Digital Voice and Picture Communication

Prof. S. Sengupta

Department of Electronics and Communication Engineering

Indian Institute of Technology, Kharagpur

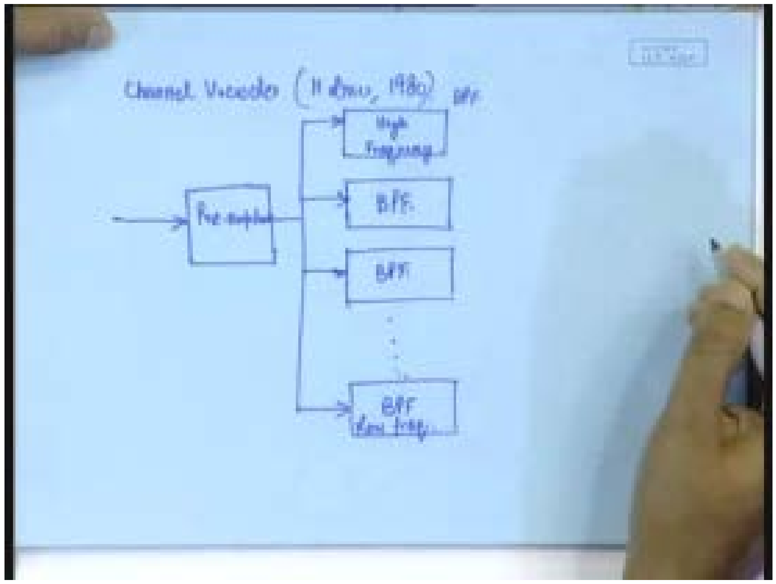
Lecture - 03

Speech Coding Objectives and Requirements

So, towards the end of the last lecture I was telling you about the speech analysis and synthesis requirements. So, as we had seen in the last class that essentially we can have a source filter model for the speech generation process and in that source filter model we can say that the source excitation is actually provided with the glottal pulses and then the vocal tract acts as a linear but time varying filter.

So, in accordance with the speech requirements we change the parameters of that filter so that different sounds as we desire can be produced. So, when we would like to have a speech synthesis what we need to have is to go in for a speech analysis and synthesis system which would look something like this. The one that I am going to describe is called as the channel vocoder and this was actually developed by homes Homer sorry Holmer in 1980 and in the channel vocoder essentially what we are going to have is that there will be a pre-emphasis filter some prefiltering basically and then the pre-emphasis outputs will be a fed to a bank of band pass filters. So here we have the high frequency and progressively the lower and lower frequency band pass filters will be used. So it is a bank of BPFs. So this is the highest frequency BPF (Refer Slide Time: 3:25) and here we have the next higher frequency or next lower frequency to that then the next BPFs and finally the lowest frequency BPF or rather in other words that will be a low pass filter. It is a bank of band pass filters that we are talking of over here so this is the lowest frequency.

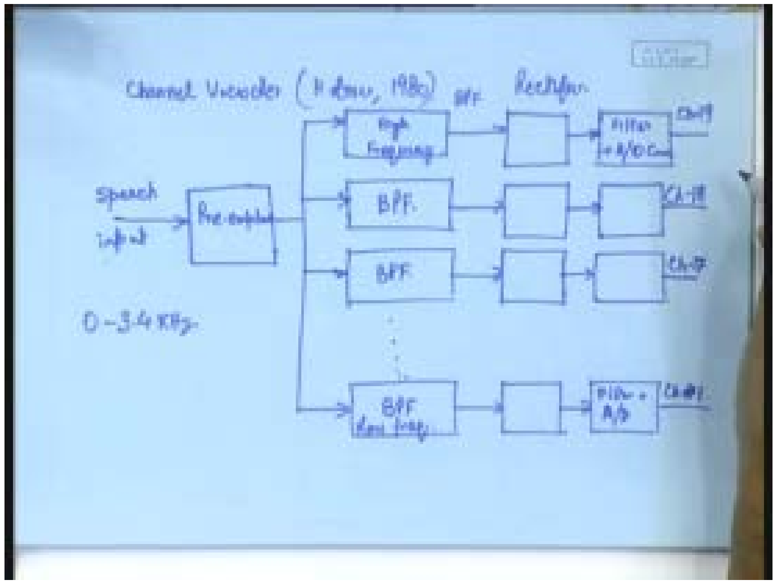
(Refer Slide Time: 3:55)



Then these filter outputs actually will be fed to rectifiers. Essentially what we want to do is to extract the energy of the different bands. So, given the speech input which is fed over here (Refer Slide Time: 4:28) we are going to analyze it into different frequency components. So, in the range of 0 to 3.4 kilo hertz we will be having a bank of such filters and individually every filter bank will every filter banks energy will be computed.

So what we have to do is that till now everything is analog but after the rectifier what we have to do is to use a smoothening filter and then A to D converters. So I am writing here as filter plus A to D converter A to D converter and that we have to do for all these channels. So typically we can show that here in Homer in Holmer’s implementation we have actually nineteen such channels. So we can number the channels like this. All these are the different filters plus A to D filter plus A to D and here this is channel number....... the channel number we start from here (Refer Slide Time: 5:44) from the lowest frequency so this is channel number 1 and so on and the last the highest frequency will be channel number 19 and the one that is next to it is channel number 18, the one next to that is channel number 17 and so on and the lowest frequency is channel number 1.

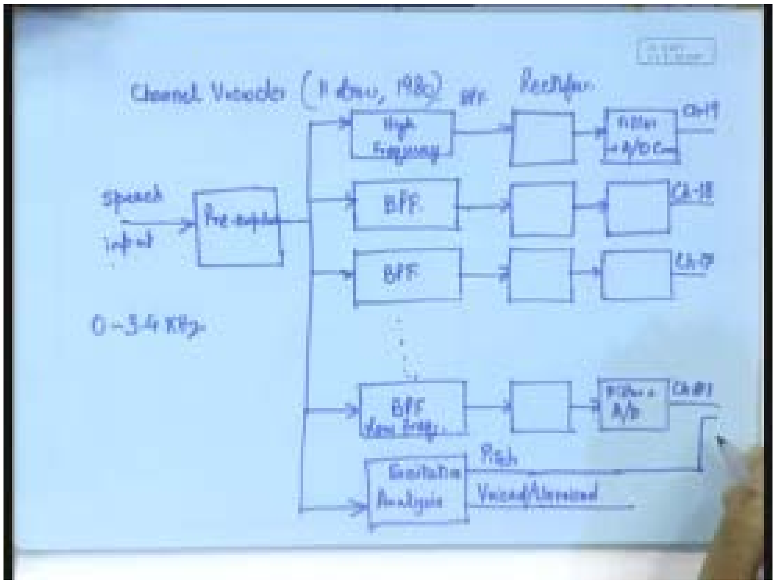
(Refer Slide Time: 6:07)



