Digital Voice and Picture Communication

Prof. S. Sengupta

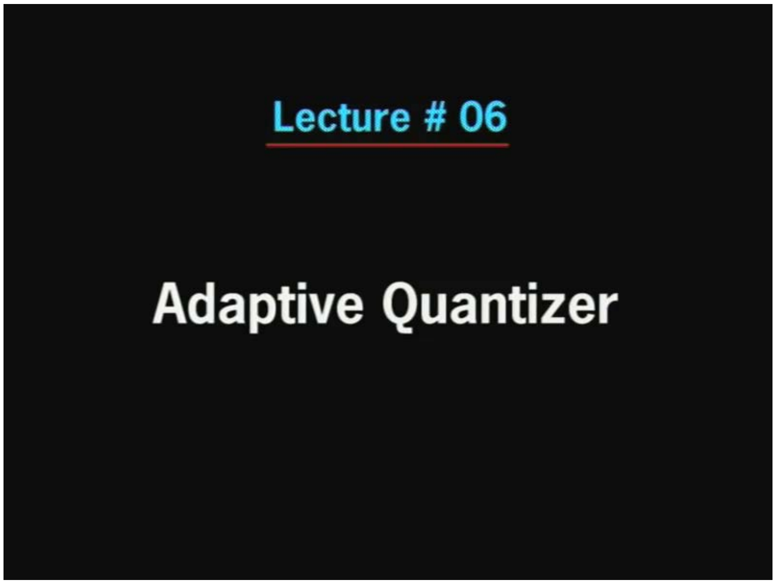
Department of Electronics and Communication Engineering

Indian Institute of Technology, Kharagpur

Lecture - 06

Adaptive Quantizer

(Refer Slide Time: 00:00:44 min)



In the last class we were discussing about the optimum quantizers and also we talked about the nonlinear quantizer in the form of the mu law. Now what essentially we had seen is that the non-uniform quantizer was less sensitive to the variance of the signal; to some extent it was but for a relatively flat zone we had seen that the signal to noise ratio was more or less constant; so that was so that means to say that it was independent more or less on the sigma x parameter that is to say the variance of the signal. And of course although it was uniform, although the SNR was uniform that is what we had shown in some of the example graphs, I mean, it was also seen that the SNR value that the overall SNR value was much lower as compared to what it would have been if we had used the uniform quantizer. But beyond a certain stage we found that the performance of the mu law was better than the uniform quantizer especially wherever we are having the ratio of sigma x by x max to be on the higher side. These details we had seen in the last class. Then we had also put forward the theory behind the optimal quantizers and there what we had said is that the quantizer parameters that means to say the decision levels and the reconstruction levels these are to be designed in accordance with the signal variance. But in the process we had also seen that especially for the speech where we have got very large variations in amplitude, even for one particular speaker under the different conditions of communication one can have different amplitude levels and that gives rise to a very large dynamic range of the values which the speech signal can attain.

And another characteristic of the speech signal is that it is not a stationery process so it varies from frame to frame and another matter of concern is that we have got speech verses silence. Therefore, considering all these considerations we had argued towards the end of the last class that it is not the right kind of solution to use the optimum quantizer although optimum quantizer when matched to the signal variance would give us a much better SNR as compared to the mu law. but mu law usage of mu law is preferred for the speech signals under different conditions.

Now today we are going to study that in order to make the quantizer getting itself adapted to the variance in the signal there is yet another strategy that is what we are going to discuss and today's topic of discussion is going to be what is called as the adaptive quantization.

As the name implies, an adaptive quantizer would mean those kinds of quantizers where the quantization parameters would be dynamically adjusted in accordance with the signal variance. Means its parameters should match according to the variation in the signal. Now what are the parameters that we can change? Now one of that is of course the step size that is the delta.

Now, so long we were assuming that the delta was a constant quantity. I mean, when we take that uniform quantizer it was absolutely equal; for all the different levels we were having a having equal step size. But in the case of the non-uniform although the step sizes are unequal, essentially we have got delta 1 delta 2 etc for the different levels but in such a case also all those different step sizes that we considered, those were also constant for a particular design of the

quantizer. Means if we consider any particular value of mu in the mu law quantizer, in that case the step sizes are essentially predesigned. But in the case of adaptive quantization the step sizes will be made to vary in accordance with the signal. So what would you expect? If the signal variance increases, if sigma x happens to increase in that case what should we logically do to the value of the delta, the step size, increase or decrease?
Yes, if sigma x increases we should also increase delta. Because essentially in the process of quantization now we can do some form of over quantization is possible because the signal level is up, the variance of the signal is higher so we can use larger value of delta.

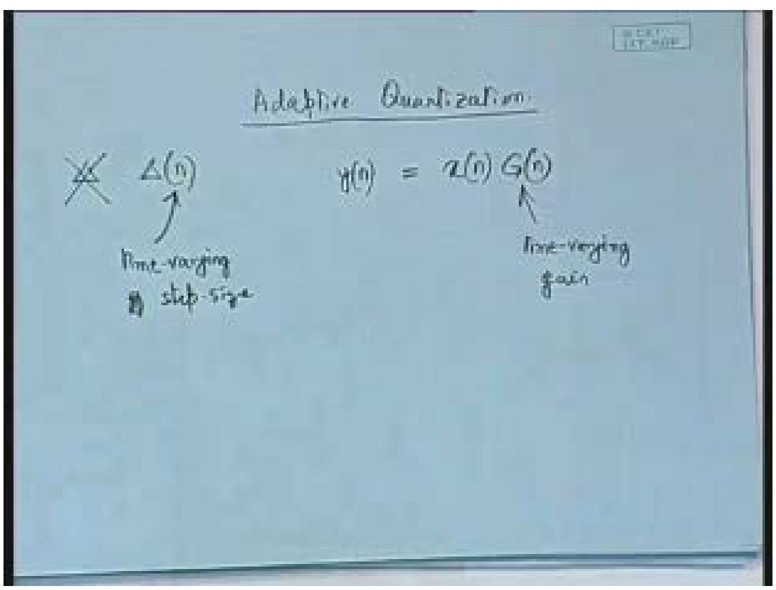
On the other hand, when we have the signal variance as smaller in that case we are going to have the delta value also as smaller. So essentially what we want is that with such a variation in the signal amplitude of the speech signal it should be changing this delta should be changing from time to time. So now delta we are not assuming as a constant delta rather we are going to say that this delta would be parameterized by n where n is going to be a parameter corresponding to the sequence. Now, delta instead of writing delta we are going to write delta as a function of n.

Now whether we will be sampling this new delta for every samples or whether we are going to sample it at a much lower frequency as compared to the sequence that means to say that the rate at which the sampling is performed, these aspects we are going to discuss in course of time. But essentially it will not be the delta anymore, the fixed delta anymore but it will be the delta as a parameter of n. now this is one way of achieving the adaptive quantization.

Yet another methodology to have an adaptive quantizer is that before you take the before you perform the quantization you multiply the signal by some gain; you just introduce a gain corresponding to the signal and that gain is something which is not a constant gain but that gain will change according to the variance of the signal. Therefore, what we want is that we will not be actually quantizing the original signal x of n, but instead we will be quantizing y of n where y of n will be given by x of n multiplied by a gain parameter G which also will be dependent upon n. So this G will be a time varying gain, this will be a time varying gain, like the way delta(n) is essentially a time varying step size or an adaptive step size. G(n) is going to be time varying gain. That will be a second approach in which case before quantizing the signal we will multiplying it by the time varying gain so we get y of n and it is the y of n which we will be quantizing and then encoding. So, at the decoder end what we have to do we have to do the reverse process. That means to say that when we recover y cap of n at the decoder there we have to divide it by G of n so as to get back the x cap of n.

