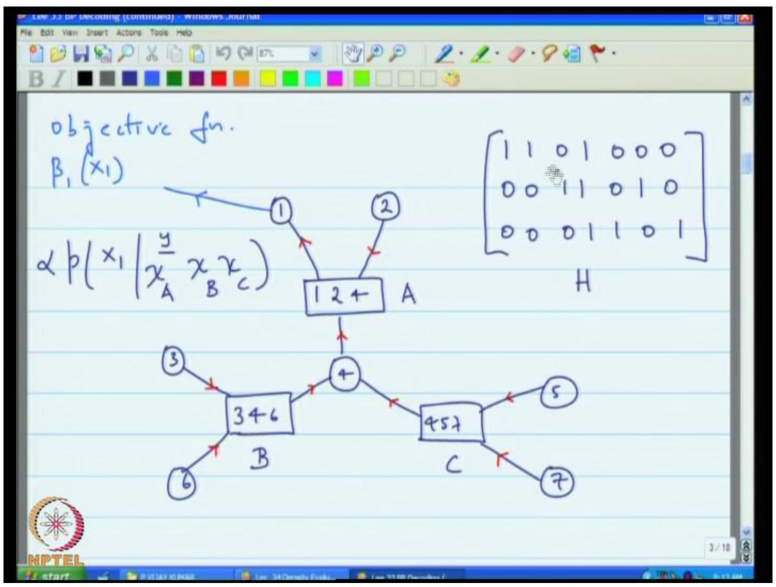
**Error Correcting Codes**
**Prof. Dr. P Vijay Kumar**
**Department of Electrical Communication Engineering**
**Indian Institute of Science, Bangalore**

**Lecture No. # 34**
**Density Evolution under BP Decoding**

Good afternoon, welcome. So, this is now entering lecture 34. And as always, let me begin by going over what we covered last time.

(Refer Slide Time: 00:44)



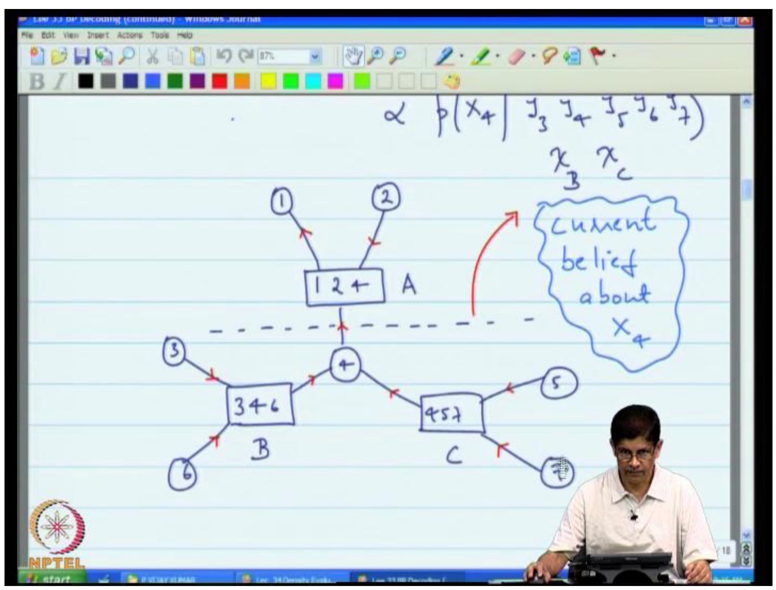

The title was belief propagation decoding continued, and in reality this was the lecture and which we actually started talking about belief propagation. And the perspective I wanted to bring in the last lecture was that as far as decoding was concerned the algorithm, the belief propagation algorithm is exactly the algorithm, that we used in conjunction with the g d l.

So, that is for that reason just to remind you that that was the case. And also to make another point that we when we use the g d l to decode the 7 4 2 block code. Then I wanted to point out that, there was a simple interpretation for every message that was passed on the edge on the on the junction tree graph for that particular code. And it had an interpretation in terms of beliefs; and in fact, it is that which actually gives us raise to the term belief propagation decoding. So, that is how I actually started out.

So, here what I pointed out was look, if you actually look at this figure over here; then you can see that you have messages, let us say that we are in interested in decoding the value of this particular code symbol. And then I looked at I tried to give you sense for so overall what we are interested in as finding out what is the probability that x 1 was transmitted given all the received symbols, and given that the code symbols themselves satisfied parity checks A, B and C.

So, this in itself is a belief, because it is the belief about x 1 based on all the evidence. But you are in above that it was interesting that you can also give interpretations to messages intermediate messages that also have been can be explained in terms of belief.

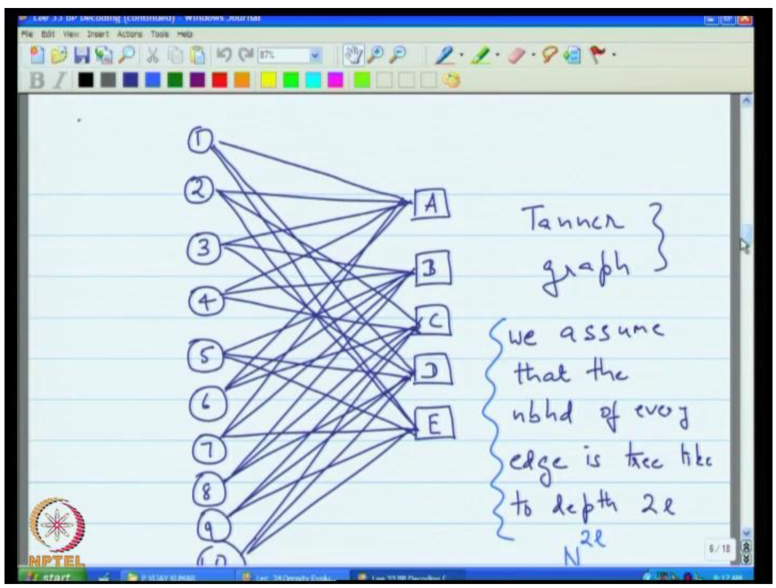
(Refer Slide Time: 02:35)




So, for example when we come down over here, then the message that was passed here, and again we are aiming to eventually compute at the objective function here, but even at the intermediate step, the message that was be cross this barrier over here was the current belief about x 4 based on some evidence.

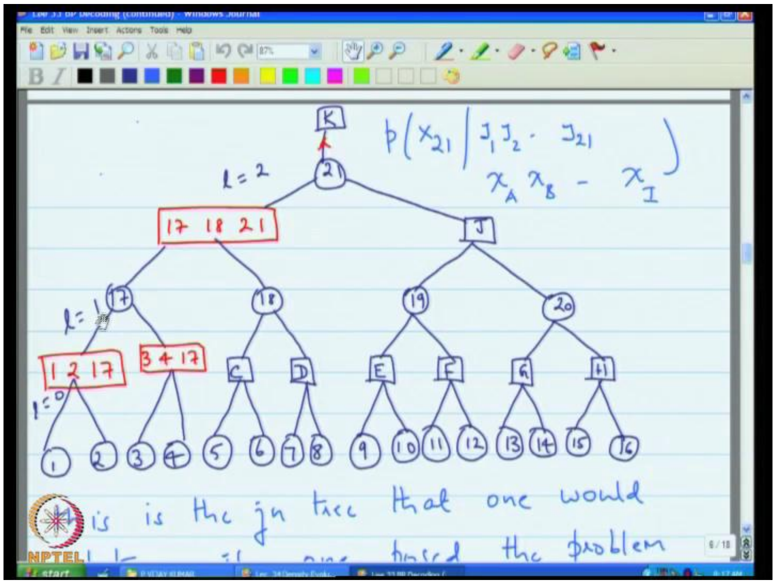
What does some mean? Well it is all the evidence which is on this side of the barrier, and this most specifically that evidence is where the received symbols y 3, y 4, y 5, y 6, y 7, and on top of that we also have that belief that also we also had knowledge of the particular parity checks B and C when rolling back even further, if you actually look at the message across this barrier again it is a belief about x 4, but this time based upon y 3, y 6, and the parity. So, when we talk about belief propagation decoding on the tanner graph, then we have doing exactly this method of message passing exactly this way of passing message is that we actually carry over. The other question that is in my mind as well, but you know that whole theory was actually developed for junction trees, where us suddenly you are in porting it now to the tanner graph of an LDPC code.

(Refer Slide Time: 04:02)




What is the connection? And then I actually pointed out that wait a minute. If you assume, so here is your tanner. If you assume that the tanner graph of an LDPC code has a tree like neighborhood to depth 2 l.

(Refer Slide Time: 04:15)



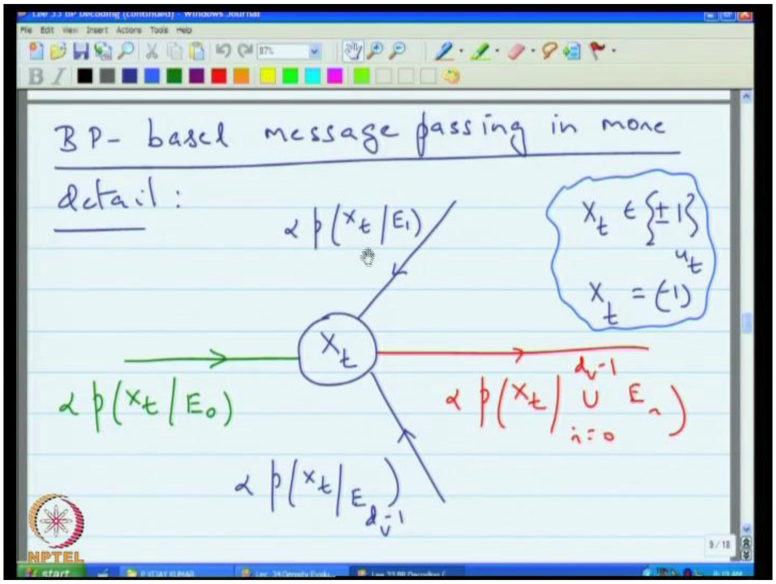

What you mean by that is that if you unravel the tree like this, and this is called the computational tree, then every node that you encounters distinct, so that means there are no cycles. For example, if this node forded to also pear here; that means, they would be a cycle in this graph. Since the nodes are all distinct at least to depth 1, 2, 3, 4 times to it to depth it, there are no cycles in the neighborhood of node 21.

So, when the neighborhood is cycle 3, then I made the point that look if you actually write down these parity checks, the way we did for the 7 4 2 code in terms of listing all the variable in the series that they actually check parity on. Then you see that in fact, it is a junction tree, for example and the way you check it, check for a junction tree is just by projection, so let us project it for a instance on variable 17. If you project on to 17, then you get the actually your connected graph, which consisting of this four nodes. So, similarly you will see that in every instance you get a connected tree, and therefore this is a junction tree. Am I right.

This is the junction tree that one would obtain if one posed the problem of computing this as an MPF problem. So we are in fact, under the tree lock assumption in the certain of a g d l and of the junction tree, and because of this, we can actually interpret every message that is being passed around in the graph as a belief.

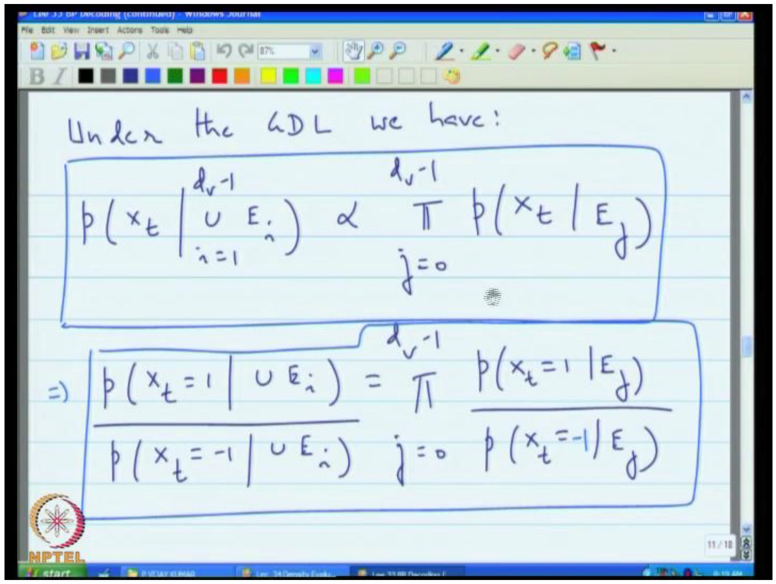
For example here, you would be passing from this node to this node a belief about x 17 based on their evidence would corresponding to receive symbols 1, 2, 3, 4; and including the knowledge that there was a parity check here and a parity check that. So, now that is why it is called belief propagation decoding.

