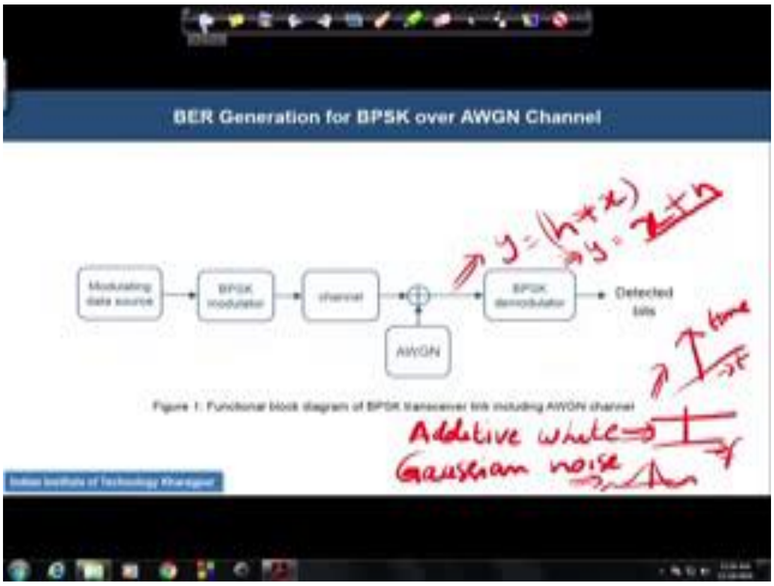
Spread Spectrum Communications and Jamming

Prof. Venu Balaji
G S Sanyal School of Telecommunications
Indian Institute of Technology, Kharaghpur

Lecture – 34
Tutorial – IV

I am Venu Balaji. I am technical class assistant to Prof. Debarati Sen. Today’s tutorial is on MATLAB code for BER generation of BPSK system over AWGN channel.

(Refer Slide Time: 00:37)



This is the block diagram for BPSK transceiver including AWGN channel. First of all, the modulating data source will be there where the transmitted data is given to BPSK modulator. BPSK modulator does the BPSK modulation, BPSK modulator first converts the data bits to bipolar signals; that means, 0 will transmitted to minus 1 and 1 will be mapped to plus 1, this logic depends upon our implementation. Then after that each this bipolar transmitted signal will be transmitted over the carrier into the channel. Channel here we are considering as AWGN channel, AWGN channel means additive white Gaussian noise channel.

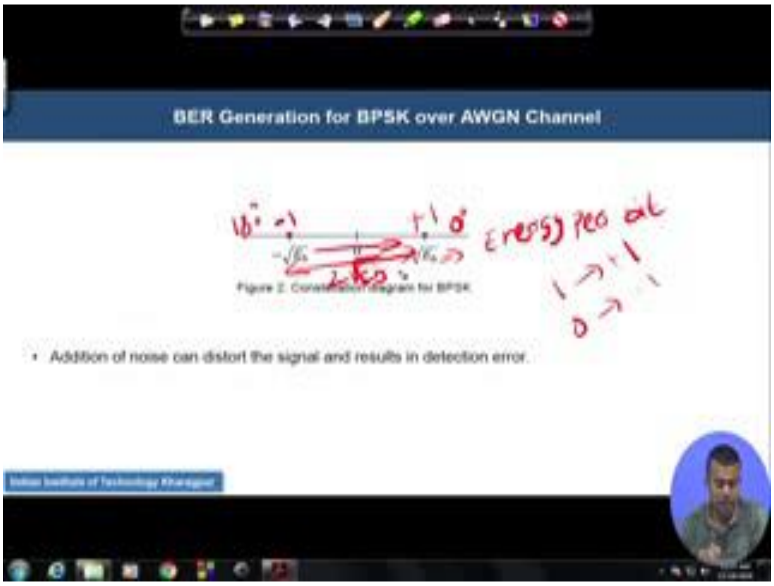
So, here additive white Gaussian noise means the noise here additive white Gaussian; additive means here the noise will be the channel will be adding noise in additive nature and white means this channel are does not account for any non-linearity in the channel or dispersion or and it does not introduce any multipath components into the channel, it treats all frequency it provides same power over all the frequencies. That means, if you look at in this one in frequency domain, it will be like flat over all frequencies.