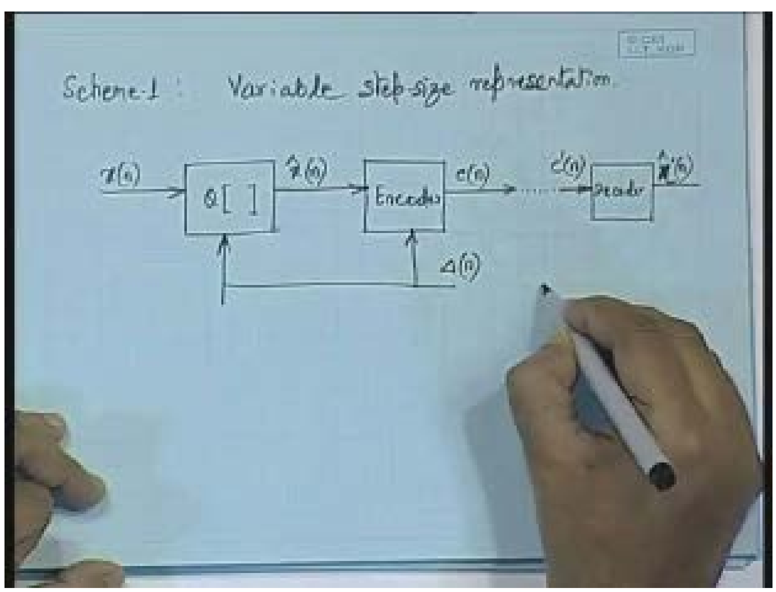
We will see the blocks of these two these two adaptive quantization schemes. Now what I said is that this delta(n) (Refer Slide Time: 10:24) should be proportional to the signal variance. Now what should this gain be? Should it be proportional to the signal variance? Directly proportional to the signal variance? It should be inversely proportional to the signal variance. Why? Because, as the signal variance changes this x(n) that changes and we have to compensate. So we have to compensate for this increase by reducing G of n. thus, G of n should have the variance in hexa reciprocal. So, as a result of that what we will try to do is that this product x(n) G(n) that should be maintained as independent as possible with sigma x, that is the basic idea that it should adapt itself to the variance of the signal.

(Refer Slide Time: 11:22)



We will see in a block diagram form that how these two quantizers, how these two adaptive quantization scheme would look like when we have the variable step size representation. This is the scheme 1 which is the variable step size representation. In this case as we said that x of n is the input so x of n will be passed through the quantizer which I am denoting by this block Q. So at the output of the quantizer we are going to write it as x cap of n which is the quantized signal and then this will be encoded. So we will put an encoder and after that we will be having the actual encoded sequence which will be formed. So encoded sequence will denoted by c of n. So c of n is the encoded sequence and this quantizer (Refer Slide Time: 12:41) as well as the encoder this will be controlled by the step size. Therefore, the step size will now be written as delta(n) as I said. So this will be on the encoder side and on the decoder side we should be receiving ideally the same c(n). But you know that in between this encoder and the decoder we are going to have the communication channel. The communication channel may introduce some amount of noise. We cannot all the time assume that............ although in many of the analysis we simple make an ideal assumption that the channel is a lossless channel but practical channels we know are not lossless. Thus, if we have to write it in a proper way we should not be writing it as c(n) but we should be writing it as c prime n where c prime n will be equal to c(n) only under the ideal lossless channel situations.

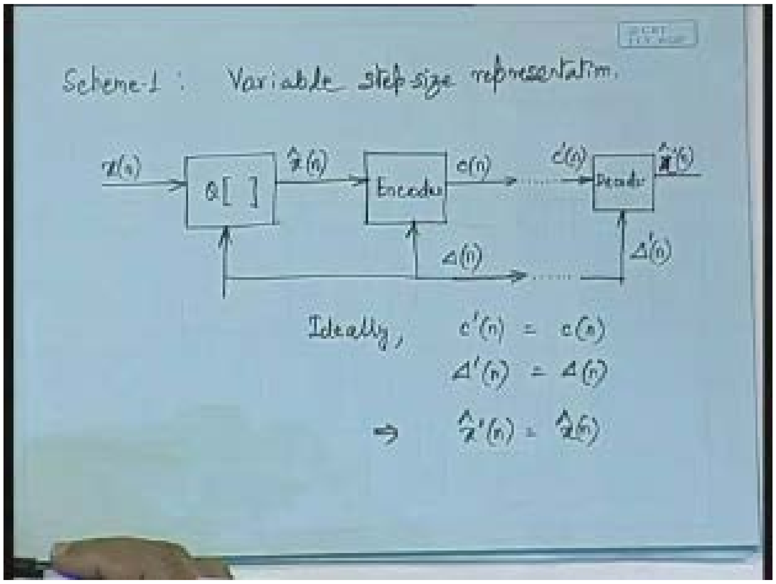
(Refer Slide Time: 14:14)



So this c prime n will pass through the reverse block of this. that means to say that it will go through the decoder and the output of the decoder will be y prime cap n and no not not y in this case we are deal dealing with the x only so this we will call as x prime n and this decoder its parameter will be controlled by delta(n) is it not? But just look at one thing that in this case this delta is a time varying quantity. Now, in the earlier case where delta was fixed then delta is well known to the encoder and the decoder beforehand because you are going to use a particular quantization scheme either a uniform quantization scheme or a non-uniform quantization scheme with a particular value of mu, so everything all the step sizes are known before end to the encoder and decoder so there was no need to transmit that information in the channel.

But, whereas in this case; because delta is varying with n, we have to update the decoder with whatever value of delta the encoder is using. That is why this value of delta(n) also should be sent into the channel. It should also be encoded and send to the channel. So we will be sending delta(n) and here ideally we should be receiving at the decoder end delta(n) only but again considering the non-ideal situations of the channel let us not write it as delta n, let us write it as delta prime n. So ideally, for this we should have c prime n is equal to c(n) for lossless; delta prime n should be equal to delta(n) and in that case only, we can expect to get that x cap prime of n is going to be x cap of n. We certainly cannot get x of n exactly because there is already a quantization process which is there. But x prime of n can be made equal to x cap of n if the channel happens to be noiseless.

