# LINEAR

# PREDICTIVE

# CODING

# OF SPEECH

# CONTENTS

# CONTENTS

# CHAPTER 1

## INTRODUCTION

The principal means of human communication is speech. Speech can be described an acoustic waveform that conveys information from speaker to listener. In modern era of communication, the systems often need to transmit, store, manipulate, recognize and create speech. Some even need to recognize identity of the speaker. For all these tasks, the speech signal is usually represented in its digital form. However, uncompressed digital form of speech needs large amount of storage space and transmission bandwidth. Consequently, compression in digital form is more versatile than analog, providing better compression ratio, consistent quality and security.

Digital speech coding or speech compression is concerned with obtaining compact digital representation of voice signal. The objective of speech coders is to represent speech signal with fewer bits while maintaining perceptual quality with reasonable computational complexity on both analysis and synthesis side. To achieve high quality speech at low bit rate, coding algorithms apply sophisticated algorithms to reduce the redundancies and perceptually irrelevant information.

Over past few decades, a variety of speech coding techniques have been proposed, analyzed and implemented. This chapter discusses the speech coders briefly along with their two basic types, i.e. waveform coders and parametric (model-based) coders. Further it talks about attributes of speech coders and their significance from designing and implementation point-of-view.

## 1.1 OVERVIEW OF SPEECH CODERS

Traditionally, speech coders are divided into two classes; waveform coders and parametric coders (also known as vocoders or source coders). Typically, waveform coders operate at higher bit rates (lower compression) and produce higher quality speech. Parametric or model based coders operate at lower bit rates but produce synthetic quality speech. Recently, a new class called hybrid coders is introduced which uses techniques of both waveform and parametric codes to achieve higher quality speech at medium bit rates.

### 1.1.1 WAVEFORM CODERS

The waveform coders almost recover the original signal back on synthesis side. They are designed to be independent of signal type, thus can be used with wide variety of signals including the ones which are other than speech. Their performance does not depend on the type of signal. Generally, they are low complexity coders producing high quality at bit rates more than 16 kbps. Waveform coding can be carried out in time as well as frequency domain.

Pulse Code Modulation (PCM) is a simplest type of coder, using fixed uniform quantizer for each sample of signal. Considering logarithmic sensitivity of human auditory system; a non-uniform quantizer yield better quality of speech than uniform quantizer at same bit-rate. Thus, CCIT standardized G.711 in 1972, a 64 kbps logarithmic PCM toll quality speech coder for telephony.

At a cost of more complexity, the toll quality can be obtained at much lower bit-rates. Differential Pulse Code Modulation (DPCM) and Adaptive Differential Pulse Code Modulation (ADPCM) reduce variance of a signal using suitable filter and the low-variance signal is then quantized with fewer bits. In ADPCM, a filter and quantizer both are adapted depending on the signal. The coders G.726 and G.727 are examples of ADPCM based coders with bit rates of 40, 32, 24 and 16 kbps.

Sub-band Coders are used for wide-band applications such as audio, which employ a series of filters (filter-bank) of band pass filter to divide the signal in frequency domain. Each sub-band signal is then encoded separately. At synthesis side, signals in all frequency bands are decoded and summed up to reproduce the signal. The advantage of sub-band coding is that the quantization noise produced in a sub-band is confined to that band. Standard G.722 is an audio codec which encodes wideband audio signal for transmission at 48, 56 and 64 kbps.

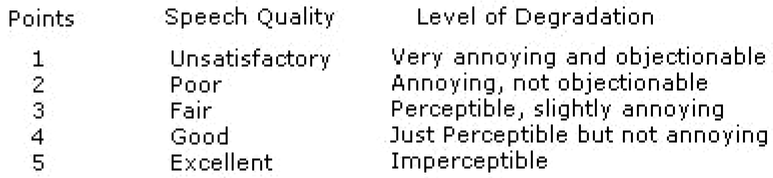
Another way is to convert the signal from time domain to a suitable domain such that the energy is concentrated in fewer coefficients, which are then quantized efficiently. An example is Adaptive Transform Coding (ATC) where the quantizer is adopted according to the characteristics of a signal. Several transforms like Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT) can also be used.

### 1.1.2 PARAMETRIC CODERS

Parametric coders try to imitate the process of human speech production using a suitable pre-defined model. The model parameters vary with signal and are used on receiver side for synthesis. The model parameters are sent instead of actual speech signal so as to achieve the compression. Almost all coders used Linear Predictive Analysis (LPC) for model estimation. The most commonly used model for speech is Auto-Regressive Model (of the form of IIR filter or All-Pole filter) with a suitable order. The reasons to select this model are well explained in [1].

For reproduction of speech, model needs a suitable excitation signal. Speech quality in source coders depends purely on the way model is excited no matter how much accurate the model is. There are numerous techniques available to decide and model an excitation signal. The coders are mainly distinguished based on the way they deal with excitation signal, sometimes also called residual or error signal. The different techniques of LPC based coders are LPC-10, MPLPC, CELP, MELP, RELP, VSELP, VELP and many more. Few are explained in consequent chapters. All these coders provide audible or more than audible quality speech at bit rates below 16 kbps; the reason why they are preferred over waveform coders. Model based coders can not reproduce original speech and are susceptible to noise if nature of the signal changes for which the model is pre-defined.

Coders like LPC-10 provide bit rates of 2.4 kbps which use model parameters along with pitch as excitation parameter. The coders with bit rates as low as 100 bps have also been designed using techniques of speech recognition and text-to-speech synthesis. More information about very low bit rate coders is given in [1].



| Points | Speech Quality | Level of Degradation |
|---|---|---|
| 1 | Unsatisfactory | Very annoying and objectionable |
| 2 | Poor | Annoying, not objectionable |
| 3 | Fair | Perceptible, slightly annoying |
| 4 | Good | Just Perceptible but not annoying |
| 5 | Excellent | Imperceptible |

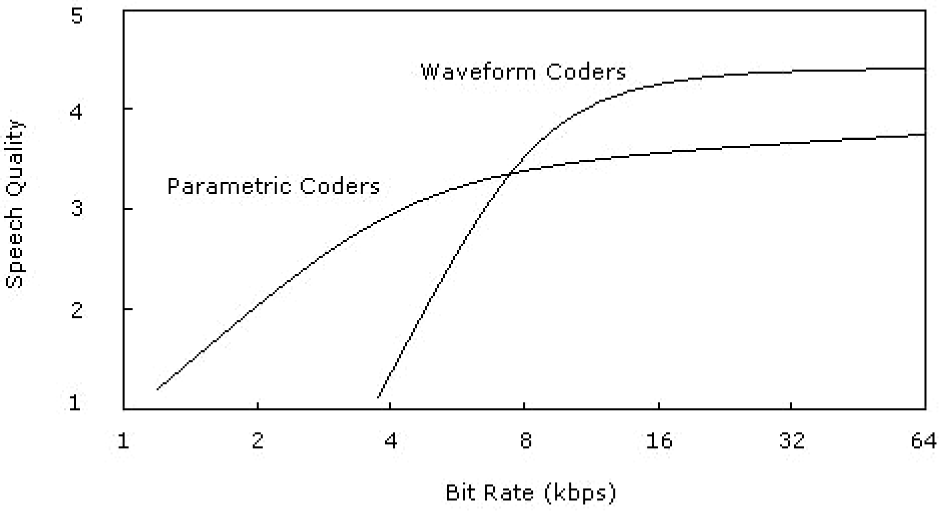


Figure 1.1 Quality of Speech Coders Compared to Bit-Rate

Hybrid coders attempt to fill the gap between waveform and parametric coders. Hybrid coders try to achieve natural quality at bit rates anything below 16 kbps for speech. However these coders are computationally more complex and need higher processing speeds.

## 1.2 ATTRIBUTES OF SPEECH CODERS

The following criterias are considered while designing or selecting a speech coding system, for that matter any compression algorithm.

- QUALITY: The quality of speech can be evaluated using extensive testing with human subjects. The quality is rated depending on the degradation in reconstructed signal. The following are the categories to describe the quality of speech coders. (1) Commentary or Broadcast quality describes wide-band speech with no perceptible degradation. (2) Toll or wireline quality speech refers to the type of speech required over PSTN. (3) Communication quality speech is completely intelligible but with noticeable distortion. (4) Synthetic speech quality is characterized by its 'robot-like' sound, lacking speaker identification ability and (5) objectionable quality with annoying noise and not suitable for implementation.

- BIT-RATE: This attribute decides the channel bandwidth required to transmit the encoded speech signal. Generally, a choice is made between fixed-rate and variable-rate coders. The variable bit-rate coders are suited in applications such as mobile communication where the channel bandwidth is dynamic due to variations in traffic. In applications, where the users are allocated dedicated channels, fixed-rate coders are more suitable. Bit-rate is represented in terms of kbps (kilo-bits per second).

- COMPLEXITY: This is evaluated by the memory requirements and computational power required for implementation. For selecting appropriate hardware, one needs to judge the coder complexity. In virtually all applications, such as live broadcasting of speech, real-time coders and decoders are required. The real-time performance demands faster processing and advanced hardware. Complexity is measured indirectly in terms of Million Instructions per Second

(MIPS) or Floating Point Operations per Second (FLOPS), the same used for evaluation of DSP processors.

- DELAY: The total delay of speech coding system is the time taken by speech to reach to listener from talker. It involves algorithmic delay, processing delay and transmission delay. Algorithmic delay is the total amount of buffering or look-ahead used by the coding algorithm. Processing delay describes time taken for processing speech. The transmission delay is induced by communication system to send signal form one point to the other. Ideally, the delay should not exceed 300 ms for practical implementation.