Now this essentially analyzes the speech signal into different filter banks. And along with that we also have the extraction of pitch information. Because, as I told you in the last class that pitch forms an essential parameter in order to analyze speech and obviously if it is needed for analysis it is needed for the synthesis also. So the pitch information has to be derived from the given speech signal. So there is an excitation analysis block. In fact pitch extraction process is quite standardized and one of the very popular techniques that can be adopted for the pitch extraction is auto correlation based method. I am not going to describe that over here and what I would suggest all of you to do is to refer to any of the standard books which I mentioned in the first lecture. So you can find some good references whereby you can learn about the pitch extraction process where as I told just now that auto correlation based method is one of the very effective and popular names.

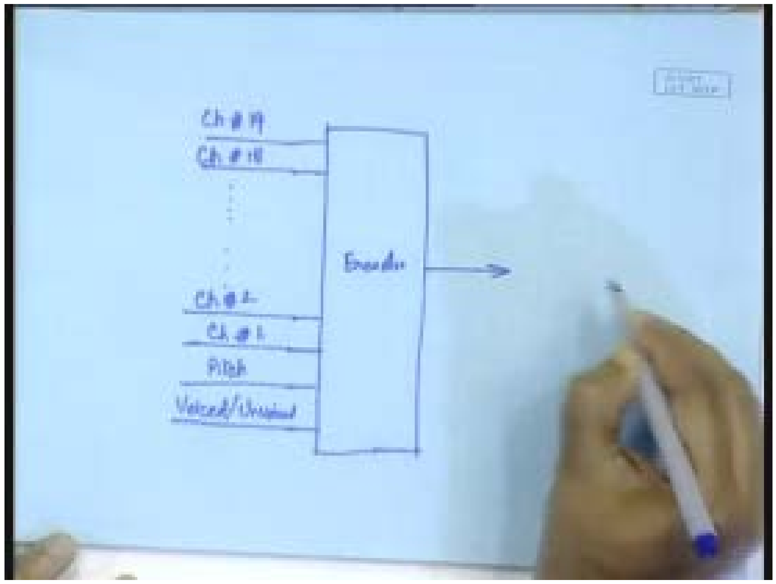
So, at the end pitch will be extracted out of this and not only that the excitation analysis block is also going to find out whether the speech signal that is present over here is a voiced speech or an unvoiced speech. So what we are going to do is to have that decision available from this excitation analysis block rather a voiced and unvoiced.

(Refer Slide Time: 8:04)



So it is a binary decision that either it is voiced or it is unvoiced and accordingly all this information that means to say these nineteen different filter bank outputs and this voiced unvoiced flag, everything together will be actually fed to the encoder. In fact there will be a proper bit allocation and everything for all these items that is to say the nineteen channels, then the pitch information also is to be represented and the voiced and unvoiced which is one bit flag that also has to be encoded. So together I mean I will just show it in a separate page where we have channel number 1 to channel number 19. So this is channel number 19 the highest frequency, channel 18 and so on up to channel 1 channel 1 channel 1 being the lowest frequency and then we have the pitch information and then the binary decision that is whether it is voiced speech or unvoiced so all these will get encoded. So there will be an encoder which will be encoding this information together. So channel 1 to channel 19 we essentially have; I mean, in the process of rectification what we have done is that we have extracted its amplitude because signals are bipolar in nature that is why it will be varying from the positive as well as to the negative so what we do in the process of the rectification is to extract the envelop of that and this in turn gives us an idea about which filter bank contributes to what proportion of the signal. In other words, it is a say kind of spectral analysis that we have done using this bank of filters.

(Refer Slide Time: 10:14)

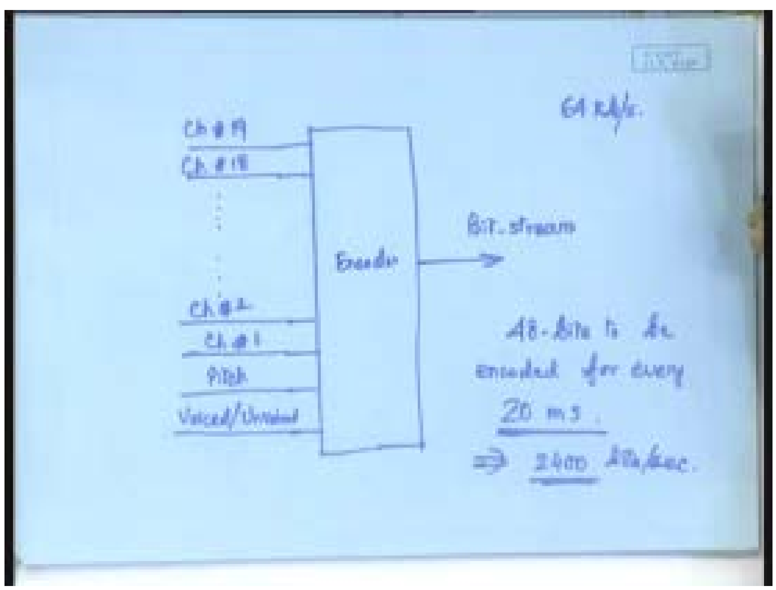


Then this encoder’s job will be......... see, basically here everything is digital in nature so all these digital information will be encoded and the appropriate bit stream will have to be generated. So this is the analysis followed by encoding scheme. So what we have here, that is to say speech followed by this bank of filters, it is a kind of analysis filtering (Refer Slide Time: 10:50) and then this speech extraction which also is essential because speech forms an integral part of the analysis system and for the proper synthesis of speech what we have to do is to extract all these individual items out of the bit stream that is generated from the encoder.

We are talking of what decoder is going to do. Decoder is going to extract all this channel information, pitch information, voiced unvoiced flag and then accordingly the pitch will excite the source and then accordingly we will be having the synthesis that is to say this channel 1 to channel 19 outputs in turn we will decide the spectral shape. So this whole thing is called as the channel vocoder (Refer Slide Time: 11:50). The word vocoder basically originates from voice encoder. It is a very popular thing and in this methodology essentially what we have done here is a kind of parametric coding; and why parametric coding is because we are not approximating the speech waveform like we do in the case of a say pulse code modulation or the adaptive pulse code modulation and so on which is essentially waveform approximating technique. But in this case it is completely parametric. We are not approximating the waveform rather we are just making the analysis and then getting the information out of it.

