

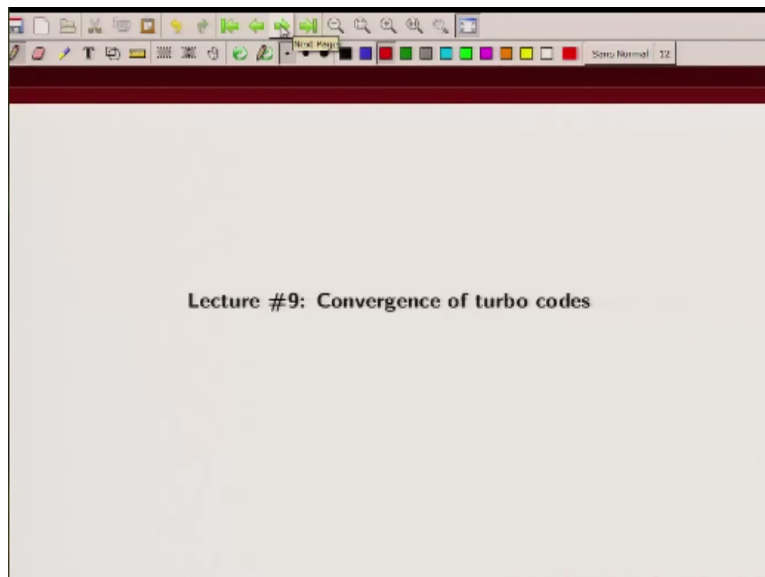
Indian Institute of Technology Kanpur
National Programme on Technology Enhanced Learning (NPTEL)
Course Title
Error Control Coding: An Introduction to Convolutional Codes

Lecture-9
Convergence of turbo codes

by
Prof. Adrish Banerjee
Dept. Electrical Engineering, IIT Kanpur

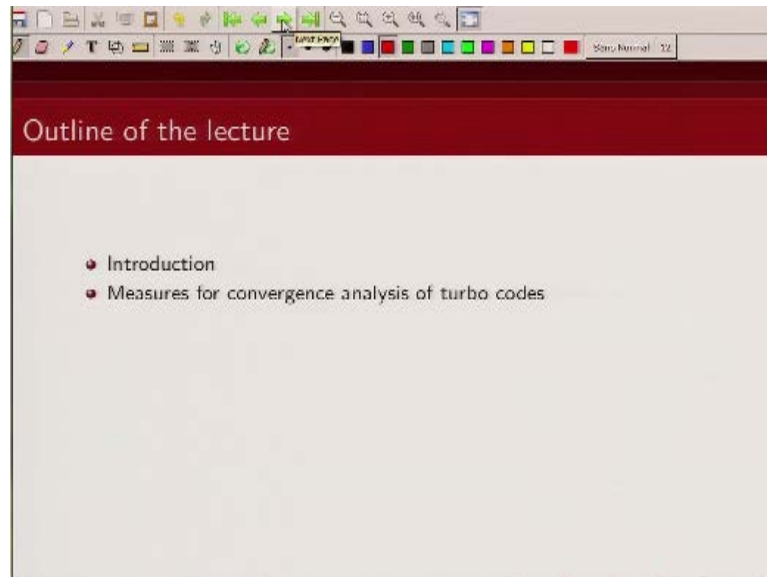
Welcome to the course on error control coding, an introduction to convolutional code. Today we are going to talk about how to analyze the performance of turbo code in low snr.

(Refer Slide Time: 00:30)



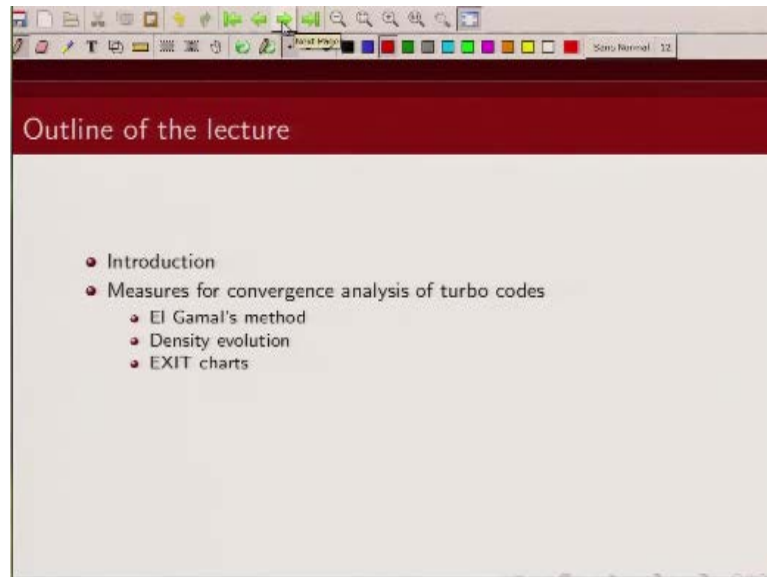
So we are going to talk about convergence, how to track the convergence of turbo iterative decoding algorithm and that is the topic of our discussion, convergence of turbo codes.

(Refer Slide Time: 00:49)



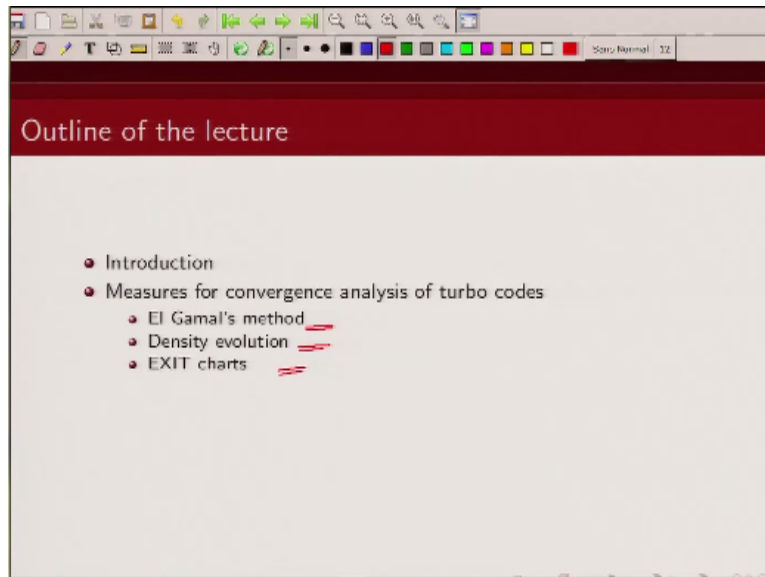
So with brief introduction we will talk about what are the various measures for convergence analysis.

(Refer Slide Time: 00:52)



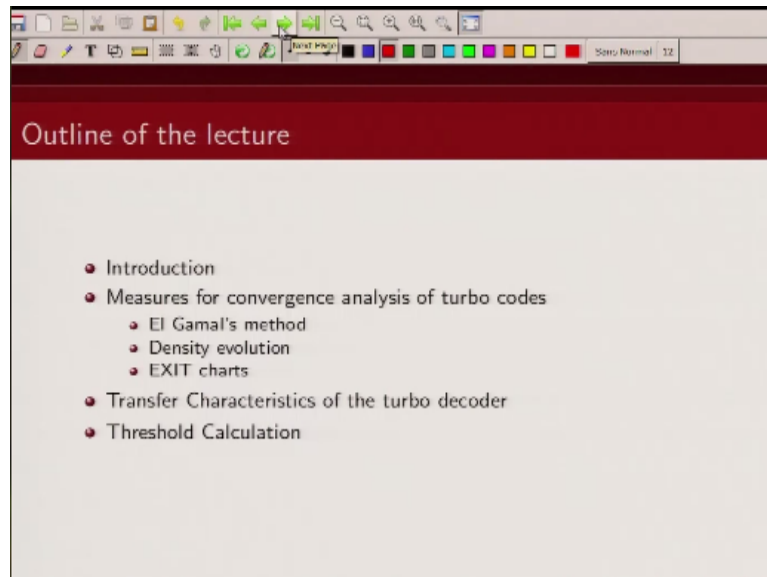
And in particular we will -- are going to talk about these three methods.

(Refer Slide Time: 00:57)



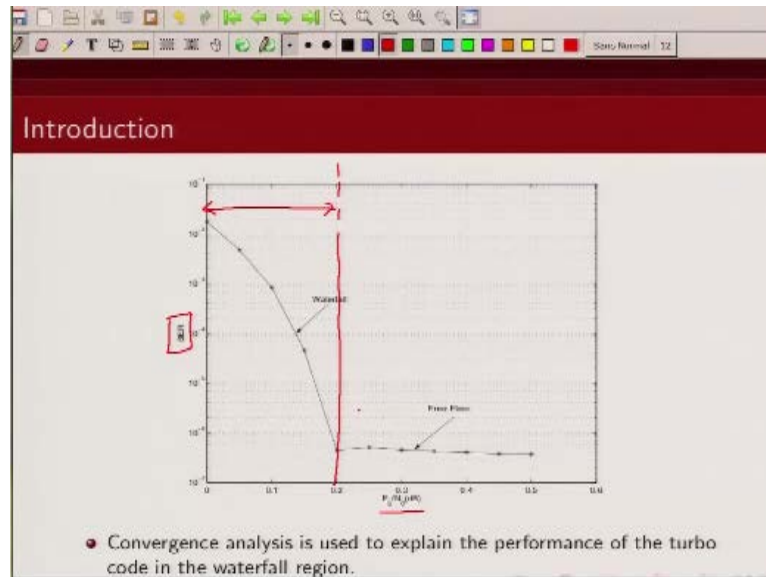
The first method which is based on Gaussian approximation and which involves tracking domain of the extrinsic values. A method proposed by El Gamal, next we will talk about a method which is proposed by [indiscernible][00:01:15] and others using density evolution and then a method which is based on mutual information, tracking mutual information proposed by 10 brink.

(Refer Slide Time: 01:28)



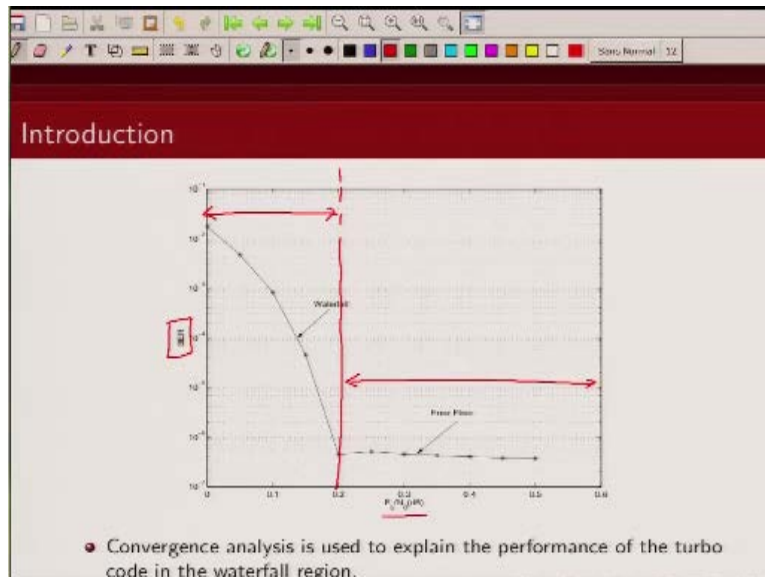
And then we will talk about, what do we mean by a transfer characteristic of a turbo decoder and how we can use it to compute the convergence threshold of a turbo code.

(Refer Slide Time: 01:40)



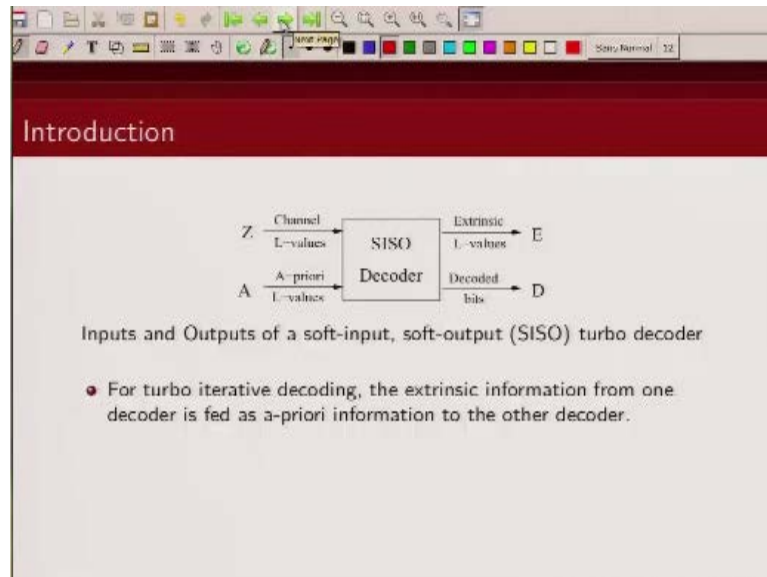
So this is a typical performance of a turbo code if we take a large block size, this is for block size, I think 65,000 plus. So if you take a large block size this is typical performance of a turbo code on the X axis I have signal to noise ratio and on the Y axis as plotted bit error rate. Now you will see there is a region, so this region which we are calling waterfall region where there is a steep fall in better rate performance.

(Refer Slide Time: 02:18)



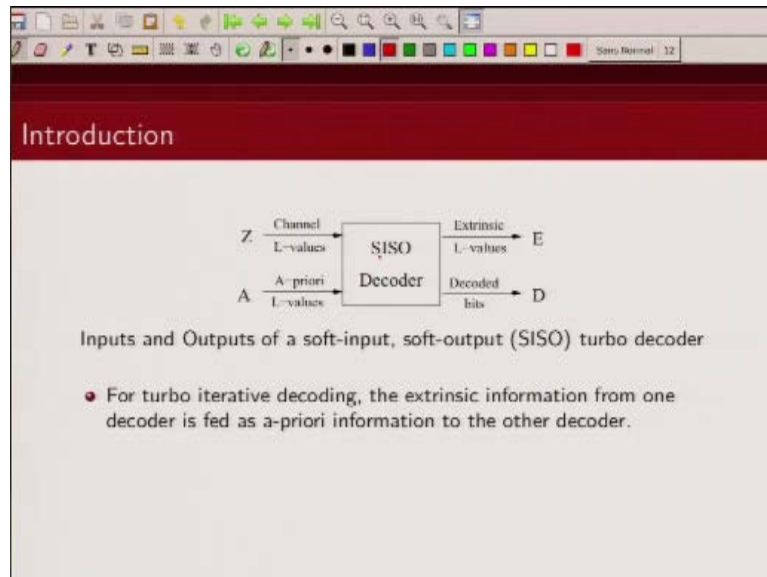
And then there is a region we call it error floor region where the BER does not improve much. So today's topic of discussion is this waterfall region, what determines the performance of turbo code in this region where it falls sharply. And how can we get some guidelines on how to choose constituent encoders so that we get a steep fall like this.

(Refer Slide Time: 02:47)



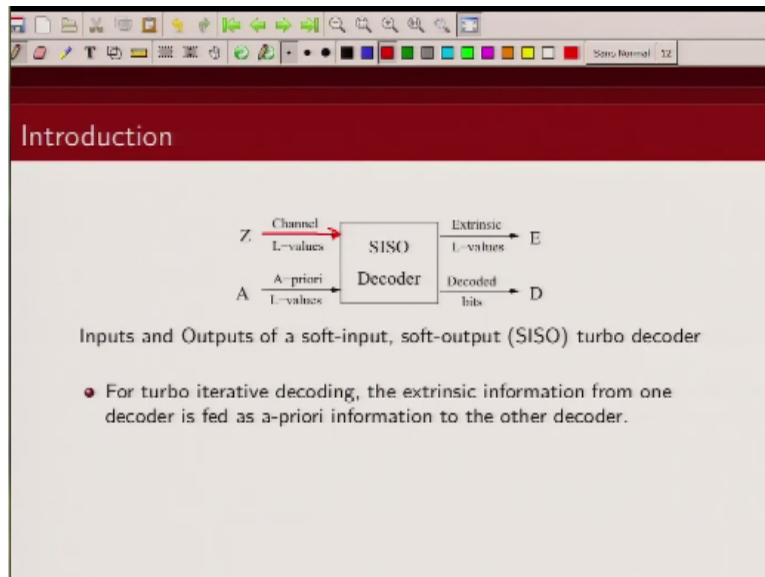
So before we study the convergence analysis, convergence of turbo code let us pay a close attention to the basic block diagram of our turbo decoder.

(Refer Slide Time: 03:01)



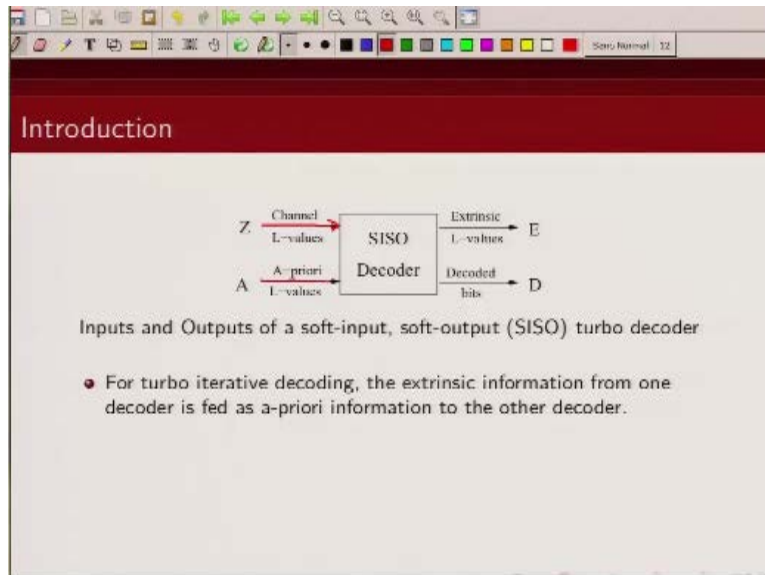
The heart of a turbo decoder is a soft-input, soft-output decoder and if you recall this soft-input, soft-output decoder takes in as input

(Refer Slide Time: 03:14)



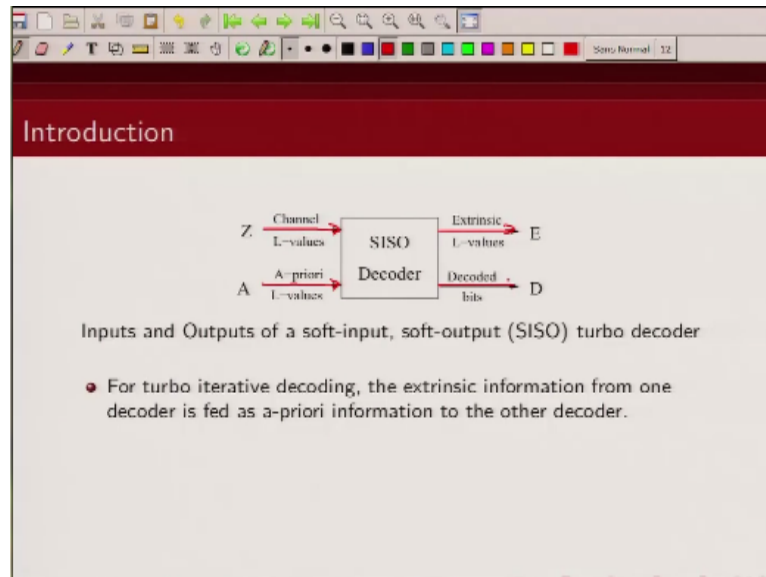
The channel received values corresponding to the information and parity bits.

(Refer Slide Time: 03:20)



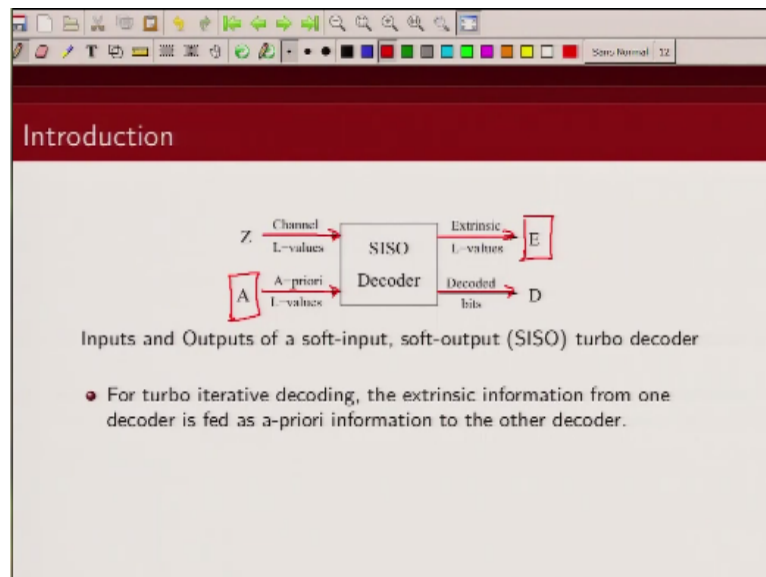
A-priori value which it receives from the other decoder which are the extrinsic values passed on to the other decoder.

(Refer Slide Time: 03:31)



And it computes extrinsic values as well as APPL values where you take a hard decision to get back your decoded bits. So if you look at a turbo decoder this is the heart of a turbo decoder, there are two such soft-input, soft-output decoder. And if you look for a particular signal to noise ratio, if you look at turbo decoder at the function of iteration you will notice.

