

Scalable Wideband Coding of Speech

Raghunandan H K



Dept. of Electrical Engineering, IIT Madras, Chennai 600036

under the guidance of

Pankaj Rabha



Texas Instruments, Bangalore, India

Project undertaken for fulfillment of the Internship requirement for awarding the
Bachelor Degree in Engineering

Contents

Abstract	ii
List of Figures	iii
List of Acronyms	iv
1. Introduction	1
1.1 Introduction to speech	2
1.2 Speech Coding	3
1.2.1 Introduction to speech coding	3
1.2.2 Narrowband coders	4
1.2.3 Wideband coders	4
1.3 Scalable Coders	9
1.3.1 Introduction to scalable coders	9
1.3.2 Types of scalable coders	9
1.4 Scalable Wideband Coding	9
1.5 Summary	10
2. Motivation	11
2.1 Introduction to cable architecture	12
2.2 Introduction to Session Initiation Protocol (SIP)	13
2.3 Introduction to Session Description Protocol (SDP)	14
2.4 A basic call setup using SIP	16
2.5 Integration of Scalable Codecs into PacketCable	18
2.6 Conclusion	19
3. Scalable Wideband Coding	21
3.1 Introduction	22
3.2 Experiments	22
3.2.1 Experiments in the LPC domain	23
3.2.2 Experiments in the residual domain	28
3.3 Scalable Wideband Coders	31
3.3.1 Encoder	31
3.3.2 Decoder	33
3.4 Results	33
Conclusion and future work	38
References	39

Abstract

In this project, we propose a bandwidth scalable scheme for coding wideband speech to work in conjunction with a CELP based baseband coder like G.729. The scalability is two layered with one layer adding the band from 3.4 kHz – 5.6 kHz and the second layer adding the band from 5.6 kHz – 7.2 kHz at bitrates of 2.7 kbps and .9 kbps respectively above the baseband bitrate. We also investigate and propose a method to integrate scalable codecs into a PacketCable network as an application to the proposed scheme. Tests show that the proposed scheme in conjunction with G.729E base coder at 15.4 kbps achieves a quality slightly better than the G.722.2 coder at 15.8 kbps and a quality equivalent to the G.722 coder at 48 kbps.

List of Figures

Chapter 1

Figure 1.1: Sagittal plane X-ray of the human vocal apparatus	2
Figure 1.2: Block Diagram of the G.729 Encoder	5
Figure 1.3: Block Diagram of the G.729 Decoder	6
Figure 1.4: Block Diagram of the G.722.2 Encoder	7
Figure 1.5: Block Diagram of the G.722.2 Decoder	8

Chapter 2

Figure 2.1: Reference architecture	13
Figure 2.2: SIP session setup example with SIP trapezoid	17
Figure 2.3: An example call flow of session initiation for scalable codecs	19

Chapter 3

Figure 3.1: Lowband synthesis filter	23
Figure 3.2: Highband synthesis filter	24
Figure 3.3: Itakura distances for the different speech classes.	24
Figure 3.4: Lower band LSFs v/s higher band LSFs of the same frame.	26
Figure 3.5: Plot of 'a' v/s 'b'	27
Figure 3.6: Plot of 'b' v/s 'c'	27
Figure 3.7: Plot of 'a' v/s 'c'	28
Figure 3.8: Encoder of Experiment 1	29
Figure 3.9: Decoder of Experiment 1	29
Figure 3.10: Encoder of Experiment 2	30
Figure 3.11: Decoder of Experiment 2	30
Figure 3.12: Encoder of Experiment 3	31
Figure 3.13: Decoder of Experiment 3	31
Figure 3.14: Obtaining highband speech	32
Figure 3.15: Plot of mean PESQ values for all speakers	34
Figure 3.16: Plot of mean PESQ values for male speakers	35
Figure 3.17: Plot of mean PESQ values for female speakers	35
Figure 3.18: Plot of mean PESQ values for all speakers	37

List of Acronyms

ACB	- Adaptive Codebook
AMR-WB	- Adaptive Multirate Wideband Coder
CDF	- Charging Data Function
CELP	- Code Excited Linear Prediction
FCB	- Fixed Codebook
LPC	- Linear Predictive Coding
LSP	- Line Spectral Pairs
MOS	- Mean Opinion Score
PAM	- PacketCable Application Manager
PESQ	- Perceptual Evaluation of Speech Quality
QoS	- Quality of Service
RTP	- Realtime Transport Protocol
S-CSCF	- Serving Call Session Control Function
SDP	- Session Description Protocol
SIP	- Session Initiation Protocol
UDP	- User Datagram Protocol
UE	- User Equipment

CHAPTER 1

INTRODUCTION

1.1 Introduction to speech

1.1.1 Speech Production

Speech is produced by the human vocal apparatus shown in Figure 1.1. The vocal tract (dotted line) begins at the opening between the vocal cords (glottis) and ends at the lips. It thus consists of the pharynx and the oral cavity. The nasal tract begins at the velum and ends at the nostrils. When the velum is lowered the nasal tract is acoustically coupled to the vocal tract to produce the nasal sounds of speech.

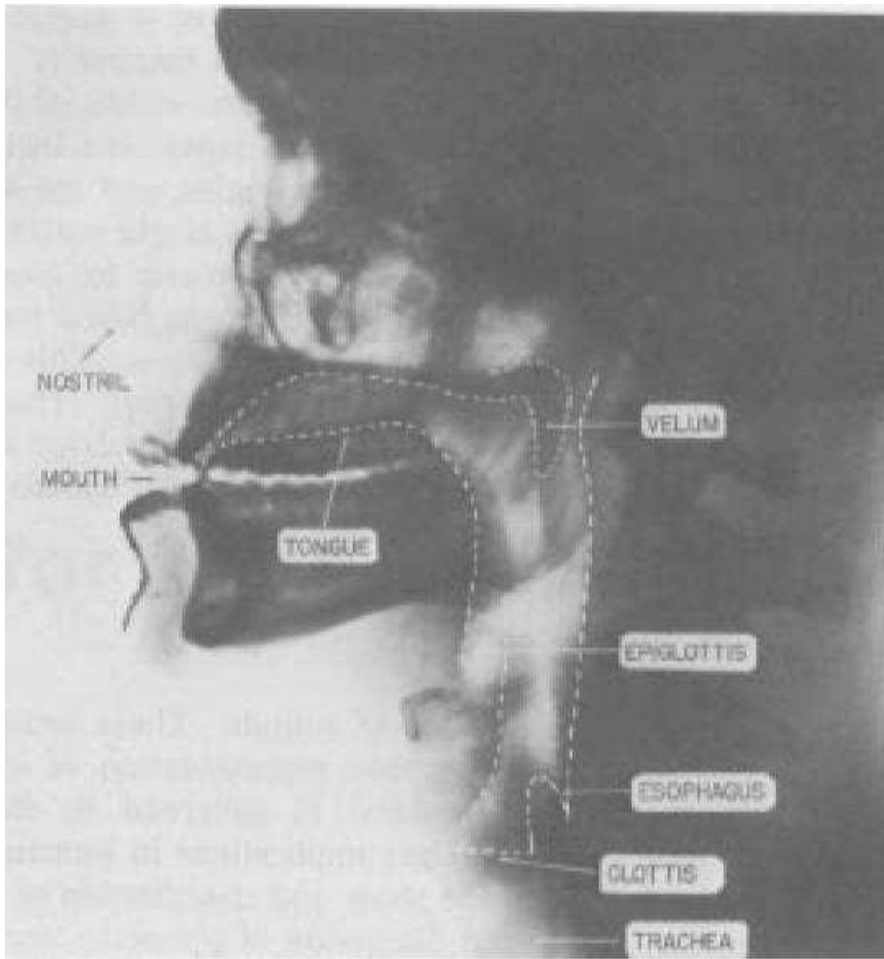


Figure 1.1: Sagittal plane X-ray of the human vocal apparatus (Flanagan et al [1])

The sub-glottal system consisting of the lungs, bronchi and the trachea serves as a source of energy for the production of speech. Speech is the acoustic wave resulting from the disturbance produced by the sub-glottal system and modulated by the vocal tract. The shape of the vocal tract plays a key role in the type of speech produced. Consequently, speech production can be modeled as an excitation filtered through a time-varying filter.

1.1.2 Classification of speech sounds

Speech sounds can be classified into 3 distinct classes according to their mode of excitation.

Voiced sounds are produced by forcing air through the glottis with the tension of the vocal cords adjusted so that they vibrate in a relaxed oscillation, thereby producing quasi periodic pulses of air which excite the vocal tract. Some of the voiced sounds are /i/ (as in we), /e/ (as in chase) etc.

Fricative or unvoiced sounds are generated by forming a constriction at some point in the vocal tract, and forcing air through the constriction at high enough velocity to cause turbulence. “sh” (as in should) is an example of an unvoiced sound.

Plosive sounds result from making a complete closure in the vocal tract, building up pressure behind the closure and abruptly releasing it. Plosive excitation is involved in creating the “cha” sound of chase.

Most languages, including English, can be described by a set of distinctive sounds called phonemes. The four broad classes of phonemes for English are vowels, diphthongs, semivowels, and consonants. These are further divided into sub classes. A detailed description of the speech production process and the classification and properties of speech sounds is given in [2].

1.2 Speech Coding

1.2.1 Introduction to speech coding

For speech to be processed effectively, it is necessary to represent speech digitally. This is done by a D/A converter which samples the input speech and quantizes it to a finite number of levels. Quantization is essentially an irreversible process and the difference between the original speech signal and the quantized signal is called quantization noise. This quantization noise depends on the type of quantizer used and the number of bits used to represent one sample. Usually, 16-bit or 8-bit linear quantizers are used. This gives rise to a huge bitrate for speech (128 kbps for 16 bit speech at 8 kHz sampling). This signal is of very high quality but puts a huge load on storage and transmission systems. Hence speech coding techniques are used to reduce the bitrate of speech.

Speech coding is the act of transforming speech into a different representation. The transformation can be reversible, that is the input speech can be recovered exactly by a corresponding inverse transformation, or it can be irreversible, where the input speech cannot be recovered exactly. In any coding system, the main idea is to reduce the bitrate

while still maintaining an acceptable perceptual quality of speech. Refer [3] for a tutorial review of speech coding techniques.

1.2.2 Narrowband Coders

Human speech consists of frequencies from 0 – 8 kHz. But most of the information is contained in (300 Hz – 3.4 kHz). Hence many coders restrict speech to this band before coding. These coders are called narrowband coders. In general, the narrowband speech contains first three to four formants of speech. The first two formants are decisive for the formation of the different vowels and phonemes. The bandwidth limitation of the narrowband speech imposes a limit on the quality of speech.

1.2.2.1 The G.729 Coder

One of the most popular narrowband coder is the G.729 coder. It is based on the Code Excited Linear Prediction (CELP) model with an algebraic structure for the fixed codebook [4]. This coder operates at 8 kbps and produces toll quality speech.

