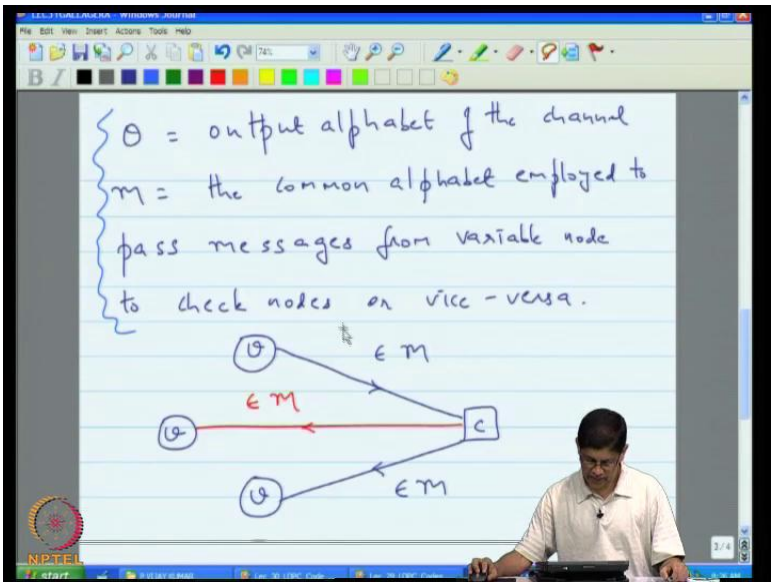
Error Correcting Codes
Prof. Dr. P. Vijay Kumar
Electrical Communication Engineering
Indian Institute of Science, Bangalore

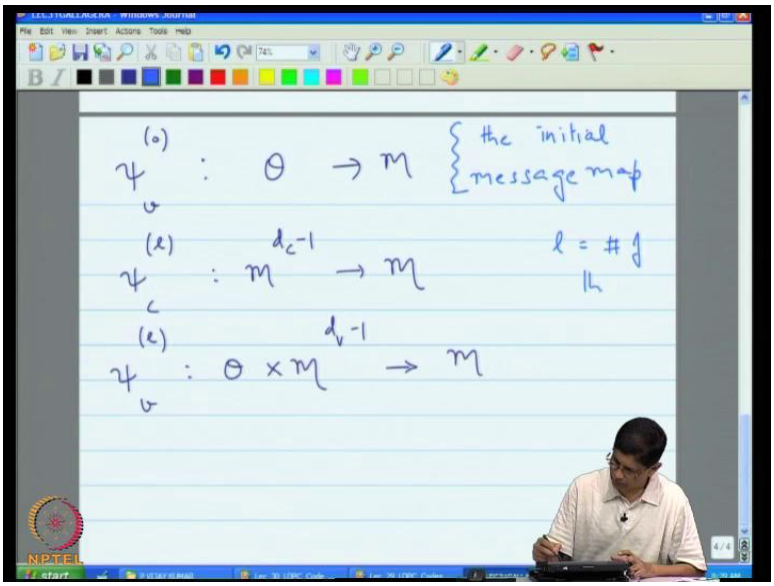
Lecture No. # 31
Gallager Decoding Algorithm "A"

(Refer Slide Time: 00:22)



Good afternoon, welcome back. So, this is lecture thirty one and I will give this the name, this will call this Gallager decoding algorithm "A". So, in the last lecture, what we did was we basically introduced terminology, relating to LDPC codes. So, we talked about edges, paths and neighbor hoods and then, we also talked channel models and I introduced the channel symmetry conditions. And towards the end we were talking about going to talk about channel output and input alphabet. So let me, copy down a couple of the figures from the end of the last lectures.

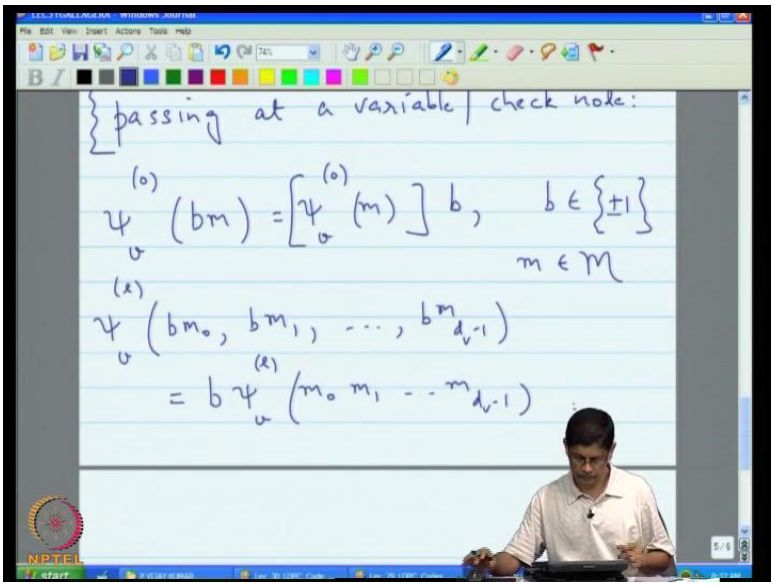
(Refer Slide Time: 04:00)



So, I think these two pages would be nice to have. So, we were talking about the alphabet from, which the messages there are actually passed are drawn from and I pointed out that these messages are actually subsets of the real numbers and also pointed out. There is a common message alphabet that is actually used to communicate messages, forth create between the check node and the variable node. This green arrow here represents the output of the channel. So, that potentially could be a different alphabet so, going to denote that. Now message passing can also be viewed as a mapping from the incoming messages or from the outgoing message.