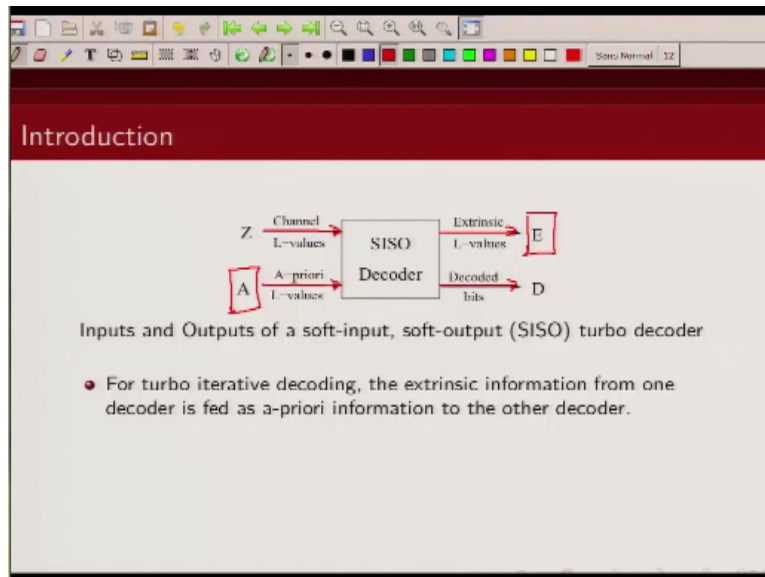
(Refer Slide Time: 04:02)



The only thing changing with iteration is this extrinsic value and a-priori value. So with iteration your – initially you do not have any estimate on a-priori value, you assume that it is the bits are equally likely to be 0 and 1. But subsequently with iteration when your extrinsic values are generated those are passed on as a-priori value.

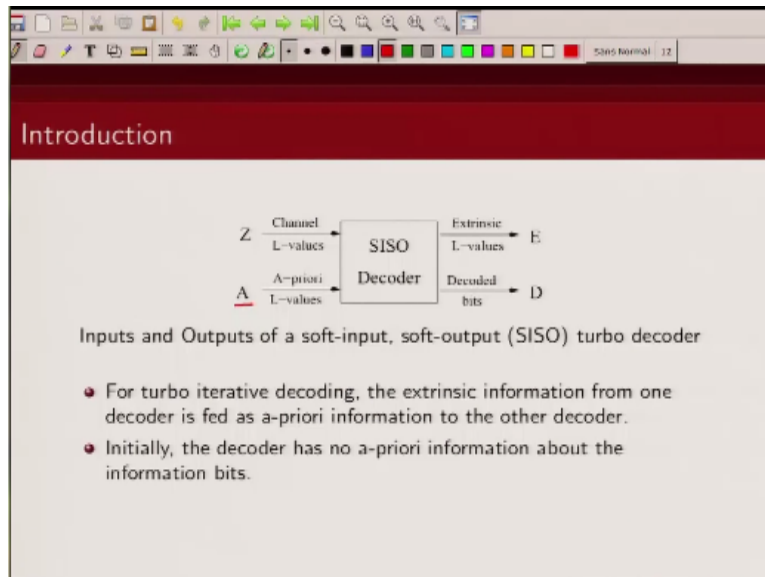
Now the channel L values remains same for a fixed signal to noise ratio for a received bit, the channel L value remains the same. Only thing changing with iteration are these two quantities.

(Refer Slide Time: 04:42)



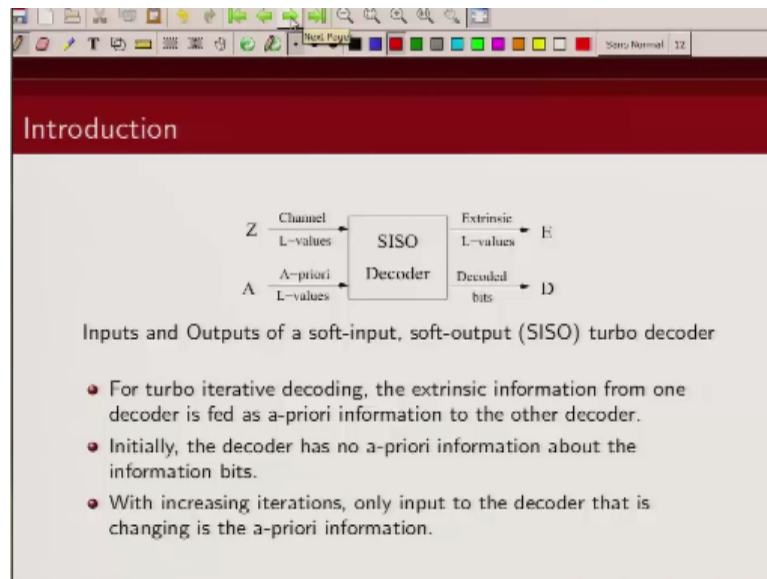
This a-priori value and the extrinsic value. So if we can track with iteration how our extrinsic information is growing with this a-priori information that will give us some clue about the performance of turbo code at waterfall region.

(Refer Slide Time: 05:03)



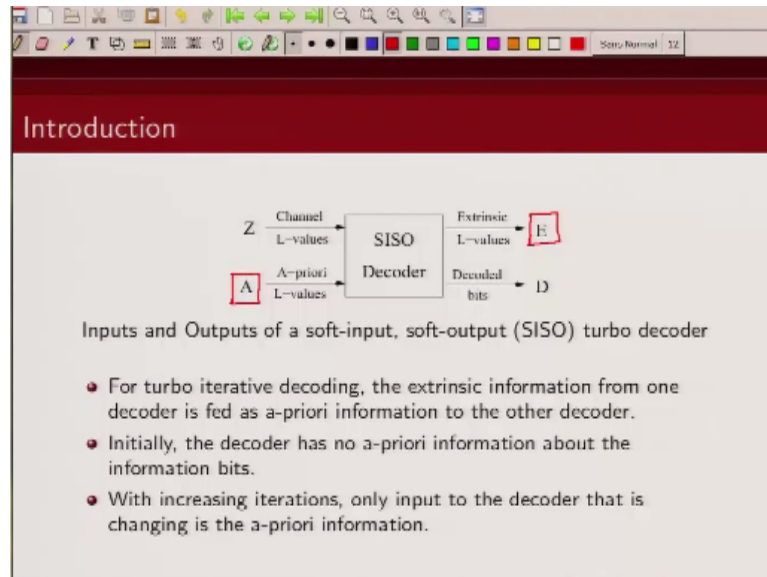
So as I said initially we do not have any a-priori value, but subsequently after one half iteration extrinsic informations are generated and that is passed on as a-priori value.

(Refer Slide Time: 05:17)



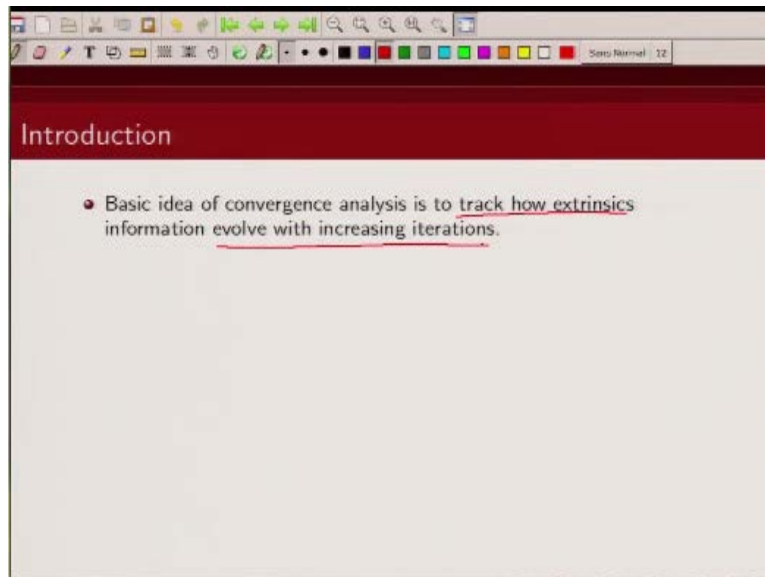
To this soft-in, soft-output decoder. And again I emphasize the only thing changing with iteration are these extrinsic values and a-priori value. So if you want to track how your turbo decoder is working with iteration you need to track these two quantities.

(Refer Slide Time: 05:41)



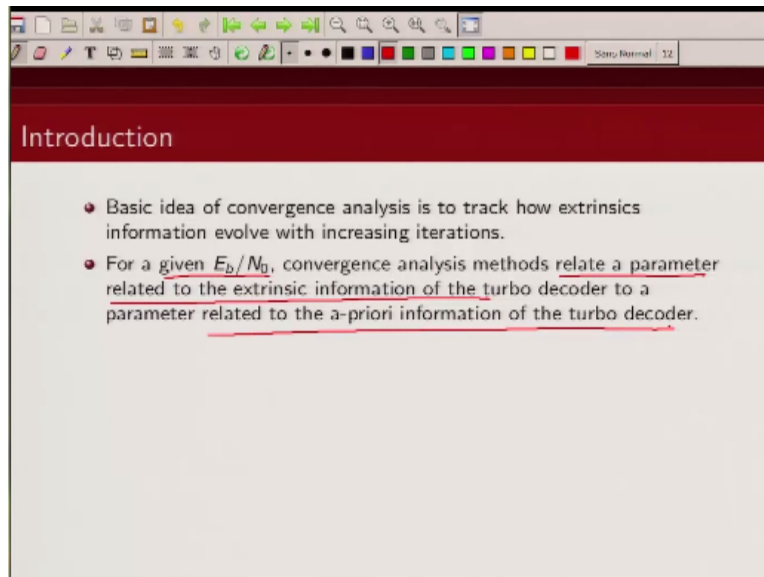
And we are going to talk about what are the various measures that we can use to track these two quantities.

(Refer Slide Time: 05:50)



So basic idea of convergence of turbo code or convergence analysis of turbo code is to track how these extrinsic information are evolving with increased iteration. So if you feed in better a-priori value, how is your extrinsic information evolving?

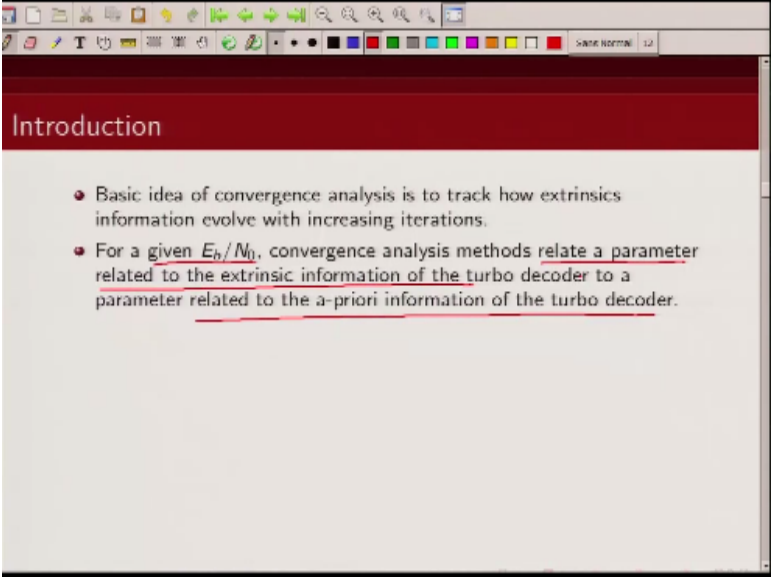
(Refer Slide Time: 06:14)



So what we do is for a fixed signal to noise ratio we have a set of received value. So what we do is we try to relate a parameter which is related to the extrinsic information of the turbo decoder and we try to relate it to a parameter which is related to the a-priori information. As I said in this soft-input, soft-output decoder only thing changing is this a-priori information and this extrinsic information.

So we want to track how these extrinsic information and a-priori information are growing with iteration. So what we are going to do in this convergence analysis is we are going to

(Refer Slide Time: 07:03)

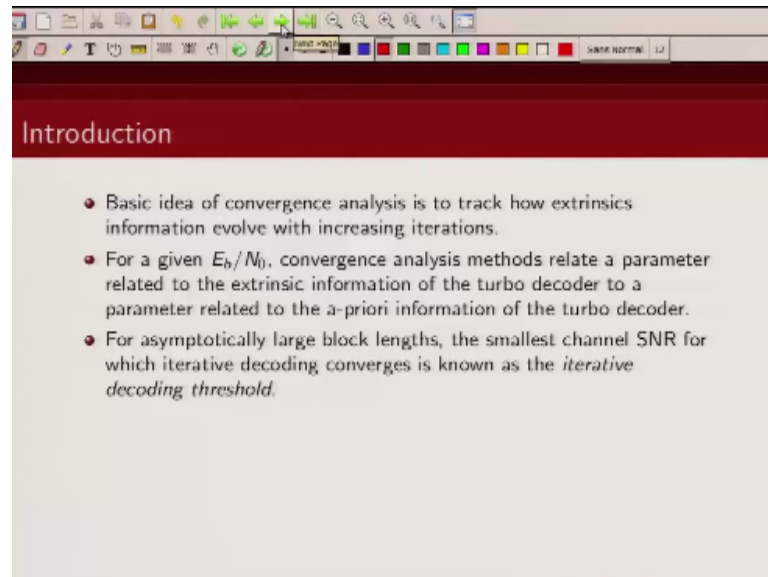


The image shows a presentation slide titled "Introduction" with a red header. The slide content is as follows:

- Basic idea of convergence analysis is to track how extrinsics information evolve with increasing iterations.
- For a given E_b/N_0 , convergence analysis methods relate a parameter related to the extrinsic information of the turbo decoder to a parameter related to the a-priori information of the turbo decoder.

Track a parameter which is related to the extrinsic information and we will see how that parameter will change when a parameter at the input side which is the a-priori value is also change.

(Refer Slide Time: 07:18)

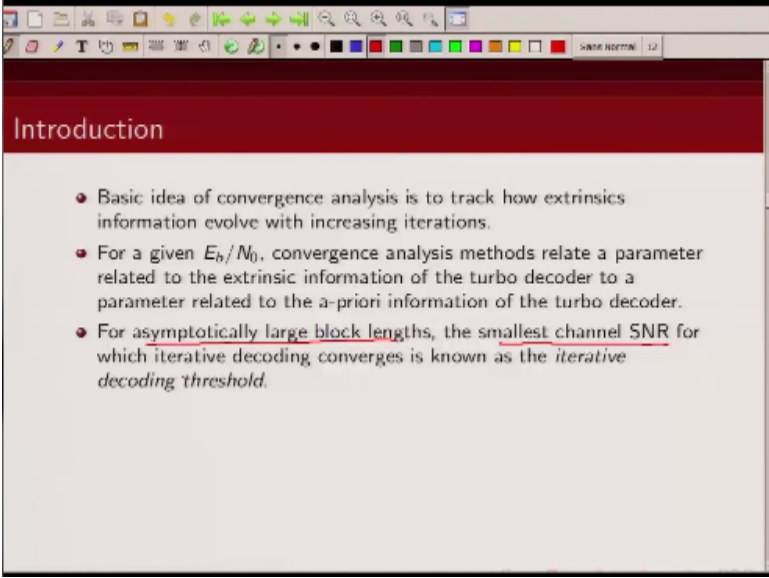


The image shows a presentation slide with a dark red header containing the word "Introduction" in white. Below the header, on a light beige background, are three bullet points. The first bullet point discusses tracking extrinsic information. The second bullet point relates parameters of the turbo decoder to E_b/N_0 . The third bullet point defines the iterative decoding threshold for large block lengths. The slide is displayed within a window that has a standard toolbar at the top and a status bar at the bottom.

- Basic idea of convergence analysis is to track how extrinsics information evolve with increasing iterations.
- For a given E_b/N_0 , convergence analysis methods relate a parameter related to the extrinsic information of the turbo decoder to a parameter related to the a-priori information of the turbo decoder.
- For asymptotically large block lengths, the smallest channel SNR for which iterative decoding converges is known as the *iterative decoding threshold*.

And for an asymptotically

(Refer Slide Time: 07:21)

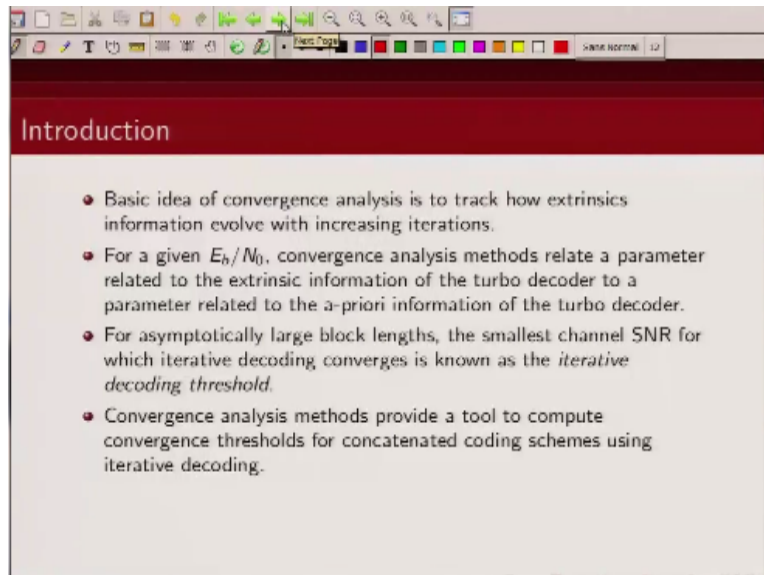


Introduction

- Basic idea of convergence analysis is to track how extrinsic information evolve with increasing iterations.
- For a given E_b/N_0 , convergence analysis methods relate a parameter related to the extrinsic information of the turbo decoder to a parameter related to the a-priori information of the turbo decoder.
- For asymptotically large block lengths, the smallest channel SNR for which iterative decoding converges is known as the *iterative decoding threshold*.

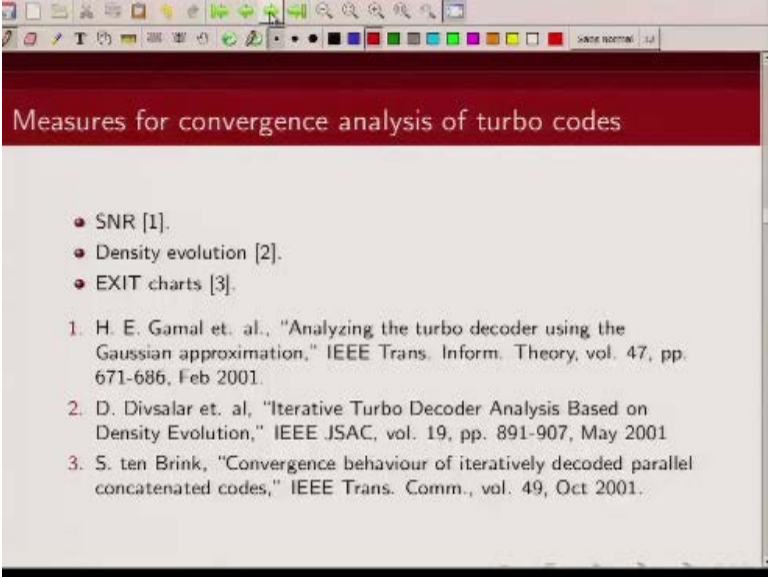
Large block size the smallest channel SNR for which iterative decoding algorithm converges is known as decoding threshold, so this iterative decoding threshold will be away from your channel capacity typically.

(Refer Slide Time: 07:40)



Now this convergence analysis tool is a very, very powerful tool to analyze these kinds of iterative decoding algorithm, it gives us tool to analyze the performance of concatenated schemes that use iterative decoding algorithm, it gives us tool to design our constituent encoder, it gives tools to design our punctuating pattern so it is a very, very interesting tool for analysis in the waterfall region.

(Refer Slide Time: 08:14)



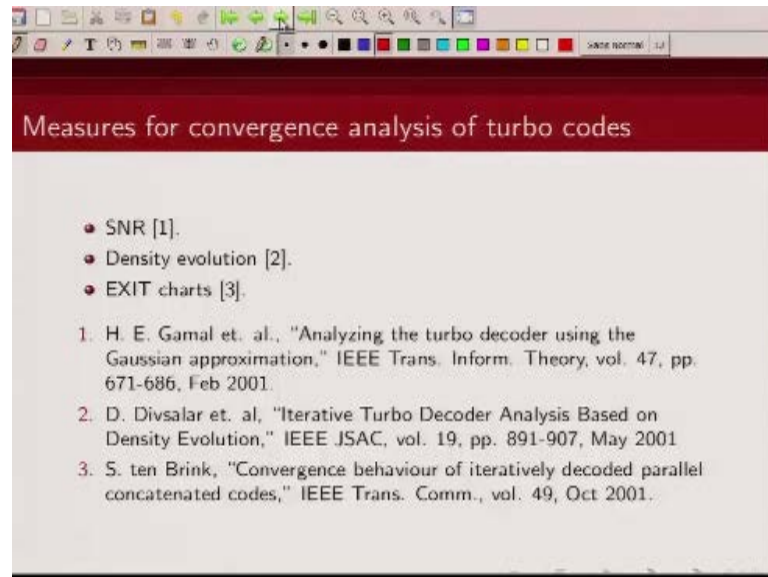
Measures for convergence analysis of turbo codes

- SNR [1].
- Density evolution [2].
- EXIT charts [3].

1. H. E. Gamal et. al., "Analyzing the turbo decoder using the Gaussian approximation," IEEE Trans. Inform. Theory, vol. 47, pp. 671-686, Feb 2001.
2. D. Divsalar et. al, "Iterative Turbo Decoder Analysis Based on Density Evolution," IEEE JSAC, vol. 19, pp. 891-907, May 2001
3. S. ten Brink, "Convergence behaviour of iteratively decoded parallel concatenated codes," IEEE Trans. Comm., vol. 49, Oct 2001.