## 1.3 OBJECTIVE OF OUR PROJECT

The various types of speech coding algorithms and their attributes have been explained. The waveform coder as mentioned above cannot provide bit-rates below 16 kbps. On the other hand, demands for lower bit-rate in modern communication systems have made Parametric or Model-based coders more popular. The most common of all is Linear Predictive Coding (LPC).

The aim of our project is to implement LPC based speech coder which will provide reasonable quality speech at a bit-rate of around 8 kbps. We are using Matlab for simulating the algorithms before implementing those in C language. If time permits, we would like to implement the algorithm or a part of it on a DSP processor.

## 1.4 ORGANIZATION OF THE REPORT

Before starting with processing a speech, it is important to understand its nature, since almost all algorithms employ the characteristics of speech itself to achieve compression. Thus next chapter describes speech production and perception, along with their time and frequency domain characteristics.

Third chapter describes complete linear prediction analysis and the algorithms involved, from theory as well as implementation point-of-view by including extensive mathematical analysis. A section of it describes various techniques based on Linear Prediction Analysis which differ in a way they model or produce excitation signal.

Fourth chapter includes the details of MP-LPC (Multi-Pulse Linear Prediction Coding) implementation which we have accomplished in C. It describes both analysis and synthesis techniques with their performance evaluation based on the criterias mentioned in previous section. It also compares the coder we have developed with the standard coders and their attributes for evaluation purpose.

# CHAPTER 2

## SPEECH SIGNAL

Speech coding algorithms can be made more efficient by removing the irrelevant information from speech signal. In order to design a speech coding algorithm, it is necessary to understand the human speech production, its perception by human ear, so that the redundancies can be removed. Also the nature of speech in time domain is important to understand its behaviour and relation with different kinds of voices, especially for time-domain coders. Frequency domain behaviour is equally important. The subsequent sections talk about these things in detail.

### 1.1 HUMAN SPEECH PRODUCTION

Regardless of the language spoken, all people use relatively same anatomy to produce speech. Thus the nature of speech signal is independent of language and is only decided by the laws of Physics. Acoustically, speech can be described as fluctuations in pressure propagating through air which are sensed by human ear and decoded by brain for perception.

The speech production process can be summarized as air being pushed by lungs through the vocal tract involving various vocal organs and out through the mouth to generate speech. The vocal tract system with vocal organs is shown in figure 2.1. In more details, the lung pressure in larynx forces the air flow through the tensioned vocal cords, as a result, producing vibrations. Leaving the oral and nasal cavities, air flows through the mouth and nose. The fundamental frequency (described as pitch) of vibration of cords is an inherent property speech, and depends on mass and tension of vocal cords. The average fundamental frequency is about 130, 220 and 300 Hz for men, women and children respectively.

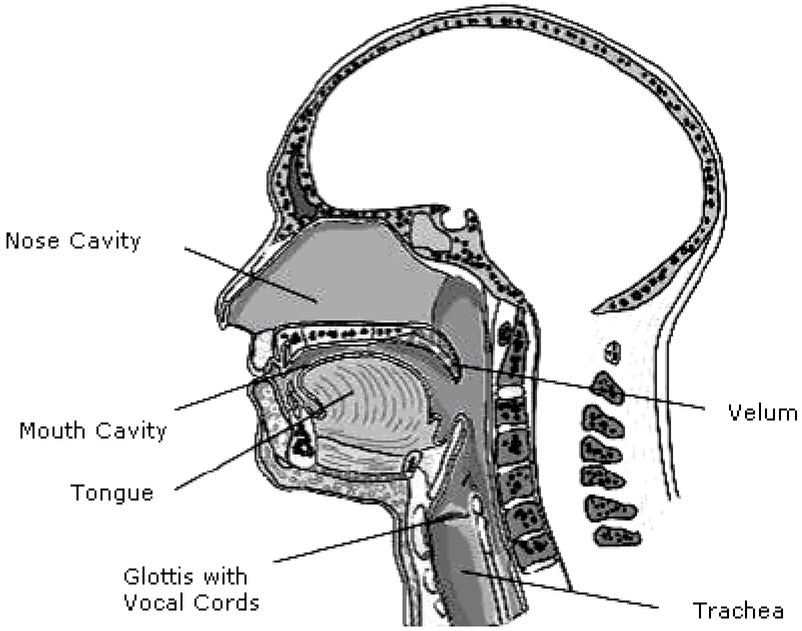


Figure 2.1 Human Speech Production System

In this context, lungs can be thought of as a source of energy and vocal tract as a filter that shapes the source signal to produce various types of sounds that make up speech. The model based coders and text-to-speech synthesizers try to model vocal tract system as a time-varying digital filter to produce speech. This model when excited by suitable excitation signal produces speech.

## 2.2 TIME AND FREQUENCY DOMAIN REPRESENTATION OF SPEECH

Speech in general can be analyzed from different points of view. Phonetics is a branch of acoustic speech production, perception and its acoustic analysis, which assumes speech properties to be independent of the language spoken. While human can produce large number of sounds, each language has a set of linguistic units called phonemes. It can be described as a smallest meaningful contrastive unit in the phonology of a language. Each word is made up by series of phonemes. Most languages have 20 to 40 phonemes providing an alphabet of sounds which is the unique description of the words in a given language. There are two main categories of phonemes, voiced and unvoiced, that are considered by few LPC based coders while analyzing and synthesizing speech signal.

Voiced sounds are usually vowels and often have high average energy levels and very distinct resonant frequencies. Voiced sounds are generated by air from the lungs being forced over the vocal cords. As a result the vocal cords vibrate in a somewhat periodic pattern that produces a series of air pulses called glottal pulses. The rate at which the vocal cords vibrate determines pitch of the sound produced. These air pulses that are created by the vibrations finally pass along the rest of the vocal tract where some frequencies resonate. It is generally known that women and children have higher pitched voices than men as a result of a faster rate of vibration during the production of voiced sounds. It is therefore important to comprise pitch period in the analysis and synthesis of speech.

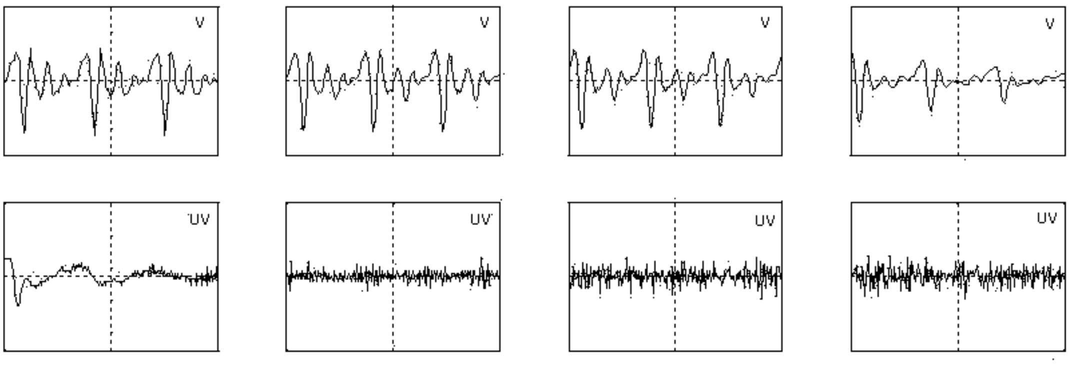


Figure 2.2 Various Voiced and Un-voiced Speech Segments

Unvoiced sounds are usually consonants and generally have less energy and higher frequencies then voiced sounds. The production of unvoiced sound involves air being forced through the vocal tract in a turbulent flow. During this process the vocal cords do not vibrate, instead, they stay open until the sound is produced. Pitch is not an important attribute of unvoiced speech since there is no vibration of the vocal cords and no glottal pulses produced. The categorization of sounds as voiced or unvoiced is an important consideration in the analysis and synthesis process.

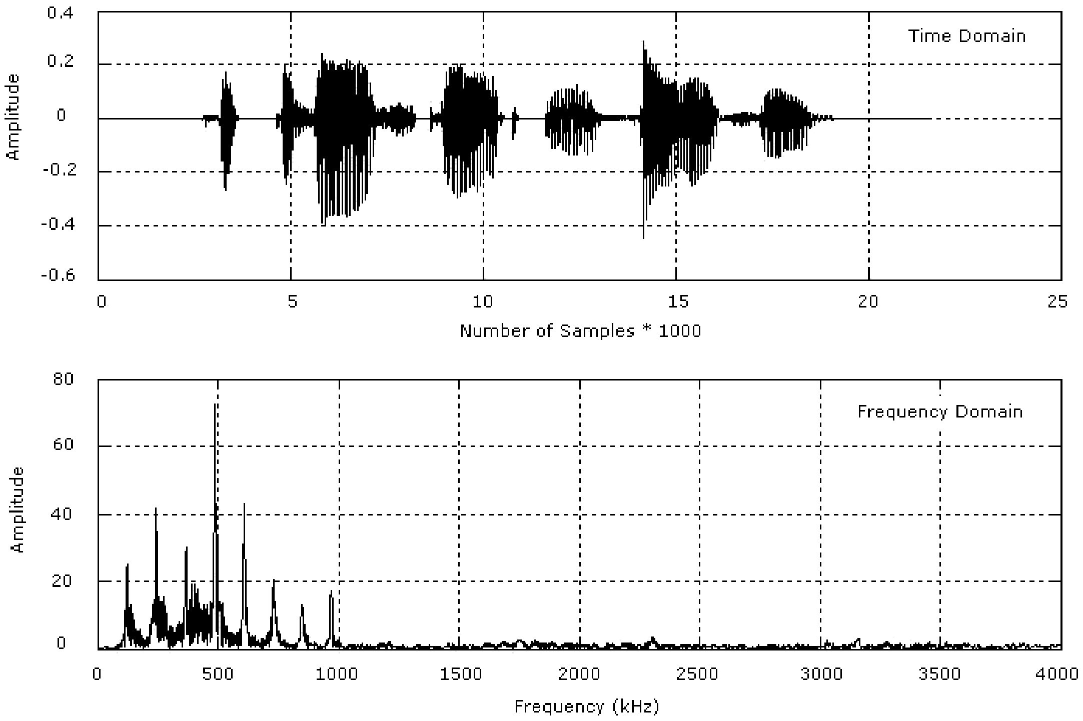


Figure 2.3 A speech Signal in Time and Frequency Domain

A pure speech signal contains dominant frequency components in range of 50-3500 Hz. That means, the removal of components of higher frequencies above 3500 Hz does not affect the speech quality seriously. Even if the frequencies above 1 KHz are removed, around 70% of the speech can still be recognized.