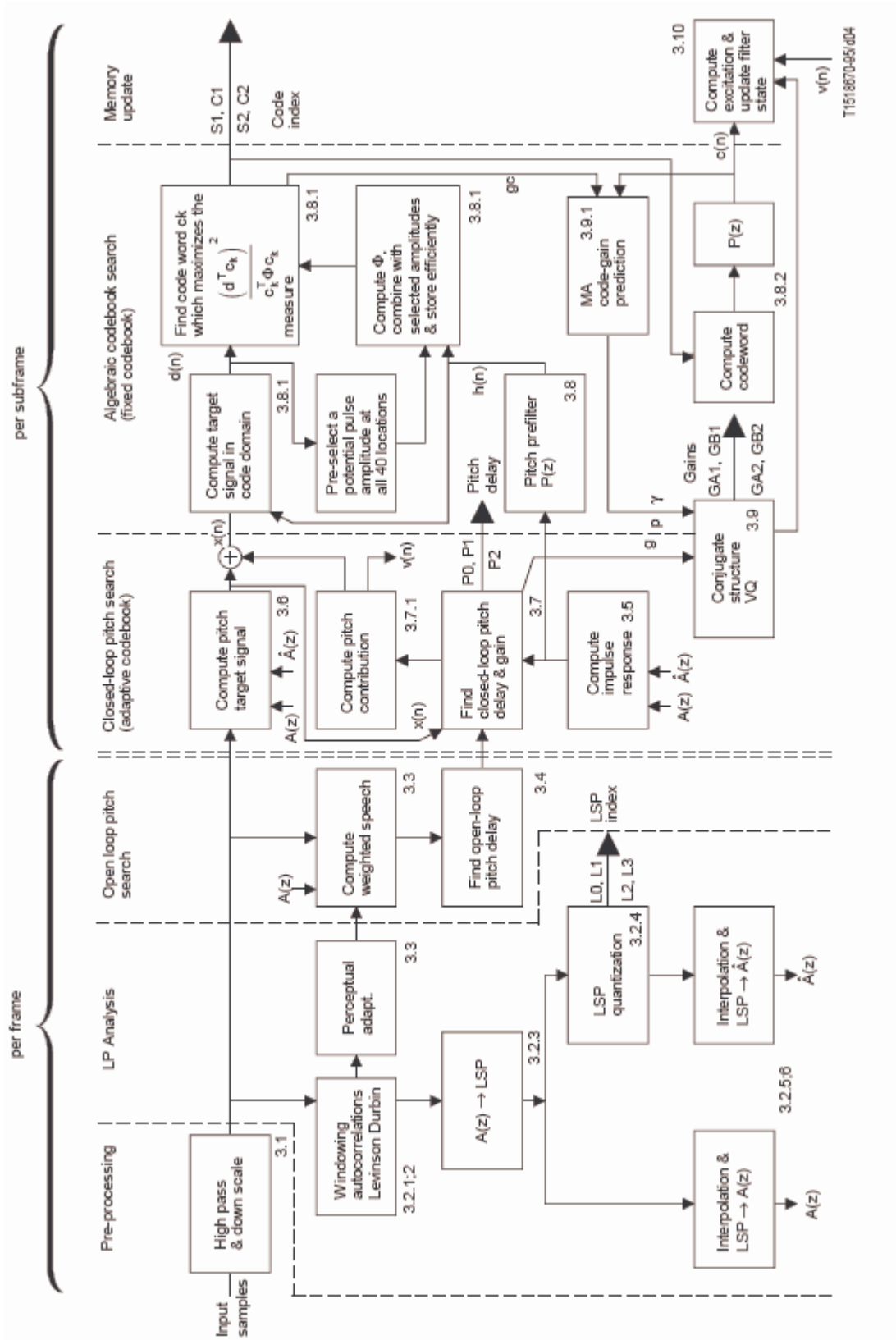
Some of the key features of the G.729 coder are:

- Sampling rate 8 kHz
- Frame size of 10 ms
- Uses Levinson Durbin algorithm [5] for the LPC filter
- A single tap Long term filter is used to remove the fine structure
- The resulting excitation is modeled with Fixed codebooks
- Mean Opinion Score (MOS) : 4.1
- Algorithmic delay : 15 ms

The Encoder and Decoder Block diagrams for the G.729 codec are given in Figure 1.2 and Figure 1.3 respectively. A detailed description of the working of the G.729 codec is given in [4].

1.2.3 Wideband Coders

Wideband speech is made of frequencies between 50Hz to 7 kHz and is normally sampled at 16 kHz. The quality of wideband speech is close to face-to-face communications quality. Hence, the usage of wideband speech results in major subjective improvements in speech quality. The low frequency components between 50Hz to 300Hz that is not present in the narrowband speech contributes to naturalness, presence and comfort in perceived quality. The high frequency components between 3.4 kHz and 7 kHz provides better fricative differentiation, hence increases intelligibility. Speech of bandwidth 50Hz to 7 kHz not only increases the intelligibility and naturalness of speech, but also adds feeling of transparent communication.



T1516870-95:004

Figure 1.2: Block Diagram of the G.729 Encoder

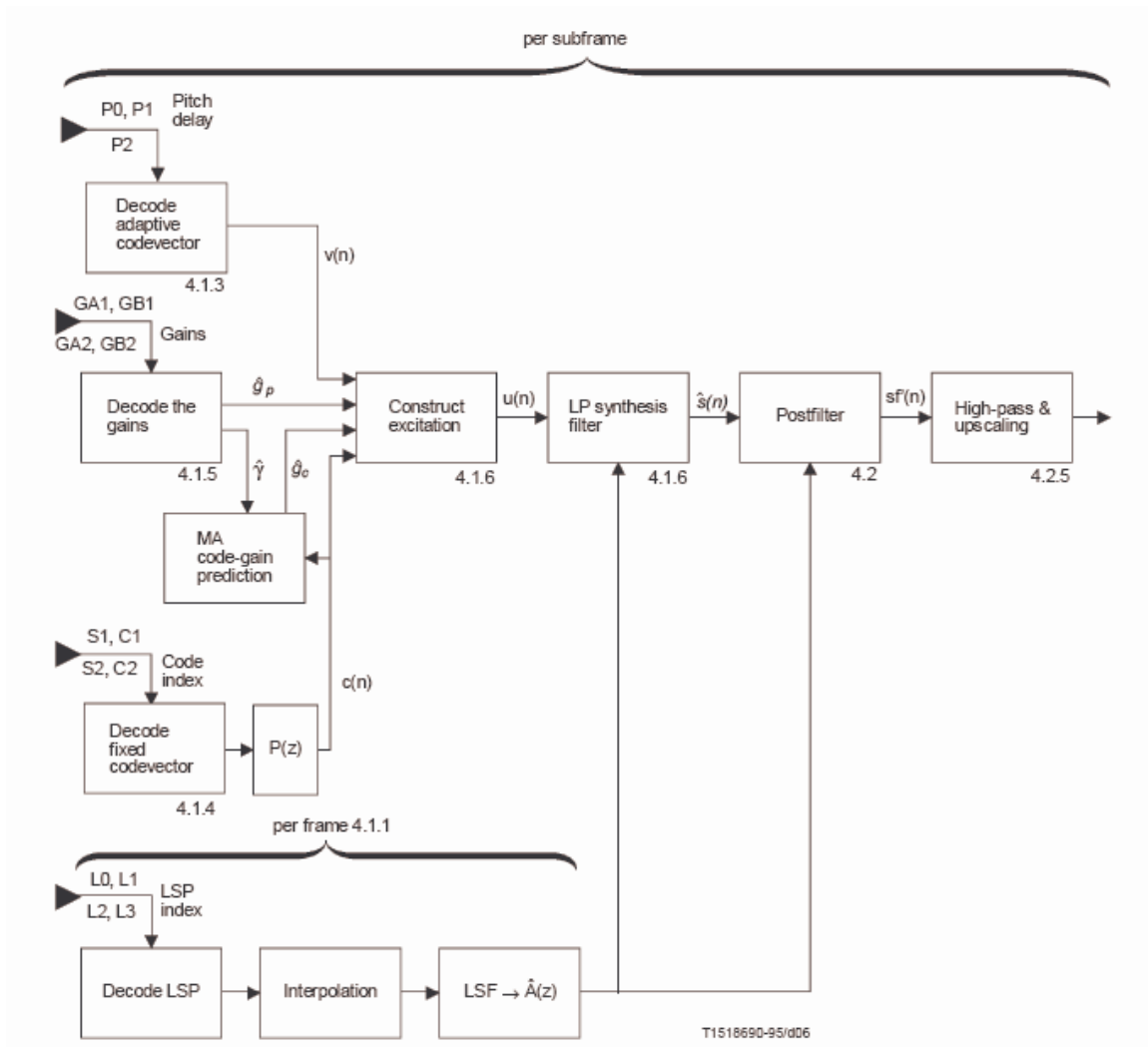


Figure 1.3: Block Diagram of the G.729 Decoder

1.2.3.1 The G.722.2 Coder

The G.722.2 Coder provides wideband speech at around 16 kbps using Adaptive Multirate Wideband (AMR-WB) [7]. The keys features of this coder are:

- Sampling rate of 16 kHz
- Frame size 20 ms
- Voice Activity Detection is used
- Two sub-bands from (0 – 6.4 kHz) and (6.4 kHz – 7.2 kHz)
- Different codebooks are used for the different modes

The Encoder and Decoder Block diagrams for the G.722.2 codec are given in Figure 1.4 and Figure 1.5 respectively. A detailed description of the working of the G.722.2 codec is given in [7].

