

Performance Analysis of Speech Enhancement Algorithm for Robust Speech Recognition System

C.GANESH BABU¹, Dr.P.T.VANATHI², R.RAMACHANDRAN³, M.SENTHIL RAJAA³,
R.VENGATESH³

¹ Research Scholar (PSGCT)
Associate Professor / ECE,
BIT,
Sathyamangalam,
India.

E-mail:bits_babu@yahoo.co.in

² Assistant Professor / ECE,
PSGCT,
Coimbatore,
India.

E-mail:ptvani@yahoo.com

³ UG Scholar
BIT,
Sathyamangalam,
India.

Abstract: - Widely Speech Signal Processing has not been used much in the field of electronics and computers due to the complexity and variety of speech signals and sounds with the advent of new technology. However, with modern processes, algorithms, and methods which can process speech signals easily and also recognize the text. Demand for speech recognition technology is expected to raise dramatically over the next few years as people use their mobile phones as all purpose lifestyle devices. In this paper, an implementation of a speech-to-text system using isolated word recognition with a vocabulary of ten words (digits 0 to 9 with each 100 samples) and statistical modeling (Hidden Markov Model - HMM) for machine speech recognition was undertaken. In the training phase, the uttered digits are recorded using 8-bit Pulse Code Modulation (PCM) with a sampling rate of 8 KHz and saved as a wave file using sound recorder software. The system performs speech analysis using the Linear Predictive Coding (LPC) method of degree. From the LPC coefficients, the weighted cepstral coefficients and cepstral time derivatives are derived. From these variables the feature vector for a frame is arrived. Then, the system performs Vector Quantization (VQ) utilizing a vector codebook which result vectors form of the observation sequence. For a given word in the vocabulary, the system builds an HMM model and trains the model during the training phase. The training steps, from Speech Enhancement to HMM model building, are performed using PC-based Matlab programs. Our current framework uses a speech processing module includes Speech Enhancement algorithm with Hidden Markov Model (HMM)-based classification and noise language modeling to achieve effective noise knowledge estimation.

Key-Words: Hidden Markov Model, Vector Quantization, Speech Enhancement, Linear Predictive Coding, Speech Recognition.

1 Introduction

Currently there are many technical barriers in which the speech recognition system from meeting the modern application. An important drawback affect most of these application is harmful environmental noise and it reduces any system performance. Some of the system which is highly affected is new wireless communication voice services and mobile technology. The quality of speech can be enhanced by noise reduction algorithm. In this paper, Speech Enhancement Algorithm is used to suppress the noise from the input noisy signal [1]. The proposed method of Speech Recognition System for Robust noise environment is shown in the figure 1.



Fig.1 Proposed Robust Speech Recognition System

The paper is organized as follows. Section 2 gives the brief outlook of Adaptive Gain Equalization (AGE) for Speech Enhancement. Section 3 reviews the Hidden Markov Model. Section 3.1 discusses the Linear Predictive Coding Analysis. Section 3.2 gives

the Vector Quantization and says how samples are trained and also the recognition of speech samples. Results and discussions are tabulated and discussed in Section 4. The paper is concluded in Section 5.

2 Adaptive Gain Equalization

The Adaptive Gain Equalization (AGE) method for Speech Enhancement separates itself from the traditional methods of improving the Signal to Noise Ratio (SNR) of a signal corrupted by noise, through moving away from noise suppression and focusing primarily on speech boosting. Noise suppression traditionally, like spectral subtraction, looks at subtracting an estimated noise bias from the signal corrupted by noise. Whereas speech boosting aims to enhance the speech part of the signal by adding an estimate of the speech itself, thus boosting the speech part of the signal. The difference between noise suppression and speech boosting is presented in figure 2. It shows the noise estimate being subtracted from a noise corrupted signal. While in figure 2 an estimate of the speech signal is used to boost the speech in the noise corrupted signal.

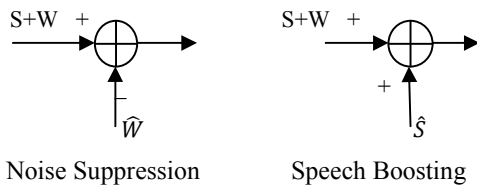


Fig 2. Difference between Noise Suppression and Speech boosting

The AGE method of Speech Enhancement Algorithm (SEA) relies on a few basic ideas [13]. The first of which is that a speech signal which is corrupted by band limited noise can be divided into a number of subbands and each of these subbands can be individually and adaptively boosted according to a SNR estimate in that particular subband. In each subband, a short term average is calculated simultaneously with an estimate of a slowly varying noise floor level [3]. By using the short term average and floor estimate, a gain function is calculated per subband through dividing the short term average by the floor estimate. This gain function is multiplied with the corresponding signal in each subband to form an output per subband. The sum of the outputs from each subband forms the final output signal, which

should contain a higher SNR when compared to the original noisy signal.

