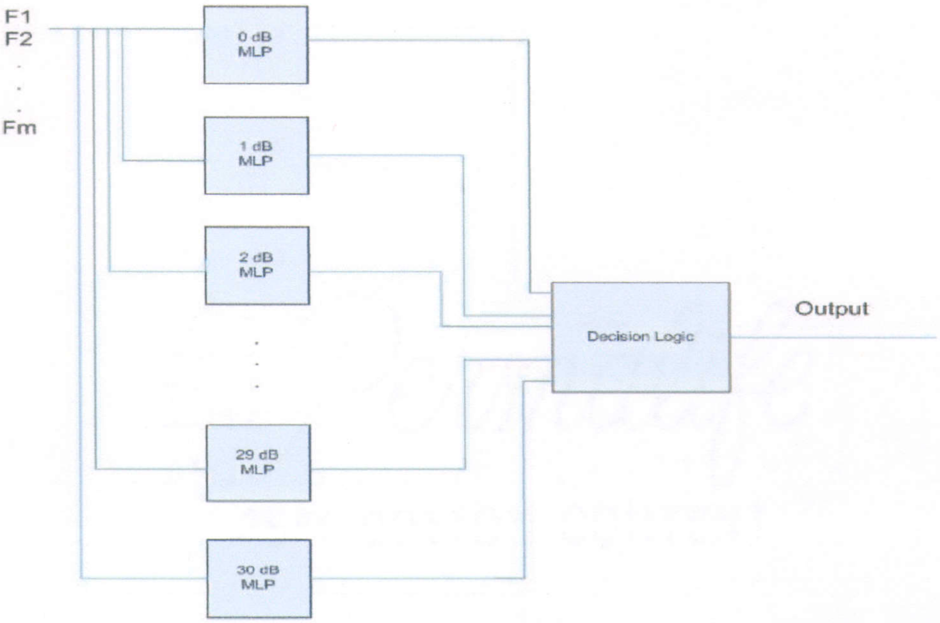


Figure 7: Multiple Multilayer Perceptron Architecture

Initial experiments were performed using a total of 31 MLPs, from 0dB to 30dB. Test signals were corrupted with known levels of noise, from 0dB to 30dB.

Preliminary results showed that the OAAE results for the extreme values (0dB and 30dB) were at least 1dB higher than their respective neighboring SNR levels. Part of that problem originated from the soft decision method itself. If the true SNR of a test signal is 30dB and the classifier system outputs 30dB as the hard decision estimate, the soft decision estimation will automatically be flawed. This occurs because the soft decision method chooses the top three classes which showed the most resemblance to the test signal in computing its estimate. For the same example, even if the second and third classes which showed the most resemblance were 29dB and 28dB respectively, the final soft decision estimation would be further away from 30dB than the user would like to tolerate. Since the soft decision calculates the top three classes in its estimate, the inclusion of 29dB and 28dB would skew the soft decision estimate further away from the true SNR (30dB). The motivation to reduce this inherent error would be to train a classifier for -1dB and 31dB in the multiple classifier system. The test data will remain in the range from 0 to 30dB, but the training data now consists of features extracted from -1dB to 31dB.

3.5.2.1 Train / Test with Same Type of Additive Noise

For this particular application, we will be looking into three types of additive noise: white Gaussian, pink, and telephone channel noise (CPV) [27]. A robust system capable of handling any type of noise would be ideal, but these three types of noise will be our main focus for training data. Each type of additive noise has a distinct noise spectrum which will ultimately affect the speech features in their own unique fashion. White Gaussian noise is a random signal with a flat power spectral density. Pink noise is a signal with a frequency spectrum that the power spectral density is the reciprocal of the frequency. Telephone channel noise has a bandpass type of noise spectrum, with different cutoff frequencies and roll off for different types of telephone channel noise. Two types of telephone channel noise will be utilized in this approach, which are the continental poor-voice (CPV) and the continental mid-voice (CMV) [27]. Each type of noise will affect the features in a different manner, with the hope that the multiple classifier system is able to discern among different classes for each feature.

In order to determine the classification capability of the features with additive noise, several different combinations of training and testing data needed to be exhausted. A set of classifiers were trained and tested on just white Gaussian noise for all seven features. Following the multiple classifier system trained with AWGN tests, experiments were also conducted with training classifier systems based solely on pink or telephone channel noise and testing with the same type of noise.

3.5.2.2 Train / Test Mismatch

For our proposed approach, it would be nearly impossible to train a classifier on every type of noise which can occur across a transmission channel. Instead, a few experiments were conducted to determine the Overall Average Absolute Error (OAAE) when a classification system is trained on one specific type of noise, and tested with data corrupted by a different type of noise. This methodology gave a generalization on the risk involved with training a system purely on one type of noise and having test data of another nature.

This particular robustness test was approached in three different scenarios:

- Train classifiers on CEP feature corrupted with white Gaussian noise; test with CEP corrupted with pink noise.
- Train classifiers on LSF feature corrupted with white Gaussian noise; test with LSF corrupted with telephone channel noise (CPV).
- Train classifiers on PFL feature corrupted with pink noise; test with PFL corrupted with telephone channel noise (CPV).

3.5.2.3 Train Noise Robust System

3.5.2.3.1 Analysis without Principal Component Analysis

After the results for the previous section had been obtained, a classification system was created and trained on three types of noise: white Gaussian, pink, and telephone channel noise. Due to the dimensionality of the data and the number of training vectors, the classifiers required to differentiate between classes become computationally complex. The overall capability of the system in estimating the SNR of a corrupted speech signal decreases due to that complexity. Following the training of the classifiers, the test data consisted of data also corrupted by the same three types of noise.

3.5.2.3.2 Analysis with Principal Component Analysis

As an additional method to aide in the training process, Principal Component Analysis (PCA) was implemented to reduce the amount of training data necessary to successfully solve the problem. With several features, PCA increased the overall performance of the system when compared to the previously run 12-dimensional tests. This is due in part to the transformation of the data via PCA, which ultimately decreased the number of dimensions necessary to discern between classes.

3.5.2.4 Train Noise Robust System, Test Mismatch

Once a robust classifier system had been trained on three different types of noise, another experiment was executed which involved testing with a type of noise that has not been seen by the system. This type of experiment will show how risky it could be to train a system on several types of noise, and encounter a different type of noise with the test data. The system specifications were chosen by selecting the PCA scenarios which was seen to be the most robust during the previous trials, as well as the original 12-dimensional scenario. These classifiers were then trained again with three types of noise, and tested with data corrupted by a different type of telephone channel noise known as continental mid-voice (CMV).

3.5.3 Decision-Level Fusion

Once all of the previously mentioned procedures have been exhausted, the final step consists of fusing together some of the best classifier opinions. If certain features regularly outperform others at particular SNR levels, then a fusion technique could be implemented to improve the estimate. Figure 8 illustrates the decision-level fusion of two separate multiple classifier systems, trained on cepstrum and variance of cepstrum.

There are a wide variety of fusion approaches, but the particular methods implemented for these features include the mean, median, and trimmed mean of the estimates. Once all possible feature combinations have been exhausted for all seven features, performance values are analyzed to determine optimal estimation system conditions.

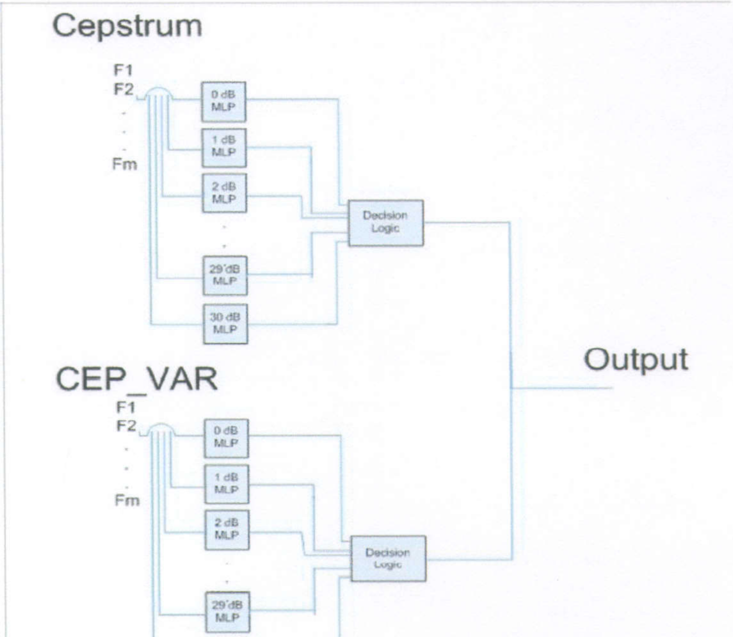


Figure 8: Decision-Level Fusion shown with two separate Multiple Classifier Systems trained on Cepstrum and Variance of Cepstrum Data.

CHAPTER 4 – IMPLEMENTATION AND RESULTS

This chapter has three sections. Section 4.1 describes the test results and procedures used with just a single MLP classifier. Section 4.2 explains the results for the series of experiments which were conducted with a multiple classifier system. Section 4.3 describes the overall results with decision-level fusion, which takes the results of several classification systems in making an overall estimation.

4.1 Single Classifier System

4.1.1 5dB Class Increment

Before the introduction of the OAAE measure, the performance standard which was used was simply the percentage of test signals correctly classified divided by the total number of test signals. For these particular tests, a single classifier was trained on data from seven different classes, which consisted of codebooks of 128 vectors extracted from SNR levels 0 to 30dB in 5dB increments. The LBG algorithm was utilized in this training phase, as several data manipulations were applied to determine an optimal training size. Training vectors of size 32, 64, 128, and 256 were used for the experiments. Data sizes larger than 256 vectors resulted in a performance decrease of 50%, and will not be discussed.

In addition to determining an optimal training size, the number of hidden layer nodes was also altered. The performance of a classifier can rely heavily on the construction of the network, which is one of the main factors for altering the free parameter. If the network is created with too many hidden layer nodes, often the network will end up overtraining the data which would result in a decrease in performance.

These 5dB spacing tests were conducted for each of the seven features, but only the LP cepstrum results are shown in the following figures. This is due in part to the poor performance of a single classifier with the remaining features, with the only useful information corresponding to the training data size. The number of hidden layer nodes showed to have an minimal effect on the performance of a single classifier system, but will become crucial when constructing the multiple classifier system.

Figure 9 and Figure 10 both illustrate that a small number of training vectors was not able to provide the classifier system with enough class information to accurately discern between classes. Figure 11 and Figure 12 correspond to a single classifier system trained with 128 and 256 vectors for each class, respectively. The performance for both 128 and 256 vectors is higher than 32 and 64 vectors, with the best performance occurring at 0dB. While a classifier system trained with 256 vectors from each of the seven classes resulted in poor performance for all SNR levels, a more accurate performance measure was created to gather more information regarding the estimation error for a classifier system at a certain SNR level. This method is known as the Overall Average Absolute Error (OAAE), and was used for the remaining tests.

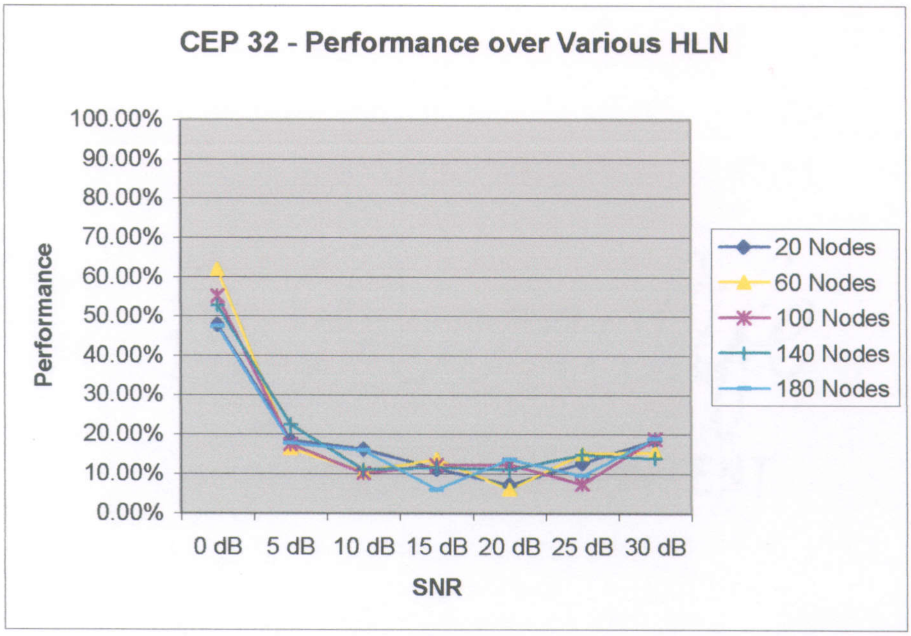


Figure 9: LP Cepstrum: 5dB Performance over Varying HLN, Training Size = 32 vectors.

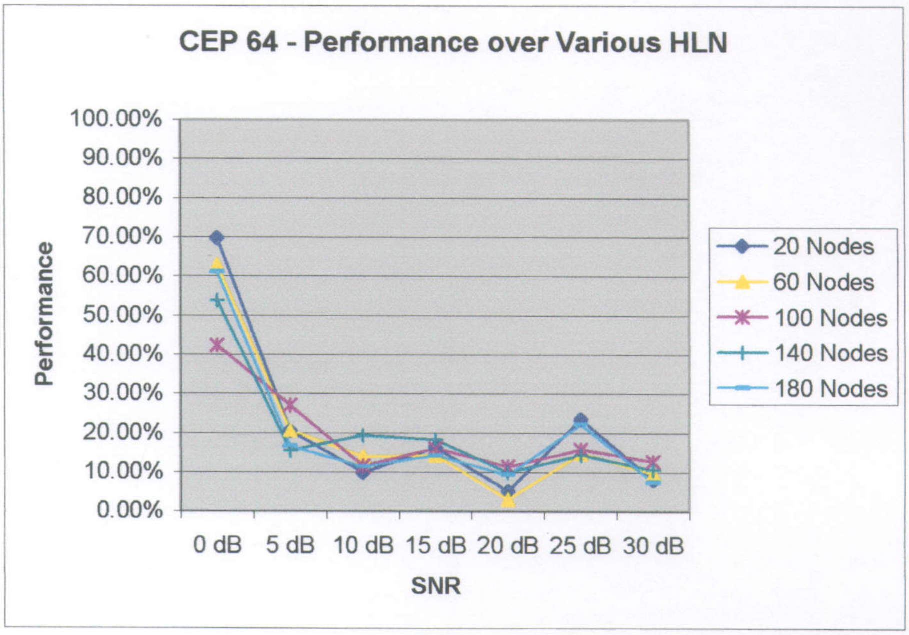


Figure 10: LP Cepstrum: 5dB Performance over Varying HLN, Training Size = 64 vectors.

