August 24, 2012

Waveform Coding Algorithms: An Overview

RWTH Aachen University

Compression Algorithms
Seminar Report

Summer Semester 2012

Adel Zaalouk - 300374

Aachen, Germany

Contents

Chapter 1

An Introduction to Speech Coding

1.1 What is Speech Coding ?

Speech coding can be define as a the procedure of representing a digitized speech signal as efficiently as possible, while maintaining a reasonable level of speech quality as well as a reasonable level of delay.

1.2 A Walk Through The History of Voice Compression

Here is a Glimpse over the history of speech coding.

- 1926 Pulse Code Modulation (PCM) was pointed out by Paul M. Rainey and independently by Alex Reeves (AT&T Paris) in 1937. However, it was only deployed in the the US at 1962.
- 1939 Channel vocoder -First analysis by synthesis system developed by Homer Dudley of the AT&T labs - VODER.
- 1952 Delta Modulation was proposed, Differential Pulse Code Modulation (DPCM) was invented.
- 1957 u-law encoding was proposed (Standardized later for the Public Switching Telephone Network in 1972 (G.711)).
- 1974 Adaptive Differential Pulse Code Modulation (ADPCM) was developed.

- 1984 CELP Vocoder was proposed (Majority of coding standards for speech signal today use a variation of CELP).

1.2.1 Why Voice Compression

Now comes an important question. Why do we need voice compression anyways ? before answering this question lets first have a look at the structure of an encoder and a decoder and try to analyze each block individually.

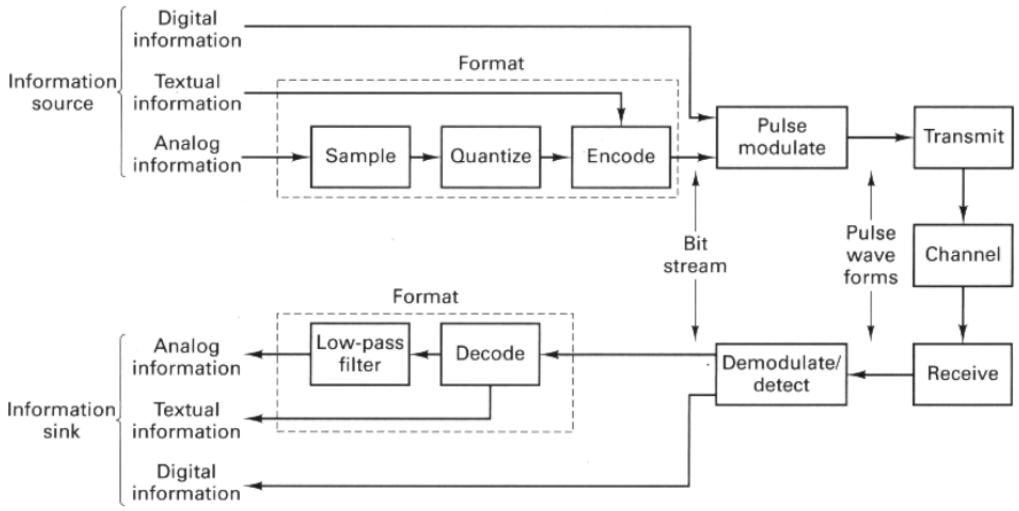

Figure 1.1: Formatting and Transmission of Baseband Signals [5]

Filtering And Sampling

Sampling is the process of representing a continuous time signal by a set of deltas shifted in time. Sampling process is the link between an analog and a digital representation of a signal. Basically, there are many ways to implement the sampling process, the most famous one is called *sample-and-hold* operation. The output of the sampling process is called Pulse Amplitude Modulation (PAM), this is because the output can be described as a sequence of pulses with amplitudes derived from the input waveform samples. Depending on the sampling resolution the original signal can be retrieved from this set of PAM waveform samples by simple low pass filtering.

The sampling process is not perfect however. To present an infinite set of amplitudes of a continuous signal with a finite set of samples might lead

to an incorrect signal reconstruction. This can happen if we "under sample" the signal. Under sampling, means that the signal is not represented with enough samples. When the signal is under sampled, we have what is called as "Aliasing". Aliasing, just means that the original signal became indistinguishable or in-retrievable from the set of samples.

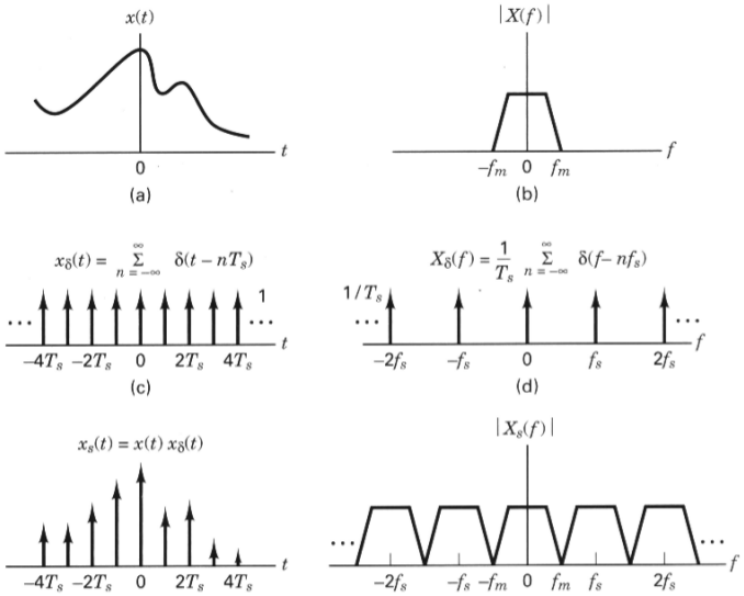

Figure 1.2: Sampling In Time And Frequency Domain [5]

To solve the problem of aliasing. Two scientists, namely "Harry Nyquist" and "Claude Shannon" came a with a solution, and they defined this solution by,

"If a function x(t) contains no frequencies higher than B hertz, it is completely determined by giving its ordinates at a series of points spaced 1/(2B) seconds apart."

That is, to avoid the problem of under sampling. The signal should be sampled at a rate that is greater than or equal twice the maximum signal bandwidth.

$$f_s \geq 2f_m \tag{1.1}$$

To see this, let's have a look at Figure 1.3 and Figure 1.4.

