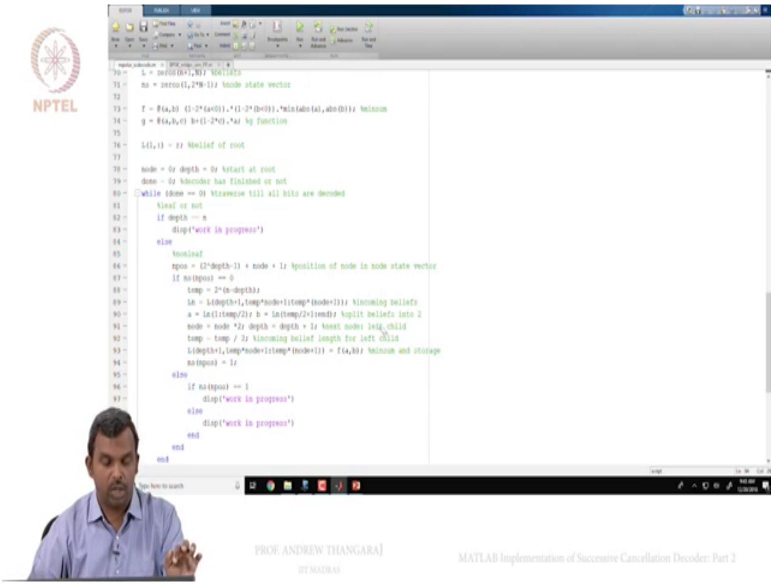
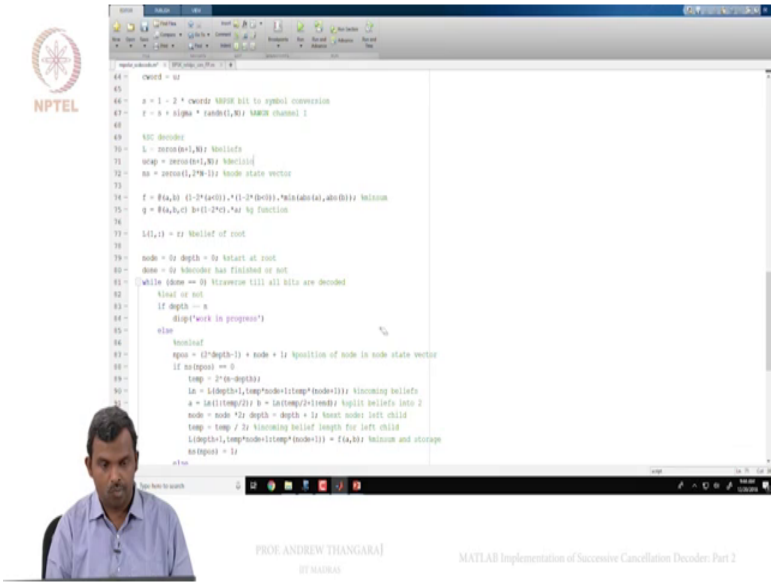
LDPC & Polar codes in 5G Standard
Prof. Andrew Thangaraj
Department of Electrical Engineering
Indian Institute of Technology Madras
MATLAB Implementation of Successive Cancellation Decoder Part 2

(Refer Slide Time: 0:16)



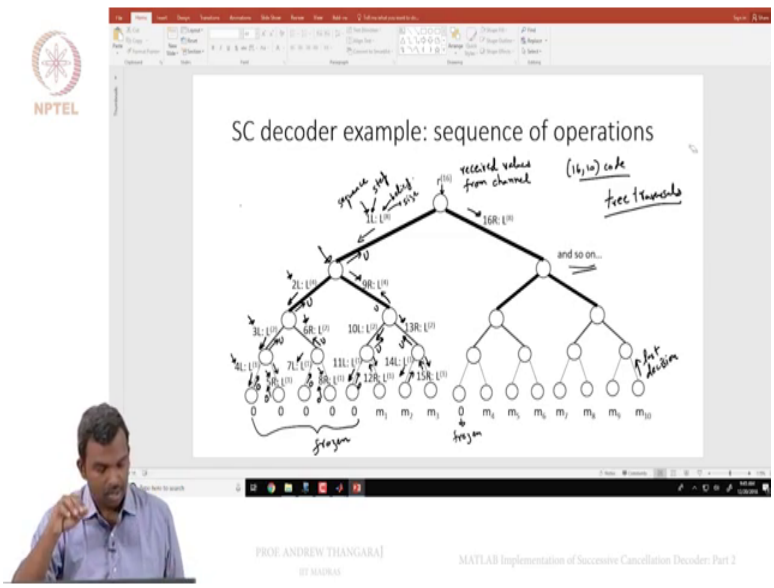
Okay let us continue our coding and you see what the coding involves right, so you have the storage and you have to pull it from storage, pull what you need from storage to the processing and push it back in to the storage okay, so that is how the storage operates, I need to at just this lengths and all that is 2 power N minus depth and all that interest a picture, that is not very difficult to do as we can see, of course there are many more efficient ways in which this can be done, I am not focusing on that okay.

(Refer Slide Time: 0:45)



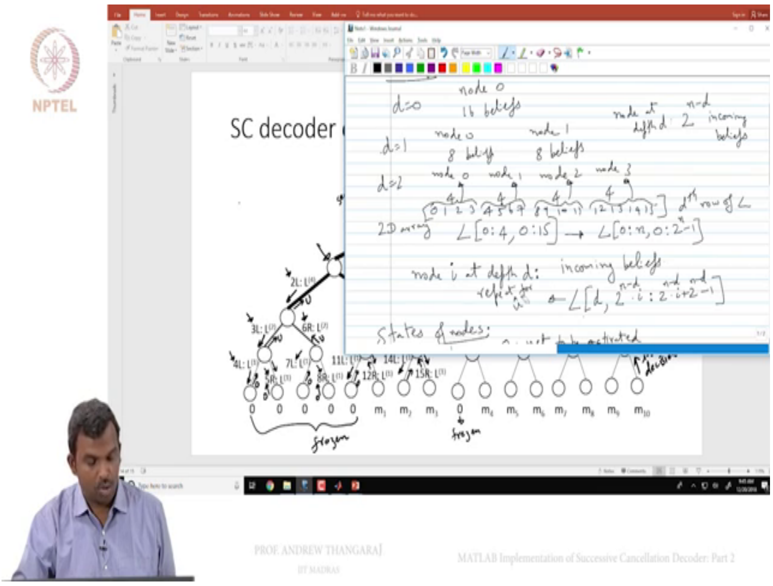
So next is then the state is at, when the node state is at 1 which means the left is already done okay and it is receiving the decision okay, so if you remember this U will flow back also okay and one needs to remember that also okay so because that is important, I did not have the storage for the U cap and that is also the same as this okay, so beliefs and then I have the U cap also okay.

(Refer Slide Time: 1:25)



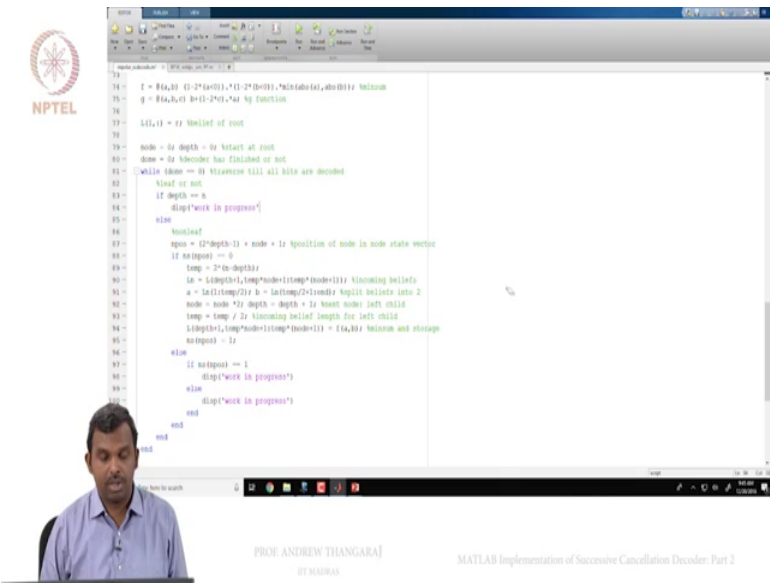
So once again let me emphasise where this is coming from okay, so this is the decoder, you remember this decoder right, so we stored the incoming beliefs, now what about the decisions that are flowing back? Those also need to be stored okay and you have to restored them and then process them later also, so you have to keep them, so for the decisions also will use the same sort of a array structure as we used for L okay, we had L is a big array which stores all the beliefs, likewise will have U cap as the array which stores all the decisions that are flowing back okay and the position is exactly the same as L, wherever you computed the Ls position the same place you will put the U position also okay, so that is the idea.

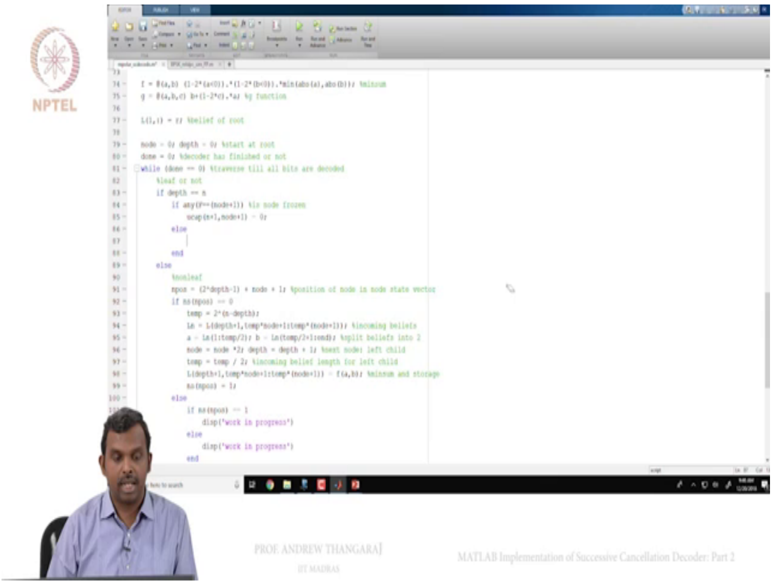
(Refer Slide Time: 2:12)



So maybe I go back to this thing and show you how I organise the L, you remember how we organise L right, so it is the same way will organise the U cap also, the same way in which, so the same way U cap also okay, repeat for U cap okay, so that is what I am going to do, so let me not spend too much time on it we can start coding.

(Refer Slide Time: 2:35)

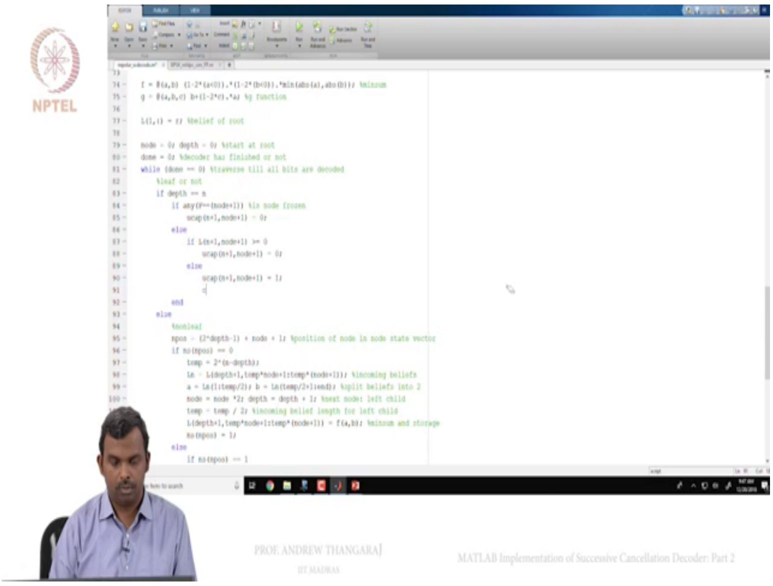


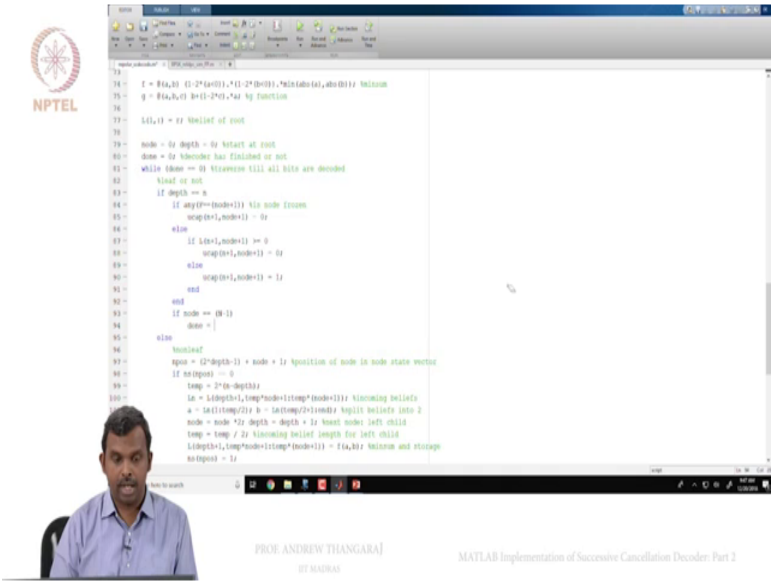


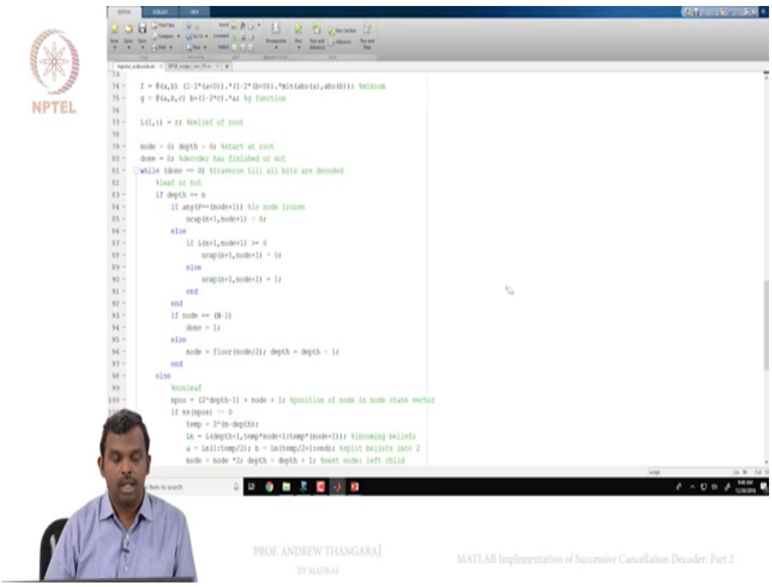
So U cap is like that okay, so U cap comes into operation when you either make a decision at the leaf right of you do a step U or even a step R right, even in step R the incoming U cap from the child node is used okay, so maybe I will start with what happens at depth N and how U cap is used here okay, so at depth N I am going to check if node plus 1 is frozen or not okay, so how do I check if node plus 1 is frozen, there is a command you can use here any of F equals equals node plus 1 okay.

So this is node frozen, remember I am at depth N and I am at bottommost thing okay, so I am at the bottom most, the bottommost is easy to address the position in the U cap array, if it is frozen, so I will just explicitly store U cap to be 1 node plus 1 equals 0 okay, so in the bottommost you have the entire length coming in okay and then every node is just one position away, you do not have to do any calculation for the position, that is one thing nice.

(Refer Slide Time: 4:03)



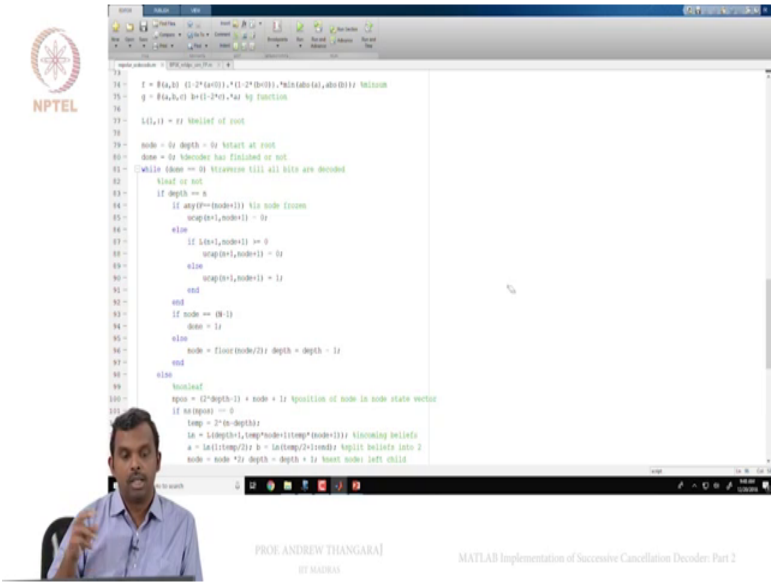




So here if, so this is the decision right, so if L of N plus 1, node plus 1 is greater than 0, greater and equal to 0, U cap of N plus 1, node plus 1 equals to 0 else U cap of N plus 1, node plus 1 equal to 1 that is, so that is the decision okay, so once you finish the decision, you are not done, you have to handover to the parent right, so how do you hand over to the parent, I do node equals.

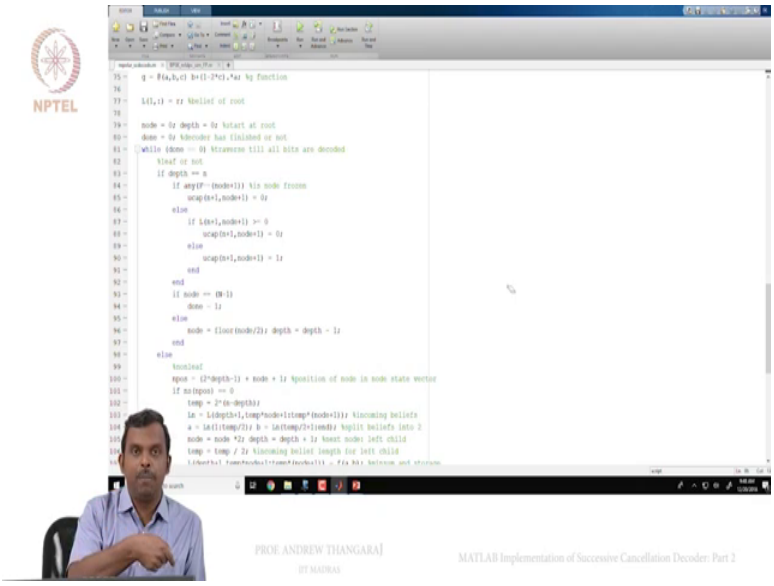
So one thing you can do is you can check if this is this is the last one, if the node equal to equal to N minus 1, so if you have made a decision on the last node, you can set done equals 1 else if you are not done the last one you can make node as floor of node by 2 okay, so this is my flooring and then depth is depth minus 1, so you proceed further okay, so this is going of to the next one.