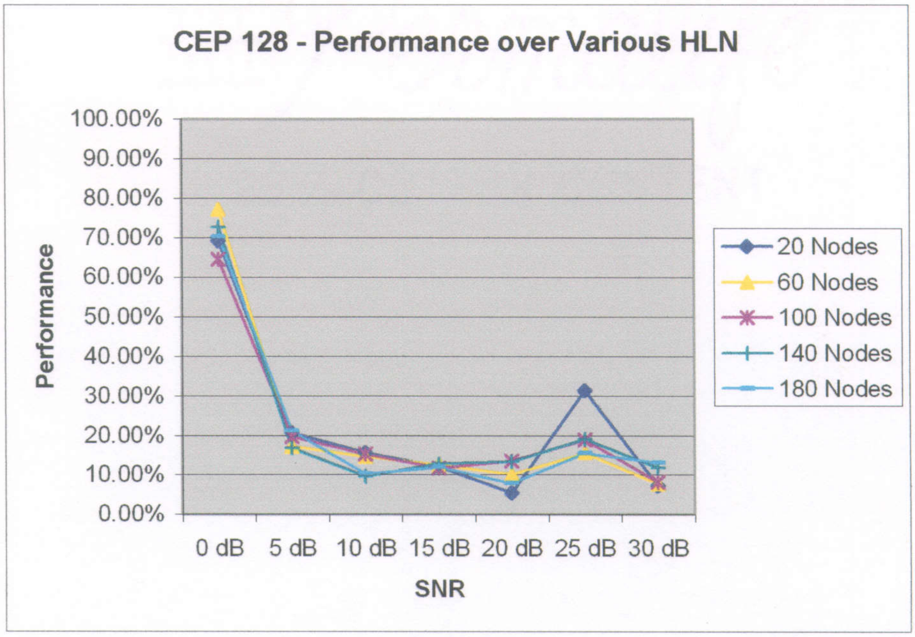


Figure 11: LP Cepstrum: 5dB Performance over Varying HLN, Training Size = 128 vectors.

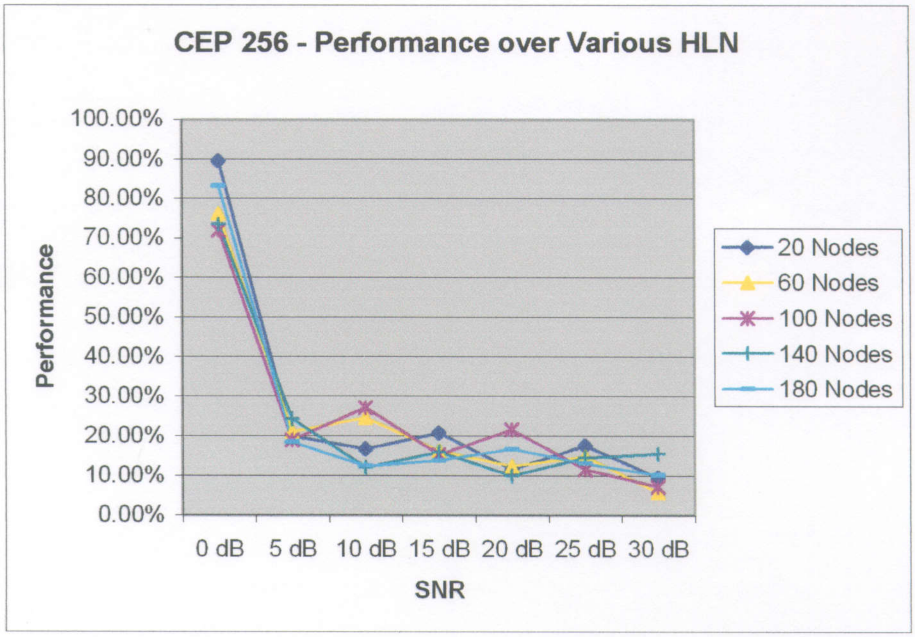


Figure 12: LP Cepstrum: 5dB Performance over Varying HLN, Training Size = 256 vectors.

4.1.2 1dB Class Increment

Before implementing a multiple classifier system, the same single MLP architecture is created, with data extracted in increments of 1dB. Again, due to the poor performance measure used with the 5dB spacing, the Overall Average Absolute Error (OAAE) will be used to compare estimation performance among varying parameters.

The 31 class classifier system resulted in a poor estimation, supporting the initial prediction that single classifier system is not sufficient for these particular features. Both the hard and soft decision OAAE values were collected over varying training data sizes, as well as a varying number of hidden layer nodes. Due to the increased amount of training data (31 classes, 12-dimensional data), an increased number of hidden layer nodes were utilized.

Table 5: Hard Decision Results for various Training Data Size, Single MLP

	# HLN				
Data Size	**100**	**300**	**500**	**700**	**900**
8	7.91 ±2.87 dB	7.35±1.61 dB	7.01±1.66 dB	6.86±1.97 dB	6.82±2.23 dB
16	4.77±1.82 dB	6.13±2.19 dB	7.00±1.95 dB	7.10±2.35 dB	11.67±0.62 dB
32	7.52±0.28 dB	8.51±1.66 dB	7.56±0.96 dB	8.08±2.75 dB	8.70±0.75 dB
64	7.23±0.76 dB	6.78±1.34 dB	6.64±0.97 dB	7.12±1.13 dB	5.67±2.18 dB
128	6.44±2.77 dB	**4.46±1.38** dB	8.21±3.06 dB	6.73±1.84 dB	6.86±2.68 dB
256	7.02±1.11 dB	8.03±2.99 dB	6.42±3.20 dB	6.75±2.65 dB	9.95±0.78 dB
512	9.64±2.09 dB	6.79±3.11 dB	6.58±1.66 dB	8.42±2.17 dB	7.56±1.54 dB

Table 6: Soft Decision Results for various Training Data Sizes, Single MLP

	# HLN				
Data Size	**100**	**300**	**500**	**700**	**900**
8	6.07±1.69 dB	7.54±1.68 dB	6.23±1.09 dB	6.50±1.29 dB	7.55±0.90 dB
16	8.84±1.46 dB	6.51±2.91 dB	7.10±0.74 dB	7.13±1.84 dB	7.15±0.54 dB
32	8.22±0.46 dB	5.32±1.39 dB	8.17±2.48 dB	7.47±0.56 dB	7.25±1.29 dB
64	7.94±0.99 dB	8.97±2.24 dB	5.33±1.61 dB	7.92±2.05 dB	6.61±1.29 dB
128	6.58±2.80 dB	**4.61±1.33** dB	7.93±2.84 dB	6.63±1.10 dB	7.33±2.52 dB
256	7.36±0.75 dB	7.87±3.05 dB	6.22±2.99 dB	6.75±2.61 dB	9.71±0.54 dB
512	8.81±2.05 dB	5.92±2.27 dB	6.61±1.52 dB	6.23±1.76 dB	7.45±1.42 dB

Table 5 and Table 6 depict the hard and soft decision results obtained by using a single MLP architecture with 1dB training increment over 5 trials. The number of training vectors, as well as the number of hidden layer nodes of the classifier, was altered. This was done to determine the optimal conditions for using a single MLP with 12-dimensional cepstrum data. For both hard and soft decision, the optimal training/testing conditions occurred at a training data size of 128 vectors, with 300 hidden layer nodes. This resulted in a hard decision OAAE of 4.46±1.38 dB, and a soft decision OAAE of 4.61±1.33 dB.

These particular tests were conducted a total of 5 times to gain knowledge of the variation of performance, with the values in Table 5 and Table 6 representing those gathered results. The possibility of overfitting the data is also a major concern for these tests. The number of hidden layer nodes required to train the data will decrease once a multiple classifier system is implemented as each classifier becomes a 2-class classifier.

4.2 Multiple Classifier System

This section presents the results of the various multiple classifier system architectures which were implemented in determining conditions for a robust SNR estimation system.

4.2.1 Train AWGN, Test AWGN

While three different types of noise will ultimately be introduced to the classification system, it was determined that the path to search for optimal conditions must first analyze each type of noise individually. As can be seen from the previous section, a single MLP classifier produced a best soft decision OAAE value of 4.61±1.33, which is far from ideal for an estimation system.

Instead, a multiple classifier system was created that combines classifier outputs to determine an estimate of the SNR of the unknown corrupted signal. This methodology was implemented by training 33 MLP classifiers, each being trained on an SNR level from -1dB to 31dB on data corrupted with purely additive white Gaussian noise (AWGN). The anti-data for each individual MLP consisted of compressed data from all other SNR levels concatenated together to equal the same length as the original training data. The anti-data comprised of 4 training vectors from each of the 32 remaining SNR levels, totalling 128 training vectors.

One of the main motivational reasons for equal lengths of training data and anti-data was to eliminate any bias which may be inherent when training the MLP itself. This phenomenon occurs due to the classifier recognizing an overload of data for a particular class as the probability of that class occurring in the test data. For the SNR estimation system, a SNR decibel level is equally likely to occur as any other SNR level. Other pattern recognition problems may welcome the overload of training data for a specific class, but training data remained equal for all tests to remove any bias.

Before any assumptions could be made as to the size of the training data which should be used, the cepstrum feature was again implemented with several classifiers of varying sizes of training data. Table 7 through Table 10 show the results from using a 33-MLP classifier system with training data sizes of 32, 64, 128 and 256 respectively. The training data size which resulted in the lowest OAAE values would be used throughout all future classifier systems, which was 128 vectors. While each feature is unique, the cepstrum feature was used to draw conclusions to the size of the training data required for optimal classification for all features. Exhausting all possibilities for each feature with different network structures and different training sizes would be computationally expensive.

Table 7: Hard and Soft Decision OAAE Results for 33-MLP system, each MLP trained on 32 training vectors and 32 anti-data vectors.

Train Size=32		
HLN	**Hard**	**Soft**
10	6.01 ± 1.74 dB	4.68 ± 2.10 dB
30	5.77 ± 1.53 dB	4.35 ± 1.74 dB
50	5.76 ± 1.42 dB	5.34 ± 1.91 dB
70	5.81 ± 1.48 dB	5.64 ± 1.73 dB
90	6.04 ± 1.19 dB	5.87 ± 1.65 dB

Table 8: Hard and Soft Decision OAAE Results for 33-MLP system, each MLP trained on 64 training vectors and 64 anti-data vectors.

Train Size=64		
HLN	Hard	Soft
10	6.10 ± 1.68 dB	5.34 ± 1.68 dB
30	5.32 ± 1.68 dB	5.35 ± 1.68 dB
50	5.54 ± 1.68 dB	5.45 ± 1.68 dB
70	7.76 ± 1.68 dB	5.81 ± 1.68 dB
90	8.34 ± 1.68 dB	6.07 ± 1.68 dB

Table 9: Hard and Soft Decision OAAE Results for 33-MLP system, each MLP trained on 128 training vectors and 128 anti-data vectors.

Train Size=128		
HLN	Hard	Soft
10	4.32 ± 1.35 dB	4.31 ± 1.41 dB
30	7.27 ± 2.43 dB	4.38 ± 1.80 dB
50	5.04 ± 1.74 dB	4.45 ± 1.69 dB
70	6.31 ± 1.13 dB	5.30 ± 1.96 dB
90	6.83 ± 1.79 dB	6.13 ± 1.87 dB

Table 10: Hard and Soft Decision OAAE Results for 33-MLP system, each MLP trained on 256 training vectors and 256 anti-data vectors.

Train Size=256		
HLN	Hard	Soft
10	4.66 ± 1.41 dB	4.41 ± 1.35 dB
30	5.32 ± 1.54 dB	4.64 ± 1.62 dB
50	6.81 ± 1.72 dB	4.68 ± 1.51 dB
70	6.93 ± 1.78 dB	4.82 ± 1.72 dB
90	7.01 ± 1.69 dB	4.91 ± 1.93 dB

Once it had been established that a training size of 128 vectors was most beneficial, each of the features were trained and tested using a multiple classifier system. The number of hidden layer nodes was also altered, as each feature and its respective data may require a more complex network to accurately model the data. All seven features were implemented in this fashion, with the lowest hard and soft OAAE values shown in Table 11. Table 11 below displays the optimal feature thus far is the variance of the cepstrum, with an overall soft OAAE of 3.40 ± 0.14 dB. The number of hidden layer nodes which resulted in the lowest OAAE values is displayed underneath the feature in parentheses.

Table 11: AWGN - Hard and Soft Decision OAAE Results, Training Size = 128.

	Hard	Soft
ACW **(10)**	4.65 ± 1.53 dB	3.45 ± 0.66 dB
CEP **(10)**	4.32 ± 1.34 dB	4.31 ± 1.29 dB
LAR **(40)**	4.23 ± 1.19 dB	3.92 ± 1.28 dB
LSF **(40)**	4.12 ± 0.50 dB	3.76 ± 0.23 dB
PFL **(40)**	5.24 ± 1.78 dB	4.47 ± 0.68 dB
REFL **(10)**	4.51 ± 1.30 dB	3.57 ± 0.70 dB
CEP_VAR **(40)**	3.86 ± 0.12 dB	3.40 ± 0.14 dB

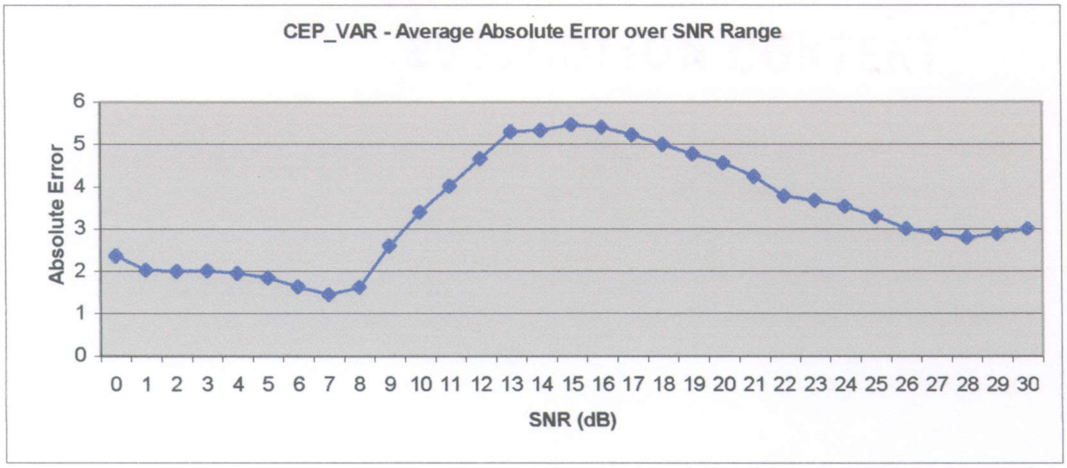


Figure 13: CEP_VAR - AWGN: Absolute Error over SNR Test Range

A plot of the average absolute errors across the range of SNR test levels can be seen in Figure 13, depicting the areas of the test spectrum which are most often misclassified. The variance of the cepstrum has a difficult time estimating SNR levels in the middle of the SNR spectrum. If another feature is better able to classify the mid-spectrum SNR levels, a fusion method would likely reduce the average absolute error as well as the overall average absolute error values. Implementation of PCA will ultimately lower the estimation error. Also, this is just the first step in designing a robust estimation system as these classifiers have only been trained on one type of noise.

