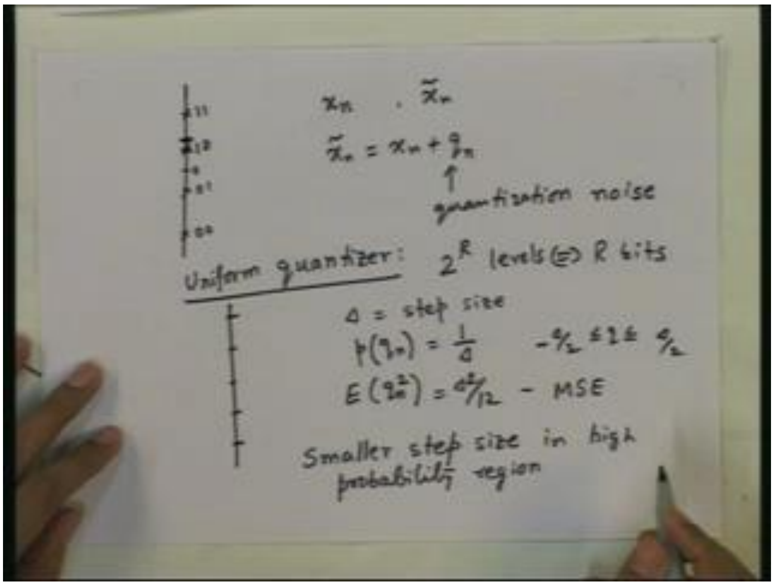
Digital Communication

Prof. Bikash Kumar Dey
Department of Electrical Engineering
Indian Institute of Technology, Bombay

Lecture - 03
Quantization, PCM and Delta Modulation

Hello everyone, today we will discuss quantization. In the last class we have discussed sampling which is an essential part of digitizing an analog signal. If you have an analog signal if you want to store or transmit the signal by digital means we need to represent a finite interval the signal, in any finite interval using finite number of bits. And, for doing that first of all we have to take only finite number of samples or values in any given finite amount of time and thus the need for digitizing in time that is sampling. And, now we are going to discuss the next essential part of digitizing an analog signal that is quantization where we discretize the signal in amplitude. So, the first of all we do sampling and then the samples are each sample is represented by finite number of bits. So, for doing that we have to select only finite number of levels in the amplitude and we cannot store or transmit all possible amplitude values.

(Refer Slide Time: 02:02)



So, let us say we have this is the amplitude scale and we have this is 0 and we choose a finite number of points from here that we will represent using bits. Let us say this point, this point, this point and this point these 4 points. So, whatever is the value of the sample; let us say something here we will choose the nearest level that we can quote. So, we will choose this point and store the corresponding bits. So, we can express this level by 0, 0 this by 0, 1 this by 1, 0 and this level by 1, 1. So, if you have this amplitude we will simply store or transmit 1, 0. So, by doing this obviously, there is a loss in terms of error.

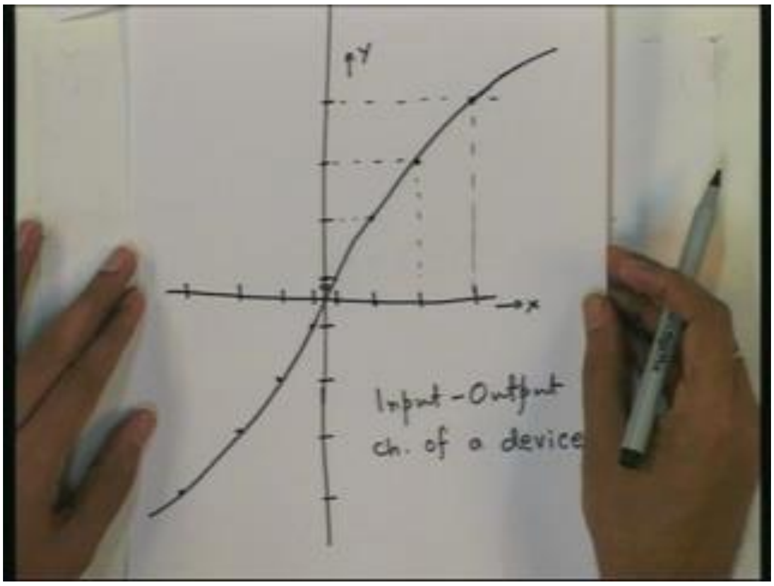
So, why is that? Because the actual value of the signal of the sample is this point whereas the receiver has no way of knowing what the actual value was; but it can only see this level so this is the error. So, if we denote the actual sample value at the n th time that is the x; if x n is the n th sample and x n tilde is the quantized value, quantized level for x n then x n tilde a is x n plus some quantization noise. So, this part is called quantization noise ok. Now, we can choose these levels in different ways; the most simple way to choose these levels is called uniform quantizer. If you choose an uniform quantizer in such way that we can, we want to store the sample values by say R number of bits; then, we can choose 2 power R numbers of levels.