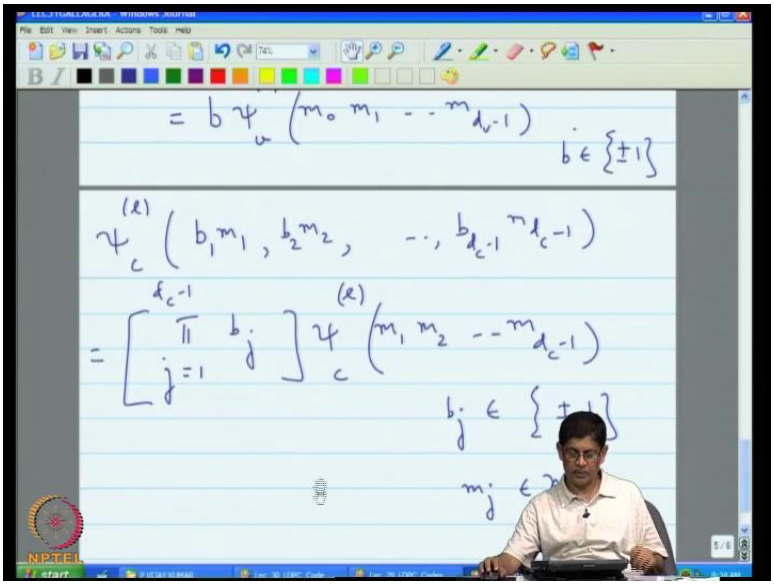
So, in that sense we actually introduce the following notation, we will actually say that psi sub v of zero, is a map from o to m and you can think of this initial message map. So, this is the variable node to the check node and what we are basically saying is look? There is an input that comes from the channel and whatever the node does it transmits. It is output based only on the input of the channel, because that is all. It is initially and this zero here denotes the number of the iterations and we are talking about the zero iteration here. Now, on the other hand at a check node so, that is c here and during the l th iteration the incoming messages. Since, there are d minus c inputs. It is going to map that end and in subsequent iterations from the variable mode. The map is output of the channel alphabet and taking to account the incoming messages, which are d v minus one in number, you map this to the message alphabet. So, these are the three maps that are carried out and l is the number of the iteration.

(Refer Slide Time: 06:29)



We got these messages, we will make certain assumptions so, assumptions concerning message passing at a variable or check node. So, we will write these down in terms of notations that, we introduced here earlier. So, we will say that psi v of 0 of b m is equal to psi v 0 of m time's b, where b belongs to plus and minus one and m belongs to the message. We also assume that psi v m of b m 0, b m 1, b m d v minus 1 is equal to b time's psi v l of m 0, m 1, m d v minus 1.

(Refer Slide Time: 09:00)

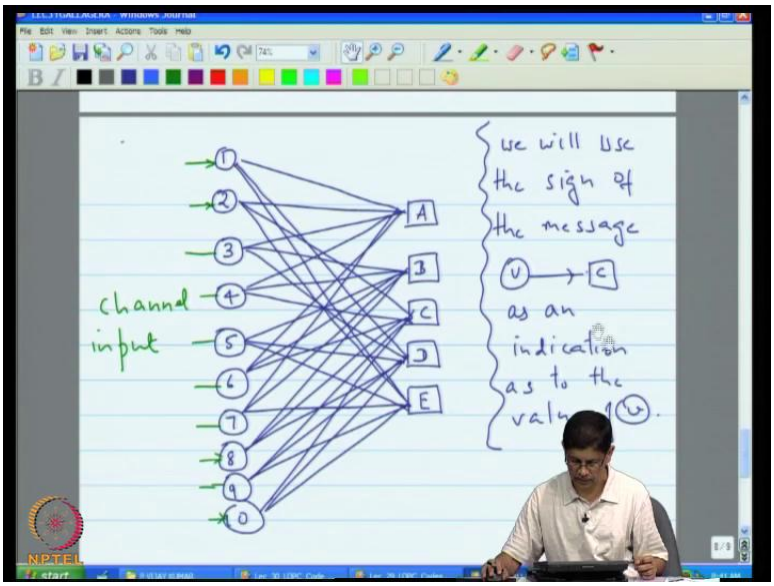


I will come back and explain in a second. At a check node we assume that the property that satisfies that again here, b is plus or minus 1. At a check node, we assume that b 1 m 1, b 2 m 2, b d c minus 1, m d c minus 1 is equal to the product j is equal to 1, d c minus 1of the b of j times psi c of l of m 1, m 2, m d c minus 1. And again this b of j is either plus or minus 1 and of course the m j is drawn from the words. So, in words what we are saying is look that in variable node? Initially, that whatever mapping is conducted at the variable node must have the property that. If the sign of the incoming signal is changed then that outgoing signal must also change. So, if you multiply m by minus 1 then the new message is whatever the message was before, when m was the input times b.

In general, this is for the initial iteration. So, the initial iteration is there is an incoming message from the incoming channel; there are no other incoming messages. Subsequently you have messages coming from all of these nodes. So, what is saying is that if each of these same multiplied by minus 1 then the output message will also be multiplied by minus 1. So, that is what saying that, if each of the incoming messages including the messages that comes from the output of the channel. If they all multiplied say by minus 1, then the message that the way node is required is going to send out will also change that sign and a check node it is a little bit more complicated. So, we say that reject node the requirement is that let say that the individually the incoming messages and there are these c minus 1.