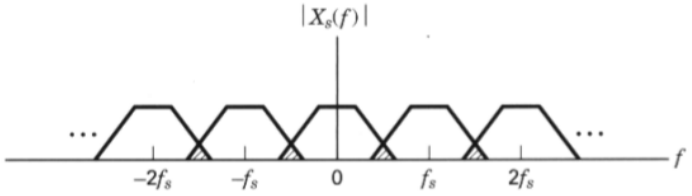

Figure 1.3: Aliasing Due To Under sampling [5]

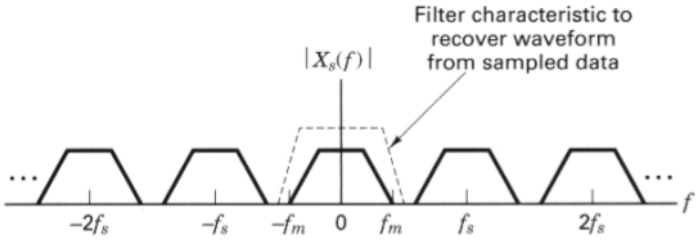

Figure 1.4: Sampling By Following The Nyquist Sampling Theorem [5]

Now, having defined one of the main concepts of transforming a speech signal from analog to digital, let's get back to the question "Why do we need voice compression". According to studies in the field of psycho-acoustics, it has been found that the human speech contents lies in between 300 - 3400 Hz. So, according to the Nyquist theorem that we have just discussed, we should sample the signal with regard to equation 1.1. This implies that the sampling rate should be greater than two times the maximum bandwidth $f_s = 2 \times f_m = 2 \times 3.4 \approx 8kHz$.For good signal quality, let's say that each sample is represented by 16 bits. Therefore, the total bit rate will be $Bitrate = 8kHz \times 16bits = 128kbps$. In some cases the bit rate is even more. For example, in Skype the bit rate used can be 192 kBps (using 16 kHz sampling frequency).

The purpose of speech coding is to reduce the rate required for speech, as can be seen from the following figure.

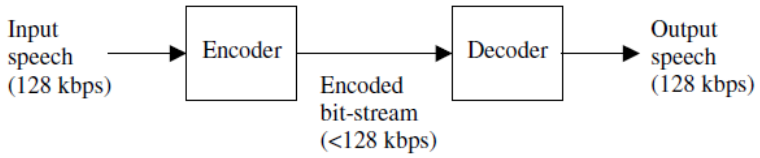

Figure 1.5: Source Coding Concept [1]

Data rate is not the only important metric to consider, other parameters

like delay for example should be kept in mind. The most important metrics to keep in mind while designing a speech coder are,

Low Bit-rate By using a lower bit rate, a smaller bandwidth for transmission is needed , leaving room for other services and applications .

High Speech Quality Speech quality is the rival of low bit rate. It is important for the decoded speech quality to be acceptable for the target application.

Low Coding Delays The process of speech coding introduce extra delay, this might affect application that have real time requirements.

To see this clearly, we are going to look at the factors affecting the delay, these factors are shown in the following figure,

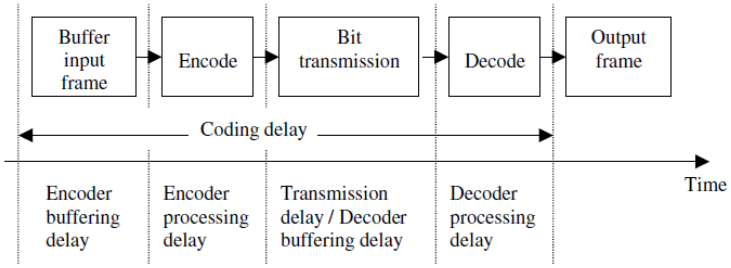

Figure 1.6: Factors Affecting The Delay In A Speech Coder [1]

1.3 Categories of Speech Coding

Speech coding is divided into three main categories,

1. Waveform Codecs (PCM, DM, APCM, DPCM, ADPCM)

   Waveform Codecs gives hight speech quality, without any prior knowledge of how the signal to be coded was generated, to produce a reconstructed signal whose waveform is close as possible to the original.

2. Vocoders (LPC, Homo-morphic, ... etc)

   The vocoder looks at how the speech characteristics change over time. A representation of these modified frequencies is produced as a result at any particular time as the user speaks. In another words, the original signal is split into different frequency bands (The more frequencies used to represent the signal, the more accurate the analysis). The level of the signal in each of these frequency bands gives a direct representation of the spectral Energy content of the signal.

3. Hybrid Coders (CELP, SELP, MELP, RELP, APC, SBC, ... etc)

   Hybrid coding is an intermediate type of coding that between waveform and source coding.

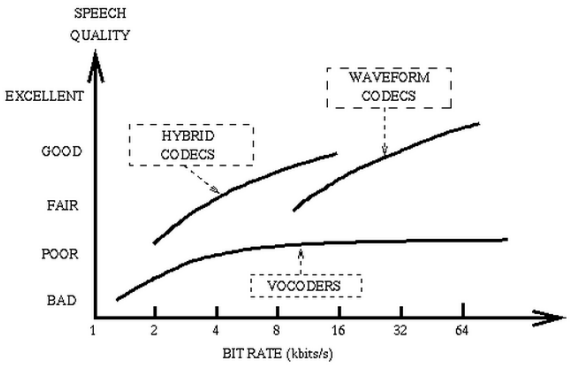

Figure 1.7: Speech Coding Categories [6]

This report work will be focusing more into the waveform coding category and just scratching the surface for the other categories.

Chapter 2

Concepts

This chapter will be focusing into the main concepts of the Waveform speech coding category.

2.1 Quantization

Quantization is the process of transforming the sample amplitude of a message into a discrete amplitude from a finite set of possible amplitudes.