Now in this process it is possible to represent the information in a much compact way as compared to the standard waveform approximating techniques. In fact what one can do is that in Holmers channel vocoder we require 48 bits to be encoded for every 20 millisecond so 48 bits to be encoded for every 20 milliseconds. That means to say that effectively the bit rate will be 2400 bits per second. Just try to compare this with the pulse code modulation technique which I had told that that is 64 kilobits per second and why 64 kilobits per second is because 3.4 kHz being the speech signal bandwidth we decided to sample it at 8 kHz and represent 8 bits for each of the samples so that gives rise to 64 kilobits per second as against we have got only 2400 bits per second.

(Refer Slide Time: 14:09)

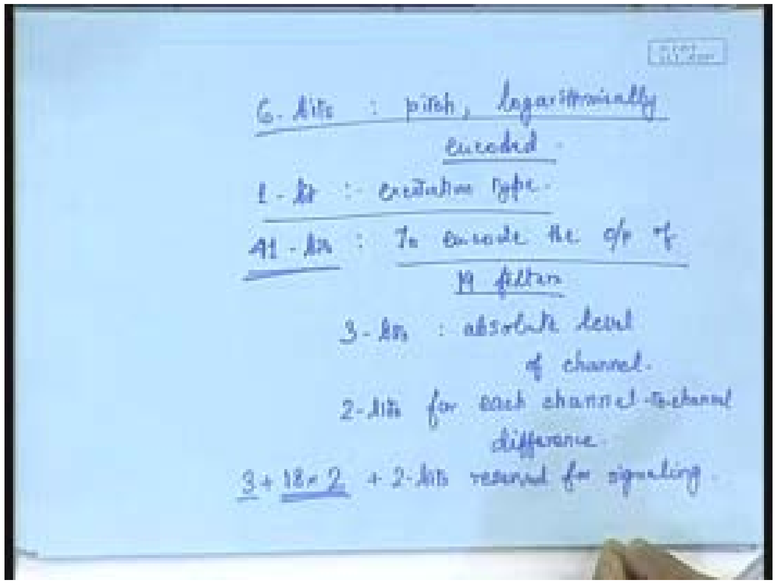


And again the reason for 20 milliseconds I think should be clear from my earlier lecture where I have mentioned that speech signal is essentially quasi-stationary and that quasi-stationary behavior is assumed for a time between 20 milliseconds to 30 milliseconds that is something quite typical. So, in 20 milliseconds of interval you can say that all these parameters that we are extracting; those parameters may be held constant during this 20 milliseconds of time.

And what is the division of these 48 bits?
Well, it goes like this that we have 6 bits allocated to encode the pitch information. So 6 bits are needed for the pitch information encoding and it is in fact logarithmically encoded. So pitch which is logarithmically encoded that requires 6 bits of information. then we require 1 bit to specify the excitation time and what is that excitation time it is the voiced unvoiced decision, and then we require 41 bits to encode the output of nineteen filters; 41 bits to encode the output of nineteen filters and how that is arranged; actually 3 bits are there for encoding the absolute level of channel 1. Channel 1 is channel 1 means remember channel 1 is the lowest frequency so its absolute level is encoded in 3 bits; absolute level of what? It is the absolute level of the rectified and digitized signal. That is to say essentially it represents what is the amplitude of the signal component that is present for the lowest frequency band; so that requires 3 bits.

And then from channel 2 onwards we do not encode the absolute level rather we just encode the channel to channel difference. It means, when we are encoding channel number 2 channel number 2 will be the very next filter bank from the lowest frequency and its output will be encoded using the channel to channel difference and channel to channel difference requires less number of bits. Whereas 3 bits are needed absolute level, the channel to channel incremental difference needs only 2 bits so 2 bits for each channel to channel difference; and how many such channels are to be encoded? We have already encoded the channel number 1 so that requires 3 bits and then the remaining eighteen channels; 2 bits for each channel to channel difference.

(Refer Slide Time: 18:28)



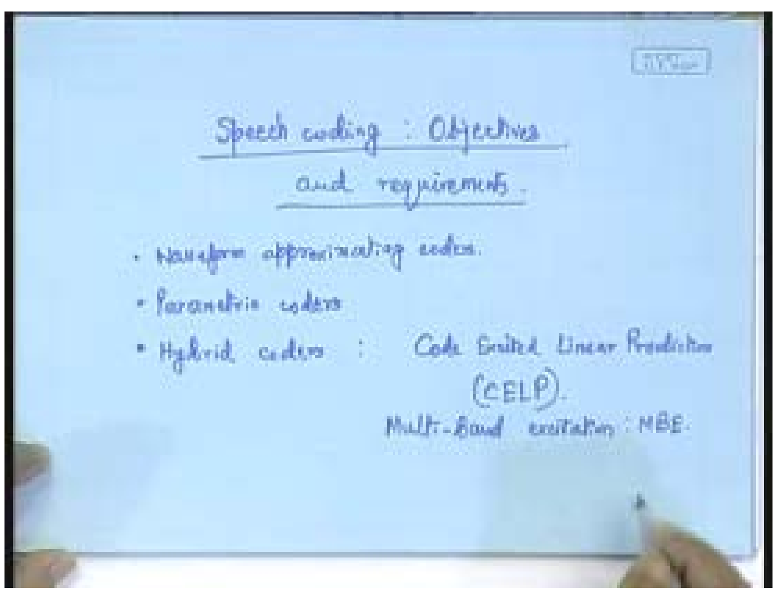
So what we have is 3 plus.......... for eighteen such channel to channel difference we have 18 into 2 so 36 plus 3 so 39 and we reserve 2 bits for signaling or monitoring, so 2 bits are reserved for signaling. So, in all we have 3 plus this channel to channel 18 differences which is 36 plus 2 bits for signaling so all together it is 3 plus 36 plus 2 so 41 bits. So 41 bits for all the nineteen filters, 1 bit for excitation, 6 bits for pitch so these things together makes it 48 bits. So 48 bits we need to encode for every 20 millisecond of time requiring 2400 bits per second which is pretty low; I mean astonishingly low if you compare that with the regular PCM's bit rate.

So now, having understood all these we are in a position to talk about the main topic of today's lecture which is actually speech coding. I will talk about some basic fundamentals of speech coding and in this I will be talking about its objectives and requirements.