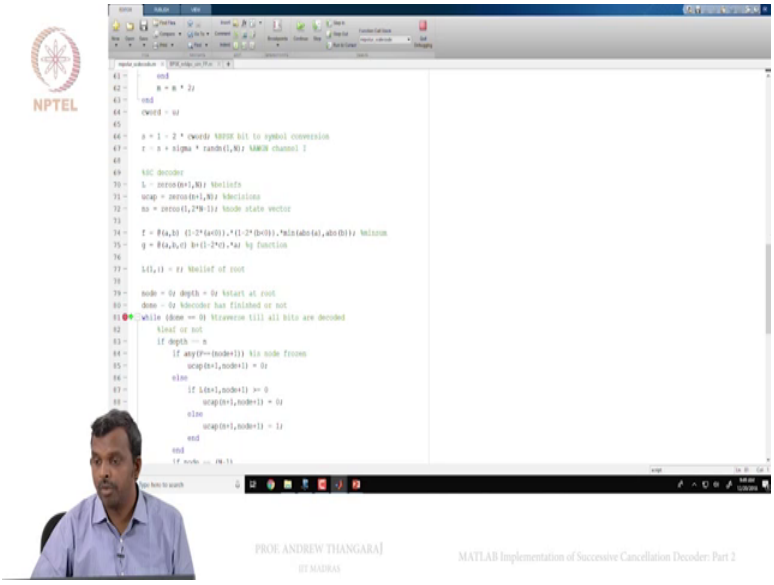
(Refer Slide Time: 5:24)



I am not going to bother too much about the changing the state of the leaf node okay because it is not so important, once you get there you design and you go on okay, so this is what the leaf node does? Once you come to the leaf node it will go through okay.

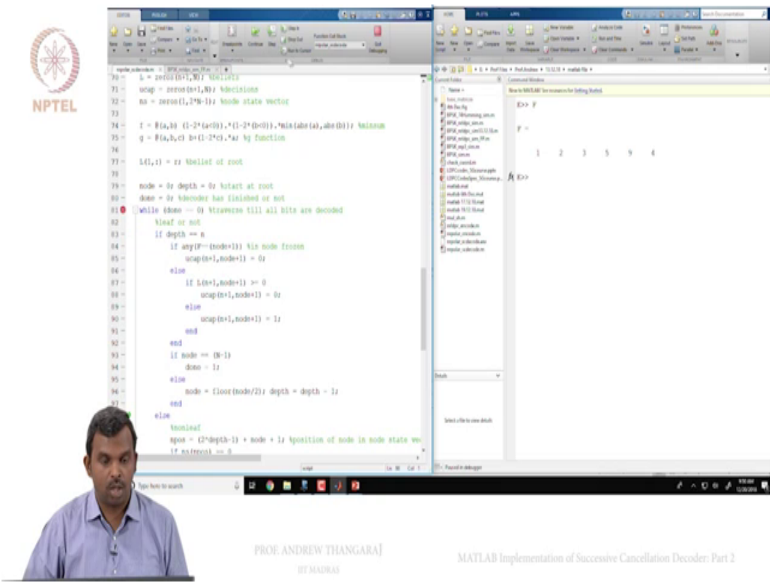
(Refer Slide Time: 5:40)

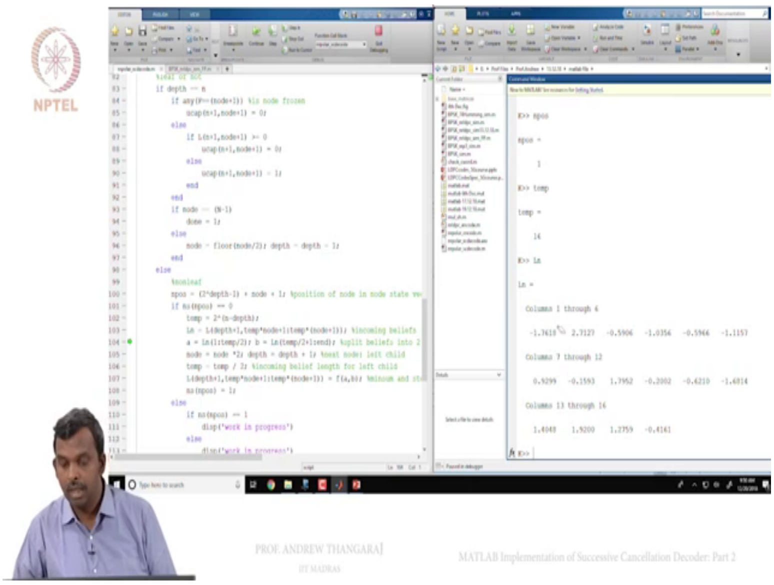


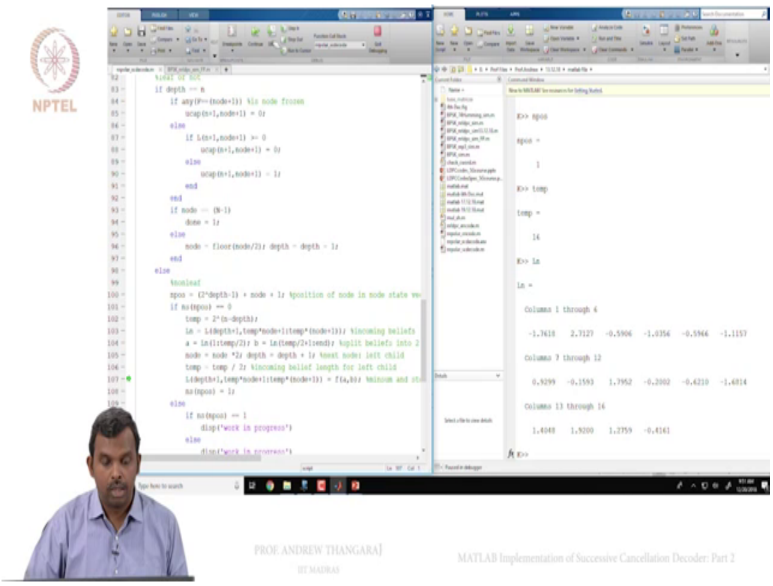


So in fact we have done enough so that even simulate and see whether things are happening correctly or wrongly, it is not too bad at this point I believe you can go up to some point because remember you do a lot of down steps initially and then you make a decision okay and then we can avoid the work in progress quite a bit if you do this, so let us, I think it is good at this point to run it and then see what happens, so let me come up with this point and set up a breakpoint here hopefully it will run and let us run okay when you run it at debugger mode we have to go back to the okay.

(Refer Slide Time: 6:19)



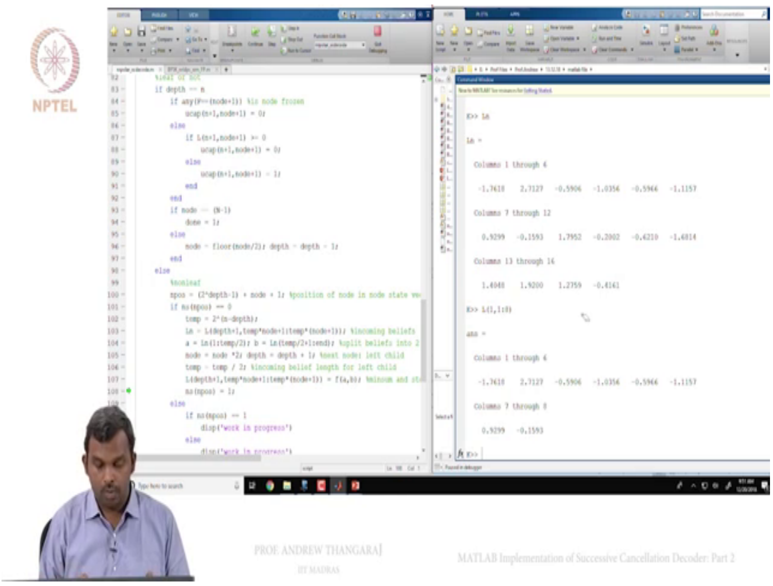


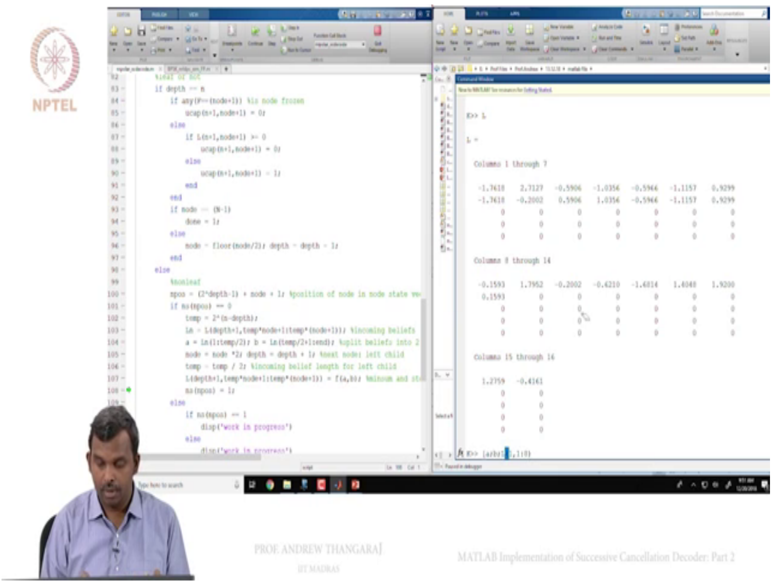


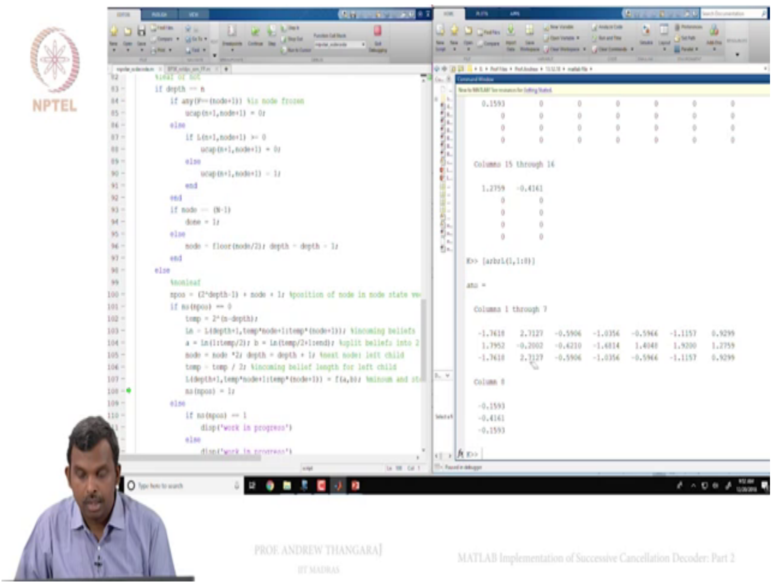
So we have done and this is started, remember our starting point was this is N equals 16 right, yes 16 and 10, so maybe we can see the frozen positions just to be sure, 1, 2, 3, 5, 9, 4 it is okay, so that is the frozen positions, so let us that going in okay, so we step here and the node and depth they should not happen is not a leaf okay and then it is calculate the position, depth in 0, node is 1 so N Poss should work out as 1, so this will be to okay anyway so I check it out N Poss is 1 that is okay and NS of N Poss will be 0, so no problem here and you compute temp, temp will be 16 okay and then depth the 0, so node 0, temp is 16.

So this will be 1 all the way up to 16 okay, so LN will be the same as L, so if we see LN it will have 16 values and this is the received values from the channel okay and then you store temp by 2, A and B, A will be the first part, B will be the second part and then you go to the next node, node is again 0 but depth is 1 okay and temp is become temp by 2 and then you do the storing of L okay.

(Refer Slide Time: 7:49)

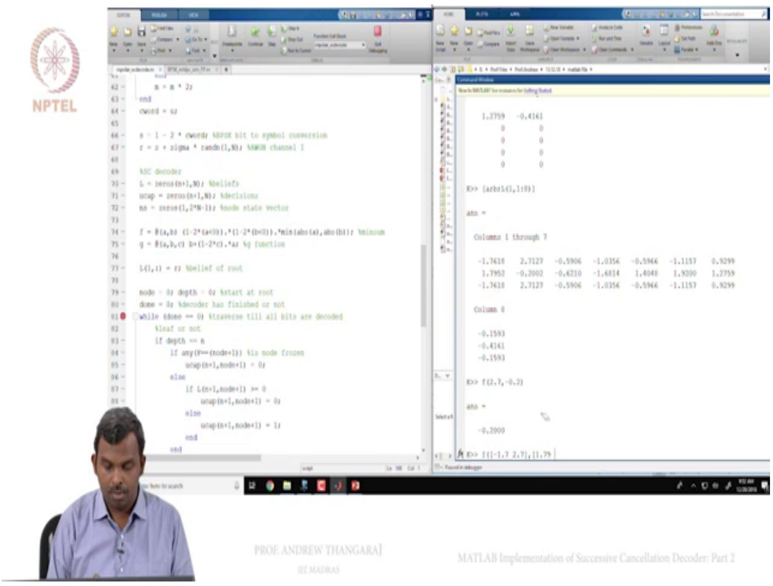


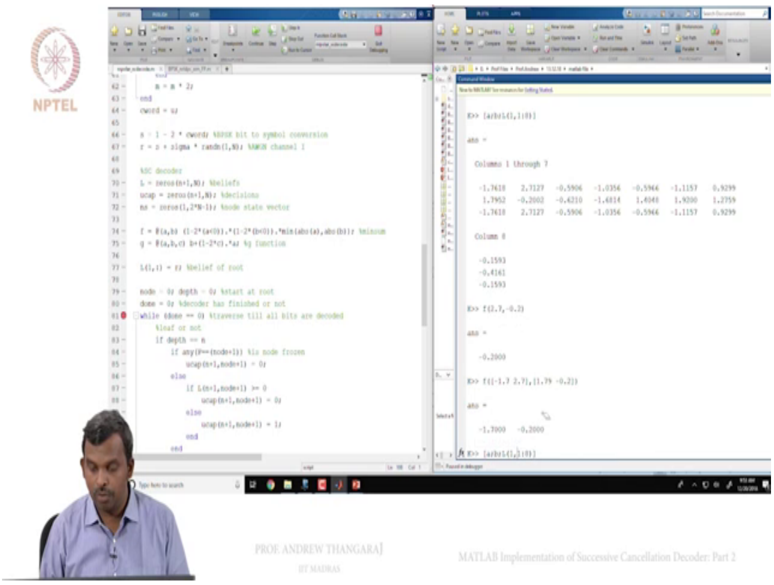


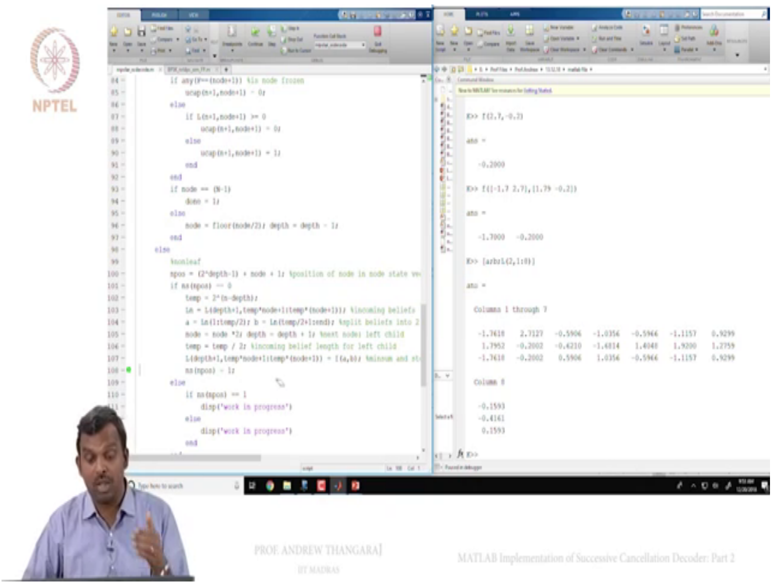


So let us see L of 1, 1 colon 8 right so this is what I will know, the remaining will actually be 0 and L, so we will just do this and how do you see this, so maybe we can see a little bit better, we do this, so we can see A, we can see B and then we can see L okay, maybe we can see all okay, so this became a bit ugly, so let us see A, B and then L of 1, 1 colon 8 okay, so you can see the minsum is working correctly, so you the minsum is working not so correctly, why is that? Okay so I think, so something is wrong with this minsum, I do know why A and B together should give you this because this minsum is supposed to be minus 0.2 is not it, so something slightly wrong in my F construction.

(Refer Slide Time: 8:59)

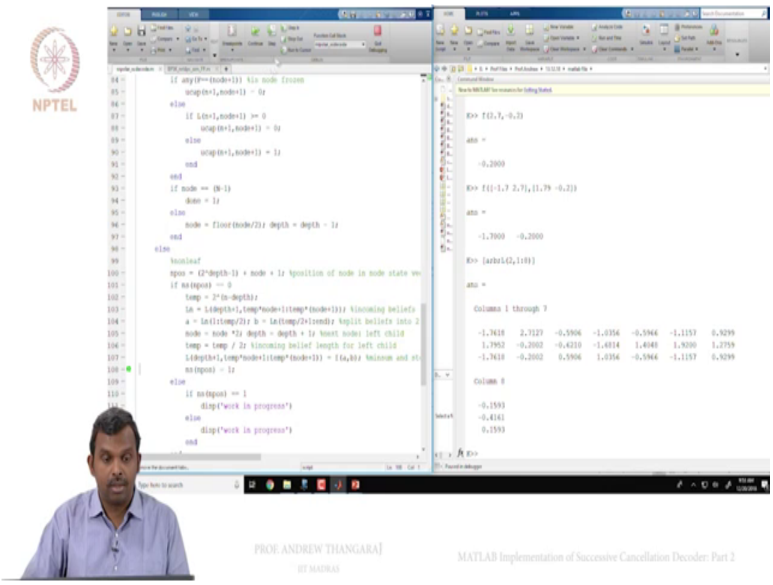






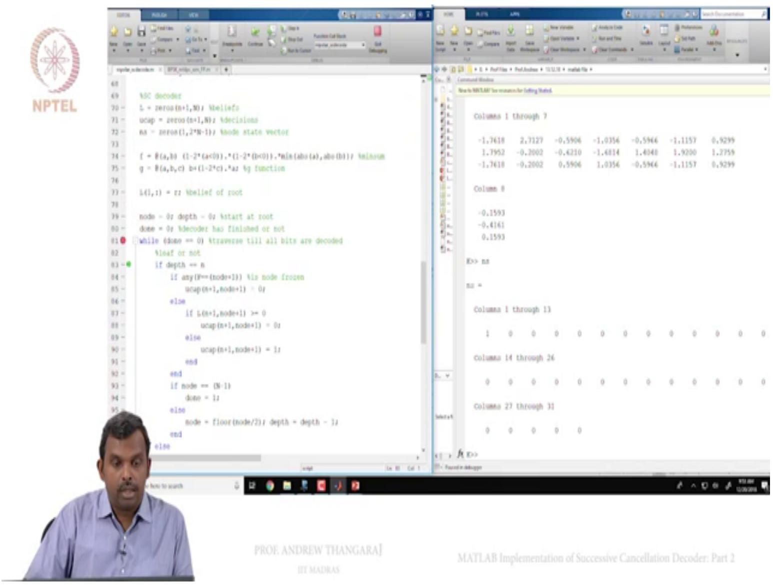
So let me just check this out, so if I do F of 2.7 and minus 0.2, I should get minus 0.2 but that seems to be working okay but F of let me check this minus 1.7, 2.7, 1.79 minus 0.2 let see, this is also working out okay oh okay, so I think I know what mistake I made an error, this is L of 2, okay sorry okay, so I was right program was working correctly, I was just not checking the correct L, it needs to go to depth plus 1 right, so the next storage.

