VoIP data Rate Reduction Exploiting Linear Prediction Coefficients Redundancy

Islam Younis Morshed Amro

Chapter One

Introduction to Networking and Signal Processing Approaches

1.1 Introduction

Voice over Internet Protocol (VoIP), is a new technology that aims to transmit voice as packets over the internet, so that the wired and the wireless systems will be holding the same type of data and serve the same purpose, rather than having two separate systems, one for voice and another for data. This actually is part of a bigger problem which is converting all types of communication into data communication and converge to the point that the use of different media will serve the same purpose, and follow the same constrains, respecting the privacy of each type of media, i.e., video constrains and audio constrains, etc.

Through the literature review, we have noticed that the state of mind for all the resources (authors and even technology vendors) states the VoIP as a telephone systems alternative, which is true since the VoIP system in its best shapes is a replacement for what currently exists, and a feasible migration strategies should be introduced with this new topic.

The problems of VoIP technology arises from the networking part and the codec part, the networking related problems are not in the scope of our research.

The codec used in the VoIP embodies the process of compressing and decompressing the audio signals, this operation is being adapted for evident reasons, the codec related constraints are:

- Maintaining the data rate of the transmitted signals as lower as possible.
- Complexity, the process of reducing the data involves further algorithmic operations that would increase the complexity.
- The increase of complexity would increase the processing time, this means we are encountering a higher delay.
- All of the operations are governed by maintaining a good quality for the signal in an objective manner. (Chou 2003)

The reason why we have to go through all of these investigations is to increase the resource utilization in our networks, the lower that data rates per VoIP connections means that we can multiplex more simultaneous calls in our link according to its capacity. Since the increase of the link capacity is always more expensive when it comes to the business levels. (Cisco 2003)
The data rate reduction with a reasonable complexity, delay and acceptable quality has always been the direction of research in the voice communication environments, but the demand on lower bits data rate has increased with the deployment of the VoIP technology (Orthman 2004).
This research is aiming at reducing the data rate of transmitting the human voice in a VoIP communication environment that uses the Linear Prediction Coding (LPC) as the base of its codecs as explained in chapter four.

The packetised information in VoIP or what is being recognized as packet payload is the LPC parameters, therefore the LPC parameters dictates that data rate of the transmission.

These parameters are the digital filter coefficients, the excitation information and gain, this information expresses a certain window of the signal called frame, and this frame has a known fixed length.

The data rate reduction approaches are subjected about these parameters, as seen in chapter two, some approaches deals with the digital order, this is seen in the new iLBC, but most of the approaches deals with the excitation information as in CELP, and others deals with reducing the data rate by reducing the window size as LD-CELP.

Our research works on reducing the data rate exploiting the linear prediction coefficients redundancy, this concept is being elaborated in chapter four.
Filter coefficients redundancy is the process of representing a certain frame filter coefficients with another frame filter coefficients, the replacement is done according to an acceptable distance between each corresponding lag elements values of the two filters.

As a development of this simple concept, we built us a digital filter coefficients dictionary, the entries of this dictionary were obtained previously from a certain speakers, and this dictionary is to be shared between the sender and the receiver, when a the speakers starts to speak, each of his frames are being analyzed and the coefficients are being deduced, then this frame coefficient is being compared with the dictionary entries against the acceptable distance, when we have a match, the index of the matched coefficient is being transmitted instead of the whole coefficient string since the receiver had a copy of that dictionary, we are transmitting in this case a one value which is the index instead of M values which the number of coefficients in that frame, further elaboration of this concepts is seen in chapter four including the research results.

This process is being simulated in the following steps

- Generate the dictionary from previous samples (wav files) of a certain speaker.
- Use a new sample of the same speaker in investigating the replace
- Start the LPC analysis.
- At the end of calculating each frame coefficients, start searching for its match in the dictionary respecting the acceptable distance condition.
- This distance is previously specified, i.e. before starting the execution.
- If there is a match take the value in the dictionary and save it in the output file respecting the frame order, and discard the value being calculated (i.e. the frame filer coefficients is being expressed in term

of its corresponding index in the dictionary, so the transmission will be for its index only, rate reduction achieved)

- If there is no match in the dictionary save the originally deduced coefficients in the out put file, and update the dictionary with this value, since it has not learned about it yet, this value is being saved at the end of the bottom of the dictionary. (i.e. the whole string coefficients will be transmitted, no rate reduction is being achieved)
- The output file then is being synthesized back to a wav file, this file is being assessed by the MOS, success indication is taken form this file

The methods that is being used in the quality assessment for speech signals is being introduced in chapter three.

Regarding the complexity issue, we have added further complexity for the operation, but the complexity analysis is not in scope of this research, we focused on the usability of the digital filter coefficients redundancy concept in data rate reduction.

Chapter five includes a theoretical adaptation of this concept in the VoIP environment, an introduction for the RTP packet structure modification was done, and how packets holds data rates information, and how the transmission reception flow works.

In the following; we will build up the concepts needed by this technology, starting from the VoIP as a telephone alternative, the technology constraints, and then move to the theory behind the codec used by VoIP.

1.1.1 Telephone Alternative

This means using a VoIP system to make a voice call to another person, and this can be done in several ways.

A PC might be connected directly to network, talking to another PC common on the same media; the PC should be equipped with a soundcard, speakers, microphone and a VoIP application that will do the whole work. The second scenario is having the telephone connected to the PC, the call is made like any other regular call, this needs to have an always on directly connected computer to the network.

The last case comes with omitting the computer and having a voice gateway, this gateway has all the telephones connected about it, and connects the public telephone network with a computer network and performs the necessary actions and conversations to make the call possible. (Cisco Systems 2002) (Ohrtman 2004)

1.1.2 Components of a VoIP System

The components functionalities and the entire process of the VoIP operations are illustrated in figure 1.1. This is not the architecture of the system since we found this slightly away from the terrain.

The processes sequence of sending signals is expressed by arrows pointing downward, and the arrows pointing upward define the processes sequence of speech signals reception. The label of a box contains two items, the left one indicates processing occurring at this layer when we are sending a speech signals and the right one indicates the reception of the signals. They are being grouped together because they function at the same level, and the right item does almost the opposite of the left one. (Cisco Systems 2002) (Ohrtman 2004)

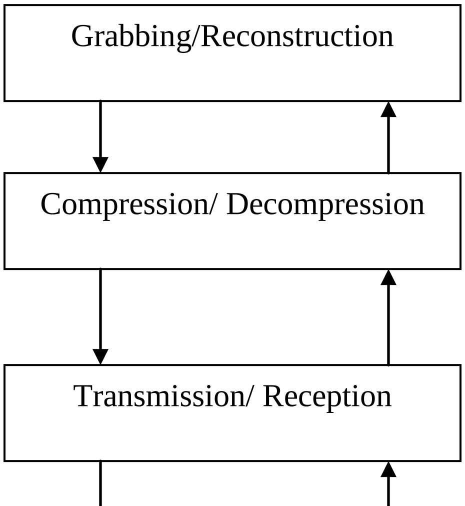


Figure 1.1: VoIP Process

1.1.3 Grabbing and Regeneration

A digitization process is required before we are able to transmit voice over a network, this is the analogue to digital conversion, in general the microphone acquires the speech signal and the soundcard performs the conversion process, this process is known in voice communication as grabbing. (Cisco Systems 2002)

To maintain the real time aspects of the conversation, speech is subdivided into blocks, we have to send to the other end as much as we can of these blocks, the receiver must always have data to display respecting the delay constraints, the whole process of conversion and sending must go with an acceptable delay. When the digitized block is being received; it should go back to its previous shape before the conversion, this is also performed by the soundcard, granting that we have regenerated a signal that holds most of the properties of the signal being digitized by the sender. The regeneration is the reverse operation of grabbing. (Cisco Systems 2002)

1.1.4 Compression and Decompression

Bandwidth is a key element in the communication process, and any digitization process should take into consideration the bandwidth of the environment we are going to use in our transmission, this dictates the need for data compression. Various compression schemes are used to reduce the required bandwidth for voice communication.

Some voice compression approaches use general compression techniques which are also applicable to other kinds of data; other types uses the fact that we are dealing with voice information to achieve large compression ratios. The decompression is the inversion of the compression process, this inversion might be a procedural inversion, i.e. use the opposite procedures sequence being used in the compressor, other approaches use mathematical inversion if a mathematical compressor were used, a combination of the two types is commonly used. (Ohrtman 2004)

1.1.5 Transmission and Reception

In this portion we care about the end to end problems between the sender and receiver, where blocks have to be sent from source to destination across the network. We are concerned now with the channel delay and losses, the order of the received data blocks is important here and the quality of the overall transmission of the voice signals. (Cisco Systems 2002)

1.2. Introduction to TCP/IP

Now we will start talking about the logical infrastructure of the VoIP, which is the TCP/IP. The TCP/IP, the Transmission Control Protocol and the Internet Protocols are the underlying infrastructure set of protocols that are responsible for the transmission and reception of data over the internet. The following will be a general review for the protocol. Over the TCP/IP infrastructure; all kinds of communication occur, i.e. Data, Audio and Video on the same internetworking media, using the same protocols and topologies for evident reasons.

1.2.1 The TCP/IP Communication Protocols Suite

The TCP/IP is made up of several protocols; the two main ones are the TCP and the IP. It was basically built into the UNIX operating system, making it the de facto standard for transmitting data over networks, even network operating systems that have their own protocols, such as Netware, also support TCP/IP. (Cisco Systems 2002)

The TCP/IP is one implementation of the famous standard ISO-OSI (Open Systems Interconnection) layered model. All the references introduce the TCP/IP as identical to the ISO-OSI model even though it is different, but its easier to understand the TCP/IP stack if it was introduced as an ISO-OSI model.

1.2.2 TCP/IP Implementation

The specialty of the TCP/IP as an ISO-OSI implementation comes from its organization and the declared role of each layer. The following represents the TCP/IP stack. (Rao et al. 2003)

4) Application
3) Transport
2) Internet
1) Host to Network

Figure 1.2: TCP/IP Implementation

The host to network layer

This layer is quit identical to the physical layer of the ISO-OSI. It is the interface between the logical and physical components of the network, and supplies the TCP/IP with the media interportability it needs. (Bryan 2004) (Ohrtman 2004)

The internet layer

In this layer; several packets that flow between networks and expect to be directed to where they should go. So this layer is concerned about having each packet routed to the most likely true destination; most of the packet loss happens in this layer since it has poor functionalities and depends highly on the process organization supported from higher layers. The only data is being added to the packet here is the IP address datagram, which organizes the source, destination and route operations.

The Transport Layer

Here; higher functionalities are being specified to have a valid session, and the problem of having multiple applications using the same IP is being solved, the problem of multiple applications using the same IP is solved using an extra naming approach, to make sure that multiple applications can use the network facilities at once, which is basically the port. Here also, the transport layer is the first real end to end layer.

The TCP/IP model has two major transport layer protocols. One of them is the Transmission Control Protocol (TCP). This protocol transforms the connectionless unreliable packet caused by the lower internet layer into a connection oriented and reliable byte stream. It is a very important protocol since it makes reliable communication possible. This is achieved by adding a sequence of states that a session must follow, this sequence is referred to as the session states. Networks intrusions usually discovered after someone shows an irregular state behavior.The other protocol is the User Datagram Protocol (UDP). This is a protocol for applications that do not need the services offered by TCP or want to use a protocol of their own. The User Datagram Protocol is merely a small extension to IP. It is also an unreliable packet based connectionless protocol and the only real extensions to IP itself are the presence of a port number and an optional checksum of the data, this makes a good choice for applications that avoid processing delays. (Karrenberg 2002) (Ohrtman 2004)

The Application Layer

This is where the networking applications reside. Like the HTTP, virtual terminal applications (TELNET protocol), file transfer utilities (FTP protocol) and electronic mail (SMTP protocol). (Bryan 2004)

1.2.3 How IP Works

Packet Format

packets Sent by the IP layer consist of an IP header, followed by the transmitted data. The IP header is shown in figure 1.3 below. The least significant bit on the left is numbered zero. The most significant bit on the right is numbered thirty one. Transmission is done in network in octet order. This means that the most significant octets are sent first and the least significant octets are sent last.

<table>
<tr><td colspan="3">0</td><td colspan="3" align="right">31</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td>version</td><td colspan="2">IHL</td><td>Type of Service</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="4">Identification</td><td>Flags</td><td>Fragment Offset</td></tr>
<tr><td colspan="2">Time To Live</td><td colspan="2">Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="6">Source IP Address</td></tr>
<tr><td colspan="6">Destination IP Address</td></tr>
<tr><td colspan="6">Options (not mandatory)</td></tr>
</table>

Figure 1.3: Packet format

The version field should contain the value four for the current version of the Internet Protocol (IPv4 or 6) (Cisco Systems 2002). The IHL field contains the Internet Header Length; it specifies the length of the header in units of 32 bit words. (Swale 2001) The next field is the Type of Service (TOS) field. This field was meant to supply a Quality of Service (QoS) mechanism. The Time to live (TTL) is used to limit the lifetime of a datagram. Routers decrement the value by at least one, if the packets stay a long time in the router queue, the TTL value should be decreased by number of seconds the datagram spent in queue. When the counter is zero, the datagram is discarded. The protocol field is used to specify the internet or the transport layer protocol is being used. The header checksum is used to check the validity of the datagram.

A minimal IP header contains the source IP address and the destination IP address. These addresses are seen in every datagram since the internet layer operates in a connectionless way, this is how basically packets are being routed between networks in the back to back connections, and also how packets are bring delivered to hosts in the back to end connection. (Bryan E. Carne 2004)(Ohrtman 2004)

1.2.4 Transport Protocols

In the previous part we have seen the internet protocol, now we will have a look on the protocols that comes over this protocol, which is being served by the IP. The Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) are the two transport level protocols in the TCP/IP.

1.2.4.1 TCP

The TCP is the most used transport of the two protocols in general, where a transformation of the unreliable packets transmission based of the internet layer into a reliable byte stream occurs. The protocol is designed for communication between two ends. details of this protocol was avoided since it is not a concern of voice transmitting environment, but after the session is being established, it has a unique picketing approaches and a well known packet types that ensure a healthy sequence of session. (Galis 2004)

1.2.4.2 UDP

We have lowered the profile of the UDP already, accusing it by adding nothing special on the functionality level of the IP, but avoiding the functionality complications has made it an excellent choice for the real-time applications, since it is relatively very fast compared to the TCP. The UDP just adds its header to the fragment making a packet, and pass it to the IP layer to transmit it. This means that just like IP itself, UDP is a best effort service. No guaranteed delivery is given. The UDP header is shown in figure 1.4 below. The header contains the source and destination ports, which identify the sending and receiving applications. It also contains the number of data bytes which must be sent and finally the header contains space for an optional checksum. (Bryan 2004)

0	31
Source Port	Destination Port
Length	Checksum

Figure 1.4: the UDP Header

Since the service which UDP offers is almost identical to the service of IP itself, it is possible for applications to send UDP datagrams to a multicast address and to receive UDP datagrams from a multicast group. Extra functionalities are required by the voice transmission that UDP does not support, this is achieved by extra protocols added over the UDP that will compensate the missing functionalities as mentioned next. (Bryan 2004)

1.2.5 Transport Layer Voice Communication Protocols

1.2.5.1 Real Time Transport Protocol (RTP)

Real-time transport protocol (RTP) provides an end to end network transport function and is recommended for applications transmitting in a real time manner, such as audio and video. RTP is augmented by a control protocol (RTCP) to monitor data delivery and network statistics. Together they resolve many of the problems a UDP network environment may experience, such as lost packets, jitter, and out of sequence packets; these concepts will be exposed later in the chapter.

1.2.5.2 Real Time Streaming Protocol (RTSP)

Same as the RTP but in addition it is related to streaming applications like audio and video transmission. This increases the extra overhead for buffering purpose that the RTP doesn’t supply. This is due to the massive requirement of buffering in the streaming environment. It is impractical to overload the functionalities of the RTP with such overhead, so special protocols were committed for this specific environment. (Ohrtman 2004)

1.2.5.3 Session Initiation Protocol (SIP)

Session Initiation Protocol (SIP) is a control protocol that initiates, establishes and terminates a session or a call. SIP can be used with other call setup and signaling protocols, it provides the functionalities of call routing that is present in the PSTN. (Ohrtman 2004)

1.2.5.4 ITU H.232 Standards

This is the group of conferencing related protocols; it is often called the umbrella specification. This is because it uses several other (International Telecommunication Union) ITU recommendations to provide its functionality, unlike the previous protocols, it is a mature version of introducing the multimedia transmission standards in an industrial shape. The diversity that multimedia types require is well supported in these standards, unlike the strict networking concepts seen in other protocols. The structure of the H.323 architecture is illustrated in figure 1.5.

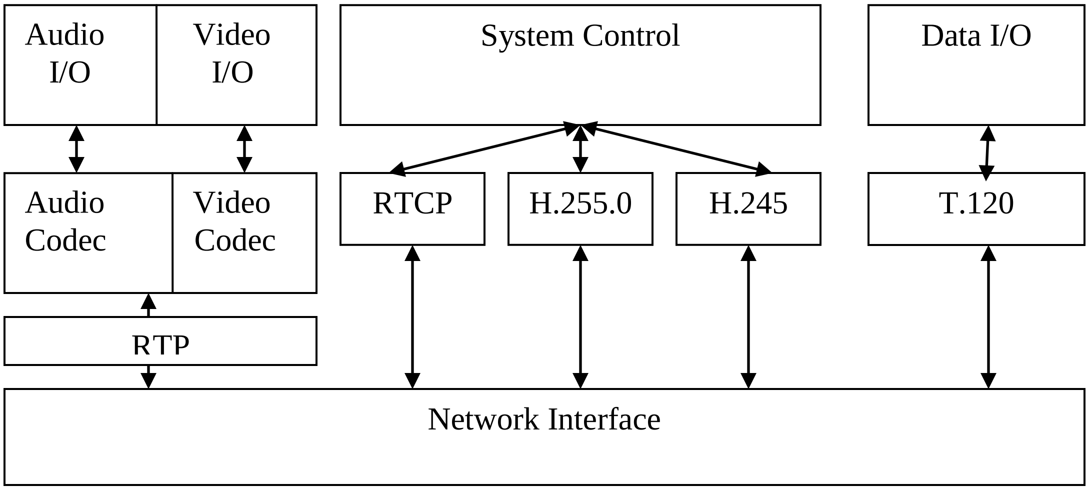


Figure 1.5: the H.232 Standards

The audio coders are the ITU-T G. standards which will be mentioned later. The video coders are H.261 and H.263. Both audio and video are encapsulated in RTP packets and then transmitted across the network. The call set up is prepared by the H.225.0 and H.245. After the call is set up; the system is entitled to specify the coders according to thier capabilities or the user selection, when the call has been established. The H.232 model is the real mature version of the VoIP communication protocols, and heavily implemented in the industry. (Cisco Systems 2002)(Ohrtman 2004)

1.3 VoIP as a Voice Communication System

We have decided to include the general voice communication systems characteristics along with the VoIP communication system characteristics. There are some general problems that reside in the voice communication systems, the data rate reduction and its conjugates seem the only advantages that VoIP has come up with, since it has made the classic constraints more complicated and annoying, the following will be an exploration for the VoIP constraints as a voice communication system.

1.3.1 Sampling Rate and Quantisation

It is known that a higher sampling rate and a smaller quantisation step imply a better representation of the original signal. But this also means that more digitized information will be transmitted and more bandwidth is required. So we have to determine how much information is necessary to hold a telephone quality conversation, recall that telephony quality is the ultimate that VoIP tends to have.

The Nyquist theorem is important in this matter. It states that the minimum sampling frequency should be twice the maximum frequency of the analogue signal. (Oppenheim et al. 2000)

Little thinking about this concludes that, to capture N cycles of a signal, you should be able to cover the area of 2N of the signal or you will be unable to catch the signal highest and/or lowest frequencies. The case of missing these frequencies is called aliasing, which is encountering a certain sampling window frequency into another adjacent sampling window.

The human speech can have higher frequencies up to 12 KHz (Chu Wai C. 2003). But the telephone coders assume that 4000 Hz allows high quality communication. Following up with Nyquist theorem, the sampling rate of 8000 Hz is adequate for sampling, with 8 bit being used for digitization, and the 64 Kbps number comes up. (Diniz 2002)

Normal telephone call wastes a large amount of bandwidth. When someone is not speaking, the bandwidth stays assigned without being used. With packetised voice this bandwidth could be used by other calls or applications. (Swale 2001)

1.3.2 Packet Length

With Voice over IP; the problems come with networking that affects the operation flow in voice transmission, in general; packets can get corrupted or even lost. To reduce the amount of lost information, a packet should contain only a very small amount of the voice signal. This way, if a packet gets lost, only a little fraction of the conversation will be missing, which is very unlikely to disturb the conversation. A new approach of tolerance is introduced by splitting the line spectral pairs into odd and even parts and fill each part in a separate packet. (Benjamin et al. 2005)

1.3.3 Buffering

The problem of jitter, which is having the packets in an out of order manner with and annoying regimentation of a certain connection among other connections fragments, this causes the packets to be mixed up together. To avoid jitter, buffering is used, within the buffer, a reordering operation is carried out with having a certain amount of speech blocks that will be used for the playback as illustrates in figure 1.6.