Now I have already told in the first lecture itself that the speech coders can be divided into two distinct classes: one is what I mentioned as the waveform approximating coders and the other is the parametric coders. Examples you already know; waveform approximating coders: PCM, ADPCM, these are waveform approximating coders and parametric coders as I described just now, one form of parametric coder is the channel vocoder which I just now described. But there are other advanced parametric coders which have been developed till date and many of these parametric coders are highly efficient and they are capable of producing a natural sound. I mean, it can produce something very close to the natural speech signals and that is why it is very popular. Now, there is also a third class which we sometimes talk about and that is called as hybrid coders. In fact hybrid coders try to take the advantage of both these methods that is waveform coding as well as parametric coding and some of the examples of hybrid coders, very popular example is what is called as code excited linear prediction.

Now, linear prediction is required for the parameters; because as I have already told you in my earlier lecture that essentially the vocal tract is modeled as a linear filter so you need to have some linear parameters which is based on the linear prediction. So linear predictive coefficient is to be encoded and then this is code excited and in short form this is referred to as C E L P or CELP and also the multiband excitation technique. We are not going into details of this right now. But later on we will be talking more about the parametric encoding scheme. So multiband excitation basically corresponds to what we call as the MBE; in short form it is referred to as MBE.

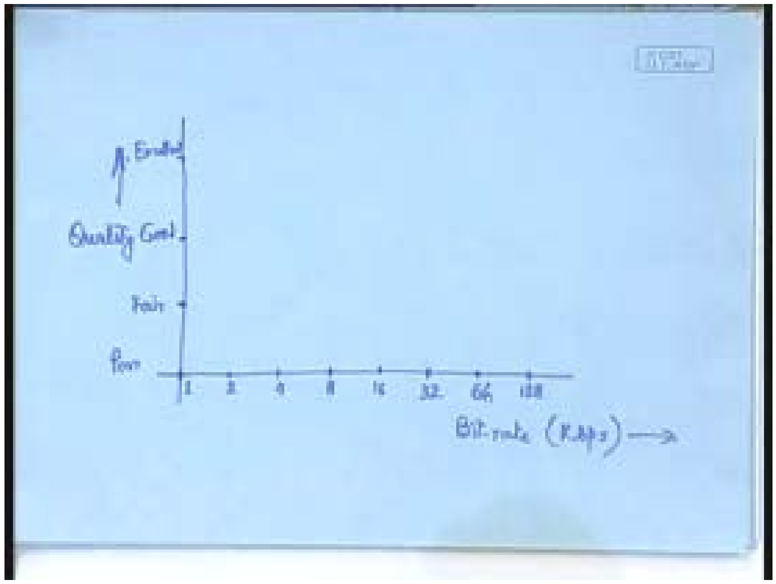
(Refer Slide Time: 22:09)



Now just to tell you about some qualitative difference between these two major forms of coders that is waveform approximating and parametric, we can just give a subjective comparison in this manner. We just plot, in the x axis we indicate the bit rate and the bit rate we indicate in units of kilobits per second and what we do is that we choose a logarithmic scale to represent this so here I write 1 which means that it is 1 kbps and then we just keep dividing into regular intervals, this is 2, this is 4, this is 8, this is 16 so you can understand that it is divided in logarithmic scale 32, 64, 128 and so on. Hence, more than that we do not require because at the high end we have up to 64 kilobits per second that is for the pulse code modulation as I have talked of.

And here (Refer Slide Time: 23:37) on the y axis we indicate the quality. Now when we say quality it is somewhat subjective. So how do we assess the quality subjectively? Well, there what we do is that we just form kind of a mean opinion score that is to say that if you ask a group of people to evaluate the performances of the different coders and if you ask all those listeners to make a choice that on a scale of 4 or a scale of 5 you just indicate that whether it is of excellent quality or whether it is good quality or fair quality then you collect all these responses and using these responses you just form a statistics that whether most of the people really liked this coder output or whether this coder output is not being talked not being rated well by the listeners; so like that you have some kind of a subjective assessment and then you just have it as a scale of poor, poor is the lowest, fair, then good and at the highest level you can have it as excellent.

(Refer Slide Time: 25:00)



So, definitely what we are trying to do is that we should definitely have we should definitely try to reach the excellent. The ideal situation would be that we should reach excellent quality at very low bit rate. But as you know that is not possible. In fact the tradeoff goes in the other way. That means to say that, as we decreases the bit rate, if we go more on this side, decreasing the bit rate obviously requires sacrificing the quality. Basically there is loss of information. If you are going in for waveform approximation techniques then the way by which you can achieve a lower bit rate is to use very coarse quantizations. So coarse quantization means you are sacrificing the finer levels; you are very crudely quantizing and as a result of that your quality is going to suffer. So definitely you can expect that for a very low bit rate the quality will be poor anyway.

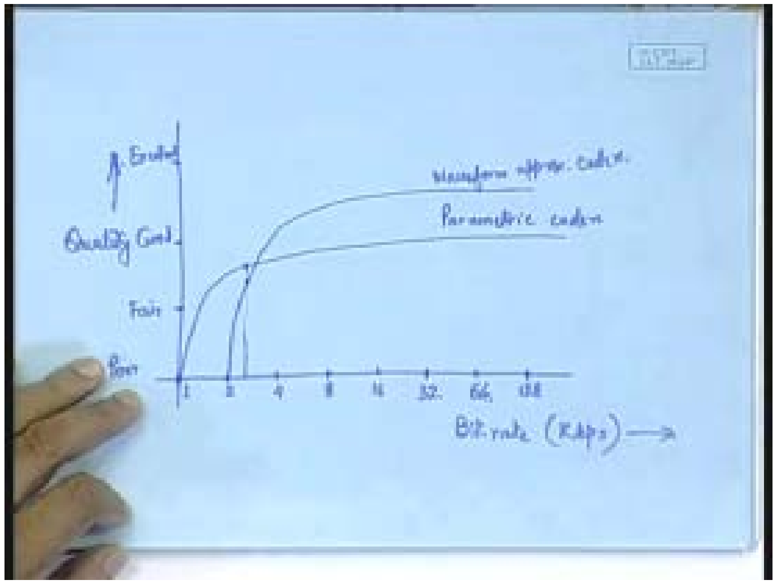
Going by this bit rate figure, for waveform approximating coder one can just forget about having anything in the 1 kilobits per second, impossible. So we have to try that out the waveform approximating coders and after 4 or 8 kilobits per second generally the quality become somewhat acceptable. So, if we draw a curve based on a number of listeners' observations the listeners normally tend to have a rating as something like this that 2 kilobits per second they would definitely call the presentation to be poor but when you go in for a higher a bit rate then your quality improves.

