LDPC and Polar codes in 5G Standard

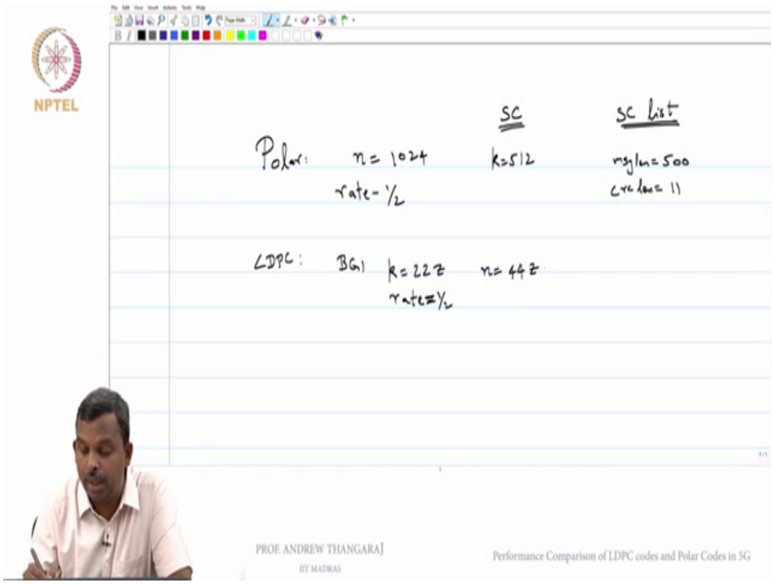
Professor Andrew Thangaraj
Department of Electrical Engineering
Indian Institute of Technology, Madras

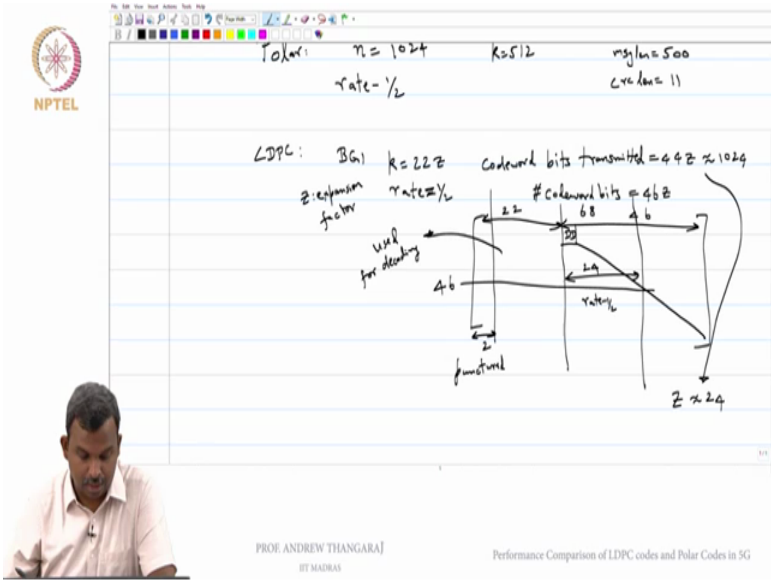
Performance Comparison of LDPC codes and Polar codes in 5G

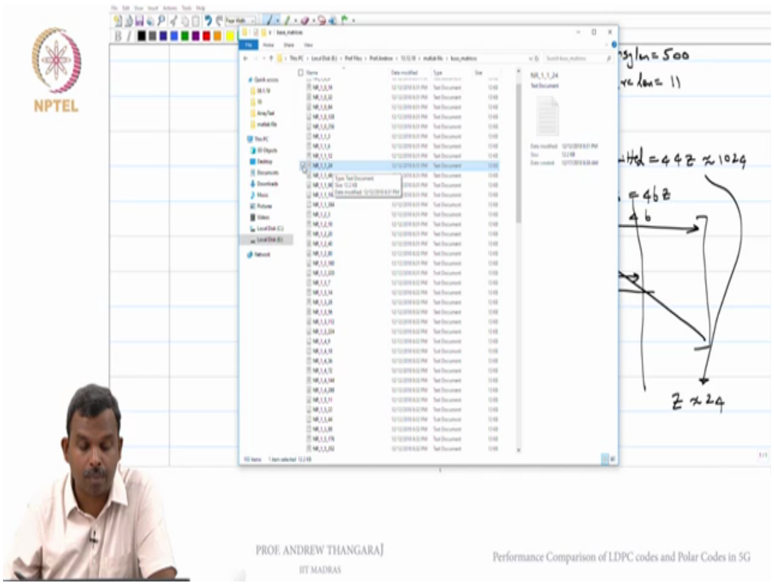
Hello and welcome to this lecture. So this is probably the last lecture in the class, we will be talking about doing simulations and comparing error control codes and making plots of error rates versus eb over n not. So that is the main agenda for this class, for this short lecture here. We have already written decoders for LDPC codes, we have already written decoders for Polar codes, the successive cancellation decoder, successive cancellation list decoder we want to compare all three in a comparable scenario, maybe similar rates and then similar block lengths and see what happens, what performs better?