The AGE acts as a speech booster, which is adaptively looking for a subband speech signal to boost. Outlining that speech energy is a highly non-stationary input amplitude excursion, if there is no such excursions no alteration to the subband will be performed, the AGE will remain idle, as a result of the quotient between the short term magnitude average and the noise floor estimate being unity, with them being approximately the same [14]. If speech is present the short term magnitude average will change with the noise floor level remaining approximately unchanged, thus amplifying the signal in the subband at hand due to the quotient becoming larger than unity.

We have an acoustical discrete time speech signal denoted $s(n)$ and a discrete time noise signal denoted $w(n)$. The noise corrupted speech signal $x(n)$ can then be written as

$$x(n) = s(n) + w(n) \tag{1}$$

By filtering the input signal $x(n)$ using a bank of K bandpass filters, $h_k(n)$, the signal is divided into K subbands, each denoted by $x_k(n)$ where K is the subband index. This filtering operation can be written in time domain as

$$x_k(n) = x(n) * h_k(n) \tag{2}$$

Where $*$ is the convolution operator, In the ideal case, the original signal can be described as

$$x(n) = \sum_{k=0}^{K-1} x_k(n) = \sum_{k=0}^{K-1} s_k(n) + w_k(n) \tag{3}$$

Where $s_k(n)$ is the speech part subband k and $w_k(n)$ is the noise part subband k . Output $y(n)$ is formed by

$$y(n) = \sum_{k=0}^{K-1} G_k(n)x_k(n) \tag{4}$$

Where $G_k(n)$ is a weighing function that amplifies the band gain during the speech activity. Since $G_k(n)$ introduces the gain to each subband.

Now we have to find the gain function that weights the input signal subbands using the ratio between $s_k(n)$ and $w_k(n)$ i.e. a short term noise estimate. The gain function in each subband is found by using the ratio of a short term exponential magnitude average,

$A_{x,k}(n)$ based on $|x_k(n)|$, and an estimate of the noise floor level $A_{x,k}(n)$. The short term average in subband k , $A_{x,k}(n)$, is calculated as

$$A_{x,k}(n) = (1 - \alpha_k)A_{x,k}(n - 1) + \alpha_k|x_k(n)| \quad (5)$$

The suitable value for α_k can be found using the following equation

$$\alpha_k = \frac{1}{T_{s,k}F_s} \quad (6)$$

Where F_s is the sampling frequency and $T_{s,k}$ is the time constant.

2.1 Non Linear spectral Subtraction

The basics of nonlinear spectral subtraction techniques (NSS) reside in the combination of two main ideas [2]:

- The noise-improvement model is used which is obtained in the course of a speech pause.
- The nonlinear subtraction is used when a frequency-dependent signal-to-noise ration (SNR) is obtained. This means that in spectral subtraction a minimal subtraction factor is high SNR is used in turn.

3 Hidden Markov Model

As mentioned above the technique used to implement speech recognition is Hidden Markov Model (HMM). The HMM is used to represent the utterance of the word and to calculate the probability of that the model which created the sequence of vectors [4, 12]. There are some fundamental problems in designing of HMM for the analysis of speech signal.

The present hidden Markov Model is represented by

$$\lambda = (\pi, A, B) \quad (7)$$

π = initial state distribution vector.

A = State transition probability matrix.

B = continuous observation probability density function matrix.

Given appropriate values of A, B and π , the HMM can be used as a generator to give an observation sequence

$$O = O_1 O_2 \dots \dots O_T \quad (8)$$

(Where each observation O_t is one of the symbols from the observation symbol V and T is the number of observation in the sequence) as follows:

- i) Choose an initial state $q_1 = S_i$ according to the initial state distribution π .
- ii) Set $t = 1$
- iii) Choose $O_t = v_k$ according to the symbol probability distribution in state S_i .
- iv) Transit to a new state $q_{t+1} = S_j$ according to the state transition probability distribution for state S_i .
- v) Set $t = t + 1$ (return to step3) if $t < T$; otherwise terminate the procedure.

The above procedure can be used as both a generator of observations, and as a model for how a given observation sequence was generated by an appropriate HMM.