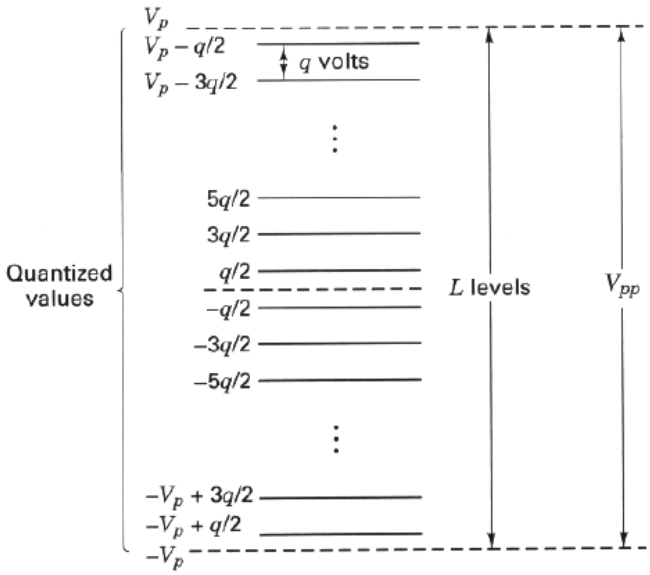

Figure 2.1: Structure Of A Quantizer [5]

As can be seen for Figure 2.1 the quantizer consists of "L" quantization levels and it has a peak to peak voltage V_{pp} and step sizes of "q" volts. To get a feeling of how quantization works, let's have a look at the follwoing figure,

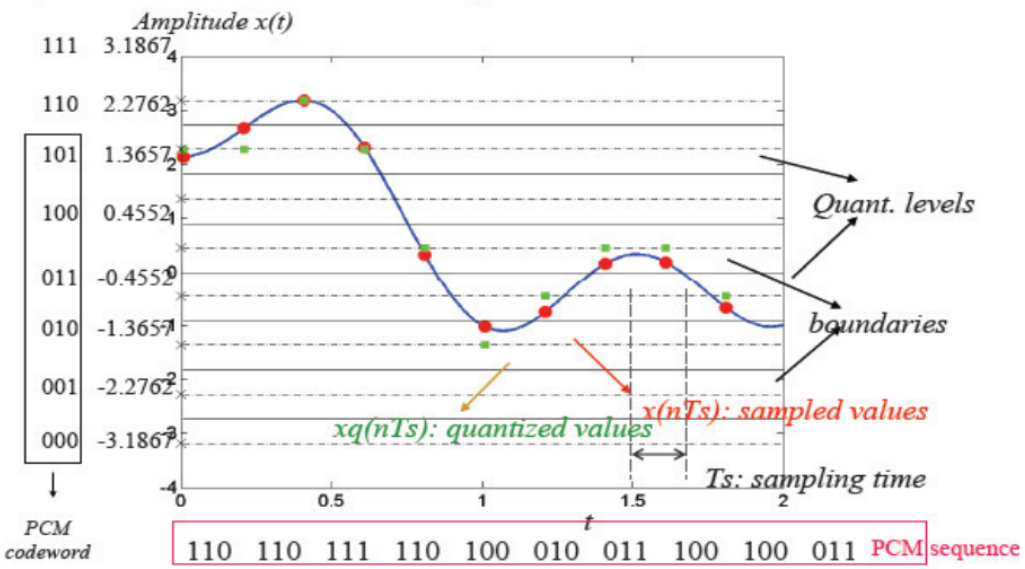

Figure 2.2: Quantization Example [5]

In Figure 2.2, the green dots represents the original sampled values, and the red dots represents the quantized values. As we can see that the original sampled values are mapped to the quantized values, this is because the goal of quantization is to map an infinite set of samples to a finite set, and so there could be two samples of different values, that are mapped to the same quantized values, and this causes what is called "Quantization Noise".

2.1.1 Classification Of Quantization Process

The quantization process is classified into two main categories,

Uniform Quantization The representation levels are equally spaced.

Non-Uniform The representation levels have variable spacing from one another .

Further, the uniform quantization category is subdivided into,

- Midtread Type Quantization
- Midtrise Type Quantization

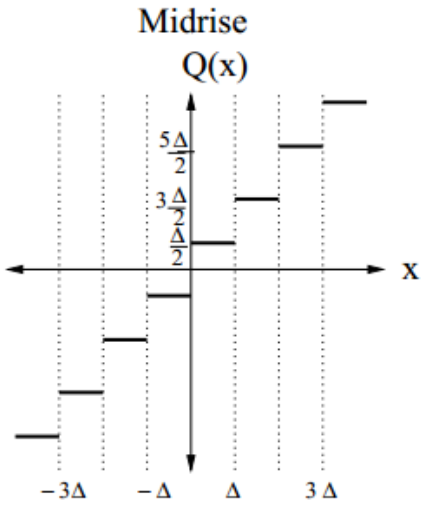

Figure 2.3: Midrise Uniform Quantier [7]

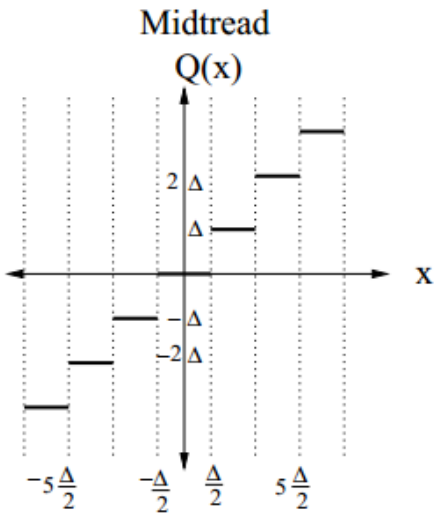

Figure 2.4: Midtread Uniform Quantizer [7]

The difference between mid rise and mid tread quantizers is not that big. However, each of them has its own advantages and disadvantages. Mid rise quantizer's disadvantage is that it does not have a zero-level, this means that weak or unvoiced signals will have to use the first level of the quantizer because they have no zero-level to map to. On the other hand, the mid tread quantizer has a zero level, but it only has an odd number of levels although it was given "B" bits which should always yield an even number of levels 2^B. This leads to underutilization, and less efficient use of the quantization levels.