Now there are lot of parameters to choose and all that one can keep exploring the space a lot, I have done it for some typical case, you are welcome to try it for other cases and compare yourself, okay. So let me show you how I go about doing it, so that we clear on how it is done, okay.

(Refer Slide Time: 1:12)







So the first thing is I want to compare so now for the polar code I know that the block length the maximum block length atleast is 1024, so we will fix that as 1024 and I will want a rate of half, okay. So now if you do SC decoding then you do not need CRC so you can take K to be 512, on the other hand if you do SC list decoding successive cancellation list decoding then you cannot take K equals 512 because you need that CRC, so how many ever bits of CRC you have to add you take, standard specifies multiple lengths of CRC, what we are going to do is we will take the message length to be 500 and then the CRC length to be 11, so 11 is one CRC polynomial which is there in the 5G standard, so we will pick from there.

So you see the total length is 511 and this is sort of comparable roughly comparable. Now what do we do for LDPC? Now for LDPC remember there is base graph 1 and base graph 2. Now if I want rate half with base graph 1, if you remember the K gets fixed in LDPC, K becomes the base graph 1 it is 22 times Z. So if that is the case then for rate half okay roughly half I will put, so we will not write exactly half, maybe we will get exactly half.

You need n to be 44 into Z, so 44 Z into so when I say n so let me say what 44 Z needs to be the code word bits transmitted equals 44 into Z, remember Z is the expansion factor, right so you always have an expansion factor. But there is also this thing about puncturing, so you remember the first 2Z bits always get punctured in the LDPC code, so that always happens. So actual number of code word bits will actually be 46 into Z, so it will not be 44 into Z, 46 into Z because and then 2Z gets cut off and then in terms of the parity check matrix also you have to pay some attention here.

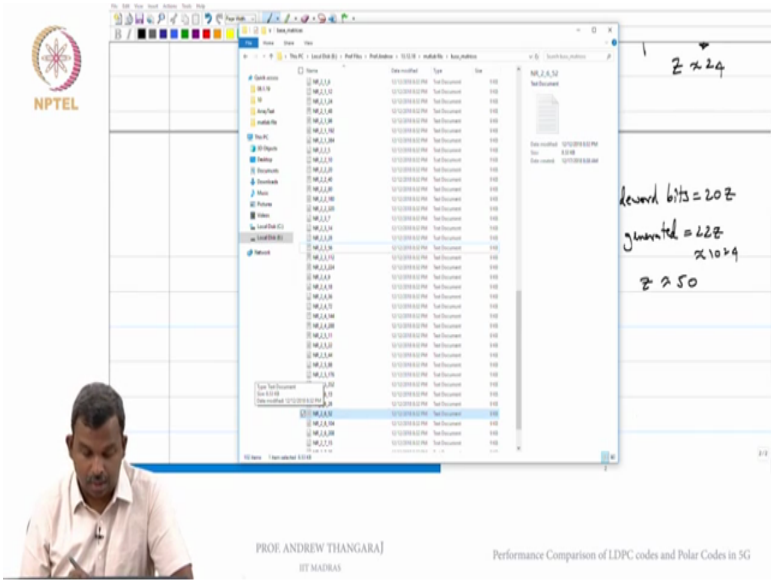
So you have the 22, so actually the overall parity check matrix is 46 by 68, right that is the base graph 1, so that is the base graph 1 here so you will have the first 22 rows being the message then the remaining 46 being parities, out of these the first two are punctured the message ends here and then you will start getting the parities and remember the parities there will be the short block which has double diagonal structure and then after that it is just diagonal (I didn't get this right) diagonal right upto this.