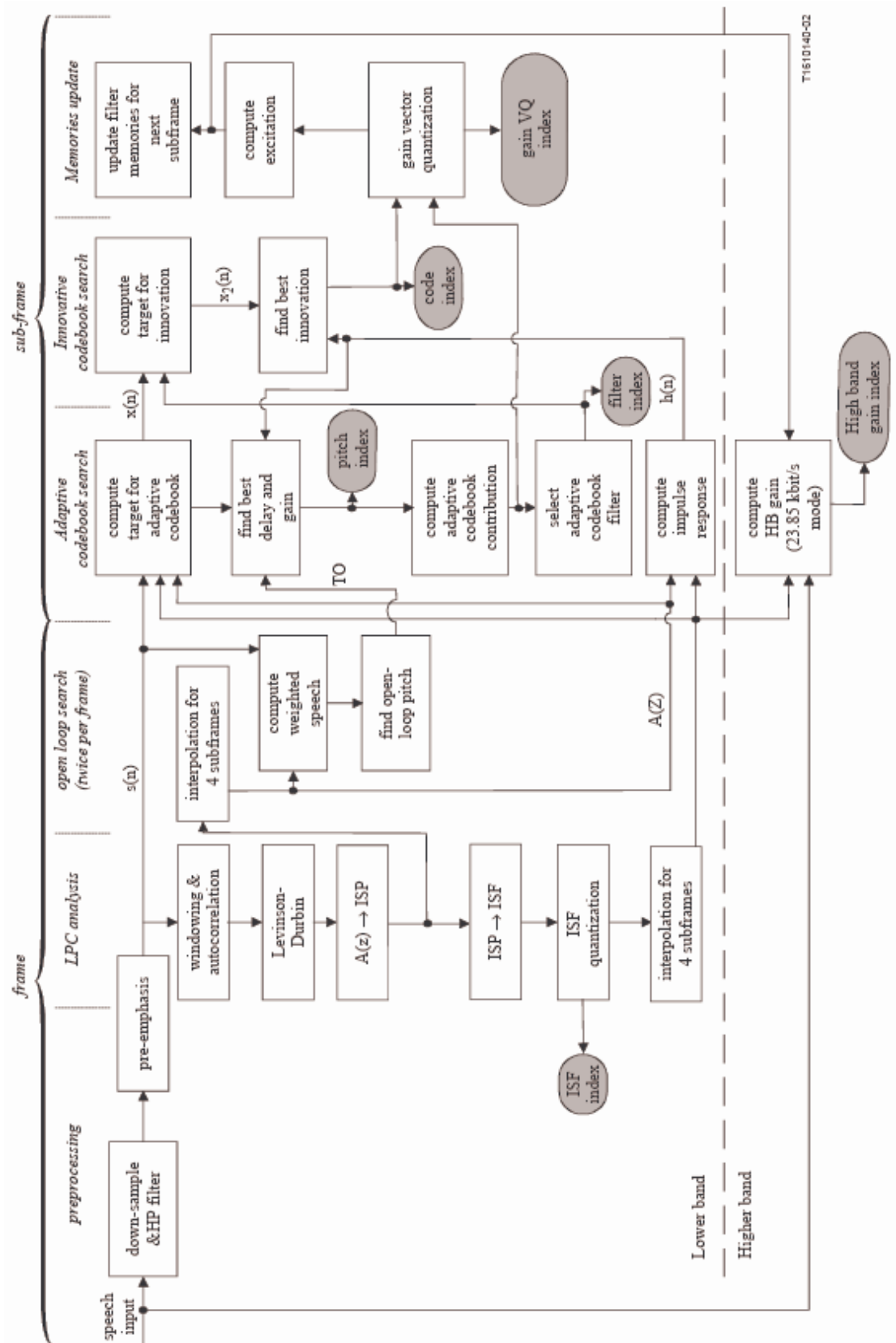


Figure 1.4: Block Diagram of the G.722.2 Encoder

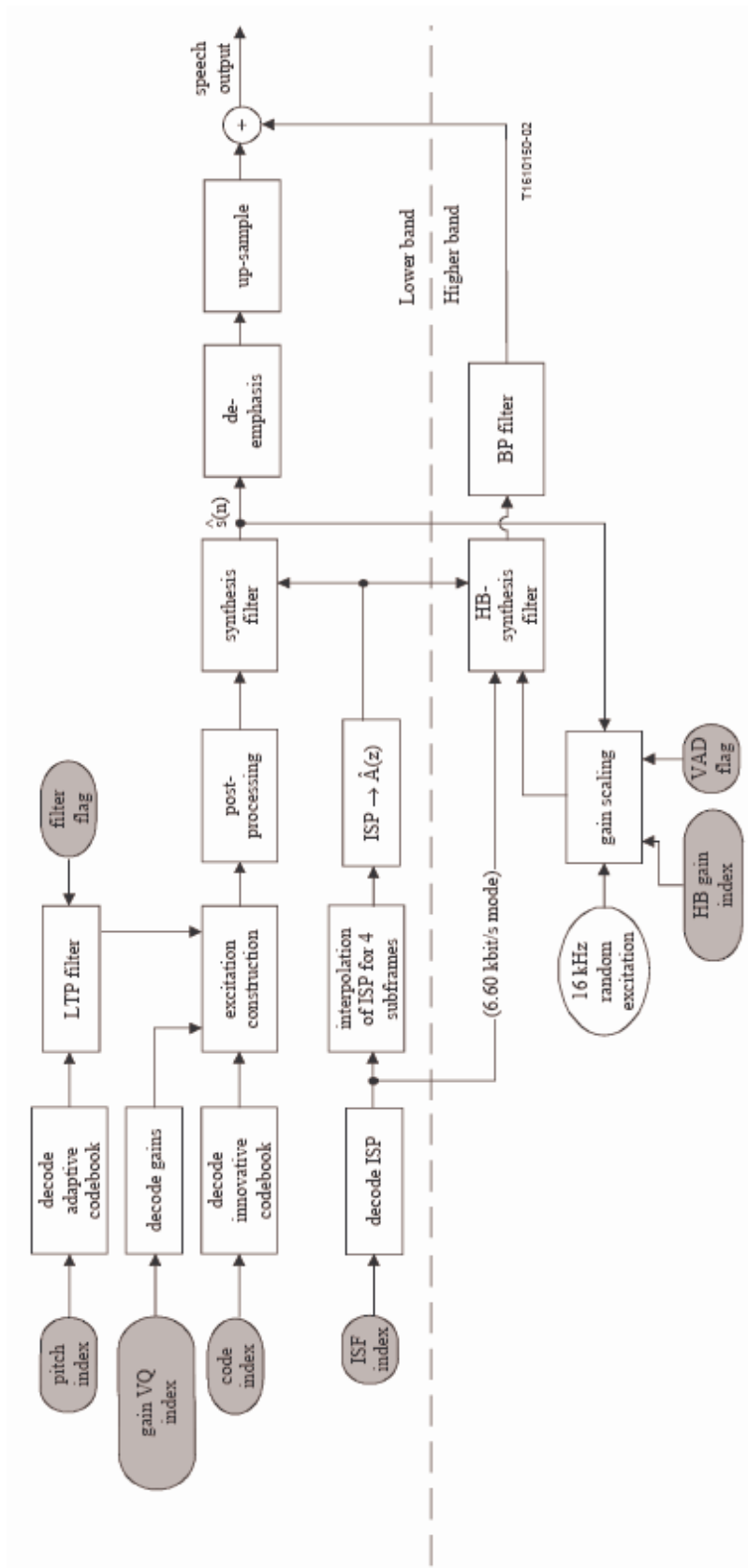


Figure 1.5: Block Diagram of the G.722.2 Decoder

1.3 Scalable Coders

1.3.1 Introduction to scalable coding

Scalable coding is a technique in speech coding where the output bitstream (the encoded speech) is structured in layers. Speech can be synthesized by using one or more of the output layers. The base layer (first layer) consists of the most important information regarding the input speech and is crucial for synthesizing intelligible speech. The other layers (sometimes called the enhancement layers), improve the perceptual quality of speech. Hence the coder can operate at different bitrates depending on the number of enhancement layers selected. This is useful in transmission over congested networks as the enhancement layers can be dropped during network congestion while still maintaining intelligible quality of speech.

1.3.2 Types of Scalable Coders

Scalable coders are divided broadly into two types based on the parts of speech they enhance.

Bitrate Scalable Coders

In bitrate scalable coders, the enhancement layers improve speech in the same band as the base layer. These coders depend heavily on the functioning of the base layer. They improve the quality of speech by encoding the difference between the input speech and the speech synthesized by the base layer. [8] and [9] are examples of bitrate scalable coders.

Bandwidth Scalable Coders

Bandwidth scalable coders improve the quality of speech by encoding the frequency bands left out by the base layer. These coders can operate independently of the base layer, but can be more efficient if they use the base layer information. [10][11][12][13] are examples of bandwidth scalable coders.

1.4 Scalable Wideband Coding

A bandwidth scalable coder with the G.729 base coder has been proposed in [14]. Here different structures have been discussed for the enhancement coder. In all the structures separate ACBs and FCBs are used in the enhancement coder.

In the present work, we propose a scalable coding scheme where the enhancement coder uses the ACBs and FCBs of the baseband coder. This results in a significant decrease in the bitrate of the enhancement layers. Also an additional level of scalability is introduced. The developed coder is scalable with the first enhancement layer (3.4 kHz – 5.6 kHz) at 2.7 kbps and the second layer (5.6 kHz – 7.2 kHz) at 0.9 kbps.

1.5 Summary

In this document, we report the experiments done on wideband speech samples and the wideband scalable coder developed using the results of the experiments. Chapter 2 gives the motivation for the development of scalable coders with the example of the PacketCable network. Chapter 3 discusses the present work. Conclusion and suggestions for future work , and References are provided at the end.

CHAPTER 2

MOTIVATION

2.1 Introduction to cable architecture

PacketCable is a CableLabs specification to support the convergence of voice, video, data and mobility technologies. It defines an architecture and a set of open interfaces that leverage emerging communications technologies to support the rapid introduction of new IP-based communications into the cable network.

PacketCable uses an SIP-based signaling [15] model for setting up and tearing down sessions between User equipments (UE). It defines a function called Serving Call Session Control Function (S-CSCF) which acts as the SIP-registrar function and routes calls to the remote S-CSCFs.

QoS management in PacketCable is done through a PacketCable Application Manager (PAM). The SIP requests are proxied to the PAM which interprets the SDP message body to determine the resources needed.

Various network elements generate billing events which are routed to a Charging Data Function (CDF) for keeping records and billing.

Figure 2.1 shows the basic architecture of the PacketCable network. Refer to [16] for a complete description of the PacketCable network elements and the interfaces between them.

One of the key issues in this network is the QoS resource management. PacketCable provides access network QoS resource management through the PacketCable Application Manager (PAM) but does not specify backbone network QoS management protocols. Using scalable codecs helps better backbone network resource management in the following way. The enhancement layers are marked for lower priority and can be dropped to relieve network congestion. These packets are not retransmitted (but instead interpolated) when packet loss is discovered. So, it does not result in subsequent congestions. Also the quality loss will be minimized since the scalable codecs are designed for packet loss situations

A second advantage in using scalable codecs is the flexibility it provides for the customers to choose from a wide variety of quality vs charging options. For eg., one might choose to have only the base layer for normal informal talk, but might switch to a high quality (one or more enhancement layers) session for a business call without a very significant increase in calling rate (since the enhancement layers are charged less because of low priority).

In this chapter, we investigate the possibility of integrating scalable codecs into the PacketCable architecture with minimum signaling overhead, minimum change to network elements and interoperability with non scalable codec-enabled UEs.

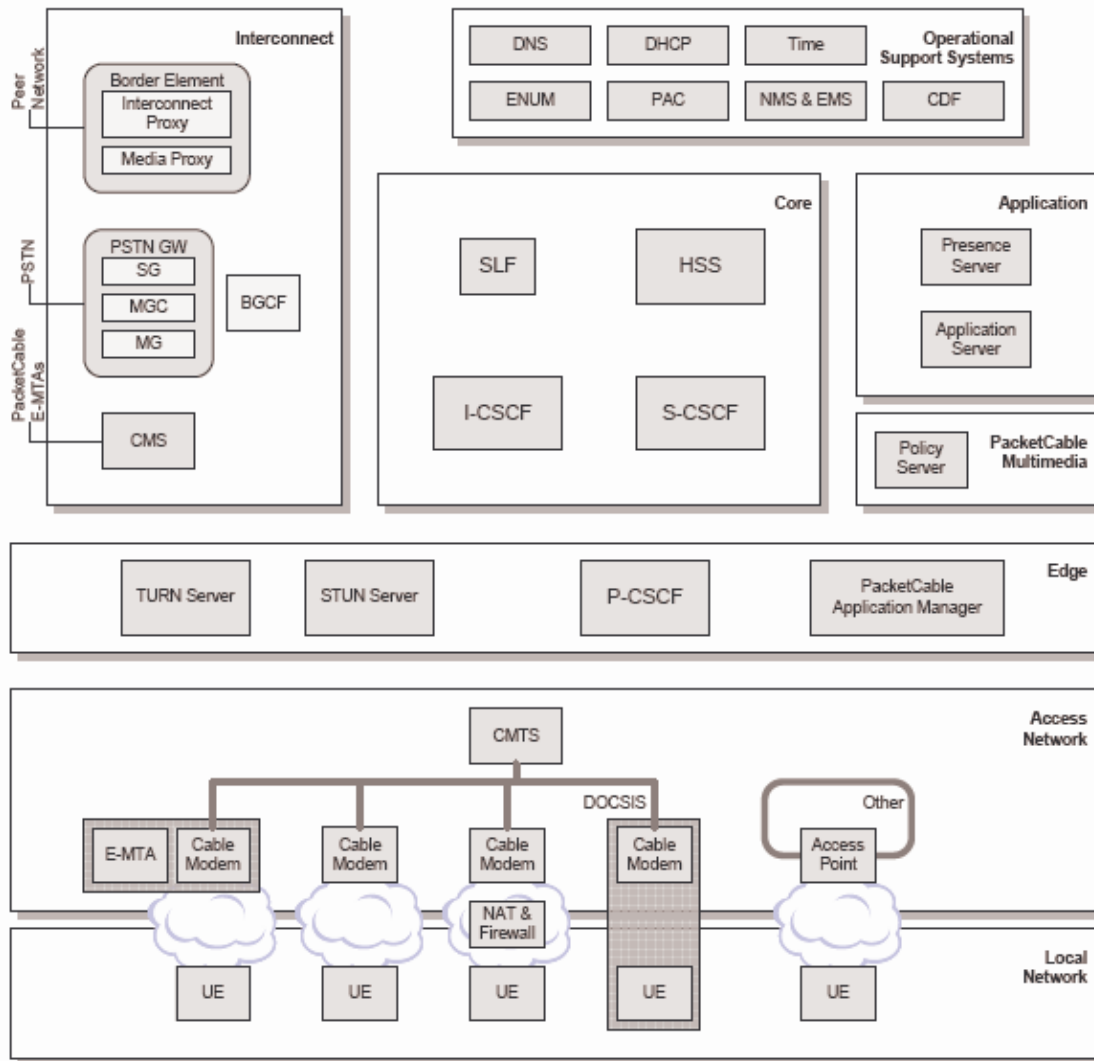


Figure 2.1: Reference architecture

2.2 Introduction to Session Initiation Protocol (SIP)

There are many applications of the Internet that require the creation and management of a session, where a session is considered an exchange of data between an association of participants. The implementation of these applications is complicated by the practices of participants: users may move between endpoints, they may be addressable by multiple names, and they may communicate in several different media – sometimes simultaneously. Numerous protocols have been authored that carry various forms of real-time multimedia session data such as voice, video, or text messages. The Session Initiation Protocol (SIP) works in concert with these protocols by enabling Internet endpoints (called user agents) to discover one another and to agree on a characterization

of a session they would like to share. For locating prospective session participants, and for other functions, SIP enables the creation of an infrastructure of network hosts (called proxy servers) to which user agents can send registrations, invitations to sessions, and other requests. SIP is an agile, general-purpose tool for creating, modifying, and terminating sessions that works independently of underlying transport protocols and without dependency on the type of session that is being established.

SIP supports five facets of establishing and terminating multimedia communications:

- User location: determination of the end system to be used for communication
- User availability: determination of the willingness of the called party to engage in communications
- User capabilities: determination of the media and media parameters to be used
- Session setup: "ringing", establishment of session parameters at both called and calling party
- Session management: including transfer and termination of sessions, modifying session parameters, and invoking services

Refer to Section 2.4 for an example of a basic call setup in SIP.