A speech is often described as a non-stationary random or stochastic process. For statistical analysis, speech is assumed to be stationary over short period of time, usually 10-30 ms. A short term speech signal and its magnitude response is shown in figure 2.4. As seen, the first formant $F_1$ is due to the fundamental frequency or pitch of voiced segment and the following $F_2$, $F_3$ and $F_4$ are due to the resonant frequencies of vocal tract, called formants. Parametric coders even analyze the formants in frequency domain for modelling the speech signal.

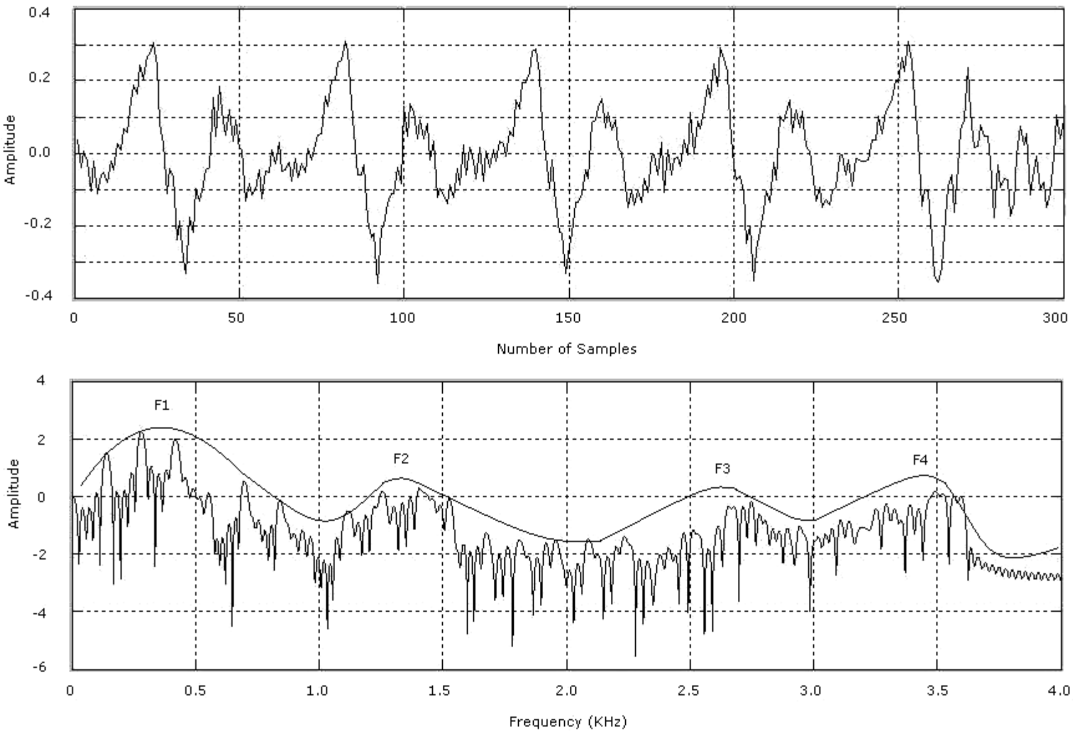


Figure 2.4 Short-term Amplitude Response of Voiced Speech Segment

## 2.3 SPEECH PERCEPTION

One of the major performance measures of speech and audio coding is determined by how well the speech is perceived. If the redundancies in speech are spotted and if the perceptual properties of ear are exploited properly, good audible performance can be achieved at lower bit-rates.

The human hearing system acts as a filter banks and is sensitive only to the frequencies from 20 Hz to 16000 Hz. This can be verified by writing a simple C code using a function 'void sound(unsigned frequency)', which uses internal PC speakers. Perceptual experiments have shown that range of 200-3700 Hz is important to speech intelligibility. For this reason, a speech signal is sampled at least at 8000 Hz. Also there is a limit to a sensitivity of human ear. If the sound is too weak, it will not be detected. This lower limit is called threshold of audibility. Threshold varies with frequency and decides how much noise can be added at each frequency so that it is not perceptible.

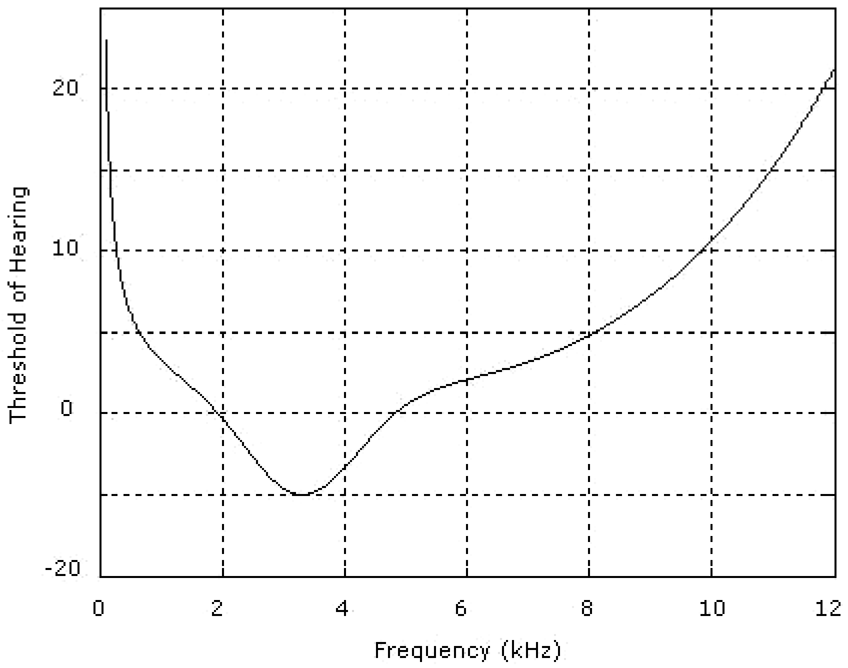


Figure 2.5 Relation between Threshold of Audibility and Frequency

The threshold of audibility is well approximated by a non-linear function, with respect to a young listener with acute hearing and quite surrounding.

$$T(f) = 3.64 * f^{-0.8} - 6.5 * e^{-0.6*SQR(f-3.3)} + 10^{-3} * f^{4}$$

where, T(f) - dB

f - mHz

The above relation between threshold of audibility and frequency is an important consideration in sub-band coders and perceptual coders, for applications such as audio and high quality speech coding.

# CHAPTER 3

## LINEAR PREDICTION OF SPEECH

As discussed in previous chapters, linear prediction analysis is a parametric coding technique which assumes a suitable model of a vocal tract system to synthesize speech. The model is a time-varying digital filter changing its parameters as per change in a segment of speech signal. Selection of proper model is a crucial task since in model-based techniques; system performance purely depends on the accuracy of the model.

A linear predictor, as name suggests, predicts the current value from previous values of speech samples on analysis side. It is called linear simply because, the equation is linear mathematically (i.e. no square term of dependant variable). Though non-linear predictors have also been derived and implemented, linear ones are preferred because of their simplicity. The sum is weighted using filter coefficients which needs to be calculated for a group of samples. The order of predictor filter is decided by the number of previous samples considered. For analysis, a speech segment needs to be buffered. Generally, the size varies over 10-30 ms, since over this short segment, the samples are highly correlated. Mathematically, predicted value is given by,

$$x'(n) = a_1 . x(n-1) + a_2 . x(n-2) + \ldots + a_p . x(n-p) \qquad \text{- Eq. 3.1}$$

where, $a_1, a_2, \ldots, a_p$ are the filter coefficients

and $p$ is the order of the filter

## 3.1 FIRST ORDER LINEAR PREDICTOR

To understand how a linear predictor modifies the signal, lets take a simplified case of 1st order predictor as shown in following figure. Let x(n) be the discrete time-invariant input speech signal.