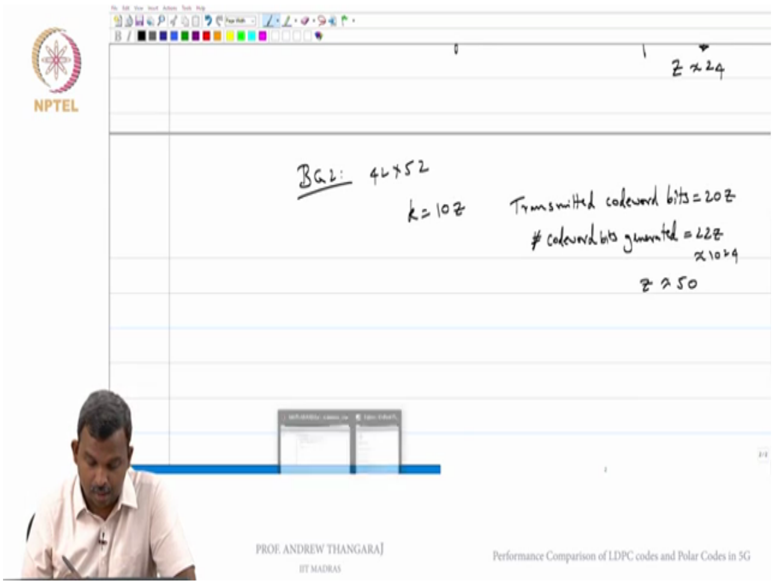
So now what is the meaning of saying number of code word bits is 46 into Z, here is where the rate matching comes in I have 22 Z here, I need totally 46 into Z so how many more will I take? I will take 24 Z more, so how many parities will I take for rate half? For rate half I will take 24 parities, so I will stop there. So when you stop there, remember there are 24 parities that you got here beyond that you are puncturing so you are not transmitting those code words beyond 44, so your parity check matrix sort of shrinks.

So you have to use only this part of the parity check matrix for decoding, it is used for decoding. So this is something that we saw before in one of the lectures on how to adopt the rate matching to the LDPC codes. So this is just some numbers, remember even though you have 46 Z the first 2 Z gets punctured, only 44 Z is transmitted so you have a rate of half it is number 1.

The next thing is I want a block length around 1024, so this needs to be around 1024 so that also sort of fixes Z, okay so from here if you see the value of Z will get fixed around what is 1024 divided by 44, it is I think roughly around 24, so this gives you a Z of around 24, okay and if you go and now look at the list of expansion factors possible in the LDPC code you can do that if you see the base matrix the list of base matrices, you see 24 occurs here okay that is the base matrix that you can do NR_1_1_24 that is possible, okay so this is with BG1.

(Refer Slide Time: 6:18)