2.3 Introduction to Session Description Protocol (SDP)

SDP [17] is intended for describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation. The purpose of SDP is to convey information about media streams in multimedia sessions to allow the recipients of a session description to participate in the session. SDP is primarily intended for use in an internetwork, although it is sufficiently general that it can describe conferences in other network environments.

Session Description primarily includes:

- Session name and purpose
- Time(s) the session is active
- The media comprising the session
- Information to receive those media (addresses, ports, formats and so on)

The list of SDP descriptors is given below. The ones marked with a * are optional.

Session description

v= (protocol version)
o= (owner/creator and session identifier).
s= (session name)
i=* (session information)
u=* (URI of description)
e=* (email address)
p=* (phone number)
c=* (connection information - not required if included in all media)
b=* (bandwidth information)
One or more time descriptions (see below)
z=* (time zone adjustments)
k=* (encryption key)
a=* (zero or more session attribute lines)
Zero or more media descriptions (see below)

Time description

t= (time the session is active)
r=* (zero or more repeat times)

Media description

m= (media name and transport address)
i=* (media title)
c=* (connection information - optional if included at session-level)
b=* (bandwidth information)
k=* (encryption key)
a=* (zero or more media attribute lines)

An example SDP description is:

```
v=0
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 31
m=application 32416 udp wb
a=orient:portrait
```

The most important descriptor for the present work is the media descriptor. Let us try to understand the first media descriptor in the above example:

- The first sub-field specifies the type of media. Presently defined types are “audio”, “video”, “application”, “data” and “control”
- The second sub-field specifies the transport port to which the media will be sent
- The third sub-field specifies the transport protocol that will be used. Here it is the IETF's Realtime Transport Protocol [18] using the Audio/Video profile [19] carried over UDP
- The fourth sub-field is the media format. For RTP this will be the payload type. A number of standard payload types are defined in [19]. Experimental payload types can be specified using the rtpmap attribute (see below) to map them to a payload type above 96. Experimental payload types must begin with X-

The rtpmap attribute is used as follows:

```
a=rtpmap:<payload type> <encoding name>/<clock rate>[/<encoding parameters>]
```

For example:

```
m=video 49232 RTP/AVP 99  
a=rtpmap:99 X-GSMLPC/8000
```

Here the payload type GSMLPC is mapped to payload type number 99 using the rtpmap attribute.

2.4 A basic Call Setup using SIP

Figure 2 shows a typical example of a SIP message exchange between two users, Alice and Bob. (Each message is labeled with the letter "F" and a number for reference by the text.) In this example, Alice uses a SIP application on her PC (referred to as a softphone) to call Bob on his SIP phone over the Internet. Also shown are two SIP proxy servers that act on behalf of Alice and Bob to facilitate the session establishment. This typical arrangement is often referred to as the "SIP trapezoid" as shown by the geometric shape of the dotted lines in Figure 2.2.

Alice "calls" Bob using his SIP identity, a type of Uniform Resource Identifier (URI) called a SIP URI. It has a similar form to an email address, typically containing a username and a host name. In this case, it is sip:bob@biloxi.com, where biloxi.com is the domain of Bob's SIP service provider. Alice has a SIP URI of sip:alice@atlanta.com. Alice might have typed in Bob's URI or perhaps clicked on a hyperlink or an entry in an address book. SIP also provides a secure URI, called a SIPS URI. An example would be sips:bob@biloxi.com. A call made to a SIPS URI guarantees that secure, encrypted

transport (namely TLS) is used to carry all SIP messages from the caller to the domain of the callee. From there, the request is sent securely to the callee, but with security mechanisms that depend on the policy of the domain of the callee.

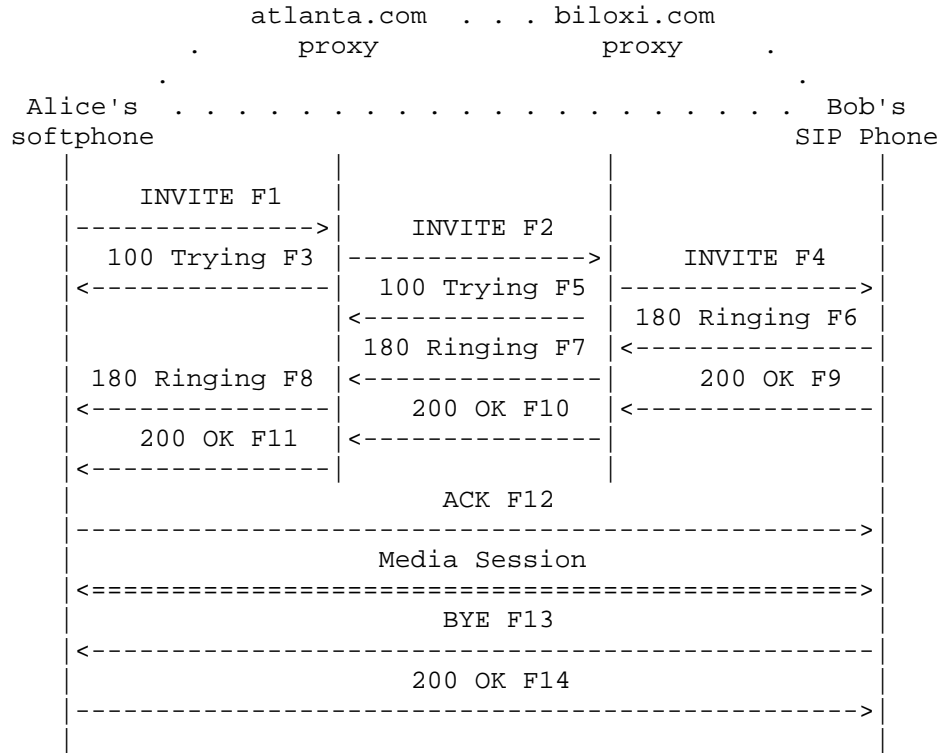


Figure 2.2: SIP session setup example with SIP trapezoid

The different messages are explained below:

F1 : This is the INVITE request from Alice to Bob. This message contains the To and From addresses, a Call-Id, and the type of media that Alice wants to use (in the SDP), in addition to other information

F3,F5 : These messages mean that the server is trying to service the request

F2,F4 : The INVITE request is forwarded to Bob

F6,F7,F8 : Bob's phone is located and his phone "rings". This is conveyed to the caller. The phone at Alice can feed a "ringing" tone to alert Alice

F9,F10,F11 : Bob accepts the phone call and the OK message is relayed to Alice.

F12 : An acknowledgement is sent from Alice and the session starts

F13 : Bob decides to terminate the call and his phone sends the BYE message

F14 : Alice sends the OK response and the session is broken down

2.5 Integration of Scalable Codecs into PacketCable

A method for integrating scalable codecs into PacketCable networks is presented below. One of the factors considered for suggesting the method is interoperability with other non scalable codec-enabled UEs. The method suggested here works with any UE compliant with the PacketCable specifications. Another factor considered was minimization of the changes needed to network elements in an already deployed network. The method presented needs only software changes to the PAM (for marking the packets) and the Charging Function (for billing). The method is described below.

The originating UE that wishes to make a call using scalable codecs includes in its SDP [17] of the initial INVITE request an offer[20] with a format type that needs to be standardized but is designated here as X-SC1. We define here two format-specific parameters for the fmp attribute. They are the min-n and max-n for the maximum and minimum no. of enhancement layers that it wishes to use (min-n is usually zero but is included for completeness). It also includes other codecs that it supports. The relevant portion of a sample SDP is shown for the case of a single audio stream:

```
m=audio 50000 RTP/AVP 100 20 3
a=rtpmap: 100 X-SC1/8000
a=fmp: 100 min-n=0;max-n=3
...
...
```

If the terminating UE is not scalable codec-enabled or it does not wish to use the offered scalable codec(s), it returns an SDP with the preferred codec which the originating UE should accept and the session proceeds as usual. Or it might terminate the session with a 606 response if it supports none of the codecs specified and might include the list of codecs it supports in the response.

If the terminating UE wants to use the scalable codec it returns a 183 (session progress) response. The answer will contain the specified codec in the SDP. Both min-n and max-n are set to the no. of enhancement layers it wishes to use. An example response is shown below:

```
m=audio 50000 RTP/AVP 100
a=rtpmap:100 X-SC1/8000
a=fmp: 100 min-n=2;max-n=2
```

The originating UE processes this response and sends a PRACK[21] request with a modified SDP. This contains the additional media streams necessary for the enhancement layers. For this example, the SDP is shown below.

```
m=audio 50000 RTP/AVP 100
a=rtpmap:100 X-SC1/8000
```

```
m=audio 50002 RTP/AVP 101
a=rtpmap:101 X-SC1-EXT1/8000
```

```
m=audio 50004 RTP/AVP 102
a=rtpmap:102 X-SC1-EXT2/8000
```

This will be accepted the terminating UE which sends a 200 (OK) response to start the session. An example call flow is shown in Figure 2.3.

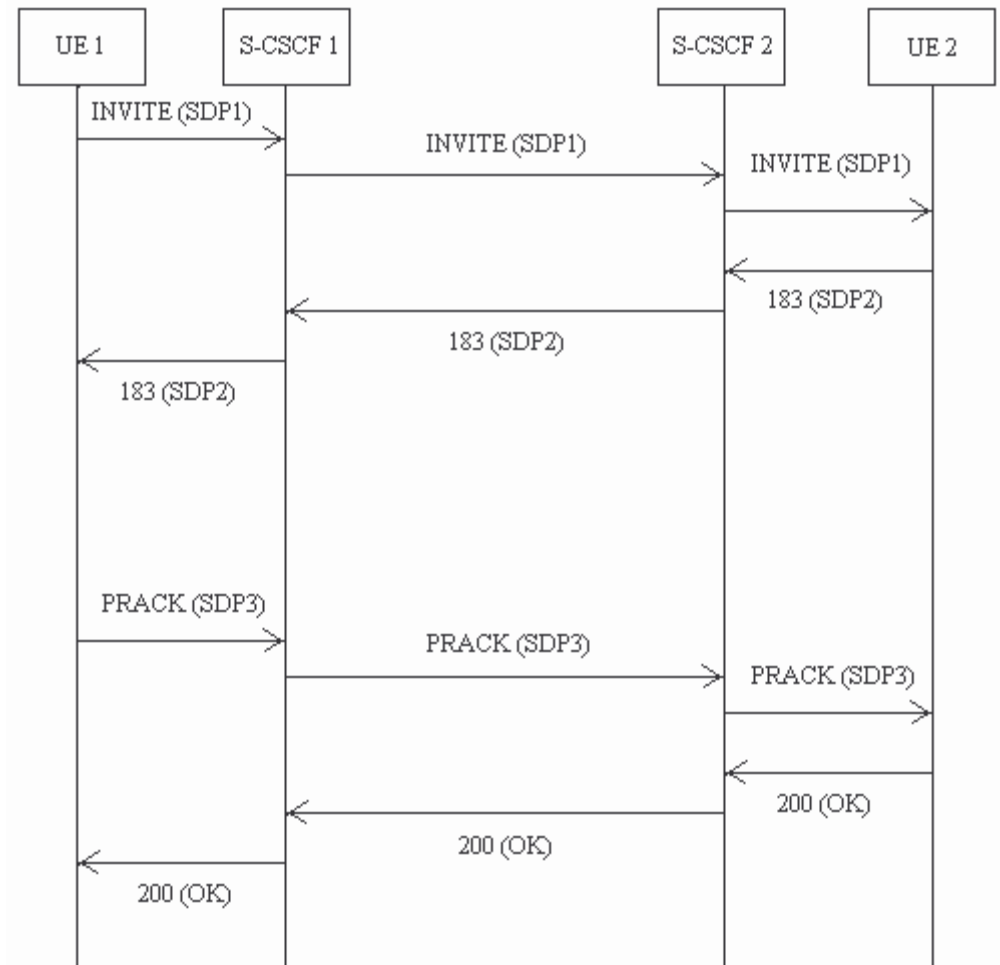


Figure 2.3: An example call flow of session initiation for scalable codecs

The above method needs a minor software change to the PAM which has to detect the enhancement media streams and assign them a lower priority. The actual assignment depends on the backbone protocol used. Also the billing function needs to be changed slightly so that the low priority enhancement layers are charged lesser than the base layer.