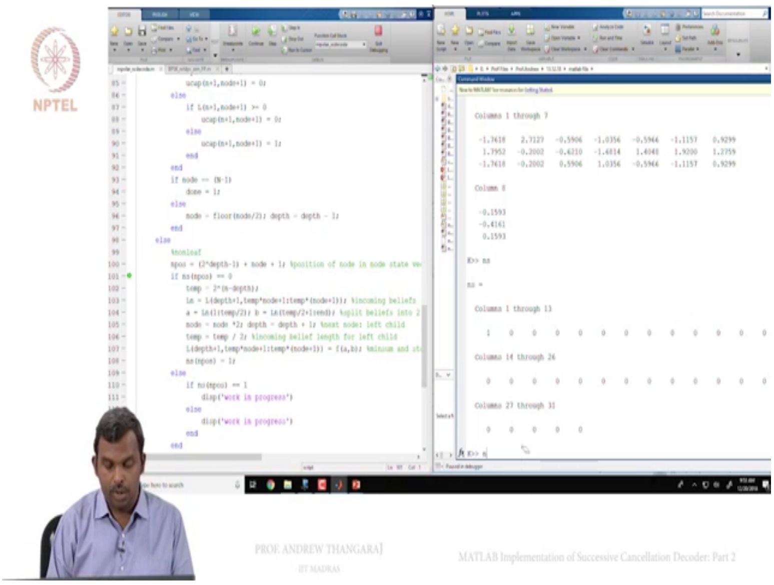
(Refer Slide Time: 9:56)

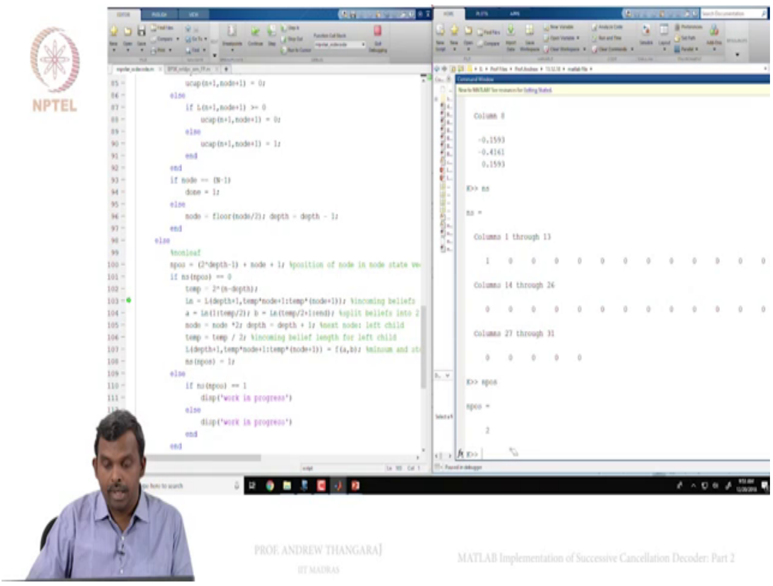


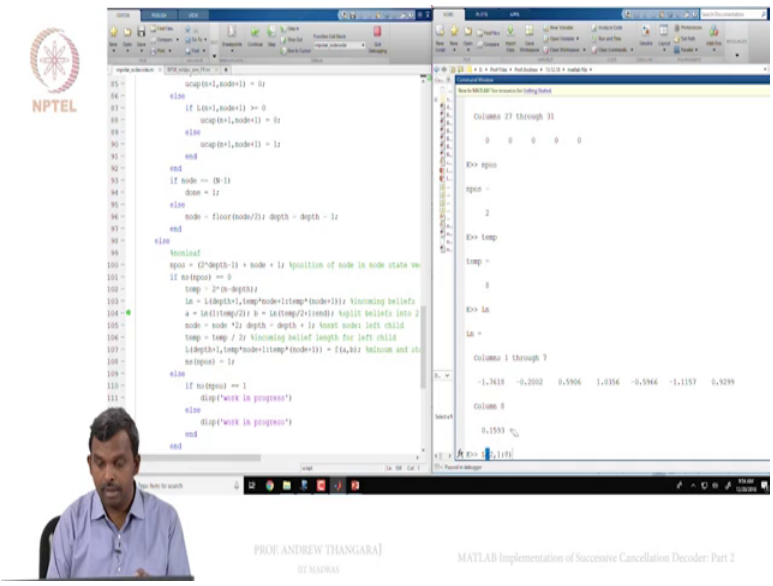
So incoming in the next node and that is coming out correctly okay, it is a minsum, you can see the product of the signs and the lower of the two values being retained here okay, so this is the belief here and the state goes into the next one okay, so there is the end, so if you want you can see the node states, we have big matrix but the first one alone is 1 okay.

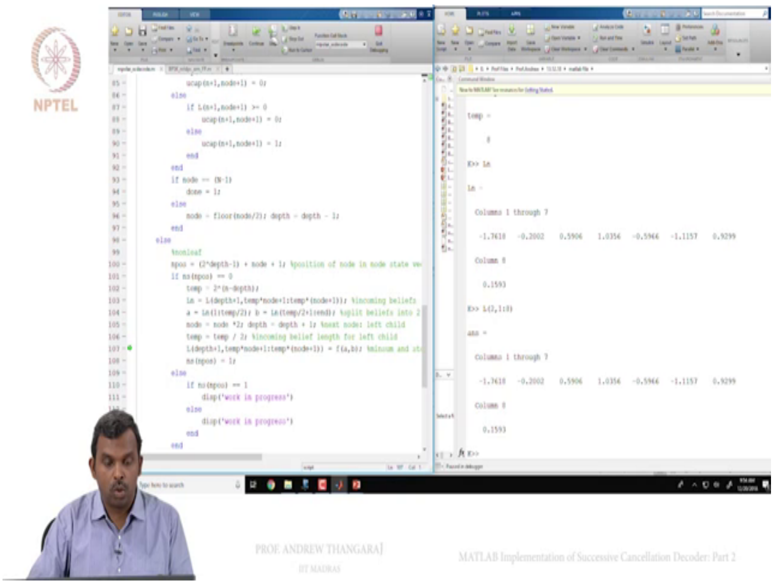
(Refer Slide Time: 10:14)




NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2

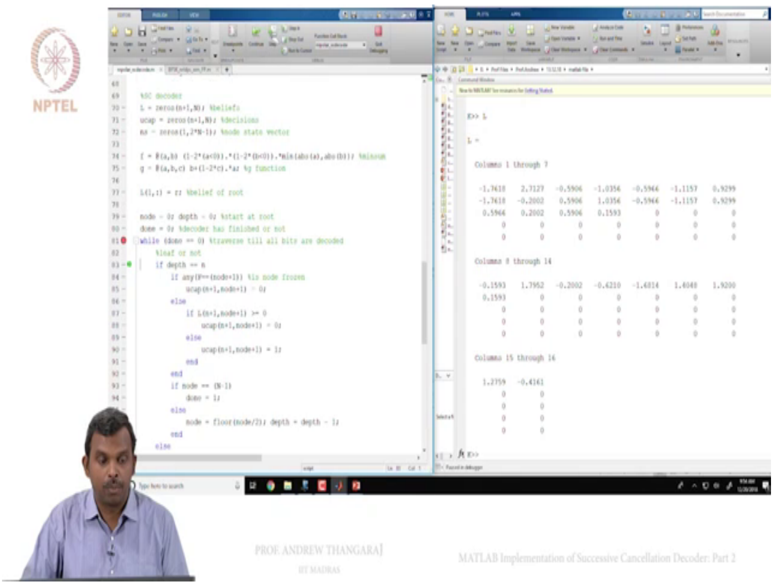

NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2


NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2



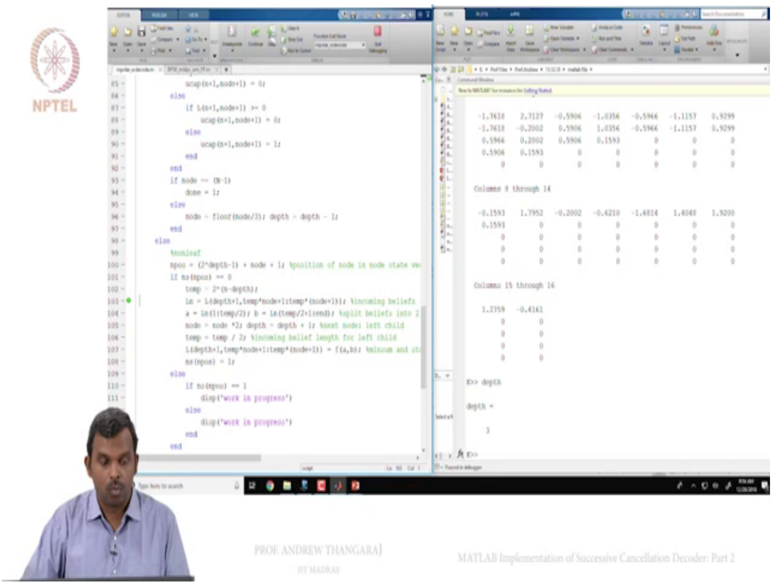
So the next one, in the next step we are again you do not have this but then if you look at N Poss, N Poss will be 2 I think okay, so that is the correct N Poss, the second position and then again it will be 0 okay, so again it will do this, this will work out hopefully correctly temp will be 8 okay and then LN will pulled out correctly the values that we had, you can see L of 2, 1 colon 8 and agrees with what it pulled out and then you do the same thing as before A, B and then you go to the next node, temp become 4 and then you do L okay and then you set it as N Poss.

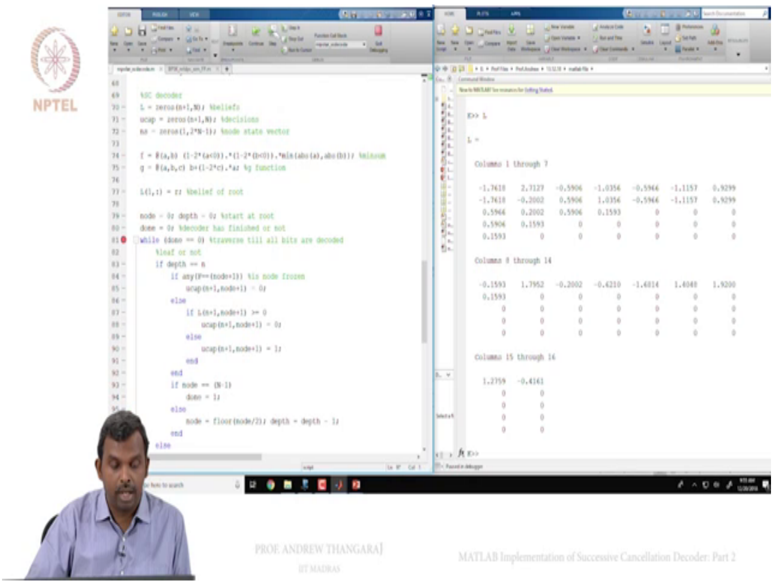
(Refer Slide Time: 11:01)



So if you now you get L will see this big matrix, initially there were 16 values then you have 8 values then you have 4 values okay and then this 4 values you will see are the minsum of this two and you will proceed like this, so you can go up to the leaf node actually.

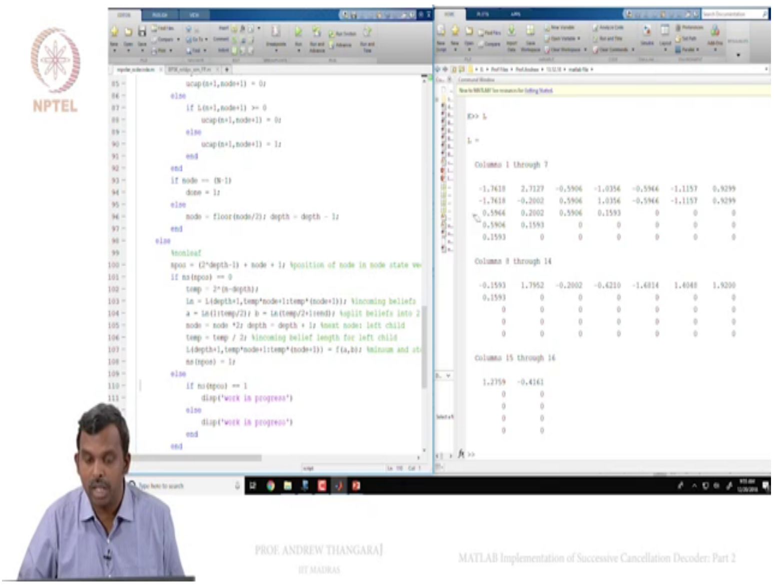
(Refer Slide Time: 11:19)





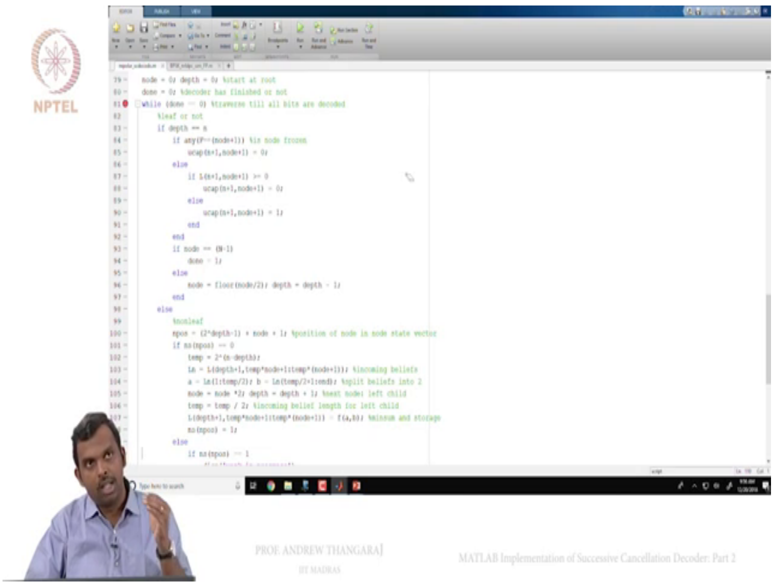
So you go to the next step again the same process continues okay, see L okay come up to this two, now are at the depth of 3, so we are almost at the leaf node, then we can do one more we are ready to leaf node now okay, so you can look at L, you see this triangular sort of structure right 16, 8, 4, 2, 1 and this is your belief for the U1 okay but interestingly U1 is I think frozen, so you if you have depth N it is frozen and then you totally ignored it but still it is okay and node is not N minus 1 and then you go to the next node, node is still 0 you have depth 1.

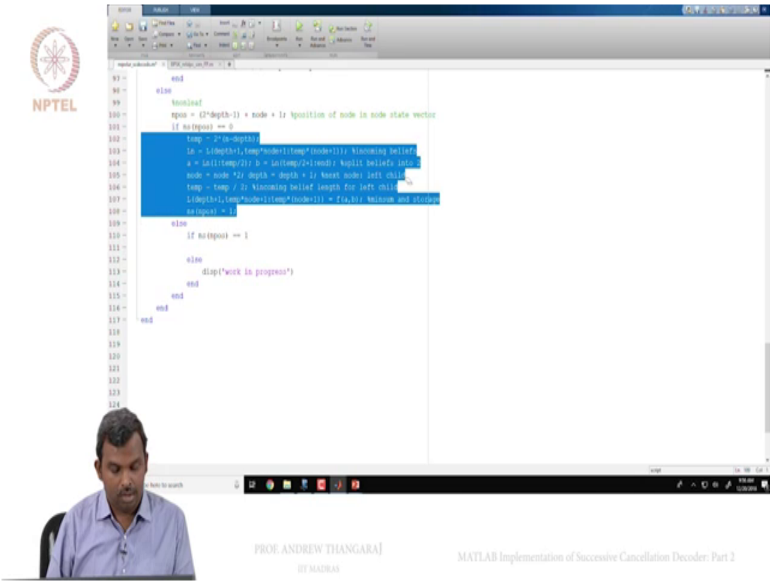
(Refer Slide Time: 12:14)



So you have come up to the next one depth higher okay and here if you do, you know it is not a leaf node okay, you got to the right position but now NS of N Poss will be 1 okay and we have not written code for it, so let us start writing for that okay, so hopefully you saw how the decoder was working, I would give you an idea of how this thing works, how the traversal happens.

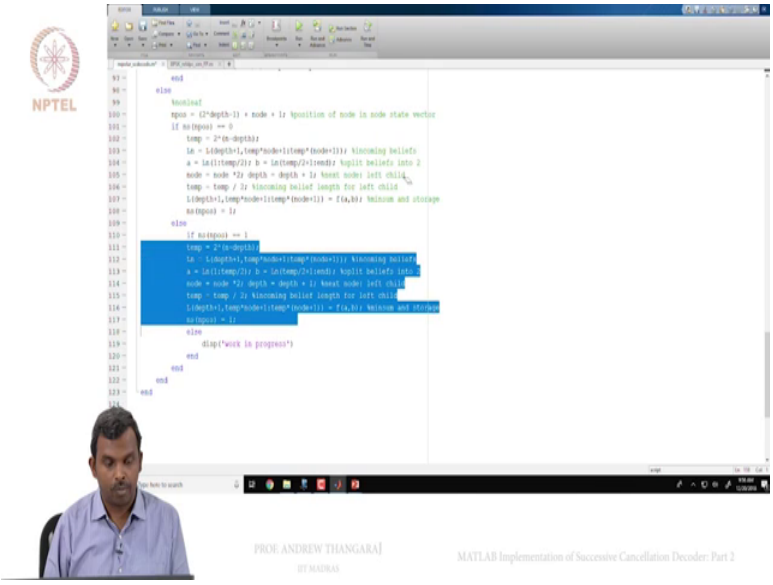
(Refer Slide Time: 12:42)

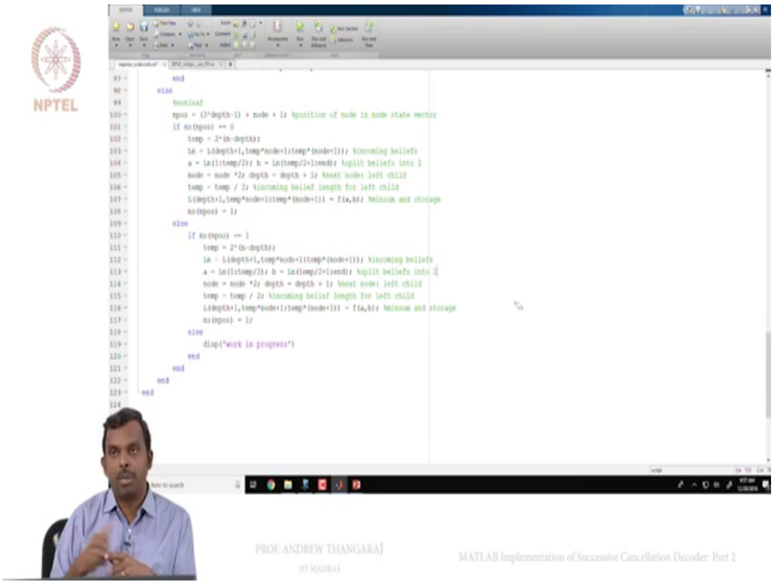




So keep that in mind, so it goes left left left left and then it made a decision and it wanted to go right but we have not written code for going right so let us try and do that now, so once you have figure out how to write for left, the Right is also not very difficult to write down, so I know this, so I will again start the same sort of logic, I need the incoming ones, I need the split okay, I need to know the next node, I need to know the next temp except for this everything else will be the same.