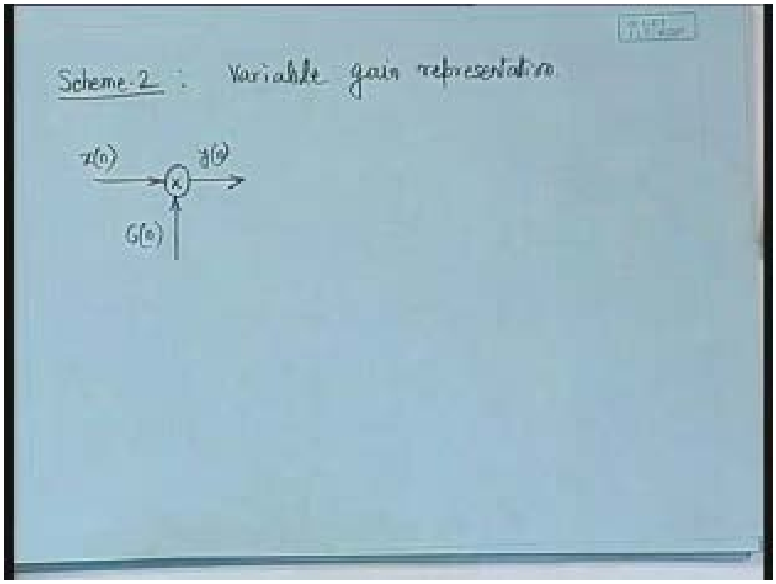
(Refer Slide Time: 16:23)



Now what we have represented about this delta(n) is okay as a scheme. But we have to definitely study that how will this delta(n) be decided. Because we said that this delta(n), because this because of this dependence upon n it is going to be adaptive. So it must somehow know that how the signal variance is changing. It should monitor the signal. How it monitors the signal; that aspect we will come to shortly. This is about the scheme 1 and let us have a look at the scheme 2 where instead of providing a time varying step size, we are going to provide a time varying gain. Therefore, the scheme 2 would like this.

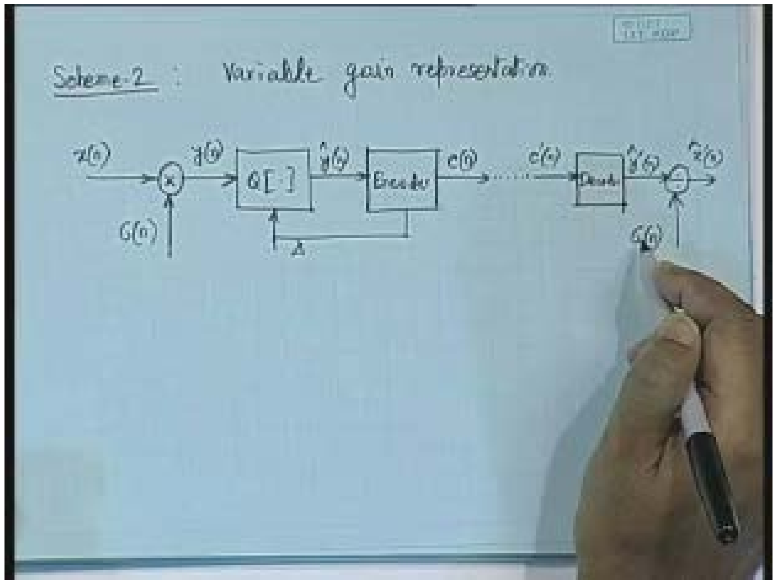
This is the variable gain representation. In this case the input is x(n) and we have a block where a multiplication will be performed and this multiplication will be by G of n. Just look at this part that we are saying y(n) is x(n) into G(n) (Refer Slide Time: 18:03). So this is x(n) and the G(n) is the time varying gain. How we derive it, that we will see and the output of this will be the product of x(n) G(n) which is which we are calling as y(n).

(Refer Slide Time: 18:18)



So now, instead of quantizing x(n) which we did in the case of the variable step size (Refer Slide Time: 18:22) where x(n) was directly quantized, in this case y(n) will be quantized. So this y(n) will be put through the quantizer, the Q block and the output of this will be y cap of n the quantized y and then we will pass it through the encoder, the encoder generates the encoded sequence c(n). Now in this case the quantizer and the encoder should be controlled by the quantization parameter delta. But in this case delta is fixed so we need not have to transmit the delta into the channel. And c of n goes into the channel and c of n gets decoded, I mean, at the input of the decoder we will call it as c prime n as before and then it goes to the decoder so the decoder output we are going to write as y prime cap n because it is the y cap of n which we are going to recover; exactly y cap n if the channel is lossless but otherwise it will be in general y cap n and this y cap n will be put through a divider block where this y prime cap n will be divided by the gain G of n, the time varying gain G of n, and we should be getting x prime n cap. This is what we can expect to get. So x(n) here x prime cap n over here.

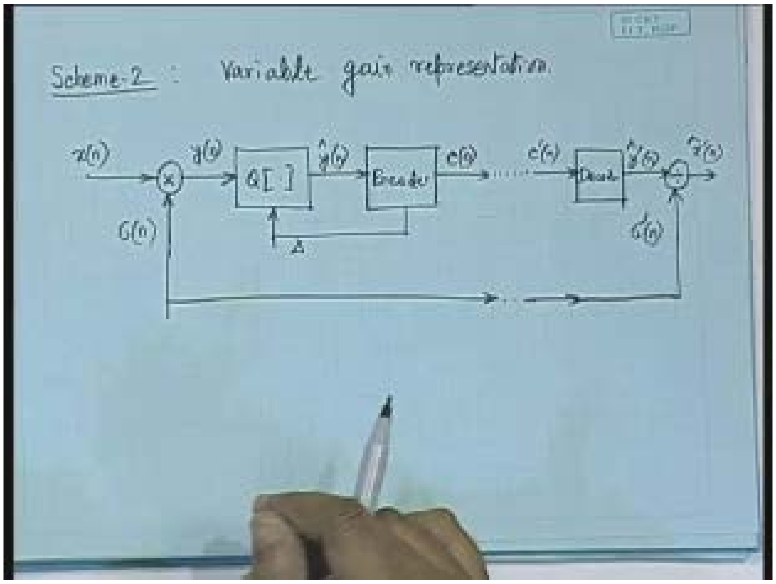
(Refer Slide Time: 20:15)



Now, this time varying gain what we are showing over here, this time varying gain must be known to the decoder also. the encoder is using the time varying gain to multiply x of n, so likewise the decoder should know the same G of n in order to divide y prime cap of n. So this G of n information should be sent in to the channel. So here we must send G of n (Refer Slide Time: 20:49) and here we will be receiving what we will write as G prime n. So it will be y prime cap n divided by G prime n which gives rise to x prime cap n. So this is going to be the reconstructed or the decoded signal.

Hence, under the noiseless channel case, x prime cap n is going to be x prime n. Now we do not have any x prime n directly in this block diagram because we are not quantizing x(n) rather we are quantizing y of n. But x prime n will be the equivalent quantized representation of the signal x of n. So in this case what we have to do is to transmit both c(n) and G(n). In the earlier case we were transmitting c(n) and delta(n) but in variable gain representation the c(n) and G(n) that is what we have to send.

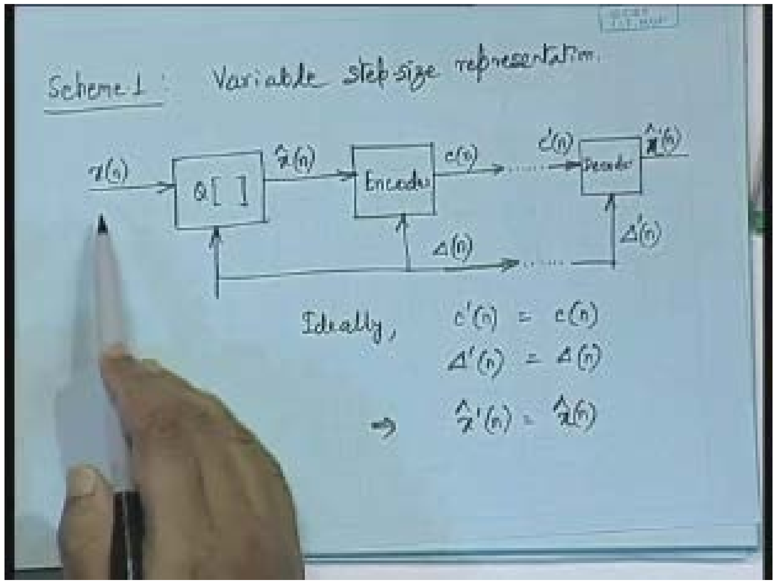
(Refer Slide Time: 22:02)



Now we will treat both these schemes simultaneously because in whatever methodology we can extract the time varying properties of the signal, we can use it either to control the adaptive step size or to control the adaptive gain. In principle both are going to be same so that is why we are treating them simultaneously. Now, one of the aspects that we can think over is that to look at scheme 1 or look at scheme 2. What are we going to, I mean, what is our requirement? We must formulate this delta(n) properly depending upon the signal. We have to monitor the signal and based on that we must decide that what should be the value of this delta n. now where we can monitor the signal.