So, for example here in this example we have taken 2 bits and so there are 2 powers 2 which is 4 there are 4 levels ok. So, if we have 2 powers R number of levels then we will have a certain delta. So, here the uniform quantizer basically means that the gap between the conjugative levels is fixed. So, if you that is the example taken here. So, the difference between the conjugative levels is uniform that is why the quantizer is called uniform. If delta is the step size of the quantization and if delta is small enough, and the probability density function of the sample is smooth enough. Then, we can see that the density function of the quantization noise q n will be almost uniform; if we of course, exclude the boundary regions.

Now, if we have this quantization noise density in the range of minus delta by 2 to delta by 2; then, what is the expected value of the energy of the noise that is delta square by 12. This can verified by computing using this uniform density function. So, this is the mean square error for a uniform quantizer. Now, we can also design non-uniform quantizer and often that is very useful. Because many practical sources have values which are dense, which have probability density function high in certain regions and low in certain other regions; because the density function often is not uniform. For example, speech signal is more likely to take small value than high values. So, for such a signal we need to quantize using smaller step size in the high probability regions and bigger step size in the low probability regions; because that is what will minimize the error the mean square error ok.

So, what is desired is smaller step size in high probability region. And, we will see a simple way of doing this quantizer, quantization using uniform quantizer itself.

(Refer Slide Time: 08:41)

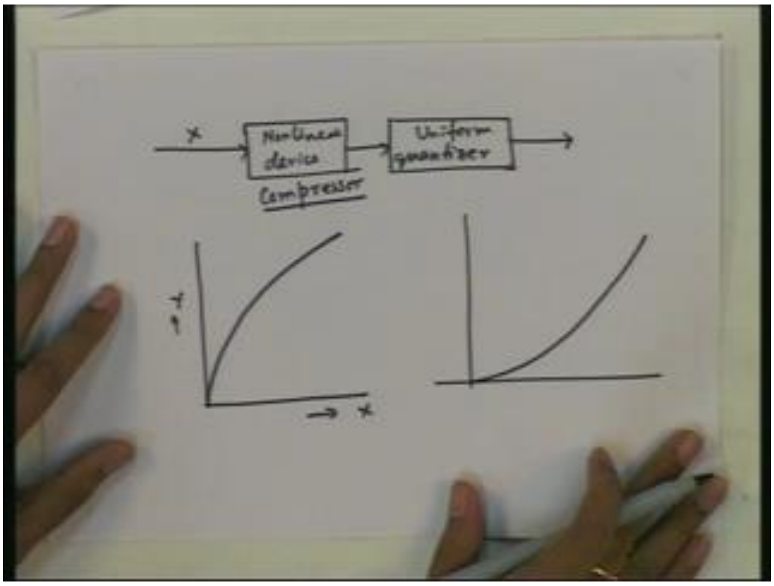


So, let us say we choose a certain set of levels for a particular source. Let us say we want to choose this is 0 and let us consider a signal like speech signal which takes small values with high probability and high values with small probability. Then, we want to quantize the small values with dense levels and high values with sparse levels. So, let the levels be like this these are the levels; so, we are considering here 8 levels. So, we can code the quantize signal using 3 bits and we can do this quantization in the following way. Let us consider a uniform quantizer on the other hand these are the levels at which we want to quantize the input signal; but let us also consider a uniform quantizer.

How will it quantize? It will quantize at a fixed step size; so, let the step size be like this let us take 8 such levels ok. So, the input signal takes values in this x axis; let us say x is the input sample value and we want to transform this x in such a way that this level, this value becomes this value. The first level on the x axis becomes the first level on the y axis. So, if we want to draw the characteristic of the device through which we will pass x that should transform this value to this value. So, this is the input, output characteristic of the device through which we will pass the sample value and then do uniform quantization.

So, it will transform this value to this value, then this value there is the second level to the second level of the uniform quantizer that is this is also a point on the transfer characteristic; similarly, we will get all the other points. So, if you join these points we get something like this and we will see that if you had more points it will go like this. So, this is the input, output characteristic of a device that we will use for doing this non uniform quantization. So, we have the central value x to be quantized we pass it through a non-linear device which we will convert this value x to a value y based on this characteristic. And, then we will do uniform quantization on y and transmit the coded bits accordingly. Now, the advantage here is that we do not have to the non uniform quantizer; because very simple to implement is fully for implementational point of view.