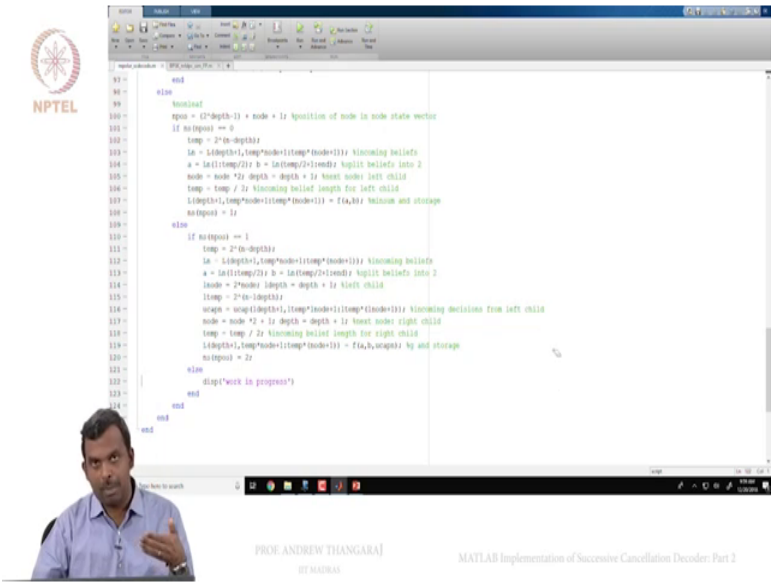
(Refer Slide Time: 13:22)





So I can do a cut and paste here and then modify okay, so you will see there are few changes we have to make here because the state is different, you also need like LN, you also need the, so we got the incoming beliefs, split and all that that is good but we also need the incoming message right, so I am in state 1 okay, so if I am in state 1 it means my left child is giving me something okay.

(Refer Slide Time: 14:00)

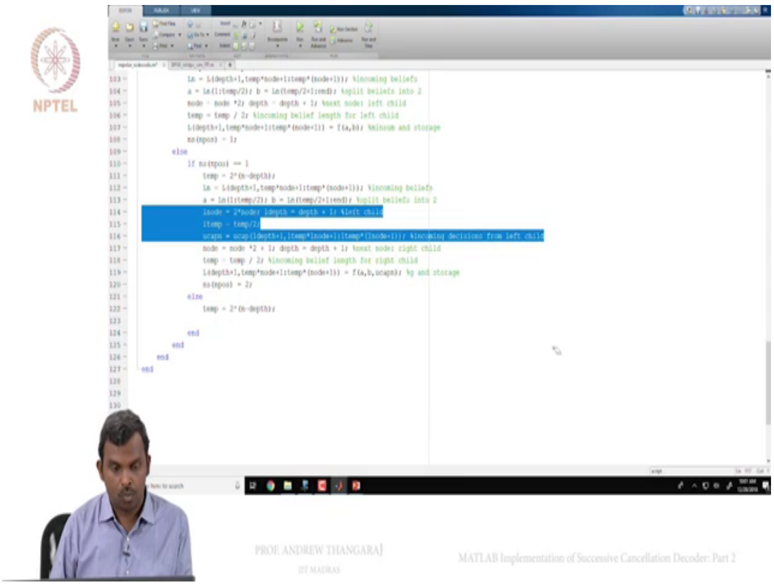


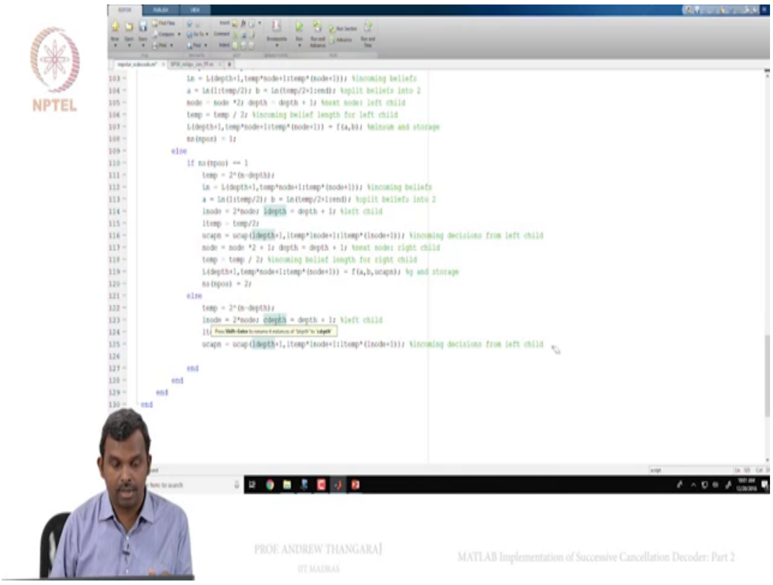
So you have to go to the left child and pick up the U cap from the left child okay, so that is what is important, the left child is U cap is what you have to pick up okay, so what is left child you are at the node and depth, so the left child is 2 into node and the left depth is depth plus 1 right, this is your left child coordinates so to speak okay and then once you do that you can have an L temp which is 2 times 2 power N minus L depth okay and then you can pick up U cap of the node okay the incoming message from the left child which will be U cap of L depth plus 1, L temp into L node plus 1 colon L temp into L node plus 1 okay.

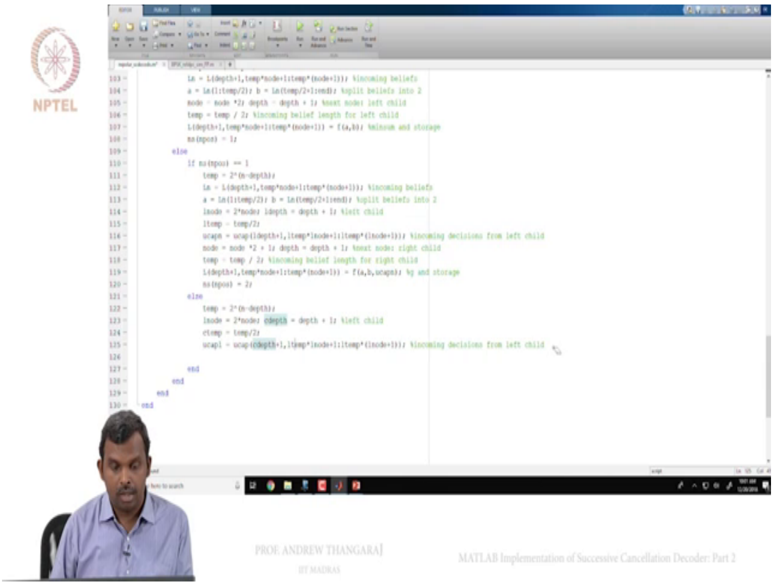
So this is incoming decisions from left child okay, so hopefully this is clear to you remember when you do the G function, you not only need the incoming beliefs you also need the incoming decisions from the left child okay, so you have to compute who your left child is, go access the corresponding storage for U cap then pull out the storage okay, so for some reason it is warning me that he did not do something correctly I think I did not close the brackets okay, so that is it.

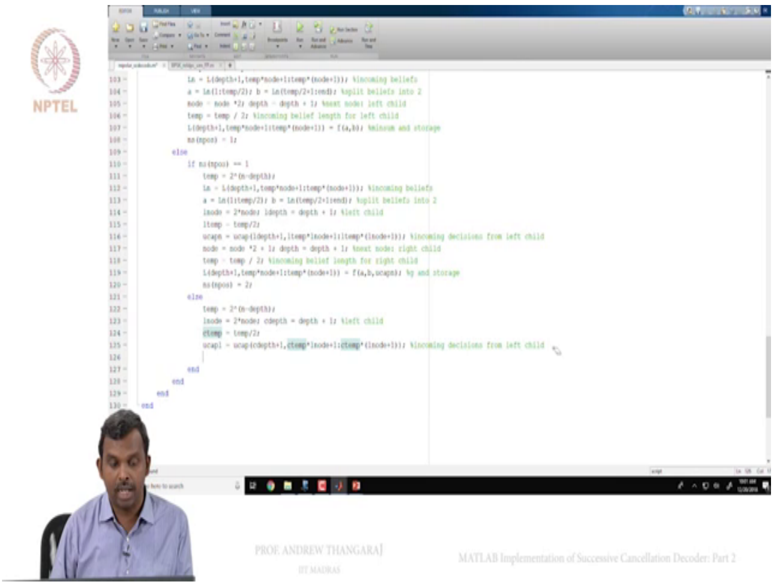
So once you do this, then you go to the next node, now next node has to be the right child okay, the right child is the node into node 2 plus 1 temp still goes down by 2 okay and then you go to the right child, so know you can do A B and then U cap okay, so this is G and storage and now you can make the NS stoop is that okay, so maybe little bit inefficient but I think if we managed to the right thing as well.

(Refer Slide Time: 16:37)




NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2

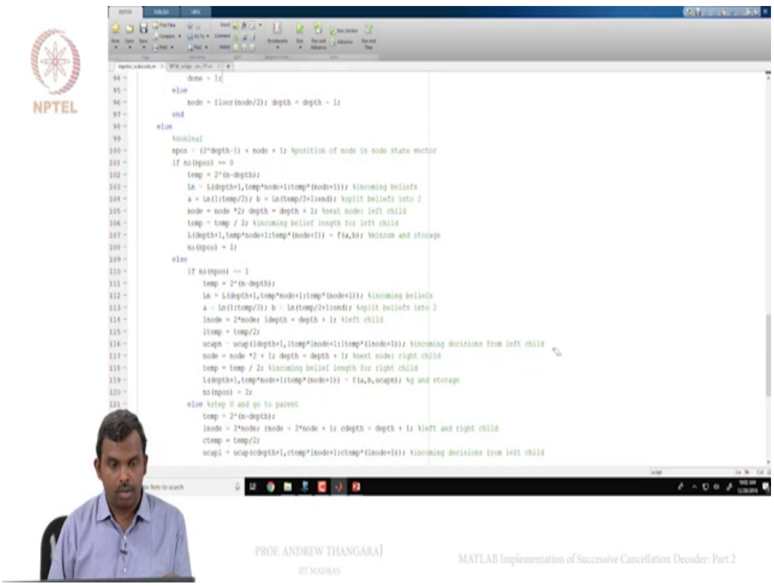

NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2

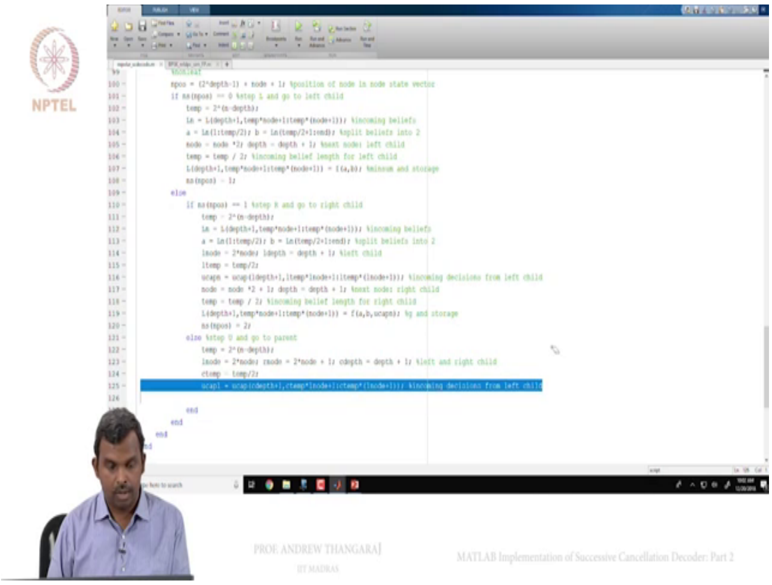

NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2

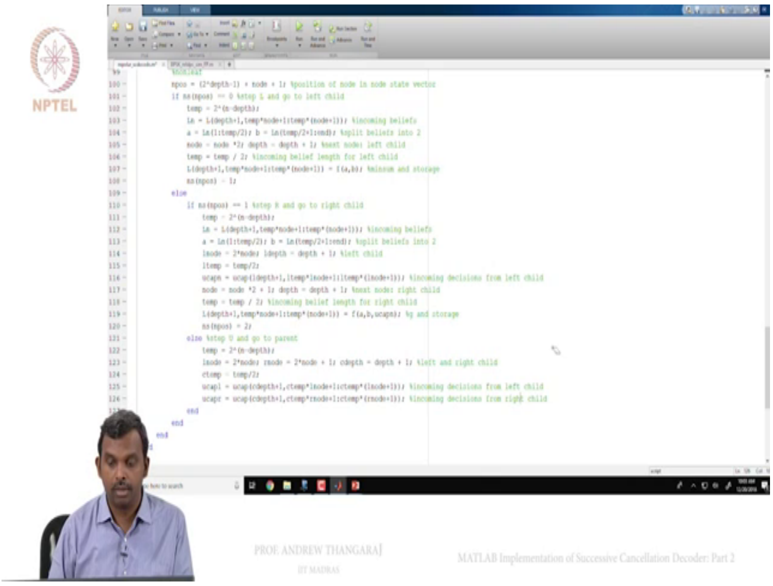
So let us code even the step U which is going back up to the parent, so the so far so good we have not done anythings too wrong, so let us do going up to the parent, going up to the parent is actually also very similar to this in some sense, so in fact here even for L temp I do not think I need, so L depth was just depth plus 1, I can simply say L temp is temp by 2 I think right, so when you go to the left child L temp becomes temp I do just to avoid a little bit of computation that okay.

Alright so this, so let us figure out what to do when NS states is 2 okay, so when state is 2 what you need to do is compute temp okay, so let us compute temp that is okay, you do that and then we do not need A or B but we need the U cap N from the left child okay left node is this, left depth is this okay, so I call it C depth because that is the childs depth and then C temp which is childs temp okay and then U cap left is U cap of C depth plus 1, C temp into L node plus 1, C temp into L node plus 1, so this is incoming decisions from left child.

(Refer Slide Time: 18:30)

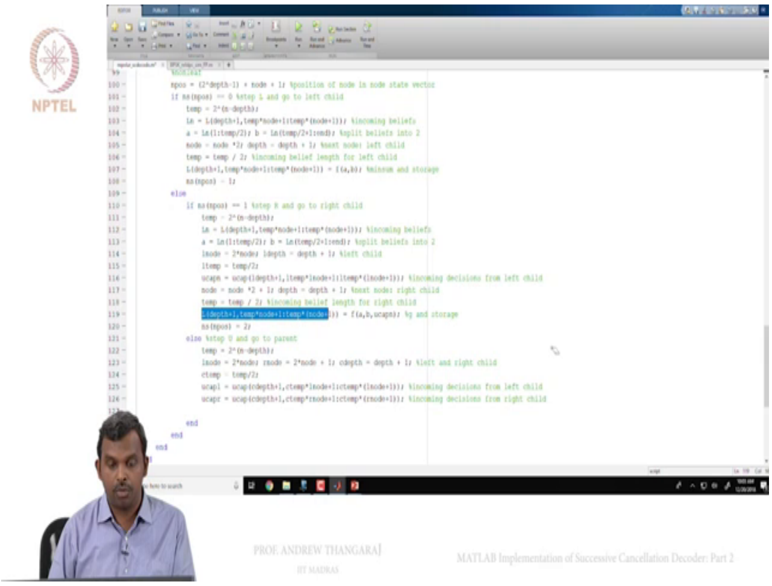


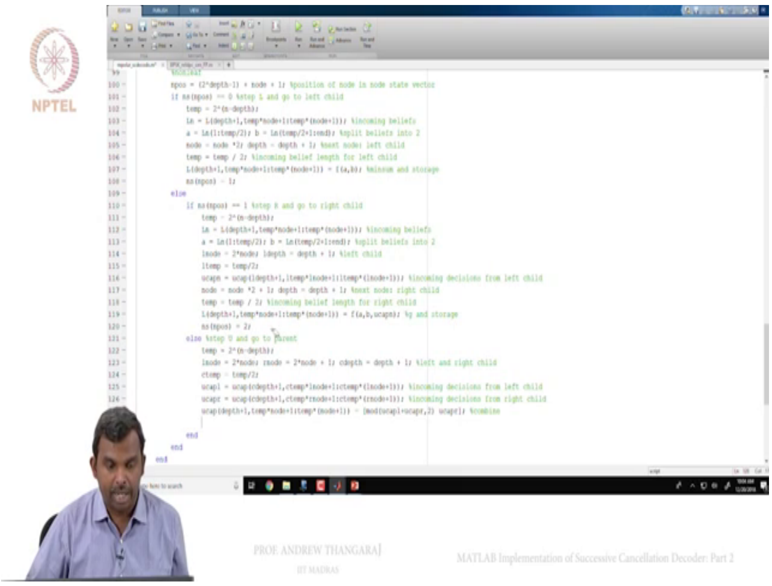


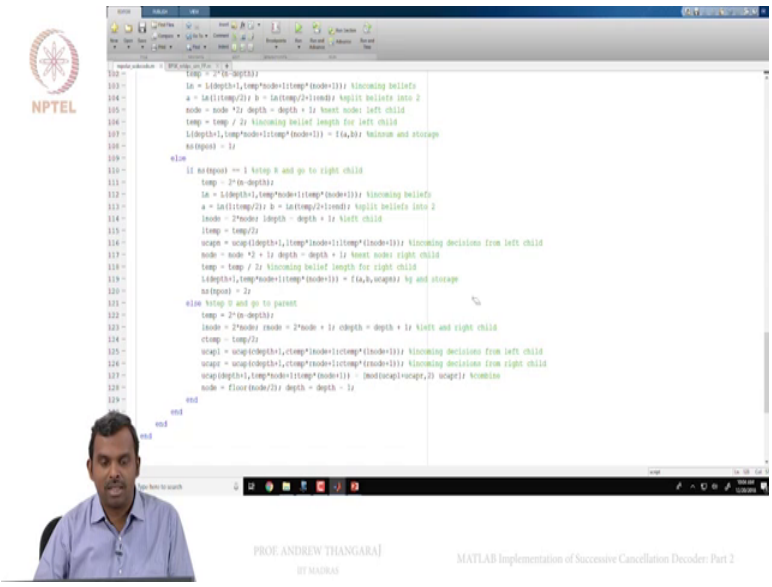


So I need U cap R for which I will need R node okay, so I will put R node also right, so you are, so save this is else I need to do step U okay step U and go to parent, remember this, so this is a step L, this is step R okay, so something very similar with U cap 2, U cap R which is R node incoming decisions from the right child okay.

(Refer Slide Time: 19:50)



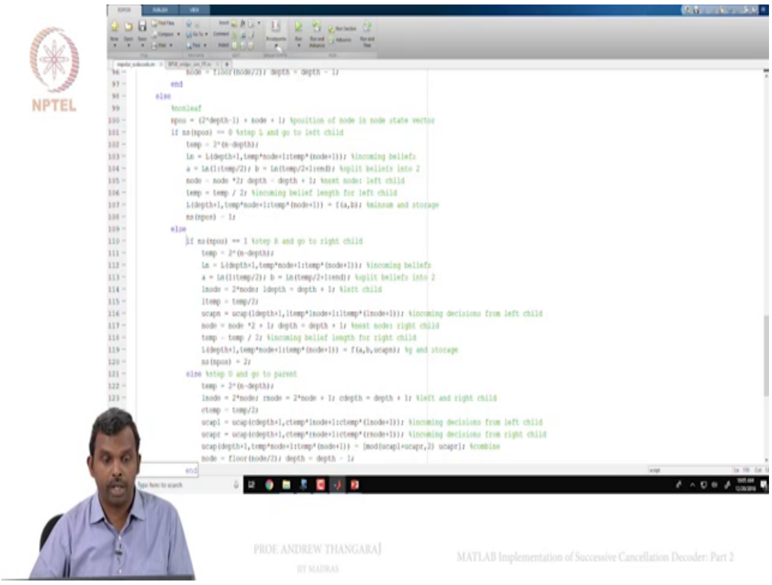


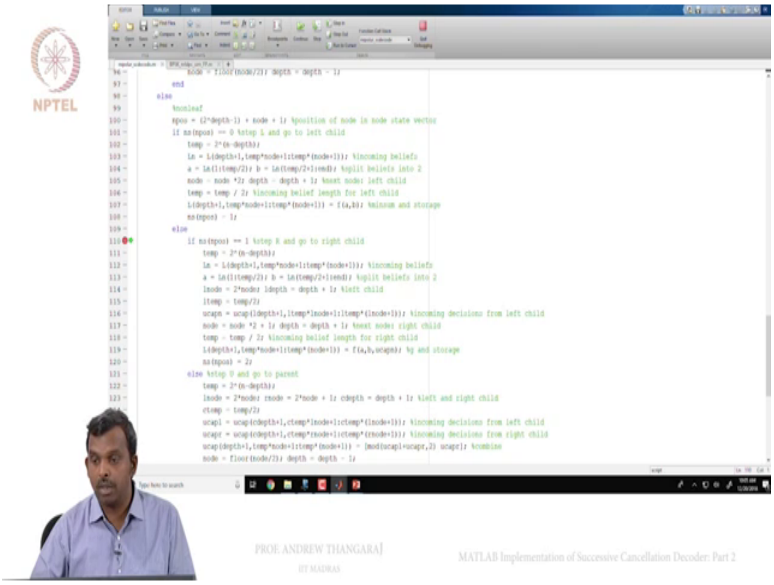


So now we are ready to do step 2, step U but for step U you also need the L to come in properly, so that is okay I think we can do that, so just like this assignment you have the assignment for U cap equals, so this is U cap L, we have to do mod 2 right, mod of U cap L plus U cap R, 2 and then U cap R there you go, so I think that step U I think that is correct okay.