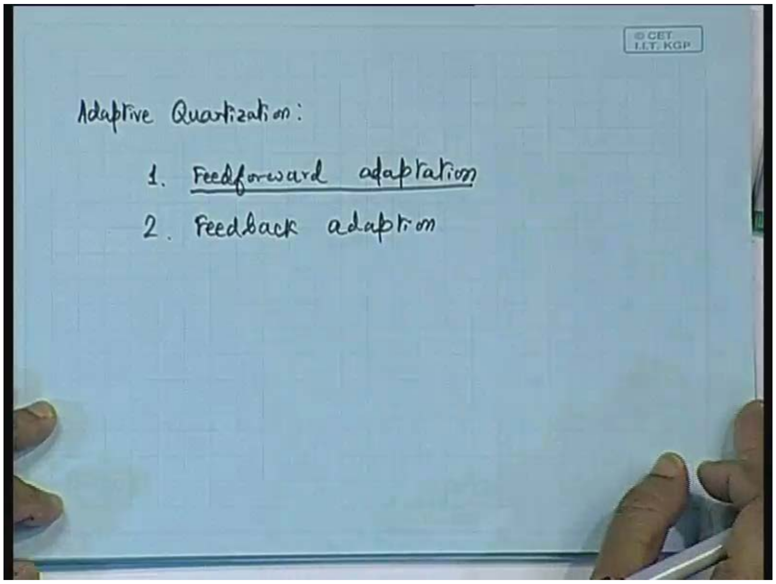
Now we have got if you look at the block diagram over here this is the input part of the signal (Refer Slide Time: 23:13) and this c(n), c(n) is also carrying the signal, but only thing is that c(n) is the coded form of the signal.

(Refer Slide Time: 23:31)



So we can either derive the parameters delta(n) from the original signal x of n or we can derive delta(n) by looking at the encoded sequence c of n. In case we are estimating delta(n) from x(n) directly in that case we must have a block which computes the adaptive step size in the forward loop. Whereas if we are going to compute delta(n) by looking at c(n) that is to say the encoded sequence in that case we must have an adaptive quantizer computation loop which will be in the feedback path of the encoder. So there are two schemes, there are two methodologies for achieving adaptive quantization and one of the methodology is to use a feedforward adaptation and the second is what is called as the feedback adaptation.

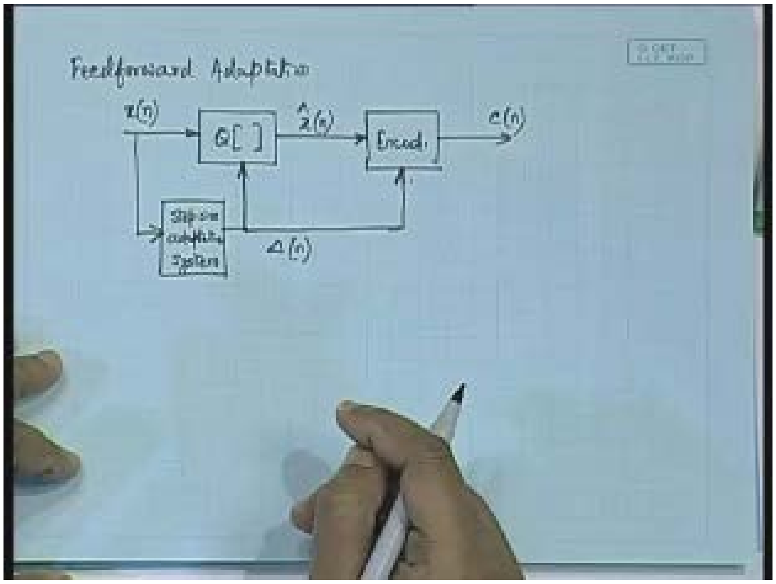
(Refer Slide Time: 00:24:36 min)



So adaptive adaptive quantization, we can say that, from the adaptation point of view one can have either a feedforward adaptation or it could be a feedback adaptation. The feedforward or the feedback this methodology should be possible for both variable step sizes as well as for variable gain. Because, if we take the feedforward; the feedforward mechanism could be used for both. So let us see the step size adaptation.

So let us first discuss the feedforward adaptation. Thus, in feed forward adaptation what we need to do is that the block diagram; I mean, if we have understood the earlier block diagram which is a general block diagram where we did not talk about how the adaptation is going to take place but in this case specifically for a feedforward adaptation, so this is going to be the feedforward adaptation x(n) goes through Q, output is x cap n and that goes to the encoder this goes to c(n) up to this we do not have any change, only thing is that here we are going to have delta(n) that also we had seen in the variable step size representation. But the only new addition will be that this delta(n) will be decided by a block which will be in the feedforward which basically we say which is a step size adaptation system.

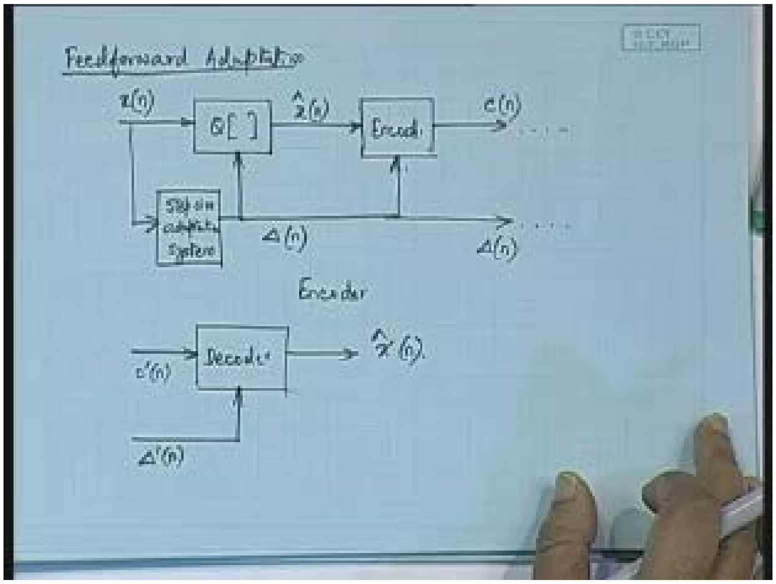
(Refer Slide Time: 26:57)



Now what is this adaptation system?