(Refer Slide Time: 13:08)

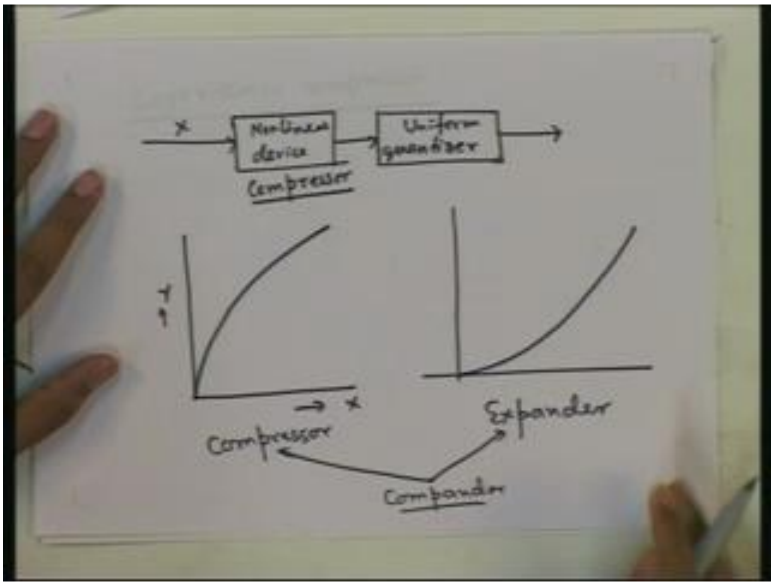


So, the block diagram of the quantizer will be like this that x is passed through a non-linear device. The device is obviously non-linear because this which is very clear from the plot of its characteristic; and then this is passed through a uniform quantizer. So, here we see that given any set of desired levels for a non uniform quantizer we can implement it using such a block diagram, using a non uniform quantizer, using a uniform quantizer and a suitable non-linear device. Now, if we observe this characteristic of the device we see that it actually compresses the signal it takes the signal and compresses the high values it amplifies the low values, small values of the signal and attenuates the high valued signal and then does uniform quantization.

So, this is called this device is called compressor device; because it compresses the input signal. And, then the how would we do the, how do we recover the original signal back? We cannot recover it fully; because of course quantization noise. But how do we at least get the from these values, how do we get these values we have to pass it through the inverse device and the transfer characteristic of that will be like this. If we draw only the positive the first quadrant of the plot; the compressors characteristic is like this is x this is y so the obviously the inverse will be like this.

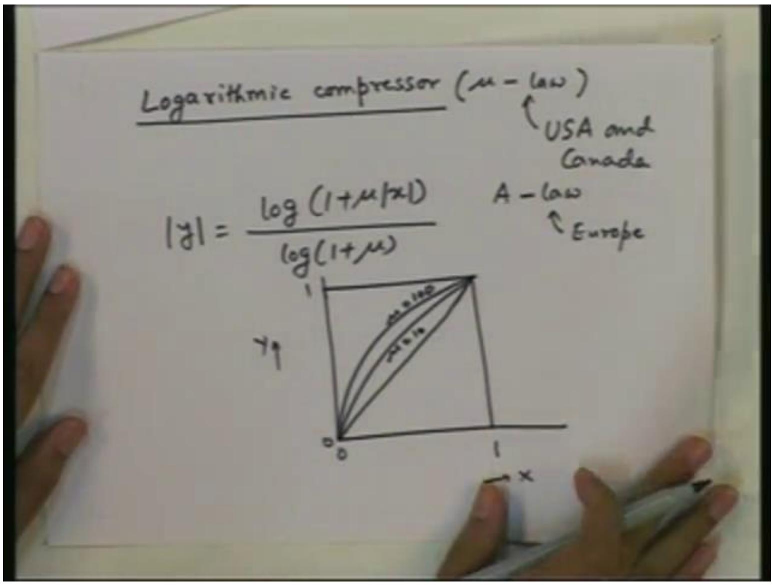
So, at the receiver or at the decoder we will simply pass the signal through first of all we will convert the bits into levels of the uniform quantizer. So, that is the and then that is passed through another non-linear device which is inverse of that which is present at the transmitter. So, we can see that any non uniform quantizer can be implemented in terms of a non-linear device and a uniform quantizer.

(Refer Slide Time: 17:15)



Now, there are some very useful compressors that are used in practice, most of the times the compressor is logarithmic in nature there is reason for this which we will not discuss in this class. Before we discuss logarithmic compressor from these plots one can see that this is the inverse of the compressor; so, this is called expander, this is compressor and that is expander. And, so these 2 combined is called compander and this procedure is called companding. So, this method of doing non-uniform quantization for signals like speech signal for which the low values are more probable then the high values this method is called compander okay.