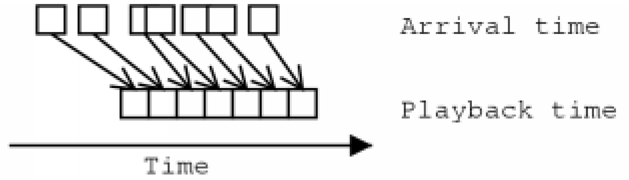


Figure 1.6: Buffering to avoid jitter

This adds an extra delay for the whole process, but the smoothness of the playback voice is granted, and any error correction operation can be masked with this delay time. (Cisco Systems 2002)

1.3.4 Delay

Large delays distort conversations. Delay can be of two types. The first is fixed. This delay is always present due to buffering and link transmission rate. The second is variable. This comes from the dynamic nature of the networks which is embodied by packet queuing in routers, congestions etc.

The components shown in figure 1.7 determine the amount of fixed delay for a VoIP system.

Sampling Delay
Compression Delay

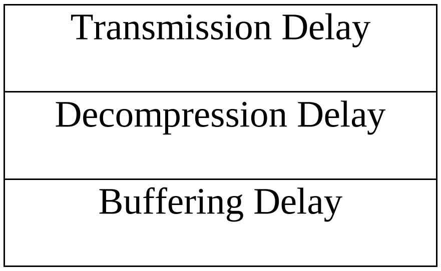


Figure 1.7: Types of delay

The sampling delay is the delay caused by the sampling of the voice signal. If the sampling interval is one second, the total delay will be at least one second since no processing will take place until the sample is ready for manipulation. Keeping this delay as small as possible means shortening the sampling period, on the other hand, this will increase the required data rate. The compression and decompression delays are caused by compression and decompression algorithms respectively. The transmission delay is the delay which is caused by link capacities. Studies made by (Young et al. 2002) shows that when the delay exceeds 800 ms, a normal telephone conversation becomes very hard to do. Besides that; a delay of 200 to 800 ms is tolerable for short portions of the communication. However, in general a delay below 200 ms has got to be attained to hold a pleasant conversation.

Buffering delay was previously introduced. (Cisco Systems 2002)

1.3.5 Voice Coding Techniques in VoIP Systems

In order to reduce bandwidth in the transmission of speech signal, speech coding is employed to compress the speech signals. In general, speech coding techniques are divided into three categories: Waveform coders, Vocoder and Hybrid coders.

- Waveform coders: only explore the correlation in time-domain and frequency- domain and attempt to preserve the general shape of the signal waveform. e.g. G.711 PCM (64Kb/s) and G.726 ADPCM (40/32/24/16 Kb/s). (Cisco Systems 2002)
- Voice coders (vocoders): based on simple (voiced/unvoiced) speech production model and no attempts are made to preserve the original speech waveform. The speech is synthetic. e.g. 2.4/1.2 Kb/s LPC.(Cisco Systems 2002)
- Hybrid coders: incorporate the advantages of waveform coders and vocoders to achieve good speech quality at 4.8 to 16 Kb/s. Includes all the modern codecs, e.g. G.729 CSACELP (8Kb/s), G.723.1 MP-MLQ/ACELP (6.3/5.3 Kb/s), AMR (Adaptive Multi-Rate, ACELP) and iLBC (Internet Low Bit Rate Codec). (Ohrtman 2004) (Chu 2003)

1.4 Speech Processing as a Stochastic Process and Its Models

Speech is usually modeled as a random process and some of its properties can be captured using a simple model. Being able to represent speech according to that model and estimating the model's parameters will allow us to deduct related information using alternative media.

The power spectral density has an essential role in the speech processing methods. Depending on the fact that human auditory system relies heavily on the power distribution in the frequency domain, many methods were developed in analyzing this distribution. The following is building up the concepts for both spectrum estimation and the concept of the Periodogram, parametric method of spectral estimation

1.4.1 The Power Spectral Density

The average power of a deterministic signal $x[n]$ is given by

$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} \left(|\, x[n]\,|\right)^{2} \tag{1.1}$$

Defining the function over an interval N

$$x_N[n] = \begin{Bmatrix} x[n], |n| \le N \\ 0, otherwise \end{Bmatrix} \tag{1.2}$$

expresses the average power of the signal as the area covered in the spectrum (integration of the Fourier transform of the signal), the following equation is otained by using the Parseval theorem. (Oppenheim et al. 2000)

$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} \left(|x[n]|\right)^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} \lim_{N\to\infty} \left(\frac{|X_N\left(e^{j\omega}\right)|^2}{2N+1} \right) d\omega \tag{1.3}$$

Where

$$x_N[n] \xleftrightarrow{F} X_N\left(e^{j\omega}\right) \tag{1.4}$$

Fourier transform pairs are defined in the following two equations

$$X_N\left(e^{j\omega}\right) = \sum_{-\infty}^{\infty} x_N[n] e^{-j\omega n} \tag{1.5}$$

$$x_N[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X_N(e^{j\omega}) e^{j\omega n} d\omega \tag{1.6}$$

1.4.2 Average Power of a Stochastic Process

For a stochastic process $x[n]$, (1.3) represents a one sample power realization. For the signal intended we have to take the expectation of the signal to obtain the average power P. The average power of a random signal is expressed as follows :

$$P = \frac{1}{2\pi} \int_{-\pi}^{\pi} \lim_{N \to \infty} \left(\frac{E\left\{ | X_N \left(e^{j\omega} \right) |^2 \right\}}{2N+1} \right) d\omega, \tag{1.7}$$

Where E{.} is the expectation operator.

1.4.3 Definition of the Power Spectral Density

The power spectral density (PSD) function $S\left(e^{j\omega}\right)$ of a stochastic process is defined by :

$$P = \frac{1}{2\pi} \int_{-\pi}^{\pi} S\left(e^{j\omega}\right) d\omega, \tag{1.8}$$

This indicates the area of the spectrum spreading in the frequency domain. Where P is the average power of the stochastic signal x[n]. comparing (1.7) to (1.8) leads to the following equation that represents the PSD:

$$S\left(e^{j\omega}\right) = \lim_{N \to \infty} \left(\frac{E\{| X_N (e^{j\omega}) |^2\}}{2N+1} \right) \tag{1.9}$$

1.4.4 Periodogram

For the sake of simplicity in the mathematical representation, speech signals are introduced as wide sense stationary signal (WSS) since it is easier to trace such equation because thy have a constant average power. In addition to the properties of the WSS signal and its processing advantages might be applicable.

A very important note is to be mentioned is that; in order to have the speech signal as a stationary signal we should frame the signal into small periods of time.

We are interested in expressing the signal power in term of its autocorrelation function, so we can go on with our calculations by masking the signal power spectral density (PSD) by another value, which is deduced by the signal autocorrelation function.

Our intention in this case is to express the PSD function in term of the signal autocorrelation function. Referring to (Chu 2003) we can see that the Fourier transform of the average of the autocorrelation function represents the Fourier transform pairs of the PSD function. With a little modification in the autocorrelation function, we can directly express the PSD function as a

Fourier transform of the autocorrelation function values at specific points. (Chu 2003)

Now, we are able to express our signal in the frequency domain by taking the Fourier transform of its autocorrelation function, since we considered our signal as a stationary signal.

Consider the N point sequence $x[n]$ $, n = 0, \cdots\cdots, N-1$. Then; the Periodogram $I_N\left(e^{j\omega}\right)$ is defined as

$$I_N\left(e^{j\omega}\right) = \frac{1}{N} | X_N\left(e^{j\omega}\right)|^2 \tag{1.10}$$

Where

$$X_N\left(e^{j\omega}\right) = \sum_{n=0}^{N-1} x[n]e^{-j\omega n} \tag{1.11}$$

is the discrete Fourier transform of the finite sequence $x[n]$. When the finite sequence is selected through a window sequence $w[n]$, that is

$$X_N\left(e^{j\omega}\right) = \sum_{n=0}^{N-1} w[n]x[n]e^{-j\omega n} \tag{1.12}$$

the resultant frequency function by definition is known as the modified Periodogram

Theorem (1.1) *We are given the* N *points real sequence* $x[n]$, $n = 0,\cdots\cdots, N-1$ *then*

$$I_N\left(e^{j\omega}\right) = \sum_{l=-(N-1)}^{(N-1)} R[l]e^{-j\omega l} \tag{1.13}$$

With
$$R[l] = \frac{1}{N}\sum_{m=0}^{N-1} w[m+1]w[m]x[m+l]x[m] \tag{1.14}$$

being the autocorrelation function of the sequence w[n]x[n]. Thus, the Periodogram is related to the autocorrelation function with respect to the Fourier transform.

1.4.5 Autoregressive Model

Since the power of the output signal (PSD) is represented by a constant noise spectrum multiplied by the square magnitude of the filter, this means that the variable that controls the power of the output signal is the values of the filter magnitudes, i.e. dominator of polynomial that represents the filter, a further elaboration of this concept can be found in the next chapter. (Chu 2003) (Kondoz 1994) (Rabiner et al. 1996)

The representation of such sequence $x[n]$ as an autoregressive process satisfies the following difference equation :

$$x[n] = -a_1 x[n-1] - a_2 x[n-2] - \cdots - a_M x[n-M] + v[n] \tag{1.15}$$

Where $x[n-1], \ldots x[M-1]$ are the auto regression parameters and $v[n]$ represent the white noise process.

The value of $x[n]$ is equal to a linear combination of the past values of the process $x[n-1], \ldots x[M-1]$, plus the error term $v[n]$, the process $x[n]$ is said to be regressed on $x[n-1], \ldots x[M-1]$; in particular, $x[n]$ is regressed on previous values of itself, therefore the name is autoregression (AR). (Chu 2003)

As we can see here a linear representation of the signal, although there is a new research direction that adapts a nonlinear representation of the signal with a slightly high order of the system, but this is beyond our research objective. (Kumar et al. 1997)

1.4.6 System Function of the AR Analyser

Taking the Z transform of the general AR equation (1.15) will get

$$H_A(z) = \frac{V(z)}{X(z)} = \sum_{i=0}^{M} a_i z^{-i} \tag{1.16}$$

Where $H_A(z)$ is the system function of the AR analyzer, which is a digital filter, noting that a_0 is equal to 1 in the previous equation. (Chu 2003) (Kondoz 1994)

1.4.7 System Function of the AR Process Synthesizer

With the white noise $v[n]$ acting as input, we can use the system given by

$$H_S(z) = \frac{V(z)}{X(z)} = \frac{1}{H_A(z)} = \frac{1}{\sum_{i=0}^{M} a_i z^{-i}} \qquad (1.17)$$

as a synthesizer for the AR signal. This process is an inversion process since the previous equation is an all zero filter (FIR) and the synthesizing filter is an all pole (IIR). The synthesizer takes white noise as an input in order to produce an output that represents an AR signal. Since both filters are inverse for each other, they can be represented as the following:

$$H_S(z) = \frac{1}{(1-p_1 z^{-1})(1-p_2 z^{-1}).....(1-p_M z^{-1})} \qquad (1.18)$$

Where $p_{1,\dots}, p_M$ are the poles of $H_S(z)$ and the roots of the characteristic equation

$$1+a_1 z^{-1}+a_2 z^{-2}+.......+a_M z^{-M}=0 \qquad (1.19)$$

Synthesizing of the AR process is always done by filtering white noise signal by an all pole filter. (Chu 2003)

1.4.9 The Power Spectral Density and the AR Process

The AR process is an output of an LTI system, the previous theorem tells us a fact about the power of such a signal in such a system, and it is characterized by the function $H_S(z)$. when the input of the system is white noise, PSD is a constant equals to σ_v^2,the variance of the input signal $v[n]$, for the signal $x[n]$ the output PSD of the output AR process is given by

$$S_x\left(e^{j\omega}\right)=|H_S\left(e^{j\omega}\right)|^2\,\sigma_v^2 \qquad (1.20)$$

Which is a multiplication of the square value of the frequency response and the variance of the input signal.

1.4.10 The Normal Equation

Since v[n] represents white noise sample at time instance n. it is not correlated with x[n-1] for $1 \leq l$. That is

$$E\{v[n]x[n]\} = \sigma_v^2 \qquad (1.21)$$

This is the cross correlation between $x[n]$ and $v[n]$ is given by the variance of $v[n]$.Multiplying both sides of (1.15) gives the system of equations

$$\begin{aligned} &R_x[0] + a_1 R_x[1] + ... + a_M R_x[M] = \sigma_v^2 \\ &R_x[1] + a_1 R_x[0] + ... + a_M R_x[M-1] = 0 \\ &\qquad\qquad\vdots \\ &R_x[M] + a_1 R_x[M-1] + ... + a_M R_x[0] = 0 \end{aligned} \qquad (1.22)$$

Or the matrix of the form

$$\begin{pmatrix} R_x[0] & R_x[1]\ldots & R_x[M] \\ R_x[1] & R_x[0] & R_x[M-1] \\ \vdots & \ddots & \vdots \\ R_x[M] & R_x[M-1]\cdots & R_x[0] \end{pmatrix} \begin{pmatrix} 1 \\ a_1 \\ a_2 \\ .. \\ .. \\ a_M \end{pmatrix} = \begin{pmatrix} \sigma_v^2 \\ 0 \\ 0 \\ .. \\ .. \\ 0 \end{pmatrix} \qquad (1.23)$$

This equation is known as the normal equation of the WSS AR processes. Given the auto correlation sequence $R_x[0], R_x[1], \ldots R_x[M]$, the system in (1.23) can be solved to obtain the model parameters a_i.

1.4.11 The Autocorrelation Estimation

The estimation of the inputs of the system (1.23) is a key element in the linear prediction coding, no matter what the application or the dimension was, we should obtain this component of the model to be able to go on with our solution.

Since we are moving toward an WSS signal representation, and we know that we are obtaining this with our signal only when we look to the signal through a window, the autocorrelation function has to be estimated for the system every short interval of time, meaning that, the system of equations will be repeated every short interval of time. Fundamentally, there exist two approaches to estimate the auto correlation function: the nonrecursive method and the recursive method. The first one uses a finite length function to use as a window for the system, and the second method uses an infinite function to express the window. (Chu 2003)

The autocorrelation function of a real discrete time signals $x[n]$ at a lag l is defined as:

$$R_x[l] = A\{x[n]x[n+l]\} = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x[n]x[n+l] \tag{1.24}$$

The following will be an implementation of some estimators.

Examples of the non recursive estimation:

1. Hamming Window: In this case, a function defined in a closed interval is used as w[n], the function is defined by the following function

$$w[n] = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{N-1}\right), 0 \le n \le N-1, \\ 0, otherwise \end{cases} \qquad (1.25)$$

2. Rectangular window
3. Trapezoidal window

Examples of the recursive window

1. Barnwell window, the signal is windowed by a function with an infinite length of sequence as a window, the estimator is expressed as:

$$w[n] = (n+1)\alpha^n u[n] \qquad (1.26)$$

This is an infinite window function as we can see where α is a positive constant and u[n] is the unit step function. The z transform of the window of given by

$$W(z) = \frac{1}{(1-\alpha^{-1})^2} \tag{1.27}$$

2. Chen Window, This window is a combination between recursive and non recursive. The window function is defined by

$$w[n] = \begin{cases} 0, n \le 0, \\ \sin(cn), 0 \le n \le L, \\ b\alpha^{n-L-1}, L+1 \le 1, \end{cases} \tag{1.28}$$

Where L is the length of the non recursive section of the window and α , b, and c are the constants that must be found for a particular window specification. For further information and how the parameters should be selected, refer to (Chu 2003). This is the basic knowledge needed to be familiar with the autocorrelation estimation, the $w[n]$ is the window function in general, the estimation process can be found in (Chu 2003). The shape of the used function and the series bound used in estimating the window specified the type of the estimation. Weather it is finite of not, all of this is supposed to serve the normal equation of 1.15.

Chapter Two

Linear Prediction

2.1 Audio Signal Representation Methods

2.1.1 Time Domain Representation

This method is the most basic method to represent a signal, which is seen as an analogue value that is measured with time. In general, most of the important information can not be seen from this prospectus; some basic information only can be obtained from this representation. Most of the signal processing methods do not prefer to use this representation due to the importance of having the signal representable in a mathematical formula. Most of the time, it is very hard to obtain such formula for most of the signals. (Oppenheim et al. 2000)

2.1.2 Transfer Domain Representation

This is the process of mapping the function values into another domain meaning changing there corresponding values according to a kernel function that rules the domain to range mapping process from the source domain to the destination domain. With this process, additional information might be obtained about the signal when it is moved to the new domain. There are several famous transfer functions, like

- Laplace Transform.
- Fourier Transform.
- Discrete Cosine transform.

A very good result in general is obtained from the transfer functions, that is, we are able to see the real parameters that we are concerned about. The most important component of the signal is the Power, this cannot be seen from the time domain.

2.1.3 Digital Filter Representation

In a very special case, which depends on the signal type, meaning its frequency ranges and its power spectrum, a signal can be seen as an output of a digital filter. This came from the fact that some signals might be written in term of linear difference equations, and the parameters of this equation might be determined from the time domain information. This is the very basic idea that has led to the new science of the linear prediction coding. (Chu 2003)

2.2 The LPC Model

The Linear Prediction Coding model (LPC) is a way of finding some model to express the system. Before we go into the details of the LPC model, we should have some background about human vocal system.

2.2.1 Physical System

Humans speak with the following mechanism

- Air is pushed out through the tract where the voice is initiated, and then shaped along the way as we desire.
- A combination of these operations in the previous point forms the basic sounds that form the sounds of letters and then words and so on.

- Sounds that are empowered by the vibrations are known as voiced sounds, the others are unvoiced sounds.
- For voiced sound, vocal cords vibrate. The rate at which the vocal cords vibrate determines the pitch of your voice. Women and young children tend to have high pitch, i.e., fast vibration, while adult males tend to have low pitch, i.e., slow vibration.
- For unvoiced sound, your vocal cords do not vibrate but remain constantly opened.
- The shape of your vocal tract determines the sound that you make.
- As you speak, your vocal tract changes its shape producing different sounds.
- The shape of the vocal tract changes relatively slowly (on the scale of 10 msec. to 100 msec.).
- The amount of air coming from your lungs determines the loudness of your voice. (Chu. 2003)

The following figure shows the components of human vocal system

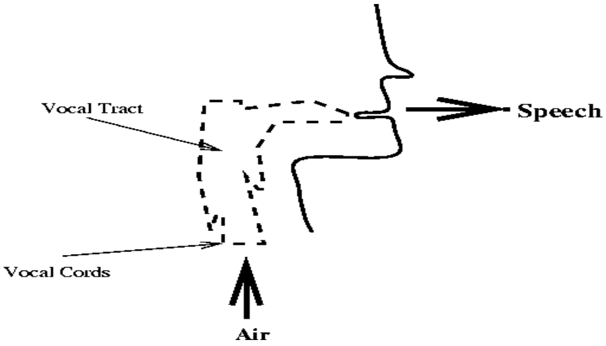


Figure 2.1: Human Vocal System

2.2.2 Mathematical Modeling

The representation of the human vocal system is expressed as the LPC Model, figure 2.2 shows the model

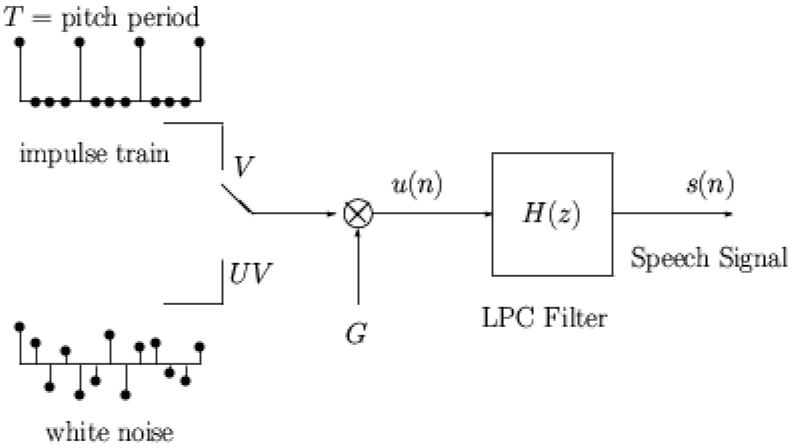


Figure 2.2: the LPC Model

LPC Model Components

- Voice is either voiced or unvoiced, V or UV.
- Voiced sounds are generated as an impulse train with known frequency T (pitch).
- unvoiced sounds are generated from white noise, obtained from white noise generator.
- Speech loudness (which is obtained by the amount of air we are pushing) is expressed as the Gain G.
- The combination of either voiced or unvoiced information is gained by G value resulting in the signal u(t), which is implemented in its turn to digital filter (LPC Filter).
- The digital filter of order M output expresses s(n), which is a very acceptable version of the input signal. (Chu 2003) (Kondoz 1994)

This is the explanation of figure 2.2, marking the main components of the LPC Model. But how do we obtain these parameters in order to have a model that describes the human physical model. We are entitled to deduce the following parameters:

1. The voicing information, whether it is voiced or not.

2. In the case of voiced. Specify the pitch (sometime expressed as formant or the fundamental frequency).
3. The gain of the signal.
4. The digital filter coefficients that will synthesis the signal.
5. And finally; how can we do all of this with the least possible error, respecting the classic constrains: quality, complexity, over all delay?.