But beyond certain point, like here you can see that after 32 kilobits per second to 64 kilobits per second the improvement may be a just a bit marginal, marginal improvement but nobody is really rating it to be excellent but something between good and excellent. If you want excellent may be you have to go in for a much higher bit rate so that essentially, I mean, unless you have absolutely lossless encoding it is not possible to reach the excellent verity because excellent category means that the original speech and the decoded and the constructed speech they should be indistinguishable from each other.

So these are the curves that we should get for waveform approximating coders (Refer Slide Time: 28:07). And then for the parametric coders actually what happens is parametric coders the one that I have described a little while back, one thing is that you are not exactly using things like a quantizer etc so there is no waveform approximation that you are trying out. You are going into the totally different domain that you are making an analysis and then just subdividing into band representations and those bands are quite a crude form.

Even though you have nineteen filter bands but nineteen filter bands representing a band of 3.4 kHz definitely means that you have some good amount of bunching in the sense that a group of frequencies you are clubbing into one particular channel. So obviously you have loss of information and the parameters that you are extracting is definitely not giving the kind of natural feeling and as a result of that the performance of parametric coders are generally much poorer as compared to the waveform approximating coders. So something of this nature, a curve of this nature may be obtained. So this is for parametric coders.

(Refer Slide Time: 29:47)



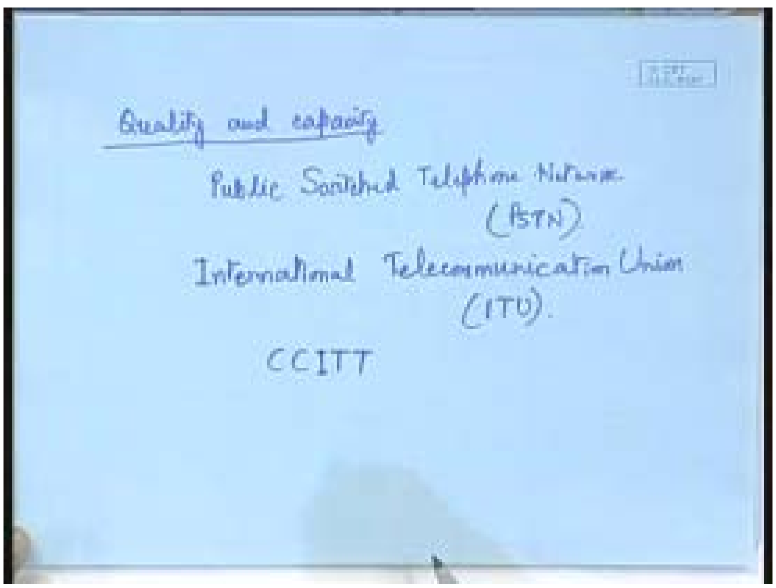
But definitely, by comparing these two curves we can have a good inference in the sense that just see that if you want a very low bit rate coder then realizing it using waveform approximating coders will not yield you a better result because like say for example you take it somewhere around 3 kilobits per second or so, you definitely find that the parametric encoders are better in performance as far as this curve is concerned. only for higher bit rate, when you can have bit rate beyond 4 kHz then definitely according to this curve we find that the waveform approximating coders they perform better as compared to the parametric coders.

Hybrids okay; hybrids will fall somewhere in between this waveform approximating and the parametric coder class. Now, some of the basic objectives of speech encoding we need to understand and that is to say that what is the quality and what is the quality requirements of this.

As I was telling you a little while back that there is always a tradeoff between quality and capacity. You go in for a higher capacity channel, you can deliver a good quality speech output and you compromise on the capacity that is to say you go in for if you are constrained to use a channel of lower bit rate capacity then quality is definitely going to suffer. Now this tradeoff is very easily understood but the point is that there should be some regulatory bodies to monitor the quality. Especially whenever you are working in the public domain like the public switched telephone networks; whenever you are in the Public Switched Telephone Network domain which in short form we all call as the PSTN; for PSTN what should be the quality requirement for a given capacity that must be fully specified. That quality anyway we should reach we have to reach so that is specified by some regulated by some regulatory bodies.

One international regulatory body that specifies for PSTN and other networking requirements other channels and that regulatory body is the International Telecommunication Union or ITU. In older times it was known as the Consultative Committee of International Telephones and Telegraphs CCITT. So all the old timers must have heard about CCITT standards and now it is called as the ITU standard, the International Telecommunication Union.

(Refer Slide Time: 33:29)



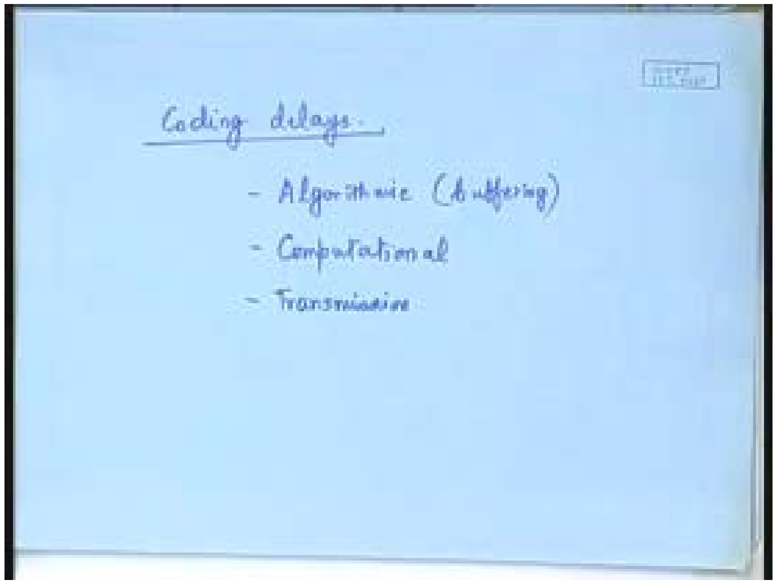
Of course there are other regulatory bodies for specifying the GSM speech quality standards or for the CDMA speech quality standards for the mobile phone applications. So essentially what one has to do is that the speech coders that we design should be designed according to one such standard specified by the ITU so that one follows the regulations very strictly. Of course for absolutely private networks like within one enterprise or say for military systems they can afford to deviate from the ITU standards. But as long as you on in the Public Speech Telephone Network domain you have to abide by the ITU. So definitely the quality verses capacity compromise; you have to do in accordance with the standards and various standards have come up. We will very shortly talk about some of these standards.