So once you do this step you to go back to parent and going back to the parent is done here, so floor of node equals floor of node by 2 and depth equals depth minus 1 then there you go that is it, so the same steps that we did before, we implemented and in fact looks like we are done with the decoder, we have implemented all steps but we have to check whether everything is happening correctly or not, we have not then the debugging part but as far as the basic implementation is concerned we are more or less done.

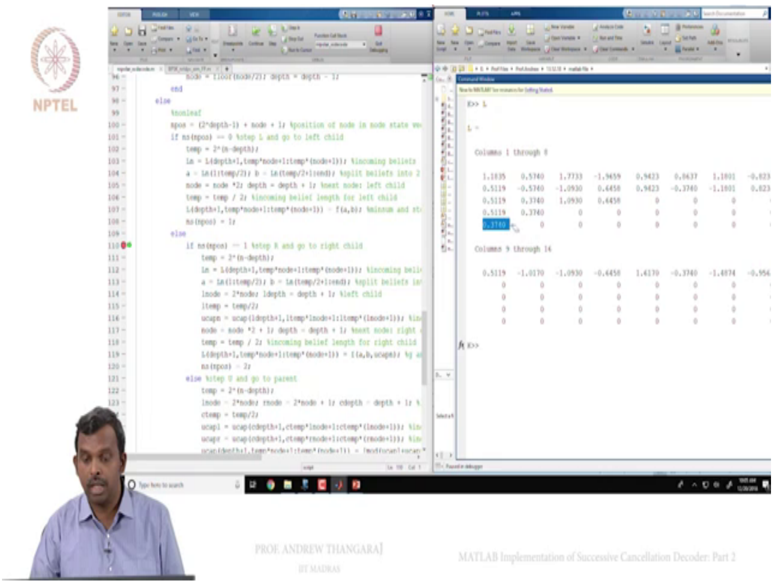
(Refer Slide Time: 21:26)

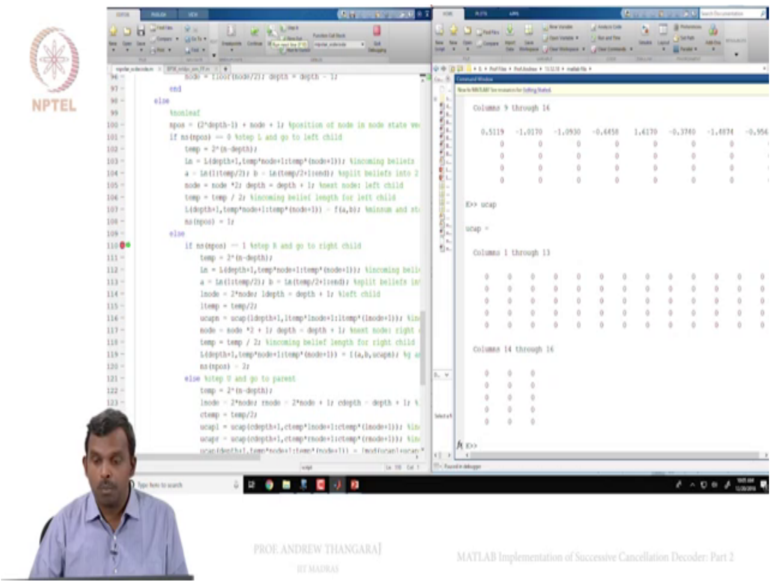


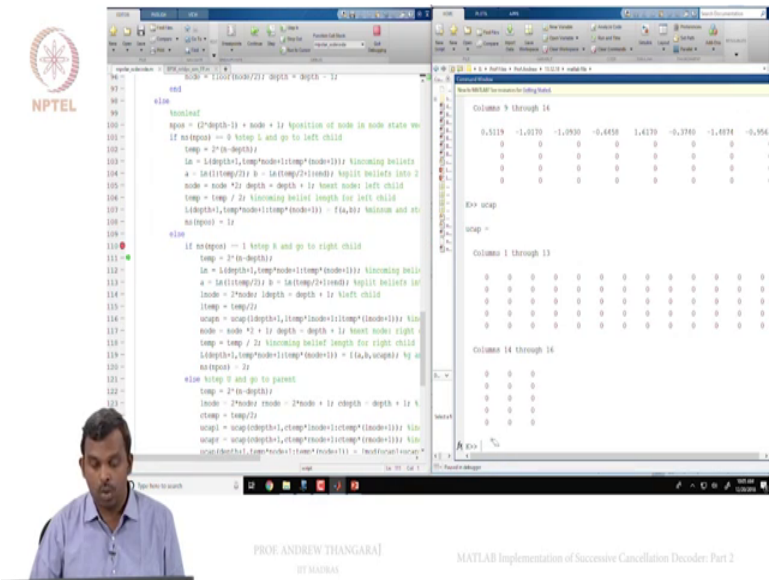


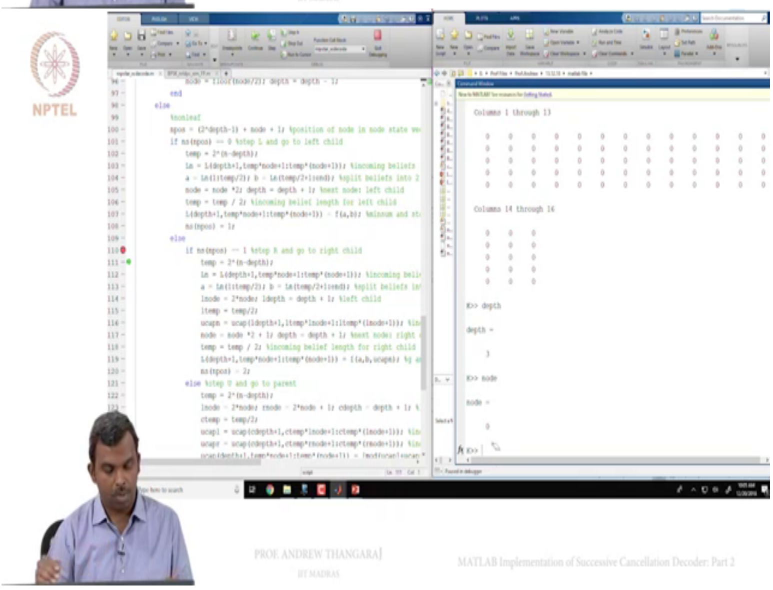
So since we already checked the first initial part going up to the first decision I clear this breakpoint and I put a breakpoint here okay, so when you have to come here okay, so when you have to do over here may be okay, so I put a breakpoint here so that I can run this and I have done all the left steps okay and I made a decision and now I have to come back.

(Refer Slide Time: 21:56)




NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2

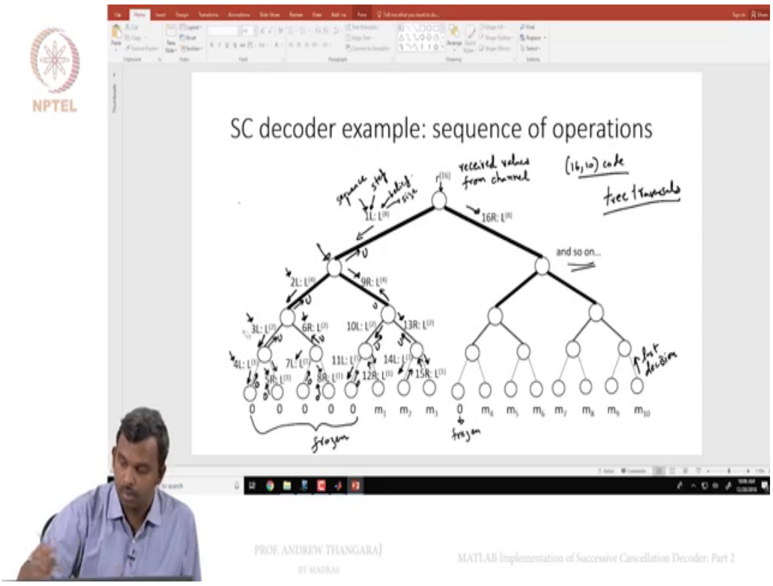

NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2

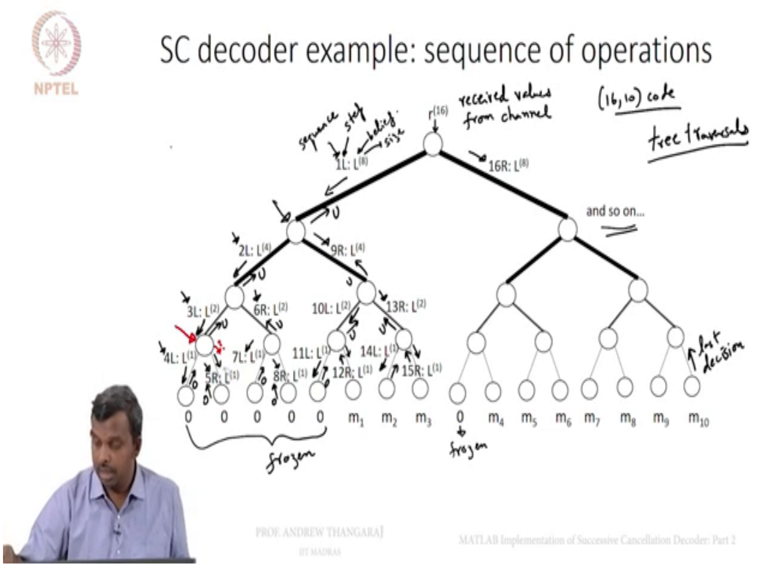

NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2

So just to show you that all that work correctly, let us go back and check the decoder once again, I will clear the screen and I show you the L okay, so you can the L, let us comeback down to the position here and you can also see the U cap it is just 0 at this point, it is no big B that is where it is and now we are ready to check the right operation okay.

So this point N Poss should be 1 okay, so it is going to, remember what is the node and the depth, so depth is 3 and the node is 0, so where am I now have comeback all the way to the leaf, I have made this decision on the leaf and then I have come back to the previous depth okay, so you remember once again.

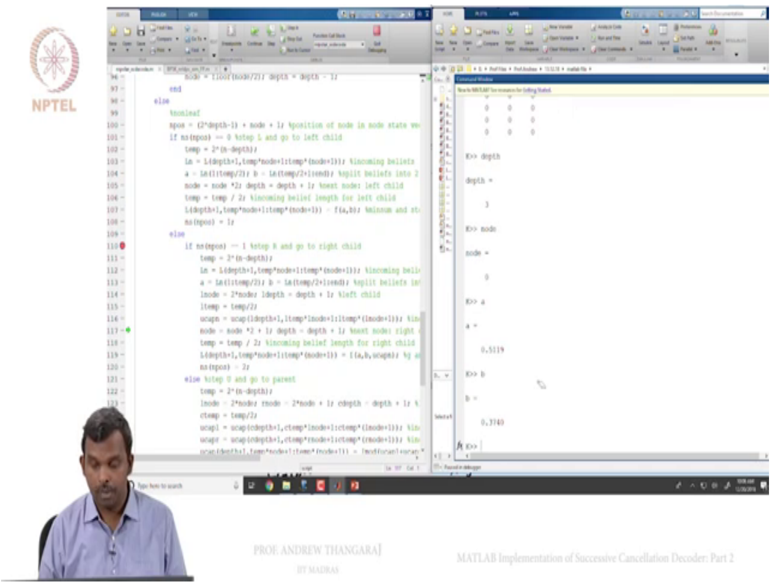
(Refer Slide Time: 22:38)

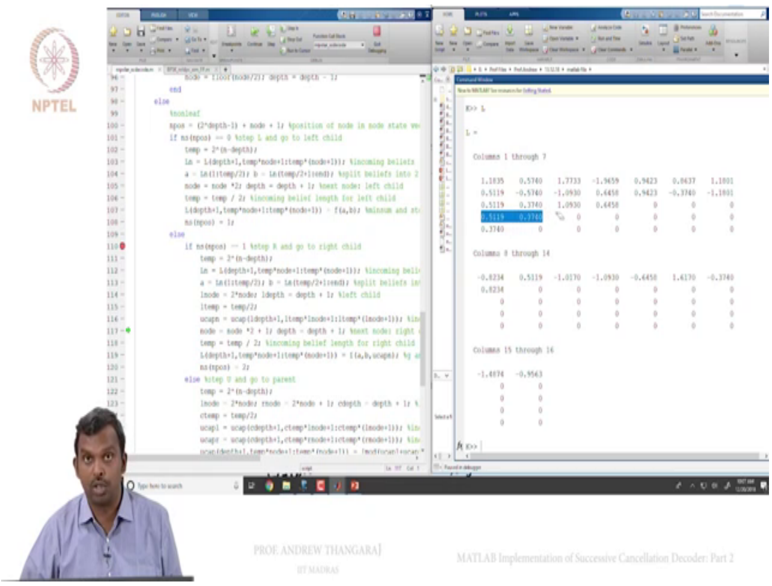


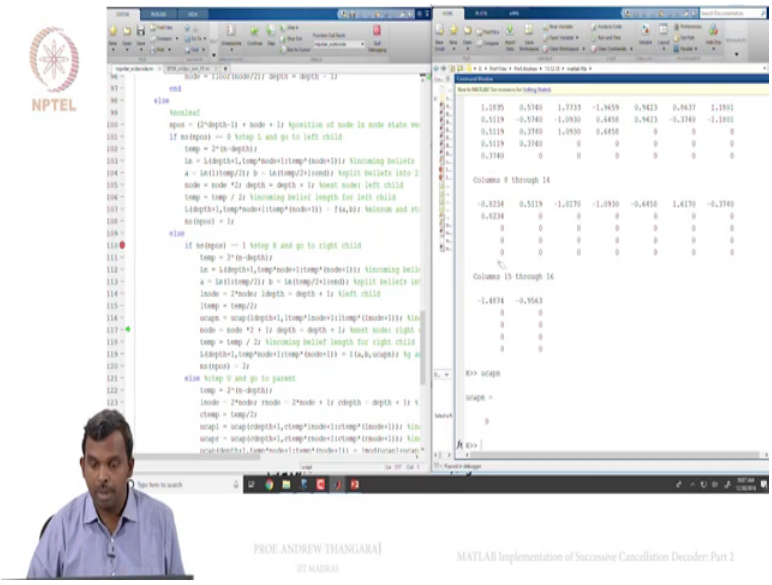


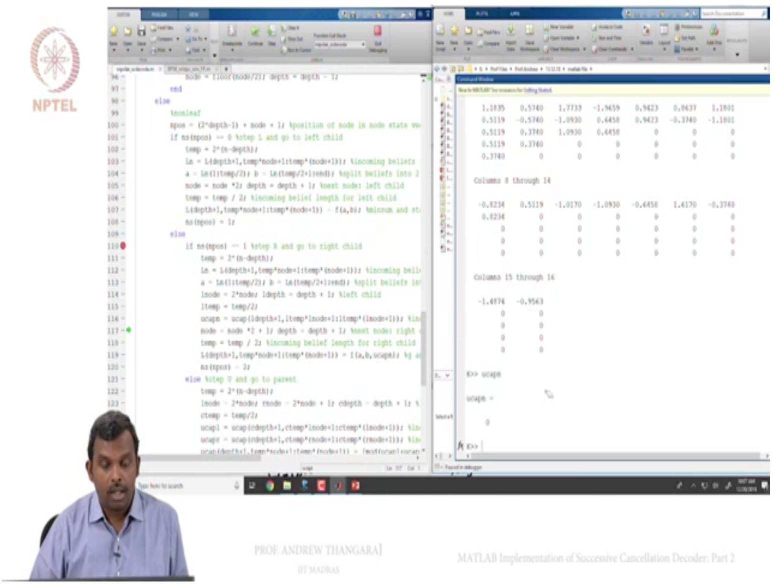
So maybe I should show you go back and show you where we are all on graph, we are somewhere here okay, so here is where we are in the decoder okay, so we went all the way down made a decision and then we came back and we want to go back here okay, so we need to do that operation and let us check whether the operation is happening correctly or wrong okay.

(Refer Slide Time: 23:07)





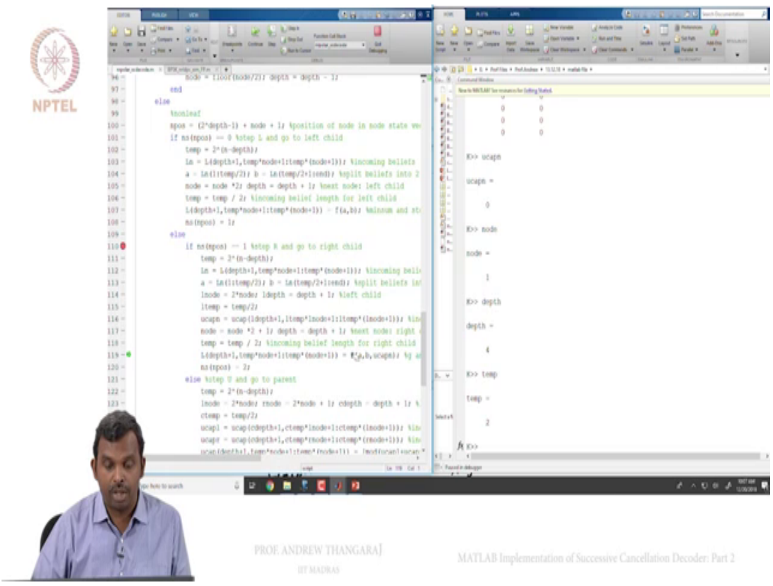


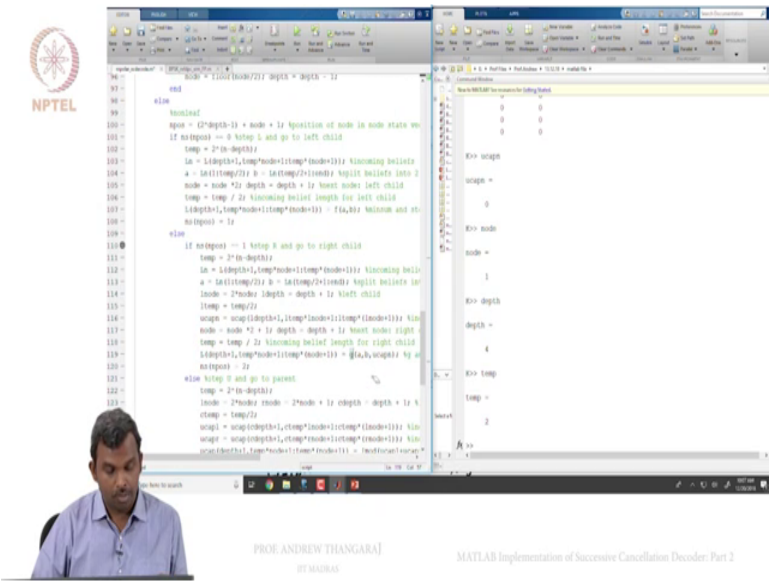


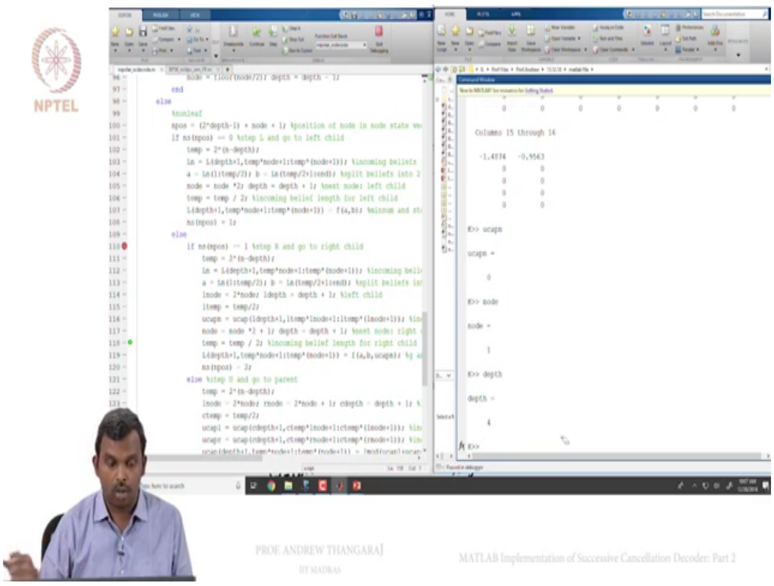
So that is where we are, step R I am going to write child temp that should be okay, LN that should be okay, A and B that should be okay and then it finds the left child that should be okay and the left temp and then the U cap and it should be okay, so let us check if all those things happen correctly or not okay.