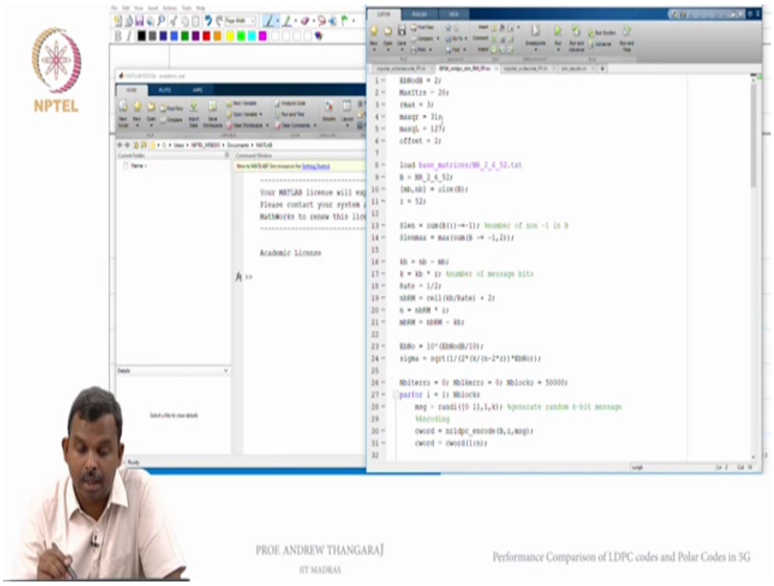


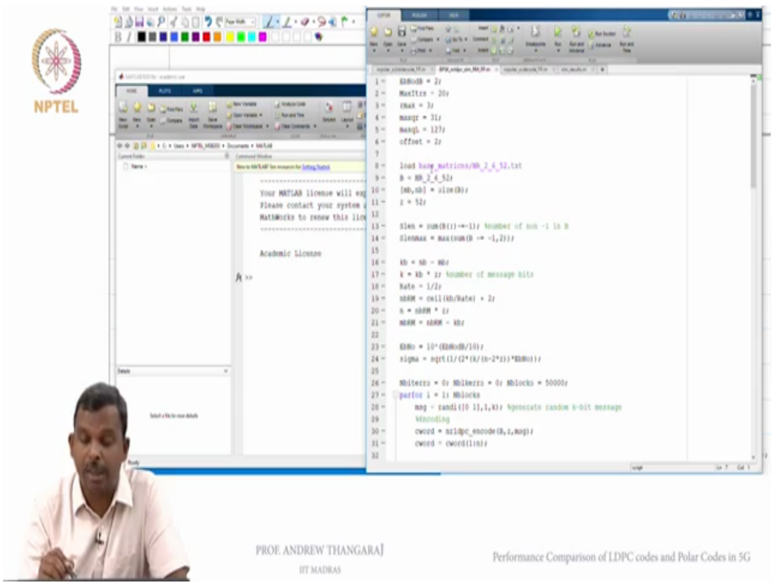


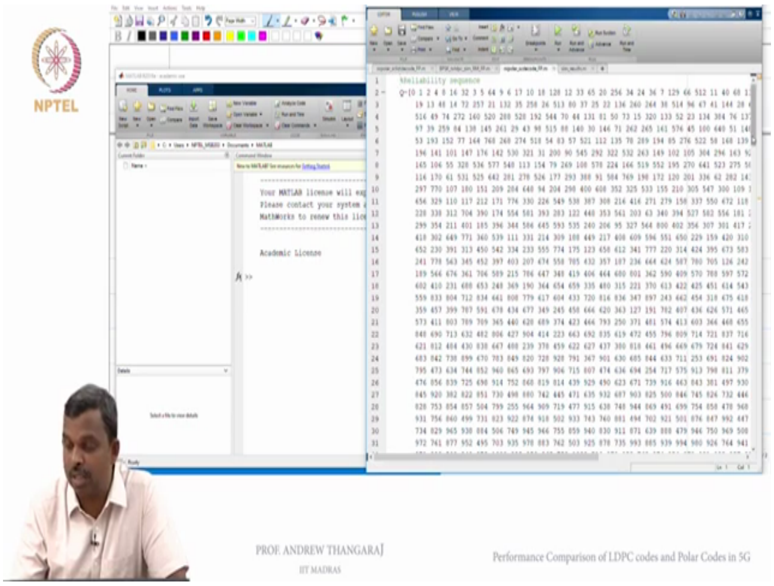
You can do a similar sort of setup for BG2, I will quickly write that down without showing you all the pictures, so if you do BG2 if you do BG2 then you remember BG2 is 42 by 52, so your K becomes 10 times Z and your code word bits transmitted the code word bits transmitted is 20 times Z which means total number of code word bits generated is 22 into Z so that is the logic for getting rate half.

And now this is this have to be roughly 1024 then Z is about 52 I think some 50 or something we will get. So if you go back and look at the list of possible things for NR2 you get something around 56 I think, so there is this 56 this comes here, so you can pick 2_ 3_56 if you like or maybe something else also that we will see (())(7:31) there is also this 2_6_52 you can pick I think this is better 2_6_52 is closer so this is something that one can pick in the from the possibilities and again rate is going to be half you have to take one part of the parity check matrix and decode.

(Refer Slide Time: 7:53)







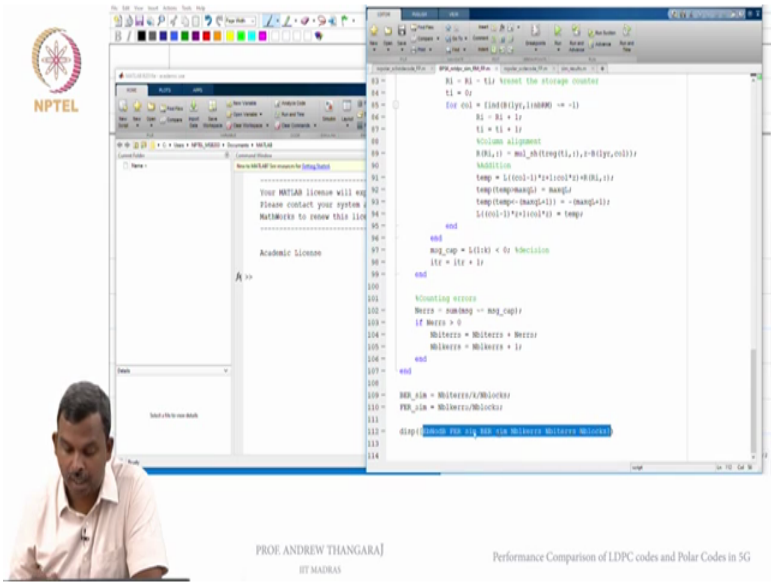
So let me show you how the MATLAB code works out just to show you one possible case here this is the LDPC part we will come to the polar part later on, so you can see that time I am taking in I am reading NR_2_6_52 that is the base matrix and Z becomes 52 and then you can see I am setting rate as half and then this gives you that just the NBRM which is the part of the total number of code word bits generated, so this is divided by rate plus 2 the additional 2 that you do for the puncturing and then you reduce the parity check matrix to NBRM minus KB only that much you take.