Table 12: CEP_VAR - AWGN - Average Absolute Error for Specific SNR Range

SNR	AAE
0-5	2.04 dB
0-10	2.09 dB
0-15	2.99 dB
0-20	3.47 dB
0-25	3.51 dB
0-30	3.42 dB

4.2.2 Train Pink, Test Pink

A set of 33 classifiers were created and trained with corrupted data corresponding to a specific SNR level from -1 to 31dB. Instead of white Gaussian noise, the additive noise was pink noise. Rather than conducting the tests which determine the optimal training size for pink noise data, the previous result of 128 vectors will be used and continued for the remainder of the project.

Our goal is to develop a robust system trained on three different types of noise, so these experiments are rather important to determine the classification capability of the extracted features with pink noise. The variance of the cepstrum was again the most prominent feature, with a soft decision OAAE value of 3.95 ± 0.29 dB. The number of hidden layer nodes which resulted in the lowest OAAE values is displayed underneath the feature in parentheses.

Table 13: CEP_VAR - Pink - Hard and Soft Decision OAAE Results, Training Size = 128.

	Hard	**Soft**
ACW **(40)**	5.60 ± 0.70 dB	3.98 ± 0.71 dB
CEP **(40)**	5.68 ± 1.75 dB	4.77 ± 0.93 dB
LAR **(40)**	6.72 ± 2.67 dB	4.13 ± 0.69 dB
LSF **(40)**	5.20 ± 0.65 dB	4.98 ± 0.43 dB
PFL **(40)**	4.91 ± 1.19 dB	4.15 ± 0.48 dB
REFL **(10)**	6.12 ± 1.90 dB	4.46 ± 1.18 dB
CEP_VAR **(40)**	4.82 ± 1.10 dB	3.95 ± 0.29 dB

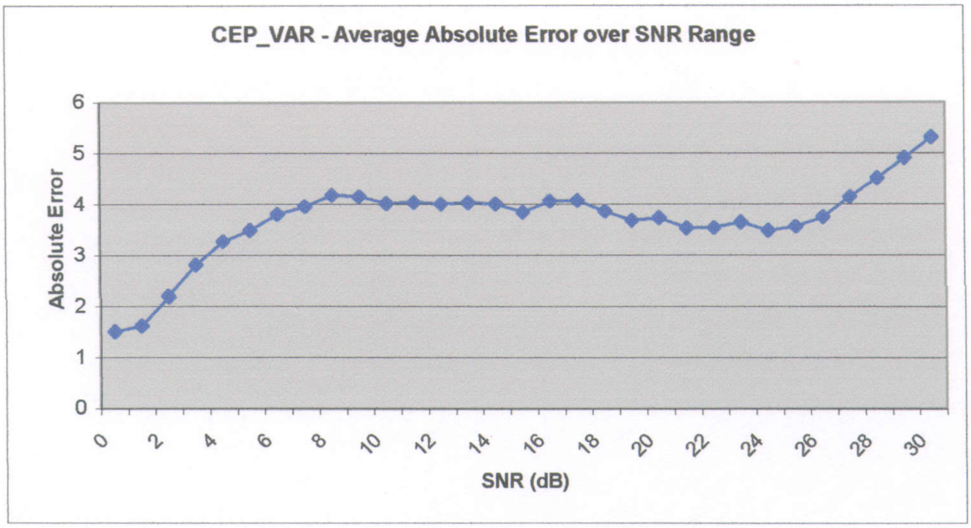


Figure 14: CEP_VAR - Pink: Absolute Error over SNR Test Range

Rather than relying solely on the OAAE value for a classifier system, it may also be useful to examine a plot of the average absolute errors over the range of test SNR decibel levels. Figure 14 is a plot which illustrates SNR ranges where the greatest errors are being made during the estimation process. The majority of errors during the estimation process occur at the high SNR levels, emphasizing the classifier's inability to distinguish between relatively 'clean' speech signals.

Table 13 displays the average absolute error for specific SNR ranges, to better gauge the performance of the feature with this type of classifier structure. While the overall average absolute error provides a simple method of comparison, it may not be explaining the entire chronicle of results.

4.2.3 Train CPV, Test CPV

In addition to white Gaussian noise and pink noise, telephone channel noise (CPV) is also a type of noise of interest for this approach. Training data size remained constant at 128 vectors, with an MLP trained at each SNR level from -1dB to 31dB. The variance of the cepstrum revealed to be the best feature for CPV noise. The soft decision OAAE value of 4.02 ± 0.15 dB was slightly ahead of the second best feature, reflection coefficients. The number of hidden layer nodes which resulted in the lowest OAAE values is displayed underneath the feature in parentheses.

Table 14: CEP_VAR - CPV - Hard and Soft Decision OAAE Results, Training Size = 128.

	Hard	Soft
ACW **(10)**	5.7 ± 1.59 dB	5.02 ± 1.30 dB
CEP **(40)**	5.92 ± 1.26 dB	4.69 ± 0.86 dB
LAR **(70)**	5.90 ± 1.76 dB	4.53 ± 0.50 dB
LSF **(10)**	6.83 ± 0.77 dB	4.98 ± 0.62 dB
PFL **(40)**	5.31 ± 0.89 dB	4.82 ± 0.99 dB
REFL **(70)**	5.62 ± 1.92 dB	4.24 ± 0.18 dB
CEP_VAR **(10)**	5.40 ± 0.55 dB	4.02 ± 0.15 dB

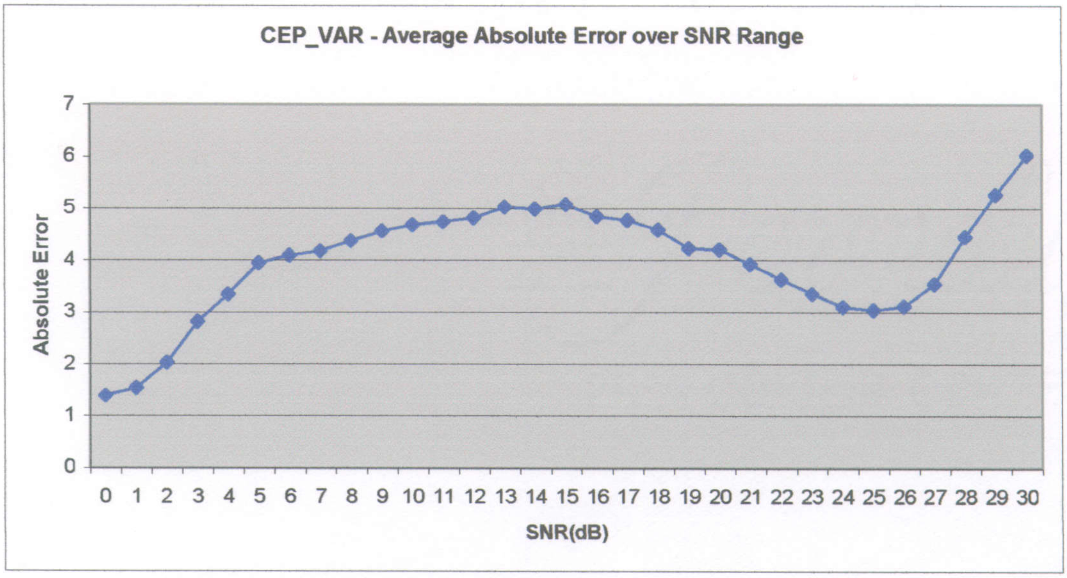


Figure 15: CEP_VAR - CPV: Absolute Error over SNR Test Range

Aside from just gauging the performance of the feature by the overall average absolute values, Table 15 illustrates the average absolute error for a specific range of SNR levels. This was computed in an attempt to further understand which SNR ranges had the poorest estimation, for a particular feature.

Table 15: CEP_VAR - CPV - Average Absolute Error for Specific SNR Range

SNR	AAE
0-5	2.52 dB
0-10	3.37 dB
0-15	3.86 dB
0-20	4.02 dB
0-25	3.90 dB
0-30	4.00 dB

4.2.4 Relative Robustness

Before classifiers were trained on all three types of noise, a simple analysis was conducted to determine the risk associated with a mismatch of train and test data. This study will determine how well an estimation system trained on one type of noise can estimate a different type of noise. The following experiments were implemented:

- Train additive white Gaussian noise / Test Pink noise - CEP
- Train additive white Gaussian noise / Test CPV noise - LSF
- Train Pink noise / Test CPV noise - PFL

This simple analysis will ultimately determine whether it's necessary to train classifiers on all three types of noise, or whether training on one type of noise will suffice. The mismatch of train and test data proves to be costly; with the resulting soft decision OAAE values averaging over 12 dB.

4.2.4.1 Train AWGN, Test Pink

A group of 33 classifiers were trained with cepstrum data extracted from speech signals corrupted with additive white Gaussian noise. Once the classifiers were trained, the testing process continued with using data comprised of cepstrum features extracted from pink noise corrupted signals. The corresponding OAAE values can be seen below:

Table 16: Train/Test Mismatch #1

Hard Decision OAAE	Soft Decision OAAE
13.01 dB	11.91 dB

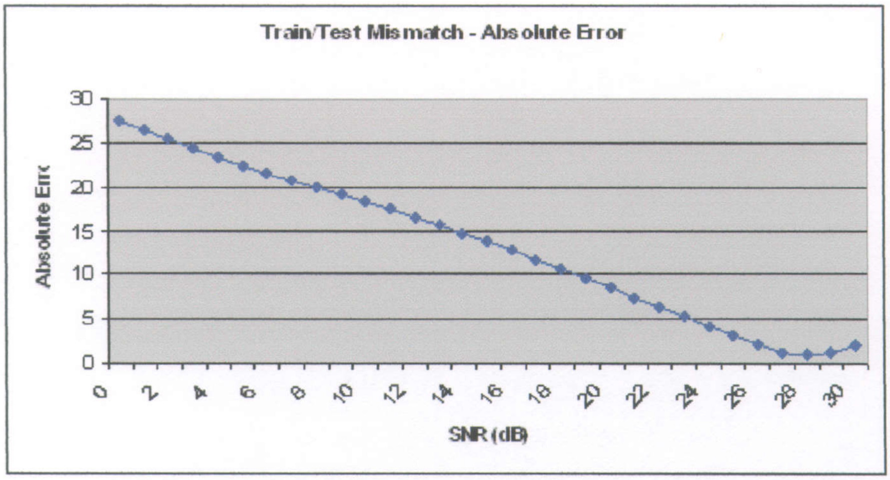


Figure 16: Train/Test Mismatch #1- Absolute Error over Test SNR Range

Both Table 16 and Figure 16 depict the risk associated with having a mismatch between train and test data. The plot of the average absolute errors in Figure 16 appears to be transversed, but the differences between the train and test data are large enough to skew the resulting estimate of the entire system. This illustrates that a system trained on only one type of noise would be devastating in estimating the SNR of an unknown signal, unless the type of additive noise was known.

4.2.4.2 Train AWGN, Test CPV

A second trial for the mismatch of data was performed featuring a classification system trained on LSF vectors extracted from AWGN corrupted signals, and tested LSF features extracted from CPV corrupted signals. The mismatch was detrimental to the system, as the soft decision OAAE was 10.46dB. The corresponding OAAE values can be seen below:

Table 17: Train/Test Mismatch #2

Hard Decision OAAE	Soft Decision OAAE
16.00 dB	10.46 dB

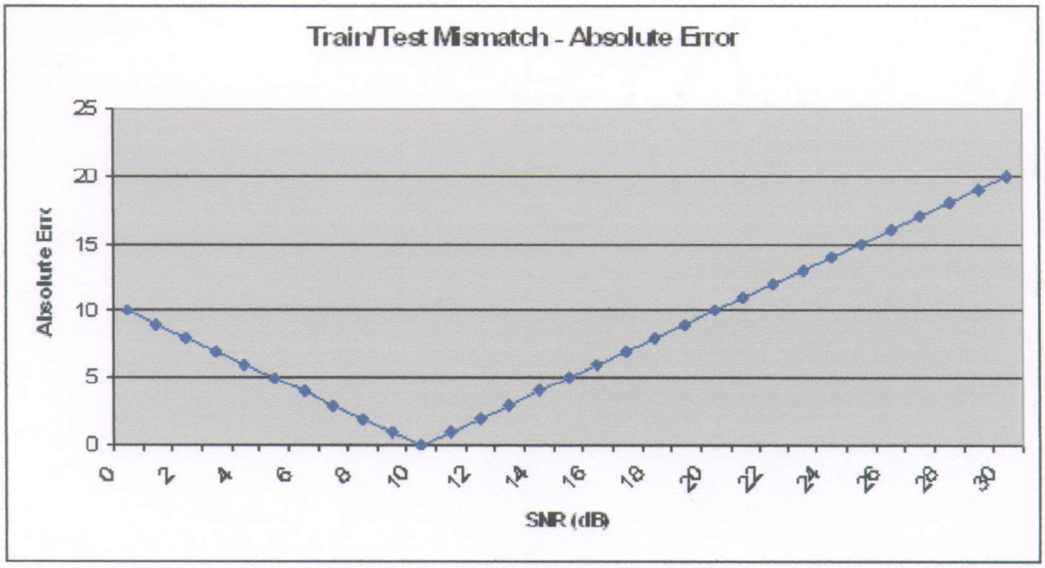


Figure 17: Train/Test Mismatch #2 - Absolute Error over Test SNR Range

4.2.4.3 Train Pink, Test CPV

The final trial for the mismatch of data consisted of a classification system trained with PFL vectors extracted from corrupted pink noise signals, and tested with PFL vectors extracted from corrupted CPV signals. The mismatch was detrimental to the system, as the soft decision OAAE was 10.46dB. The corresponding OAAE values can be seen below:

Table 18: Train/Test Mismatch #3

Hard Decision OAAE	Soft Decision OAAE
16.00 dB	13.95 dB

4.2.5 Robust Estimation System

The results from the previous section clearly illustrate the need for an abundant amount of training data with not only with one type of noise, but rather on data collected from speech signals corrupted with different types of noise. If an estimation system were to be implemented in a real-life scenario, the type of noise which may occur is quite hard to predict. This would motivate avoiding an estimation system trained on only one type of noise, and propel towards a robust estimation system.