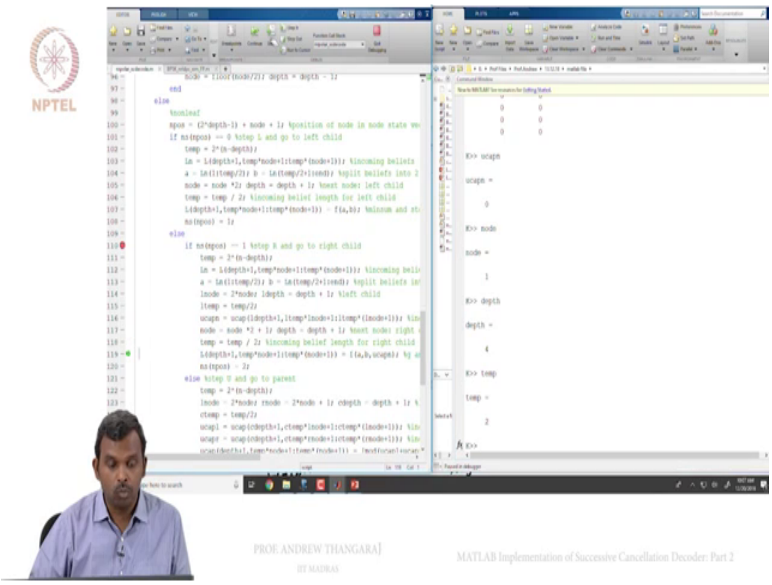
So this is A and B have to be just single values okay and this should just be 0.5149 and 0.3740 okay, so that should be the A and B and U cap N must just be 0 okay, so it should be 0 if it would have picked up the correctly U cap N it assume did that okay, this U cap N it would have picked up okay, so that is okay if you want you can go in and check it but I think it is okay.

(Refer Slide Time: 24:04)



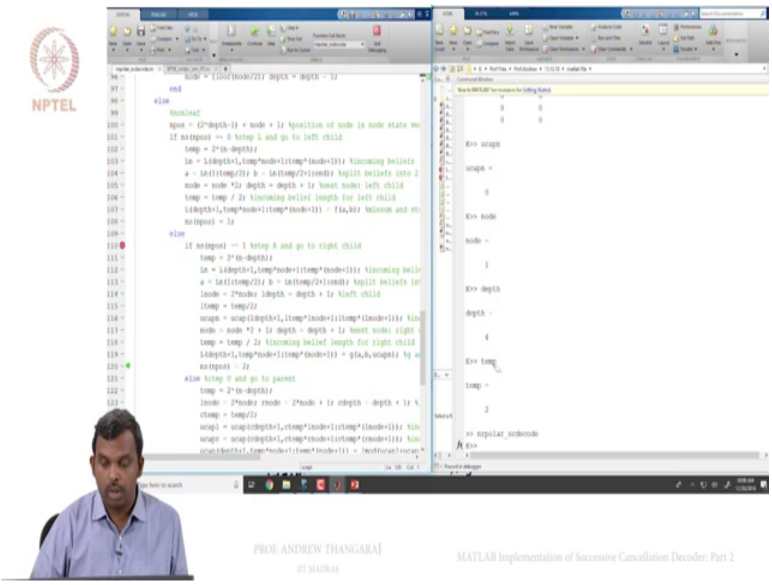


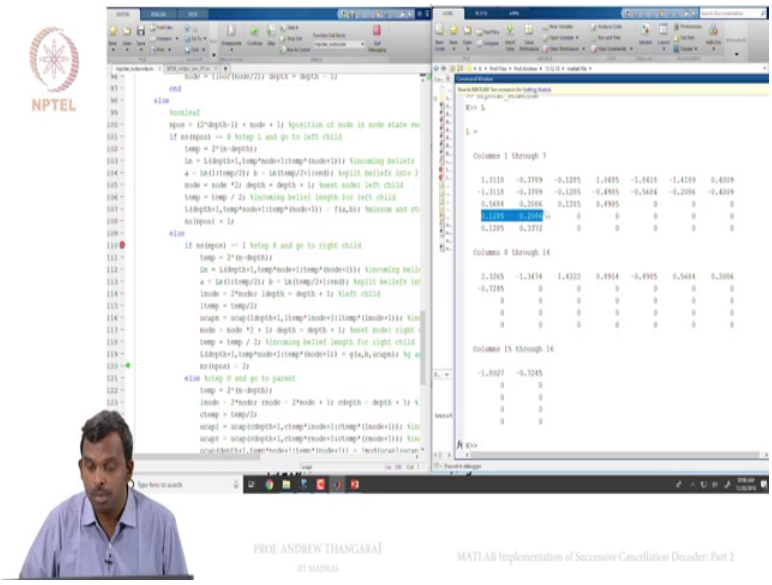


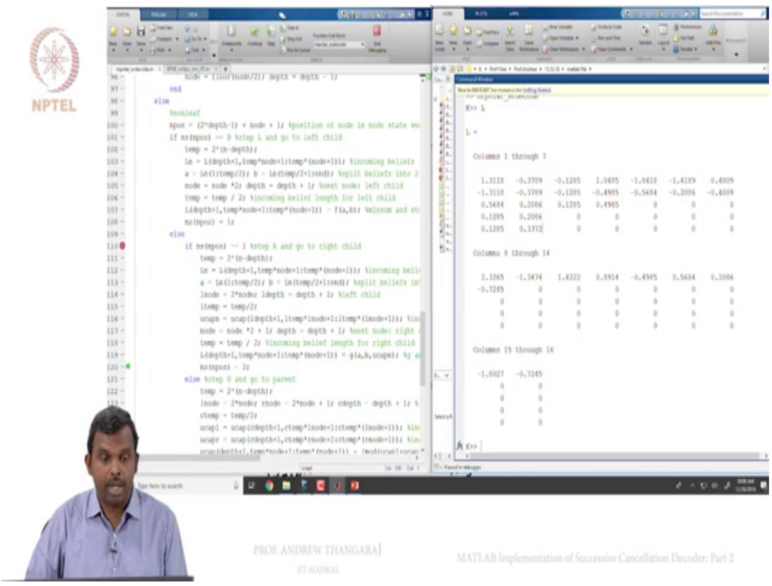


So then it did all that and then is going to the next node okay, so if you look at it now node will be 1 and the depth will be 4 okay, it is gone to the next node, temp probably was 2, so temp needs to go back to 1 okay and then you need to assign it to the correct L right, L is I put at F here it should be G okay, so that is where the array will come, if I run this, this going to be an error, so I will put G here okay, so I made that mistake it should be G, so this will run the correct thing okay.

(Refer Slide Time: 24:40)

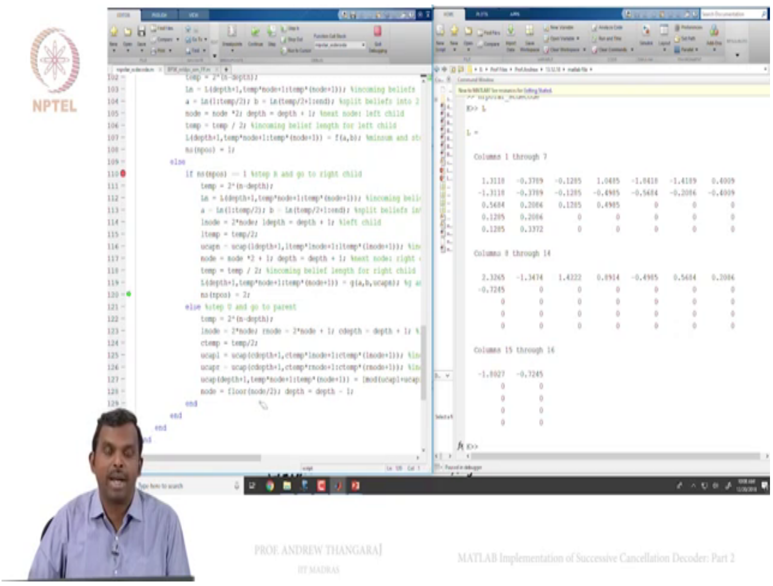


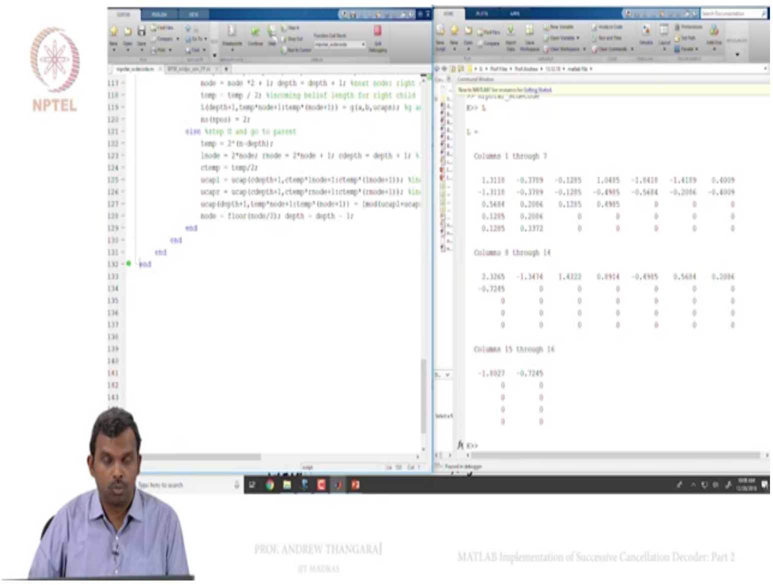




So let us run it once again and quickly step through all the way up to here, things are working correctly and then this assign the L for that kind okay, so let see what happen to the L, this happened correctly or not yes okay, so this must be just the sum of these two and you can see 0.33 is correct okay, so it added these two and you got the sum, that is the L value okay.

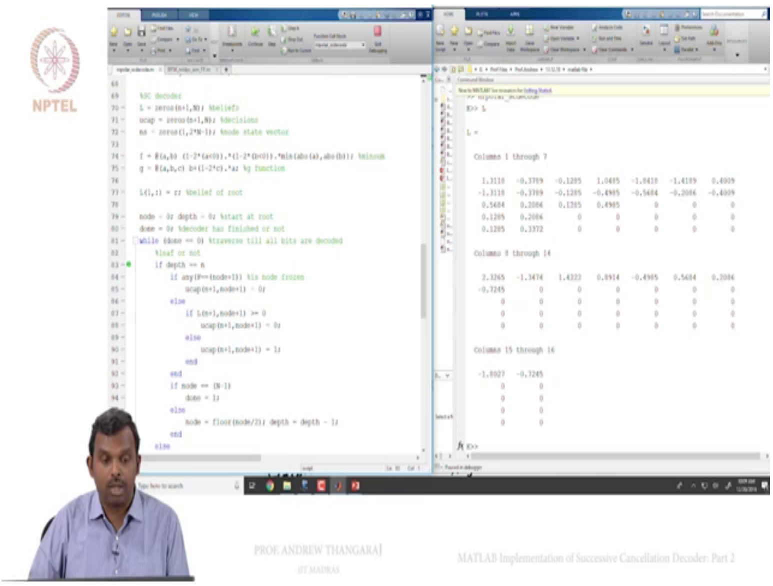
(Refer Slide Time: 25:10)

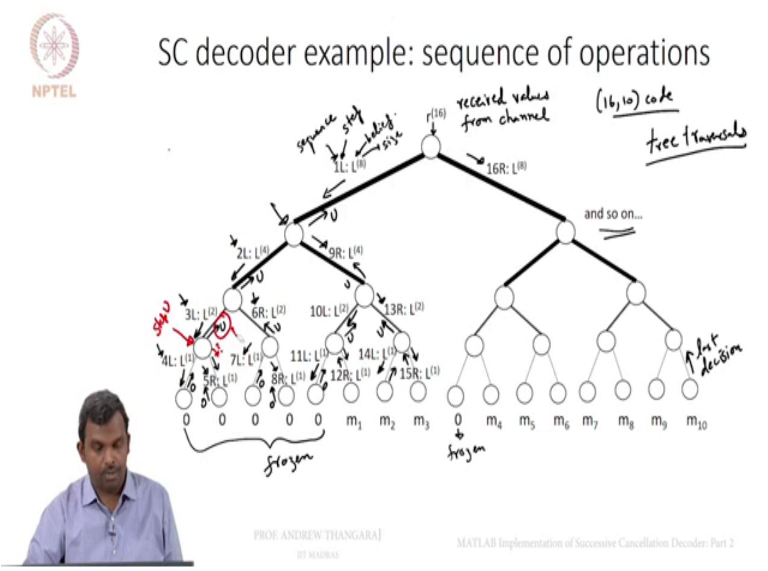


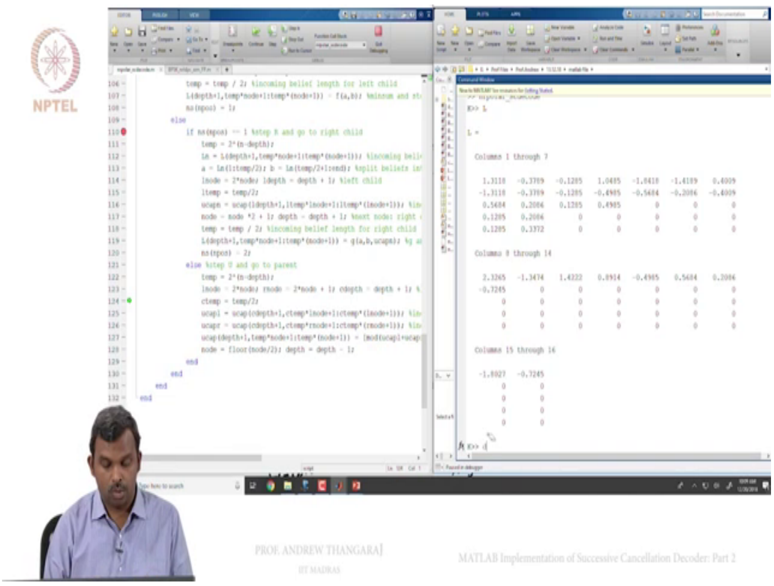


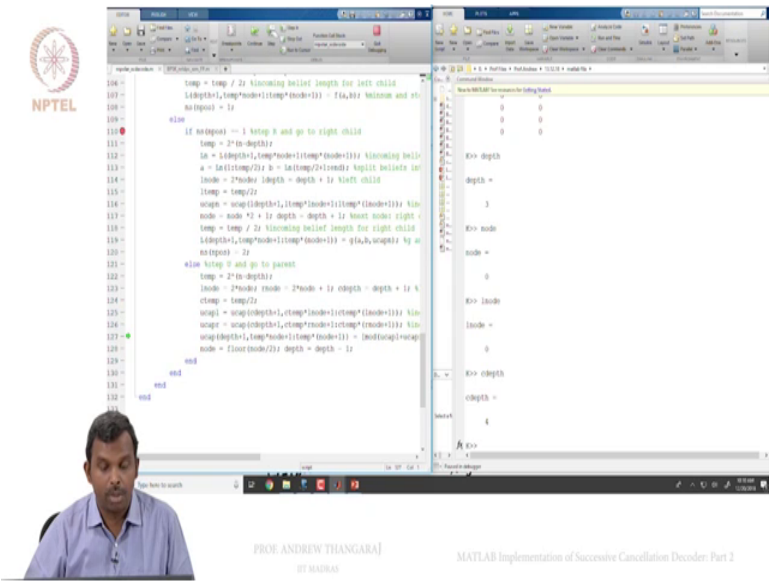
So the next step and made the state as two for this and see I am not bothering once the U happens to change the state because when I could change it but I will not come back to it ever again, so it does not matter too much, then we continue and then it will go to decision okay, so it will make a decision, it is frozen again and it is not so it is goes back up in depth to the next one okay.