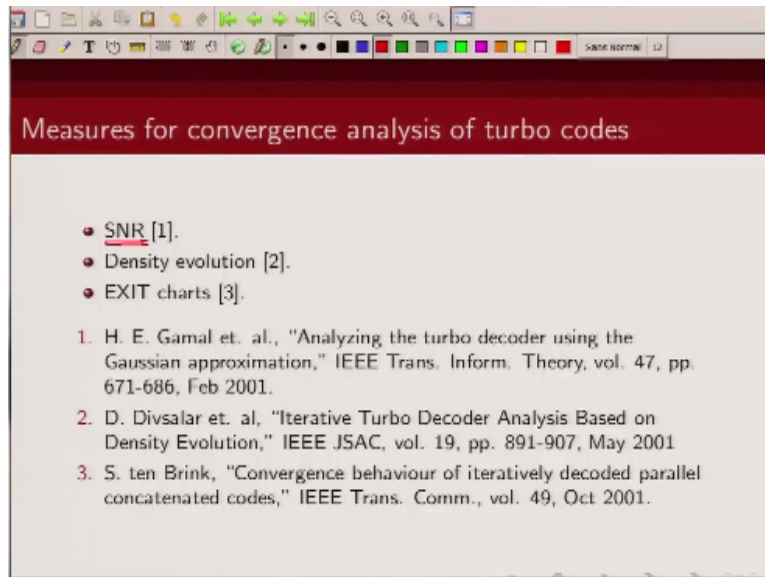
So as I said there are three popularly known techniques for convergence analysis.

(Refer Slide Time: 08:27)



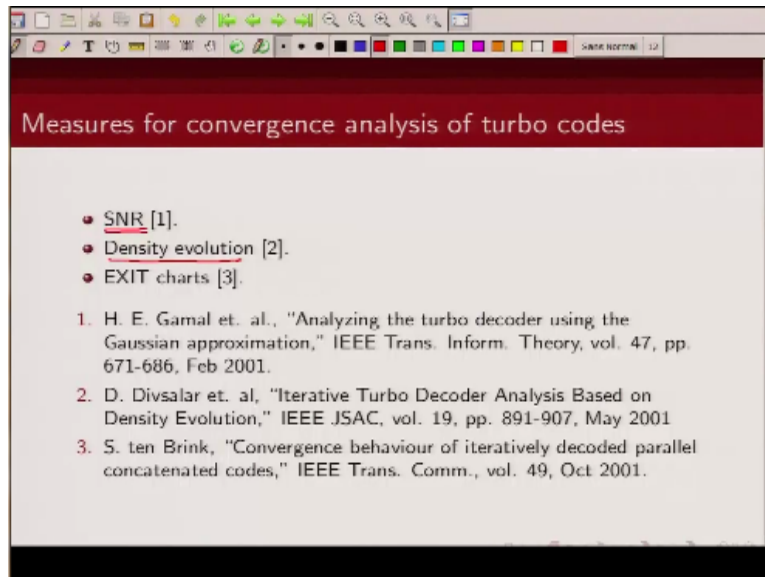
And as I said the idea of these techniques is the track one parameter which is related to the extrinsic information and track the same parameter related to the a-priori information, so this technique by El Gamal

(Refer Slide Time: 08:43)



Makes use of Gaussian approximation and it tracks the signal to noise ratio so it tracks the signal to noise ratio of the extrinsic information and observes how this SNR extrinsic information grows when you change the SNR of the a-priori information. In the density

(Refer Slide Time: 09:04)



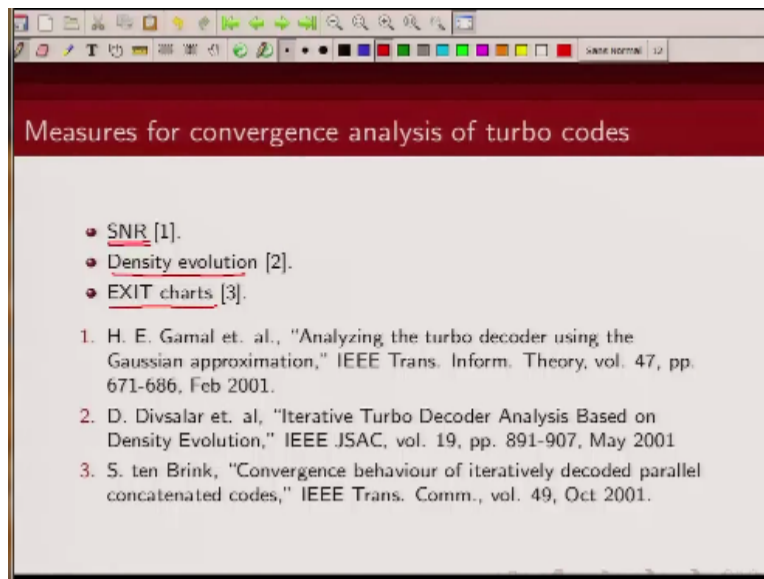
Measures for convergence analysis of turbo codes

- SNR [1].
- Density evolution [2].
- EXIT charts [3].

1. H. E. Gamal et. al., "Analyzing the turbo decoder using the Gaussian approximation," IEEE Trans. Inform. Theory, vol. 47, pp. 671-686, Feb 2001.
2. D. Divsalar et. al, "Iterative Turbo Decoder Analysis Based on Density Evolution," IEEE JSAC, vol. 19, pp. 891-907, May 2001
3. S. ten Brink, "Convergence behaviour of iteratively decoded parallel concatenated codes," IEEE Trans. Comm., vol. 49, Oct 2001.

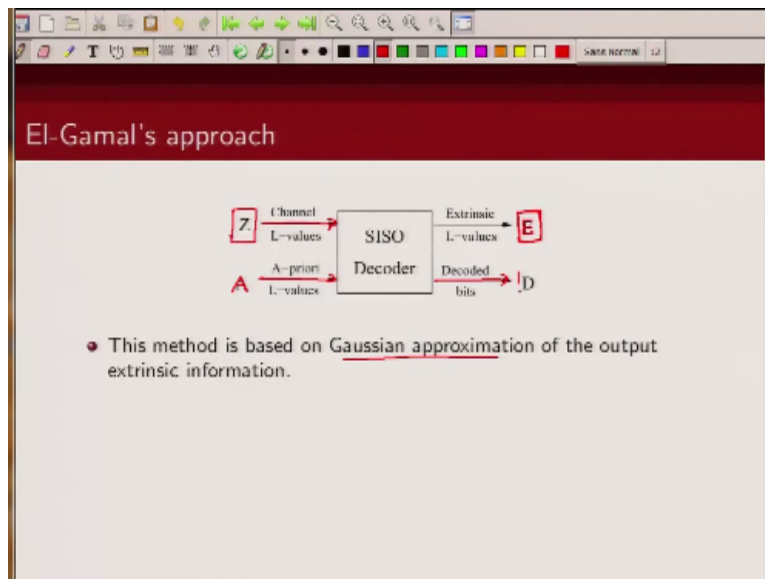
Evolution method by Divsalar and others they actually see that density of this extrinsic information how does that grow with iteration.

(Refer Slide Time: 09:17)



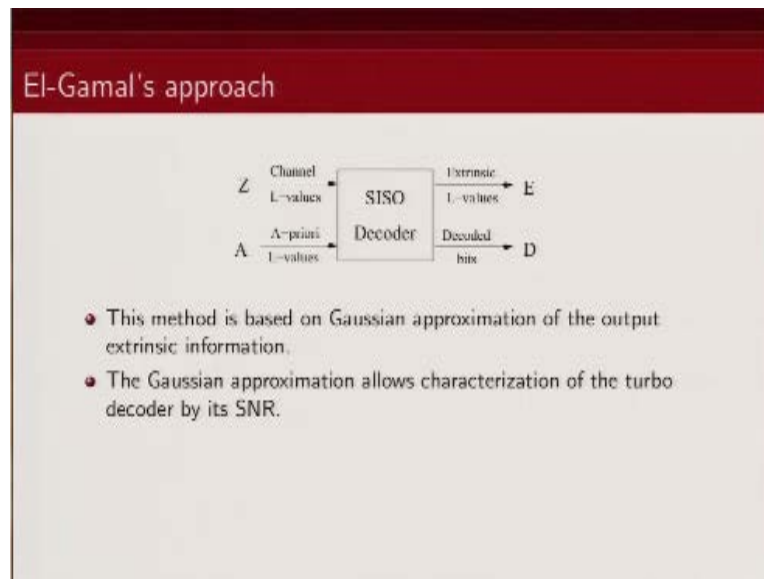
And this approach of Ten Brink which is known as extrinsic information transfer chart, it uses mutual information as a parameter to observe how with iteration your extrinsic information is growing, and these are the three references, the first one corresponding to this SNR technique, the second one corresponding to this density evolution technique and third corresponds to this exit chart technique. So the El Gamal's approach is based on Gaussian approximation

(Refer Slide Time: 09:56)



Of this output extrinsic information, so note there are two inputs to my soft inputs output decoder, one which I am referring by Z which is the channel received L values, the second one is this a-priori values and there are two outputs, one is this extrinsic information and the other one is this APPL values, if I take a hard decision on that what I get is my decoded bits.

(Refer Slide Time: 10:34)



Now we are using this Gaussian

(Refer Slide Time: 10:38)

The slide is titled "El-Gamal's approach". It features a block diagram of a SISO (Soft Input Soft Output) decoder. The diagram shows a central box labeled "SISO Decoder". To the left, there are two inputs: "Z" labeled "Channel L-values" and "A" labeled "A-priori L-values". To the right, there are two outputs: "E" labeled "Extrinsic L-values" and "D" labeled "Decoded bits".

- This method is based on Gaussian approximation of the output extrinsic information.
- The Gaussian approximation allows characterization of the turbo decoder by its SNR.
- For an AWGN channel,

$$z = x + n$$

where z is the received channel value, x is the transmitted bit ($= \pm 1$), and n is Gaussian distributed with zero mean and variance $N_0/2$.

Approximations so assume so we have one Gaussian channel

(Refer Slide Time: 10:41)

El-Gamal's approach

Diagram illustrating El-Gamal's approach for turbo decoding:

- Input: Channel L-values (Z)
- Block: SISO (Soft Input Soft Output)
- Output: Extrinsic L-values (E)
- Block: Decoder
- Input: A-priori L-values (A)
- Output: Decoded bits (D)

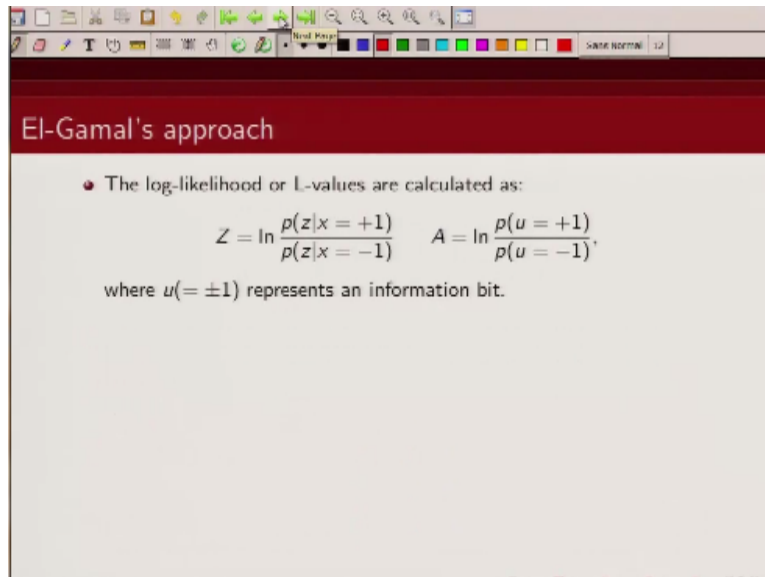
- This method is based on Gaussian approximation of the output extrinsic information.
- The Gaussian approximation allows characterization of the turbo decoder by its SNR.
- For an AWGN channel,

$$z = x + n$$

where z is the received channel value, x is the transmitted bit ($= \pm 1$), and n is Gaussian distributed with zero mean and variance $N_0/2$.

So if x was a modulated signal and n is my Gaussian noise so what I received is Z .

(Refer Slide Time: 10:51)



The image is a screenshot of a presentation slide. At the top, there is a red header bar with the text "El-Gamal's approach" in white. Below the header, the slide content is on a light gray background. It starts with a bullet point: "• The log-likelihood or L-values are calculated as:". This is followed by two mathematical expressions:
$$Z = \ln \frac{p(z|x=+1)}{p(z|x=-1)} \quad A = \ln \frac{p(u=+1)}{p(u=-1)},$$
 and then a line of text: "where $u(=\pm 1)$ represents an information bit." The slide is displayed within a window that has a standard toolbar at the top with various icons for navigation and editing.

El-Gamal's approach

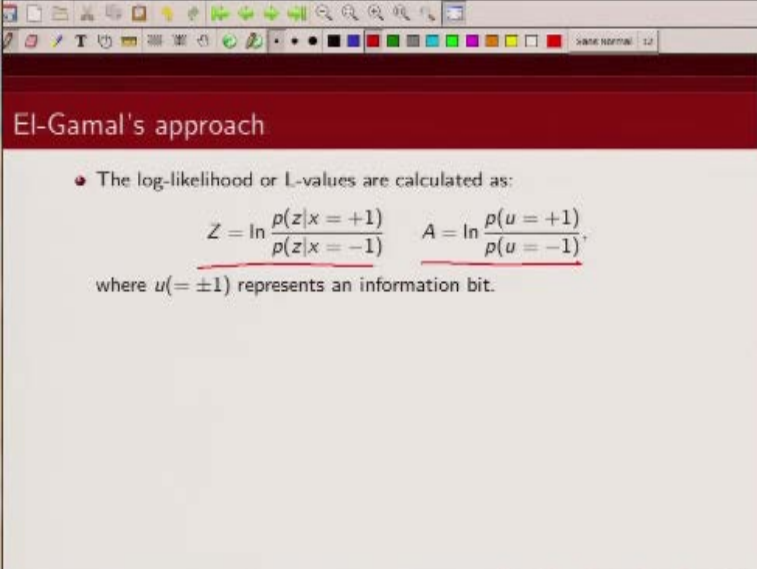
- The log-likelihood or L-values are calculated as:

$$Z = \ln \frac{p(z|x=+1)}{p(z|x=-1)} \quad A = \ln \frac{p(u=+1)}{p(u=-1)},$$

where $u(=\pm 1)$ represents an information bit.

Now the likelihood ratio of Z

(Refer Slide Time: 10:51)



The screenshot shows a presentation slide with a red header bar containing the text "El-Gamal's approach". Below the header, there is a bullet point stating "The log-likelihood or L-values are calculated as:". This is followed by two equations: $Z = \ln \frac{p(z|x=+1)}{p(z|x=-1)}$ and $A = \ln \frac{p(u=+1)}{p(u=-1)}$. Red underlines are drawn under the denominators of both fractions. Below the equations, a line of text states "where $u(=\pm 1)$ represents an information bit."

El-Gamal's approach

- The log-likelihood or L-values are calculated as:

$$Z = \ln \frac{p(z|x=+1)}{p(z|x=-1)} \quad A = \ln \frac{p(u=+1)}{p(u=-1)},$$

where $u(=\pm 1)$ represents an information bit.

We can write it like this, similarly this a-priori information the L value of that I can write it like this.

(Refer Slide Time: 11:05)

El-Gamal's approach

- The log-likelihood or L-values are calculated as:
$$Z = \ln \frac{p(z|x = +1)}{p(z|x = -1)} \quad A = \ln \frac{p(u = +1)}{p(u = -1)},$$
- where $u(\in \pm 1)$ represents an information bit.
- For large block sizes, the probability distribution of the a-priori L-values p_A , are assumed to be Gaussian. In particular, the a-priori L-value A can be modeled as
$$A = \mu_A \cdot u + n_A$$

where the n_A is a zero mean Gaussian random variable with variance σ_A^2 that satisfies the following condition

$$\mu_A = \frac{\sigma_A^2}{2}, \quad (\text{consistency condition})$$

Now for large block sizes this a-priori distribution is

(Refer Slide Time: 11:08)

El-Gamal's approach

- The log-likelihood or L-values are calculated as:

$$Z = \ln \frac{p(z|x=+1)}{p(z|x=-1)} \quad A = \ln \frac{p(u=+1)}{p(u=-1)},$$

where $u(\in \pm 1)$ represents an information bit.

- For large block sizes, the probability distribution of the a-priori L-values p_A , are assumed to be Gaussian. In particular, the a-priori L-value A can be modeled as

$$A = \mu_A \cdot u + n_A$$

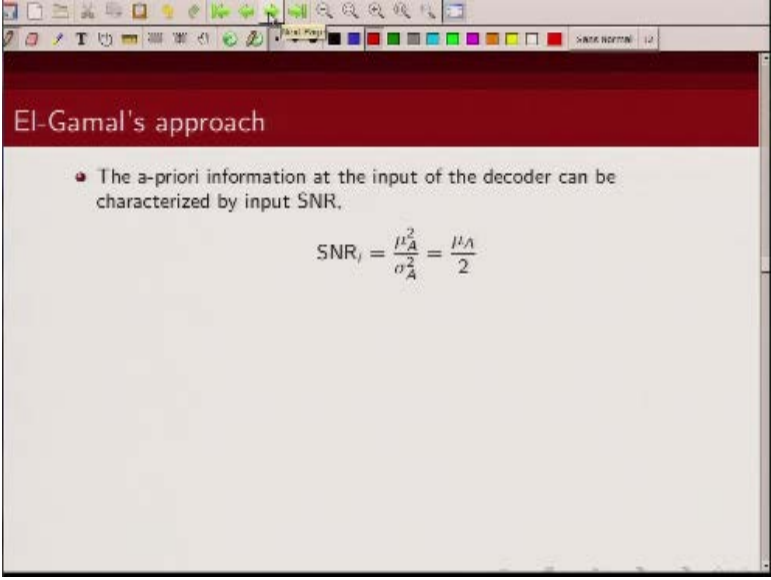
where the n_A is a zero mean Gaussian random variable with variance σ_A^2 that satisfies the following condition

$$\mu_A = \frac{\sigma_A^2}{2}, \quad (\text{consistency condition})$$

Assumed to be Gaussian, so we model this a-priori L value in this particular way in this El Gamal's approach so in El Gamal's approach we modeled our a-priori information as Gaussian and we generate it like this A is μ_A times input + some Gaussian noise, and they have also observed what they call consistency condition so they assume the mean and variants are related in this particular fashion.

So what happens is if you make this Gaussian assumption and you make this assumption that mean and variants are related then you essentially need to track only one parameter. So you for example with just a mean you can track your Gaussian distribution, because mean and variants are related.

(Refer Slide Time: 12:14)



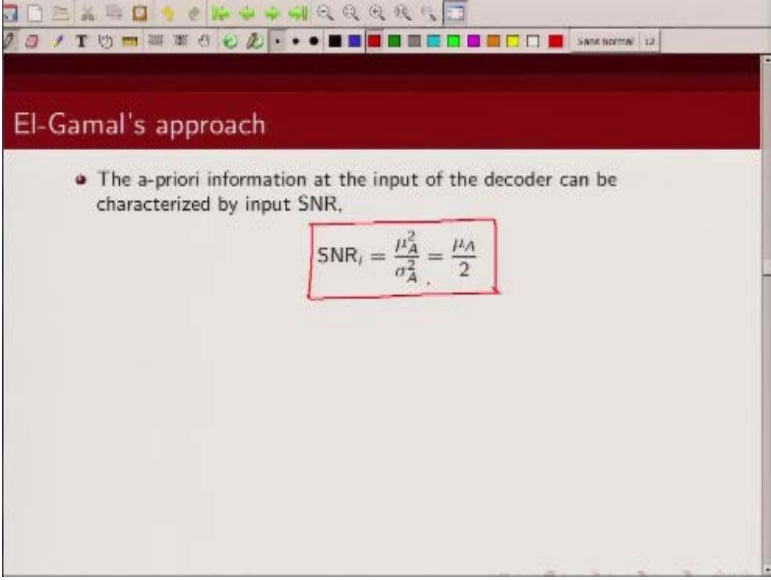
El-Gamal's approach

- The a-priori information at the input of the decoder can be characterized by input SNR.

$$\text{SNR}_I = \frac{\mu_A^2}{\sigma_A^2} = \frac{\mu_A}{2}$$

Now similarly we can define input

(Refer Slide Time: 12:18)



The screenshot shows a presentation slide with a red header bar containing the text "El-Gamal's approach". Below the header, there is a bullet point stating: "The a-priori information at the input of the decoder can be characterized by input SNR,". Below this text, the equation $SNR_i = \frac{\mu_A^2}{\sigma_A^2} = \frac{\mu_A}{2}$ is displayed and enclosed in a red rectangular box. The slide is viewed through a software window with a standard toolbar at the top.