A robust system was devised and created consisting of training data collected from three different types of noise. The training data for each MLP represented features extracted from three different types of noise, for a specific SNR level. The test data consisted of three individual sets of extracted feature data: signals corrupted with AWGN, signals corrupted with pink noise, and signals corrupted with CPV channel noise. Once the network was successfully trained, the three individual test sets were passed through the network one at a time to determine the associated estimation error.

In addition to training and testing with the full 12-dimensional data, principal component analysis (PCA) was implemented to convey the most amount of information in the least number of dimensions. All PCA possibilities were exhausted in search of optimal training/testing conditions. This section contains those results for all seven features.

4.2.5.1 Adaptive Component Weighting Cepstrum (ACW)

The ACW feature was analyzed and implemented in a classifier system with each of the classifiers being trained on all three types of noise. Each classifier's output was used in determining both the hard and soft decision OAAE values. After the classifiers had been trained, the first test data set consisted of speech signals corrupted with additive white Gaussian noise. The data set was passed through the group of classifiers and an average absolute error (AAE) was computed for each SNR level. Once the first data set was completed and the OAAE values for both hard and soft decision were calculated, the second data set was inputted to the system. OAAE values were computed for each individual data set, with the overall performance of the classification system being the average of the OAAE values. Only the soft decision OAAE values are shown for the overall performance of each classification system, as the hard decision OAAE was not a value representative of the classification performance. Table 19 has both the hard and soft decision OAAE values for each type of test noise, as well as the overall soft decision OAAE.

Table 19: ACW – 12 Dimensional Non-PCA Results – 10 HLN

	Hard Decision	Soft Decision	Overall Soft Decision
AWGN	6.19+/-0.68 dB	5.67+/-0.94 dB	5.11+/-0.46 dB
PINK	5.63+/-0.42 dB	4.75+/-0.27 dB	
CPV	5.73+/-0.35 dB	4.92+/-0.15 dB	

The results for the ACW feature portray that when a classifier is trained on three different types of noise, it displays some bias towards one type of noise. The trained classifiers resulted in bias towards pink noise, with white Gaussian noise and telephone channel noise slightly behind. This bias is created due to the different types of data being fed into the classifier and relying upon the network to discern amongst all three.

In addition to training classifiers on the 12-dimensional data, Principal Component Analysis (PCA) was also implemented and exhausted for all possible combinations. PCA trials were conducted for all possible dimensions (n < 12), with the classifiers being trained on the transformed data. Table 20 illustrates the performance of the classifier system on each type of noise, in descending order of dimensionality. With PCA, the data has been transformed into a new feature space with the number of dimensions retained ($n < 12$) corresponding to the n highest eigenvalues. The table of results (Table 20) illustrates that the lowest calculated OAAE value occurs when training only 6 dimensions (5.45+/-0.19 dB), corresponding to the transformed data with the 6 largest eigenvalues. The number of hidden layer nodes which resulted in the lowest OAAE values is displayed underneath the feature in parentheses.

When comparing the results for the 12-dimensional ACW and with PCA, the results illustrate that PCA does not help to reduce the OAAE. One of the explanations to this result is that ACW is known to be a feature that is robust to noise, which in turn will complicate the classifier's ability to discern between SNR levels. PCA was able to compact the necessary information of the data into 6 transformed dimensions, reducing the amount of training data and the computational complexity of the neural networks.

Table 20: ACW PCA OAAE Results

Dimension	Type of	PCA		
		Hard (dB)	Soft (dB)	Overall Soft (dB)
11 (40)	AWGN	7.16+/-1.56	6.55+/-2.24	5.73+/-0.71
	PINK	6.16+/-0.39	5.10+/-1.31	
	CPV	6.27+/-0.63	5.53+/-2.48	
10 (70)	AWGN	7.80+/-1.17	7.44+/-1.50	6.89+/-0.53
	PINK	7.41+/-1.42	6.91+/-1.75	
	CPV	7.35+/-1.38	6.72+/-1.99	
9 (40)	AWGN	6.44+/-0.85	5.30+/-0.25	5.73+/-0.37
	PINK	7.84+/-0.60	6.02+/-0.23	
	CPV	7.46+/-0.55	5.89+/-0.01	
8 (70)	AWGN	7.32+/-0.16	6.84+/-0.06	6.52+/-0.28
	PINK	6.96+/-0.34	6.45+/-0.15	
	CPV	6.87+/-0.21	6.26+/-0.05	
7 (70)	AWGN	7.24+/-1.37	6.26+/-1.31	6.43+/-0.44
	PINK	7.28+/-1.87	6.07+/-1.52	
	CPV	7.63+/-1.12	6.96+/-1.04	
6 (70)	AWGN	6.75+/-0.36	5.23+/-0.25	**5.45+/-0.19**
	PINK	6.95+/-0.71	5.52+/-0.23	
	CPV	6.69+/-0.22	5.61+/-0.60	
5 (40)	AWGN	7.96+/-2.32	6.76+/-2.45	6.90+/-0.19
	PINK	7.14+/-0.84	6.21+/-1.80	
	CPV	8.34+/-0.26	7.13+/-2.18	
4 (40)	AWGN	7.54+/-1.08	5.80+/-0.07	5.83+/-0.37
	PINK	7.94+/-0.52	6.24+/-0.10	
	CPV	6.55+/-0.06	5.46+/-0.78	
3 (40)	AWGN	7.50+/-0.70	6.29+/-0.06	6.53+/-0.47
	PINK	7.42+/-0.66	6.20+/-0.30	
	CPV	8.23+/-0.89	7.09+/-1.51	
2 (40)	AWGN	8.30+/-0.84	6.63+/-0.89	6.71+/-0.79
	PINK	8.65+/-1.88	6.73+/-0.89	
	CPV	8.45+/-0.19	5.75+/-0.22	
1 (40)	AWGN	8.46+/-0.77	8.38+/-0.26	7.21+/-1.20
	PINK	8.61+/-1.16	7.64+/-0.49	
	CPV	8.08+/-0.02	5.75+/-0.22	

4.2.5.2 LP Cepstrum (CEP)

Unlike ACW, principal component analysis aided in improving the estimation performance of the cepstrum feature. The overall soft decision OAAE value for the 12-dimensional cepstrum feature was 6.23+/-0.60 dB. Once PCA was implemented, the optimal dimensionality of the training data was reduced to only 5 dimensions with an overall soft decision OAAE value of 4.03+/-0.09 dB. That equates to a performance increase of over 2dB and a reduction in dimensionality from 12 to 5. The computational complexity of the network therefore decreases, which also decreases the amount of time required to train the set of classifiers.

Table 21: CEP – 12 Dimensional Non-PCA Results – 10 HLN

	Hard Decision	Soft Decision	Overall Soft Decision
AWGN	7.16+/-0.29 dB	6.29+/-0.31 dB	6.23+/-0.60 dB
PINK	6.68+/-0.66 dB	5.58+/-0.39 dB	
CPV	7.46+/-0.52 dB	6.83+/-0.73 dB	

The 5-dimensional PCA results shown in Table 22 illustrate that the classifiers did not show any bias towards one type of noise over another. The bias which was evident in the 12-dimensional classification trials is nonexistent with the 5-dimensional PCA trial. The soft decision OAAE values for each type of noise were almost identical and further reinforce that the LP cepstrum feature benefited from PCA.

Table 22: CEP PCA Results

Dimension	Type of	PCA		
		Hard (dB)	Soft (dB)	Overall Soft (dB)
11 (10)	AWGN	7.16+/-1.98	6.46+/-1.75	6.04+/-0.36
	PINK	6.71+/-1.08	5.70+/-0.60	
	CPV	7.00+/-1.85	5.96+/-1.36	
10 (10)	AWGN	7.80+/-2.11	6.85+/-2.17	6.44+/-0.53
	PINK	7.30+/-1.63	5.80+/-1.56	
	CPV	7.64+/-1.76	6.67+/-1.65	
9 (10)	AWGN	5.80+/-0.95	5.13+/-0.99	4.95+/-0.19
	PINK	5.55+/-0.99	4.73+/-0.92	
	CPV	5.66+/-1.17	5.00+/-1.23	
8 (10)	AWGN	6.58+/-0.84	5.69+/-1.29	5.40+/-0.26
	PINK	6.06+/-0.83	5.16+/-0.71	
	CPV	6.06+/-1.26	5.34+/-1.64	
7 (10)	AWGN	5.41+/-1.07	4.50+/-0.50	4.52+/-0.33
	PINK	5.52+/-0.65	4.88+/-0.52	
	CPV	4.80+/-0.38	4.18+/-0.59	
6 (10)	AWGN	4.98+/-0.62	4.23+/-0.68	4.30+/-0.13
	PINK	5.11+/-0.63	4.46+/-0.57	
	CPV	5.09+/-0.39	4.22+/-0.42	
5 (10)	AWGN	4.65+/-0.22	4.13+/-0.14	**4.03+/-0.09**
	PINK	4.68+/-0.21	4.02+/-0.21	
	CPV	4.94+/-1.05	3.94+/-0.54	
4 (10)	AWGN	6.85+/-1.05	5.08+/-0.65	4.71+/-0.53
	PINK	4.73+/-0.33	5.00+/-1.60	
	CPV	5.23+/-0.90	4.07+/-0.17	
3 (10)	AWGN	5.66+/-0.45	5.34+/-0.55	4.82+/-0.43
	PINK	4.82+/-0.87	4.52+/-1.11	
	CPV	4.87+/-0.73	4.59+/-0.95	
2 (10)	AWGN	5.30+/-0.71	5.04+/-0.49	4.88+/-0.14
	PINK	5.37+/-0.42	4.81+/-0.34	
	CPV	5.49+/-0.62	4.77+/-0.50	
1 (10)	AWGN	9.41+/-1.55	9.17+/-2.19	7.76+/-1.23
	PINK	7.12+/-1.44	6.63+/-1.81	
	CPV	8.00+/-0.63	7.46+/-0.85	

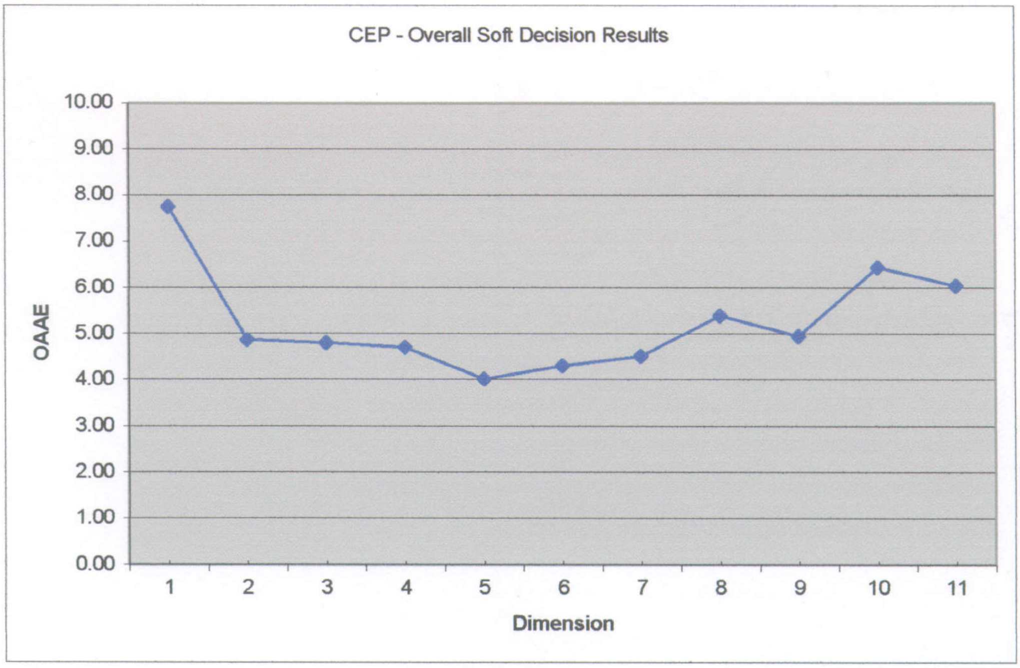


Figure 18: CEP - Plot of the Overall Soft Decision results across PCA Dimensions.

Figure 18 illustrates the overall soft decision OAAE values for each of the corresponding PCA dimensions. When only 5 dimensions of the transformed feature are retained, the system is better suited to estimate the SNR of the unknown speech signal. A trend also exists between the OAAE of a classifier system and the number of transformed dimensions required for training. The transformed cepstrum data required 5 dimensions to achieve optimal performance, which was a significant improvement over the original 12-dimensional implementation.

4.2.5.3 Log Area Ratios (LAR)

The 12-dimensional LAR tests resulted in a heavy bias towards pink noise. White Gaussian noise and CPV noise both had OAAE values significantly higher than pink noise, which can deduce an explanation that the classifier was unable to correctly discern between SNR levels for certain types of noise.

Once PCA had been implemented with the original 12-dimensional data, a performance increase was clearly evident and can be seen in Table 24. PCA allowed the dimensionality of the training data to be cut in half, with the lowest OAAE value recorded at 6 dimensions.

Table 23: LAR – 12 Dimensional Non-PCA Results – 10 HLN

	Hard Decision	Soft Decision	Overall Soft Decision
AWGN	8.77+/-1.19 dB	8.19+/-1.13 dB	6.90+/-1.24 dB
PINK	6.50+/-0.90 dB	5.58+/-0.91 dB	
CPV	7.87+/-1.31 dB	6.94+/-1.28	

The overall soft decision OAAE performance of 6 dimensions was 4.99+/-0.09 dB, which was almost 2dB lower than the original 12-dimensional results. For the log-area ratios, PCA is beneficial in achieving better SNR estimation than the original 12-dimensiona features, with a fewer number of dimensions.