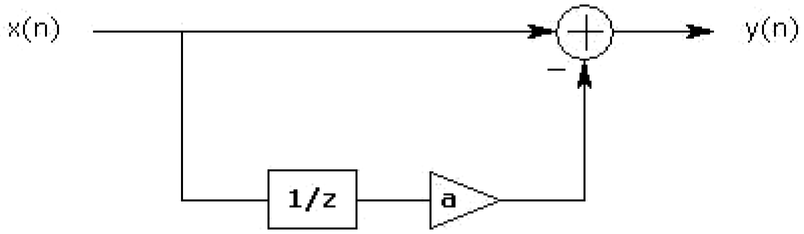


Figure 3.1 1st Order Predictor

Thus,

$$y(n) = x(n) - a \,.\, x(n-1) \qquad \text{- Eq. 3.2}$$

then the variance i.e. power of the signal can be expressed as (assuming x(n) and a real),

$$E\{ y(n)^2 \} = E\{ [\, x(n) - a \,.\, x(n-1) \,]^2 \} \qquad \text{- Eq. 3.3}$$

$$R_{yy}(0) = R_{xx}(0) \,[\, 1 - 2a \,.\, r_{xx}(1) + a^2 \,] \qquad \text{- Eq. 3.4}$$

where, $R_{xx}(n)$, $R_{yy}(n)$ are the auto-correlation functions associated with x and y respectively; $r_{xx}(n)$ represent normalized values.

The energy or variance represented by $R_{yy}(0)$ is minimized if $r_{xx}(1)$ equal to a. Thus from eq. 3.4, we get,

$$\sigma_y^2 = (1 - a^2) \,.\, \sigma_x^2 \qquad \text{- Eq. 3.5}$$

Clearly, if a is closer to unity, the variance of output y(n) would be much less than that of input x(n). Thus, the quantization of y(n) required fewer bits keeping the quantization noise power same. To ensure that the filter is stable, it is necessary that |a| < 1. This is quite obvious since coefficient $r_{xx}(1)$ is always less than one. The details of relation between variance and number of bits needed for quantization are given in [2].

For practical implementation, predictor in figure 3.1 needs to be represented by the closed-loop scheme as shown below. The scheme generating lower word-length is called encoder while the one which reconstructs the signal, is called decoder.

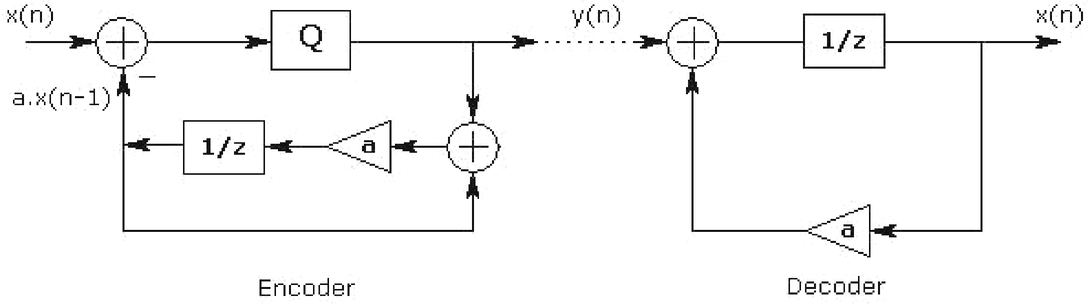


Figure 3.2 First Order Encoder and Decoder

The figure 3.3 shows random signals with 1st order prediction applied. Dotted signal shows y(n), the output of predictor. It clearly shows that the variance (or range) of the signal is reduced so that it can be quantized using lesser amount of bits.

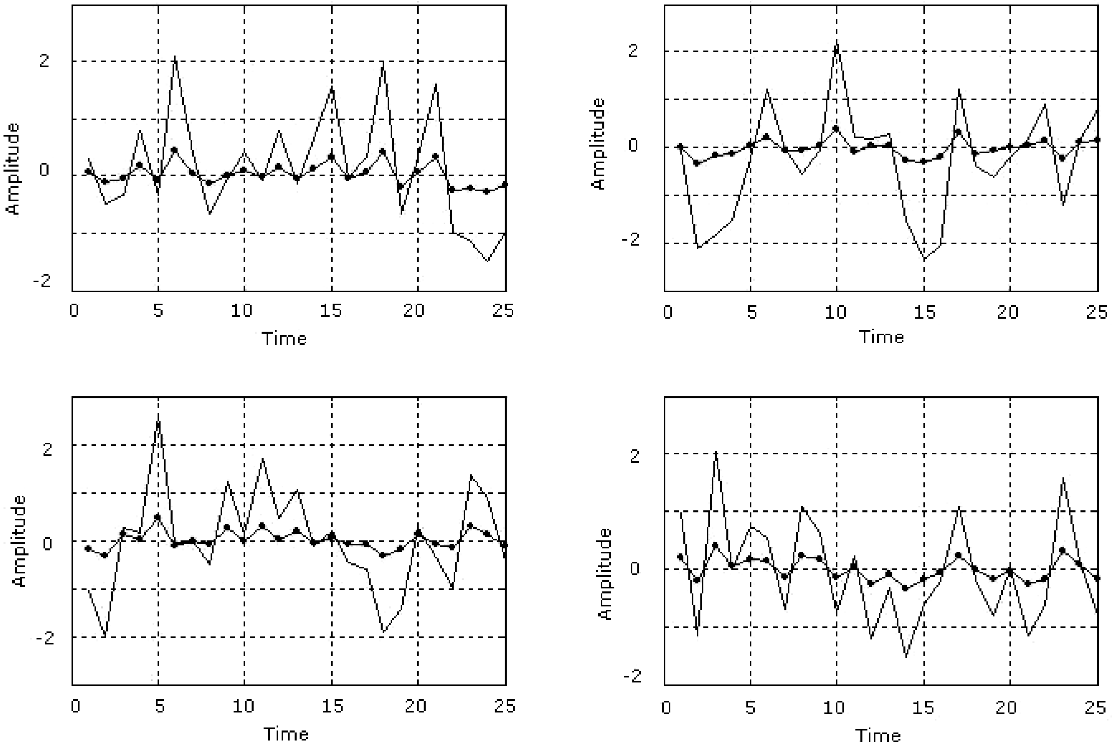


Figure 3.3 Various Random Signals filtered with 1st order predictor filter

## 3.2 LINEAR PREDICTION ANALYSIS

In analysis phase, we need to find the appropriate filter coefficient vector which is capable of predicting current value of signal from the previous ones. Consider, a linear predictor of order p which forms prediction of the value x'(n), by weighted linear combination of past values i.e. x(n-1), x(n-2), … , x(n-p). Hence, the predicted value is given by,

$$x'(n) = \sum_{k=1}^{p} - a_P(k) \,.\, x(n - k) \qquad \text{- Eq. 3.6}$$

where, $-a_P(i)$ are called predictor coefficients. In practice, for a random process, predictor will only be able to predict the correlated portion of the signal. Thus the non-zero error between actual and predicted value will be because of random component or a white noise. This non-zero component is called residual error and given by,

$$e(n) = x(n) - x'(n)$$

i.e.

$$e(n) = x(n) + \sum_{k=1}^{p} a_P(k) \,.\, x(n - k) \qquad \text{- Eq. 3.7}$$

We view linear prediction as being equivalent to linear filtering when predictor is embedded in linear filter. If the x(n) is the input and error signal e(n) is output, the above equation is called prediction-error filter, can be represented in transfer function form as,

$$A_P(z) = E(z) / X(z) = \sum_{k=0}^{p} a_P(i) \,.\, z^{-K} \qquad \text{- Eq. 3.8}$$

where, $a_P(0) = 1$

Thus, if a speech input x(n) is known, the corresponding residual error can be evaluated for each sample of input. This residual signal is an ideal excitation to a filter on synthesis side, which is an inverse form of the one in above equation. Various coders using linear prediction for modelling differs only in a way they deal with residual signal.

If the predictor coefficients of filter provide best all-pole model for the power density function of a random process x(n), the mean square error between actual and predicted values will be minimum. Let this mean square error (MSE) be,

$$Ep = (1/N) \sum_{n=0}^{N-1} [\, x(n) + \sum_{i=1}^{p} a_P(i) \, . \, x(n - i) \,]^2 \qquad \text{- Eq. 3.9}$$

where, N is the number of samples in x(n)

p is order of the filter

Now differentiating mean square error with respect to each coefficient $a_j$ and equating it to zero will lead to set of linear equations whose solution represents optimum set of coefficients of given x(n),

$$\delta Ep / \delta a_j = -2/N \sum_{n=0}^{N-1} [\, x(n) + \sum_{i=1}^{p} a_P(i) \, . \, x(n - i) \,] \, . \, x(n - j) \qquad \text{- Eq. 3.10}$$

by equating above equation to zero, we get,

$$\sum_{i=1}^{p} a_P(i) \, . \, R(i - j) = - R(j) \qquad \text{where, } j = 1, 2, \dots , p \qquad \text{- Eq. 3.11}$$

The minimum mean square error (MMSE) of prediction is given by,

$$E_P = R(0) + \sum_{i=1}^{p} a_P(i) \, . \, R(i) \qquad \text{- Eq. 3.12}$$

writing above equation in matrix notation,

$$\mathbf{R\,a} = -\,\mathbf{P}$$

where, **R** is a square matrix with order p-by-p

**a** and **P** are vectors of order p-by-1

$$\mathbf{a} = -\,\mathbf{R}^{-1} \, . \, \mathbf{P} \qquad \text{- Eq. 3.13}$$

The filter coefficient vector **a** can be evaluated using above expression. The main concern here is the calculation of inverse of matrix **R**, since the order of predictors is usually high; in range of 10 to 16. The usual method of evaluation is of complexity O($p^3$).