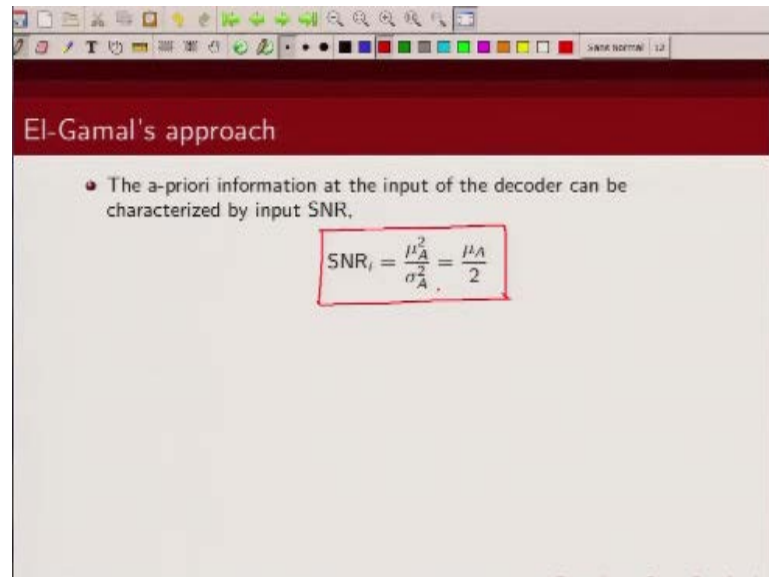
El-Gamal's approach

- The a-priori information at the input of the decoder can be characterized by input SNR,

$$SNR_i = \frac{\mu_A^2}{\sigma_A^2} = \frac{\mu_A}{2}$$

SNR of the a-priori information, this is μ^2 / σ^2 now $\sigma^2 / 2$

(Refer Slide Time: 12:18)



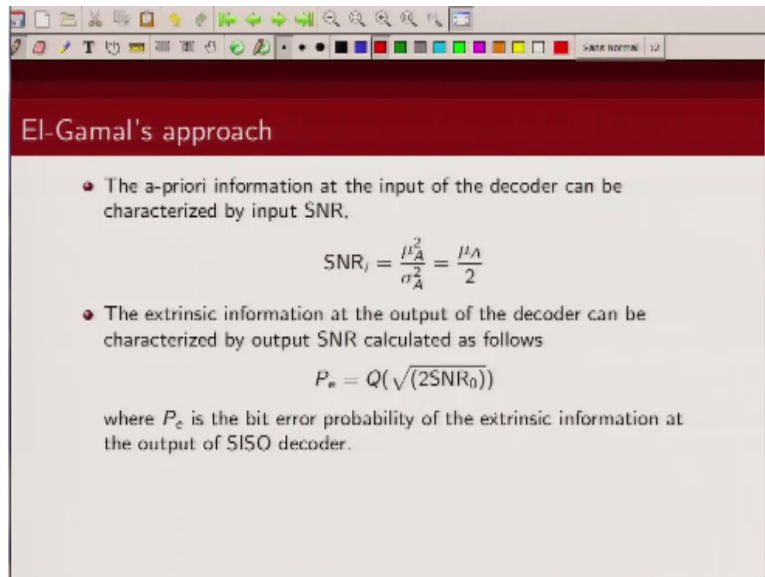
El-Gamal's approach

- The a-priori information at the input of the decoder can be characterized by input SNR,

$$\text{SNR}_i = \frac{\mu_A^2}{\sigma_A^2} = \frac{\mu_A}{2}$$

Is μ_A so our input SNR is given by the mean of the a-priori information divided by 2.

(Refer Slide Time: 12:37)



The slide is titled "El-Gamal's approach" in a red header. It contains two bullet points. The first bullet point states: "The a-priori information at the input of the decoder can be characterized by input SNR," followed by the equation
$$\text{SNR}_i = \frac{\mu_A^2}{\sigma_A^2} = \frac{\mu_A}{2}$$
. The second bullet point states: "The extrinsic information at the output of the decoder can be characterized by output SNR calculated as follows" followed by the equation
$$P_e = Q(\sqrt{2\text{SNR}_0})$$
. Below this equation, a note states: "where P_e is the bit error probability of the extrinsic information at the output of SISO decoder."

El-Gamal's approach

- The a-priori information at the input of the decoder can be characterized by input SNR,
$$\text{SNR}_i = \frac{\mu_A^2}{\sigma_A^2} = \frac{\mu_A}{2}$$
- The extrinsic information at the output of the decoder can be characterized by output SNR calculated as follows
$$P_e = Q(\sqrt{2\text{SNR}_0})$$

where P_e is the bit error probability of the extrinsic information at the output of SISO decoder.

And since our output is approximated as Gaussian so we can calculate

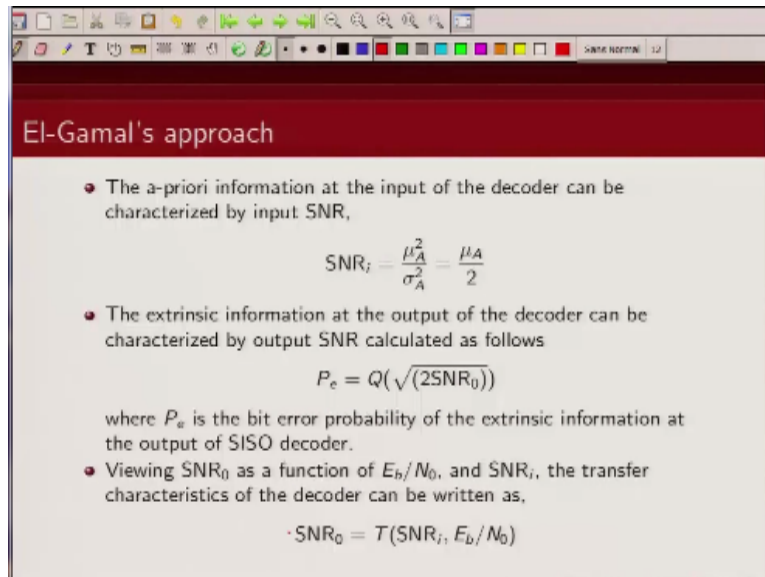
(Refer Slide Time: 12:46)

El-Gamal's approach

- The a-priori information at the input of the decoder can be characterized by input SNR,
$$\text{SNR}_i = \frac{\mu_A^2}{\sigma_A^2} = \frac{\mu_A}{2}$$
- The extrinsic information at the output of the decoder can be characterized by output SNR calculated as follows
$$P_e = Q(\sqrt{2\text{SNR}_0})$$
where P_e is the bit error probability of the extrinsic information at the output of SISO decoder.

The output probability of error as the function of output SNR; they are related to using this Q function.

(Refer Slide Time: 12:59)

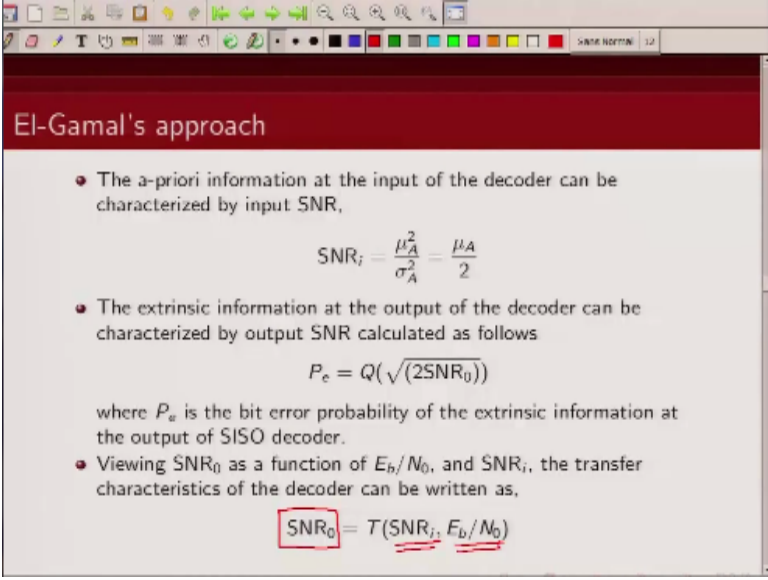


El-Gamal's approach

- The a-priori information at the input of the decoder can be characterized by input SNR,
$$\text{SNR}_i = \frac{\mu_A^2}{\sigma_A^2} = \frac{\mu_A}{2}$$
- The extrinsic information at the output of the decoder can be characterized by output SNR calculated as follows
$$P_e = Q(\sqrt{2\text{SNR}_0})$$
where P_e is the bit error probability of the extrinsic information at the output of SISO decoder.
- Viewing SNR_0 as a function of E_b/N_0 , and SNR_i , the transfer characteristics of the decoder can be written as,
$$\text{SNR}_0 = T(\text{SNR}_i, E_b/N_0)$$

Now so what we can do is we can write this output

(Refer Slide Time: 13:02)



El-Gamal's approach

- The a-priori information at the input of the decoder can be characterized by input SNR,
$$\text{SNR}_i = \frac{\mu_A^2}{\sigma_A^2} = \frac{\mu_A}{2}$$
- The extrinsic information at the output of the decoder can be characterized by output SNR calculated as follows
$$P_e = Q(\sqrt{2\text{SNR}_0})$$
where P_e is the bit error probability of the extrinsic information at the output of SISO decoder.
- Viewing SNR_0 as a function of E_b/N_0 , and SNR_i , the transfer characteristics of the decoder can be written as,
$$\boxed{\text{SNR}_0} = T(\text{SNR}_i, E_b/N_0)$$

SNR in terms of input SNR and our operating signal to noise ratio, so what we can do is we can view the output SNR of the extrinsic information as a function of input SNR of a-priori information as well as the channel operating signal to noise ratio.

(Refer Slide Time: 13:34)

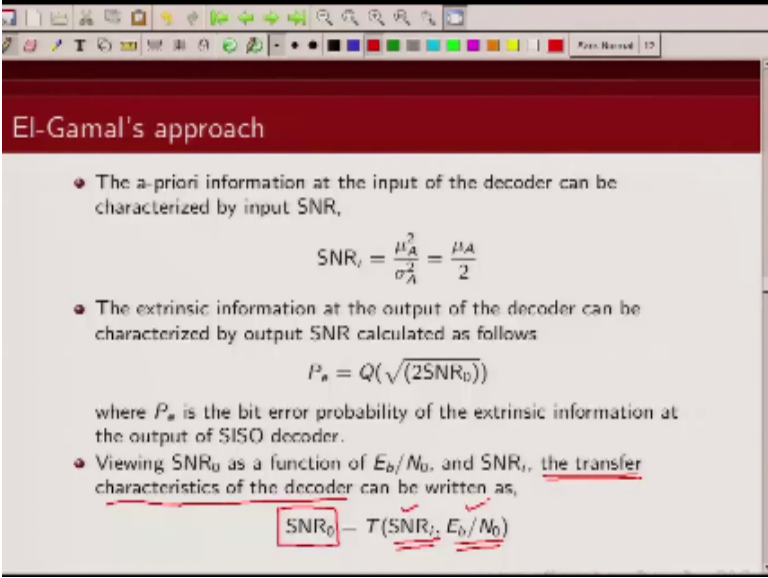
El-Gamal's approach

- The a-priori information at the input of the decoder can be characterized by input SNR,
$$\text{SNR}_i = \frac{\mu_A^2}{\sigma_A^2} = \frac{\mu_A}{2}$$
- The extrinsic information at the output of the decoder can be characterized by output SNR calculated as follows
$$P_e = Q(\sqrt{2\text{SNR}_0})$$

where P_e is the bit error probability of the extrinsic information at the output of SISO decoder.
- Viewing SNR_0 as a function of E_b/N_0 , and SNR_i , the transfer characteristics of the decoder can be written as,
$$\boxed{\text{SNR}_0} = \underline{T(\text{SNR}_i, E_b/N_0)}$$

So this is crucial so this is basically what I call the transfer characteristics of the decoder because my decoder is a function of a-priori inputs as well as channel received values. Now a channel received value is a function of channel operating SNR and what I get a-priori information is a function of a-priori input SNR, so I can view my SNR of the extrinsic information, I can view

(Refer Slide Time 14: 09)



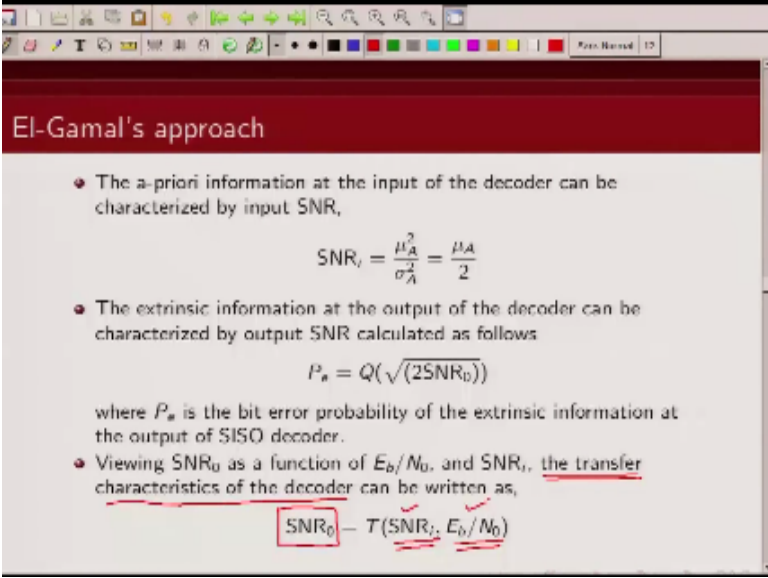
The screenshot shows a presentation slide with a red header bar containing the title "El-Gamal's approach". The slide content includes three bullet points. The first bullet point states that a-priori information at the decoder input is characterized by input SNR, followed by the equation $SNR_i = \frac{\mu_a^2}{\sigma_a^2} = \frac{\mu_a}{2}$. The second bullet point states that extrinsic information at the decoder output is characterized by output SNR, followed by the equation $P_e = Q(\sqrt{2SNR_o})$ and a note that P_e is the bit error probability. The third bullet point states that viewing SNR_o as a function of E_b/N_0 and SNR_i , the transfer characteristics of the decoder can be written as, followed by the boxed equation $SNR_o = T(SNR_i, E_b/N_0)$. The slide also features a standard software toolbar at the top.

El-Gamal's approach

- The a-priori information at the input of the decoder can be characterized by input SNR,
$$SNR_i = \frac{\mu_a^2}{\sigma_a^2} = \frac{\mu_a}{2}$$
- The extrinsic information at the output of the decoder can be characterized by output SNR calculated as follows
$$P_e = Q(\sqrt{2SNR_o})$$
where P_e is the bit error probability of the extrinsic information at the output of SISO decoder.
- Viewing SNR_o as a function of E_b/N_0 , and SNR_i , the transfer characteristics of the decoder can be written as,
$$SNR_o = T(SNR_i, E_b/N_0)$$

It as a function of input SNR of a-priori values as well as channel operating channel signal to noise ratio. So this relation characterizes how my decoder will behave because remember with iteration your extrinsic information is changing as a function of a-priori value and what is your operating channel SNR, so this transfer function

(Refer Slide Time 14: 37)



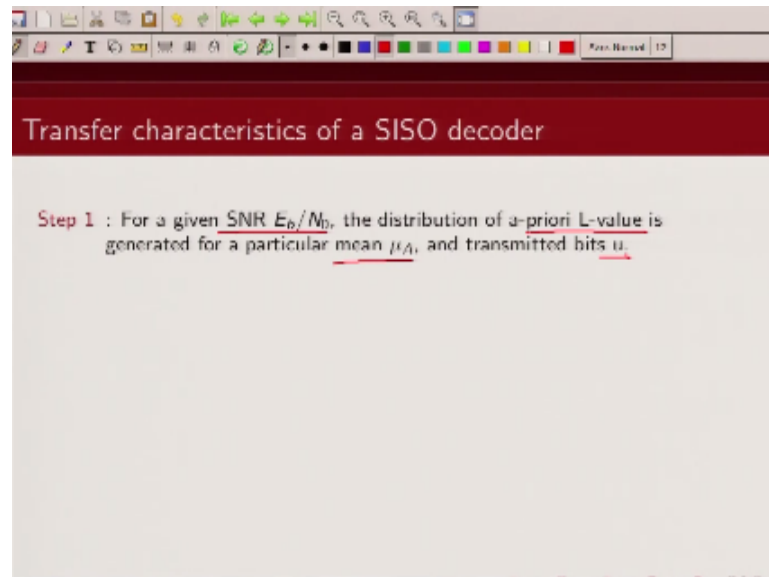
The screenshot shows a presentation slide with a red header bar containing the title "El-Gamal's approach". The slide content includes three bullet points. The first bullet point states that a-priori information at the decoder input is characterized by input SNR, followed by the equation $SNR_i = \frac{\mu_a^2}{\sigma_a^2} = \frac{\mu_a}{2}$. The second bullet point states that extrinsic information at the decoder output is characterized by output SNR, followed by the equation $P_e = Q(\sqrt{2SNR_o})$ and a note that P_e is the bit error probability. The third bullet point states that viewing SNR_o as a function of E_b/N_0 and SNR_i , the transfer characteristics can be written as, followed by the boxed equation $SNR_o = T(SNR_i, E_b/N_0)$. The slide is displayed in a window with a standard toolbar at the top.

El-Gamal's approach

- The a-priori information at the input of the decoder can be characterized by input SNR,
$$SNR_i = \frac{\mu_a^2}{\sigma_a^2} = \frac{\mu_a}{2}$$
- The extrinsic information at the output of the decoder can be characterized by output SNR calculated as follows
$$P_e = Q(\sqrt{2SNR_o})$$
where P_e is the bit error probability of the extrinsic information at the output of SISO decoder.
- Viewing SNR_o as a function of E_b/N_0 , and SNR_i , the transfer characteristics of the decoder can be written as,
$$SNR_o = T(SNR_i, E_b/N_0)$$

Will give me how my decoder this soft input, soft output decoder how it will perform as a function of a-priori value and the channel operating SNR.

(Refer Slide Time 14: 51)

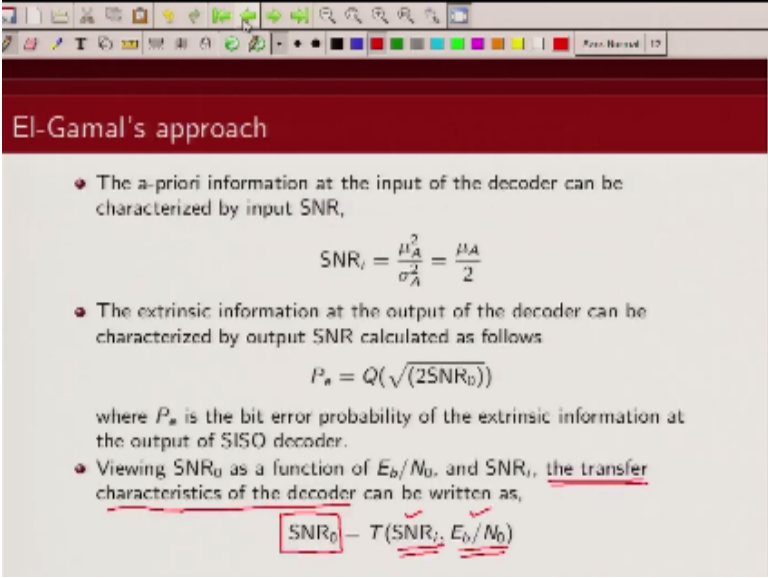


Transfer characteristics of a SISO decoder

Step 1 : For a given SNR E_b/N_0 , the distribution of a-priori L-value is generated for a particular mean μ_A , and transmitted bits u .

So then how do we draw the transfer characteristics for a given signal to noise ratio, the distribution of a-priori L values is generated for a particular mean new way and transmitted bit you.

(Refer Slide Time 15: 10)



The screenshot shows a presentation slide with a red header bar containing the title "El-Gamal's approach". The slide content includes two bullet points. The first bullet point states that a-priori information at the decoder input is characterized by input SNR, followed by the equation $SNR_i = \frac{\mu_A^2}{\sigma_A^2} = \frac{\mu_A}{2}$. The second bullet point states that extrinsic information at the decoder output is characterized by output SNR, followed by the equation $P_e = Q(\sqrt{2SNR_0})$. A text line explains that P_e is the bit error probability of the extrinsic information at the output of a SISO decoder. The third bullet point states that viewing SNR_0 as a function of E_b/N_0 and SNR_i , the transfer characteristics of the decoder can be written as, followed by the equation $SNR_0 = T(SNR_i, E_b/N_0)$. The equation is annotated with red boxes and checkmarks: a box around SNR_0 , a checkmark over SNR_i , and a checkmark over E_b/N_0 .