You can consider this channel as flat fading channel. And the noise whatever is inherent in the channel will be Gaussian noise, Gaussian or you can call it as normalized distribution, the channel consist of many noise sources like black body radiation from earth and thermal noise introduced by the electronic components and many other sources will be there. But according to central limit theorem of probability theory when you include all the noise when you when you have s, many random variables, when you take the summation of all the random variables, their probability distribution will be like convolution among the probability distribution of all the random variables will be convolved. And finally, it will produce Gaussian distribution, so that is why the noise that is present in the additive white Gaussian, this AWGN channel is nothing but Gaussian noise or normalized distribution.

Then in frequency domain the power of the channel is same over all frequencies that mean, it is flat fading channel. If you consider this one in the frequency domain in time domain, time domain it will be like impulse kind of response. Gaussian noise will be like this is time domain nature. Channel, here channel is nothing but it is like it is like a filter with some tapping with some coefficients. Here, if you look at in time domain, the time domain response of channel is nothing but one filter tap coefficients. If you convolve this channel response with that transmitted BPSK signal, it will be like here if you consider this one as y equal to h convolution with suppose x is nothing but transmitter signal, then here h will be like impulse kind of channel behavior.

So, this will give you to the output of the channel is nothing but convolution of impulse response with anything is nothing but the same signal you will get that means, here output of the AWGN channel is nothing but the transmitted BPSK signal over noise gets added. So, the transmitted signal with the addition of noise will be given to BPSK demodulator; BPSK demodulator does the demodulation and then we will get the detected bits.

(Refer Slide Time: 06:01)



This is the BPSK constellation. BPSK constellation, in BPSK, as we are transmitting 0 and 1 bits; here one is mapped to plus 1, and 0 is mapped to minus 1. So, if you draw the constellation of this BPSK diagram, it will be like plus 1 will be having energy of square root of E b where E b is the energy per bit. This minus square root of E b is the energy per transmitted minus 1 means corresponding to 0, it will be like minus square root of E b, both the see both the transmitted bits are separated by 180 degree phase shift, this will be like 0 degrees and this is 180 degrees. And the distance between the constellation points is 2 into square root of E b.

This BPSK modulation is more robust to the amount of noise required or the amount of the distortion required to make the system unable to detect the bits. That means, you require very big amount of noise or distortion required to move the signal points to this side, so that you cannot detect, that means if the distance between the signal points becomes less that means, when the signal points come closer to each other, then you will not be able to detect the signal. But here in the case of the BPSK signal, BPSK modulation, you are having the maximum distance between the signal points. So, that you need a very high amount of noise introduction in the system, so that you will not be able to detect the bits generally which will not be a feasible that is why BPSK system is more robust to any amount of distortion introduced in the channel.

(Refer Slide Time: 08:41)