Now that we have derived equations to find coefficient vector **a**, the synthesis filter can be derived from equation 3.7 by considering x(n) as an auto-regressive process generated from input e(n). Thus in its transfer functions form, generalized all-pole synthesis filter will be,

$$H(z) = \frac{G}{1 + \sum_{k=1}^{p} a_P(k) \, . \, R(k)} \qquad \text{- Eq. 3.14}$$

Therefore, the difference equation for the input-output relationship is,

$$x(n) = G.e(n) + \sum_{k=1}^{p} a_P(k) \, . \, x(n - k) \qquad \text{- Eq. 3.15}$$

where, G is given by,

$$G = [\, R_P(0) - \sum_{k=1}^{p} a_P(k) \, . \, R_P(k) \,]^{1/2} \qquad \text{- Eq. 3.16}$$

## 3.3 ALGORITHMS TO EVALUATE LP COEFFICIENTS

This section describes the computationally efficient algorithms available to derive vector **a** in equation 3.13. These methods exploit the properties of correlation matrix **R** which is of the form,

$$\mathbf{R} = \begin{bmatrix} R(0) & R(1) & . & . & . & . & . & R(p-1) \\ R(1) & R(0) & . & . & . & . & . & R(p-2) \\ . & . & & & . & & & . \\ . & . & & & & & . & . \\ R(p-1) & R(p-2) & & . & . & . & . & R(0) \end{bmatrix}$$

As shown, the matrix **R** possess following properties:

1. Symmetric - $\mathbf{R} = \mathbf{R}^T$ (for real signal)
2. Semi-definite - $\mathbf{X}^T \mathbf{R} \mathbf{X} >= 0$ (for any $\mathbf{X} != 0$)
3. Toeplitz - $\mathbf{R}(i,j) = \mathbf{R}(i+1,j+1)$ (i.e. constant diagonals)

These properties are used for finding computationally efficient algorithms for inverting the matrix. The complexity of computation for such matrices is $O(p^2)$.

Most commonly used algorithms Levinson-Durbin Recursion and Schur Algorithm are described below.

### 3.3.1 LEVINSON-DURBIN RECURSION

Here we outline the basic steps involved in the algorithm. The key to Levinson-Durbin method which exploits Toeplitz property of the matrix, is to proceed recursively; beginning with predictor of order n=1, we increase the order recursively, to obtain solution to the subsequent higher orders using lower order solutions. From equation 3.11 and 3.12,

$$\sum_{i=0}^{p} a_P(i) \,.\, R(i - j) = \begin{cases} E_P & - \ j = 0 \\ 0 & - \ j = 1,2,\ldots,p \end{cases} \qquad \text{- Eq. 3.17}$$

where, $a_P(0) = 1$

The solution to 1st order predictor can be obtained from above equation and is given by,

$$a_1(1) \; = \; - \; R(1) \, / \, R(0) \qquad \text{- Eq. 3.18}$$

and the resulting MMSE is,

$$\begin{aligned} E_1 \; &= \; R(0) \; + \; a_1(1) \, . \, R(-1) \\ &= \; R(0) \, . \, [\, 1 - \; |\, a_1(1) \,|^2 \,] \qquad \text{- Eq. 3.19} \end{aligned}$$

As $a_P(p) = K_P$, $a_1(1)$ is the first reflection coefficient of lattice filter. For more details on lattice filters and reflection coefficients, refer [3] and [4]. Few literature, even refer to reflection coefficients as Partial Correlation coefficients or PARCOR coefficients.

To find recursive relation, the next step is to solve for coefficients $a_2(1)$, $a_2(1)$ of the second order predictor and express the solution in terms of $a_1(1)$. The two equations obtained from equation 3.17 are,

$$a_2(1) \, . \, R(0) \; + \; a_2(2) \, . \, R(-1) \; = \; - \, R(1)$$

$$a_2(1) \, . \, R(1) \; + \; a_2(2) \, . \, R(0) \; = \; - \, R(2)$$

By solving above equations using value of $a_1(1)$ in equation 3.18, we obtain the solution as,

$$a_2(2) \; = \; \frac{- \; R(2) - a_1(1) \, . \, R(1)}{R(0) \, . \, [\, 1 - \; |\, a_1(1) \,|^2 \,]}$$

Thus, from equation 3.13, we get,

$$a_2(2) \; = \; \frac{- \, R(2) - a_1(1) \, . \, R(1)}{E_1} \qquad \text{- Eq. 3.20}$$

Similarly,

$$a_2(1) \; = \; a_1(1) \; + \; a_2(2) \, . \, a_1(1) \qquad \text{- Eq. 3.21}$$

Thus, from equations 3.20 and 3.21, we can obtain coefficients of second order predictor using $a_1(1)$ and correlation values. Proceeding this way, we may express coefficients of $m^{th}$ order predictor in terms of coefficients of $(m-1)^{Th}$ predictor.

The above equations in their generalized form are represented as,

$$a_m(m) = K_m = \frac{- R(m) - \mathbf{\Gamma}_{m-1} . \mathbf{a}_{m-1}}{E_{m-1}} \qquad \text{- Eq. 3.22}$$

$$a_m(k) = a_{m-1}(k) + K_m . a_{m-1}(m - k) \qquad \text{- Eq. 3.23}$$

- for $k = 1, 2, \dots , m-1$

where, $m = 1, 2, . . . , p$

and $\mathbf{\Gamma}_{m-1} = [ R(m-1) \; R(m-2) \; . . . . \; R(1) ]$

$\mathbf{a}_{m-1} = [ a(m-1) \; a(m-2) \; . . . . \; a(1) ]^T$

Finally, the expression for MMSE of mth order with respect to MMSE of (m-1)th order is given by,

$$E_m = E_{m-1} . (1 - K_m^2) \qquad \text{- Eq. 3.24}$$

Where, $E_0 = R(0)$

Since, reflection coefficients satisfy the property that $|K_m| <= 1$, the MMSE always satisfies the following condition,

$$E_0 \geq E_1 \geq E_2 \geq . . . \geq E_p \qquad \text{- Eq. 3.25}$$

This is quite obvious since the model accuracy increase with the increase in order of the predictor decreasing the mean square predictor error. This property is useful in verification of Levinson-Durbin algorithm, whenever implemented in C or on a DSP processor. There is another method, which uses parallelism in the algorithm to compute LP coefficient is Schur Algorithm, provides complexity of the order O(p) with the help of parallel processing.

### 3.3.2 SCHUR ALGORITHM

The Schur algorithm achieves minimum complexity by avoiding the calculations for inner products i.e. product of $\mathbf{\Gamma}_{m-1}$ **.** $\mathbf{a}_{m-1}$ in Levinson-Durbin. The recursive procedure for finding PARCOR coefficients is described below. The details of how these steps are derived and implementation of algorithm on parallel architecture, are given in [3].

**Initialization:** The matrix called Generator matrix $\mathbf{G}_0$ of the order 2-by-(p+1) needs to be initialized as,

$$\mathbf{G}_0 = \begin{bmatrix} 0 & R(1) & R(2) & . \; . \; . \; . & R(p) \\ R(0) & R(1) & R(2) & . \; . \; . \; . & R(p) \end{bmatrix}$$

**Step 1:** Shift the second row of generator matrix to right by one and discard the last element of this row. Take zero from left as the first element to get new generator matrix as,

$$\mathbf{G}_1 = \begin{bmatrix} 0 & R(1) & R(2) & . \; . \; . \; . & R(p) \\ 0 & R(0) & R(1) & . \; . \; . \; . & R(p-1) \end{bmatrix}$$

Now, the negative ratio of elements in second column yields the reflection coefficient, $K_1 = - R(1) / R(0)$.

**Step 2:** Multiply the generator matrix $\mathbf{G}_1$ by a 2-by-2 matrix,

$$\mathbf{V}_1 = \begin{bmatrix} 1 & K_1 \\ K_1 & 1 \end{bmatrix}$$

the multiplication gives 2-by-(p+1) matrix as shown below,

$$\mathbf{V}_1\mathbf{G}_1 = \begin{bmatrix} 0 & 0 & R(2) + K_1.R(1) & . \; . \; . \; . \; . & R(p) + K_1.R(p-1) \\ 0 & R(0) + K_1.R(1) & . \; . \; . \; . & . \; . \; . \; . \; . & R(p-1) + K_1.R(p) \end{bmatrix}$$

**Step 3:** Shift the second row of $\mathbf{V}_1\mathbf{G}_1$ matrix to right by one, discard the last element and take up zero from left, to get the matrix $\mathbf{G}_2$,

$$G_2 = \begin{bmatrix} 0 & 0 & R(2) + K_1.R(1) \; ..... & R(p) + K_1.R(p-1) \\ 0 & 0 & R(0) + K_1.R(1) \; ..... & R(p-2) + K_1.R(p-1) \end{bmatrix}$$

The negative ratio of elements in third column of $\mathbf{G}_2$ yields second reflection coefficient i.e. $K_2 = [\ R(2) + K_1.R(1)\ ] / [\ R(0) + K_1.R(1)\ ]$

Step 2 and 3 are repeated until we have solved for all p reflection coefficients. In general the 2-by-2 matrix in step m is,

$$\mathbf{V}_m = \begin{bmatrix} 1 & K_m \\ K_m & 1 \end{bmatrix}$$

and the multiplication of $\mathbf{V}_m$ by $\mathbf{G}_m$ yields $\mathbf{V}_m\mathbf{G}_m$. As in step 3, we shift the second row of $\mathbf{V}_m\mathbf{G}_m$ one place to the right and thus we obtain new generator matrix $\mathbf{G}_{m+1}$

As mentioned above, Schur gives complexity proportional to O(p) while parallel processing. Since we are using serial processing; for the implementation purpose, Levinson-Durbin Recursion is selected.