Another requirement that comes for the specification of the speech coding is the amount of coding delay. Now, coding delays essentially has got three components. The coding delays may be algorithmic. Essentially the algorithmic coding delays arise out of buffering of signal. You see, your algorithm may be requiring analysis of the past samples and if you want to analyze the past samples several past samples have to be used for the analysis, you have to have some kind of a buffering or you want to measure some statistics, you have to store it for some time so the buffering is required. So, algorithmic coding delays are very much possible.

The next thing is the coding delay that results from computations. So those are referred to as computational coding delays. So it means to say that essentially in a very very in a different language we can also call that as the processing delay. Actually, whenever we are using a very standard PCM for example, there we hardly have any computation. Because all that we are doing in a PCM is that we are taking sample by sample and then we are just approximating that sample using some quantization and the quantized signal we are representing by bits and it is that bits which we are sending in the bit stream, as simple as that; so there is no computational aspect that is involved. Whereas as you go in for more and more sophisticated codecs codec designs there the computational design there the computational coding delays comes into picture.

So we will be seeing that if we have to encode the speech at a very low bit rate for which we have to use the parametric coders but to extract the parameters for that to derive or to extract the linear predictor coefficients what we have to do is to do lot of computation and processing; lot of computation and processing will be required within the computer. So, naturally more amount of processing gives rise to processing or computational delay. So, that becomes significant when you go in for lower bit rate coders.

(Refer Slide Time: 38:25)



And the third delay is of course due to the transmission. Now this is certainly not within in your control because if you are doing long distance communication then transmission delays are definitely inevitable. But what can do is that, from the encoder itself one should design the encoder in such a way that the algorithmic and the computational delays are minimized.

In fact we would be calling those speech encoders to be more efficient for which the algorithmic and computational delays are minimized to a great extent. In fact what bad effect do we normally see when there are coding delays; what we have is that when coding delays occur that time that time the speaker is often irritated with a kind of echo that takes place. If there is a coding delay then from the source to the destination it ultimately gets played back at the destination at a later instant which may be something of the order of lets say 100 milliseconds or 200 milliseconds and if that happens then there will be a round trip of that because the speech signal what is produced at the receiving end a part of that again comes back to you in the form of echo and that echo is really interfering with your normal speech process.

I do not know whether it must be that some of you must have experienced this kind of echo effects.

Therefore, coding delays should be minimized for the reduction of the echoes. In fact when you have something like a 64 kilobits per second PCM then neither do you have any algorithmic delay nor do you have any computational delay; it is just the transmission delay that comes into effect and normally with transmission delays we do not have that much of problem unless it is a very long root that it is taking otherwise this is perfectly fine. But algorithmic and computational delays are definitely something that needs to be minimized.

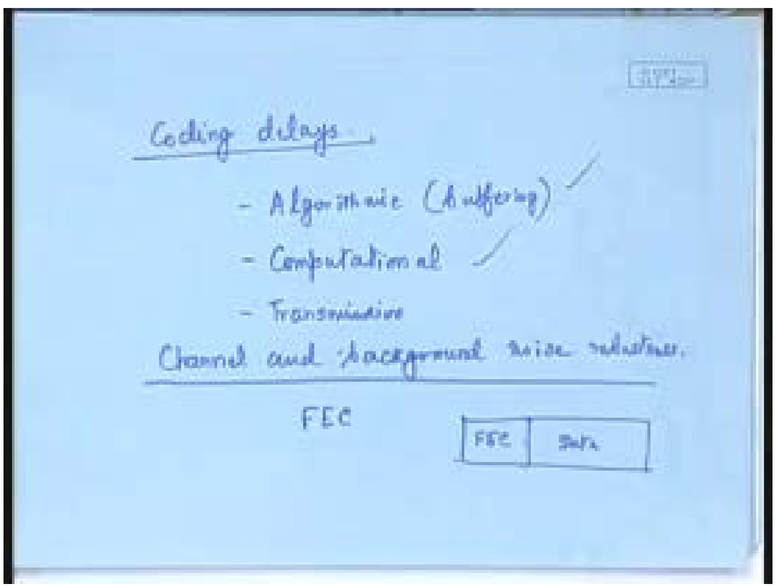
And then while designing the speech coders one should also take into consideration one should also take into consideration the channel and the background noise and whatever encoders that we design should be adequately robust against these. So channel and background noise robustness that becomes a very important consideration.

Typically the source coding of speech that requires only a fraction of the total channel capacity. Because what we have to do is that in order to make it somewhat robust against errors we generally incorporate some forward error correction codes or what we call as the FEC. So, FEC requires a substantial fraction of the bits stream like....... if your actual data bits stream or the speech samples that takes this much your FEC may be taking this much. So definitely since the error robustness requires lot of extra information to be sent in, it is tending to be somewhat inefficient.

Therefore, rather than having data or rather the source coding outputs as separate and then embedding the channel coding by providing the forward error corrections and having a complex scheme like that it is better that if we have a combined channel and source coding in the sense that at the time of source coding itself if it is possible to specify that what should be the that during the source encoding itself if you can take care of the robustness so that if a part of the data is lost if a packet is lost then we should be able to recover the bit stream with minimum amount of perceptible loss, that is important because for speech at least the perception is highly crucial whereas in in the case of images it may be not that crucial because some of the pixels may get disturbed, you may not be able to view some pixels if it is effected by noise or even if there is a loss of one single frame you may not notice that because if the very next frame onwards is available correctly then you will not even notice anything whereas in the case of speech if there is a discontinuity then it really affects the intelligence aspect. That means to say that your understanding will be completely affected, your understanding of speech output will be affected if you have such kind of discontinuity. So error robustness is very important and more so from the consideration of today's technology of wireless communication, mobile communication because the mobile communication is definitely subject to burst errors and packet losses are possible. So one has to take care of the channel as well as background noise.

So far we were talking about the channel noise reduction. The same thing also applies for the background noise reduction that whenever you have background noisy background environment then your speech encoding should be able to extract the signal or maximize the signal and reduce the noise content at the source itself and then you can have a better background noise reduction, robustness against background noise.