2.6 Conclusion

We have investigated the possibility of using scalable codecs in a PacketCable architecture. We have outlined the method to be followed in such a situation for minimum signaling overhead, minimum change to network elements and interoperability with non scalable codec-enabled UEs. Though the method has not been standardized, we hope that a standard will soon be worked out.

CHAPTER 3

SCALABLE WIDEBAND CODING

3.1 Introduction

Wideband coding of speech refers to coding of speech in the 50 Hz – 7.2 kHz range as against narrowband coders which code speech only in the 300 Hz – 3.4 kHz range. The narrowband coders are used in most traditional systems including the telephone networks. Although narrowband speech is mostly intelligible, it becomes less intelligible in unvoiced regions like fricatives. For example, a ‘P’ uttered over the telephone is usually heard as a ‘B’, and a ‘D’ is heard as a ‘T’. Also narrowband speech is muffled and unnatural and makes speaker identification difficult.

Wideband coders encode speech in a larger range which adds naturalness to speech and makes it more intelligible. Also a feeling of comfort is added as speaker identification is easier and the speech is closer to face-to-face communication. But the main impediment to the deployment of wideband coders is the bitrate necessary to transmit wideband speech. While good narrowband coders like the G.729 work at 8 kbps, there is a need to develop wideband coders to work at these rates at the same time maintaining the quality of decoded speech.

A number of wideband coders have been proposed to work at low rates like the “Adaptive Multirate – Wideband” (AMR-WB) Coder which works from 6.6 kbps – 23.85 kbps. Since most of the future communication is expected to be packet communication, there is a need to develop scalable coders. Scalable coders have the advantage that they help ease network congestion by allowing higher layer packets to be dropped. In the case of non-scalable coders, packets dropped due to network congestion are retransmitted, leading to further congestion of the network. On the other hand, scalable coders do not transmit dropped higher layer packets, thus helping in easing the congestion.

Attempts have been made to develop wideband scalable coders to work at around 16 kbps like in [13] and [14]. Here we develop wideband coders scalable with any CELP based coder like G.729 or G.729E with 2 enhancement layers at 2.7 kbps (3.4 kHz – 5.6 kHz) and .9 kbps (5.6 kHz – 7.2 kHz). But before that, we report the experiments conducted on wideband speech to gain an understanding that will enable us to develop the coders in Section 3.2. Section 3.3 describes the coders developed. The results are presented in Section 3.4

3.2 Experiments

In the CELP paradigm, speech can be represented by the synthesis filter and the excitation. Going by this idea, experiments were conducted, both on the synthesis filters and the excitation signals. These experiments are presented below. We have not used the results of some of the experiments. Nonetheless, these experiments and the results are presented for the sake of completeness and future study.

3.2.1 Experiments in the LPC Domain

Since there is feeling that the highband speech (4 kHz – 8 kHz) is not totally uncorrelated from the narrowband speech, attempts are made to capture the correlation between them and use them for developing low bitrate wideband coders. One such attempt is to derive the synthesis filter for the highband from the lowband.

In this experiment, speech is lowpass filtered and LPC analysis of order 10 is performed. Similarly, the highpass component is LPC analyzed (order 10) and the synthesis filters obtained are compared. For ease of comparison, the highband synthesis filter is shifted from (4 KHz – 8 kHz) band to (0 kHz – 4 kHz) band. The filters obtained show considerable correlation with respect to peaks. One such example is given below in Figures 3.1 and 3.2.

As can be seen from the figures, there exists a good correlation between the lowband and the highband synthesis filters. This suggests that the higher band filter can be derived from the low band filter with some additional information.

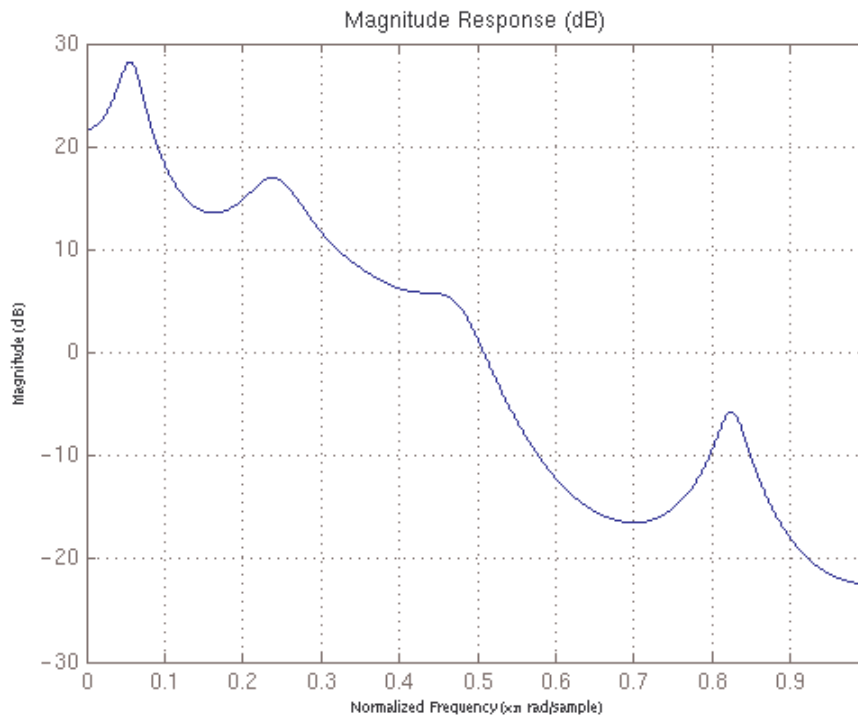


Figure 3.1: Lowband synthesis filter

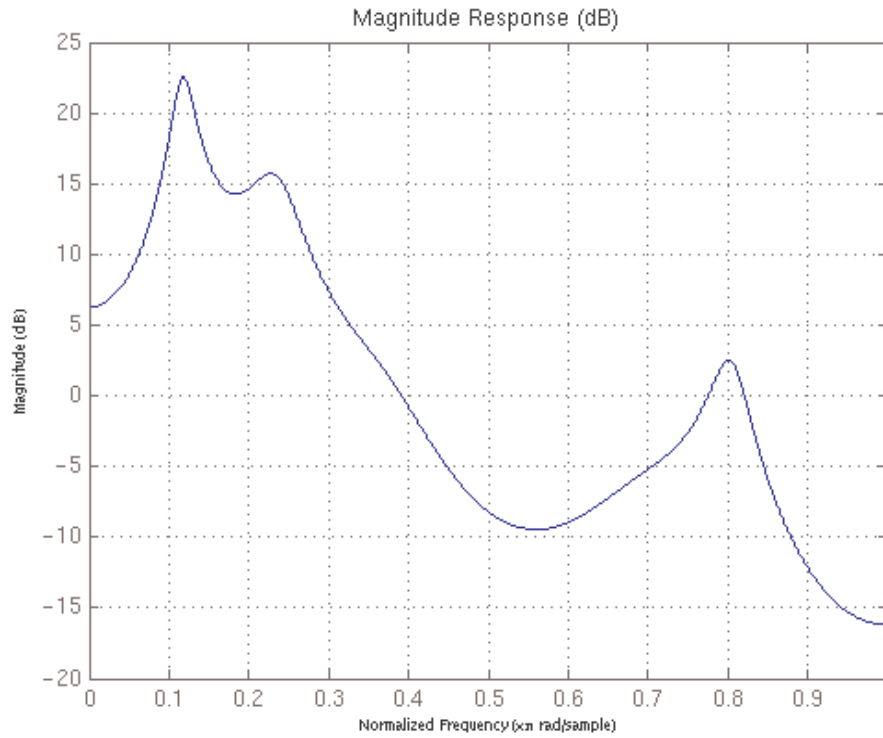


Figure 3.2: Highband synthesis filter

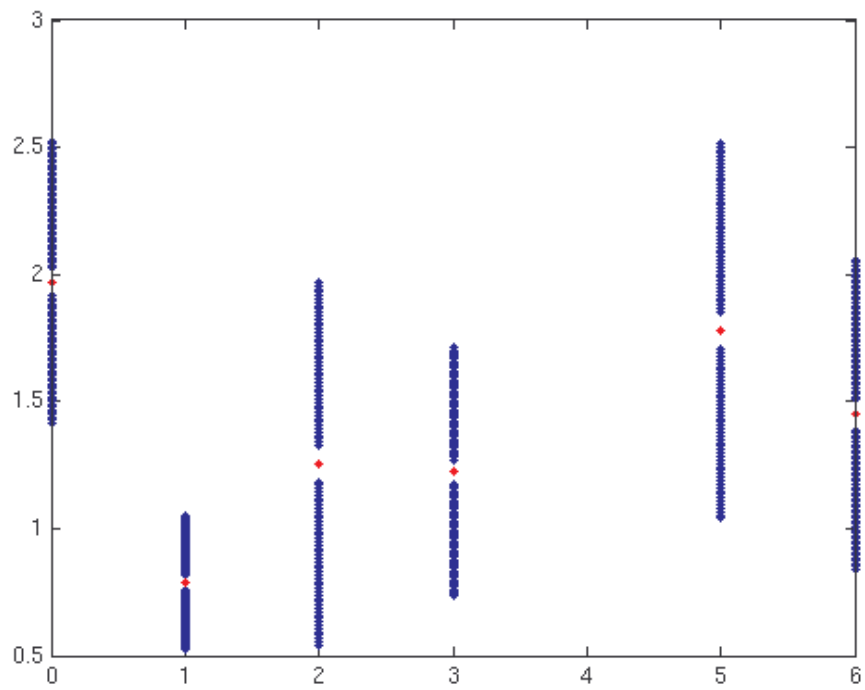


Figure 3.3: Itakura distances for the different speech classes.

Next, the similarity of the filters with respect to the speech type was investigated. A classifier of speech was used which would classify speech into one of the six classes namely silence(0), noise-like(1), unvoiced(2), onset(3), non-stationary voiced(5) and stationary voiced(6). LPC analysis was done and itakura distances were found between the low band and the high band filters. These distances were averaged within a particular class of speech to know the variation in the similarity of the filters with respect to the class of speech. The result is shown below in Figure 3.3.

The results presented in Figure 3.3 are not very encouraging. But as can be seen from the synthesis filters, although they show a correlation, the peaks and valleys are shifted to the left or right with respect to one another. This partly explains the bad results of Figure 3.3 To capture this correlation between the filters, the LPCs are converted to LSFs and studied.

LSF Domain

As noted in the previous section, the LPCs of the lowband and highband speech are converted into LSFs to look for a correlation between them. The obtained LSFs of the highband are plotted against the LSFs obtained for the lowband of the same frame. The plot is presented in Figure 3.4.

As can be seen from the graph, the LSFs of the highband show a “linear-like” relationship with respect to the LSFs of the lowband. Also, the higher band LSFs of the present frame show a linear relation with the LSFs of the previous frame. Hence an attempt is made to predict linearly the high band LSFs of the current frame from the low band LSFs of the current frame and the high band LSFs of the previous frame

The prediction equation is:

$$\hat{L}_{h,i}(k) = aL_{l,i} + bL_{h,i-1} + c$$

where ‘i’ refers to the current frame and ‘i-1’ refers to the previous frame. ‘L’ are the LSFs and ‘k’ is the index referring to one of the 10 LSFs. The subscripts ‘h’, and ‘l’, stand for the highband and the lowband respectively.

The prediction error is:

$$\sum_{k=1}^{10} (\hat{L}_{h,i}(k) - L_{h,i}(k))^2$$

Minimizing this error gives a linear equation for a, b, and c. These are solved and transmitted to the decoder. Also a random error uniformly distributed in (-0.1 to 0.1) was added to a, b, and c to simulate quantization effects. The high band LSFs are decoded at the decoder based on the received a, b, and c. This method can give rise to LSFs below zero or above π . In such cases the decoded LSFs are shifted and scaled to bring them into

the range $0 - \pi$. These are then converted to LPC coefficients and the synthesis filter is reconstructed.