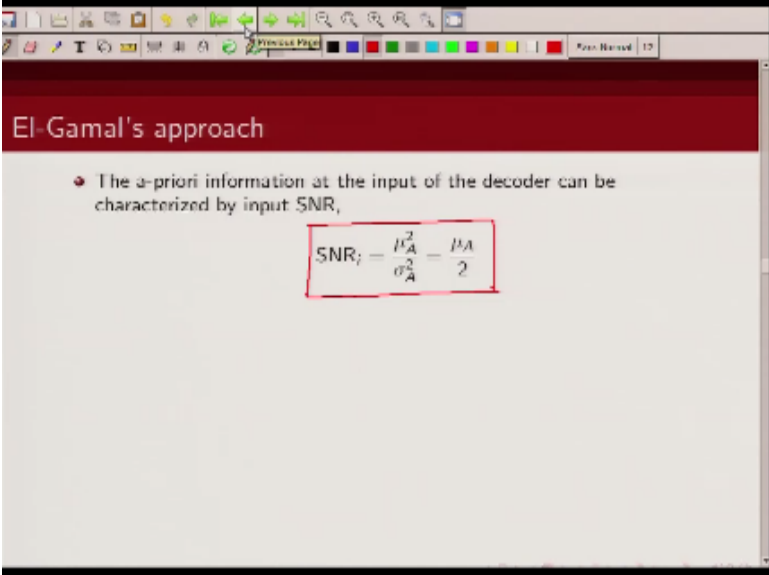
El-Gamal's approach

- The a-priori information at the input of the decoder can be characterized by input SNR,
$$SNR_i = \frac{\mu_A^2}{\sigma_A^2} = \frac{\mu_A}{2}$$
- The extrinsic information at the output of the decoder can be characterized by output SNR calculated as follows
$$P_e = Q(\sqrt{2SNR_0})$$

where P_e is the bit error probability of the extrinsic information at the output of SISO decoder.
- Viewing SNR_0 as a function of E_b/N_0 , and SNR_i , the transfer characteristics of the decoder can be written as,
$$SNR_0 = T(SNR_i, E_b/N_0)$$

How? We know that

(Refer Slide Time 15: 11)

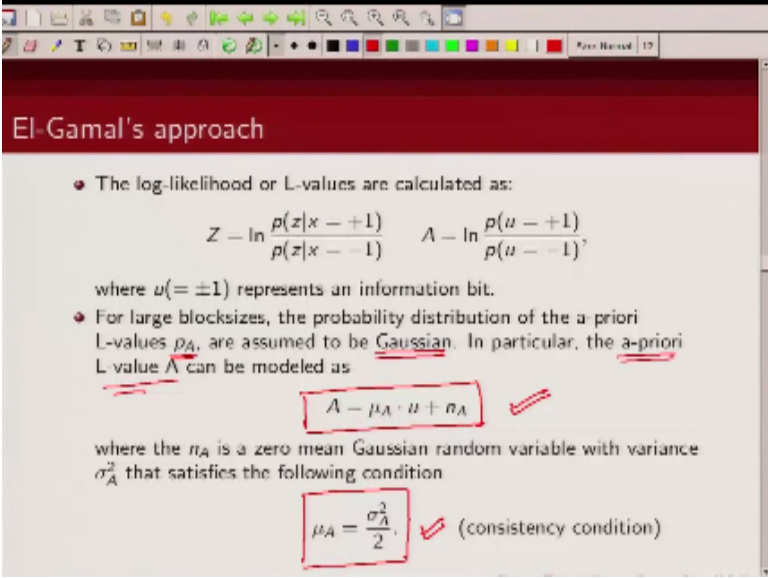


El-Gamal's approach

- The a-priori information at the input of the decoder can be characterized by input SNR,

$$\text{SNR}_i = \frac{P_A^2}{\sigma_A^2} = \frac{P_A}{2}$$

(Refer Slide Time 15: 12)

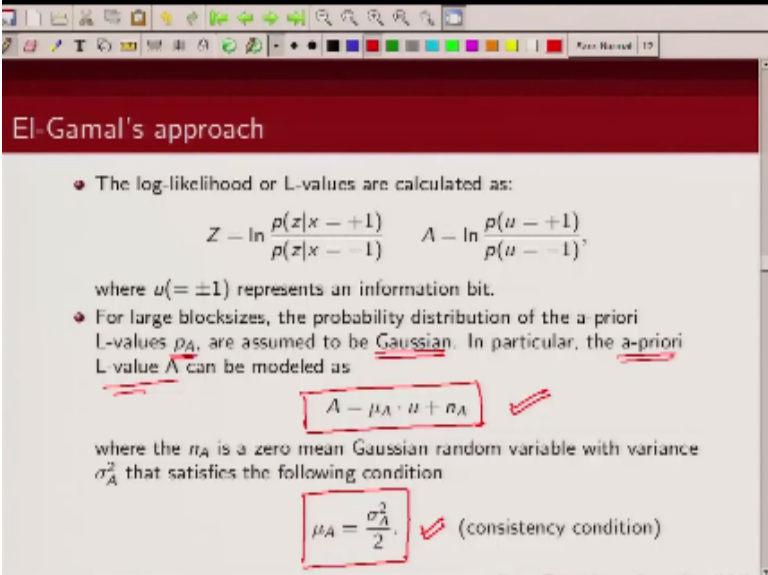


The slide is titled "El-Gamal's approach" in a red header. It contains the following content:

- The log-likelihood or L-values are calculated as:
$$Z = \ln \frac{p(z|x=+1)}{p(z|x=-1)} \quad \Lambda = \ln \frac{p(u=+1)}{p(u=-1)},$$
- where $u(=\pm 1)$ represents an information bit.
- For large block sizes, the probability distribution of the a-priori L-values p_A , are assumed to be Gaussian. In particular, the a-priori L-value Λ can be modeled as
$$\Lambda = \mu_A \cdot u + n_A$$
- where the n_A is a zero mean Gaussian random variable with variance σ_A^2 that satisfies the following condition
$$\mu_A = \frac{\sigma_A^2}{2}, \quad \checkmark \text{ (consistency condition)}$$

We are modeling our a-priori information like this and of course we are assuming consistency condition, so the mean and variants of the mutual the a-priori information is related like this.

(Refer Slide Time 15: 29)

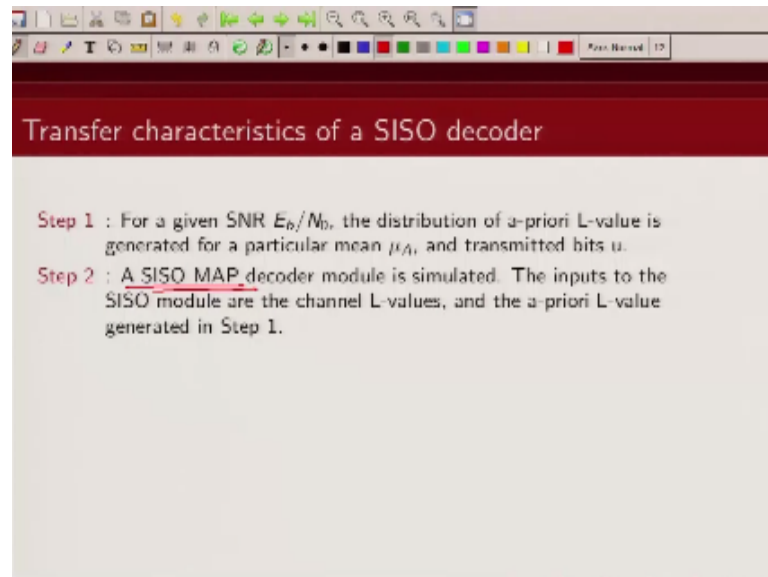


The slide is titled "El-Gamal's approach" in a red header. It contains two bullet points. The first bullet point states that log-likelihood or L-values are calculated as: $Z = \ln \frac{p(z|x=+1)}{p(z|x=-1)}$ and $\Lambda = \ln \frac{p(u=+1)}{p(u=-1)}$. It then explains that $u (= \pm 1)$ represents an information bit. The second bullet point states that for large block sizes, the probability distribution of the a-priori L-values p_A are assumed to be Gaussian. In particular, the a-priori L-value Λ can be modeled as $\Lambda = \mu_A \cdot u + n_A$, where n_A is a zero mean Gaussian random variable with variance σ_A^2 that satisfies the following condition: $\mu_A = \frac{\sigma_A^2}{2}$. This condition is labeled as the "consistency condition".

El-Gamal's approach

- The log-likelihood or L-values are calculated as:
$$Z = \ln \frac{p(z|x=+1)}{p(z|x=-1)} \quad \Lambda = \ln \frac{p(u=+1)}{p(u=-1)},$$
where $u (= \pm 1)$ represents an information bit.
- For large block sizes, the probability distribution of the a-priori L-values p_A , are assumed to be Gaussian. In particular, the a-priori L-value Λ can be modeled as
$$\Lambda = \mu_A \cdot u + n_A$$
where the n_A is a zero mean Gaussian random variable with variance σ_A^2 that satisfies the following condition
$$\mu_A = \frac{\sigma_A^2}{2}, \quad \checkmark \text{ (consistency condition)}$$

(Refer Slide Time 15: 31)



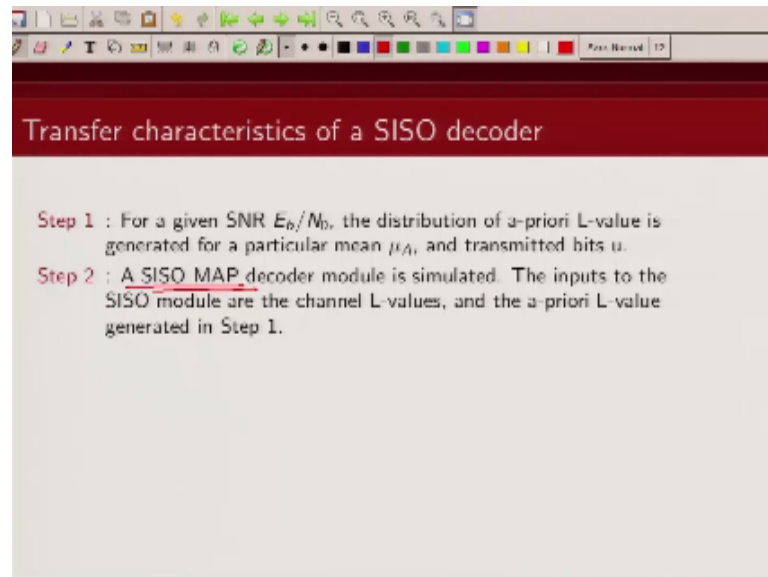
Transfer characteristics of a SISO decoder

Step 1 : For a given SNR E_b/N_0 , the distribution of a-priori L-value is generated for a particular mean μ_A , and transmitted bits u .

Step 2 : A SISO MAP decoder module is simulated. The inputs to the SISO module are the channel L-values, and the a-priori L-value generated in Step 1.

So next step is we simulate a soft input, soft output decoder so we feed in these two input, one is this channel received SNR, other is this a-priori information.

(Refer Slide Time 15: 44)



Transfer characteristics of a SISO decoder

Step 1 : For a given SNR E_b/N_0 , the distribution of a-priori L-value is generated for a particular mean μ_{A_i} and transmitted bits u .

Step 2 : A SISO MAP decoder module is simulated. The inputs to the SISO module are the channel L-values, and the a-priori L-value generated in Step 1.

Which we have modeled as Gaussian. We feed these two inputs to the decoder and what comes out as output are these extrinsic values.

(Refer Slide Time 15: 58)

Transfer characteristics of a SISO decoder

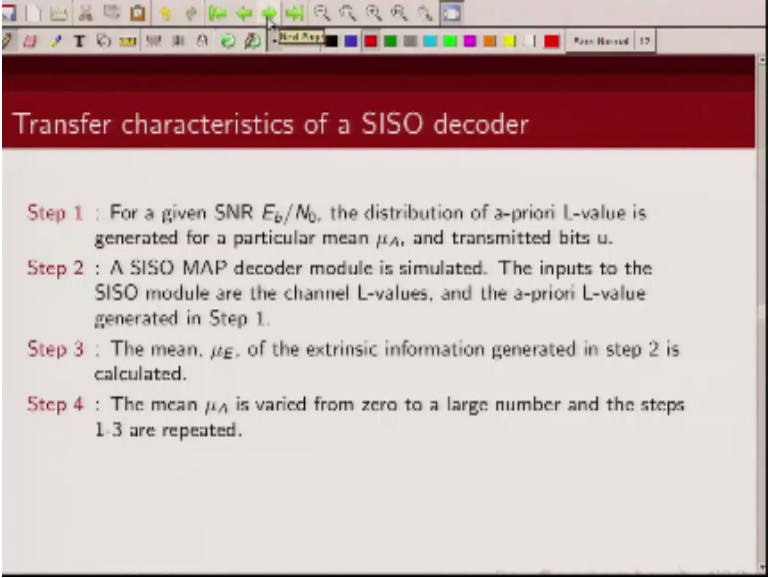
Step 1 : For a given SNR E_b/N_0 , the distribution of a-priori L-value is generated for a particular mean μ_A , and transmitted bits u .

Step 2 : A SISO MAP decoder module is simulated. The inputs to the SISO module are the channel L-values, and the a-priori L-value generated in Step 1.

Step 3 : The mean, μ_E , of the extrinsic information generated in step 2 is calculated.

And we compute the mean of the extrinsic values.

(Refer Slide Time 16: 06)

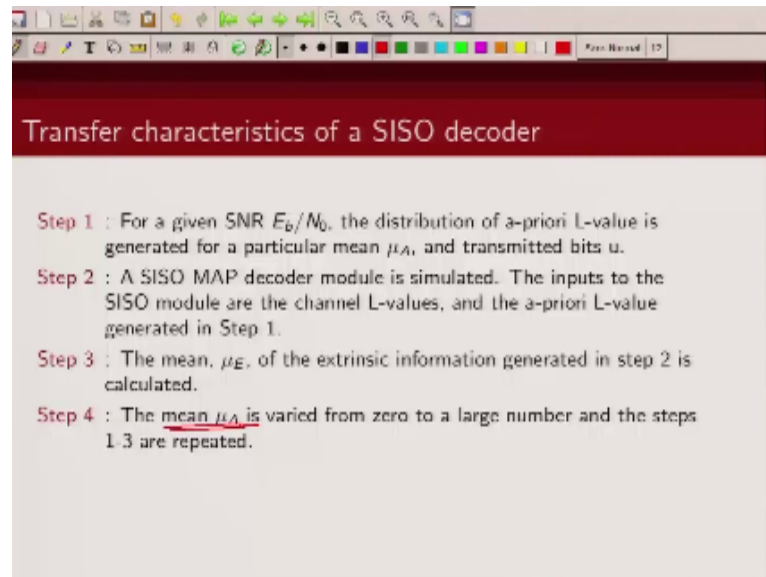


Transfer characteristics of a SISO decoder

- Step 1** : For a given SNR E_b/N_0 , the distribution of a-priori L-value is generated for a particular mean μ_A and transmitted bits u .
- Step 2** : A SISO MAP decoder module is simulated. The inputs to the SISO module are the channel L-values, and the a-priori L-value generated in Step 1.
- Step 3** : The mean, μ_E , of the extrinsic information generated in step 2 is calculated.
- Step 4** : The mean μ_A is varied from zero to a large number and the steps 1-3 are repeated.

Now we know that our signal to noise ratio because we are making Gaussian assumption of signal to noise ratio is related to the mean now as I said with iteration my a-priori information is changing so now are going to change the

(Refer Slide Time 16: 25)

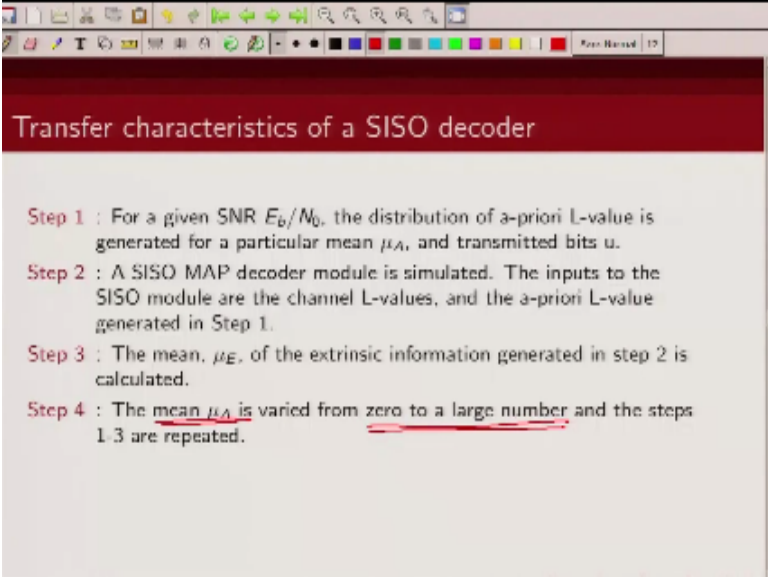


Transfer characteristics of a SISO decoder

- Step 1 :** For a given SNR E_b/N_0 , the distribution of a-priori L-value is generated for a particular mean μ_A , and transmitted bits u .
- Step 2 :** A SISO MAP decoder module is simulated. The inputs to the SISO module are the channel L-values, and the a-priori L-value generated in Step 1.
- Step 3 :** The mean, μ_E , of the extrinsic information generated in step 2 is calculated.
- Step 4 :** The mean μ_A is varied from zero to a large number and the steps 1-3 are repeated.

Mean of the a-priori information and then we will against simulate this soft input, soft output decoder and we will try to see what happens to the extrinsic information mean, how much it is growing with change in input a-priori information mean.

(Refer Slide Time 16: 47)



Transfer characteristics of a SISO decoder

- Step 1** : For a given SNR E_b/N_0 , the distribution of a-priori L-value is generated for a particular mean μ_A , and transmitted bits u .
- Step 2** : A SISO MAP decoder module is simulated. The inputs to the SISO module are the channel L-values, and the a-priori L-value generated in Step 1.
- Step 3** : The mean, μ_E , of the extrinsic information generated in step 2 is calculated.
- Step 4** : The mean μ_A is varied from zero to a large number and the steps 1-3 are repeated.

So this process is done, so we repeat this by varying our a-priori information mean.

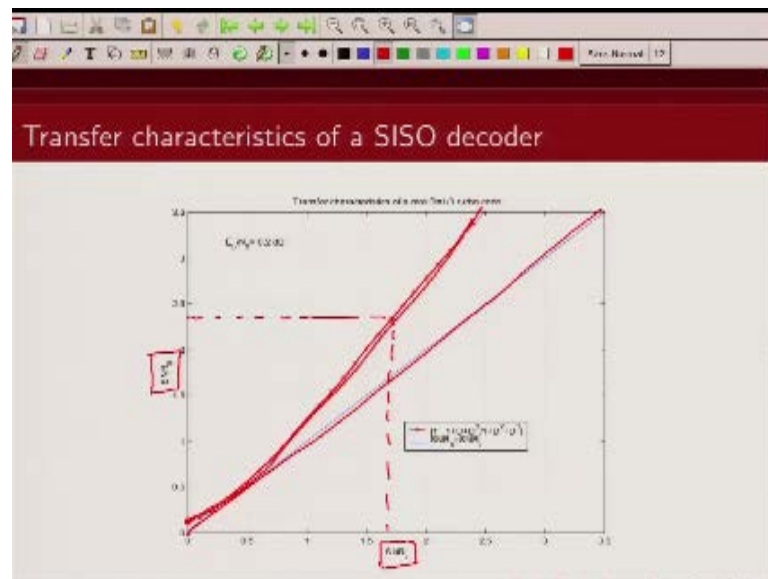
(Refer Slide Time 16: 59)

Transfer characteristics of a SISO decoder

- Step 1 :** For a given SNR E_b/N_0 , the distribution of a-priori L-value is generated for a particular mean μ_A , and transmitted bits u .
- Step 2 :** A SISO MAP decoder module is simulated. The inputs to the SISO module are the channel L-values, and the a-priori L-value generated in Step 1.
- Step 3 :** The mean, μ_E , of the extrinsic information generated in step 2 is calculated.
- Step 4 :** The mean μ_A is varied from zero to a large number and the steps 1-3 are repeated.
- Step 5 :** The set of $(\text{SNR}_i, \text{SNR}_o)$ for different values of μ_a is plotted. This is then used as the transfer characteristics for the SISO module for that particular code, and channel SNR E_b/N_0 .

And finally what we do we plot this input, output relation for a particular channel SNR so this is my input a-priori SNR, this is the extrinsic information SNR, we plot it for a particular value of signal to noise ratio, and this is my transfer characteristic for that particular decoder which is a function of channel operating SNR and of course it is a function of the constituent encoders that I have used.

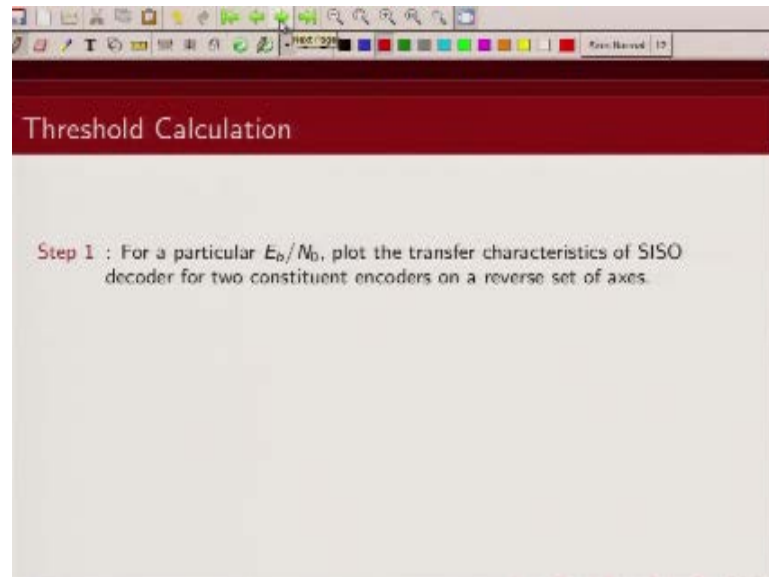
(Refer Slide Time 17: 37)



So here basically I have plotted with red curve, I have plotted transfer characteristics of one such code, it is a eight state code, what I have here at the input side is SNR of a-priori information and what I have here on the output side is SNR of the extrinsic information and this is how my, so initially I do not have any a-priori knowledge, the extrinsic information will this, this is the amount of extrinsic information which is generated. So this was, this transfer characteristics will tell me if I have a particular input a-priori information.

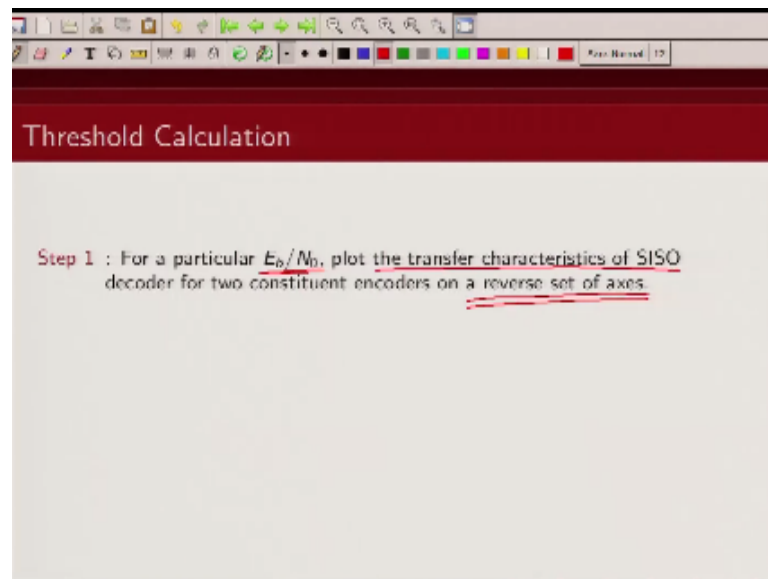
Then what is the corresponding extrinsic information SNR, and for comparisons sake I have drawn this line which is the SNR in equal to SNR out. Now if you have a symmetric turbo code you obviously would like your transfer characteristics to be above this line.