2.1.2 Human Speech

Speech can be broken into two different categories,

- Voiced
- Un-Voiced

There is a lot of literature that describes both, however it is found the best way to show the difference is by trying to pronounce (zzzzzzz) and (sssssss) the difference is that when saying the first out loud, the vocal tract vibrates causing the voice output we hear. On the other hand, the unvoiced signals does not cause any vibration. For example, the word "Goat" in Figure 2.5

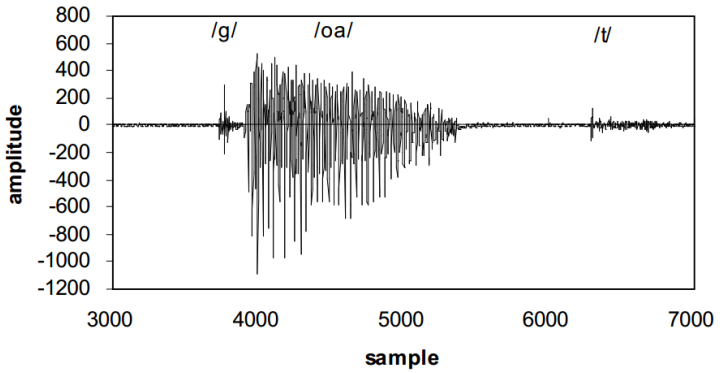

Figure 2.5: How "Goat" Looks like [7]

Goat contains two voiced signals followed by a partial closure of the vocal tract and then an unvoiced signal. Those occurs at 3400-3900, 3900-5400, and 6300-6900, respectively.

It should be noted that the peak to peak amplitudes of voiced signals are approximately 10 times that of the unvoiced signal. However, unvoiced signals contains more information and thus higher entropy than voiced signals, as a result the telephone system must provide higher resolution for high amplitude signals. Figure 2.6 shows the probability of low amplitudes is

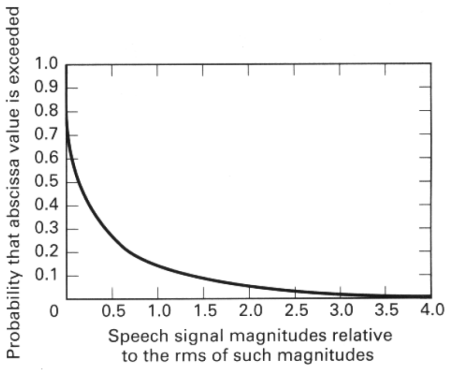

Figure 2.6: PDF Of Speech Amplitudes [5]

much higher that the probability of high amplitudes.

2.1.3 Quantization Noise

The quantization is not a perfect process as anything in this life. It is a lossy process that introduces an error compared to the original signal. An error is defined as the difference between the input signal M and the output signal V. This error "E" is called the quantization Noise $E = M - V$. Consider the following simple example,

$$M = (3.117, 4.56, 2.31, 7.82, 1)$$
$$V = (3, 3, 2, 7, 2)$$
$$E = M - V = (0.117, 1.561, 0.31, 0.89, 1)$$

Consider an input m of continuous amplitude of the range $[-M_{max}, M_{max}]$. Also, assume a uniform quantizer, how do we get the Quantization Noise Power.

Let $\delta = q = \frac{2M_{max}}{L}$ where L is the number of levels. We need to calculate the *Average Quantization Noise Power* φ^2. The average Quatization noise power is defined as,

$$\varphi^2 = \int_{-\frac{q}{2}}^{\frac{q}{2}} e^2 p(e) de \tag{2.1}$$

where $p(e)$ is the Probability Density Function(PDF) of the error and it follows a uniform distribution. And e is the error. If we do the integration we will end up with

$$\varphi^2 = \frac{q^2}{12} \tag{2.2}$$

However, we know that $\delta = q = \frac{2M_{max}}{L}$, so if we substitute in Equation 2.2, we get

$$\varphi^2 = \frac{M_{max}^2}{3L^2} \tag{2.3}$$

From this we conclude the *Average Quantization Noise Power* is inversely proportional with the number of levels in the quantizer. The more levels we have, the less error we get and vice-versa.

The main goal is to decrease the *Signal-To-Quantization-Nose-Ratio (SQNR).* So, given from before that speech signals does not require high quantization

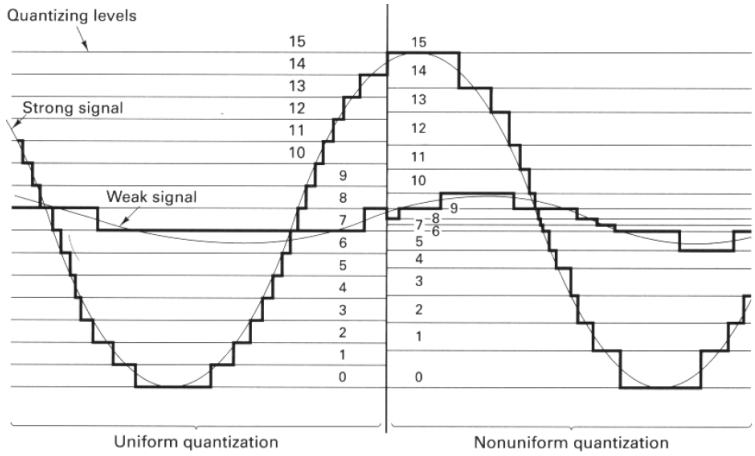

Figure 2.7: Uniform Vs. Non-Uniform Quantizer [5]

resolution for high amplitudes, why not use a non-uniform quantizer, instead of using a uniform quantizer. From Figure 2.8 we can see that for low amplitudes the non-uniform quantizer gives a fine number of levels, where for high amplitudes it gives a coarse number of levels, which matches our goal of decreasing the *SQNR* by increasing the number of levels for low amplitudes.

The question that remains, is how we can construct such a non-uniform quantizer. One way to construct such a non-uniform quantizer is to use what is called "Companding".

Companding = **Comp**ression + Exp**anding**

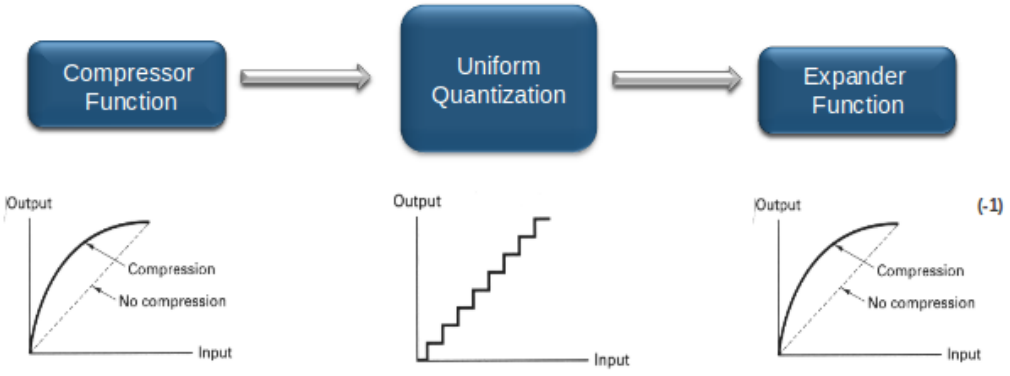

Figure 2.8: The Process Of Companding

The companding process compromises three main steps,

- Compression
- Uniform Quantization
- Expanding

In another words, companding applies a transform to simulate a non uniform signal in a uniform manner.

