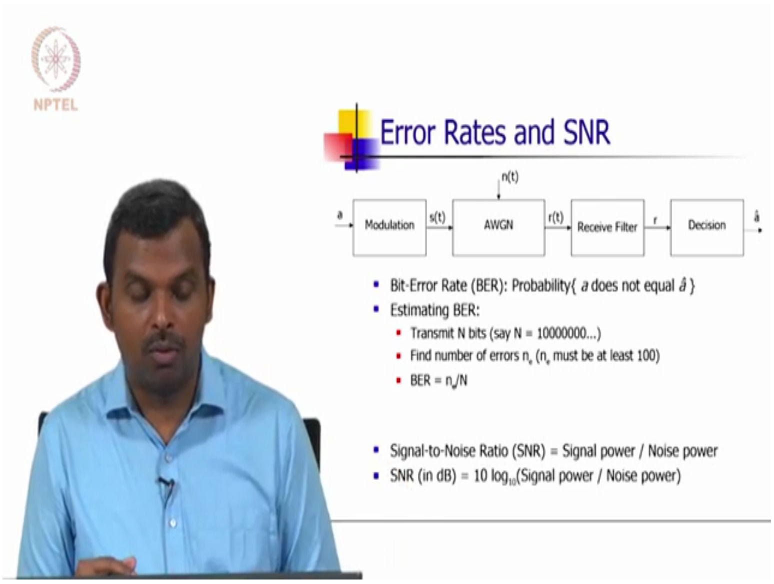
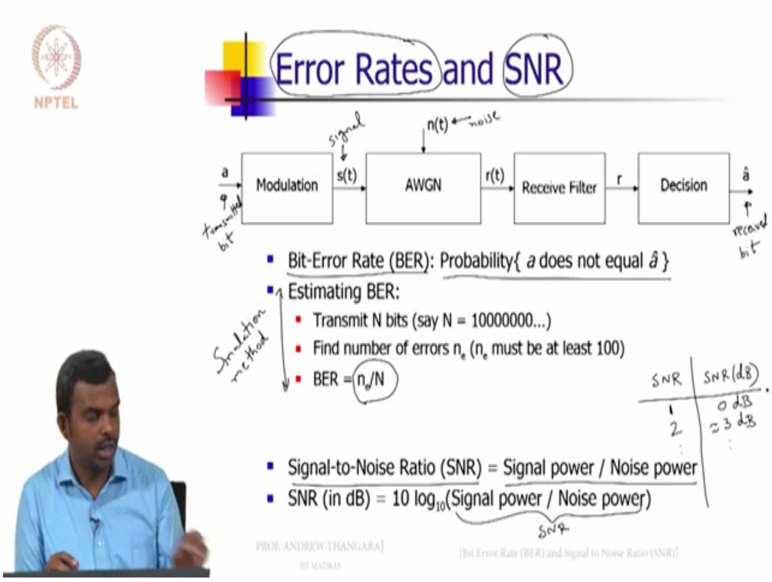
LDPC and Polar codes in 5G Standard
Bit Error Rate (BER) and Signal to Noise Ratio (SNR)
Professor Andrew Thangaraj
Department of Electrical Engineering
Indian Institute of Technology, Madras
Error Rates and SNR

(Refer Slide Time: 0:18)



So now we saw before that digital communication system has a very simple model like this, this is the continuous time model that I am showing here once again that s of t, n of t and r of t. Now what are the performance metrics or the important metric that you have to focus on in the digital communication system, there are two important metrics.

(Refer Slide Time: 0:41)



The first metric is signal to noise ratio (SNR) is an important metric Signal to Noise Ratio is very very important metric. So this is defined as signal power divided by noise power, okay so this s of t is the transmitted signal it will have a certain power, okay power is defined for signals in a certain way you can take the square integral of the signal and divide by the time, you will get the power, so that is the power of the signal.

In similar way one can also define power for the noise signal, remember noise is probabilistic one has to take expected values and all that because noise is probabilistic one has to take expected values but nevertheless one can define an ocean of a power for the noise as well. Now the ratio of these two powers, okay signal power divided by noise power is defined as the signal to noise ratio, okay so the signal to noise ratio has no units power by power it cancels this no units, okay and typically signal to noise ratio is reported in dB scale, how do you convert any number to dB? You take logarithm to the base 10 and then multiply by 10, so that is what you do to convert any number to dB.