(Refer Slide Time: 06:04)



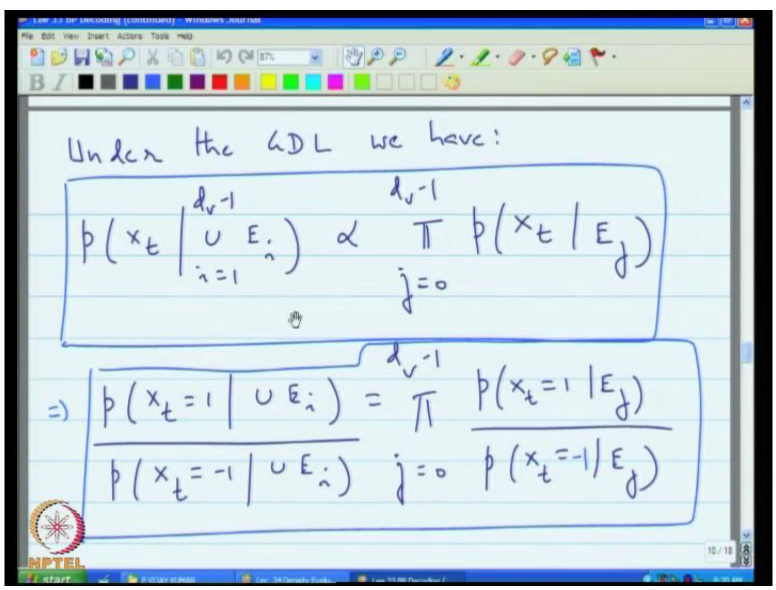

So, and in I introduce the notation of E 1 E 0 E d v minus 1. So, this stands for evidence. So, at a at a generic variable node you get some belief about x t from the channel which I call evidence is 0, but then you also get evidence from the other check node which are called event E d v minus 1. And what actually goes out is a belief about x t based on the union of all these evidence, so that is actually listed here.

(Refer Slide Time: 06:40)



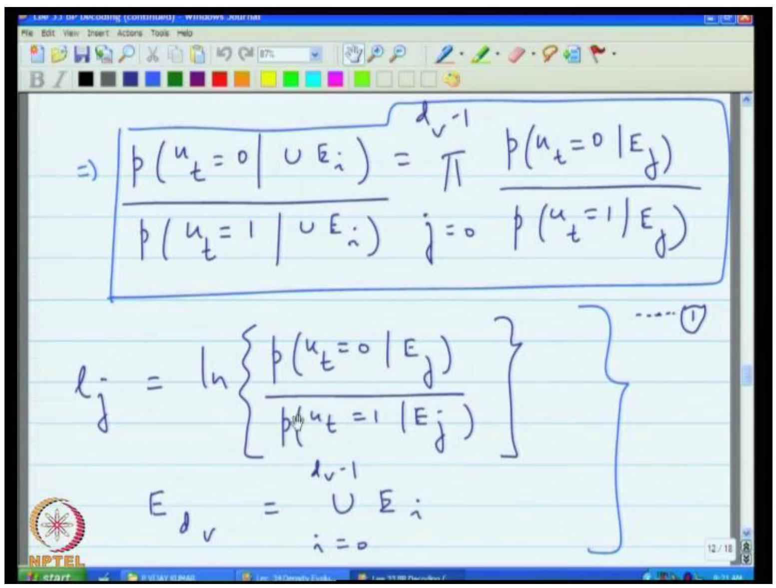

And more precisely in terms of the g d l, we know that these, because of the way g d l passes messages; because at variable node, the g d l would just collect all the incoming messages multiply them together and send them all.

(Refer Slide Time: 07:00)



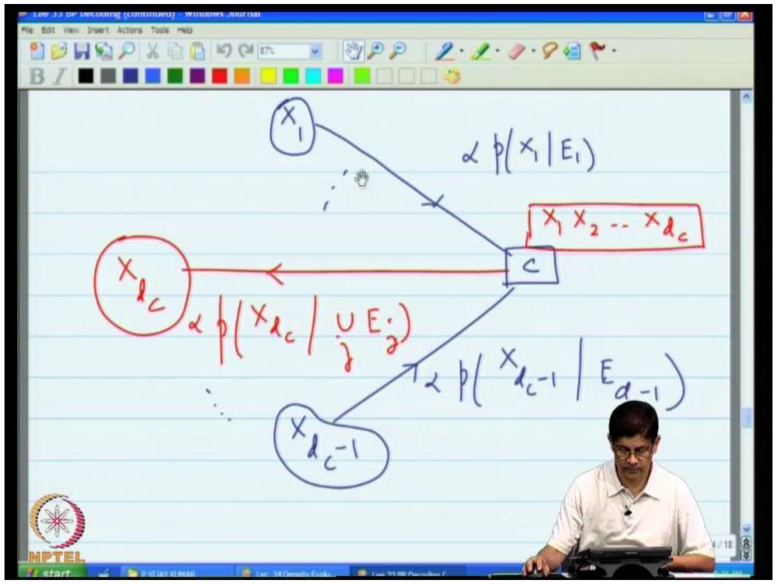

That is what we are doing here. And this proportionality is included, because well for reasons that we seen earlier. Now, we are going to do something slightly different from what we did under belief propagation, because we are going to actually interpret rewrite these messages in terms of log likelihood ratios. So, in particular here, where as this is a belief whenever going to pass on to the ratio of belief, which is a likelihood ratio and then dominantly pass to ratios; that is revaluate this remember that under belief propagation we are passing our g d l we are passing functions, but now we are evaluating the function at a particular value of x t 1 and minus 1 in this case, when you compute the ratio at this proportionality constant goes away and well after the equality. So, you have likelihood ratio here on the left which is the product of the likelihood ratios.

(Refer Slide Time: 07:53)




So, then we pass on then we take logs on both sides, and that leads as to a simple expression that is the log likelihood ratio at the outcome is simply the sum of the likelihood log likelihood ratios at the input, so very clean and simple.

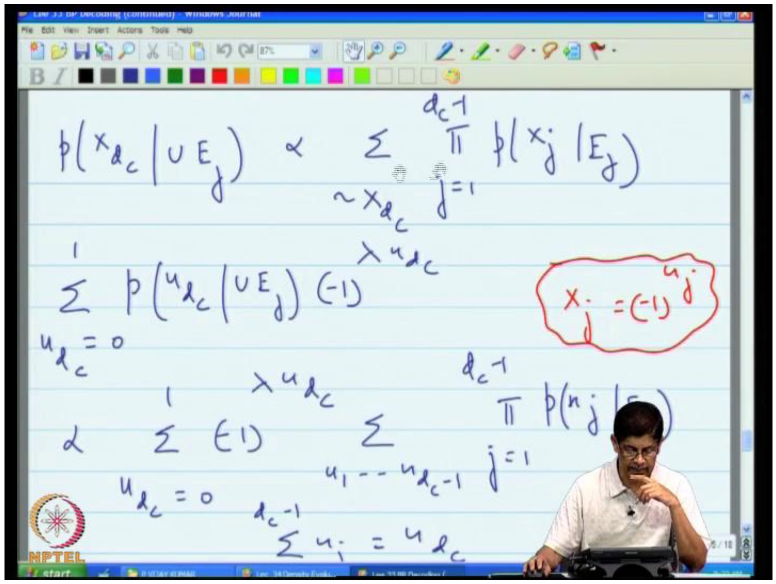
(Refer Slide Time: 08:11)




So, that expression is well what happens on the check node. Well thinks at a check node have slightly more involved, because there is a marginalization that actually take place. So, if you will recall at a check node what actually happens is these are in fact, the incoming messages, very similar to the types of messages there are incident on a variable node; what goes out is also very similar, but the relationship between input and output is like this, because they have the marginalization here.

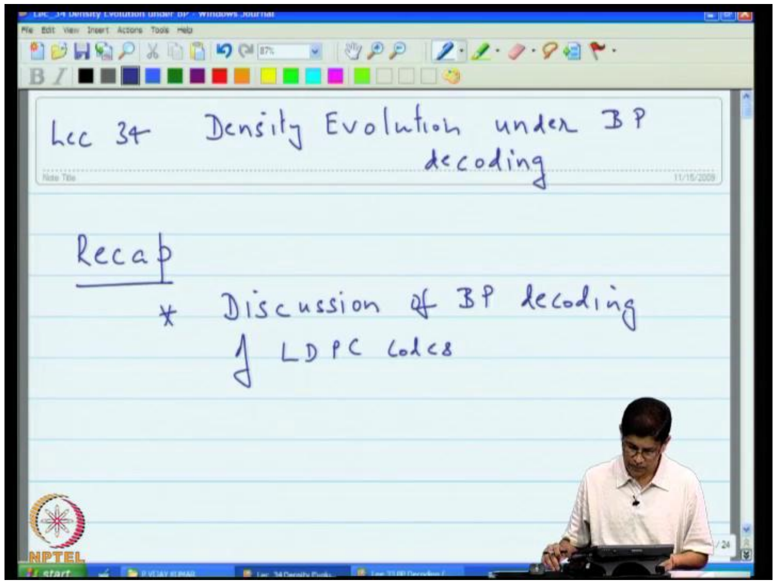
Because, if you think of this is the g d l, this check node is really a function of several variables right. So, at the local domain in the terminology of a junction tree would list all the variables. Now, on the other hand you are sending a message to a variable node that has a single variable. And therefore, the local domain consisting of a single variable, so between here and here you need to marginalize with respect to all of the other variables. So, that is the expression over here.

(Refer Slide Time: 09:13)



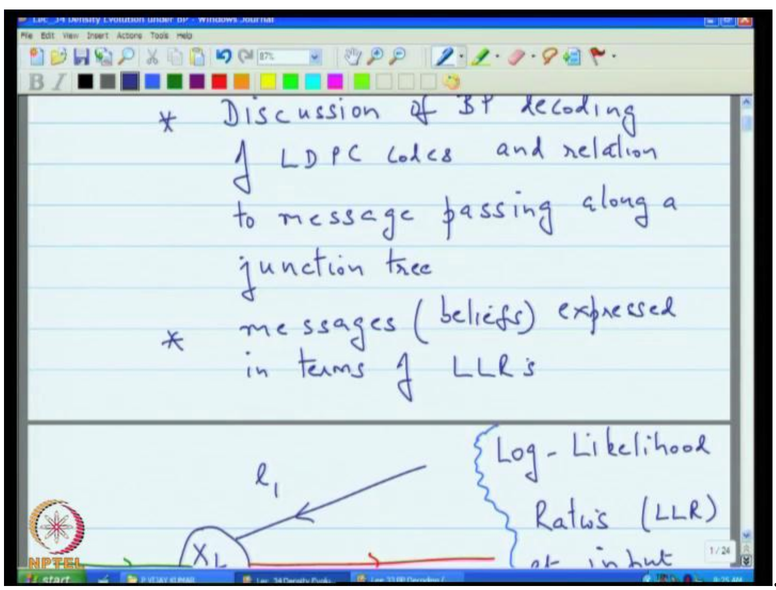
Now, once again just as we did in the case of a variable node what we would like to do is we would rewrite this in terms of log likelihood ratios. So, how can we do that? Well, so there is there is a technicality here we go through some manipulations which have interpret here, but when you are when all is sudden done what you actually get is this relationship between input and output. You get that the log likelihood ratios at the output are related to the log likelihood ratios at the input like this. So, this is where we will actually we pick up our lecture from, so because I have gone through this in some detail I am not going to put down a summary, but I am just goin g to jump or maybe I will just very briefly very quickly and briefly summarize what we did?

(Refer Slide Time: 09:55)

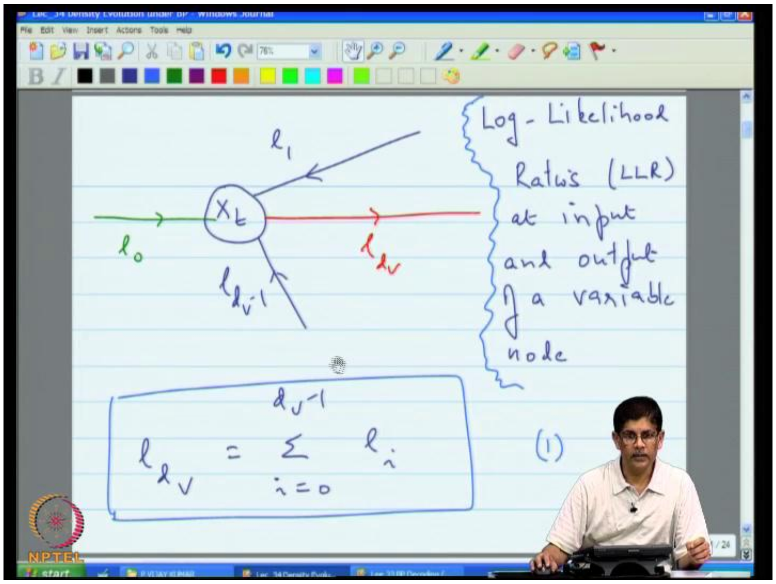



So, the recap in today is lecture of the last lecture was a discussion of belief propagation decoding of LDPC codes.