In the first step the input signal is applied to a logarithmic function and the output of this function is used in the second step. In the second step, a mid rise uniform quantizer is used to quantize the output of the compressor. Finally, the inverse of the logarithmic function used in the compression step is applied to the output of the quantizer. After following the above mentioned steps, we now have non-uniform quantizer with more levels for low amplitudes and less levels for high amplitudes as shown in Figure 2.15

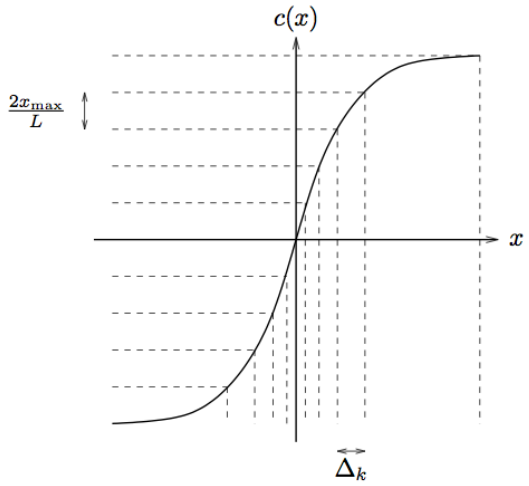

Figure 2.9: Compressor Function

2.1.4 Encoding Laws

In the previous section, the concepts behind companding was explained, however the implementation was not. There are two famous "Encoding Laws" that implement the companding technique.

- A-Law Companding

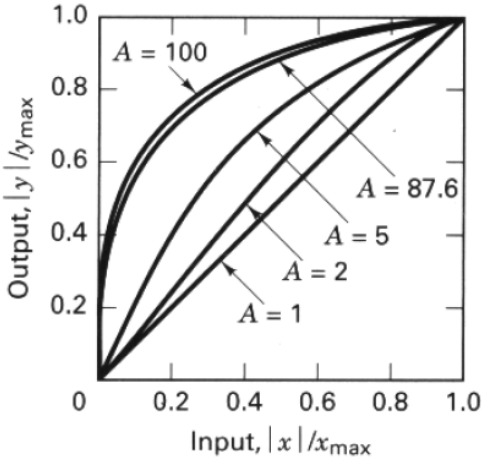

Figure 2.10: A-Law Companding [5]

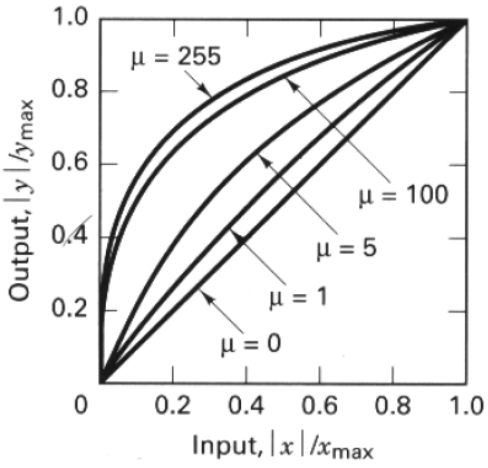

Figure 2.11: μ-Law Companding [5]

- μ-Law Companing

Equations for A-Law are,

$$y = y_{max}\frac{A(|x|/x_{max})}{1 + log_e A} sgn(x) \text{ for } 0 < \frac{|x|}{x_{max}} \leq \frac{1}{A} \tag{2.4}$$

$$y = y_{max}\frac{1 + log_e A(|x|/x_{max})}{1 + log_e A} sgn(x) \text{ for } \frac{1}{A} < \frac{|x|}{x_{max}} \leq 1 \tag{2.5}$$

Equation for μ-Law is,

$$y = y_{max}\frac{1 + log_e[1 + \mu(|x|/x_{max})]}{log_e(1 + \mu)} sgn(x) \tag{2.6}$$

For both,

$$sgn(x) = \begin{cases} +1 & x \geq 0 \\ -1 & x < 0 \end{cases} \tag{2.7}$$

The Algorithm

Logarithmic functions are slow to compute, why not approximate it. The logarithmic functions can be approximated by segments, in our case we will be using three bits that is eight segments (also called "chords") to approximate the logarithmic function.

Our goal is to transform a thirteen or a fourteen bit input to an 8 bit output, as shown in Figures 2.19 and 2.20. In Figure 2.20 **P** is the sign

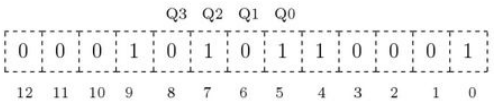

Figure 2.12: Thirteen Bits Input [5]

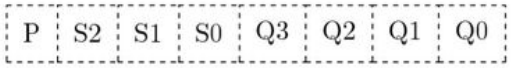

Figure 2.13: Eight Bits Output [5]

bit of the output, the **S's** represents the segment code and finally, the **Q's** are the quantization codes.

To encode an input the following algorithm is executed,

1. Add a bias of 33 to the absolute value of the input sample
2. Determine the bit position of the most significant among bits 5 to 12 of the input
3. Subtract 5 from that position, and this is the Segment code
4. Finally, the 4 bit quantization code is set to 4 bits after the bit position of the most significant among bits 5 to 12

To decode, the following algorithm is executed,

1. Multiply the quantization code by 2 and add 33 the bias to the result
2. Multiply to the result by 2 raised to the power of the segment code
3. Decrement the result by the bias
4. Use **P** bit to determine the sign of the result

Example

P	S2	S1	S3	Q3	Q4	Q5	Q6
1	1	0	0	0	1	0	1

Figure 2.14: Output Of μ-Law Algorithm

The Input to the Algorithm is -656. First, since the sample is negative, then the **P** bit should become 1. Then we add 33 to the absolute value to

bias high input values (due to wrapping), we see that the result of the addition is $689 = 0001 - 0101 - 10001$. Now, we have to find the position of the most significant 1 bit in position range [5,12], in this example it is at position 9. Subtracting 5 from the position values yields 4 (The segment code). Finally, the 4 bits after the last position are inserted as the quantization code.