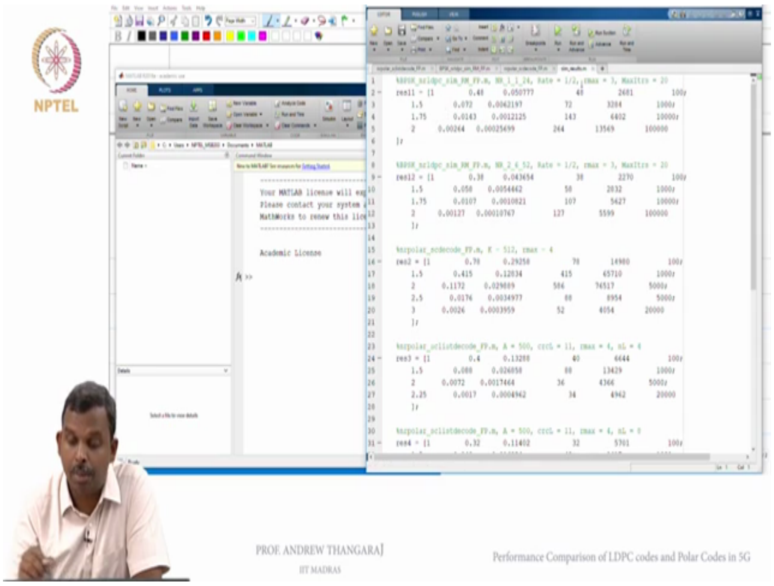
But then when you do sigma calculation you have to be very careful, remember K is the rate actual rate is not is K by N minus 2Z, so actual number of message bits divided by number of code word bits transmitted, so this N minus 2Z is quit important I am taking N to be the total length so N minus 2Z is the actual rate that you are going to use. Then after that things continue as it is there is no big change here. So maximum iterations we will take as 20, it is not a number for comparison.

So if you go to list decoding I have done similar setup here I have put A as 500, the CRC length is 11 and I have a polynomial here this is taken from the 5G standard itself and the one parameter here is list size you can vary this 4 is something you can consider, 8 also is something that one might want to consider, okay. Then of course you also have these quantization parameters, so you can put 3 here and 31 here this is 6 bits with between minus 4 to 4 and if you look at the LDPC case in the LDPC case I would put R max is 3 and max qr is 31, so I am taking minus 3 to 3 and the 6 bit quantization, okay so these are all choices so one can imagine there is lot of optimization here over how you do the quantization, how many iterations you do, what is the list size, etc. So one needs to compare and be happy about what you are doing.

So once you decide on these parameters what do you do? There is also SC decode there is no major choice here rather than quantization so one can do that also, so this is something that is done, okay.

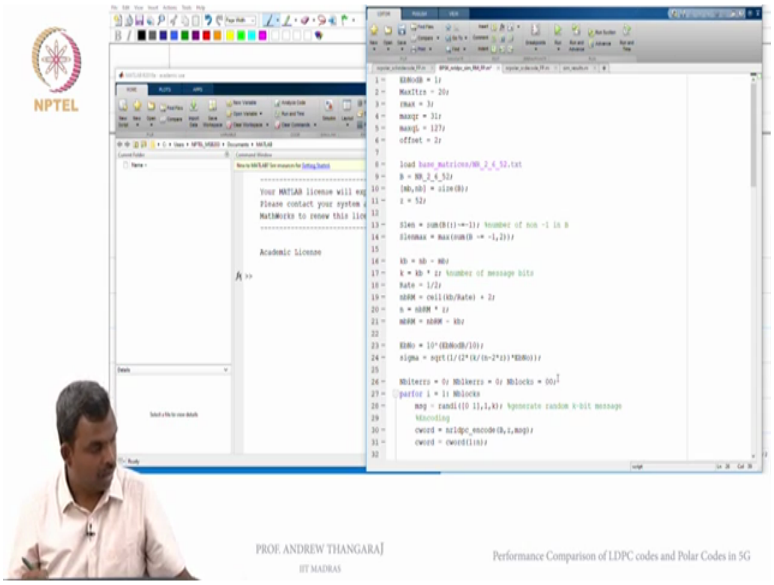
(Refer Slide Time: 10:16)