(Refer Slide Time: 10:40)



.

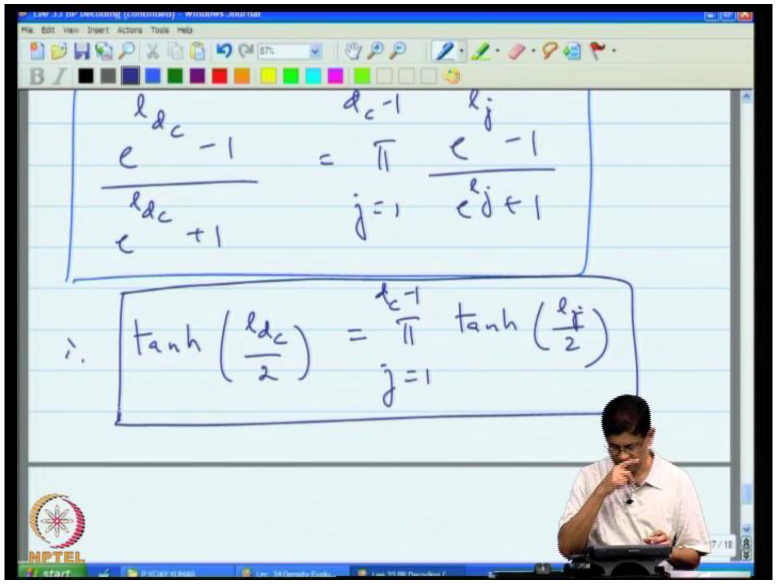



And particularly and relation to message passing along along junction trees along a junction tree; and then we expressed a messages or a rather messages and in brackets I will write beliefs expressed in terms of log likelihood ratios.

So, that is our summary and now we want to pick up the third in today's lecture. Now, today is lecture is going to actually talk about something called density evolution. In which is really similarly to what we did in the case of Galagus decoding algorithm a, there what we did was we first I outline what the decoding algorithm was, and then I have tried as far as performance analysis was concern, we were in interested in trying to actually characterize as well, if you have to run this algorithm. What is it is performance in terms of the number of erroneous messages that are passed. So, we want to do something very similar even here. So, because the calculations tend to get little bit mercy what I thought I would to do in this lecture was actually prewrite the calculation. I have pre written the lecture, and I realize that when you pre out a lecture, the lecture tends to go little bit fast I will try to slow down a little bit to compensate, but here is to recap.

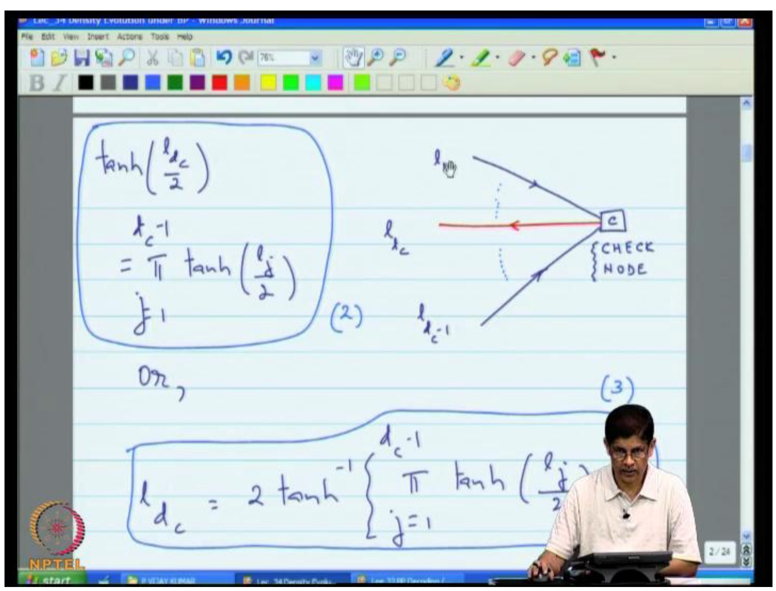
So, at a variable node we intro the conclusion that beliefs the equivalent message passing rule in terms of log likelihood ratios at a variable node is simply to take the some of the incoming log likelihood ratios including the one that comes in from the channel at a check node.

(Refer Slide Time: 13:39)



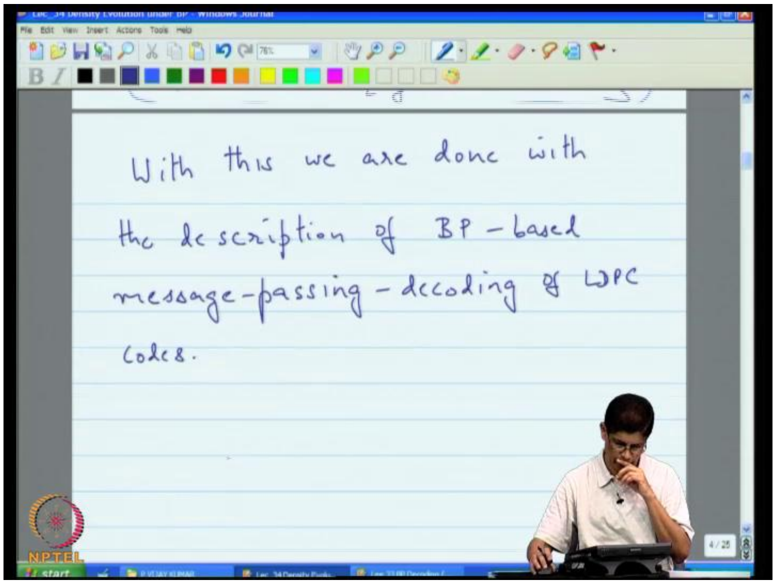
Now, at a check node although it was not obvious it is clear and I will just perhaps put this down in your last lecture itself from this it follows that therefore tan hyperbolic (l d c by 2) is the product j is equal to 1 to d c minus 1 tan hyperbolic (l j over 2). So, that was the tail end of last time's lecture. So, we pick up the thread here.

(Refer Slide Time: 14:13)



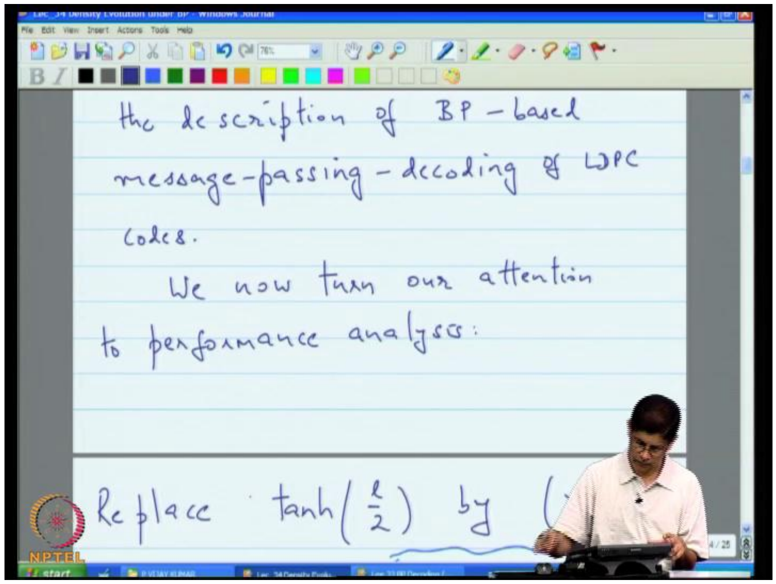
So, we know that the input output ratio at a check node is in terms of the tan hyperbolic function alright. So, the incoming are the allies the out coming is l d c. So, you can convert this in and say that l d c is therefore, twice the inverse tan hyperbolic function of the product. This describes the input output relationships. So, once you know this you already know how to actually decode because you know the messages that are going back, and forth I have told you how with how messages are passed across variable node and check node and you had done.

(Refer Slide Time: 15:08)



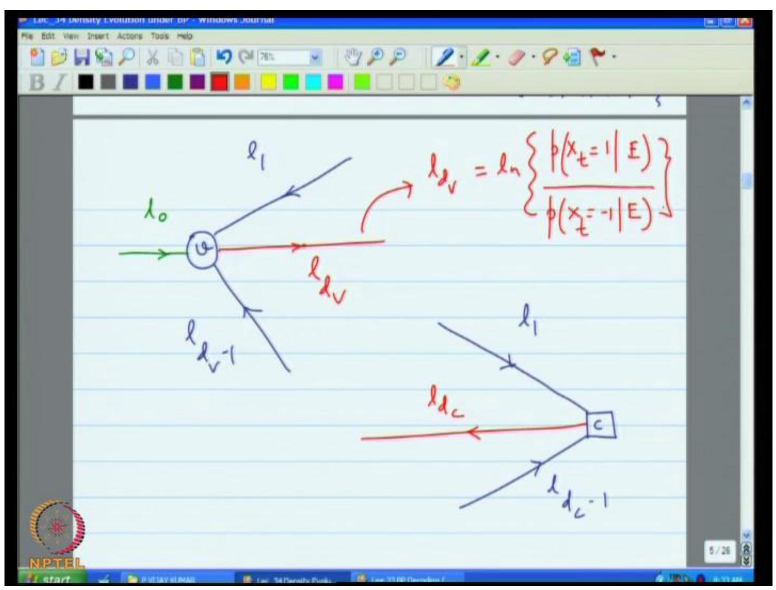

But now, we want to start performance analysis. So, may be just to actually enforce size that and you just insert a page and you say I just add a note saying with this with this we are done with with the description of belief propagation based message passing decoding of LDPC codes.

(Refer Slide Time: 16:04)




We now turn our attention to performance analysis, and the performance analysis will be carried out in terms of will be carried out we had density evolution. So, this is to be carried out using density evolution.

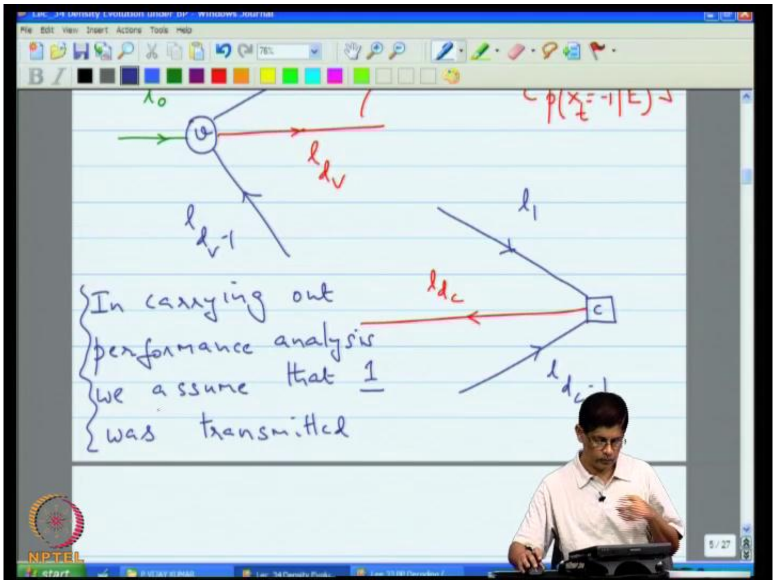
(Refer Slide Time: 17:37)




So, what does that mean? It means that so perhaps I should just say that may I should pass to illustrate that a little bit. So, what we mean by that is let see, that you have a variable node here, and you have incoming and outgoing messages l 0 l 1 l 1 l d v minus1 and l d v is your outgoing message.

And similarly at a check node and at the check node we have incoming outgoing messages which are expressed in terms of log likelihood ratios. And a remember that in here log likelihood ratio. So, it say this one l d v is l n of p some variable x t equal to 1 given some evidence divided by p of x t equals minus 1 given certain evidence.