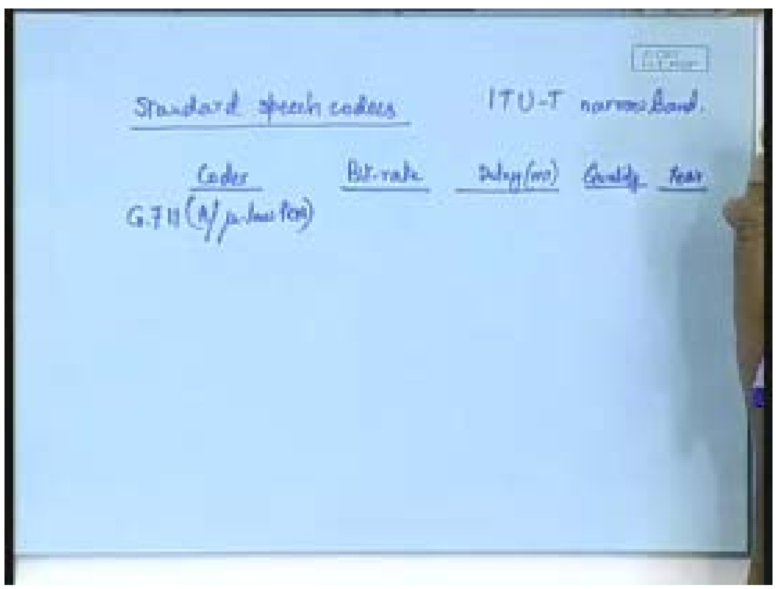
(Refer Slide Time: 45:30)



So these factors are all very important while designing the speech codecs. And as I was telling you, a number of speech coding standards have been proposed in the recent years. Why recent? I should not say recent; right since 1972 the standards have come up so we will be just listing some of the standards speech codecs which have been adopted by the ITU, ITUT narrow band speech codecs. So let us present a list of that.

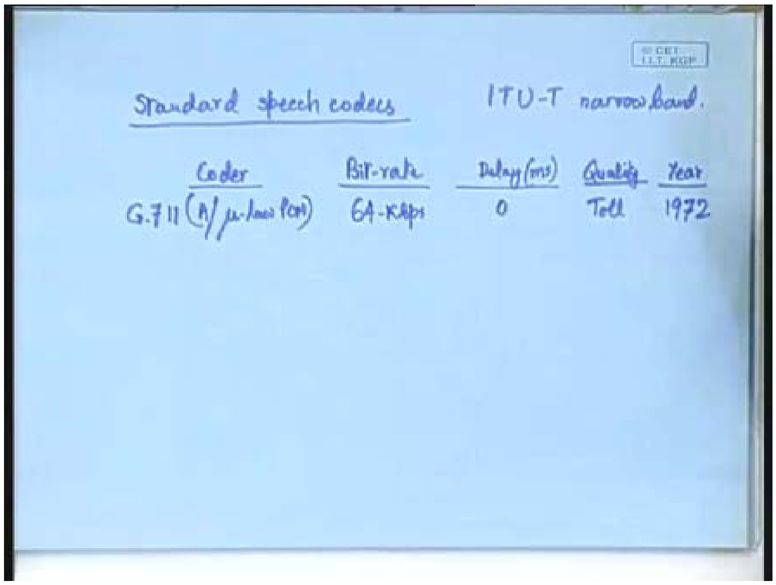
Now we begin with the oldest speech codecs. So here we just list the name of the coder or the standard number that is assigned because the standard coder has got a type number that is associated with that and then we also mention the corresponding bit rate for that coder, and then we also specify the delay, the delay we specify in milliseconds and then the kind of quality. Actually what ITUT specifies is that it should be of a good quality which refers to as the toll quality transmission. So toll quality requirements are...... at least near toll quality requirements must be met and then we also say that in which year was that standard adopted.

(Refer Slide Time: 47:42)



So we begin with the G dot 711 coder which is basically an adaptive and mu law PCM. We will be coming very shortly. I suppose, in the next one or two lectures we will be covering this adaptive and mu law PCM. Essentially as I had told you already that PCM confirms to a bit rate of 64 kilobits per seconds. So in today's technology 64 kilobits per second is definitely high. But, on the other hand, going by the coding delays and the quality its really very good that its delay you can say almost as 0 millisecond, I mean, forget about the transmission delay, but as far as the coding, as far as the algorithmic and computational delays are concerned it is 0 and the quality that this delivers is the toll quality and this was actually proposed in the year, this was adopted in the year way back in 1972, so 35 years back from today.

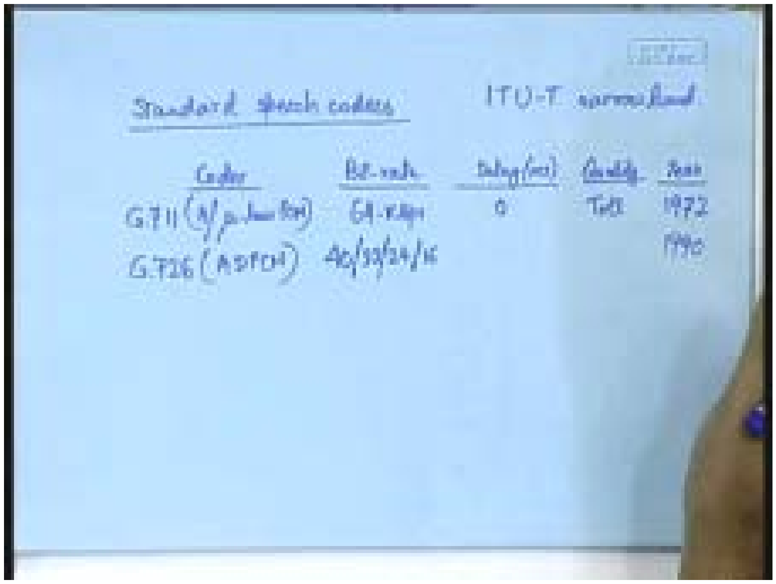
(Refer Slide Time: 49:04)



The next encoder that we have is the G dot 726. in fact G dot 726 it came much later, it came in 1990 and G dot 726 is based on the Adoptive Differential Pulse Code Modulation which we call as the ADPCM; we will of course be learning about the ADPCM with a great deal of the details in the other lectures. In the lectures to follow you will learn details about this. But just to suffice by telling you in a nutshell, here the bit rate that we have obviously is expected to me much lower than this 64 kilobits per second.

Because, as I mentioned in the very first class that differential way of representing the information reduces the dynamic range of the signal content in the sense that we are making a kind of waveform approximation so we have the input speech and then we have the speech that is predicted the speech signal that is predicted which is based on the past samples past speech samples and then it is the error in the prediction what we input; that is a simple differential pulse code modulation and we make the step size of differential pulse code modulation as adoptive giving rise to ADPCM. That caters to different bit rate requirements and the bit rates which are specified by ITUT it caters for 40 32 24 16 up to 16 kilobits per second. So 16 if you compare with 64 definitely it is 1 is to 4 a very substantial reduction.