Table 24: LAR PCA Results

Dimension	Type of	PCA		
		Hard (dB)	Soft (dB)	Overall Soft (dB)
11 (10)	AWGN	8.49+/-1.47	7.79+/-1.19	6.70+/-0.95
	PINK	6.96+/-0.99	5.84+/-0.80	
	CPV	7.22+/-1.31	6.46+/-1.22	
10 (10)	AWGN	7.19+/-0.90	6.64+/-0.86	5.98+/-0.55
	PINK	6.38+/-0.56	5.60+/-0.55	
	CPV	6.27+/-0.90	5.69+/-0.84	
9 (10)	AWGN	6.92+/-0.67	6.32+/-0.83	5.42+/-0.75
	PINK	5.66+/-0.41	4.84+/-0.22	
	CPV	5.77+/-0.35	5.10+/-0.42	
8 (10)	AWGN	7.59+/-0.57	7.08+/-0.60	6.21+/-0.71
	PINK	6.63+/-0.48	5.73+/-0.47	
	CPV	6.76+/-0.43	5.83+/-0.44	
7 (10)	AWGN	6.85+/-0.55	6.46+/-0.49	6.00+/-0.41
	PINK	6.29+/-0.26	5.60+/-0.35	
	CPV	6.44+/-0.39	5.92+/-0.36	
6 (10)	AWGN	5.73+/-0.38	5.09+/-0.36	**4.99+/-0.09**
	PINK	5.73+/-0.48	4.92+/-0.49	
	CPV	5.55+/-0.37	4.96+/-0.38	
5 (40)	AWGN	6.40+/-0.69	5.97+/-0.67	5.81+/-0.34
	PINK	5.98+/-0.40	5.40+/-0.31	
	CPV	6.43+/-0.70	6.07+/-0.63	
4 (40)	AWGN	6.44+/-0.56	5.77+/-0.65	5.66+/-0.09
	PINK	6.33+/-0.68	5.58+/-0.57	
	CPV	6.24+/-0.50	5.64+/-0.67	
3 (40)	AWGN	6.96+/-0.71	6.24+/-0.55	5.80+/-0.39
	PINK	6.18+/-0.27	5.41+/-0.24	
	CPV	6.39+/-0.57	5.74+/-0.66	
2 (40)	AWGN	7.30+/-0.18	5.73+/-0.45	6.29+/-0.69
	PINK	8.39+/-1.08	7.11+/-0.77	
	CPV	7.87+/-0.78	6.04+/-0.59	
1 (10)	AWGN	8.87+/-1.35	7.35+/-0.84	7.32+/-0.36
	PINK	8.18+/-1.31	7.56+/-1.37	
	CPV	8.66+/-1.26	7.27+/-0.83	

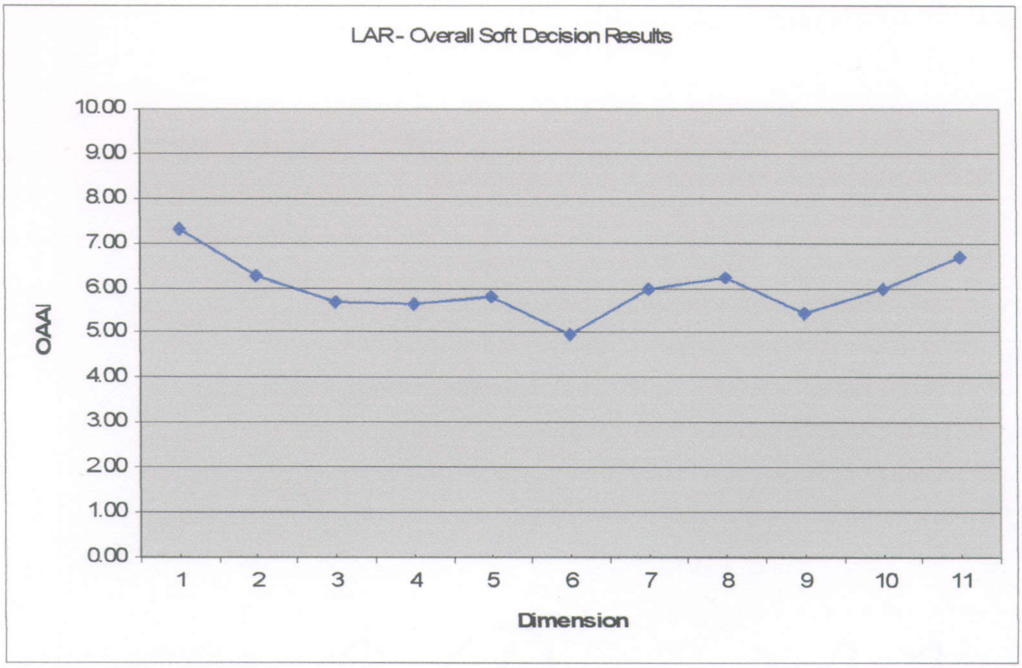


Figure 19: LAR - Plot of the Overall Soft Decision results across PCA Dimensions.

The original 12-dimensional overall soft OAAE value was 6.90+/-1.24 dB, which is easily met and exceeded with variety of PCA dimensions. Similar to the cepstrum performance plot, a continuous trend exists with the number of selected PCA dimensions and the resulting overall soft decision OAAE. The classifiers require at least 3 dimensions to achieve 1dB lower than the 12-dimensional experiments, with 6 dimensions providing an almost 2dB decrease in OAAE.

4.2.5.4 Line Spectral Frequencies (LSF)

The LSF feature is a prime example of why PCA was implemented for this particular application. The 12-dimensional soft decision results were the highest of all the features with an overall average of 7.20+/-0.09 dB. While these results would normally rule out the LSF as a suitable feature for this robust estimation system, the implementation of PCA turned out to be beneficial, decreasing the OAAE over 3dB and reducing the dimensionality from 12 to 4.

Whether the chosen number of dimensions was two or ten, the use of PCA with the LSF feature decreased the estimation error when compared to the original 12-dimensional feature results. The impressive results shown here with the LSF feature will be beneficial during the fusion process, in which several classifier systems' opinions will be combined to give an overall improved estimate. The 4-dimensional PCA results for LSF are among the top 3 results for all seven features.

Table 25: LSF – 12 Dimensional Non-PCA Results – 10 HLN

	Hard Decision	Soft Decision	Overall Soft Decision
AWGN	8.17+/-0.43 dB	7.30+/-0.42 dB	7.20+/-0.09 dB
PINK	6.74+/-0.00 dB	7.10+/-0.43 dB	
CPV	8.12+/-0.27 dB	7.20+/-0.48 dB	

Table 26: LSF PCA Results

Dimension	Type of	PCA		
		Hard (dB)	Soft (dB)	Overall Soft (dB)
11 (10)	AWGN	8.94+/-0.76	8.48+/-0.58	7.80+/-0.76
	PINK	7.64+/-0.74	6.91+/-0.66	
	CPV	8.76+/-0.94	8.01+/-0.74	
10 (10)	AWGN	8.20+/-0.32	7.58+/-0.81	7.19+/-0.52
	PINK	7.14+/-0.62	6.57+/-0.84	
	CPV	7.90+/-0.76	7.43+/-0.89	
9 (10)	AWGN	8.46+/-0.91	7.90+/-1.12	7.61+/-0.67
	PINK	7.56+/-0.87	6.80+/-1.09	
	CPV	8.90+/-1.16	8.12+/-1.25	
8 (10)	AWGN	7.79+/-0.88	7.43+/-0.85	6.55+/-0.79
	PINK	6.33+/-0.53	5.79+/-0.54	
	CPV	6.99+/-0.98	6.42+/-0.89	
7 (10)	AWGN	6.55+/-0.60	6.09+/-0.70	5.39+/-0.61
	PINK	5.63+/-0.52	4.84+/-0.51	
	CPV	5.76+/-0.63	5.22+/-0.62	
6 (10)	AWGN	5.58+/-0.28	5.01+/-0.15	4.81+/-0.24
	PINK	5.54+/-0.41	4.89+/-0.15	
	CPV	5.01+/-0.23	4.53+/-0.26	
5 (10)	AWGN	6.09+/-0.52	5.50+/-0.73	4.93+/-0.50
	PINK	5.09+/-0.45	4.46+/-0.50	
	CPV	5.51+/-0.81	4.84+/-0.84	
4 (40)	AWGN	4.86+/-0.50	3.86+/-0.40	**4.06+/-0.17**
	PINK	5.30+/-0.77	4.21+/-0.47	
	CPV	6.32+/-1.82	4.10+/-0.31	
3 (40)	AWGN	5.03+/-0.32	4.36+/-0.51	4.38+/-0.14
	PINK	5.52+/-1.00	4.53+/-0.64	
	CPV	5.13+/-0.90	4.25+/-0.61	
2 (40)	AWGN	5.75+/-1.19	4.88+/-1.08	5.20+/-0.31
	PINK	6.26+/-0.99	5.53+/-0.55	
	CPV	5.99+/-0.99	5.18+/-0.58	
1 (10)	AWGN	8.34+/-1.62	8.24+/-1.24	7.46+/-1.23
	PINK	6.64+/-0.96	5.97+/-0.78	
	CPV	8.93+/-0.58	8.17+/-0.59	

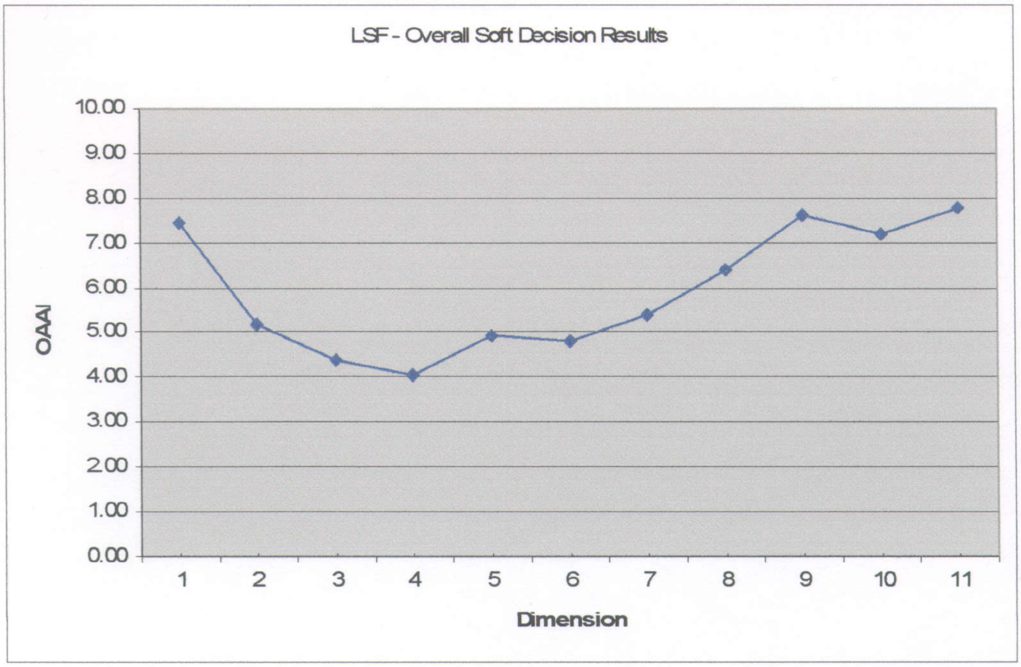


Figure 20: LSF - Plot of the Overall Soft Decision results across PCA Dimensions.

Any selection of PCA dimensions from two through ten would yield an improved estimation of an unknown signal than the original 12-dimensional data. Figure 20 is an accurate representation of the change in estimation error over the number of selected dimensions. The computational complexity of the network decreases significantly when trained with 4 dimensions rather than 12.

These results for the LSF feature are surprising due to the performance of the LSF feature when trained and tested on only one type of noise. The LSF feature was consistently one of the worst features for those tests. Clearly the implementation of PCA has greatly increased the classification capability of the feature data.

4.2.5.5 Postfilter Cepstrum (PFL)

The postfilter cepstrum results appear similar to the ACW results, in the sense that PCA did little or nothing to improve the performance of the estimation. For PFL, the initial 12-dimensional tests resulted in an overall soft OAAE value of 5.00+/-0.49 dB. The best results for PCA occurred at three dimensions (4.72+/-0.35 dB), resulting in a simple conclusion that PCA aided in reducing the dimensionality and slightly decreasing the average estimation error.

Table 27: PFL – 12 Dimensional Non-PCA Results – 10 HLN

	Hard Decision	Soft Decision	Overall Soft Decision
AWGN	5.58+/-1.88 dB	5.55+/-1.63 dB	5.00+/-0.49 dB
PINK	4.60+/-1.38 dB	4.51+/-1.43 dB	
CPV	4.98+/-0.87 dB	4.95+/-0.88 dB	

From the beginning of the implementation, it was predicted that the postfilter cepstrum feature would not be consistent for estimation. Similar to the ACW feature, the postfilter feature was initially created and established as a speech feature which is robust to additive noise. The liftering technique performed upon the LP cepstrum produces the postfilter cepstrum, in which it de-emphasizes the first few components of the LP cepstrum [11]. Those first few components vary the most with additive noise, so the de-emphasis placed on those components greatly hinders the performance of the feature. Even with the transformation of the data via PCA, the problem inherently lies in the robustness of the feature to additive noise.