So, going back here, you see that the check node here you have all these incoming messages. Since, the check node has degree d c in a d v, d c regular code and we are going to assume that this class of codes. There are d c minus 1 incoming edges and one outgoing edge, if each of these multiplied individually by a separate sign factor. So, b 1 after d, c minus 1 then what happens is that these signs factor outside and you get the products sign, which actually goes back? Which is multiplies the normal output of the channel.

(Refer Slide Time: 13:00)



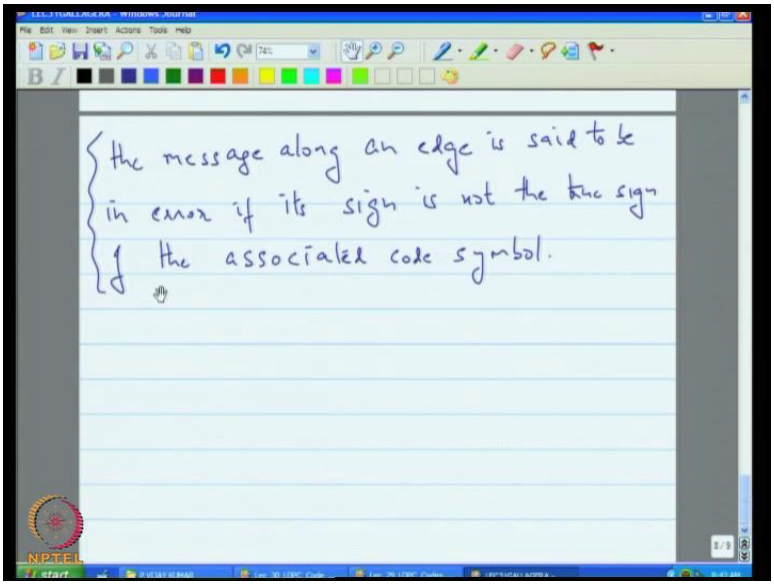
So, these conditions let us give the numbers equations so, this is 1, this is 2 and this is 3. So, equations 1 to 3 are known as the channel are known as the variable and check node symmetry conditions. Now, what I would like to actually point out next is that when you are doing message passing decoding using the tanner graph. So, perhaps for this and pull up again the picture from our last lecture. Here is the tanner graph so, what going to happen is that we are passing messages back and forth on this tanner graph and let me just add one other small change. Let us also depict the input that is coming in from the channel to each of these lines. So, this is the channel input.

Now, what we want to actually it turns out that, you are going to iteratively pass messages and then you stop, when there is a certain number of there are some conditions under, which you can stop. For example, you can fix the number of iterations after, which you stop, but then once you stop always you have is the bunch of messages. How do you make decisions on the code symbols based on these messages? So, it turns out that what you do is since here regardless of whether you are operating on a banner symmetric channel or awng channel. These messages are after all plus minus 1. So, what you do is, you stop look at the sign of the message that you are sending out from the variable node.

If that sign is positive, then you declare that corresponding variable signal to be positive. So, you have multiplied this here, you want might want to take a majority of these. The signs of the message and there are many edges and so on. That is you have actually make a decision may examine, may be in the sign. Now there are many edges and potentially each edge could be carrying a message, whose sign is either an agreement with the correct sign of the corresponding code symbol or not. So, we will actually regard an edge as being as an error, if it is sign disagrees with the sign of the corresponding code symbol. So, perhaps it is worth just taking a quick note over here.

We will use the sign of the message going from say the variable node to check node, as an indication as to the value of v. So, these value of this the particular code corresponding code symbol is what we are trying to decode? So, we will indicate this as an estimate of as a current estimate of the value of v of course this estimate will keep changing and so, we will regard this message as being an error, if the sign does not agree with true value.

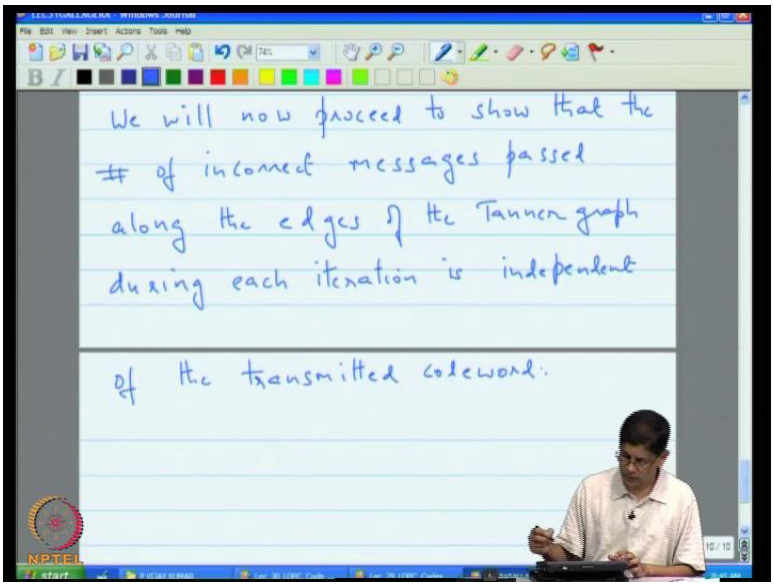
(Refer Slide Time: 18:54)



So, the message along an edge is said to be the error, if sign is not the true sign of the associated code symbol. Now the messages are being passed in both directions. So, it is meaningful to talk about in message in either direction being an error, because for example, it is going from here to here. We look at the sign of this message and compare it with the true value of this, if you are

looking in the message going from here to here. We compare the sign of this message with the true value of this particular code symbol. So, in both cases it is quite meaningful to talk about the message being an error. Again the sign of the message going from 8 to 6 is an indication of the value of 6 so, if that disagrees with the true value true sign of 6. Then we say that message is an error. Now what we want to actually show is that the number of incorrect messages passed is independent of the actual codeword transmitted.