(Refer Slide Time 18: 48)



Now how do we compute, how do we use this transfer characteristics to compute the decoding threshold? So how do we find out this SNR, minimum SNR under which our reiterative algorithm will converge, so for that we need to do this threshold computation, so how do we do this threshold computation?

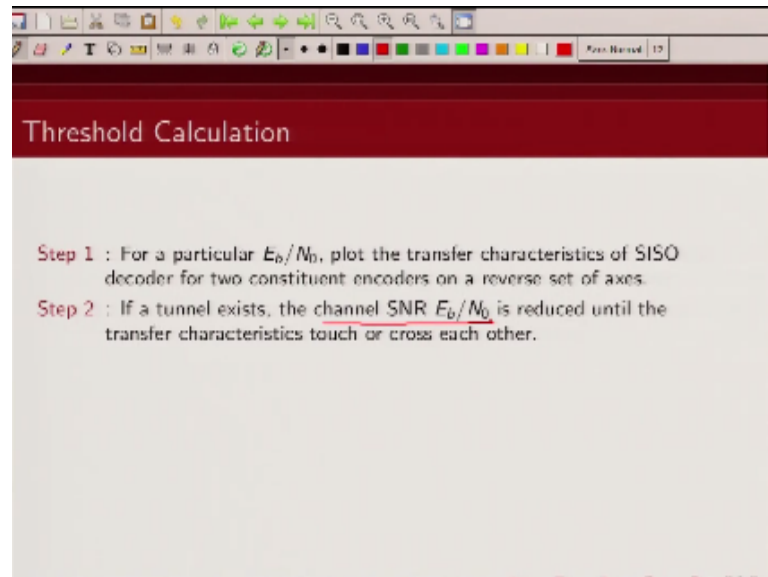
(Refer Slide Time 19: 15)



So for a particular signal to noise ratio we plot the transfer characteristics of this soft input, soft output decoder, we plot them on reverse set of axis's. Now what do I mean by reverse set of axis's? So for the first my SNR in is on X axis and SNR out is on the Y axis. Now for the second decoder my SNR in is on the Y axis, and SNR out is on the X axis. Now why do I do this, because the extrinsic information of first decoder is input to the second decoder so SNR out of the first decoder becomes SNR in of the second decoder and that is why I put this SNR in second decoder as Y axis, and the SNR out

Of the second decoder is SNR in for the first decoder because the extrinsic information from the second decoder is coming as input to the as a-priori input to the first decoder and that is the reason I plot these transfer characteristics on reverse axis's.

(Refer Slide Time 20: 42)



Now if these transfer characteristics do not cross there is a tunnel in the sense they do not touch each other then what we do is, the channel operating channel SNR is reduced until these transfer characteristics just about touch. So what is the effect of channel SNR? So as you reduce the channel SNR these transfer characteristics which have been plot on reverse axis, they come closer when you reduce the channel SNR so the smallest SNR for which there is still a tunnel that is your decoding threshold for that particular code.

(Refer Slide Time: 21:35)

Threshold Calculation

- Step 1 : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes.
- Step 2 : If a tunnel exists, the channel SNR E_b/N_0 is reduced until the transfer characteristics touch or cross each other.

(Refer Slide Time: 21:36)

Threshold Calculation

- Step 1 : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes.
- Step 2 : If a tunnel exists, the channel SNR E_b/N_0 is reduced until the transfer characteristics touch or cross each other.
- Step 3 : If the transfer characteristics touch or cross each other, the channel SNR E_b/N_0 is increased until a tunnel exists.

So if that transfer characteristic touch across each other then what we need to do is we need to increase SNR until there is a tunnel, still a tunnel.

(Refer Slide Time: 21:48)

Threshold Calculation

- Step 1 : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes.
- Step 2 : If a tunnel exists, the channel SNR E_b/N_0 is reduced until the transfer characteristics touch or cross each other.
- Step 3 : If the transfer characteristics touch or cross each other, the channel SNR E_b/N_0 is increased until a tunnel exists.
- Step 4 : The smallest channel SNR E_b/N_0 for which the transfer characteristics do not touch and a tunnel exists is the convergence threshold for that particular code.

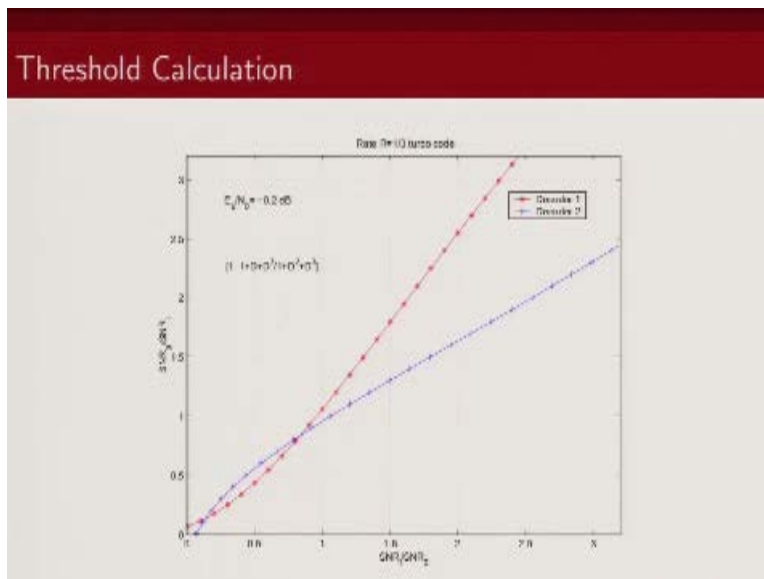
(Refer Slide Time: 21:51)

Threshold Calculation

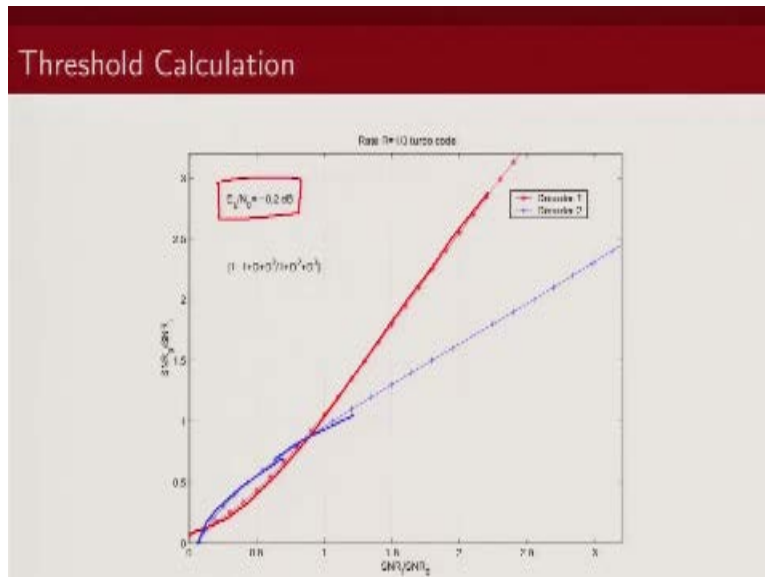
- Step 1 : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes.
- Step 2 : If a tunnel exists, the channel SNR E_b/N_0 is reduced until the transfer characteristics touch or cross each other.
- Step 3 : If the transfer characteristics touch or cross each other, the channel SNR E_b/N_0 is increased until a tunnel exists.
- Step 4 : The smallest channel SNR E_b/N_0 for which the transfer characteristics do not touch and a tunnel exists is the convergence threshold for that particular code.

So the smallest channel SNR for which these two transfer characteristics which have been plotted on reverse axis, they do not touch and a tunnel exist is basically the convergence threshold for that particular code, so that would give the SNR minimum SNR under which that particular code will converge and it will have a waterfall kind of behavior if you take large enough block size.

(Refer Slide Time: 22:23)

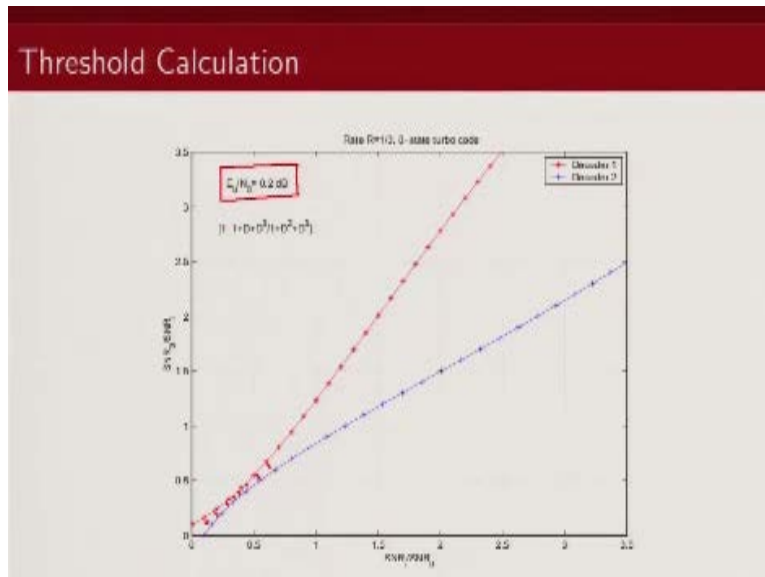


Refer Slide Time: 22:30)



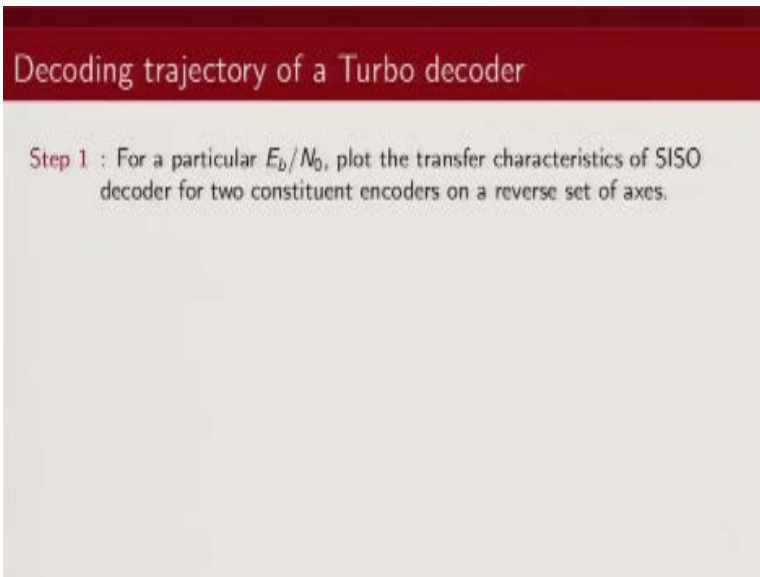
This is one example, now note here this is plotted for channel operating SNR of -0.2 db so this is in red curve with my decoder one and in blue color we have decoder two, note that these two are crossing each other so there is no tunnel.

(Refer Slide Time: 22:48)



Now same port, now I increase my SNR and I made it 0.2 db, now you can see there is a tunnel between them there is a tunnel, okay.

(Refer Slide Time: 23:05)



The slide features a dark red header with the title "Decoding trajectory of a Turbo decoder" in white. Below the header, the text "Step 1 : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes." is displayed in a dark font. The main body of the slide is a light gray color.

Decoding trajectory of a Turbo decoder

Step 1 : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes.

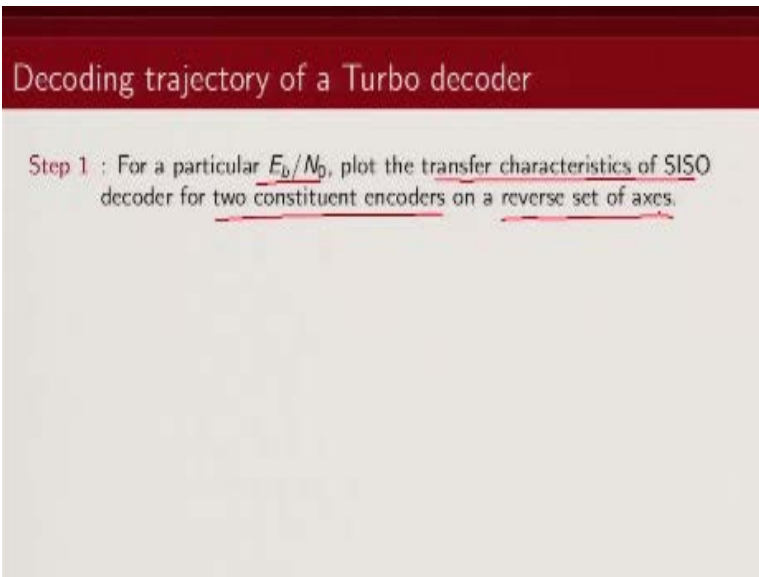
Now let us see how we can draw the decoding trajectory of a turbo decoder with the help of these transfer characteristics.

(Refer Slide Time: 23:18)

Decoding trajectory of a Turbo decoder

Step 1 : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes.

(Refer Slide Time: 23:20)



Decoding trajectory of a Turbo decoder

Step 1 : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes.

So what we do is for a particular signal to noise ratio as I said we plot these transfer characteristics of two constituent encoders on reverse set of axis's, so for decoder one SNR in will be on x axis, SNR out will be on y axis, whereas for decoder two SNR in will be on y axis and SNR out will be on x axis.

(Refer Slide Time: 23:48)

Decoding trajectory of a Turbo decoder

- Step 1** : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes.
- Step 2** : For a given E_b/N_0 , initially $\text{SNR}_i = 0$ corresponding to the first iteration of decoder 1, we determine the resulting SNR_0 (vertically) using the transfer characteristics for decoder 1.

(Refer Slide Time: 23:52)

Decoding trajectory of a Turbo decoder

- Step 1 : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes.
- Step 2 : For a given E_b/N_0 , initially $\text{SNR}_i = 0$ corresponding to the first iteration of decoder 1, we determine the resulting SNR_0 (vertically) using the transfer characteristics for decoder 1.

So initially because you do not have any a-priority knowledge about the information bits so initially the, a-priory SNR is zero.

(Refer Slide Time: 24:01)

Decoding trajectory of a Turbo decoder

- Step 1** : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes.
- Step 2** : For a given E_b/N_0 , initially $SNR_i = 0$ corresponding to the first iteration of decoder 1, we determine the resulting SNR_0 (vertically) using the transfer characteristics for decoder 1.

And this corresponds to and so we are first going to look at the transfer characteristics for the first decoder, so input we will give zero so we will try to see what is the output extrinsic SNR corresponding to this decoder one.

(Refer Slide Time: 24:20)

Decoding trajectory of a Turbo decoder

- Step 1** : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes.
- Step 2** : For a given E_b/N_0 , initially $\text{SNR}_i = 0$ corresponding to the first iteration of decoder 1, we determine the resulting SNR_0 (vertically) using the transfer characteristics for decoder 1.

So we determine the resulting output SNR which we look vertically for using the transfer characteristics for decoder one.

(Refer Slide Time: 24:30)

Decoding trajectory of a Turbo decoder

- Step 1** : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes.
- Step 2** : For a given E_b/N_0 , initially $\text{SNR}_i = 0$ corresponding to the first iteration of decoder 1, we determine the resulting SNR_0 (vertically) using the transfer characteristics for decoder 1.
- Step 3** : Since the extrinsic information at the output of decoder 1 becomes the a-priori information at the input of decoder 2, the value of SNR_0 from decoder 1 becomes SNR_i for the first iteration of decoder 2, and the resulting SNR_0 for decoder 2 is determined (horizontally) using the transfer characteristics for decoder 2.

Now as I said since the extrinsic information from the first decoder is actually a-priori value for the second decoder so what we are going to do is that particular extrinsic information will now become SNR in for the decoder two.

(Refer Slide Time: 24:50)

Decoding trajectory of a Turbo decoder

- Step 1 : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes.
- Step 2 : For a given E_b/N_0 , initially $\text{SNR}_i = 0$ corresponding to the first iteration of decoder 1, we determine the resulting SNR_0 (vertically) using the transfer characteristics for decoder 1.
- Step 3 : Since the extrinsic information at the output of decoder 1 becomes the a-priori information at the input of decoder 2, the value of SNR_0 from decoder 1 becomes SNR_i for the first iteration of decoder 2, and the resulting SNR_0 for decoder 2 is determined (horizontally) using the transfer characteristics for decoder 2.

So the SNR out that we got from the transfer characteristics of decoder one that is our new a-priori SNR in for decoder two. Now we are going to look at the transfer characteristics of decoder two and we are going to go horizontal and find a point corresponding to that particular a-priori SNR, what is the output SNR?

(Refer Slide Time: 25:26)

Decoding trajectory of a Turbo decoder

- Step 1** : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes.
- Step 2** : For a given E_b/N_0 , initially $\text{SNR}_i = 0$ corresponding to the first iteration of decoder 1, we determine the resulting SNR_0 (vertically) using the transfer characteristics for decoder 1.
- Step 3** : Since the extrinsic information at the output of decoder 1 becomes the a-priori information at the input of decoder 2, the value of SNR_0 from decoder 1 becomes SNR_i for the first iteration of decoder 2, and the resulting SNR_0 for decoder 2 is determined (horizontally) using the transfer characteristics for decoder 2.
- Step 4** : This procedure is repeated to trace the trajectory of iterative decoding.

And this process we are going to repeat to draw the decoding trajectory of turbo decoder.

(Refer Slide Time: 25:37)

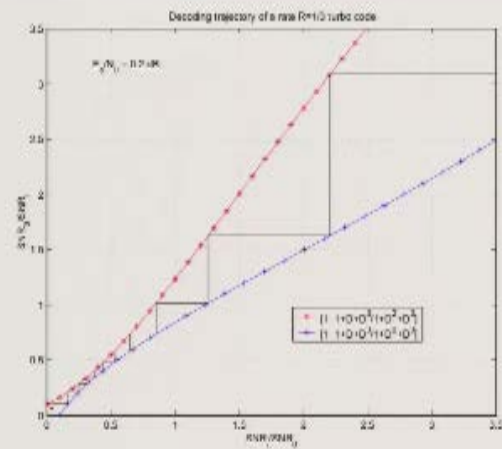
Decoding trajectory of a Turbo decoder

- Step 1** : For a particular E_b/N_0 , plot the transfer characteristics of SISO decoder for two constituent encoders on a reverse set of axes.
- Step 2** : For a given E_b/N_0 , initially $\text{SNR}_i = 0$ corresponding to the first iteration of decoder 1, we determine the resulting SNR_0 (vertically) using the transfer characteristics for decoder 1.
- Step 3** : Since the extrinsic information at the output of decoder 1 becomes the a-priori information at the input of decoder 2, the value of SNR_0 from decoder 1 becomes SNR_i for the first iteration of decoder 2, and the resulting SNR_0 for decoder 2 is determined (horizontally) using the transfer characteristics for decoder 2.
- Step 4** : This procedure is repeated to trace the trajectory of iterative decoding.
- Step 5** : If a tunnel exists between the two transfer characteristics, iterative decoding converges.

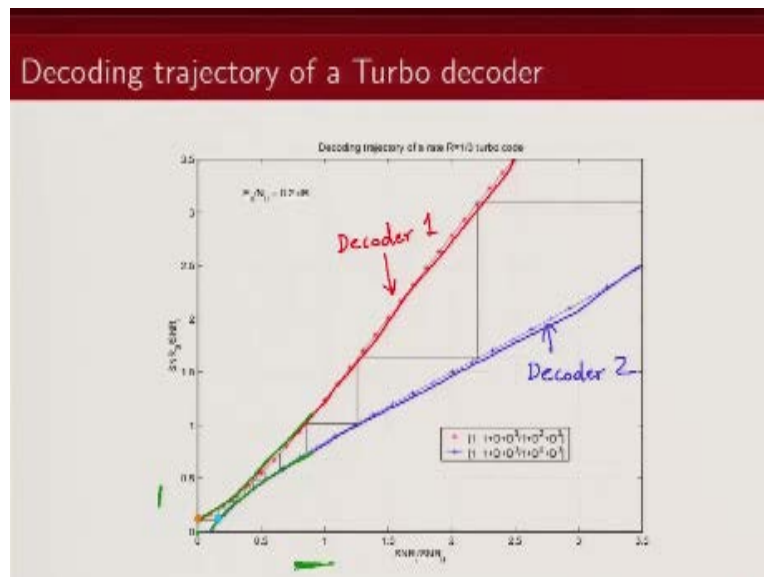
If while drawing this decoding trajectory our decoding trajectory does not get stuck, our decoding trajectory will not get stuck if there is a tunnel and if there is these transfer characteristic cross each other than our decoding trajectory will get stuck.