Let me give you a few numbers here, so if the ratio is SNR and you want to compare SNR in dB may be a few things can be looked at, SNR of 1 actually corresponds to 0 in dB, okay SNR of 2 actually corresponds to roughly 3dB, okay so this is the calculation that you can do, you can add more values here you put any other function here so remember this is actual SNR you put 1 here log 1 base 10 is 0 so you get 0 dB, if you put 2 here log 2 base 10 is around 0.3 so you get 3 times and so on, alright so you can put any other value here you will get something in dB okay this 2 to 3 is a good number to remember SNR of 2 corresponds to 3 dB, okay so it sound of those things to remember.

So SNR, so what is SNR quantify? SNR if SNR is high then it means your signal level or signal power is higher than the noise power, okay so your noise signal is significantly lower in amplitude and power and all that compared to your signal power, signal power is very big the noise power is very small when SNR is high. On the other hand when SNR is low 0 dB, if SNR is 0 dB then your signal power is equal to the noise power, okay so when signal power and noise power are the same you are likely to make many more errors, right so SNR suppose to the case when the signal power is very large and the noise power is very small you may not be making many errors, okay.

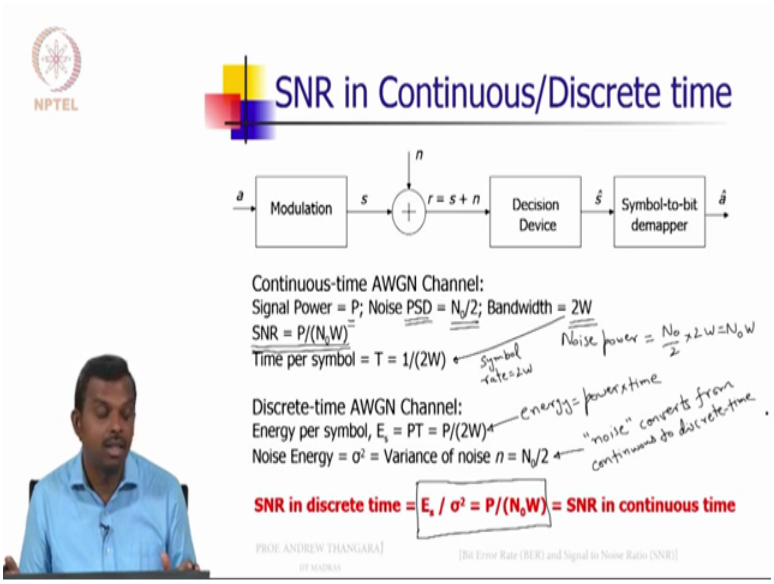
So this is so SNR in some sense quantifies (the signal) the quality of your channel, how good is your channel, okay and that is very important to know and all of our measurements and all that will be respect to SNR, okay.

Now the other very important measure in the digital communication system is error rate, okay bit error rate is the next important metric and that has a very simple definition you transmitted a, okay so this is the transmitted bit, this is the received bit, okay the transmitted bit to the received bit what is the error? Probability that a does not equal a hat, okay. So now in a typical communication system there is this Monte Carlo simulation method to estimate bit error rate, okay simulation method.

There is also a theoretical method to compute this I will talk about it soon enough, but for now let us understand what the simulation method will be? You actually transmit a large number of bits, okay so tens of thousands of bits you transmit and you know what bits were transmitted, you know what bits were received, okay after this system and then you compare you see how many errors were made and this ratio n e by N as we know from our studies of probability this is a good estimate of the bit error rate, okay the probability that a is not equal a hat a good estimate is this, okay so this is the way in which things are captured.

So as you can imagine as SNR increases your bit error rate is going to fall, okay so this is the essential trade-off in digital communication systems, (what is) how is bit error rate related to SNR, how does bit error rate vary with respect to SNR, okay so one can do also a nice and simple theoretical calculation for this.

(Refer Slide Time: 5:46)



But before that I want to spend a little bit of time in talking about this SNR in continuous time and discrete time very quickly just to convince you how this works, okay. So let us look at the continuous time channel in which the signal power is P and the noise power spectral density is N not by 2, so this is in the spectral domain you think of N not by 2 as a noise pro spectral density and let us say you are using a bandwidth of 2W, okay some number 2W hertz of bandwidth.

Now if you look at the noise power, okay noise power signal power was P, noise power is actually power spectral density times the bandwidth, so you get N not times W, okay. So in the continuous time domain the SNR becomes P by N not W, okay so this is the main calculation here, so it is a very simple calculation for the signal to noise ratio in the continuous time domain.