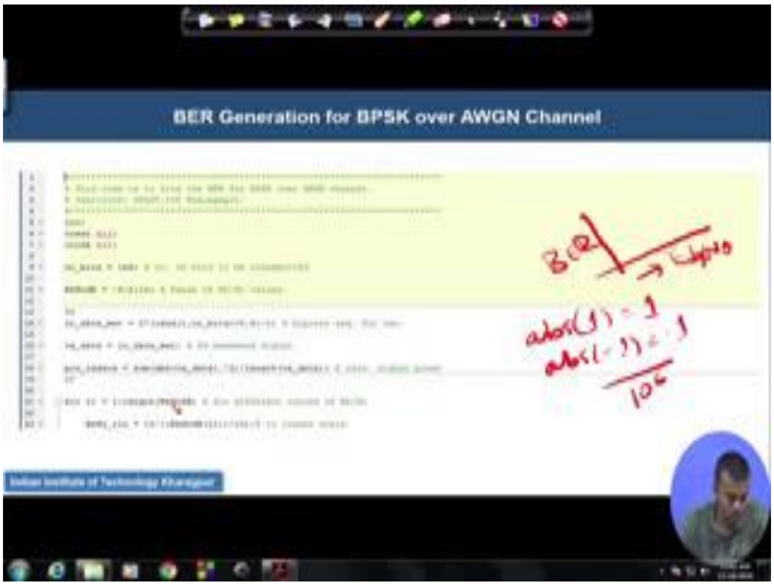
This is the MATLAB. Now, I am going to tell about the generation of BPSK, BER generation for BPSK over AWGN channel. First, I am considering here 10 to the power of 6 number of bits to be transmitted. Next, here I am going to plot BER with respect to; this is probability of error or you can say this is bit error means, how many number of bits are in error over the transmitted number of bits. Suppose, I have transmitted 10 power 6 bits means suppose if 10 bits are in error, 10 power 6 means bit error will be around 10 power minus 5 plotted against E b by N naught in dB. So, generally it will come like monotonically decreasing function. This thing I am going to simulate with the help of this MATLAB code. So, numbers of bits transmitted are 10 power 6; and E b by N naught is taken in dB scale. And I am taking from minus 4 dB to 10 dB over an increment of 2.

Next I am generating here I have written a logic to convert to map 1 to plus 1, and 1 to minus 1. So, further I have taken a distribution called rand, rand is a uniform distribution. Rand is a uniform distribution generation like it will generate the bits between it will generates random numbers which follows uniform distribution between 0 to 1. If the generated; and if how many number of bits you want to generate randomly that many number of bits you can pass as an argument to the rand function.

Suppose, you are generating you are passing random of 1 comma 10 power 6 bits that means, it will generate 1 into 10 power 6 number of bits like this. And each one will be varying from 0 to 1 means it can be 0.2, it can be 0.8 like this. But they are within the limits of 0 to 1, so that if the randomly distributed numbers are greater than 0.5 that means, I am taking here 0.5 as a threshold. So, if the generated number is greater than 0.5, then it will be considered as 1, then it is multiplied by 2, and then subtracted one means 1 is mapped to 1; if it is less than 0.5 means then it is considered as 0, 2 into 0 - 0 minus 1 minus 1.

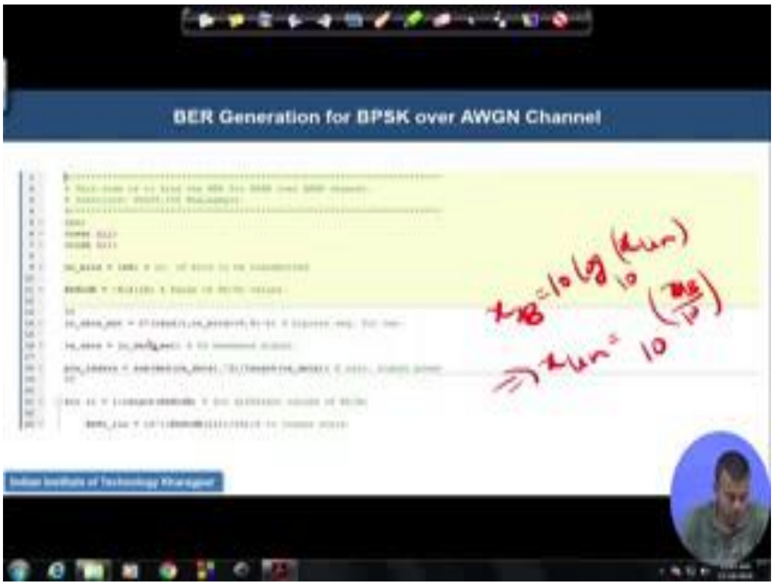
In that way, I am generating bipolar sequence from the transmitted number of bits. Next, this generate bipolar sequence for this bipolar transmitted sequence, I am calculating the power here, I have taken; this transmitted data will be a sequence of plus 1 and minus 1s. So, I am taking the absolute value of those is that sequence of plus 1 and minus 1s.