(Refer Slide Time: 25:37)





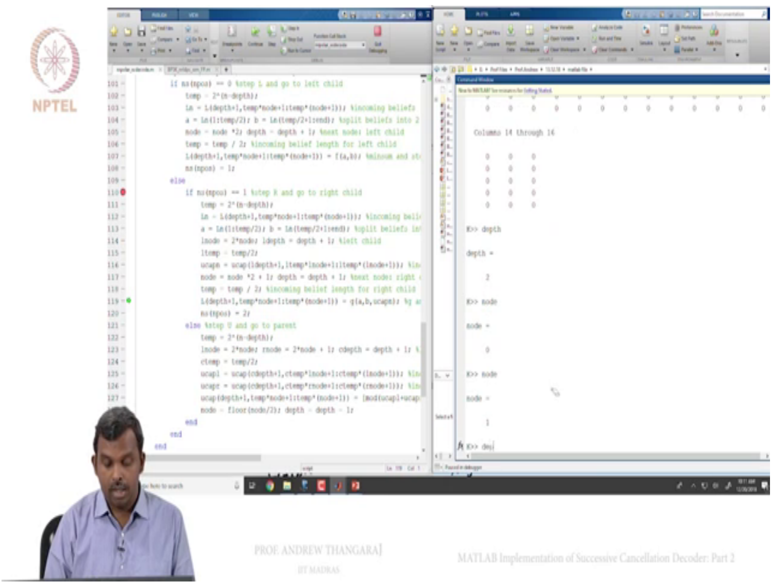


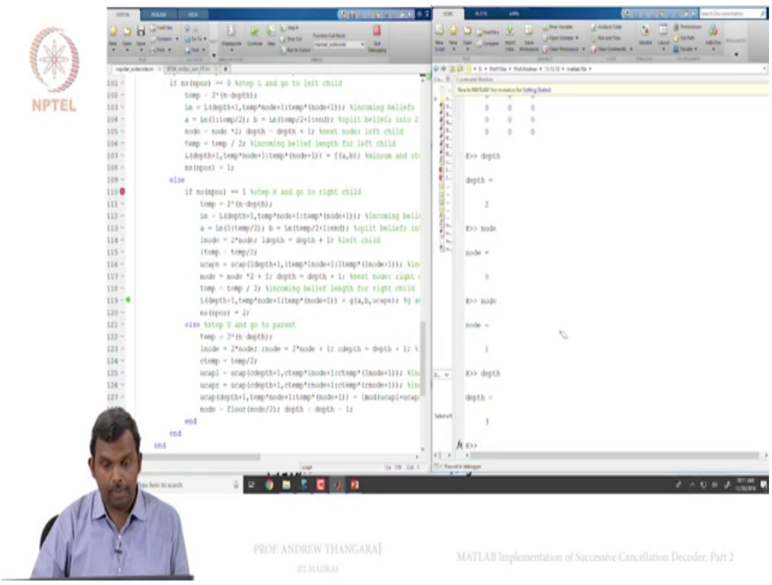


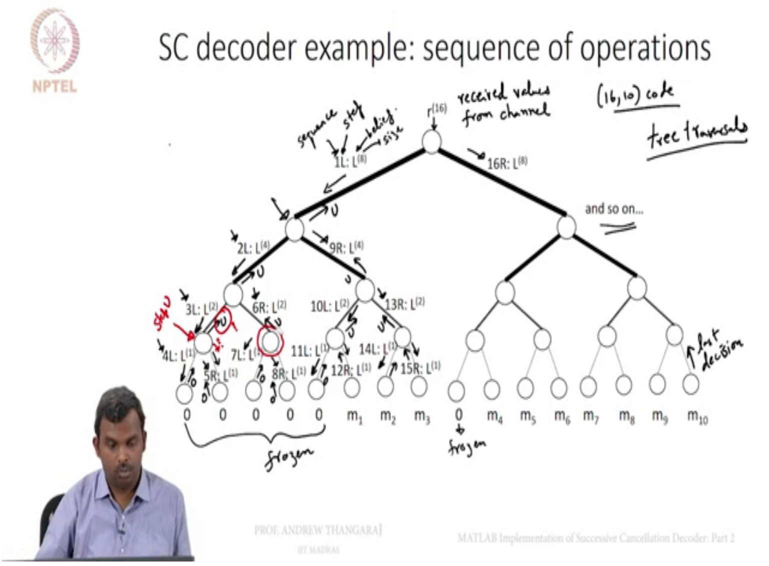
So now it will come to once again where is it come okay, so you remember it is finished the decision here and it is comeback here, now this node is here to do step U okay, so it is needs to figure out this step U here and it will do it, we will see it does it correctly okay, so depth is not N now, so N Poss it calculates and step, it is not 0, it is not 1, it will go to 2 okay.

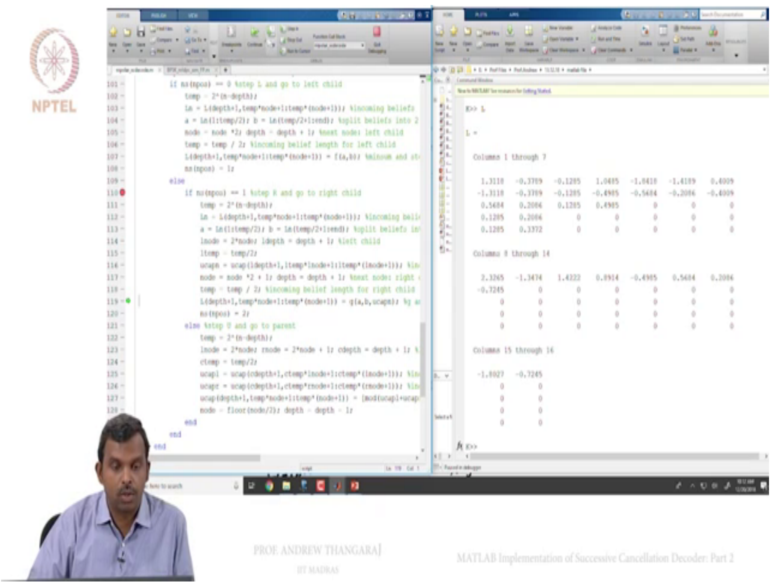
So now it has to do step 2, calculates the depth and then the L temp, L node and C depth, so let just check it out once again, so depth is should be 3, node should be 0, it going to go up right, so and then let us look at L node it will be 0 again and C depth will be 4 okay, so that is okay and then it calculates C temp that is okay, it does that C temp should just be 1 okay, I can check that it is 1 and then it pulls out the U cap both from L node and I am not, this will just be 1 bit okay.

(Refer Slide Time: 27:02)




NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2

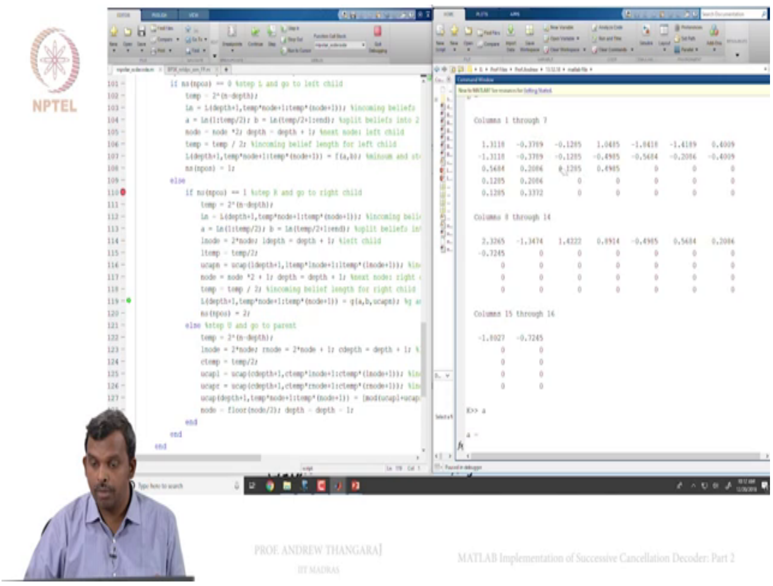

NPTEL
SC decoder example: sequence of operations
received values from channel
(16,10) code
tree traversals
sequence step
belief size
1L: L(8)
16R: L(8)
and so on...
2L: L(4)
9R: L(4)
3L: L(2)
6R: L(2)
10L: L(2)
13R: L(2)
4L: L(1)
5R: L(1)
7L: L(1)
8R: L(1)
11L: L(1)
12R: L(1)
14L: L(1)
15R: L(1)
0 0 0 0 0 m1 m2 m3 0 m4 m5 m6 m7 m8 m9 m10
frozen
frozen
first decision
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2

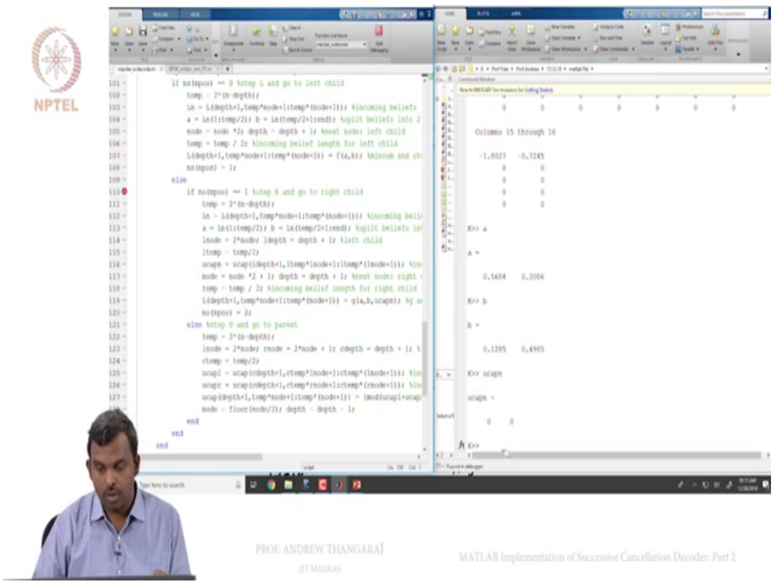

NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2

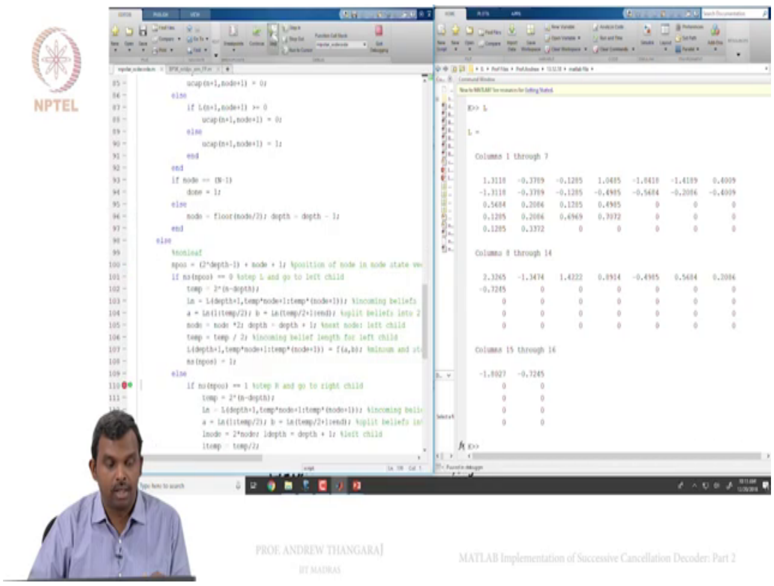
So U cap L and U cap R will both be 1 bit and there will be 0 because they were frozen positions and then you calculate the U cap for going up okay, so if you do this make it again busy ruse it is, so if CU cap I mean everything is going to be 0, so you not going to see much and then you go up to the parent okay, when you went up to the parent you would have gone to depth of 2 and the node would be 0 okay that again to 0 and then after this it will continue, depth is not N and calculates N Poss, this will be actually 1.

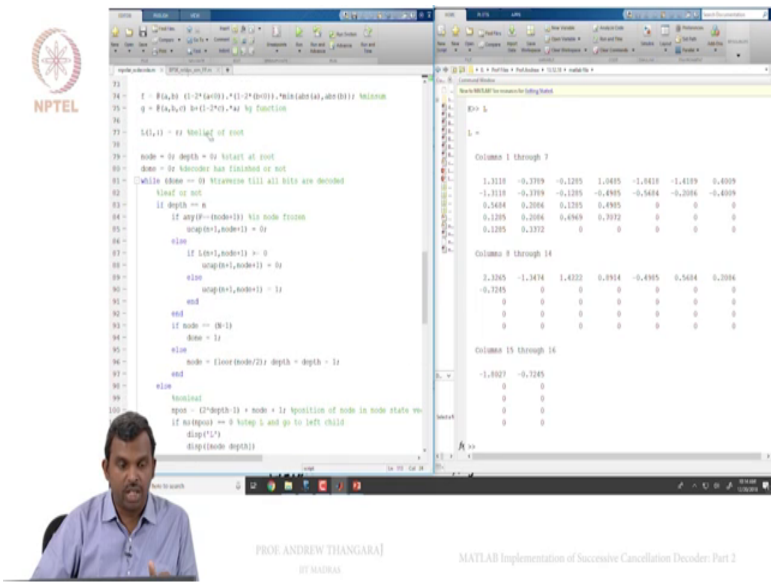
So it will do the right calculation, so it is good to check what it did here, so remember this going to go right, node is 1, depth is 3 okay, so what is node 1 step depth 3, so maybe you can see what is node 1 in depth 3, so it is come here and node while it is come here okay, it is come to this guy okay and that come there which is be sure, let me just see the L here yes, so this is fine so this goes to node one in depth 3, yes so I think I went through the L and it just is not pay attention to okay, so that is okay, so that is good.

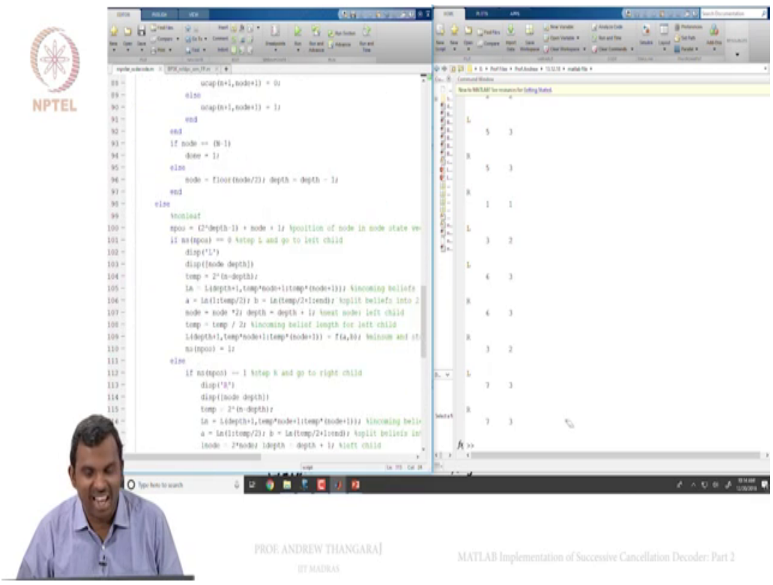
(Refer Slide Time: 28:46)




NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2


NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2


NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2



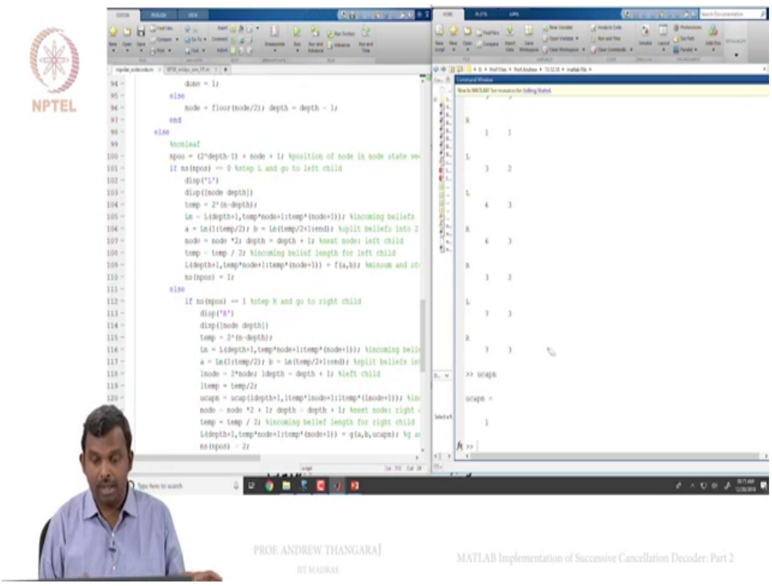
So let us take up the A and B okay A is 2 values now okay, so 0.5684, 0.2086 then B is the remaining two values 0.1285 and 0.4985 okay, so node okay for some reason I calculate it, I have gone to the next node okay alright, so this is fine, so this is good, so then you have U cap N which will be actually two 0s and then when I computes it, it will go off and then the next state goes to this.

So I have not gone to the next node, that is the next node and going to, so now this node is 1 and 2 okay, so this is okay, so this is looking correctly, so you have the Ls calculated okay, so you can go through and look at this and see how it proceeds then this will not be true, this will go left further again yes and then it will come back make a decision then it will go right again, just to check and then it will go, make a decision it is doing a lot of things.

So it seems to be working, there is no major bug here at least I am not checked it fully but at least is working, so I am going to clear all the breakpoints and just to check what is happening, so I am going to print some information out just to check that things are happening correctly or wrongly, so any time I enter my left I am going to print, print does work, so I am going to just display node and depth okay, so just to make sure I see something every time I go left and so I should say I went left right.

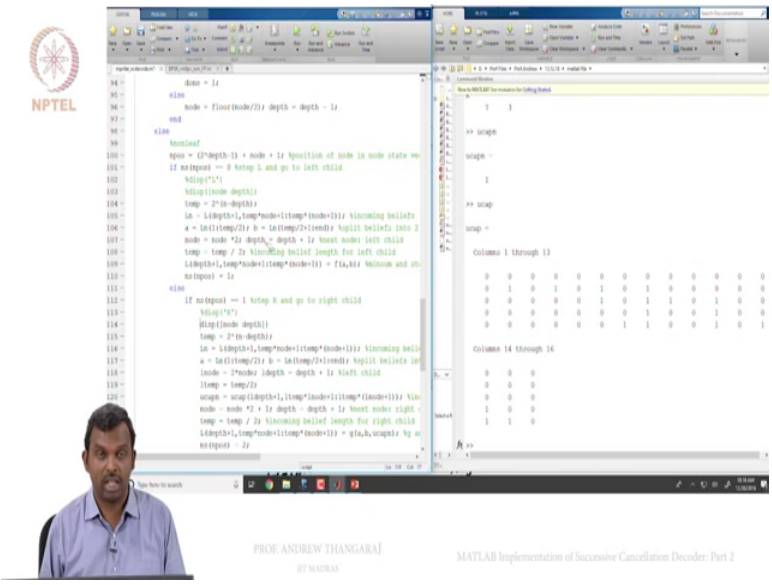
So I display left also okay, then I say when I go right I will display right just to see that whole thing is proceeding correctly or wrong and then depth and, node and depth okay just to see how the whole thing is going, than it should finish, so let me just run it once to see what it does okay, so it finished give me some answer but no idea whether the answer is correct or wrong but at least it ran.

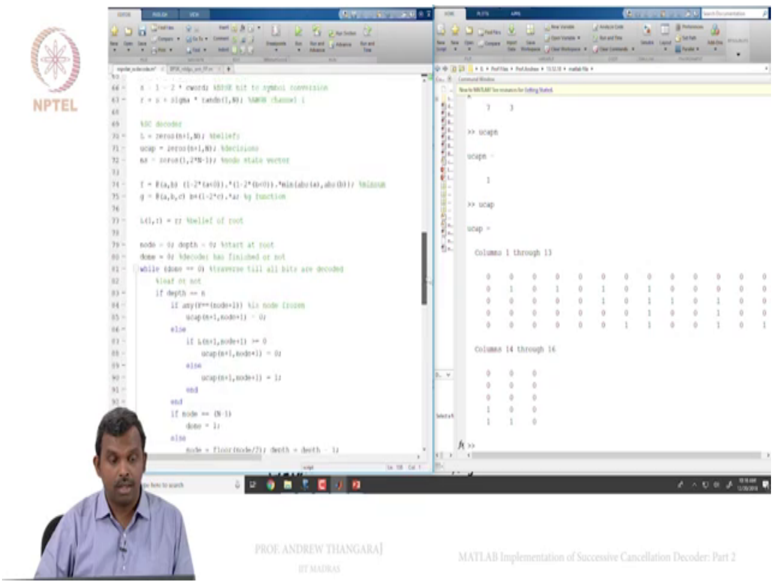
(Refer Slide Time: 31:36)



So let us see whether the sequence is correct or wrong okay, so that is the first thing one can check, L node 0 okay, so it went left to depth 1, went left to depth 2, went left to depth 3 then it went right okay to depth 3, right from depth 3 and then right from depth 2 left from, so it looks okay and it is given me some final answer, so you can check on U cap N.