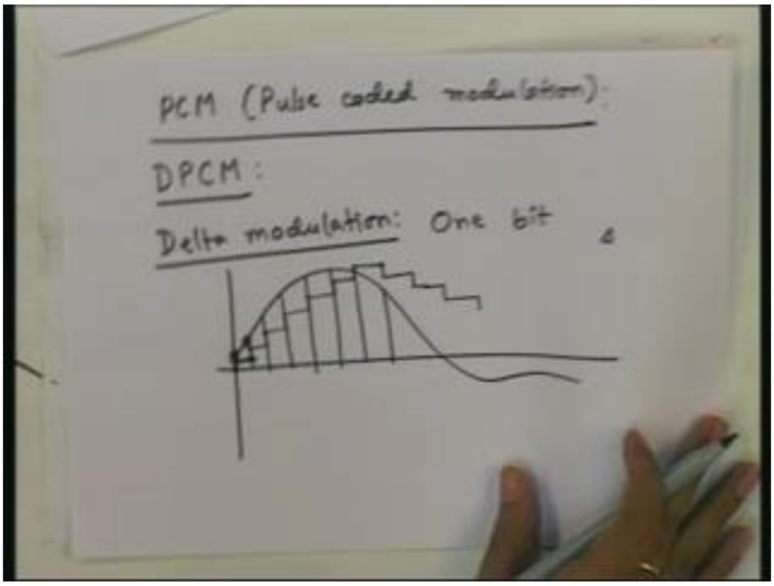
(Refer Slide Time: 18:04)



So, now let us to come back to logarithmic compressor; there are 2 different standards for this compressor, logarithmic compressor 1 is mu-law compressor which is used in the USA and Canada and another which is called A-law compressor which is used in Europe. This variables mu and A are simply because of those of the mu is the name of a variable which is used in the expression of the characteristic of the compressor. So, the characteristic of the u law compressor is this. So, a variable called mu appears in this expression and so this is called mu law compressor. Of course, we can take, we can give, we can name this variable anything else and call it some other way; but that is way it is named traditionally.

And, if we plot these characteristics for different values of mu they will look like this. We plot again only the first quadrant for mu equal to 0 we will get no compression that is the characteristic will be linear. And, for mu equal as mu increases there will be more and more compression this is for mu equal to 10, then this for mu equal to 100. And, similarly it will keep compressing more and more as mu increases okay. We will not discuss the A law compressor in detail it is similar to mu law compressor, it is also logarithmic in nature.

(Refer Slide Time: 20:56)



Now, this way of coding that is first quantizing the value of the sample and then representing the levels by suitable number of bits is also called PCM; that is abbreviation for pulse coded modulation. So, important point we will note here is that each sample is coded independently of other samples; whereas, in the schemes that we will discuss next we will see that the coded bits for the 1 sample depends on the coded what the value of previous samples were. So, once the scheme is DPCM where not the absolute value of the sample; but only the difference between the sample values, conjugative sample values is coded using finite number of bits.

So, what is done is that the transmitter as well as the encoder as well decoder both first have the initial value 0. Then, the input signal comes if it is greater than 0 it takes the difference between the value of the signal and 0 and then encodes that value. Then, next again another sample comes it takes the difference between the 2 samples and then encodes the difference using finite number of bits. Another, alternative is to we will discuss that in terms of a special case that is called delta modulation, here just like DPCM but only 1 bit is used per sample.

So, what is done is the transmitter has 0 in it is memory when it initializes it is encoding with the value 0; then, the signal value let us say signal is like this then it starts encoding it is sees that the sample value, samples are here these are the samples. The sample value is greater than 0; so, it transmits 1 and there is a fixed step size delta. So, it transmits 1 and what does the receiver do? Receiver receives 1 and it assumes that the transmitted;

the sample at the transmitter has the value delta. If it had received 0; it would have assumed that the value is minus delta. So, there is this but this is the actual value; but the receiver assumes that the value is delta which is let us say less than the actual value.