(Refer Slide Time: 20:15)



We will now show, proceed to show that the number of incorrect messages passed along the edges of the tanner graph, during the each iteration is independent of the transmitted code word. So, once we showed that to simplify the analysis, we just assume that the transmitted codeword. Because we now show that the messages that are passed are independent of the actual code word and what I will do is I first. Now let us, just go to the graph and try to explain the reasoning behind that. The reasoning behind this is this now if the transmitted code word was the all one code word then all these messages will initially take on the value at the output of the channel, this is certain value of t.

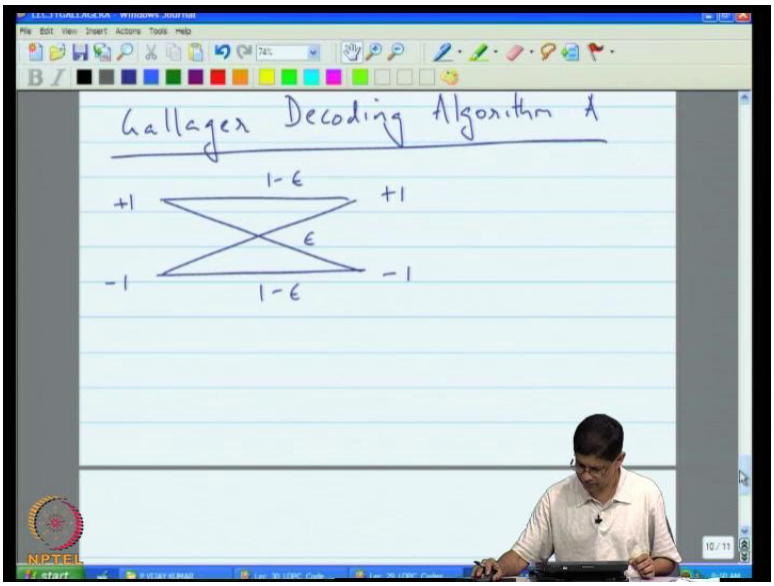
Now, if some of these code symbols were actually minus 1, then we know that there is an event. There is I mean the probability of y t being equal to I mean, probability of the output taken on

the certain value, corresponding to 1 is equal to the probability. That it takes on the value minus 1 corresponding to x t equal to minus 1. So what that means, let we see…

Absolutely let us go back actually to our symmetry assumptions. So, here if the incoming symbols changes sign as it well, because of the codeword will changes the sign then your incoming message will also changes sign. In which case, what will happen is that the messages that are passed initially will actually change sign. So, what going to happen is that, I think keeping in mind that I would like to actually get into Gallager's decoding algorithm. Let me, keep this brief basically what happens is that your messages sign of the messages. Whatever they are transmitted code word is a 1, the number of agreements the initially will not be the same. Regardless of the actual code word is and that will propagated through the system, because let us say that the initially when the messages go. They are boring, they are in signs corresponds to the signs of the incoming messages and thereafter to check with what happens is that, these signs get multiplied, because of the nature of a parity check.

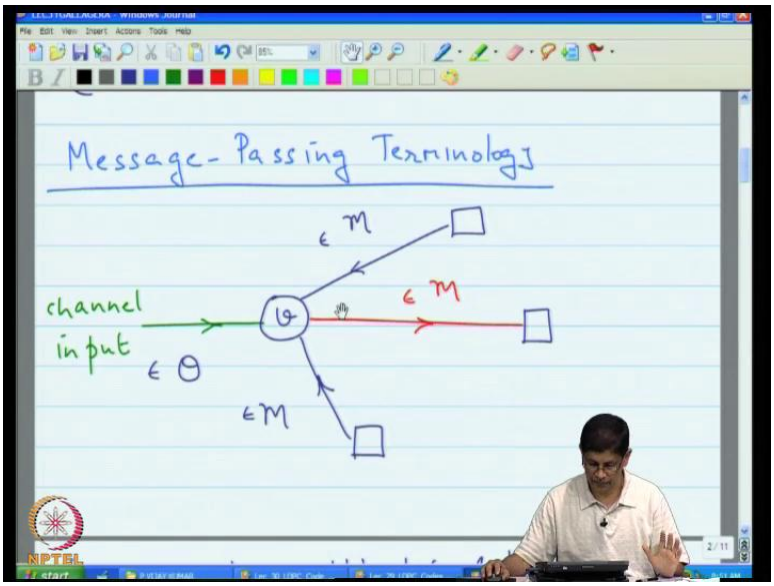
It turns out the sign of the message pass back from the check node to the variable node, is also multiplied by the corresponding value of the variable node and this keeps continuing with the net effect. That the sign of any message passed is does not depend upon what the actual code word was it is the same. So, let us one thing let us just put that down and if time permits, I will come back and explain that little bit better.