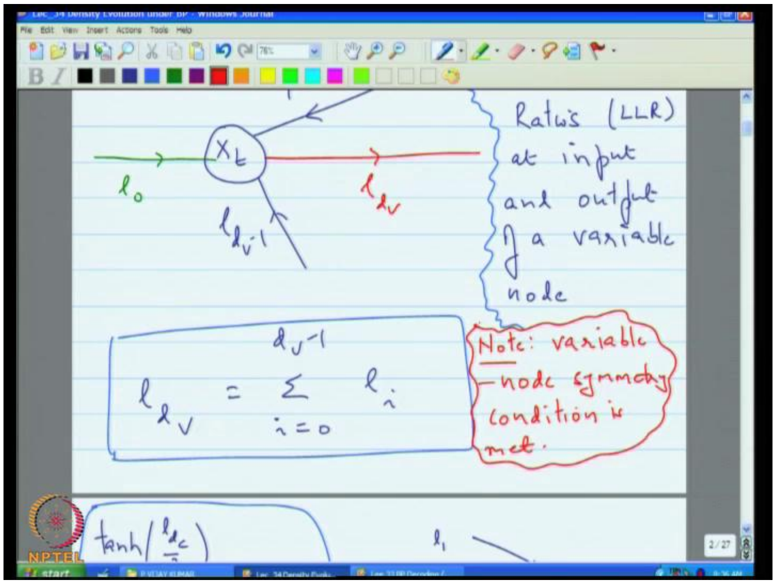
(Refer Slide Time: 20:07)




Now, when we do performance analysis we assume that the all one code word was transmitted. So, put that down here in carrying out performance analysis, we assume that the all 1 code word was transmitted. Now, we discuss this sometimes earlier when the issue of well in general, when you are doing performance analysis of LDPC code.

We said that look we can always assume that the draw 1 code word was transmitted provided we had certain conditions that were made. There is you needed the channel symmetric condition to be met, and you needed that there was symmetry in the messages that were passed across variable node and you needed symmetry in the messages that were passed across a check node. And in fact you can check that that is very easily the case, because here…

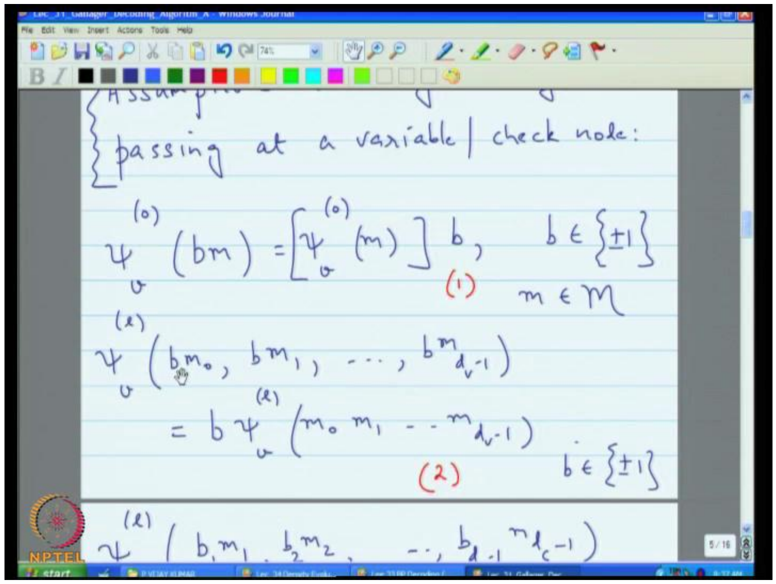
(Refer Slide Time: 21:30)




So, let us take a look at there. So, at a variable node at a variable node you have that the output messages simply the sum of the incoming messages, variable node symmetry requires that if you change the sign of all your incoming messages, the output will also change sign. So, that is clearly declared here. So, let says make a note of that by saying that I will remove this when here create a little bit space, and put down that note that the variable node symmetric condition is met.

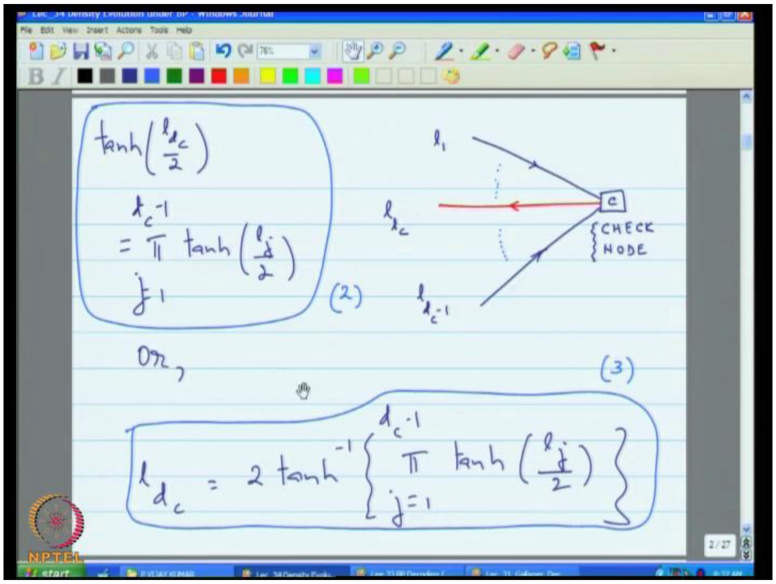
So, the variable node symmetric condition is met. At a check node, what we have is that the input output relationship is in terms of the tan hyperbolic, and our check node the check node symmetry condition that we actually required was that if you put a sign on each of the incoming messages, then the sign on the output should be the product of the incoming signs up. Let me see if I can actually pull data.

(Refer Slide Time: 23:33)




So, remember these are the at a variable node is said that if you multiply all the messages by a constant b, which is e the plus or minus 1 then the output also be multiplied. And we just check that was true for case of belief message passing based on belief, at a check node you need that if the incoming messages are signed with individual signs, then the outgoing messages are signed with the product of the signs. And I just want to actually show you that thus the keys here.

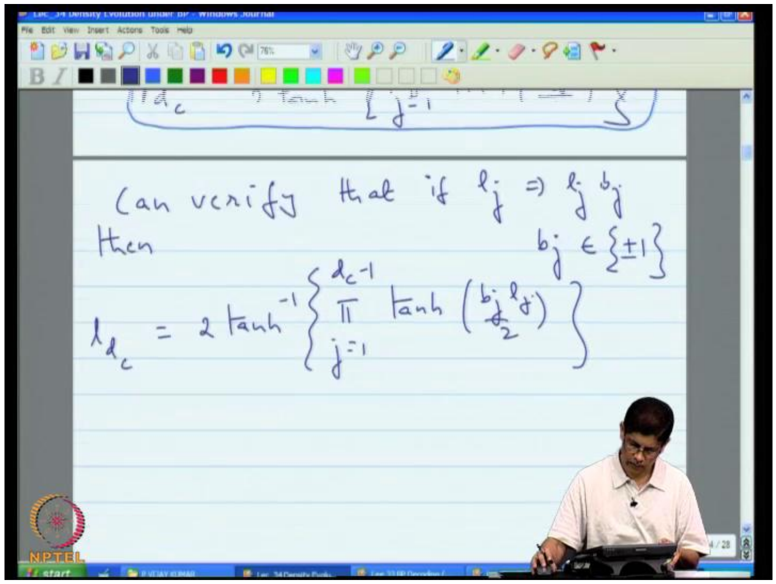
(Refer Slide Time: 23:58)




Well whiles that because supposing each of these incoming messages was multiplied where sum d, then the way then each of the tan hyperbolic function causes them to be pulled out of this, and then again the inverse is similar twice that. Well take a look at the tan hyperbolic function which have actually sketched, which is a sketch little bit later here.

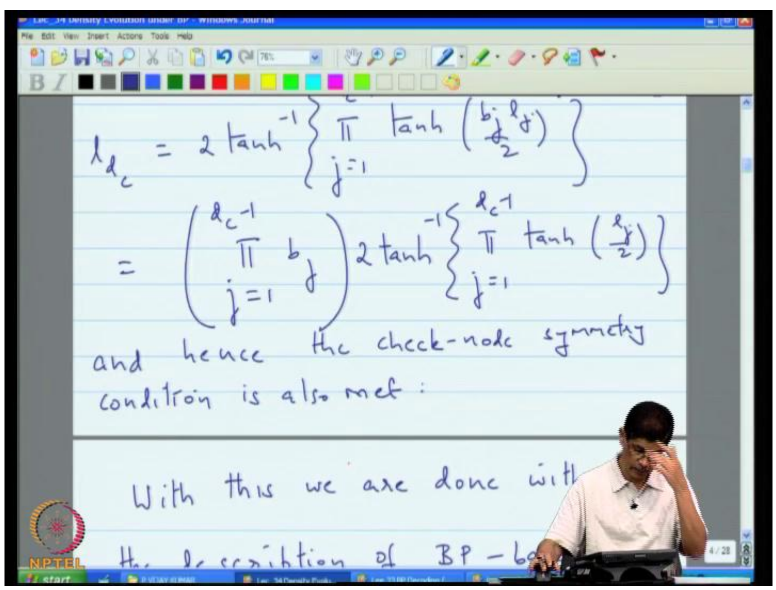
Here is a sketch of the tan hyperbolic function. So, you see that the tan hyperbolic function has the same size of the input, and in fact if the input reverse a sign as symmetry the tan hyperbolic also reverse a sign. So, what that means in particular over here is a so perhaps I will just insert a page here.

(Refer Slide Time: 25:13)



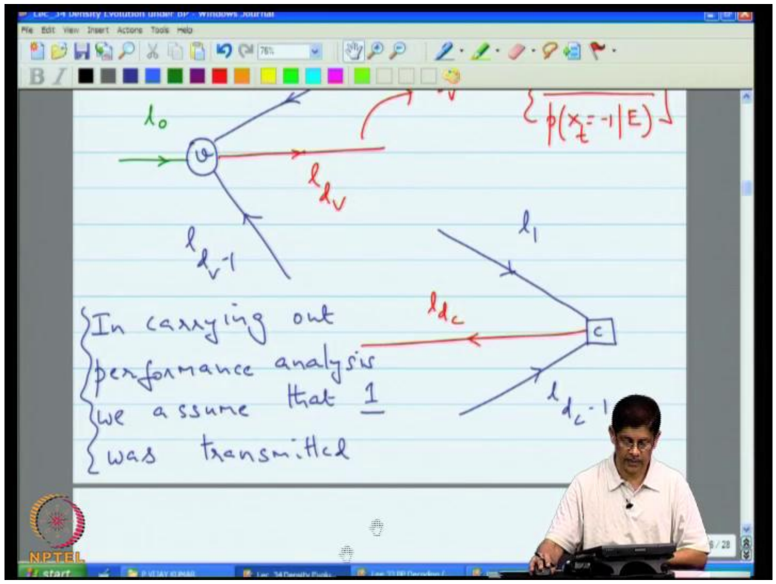

And say that so can verify that l d c that if you replace each l j by l j times b j by sign, where b j is either plus or minus 1 then l d c. The result in l d c is equal to 2 tan hyperbolic inverse the product.

(Refer Slide Time: 26:16)



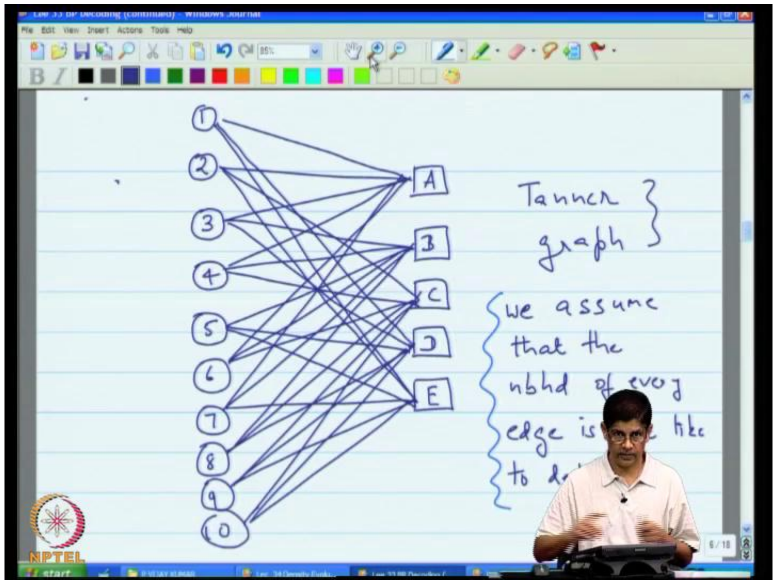

You can verify that it first of all pulls out here, and in eventually pulls out of this. So, you actually get this. And you get and hence the check node symmetry condition is also met.

(Refer Slide Time: 27:25)




So, coming down here, so now without we have justified in making assumption in carrying out performance analysis that the all one code word was actually transmitted, so that is why we now. So, we want to do belief propagation decoding and the way you do belief propagation decoding was actually say you know what? Every time I pass a message from a variable to a check node, I am going to regard the message. So, because this is an iterative process, it is an iterative process that is actually carried out on the tanner graph.