(Refer Slide Time: 25:57)

Decoding trajectory of a Turbo decoder



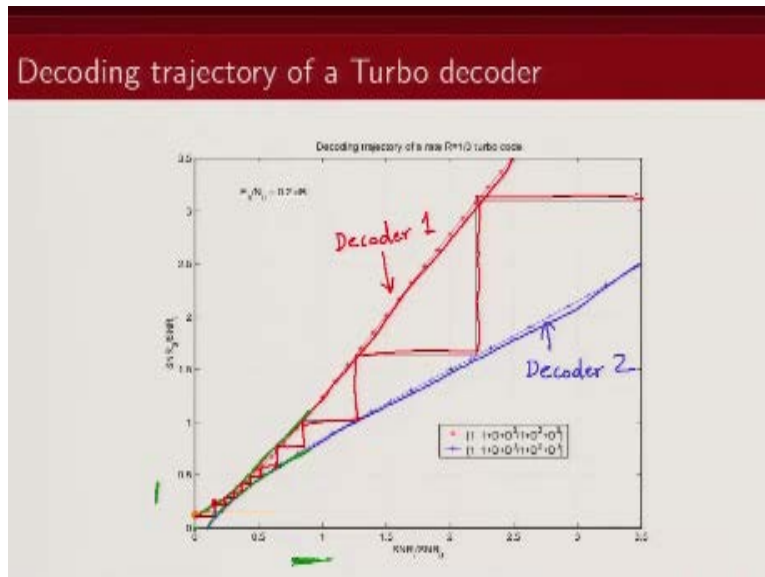
(Refer Slide Time: 26:07)



So this an example, so I have this with red that you see that is the transfer characteristics of the first decoder, this is decoder one this is the transfer characteristics of decoder one, and what you see in blue is the transfer characteristics of decoder two. They are the same encoder this is a symmetric turbo code I am considering, so how do I start? So initially I will look at the transfer characteristics of the first decoder, this is where I will look so initially I do not have any a-priory knowledge, so I will start from this point and I am looking at this curve so this is my extrinsic SNR corresponding to zero input.

Now note that this extrinsic information that we are getting from decoder one is going to be the a-priory information for decoder two so then what we will do, so we will now look at this curve which is transfer characteristics of decoder two. For decoder two this side is input and this side is output. This is input and this is output so we will look here and we will look horizontally so this is the point, so this is the point corresponding to SNR out corresponding to decoder two. Now note this extrinsic information is getting fed as a-priory information to decoder one.

(Refer Slide Time: 27:53)



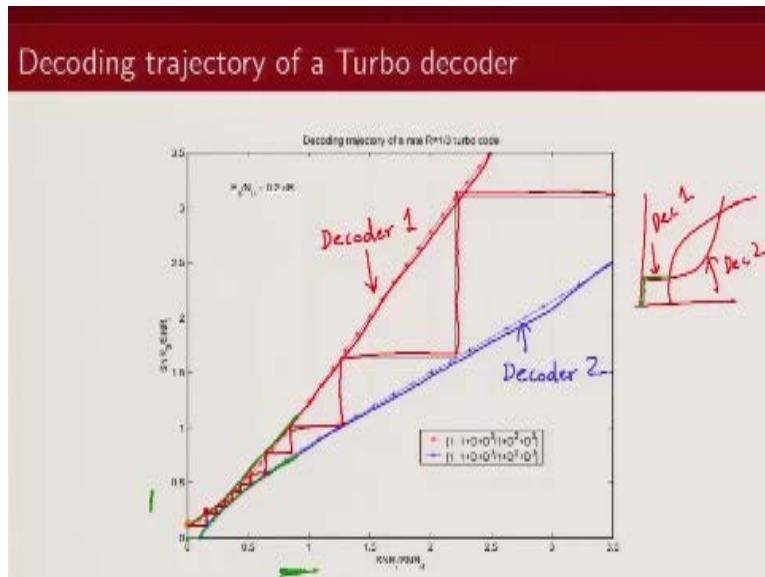
So we will look at decoder one classified characteristics and this is the point, so you can see I am going like this, you see this is how basically my decoding trajectory of my turbo decoder is happening.

(Refer Slide Time: 28:20)

Density Evolution

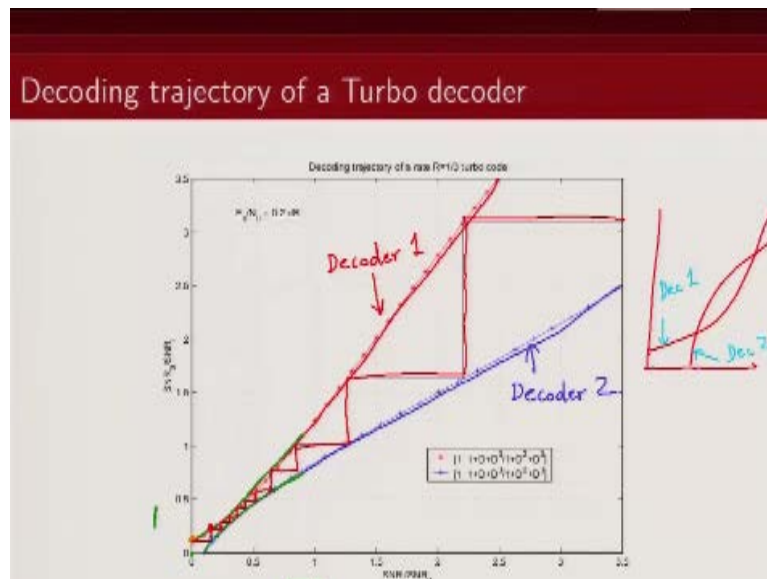
- This method is based on tracking the actual densities of the extrinsic information during each half iteration.

(Refer Slide Time: 28:23)



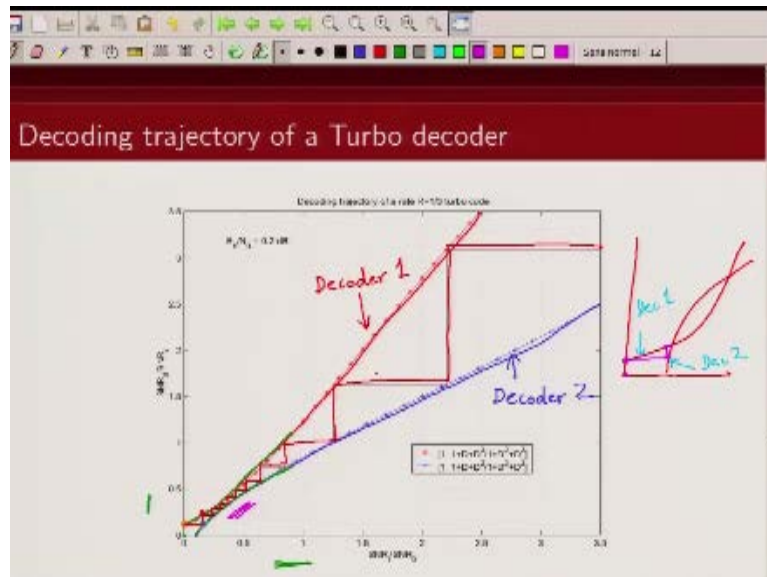
Now what would have happened if this, these curves would have got crossed? So let us look at scenario, let us say I had some curves which are like this, so let us say this is my decoder one and this is my decoder two, then would have happened is so I would have initially started with zero, I have got this, then I got this

(Refer Slide Time: 28:52)



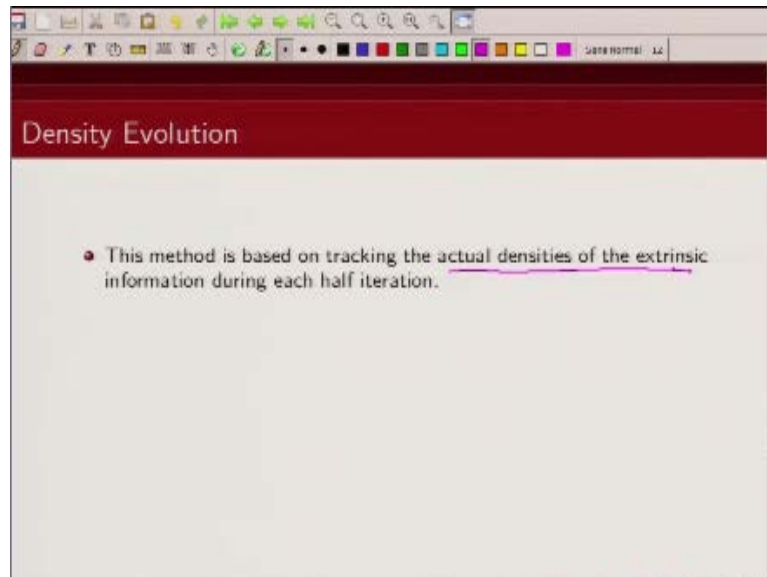
Let me draw slightly better transfer characteristics, so one second, so you draw it, let us say you draw it like this okay, Now let us draw the decoding, so this is transfer characteristics of decoder one, and this is transfer characteristics of decoder 2. So what happens here so you start off with this SNR 0 point you are getting this output SNR from the decoder 1. Now this is input to the decoder 2 so you will get to this point, then from here you will get to this point, then you get to this point and then here you are stuck because these two graphs cross each other. So what you will notice is if there is no tunnel then your decoding algorithm will get stuck and the extrinsic values will not improve.

(Refer Slide Time: 29:51)



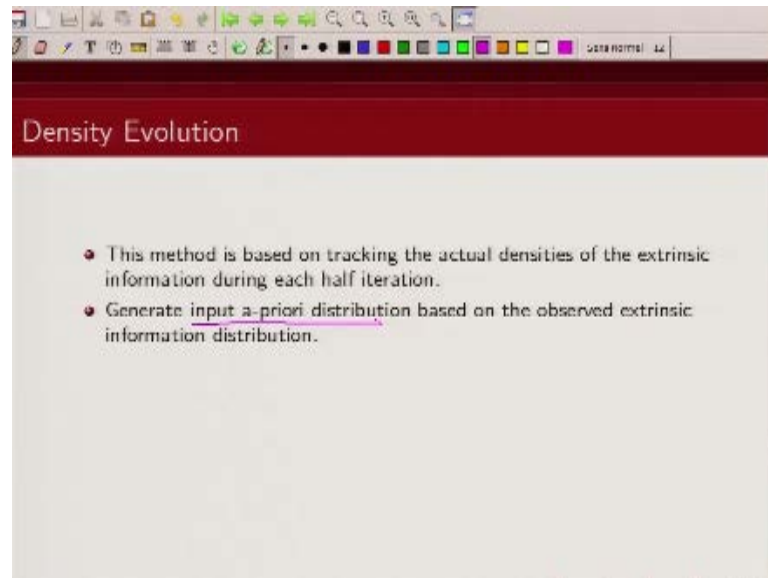
Whereas if there is a tunnel existing like in this particular case you saw that with iterations your extrinsic information is growing, and that is what we would like, so we would like to choose our encoders in such a way such that they match up in a way that there is a tunnel if we plot the decoding trajectories on reverse axis.

(Refer Slide Time: 30:18)



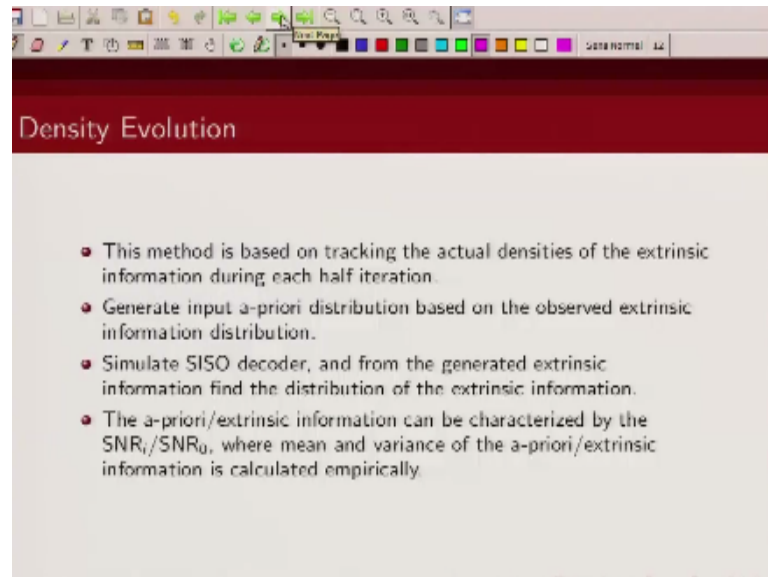
This was the method of El Gamal now the method of Divsalar, they actually use the actual densities of the extrinsic information and they track it

(Refer Slide Time: 30:34)



For finding out how it is growing with iteration. So they generated some input a-priori distribution based on observed extrinsic information.

(Refer Slide Time: 30:43)

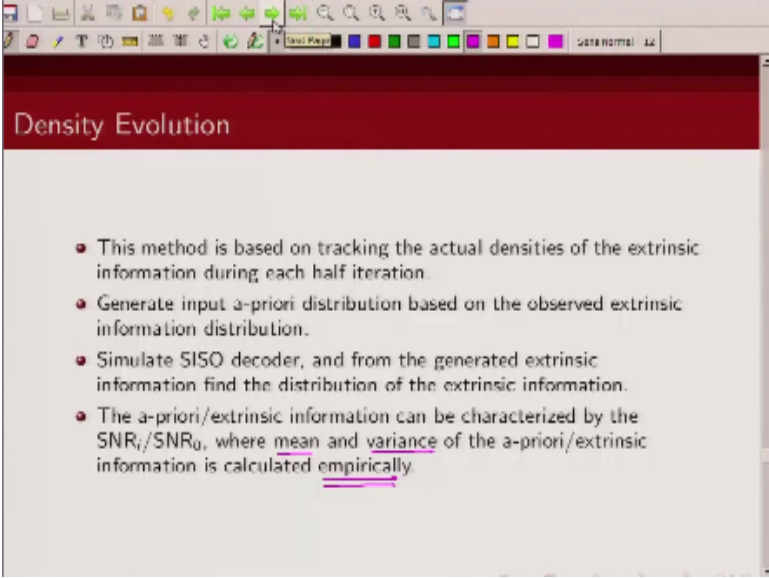


Density Evolution

- This method is based on tracking the actual densities of the extrinsic information during each half iteration.
- Generate input a-priori distribution based on the observed extrinsic information distribution.
- Simulate SISO decoder, and from the generated extrinsic information find the distribution of the extrinsic information.
- The a-priori/extrinsic information can be characterized by the $\text{SNR}_i/\text{SNR}_0$, where mean and variance of the a-priori/extrinsic information is calculated empirically.

And then they simulate the soft input, soft output decoder using this generated distribution of a-priori information, and they find out the distribution of the extrinsic information. And similarly they characterized the SNR of the input distribution as well as the output distribution using

(Refer Slide Time: 31:18)

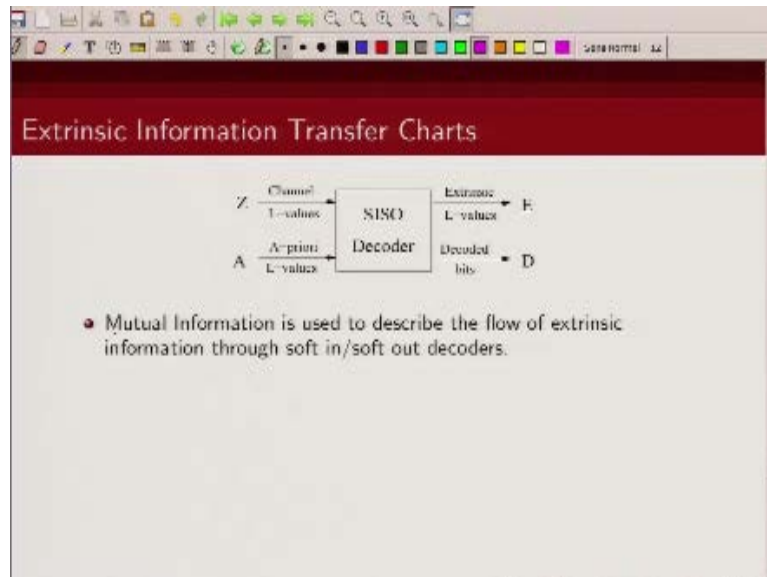


Density Evolution

- This method is based on tracking the actual densities of the extrinsic information during each half iteration.
- Generate input a-priori distribution based on the observed extrinsic information distribution.
- Simulate SISO decoder, and from the generated extrinsic information find the distribution of the extrinsic information.
- The a-priori/extrinsic information can be characterized by the SNR_i/SNR_0 , where mean and variance of the a-priori/extrinsic information is calculated empirically.

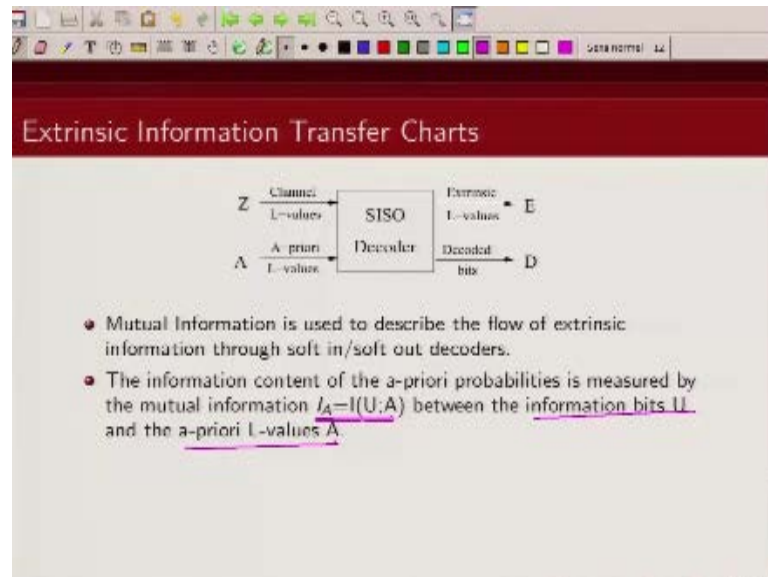
Mean and variance which was empirically computed, so they did not assumed that consistency criteria which El Gamal and others did, they actually used the observed density, they generated a-priori information based on the observe distribution of the extrinsic information.

(Refer Slide Time: 31:43)



The third method which was proposed is based on mutual information. So mutual information was used to describe the flow of information through this soft input, soft output decoder.

(Refer Slide Time: 31:58)



So there were two quantities which were described here basically one was this input mutual information which the mutual information between the information bits and the a-priori value.

(Refer Slide Time: 32:11)

Extrinsic Information Transfer Charts

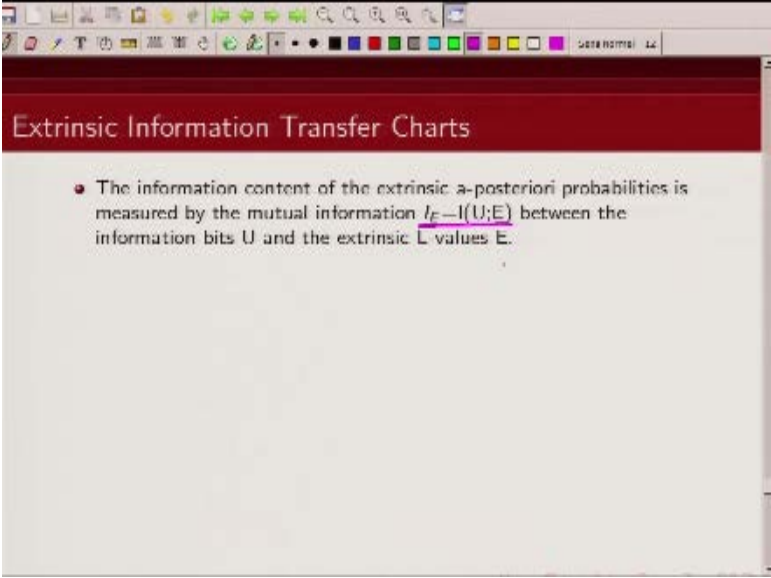
```

graph LR
    Z[Z] -- "Channel L-values" --> SISO[SISO Decoder]
    A[A] -- "A priori L-values" --> SISO
    SISO -- "Extrinsic L-values" --> E[E]
    SISO -- "Decoded bits" --> D[D]
    
```

- Mutual Information is used to describe the flow of extrinsic information through soft in/soft out decoders.
- The information content of the a-priori probabilities is measured by the mutual information $I_A = I(U; A)$ between the information bits U and the a-priori L-values A .
- The input mutual Information $I(U, A)$ is calculated as:

$$I(U; A) \triangleq \frac{1}{2} \sum_{U=-1,1} \int_{-\infty}^{\infty} p_A(\xi|U=u) \log \frac{p_A(\xi|U=u)}{p_A(\xi)} d\xi$$

(Refer Slide Time: 32:13)



The image is a screenshot of a presentation slide. At the top, there is a red header bar with the title "Extrinsic Information Transfer Charts" in white text. Below the header, the slide has a light gray background. A single bullet point is visible, stating: "The information content of the extrinsic a-posteriori probabilities is measured by the mutual information $I_E = I(U;E)$ between the information bits U and the extrinsic L values E." The slide is displayed within a window that has a standard toolbar at the top with various icons for navigation and editing.

Extrinsic Information Transfer Charts

- The information content of the extrinsic a-posteriori probabilities is measured by the mutual information $I_E = I(U;E)$ between the information bits U and the extrinsic L values E.

And the second term which was defined here was the extrinsic mutual information which is a mutual information between the input bits and the extrinsic values.

(Refer Slide Time: 32:44)

Extrinsic Information Transfer Charts

- The information content of the extrinsic a posteriori probabilities is measured by the mutual information $I_E = I(U; E)$ between the information bits U and the extrinsic L-values E .
- The probability distribution of the extrinsic L-values p_E , is computed experimentally from Monte Carlo simulations. p_E is then used to calculate the output mutual information $I(U; E)$.