(Refer Slide Time: 26:00)



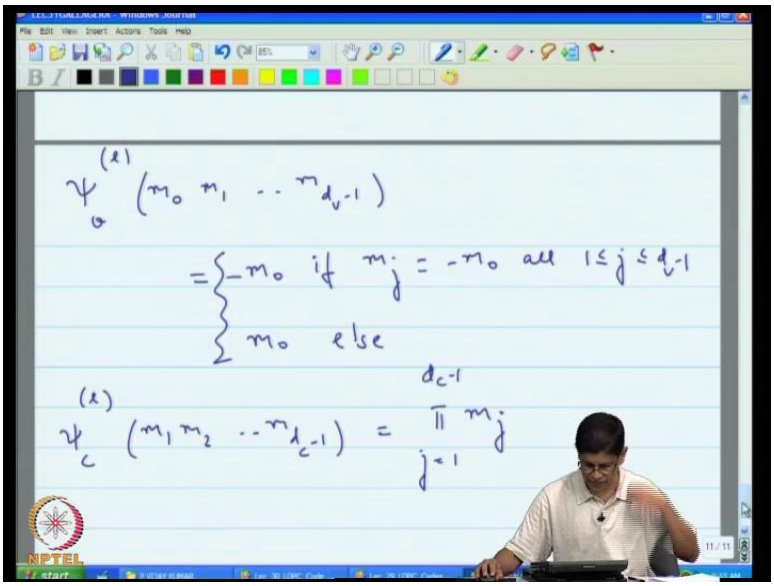
Now we come to examining a particular decoding algorithm that was introduced by Gallager, Gallager's decoding algorithm what we actually considering was the banner symmetric channel, that the input was either plus 1 and minus 1 and he specified certain specific variable and check node maps. He specified that at a variable node, the rule is that initially whatever the incoming messages that message is actually passed out and subsequently the message that passed out of the variable node is precisely. Whatever comes in from the input of the channel? Unless all the other messages over ridded.

(Refer Slide Time: 27:44)



Let use the other figures to explain that so, initially here is your variable node and here is the output of the channel. Initially Gallager's algorithm says look my channel output is plus minus 1 and j is going to request the variable node to send out. Whatever is receives from the channel a line along all the outgoing edges, thereafter again us still talking about a variable node. What it actually does is that it is going to send out, the messages are all plus minus 1 in Gallager's decoding algorithm "A". So, it is going to send out the channel input, unless is there is overriding evidence provided by all of the other inputs, if all of the other inputs have a sign that disagrees with the sign of the channel input. Then it actually sends the reverse of the sign of the channel input, otherwise it continues to send whatever the channel are input pass. At a check node in Gal lager's algorithm messages again are all plus minus 1, it just simply sends the product of the incoming messages.

(Refer Slide Time: 29:43)

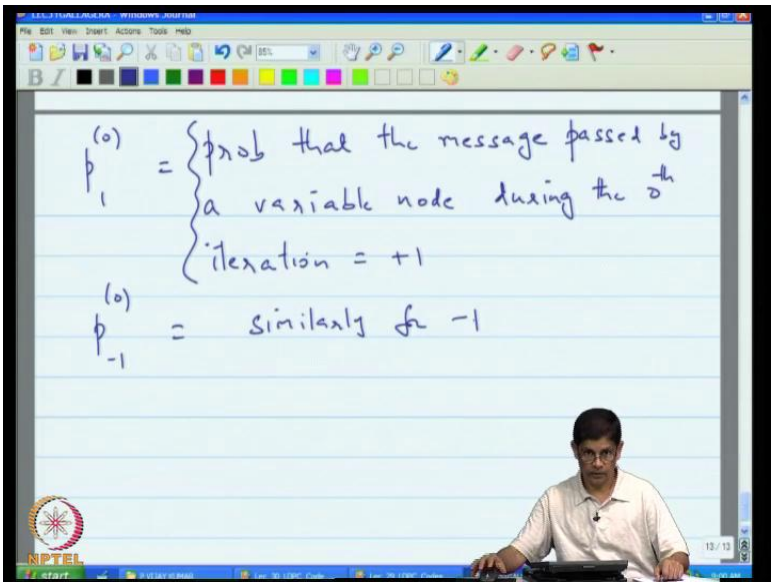


So, psi of 0 of m is simply equal to m so, here I should mention o is equal to plus 1 minus 1 and your message alphabet is also equal to plus 1 minus 1. So, the initial map at a variable node is just this, thereafter psi of v of l of m 0, m 1, m d v minus 1 is equal to minus m 0, if m j is equal to minus m 0 for all and it is equal to m 0 else. So, what this actually means is that it is going to pay very close attention to what it receives from the channel and it is going to transmit that symbol. Unless all of the other incoming messages have a sign that is different from the incoming message from the channel, in which case listen to these other messages into change the sign of outgoing message at a check node, during any iteration.

The message that is passed out is simply the product of all the incoming messages. So, with this the decoding algorithm is very clear, because what we have done is we have identified the action during decoding. That is actually taken by the variable node and we have also identified the action taken by check node, during decoding. So now, we would like to actually do performance analysis. So, the question is how well this algorithm performs? So, we will handle this by doing something that is nowadays referring to as density evolution. So, we will do this, we will evaluate performance by carrying out density evolution, by which we mean that we will determine estimate the number of incorrect messages passed, during each iteration and will actually do this iteratively.