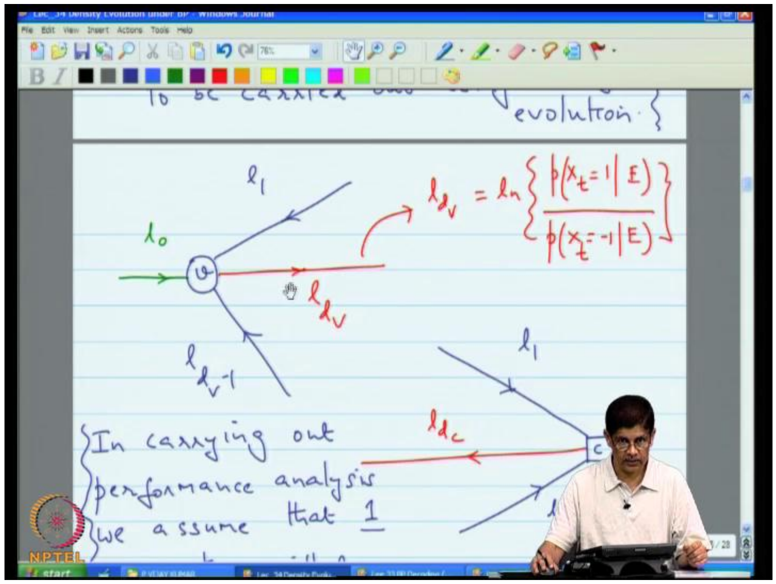
(Refer Slide Time: 28:16)




Here is a tanner graph and in belief propagation decoding what we are doing is, we are passing messages back and forth along these edges, and each of this is a log likelihood ratio. Now, we will say that a message is incorrect and we know all the code symbols can be assumed for performance analysis to be assume to be all one. So, and if your belief is correct, then since your belief is the log likelihood ratio over here, if you look over here, if in fact, this is positive; that means the probability there it is a 1 given e is greater than the probability there it is minus 1 which means that this quantity is greater than 0 right. So, if this quantity is greater than 0, then you are going to conclude that well my current belief is that is actually a 1, that 1 wins over a minus 1.

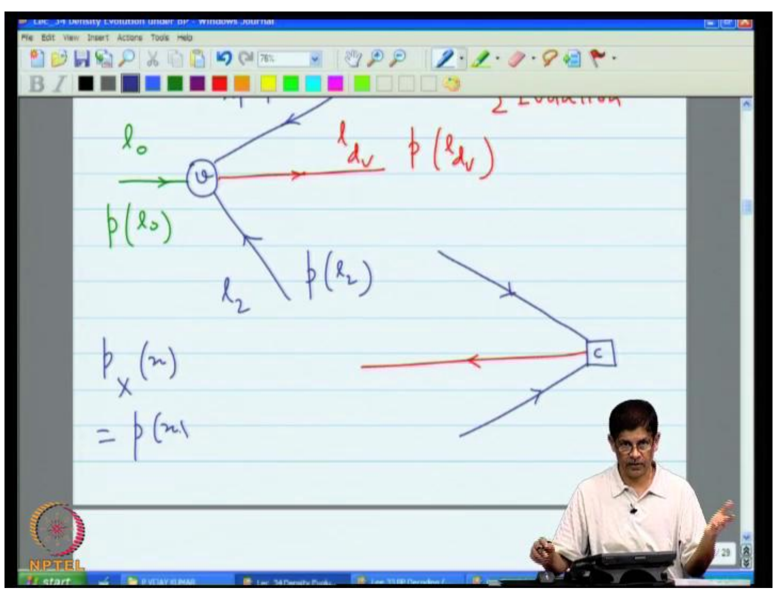
In this sense we are going to look at these message, and look at the sign of these messages; if the sign of the message is positive, then we are going to say that is a correct message. If the sign is negative, we will actually say it is an incorrect message, but how do we actually analyze, how many or what is the probability with this there incorrect or correct. So, what we are going to do instead is well, we will actually trying to do something more.

(Refer Slide Time: 29:43)




Because it is easier will not only examine the sign of these messages will actually examine their density function. So, now our interest is in saying well we actually want to see, so I let me duplicate this slide and I will put some writing on it. So, let me get rid of some of (( )) writing here.

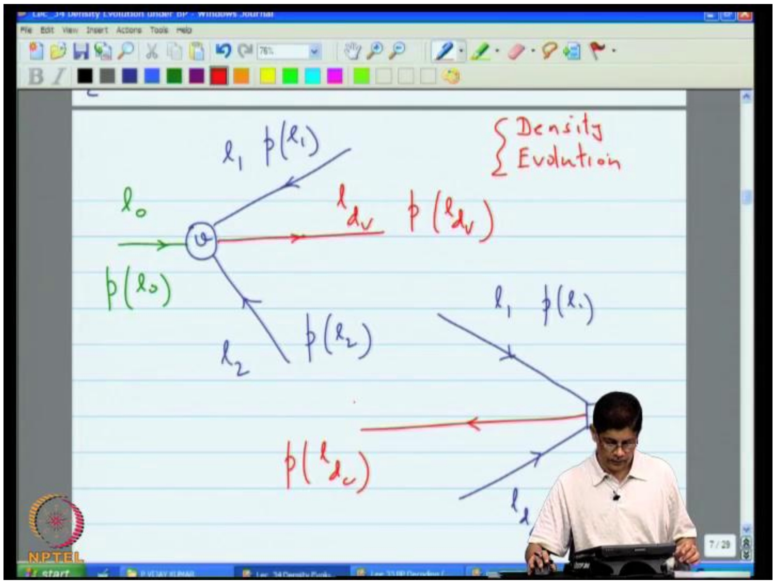
(Refer Slide Time: 30:40)



This time what we are going to be interested in knowing is not just the likelihood ratios, but rather on their densities. So, we are going to be interested in saying well look there is an l 0 here, that is true but it has a certain density function; there is an l 1 here, and there is an associated density function, and similarly over here there is an l 2, and there is a p of l 2. And at the output I have an l d v which is associated with a certain l d v and so what we are saying on a density evolution is that earlier we were varied about the messages, now we have varied about the density function.

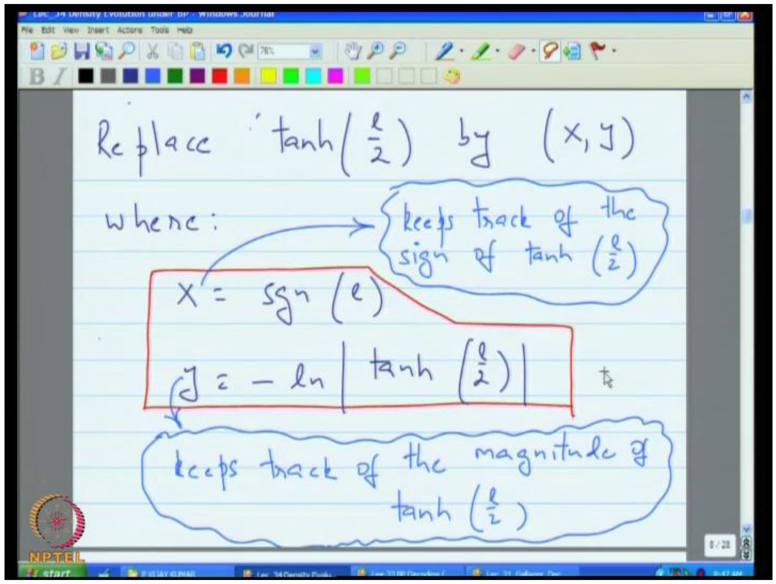
So, knowing the relationship between this variable and this other variables, we want to translate our knowledge of that relationship to actually derive these densities in terms of the other densities. By the way I am kind of using short form, because normally what you do is you would write p x of x, but I am just writing p of x saying that the underline grand variable is understood. So, hopefully that is actually clear that is a short form notation.

(Refer Slide Time: 32:41)



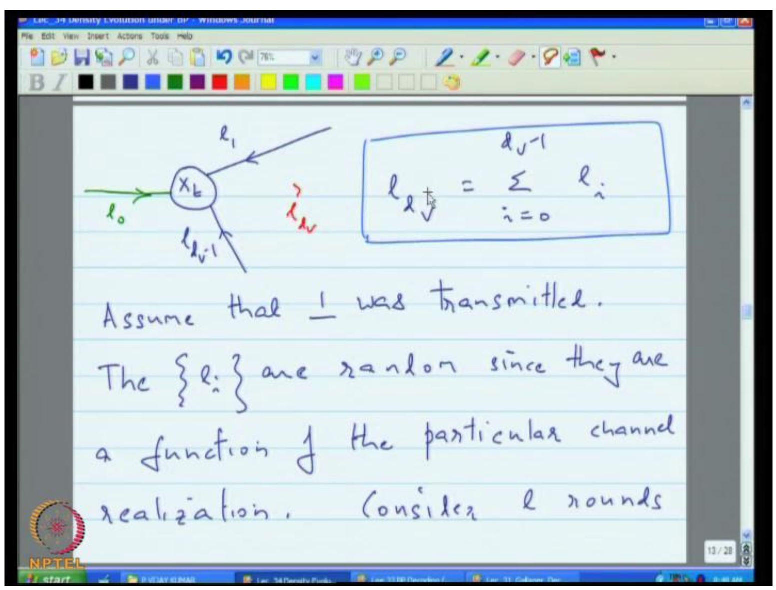

And the same thing here; that is you have an l 1 here associated with the p l 1 and so on. So, that is our next task is to actually compute density functions. And I just wanted to introduce some or let us do one thing, let me get rid of this page first here we go.

(Refer Slide Time: 33:30)



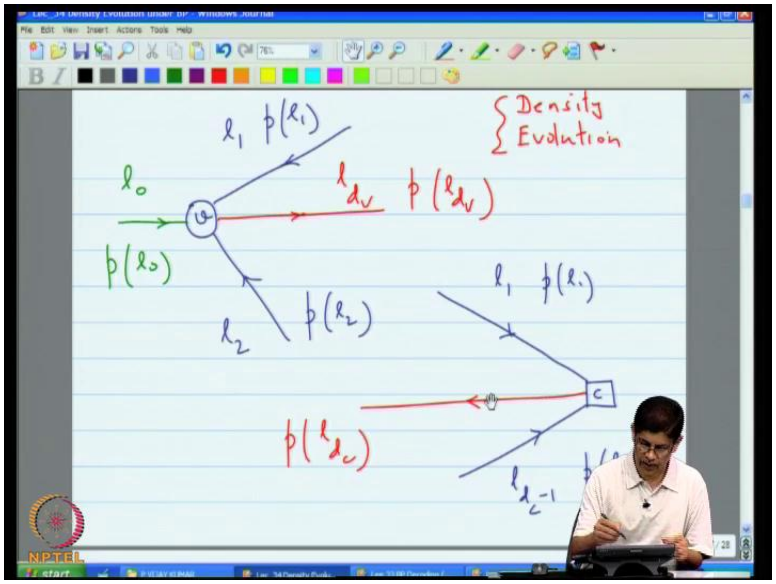

So, towards will actually do a few manipulations, before we get to the density evolution stage. So, I want to slightly change the order in which I carry out this description. So, let me just skip a head and go over here.

(Refer Slide Time: 34:59)




Let us start the discussion at a variable node and at a variable node, we know that is so I take that back. So, I see that my pages are in a certain order. So, rather than confuse you may describe them in the order in which they are, let us get back here. So, we were talking about carrying out density evolution which says let describe the densities in terms of the incoming densities, but before that they has some computations that will be found helpful. So, I am going to request your passions, because right now is not completely motivated.

(Refer Slide Time: 36:00)

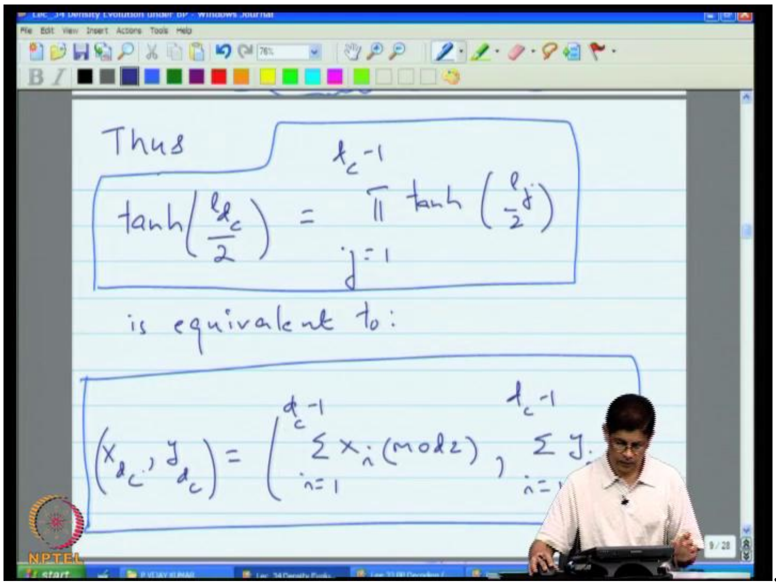


But we know that what happen there in terms of the messages is that here, the likelihood ratios simply the sum of the incoming likelihood ratios, here it is little bit more complicated. Because you have you have that your l d c the incoming and outgoing likelihood relationship is through the tan hyperbolic. And that present some challenges, so we would like to overcome them, and we will actually do a change of variables. What we will do is we will replace the tan hyperbolic function by a pair of a variables, so x will actually keep track of the sign of tan hyperbolic of l by 2, but because the tan hyperbolic function monatomic in l, and sign preserving the sign of the tan hyperbolic is the same as the sign of l.