2.3 Pitch Period Estimation

2.3.1 Frequency Domain Pitch Period Estimation

Cepstral Pitch Estimation

Cepstral Analysis provides a way for the estimation of pitch. we assume that a sequence of voiced speech is the result of convoluting the excitation sequence e[n] with the vocal tract discrete impulse response h[n]. In frequency domain, the convolution relationship becomes a multiplication relationship. Then, using property of log function log AB = log A + log B, the multiplication relationship can be transformed into an additive relationship. The cepstrum of the input signal, namely the real part expresses the pith of the speaker. The real cepstrum of a signal s[n] = e[n]* h [n] is defined as (Chou 2003)

$$
\begin{aligned}
&c[n] = F_{DTFT}^{-1}\left\{\log|F_{DTFT}\left\{s[n]\right\}|\right\} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\log|S(\omega)|e^{j\omega}d\omega \\
&where \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (2.1) \\
&S(\omega) = \sum_{-\infty}^{\infty} s[n]e^{-j\omega n}
\end{aligned}
$$

2.3.2 Time Domain Pitch Period Estimation

The fact that variations in voiced signals are so evident suggests that time domain techniques should be capable in detecting pitch period of voiced signals. Most of the time domain pitch period estimation techniques use autocorrelation function (ACF).

Properties that make ACF an attractive basis for estimating periodicities in all sorts of signals, including speech are:

- It is an even function.
- The quantity at lag 0 equals the energy for deterministic signals or the average power for random or periodic signals. This is attained by the use of small interval to have a stationary signal as mentioned in the previous chapter.

Following are the three time domain pitch estimation and detection.

2.3.2.1 Autocorrelation Method (ACM)

One of the major limitations of the autocorrelation representation of the voice is that it retains too much of the information in the speech signal. Most of the peaks in the ACF can be attributed to the damped oscillations of the local tract response, which are responsible for the shape of each period of the speech wave. Also, if the window is too short compared to the pitch period, a false pitch period estimation might occur.

Thus, in cases the autocorrelation peaks are being counted due to the vocal tract, the period of the highest peeks is expressed as the pitch, but when the response does not match the periodicity of the vocal excitation, meaning that there is high response but no periodicity can be dedicated the simple procedure of picking the largest peak in the autocorrelation function will fail. (Rabiner et al. 1996)

2.3.2.2 Center clipping Autocorrelation Method (CC-ACM)

An improvement of ACF method is CC-ACM. It belongs to the group of spectrum flattening techniques. A segment of speech to be used in computing an ACF is preprocessed by passing signal through a clipper. The clipping value is determined as 60% of the minimum of maximum amplitudes in first and last third part of the signal, this means dropping off the unnecessary information calculations seen in the ACM. (Rabiner et al. 1996) (Hess 1983)

2.3.2.3 Modified Autocorrelation Method (MACM)

The conflicting requirements in choosing optimal window size (N) in pitch detection exist. Because of the changing properties of the speech, N should be as small as possible. On the other hand, it should be clear that to get any indication of periodicity in the autocorrelation function, the window must have a duration of at least two periods of the waveform. In order to solve this conflicting requirement, modified short time ACF is used:

$$\hat{R}_x(k) = \sum_{m=0}^{N-1} x(n+m)x(n+m+k), 0 \le k \le K \qquad (2.2)$$

Where K is the greatest lag of interest. (Rabiner et al. 1996)

2.4 Two state LPC vocoder synthesizer

Speech can be synthesized from the linear prediction parameters. Figure 2.2 shows a block diagram of speech synthesizer. The time varying control parameters needed by the synthesizer are the pitch period, a voiced/unvoiced switch, gain, and predictor coefficients. The impulse generator acts as the excitation source for voiced sounds producing a pulse of unit amplitude at the beginning of each pitch period. The white noise generator acts as the excitation source for unvoiced sounds producing uncorrelated samples with zero mean Gaussian distribution. The selection between two sources is made by the voiced/unvoiced control. The gain control determines the overall amplitude of the excitation. (Chu 2003) (Hess 1983) (Rabiner et al. 1996)

2.5 The Digital Synthesis Filter

The filters

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 + \sum_{1}^{M} a_i z^{-i}} \qquad (2.3)$$

and

$$A(z) = 1 + \sum_{1}^{M} a_i z^{-i} \qquad (2.4)$$

are very essential to the LPC operation, since the first one has all poles values when represented in the Z domain, and the other has the values of only zeroes. The relation between these two filters is any of them is the inverse function for the other, where the constant M is the order of the filter, and a_i represents the filter's coefficients. M sometimes is expressed as the predictor order. And another term is used for the a_i's is the linear prediction coefficients. The first filter is a finite impulse response filter (FIR) and the second is an infinite impulse response filter (IIR), but there is a method to approximate the infinite filter to a finite form of representation. (Yamamoto et al. 2003)

There are some other filter structures that are used for this purpose, called the modified filter structures that do not follow the strictness of (2.3) and (2.4). (Harma 2001)

These filters are expressed in different realizations. The following will be about the direct form realization and the Lattice realization.

2.5.1 Direct Form Realization

Applying a signal $x[n]$ as an input to the filter and having $y[n]$ as output, the time domain difference equation for (2.3)

$$y[n] = x[n] - \sum_{1}^{M} a_i y\left[n-i\right] \tag{2.5}$$

And for (2.4)

$$y[n] = x[n] + \sum_{1}^{M} a_i x\left[n - i\right] \tag{2.6}$$

Signal Flow Graphs

The signal flow graph that represents this equation is seen in figure 2.3 and 2.4. Note that the feedback process in this type of filters cause an infinite non trivial impulse response for the filter (IIR), since the output of the system is added back to its input, thus the all zero filter has a finite impulse response (FIR), due to the fact that the output of the system is a combination of its current input and the previous values.

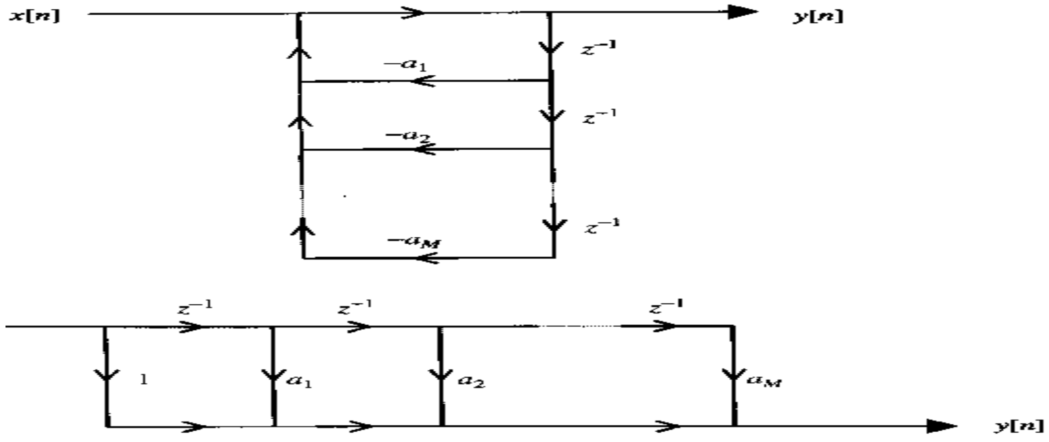


Figure 2.3: Signal Flow Graph of Direct Form IIR Filter (top) , FIR(bottom)

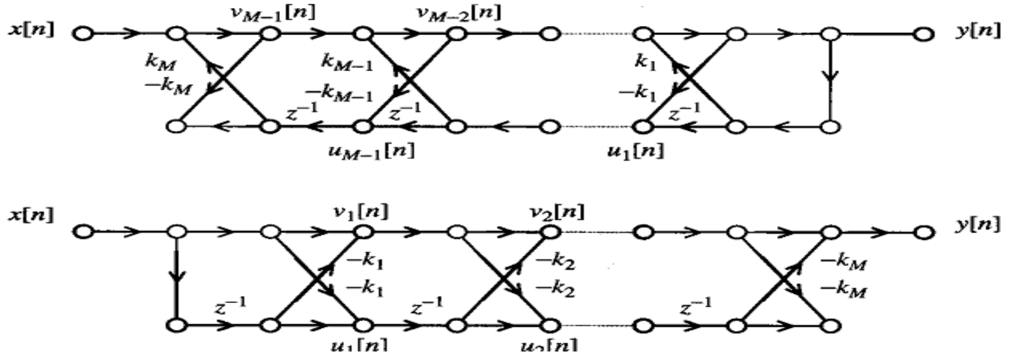


Figure 2.4: Signal Flow Graph of Lattice, FIR (bottom) (Chu 2003)

2.5.2 Lattice Realization

The Lattice realization in figure 2.4 is often referenced as the *k* realization and the direct form is referenced as the *a* realization. The lattice realization is expressed by the coefficients *k* which is recognized as the reflection coefficients. These coefficients can be deducted from the *a* coefficients through the computational loop specified below. (Chu Wai C. 2003) (Kondoz 1994)

$$\begin{aligned}
&\text{For } l = M, M-1, \ldots, 1: \\
&k_l = -a_l, \\
&a_i^{l-1} = \frac{a_i^l + k_l a_{l-i}^l}{1-k_l^2} \cdots i = 1, 2, \ldots, l-1.
\end{aligned} \tag{2.7}$$

2.5.3 Comparison between the two realizations

The direct form realization is preferred sometimes due to its simplicity and lower computing complexity, on the other hand lattices realization coefficients are the first coefficients to be obtained from the recursion used to solve the equations systems, this will be explained in more details in the Levinson-Durbin recursion section, the first values obtained from each recursion in the Levinson-Durbin recursion are the k coefficients. A further step of processing is required in order to obtain the LPC coefficients, i.e. the direct form coefficients. In addition to this, we do not have to worry about the stability of the system at any point with a lattice realization, this means that the values of the coefficients will lie on the real axis of the Z domain within the unit circle, the instability of the system will cause the filter output will go in saturation meaning the loss of the signal values. (Chu C. 2003)

2.6 The Linear Prediction

We have reviewed the important concepts of the speech signals processing techniques, and the characteristics of this signal; and the use of the AR model as a representation of the synthesized signal, basically; the linear predication is the process estimating the AR model parameters. (Rabiner 1996)

2.6.1 The Fundamental Problem of the Linear Prediction

The linear prediction process solves the problem of calculating the AR parameters of a signal, and use them in the synthesis process (the retrieval) of the signal $s[n]$, the white noise signal $x[n]$ is filtered by the AR synthesizer to obtain $s[n]$, this is done by the following equation

$$\hat{s}(n) = -\sum_{i=1}^{M} a_i s[n-i] \qquad (2.8)$$

Where the a_i are the estimate of the AR parameters and referred to as the LPC coefficients, and the constant M is the prediction order. So the predication is based on a linear combination of the M past samples, and the prediction error is expressed as

$$e[n] = s[n] - \hat{s}[n] \qquad (2.9)$$

Where $s[n]$ is original signal and $\hat{s}[n]$ is the prediction signal, figure 2.5 shows the relation between the original and the predicted signal (Chu Wai C. 2003).

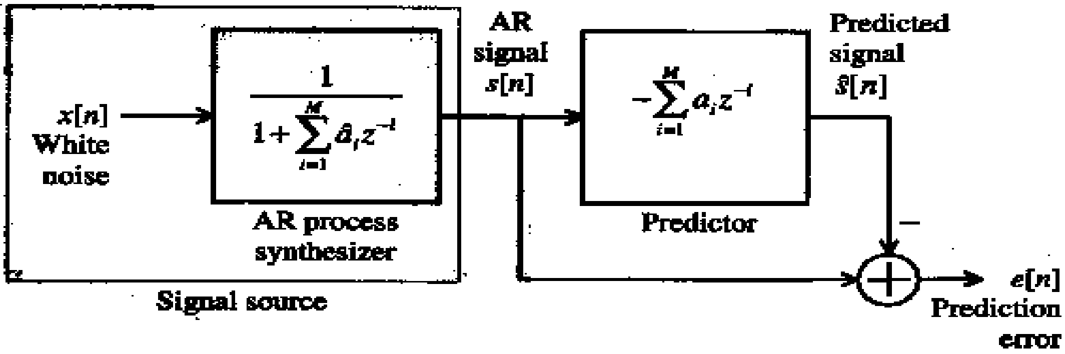

$x[n]$
White noise
$\frac{1}{1+\sum_{i=1}^{M}\hat{a}_i z^{-i}}$
AR process synthesizer
Signal source
AR signal $s[n]$
$-\sum_{i=1}^{M} a_i z^{-i}$
Predictor
Predicted signal $\hat{s}[n]$
−
$e[n]$ Prediction error

Figure 2.5: Linear Prediction System Identification. (Chu 2003)

2.6.2 Error Minimization

The system identification problem consists of the estimation of the AR parameters $\hat{a}_i$ from s[n], with the synthesis of the LPC, a criterion must be established, which is the mean prediction error, which is expressed as

$$J = E\{e^2[n]\} = E\left\{\left(s[n] - \sum_{i=1}^{M} a_i s[n-i]\right)^2\right\} \tag{2.10}$$

This term is minimized by selecting the appropriate LP coefficients, where the cost function J is precisely a second order function of the LPC and the dependency of the function on the order M and the coefficients, the optimal LPC can be found by the partial derivative of J against a_i and equate it to zero, we obtain

$$\frac{\partial J}{\partial a_k} = 2E\left\{\left(s[n] + \sum_{i=1}^{M} a_i s[n-i] \right)\ s[n-k]\right\} = 0 \qquad (2.11)$$

This equation is verified over M points, we are aiming to have $a_i = \hat{a}_i$, at this point the LP Coefficients are equal to the AR parameters . (Chu Wai C. 2003)

2.6.3 The Normal Equation

The partial derivation equation can be given as

$$E\{s[n]s[n-k]\} + \sum_{i=1}^{M} a_i E\{s[n-i]s[n-k]\} = 0, \qquad (2.12)$$

Or

$$\sum_{i=1}^{M} a_i R_s[i-k] = -R_s[k] \qquad (2.13)$$

For $k = 1, 2, \ldots, M$ where

$$R_s[i-k] = E\{s[n-i]s[n-k]\} \qquad (2.14)$$

$$R_s[k] = E\{s[n]s[n-k]\} \qquad (2.15)$$

Equation (2.14) defines the optimal LPC in the term of the autocorrelation function $R_s[l]$ of the signal $s[n]$, in matrix form this will become

$$\mathrm{R}_s a = -\mathrm{r}_s \tag{2.16}$$

Where

$$\mathrm{R}_s = \begin{pmatrix} R_s[0] & R_s[1]\ldots & R_s[M-1] \\ R_s[1] & R_s[0] & R_s[M-2] \\ \vdots & \ddots & \vdots \\ R_s[M-1] & R_s[M-2]\cdots & R_s[0] \end{pmatrix} \tag{2.17}$$

$$\mathrm{a} = [a_1 \quad a_2 \quad ..... \quad a_M]^T \tag{2.18}$$

$$\mathrm{r_s} = [R_s[1] \quad R_s[2] \quad ..... \quad R_s[M]]^T \tag{2.19}$$

Equation 2.18 is known as the normal equation, assuming that the inverse of the correlation matrix R_s exists, the optimal LPC vector is obtained with

$$\mathrm{a} = -\mathrm{R}_s^{-1}\mathrm{r_s} \tag{2.20}$$

2.6.4 Prediction Gain

The prediction gain of the predictor is given by

$$PG = 10\log_{10}\left(\frac{E\{s^2[n]\}}{E\{e^2[n]\}}\right) \tag{2.21}$$

Which is the ratio between the variance of the input signal and the prediction error. The log represents the value in decibels. The good predictor is the one that can achieve the required gain with the least possible error. (Rabiner 1996)(Chu 2003)

2.6.5 Minimum Mean-Squared Prediction Error (MMSPE)

Recall (2.14), we can see that $a_i = \hat{a}_i, e[n] = x[n]$; means that, the prediction error is the same as the white noise used to generate the AR signal $s[n]$. Indeed, this is the optimal situation where the mean-squared error is minimized, with

$$J_{\min} = E\{e^2[n]\} = E\{x^2[n]\} = \sigma_x^2 \tag{2.22}$$

Equivalently, the prediction gain is maximized. The optimal case starts to be reached when the order of the synthesizer is at least equal to the order of the AR process, a good way to be able to know a good order of the system is investigating the system gain saturation. This is achieved by plotting the system gain as a function of the system order, at a certain point, the increasing of the system order, at the beginning of this case, we have M.

Reading the function (2.14) will lead to the recognition of its unique minimum value. If the predictor order is known, then the cost function is minimized when $a_i = \hat{a}_i$, leading to $e[n] = x[n]$. This means that the prediction error is equal to the excitation signal of the AR process synthesizer. The reasonability of this comes from the fact that; the best thing that a filter can do is to whiten the AR signal $s[n]$, thus, the maximum prediction gain is given by the ratio between $s[n]$ and the variance of $x[n]$in decibels. Taking into account the AR parameters used to generate the signal $s[n]$, we have

$$J_{\min} = \sigma_x^2 = \sum_{i=1}^{M} a_i R_s[i] \tag{2.23}$$

Now we would like to use the matrix form of the equation that we have seen in (2.18) to have

$$\begin{pmatrix} \mathrm{R}_s & \mathrm{r}_\mathrm{s}^{-1} \\ \mathrm{r}_\mathrm{s} & \mathrm{R}_s \end{pmatrix} \begin{pmatrix} 1 \\ \mathrm{a} \end{pmatrix} = \begin{pmatrix} J_{\min} \\ 0 \end{pmatrix} \tag{2.24}$$

This form is known as the augmented normal equation, with 0 the Mx1 Zero vector.

Equation (2.22) can also be written as

$$\sum_{i=0}^{M} a_i R_s[i-k] = \begin{Bmatrix} J_{\min}, \ldots k = 0 \\ 0, \ldots\ldots k = 1, 2, \ldots., M \end{Bmatrix} \tag{2.25}$$

Where a_0 =1.

2.7 The Levinson-Durbin Algorithm

Consider the following linear system

$$\begin{pmatrix} R[0] & R[1]\ldots & R\ [M\ ] \\ & & \\ R[1] & R[0] & R[M\ -1] \\ \vdots & \ddots & \vdots \\ & & \ddots \\ R\ [M\ ] & R[M\ -1]\cdots & R\ [0] \end{pmatrix} \begin{pmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ \vdots \\ a_M \end{pmatrix} = \begin{pmatrix} J \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \qquad (2.26)$$

With the objective solution being the solution of the LPC $a_i, i = 1, 2, \cdots, M$, given the autocorrelation values $R\left[l\right], l = 0,1,\cdots,M$, and J represents the minimum mean square error, or the variance of the input signal, which is a white noise that is applied as input to the synthesizer, the autocorrelation value is calculated from the signal samples, and J is usually unknown . This algorithm finds the solution of Mth order system from $(M-1)th$ system. This is a recursive operation which walks on a line of order reduction, until it reaches the order zero, the value of order zero is first found and then use this value in solving the higher value solution and so on, till we reach the solution of the order M which is the objective of this recursion.

The algorithm relies on two main properties of the matrix

- The correlation matrix of a given size contains as the sub block all the lower order correlation matrices

- If $$\begin{pmatrix} R[0] & R[1]\dots & R\,[M] \\ R[1] & R[0] & R[M-1] \\ \vdots & \ddots & \vdots \\ R\,[M] & R[M-1]\cdots & R\,[0] \end{pmatrix} \begin{pmatrix} 1 \\ a_1 \\ a_2 \\ .. \\ .. \\ a_M \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ \vdots \\ \vdots \\ b_M \end{pmatrix} \tag{2.27}$$

- then $$\begin{pmatrix} R[0] & R[1]\dots & R\,[M] \\ R[1] & R[0] & R[M-1] \\ \vdots & \ddots & \vdots \\ R\,[M] & R[M-1]\cdots & R\,[0] \end{pmatrix} \begin{pmatrix} a_M \\ a_{M-1} \\ .. \\ .. \\ a_0 \end{pmatrix} = \begin{pmatrix} b_M \\ b_{M-1} \\ \vdots \\ \vdots \\ \vdots \\ b_0 \end{pmatrix} \tag{2.28}$$

The correlation matrix is invariant under interchange of its columns as then its rows. The mentioned properties are direct consequence of the fact that the correlation matrix is a Teoplitz , this is because it is square and its diagonal elements are equal, also because all the parallel lines to the diagonal aligns equal elements. (Rabiner 1996)

We consider the solution of the augmented normal equation starting from zero prediction order. It is shown that the solution of a certain order can be obtained from the result of prediction a lower order results; and so on.