After re estimate the parameters, the model is represented with the following denotation

$$\lambda = (A, \mu, \Sigma) \quad (9)$$

The model is saved to represent that specific observation sequences, i.e. an isolated word. The basic theoretical strength of the HMM is that it combines modeling of stationary stochastic processes (for the short-time spectra) and the temporal relationship among the processes (via a Markov chain) together in a well-defined probability space. This combination allows us to study these two separate aspects of modeling a dynamic process (like speech) using one consistent framework. Another attractive feature of HMM's comes from the fact that it is relatively easy and straightforward to train a model from a given set of labeled training data (one or more sequences of observations).

3.1 Linear Predictive Coding Analysis

One way to obtain observation vectors O from speech samples s is to perform a front end spectral analysis. The type of spectral analysis that is often used (and the one we will describe here) is called linear predictive coding (LPC) [5-9].The block diagram shown in figure.3 clearly explains the LPC analysis technique.

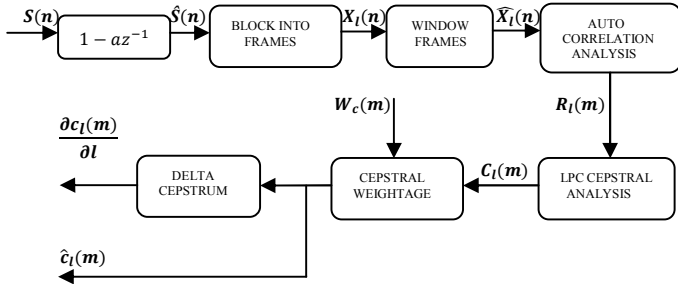


Fig.3 Block diagram showing Linear Predictive Coding Analysis

The steps in the processing are as follows:

i) Preemphasis: The digitized speech signal is processed by a first-order digital network in order to spectrally flatten the signal.

$$\hat{s}(n) = s(n) - \alpha s(n-1) \quad (10)$$

ii) Blocking into Frames: Sections of N_A consecutive speech samples are used as a single frame. Consecutive frames are spaced M_A samples apart.

$$X_l(n) = \hat{s}(ml + n), 0 \leq n \leq N-1; 0 \leq l \leq L-1 \quad (11)$$

iii) Frame Windowing: Each frame multiplied by an N_A sample window (Hamming Window) $w(n)$ so as to minimize the adverse effects of chopping an N_A samples section out of the running speech signal.

$$\tilde{X}_l(n) = x_l(n) \cdot w(n), 0 \leq n \leq N-1 \quad (12)$$

iv) Auto Correlation Analysis: Each windowed set of speech sample is autocorrelated to give a set of $(p+1)$ coefficients, where p is order of the desired LPC analysis.

$$R_l(m) = \sum_{n=0}^{N-m} \tilde{X}_l(n) \tilde{X}_l(n+m), 0 \leq m \leq p \quad (13)$$

v) LPC/Cepstral Analysis: A Vector of LPC coefficients is computed from the autocorrelation vector using a Levinson or a Durbin recursion method. An LPC derived cepstral vector is then computed up to the Q^{th} component.

$$a_l(m) = \text{LPC COEFFICIENTS}, 0 \leq m \leq p \quad (14)$$

vi) Cepstral Weighting: The Q -coefficient cepstral vector $c_l(m)$ at time frame l is weighted by a window $W_c(m)$ [5, 6]

$$W_c(m) = 1 + [(Q/2)(\sin(\pi m/Q))], 1 \leq m \leq Q \quad (15)$$

To give

$$\hat{c}_l(m) = c_l(m) \cdot W_c(m) \quad (16)$$

$$c_l(m) = \text{CEPSTRAL COEFFICIENT}, 1 \leq m \leq Q \quad (17)$$

vii) Delta Cepstrum: The time derivative of the sequence of weighted cepstral vectors is approximated by a first-order orthogonal polynomial over a finite length window of frames centered around the current vector [8, 9]

$$\Delta \hat{c}_l(m) = [\sum_{k=-K}^K k \hat{c}_{l-k}(m)]. G \quad (18)$$

where G is the gain term to make the variance of $\hat{c}_l(m)$ and $\Delta \hat{c}_l(m)$ equal.

$$Q_l(m) = \{\hat{c}_l(m), \Delta \hat{c}_l(m)\} \quad (19)$$

$$\Delta \hat{c}_l(m) = \partial \hat{c}_l(m) / \partial l, 1 \leq m \leq Q \quad (20)$$

3.1 Vector Quantization and Recognition

To use HMM with discrete observation symbol density, a Vector Quantizer (VQ) is required to map each continuous observation vector in to a discrete code book index. The major issue in VQ is the design of an appropriate codebook for quantization. The procedure basically partitions the training vector in to M disjoint sets. The distortion steadily decreases as M increases. Hence HMM with codebook size of from $M=32$ to 256 vectors has been used in speech recognition experiments using HMMs [9, 10].