The second is to keep track of the magnitude of tan hyperbolic, and will keep track of it by making use of the log function, and it will be convenient to take the negative law. But basically what we are trying to do is we are trying to say the reason for this is we going to say look at a variable node, we have this very simple relationship in terms of log likelihood ratios, here it is more complicated the relationship is terms of the tan hyperbolic.

So, the first thought that comes to my mind is well why do not we take logs on both sides then it will become an addition again. The only problem with taking logs on both sides is that this is the quantity that could be the negative or positive. So, you cannot take the log of this because when it is negative, it is not even defined. We get on that by saying well let us split this into two variables or two functions; one function which carries the sign information, and a second function which carries the magnitude. So, that is what is actually happening here.

(Refer Slide Time: 37:47)

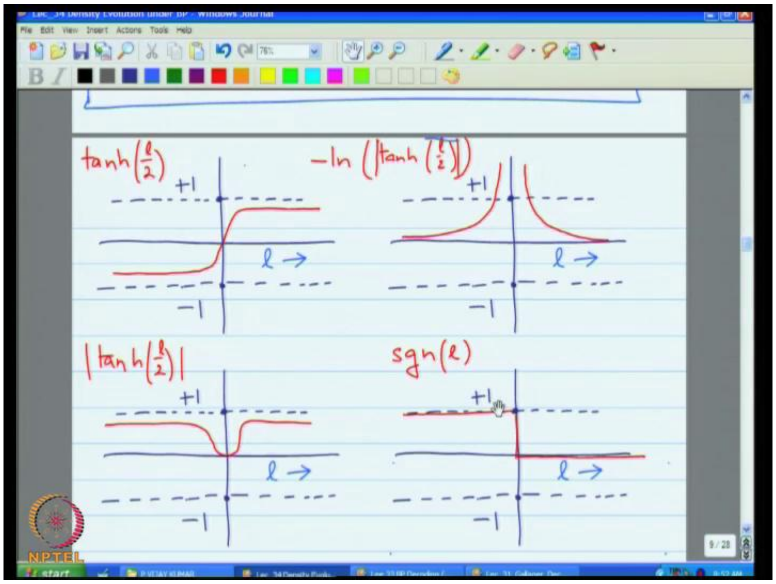


.

So, we are x keeps track of the sign and y keeps track of the magnitude, so then our original relationships in terms of type of tan hyperbolic. Can you can show there it is simply replace to saying that look the sign at the output is so this should be x d c and y d c, let me correct that; this is x d c y d c. So, that is the output x, and y's are related to the input x and y is simply by summation.

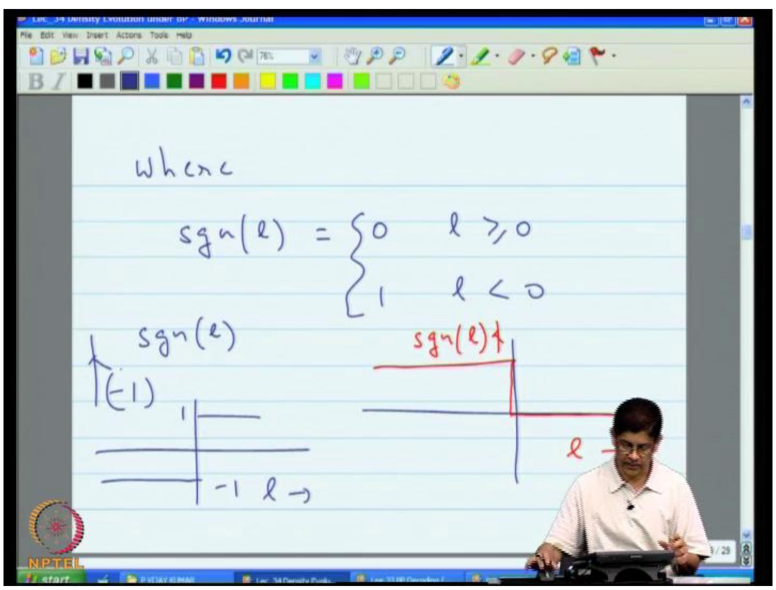
And that is not have to see, because each of this is the log of the magnitude. So, since the magnitudes are multiplied by logs are added, so that is why have this relationship. And similarly the sign you can also verify that this is true, because each of the x i is keeping track of the signs of the individual tan hyperbolic and the sign of the overall is in terms of the product of the signs.

(Refer Slide Time: 38:57)



But here, what x d c is doing is it is keeping track of the sign. So, so the onto what do you mean the sign function. So, the sign function is defined in a somewhat in a conventional way.

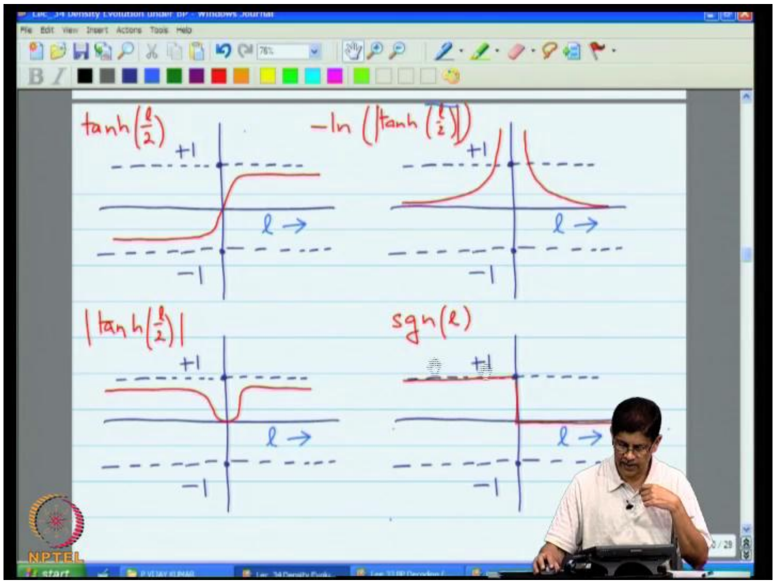
(Refer Slide Time: 39:15)



So, where sign of l is 1 a sorry 0 is for l greater than or equal to 0 and is 1 for l less than 0, that is the sign of a say you have to plot this, it would look like this it would be something that is 1 in the negative part, and then it would be 0 for the positive part. So, here I am plotting sign of l verses l now. What we conventionally think of is the sign might be or is rather is what is minus 1 to the power sign of l.

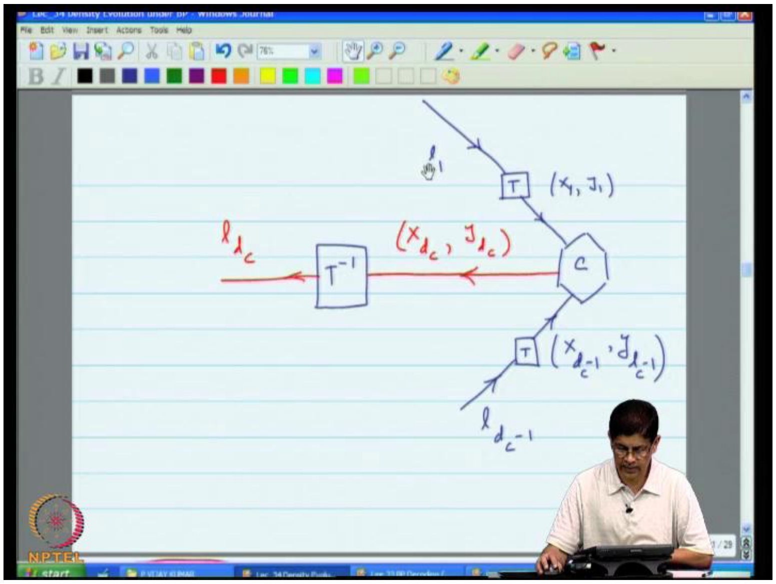
Because this function is more like the function we would have in mind there is it is 1 here and it is minus 1 for negative, but however it is convenient. So, this is a binary 0 1 function here, and this is our traditional plus minus 1 more convenient to work in terms of this. So, that is the reason while where as in the plus minus form domain, you would simply have your multiplication of the signs in the 0 1 domain, it turns out to be an addition. And the next thing about it is now that we have addition on both sides on both terms, we actually have addition all though one addition is modular to and minus this alright.

(Refer Slide Time: 41:00)




And this is your sign. Here is l tan hyperbolic when you take the log, you get a function that looks like this when you take a negative log, it looks like this the sign is this function.

(Refer Slide Time: 41:14)

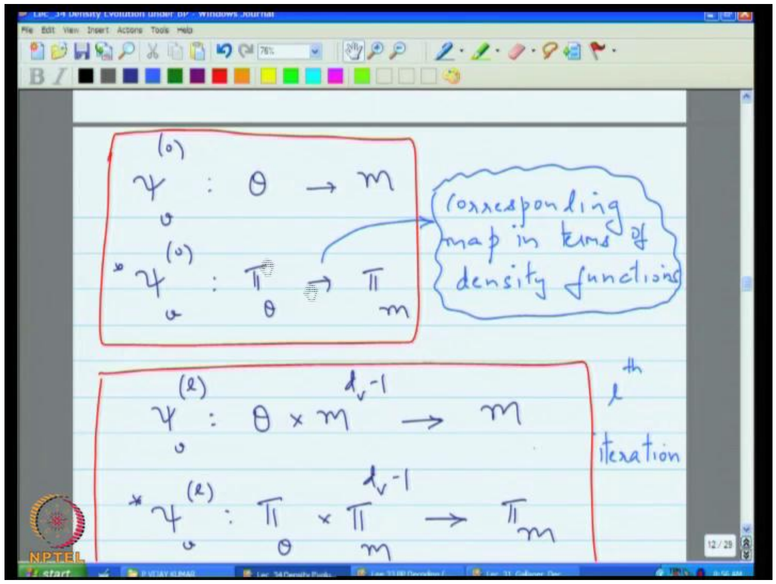



So, then if you want to represent, now in terms of this transformation what is taking place at a check node you see that having the incoming log likelihood ratios. Then there is this transformation t which takes an l to a x and a y exactly like will actually shown here, that is given an l you pass to an x and y using this map. That is where the transformation is actually carried out. Similarly, on each of the incoming messages you transform the corresponding log likelihood ratio into a a pair of functions.

One representing the sign and one representing a magnitude and then this check node simply has to actually compute the sum according to this expression over here. It is just going to compute the sum here, and then after you computed. So, x d c is the modular two sums of the incoming x j is y d c is the real sum of the incoming y j is and then you do a inverse transformation which takes you back from here to here.

So, we made this change on the check node from this other picture here, because the picture here, involve the tan hyperbolic function, and we wanted to get rid of that so we have this.

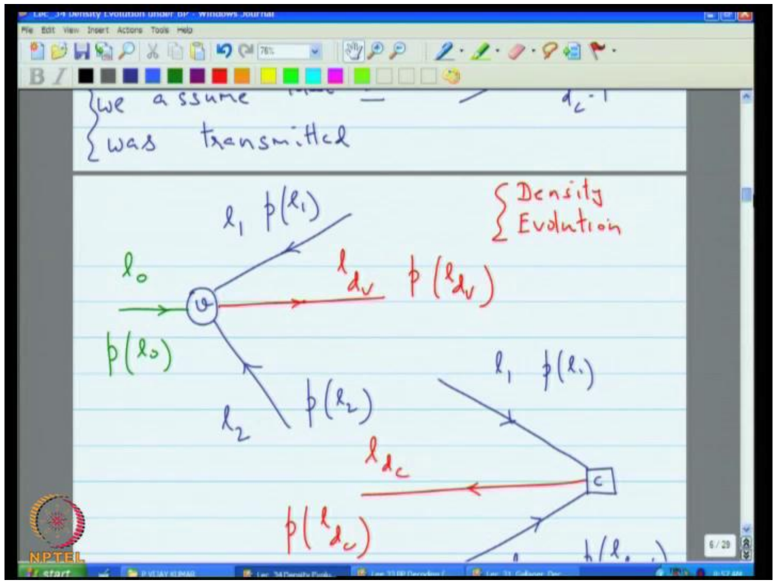
(Refer Slide Time: 42:46)




Now, density evolution says that look in abstract it says that look where as earlier we acknowledge that at every variable node you had a map, in is in the initial iteration that is in the 0 th iteration when the only input is from the channel the map is from the channel output alphabet to your message alphabet, and associated to that because anytime you have a map from one variable to another you also map that the density function on that variable towards density function on this run variable. This is the corresponding map in terms of the density function.