Table 28: PFL PCA Results

Dimension	Type of	PCA		
		Hard (dB)	Soft (dB)	Overall Soft (dB)
11 (40)	AWGN	7.85+/-1.09	7.34+/-1.28	5.68+/-1.37
	PINK	4.63+/-1.21	4.73+/-1.03	
	CPV	4.66+/-0.81	4.97+/-0.49	
10 (10)	AWGN	8.07+/-0.81	8.22+/-0.64	5.44+/-1.32
	PINK	6.64+/-0.55	6.54+/-0.67	
	CPV	5.92+/-0.69	6.03+/-0.62	
9 (70)	AWGN	5.58+/-2.00	5.70+/-1.70	5.23+/-0.39
	PINK	5.04+/-1.54	4.95+/-1.50	
	CPV	4.94+/-2.26	5.03+/-1.91	
8 (40)	AWGN	6.41+/-1.29	6.17+/-1.24	5.77+/-0.45
	PINK	5.34+/-0.82	5.25+/-0.76	
	CPV	6.01+/-1.41	5.88+/-1.24	
7 (10)	AWGN	6.18+/-1.59	6.06+/-1.22	6.29+/-0.84
	PINK	5.67+/-1.09	5.54+/-0.91	
	CPV	7.49+/-2.39	7.26+/-1.86	
6 (10)	AWGN	5.26+/-1.40	5.25+/-1.08	5.59+/-0.79
	PINK	5.02+/-0.87	4.99+/-0.66	
	CPV	6.49+/-1.59	6.55+/-1.22	
5 (40)	AWGN	5.00+/-1.70	4.82+/-1.32	5.25+/-1.15
	PINK	4.41+/-1.41	4.30+/-0.91	
	CPV	6.67+/-1.79	6.62+/-1.37	
4 (10)	AWGN	5.80+/-1.68	5.68+/-1.19	5.78+/-1.05
	PINK	4.77+/-1.77	4.72+/-1.30	
	CPV	7.39+/-2.34	6.93+/-2.10	
3 (70)	AWGN	4.32+/-1.31	4.50+/-1.29	**4.72+/-0.35**
	PINK	5.21+/-0.93	5.14+/-0.95	
	CPV	4.32+/-1.22	4.53+/-1.04	
2 (40)	AWGN	6.56+/-1.45	5.79+/-1.17	6.34+/-0.90
	PINK	6.41+/-1.38	5.78+/-1.20	
	CPV	7.94+/-2.35	7.43+/-2.14	
1 (10)	AWGN	9.32+/-1.39	8.98+/-1.21	8.54+/-0.68
	PINK	9.51+/-1.52	8.91+/-1.08	
	CPV	8.63+/-2.30	7.71+/-1.64	

4.2.5.6 Reflection Coefficients (REFL)

The reflection coefficients showed a decrease in average estimation error when comparing the 12-dimensional trial with the optimal PCA dimensions. The PCA dimension which reported the best overall soft decision OAAE was 6 dimensions, resulting in an estimation error of 4.00+/-0.68 dB. This is 1.5dB lower than what was achieved with the 12-dimensional trial, and also had twice as many data points.

One of the most noticeable results in Table 29 is the difference between the soft decision values for the three different types of noise. Clearly, the classifiers show some bias for pink noise over the other two. That bias is also present with the PCA results, as pink noise generally outperformed the other two types of noise. Since none of the other features showed this as heavily as the REFL feature, it's believed that the cause of this bias lies directly within the extracted data from all three types of noise.

Table 29: REFL – 12 Dimensional Non-PCA Results – 10 HLN

	Hard Decision	Soft Decision	Overall Soft Decision
AWGN	7.03+/-1.89 dB	6.62+/-0.96 dB	5.51+/-1.03 dB
PINK	4.77+/-1.92 dB	4.45+/-1.55 dB	
CPV	5.41+/-1.79 dB	5.47+/-1.33 dB	

Table 30: REFL PCA Results

Dimension	Type of	PCA		
		Hard (dB)	Soft (dB)	Overall Soft (dB)
11 (10)	AWGN	6.77+/-2.21	6.75+/-2.09	5.23+/-1.29
	PINK	4.21+/-1.48	4.14+/-1.47	
	CPV	4.56+/-1.96	4.81+/-1.67	
10 (10)	AWGN	6.54+/-0.78	6.54+/-0.54	5.27+/-1.27
	PINK	4.15+/-1.13	3.88+/-0.91	
	CPV	5.25+/-1.03	5.39+/-0.77	
9 (10)	AWGN	6.00+/-1.06	6.00+/-0.90	5.11+/-0.99
	PINK	3.74+/-1.17	3.97+/-0.86	
	CPV	5.18+/-1.09	5.37+/-0.97	
8 (10)	AWGN	5.43+/-1.98	5.22+/-1.98	4.76+/-0.46
	PINK	4.19+/-0.53	4.25+/-0.40	
	CPV	4.85+/-1.01	4.81+/-0.99	
7 (10)	AWGN	6.19+/-0.84	5.75+/-0.87	5.28+/-0.51
	PINK	4.94+/-1.55	4.70+/-1.03	
	CPV	5.58+/-1.42	5.39+/-1.05	
6 (10)	AWGN	4.56+/-1.16	4.46+/-1.01	**4.00+/-0.68**
	PINK	3.25+/-0.54	3.18+/-0.52	
	CPV	4.34+/-1.47	4.38+/-1.22	
5 (40)	AWGN	5.46+/-1.24	5.46+/-0.98	4.66+/-0.96
	PINK	3.72+/-1.18	3.52+/-0.98	
	CPV	4.94+/-1.44	5.00+/-1.04	
4 (40)	AWGN	5.63+/-1.24	5.59+/-0.81	4.85+/-1.02
	PINK	3.56+/-1.27	3.61+/-0.74	
	CPV	5.31+/-1.34	5.34+/-0.80	
3 (70)	AWGN	3.92+/-1.56	3.79+/-1.55	5.22+/-1.27
	PINK	5.92+/-0.83	5.46+/-1.26	
	CPV	7.09+/-1.60	6.42+/-1.78	
2 (70)	AWGN	5.00+/-2.67	4.72+/-1.69	5.16+/-0.49
	PINK	6.16+/-1.42	5.72+/-1.37	
	CPV	5.72+/-1.36	5.03+/-1.17	
1 (70)	AWGN	6.41+/-2.89	6.42+/-2.36	7.08+/-1.29
	PINK	6.85+/-2.18	6.16+/-1.66	
	CPV	10.25+/-1.73	8.64+/-0.57	

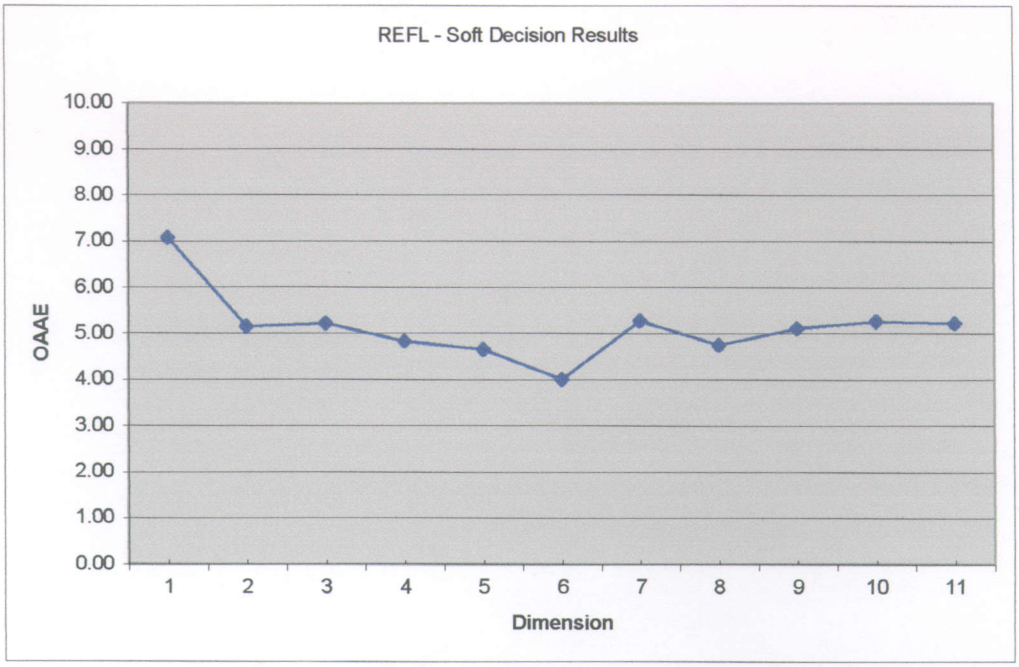


Figure 21: REFL - Plot of the Overall Soft Decision results across PCA Dimensions.

The plot seen in Figure 21 displays a consistent performance over the various PCA dimensions. Aside from three outliers, the soft decision OAAE values are around 5dB.

A majority of the other features have one prominent dimension at which the error is minimal with the rest of the dimensions linearly increasing from that point, such as Figure 20. This phenomenon occurs due to the addition of unnecessary dimensions in the classification of data. However, the REFL feature results show an almost steady performance aside from three outlying dimensions. The REFL feature is rather robust in performance when choosing a correct number of dimensions for best representation.

4.2.5.7 LP Cepstrum Variance (CEP_VAR)

Perhaps the most fascinating feature of the approach is the variance of the linear predictive cepstrum. When a set of classifiers are trained on the 12-dimensional data for all three types of noise, the overall soft decision OAAE is 4.23+/-0.37 dB. This estimation error is slightly lower than when compared to similar calculations from the other features.

Once PCA was implemented and used for the variance of the cepstrum, the estimation error decreased further. A set of classifiers only need to be trained on the first two dimensions of the PCA transformed data to achieve the lowest error, resulting in an overall soft decision OAAE of 3.83+/-0.15 dB. These findings mark this feature as the most prominent feature in estimating SNR of an unknown speech signal, with cepstrum and line spectral frequencies second and third, respectively.

Table 31: CEP_VAR – 12 Dimensional Non-PCA Results – 10 HLN

	Hard Decision	Soft Decision	Overall Soft Decision
AWGN	4.98+/-0.40 dB	3.78+/-0.14 dB	4.23+/-0.37 dB
PINK	6.11+/-0.34 dB	4.47+/-0.19 dB	
CPV	6.14+/-0.33 dB	4.45+/-0.20 dB	

Table 32:CEP_VAR PCA Results

Dimension	Type of	PCA		
		Hard (dB)	Soft (dB)	Overall Soft (dB)
11 (10)	AWGN	6.32+/-0.88	4.82+/-0.08	5.13+/-0.25
	PINK	7.40+/-1.21	5.29+/-0.31	
	CPV	7.16+/-1.20	5.28+/-0.40	
10 (40)	AWGN	4.97+/-0.41	4.22+/-0.26	4.31+/-0.11
	PINK	5.65+/-0.63	4.44+/-0.16	
	CPV	5.38+/-0.07	4.26+/-0.15	
9 (10)	AWGN	5.85+/-0.81	5.00+/-0.71	5.16+/-0.64
	PINK	7.32+/-0.61	5.35+/-0.74	
	CPV	7.44+/-0.79	5.24+/-0.59	
8 (40)	AWGN	5.74+/-0.71	4.62+/-0.66	4.95+/-0.27
	PINK	6.56+/-0.46	5.06+/-0.16	
	CPV	4.27+/-1.19	5.15+/-0.45	
7 (40)	AWGN	4.83+/-0.29	3.92+/-0.41	4.71+/-0.65
	PINK	6.35+/-0.04	5.19+/-0.28	
	CPV	6.25+/-0.26	5.01+/-0.26	
6 (40)	AWGN	4.57+/-0.35	4.06+/-0.39	5.01+/-0.79
	PINK	6.25+/-0.27	5.47+/-0.33	
	CPV	6.30+/-0.34	5.51+/-0.37	
5 (40)	AWGN	5.72+/-0.25	4.37+/-0.38	5.15+/-0.64
	PINK	7.05+/-0.18	5.56+/-0.60	
	CPV	7.02+/-0.18	5.53+/-0.57	
4 (40)	AWGN	5.40+/-0.03	4.26+/-0.52	4.76+/-0.58
	PINK	6.77+/-0.27	5.44+/-0.97	
	CPV	6.09+/-0.17	4.98+/-0.74	
3 (10)	AWGN	4.61+/-1.03	3.76+/-0.34	4.08+/-0.27
	PINK	5.06+/-1.12	4.17+/-0.51	
	CPV	5.26+/-1.20	4.31+/-0.56	
2 (10)	AWGN	4.11+/-0.12	3.70+/-0.07	**3.83+/-0.15**
	PINK	4.15+/-0.13	3.77+/-0.07	
	CPV	4.39+/-0.20	4.01+/-0.09	
1 (10)	AWGN	4.41+/-0.26	4.02+/-0.35	4.41+/-0.36
	PINK	4.70+/-0.39	4.44+/-0.40	
	CPV	5.07+/-0.42	4.78+/-0.46	

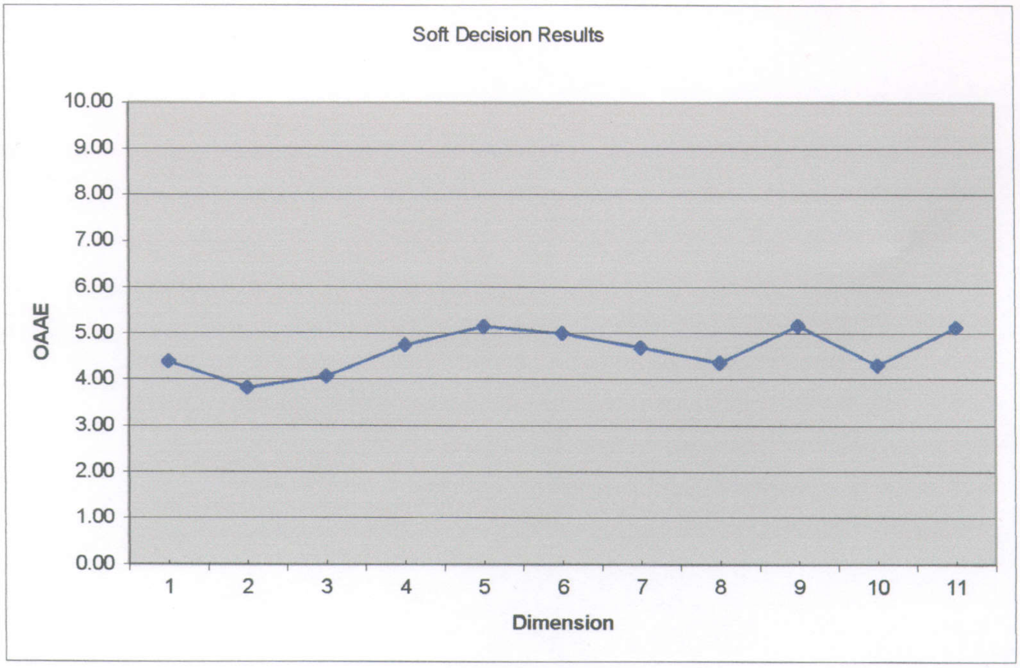


Figure 22: CEP_VAR - Plot of the Overall Soft Decision results across PCA Dimensions.

With only 2 dimensions required, the performance of the transformed feature will later be one of the main features for fusion tests. The computational complexity and time required to train the classifier definitely help this feature achieve the title as the best choice for estimating SNR of an unknown signal for three different types of noise.