To decode the sample back, first we notice that the quantization code is 101 which is 5 in decimals, so $5*2+33 = 43$. We also notice that the segment code is 100 which is 4 in decimal, so $43 * 2^4 = 688$. Now we decrement the value by 33 (the bias we added before) and we have 655. Finally, we add the sign and we have -655 as our decoded sample. It should be noticed that the quantization noise is only 1 (very small).

2.2 PCM

Pulse Code Modulation (PCM) is the process of representing quantized samples by a digital stream of bits. After sampling, we are left with Pulse Amplitude Modulated (PAM) samples. PCM takes as an input those PAM samples and uniformly quantizes them. The result of the uniform quantization is mapped to a code number. This code number is finally represented by a set of bits.

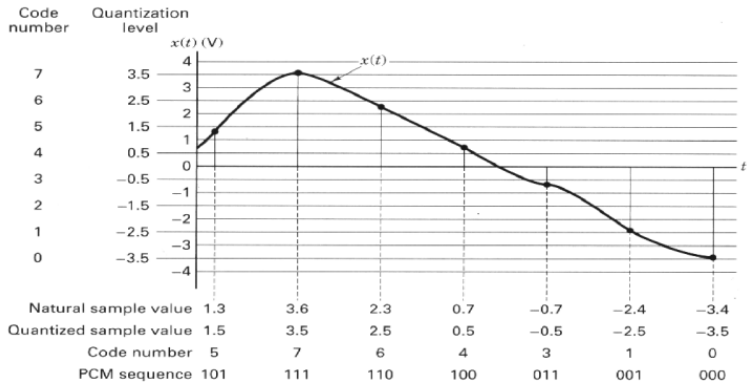

Figure 2.15: Pulse Code Modulation [5]

2.3 DPCM

Differential Pulse Code Modulation (DPCM) adds to PCM by having the following reasoning. Signals that are sampled with a rate much higher than

the "Nyquist Rate 1.2.1" have highly correlated samples, so why not use this correlation relation for our advantage. Instead of representing each sample independently, why not only encode the difference between the current sample and the previous one? By following this reasoning we will have a quantizer with much less number of bits, hence we are only encoding the difference.

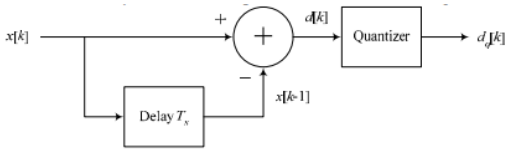

Figure 2.16: DPCM Encoder [10]

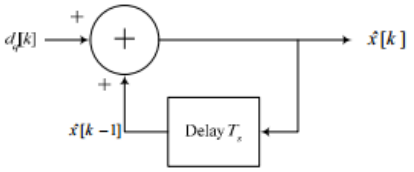

Figure 2.17: DPCM Decoder [10]

By only using the one previous sample in our calculations , we are using what is called first order prediction,

$$d[k] = x[k] - x[k-1] \tag{2.8}$$

We can also use more than one previous sample in the prediction process, in that case we call it N-Order prediction,

$$d[k] = x[k] - \sum_{k=0}^{k=N} x[k-1] \tag{2.9}$$

The DPCM approach is not perfect, it also has its problems, consider Figure 2.18 each sample $x[k]$ is subtracted from the previous sample $x[k-1]$

k	−1	0	1	2	3	4	5	6	7	8	9
$x[k]$	0	0.3	1.5	0.7	1.0	2.3	3.7	2.8	3.5	2.8	0
$x[k-1]$	0	0	0.3	1.5	0.7	1.0	2.3	3.7	2.8	3.5	3.1
$d[k]$	0	0.3	1.2	–0.8	0.3	1.3	1.4	–0.9	0.7	–0.7	–2.8
Quantization Up/Down	U (or D)	U	U	U	U	U	U	U	D	U	U
$d_q[k]$	0.5	0.5	1.5	–0.5	0.5	1.5	1.5	–0.5	0.5	–0.5	–2.5
$\hat{x}[k-1]$	0	0.5	1.0	2.5	2.0	2.5	4.0	5.5	5.0	5.5	5.0
$\hat{x}[k]$	0.5	1.0	2.5	2.0	2.5	4.0	5.5	5.0	5.5	5.0	2.5
$\hat{x}[k]-x[k]$	0.5	0.7	1.0	1.3	1.5	1.7	1.8	2.2	2.0	2.2	2.5
Err. Direction Up/Down	U	U	U	U	U	U	U	U	D	U	U

Figure 2.18: DPCM Cummulative Quantization Error [10]

and then the result is quantized. The problem arises because of the erroneous

quantization process that add noise to the original input. At the decoder, when the quantized difference $d[k]$ is added to $x'[k-1]$ a completely different $x[k]$ is perceived as the result, and this is because the decoder does not have access to the $x[k]$ used at the encoder, and due to this difference the problem of cumulative noise arises.

To solve this problem, the input to the predictor of the decoder should be the same as the one that is used as the encoder's predictor input. Consider the following two figures, the first figure will result in a cumulative noise whereas, the second figure will fix the problem by moving the quantizer inside the feedback loop to give the same input to the predictor.