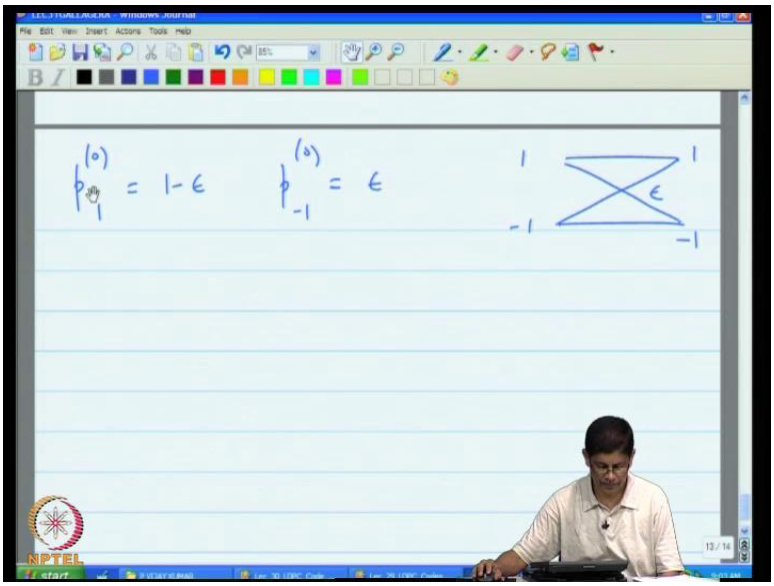
So, to do that it is (( )) we assume that the all one code word was transmitted.

(Refer Slide Time: 35:35)



And will introduce some symbols now. Let us, say that p 1 of zero denotes that the probability that the message passed by a variable node. During the 0 eth iteration is equal to a plus 1 and p minus 1, 0 is the same. Similarly, for minus 1 so, here we are going to use. Now earlier in our, we had used b and c to denote the variable in check nodes, but here what we are going to do is we are going to distinguish between variable and check nodes with regard to the probability distribution. By actually using letters p to denote probabilities associated with variable node messages and q for check node messages. Similarly, q l 1 is the probability that on the l eth iteration, the check node message is equal to 1 and similarly for minus 1. So, you would have q minus 1 of l.

(Refer Slide Time: 38:26)



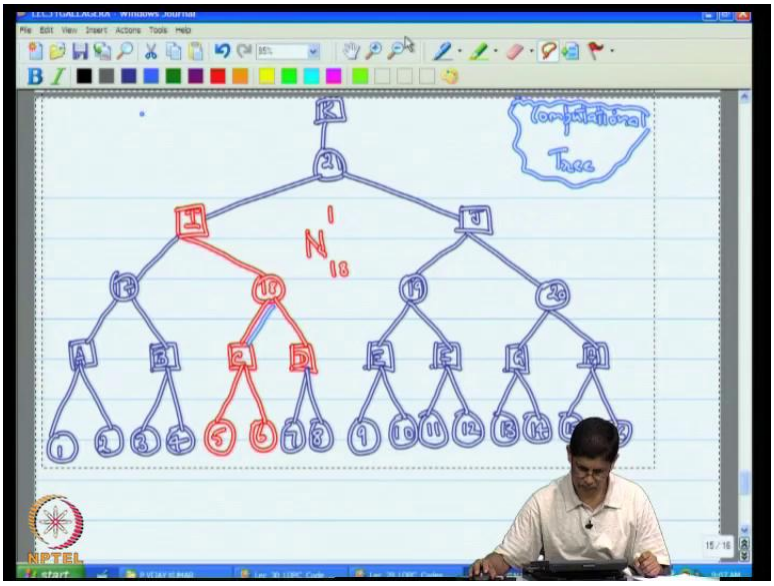
Now once we have that it is possible to iteratively determine these expressions. Now keeping in mind there are channel is a bandwidth symmetric channel, in that the cross over probability is epsilon. We know that p 0 of 1 is equal to 1 minus epsilon and that p minus 1 of 0 is equal to epsilon, because we assume that the all 0 code word was transmitted. There is another thing that I need to point out which is that when we are doing this analysis of the probability of messages being an error. See, if you assume that the transmitted code word was the all 1 code word then, if we look at your incoming messages. The probability that their 1 is equal to 1 minus 1, the probability that their minus 1 is epsilon 1 minus epsilon the other is epsilon.

And it is same for all inputs, because you are assuming that the all 1 code was transmitted so, the density function of the probability mass function of the initial message is the same for all inputs. You can see, thereafter by symmetry, because every node has left has the same degree every node on the right has the same degree. The symmetry what is going to happen is that the probability mass function associated with the message. Iteration does not depend upon a particular edge it only depends upon the iteration and also the direction, in which the messages are going or the messages are going to variables to check node or vice versa.

So, we do not have anything in this notation to denote a particular variable, because for all variable nodes it is the same. So, maybe it is worth making note of that, note that by symmetry

the probabilities the probability mass function of messages along an edge only depends upon the number of the iteration. We are using this symbol l to denote that so, example i, e on l as well as the direction of the message. That is this message going from variable node to check node or vice versa. So, we have these initial probabilities, thereafter it is not hard to compute the other probabilities in succession. For example, if you look let us say at a variable node and we are interested in the p n f the probability, let us say that the transmitted message along here is a 1. Now this is a 1, if this is a 1 and not all of these are a minus 1 or if this is a minus 1 all of those are a minus 1. Now, here is where another interesting point comes up and for that I will have to take you back to the computational tree. So let me, select this page bring it into our current lecture.