## 3.4 MODELLING OF EXCITATION SIGNAL

The excitation signal is the one which is needed for synthesis side filter H(z) to reproduce the modelled signal. An ideal excitation signal is the residual error which is obtained by filtering original signal using 1/H(z). The compression is achieved by modelling the residual signal or by finding a suitable substitution such that it will produce the effect same as residual. The synthesized speech quality purely depends on the excitation signal, thus finding optimum excitation becomes very crucial. The various types of speech coders are derived, depending upon method they use for modelling the excitation signal. Some of them are explained in brief in following section.

## 3.5 VARIOUS LPC ALGORITHMS

The dissertation in this section is focused on improvements proposed for LP analysis by modelling the excitation signal. In LPC based algorithms, all efforts are put in finding suitable excitation. The methods explained below have their own way of doing it, thus each of the techniques vary in its attributes like bit-rate, quality, complexity etc.

### 3.5.1 STANDARD LPC-10

In 1984, the United States Department of Defense produced federal standard LPC-10. The standard assumes signal being sampled at 8 kHz i.e. 8000 samples per second which are broken into frames of 180 samples. Thus each segment represents 22.5 ms of speech. In this method, the speech segment is classified as voiced or unvoiced. Depending upon such criteria, the excitation signal is periodic pulse generator for voiced segment and random noise for unvoiced segment.

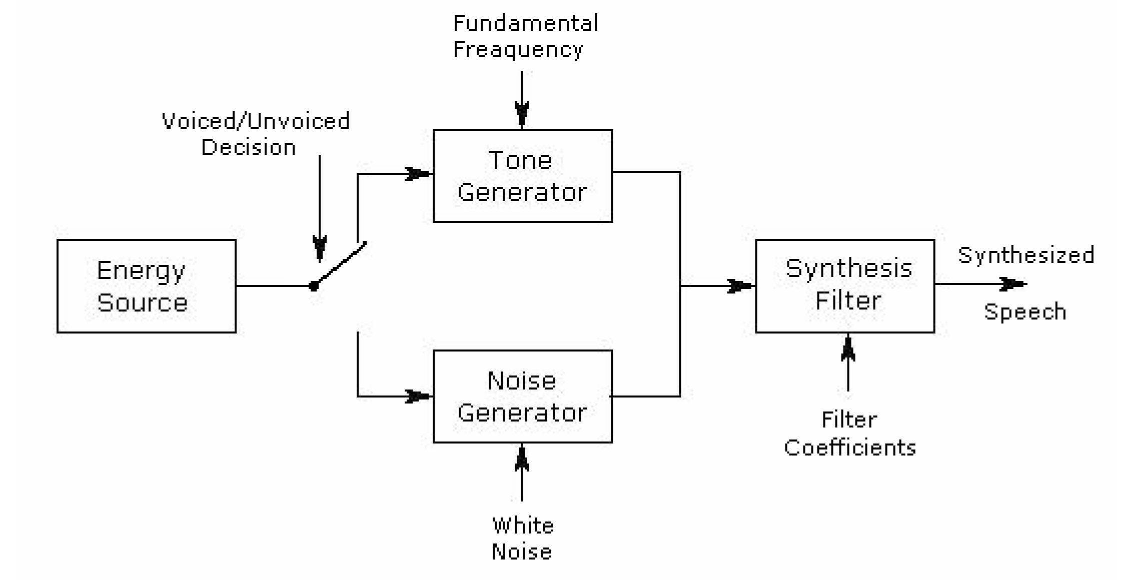


Figure 3.4 Basic Speech Synthesis Model for LPC-10

LPC-10 uses Average Magnitude Difference Function (AMDF) for finding the pitch period which is defined as,

$$\text{AMDF}(P) = (1 / N) \;.\; \sum_{i=k_0}^{N} | y_i - y_{i-P} |$$

The higher degree of approximation in excitation parameters results in poor quality of reproduced speech in LPC-10. The total numbers of bits required for each frame or segment are 54 bits which are distributed as follows –

| | | | |
|---|---|---|---|
| 1 | bit | - | voiced/unvoiced |
| 6 | bits | - | pitch period |
| 10 | bits | - | K1 and K2 (5 each) reflection coefficients |
| 10 | bits | - | K3 and K4 (5 each) |
| 16 | bits | - | K5, K6, K7 and K8 (4 each) |
| 3 | bits | - | K9 |
| 2 | bits | - | K10 |
| 5 | bits | - | gain G |
| 1 | bit | - | synchronization |
| 54 | bits | - | TOTAL BITS PER FRAME |

As, there are 180 samples per frame, therefore 44.4 frames per second results in bit rate of 2400 bits per second.

### 3.5.2 RESIDUAL EXCITED LINEAR PREDICTION (RELP)

In this method the residual is quantized severely while retaining adequate speech quality. Thus in RELP, kM bits are needed in addition to N predictive coefficients, corresponding to quantizing each sample of residual to k bits and M is the length of residual signal. RELP provides the least compression among parametric coders, results in bit rate closer to 16 kbps or even more. The only advantage of this method is that the complexity is very less.

### 3.5.3 MULTI-PULSE EXCITED LINEAR PREDICTION (MP-LPC)

The principle of MP-LPC is the Analysis-by-Synthesis technique, used while evaluating the appropriate excitation which is verified by synthesis. The error between synthesized and original signal is decides the further excitation signal. The excitation is generated in terms of fixed number of pulses per frame whose position and amplitude is to be determined. The implemented speech coder discussed in next chapter uses the same technique.

### 3.5.4 CODE-EXCITED LINEAR PREDICTION (CELP)

CELP is another method which relies on Analysis-by-Synthesis technique. The synthesis filter, on analysis side is excited by a series of residual signals which reside in a codebook. The codebook is fixed in size (varies from 1K to 4K as per the design) and its contents. The index of the entry which results in least error between synthesized and original speech, is sent to the listener along with filter parameters. The major concern here is to design a codebook which needs large amount of experimental data. While adding an entry to a book, it must be ensured that the entry similar or almost similar to it is not present already. With CELP the bit-rates around 2.4 kbps are easily achieved.

In advanced CELP method, original speech is divided into number of frames each further divided into subframes. The frame based classifier divides the frame into three categories as voiced, unvoiced and transition. The excitation vector is computed for each subframe using a search procedure through two codebooks, an adaptive codebook and stochastic codebook. During synthesis the voiced class uses adaptive codebook, whereas unvoiced class uses stochastic codebook. The transition class uses both codebooks.

### 3.5.5. VECTOR-SUM-EXCITED LINEAR PREDICTION

The VSELP technique is been standardized as IS-54 by EIA (Electronic Industries Association), which describes an algorithm that encodes speech at 8 kbps. In this method, an excitation is composed of linear (vector) combination of two different codebook excitations. The frame size is fixed to 160 samples corresponding to 20 ms for 8 kHz speech signal. The frame is further divided into subframes of 40 samples (each of 5 ms). Certain parameters are actually changed at the subframe rate to provide smooth transitions of parameters. The 159 bits per frame are divided among the parameters as shown below.

- 38 bits - filter coefficients ($a_i$)
- 5 bits - frame energy
- 28 bits - lag L (7 bits per subframe)
- 28 bits - codeword I (7 bits per subframe)
- 28 bits - codeword H (7 bits per subframe)
- 32 bits - gains ($\beta$, $\Gamma_1$, $\Gamma_2$)
- 159 bits - TOTAL PER FRAME

The quality of synthesized speech from a subjective point-of-view, is far superior to that obtained by 16 kbps ADPCM. On the other hand, VSELP is computationally complex since requires processing at frame as well as subframe level.

# CHAPTER 4

## SPEECH CODEC IMPLEMENTATION

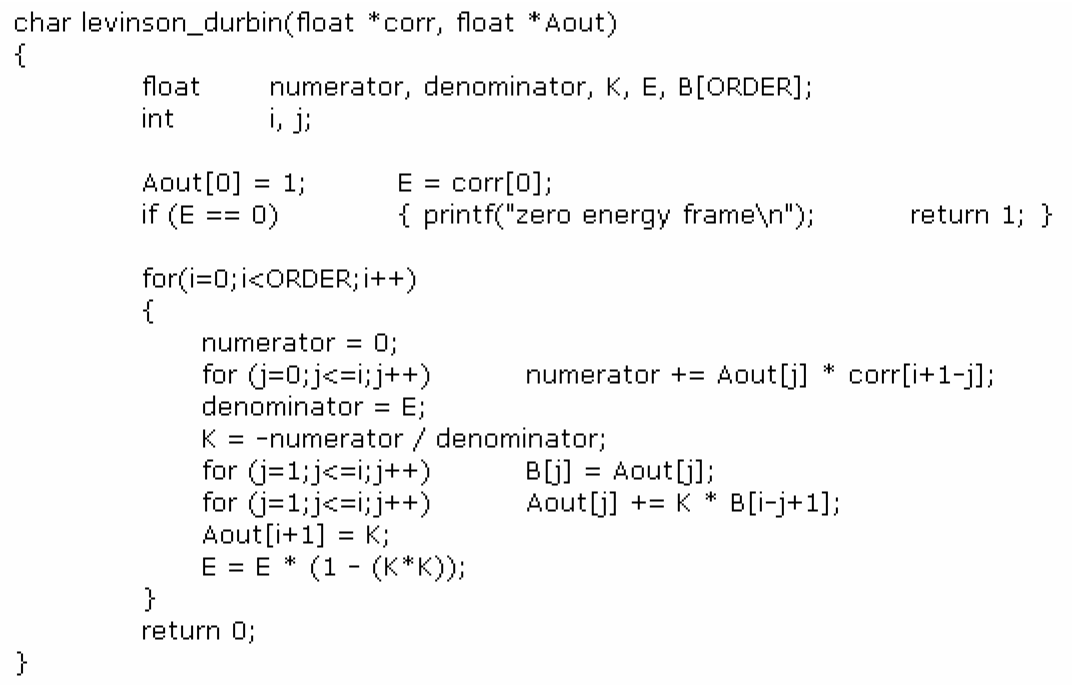
The various speech coders have been discussed in previous section. Out of those, we have managed to implement Multi-Pulse Linear Prediction Coder (MP-LPC) and its decoder in C. The coder provides bit-rate of 5.4 kbps (corresponding to compression ratio of 1:12) with much better perceptible quality of synthesized speech. This section will discuss the details of the implementation.