2.7.1 Predictor of Order Zero

In this case consider the equation

$$R\,[0] = J_0 \tag{2.29}$$

This equation is already solved, this relation states that the MMSPE is achievable with zero order predictor is given by the auto correlation of a signal at lag zero, or the variance of the signal itself, meaning the MMSPE is equal to the signal itself.

By expanding (2.29) we obtain the next dimension

$$\begin{bmatrix} R\,[0] & R\,[1] \\ R\,[1] & R\,[0] \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} J_0 \\ \Delta_0 \end{bmatrix} \tag{2.30}$$

Which is a two dimensional version of the fundamental matrix, with a_1=0. Since a_1=0 the optimal condition cannot be achieved in general, and the term Δ_0 is introduced on the right hand side to balance the equation, this quantity is found from the equation as

$$\Delta_0 = R\,[1] \tag{2.31}$$

Forming the property of the matrix, the equation becomes

$$\begin{bmatrix} R\,[0] & R\,[1] \\ R\,[1] & R\,[0] \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \Delta_0 \\ J_0 \end{bmatrix} \tag{2.32}$$

2.7.2 Predictor of Order One

We now seek to solve

$$\begin{bmatrix} R\,[0] & R\,[1] \\ R\,[1] & R\,[0] \end{bmatrix}\begin{bmatrix} 0 \\ a_1^{(1)} \end{bmatrix} = \begin{bmatrix} J_1 \\ 0 \end{bmatrix} \tag{2.33}$$

Where $a_1^{(1)}$ is the LPC predictor; the subscript means we are in predictor on level one, J_1 represents the MMSPE achievable using predictor of order one, unknowns of the system are J_1 and $a_1^{(1)}$. Consider the solution of the form

$$\begin{bmatrix} 1 \\ a_1^{(1)} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - k_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.34}$$

With k_1 being a constant, multiplying both sides by the correlation matrix we have

$$\begin{bmatrix} R\,[0] & R\,[1] \\ R\,[1] & R\,[0] \end{bmatrix}\begin{bmatrix} 0 \\ a_1^{(1)} \end{bmatrix} = \begin{bmatrix} R\,[0] & R\,[1] \\ R\,[1] & R\,[0] \end{bmatrix} - k_1 \begin{bmatrix} R\,[0] & R\,[1] \\ R\,[1] & R\,[0] \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.35}$$

Substituting (2.32), (2. 33) and (2.34) we have

$$\begin{bmatrix} J_1 \\ 0 \end{bmatrix} = \begin{bmatrix} J_0 \\ \Delta_0 \end{bmatrix} - k \begin{bmatrix} \Delta_0 \\ J_0 \end{bmatrix} \tag{2.36}$$

Then

$$k_1 = \frac{\Delta_0}{J_0} = \frac{R\,[1]}{J_0} \tag{2.37}$$

Where (2.36) is used. The LPC of this predictor is readily found from (2.35) to be

$$a_1^{(1)} = -k_1 \tag{2.38}$$

Using (2.36) and (2.37) we find

$$J_1 = J_0 \left(1 - k_1^{\,2}\right) \tag{2.39}$$

Although the first order predictor is specified, the parameter k_1 is know as the reflection coefficient (RC) representing an alternative form from the LPC, note that k_1 was derived from the previous step which is the zero order predictor. As we did in before we can expand the matrix to become

$$\begin{bmatrix} R\,[0] & R\,[1] & R\,[2] \\ R\,[1] & R\,[0] & R\,[1] \\ R\,[2] & R\,[1] & R\,[0] \end{bmatrix} \begin{bmatrix} 1 \\ a_1^{(1)} \\ 0 \end{bmatrix} = \begin{bmatrix} J_1 \\ 0 \\ \Delta_1 \end{bmatrix} \tag{2.40}$$

Where Δ_1 represents an additional term to keep the equation balanced, when the first order predictor is used and $R\,[2] \neq 0$, the quantity is solved as

$$\Delta_1 = R\,[2] + a_1^{(1)} R\,[1] \tag{2.41}$$

2.7.3 Predictor of Order Two

We go on further step by solving

$$\begin{bmatrix} R\,[0] & R\,[1] & R\,[2] \\ R\,[1] & R\,[0] & R\,[1] \\ R\,[2] & R\,[1] & R\,[0] \end{bmatrix} \begin{bmatrix} 1 \\ a_1^{(2)} \\ a_2^{(2)} \end{bmatrix} = \begin{bmatrix} J_2 \\ 0 \\ 0 \end{bmatrix} \tag{2.42}$$

The unknowns in this case are $a_1^{(2)}, a_2^{(2)}$ and J_2 consider the solution of the form

$$\begin{bmatrix} 1 \\ a_1^{(2)} \\ a_2^{(2)} \end{bmatrix} = \begin{bmatrix} 1 \\ a_1^{(1)} \\ 0 \end{bmatrix} - k_2 \begin{bmatrix} 0 \\ a_1^{(1)} \\ 1 \end{bmatrix} \tag{2.43}$$

With k_2 as the RC. Multiplying both sides by the correlation matrix leads to

$$\begin{bmatrix} J_2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} J_1 \\ 0 \\ \Delta_1 \end{bmatrix} - k_2 \begin{bmatrix} \Delta_1 \\ 0 \\ J_1 \end{bmatrix} \tag{2.44}$$

The k_2 can be found from the above equation as

$$k_2 = \frac{1}{J_1}\left(R\,[2] + a_1^{(1)} R\,[1]\right) \tag{2.45}$$

And from the previous matrix

$$a_2^{(2)} = -k_2 \tag{2.46}$$

$$a_1^{(2)} = a_1^{(1)} - k_2 a_1^{(1)} \tag{2.47}$$

Finally J_2 is found from 2.47 and the next one

$$J_2 = J_1\left(1 - k_2{}^2\right) \tag{2.48}$$

For the next step expanding the order two matrixes we have

$$\begin{bmatrix} R\,[0] & R\,[1] & R\,[2] & R\,[3] \\ R\,[1] & R\,[0] & R\,[1] & R\,[2] \\ R\,[2] & R\,[1] & R\,[0] & R\,[1] \\ R\,[3] & R\,[2] & R\,[1] & R\,[0] \end{bmatrix} \begin{bmatrix} 1 \\ a_1^{(2)} \\ a_2^{(2)} \\ 0 \end{bmatrix} = \begin{bmatrix} J_2 \\ 0 \\ 0 \\ \Delta_2 \end{bmatrix} \tag{2.49}$$

Note that

$$\Delta_2 = R\,[3] + a_1^{(2)} R\,[2] + a_2^{(2)} R\,[1] \qquad (2.50)$$

2.7.4 Predictor of Order Three

In this case; consider the solution in the form of

$$\begin{bmatrix} 1 \\ a_1^{(3)} \\ a_2^{(3)} \\ a_3^{(3)} \end{bmatrix} = \begin{bmatrix} 1 \\ a_1^{(2)} \\ a_2^{(2)} \\ 0 \end{bmatrix} - k_3 \begin{bmatrix} 0 \\ a_1^{(2)} \\ a_2^{(2)} \\ 1 \end{bmatrix} \qquad (2.51)$$

Proceeding in similar manner we arrive to the solution. (Chu 2003) (Rabiner et al. 1996)

$$k_3 = \frac{1}{J_2}\left(R\,[3] + a_1^{(2)} R\,[2] + a_2^{(2)} R\,[1]\right) \qquad (2.52)$$

$$a_3^{(3)} = -k_3 \qquad (2.53)$$

$$a_2^{(3)} = a_2^{(2)} - k_3 a_1^{(2)} \qquad (2.54)$$

$$a_1^{(3)} = a_1^{(2)} - k_3 a_2^{(2)} \qquad (2.55)$$

$$J_3 = J_2\left(1 - k_3^2\right) \qquad (2.56)$$

2.8 Types of LPC Excitation

2.8.1 Introduction

The excitation is where the researches have been going for the last twenty years, and it is really confusing and misleading to go into the excitation methods without focusing on the main idea of excitation, which is why do we excite the system, and what are the problems and solutions that excitation offer, the problem of the strict voiced an unvoiced signal arise from the following

- We do not frame the signal according to voiced and unvoiced components; we do it according to time frames, we can not be sure that we have a voiced frame and an unvoiced frame.
- The voiced signal is expressed as in impulse, and experience has shown that it is not entirely accurate to perform such reductions.

The basic thing that the excitation methods agree about is finding the format of the signal, and all of them are trying to reach this part of information considering the following conditions

- having the highest possible range of excitation signals.
- Reaching the most possible accurate spectral envelop (formant).
- Do this with the least possible bits.

- Perform this in the least possible complexity.
- Use less machine resources (time, memory).
- Minimize the error (with respect to SNR types or MOS family).

2.8.2 Codebook Excited Linear Prediction (CELP)

The CELP coder tries to overcome the synthetic sound of vocoders by allowing a wide variety of excitation signals, which are all captured in the CELP codebook. To determine which excitation signal to use, the coder performs an exhaustive search. For each entry in the codebook, the resulting speech signal is synthesized and the entry which created the smallest error is then chosen. The excitation signal is then encoded by the index of the corresponding entry. So basically, the coder uses Vector Quantisation (Chu Wai C. 2003) to encode the excitation signal. The research of these coders most focuses on the delay reductions since we are expecting huge delay due to the VQ related operations. (Zhang et al. 1997)

This technique is called an Analysis-by-Synthesis (AbS) technique, because it analyses a signal by synthesizing several possibilities and choosing the one which caused the least amount of error. (Chu Wai C. 2003) (Kondoz 1994)

This exhaustive search is computationally very expensive. Fast algorithms have been developed to be able to perform the search in real time. This process allowed lower rates, CELP techniques allow bit rates of even 4.8 kbps. (Mitra 1998) (Chu Wai C. 2003) (Kondoz 1994)

There are several methods to go slightly lower down to 2.4 kbps for these codes using some approaches like the variable rate time scale modification if it was applied to the excitation code. (Chong et al. 2003)(Beritelli 1999)(Benjamin 2005)

2.8.3 Residual Excited Linear Prediction (RELP)

The RELP coder works in almost the same way as the LPC coder. To analyze the signal, the parameters for the vocal tract filter are determined and the inverse of the resulting filter is applied to the signal. This gives us the residual signal.

The LPC coder then checked if the signal was voiced or unvoiced and used this to model an excitation signal. In the RELP coder however, the residual is not analyzed any further, but will be used directly as the excitation for speech synthesis. The residual is compressed using waveform coding techniques to lower the bandwidth requirements. RELP coders can allow good speech quality at bit rates in the region of 9.6 kbps. (Mitra 1998) (Chu Wai C. 2003)

2.8.4 Multipulse and Regular Pulse Excited coding (MPE and RPE)

Like the previous method, MPE and RPE techniques try to improve the speech quality by giving a better representation of the excitation signal. With MPE, the excitation signal is modeled as a series of pulses, each with its own amplitude. The positions and amplitudes of the pulses are determined by an AbS procedure. The MPE method can produce high quality speech at rates around 9.6 kbps.

The RPE technique works in a similar fashion, here only the pulses are regularly spaced, as the name suggests. The GSM mobile telephone system uses a RPE variant which operates at approximately 13 kbps. (Mitra 1998) (Kondoz 1994)

2.9 Excitation with the Discrete Cosine Transform (DCT)

2.9.1 Introduction

This part of the work we are going to focus on, since the DCT was used in deducting the residual signal that used in the filter excitation.

This process is carried out through framing the speech signal; the common term with the DCT for frames is blocks. A linear transformation is carried out on each block; the output of this operation is the transform coefficients, which represents the power components of the signal for the selected range of coefficients (Giridhar 1995)(Alwan 2002).

The following will be an explanation of the method that has been used in the thesis and also an explanation for the concept itself.

- The preemphasized speech signal is grouped into frames and overlapped by 10 samples between two adjacent frames.
- The DCT coefficients are quantized one by one, the numbers that were used in our case were 4 bits for each coefficient.
- The power spectrum of each frame is claimed to be the vector that holds the coefficients of the frame transformation, in our case 40 coefficients were used since the signal started to show pleasant voice at this number, the signal quality will increase if we take more coefficients, the penalty will be the data rate and the complexity.
- In the decoding phase inverse DCT is carried out.

This operation is carried out after the understanding of the distribution of the numerical values of the coefficients. (Lam 2004)

2.9.2 Spectral Modeling

A smooth spectral fit to the DCT coefficients provides an estimate of the energy of the coefficients (Alwan 2002). Where the calculation might be done using the FFT real part after reordering the in inputs of the DCT. (Storn 1996)

This energy estimate serves a dual role:

- First, it is used to determine the number of bits to be assigned to each coefficient, in such a way as to minimize the total mean-square quantization error.
- Second, the spectral estimate is used to scale the quantizer, it affects adapting the quantizer range for coefficient being quantized.

The envelope of the DCT spectrum retains the formant structure of speech. The problem of modeling the spectrum is similar to one which occurs in linear predictive coding (LPC) of speech. (Kabal et al. 1984)

And there are some new methods that deal with the spectrum of the signal directly rather than building up a codebook for the residual signal. one method is based on the distance measure in the adaptive filter context. The research showed a faster relatively good quality for the estimation of the spectral envelop of the signal. (Wei et al.. 2003)

In the search for the best possible pitch for the CELP coders happens in a closed loops search for most of the types of CELP coders. A new method was recently introduced called the joint pitch and LPC (JP-LPC). This method showed a faster convergence to the best possible pitch with an impressing SNR results. (Serizawa et al. 1999)

Chapter Three

Signal Quality Assessment

3.1 Introduction

Speech quality measurements are a crucial issue for all the speech related research for communication and commercial reasons. Quality measurement might be either subjective or objective. For objective methods, the signal is the target of the analysis, but in the subjective methods, the user interaction with the synthesized voice is the measure for determining the speech quality. The Mean Opinion Score (MOS) is the most widely used subjective measure of voice quality and is recommended by the ITU (International Telecommunication Union Aug. 1996). An MOS value is obtained as an average opinion of quality based on asking people to grade the quality of speech signals on a five point scale (Excellent, Good, Fair, Poor and Bad). under standard controlled conditions that are set out in the ITU standard. In voice communication systems, MOS is internationally accepted in both developing and commercial level metric, since it links the voice quality to the end user. The problem of subjective MOS measurement is that it is expensive, time consuming and lacking of repeatability. Besides, it cannot be used to obtain information about the daily transmission infrastructure. Figure 1.3 shows the direction of the speech quality measuring in its both types.

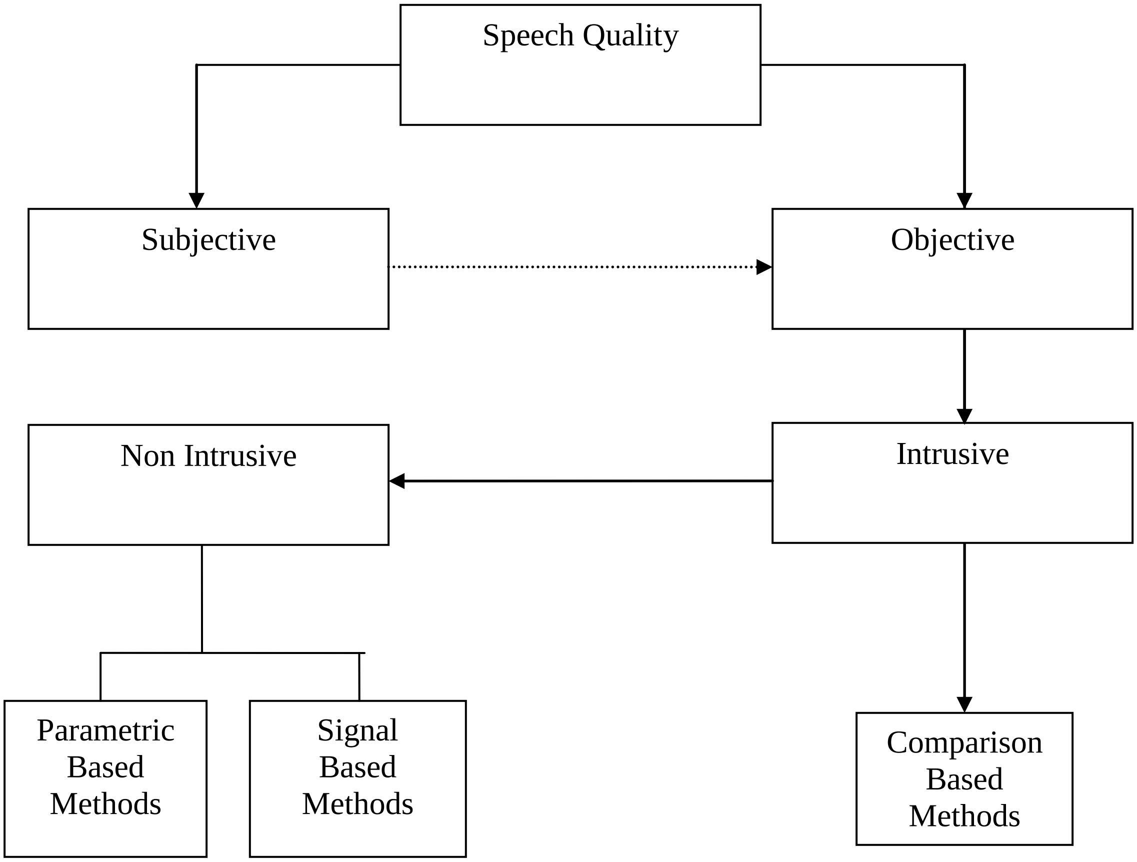


Figure 3.1: Classification of Speech Quality Assessment Methods

This figure represents all the sets of possible quality testing methods used in communication. But let us have general information about how to set up a quality test, and what to think about and how it all had started.

At the very early developmental stage of digital communication, signal to noise ratio (SNR) was often the method of evaluating the quality of wave form coding such as PCM and ADPCM, where the complications of the MOS was negligible. (Oppenheim et al. 2000)

With the developing of the LPC, it was observed that the synthetic speech was quite intelligible; however, it has a poor SNR value. The conclusion was that some new measure must be developed in order to handle these types of parametric coders. So researchers decided to think about constraints rules for the process of the speech quality assessment.

3.2 Speech Quality Measurement Constraints and Approaches

The general methods seen in figure 3.1, were developed using the general constraints seen below

- Intelligibly. Meaning; can we understand what was said?

- Naturalness and Pleasantness. There are some known type of noises that come along with speech synthesis such as: noise, echoes, muffling, and clicking.
- Speaker Recognition. Does the synthetic speech allow straightforward identification to the original speaker?

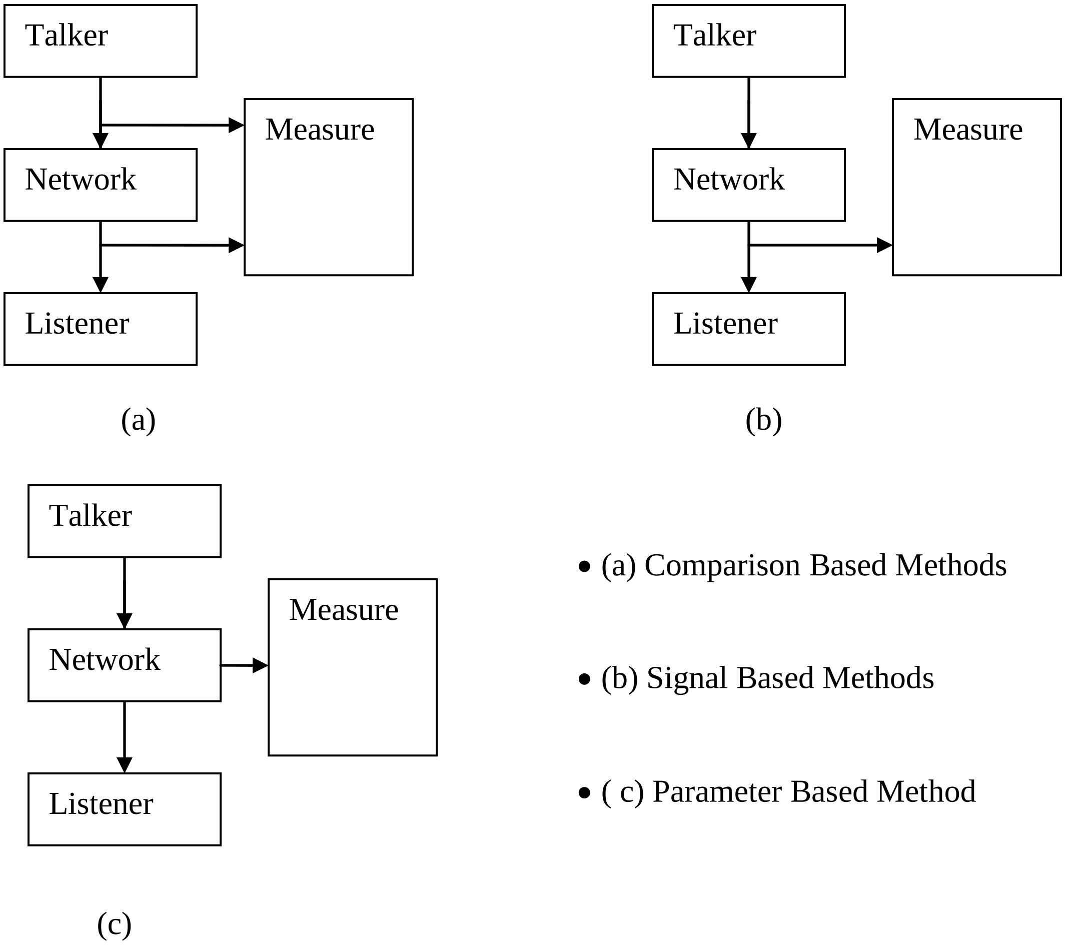
It is important to note that speech quality must be measured under varying conditions, such as:

- Dependency on Speaker; Some coders tend to show different performances with different speakers, a standard operation for the coder might be carried out with males, females and children.
- Dependency on Language: the fact says that languages have different speech sounds (phonemes) that may or may not be appropriately modeled by the coder. Different languages can discover a weakness toward a given idiom.
- Dependency on Signal Levels: Coders might show different behavior for the same sound with different power levels, different rates of signal power levels should be used to test the quality of a certain coder.