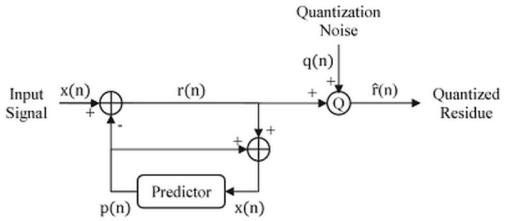

Figure 2.19: DPCM Encoder With Quantizer Outside [2]

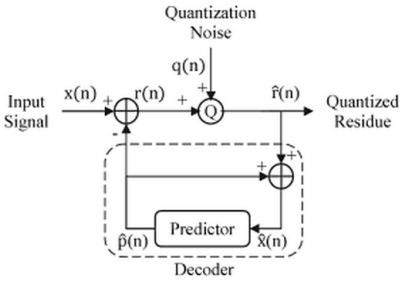

Figure 2.20: DPCM Encoder With Quantizer Inside [2]

2.4 ADPCM

Having discussed about PCM and DPCM, Adaptive Differential Pulse Code Modulation (ADPCM) is not much different. The only difference here is the "A". The "A" stands for adaptivity, the main idea here is varying the quantization step size. So for example, a four bit sixteen level quantizer have small step size between the levels for low amplitude differential input samples and large step size for high amplitude differential input samples as shown in Figure 3.1. Different rates can be achieved by ADPCM by using different number of bits for quantization, as will be shown later in the standards section.

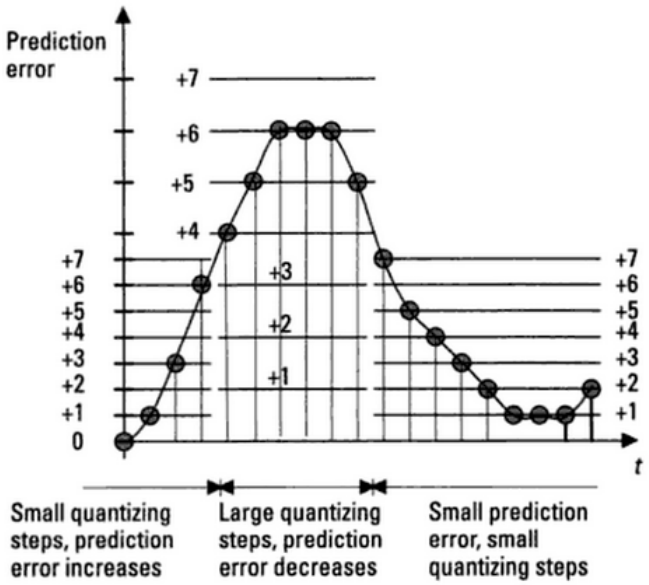

Figure 2.21: ADPCM Example [12]

Chapter 3

From Concepts To Standards

This chapter will give a brief introduction for the standards G.711 and G.726

3.1 G.711

G.711 is a Waveform codec that has been released in 1972. It's formal name is Pulse Code Modulation (PCM) since it uses it as the main concept for encoding. The G.711 standard achieves 64kbps bit rate by using 8kHz sampling frequency multiplied by 8 bits per sample.

The G.711 standard defines two main compression algorithms,

- A-Law (Used in North America & Japan)
- μ-Law (Used in Europe and the rest of the world)

A and laws algorithms as an input 14-bit and 13-bit signed linear PCM samples and Compress them to 8-bit samples.

Applications

The G711 standards is used in the following applications,

- Public Switching Telephone Network (PSTN)
- WiFi phones VoWLAN
- Wideband IP Telephony
- Audio & Video Conferencing
- H.320 & H.323 specifications

3.2 G.726

The G.726 standard makes a conversion from 64 kbps A or μ Law PCM channel to and from 40, 32, 24 and 16 kbps. This conversion is applied to raw PCM using the ADPCM encoding technique, G.726 has support for different rates by adapting the number of quantization levels,

- 4 - levels (2 bits and 16 kbps)
- 7 - levels (3 bits and 24 kbps)
- 15 - levels (4 bits and 32 kbps)
- 31 - levels (5 bits and 64 kbps)

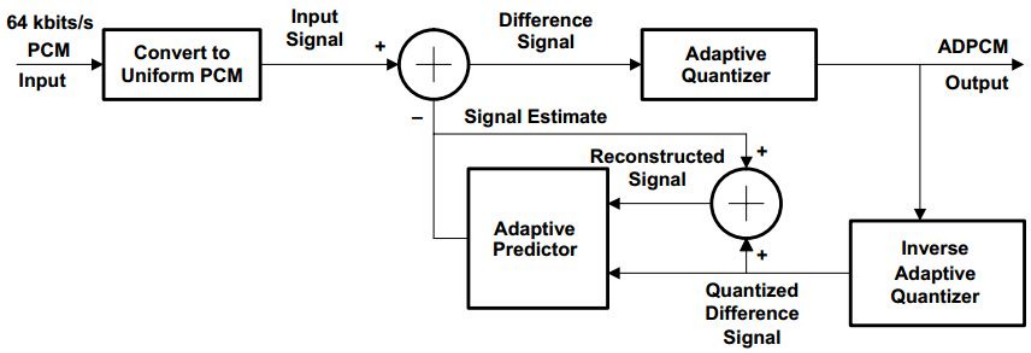

Figure 3.1: G.726 Encoder [15]

The G.726 standard includes also, the G.721 and the G.723 which both use ADPCM.

Applications

The G.726 applications are very similar to G.711's.

Chapter 4

A Performance Comparison

To give an overview about all the prominant speech codecs out there and to have a hawk's eye view of the most important metrics for speech codecs, consider Figure 4.2

Here are some remarks of the protocols mentioned on the graph,

G.711 Support very good quality, but it requires a very high data rate. It also has a very low latency (not complex)

G.726 Requires half of the rate needed by G.711, and is used in many open source frameworks like Asterisk.

G.728 Uses Code Excited Liniear Prediction (CELP) which support compression for verly low delays.

G.729 Support for very good qualit. However it has high processing delay.

G.723.1 Support for two bit rates 6.3 & 5.3 kbps using MPC-MLQ & ACELP algorithms. It also has support for very good quality.

GSM Uses Linear Prediction Coding, has support for 13kbps, it also has three versions (Half Rate, Full Rate and Enhanced Full Rate).

FS1015 Developed by the U.S and later by the NATO, it is also known as LPC10, does not require high data rate and still gives good quality. However, it has a very high delay.