(Refer Slide Time: 51:24)

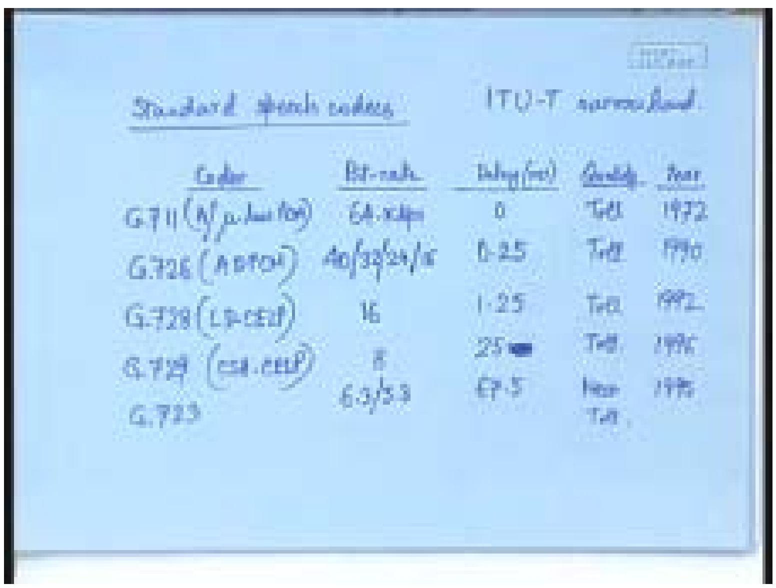


Now, ADPCM definitely requires some amount of buffering definitely because you see how are you predicting the samples. You are predicting based on the past samples. so obviously you need to have some storage so as a result of that it gives rise to delays which are of course marginal because in terms of milliseconds it is just 0.25 milliseconds of delay which is worse than this PCM but somewhat better definitely and the quality of that is also a toll quality.

The next that we talk about is the G dot 728 encoder. G dot 728 is not exactly a waveform approximation coder; it is actually a hybrid coder which is LDCELP CELP stands for Codec Excited Linear Prediction. I am not explaining this in detail right now. In fact we will be talking about the CELP and the linear prediction in general we will talk in the later lectures. This caters for a bit rate of 6 kilobits per second and the delay that is encountered in this process is somewhat more 1.25 for toll quality speech and this was proposed in 1992.

And then came G dot 729 which is also another variety of the CELP CSA-CELP and this brings down the bit rate to 8 kilobits per second. This was adopted in 1996 and this also gives toll quality. But in order to make it work at 8 kilobits per second which is a substantial reduction in bit rate eight times as compared to the original PCM but on the bad side of it, it incurs a delay of 25 milliseconds; of course I did not write milliseconds because here all the delays that I am talking of is referred to in milliseconds. So it is 25 milliseconds for that and the standard G dot 723 which was actually proposed in 1995 that gives rise to near toll quality and on the bit rate side it comes to as low as 6.3 or even as low as 5.3 but delay is tremendous; it is 67.5 milliseconds a substantial delay.

(Refer Slide Time: 54:20)

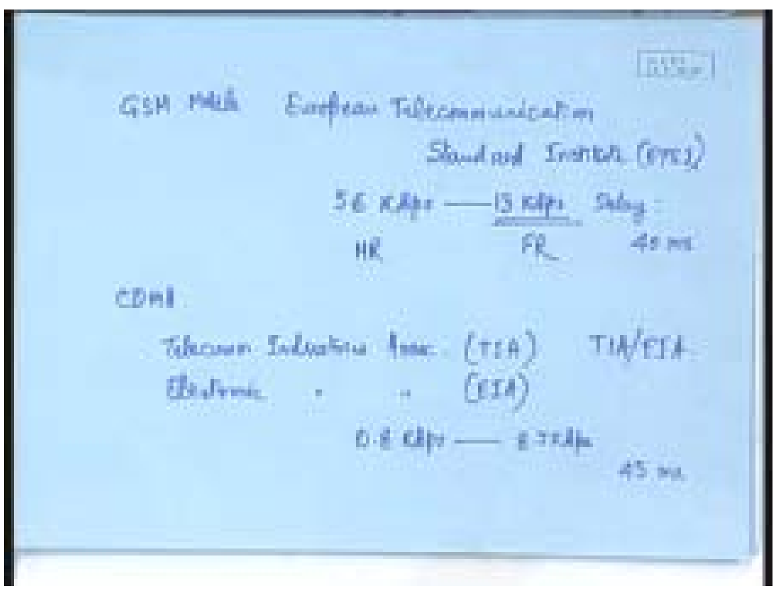


So you can see that, as we are going in for more efficient bit rate coders, the delay increases and although till here we are able to maintain the toll quality anything beyond anything below 8 kilobits per second you are no longer able to give the toll quality but you have to give a near toll quality. And in fact for the mobile standards they have accepted the near toll quality and for the GSM mobiles for the GSM, the standard that is adopted by the European telecom Standard Institute European Telecommunication Standard Institute or what is popularly known as ETSI E T S I; E T S I specifies a bit rate in the range of 5.6 kilobits per second to 13 kilobits per second.

in fact 13 kilobits per second is the one that you are getting at full rate and 5.6 kilobits per second is what you are getting at half rate and there the delay is of the order of 40 milliseconds so definitely we have to accept this higher delay because for mobiles even this 13 kbps of bit rate is on a very high side.

For the CDMA, the standard that the North American North American standards which are existing that is by the telecom industries association Telecom Industries Association or Electronic Industries Association in short form called as TIA or EIA so they had jointly formulated the standards TIA or EIA standards and that essentially specifies a bit rate of 0.8 kbps so it is really low at the lower end up to 8.5 kilobits per seconds at the higher end and the delay that we are talking of for the CDMAs the Code Division Multiple Access that is 45 milliseconds considerably high both for GSM as well as for CDMA and also for very low bit rate codecs what ITUT has proposed.

(Refer Slide Time: 57:20)



So in a nutshell what we learnt is that speech coding has got a few fundamental requirements and the essential point is that whatever speech output that we produce must adhere to certain standards that has been stipulated by ITUT, ETSI and other standard bodies. And we will be in the next lectures we will go in to the basics of the codecs design. We will start with the approximating waveform approximating codecs and we will understand that how the quantization is going to affect the quality the quantization noise, the corresponding signal to noise ratios and how we improve upon that, so that will be our topic of discussion in the next few lectures, thank you.