During the training phase the system trains the HMM for each digit in the vocabulary [11]. The same weighted cepstrum matrices for various samples and digits are compared with the code book and their corresponding nearest codebook vector indices is sent to the Baum-Welch algorithm to train a model for the input index sequence. After training we have three models for each digit that corresponds to the three samples in our vocabulary set. Then we find the average of A, B and π matrices over the samples to generalize the models.

During the recognition the input speech sample is preprocessed to extract the feature vector. Then, the nearest codebook vector index for each frame is sent to the digit models. The system chooses the model that has the maximum probability of a match.

4 Results and Discussion

Several experiments are conducted commonly to improve the speech recognition. The analysis mainly focused on enhances the quality of the recognition with different noises at different SNR's values. Speech enhancement algorithm using adaptive gain equalization gives better result in different environmental conditions. The speech enhancement algorithm produces enhanced quality of speech recognition at different SNR values which are shown in Table 1-10.

Table 1 Performance of Speech Enhancement Algorithm for digit '0'

Noise	0dB	5dB	10dB	15dB
AIRPORT	71	55	58	68
EXHIBITION	0	14	92	15
TRAIN	0	20	50	25
RESTAURANT	0	1	2	42
STREET	92	86	73	93
BABBLE	0	30	49	33
STATION	3	6	8	66
CAR	35	6	28	36

Table 2 Performance of Speech Enhancement Algorithm for digit '1'

Noise	0dB	5dB	10dB	15dB
AIRPORT	18	22	39	37
EXHIBITION	57	60	47	64
TRAIN	11	52	34	57
RESTAURANT	23	35	51	51
STREET	26	41	49	49
BABBLE	21	46	49	35
STATION	54	44	51	50
CAR	25	37	47	32

Table 3 Performance of Speech Enhancement Algorithm for digit '2'

Noise	0dB	5dB	10dB	15dB
AIRPORT	49	36	30	35
EXHIBITION	13	35	52	57
TRAIN	10	37	52	54
RESTAURANT	27	33	46	34
STREET	60	35	57	35
BABBLE	19	34	44	13
STATION	76	60	62	48
CAR	36	46	47	30

Table 4 Performance of Speech Enhancement Algorithm for digit '3'

Noise	0dB	5dB	10dB	15dB
AIRPORT	37	52	48	43
EXHIBITION	9	37	52	54
TRAIN	57	45	45	43
RESTAURANT	28	31	35	37
STREET	46	41	51	42
BABBLE	33	42	40	40
STATION	33	48	42	55
CAR	42	31	36	37

Table 5 Performance of Speech Enhancement Algorithm for digit '4'

Noise	0dB	5dB	10dB	15dB
AIRPORT	56	41	52	45
EXHIBITION	22	65	65	51
TRAIN	24	77	82	93
RESTAURANT	55	62	63	66
STREET	83	72	60	81
BABBLE	44	54	59	68
STATION	72	89	75	60
CAR	64	39	80	41

Table 6 Performance of Speech Enhancement Algorithm for digit '5'

Noise	0dB	5dB	10dB	15dB
AIRPORT	18	22	28	31
EXHIBITION	37	26	24	23
TRAIN	31	33	25	38
RESTAURANT	33	33	40	35
STREET	25	28	37	37
BABBLE	27	21	21	17
STATION	35	37	36	32
CAR	32	48	14	33

Table 9 Performance of Speech Enhancement Algorithm for digit '8'

Noise	0dB	5dB	10dB	15dB
AIRPORT	35	21	38	20
EXHIBITION	25	20	42	53
TRAIN	34	42	48	48
RESTAURANT	16	24	19	42
STREET	60	40	31	63
BABBLE	21	30	6	5
STATION	45	48	31	60
CAR	16	28	10	21

Table 7 Performance of Speech Enhancement Algorithm for digit '6'

Noise	0dB	5dB	10dB	15dB
AIRPORT	14	23	9	9
EXHIBITION	7	18	15	7
TRAIN	4	9	20	24
RESTAURANT	9	11	12	19
STREET	11	26	25	25
BABBLE	14	8	11	21
STATION	20	21	21	28
CAR	17	7	3	13