4.2.6 Train All 3, Test CMV

The motivation for the previous section was to determine the optimal conditions for training a group of classifiers on three different types of noise. The next step was to take those optimal conditions and test the classifiers with a fourth type of noise, which has not been seen. These trials will give an idea as to the robustness of each feature in estimating the SNR of an unknown signal with a previously unseen type of noise. The classifiers were trained on white Gaussian, pink, and telephone channel noise (CPV), and were tested with another type of telephone channel noise (CMV). The results for each are in Table 33.

The first two columns in Table 33 state which feature and the number of dimensions used for each trial. Column #3 re-iterates the overall soft decision results of each scenario from the previous results section. The fourth and fifth columns illustrate the new results, for which the classifiers have been tested with CMV noise.

Once again, both the cepstrum variance and the LSF lead the rest of the features with the lowest estimation error. In the case of the 12-dimensional cepstrum variance, the test results for CMV noise exceeded the previous results for white Gaussian, pink, and CPV. This is unexpected because the previous tests were trained with three types of noise and tested with each type of noise. The new test results for CMV were testing with a type of noise not previously seen by the classifier. It is hypothesized that the manner in which some of the features changed due to the CMV noise likely modeled the same manner in which the features changed for CPV noise.

In addition to these CMV results, fusion will also take place to determine whether a combination of features and their classifier system opinions will give a more accurate estimation of the SNR of an unknown signal. Rather than just focusing on the best results, all features will be considered and exhausted in determining optimal fusion conditions.

Table 33: Results for classifiers tested with type of noise not previously seen.

Feature	Dimensions	Robust: Overall Soft Decision (dB)	CMV: Hard Decision (dB)	CMV: Soft Decision (dB)
ACW (10)	12-D	5.11+/-0.46	9.53+/-0.55	8.09+/-0.46
ACW (40)	6-D, PCA	5.45+/-0.19	10.35+/-2.29	9.09+/-1.78
CEP (10)	12-D	6.23+/-0.60	7.82+/-0.81	7.22+/-0.51
CEP (40)	5-D, PCA	4.03+/-0.09	5.59+/-1.05	4.86+/-0.81
LAR (40)	12-D	6.90+/-1.24	7.32+/-1.28	6.33+/-1.53
LAR (10)	6-D, PCA	4.99+/-0.09	6.14+/-1.73	5.20+/-2.41
LSF (10)	12-D	7.20+/-0.09	6.84+/-0.87	5.87+/-0.24
LSF (10)	4-D, PCA	4.06+/-0.17	5.28+/-0.50	**4.40+/-0.19**
PFL (10)	12-D	5.00+/-0.49	8.46+/-2.25	6.25+/-1.32
PFL (70)	4-D, PCA	4.72+/-0.35	5.77+/-1.28	5.54+/-0.19
REFL (10)	12-D	5.51+/-1.03	6.78+/-0.81	6.14+/-0.44
REFL (10)	6-D, PCA	4.00+/-0.68	6.52+/-1.71	6.10+/-2.01
CEP_VAR (40)	12-D	4.23+/-0.37	5.15+/-0.12	**4.12+/-0.48**
CEP_VAR (40)	2-D, PCA	3.83+/-0.15	4.95+/-0.29	**4.49+/-0.35**

4.3 Decision-Level Fusion

The motivation behind fusion is to take the classifier system estimates for several different features and combine them in such a method to reduce the Average Absolute Error (AAE) at each SNR level [28]. These fused outputs represent each network's estimation as to the signal-to-noise ratio of the test signal.

There are a wide variety of fusion approaches but the particular methods implemented for this approach include the mean, median, and trimmed mean of the estimates. All possible feature combinations have been exhausted, with the optimal results explained in this section. The results for mean and median are the same for fusion of two features. The trimmed mean calculation for fusion of only two features is not calculated. The results for median and trimmed mean are the same for both three and four features as well [29].

Once the classifier network had been trained with all three types of noise, each type of noise was tested individually to receive the absolute error value corresponding to each utterance. The decision level estimates from several features' classifier systems corresponding to each sentence have been fused, with the results listed in the following tables.

Table 34: Fusion Results when taking Mean of Network Estimates

	Soft Decision				
Features	**AWGN**	**Pink**	**CPV**	**Overall**	**CMV**
CEP, CEP_VAR	3.03 dB	3.07 dB	2.96 dB	**3.02 dB**	3.73 dB
CEP, CEP_VAR, LSF	2.74 dB	3.06 dB	3.27 dB	3.03 dB	**3.59 dB**
CEP, CEP_VAR, LSF, PFL	3.19 dB	3.15 dB	3.12 dB	3.15 dB	3.65 dB
CEP,CEP_VAR,LSF,PFL,LAR	3.81 dB	3.38 dB	3.47 dB	3.56 dB	3.65 dB
CEP,CEP_VAR,LSF,PFL,LAR, ACW	4.16 dB	3.64 dB	3.93 dB	3.91 dB	3.96 dB
CEP,CEP_VAR,LSF,PFL,LAR, ACW,REFL	4.51 dB	3.88 dB	4.35 dB	4.25 dB	4.18 dB

Table 35: Fusion Results when taking Median of Network Estimates

Features	Soft Decision				
	AWGN	Pink	CPV	Overall	CMV
CEP,CEP_VAR	3.03 dB	3.07 dB	2.96 dB	**3.02 dB**	3.73 dB
CEP,CEP_VAR,LSF	2.87 dB	3.21 dB	3.14 dB	3.07 dB	3.74 dB
CEP,CEP_VAR,LSF,ACW	3.22 dB	3.03 dB	3.04 dB	3.10 dB	**3.61 dB**
CEP,CEP_VAR,LSF,PFL	3.17 dB	3.17 dB	2.94 dB	3.09 dB	**3.60 dB**
CEP,CEP_VAR,LSF,LAR	3.40 dB	3.00 dB	2.95 dB	3.12 dB	3.84 dB
CEP,CEP_VAR,LSF,PFL,ACW	3.71 dB	3.24 dB	3.43 dB	3.46 dB	4.00 dB
CEP,CEP_VAR,LSF,PFL,LAR	3.80 dB	3.21 dB	3.39 dB	3.47 dB	4.03 dB
CEP,CEP_VAR,LSF,PFL,LAR,ACW	4.34 dB	3.57 dB	4.23 dB	4.05 dB	3.81 dB
CEP,CEP_VAR,LSF,PFL,LAR,ACW,REFL	4.99 dB	4.11 dB	5.30 dB	4.80 dB	3.99 dB

Table 36: Fusion Results when taking Trimmed Mean of Network Estimates

Features	Soft Decision				
	AWGN	Pink	CPV	Overall	CMV
CEP,CEP_VAR,LSF	2.87 dB	3.21 dB	3.14 dB	**3.07 dB**	3.74 dB
CEP,CEP_VAR,LSF,ACW	3.22 dB	3.03 dB	3.04 dB	3.10 dB	**3.61 dB**
CEP,CEP_VAR,LSF,PFL	3.17 dB	3.17 dB	2.94 dB	3.09 dB	**3.60 dB**
CEP,CEP_VAR,LSF,LAR	3.40 dB	3.00 dB	2.95 dB	3.12 dB	3.84 dB
CEP,CEP_VAR,LSF,PFL,ACW	3.73 dB	3.34 dB	3.51 dB	3.52 dB	3.96 dB
CEP,CEP_VAR,LSF,PFL,REFL	3.81 dB	3.34 dB	3.53 dB	3.56 dB	3.98 dB
CEP,CEP_VAR,LSF,PFL,ACW,LAR	4.26 dB	3.65 dB	4.05 dB	3.99 dB	3.83 dB
CEP,CEP_VAR,LSF,PFL,ACW,REFL	4.24 dB	3.66 dB	4.15 dB	4.02 dB	4.37 dB
CEP,CEP_VAR,LSF,PFL,ACW,LAR,REFL	4.70 dB	3.97 dB	4.57 dB	4.41 dB	4.15 dB

Table 34, Table 35, and Table 36 illustrate that the optimal fusion implementation scenarios occur with either two or three features. The results for two feature fusion would be important when choosing the optimal fusion combination, but the fusion test results for CMV also explain which set of features are more robust to estimating a type of noise not previously seen by the network. Although, the average absolute error computed for each SNR level might be a better discriminant when choosing an optimal condition.

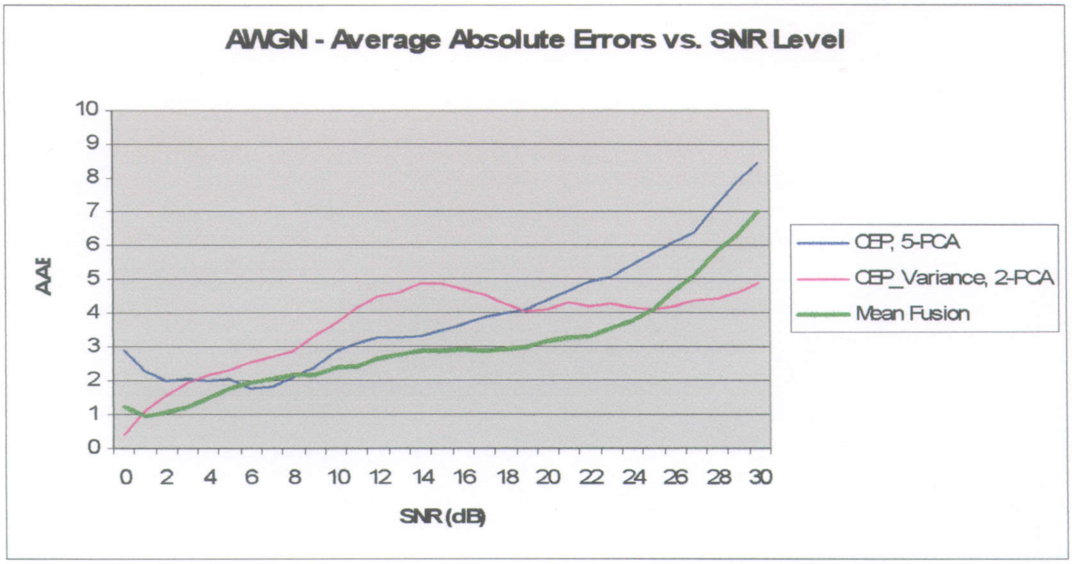


Figure 23: AWGN-Average Absolute Error over SNR Levels: Two Individual Features and Fusion

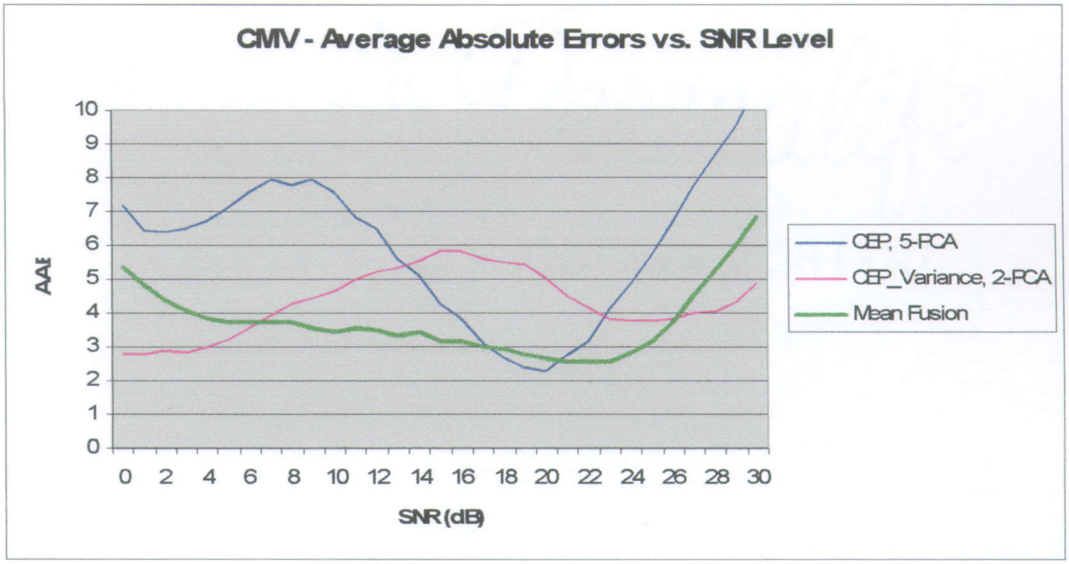


Figure 24: CMV-Average Absolute Error over SNR Levels: Two Individual Features and Fusion

Figure 23 and Figure 24 both depict the type of improvement in average absolute error with the implementation of fusion between CEP and CEP_VAR. The two figures represent white Gaussian and CMV error. The decision level fusion certainly aided in lowering the estimation error, and had no bearing on whether the type of noise had been previously seen by the classifier or not.

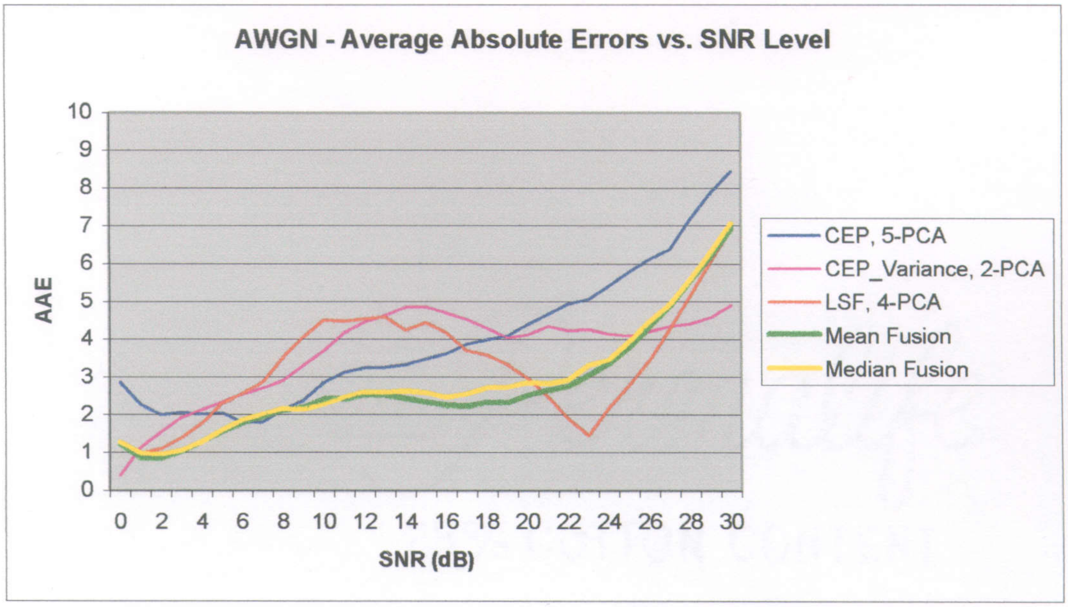


Figure 25: AWGN-Average Absolute Error over SNR Levels: Three Individual Features and Fusion

Table 37: AWGN - Average Absolute Errors within certain SNR Range

	CEP	CEP_VAR	LSF	Mean Fusion	Median Fusion
0-5 dB	2.22 dB	1.58 dB	1.47 dB	1.15 dB	1.20 dB
0-10 dB	2.21 dB	2.25 dB	2.40 dB	1.58 dB	1.61 dB
0-15 dB	2.55 dB	2.99 dB	3.05 dB	1.85 dB	1.91 dB
0-20 dB	2.90 dB	3.31 dB	3.17 dB	1.97 dB	2.10 dB
0-25 dB	3.34 dB	3.49 dB	2.98 dB	2.19 dB	2.33 dB
0-30 dB	3.96 dB	3.65 dB	3.33 dB	2.74 dB	2.87 dB

Figure 25 depicts that both types of fusion help the system in estimating the SNR of an unknown signal. The figure and table associated with these results are for training a system on all three types of noise, but only tested on AWGN. The results for the other two types of noise follow the same trend, which ultimately decreases the average absolute error of the entire system.