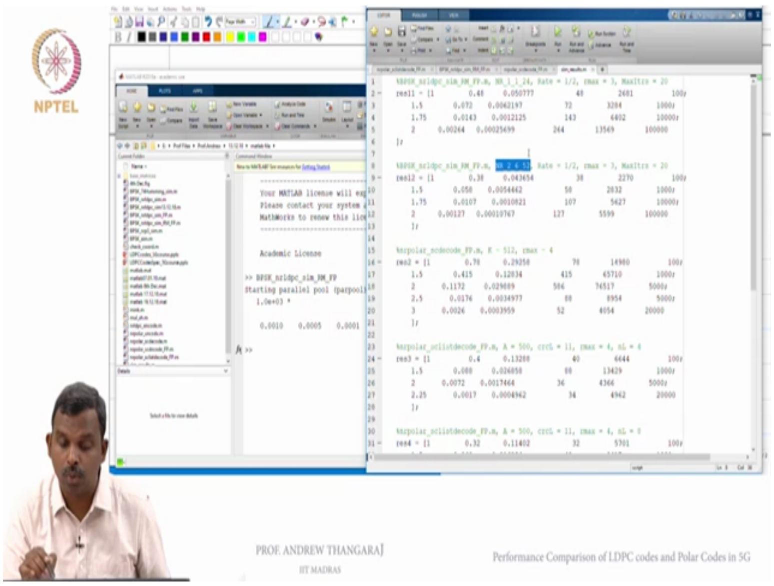




Now quick word about simulations when you start simulating you need enough errors so here are some simulation results I have already done some simulations here and I compile the simulation results and you can see here this is so this is another MATLAB file where I like to keep all my simulation results. So if you remember what comes out at the end of the simulation, what we are printing is eb over n not FER simulated, DER simulated, number of block errors, number of bit errors, number of blocks. So this is what we print at the end of the simulation so we get those values I collect and put them into a matrix like this, okay.

(Refer Slide Time: 10:48)



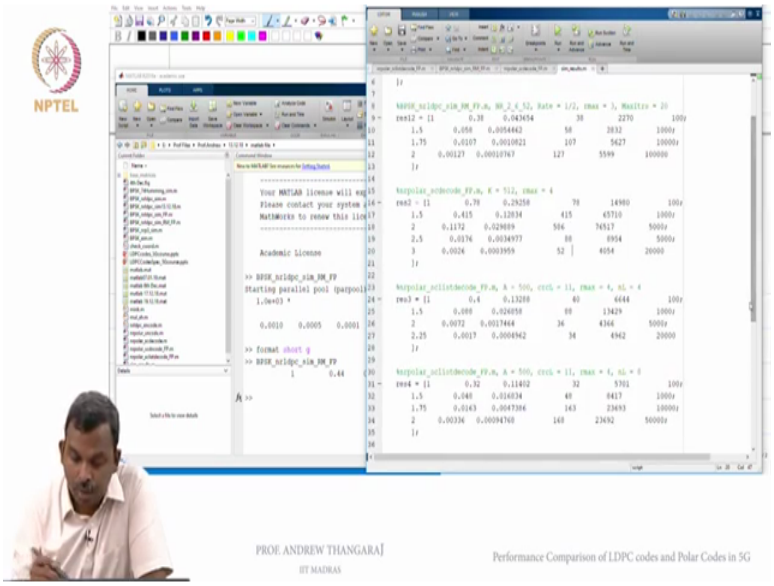


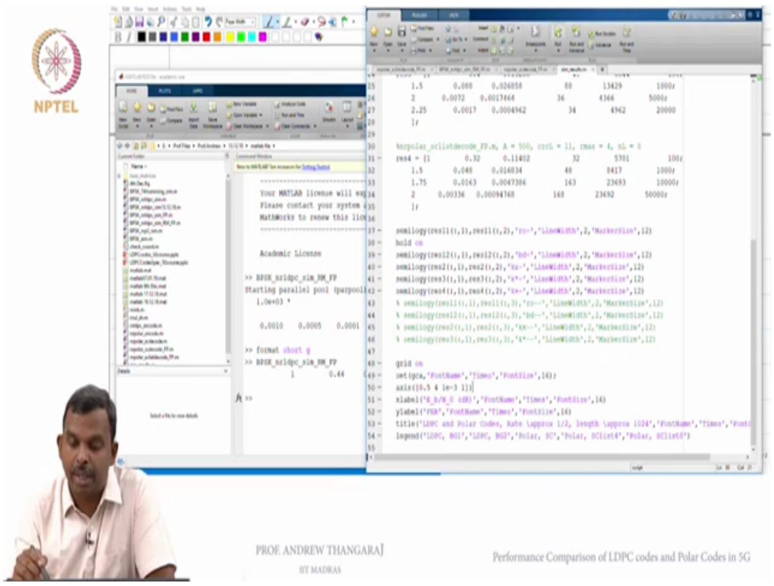
So for instance if you want to see what happens here this is let me do for a small SNR like row 1 which is only 100 blocks it will finish a little faster. So if you do that if you change this to 1 dB then the number of blocks near is 100 then I can go ahead and run, but you will see notice one small change here so I am doing this (())(11:16) for this is parallel for loop, if you have the parallel completing tool box of MATLAB this is very useful this makes this old for loop run in multiple threads with different values.