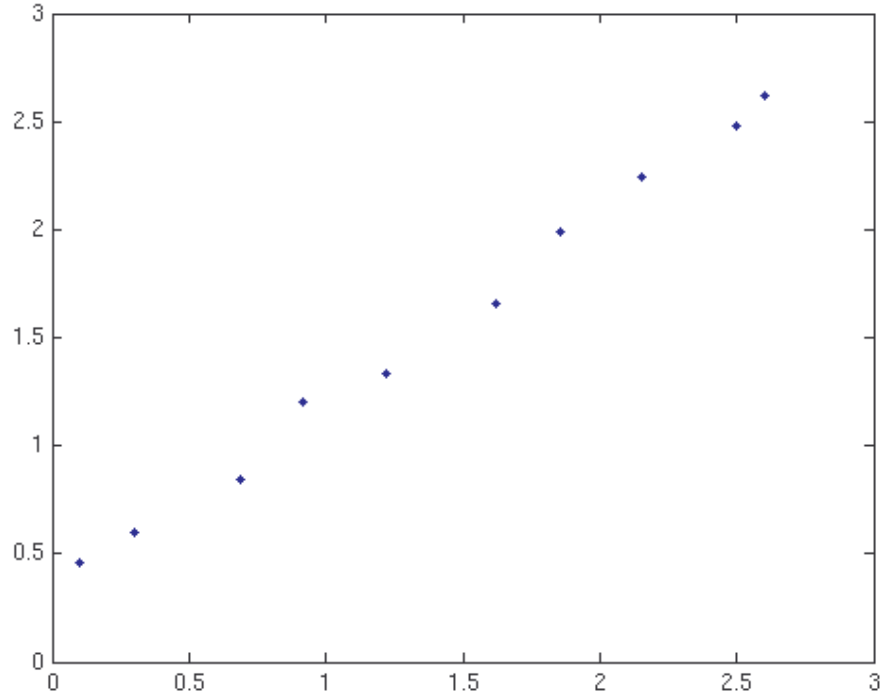


Figure 3.4: Lower band LSFs v/s higher band LSFs of the same frame.

To test the above method, wideband speech was filtered into lower and higher sub bands and LPC analysis was done on both the signals. The resulting LPCs are converted into LSFs and the prediction coefficients (a,b,c) are found. The decoder side LSFs are obtained by linear prediction as described above and converted into LPC coefficients. The residues of both the sub bands are filtered through the synthesis filters. Both the signals are upsampled and added and then compared with the original speech.

Though the PESQ scores were good, the resulting speech obtained showed problems in regions with considerable high frequency components. Some of these regions had audible artifacts in the decoded speech. The reason for this is probably the high gain seen in the decoded synthesis filters for the higher sub band, thus amplifying the high frequency components leading to artifacts. This problem of the filters becoming unstable is presumably because of arbitrary shifting of out-of-range LSFs into the $0 - \pi$ range.

Clustering of predictive coefficients

The above method might be useful only if the predictor coefficients can be quantized effectively. Hence the predictive coefficients a, b, and c were plotted in pairs to

investigate their clustering behavior. The plots are shown in Figures 3.5, 3.6, and 3.7. As can be seen from the plots, these coefficients seem to yield effectively to Vector Quantization.

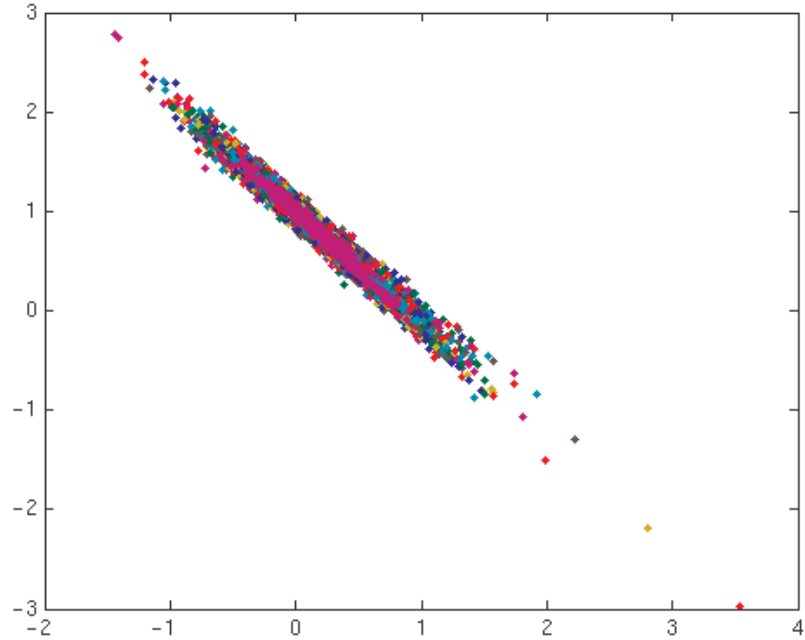


Figure 3.5: Plot of 'a' v/s 'b'

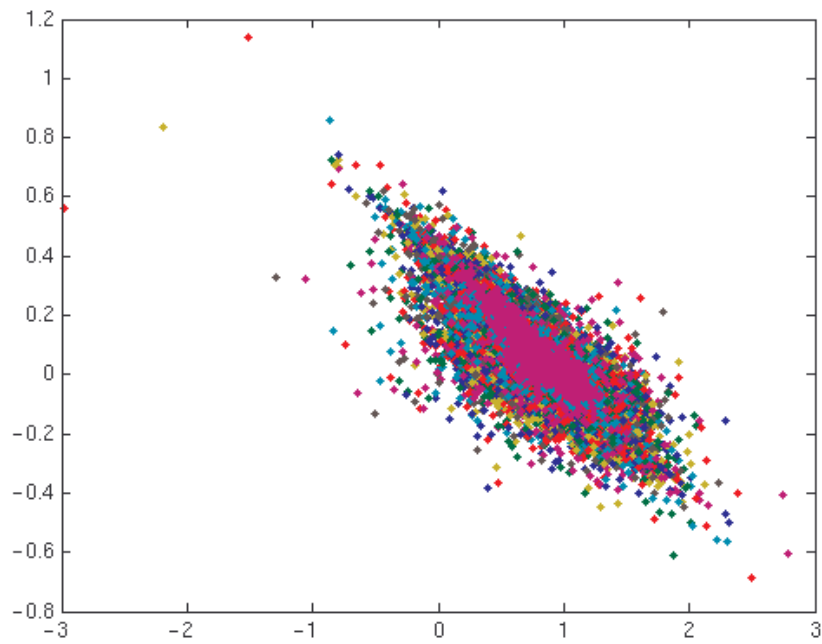


Figure 3.6: Plot of 'b' v/s 'c'

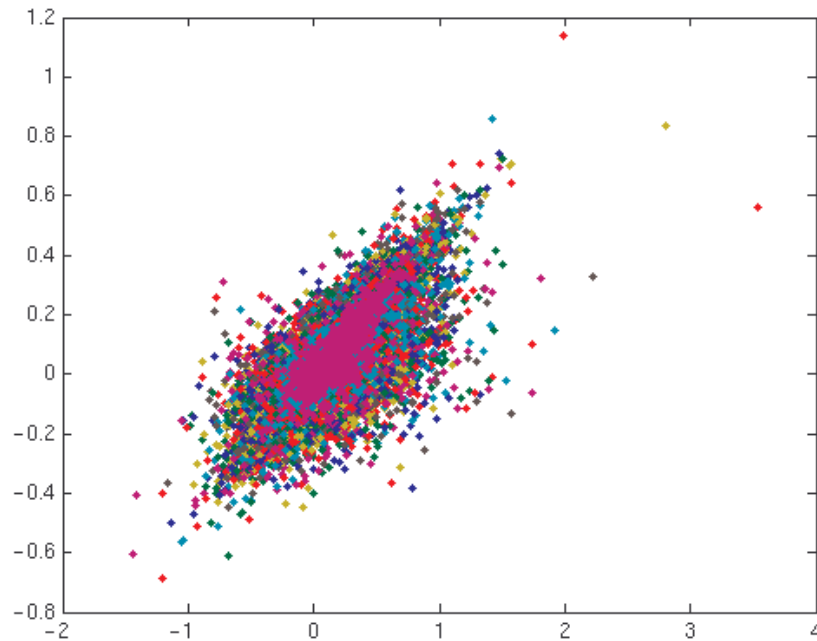


Figure 3.7: Plot of 'a' v/s 'c'

Conclusion

In this section, we tried a method of obtaining synthesis filter of the highband from the lowband. But the decoded speech showed audible artifacts in regions with large high frequency components. Hence, this method as such is not usable and we have not used it in the coders developed. Further study might be needed to use this method effectively.

3.2.2 Experiments in the residual domain

The residue obtained after the speech is filtered through the inverse synthesis filter is usually modeled using the Adaptive Codebook and the Fixed Codebook. The Adaptive Codebook captures the pitch periodicity of the residue and the Fixed Codebook captures the excitation obtained after passing the residue through the pitch filter. The ACB and FCB indices and gains together occupy a huge chunk of the transmitted bitstream. Hence attempts were made to reduce the amount of transmitted information. Here we present some of the experiments done in this regard and the problems encountered.

Experiment 1

The first experiment done was to get rid of the ACB information transmitted. Here the lowband is encoded using the standard G.729 like coder. That is LPC analysis of order 10 is done on the lowpass speech and it is passed through the inverse synthesis filter

obtained. This residue is analyzed to find the pitch and the pitch gain to form the Long Term Filter (pitch filter). The residue is long term filtered to obtain the excitation which is modeled by the FCB.

For the highband, LPC analysis of order 10 is done and the highband residue is obtained. Then it is passed through the pitch filter of the lowband. The obtained excitation is modeled through the FCBs as usual.

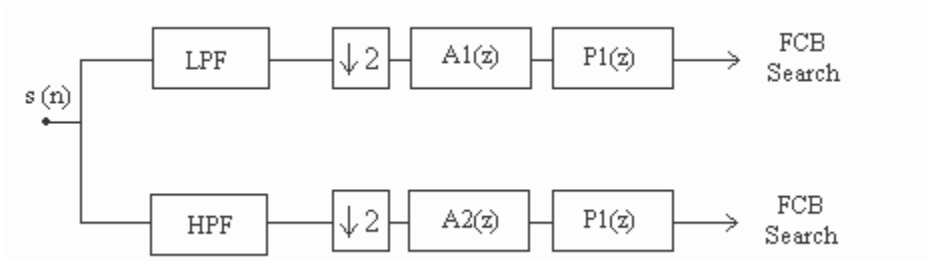


Figure 3.8: Encoder of Experiment 1

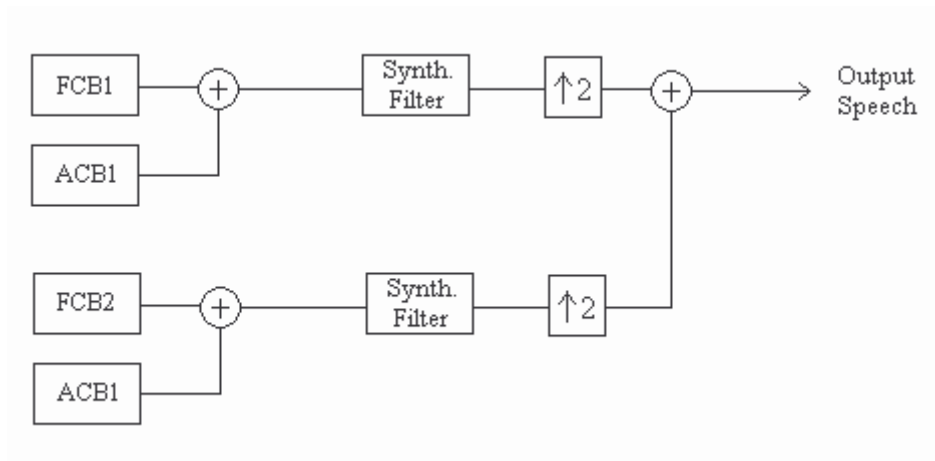


Figure 3.9: Decoder of Experiment 1

In the decoder, both the excitations are passed through the same pitch filter and different synthesis filters and then added to form the decoded speech. By this method the ACB index and gain need not be transmitted. But the problem with this approach is that the excitation obtained for the highband is not random and might contain some pitch periodicity. Hence the FCBs may not be able to model them adequately.

Experiment 2

As described in the previous experiment, the highband excitation is not modeled adequately by the FCBs when it contains some pitch periodicity. Hence a second approach was tried. Here, the highband speech was encoded in a manner similar to the

lowband speech. But the pitch information was discarded. At the decoder, the highband speech was decoded using the lowband pitch filter.