(Refer Slide Time: 32:07)

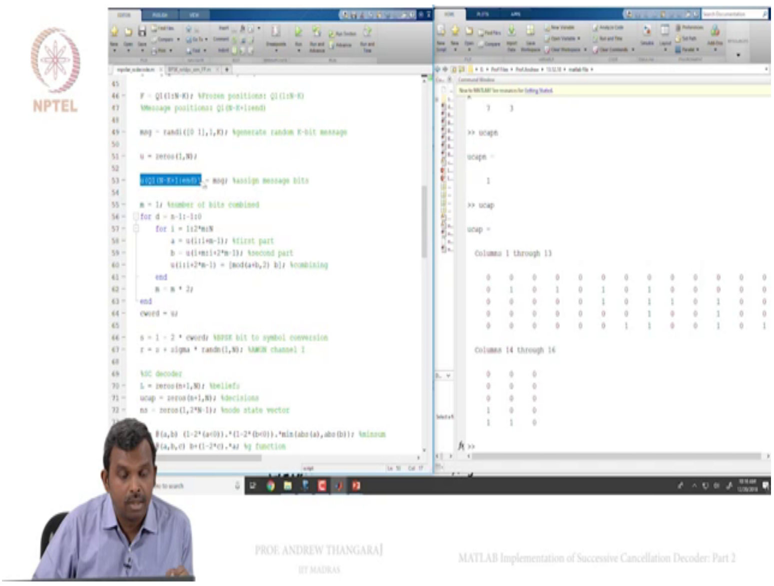


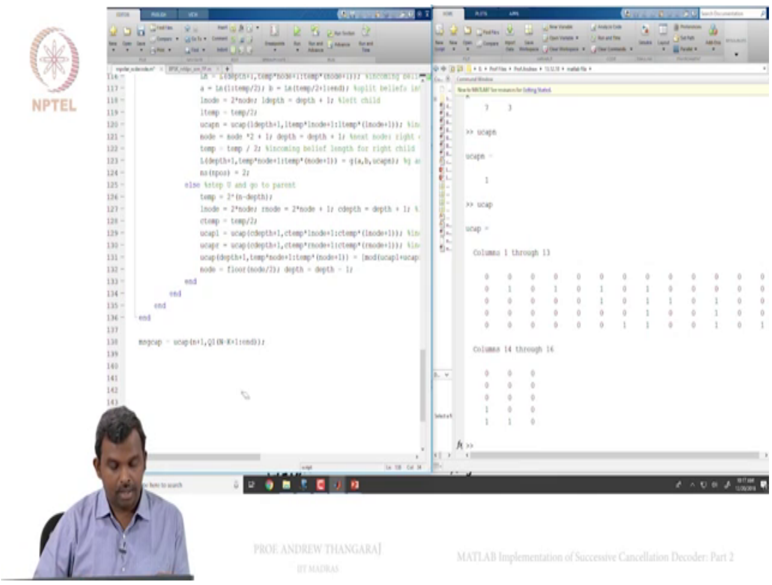


So if you look at the U cap, this is the overall U cap but remember the actual message is in the last row okay I did not pull it out in the code but the actual message is in the last row, I was just checking here to ensure that everything is happening correctly, it looks to be working fine so the message is in the last row okay.

So let us maybe not print is things, this is just some debugging to check everything is working correctly, it seems to be working correctly, so I am done with the decoder as I now I can assign my M cap right, so my actual, so I have to pull out the messages from the frozen positions or I could just compare the U cap of the last one with the U right, so U is my actual code word or code word itself, so that is something that one can do, let me save this okay.

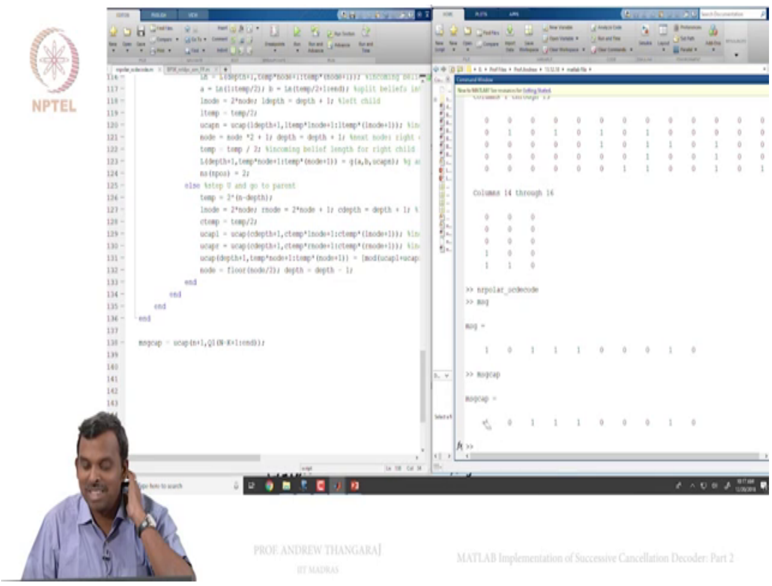
(Refer Slide Time: 33:08)

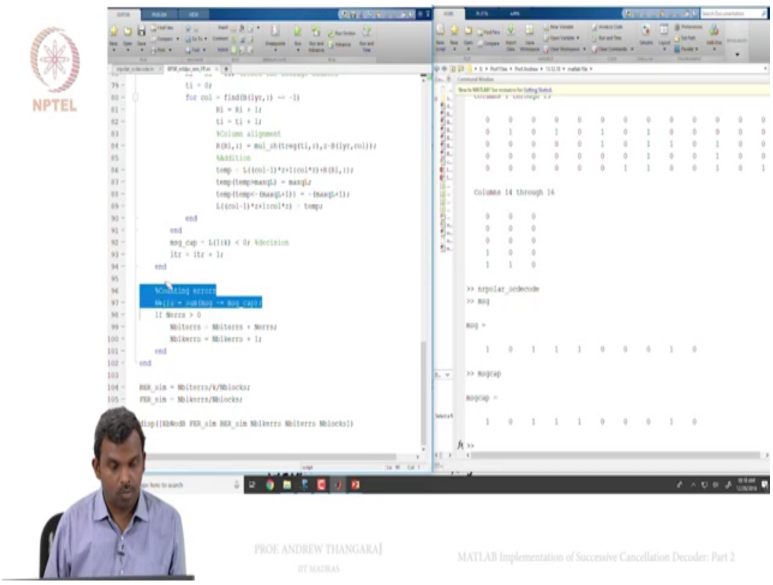


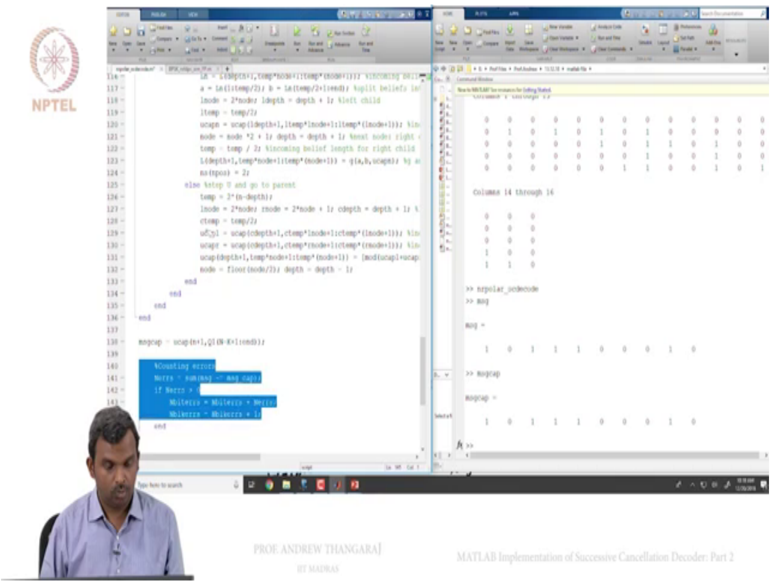


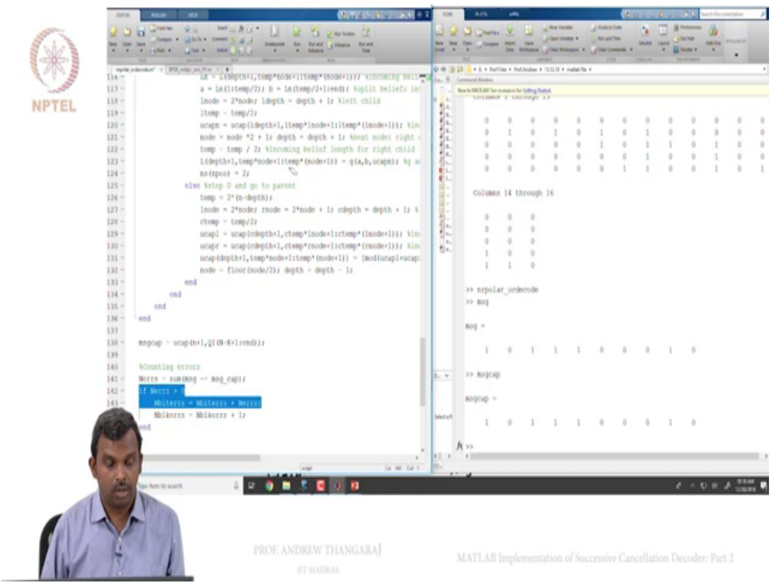
So you can see if these two are equals, so the message went into view of this right, so this is where the message went in to okay, so if I pull out this I should get the message cap back okay, so let us that U cap of N plus 1, the same Q1, there was the Q1 of N minus K plus 1 colon N okay, so this is the position where the thing got inserted, so the same thing in the last row should be my message cap okay.

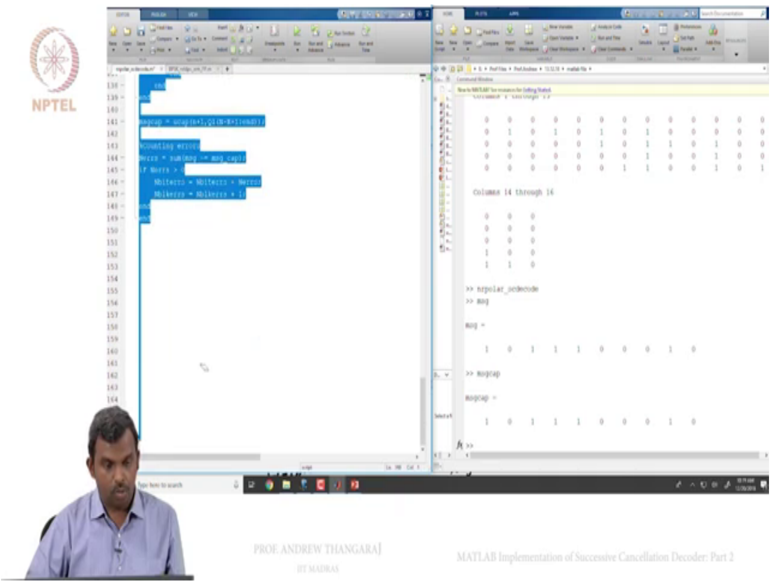
(Refer Slide Time: 33:58)

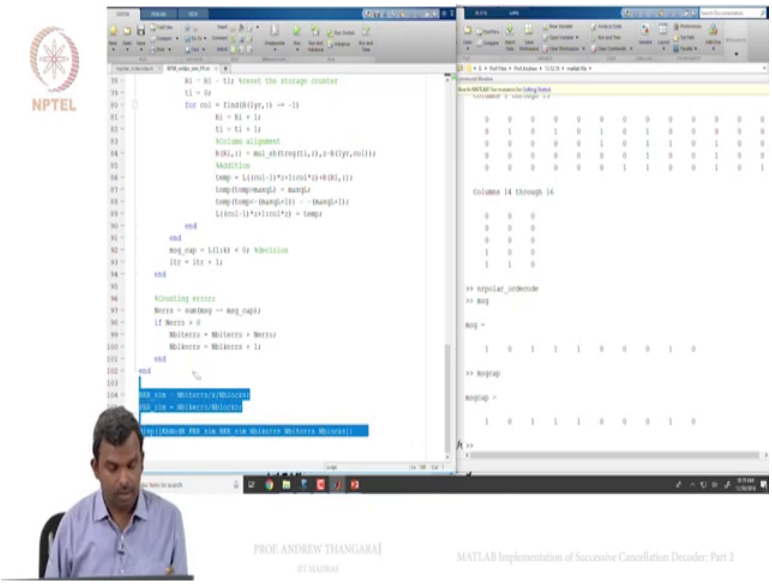


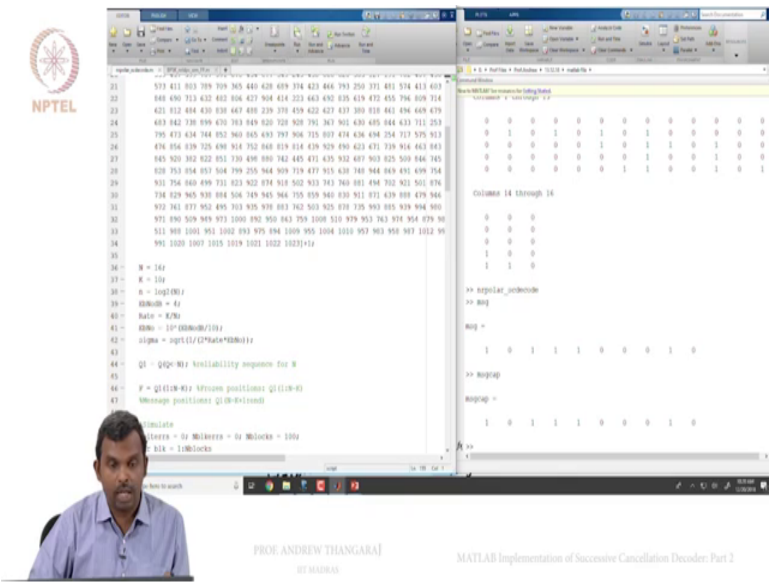

NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2

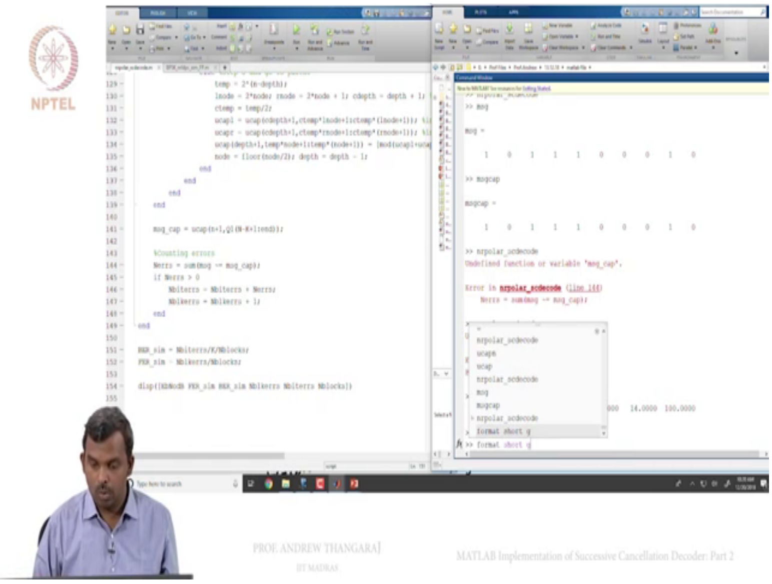

NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2

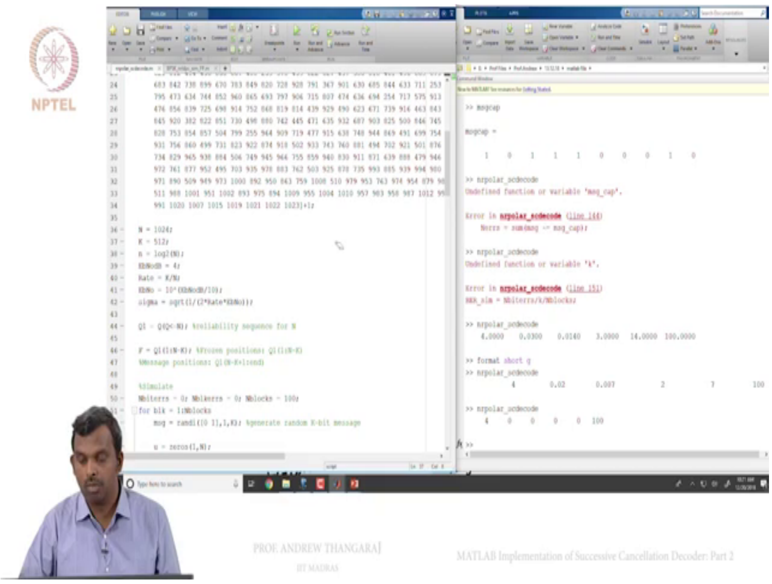

NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2

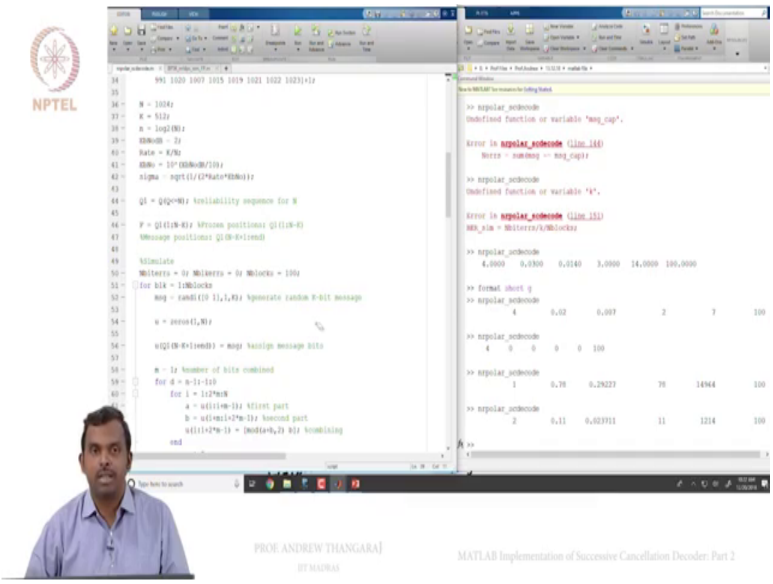

NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2


NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2


NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2


NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2


NPTEL
PROF. ANDREW THANGARAJ
IIT MADRAS
MATLAB Implementation of Successive Cancellation Decoder: Part 2



So let us run this and then see if were at this highest NR if it is working or not, you say you had a message and you had a message cap and they are the same very nice is not it, so it is good to see that the message in the message cap agree maybe there were no errors in the channels but still our decoder is working and it is agreeing, so you can again to the same thing as before our way in which count the errors, you remember I do the error counting okay, so you can do that okay, so I am not putting it in a loop.

So this part is not so relevant but anyway, so I am not putting it in a loop for calculating the message and sending it out, so maybe that part we can add, so how do you add that, so you put this here and blocks, where do I put this, I needed below this simulate and then I need a for loop, for loop needs an N and you know the whole thing can be intended, intended a little bit so that it goes off that key okay, so I think I have done most of what I wanted to do.

So I put blockers 0 and then I am simulating for every block and every block I generate a message, I had code word and I transmitted, I assigned these things to 0, I think I am doing all of that correctly than simulating it, decoding it, getting a message cap and checking it up okay and then we can now print, the B are same and the, and this, there you go.

So this will even simulate 100 blocks for the 1610 code at 4 DB SNR so let us do that, there you go okay, did I not put underscore okay sorry, there you go, it got to change the capital K okay, so now we are ready to simulate and you get some answers okay, so there were 3 block errors out of 100, format short G, opps what did I do.

So one can decode, so you get 2 errors out of 100, 3 errors out of 100, so this is basically working, so maybe we want to be very more experimental, so let us try the largest block length that the 5G standard allows, 1024 and 512, so this is going to be big and let us see what it does, no errors at 4 DB is huge SNR, so let us make it may be 1 DB, maybes will see some errors out of 100, 78 were an errors, so maybe 1 DB is not too great, maybe 2 DB, 11 were an errors so on.

So the basic decoder I believe is working, this is the successive cancellation decoder for polar codes and you saw it is reasonably easy to code, it is a short piece of code that gives you good answers okay of course one can make it more efficient, one can make it more interesting, one can make modifications we can keep on working on it and what we will do the next week is to look at list decoding for polar codes, so list decoding is quite important that is the decoder which actually works in practice, it gives you better gains than this, will see that in the next class, in the next lecture and as far as this lecture is concern we are done. Thank you very much.