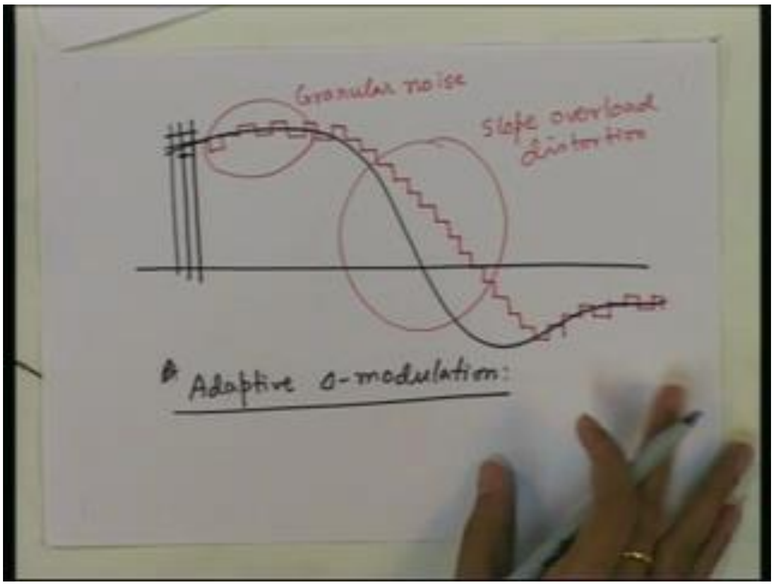
Then, it takes the next sample and it is still greater than the value that is there in the memory of the transmitter as well as the receiver; receiver also has delta value and transmitter also keeps track of what value the receiver will keep computing. So, delta is presently there in the memory both the transmitter and the receiver it sees the next sample value and if it is greater than the present value at the memory it transmits 1, if it is less than the present value in the memory in transmit 0. So, what does the receiver do? Receiver sees that for this example it receives 1 and it increases the value by another delta so like this. Then, the next value is still greater than that what is there in the memory; so, this is the value chosen so, it keeps doing that.

So, it is trying to track the original signal; but because the signal is changing faster than it can track it is not able to track it fully, so it tracks like this. Then, here it sees that the value in the memory or of the receiver as well as the transmitter is greater than the sample value of the signal. So, it transmits 0 and the next value becomes the present value in the memory minus delta; so, then it tries to track like this okay. So, there is an advantage of delta modulation when there is a high correlation between the samples that is when the samples are not changing their values very fast. Because only when the samples are not changing very fast the delta modulation can track it is value.

If for example, here the signal is changing the slope of the signal is high and as a result the value in the memory of the transmitter and the receiver cannot track it fully. So, how do we enforce this condition in practice that is we want the sample values to change slowly so that we can do delta modulation. So, the advantage of the delta modulation is that it uses only 1 bit unlike PCM; where it uses usually 8 bits or 4 bits depending on what is the resolution we want. So, here we need only 1 bit; but we need the samples to satisfy the added 1 extra condition that is there values should not change very fast. And, we can actually impose that condition in practice by having a higher sampling frequency than Nyquist rate.

So, if we take Nyquist; if you sample the signal the at Nyquist rate we have seen in the previous class that it is sufficient for sampling the band limited signal, it is sufficient to recover the original signal back from the sample signal. But then if you sample at exactly Nyquist rate what will happen is that the sample values will change arbitrarily. And, as a result delta modulation will not be able to track the sample values fast enough. And, for delta modulation to what we need higher sampling frequency and that of course again results in higher number of bits per seconds. Because higher sampling frequency means higher number of samples per second and even if you have 1 bit per sample we will end up using quite a lot of bits per second.

(Refer Slide Time: 29:07)

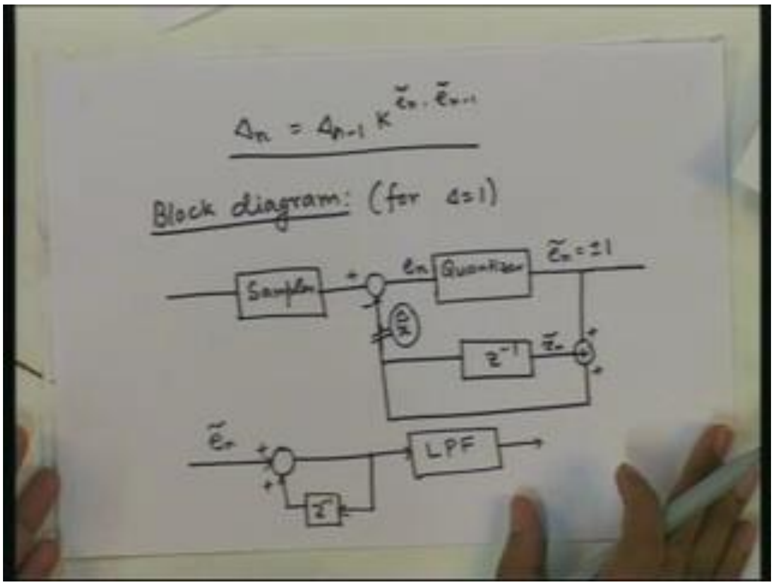


So, let us now discus 2 common types of distortions that delta modulation suffers from. Let us discuss that with an example; let us say the signal is like this and the step size is the sampling interval is like this and the step size is of this size approximately. So, how does you track the, how does the quantizer track the, how does the delta modulator track the, values of the samples? It has this value it is trying to track the value of the signal. So, the red signal is how the quantizer is tracking the value of the original signal that is this is the value in the memory of the transmitter as well as the receiver. As, we can see there is high distortion in this range where the signal is changing very fast. And, the quantizer is not able to track the signal value.