- Background Noise: Background noise is almost inevitable due to the increasing of portability. Typical noise sources include car, street, and transient noise, music, and interfering speakers.
- Tandem Coding: Codec interchanging and cascading things that happen in the entire communication environment, we should be aware that this operation will not affect our coder.
- Channel Errors: Communication channels contain some sort of errors; we certainly have models for these errors these days. Quality of the coder can be measured under different bit error rates.
- Nonspeech Signal. A coder must also be tested according to non speech signals; if it is going to be adapted in a commercial level, music is a good example of these signals .(Chu Wai C. 2003)

3.3 Modern Speech Quality Measurement Approaches

According to the previous information, the real interaction has replaced the mathematical measurements approaches; the real interaction became the base for measurements. This has opened the door for the objective measurement of voice quality in modern communication networks. The objective measurement is categorized as intrusive or non-intrusive. The intrusive method is the most accurate since it comes as a comparison between the sent signal and the conveyed signal. This means a duplicate bandwidth is required to have intrusive method carried out. This is why this method is dropped when we are talking about live traffic quality monitoring. The nonintrusive method is based on the latest ITU standard, P.862, Perceptual Evaluation of Speech Quality (PESQ) (International Telecommunication Union May. 2001). A comparison between the reference and the degraded speech signals is carried out to obtain the MOS score, as shown in Figure 3.2(a). Nonintrusive method does not include injecting the source signal. This makes it a good approach for monitoring live traffic. There are two categories for the non-intrusive methods, which are signal based and parameter based methods. Figure 3.2(b) and (c) show operations flow in both cases. The E- model is a very interesting parameter based method since it can predict the MOS score from the IP network parameters (R. G. Cole and J. Rosenbluth 2001) (Clark 2001). The vocal tract model is the most famous signal based analysis used by the ITU (International Telecommunication Union 2003) that predicts voice quality by analyzing the dragged signal only.



- (a) Comparison Based Methods
- (b) Signal Based Methods
- (c) Parameter Based Method

Figure 3.2: Three Main Categories of Objective Quality Measurement

3.4 Subjective Speech Quality Measurement

Subjective methods are mostly used for benchmarking; The ITU P.800 (International Telecommunication Union Aug. 1996) includes several methods for subjective assessment for the quality of transmission. The most used among them is Absolute Category Rating (ACR), and the Degradation Category Rating (DCR) is also used, which is a Degradation Mean Opinion

Score (DMOS); the opposite direction of satisfaction. MOS tests are carried out under standard conditions in acoustic rooms, with identical testing conditions, the complication and the expansiveness of this test arises from these needs.

3.4.1 Absolute Category Rating (ACR)

For Absolute Category Rating (ACR) listening test, subjects arbitrary listeners to rates a given speech without having the chance for listening to the original voice. That rate comes in a scale as shown in Table 3.1. The average of opinion scores of the subjects gives the Mean Opinion Score (MOS).

Category	Speech Quality
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table 3.1: Absolute Category Rating (ACR)

3.4.2 Degradation Category Rating (DCR)

Degradation Category Rating (DCR) is normally used to serve the purpose of detecting the non quality related information. Users are not asked to seek good information but rather measure the level of their dissatisfaction. DCR procedure uses an annoyance scale and a quality reference. Subjects are asked to rate annoyance or degradation level by comparing the speech utterance being tested to the original (reference). The rating scales or the degradation levels are shown in Table 3.2.

Category	Degradation Level
5	Inaudible
4	Audible but not annoying
3	Slightly Annoying
2	Annoying
1	Very Annoying

Table 3.2: Degradation Category Rating (DCR)

3.5 Intrusive Speech Quality Measurement

Intrusive objective speech quality measurements use two input signals, a reference (or original) signal and the degraded (or distorted). They are more accurate to measure end to end speech quality, but unsuitable for live network traffic due to the need of injecting the reference signal. The major two groups of this type are; the frequency domain and the time domain group measures, such as Signal-to-Noise Ratio (SNR) and Segmental Signal-to-Noise Ratio (SNRseg).

For a given original speech $x[n]$ and the synthetic version $y[n]$, the SNR is defined by:

$$SNR = 10\log_{10}\left(\frac{\sum_n x[n]^2}{\sum_n (x[n]^2 - y[n]^2)}\right) \qquad (3.1)$$

The Segmental Signal-to-Noise Ratio (SNRseg) is a refinement with respect to conventional SNR measure, and is created to handle the dynamic nature of nonstationary signals such as speech. The definition of SSNR is

$$SSNR = \frac{1}{N}\sum_{m=1}^{N} SNR_m \qquad (3.2)$$

That is, it is an average of SNR values obtained for isolated frames, where the frame is block of samples. The SNR in (3.2) is computed for frames of 10 to 20 ms in duration.

The SNR and SSNR measure are a wave form coders assessment approaches. As we can see from (3.1) and (3.2), both the SNR and SSNR measures signal alignments, amplitudes and phase, which is a subject of sever change in the linear prediction coders. Most of the modern coders with low bit rate do not preserve the original shape of the source signal, indeed, it might be highly variant, this makes SNR and SSNR meaningless for the evaluation of these coders. These methods are very simple to implement, but are not suitable for estimating the quality for low bit rate codec and modern networks.

The second group is spectral domain measures, the Linear Predictive Coding parameter distance measures, and the cepstral distance (European Telecommunications Standards Institute 1999). These measures are concerned about the signal transfer domain issues, measuring the modifications of the frequency domain. This is more practical than the time domain approaches since it focus on the main issues that the LPC model is built on. The most famous measures used are Perceptual Speech Quality

Measure (PSQM) (International Telecommunication Union 1998) (Beerends et al. 1994), Perceptual Assessment of Speech Quality (PAMS) (International Telecommunication Union Aug. 1996) (Rix et al. 1999) Measuring Normalizing Blocks (MNB) (Voran 1999) (Voran 1999), Enhanced Modified Bark Spectral Distortion (EMBSD) (Yang 1999), (Yang et al. 1998) and Perceptual Evaluation of Speech Quality (PESQ) (Clark April 2001) (Rix et al. 2001) which is the latest ITU standard for assessing speech quality for communication systems and networks.

3.6 Non-intrusive Speech Quality Measurement

3.6.1 General Concept

Unlike intrusive methods, in which, a reference or test signal is injected into the tested system or network and live traffic has to be interrupted during the test, non-intrusive speech quality measurement methods do not need the injection of a reference signal and are appropriate for monitoring live traffic. There are two categories of non-intrusive speech quality prediction methods. One is to predict speech quality directly from varying IP network impairment parameters (e.g. packet loss, jitter and delay) and non-IP network parameters (e.g. codec, echo, language and/or talker issues) as shown in Method 2 (parameter based) in Figure 3.3. The purpose is to establish the relationship between perceived speech quality and network or non network related parameters. One of the methods is the E-model. The other category is to predict speech quality directly from degraded speech signal (or in service signal) using signal processing methods as in Method 1 (signal based or output-based). The in-service speech signal can be derived directly from T1/E1 links as shown in the figure. Representative methods are INMD (in-service, non-intrusive measurement devices)/CCI (call clarity index) (International Telecommunication Union 1996) (International Telecommunication Union May. 2000), vocal tract model (Clark April 2001) and machine speech recognition (Jiang et al. 2002).

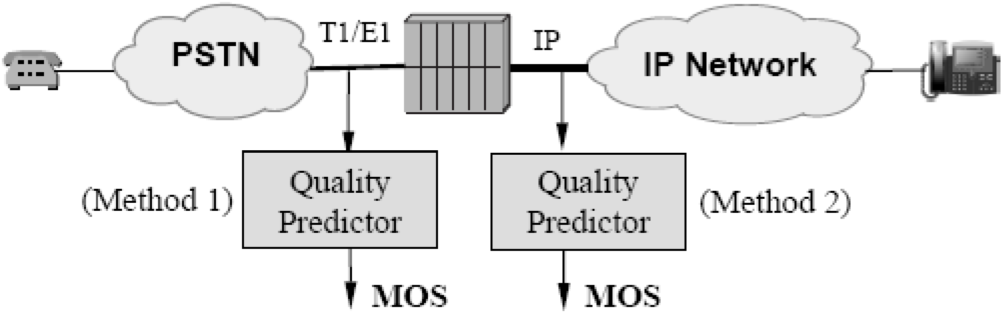


Figure 3.3: Non-intrusive Speech Quality Measurements (Cisco 2002)

3.6.2 E-model

The E-model is named after the European Telecommunications Standards Institute (ETSI) (European Telecommunications Standards Institute 1996). It was organially proposed as planning tool (European Telecommunications Standards Institute 1999) (International Telecommunicaion Union 2000) (Johannesson. 1997), but it is used now to predict quality for VoIP applications (Cole et al. 2001) (Markopoulou et al. 2002). It uses the transmission rating scale, R (Moller et al. 2002) (Clark 2001). The E-model makes a combination of the effects of various transmission parameters into a rating factor, R (which lies between 0 and 100), so MOS scores can be derived. The rating factor R is given by the following:

R is given by:

$$R = R_0 - I_s - I_d - I_c + A \qquad (3.3)$$

Where

R_0 : S/N at 0 dB point (groups the effects of noise).

Is : impairments and defects that occur simultaneously with speech (e.g. quantization noise, received speech level).

Id: impairments that are delayed with respect to speech (e.g. talker/listener echo and absolute delay).

Ic: Effects of special equipment or equipment impairment (e.g. codecs, packet loss and jitter).

A : Advantage factor or expectation factor (e.g. 0 for wire line and 10 for GSM) .

ITU G.109 (International Telecommunication Union Aug. 1998) defines the speech quality classes with the Rating (*R*), as illustrated in Table 3.3. A rating below 50 indicates unacceptable quality.

R-Value Range	**100 – 90**	**90-80**	**80-70**	**70-60**	**60-50**
Speech Transmission Quality Category	Best	High	Medium	Low	Poor
User’s Satisfaction	Very Satisfied	Satisfied	Some Users Dissatisfied	Many Users Dissatisfied	Nearly All Users Dissatisfied

Table 3.3: Speech quality classes according to E-model

MOS score can be derived from R value by using the equations in ITU G.107 (International Telecommunication Union 2000). For VoIP applications, the impact of IP network impairment is expressed with the *Ic* value, a good idea for master thesis is building up a fuzzy model for the E-Model.

Chapter Four

Digital Filters Coefficients Redundancy

4.1 Introduction

During the analysis of the LPC model, which was introduced in chapter two, we have seen that the signal retrieval is dependent on the voicing information. This information is expressed either as white noise or periodic impulse (or any other excitation method), and the signal gain, in addition to the digital filter parameters. Not to forget the mentioning of the system order that affects the process dramatically. For the sake of recognition and voice conversion we are obliged to have higher orders of the linear system. (Kain et al. 2000) (Chou 2002)

The LPC starts to show a very good behavior quality wise around the orders of 12 and 15 (Singer et al. 1999). although a very good spectrum modeling and analysis would give a lower order of analysis like 8 if the spectrum is being treated as pairs, this will cause the quality to be like the order 12 and 15, but with actual order is three. (Alku et al. 2004)

We have started from a basic idea of the correlation between the odd linear prediction coefficients together, and the even linear prediction coefficients together, this have lead to the idea of having a possible replacement for these coefficients, but the correlation was not a reliable measure and not a strong base for the generalization of the concept

4.2 Assumptions

Different ways and approaches were introduced for realizing the LPC process, and each achieves certain desired outputs using different analysis prospectus. We are claiming in the thesis that there are a closed set of phones that a particular human can generate physically, but the mathematical model that describes the system did not show such behavior, i.e. it has no closed set of numerical values.

Maybe this problem is caused by the mathematical model, since it is based on statistical representations, regional analysis for the spectrum or the cepstrum in some cases. We focused in our study on the digital filter values, and we would like to achieve data rate reduction from the study of the digital filters coefficients properties.

How do we obtain the digital filters coefficients?

From section 2.4 we saw that, for a single frame, the digital filters coefficients are obtained by solving the linear system

$$\begin{pmatrix} R[0] & R[1]\ldots & R\,[M\,] \\ & & \\ R[1] & R[0] & R[M-1] \\ \vdots & \ddots & \vdots \\ & & \ddots \\ R\,[M\,] & R[M-1]\cdots & R\,[0] \end{pmatrix} \begin{pmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ \vdots \\ a_M \end{pmatrix} = \begin{pmatrix} J \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{pmatrix}$$

This system is solved using several methods as Levinson-Durbin recursion or (Leroux-Gueguen Algorithm) over an infinite interval around 1 and -1. This makes the repetition of these values very hard to obtain.

4.3 General Solution Strategy

The autocorrelation function is used to estimate the power spectral density of the signal, using information obtained from the time domain, and then represent the signal as a linear combination of regressed groups of highly correlated sub groups, where each sub group is multiplied with the prediction coefficient at that sub-group of points, a noise term is added after that to the equation. This is not a deterministic representation of the system, there is an error to reduce, the model does not supply a unique values that describe the system. Let us make use out of this, and do something to the difference equation coefficients, by replacing them according to some criteria. We are working in a noisy environment after all, and the noise is part of our difference equation, and we think that the equation is tolerant for slight modifications.

4.4 Definition of Digital Filters Redundancy

It is the process of expressing the values of a digital filter coefficients vector by another vector that was previously encountered and saved in an easily accessible space.

Consider the two vectors that contain filters coefficients $X = \left[a_{1,} \cdots, a_M\right] \& Y = \left[b_{1,} \cdots, b_M\right]$.

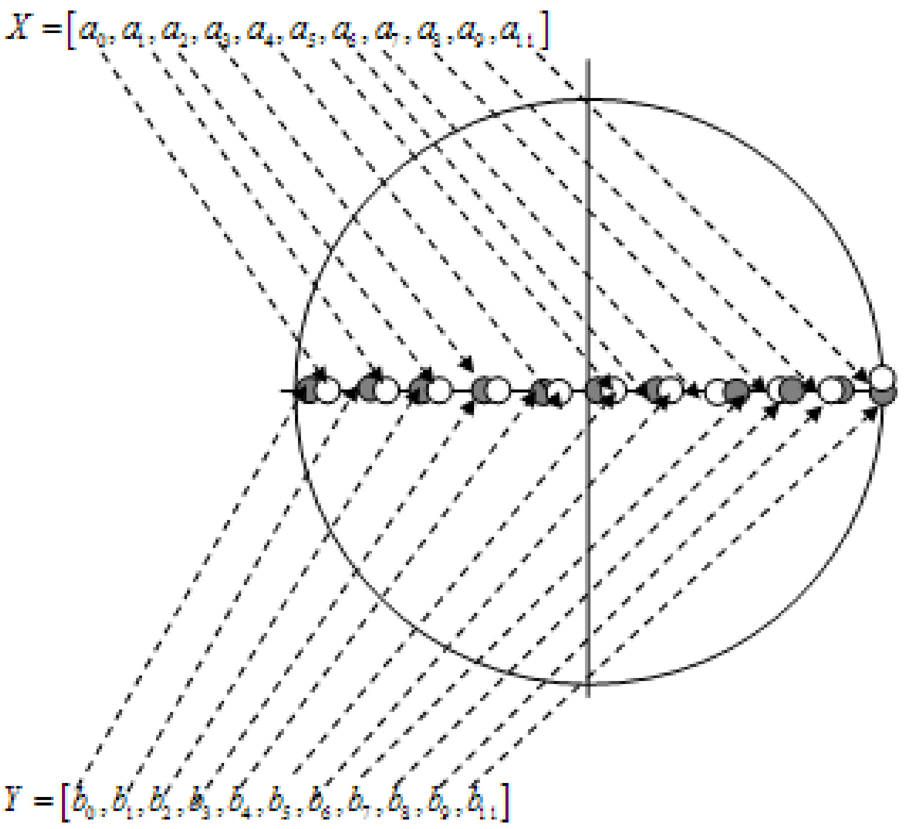

$X = [a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{11}]$
$Y = [b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{11}]$

Figure 4.1: Digital Filter Coefficients String Values

We can see from figure 4.1 the difference between Y and X does not equal zero, since the poles of the digital filter that they represent are not located exactly on the same points.

But let us consider the two FIR filters

X = [1.0 0.23 0.27 -0.04 0.13 -0.04 -0.01 -0.17 -0.02 -0.07 -0.03] and

Y = [1.0 0.31 0.32 -0.11 0.21 -0.12 -0.11 -0.21 -0.09 -0.11 -0.07]

There impulse responses are plotted respectively in figures 4.2 and 4.3. Note that they do not have the same response since they do not have the same digital filter coefficients string.

Let us consider the vector Z which is obtained by the following conditions

$$Z_i = \begin{cases} |a_i - b_i|, 0 < a_i, 0 <, b_i \\ ||a_i| - |b_i||, a_i < 0, b_i < 0, \\ \end{cases} \qquad (4.1)$$

Finding Z from (4.1) , X and Y we obtain

Z= [0 0.08 0.05 0.07 0.08 0.08 0.1 0.04 0.07 0.04 0.04]

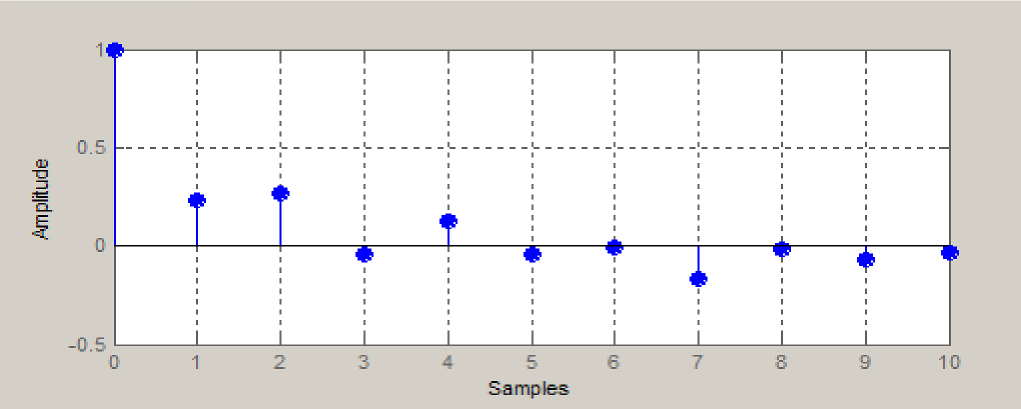


Figure 4.2: Filter X Impulse Response

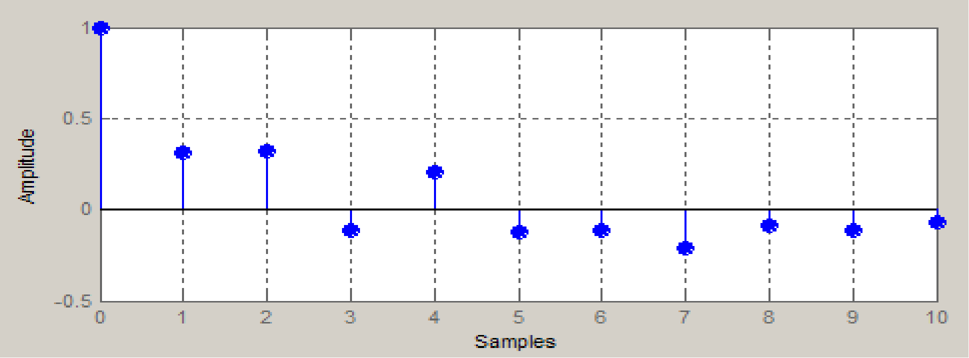


Figure 4.3: Filter Y Impulse Response

As we can see the slight difference in the response between X and Y, with Z represents the algebraic difference in there responses. The Z vector represents the distance victor between each zero of the filters. And this arises the questions

- Is this difference in response tolerable?
- If yes, what is the distance range that we can tolerate?
- What is the impact of this distance on the solution range of equation 4.1, can we have a closed set of solutions?
- How can we tell that this distance is tolerable or not?
- How far can we go in the generalization of this?
- What is the impact of the excitation signal for this filter, related to the distance measure?