IS-54 Digital AMPS (TDMA in digital cellular Telephony)

IS-69 North American CDMA (Digital Cellular Telephony)

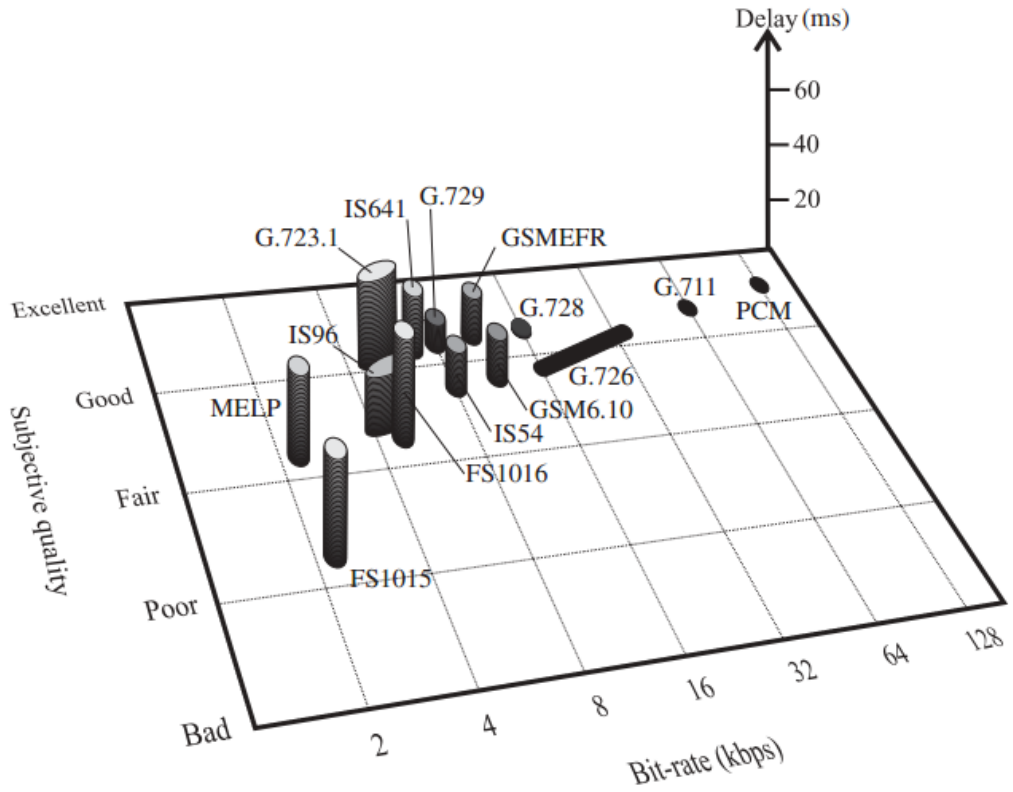

Figure 4.1: A Performance Comparison Between Speech Codecs [1]

MELP Mixed Exitation Linear Prediction, founded by the U.S DoD Speech Coding Team and it is mostly used for military applications, it has good speech quality, very low data rate and acceptable delays.

Year Finalized	Standard Name	Bit-Rate (kbps)	Applications
1972[a]	ITU-T G.711 PCM	64	General purpose
1984[b]	FS 1015 LPC	2.4	Secure communication
1987[b]	ETSI GSM 6.10 RPE-LTP	13	Digital mobile radio
1990[c]	ITU-T G.726 ADPCM	16, 24, 32, 40	General purpose
1990[b]	TIA IS54 VSELP	7.95	North American TDMA digital cellular telephony
1990[c]	ETSI GSM 6.20 VSELP	5.6	GSM cellular system
1990[c]	RCR STD-27B VSELP	6.7	Japanese cellular system
1991[b]	FS1016 CELP	4.8	Secure communication
1992[b]	ITU-T G.728 LD-CELP	16	General purpose
1993[b]	TIA IS96 VBR-CELP	8.5, 4, 2, 0.8	North American CDMA digital cellular telephony
1995[a]	ITU-T G.723.1 MP-MLQ / ACELP	5.3, 6.3	Multimedia communications, videophones
1995[b]	ITU-T G.729 CS-ACELP	8	General purpose
1996[a]	ETSI GSM EFR ACELP	12.2	General purpose
1996[a]	TIA IS641 ACELP	7.4	North American TDMA digital cellular telephony
1997[b]	FS MELP	2.4	Secure communication
1999[a]	ETSI AMR-ACELP	12.2, 10.2, 7.95, 7.40, 6.70, 5.90, 5.15, 4.75	General purpose telecommunication

Figure 4.2: Remarks about Speech Codecs [1]

Chapter 5

Summary & conclusion

5.1 Summary

- The quantization concepts was explained in all it's flavors, then the categories of waveform coding (PCM,DPCM and ADPCM) has been discussed and illustrated. A brief overview for the standards (G.711 & G.726) has been given, and finally a comparison was shown for the most prominent speech codec's out there.

5.2 Conclusion

Speech coding is an important concept that is required to efficiently use the existing bandwidth. There exist many important metrics to keep in mind when doing speech coding. It is important for a good speech coder to balance those metrics. The Most important ones are data Rate, Speech Quality and Delay. Waveform codec'sachieves the best speech quality as well as low delays. On the other hand, Vocoders achieves low data rate but at the cost of delays and speech quality and finally, Hybrid coders achieves acceptable speech quality and acceptable delay and data rate.

Bibliography

[1] *Speech Coding Algorithms: Foundation and Evolution of Standardized Coders*: Wai C. Chu .

[2] *Principles of Speech Coding*: Tokunbo Ogunfunmi

[3] *Speech Coding: A Tutorial Overview*: Andreas S. Spanias

[4] *Science of Speech Coding*: Sanjeev Gupta

[5] *Digital Communication Fundementals & Applications*: S. Klar

[6] *Speech Coding* HTTP://WWW-MOBILE.ECS.SOTON.AC.UK/SPEECH_CODECS/

[7] *A-Law and mu-Law Companding Implementations Using the TMS320C54x*

[8] *Signal Quantization and Compression Overview* HTTP://WWW.EE.UCLA.EDU/~DSPLAB/SQC/OVER.HTML

[9] *Data Compression Introduction to lossy compression*: Michael Langer

[10] *Ch. VI Sampling & Pulse Code Mod. Lecture 25* : Wajih Abu-Al-Saud

[11] *Audio Coding: Theory And Applications* : Yuli You

[12] *Introduction to telecommunication Networks Engineering*: Tarmo Anttalainen

[13] *Wikipedia G711:* HTTP://EN.WIKIPEDIA.ORG/WIKI/G.711

[14] *Data Communication the Complete Reference*: David Salomon

[15] *ITU CCIT Recommendation G.726 ADPCM*