(Refer Slide Time: 12:36)



Absolute value of plus one or minus one will be like absolute value of 1 will be 1; absolute value of the minus 1 will be minus 1. If it is complex number then you will get complex magnitude. And I am squaring them that means, I am calculating the energy of each bit, then I am summing all of the energy I am summing the energy of the all bits then I am dividing by total number of bits; that means, 10 power 6 that will give me the power of transmitted data. Now, as I mentioned earlier I am going to plot bit error with respect to E b by N naught, so that is why I am going to vary E b by n naught regularly. For that I am going to run a loop, for loop with respect to the length of the E b by N naught that is what I mentioned here. For loop, I am running from one to length of E b by N naught in dB scale.

(Refer Slide Time: 13:49)



Next I am converting E b by N naught in dB to the linear scale. As you know, suppose x is in dB it will be like it can be written as 10 log x of linear to the base 10;that means, x linear can be written as x dB by 10 10 to the power of x db by ten. That is what happening here E b by N naught linear is nothing but 10 to the power of E b by N naught dB of each element divided by 10.

(Refer Slide Time: 14:24)

Next, I am generating noise, for generating noise inherently present in any channel is generally complex noise. So, to generate complex noise, I am considering a random distribution, random distribution as I mentioned the noise present in the channel is Gaussian noise or normally distributed noise that can be generated by using random function. Here you have to pass the argument of length of the transmitted data so that the generated noise and your transmitted bits will be of the same length that means, the frequency of the transmitted data and that means, bandwidth of the noise data will be same.

As here I am generating the complex noise, then I am scaling with power of the transmitted data and E b by N naught in linear scale that will give me the noise. The noise as I told you earlier y equal to h conjugate x sorry h convolution x plus n, here h is impulse response that will may give the transmitted data plus noise so that is what I am doing here. Here received data is nothing but transmitted data plus noise. This received data is given to the BPSK demodulator. BPSK demodulator will be having some threshold in order to detect the bits detect. So, here if the received data is greater than 0; that means, if the received data than 0, I am combining multiplying with two and one that means, if the received data is plus 1, 2 into 1 minus 1 will give me the 1.

If the received data is less than 0 means if the received data is greater than 0 then only it will give me 1. If received data is not greater than zero then it will written 0 then 2 into 0 minus 1 will give me minus 1. In that way by keeping a threshold of zero I am able to de map the signals, I am able to detect the signals this is decoded signal, this decoded signal. Then again compared with the transmitted data if both the transmitted data and the decoded signal are same, then that will give me the zero output otherwise it will give me the output one.

Suppose here I am transmitting 1 minus 1 1, here I am receiving 1 plus 1. Then by comparing with this inequality operator this will give me the output of 0 1 1. And I am summing all of them then sorry it will be 0, then after summation this will be equal to 1 that means total number of bits that are in error is 1. Similarly, as we are comparing those 10 power 6 transmitted bits and the decoded signal over AWGN channel and I am summing all of the bits that will give me the probability of error in AWGN channel. So, I am dividing with the total number of bits, total number of bits means 10 power 6 this will give me the probability of error over AWGN channel with respect to simulation.

This simulation will run for different number of as I am running loop for loop for different number of E b by N naught. So, this will give me the probability of error with respect to each E b by N naught. Next this is with respect to this is simulated probability of error, next time calculating probability of error of probability of error of BPSK system core AWGN channel theoretically; theoretically, it follows 1 by 2 into error complement sorry.