Now how do you get this noise power spectral density? In actual physical system there are devices that will tell you what the PSD is and many scopes another devices have ways of measuring this noise power spectral density, so you can get N not by 2 and you know what bandwidth you will be using so you can compute this quantity, so all these quantities are perfectly computable in a physical system, okay.

Now if you are using a bandwidth of 2W it turns out that is this Ny quist rate of transmission in the channel you can send one symbol the time that you can spend for it is 1 by 2W, okay in the other way the other way to put it is the symbol rate is 2W, okay 2W symbols per second you can send without any interference, etc. So the time that you have in one symbol is 1 by

2W, okay so the power is P and the time for which you can send that one symbol the signal that you can send is 1 by 2W these are rough estimates but this is pretty good.

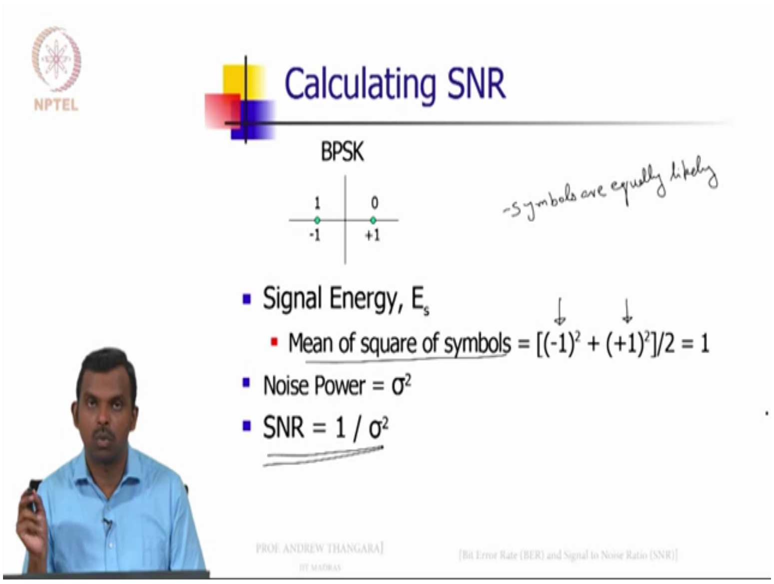
So when you convert from continuous time to discrete time one does not think of power, one thinks of energy, okay so energy because you are sending a certain signal for a particular time the total accumulated energy corresponding to that signal that you sent is a good measure of the energy you expended for transmitting that particular bit, okay. So now energy is power into time, okay power was P, the time was 1 by 2W so this is energy per symbol, okay so (this is a) this comes from energy equals power into time, okay.

So now how does the noise get converted into discrete time? So this requires a little bit of calculation, okay I am not going to go through that there is lots of theory behind it, so it turns out when you convert noise from continuous time whatever continuous time model you might have to discrete time, the noise energy is simply the variance of the noise and that is simply the level of the noise PSD, okay N not by 2. So this is how noise converts from continuous to discrete time. So the variance or energy in the noise is simply sigma squared and the energy in the symbol is P times T is P by 2W.

So the main moral of the story is if you take E s by sigma squared in the discrete time domain that exactly equals P by N not W in the continuous time domain. So the main story in this slide is you do not have to worry about the continuous time situation at all, you only worry about the energy in the symbol and the noise variance in the discrete time domain and your SNR in discrete time you define it as E s by sigma square energy in the symbol divided by sigma squared and you are fine, so that is completely enough.

So keeping this in mind we will never ever go back to the continuous time domain again, we will only be in the discrete time domain and we will be happy with the calculation of the SNR in the discrete time.

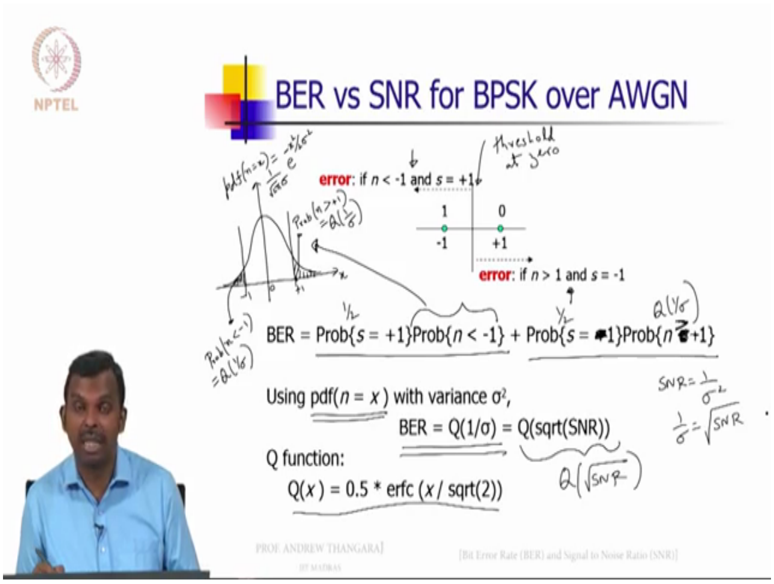
(Refer Slide Time: 10:08)