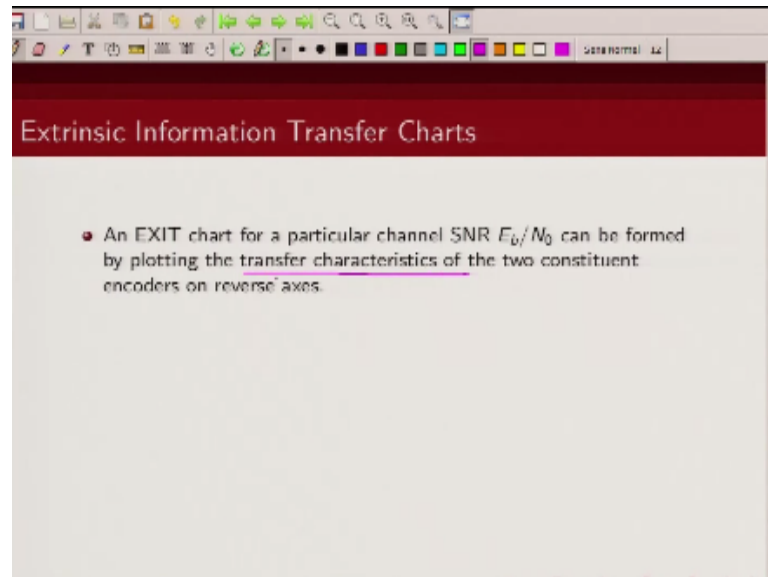
$$I(U; E) \triangleq \frac{1}{2} \sum_{U=-1,1} \int_{-\infty}^{\infty} p_E(\xi|U=u) \log_2 \frac{p_E(\xi|U=u)}{p_E(\xi)} d\xi$$

- Viewing I_E as a function of I_A and E_b/N_0 , the extrinsic information transfer characteristic of an encoder is defined as

$$I_E = T(I_A, E_b/N_0).$$

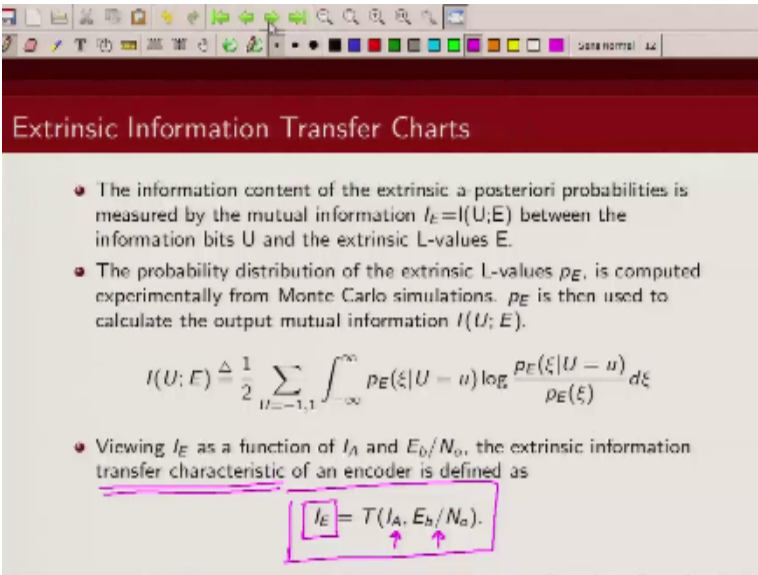
So what was done in this technique was you can view the mutual information corresponding to the input and extrinsic value as a function of mutual information of a-priori values and information bits, and operating signal to noise ratio. So this was the transfer function which was considered in this extrinsic information chart that viewing the output mutual information between the extrinsic information and the information bit as a function of mutual information between the a-priori and the information bits and signal to noise ratio.

(Refer Slide Time: 33:11)



So how was exit chart created, so they plotted this transfer characteristics which was given by

(Refer Slide Time: 33:21)



The slide is titled "Extrinsic Information Transfer Charts" in a red header. It contains two bullet points, a mathematical formula, and a definition of the extrinsic transfer characteristic. The formula for $I(U; E)$ is given as $\frac{1}{2} \sum_{U=-1,1} \int_{-\infty}^{\infty} p_E(\xi|U=u) \log_2 \frac{p_E(\xi|U=u)}{p_E(\xi)} d\xi$. The definition states that I_E is a function of I_A and E_b/N_0 , and the equation $I_E = T(I_A, E_b/N_0)$ is boxed with arrows pointing to I_A and E_b/N_0 .

Extrinsic Information Transfer Charts

- The information content of the extrinsic a posteriori probabilities is measured by the mutual information $I_E = I(U; E)$ between the information bits U and the extrinsic L-values E .
- The probability distribution of the extrinsic L-values p_E , is computed experimentally from Monte Carlo simulations. p_E is then used to calculate the output mutual information $I(U; E)$.

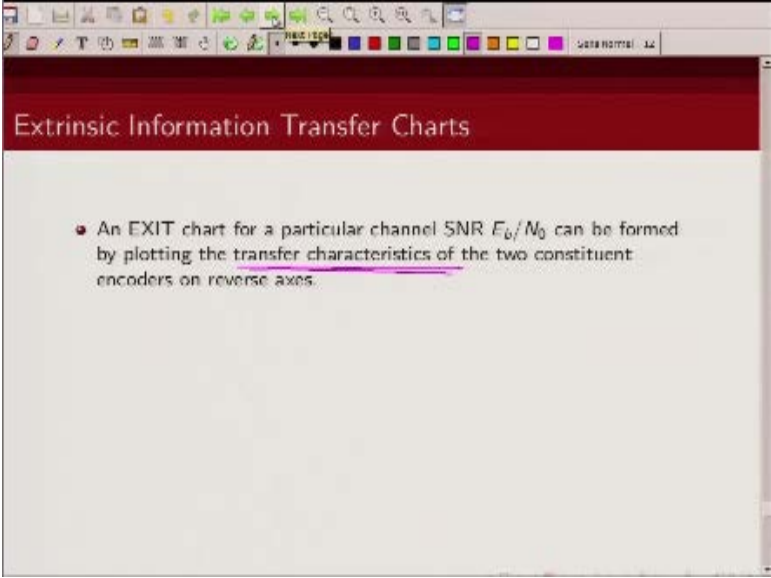
$$I(U; E) \triangleq \frac{1}{2} \sum_{U=-1,1} \int_{-\infty}^{\infty} p_E(\xi|U=u) \log_2 \frac{p_E(\xi|U=u)}{p_E(\xi)} d\xi$$

- Viewing I_E as a function of I_A and E_b/N_0 , the extrinsic information transfer characteristic of an encoder is defined as

$I_E = T(I_A, E_b/N_0).$

This, they plotted this transfer characteristics

(Refer Slide Time: 33:28)

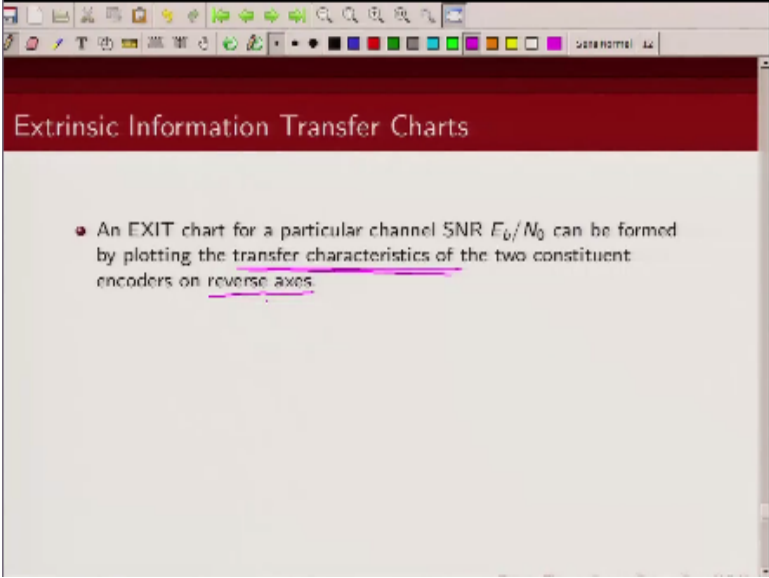


Extrinsic Information Transfer Charts

- An EXIT chart for a particular channel SNR E_b/N_0 can be formed by plotting the transfer characteristics of the two constituent encoders on reverse axes.

For two constituent decoders on reverse axis's

(Refer Slide Time: 33:34)

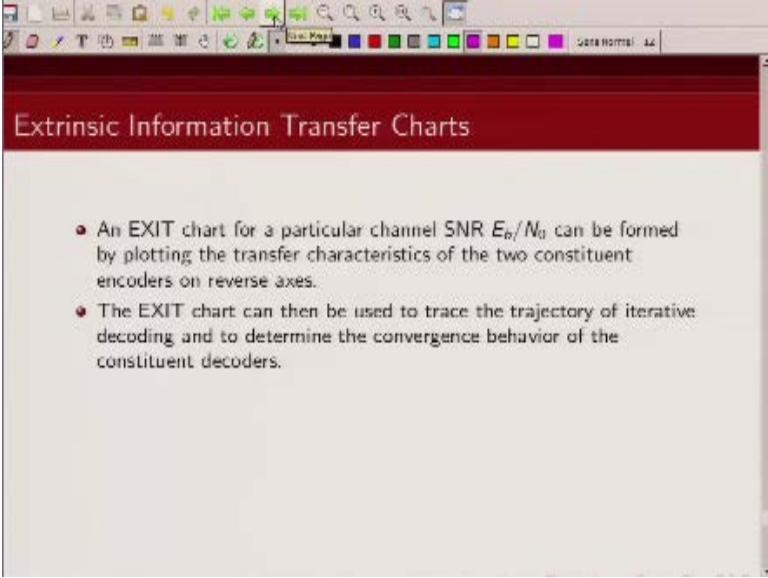


Extrinsic Information Transfer Charts

- An EXIT chart for a particular channel SNR E_b/N_0 can be formed by plotting the transfer characteristics of the two constituent encoders on reverse axes.

Similar to El Gamal's technique

(Refer Slide Time: 33:35)



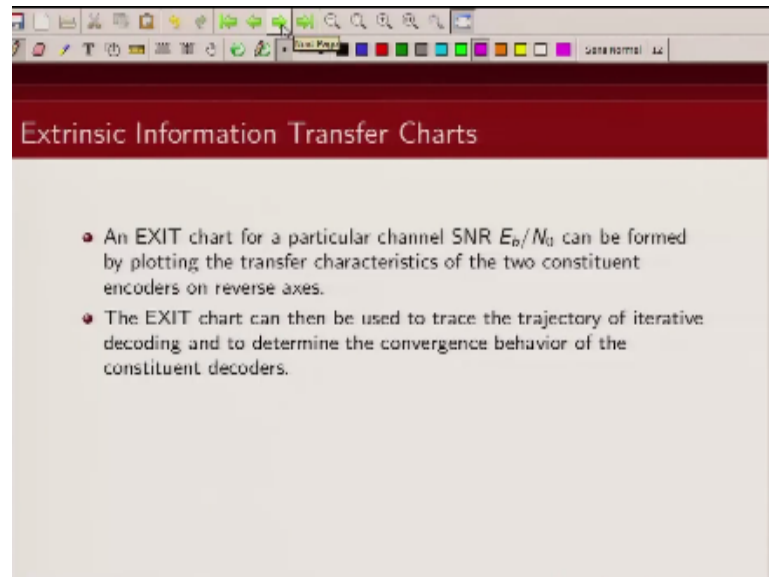
The image is a screenshot of a presentation slide. At the top, there is a red header bar with the title "Extrinsic Information Transfer Charts" in white text. Below the header, the slide has a light beige background. It contains two bullet points, each preceded by a red circular icon. The first bullet point discusses forming an EXIT chart for a specific channel SNR by plotting transfer characteristics on reverse axes. The second bullet point explains the use of the EXIT chart to trace iterative decoding trajectories and assess convergence. The slide is displayed within a window that has a standard toolbar at the top and a status bar at the bottom.

Extrinsic Information Transfer Charts

- An EXIT chart for a particular channel SNR E_b/N_0 can be formed by plotting the transfer characteristics of the two constituent encoders on reverse axes.
- The EXIT chart can then be used to trace the trajectory of iterative decoding and to determine the convergence behavior of the constituent decoders.

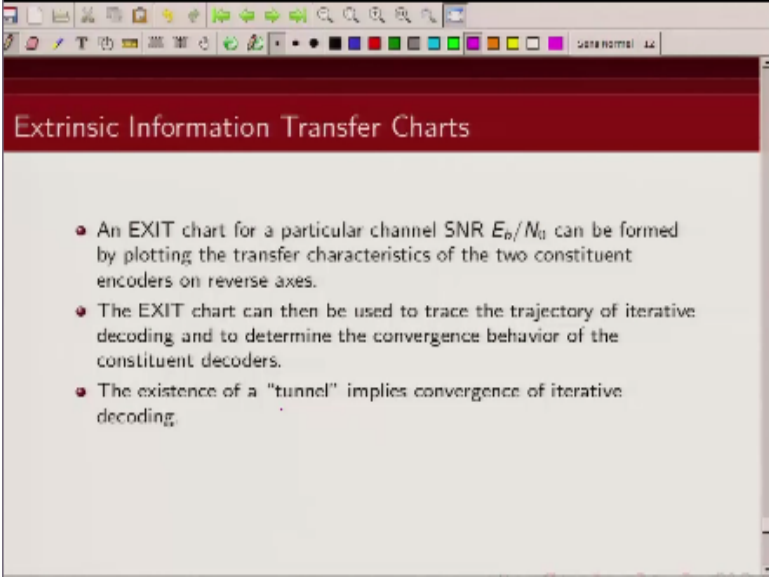
The difference is El Gamal used mean as SNR, here they use mutual information.

(Refer Slide Time: 33:45)



So very similar idea so, so these transfer functions were plotted on reverse axis's, initially you do not have any a-prior knowledge so then input a-priori mutual information is 0 and then after 1 of iteration you get some extrinsic information so you have some positive mutual information, and then you pass that as a input to second decoder and the decoding will progress if there is a tunnel otherwise it will get stuck.

(Refer Slide Time: 34:20)

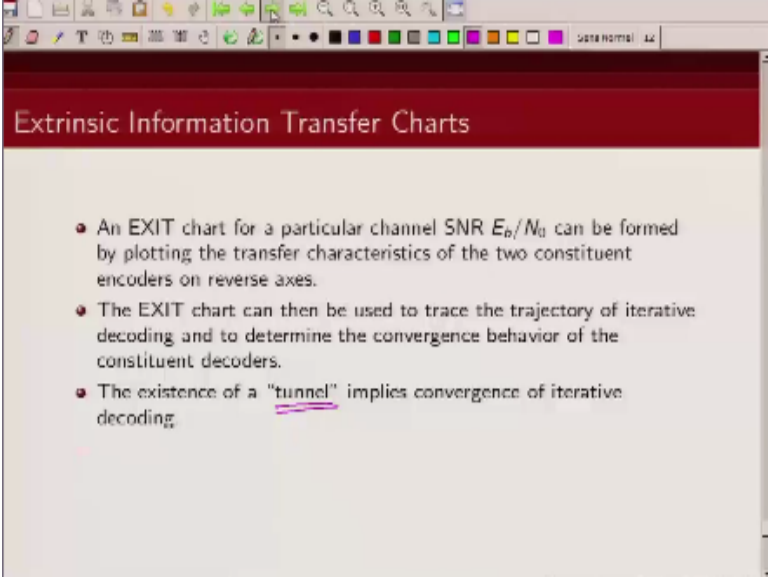


Extrinsic Information Transfer Charts

- An EXIT chart for a particular channel SNR E_b/N_0 can be formed by plotting the transfer characteristics of the two constituent encoders on reverse axes.
- The EXIT chart can then be used to trace the trajectory of iterative decoding and to determine the convergence behavior of the constituent decoders.
- The existence of a "tunnel" implies convergence of iterative decoding.

So as I said the, whether the decoding algorithm will converge or not is, can be viewed by plotting these transfer characteristics on reverse axis's, and seeing whether a tunnel exist between them or not.

(Refer Slide Time: 34:43)

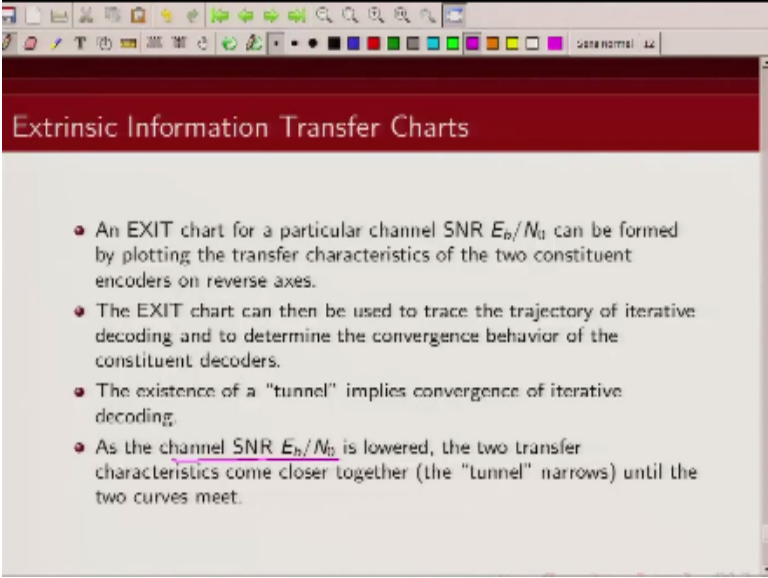


The image shows a presentation slide with a dark red header bar containing the title "Extrinsic Information Transfer Charts". Below the header, the slide background is light beige. It features three bullet points, each preceded by a red circular icon. The text is in a black serif font. The first bullet point discusses forming an EXIT chart for a specific channel SNR. The second bullet point describes how the EXIT chart is used to trace the trajectory of iterative decoding. The third bullet point states that the existence of a "tunnel" (underlined in pink) implies convergence. The slide is displayed within a window that has a standard toolbar at the top and a status bar at the bottom right showing "Slide Number: 44".

Extrinsic Information Transfer Charts

- An EXIT chart for a particular channel SNR E_b/N_0 can be formed by plotting the transfer characteristics of the two constituent encoders on reverse axes.
- The EXIT chart can then be used to trace the trajectory of iterative decoding and to determine the convergence behavior of the constituent decoders.
- The existence of a "tunnel" implies convergence of iterative decoding.

(Refer Slide Time: 34:43)

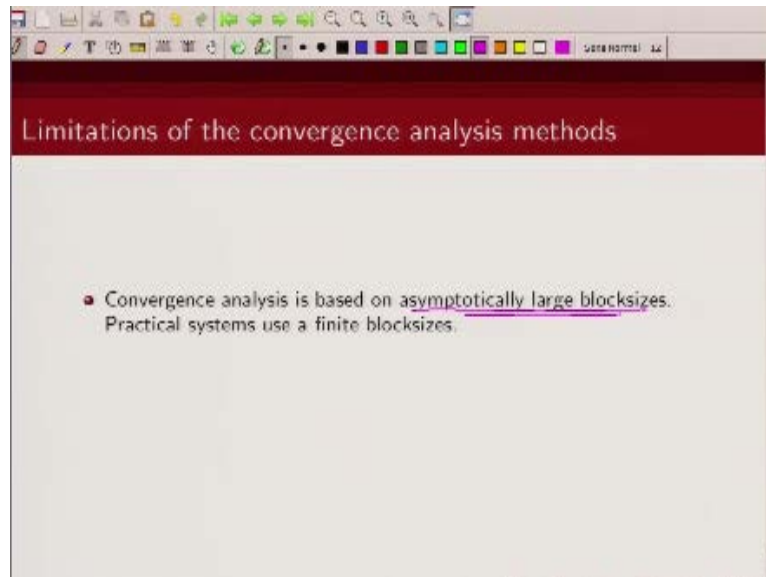


Extrinsic Information Transfer Charts

- An EXIT chart for a particular channel SNR E_b/N_0 can be formed by plotting the transfer characteristics of the two constituent encoders on reverse axes.
- The EXIT chart can then be used to trace the trajectory of iterative decoding and to determine the convergence behavior of the constituent decoders.
- The existence of a "tunnel" implies convergence of iterative decoding.
- As the channel SNR E_b/N_0 is lowered, the two transfer characteristics come closer together (the "tunnel" narrows) until the two curves meet.

Now what happens if we reduce the channel operating SNR, if we reduce channel operating SNR then these curves come closer until a point will come when they will barely touch or they will just touch or cross each other, so the point minimum SNR where they are still there is a tunnel that is your threshold, decoding threshold. So we have specified various methods for tracking the mutual information, tracking the extrinsic information and a-priori information and this can be used to see how our constituent encoders will behave how the turbo code, turbo decoder will behave under iterative decoding algorithm. Now what are the limitations of this analysis approach?

(Refer Slide Time: 35:49)



Now this approach assumes that we have very large block sizes, so this convergence analysis results hold for very large block sizes but in practical systems we use small size block sizes, so the thresholds predicted by this method may not be consistent when we use small block sizes, and of course there are some assumptions for example in El Gamal's technique we use Gaussian assumptions, we made assumption of consistency condition those conditions may or may not hold, okay. So with this I will conclude this discussion on convergence analysis of turbo codes, thank you.

Acknowledgement

Ministry of Human Resource & Development

Prof. Satyaki Roy

Co-ordinator, NPTEL IIT Kanpur

NPTEL Team

Sanjay Pal

Ashish Singh
Badal Pradhan
Tapobrata Das
Ram Chandra
Dilip Tripathi
Manoj Shrivastava
Padam Shukla
Sanjay Mishra
Shubham Rawat
Shikha Gupta
K. K. Mishra
Aradhana Singh
Sweta
Ashutosh Gairola
Dilip Katiyar
Sharwan
Hari Ram
Bhadra Rao
Puneet Kumar Bajpai
Lalty Dutta
Ajay Kanaujia
Shivendra Kumar Tiwari

an IIT Kanpur Production

©copyright reserved