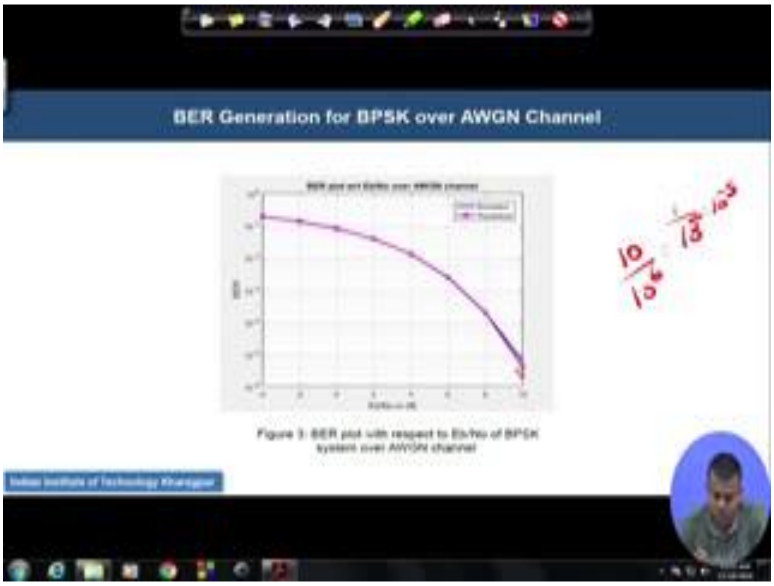
(Refer Slide Time: 19:06)

This is probability of error theoretically is 1 by 2 into e r f c of square root of E b by N naught. So, I am implementing that here this will give me the probability of error theoretically. Then I am going to plot both probability of error in simulation and probability error with theoretically on semi log semi log y using semi log y command. This semi log y command will generate a plot by taking this x component, and it plots graph with respect to x component, and y component will be taken in logarithmic mode. So, probability of the error means here y component is probability of error, this will be taken in logarithmic mode. So, it will plot probability of error in simulation mode. And I am holding the graph then I am plotting the graph with respect to then I am plotting the graph of BER with respect to theoretical.

These are like x label you have to mention, what you are taking on the x label and y label. Then legends means at the corner generally if you do not mention the position of the legend is generally it will be like northeast, it will mentioned what you have plotted.

Here I am plotting simulated and theoretically probability of errors that will be listed with respect to color in the legend box. Then you have mentioned the title, then in order to make the grid on you have to give this command grid on command.

(Refer Slide Time: 21:01)



This is the BER plot that we got after running that MATLAB code. This is the BER. The BER means the probability of the error on y-axis, x-axis E b by N naught in dB. As you see over AWGN channel the simulated the probability of error that we simulated over the AWGN channel, and theoretical probability of error over AWGN channel are almost same that means, this graph can evaluate like this. To generate a probability of 10 power minus 1, you require this 1, 10 power minus 1 you have to supply at least minus 1 dB sorry. To generate a probability of 10 power minus 1, you are required E b by N naught should be minus 1 dB. To generate the probability of error of 10 power minus 2, you require E b by N naught of 3 dB something like that.

And the only difference comes as the E b by N naught increases means if you want to generate, this is like as you are going from top to bottom along the y-axis the BER gets decreases. That means, out of 10 power 6 bits here, if you take probability of error as 10 power minus 5 that mean out of 10 power 6 transmitted bits only 1 bit is in error, sorry 10 bits are in error 10 by 10 power 6; that means, 1 by 10 power 5. That means, 10 power minus 5; that means, out of transmitted 10 power 6 bits only 10 bits are in error. To generate that probability of error E b by N naught you require an E b by N naught of around some nine 0.4 or 5 something like that. In the case of theoretical mode in the case simulation there will be you need an extra addition of 0.4 or 0.5 dB extra means there will be a mismatch between simulated probability of error and theoretical probability of error. You can say always theoretical calculations are much more accurate than the simulated that is what we observe in the generated graph.