Now we can build our case.

4.5 Problem Statement

Reduction of transmitted VoIP information in an LPC codec based environment using the synthesis filters coefficients redundancy concept.

4.6 Verification Procedure

As we can see; we are taking the problem with many questions to answer, so we decided to define a verification procedure, we need to verify the use of the coefficients redundancy concept for the sake of data rate reduction in the VoIP.

The LPC model states that the speech signal can be expressed as an output of a digital filter of an appropriate order, with a known input type which aims to represent the important power regions of the signal spectrum. This is a very general way to express the system. If we revisit the first assumption, which I claim to be true, meaning that there is a closed set of voices and phones a human can conduct, then that mathematical model that represents this system has to be of a closed set of values too. But as we all know this does not happen, and this is due to the use of lots of random variables during the processing operations. We would like to mention that dictionaries of residual signals are perfectly functioning in CELP, so can we do something like this in the digital filter coefficients string?

In general, when someone talks for a long time it is very likely that this person has used most of the possibilities that his vocal system can generate. So we can approach the closed set of values in the numerical values he can generate, in both components of the model, i.e., the digital filters values and the excitation code. Starting from this point, we did the following to support our claim:-

- Gathering four audio samples (two males, two females), each sample is represented in group of wav files (60% of the whole wave files time for a sample).
- Deduce the filter Coefficients the sample to create digital filters coefficients dictionary using a group of files represents that sample.
- For the other group of files represents that same sample; deduce the filter Coefficients to encode against this dictionary (compare and replace).
- Merge the coefficient tables in one optimized table (repetition and silence elimination).
- Result: we have a vector table that might be used as a coefficients dictionary then encode new samples against this dictionary with respect to the threshold

4.7 Compare and Replace Algorithm

After Deducing the filter Coefficients of 60% of each sample, we are expected to have a filters dictionary M, for each sample that we are going to encode; there is a vector table S that contains the coefficients, if we are going to search for a certain value in this dictionary, algorithm in figure 4.4 was used and its implementation is in the annexes.

```
For i=0 to i = end of S

     While not found and J<end of M
      If Compare(S(i),M(j))=true
       Found = true
       Else
       J=j+1
    End While

  End For

  Boolean function compare(A,B)
  Int x=0
  If  a[x]<b[x]±t and x<= order
  X=x+1
       Else
```

Figure 4.4: Compare and Replace Algorithm

4.8 The Mathematical Notation of the Operation

We are aiming at generating a digital filter's coefficients profile. This profile holds most of the values of the digital filter coefficients that are more likely to be encountered.

Assume V a vector holding the digital filter's coefficients for a given frame, where $V(i) = a_i$ is a single value of the coefficients at i shifting element of the filter, where a_i is a real number that represents $x \in [-1,1]$ under the normalization condition. (Chu 2003)

4.9 The Generation of the Testing Dictionary

Assume we have a certain male or female that has ten audio samples, each is around 40 seconds, Assume the signal S represents each sample modeled in order M system and divided into N samples.

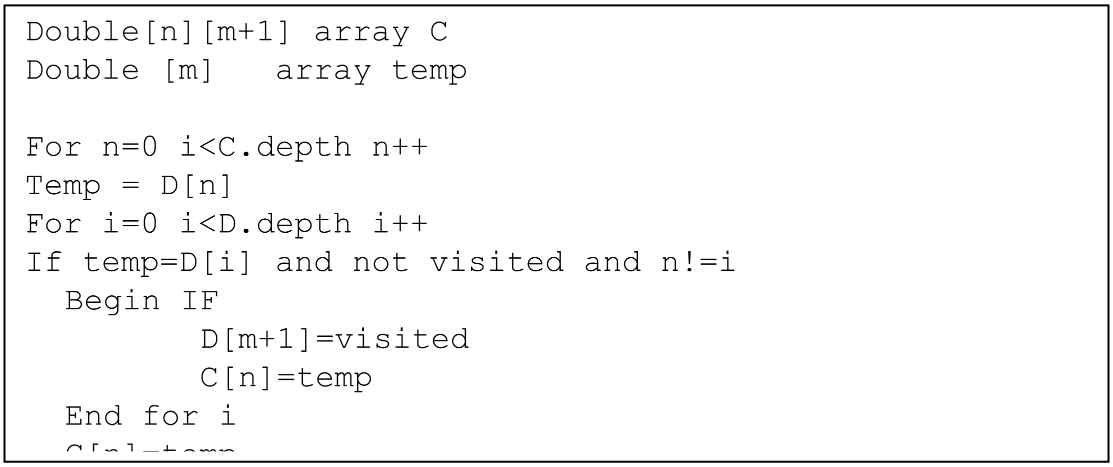
We will divide each individual sample into two groups, first group will be used to generate the coefficients dictionary, and the other group will be used to verify if there is a redundant value within one left sample.

4.10 Steps of Generating the Dictionary

- Perform the Levinson-Durbin recursion for each sample signal S_i .
- This will generate a vector space D_i, where D_i is a two dimensional matrix with $(n_i \times m)$ dimensions, m is the order of the system and n is the number of frames within the signal S_i .
- All the signal samples dictionaries are gathered in one big dictionary by concatenation on n_i generating D_{all} matrix with dimensions $\left(n_{\sum_{1}^{x} n}, m \right)$

where x is the number of all frames in the sample signals used in the generation process, and m still order of the LPC system.

- A D_{all} optimization process is carried out to make sure that there is a unique value inside the matrix for all values of its rows using the following algorithm



```
Double[n][m+1] array C
Double [m]   array temp

For n=0 i<C.depth n++
Temp = D[n]
For i=0 i<D.depth i++
If temp=D[i] and not visited and n!=i
  Begin IF
         D[m+1]=visited
         C[n]=temp
  End for i
  C[n]=temp
```

Figure 4.5: Generation of The Testing Dictionary

After passing this operation we have thousands of digital filter coefficients values. Now we are going to use this dictionary D_{all} to investigate the redundancy operation for samples that did not participate in the makeup of the D_{all}, we called this signal S_c.

The operation will look like this:

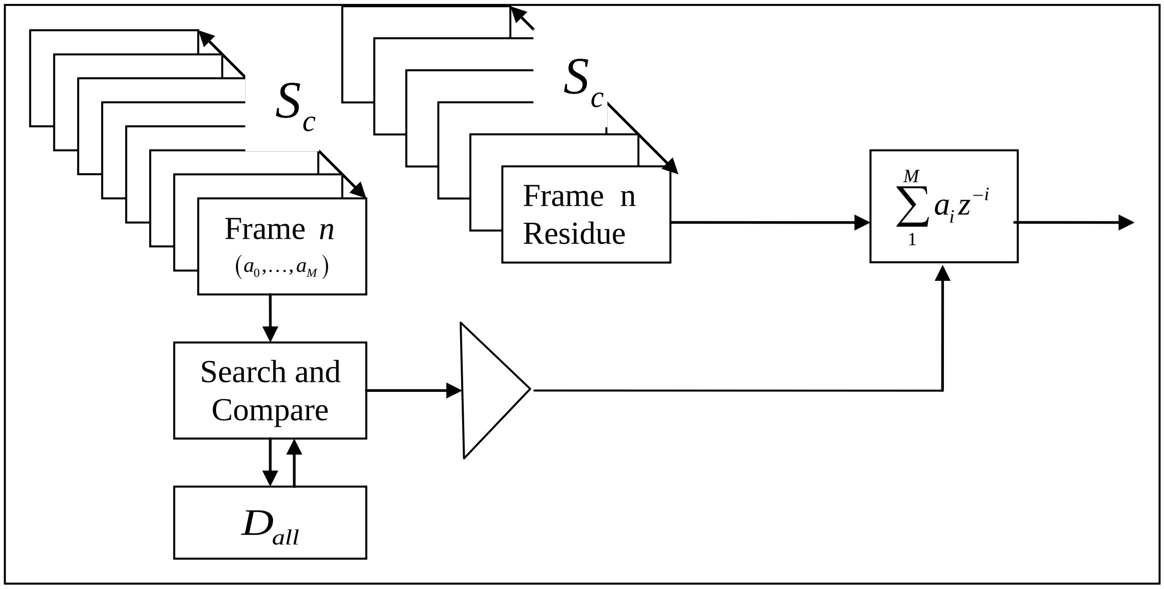


Figure 4.6: Generation and Usage of the Replacement Dictionary

4.11 Coefficients Dictionary Usage

We are mapping the values of coefficients vector from $x \in [-1,1]$ to a closed set or semi closed set where we are claiming that D_{all} is a closed form of the values represented in $x \in [-1,1]$ plus minus t. In other words, we are solving the linear system over closed sub intervals within that represented in the following interval on the z plane $Y = [-1 \pm t, -0.9 \pm t, -0.8 \pm t, \ldots, 0, \ldots, 0.8 \pm t, 0.9 \pm t, 1 \pm t]$. Now we have the Y interval with N possible outputs. Recall the signal S_c for every frame there is a vector V that holds the coefficients of the signal at that frame, where $V = D_i$, which is the row of that matrix that holds the coefficients of the given s_c

We would like to map

$$\begin{array}{l} V(i) = x \\ where \\ x \in I \end{array} \qquad (4.2)$$

Into the following representation

$$\begin{array}{l} V(i) = x \\ where \\ x \in Y \end{array} \qquad (4.3)$$

Where x is defined on Y under the condition $x = b$, where b is obtained from a previously encountered set defined on Y, and the distance between x and b is less than Z, where Z is obtained through the following

$$Z \;= \begin{cases} | x - b |, \;\; 0 < x \;\; ,0 <, b \\ \| x \;\; | - | b \;\; \|, x \; < 0, b \;\; < 0, \\ \end{cases} \qquad (4.4)$$

Z represents the distance t in the interval Y, for every value in D_i. Now recall the values of D_i in the signal that we want to encode over $\boldsymbol{D_{all}}$ where

$$\begin{array}{l} D_i : I \to V \\ becomes \\ D_i : Y \to V \end{array} \qquad (4.5)$$

Using the previous condition in equation 4.6.

4.12 Replacements Results

We are able now to represent values from D_i using values in D_{all} that are very close to values of D_i in the z plane. Instead of transmitting the values in the row of D_i, we can transmit the index of semi identical ones in the D_{all} with respect to the previous condition, now we can make this operation into an algorithm as shown in figure 4.7.

4.13 The filter coefficients replacement algorithm

```
Load D_all ; coefficients  dictionary

While signal frames are not over
Calculate V                 ; frame coefficients vector
While not found and i<=D_all.lenght
    If compare(V,D_all[i])
    Beging
    D_i=D_all
    Found = true
    End if
    If i= D_all.lenght
    D_i=V
End inner while
End Outer while
```

Figure 4.7: Filter Coefficients Replacement Algorithm

4.14 Research Results

According to the verification procedure in section 4.6; the input data was obtained from the calculation of the LPC parameters for the mentioned samples forming the dictionary of the linear prediction coefficients dictionary, and then the reduction operation is carried out on the new sample files.

Let us have a look at the following table

Sample	# Of Signal Frames	# Of Dictionary Entries	Used Distance T	Number Of Replacements	Coeff. Reduction Ratio (CRR)Within Signal %	Minimum. Transmission At CRR Kbps	Segmental Signal To Noise Ration Db	MOS
Female 1	372	3861	0	0	0%	7.71	0.9102	3.17
			0.1	68	%18	5.43	-2.2421	3.08
			0.2	194	%52	5.43	-4.6728	2.75
			0.35	372	%100	5.43	-8.6986	2.5
Female 2	593	3740	0	0	0%	7.71	0.4880	3.33
			0.2	397	%67	5.43	-0.7264	2.75
			0.3	593	100%	5.43	-3.336	2.51
Male 1	859	4000	0	0	0%	7.71	0.7906	3.2
			0.1	484	51%	5.43	-1.261	3.12
			0.2	859	%100	5.43	-3.8512	3.08
Male 2	941	6120	0	0	%0	7.71	0.4836	3.33
			0.1	198	%21	5.43	-1.1716	3.16
			0.2	443	%47	5.43	-2.9430	3
			0.3	941	%100	5.43	-4.5828	3

Table 4.1: Research Results

This table represents the results of the research carried out using Matlab code that is available in the annex of the thesis, with the cooperation of the related Java files we used for the replacement purpose.

The tests as mentioned before were carried out using the two male samples and two female samples, as seen in the first column, the second column holds the number of frames in that sample, I would like to recall each sample means a 20 ms long data, column 3 represents the previously generated filter coefficients entries, that forms the dictionary that we will use to reduce our data rate against. The fourth column is the maximum acceptable distance, which is going to be used by the algorithm in Figure 4.7; we control the number of the replacement by controlling the acceptable. The fifth represents the number of the frames that are being replaced, where the sixth column represents the replacements percentage, was called the coefficient reduction ratio (CCR). The seventh column represent the minimum possible transmission data rate. The eighth columns represents the segmental signal to noise ratio of the generated file after replacement with the input file without any replacement, we see that the higher the replacement the higher the SSNR. The last column represents the MOS test results, it was generated in a sonic isolated room with 12 persons, and the questionnaire used in testing is in the annex.

Referring to frame types in table 5.1 and 5.2 the data rates are calculated. After this, the following claims arises

1. Coefficient's replacements might lead to data rate reduction with a relatively acceptable quality.
2. The longer the dictionary entries the higher the replacements with a better quality.
3. The more replacements we have the lower quality we have.
4. The higher the replacements the higher the SSNR.
5. The replacements have shown a better quality with male voices

4.15 Complexity Analysis

An algorithm complexity is the combination of its sub units complexity, the main algorithm components used by our codec are

- The calculation of the LPC coefficients with the Levinson-Durbin . (Tankelevich 2005)
- The pre-emphasis filter.
- The reconstruction of the LPC.
- The synthesis inverse filter calculation.

- Search profile depth
- The DCT (if the fast algorithm is applied) for residual calculation. (Tankelevich 2005)

The following represents the complexity table:

Function	Function Complexity Variables	Function Complexity
Levinson-Durbin	n represents the system order, in our case equals 12.	$O\left(n^2\right)$
The Pre-Emphasis Filtration	n additions for the filter summation component of the series. m represents the number of the coefficients multiplications in the filter.	$O(n^2)+O(m^2)$
The Reconstruction of the LPC	n additions for the filter summation component of the series. m represents the number of the coefficients multiplications in the filter.	$O(n^2)+O(m^2)$
Search Profile Depth	n the profile depth.	$O(n)$
DCT	n is the number of mapping points of the transfer function.	$O(n^2)$

Table 4.2: Complexity Table

4.16 Codec overall delay

The overall delay of the systems tells one how long it takes from the input of the first sample of speech into the system until the first sample of the synthesized speech is available at the output of the system. This is clearly an important number, since one would like to process the data in real time. If an LPC vocoder is used in a communication system, we cannot accept a large delay of the transmitted speech signal. Humans are able to perceive delays of speech within several hundred of milliseconds during a talk at the telephone. For our work the limit was set to a maximum allowed value of 100 to 300 ms. For our proposed solutions the overall delay is 30 ms, since the window length is 20ms and the overlapping is 10ms. In other words, the system needs to have at least 30 ms of input data before the first calculation can be done. Of course, the calculation time needs to be added to this delay. It is not possible to come up with this number, since we employed Matlab. The calculation time therefore depends on the speed of the used computer.

Chapter Five

Transmission and Reception

5.1 Introduction

This chapter aims to place the codec we are suggesting in the VoIP framework. In chapter one we have shown that there are different ways to create VoIP connectivity, i.e., different protocols and standards to do that, but all of these protocols totally agree on the codec component and the RTP transmission mode. Both of the SIP and the H.323 have the following component in their stack seen in the figure 5.1, and the work we are intending to present is based on this part of the protocol stack, we will call it the Alpha stack, which is a hypothetical representation of the parts of the protocols seen before.

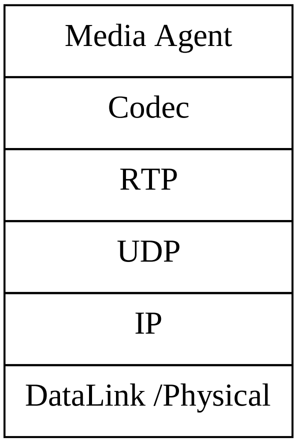


Figure 5.1: Simple VoIP Stack

- Media Agent, the application level of the stack where the user interacts with the VoIP system.
- Codec, the compression component of the system, we will focus our work on this block.
- RTP, the voice transport component, the LPC information is what the RTP packets hold.
- UDP, the transport protocol.
- IP, the network to network connection engine.
- DataLink and the Physical part is how the connection is carried out from the logical and physical state of mind.

We are not intending to suggest anything on the signaling information or the call connecting and routing parameters since thy are beyond the scope of this work.

5.2 Suggested RTP Packet Structure

The codec compresses each 20 milliseconds sampled at 16000Hz, coded on 8 bit sampled input speech, each frame is expressed by the following:

- LPC coefficients.
- Frame Gain information.
- DCT coefficients (first 40 coefficients that will represent the residue).

We have two types of frames to transmit:

- Full frame that contains all the typical frame information.
- Frame that includes coefficients index that were found previously beside the DCT coefficients and the frame gain.

The following represents the values in numbers:

1. **Frame Type One (normal frame)**

Contents	Number of	Minimal	Total bits

	values	representation bits	
LPC Coefficients	12	8	96
DCT Coefficients	40	4	160
Gain	1	5	5
Total			257

Table 5.1: Frame Type One

2. **Frame Type Two (reduced frame)**

Contents	Number of values	Minimal representation bits	Total bits
LPC Coefficients	1	16	16
DCT Coefficients	40	4	160
Gain	1	5	5
			181

Table 5.2: Frame Type Two

This means that we have to include two bit rates in our RTP packet; 257 bit for high rate and 181 bit for lower rate. This is conducted as Rate 1 (257 bits) and Rate ½ (181 bits).

5.3 The RTP Packet Organization

An RTP packet in general takes the following organization

0									1											2									3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
RTP Header																														

RR	LLL	NNN				one or more codec data frames

Figure 5.2: RTP Packet Organization

The RTP header has the expected values as in the general standards in (McKay 2003). The codec data frames are aligned on octet boundaries. And also we will interleave the codec together, we are having different data rates for the same source, and we also can use different codec types in the case of H.232, our packet parameters needs to be aware of this point. The RTP packets hold CODEC (the upper case), this CODEC is a result of interleaving different codec data rates, and different codec types, i.e., audio or video. This packet structure is a modified form of the CELP packet.

1. Reserved (RR): 2 bits, must be set to zero by sender, and be ignored by receiver. It is a good practice to include such fields.
2. Interleave (LLL): 3 bits, must have a value between 0 and 5 inclusive. The remaining two values (6 and 7) must not be used by senders. If this field is non-zero, interleaving is enabled. All receivers must support interleaving. Senders may support interleaving. Senders that do not support interleaving must set field LLL and NNN to zero.

3. Interleave Index (NNN): 3 bits, must have a value less than or equal to the value of LLL. Values of NNN greater than the value of LLL are invalid. (RFC 2658)

5.4 CODEC Data Frame Format

The output of the codec must be converted into RTP CODEC data frames for inclusion in the RTP payload as follows:

Octet 0 of the CODEC data frame indicates the rate and total size of the CODEC data frame as indicated in this table, recall that the rate indicates weather we are sending a full frame or a reduced frame, this information will be useful for the signal retrieval.

Octet 0	Rate	Total Codec Data Frames In Octets
1	1	33
2	1/2	23

Table 5.3: RTP CODEC data rates

The bits as numbered in the standard from highest to lowest are packed into octets. The highest numbered bit (257 for Rate 1, 181 for Rate 1/2) is placed in the most significant bit of octet 1 of the CODEC data frame. The second highest numbered bit (257 for Rate 1, etc.) is placed in the second most significant bit (Internet bit 1) of octet 1 of the data frame. This continues so that bit 258 from the standard Rate 1 frame is placed in the least significant bit of octet 1. Bit 257 from the standard is placed in the most significant bit of octet 2 and so on. The remaining unused bits of the last octet of the CODEC data frame must be set to zero.

Finally, the regular frame which holds the content of table 5.1 and the reduced frames holding the information in table 5.2. The receiver differentiates between the frame types according to the content of octet zero of the codec data frame. And the action of saving and retrieval is taken according to that.

5.5 The Sender Responsibilities:

- Perform the LPC operations.
- Make a time stamp for the frame.
- Decide whether the coefficients were previously encountered or not
- If they were not encountered transmit on rate one, and save the coefficients with index.
- If they were encountered transmit on rate ½, the coefficients here are expressed by the index.

- Convert the codec frames into transmission frames by aligning as octets as seen in previous section.