The distinction between this and this is the presents of this star. We will say that this mapping here induces this mapping on the density functions. Similarly, at the l th iteration, there is and you can see that here, in terms of messages there is a mapping from the channel output alphabet and the message input alphabet to the output message alphabet, and which is what this is there are d v minus 1 incoming messages.

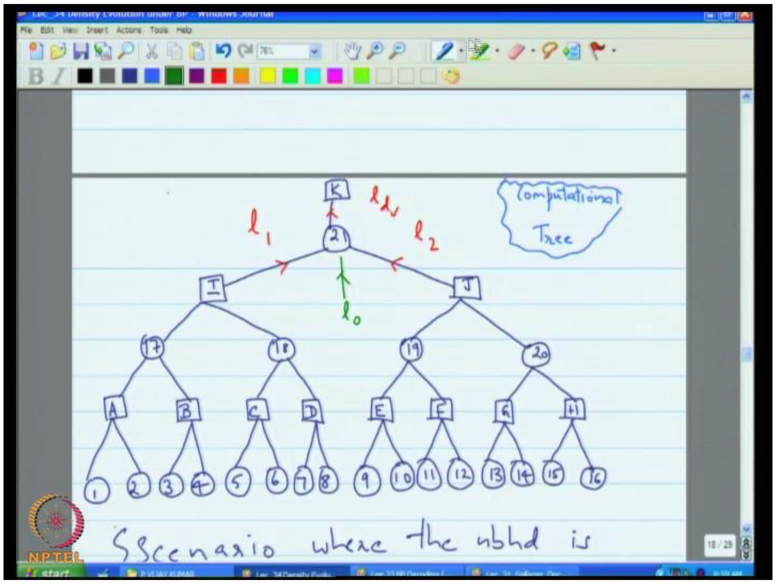
(Refer Slide Time: 43:42)




We assume that the message alphabet along any edge is the same, but that the channel income output alphabet could be different. So, this is the mapping that is carried out at a variable node and the corresponding mapping in terms of density function is given by this put the star only thing is that you see that I actually take I am taking that mapping on the density functions. Actually I have a product of density functions here.

And that is should be some we need to pass that because I mean why do you how you justified in taking the product; that seems to employ that your incoming variables are random or linear or statistically independent. And that is the case, because you see your output here is the sum of this input variables.

(Refer Slide Time: 44:51)



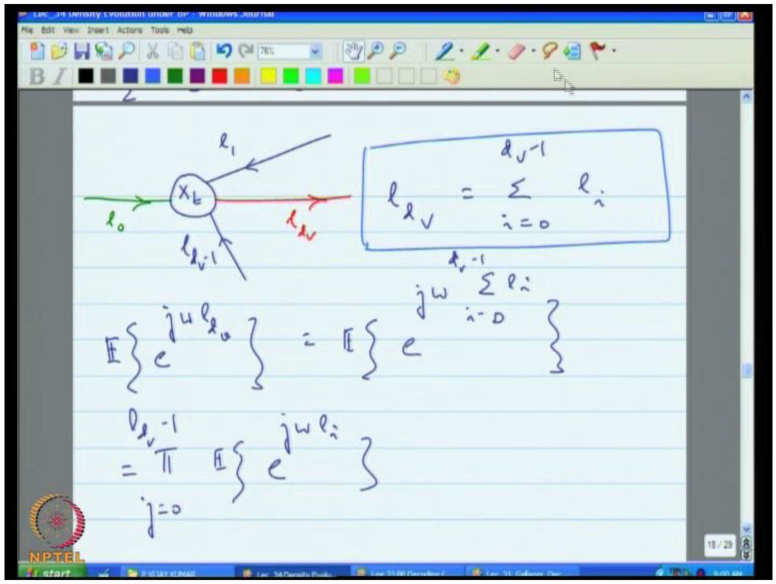

But if you assume that the neighborhood is tree like right; that means, that if you unravel the tanner graph you will get a graph like this. What we want to say point out is that look the message that is been passed the message, that is for instance that is being passed here. And the message that is being passed here, this message is now a function of all these received variables. This message which is a likelihood ratio so this is for instance this would could be an l 1 this could be an l 2. and what actually goes out is an l d v there is also a channel input which is an l 0.

So, now each of these inputs, so l 0 is a function only of the corresponding received symbols. That is y 21 l 1 is the function of all these received symbols and also a makes use of the fact. That there are certain parity conditions that the code symbol satisfied l 2 corresponds to this branch of the tree. And therefore, uses these received symbol knowledge as well as these parity checks. Since the received symbols are independent given the transmitted code word these messages are independent, so that is important.

So, the incoming messages under the tree like assumption can be assumed be independent and this is the reason. Why you actually write the product density function, why you write it in terms of the product of the density functions rather than adjoin density function.

Similarly, at a check node the messages are make from d c minus 1 fold messages and this is the corresponding message on the map on the density functions.

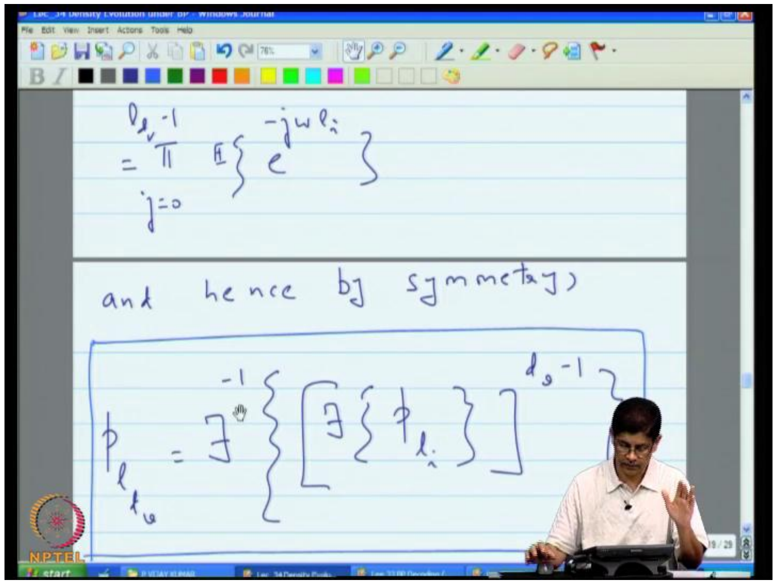
(Refer Slide Time: 46:53)



So, we want to actually say that look I now we are getting down to specifics, we know the input output relationship in terms of messages is given by this. What can you say about the densities? So, the trick is to actually where to the characteristic function, we take the characteristic function here, and because these are independent this expression breaks down to the product of individual characteristic functions.

But, any characteristic function is more or less the Fourier transform apart from a sign. So, In fact, let us in order to make it exactly the Fourier transform like simply put a negative sign on both sides. So, then the characteristic function as defined in this manner is the product of these by independence, and each of this is a Fourier transform. So, what that means is that?

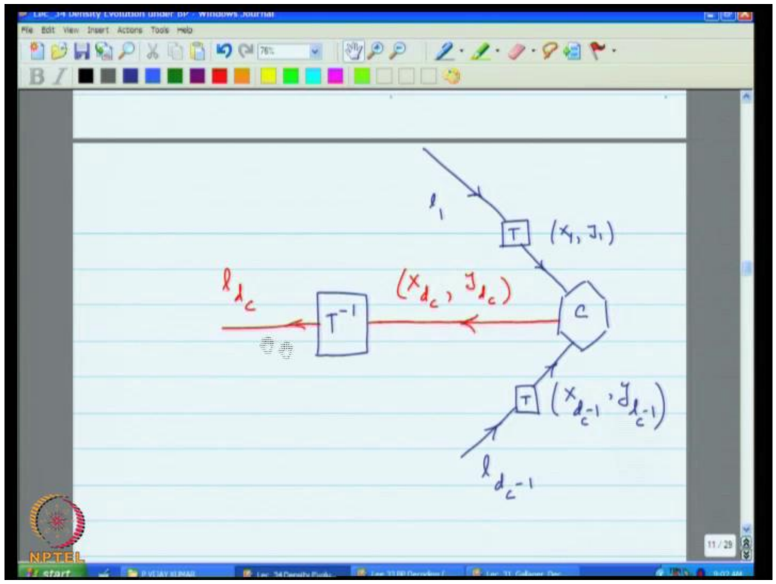
(Refer Slide Time: 47:50)




The density function of the output message can be obtained by taking the Fourier transforms of the incoming density functions multiply by them together. Multiplying them together, but a symmetry there are all actually identical. So, you can race it to this power then you take the fully inverse.

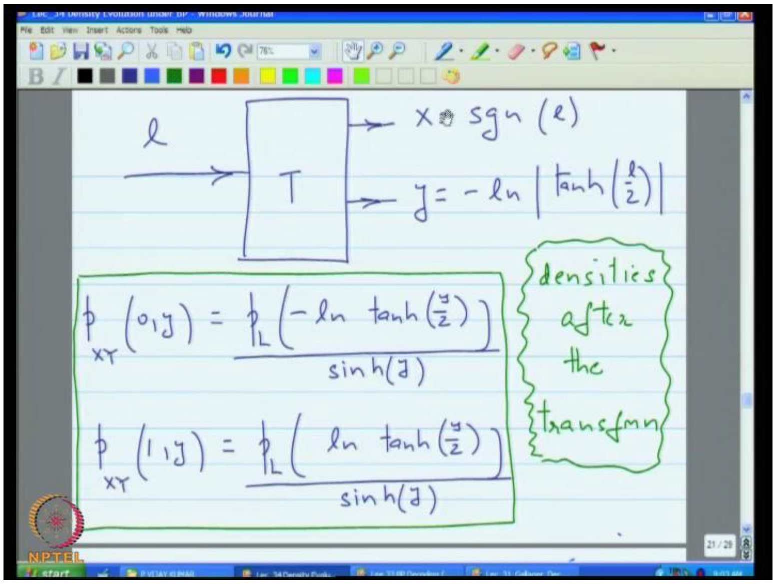
There with this we were actually accomplished density evolution at a variable node that is where able to relate the density function of the output message to the density functions of the input. So, that was attentively end list at checks node things are easy, but more complicated where is that?

(Refer Slide Time: 48:32)



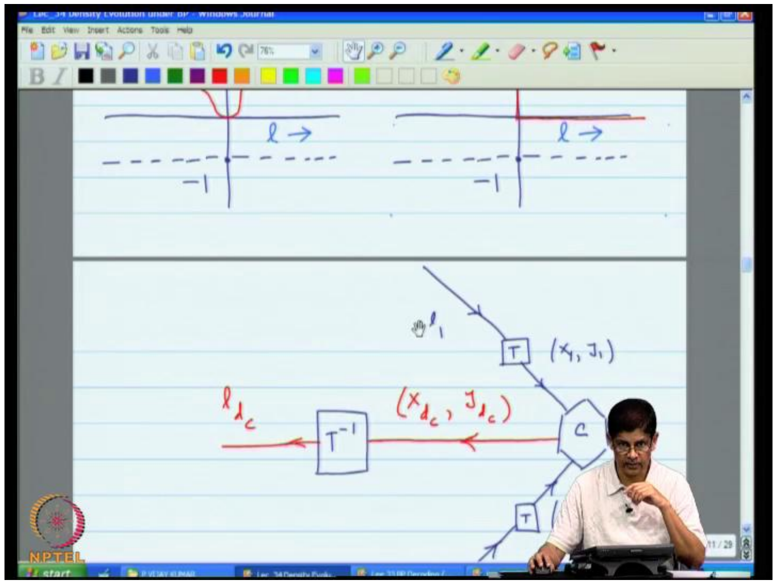

Because now here check node look likes this the mapping at a check node is in fact, a series of first of all each of the individual messages is passed across a transformation tree, and then there after after the check node carries out, it it is map then there is an inverse transformation. So, in some sense we have to actually trace the evolution of the density as it goes across the transformation. So, that is step one across the check node that is step two and then across the reverse transformation. One we now have to keep track of three transformations of the density. And that is what will actually did.