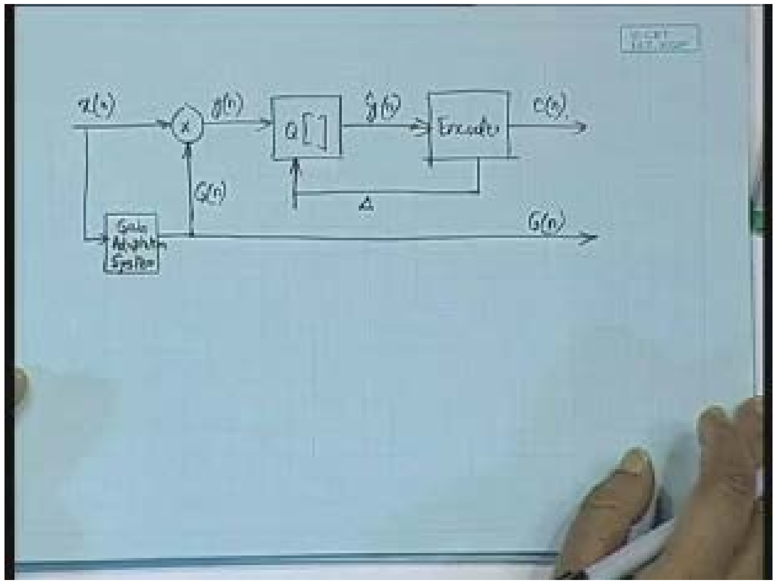
Definitely we understand it by now that the step size adaption system is going to compute the compute the variance of the signal. So it is the variance signal computation which is going to take place in this block diagram that also we can say very definitely. So this block comes in the feedforward loop and this delta(n) even in this scheme also we need to transmit delta(n) as before. So the encoder feedforward adaption encoder will carry both c(n) and delta(n) into the channel and correspondingly at the decoder there is no change; decoder will be getting c prime n and there will be a decoder block over here and then we are going to have delta prime n and then the output of this will be x prime cap of n. So this is the overall feedforward adaptation scheme and this is for the step size adaption.

(Refer Slide Time: 28:18)



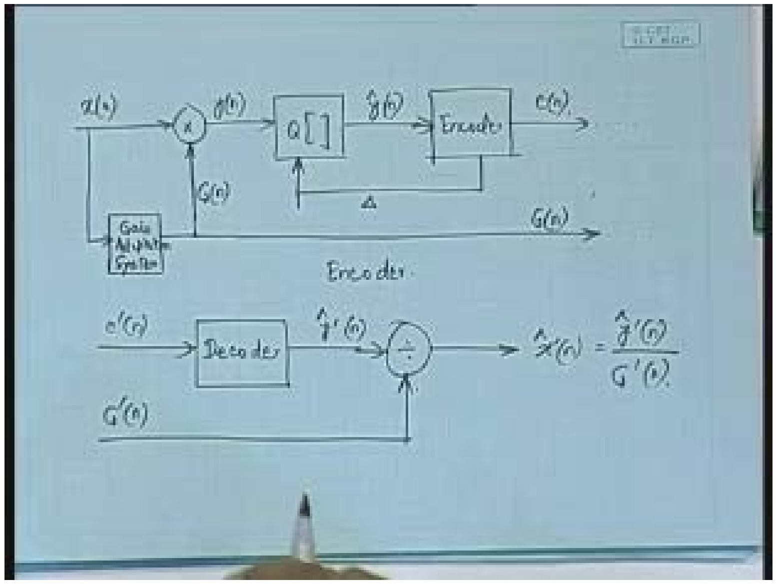
Now if we see the gain adaptation system that also will look somewhat similar. So what we need to do is that, for the gain adaptation x of n will be here and here we have the multiplier and this is the y(n), then we are going to have Q y cap n at the output, the encoder and output of the encoder is c of n and the quantizer and the encoder has got the parameter delta which in this case is a constant parameter delta that is what we are going to have and now what we have to do is to derive this G(n) from x(n). Again we are going to adapt the same feedforward mechanism. Therefore, in the feedforward mechanism we will feed this x(n) the input directly and then that goes through what is called as the gain adaptation system and the gain adaption system block that goes to the G(n) and this G(n) needs to be transmitted.

(Refer Slide Time: 29:50)



Now G(n) also should be going from the output of the encoder so this will be the encoder overall block diagram and the decoder will be, this c prime n will be received, it goes through the decoder and as before we are going to have y prime cap of n and there will be a divider block and it will be divided by instead of G(n) we write G prime n and this will be x prime cap n which will be essentially y prime cap n divided by G prime of n. So this is the feedforward adaptation only but based on the gain adaptation. Earlier was the step size adaptation, in this case it is the gain adaptation.

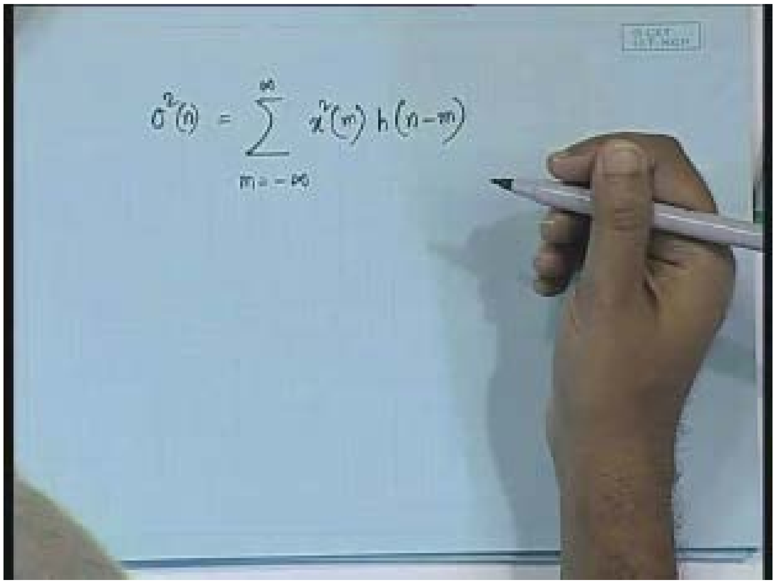
(Refer Slide Time: 30:53)



Now what we have to do is to know about this step size adaption or this gain adaptation that how exactly we do it. Now as we said that one has to estimate the signal's variance, now the variance what we have seen is that, one of the approaches is that we can estimate the variance by looking at the short time energy. The signal is a zero mean signal that is what we are assuming. So the short time energy what we are having is giving us a measure of the variance and how we are going to express that?

The short time energy, we are going to write it as sigma square n which will be the summation of x square m into the filter function the low pass filter function h of n minus m and we are going to sum it up for m is equal to minus infinity to plus infinity, this is an infinite summation but the truncation will be definitely obtained by this h of n minus m function which is typically an FIR filter or even if it is an IIR filter it should be a realizable IIR filter.