So, this is called slope overload distortion; this happens because of high slope in the transmitted, in the source signal. And, there is a different kind of distortion in the region where the transmitted signal is almost constant, here the value of the signal in the quantizer memory keeps togging between 2 levels. And, ideally it should simply settle to the correct value and here we see that does not happening; but it keeps togging between 2 values keeping the actual value in the middle and that is called granular noise. So, there is noise even when the source signal is not changing, even when the source signal is constant, there is a noise quantization noise so that is called granular noise.

So, how do we decrease granular noise; we can decrease granular noise by decreasing the step size delta. How do we decrease slope overload distortion? To, decrease slope overload distortion we actually have to increase the step size; because only than we will able to track the fast changing signal value. So, we have a 2 different type of constants to reduce granular noise and slope overload distortion. So, we have basically a conflicting interest. To decrease granular noise we need smaller step size; whereas, to decrease slope overload distortion we need higher step size. So, the solution to this is adaptive delta modulation; where we will adapt the step size depending on the need.

(Refer Slide Time: 34:10)



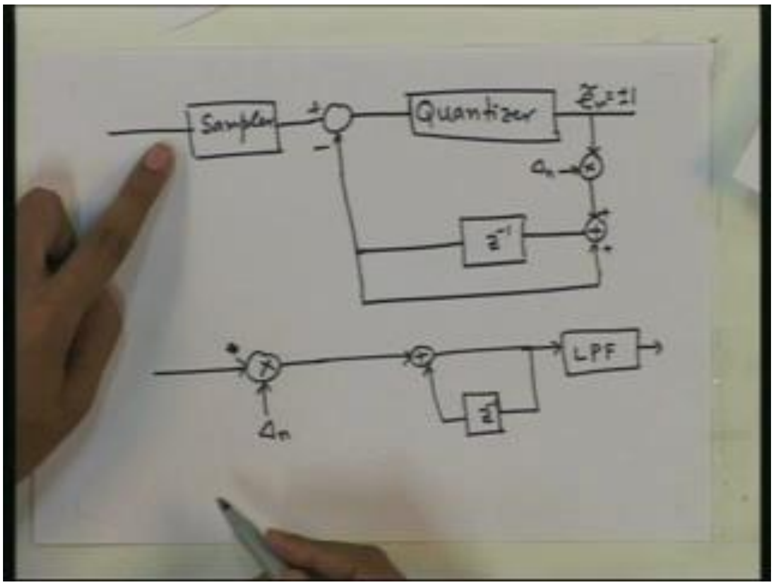
So, in adaptive delta modulation the n th step size is taken as a function of the previous step size and also what is the present signal value? And, the value in the memory of the quantizer? So, a simple adaptive delta modulation scheme is where the present step size is simply a factor of the previous step size. So, it is either divided by K or multiplied by K. So, we will see how it is actually en tilde dot en minus 1 tilde; where en tilde and en minus 1 tilde are the binary values that are being transmitted after delta modulation. We will discuss this in a little more detail after we discuss the block diagram of the delta modulator; where the symbols en tilde and en minus 1tilde will be clear.

So, this block diagram; let us first see for delta equal to 1 the simple case delta equal to 1, there is a fixed step size delta equal to 1. So, first of course we have the sampler and then we have subtractor and this is the output is en that is the error between the actual value of the sample and the value that is there in the memory and that is the value that is coming here. So, this is denoted we will see that this en will be encoded using 1 bit. So, this a single bit quantizer. So, this is en tilde that is the quantized value of en it is quantized to either 1 or minus 1. And, that is what is transmitted plus 1 or minus 1 you call plus 1 as 0 and minus 1 as 1 or whatever you can encode it using a bit.

So, we have this is the value in the memory. So, what do we do in the next time instances, for the next sample we take the value in the memory and then we add 1 or minus 1 to it 1 is the step size. So, we add 1 or subtract 1 depending on the transmitted bit. So, that is what is done here; we add this value with en tilde and then that is the memory value that will be stored for the next sample. So, that is passed through a delay and that is the memory in the next time interval okay. And, the receiver works just like this it is also receiving en tilde. So, it will just do this operation and we will it will get this is the value in the memory at the transmitter, this value is the value that the receiver also will have by doing the same operation on en tilde so this is the receiver.