The sender is following this flowchart

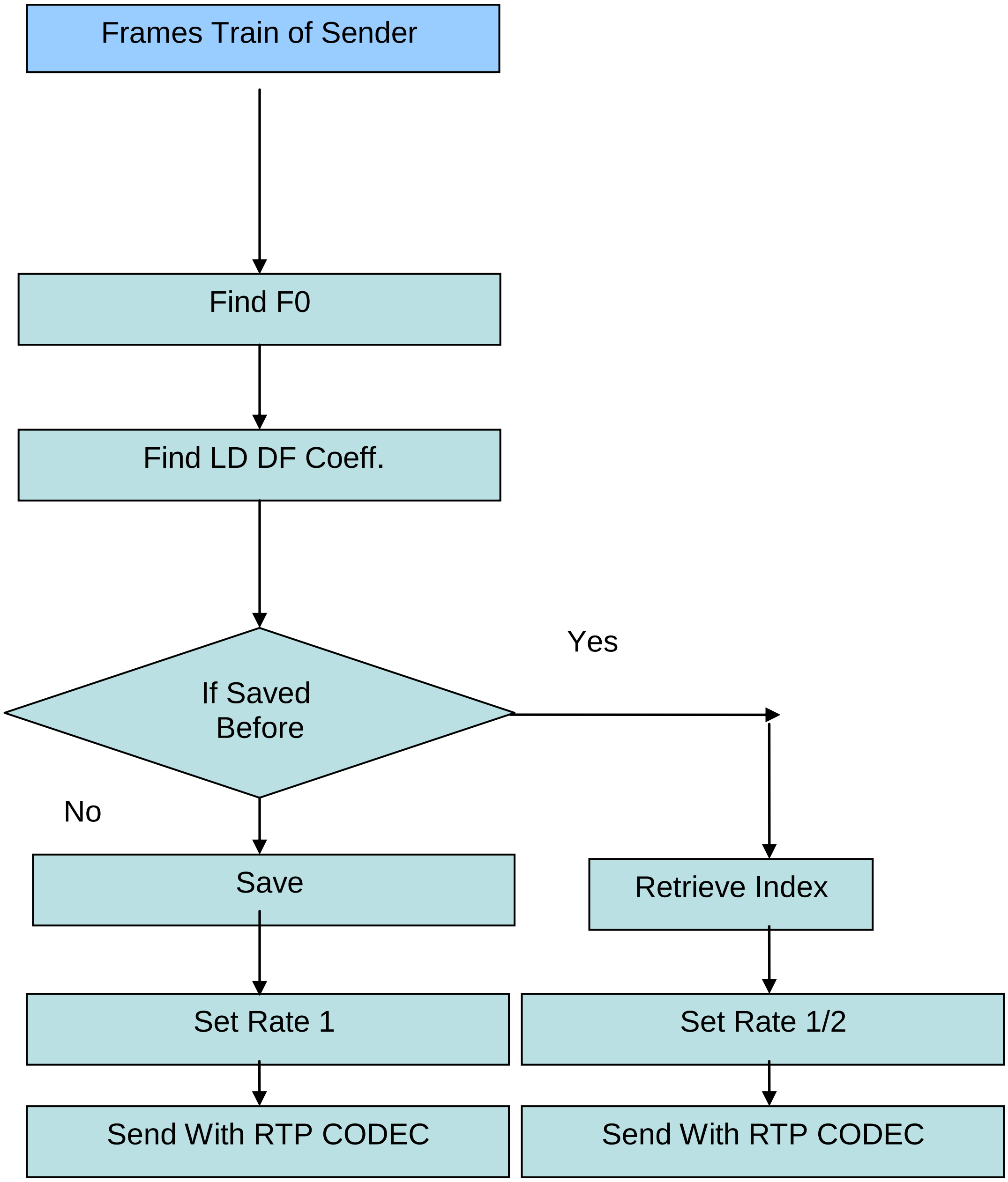


Figure 5.3: Sender Algorithm

5.6 Receiver Responsibilities

- Check for packet validity (beyond the scope).
- Expand the packet and organize its contents according to the frame's time stamp.
- Deduce the frames types from the packet by reading octet zero.
- If its of rate one deliver to buffer with respect to time stamp counter
- If it is of rate zero, use the index to retrieve the value of the coefficients and then place in the reading buffer accordingly with time stamp.

Receiver Flow Chart:

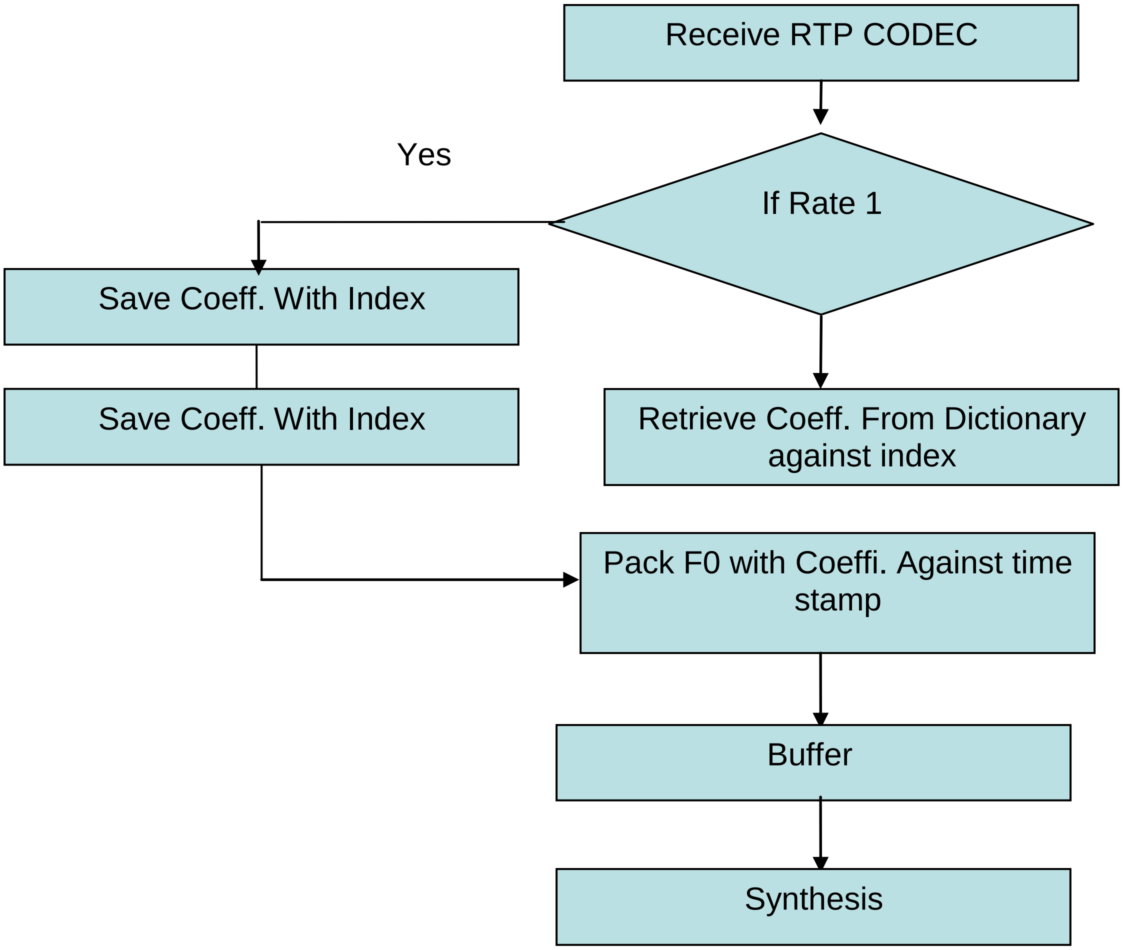


Figure 5.4: Receiver Algorithm

Conclusion

Voice over Internet Protocol (VoIP), is a new technology that aims to transmit voice as packets over the internet, so that the wired and the wireless systems will be holding the same type of data and serve the same purpose, rather than having two separate systems, one for voice and another for data. This actually is part of a bigger problem which is converting all types of communication into data communication and converge to the point that the use of different media will serve the same purpose, and follow the same constrains, respecting the privacy of each type of media, i.e., video constrains and audio constrains, etc.

The data rate reduction approaches are subjected about these parameters, as seen in chapter two, some approaches deals with the digital order, this is seen in the new iLBC, but most of the approaches deals with the excitation information as in CELP, and others deals with reducing the data rate by reducing the window size as LD-CELP.

Our approach worked with the LPC coefficients as seen in chapter four.

We have started from a basic idea of the correlation between the odd linear prediction coefficients together, and the even linear prediction coefficients together, this have lead to the idea of having a possible replacement for these coefficients, but the correlation was not a reliable measure and not a strong base for the generalization of the concept, since this correlation is caused by the properties of the normal equation, and we think that it applies when the noise term in the normal equation is minimized, we know already that the noise term might not be minimized even for the voiced information.

We avoided the use of quantization of the coefficients since it modifies the whole characteristics of the filter, so we thought of using values that the system or the speaker already produces, and use these values to express the values that fall around those values. We have built up a dictionary of digital filters coefficients and compared the values that instantaneously encountered to that dictionary entry, a smooth comparison was carried out using the maximum accepted distance concept, and data rate reduction was achieved through the transmission of the dictionary entries rather than the whole coefficients string for the filter.

Quality measurements were carried out using the Segmental signal to noise ratio and the MOS test. The results were good for the MOS, but for the segmental signal to noise ratio, it was increasing in the negative direction as we increased the acceptable distance, and the MOS result was falling down as well. The richer the dictionary, the highest replacements are achieved preserving a good quality and vice versa. But I think we are able to say that digital filter coefficients replacement is a useable approach.

After we are done with this, we had to embed this result in the VoIP framework. This is done with suggesting a modification on the RTP packet that is compatible with what we are doing, an RTP packet that can handle the different data rates and can have the necessary information for the driving of the replacement process. This dictated a definition of a process flow for the sender and the receiver as well. Sender responsibilities were clarified and the cooperating responsibilities on the receiver side too.

We have stopped our research on this point leaving many questions unanswered, this might be a good entry point for the future work, these questions might be:

- What will the result be if different conditions were used for the comparison process? meaning having different maximum accepted distances, each is applied on a certain shifting element.
- Can we use this approach in the data clustering for the voice recognizers?
- Can this approach be generalized for different excitation types ?

These are the last words we finish our research with, hoping we have added a tiny drop to the endless ocean of knowledge, and hoping we are able to gain the blessings of God.

The End

Ramallah 2005-06-27

Annex I

The Matlab Code

```
clc; % clear the command line
clear all; % clear the workspace
%
% system constants
% ---------------
InputFilename = 'C:\MT\originaL\female1.wav'; %change it
according to your wave files
FileOutput = 'c:\MT\Modified\male11.wav';
[inspeech, Fs, bits] = wavread(InputFilename); % read the
wavefile

outspeech2 = speechcoder2(inspeech);

disp('Press a key to play the original sound!');
pause;
soundsc(inspeech, Fs);

disp('Press a key to play Modified sound!');
pause;
soundsc(outspeech2, Fs);
wavwrite (outspeech2,Fs,FileOutput);

End


function [aCoeff,resid,pitch,G,parcor,stream] =
proclpc(data,sr,L,fr,fs,preemp)
```

```
% USAGE: [aCoeff,resid,pitch,G,parcor,stream] =
proclpc(data,sr,L,fr,fs,preemp)
%
% This function computes the LPC (linear-predictive coding)
coefficients that
% describe a speech signal. The LPC coefficients are a
short-time measure of
% the speech signal which describe the signal as the output
of an all-pole
% filter. This all-pole filter provides a good description
of the speech
% articulators; thus LPC analysis is often used in speech
recognition and
% speech coding systems. The LPC parameters are
recalculated, by default in
% this implementation, every 20ms.
%
% The results of LPC analysis are a new representation of
the signal
% s(n) = G e(n) - sum from 1 to L a(i)s(n-i)
% where s(n) is the original data. a(i) and e(n) are the
outputs of the LPC
% analysis with a(i) representing the LPC model. The e(n)
term represents
% either the speech source's excitation, or the residual:
the details of the
% signal that are not captured by the LPC coefficients. The
G factor is a
% gain term.
%
% LPC analysis is performed on a monaural sound vector
(data) which has been
% sampled at a sampling rate of "sr". The following
optional parameters modify
% the behaviour of this algorithm.
% L - The order of the analysis. There are L+1 LPC
coefficients in the output
% array aCoeff for each frame of data. L defaults to 13.
% fr - Frame time increment, in ms. The LPC analysis is
done starting every
% fr ms in time. Defaults to 20ms (50 LPC vectors a second)
generating 50
% victors per socound to have he main need to get th
% fs - Frame size in ms. The LPC analysis is done by
windowing the speech
% data with a rectangular window that is fs ms long.
Defaults to 30ms
% preemp - This variable is the epsilon in a digital one-
zero filter which
% serves to preemphasize the speech signal and compensate
for the 6dB
% per octave rolloff in the radiation function. Defaults to
.9378.
%
% The output variables from this function are
% aCoeff - The LPC analysis results, a(i). One column of L
numbers for each
% frame of data
% resid - The LPC residual, e(n). One column of sr*fs
samples representing
% the excitation or residual of the LPC filter.
% pitch - A frame-by-frame estimate of the pitch of the
signal, calculated
% by finding the peak in the residual's autocorrelation for
each frame.
% G - The LPC gain for each frame.
% parcor - The parcor coefficients. The parcor coefficients
give the ratio
% between adjacent sections in a tubular model of the
speech
% articulators. There are L parcor coefficients for each
frame of
% speech.
% stream - The LPC analysis' residual or excitation signal
as one long vector.
% Overlapping frames of the resid output combined into a
new one-
% dimensional signal and post-filtered.
%
% The synlpc routine inverts this transform and returns the
original speech
% signal.
%


if (nargin<3), L = 11; end
if (nargin<4), fr = 20; end
if (nargin<5), fs = 30; end
if (nargin<6), preemp = .9378; end

[row col] = size(data)     %Invstigating the data sizes,
the row is the depth of the adusio file and the col is the
legnht of the files
if col==1 data=data'; end

nframe = 0;
msfr = round(sr/1000*fr); % Convert ms to samples
msfs = round(sr/1000*fs); % Convert ms to samples
duration = length(data)
speech = filter([1 -preemp], 1, data)'; % Preemphasize
speech
msoverlap = msfs - msfr;
ramp = [0:1/(msoverlap-1):1]'; % Compute part of window


for frameIndex=1:msfr:duration-msfs+1 % frame rate=20ms
frameData = speech(frameIndex:(frameIndex+msfs-1)); % frame
size=30ms
nframe = nframe+1;

autoCor = xcorr(frameData); % Compute the cross correlation
autoCorVec = autoCor(msfs+[0:L]);

% Levinson's method
err(1) = autoCorVec(1);

A = [];
for index=1:L

numerator = [1 A.']*autoCorVec(index+1:-1:2);
denominator = -1*err(index);

k(index) = numerator/denominator; % PARCOR coeffs
A = [A+k(index)*flipud(A); k(index)] ;
err(index+1) = (1-k(index)^2)*err(index);



end
%     nframe;
% A;
% k;
    aCoeff(:,nframe) = [1; A];
```

```
parcor(:,nframe) = k';

% Calculate the filter
% response
% by evaluating the
% z-transform
if 0
gain=0;
cft=0:(1/255):1;
for index=1:L
gain = gain + aCoeff(index,nframe)*exp(-i*2*pi*cft).^index;
end
gain = abs(1./gain);
spec(:,nframe) = 20*log10(gain(1:128))';
% plot(20*log10(gain));
% title(nframe);
% drawnow;
end

% Calculate the filter response
% from the filter's impulse
% response (to check above).
if 0
impulseResponse = filter(1, aCoeff(:,nframe), [1
zeros(1,255)]);
freqResp = 20*log10(abs(fft(impulseResponse)));
plot(freqResp);
end

errSig = filter([1 A'],1,frameData); % find excitation
noise

G(nframe) = sqrt(err(L+1)); % gain
autoCorErr = xcorr(errSig); % calculate pitch & voicing
information
[B,I] = sort(autoCorErr);
%B is the sorted version of the input array
%I is the lenght of the new sorted one (index)


num = length(I);
if B(num-1) > .0001*B(num)
pitch(nframe) = abs(I(num) - I(num-1));
else
pitch(nframe) = 0;
```

```
end

% calculate additional info to improve the compressed sound
quality
resid(:,nframe) = errSig/G(nframe);
if(frameIndex==1) % add residual frames using a trapezoidal
window
stream = resid(1:msfr,nframe);
else
stream = [stream;
overlap+resid(1:msoverlap,nframe).*ramp;
resid(msoverlap+1:msfr,nframe)];
end
if(frameIndex+msfr+msfs-1 > duration)
stream = [stream; resid(msfr+1:msfs,nframe)];
else
overlap = resid(msfr+1:msfs,nframe).*flipud(ramp);
end
end
% disp('the number of samples was ');nframe
% disp('what frame do you like to analyze ')
% dd= input('prompt_string')
% ff=aCoeff(:,dd) ;
% cc=parcor(:,dd);
nframe

% subplot(5,2,1); bar(ff)
% subplot(5,2,2); bar(cc)
% subplot(5,2,3); plot(pitch)
% subplot(5,2,4); plot(G)
% subplot(5,2,5); plot(B)
stream = filter(1, [1 -preemp], stream)';
%  savefile = 'c:\sample5_work\5_COEF_.mat'
savefile = 'c:\female1.mat'
% % savefile2 = 'c:\todaytest2.mat'
%
 temp=aCoeff;
 temp=temp';
 save(savefile,'temp');
% % save(savefile,'temp');
 Exit


function [ outspeech ] = speechcoder2( inspeech )
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%
%
% Speech Coding using Linear Predictive Coding (LPC)
% The desired order can be selected in the system constants
section.
% For the excitation the residual signal is used. In order
to decrease the
% bitrate, the residual signal is discrete cosine
transformed and then
% compressed. This means only the first 50 coefficients of
the DCT are kept.
% While most of the energy of the signal is stored there,
we don't lose a lot
% of information.
%
% Parameters:
% inspeech : wave data with sampling rate Fs
% (Fs can be changed underneath if necessary)
%
% Returns:
% outspeech : wave data with sampling rate Fs
% (coded and resynthesized)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% arguments check
% ---------------
if ( nargin ~= 1)
error('argument check failed');
end;
savefile = 'c:\mt\dct_resd.mat'
%
% system constants
% ----------------
Fs = 16000; % sampling rate in Hertz (Hz)
Order = 10; % order of the model used by LPC


%
% main
% ----

% encoded the speech using LPC
```

```
[aCoeff, resid, pitch, G, parcor, stream] =
proclpc(inspeech, Fs, Order);

% perform a discrete cosine transform on the residual
% resid = dct(resid);
resid = dct(resid);
temp=resid;
[a,b] = size(resid);
% only use the first 40 DCT-coefficients this can be done
% because most of the energy of the signal is conserved in
these coeffs
%resid = [ resid(1:40,:)];
resid = [ resid(1:40,:); zeros(430,b) ];
% quantize the data
resid = uencode(resid,4);
resid = udecode(resid,4);

% perform an inverse DCT
% resid = idct(resid);

% % savefile2 = 'c:\todaytest2.mat'
%

temp=temp';
% save(savefile,'temp');
save(savefile,'temp');
resid = idct(resid);
% add some noise to the signal to make it sound better
noise = [ zeros(50,b); 0.01*randn(430,b) ];
resid = resid + noise;

% decode/synthesize speech using LPC and the compressed
residual as excitation
outspeech = synlpc2(aCoeff, resid, Fs, G);


function synWave = synlpc2(aCoeff,source,sr,G,fr,fs,preemp)
% USAGE: synWave = synlpc(aCoeff,source,sr,G,fr,fs,preemp);
%
% This function synthesizes a (speech) signal based on a
LPC (linear-
% predictive coding) model of the signal. The LPC
coefficients are a
% short-time measure of the speech signal which describe
the signal as the
```

```
% output of an all-pole filter. This all-pole filter
provides a good
% description of the speech articulators; thus LPC analysis
is often used in
% speech recognition and speech coding systems. The LPC
analysis is done
% using the proclpc routine. This routine can be used to
verify that the
% LPC analysis produces the correct answer, or as a
synthesis stage after
% first modifying the LPC model.
%
% The results of LPC analysis are a new representation of
the signal
% s(n) = G e(n) - sum from 1 to L a(i)s(n-i)
% where s(n) is the original data. a(i) and e(n) are the
outputs of the LPC
% analysis with a(i) representing the LPC model. The e(n)
term represents
% either the speech source's excitation, or the residual:
the details of the
% signal that are not captured by the LPC coefficients. The
G factor is a
% gain term.
%
% LPC synthesis produces a monaural sound vector (synWave)
which is
% sampled at a sampling rate of "sr". The following
parameters are mandatory
% aCoeff - The LPC analysis results, a(i). One column of
L+1 numbers for each
% frame of data. The number of rows of aCoeff determines L.
% source - The LPC residual, e(n). One column of sr*fs
samples representing
% the excitation or residual of the LPC filter.
% G - The LPC gain for each frame.
%
% The following parameters are optional and default to the
indicated values.
% fr - Frame time increment, in ms. The LPC analysis is
done starting every
% fr ms in time. Defaults to 20ms (50 LPC vectors a second)
% fs - Frame size in ms. The LPC analysis is done by
windowing the speech
```

```
% data with a rectangular window that is fs ms long.
Defaults to 30ms
% preemp - This variable is the epsilon in a digital one-
zero filter which
% serves to preemphasize the speech signal and compensate
for the 6dB
% per octave rolloff in the radiation function. Defaults to
.9378.
%
%

if (nargin < 5), fr = 20; end;
if (nargin < 6), fs = 30; end;
if (nargin < 7), preemp = .9378; end;

msfs = round(sr*fs/1000);
msfr = round(sr*fr/1000);
msoverlap = msfs - msfr;
ramp = [0:1/(msoverlap-1):1]';
[L1 nframe] = size(aCoeff); % L1 = 1+number of LPC coeffs

[row col] = size(source);
if(row==1 | col==1) % continous stream; must be windowed
postFilter = 0; duration = length(source); frameIndex = 1;
for sampleIndex=1:msfr:duration-msfs+1
resid(:,frameIndex) = source(sampleIndex:(sampleIndex+msfs-
1))';
frameIndex = frameIndex+1;
end
else
postFilter = 1; resid = source;
end

[row col] = size(resid);
if col<nframe
nframe=col;
end

for frameIndex=1:nframe
A = aCoeff(:,frameIndex);
residFrame = resid(:,frameIndex)*G(frameIndex);
synFrame = filter(1, A', residFrame); % synthesize speech
from LPC coeffs
```

```
if(frameIndex==1) % add synthesized frames using a
trapezoidal window
synWave = synFrame(1:msfr);
else
synWave = [synWave; overlap+synFrame(1:msoverlap).*ramp;
...
synFrame(msoverlap+1:msfr)];
end
if(frameIndex==nframe)
synWave = [synWave; synFrame(msfr+1:msfs)];
else
overlap = synFrame(msfr+1:msfs).*flipud(ramp);
end
end;

if(postFilter)
synWave = filter(1, [1 -preemp], synWave);
end
```

```
function SNR = segsnr(orig_file, coded_file),

% SNR = SEGSNR(ORIG_FILE, CODED_FILE)
%
% Output : SNR is the segmental SNR
%
% Inputs: ORIG_FILE is the orignial speech file sampled at
16 kHz
%         CODED_FILE is the coded speech file sampled at 16
kHz
%
% NOTE: BOTH FILES SHOULD HAVE THE SAME LENGTH !

if nargin < 2,
    errordlg('SNR = segsnr(orig_file, coded_file)')
end

S1 = wavread(orig_file);
S2 = wavread(coded_file);
L = length(S1);
WINDOWLENGTH = 20*16;
```

```
if length(S1) ~= length(S2)
    errordlg('File do not have the same length');
else
    F1 = enframes(S1, WINDOWLENGTH);
    F2 = enframes(S2, WINDOWLENGTH);
    NFRAMES = size(F1, 1);              % The frames are
line by line
    for i = 1 : NFRAMES
        segsnr(i) = 20 * log10 ( norm(F1(i,:)) /
norm(F1(i,:)-F2(i,:)) );
    end
end

SNR = mean(segsnr);


%--------------------

function f=enframes(x,win,inc)


nx=length(x);
nwin=length(win);
if (nwin == 1)
   len = win;
else
   len = nwin;
end
if (nargin < 3)
   inc = len;
end
nf = fix((nx-len+inc)/inc);
f=zeros(nf,len);
indf= inc*(0:(nf-1)).';
inds = (1:len);
f(:) = x(indf(:,ones(1,len))+inds(ones(nf,1),:));
if (nwin > 1)
    w = win(:)';
    f = f .* w(ones(nf,1),:);
end
Alwan 2002
```