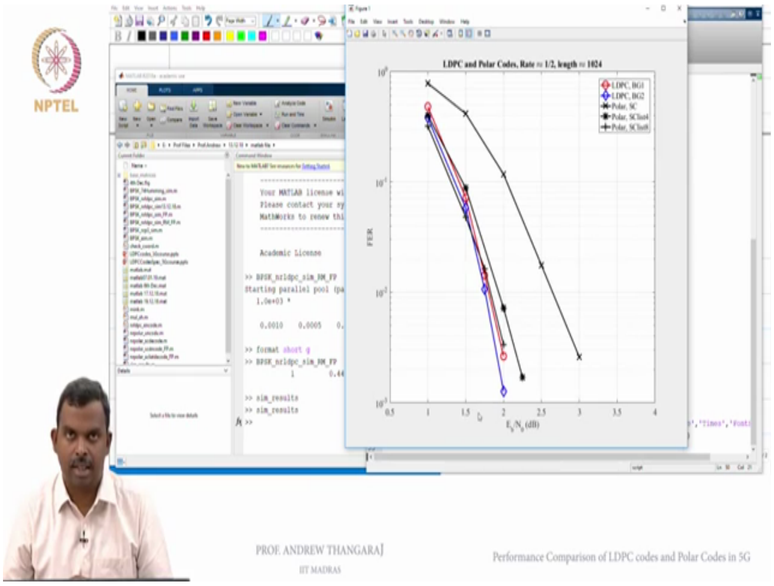
So there are some conditions under which parallelization is possible mostly it will work we can just put more variables inside the loop if you are not getting things to work then it will work and this will run as many times as (())(11:41) parallel processes you have it will work very fast. So (())(11:43) is a good thing to do to speedup simulations. So let us run this for 1 dB eb over n not and so it will start the parallel pole it will take a little bit of time but you can run this and it will work reasonably fast and you will get the answer so this is the NRBG 2 case and you can see how I have taken those values and put them here, okay.

So just to continue with what the simulation results is the first block is for BG1 and Z equals 24, rate half r max 3 and the next array that I have here is 2_6_52 again 20 iterations r max is 3 rate half and 2_6_52 so these both of them will have something similar.

(Refer Slide Time: 12:29)







So let us do format short g maybe you can run this again it ran pretty fast so we will run it again to get the access if we want. So I am expecting something of this sort so from the simulation run let us see we get that, there you go so it is 0.44 that is pretty close enough 0.38, 0.44 you will see in the plot it will not make too much of difference, okay. So similarly one can run the list decoder so you have the successive cancellation decoder for polar codes I have run that I have collected different values.

So now few comments on how I choose these values, so for 1 dB it is a low enough SNR I will run for only 100 blocks, so I get enough fair as I get some 78 errors I am okay. So if you go to 1.5 dB I am running for 1000 blocks only then I will get some 400 odd errors otherwise I cannot rely on the errors, remember number of block errors needs to be of the order of 40, 50, 60 or so to have some confidence in that value otherwise you know it is going to be very low you to worry mean, standard deviation, this, that all sorts of statistics under the picture.

So to get rid of all that you run long enough so that you get enough errors and then you do not have to worry too much about being in (())(13:41), okay and likewise for 2 dB you have to go to 5000, I have gone to 5000 which gave me about 500 errors which is very very good, for 2.5 dB again ran for 5000 I got 88 errors and a 3 dB I ran 20000 blocks and I got 52 errors so that is good so number of block errors is significant so these values are correct.

So you will see similarly for other cases also I have run multiple eb over n not's and different block sizes to get enough errors. Now this is very important, so you might wonder how do you choose which eb over n not to run from its trial and error you keep trying with 100 blocks, you keep trying from some 3 dB or so and keep reducing till you get some errors and then you slowly increase so that is the idea. So through trial and error we will quickly get it or if you have some idea on how the performance will be you can start at that point.

So for the successive cancellation list decoding I have done two cases so the first case is NL equals 4 number of list size is 4, the other case is list size is 8. So again I have done simulation for different SNR's and I have plotted and I have gathered all the results to get it, so I am not showing you more simulation runs but this is how I do when calculate I simulate gather the data and put them all into one file.