## 4.1 MULTI-PULSE ANALYSIS

The analysis phase is divided into two steps viz. one is evaluation of LP filter parameters which represent the filter, includes gain and filter coefficients and the other is to find suitable excitation signal.

### 4.1.1 FILTER PARAMETER ESTIMATION

The parameter estimation technique is common almost for all speech coders, which uses either Levinson-Durbin or Schur Algorithm to evaluate coefficient vector. Since we are not using hardware which supports parallel processing, we decided to use Levinson-Durbin. The details of Levinson-Durbin have been discussed in section 3.3.1. The equations 3.14, 3.15 and 3.16 are used to calculate filter coefficients recursively. A C function for the same is given in figure below. The parameters passed to this function are, an input correlation vector pointer (float *corr) and the output coefficient vector pointer (float *Aout). On success the function returns zero, the return value one suggests that the frame has zero energy implying that the parameters are zero and there is no need to calculate those.



```
char levinson_durbin(float *corr, float *Aout)
{
          float     numerator, denominator, K, E, B[ORDER];
          int       i, j;

          Aout[0] = 1;        E = corr[0];
          if (E == 0)         { printf("zero energy frame\n");       return 1; }

          for(i=0;i<ORDER;i++)
          {
               numerator = 0;
               for (j=0;j<=i;j++)        numerator += Aout[j] * corr[i+1-j];
               denominator = E;
               K = -numerator / denominator;
               for (j=1;j<=i;j++)        B[j] = Aout[j];
               for (j=1;j<=i;j++)        Aout[j] += K * B[i-j+1];
               Aout[i+1] = K;
               E = E * (1 - (K*K));
          }
          return 0;
}
```

Figure 4.1 C Function for Levinson-Durbin Recursion

### 4.1.2 SELECTION OF EXCITATION PARAMETERS

The synthesis side filter needs to be excited by suitable excitation which is calculated in MP-LPC by using Analysis-by-Synthesis technique. The speech coders like MP-LPC can afford to use synthesis multiple times for analysis since the complexity of synthesis filter is very less, as it required just a filtering operation. The synthesis just ensures whether the evaluated excitation signal is sufficient, if not, the excitation signal is modified accordingly and the process continues. The block diagram of such technique is shown in figure 4.2.

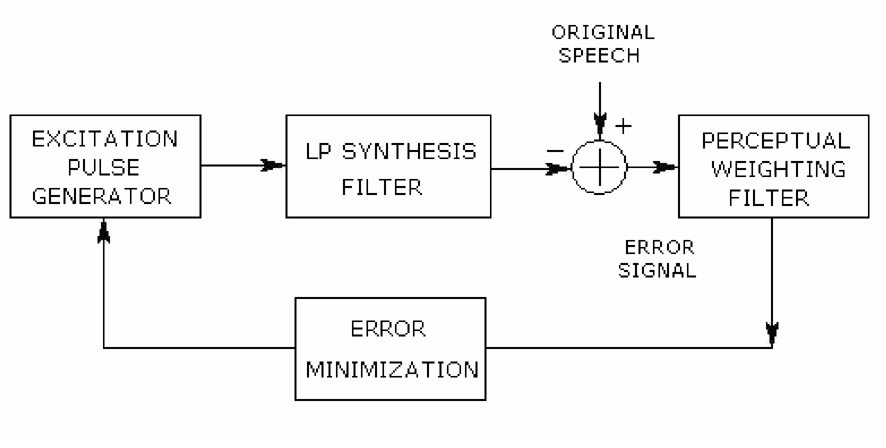


Figure 4.2 Block Diagram of MP-LPC

To find appropriate excitation pulse, we need to find its position as well as amplitude. A pulse of amplitude $A_m$ at position m will generate signal which is a convolution of impulse response of synthesis filter $H_n$ and the input pulse $A_m$. Thus first pulse can be found by placing it at such a position that it minimizes the error between original signal and impulse response of the filter. This error can be represented as,

$$E(n) = [ D(n) - Am . H(n-m) ] \qquad \text{- Eq. 4.1}$$

Thus, the mean square error for a frame of length N is,

$$E = 1/N \sum_{n=0}^{N-1} [ D(n) - Am . H(n-m) ]^2 \qquad \text{- Eq.4.2}$$

The pulse magnitude can be evaluated by applying least mean square error method. i.e. by differentiating the square error with respect to Am and equating it to zero. Thus,

$$\delta E^2 / \delta Am = - 2/N \sum_{n=0}^{N-1} H(n-m) [ D(n) - Am . H(n-m) ]$$

equating this equation to zero gives,

$$Am = \Phi m / \Phi mm \qquad \text{- Eq. 4.3}$$

where, $\Phi m = \sum D(n) . H(n-m)$

and $\Phi mm = \sum H(n-m) . H(n-m)$

The value of Am needs to be substituted in equation 4.2, to eliminate the amplitude dependant term, and to obtain expression only in terms of pulse position m. This expression for mean square error (MSE) turns out to be,

$$E = 1/N \left[ \sum_{n=0}^{N-1} D^2(n) - \Phi^2 m / \Phi mm \right]^2 \qquad \text{- Eq.4.4}$$

Clearly, the error will be minimum if second term is maximum. But, the denominator term Φmm is constant. Only the value of numerator varies with change in m. Therefore, the term Φm needs to be maximum for mean square error to be minimum, suggests that the value of m which gives maximum Φm is the most appropriate position for the pulse. Thus, using original speech D(n) and impulse response of filter designed H(n), value for pulse magnitude along with its position can be calculated using equations 4.3 and 4.4.

The subsequent pulses could be determined by following the same procedure using modified desired signal, obtained by removing the effect of previous pulses. Thus, for following pulses, desired signal will change to,

$$\check{D}(n) = D(n) - Am \,.\, H(n-m) \qquad \text{- Eq. 4.5}$$

Instead of one, we can even place more than one pulse at a time so that the mean square error is reduced. Such method is called Multiple Pulse Placement. The equation 4.1 in this case will get modified to,

$$E(n) = [\, D(n) - \sum_{i=1}^{N_P} Am_i \,.\, H(n-m_i) \,] \qquad \text{- Eq. 4.6}$$

The further analysis will be the same as it was for single pulse placement, which gives solution of the form,

$$\mathbf{A}m = (\mathbf{\Phi}mm)^{-1} \,.\, \mathbf{\Phi}m \qquad \text{- Eq. 4.7}$$

where, $\mathbf{A}m$ is column vector of order $N_P$-by-1

$\mathbf{\Phi}mm$ is a square matrix of the order $N_P$-by-$N_P$

$\mathbf{\Phi}m$ is also a column of order $N_P$-by-1

The computation of matrix inverse in above equation again is of complexity $O(N_P^3)$ by normal method. Cholskey Decomposition is the algorithm which reduces the complexity to $O(N_P^2)$. The algorithm is described in [3]. More details on pulse placement in Multi-pulse Linear Prediction are given in [12].

## 4.2 MP-LPC SYNTHESIS

For synthesis, excitation signal needs to be convoluted with impulse response of the filter. An excitation signal is determined by the pulses evaluated during analysis. The signal except at pulse positions remains zero. From experimental results, it is observed that at least four pulses (depends on pitch and frame length) are needed to reconstruct the fairly perceptible speech. The following figure show how few pulses almost reproduce the speech signal.

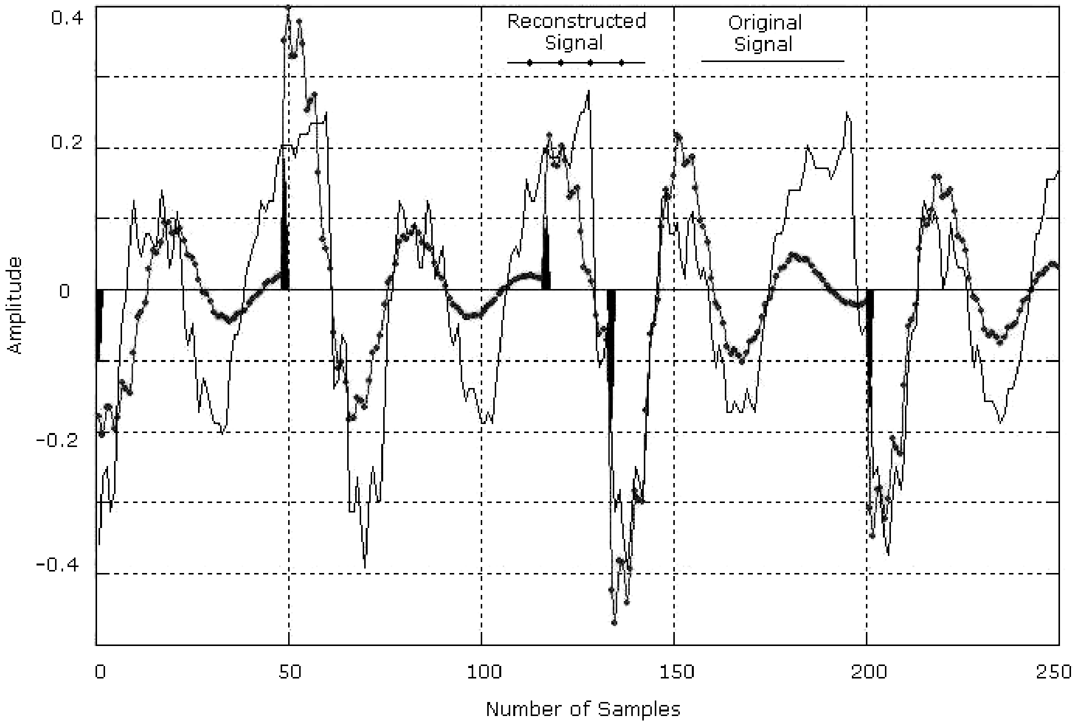


Figure 4.3 Reconstructed Frame using MP-LPC (five pulses)

For 5.4 kbps coder, we are generating five pulses. Of course, the quality improves with addition of more pulses, but the concern here is, how much it improves with each pulse. To analyse that, we have evaluated mean square error for a complete test speech signal with number of pulses varying from 1 to 30. The figure below shows the variation in MSE with number of pulses.