Annex II

Java Code

```
import java.io.*;
import java.util.*;
import java.text.DecimalFormat;


public class tProfileManger  {
/////////////////////


//////////////////////////////////////



public static boolean compare(double dum1[],double
dum2[],double t)
{


//print1(dum1,dum2);
int i=0;
boolean x= true;
while (x&&i<11)
{
//comparesmooth(dum1[i+1],convertformat(dum2[i]));
if (comparesmooth(dum1[i],convertformat(dum2[i]),t))
i++;

else return false ;
}

System.out.println("Am A CowBoy......:D:D:D:D:D:D");
return true;
}
public static void print1(double dum11[],double dum12[])
{



for (int j=0;j<dum11.length;j++)
  {System.out.println(dum11[j]);


}
```

```
 System.out.println("--------------------------------------
- ");

 for (int j=0;j<dum11.length;j++)
   {dum12[j]=convertformat(dum12[j]);
   System.out.println(dum12[j]);
 }

System.out.println("***************************************
*** ");



}

public static double  convertformat(double x)
{DecimalFormat resultFormat = new DecimalFormat("0.00");

double d;
String s=new String() ;
  s=java.lang.String.valueOf(x);
  s=resultFormat.format(x);

x=java.lang.Double.parseDouble(s);
d=java.lang.Double.parseDouble(s);
//System.out.println(s);
// System.out.println("#################################
");
return d;
}
public static  boolean comparesmooth(double x, double
y,double t)

{
  if ((x>0)&(y>0)&&(Math.abs((x)-(y))<=t)) return true;
 else
   if ((Math.abs(Math.abs(x)-Math.abs(y))<=t)) return true;

return false;
}

}// end of the whole class


import java.io.*;
import java.util.*;



public class BaseMaker {

  public static void main(String[] args) {

  try {
    BufferedWriter out1 = new BufferedWriter(new
FileWriter("c:/islam_outfile"));
  }
  catch (IOException ex) {
  }


  try {


    readText("c:/islam_all.txt");

  }
  catch (IOException e) {
  e.printStackTrace();
  }
  }
  /*Strarting the work here */



//public static void








  public static void readText(String fileName) throws
IOException {

  Reader reader = new FileReader(fileName);
  double[][] c = new double [13600][12];
  int i=0;
  BufferedReader bufferedReader = new
BufferedReader(reader);
  String nextLine;
  while ((nextLine = bufferedReader.readLine()) != null) {

  StringTokenizer tokenizer = new
StringTokenizer(nextLine);
//System.out.println(i);


  try {
    c[i][0]=Double.parseDouble(tokenizer.nextToken());
    c[i][1]=Double.parseDouble(tokenizer.nextToken());
    c[i][2]=Double.parseDouble(tokenizer.nextToken());
    c[i][3]=Double.parseDouble(tokenizer.nextToken());
    c[i][4]=Double.parseDouble(tokenizer.nextToken());
    c[i][5]=Double.parseDouble(tokenizer.nextToken());
    c[i][6]=Double.parseDouble(tokenizer.nextToken());
    c[i][7]=Double.parseDouble(tokenizer.nextToken());
    c[i][8]=Double.parseDouble(tokenizer.nextToken());
    c[i][9]=Double.parseDouble(tokenizer.nextToken());
    c[i][10]=Double.parseDouble(tokenizer.nextToken());


  i++;
   // c[i][11]=Double.parseDouble(tokenizer.nextToken());
  }
  catch (NumberFormatException ex) {

    System.out.println("Hello ,,,,Check your Inputs");
  }
       for(int
knops=0;knops<c.length;knops++){c[knops][11]=1;



       }
    ;




  }bufferedReader.close();
  System.out.print("this pashe is done ");
reductio(c)
;  //
```

```
//smothcompare(c);
//tostring(c);
  }//method end

  public static void reductio(double[][] a )

   {
     double[] row1 = new double[11];
     double[] row2 = new double[11];
     double[][] Coeff = new double[13600][11];
     int count=0;


    try {

      BufferedWriter out3 = new BufferedWriter(new
FileWriter("c:/islam_MyVector.mat"));

      for(int RedIndex=0;RedIndex<a.length;RedIndex++)
      {//begin out for
       if (a[RedIndex][11]==1)
                    {
                      for (int tempo=0;tempo<11;tempo++)

{Coeff[count][tempo]=a[RedIndex][tempo];
                          row2[tempo]=a[RedIndex][tempo];
                         }count++;
                          for (int h=0;h<13600;h++)
                              {//visiting loop
                              for (int
tempx=0;tempx<11;tempx++)
                                row1[tempx]=a[h][tempx];


                                if
(rowcom(row1,row2)&&RedIndex!=h)
                                {a[h][11]=0;
                                  a[RedIndex][11]=0;
                                  System.out.println("Frame
"+h+"and frame "+RedIndex+"  are equal rows");
                                }
                              }

                    }
```

```
         }//end out fore


         for (int f=0;f<=count;f++)
     {
       for (int ff=0;ff<11;ff++)
     {   out3.write(String.valueOf(Coeff[f][ff]+"\t"));

     }
     out3.newLine();
}
out3.close();
    }//end try
    catch (Exception ex) {
    }

     //////////////////////////

     //////////////////////////


System.out.println(">>>>>>>>>>>>>>>>>>>>>>>>"+count+"<<<<<<
<<<<<<<<<<<<<<<<<<<");


  }//end reduction

public static boolean rowcom(double dum1[],double dum2[])
{

int i=0;
boolean x= true;
while (x&&i<11)
{
if (dum1[i]==dum2[i])
i++;
else return false ;

}


return true;

}
```

```
}//end file
```

References

1. Alku P., Backstrom T. (2004). “Linear predictive method for improved spectral modeling of lower frequencies of speech with small prediction order”, IEEE Transaction on speech and audio Processing, VOL. 12, NO. 2, IEEE.

2. Allnatt J. (1975). “Subjective Rating and Apparent Magnitude” International Journal Man - Machine Studies, vol. 7.

3. Alwan Abeer (2002). “ Wideband Speech Coding with Linear Predictive Coding(LPC)”, University of California at Los Angeles, Department of Electrical Engineering, LA, CA.

4. Bateman A., Paterson-Stephnes I. (2002). “The DSP Handbook,

Algorithms, Applications and Design Techniques", Prentice Hall, London, UK.

5. Beerends J. G., Stemerdink J. A. (1994). "A Perceptual Speech Quality Measure Based on a Psychoacoustic Sound Representation" J. Audio Eng. Soc., vol. 42, no. 3.

6. Benjamin W. Wah (2005). "LSP-Based Multiple-Description Coding for Real-Time Low Bit-Rate Voice Over IP", IEEE Transactions on Multimedia, Vol. 7, No. 1, IEEE.

7. Beritelli F. (1999). "A Modified CS-ACELP Algorithm For Variable Rate Speech Coding Rebuts In Noisy Environments", IEEE signals Processing Letters. Vol. 6, No. 2, IEEE.

8.. Bryan E. Carne (2004). "A Professional Guide to Data Communication in a TCP/IP World", Artech House, Norwood, MA.

9. Childer (2000). "Speech Processing and Synthesis Toolbox", Wiley, New York, NY.

10. Chong N., Cox R. (2003). “An intelligibility Enhancement for Mixed Excitation Linear prediction Coder”, IEEE signal Processing Letters. Vol. 10, No. 9, IEEE.

11. Chou Wu, Juang Biing (2002). “Pattern Recognition in Speech and Language Processing”, CRC, Boca, FL.

12. Chu W. C. (2003). “Window Optimization In Linear Prediction Analysis”, IEEE Transaction on Speech and Audio Processing, Vol. 4, No. 6, IEEE.

13. Chu Wai C. (2003). “Speech Coding Algorithms”, Wiley, New York, NY.

14. Cisco Systems (2002). “Cisco Voice Over IP”, Cisco Press, World Wide.

15. Clark A. D. (April 2001), “Modeling the Effects of Burst Packet Loss and Recency on Subjective Voice Quality (2001).” in Proc. of

IPTEL’2001, New York, USA.

16. Clark A. D., (2001). “Modeling the Effects of Burst Packet Loss and Recency on Subjective Voice Quality” in Proc. of IPTEL’2001, New York, USA.

17. Cole R. G. and J. Rosenbluth (2001) “Voice over IP Performance Monitoring”, Journal on Computer Communications Review, Vol. 31.

18. Diniz P., da Silvo E., Netto S. (2002). “Digital Signal Processing, System and Analysis”, Cambridge University Press, Cambridge, UK.

19. EURESCOM Project P905-PF (2000). “AQUAVIT - Assessment of Quality for Audio-Visual signals over Internet and UMTS – Deliverable 2: Methodology for subjective audiovisual quality evaluation in mobile and IP networks”.

20. European Telecommunications Standards Institute (1996). “Speech

Communication Quality from Mouth to Ear of 3.1 kHz Handset Telephony across Networks," Tech. Report. ETR 250.

21. European Telecommunications Standards Institute (1999). "Specification and Measurement of Speech Transmission Quality; Part 1: Introduction to Objective Comparison Measurement Methods for One-way Speech Quality Across Networks," ETSI Guide, EG 201 377-1 V1.1.1.

22. Galis A., Denazis S., Brou C., Klein C. (2004). "Programmable Networks foe IP Service Deployment", Artech House, Norwood, MA.

23. Giridhar Mandyam, Nasir Ahmed and Neeraj Magotra (1995). "Application of the Discrete Laguerren Transform to Speech Coding",Proceedings of the Conference Record of the Twenty-Ninth Asilomar Conference on Signals, Systems and Computers (ASILOMAR '95) IEEE.

24. GunzuhanE. , Motahan K. (2001). “Linear Prediction Based Packet Loss Concealment Algorithm for PCM Coded Speech”, IEEE Transaction on Speech and Audio Processing, Vol. 47, No. 10, IEEE.

25. Harma A. (2001). “Linear predictive Coding with Modified Filter Structure”, IEEE Transaction on Speech and Audio Processing, Vol. 47, No. 10, IEEE.

26. Hess Wolfgang (1983). “Pitch Determination of Speech Signal”, Springer-Verlag, Berlin, Germany.

27. International Telecommunication Union (1997). “Improvement of the P.861 Perceptual Speech Quality Measure”, ITU-T Contribution Com12-20.

28. International Telecommunication Union (1998). “Objective Quality Measurement of Telephone band (300-3400 Hz) Speech Codecs”, ITU-T Recommendation P.861.

29. International Telecommunication Union (2000). “The E-model, A Computational Model for Use in Transmission Planning”, ITU-T Recommendation G.107.

30. International Telecommunication Union (1996). “In-service, Non-intrusive Measurement Device Voice Service Measurements”, ITU-T Recommendation P.561.

31. International Telecommunication Union (1996). “Methods for Subjective Determination of Transmission Quality”, ITU Recommendation P.800.

32. International Telecommunication Union (1998). “Definition of Categories of Speech Transmission Quality”, ITU-T Recommendation G.109.

33. International Telecommunication Union (2000). “Analysis and Interpretation of INMD Voice service Measurements”, ITU-T Recommendation P.562.

34. International Telecommunication Union (2001). “Perceptual Evaluation of Speech Quality (PESQ), An Objective Method for End-to-end Speech Quality Assessment of Narrowband Telephone Networks and Speech Codecs”, ITU-T Recommendation P.862.

35. International Telecommunication Union (2003). “One-way Transmission Time”, ITU-T Recommendation G.114.

36. ITU H.232 Home itu.int/itudoc/itu-t/aap/sg16aap/history/h323am3

37. Jiang W., Schulzrinne H. (2002). “Speech Recognition Performance as an Effective Perceived Quality Predictor”, Proceedings of International Workshop on Quality of Service (IWQOS), Miami, FL, USA.

38. Johannesson N. O. (1997). “The ETSI Computation Model: A Tool for Transmission Planning of Telephone Networks”, IEEE Communications Magazine, IEEE.

39. Kabal Peter And Rafi Rabipour (1984). “Computational Considerations in Adaptive Transform Coding of Speech”, Proc. Biennial Symp. Commun.. Kingston, ON, USA.

40. Kain Alexander, Michael W. (2000). “Design And Evaluation of A Voice Conversion Algorithm Based On Spectral Envelope Mapping And Residual Prediction", Center for Spoken Language Understanding (CSLU),Oregon Graduate Institute, OR 97006, USA

41. Karrenberg (2002). “An Interactive Multimedia Introduction to Signals Processing”, Springer, Berlin, Germany.

42. Kataoka A. et al (1996). “An 8 kbps Conjugate Structure CELP Speech Coder”, IEEE Transaction on Speech and Audio Processing , Vol. 4,No. 6, IEEE.

43. Kondoz A. M. (1994). “Digital Speech Coding for Low Bit Rate Communication Systems”, Wiley, New York, NY.

44. KumarA., Gersho A. (1997). “LD-CELP Speech Condign with Nonlinear Prediction”, IEEE Signal Processing Letters. Vol. 4, No. 4, IEEE.

45. Lam E. (2004). “Analysis of the DCT Coefficient Distribution for the Document Coding”, IEEE Signal Processing Letters. Vol. 11, No. 2, IEEE.

46. Lee C., Shoham Y. (2003). “Trellis Code Exited Linear Prediction Speech Coding”, Transaction on Speech and Audio Processing, Vol. 4, No. 6, IEEE.

47. Markopoulou A. P. , Tobagi F. A. ,Karam M. (2002). “Assessment of VoIP Quality over Internet Backbones”, in Proc. of IEEE Infocom, Vol. 1, New York, USA.

48. Marven C.and Edwards G. (1996). “A Simple Approach to DSP”, Wiley, New York, NY.

49. McClellan J. and others (1998). “Signal Processing Using Matlab”,

Prentice-Hall, Sadler River, NJ.

50. MitRa S. K. (1998). “Digital Signal Processing, A Computer Based Approach”, McGraw-Hill, New York, NY.

51. Moller S., Berger J. (2002). “Describing Telephone Speech Codec Quality Degradations by Means of Impairment Factors,” J. Audio Eng. Soc., Vol. 50.

52. Ohrtman (2004), “Voice Over 802.11”, Artech House, Norwood, MA, USA.

53. Oppenheim, Nawab Hamid, Alan V., Alan S. Willsky (2000). “Signals and Systems”, Prentice-Hall, Sadler River, NJ.

54. Rabiner L., Schafer R. (1996). “Digital Processing of Speech Signals”, Prentice-Hall, E. Cliffs, NJ.

55. Rao A. V. et al. (2003). “Pitch Adaptive Window for Improved Excitation Coding in Low Rate CELP Coder”, Transaction on

Speech and Audio Processing, Vol. 4, No. 6, IEEE.

56. Rix A., Reynolds R., Hollier M. (1999).“Perceptual Measurement of End-to-end Speech Quality over Audio and Packet-based Networks”, in AES 106th Convention, Munich, Germany.

57. Rix A. W., Beerends J. G., Hollier M. P., Hekstra A. P. (2001). “Perceptual Evaluation of Speech Quality (PESQ) - A New Method for Speech Quality Assessment of Telephone Networks and Codecs”, in Proceedings of 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), IEEE.

58. Serizawa M., Gersho A. (1999). “Joint Optimization of LPC and Closed-loop Pitch Parameters in CELP Coders”, IEEE Signal Processing Letters. Vol. 6, No. 3, IEEE.

59. Singer A. C., Feder M. (1999). “Universal Linear Prediction by Model Order Weightning”, IEEE Transaction on Signal Processing,

Vol. 47, No. 10, IEEE.

71. Storn R. (1996). “Efficient Input Reordering for the DCT Based on a Real Valued Dimension in Time FFT”, IEEE signal Processing Letters. Vol. 3, No. 8, IEEE.

72. Swale K. (2001). “Voice Over IP: Systems and Solutions”, BT Exact, Institute of the Electrical Engineers, London, UK.

73. Tankelevich R. (2005). “Algorithms Complexity Analysis”, Personal Website.
http://www.mines.edu/~rtankele/

74. Voran S. (1999). “Objective Estimation of Perceived Speech Quality - Part I: Development of the Measuring Normalizing Block Technique”, IEEE Transaction on Speech and Audio Processing, Vol. 7. IEEE.

75. Watson A., Sasse M. A. (1998). “Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications”, in Proceedings of ACM Multimedia ’98, Bristol, England. ACM.

76. Wei B., Gibson J. (2003). "A New Discrete Spectral Modeling and an Application to CELP", IEEE Signal Processing Letters. Vol. 10, No. 4, IEEE.

77. Yamamoto et al. (2003). "Optimized FIR Approximation for Discrete-time IIR filters", IEEE Signal Processing Letters. Vol. 10, No. 9, IEEE.

78. Yang W. (1999). "Enhanced Modified Bark Spectral Distortion (EMBSD): An Objective Speech Quality Measure Based on Audible Distortion and Cognition Model", PhD Dissertation, Temple University, USA.

79. Yang W., Benhouchta M., Yantorno R. (1998). "Performance of a Modified Bark Spectral Distortion Measure as An Objective Speech Quality Measure," in Proc. of IEEE ICASSP, IEEE.

80. Young M., Vafin R. (2002). "Time Synchronization for VoIP Quality of Services" ,IEEE Internet Computing, IEEE.

Zhang J. , Yu T. (1997). “A 4.2 kbps Low Delay Speech Coder with
81. Modified CELP”, IEEE Signal Processing Letters. Vol. 4, IEEE.