So nice thing once you put them all into as arrays in one file I have collected them as arrays in one file is you can start plotting. And for plotting this is a typical command that I have used, so maybe I will take this command here that is res 4 so that we can associate with the matrix here. So I like to use the semi log y, so semi log y keeps the x axis in linear scale, remember the x axis is already in eb over n not, it is going to be eb over n not so while it is logarithmic in some sense but it is only linear in eb over n not, so 1, 1.5, 1.75, 2 you wanted to be in linear scale.

And then the y axis you want to put in log scale so semi log y is a nice command to use and you can see I am taking the array res 4 of colon comma 1 which is the first column of res 4 and res 4 of colon comma 2 which is second column of res 4, so I am plotting the first column versus the second column, okay and there are some parameters here I am arranging for what type of line there should be, what type of marker should be, what colour should it be K is actually black plus means plus marker will come and then dash means continuous solid line, okay.

And then I pick these lines with because MATLAB's default line widths and marker size is a way too small, so I will like to keep a larger line width and larger marker size so that the plot comes out quite well and you will see here also I have turned hold on, so hold on makes all the plots appear in the same figure, grid on gives you the grid and then I will set a lot of properties mostly this is got to do with making the fonts bigger again MATLAB's default font is very small so I will like to make it bigger then setting the access and all that.

So let us run this so we have done all the simulations when you want to plot you do not have to redo the simulations, the data is there and you can readymade get the answer. So I think something very bad happened there so let me run it, okay so there comes the plot. So let me make it bigger for you so you can see the plot here so I have put a heading LDPC and polar codes, rate is roughly half around half, length is about 1024, I am plotting the frame error rate versus eb over n not, you can also plot the bit error rate, the frame error rate is quite nice, okay.

On the right most side you have this plot here and that is the polar code, okay the x mark is coming here the polar code successive cancellation decoder, so you can see polar code successive cancellation decoder and that is the plot here (so this is let me see if it will let me write anything here okay so it is not letting me write) so I cannot show you what this is but you can see what it is, so this x mark on the right most is the polar code with successive cancellation and it is the poorest among all of them, it is on the right most so 3 dB only it comes to around 10 power minus 3 frame error rate, so 10 power minus 3 frame error rate is a good target to have for simulations it is not too bad.

So if you do polar code with successive cancellation decoding list 4 you improve significantly so this star sort of plot mark is that one, you improve significantly around 2.1 dB or so you are getting 10 power minus 3 and if you increase the list size to 8 which is the plus mark here it becomes more to the left and it is pretty good at that point, okay also shown is the LDPC with base graph 1 and LDPC with base graph 2 and these are also quite comparable at that block length, maybe slightly to the left of polar but this has got to do maybe with choice of parameters, quantization, this, that, this is not really that much of a difference here it is within a fraction of a dB, okay.

So you see the polar code and the LDPC code are pretty good at these comparable block lengths and they are quite nice when you quantize them and simulate etc it works quite well, okay. So this is a typical picture that you mean if you if you were to write a paper about error control codes to write a report you have to include a plot like this, so you take the different codes you want to compare, make them sort of comparable, then run the decoders, (calculate the simulation) collect the simulation results together then make a plot and compare and show how it looks.

So we have done that so this is what we wanted to do in this code and to briefly summarize what we wanted to cover as far as this course on LDPC polar codes in 5G is concerned we wanted to study LDPC codes and polar codes in the 5G standard we did that from an implementation point of view how to get the encoder implemented, how to get the decoder implemented, we wrote MATLAB code for both of those, we did the simulations, here we go finally, we have compared and we know how these things work, okay along the lines you learnt about fix point decoding, how to implement how to show simulations of that, how to do this rate matching a little bit and quite a few ideas that are there in the 5G standard.

So there are many things we did not do there are in the 5G standards but we did not do, so I want to be very clear on that so you will see if you go read the standard description you will see quite a few things we did not do particularly in the polar code there is some interleaving that is needed because of the sequential decoding if there is a deep fade the polar code will fail, so to make that get over that there is some interleaving that is needed in the polar code, the LDPC code does not need an interleaving so that is one added thing we did not do at all and for very short block lengths to help the successive cancellation decoder for the polar code there is in the standard they add some 1 or 2 parity check bits so that part also was not touched upon at all in this and then also there is for some cases interleaving of the CRC bits so that you can use the early CRC's to direct your sequential decoder in the direction you want, okay.

So all of those things we did not include in our simulations they are not too difficult I will encourage you to read those things on your own and think about how you might want to implement, okay anyway thanks a lot for staying through in this course till the end and this is the last lecture and all the best for your future studies, thank you very much.