(Refer Slide Time: 49:15)



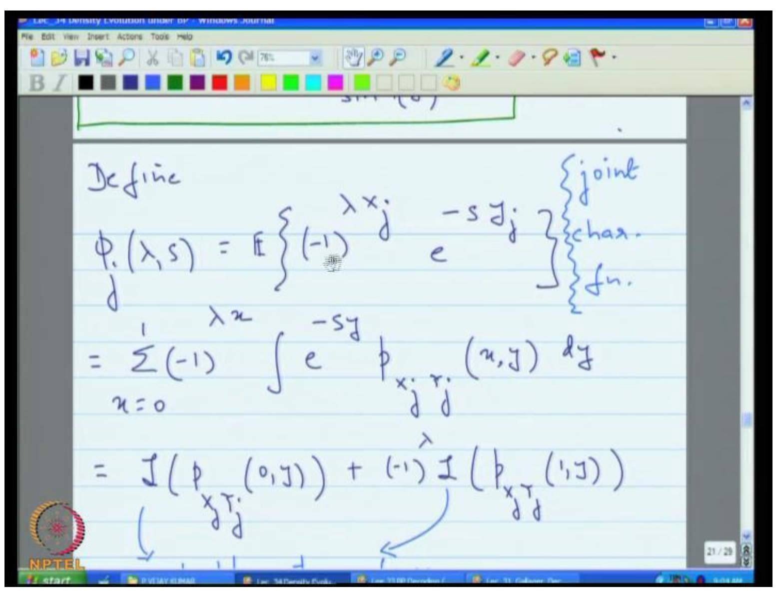

So, first step there is incoming density incoming log likelihood ratios transform from an l to an x and a y, and you can show that if you have given the density function of l, then they join density function of x and y is given through this pair of expressions. Why does it look like this, because x is a discrete random variable y is continues. So, x can take on values either 0 or 1 where as y being continues. So, we break this joint distribution function into two parts 1 which corresponds to x equal to 0, and 1 is corresponds to x equal to 1 and we get this expression. This involves some change of variables transformations which have will not discuss in any great details here.

(Refer Slide Time: 50:15)




Now what does what is that a accomplish that is accomplished; that means that we now know how to actually go from a density. Here given here to a density on the x 1 y 1 now we want to go given all the incoming densities on x y to the density on the outgoing x y. Let us do that next.

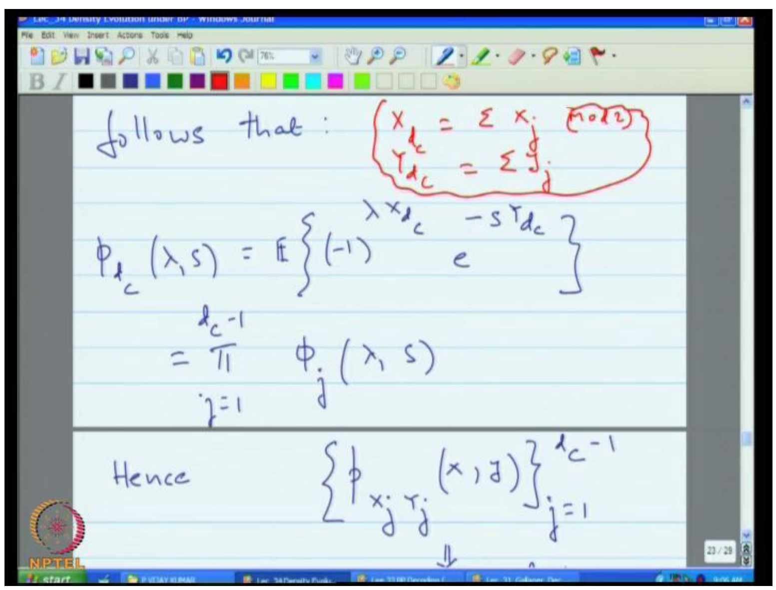
(Refer Slide Time: 50:33)



So, here let us define a joint characteristic function of x and y in this manner. It is minus 1 to the lambda x j e to the minus y j a lambda s. Lambda takes some values either 0 or 1 s you can think of s a complex variable as you would in the case of say a Laplace transform. So, this can be evaluated by actually evaluating it. Now, this expectation is over both x and y. So, x take on value 0 and 1 and y takes on a continue values. So, this expectation evaluates to this and so it is an average.

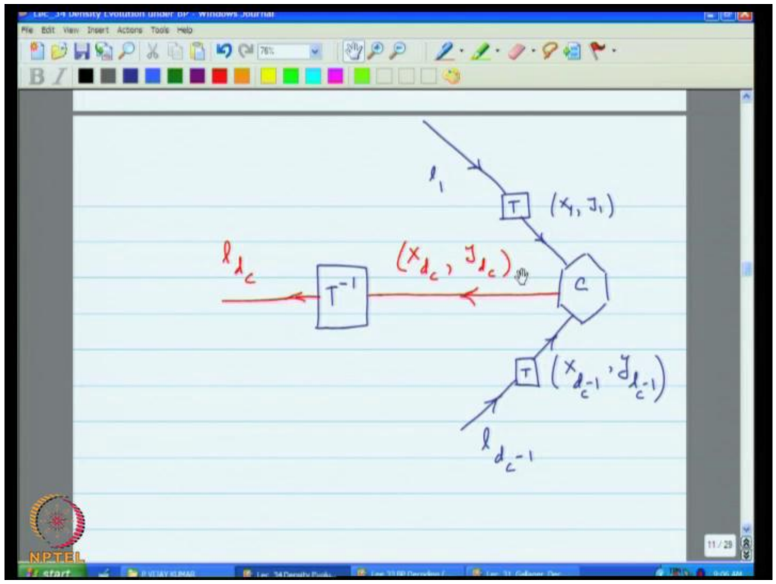
So, it is an average of all the values of x, but there is only two of them in average over all the values of all and that is a continue. So, that why we have a sum and an integral, and you can actually see that if you separate the parts corresponding to x equal to 0 and x equal to 1 that one turn is a Laplace transform of p 0 y. And the second is the Laplace transform of p 1 y, and then there is the minus 1 to the lambda which separates these two terms now since the l j, and hence the x j y j are statically independent.

(Refer Slide Time: 52:07)



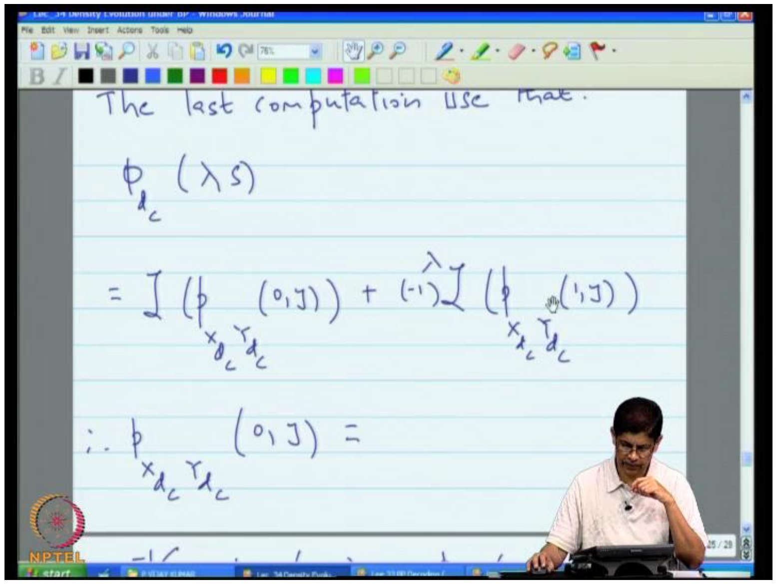

Now, I am going back over here to at the output we can do the same thing and because x d c. Now, we call that x d c is equal to the sum of x j (mod 2). So, this expectation evaluates to this. So, it is an average y d c is the sum of y j. So, keeping this in mind here it follows that you write this out and again by independence this breaks down into the product of the individual joint characteristic functions so that. So, what this is telling as is the way you can actually. So, let me go back to the perspective lecture this one.

(Refer Slide Time: 53:01)



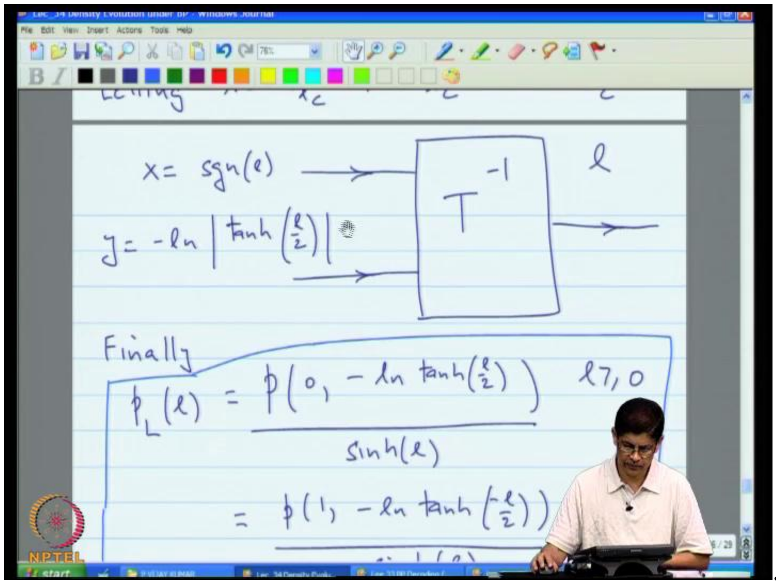

So, now we are trying to go from the density functions on the messages that are coming here to here, and what we are saying is you can go from these density functions to this as follows, you can go from the join density function, you can evaluate the characteristic function. And this set for all the inputs then by multiplying them in this way you can actually go to the joint characteristic function at the output from, which you can actually take the inverse, I mean you can go from joint density function to the joint characteristic function, but you can also go backwards.

(Refer Slide Time: 53:49)



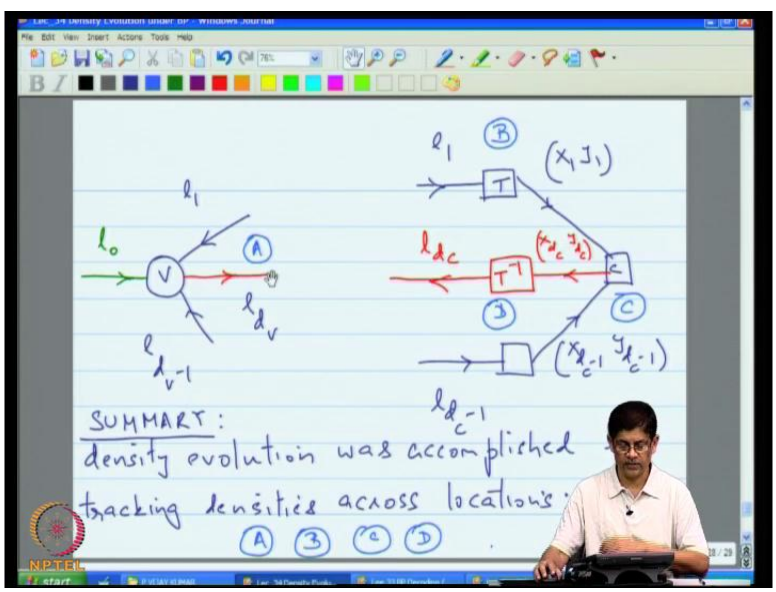

So, how do you actually go backwards if you given the joint character function then we are given this expression over here. Evaluate this for lambda equal to 0 and 1 and in terms of these transforms you can actually show that you can to cover p x d c y d c 0 y by taking the Laplace inverse of the sum divided by 2, and p of x d c y d c 1 y by taking the Laplace inverse of the difference. So, in this way you can actually, so that means that now we know how to go from a density function here, the density function here using joint characteristic function it remains to go across this transformation.

(Refer Slide Time: 54:51)




So, here is the Inverse transformation. So, we are given these x and y density function of x y and me took find the density function of l, again using this particular nature of this transformation you can actually show that the density function of the output is related to the joint density functions like this.

(Refer Slide Time: 55:12)



So, with that we were actually completed the entire density evolution, because we were able to relate at a variable node. The incoming density is to the outgoing density, check node we have to work a little bit harder we have to work in three steps; we could do this at once shot, but here we need it to work in three steps; step at b, step at c, and the step at d. We have to do density a transformation of density is across t, and then we have to do joint characteristic function to derive the density function here, and then do a transformation of density is correspond t inverse in here.

But with the net result that once you apply this you can actually keep track of the density functions at either check node or a variable node. And that basically is how you do density evolution. When you are doing belief propagation decoding?

So, I think our timing has been very good, I have got just under the minute left to summarize what we did today was we followed up on the earlier discussion on how belief propagation is carried out and we went to the harder task of actually saying well now how do we analyze it is performance?

So, we want to keep track of the likelihood there it is going to make errors, which one the one method that people know is to actually do this per keeping track of the density function of the messages that we are actually transmitting, and this is computationally a little bit mercy. I try to take you quickly through that the details are in the write up, but admittedly it was a little fast. So, I let you pore over this on your own and we will stop and continue next time, thank you.