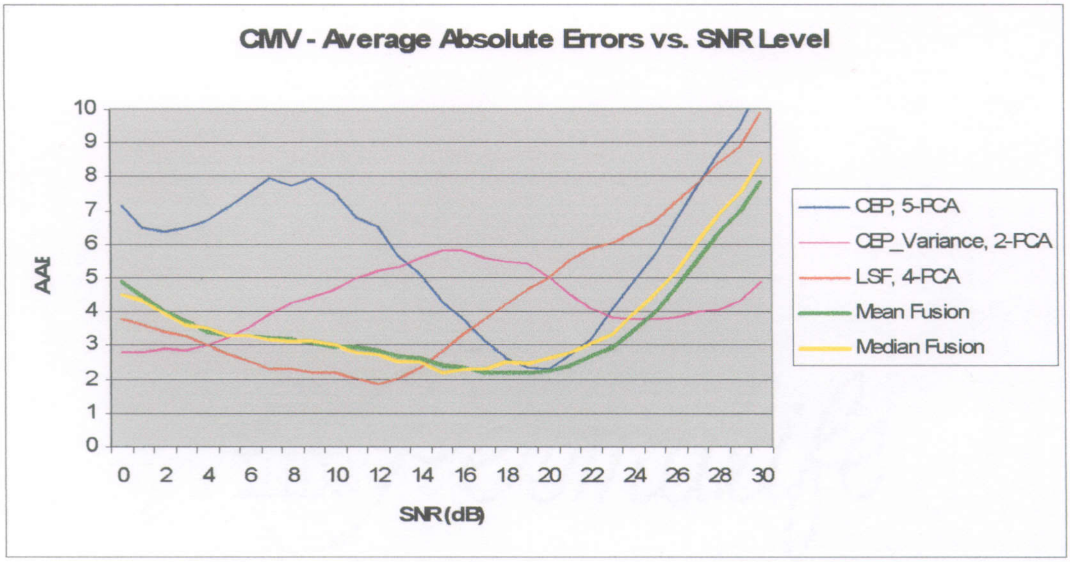


Figure 26: CMV-Average Absolute Error over SNR Levels: Three Individual Features and Fusion

Table 38: CMV - Average Absolute Errors within certain SNR Range

	CEP	CEP_VAR	LSF	Mean Fusion	Median Fusion
0-5 dB	6.72 dB	2.93 dB	3.32 dB	3.95 dB	3.88 dB
0-10 dB	7.19 dB	3.50 dB	2.86 dB	3.59 dB	3.55 dB
0-15 dB	6.71 dB	4.09 dB	2.66 dB	3.32 dB	3.24 dB
0-20 dB	5.79 dB	4.42 dB	3.03 dB	3.07 dB	3.05 dB
0-25 dB	5.47 dB	4.35 dB	3.63 dB	3.07 dB	3.15 dB
0-30 dB	5.99 dB	4.33 dB	4.40 dB	3.59 dB	3.74 dB

Fusion from estimates of test signals with a type of noise not seen by the classifier is useful, as certain features had a better estimate at specific SNR levels than others. The cepstrum AAE plot displays how the median and trimmed mean would be particularly useful in fusing the low SNR estimates.

CHAPTER 5 – CONCLUSIONS

This chapter has been divided into three sections. Section 5.1 provides a synopsis of the previous content mentioned in the earlier chapters. Section 5.2 outlines of the accomplishments of the research involved in this thesis, and lastly Section 5.3 provides recommendations and guidelines for future work.

5.1 Synopsis of Thesis

Chapter 1 presents the initial problem, as well as the complications of estimating an unknown signal without any prior information. Chapter 2 provides the general background information required to understand the proposed approach, including the derivations of the features and the structure of the classifiers. Chapter 3 provides a detailed synopsis of the approach including the methodology behind the training and testing procedures. Chapter 4 states the results gathered from each step of the approach. The improvement in results from one step to the next justified the approach taken in training and creating a robust estimation system.

5.2 Summary of Accomplishments

The goal of this thesis was to research and implement suitable features for the development of a robust signal-to-noise ratio estimation system through the use of a neural network. From the gathered results, the system will estimate the SNR of an unknown signal with an average estimate error of around 3dB through the use of decision level fusion. A comparison to gauge the relative accomplishment of the approach is shown below:

1. *To investigate seven speech signal features used primarily in speaker recognition system and their possible contribution to a signal-to-noise ratio estimation scheme.*

- In addition to implementing six features established for speaker identification, a new feature is devised and implemented for SNR estimation. This ultimately proves to be the most useful feature, as it consistently produced results exceeding the original expectations of the feature.

2. *To implement a neural network classifier approach to estimating signal-to-noise ratio.*

- While the MLP approach was implemented and nearly exhausted for the features of interest, there are still several other neural network classifiers which could also have been implemented. An ensemble of neural network classifiers could trained on specific conditions and different types of noise could ultimately decrease the estimation error. Principal component analysis increased the performance of the MLP classifier when compared to the original 12-dimensional results.

3. *To investigate the effect of classifier parameters in determining optimal neural network architecture.*

- The free parameters of the MLP, including the number of hidden layer nodes, were seen to have a drastic effect on a classifier's performance. A common downfall to training multilayer perceptron is the ever present notion of overtraining a classifier. For this reason, all of the tests conducted for this approach had the number of hidden layer nodes varied to determine optimal network architecture, as the number of hidden layer nodes required for optimal performance varied among the features.

4. *To study the effects of the extracted speech features when the original speech signal has been corrupted with one of three different types of noise; Additive White Gaussian Noise, Pink Noise, and Telephone Channel Noise.*

- The performance of the multiple classifier system, when tested on particular types of noise, showed to vary with each of the features. Some of the features are known for their robustness to additive noise, which was evident in their overall estimated error. Each of the different types of noise affected the classification ability of the features.

5. *To create and train a multiple classifier system robust to all three types of noise.*

- The preliminary tests which were run on only one type of noise gave rather subtle hints as to the performance of a particular feature when trained with all three types of noise. The variance of the cepstrum proved to be the most consistent feature for estimating the SNR of an unknown speech signal.

6. *To implement several fusion techniques in determining the optimal features to create a more robust estimation system.*

- Following the robust estimation system design, several features were fused together to lower the estimation error. The cepstrum, variance of the cepstrum, and LSF were among the most favorable features as the fusion combined network estimates over the entire test SNR spectrum. This is due in part to each feature and its respective classifiers' unique estimation ability within certain SNR decibel ranges.

5.3 Recommendations and Directions for Future Work

From viewing the results section, several conclusions can be drawn as to methods for improving the approach and the performance of the proposed estimation system. While the features can not be completely ruled out on their ability to model changes in noise level, much of the improvement lies in the design of the classifier and structure of the neural network.

The multilayer perceptron is perhaps one of the most widely used neural networks due to several of its characteristics, mainly being a global approximator. However, the downfall of the MLP lies in the training process as a created classifier's weights are never the same. The initial weights are chosen at random, and although the training data may remain the same, the ability of a classifier in modeling the training data is quite unpredictable on a regular basis. Rather than relying on 33 classifiers, instead an ensemble with more classifiers would be better suited to solving this estimation problem.

The underlying approach for an ensemble of classifiers states that a large set of diverse classifiers with each being trained on a subset of data from the same data set. Ideally, the ensemble of classifiers would combine outputs using either a simple or weighted majority vote [28]. This would mean that the output of the network does not rely on just a single of a small subset of classifiers, but rather a large collection of diverse classifiers. With 31 different SNR levels and various types of noise, an ensemble approach would be acceptable and could possibly outperform the results previously gathered.

A VQ classifier is also currently being investigated, which may prove to be a more consistent classifier than the multilayer perceptron. One of the downfalls of the

MLP is that the network has to differentiate and design boundary lines between the different classes. If the data is overlapping, which is the case for most of the implemented features, the MLP starts to fail as a classifier. Only in a situation such as the ensemble of classifiers might the neural network prevail, as that uses a 'mixture of experts' to determine class information. Overall, a feature based neural network approach to estimating the signal-to-noise ratio of an unknown signal proved to be a feasible approach, but there are several other classifiers which may be more suited for this type of classification problem.

References

[1] M. C. Huggins and J. J. Grieco, "Confidence Metrics for Speaker Identification", Int. Conf. on Spoken Language Processing, Denver, Colorado, 2002.

[2] M. C. Huggins and J. J. Grieco, "Speaker Identification Confidence Metrics For Heterogeneous Model Spaces", Proc. of the 8th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, Florida, pp. 440--443, July 2004.

[3] Rangachari, S., Loizou, P.C.; Yi Hu, "A noise estimation algorithm with rapid adaptation for highly nonstationary environments", IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, 2004, Volume 1, 17-21 May 2004

[4] Cohen, I., Berdugo, B., "Noise estimation by Minima Controlled Recursive Averaging for Robust Speech Enhancement", IEEE Signal Processing Letters, Vol. 9, No. 1, January 2002

[5] Krom, G., "A new cepstrum-based technique for the estimation of spectral signal-to-noise ratios in speech signals", ESCA Workshop on Speaker Characterization in Speech Technology, Edinburgh, Scotland, June 1990

[6] H. Hirsch and C. Ehrlicher, "Noise estimation techniques for robust speech recognition," in Proc. Int. Conf. on Acoust., Speech and Signal Processing (ICASSP), 1995, pp. 153–156.

[7] M. Zilovic, R. Ramachandran, "Speaker identification based on the use of robust cepstral features obtained from pole-zero transfer functions," IEEE Trans. Speech Audio Processing, vol. 6, pp. 260-267, May 1998

[8] K. T. Assaleh and R. J. Mammone, "New LP-derived features for speaker identification", IEEE Trans. on Speech and Audio Proc., vol. 2, pp. 630--638, Oct. 1994.

[9] T. F. Quatieri, Discrete Time Speech Signal Processing Principles and Practice Prentice Hall PTR, 2002.

[10] L. R. Rabiner and R. W. Schafer, Digital Processing of Speech Signals, Englewood Cliffs, NJ: Prentice-Hall, 1978

[11] Mammone, R.J., Zhang, X., Ramachandran, R.P., Robust Speaker Recognition, IEEE Signal Processing Magazine, Sept. 1996

[12] Reynolds, D.L., Head, L.M., Ramachandran, R.P., ASIC Implementation of Efficient Line Spectral Frequency Computation for Speech Coding Applications

[13] R. C. Rose. Environmental robustness in automatic speech recognition. *Robust2004 - ISCA and COST278 Workshop on Robustness in Conversational Interaction*, August 2004.

[14] Moreno P.J., Raj B., Gouvea E. and Stern R.M., "Multivariate-Gaussian-based cepstral normalization for robust speech recognition", Proc ICASSP95, May 1995.

[15] Acero A. and Stern R.M., "Environmental Robustness in Automatic Speech Recognition", ICASSP90, May 1990.

[16] Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COMM-28, pp. 84-95, Jan. 1980

[17] Duda, Richard O., and Peter E. Hart, Pattern Classification, 2nd ed. Wiley Interscience, 2000.

[18] Haykin, S., Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice Hall, Upper Saddle River, NJ, 1999

[19] Farrell, K.R., Mammone, R.J., Assaleh, K. T., Speaker Recognition Using Neural Networks and Conventional Classifiers, IEEE Trans. Speech Audio Processing, Vol. 2, No. 1, Part II, January 1994

[20] Kuncheva L., Combining Pattern Classifiers: Methods and Algorithms, John Wiley and Sons, Inc., Hoboken, NJ, 2005

[21] Wang, Xuechuan; Paliwal, Kuldip K.; Feature Extraction and Dimensionality Reduction Algorithms and Their Applications in Vowel Recognition, Journal of Pattern Recognition Society 36 (2003): 2429-2439.

[22] Jehan, T., Creating Music by Listening, PhD Thesis, Massachusetts Institute of Technology.

[23] Z. H. Hu, "Understanding and adapting to speaker variability using correlation-based principal component analysis", Dissertation of OGI. Oct. 10, 1999.

[24] Ding, Peilv; Zhang, Liming; Speaker Recognition Using Principal Component Analysis, International Conference on Neural Information Processing. Shanghai, China, 2001.

[25] Max Planck Institute for Psycholinguistics, "Computer Corpora and Databases: Timit Speech Database" June 1997, http://www.mpi.nl/world/tg/corpora/timit/timit.html

[26] Devore, Jay L.; Probability and Statistics for Engineering and the Sciences; 6th Edition; Belmont, CA; Brooks/Cole, 2004, pg 282-285.

[27] J. Kupin, “A wireline simulator (software),” CCR-P, Apr. 1993.

[28] Polikar R., “Ensemble Based Systems in Decision Making,” *IEEE Circuits and Systems Magazine*, vol.6, no. 3, pp. 21-45, 2006

[29] Ondusko, R., Marbach, M., McClellan, A., Ramachandran, R., Head, L M., “Blind Determination of the Signal to Noise Ratio of Speech Signals based on Estimation Combination of Multiple Features”, Asia Pacific Conference on Circuits and Systems, Singapore, 2006