En tilde is the input and then the signal is usually passed through a low pass filter; so that so as to just smooth smoothen the signal. As, you can see from the nature of the signal at the memory of the transmitter or the receiver it is rectangular small rectangular steps. So, that needs to be smoothened and then is done by low pass filter. So, this is how we recover the signal x n tilde hat which is also constructed reconstructed at the transmitter. So, the transmitter as well as the receiver both reconstructs this signal x tilde hat. And, now if you have adaptive delta modulator we need to change this step size. So, it will still transmit 0,1 or plus 1 minus 1 whatever; but while doing the processing, further processing it will needs to multiply this by the step size because the step size is something other than 1.

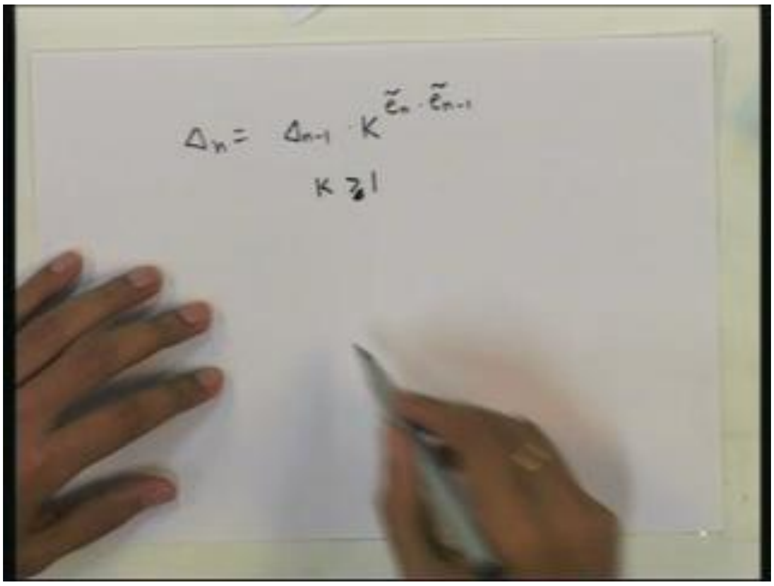
(Refer Slide Time: 40:18)



So, for adaptive delta modulator we will have the same block except added to another feature to it. So, this is en tilde which is plus 1 or minus 1; but this is then scaled by the step size and then the further processing is same as before okay. Similarly, we have to scale this at the receiver also we need to first scale the incoming value which is plus minus 1 by delta n. So, the posing is same as before except for this scaling of the input in the receiver. And, similarly here also at the transmitter the same as before except the scaling of the output by delta n for further processing.

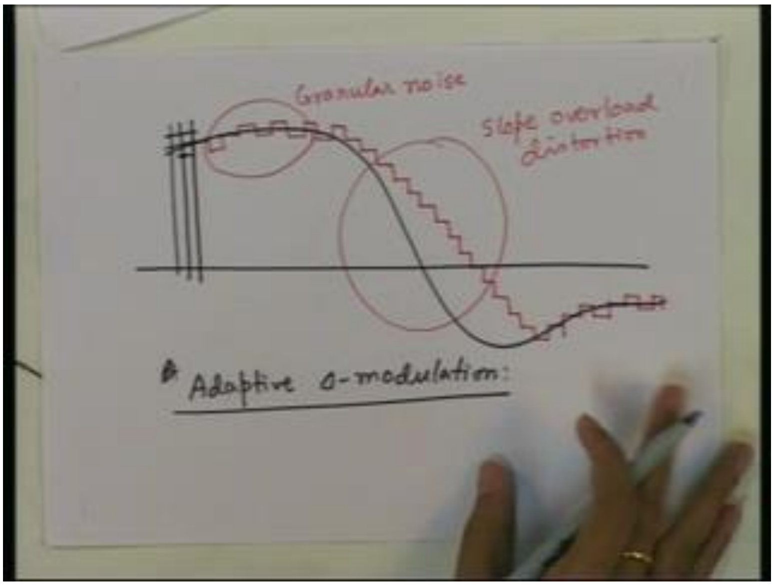
So, how do we get delta n at the receiver? The delta n at the receiver can be; if getting delta n for adaptive delta modulator here requires the knowledge of the input signal, then the receiver cannot construct it from the input. So, in that case there needs to be some overhead transmission to communicate how the delta n is changing; but that is even often not considered practical. And, usually techniques are used that that do not need extra transmission. So, where delta n can be can be constructed from en tilde itself.

(Refer Slide Time: 43:17)



So, 1 such technique is what we were just discussing; where the delta n is delta n minus 1 time s a constant K for what en tilde time s en minus 1 tilde. So, what it says is that of course K is a constant greater than 1. So, what it says is that if en and en minus 1 tilde are both same; then, the previous step size is multiplied by K and if en tilde and en minus 1 tilde are different then it is divided by K. So, what is the reason for that? The reason is let us consider the slope overload distortion and granular noise.