Table 10 Performance of Speech Enhancement Algorithm for digit '9'

Noise	0dB	5dB	10dB	15dB
AIRPORT	3	2	6	9
EXHIBITION	1	16	10	19
TRAIN	11	19	13	20
RESTAURANT	12	14	16	18
STREET	15	8	5	15
BABBLE	1	5	5	11
STATION	12	15	13	18
CAR	8	1	14	18

Table 8 Performance of Speech Enhancement Algorithm for digit '7'

Noise	0dB	5dB	10dB	15dB
AIRPORT	31	35	57	35
EXHIBITION	18	39	53	34
TRAIN	22	33	43	56
RESTAURANT	45	56	61	65
STREET	52	46	59	51
BABBLE	53	42	23	34
STATION	34	54	48	63
CAR	33	26	50	32

5 Conclusion

The experimental results which are shown in Table 1-10 clearly prove the Speech Enhancement Algorithm works for different noise sources at different SNR values. For number '0' the AGE algorithm works better for airport and street noises. For number '1' it performs well for exhibition and station noises. For numbers '2','4',and'7' the AGE performs better recognition for street and station noises. For numbers '5' and '6' the SEA works well for station and restaurant noises. For number '8' the performance of SEA is good for restaurant and street noises. For number '9' the enhanced recognition occurs for train and restaurant noises. Hence the speech enhancement algorithm works better for different noises at different environmental noises.

References:

- [1] Ramirez, J.C.Segura, C.Benitez, A.de la Torre, A.Rubio, "Voice activity detection with noise reduction and long-term spectra divergence estimation" *IEEE International Conference on Acoustics, speech and Signal Processing* pp.1093-6, Volume 2, Issue, 17-21 May 2004.
- [2] J.Poruba, "Speech Enhancement based on non linear Spectral subtraction," *Proceeding of the Fourth IEEE International Conference on devices, Circuit and System*, pp T031-1-T031-4, April 2002.
- [3] Nils Westerlund, Mattia Dahl, Ingvar Claesson, "Speech Enhancement using on adaptive gain equalizer with frequency dependent parameter settings", *Proceeding of the IEEE* vol.7 ,pp. 3718-3722, 2004.
- [4] Lawrence R.Rabiner, "A tutorial on Hidden Markov Model and selected applications in speech recognition", *Proceedings of the IEEE*, vol.77, no.2, pp. 172-175, February 1989.
- [5] J. Makhoul, "Linear Prediction a Tutorial view," *Proceedings of the IEEE*, Vol. 63, No. 4, pp. 215-230 April 1975.
- [6] J.D.Markel and A.H.Gray Jr., "Linear Prediction of Speech", *Newyork, NY: springer-Verilag*, pp.71-75 1976.
- [7] Y.Tokhura, "Aweighted cepstral distance measure for speech recognition," *IEEE Trans. Acoust speech signal processing*, vol.ASSP-35, no.10, pp.1414-1422, October 1987.
- [8] B.H.Juang, L.R.Rabiner and J.G.Wilpon, "On the Use of Bandpass filtering in speech recognition " *IEEETrans. Acoust Speech signal processing*, vol.ASSP-35, no.7, pp947-954, July 1987.
- [9] J. Makhoul, S.Roucos and H.Gish, "Vector Quantization In Speech Coding", *Proc.IEEE*.vol.73,no.11,pp.1551-1558, November 1985.
- [10] L.R.Rabiner, S.E.Levinson and M.M.Sondhi, "On The Application Of Vector Quantization And Hidden Markov Models To Speaker-Independent Isolated Word Recognition", *Bell Syst.Tech.J.*, vol.62,no.4,pp.1075-1105, April 1983.
- [11] M.T.Balamuragan and M.Balaji, "SOPC- Based Speech to Text Conversion Embedded processors design contest-outstanding", pp83-108, 2006.
- [12] Y. Ephraim and N. Merhav, "Hidden Markov Processes" *IEEE Trans. Inform. Theory*, vol. 48, pp. 1518-1569, June 2002.
- [13] Yi Hu, Philipos C. Loizou, "Subjective comparison and evaluation of speech enhancement algorithms", *Speech Communication* 49, pp.588-601, Decmber 2006.
- [14] Sundarrajan Rangachari, Philipos C. Loizou, "A noise-estimation algorithm for highly non-stationary environments" *Speech Communication* 48, pp.220-231, August 2005.