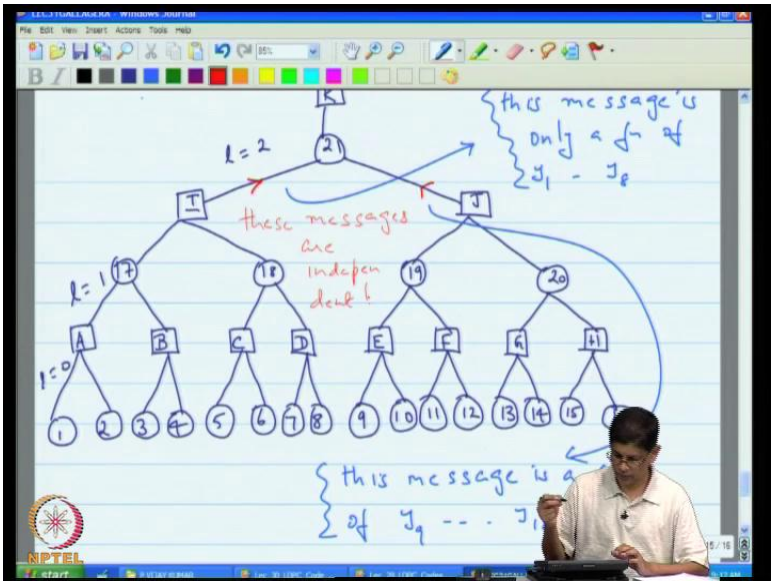
(Refer Slide Time: 44:00)



So, here is a computational tree and let us get rid of these colors. We do not need them anymore; you can just have the whole can have the whole figure in one color. Now, we are doing this density relation under the assumption that the transmitted code word is a 1, in particular we know what the transmitted code word is… Now, let me just put down here so, consider let us say a message that is passed during a certain iteration. Let us say the second iteration a message that is actually passed from check node to a variable node. Now, the message that is going to be send is a function of the incoming message. Now, this incoming message in turn is a function of these incoming messages and if you trace it back eventually, whatever the incoming messages are

dependant only on the inputs received across the channel. For example, here the message transmitted here, during the certain iteration this message is a function of… Let us say you are talking about… So, the initially let us say the message that was passed here and on the second iteration, there was a message that was passed from here to here.

(Refer Slide Time: 46:00)

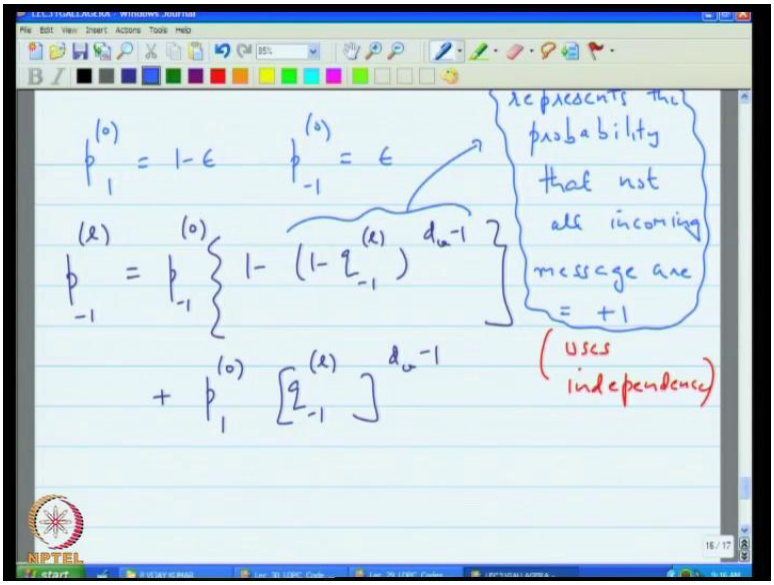


So, I can actually denote that by saying let us say that this on l equal to 2 and this was the case, when l equal to 1, and this was the message passed from l equal to 0. Remember that particular iteration consists of a message from a check node to variable node and then back. So, this was one iteration; this was the initial phase. So, during the second iteration the message that is passed here is a function of what was incoming here, but what was coming here from incoming here is a function of what was incoming here and so on. And if you trace it back this message is a function only of the received symbols y 1 through y 8 message. So, this message makes a note of that. This message is only a function of y 1 through y 8.

Now in similar way, if you ask yourself a question of what is this message a function of… Then you see that this message is a function of y 9 through y 16. Now, given the transmitted symbol the received symbols are only a function of the channel loss and therefore they are independent; and therefore these messages, which are a function of these independent variables, are also independent. So, when we are actually computing probabilities, and we want to find out the

probabilities of an outgoing message given probabilities of the incoming message, we should we can we are free to make use of this independence. So, that is so in conclusion here, we just write these messages are independent.