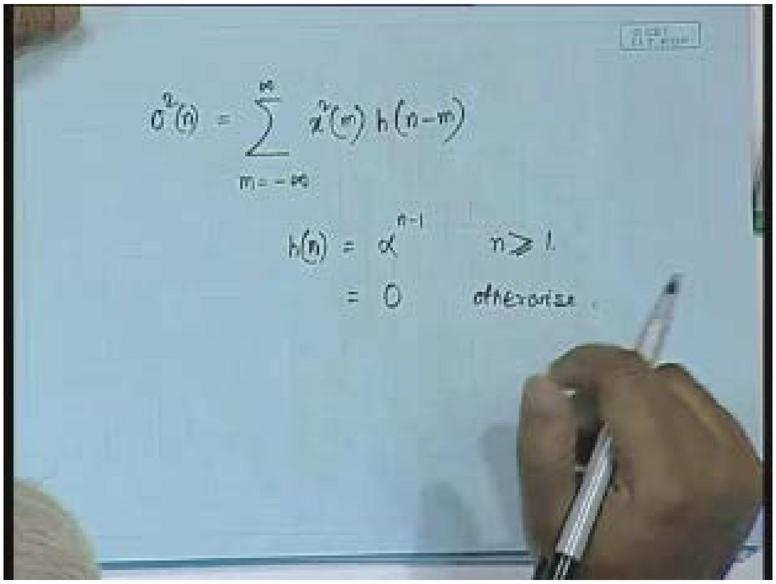
(Refer Slide Time: 32:34)



Now, one of the simple examples that we can take for this is that, we can take h of n to be alpha to the power n minus 1 for n greater than or equal to 1 and this we can say that is 0 otherwise that means to say that for n equal to 0 or any negative values of n rather n less than unity, we are going to have the value of h(n) to be equal to 0. This function we had seen. In earlier examples also we were considering these functions.

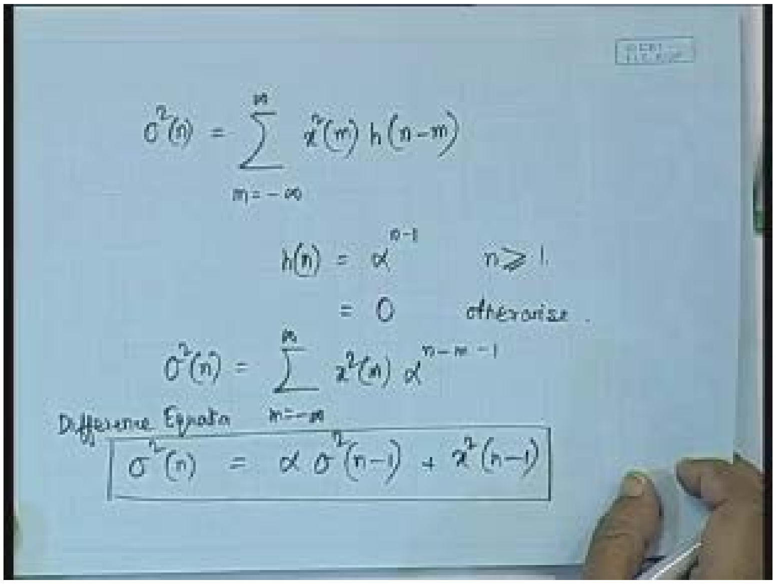
Now, for stability of course we are going to have the parameter alpha lying between 0 and 1. So using this we can write down the sigma square n as the summation m is equal to minus infinity to plus infinity summation x square m alpha to the power n minus m minus 1 and this we can also write down in the form of a difference equation.

(Refer Slide Time: 33:56)



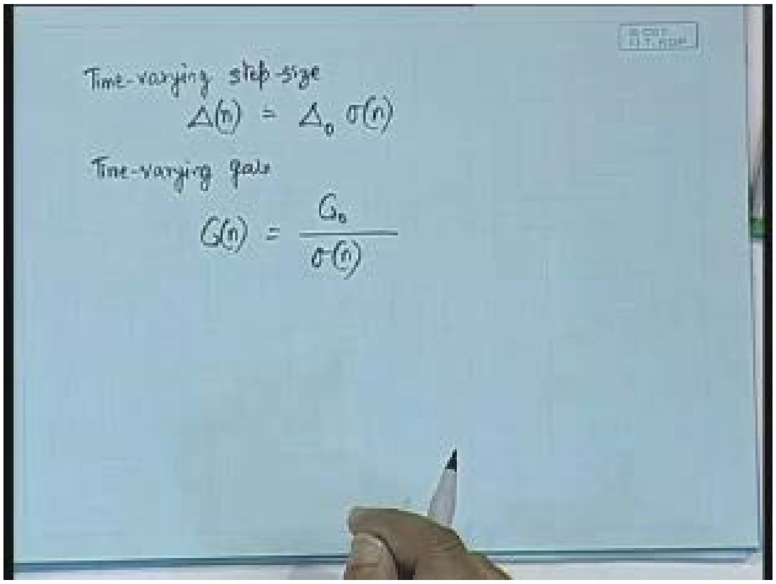
Therefore, in the form of difference equation, we can write it as sigma square n is equal to alpha into sigma square n minus 1 plus x square n minus 1. So this is going to be quite a useful expression for us that we know the earlier estimate, the previous estimate of the signal variance multiply it by this parameter alpha then we are going add up this x square n minus 1th term that means to say the new sample we are adding and that gives us the new estimate of sigma square n. So it could be used. This is a recursive way of computing sigma square n. So, if we get an updated updated estimate of this sigma square n then accordingly we can vary the step size. In fact the step size is made proportional to the standard deviation of the signals.

(Refer Slide Time: 34:59)



Hence, what we are going to write is that delta(n) will be equal to some constant delta(0) into sigma of n; sigma of n is something which will be estimated given this equation. So this is about the time varying step size, this is the time varying step size (Refer Slide Time: 35:34) and the time varying gain will be similar. Only thing is that here it will not be multiplied by sigma(n) but it will be divided by the standard deviation sigma(n). So time varying gain is given as G of n which will be written as some constant gain G(0) upon sigma n.

(Refer Slide Time: 36:06)



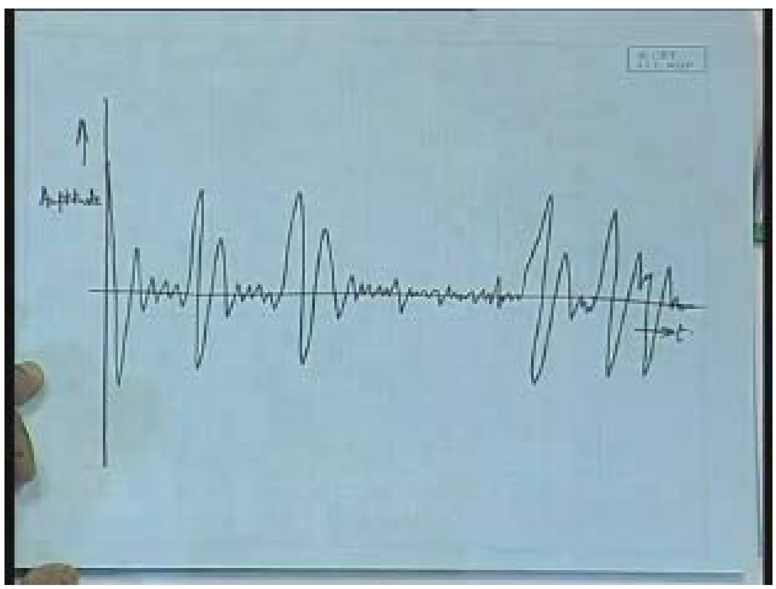
So now there are two issues, that is what we have to decide that how would you we how often are we going to change this sigma(n). Are we going to change it too fast or are we going to change it too slow?

I think that for short time signal analysis, for short time speed signal analysis we have been encountering this question again and again. Now intuitively it obviously appears that if we are going for too much of short time in the computation of sigma(n) in that case as the normal speech parameters vary that means to say that with the harmonics that the vocal cord oscillations are giving that also will be picked up by the sigma(n) variations and then accordingly we will be changing the step sizes for everything and that means to say that although we want those variations to be there but we will try to have some kind of an equalization effect. We will try to give equal treatment to all and even to the value of the sigma(n) at the non-speech sequence that also we will abruptly vary.