(Refer Slide Time: 44:21)

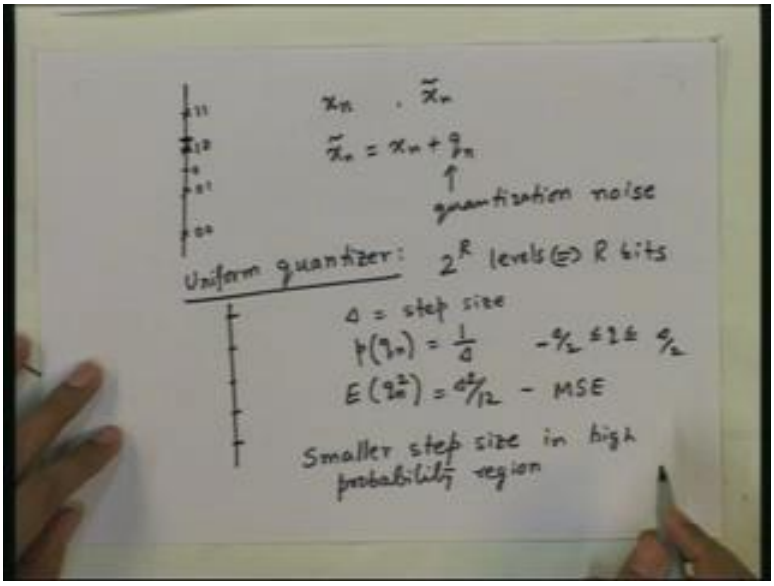


Here, we see that when the signal value is almost constant there is granular noise; the en is changing it is sign in every sample. So, when it is doing that en tilde time s en minus 1tilde will be minus 1. And, if it is minus 1; if the step size will be divided by K that is the present step size will be smaller than the previous step size. And, that means it will minimize the granular noise; because it is decreasing the step size when it is required. So, to reduce granular noise we need to decrease the step size and that is what is happening using this scheme. And, let us now consider the slope overload distortion; here, since the signal is changing at a higher rate, at a higher slope the en tilde is not changing either it is 1 for quite some time or minus 1 for quite some time.

If the signal is changing in the positive direction at a higher slope than en is en tilde is 1 for quite some time. If it is changing in the negative direction than en tilde will be minus 1 for some time. In either case en tilde time s en minus 1 tilde will be 1 and as a result the step size will be multiplied by K; which is greater than 1 and as a result the step size will keep increasing. So, the idea is that if the quantizer sees that the signal value is constantly higher than the value in the memory or constantly lower than the signal in the memory it needs to increase the step size. So, as to track the signal value the present step size is not sufficient.

So, it is increasing the step size whereas if the, if it sees that the signal value is greater in 1 sample and in the next instance it is smaller and soon. It is changing in every sampling instance then it knows that the signal is not changing fast it is almost constant. And, so to be able to track it better it needs to decrease the step size; so, as to minimize granular noise. So, it is trying to improve, it is trying to reduce granular noise as well as slope overload distortion by changing the step size in this fashion okay.

(Refer Slide Time: 48:00)



So, in today s class we have discussed quantization, we have discussed first uniform quantizer; where the levels are chosen uniformly keeping the gap between consecutive levels fixed. And, we have seen that the mean square error, mean square quantization error will be delta square by 12; where delta is the step size. And, as a result as what 1 can expect the mean square error is less when the step size is small and it is more if the step size is large. And, uniform quantizer is not the best quantizer for many practical source signals.

For example, speech signal takes smaller values with high probability. And, as a result it will be better to have dense levels in the near 0 region; and sparse levels away from 0. So that the mean square error is reduced because the values that are more frequent those values should be quantized at a dense, using dense levels. And, we have seen that any non-uniform quantization can be done by using a non-linear device and uniform quantizer. First a non-linear device can be designed. So, that the non-uniform quantization can be done by first passing the sample value through the non-linear device and then we uniform quantization.

And, for signals like speech signal for which the lower values are more reliable and the higher values are less reliable, the non-linear device will do some compression. It will attenuate the high values and amplify the low values and then the uniform quantizer will do it is job. So, this non-linear device will be a compressor for signals like speech signal.

And, the inverse device which will be used for used at the received will do the expansion to get the original signal back. And, so the compressor and expander will be used at the transmitter and the receiver respectively and this whole operation is do called companding.

And, then we have discussed adaptive delta modulation for avoiding or for minimizing granular noise and slope overload distortion. We have seen that to reduce granular noise we need to have smaller step size whereas, to reduce slope overload distortion we need to have higher step size. Now, to meet both ends we need to reduce the step size when needed and increase the step size when needed. And, that is exactly what is done in adaptive delta modulation.

Thank you.