So let us see how to calculate SNR for a particular modulation scheme, the modulation scheme that we will use is BPSK, so it turns out the signal energy is mean of the square of the symbols, ok mean of the square of the symbols and usually we will assume that the symbols are equally likely, equally likely meaning they are uniform, plus 1 happens with probability half, minus 1 with probability half. So how do you compute the signal energy? You square plus 1, you square minus 1, right so this is 1 symbol plus 1 you squared it and another symbol minus 1 you squared it and then you divide by 2, so this is the mean of the square of the symbol so you get 1, okay.

In your BPSK model this E s the signal energy quantity has this very simple expression which is 1, okay and this is equivalent to the energy in the continuous time model, it is exactly the same, of course there will be some units and all that which will be different but this 1 is a good model for that and noise power is sigma squared, so noise power remember in the discrete time domain noise power if you want to simulate and if you want to run it is just Gaussian distribution, you can generate Gaussian Noise of any variance, so whatever variance you pick is the sigma square so SNR becomes 1 by sigma square, so that is the main story here. So in the discrete time model assuming BPSK, SNR is 1 by sigma square, it is a very straight forward simple calculation.

(Refer Slide Time 11:50)



So now that we have SNR how do we do bit error rate? So for BPSK, AWGN what is BER versus SNR, how do you compute bit error rate? So here is a picture to show what happens, so remember we are going to use a threshold at 0, so we are using a threshold at 0 and if you transmit a plus 1, right you are going to say the received value if it is positive I will not make any error, on the other hand if you transmit a plus 1 and the received value becomes negative you will make an error, so that is what is this situation here, you transmitted a plus 1 but the noise was so high on the negative direction that it pushed you over the threshold became less than minus 1 so you made an error.

And here is the other situation when you transmitted a minus 1 but you made an error because noise was very high it became greater than 1 so it pushed you to the right of 0, so I can describe the probability of error the bit error rate expression has two parts to it the sum of two parts, the first part is the probability that s equals plus 1 and then the noise is very high n negatives so that you make an error, the second part is probability that s is minus 1 and then the noise is very high n positive and you make an error.

Now the first part in this probability is easy, probability that s equals plus 1 is half, probability that s equals minus 1 is half that is easy. Now what about this guy? Probability that n is less than minus 1. So now to compute this one needs the pdf of the noise, the noise is Gaussian distributed so pdf is like this the noise is Gaussian and the pdf of noise has this formula 1 by root 2 Pi sigma e power minus x squared by 2 sigma square that is the formula and if you actually plot it you will have some fall like this, it will look like that, it will never quite go to 0 but it will go very very close to 0, so 0 then you have plus 1 here let us say and minus 1 here let us say, it will be symmetric I am not drawn it quite symmetrically but it will be symmetric.

Now what is this probability that n is greater than plus 1 or less than minus 1? It is this area, this area is the probability that n is less than minus 1 and this area which (maybe I should redraw this let me just see this, okay let me redraw this) so it should be plus 1 somewhere here and this area out here is the probability that n is greater than plus 1, okay. Now those areas can be represented in terms of the Q function, this Q function is defined here I will not go into details here the Q function is available in most platforms computing platforms you can use it.

So if you write it down carefully you will see that these two probabilities are actually equal to Q of 1 by sigma and this is also equal to Q of 1 by sigma, okay. So once you write that down, this is Q of 1 by sigma, this is also Q of 1 by sigma. So if you multiply by half and add it up you will simply get that the overall BER is Q of 1 by sigma. Now if you remember SNR is 1 by sigma squared, so what is 1 by sigma? Square root of SNR, okay so the probability of (error) bit error rate is nothing but Q of square root of SNR.

So this is the very classic formula for BPSK, AWGN I just quickly showed you derivation for how it works, so this is bit error rate versus SNR for BPSK, AWGN. So this is the very nice complete theoretical calculation.