(Refer Slide Time: 49:52)



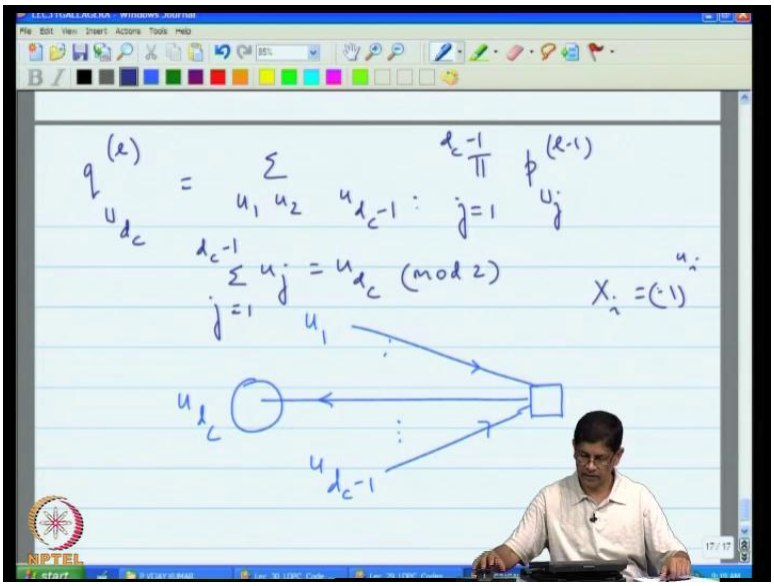
So, making use of this, it follows that so, we already started out and let me copy from what we had earlier which was this probability? Thereafter, we can actually say that p minus 1 of l is p minus 1of 0 times 1 minus, 1 minus q minus 1of l to the d v minus 1 plus p 1of 0 times q minus 1 of l to the d v minus 1. So, what does this expression tell us it is actually straight forward, it is just saying that look, if you are looking for the probability that a variable node is going to pass a minus 1 of the l th iteration. And that can happen only if the input channel input is a 1and not all the inputs that see in the check node are all 1. So, this is probability of them being minus 1, this is the probability that the 1and since these messages are independent.

I can raise it to the power and I want to join the probability of them being all equal to 1added their all not equal to 1. So, just wanted to make a note of that, this quantity here, really represents the probability that not all incoming messages are equal to plus 1. A side remark is that this uses the independence, which are remarked upon a few minutes ago? Now, the q s represents the probabilities of messages going from the check node to the variable node. So, we have the p minus 1in terms of the q s. So, how about the q s themselves can also be computed in this way.

We can actually say let us go back here and look at a picture of a check node. So, the probability that we are trying to evaluate now is saying look.

Here is a check node and I am going to pass a message from the check node to variable node and I want to know, what is the probability? That it is minus 1 or plus 1 keep in mind. That iteration begins by a message from a check node to a variable node. If, you are talking about the l th iteration here, the incoming messages are messages. Those are transmitted during the l minus 1 th iteration. So, that is just forgetting our notations. Now how do you compute that? Now that is a little bit trickier the way we computed? This is the way that Gallager computed.

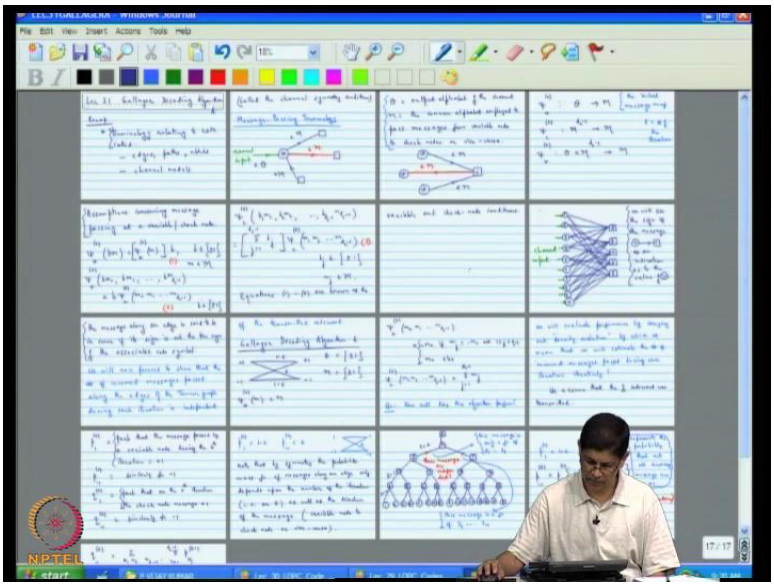
(Refer Slide Time: 54:09)



It is to say that q u d c of l is the sum u 1 the sum over u 1, u 2, u d c minus 1, subject to sigma u j equal to u d c and the product j equal to 1 d c minus 1 p u j of l minus 1. So, what this is actually saying is look, after all the probability that u d c is a 1 perhaps the picture. So, we just have 2 minutes left so, let me just draw a picture here. So, the pictures one in which we have a check node here and you are asking the question, what the probability? That is outgoing message is a 1 and here are the incoming messages. So, this is going to node u d c, this is coming from u 1. This is coming from u d c minus 1 so, there are some other incoming nodes as well.

So, what this computation here is saying is that look? The probability that this message is going to be a 1 is if all combinations of the incoming messages, whose which has such I will written sum over here. Which needs some explanation? So, let me just do this, let me add mod 2, I remind you that x i is minus 1 to the u i. We will continue this particular calculation in the next lecture. Let us just end of the quick summary.

(Refer Slide Time: 56:40)



So, we look at code words to Gallager decoding algorithm and by first introduced the message passing terminology, in terms of describing the input and output, alphabets as well as the maps. Then we specified the decoding algorithm and we were doing, what is called density evolution on this graph. We will continue next time.