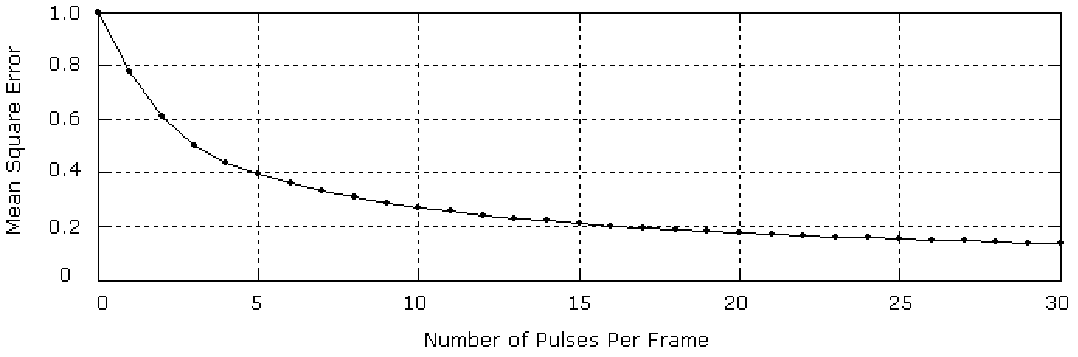


Figure 4.4 Comparison between MSE and Number of Pulses

The mean square error will be equal to energy of the signal when there is no excitation. Here that value is one since the values of MSE are normalized. As graph shows, half of the signal is recovered with only three pulses. The curve tends to go flat at higher values suggests that increase in number pulses does not improve the quality in equal proportion. Therefore, just increasing number of pulses is not the solution to achieve higher quality. The other methods for noise reduction like Gradient Adaptive Lattice (GAL) predictor are discussed in [7] and [13].

### 4.3 PERFORMANCE MEASURES

This section will discuss the attributes of implemented MP-LPC speech coder which is described so far in this chapter. The main four attributes of speech coder are bit-rate, complexity, delay and quality.

- BIT-RATE: The total number of bit required to represent parameters of a frame are distributed as follows,

| | | | | |
|---|---|---|---|---|
| ▪ | 10 | bytes | - | filter coefficients, $a_1$ to $a_{10}$ (one byte each) |
| ▪ | 1 | byte | - | gain G |
| ▪ | 5 | bytes | - | pulse positions (one byte each) |
| ▪ | 5 | bytes | - | pulse amplitudes (one byte each) |
| ▪ | 21 | bytes | - | TOTAL |

  A frame size of 250 samples results in 32 frames per second, which results in bit rate of bit-rate of 5376 bps (Approx. 5.4 kbps).

- QUALITY: It could be represented in terms of Signal-to-Noise Ratio (SNR). The following graph shows how the quality in multi-pulse excited prediction varies with number of pulses per frame.

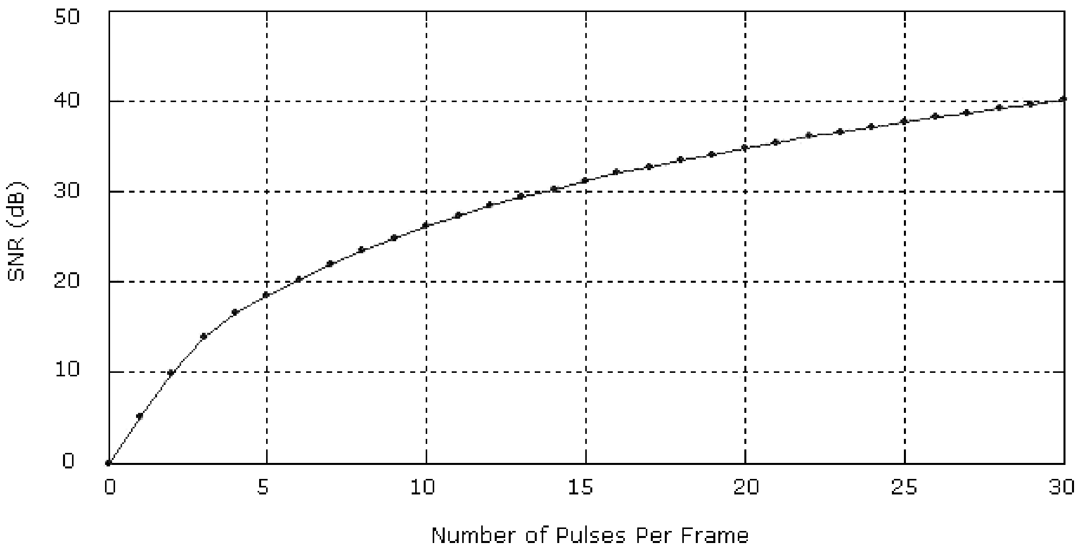


Figure 4... Variation in SNR with Number of Pulses per Frame

- COMPLEXITY: The model-based speech coders involve extensive, arithmetic calculation. The complexity is decided mainly by number of arithmetic operations needed per second. Other overheads due to looping, conditional statements and I/O are normally ignored. Doing so, the implemented coder contains approximately 5.3M additions and multiplications per second. The complexity and processing delay increases in equal proportion with increase in number of pulses. This is the reason why we selected only five pulses per frame.

- DELAY: This attribute matters only in real-time implementation of speech coder. Since the signals are processed by breaking them up into frames, the coder needs to accumulate complete frame before processing. Thus a signal gets delayed by the time equivalent to the frame length in ms, which is 31.25 ms in case of our implementation (called algorithmic delay). The signal is further delayed during processing which purely depends upon the hardware configuration used (called processing delay). Therefore the total delay will be accumulation of arithmetic, processing and transmission delay where last one depends on the performance of communication system.

# CHAPTER 5

## CONCLUSION

The MP-LPC based speech coder discussed in this document is implemented in C and Matlab. The Matlab was used to verify and simulate the algorithm before their C implementation. The C code is capable of processing 8-bit linear PCM (.wav format) speech and provides compression around 1:12.

### 5.1 SUMMERY OF WORK DONE

To start with the project, we studied the principle of Linear Prediction. In analysis, phase, after finding the filter parameters, we observed the effects of various excitations like the one given in LPC-10, at regular intervals based on pitch period. This method did not provide better quality due to higher degree of approximation in excitation. Then we studied the coders which make lesser approximations to provide better quality at a cost of slight increase in bit-rate and complexity. Among these techniques, we successfully implemented Multi-pulse Excited Linear Prediction. Since the excitation is evaluated using MMSE method, MP-LPC is least approximated and more accurate compared to other coders like CELP. The minimum bit-rate of 5.4 kbps is achieved with five pulses per frame.

### 5.2 FUTURE DIRECTIONS

In order to improve the performance of coder, noise reduction techniques such as Gradient Adaptive Lattice (GAL) can be used. Since the complexity of decoder is quite less as compared to the coder, a cost effective decoder can be designed for devices which are speech or voice enabled. For real-time implementation, encoders will need device with high processing capability and more memory.

# A. REFERENCES

[1] Speech Signal Processing

*- by Morgan, Gold*

[2] DSP in Telecommunications

*- by Kishan Shanoi*

[3] Algorithms for statistical signal processing

*- by Proakis, Rader*

[4] Digital Signa Processing

*- by Proakis*

[5] Linear Predictive Coding

*- by Jeremy Bradbury*

[6] Robust Linear Prediction Analysis for Low Bit-Rate Speech Coding

*- by Nanda Prasetiyo Koestoer*

[7] Optimized Implementation of Speech Processing Algorithms

*- by Sara Grassi*

[8] Interpolation of Linear Prediction Coefficients for Speech Coding

*- by Tamanna Islam*

[9] Coding of Excitation Signals in a Waveform Interpolation Speech Coder

*- by Mohammad M. A. Khan*

[10] Very Low Bit-Rate Speech Coding using Speech Recognition, Analysis and Synthesis

*- by Jukka Kivimąki*

[11] Modified LPC Parameter Dynamics to Improve Speech Coder Efficiency

*- by Wesley Pereira*

[12] Efficient Computation and Encoding of The Multipulse Excitation for LPC

*- by M. Berouti, H. Garten.*

[13] Methods of Noise Cancellation based on EM Algorithm

*- by Meir Feder, Alan V. Oppenheim*