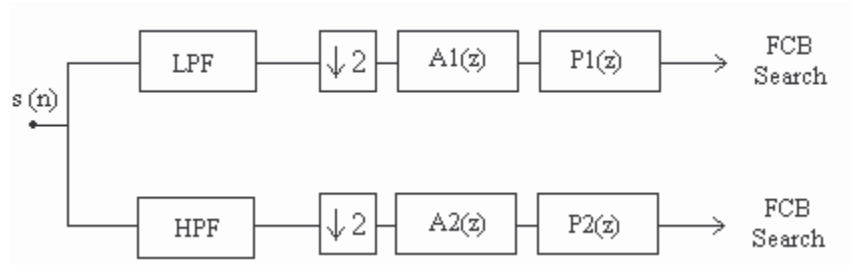


Figure 3.10: Encoder of Experiment 2

In this approach, the highband excitation is random and the FCBs can model them adequately. But the residue obtained after the pitch filter in the decoder is not identical to the residue in the encoder even in the absence of other losses. But the obtained speech was of reasonable quality, thus showing that the highband speech might not be very sensitive to pitch changes.

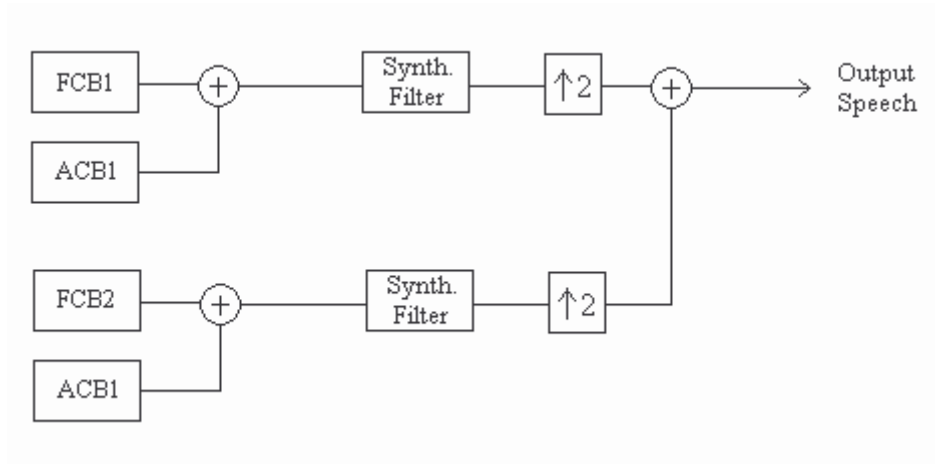


Figure 3.11: Decoder of Experiment 2

Experiment 3

It was learnt from the previous experiment that the pitch of the highband is not very critical. It is also known that the excitation after the pitch filtering is mostly random (white). Hence we try to derive the highband residue from the lowband residue.

The lowband speech is encoded in the manner given in experiment 1. The highband speech was LPC analyzed and passed through the inverse synthesis filter to obtain the residue. The energy of this residue is measured and transmitted as a fraction of the energy of the lowband residue. At the decoder, the lowband reconstructed residue is multiplied

with this fraction and passed through the highband synthesis filter to obtain the highband speech.

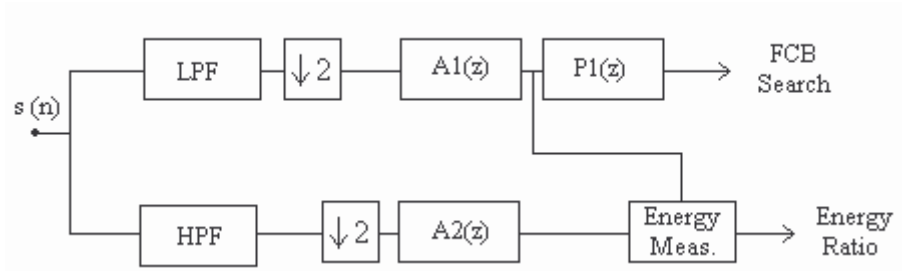


Figure 3.12: Encoder of Experiment 3

This method worked well with all the speech samples tried and was adopted in the wideband codec developed with minor modifications. We describe the final coder developed in the next section.

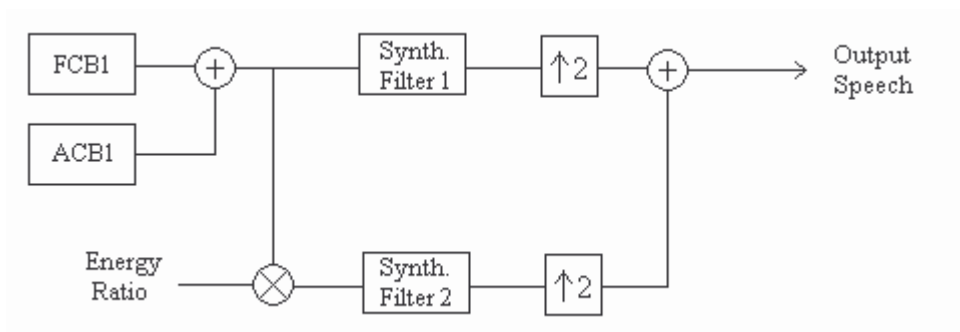


Figure 3.13: Decoder of Experiment 3

3.3 Scalable Wideband Coders

The scalable wideband coder developed here is scalable with any CELP based coder like the G.729 coder. We only assume that the baseband coder uses the CELP paradigm and uses the synthesis filter – residue based approach. The different operations followed in the codec are described below.

3.3.1 Encoder

Obtaining highband Speech

The input wideband speech is lowpass filtered using a filter with a cut-off at 4 kHz. The resulting speech is passed to the standard baseband coder modified to output the energy

of the lowband residue. The lowband speech is synthesized locally and subtracted from the input wideband speech. The resulting signal is filtered through a highpass filter with a cut-off at 3.4 kHz to obtain the highband speech. This is shown schematically in Figure 3.14.

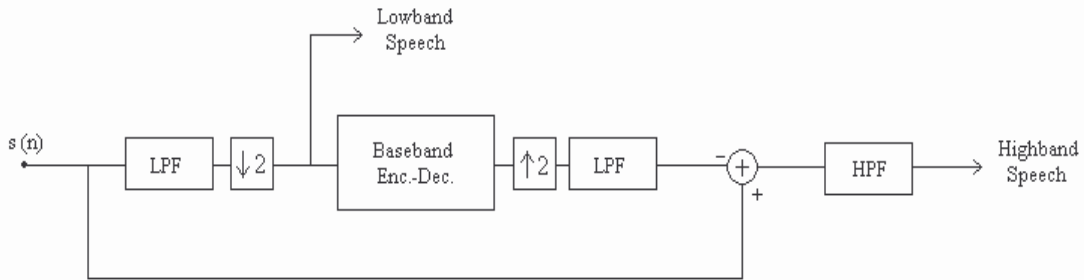


Figure 3.14: Obtaining highband speech

LPC Analysis

The highband speech obtained in the previous section is windowed with a hamming-cosine window without lookahead. 10^{th} order LPC analysis is done on this signal. These LPC coefficients are converted into LSFs and quantized with 18 bits per frame. The highband speech is passed through the inverse synthesis filter obtained. The resulting residue is filtered into two bands – one from 3.4 kHz – 5.6 kHz and another band from 5.6 kHz – 7.2 kHz. The energy of the two bands are measured separately on a sub-frame basis and the ratio of this energy to the energy of the lowband residue is found. Hence there are two gains transmitted every sub-frame. The gains for the two sub-frame are vector quantized with 9 bits per frame each (512 codewords). Hence the additional bitrate needed is 2.7 kbps for the first sub-band from 3.4 kHz – 5.6 kHz and .9 kbps for the second sub-band from 5.6 kHz – 7.2 kHz.

Bit Allocation

The bit allocation table is shown below in Table 3.1:

	Sub-band 1	Sub-band 2
LSF	18	
Gain	9	9
Total	27	9

Table 3.1: Bit allocation for the enhancement layers

3.3.2 Decoder

At the decoder, the lowband speech is decoded by the baseband decoder. The baseband decoder also passes the lowband residue to the highband decoder. At the highband decoder, the two sub-bands are decoded separately as follows.

Sub-band 1

The lowband residue is upsampled and passed through a band-pass filter from 3.4 kHz – 5.6 kHz. This is multiplied by the gains for this band on a sub-frame basis. This forms the residue for the first sub-band. This residue is passed through the synthesis filter for the highband to obtain the first sub-band speech.

Sub-band 2

For the second sub-band from 5.6 kHz – 7.2 kHz, random excitation is used. This is because the very high bands do not show any pitch periodicity. The random excitation is weighted with the second set of gains on a sub-frame basis to obtain the residue for the second sub-band. This is passed through the synthesis filter for the highband to obtain the second sub-band speech.

The lowband and the highband speech are added by taking into account that the highband speech is delayed with respect to the lowband speech because of the filters. This is compensated by adding zeros to the beginning of the lowband speech. In the next section we present the results obtained by using the above codec with the G.729 and the G.729e coder. We also present the results obtained for the AMR-WB (G.722.2) codec for comparison.

Post Processing

Post processing includes filtering the speech for bandwidth expansion. This is done to shape the noise similar to the signal such that the signal to noise ratio remains approximately constant throughout. This includes filtering with a long-term filter, a short-term filter and a tilt compensation filter.

3.4 Results

The codecs developed were used with G.729 and G.729E base coders and were tested using PESQ. The standard G.722.2 codec at 15.8 kbps and the G.722 codec at 48 kbps were also applied to the same set of speech files and tested using PESQ for comparison. The set consisted of 8 speakers, 4 male and 4 female, of American English.

Test Procedure

Testing of codecs should ideally be done with trained listeners in sound-proof conditions. But in cases of non-availability of resources for conducting these tests, one can use programs like PESQ (Perceptual Evaluation of Speech Quality) for testing the codecs. These programs try to simulate human listening by various acoustic models. These programs give a good idea about the perceptual quality of speech and are used here.

Mean values

The absolute mean of the PESQ values have been plotted in Figure 3.15. As can be seen from the figure, the developed coder is better than AMR-WB for both nominal and low level speech files.

Male and Female speakers

The mean of the PESQ values for the male and female speakers are plotted in Figures 3.16 and 3.17 respectively.

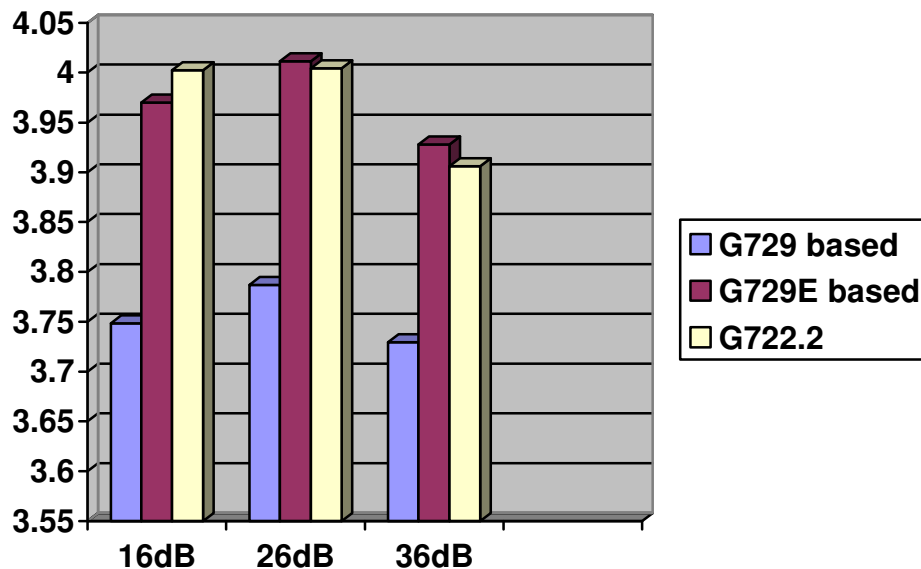


Figure 3.15: Plot of mean PESQ values for all speakers

As can be seen from the plots the G729E based coder is considerably better in the case of female speakers than the G.722.2 codec but is worse in the case of male speakers. Since the mean of the two codecs is the same, this means that the G729E based codec is less sensitive to gender than the AMR-WB codec.

The developed coder was also tested with only one sub-band (3.4 kHz – 5.6 kHz) included. The PESQ values did not vary much but the decoded speech was of slightly lower quality in the unvoiced regions.