On the other hand, we should not think of....... see, after all this sigma(n) how we have derived? Sigma(n) we have derived through a process of low pass filtering. This h of n minus m this is giving an effect of low pass filtering. So definitely this low pass filter is going to have some upper cutoff frequency and this upper cutoff frequency and because this upper cutoff frequency is there, there is no point in trying to have a sigma(n) which will be sampled at a rate which is considerably higher than that of the upper cutoff frequency of the low pass filter.

In fact I mean, sigma(n) happens to be a much slower process as compared to the speech sampling. Speech sampling is going on at 8 kHz or 10 kHz rate but that does not mean that sigma(n) will track accordingly. Just a very simple example that if we take a speech waveform something like this say a typical speech waveform let us say that this is the amplitude this axis we are plotting the amplitude of the speech signal and this axis is as it is the time axis (Refer Slide Time: 39:17). So we have something like this.

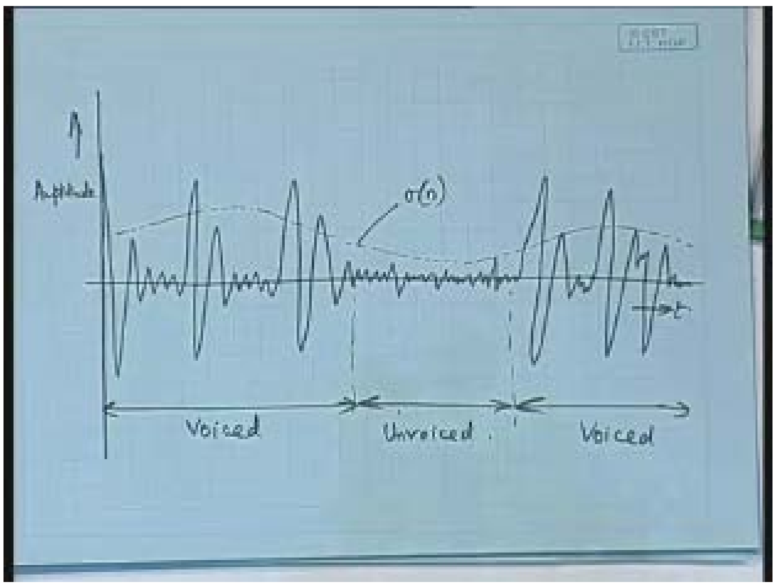
(Refer Slide Time: 39:42)



What we tried to show over here in this plot is that (Refer Slide Time: 39:52) this part we can say is the voiced speech, this duration is the unvoiced speech. Again this part happens to be the voiced. So this is voiced, this is also voiced and this is unvoiced. Now this is how the signal amplitude is varying or rather instead of signal amplitude we can talk about the signal energy also.

Now what happens is that the sigma(n) estimate sigma(n) estimate will not exactly track down according to this, it will not track down this variation but the sigma(n) variation will be relatively slower. So may be that here we are having a situation as something like this. This is how the sigma(n) varies. So sigma(n) variation you can see that it is a smooth process. It is a low pass filter version so it is a smoother process. In fact again look at the difference equation.

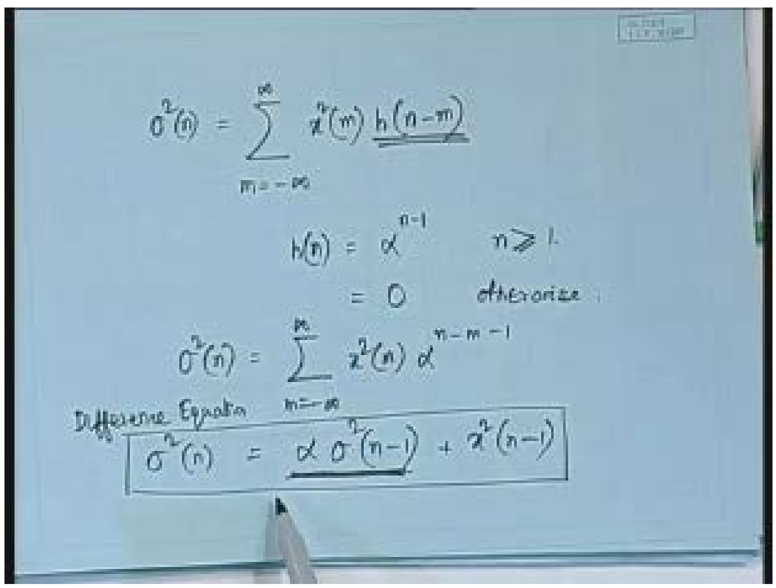
(Refer Slide Time: 41:17)



The difference equation estimates this sigma(n) according to this. So there is a parameter alpha which will be associated. Now just qualitatively think that what will be the effect if alpha is high. High means okay, for stability we have to make alpha restricted to 1. Supposing it is very close to 1 say it is equal to 0.99 that is one type of case that is what we can say and the other case is that where alpha will take will take to be much smaller.

Let us say that we take two situations: alpha is equal to 0.99 and another case we take where alpha is lower than that alpha is 0.9. So looking at the difference equation how will these two values of alpha make a difference. You see, there are two terms in this sigma square n computation. One is alpha sigma square n minus 1 and the other is the addition with x square n minus 1. Now if you make this quantity larger; by having a larger value of alpha means that you are inviting more contributions of this. That means to say that the past variance, past estimate of variance is going to have more effect on the present estimate of variance which means to say that it does not allow us to make a revised estimate of sigma(n) that fast. Whereas if we make alpha smaller in that case the contribution of this alpha sigma square n minus 1 term will be lower so the signal variation which will be reflected by this x square n minus 1 that will influence the sigma square n more rapidly. So now, with lesser value of alpha there will be more rapid variation to the signal variance whereas with higher value of alpha the variation in sigma square n will be much smaller.

(Refer Slide Time: 43:41)



Hence, as a result of that, in a typical situation one would see that if we just think of a curve like this (Refer Slide Time: 43:53) for the amplitude versus time and this is this is what we are plotting for x of n. Now let us say that if we have a gain adaptive system, if we follow the gain adaptation; gain adaptation means that we are going to multiply this x of n by G of n. So if sigma(n) changes very rapidly, sigma(n) changing very rapidly would obviously mean that this x of n G(n) or rather the y(n) y(n) will have less sensitivity to the signal variance. So as a result of that this variation which you are having all these variations will be properly captured by this sigma(n) but it also takes care, I mean, it will it will try to make more droops in between okay.

the revised sigma estimate may be something like this that here a fall then here again because of this there is a rise, then there is a fall, then again there is a rise, then there is a fall like this (Refer Slide Time: 45:13) so the sigma's variation will not be a smoother variation the sigma variation will be much rapid as a result of that and because of that we will be having a more uniform....................