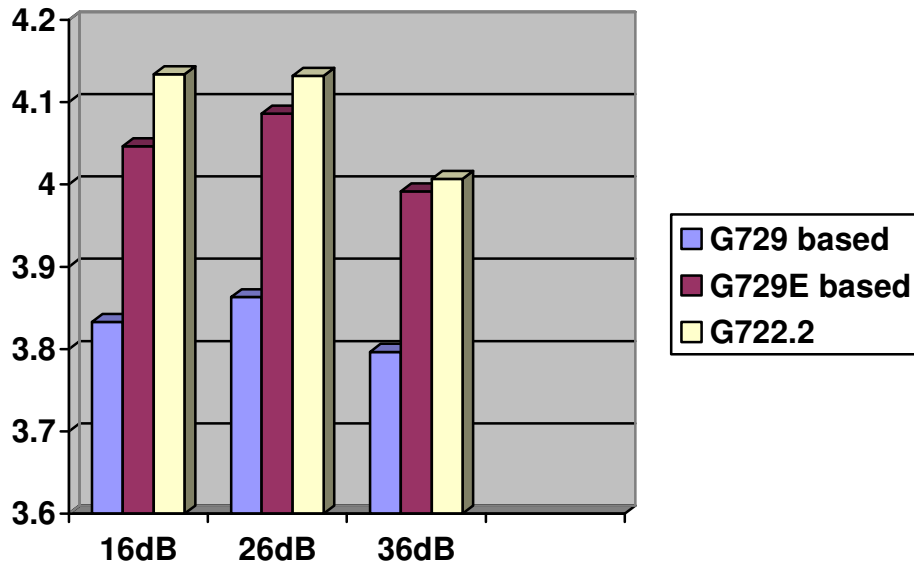


Figure 3.16: Plot of mean PESQ values for male speakers

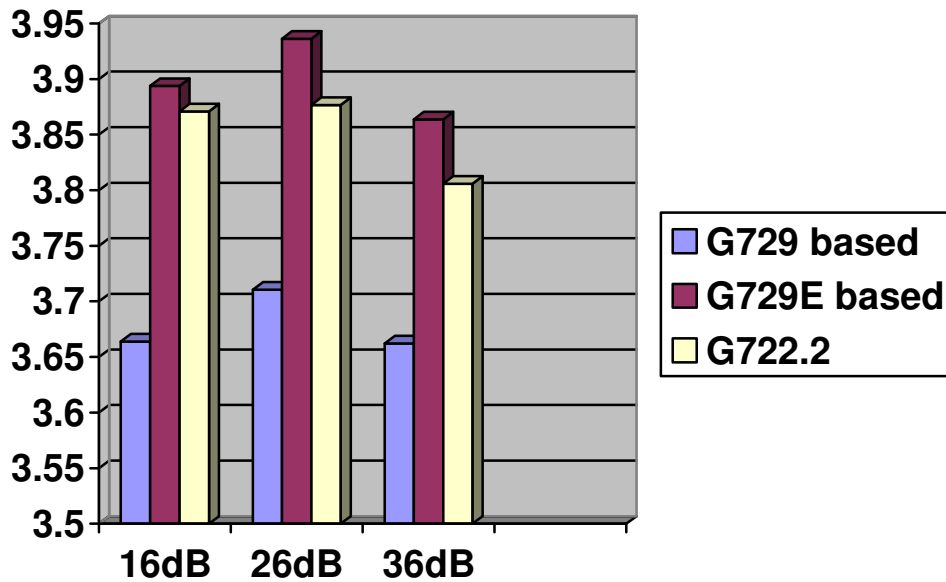


Figure 3.17: Plot of mean PESQ values for female speakers

NWT Test Results

The G729 and the G729E codecs were tested according to NWT with the G.722.2 coded as the reference. The final results are presented below in Table 3.2 and Table 3.3 respectively.

G729 based:

Condition	Test - TT		Reference - MM		Mean Dif.	t	Test (NWT)
	Mean	Stdev	Mean	Stdev			
Low level	3.7294	0.0915	3.9061	0.1225	0.1767	11.3254	Worse
Nominal level	3.7869	0.1063	4.0042	0.1507	0.2174	11.5485	Worse
High level	3.7484	0.1573	4.0022	0.1573	0.2538	11.1828	Worse

Table 3.2: Test results for the G729 based codec

G729E based:

Condition	Test - TT		Reference - MM		Mean Dif.	t	Test (NWT)
	Mean	Stdev	Mean	Stdev			
Low level	3.9277	0.0816	3.9061	0.1225	-0.0216	-1.4353	Equivalent
Nominal level	4.0112	0.1018	4.0042	0.1507	-0.0070	-0.3766	Equivalent
High level	3.9699	0.1573	4.0022	0.1573	0.0323	1.4226	Equivalent

Table 3.3: Test results for the G729E based codec

As can be seen from the tables above, the G729E based codec gives a quality equivalent to and the G729 based coder gives a quality worse than, the G.722.2 codec. This is another evidence that the G729E based codec at 15.4 kbps is of comparable quality to the G.722.2 based coder at 15.8 kbps.

The G729E based codec was also tested was also tested with the G.722 codec at 48 kbps as the reference. The results are tabulated below in Table 3.4

Condition	Test - TT		Reference - MM		Mean Dif.	t	Test (NWT)
	Mean	Stdev	Mean	Stdev			
Low level	3.9277	0.0816	3.8176	0.0966	-0.1101	-8.5340	Better
Nominal level	4.0112	0.1018	3.9741	0.1247	-0.0371	-2.2567	Equivalent
High level	3.9699	0.1212	4.0158	0.1212	0.0459	2.6246	Equivalent

Table 3.4: Test results for the G729E based codec against the G.722 codec

As can be seen from Table 3.4, the developed codec is equivalent to the G.722 codec at 48 kbps at High and Nominal signal levels and better than the G.722 codec at low signal levels.

The results are consolidated in Figure 3.18 which shows the mean PESQ values for the developed codec (G729E based) against the G.722.2 codec at 15.8 kbps and the G.722 codec at 48 kbps.

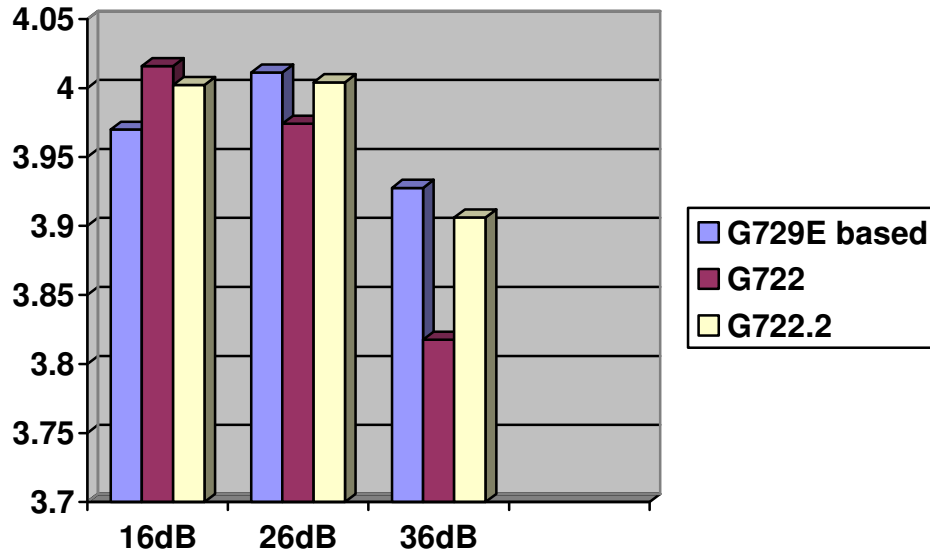


Figure 3.18: Plot of mean PESQ values for all speakers

It can be seen from the above graph that except at high signal levels, the developed coder at 15.4 kbps is better than the G.722.2 coder at 15.8 kbps and the G.722 coder at 48 kbps with the added functionality of scalability.

Conclusion and Future Work

In the present work, we have explored the various methods that can be used in wideband coding of speech. Experiments were conducted on wideband speech in the LPC and the residual domain to capture a correlation between the lowband speech and the highband speech. Based on the results of the experiments wideband scalable codecs were developed and tested. Here we have used the novel idea of using only the lowband excitation with a gain factor for the highband, thus reducing the bitrate considerably. The G729E based codec (at 15.4 kbps) has been shown equal in quality to the AMR-WB codec at 15.8 kbps. Also the codec was found to be of comparable quality to the G.722 [10] codec at 48 kbps.

The prediction of the highband LPC filter from the lowband filter proposed in Chapter 3 was not used in the final codec due to the audible artifacts present in the decoded speech. Future work might modify this procedure and make it usable, thereby reducing the bitrate even further. Another idea for future work is to modify the highband residue at the encoder to have the pitch of the lowband residue and then transmitting only the difference of the highband pitch from the lowband pitch. The reverse can be done at the decoder to obtain the highband speech. This can give better results than the present codec and can be scalable with the present scheme, with the added layer acting as a bitrate scalable layer.

The lower band of 50 Hz – 300 Hz was also not included in the present scheme. One idea to include this might be to take the Fourier series and transmit only the first few coefficients. Another problem for further study is the optimization of the post filter for the highband. In the present scheme, the post filter coefficients are the same as that of the lowband post filter. Experimenting with different coefficients for the highband postfilter might give an optimized filter resulting in a further increase in the quality of decoded speech.

References

- [1] J. L. Flanagan, C. H. Coker, L. R. Rabiner, R. W. Shafer, and N. Umeda, "Synthetic Voices for Computers", IEEE Spectrum, Vol. 7, No. 10, pp. 22-45, October 1970.
- [2] L. R. Rabiner, R. W. Shafer, "Digital Processing of Speech Signals", Prentice-Hall, New Jersey
- [4] "Coding of speech at 8kbps using conjugate structured algebraic-code excited linear-prediction (CS-ACELP)", ITU-T Recommendation G.729, March, 1996.
- [3] A. S. Spanias, "Speech Coding: A Tutorial Review", Proc. of the IEEE, Vol. 82, No. 10, October 1994.
- [5] J. Makhoul, "Linear Prediction: A tutorial Review", Proc. IEEE, Vol. 63, pp. 561-580, 1975.
- [6] Noll, P., "Wideband speech and audio coding", IEEE communications Magazine, pp.34-44, November, 1993
- [7] "Wideband coding of speech at around 16 kbit/s using Adaptive Multi-rate Wideband (AMR-WB)", ITU-T Recommendation G.722.2, January, 2002.
- [8] R. D. De lacovo, D. Sereno, "Embedded CELP coding for variable bit-rate between 6.4 and 9.6 kb/s", Proc. ICASSP'91, pp. 681-683, 1991.
- [9] A. L. Guyader, E. Boursicaut, "Embedded wideband VSELP speech coding with optimized codebooks", Proc. IEEE Workshop on speech coding, pp. 15-16, 1993.
- [10] "7 kHz audio-coding within 64 kbit/s," ITU-T Recommendation G.722, 1988.
- [11] J. Suzuki, N. Ohta, "Variable rate coding scheme for audio signal with subband and embedded coding techniques", Proc. ICASSP'89, pp. 188-191, 1989.
- [12] A. Kataoka, S. Kurihara, S. Sasaki, S. Hayashi, "A 16-kbit/s wideband speech codec scalable with G.729", Proc. EUROSPEECH, pp. 1491-1494, 1997.
- [13] T. Nomura, M. Iwadare, M. Serizawa, K. Ozawa, "A bitrate and bandwidth scalable CELP coder", Proc. ICASSP '98, pp. 341-344, 1998.

- [14] K. Koishida, V. Cuperman, A. Gersho, “A 16 kbit/s bandwidth scalable audio coder based on the G.729 standard”, Proc. of ICASSP '00, Vol. 2, pp. 1149-1152, June 2000
- [15] IETF RFC 3261 “SIP: Session Initiation Protocol”, June, 2002
- [16] “PacketCable Architecture Framework Technical Report”, www.packetcable.com/specifications/specifications20.html, April, 2006
- [17] IETF RFC 2327 “SDP: Session Description Protocol”, April, 1998
- [18] IETF RFC 1889 “RTP: A Transport Protocol for Real-Time Applications”, January, 1996
- [19] IETF RFC 1890 “RTP Profile for Audio and Video Conferences with Minimal Control”, January, 1996
- [20] IETF RFC 3264 “An Offer/Answer Model with the Session Description Protocol (SDP)”, June 2002
- [21] IETF RFC 3262 “Reliability of Provisional Responses in the Session Initiation Protocol (SIP)”, June 2002