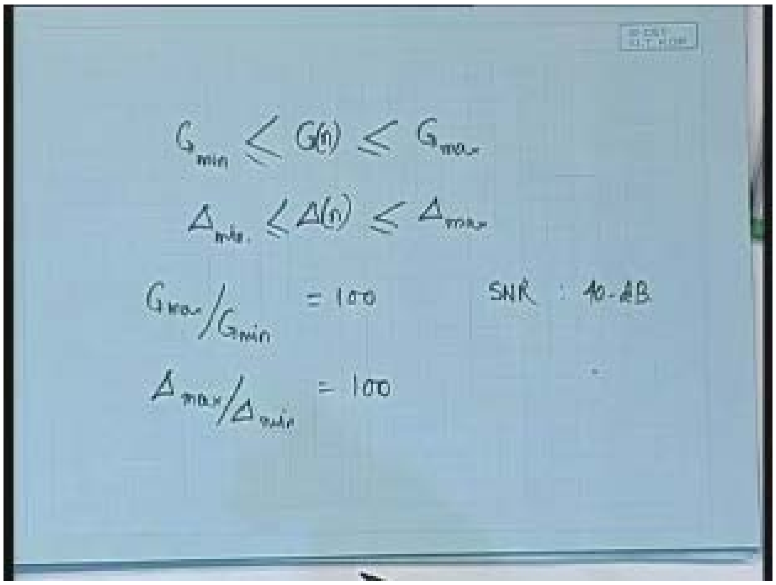
Now here the sigma(n) varies very much so here you see that sigma(n) falls and sigma(n) falls more rapidly means that it will give rise to higher value of G(n), so higher value of G(n) means that these parts will be more emphasized. So when we see the y(n) waveform we will see that these parts will be boosted and again these parts will not be boosted as much. So as a result of that the amplitude will try to have more amount of uniformity, I mean less dependence on variance which is not always desirable because it should actually capture the properties of the speech waveform.

Therefore, as result of that it is a good practice to have the sigma(n) to be as low a frequency sampling as possible without of course this term being the short time nature of the signal because if we make it too much low pass filtered, if the pass filtering effect is highly pronounced in that case the difficulty that we will be encountering is that from one frame to the other the variation of that sigma(n) will not be that much so sigma(n) will be unable to track down the variation of the signal at all which is also not very desirable. So definitely there is some amount of compromise.

Another thing is that this G(n) and sigma(n) as we are seeing over here that for speech waveforms this sigma this delta(n) and G(n) whatever adaptation we have, the step size adaptation and the gain adaptation, it is going to be performed over a wide range. But for practical implementation that range needs to be limited.

Now, when we talk about the gain, the gain should be restricted within some practical bounds. So we can say that it is restricted within G min the minimum gain and G max the maximum gain and likewise the step size delta(n) should be restricted within delta max and delta min and typically this ratio of these limits, it is seen that, whether we call of G max by G min; in all typical applications G max by G min or the delta max by delta min this is of the order of 100, 100 means that it contributes to an SNR of 40 dB. So there could be a variation in gain as much as 40 dB and then likewise there could be a variation in the step size by as much as 40 dB.

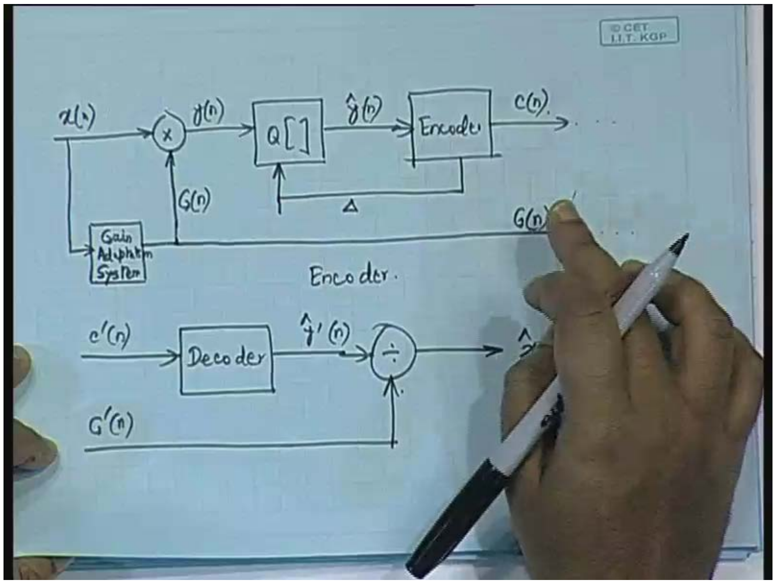
(Refer Slide Time: 48:43)



This is from the practical implementation consideration. We have seen that, of course this is the kind of adaptation that one should do. Means the delta(n) and G(n) that should be derived from the signal properties. Now here we made use of the x(n) directly in this equation (Refer Slide Time: 49:16) that sigma square n is derived directly from x square m.

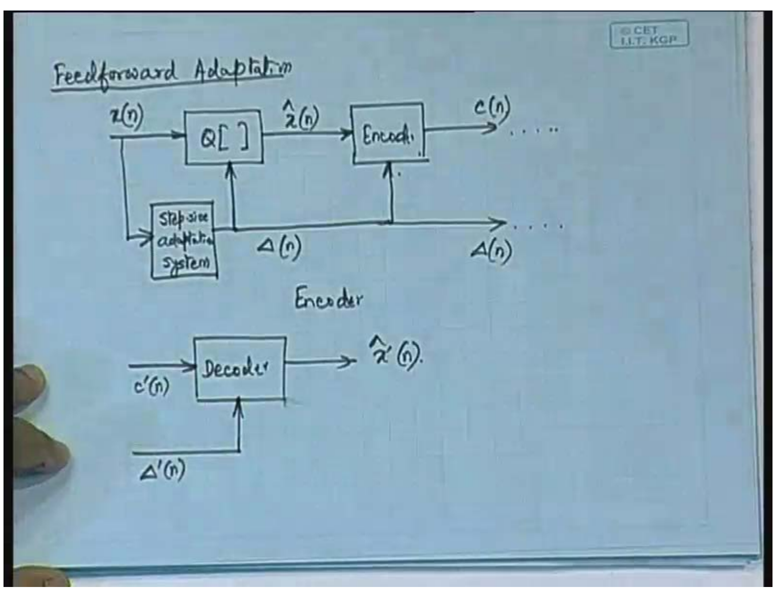
Now it could also be derived in a feedback mechanism also that is what I mentioned little earlier and in that case it leads to the feedback adaptation system. Now the feedback adaptation system I will be discussing in the next class, I will begin with a feedback adaptation system and there we will see we will notice one thing that............. just look at this diagram where we are having a step size adaptation or in this case the gain adaptation.

(Refer Slide Time: 00:49:58 min)



We need to transmit c(n) as well as G(n) for gain adaptation.

(Refer Slide Time: 00:50:02 min)



And for step size adaptation we need to transmits c(n) and G(n). But supposing step size adaptation only we employ and instead of having the step size adaptation through x(n) we have the step size adaptation through c(n) because c(n) is also carrying the signal information in that case if the delta(n) information can be derived from c(n) in that case the encoder as well as the decoder can derive that information from c(n) only. So there will not be any need to transmit the information pertaining to delta(n) separately. That will be one of the advantages of the feedback adaptation mechanism but all advantages are every advantage is also associated with certain disadvantage. So we will also see that what are the disadvantage of the feedback adaptation.

In fact in a feedback adaptation not only can step size adaptation leads to the fact that only c(n) will be sufficient, even for gain adaptation also that if have the gain adaptation from this c(n) (Refer Slide Time: 51:21) then c(n) itself should be a sufficient information, we need not have to send the G(n) information into the channel, we can derive the G(n) information from c(n) itself at the decoder also. So we will see that in the next class, till then, thank you.