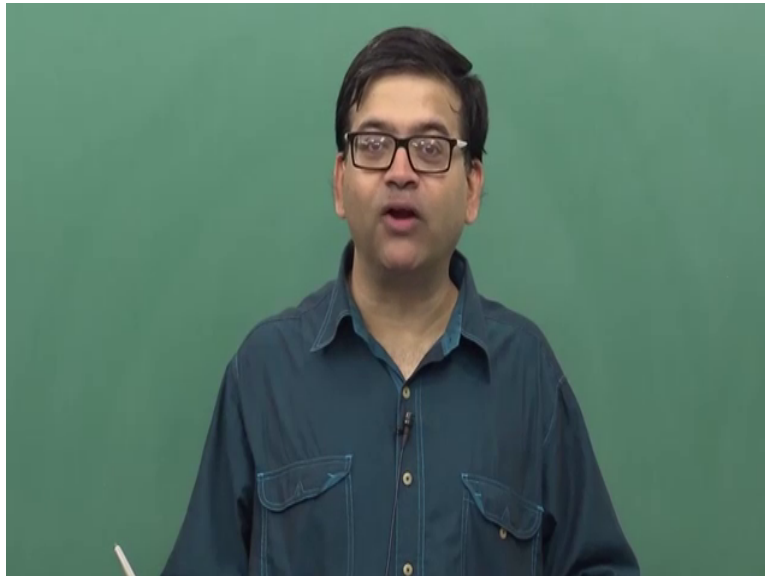


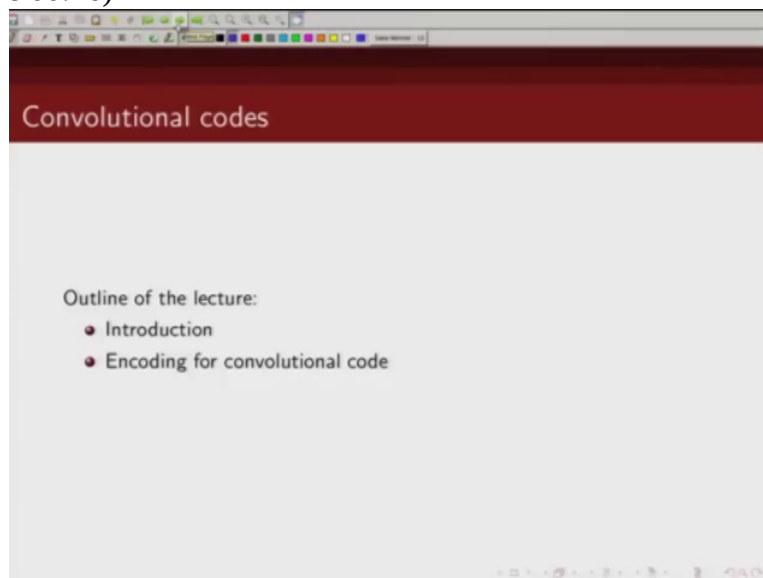
**An Introduction to Coding Theory**  
**Professor Adrish Banerji**  
**Department of Electrical Engineering**  
**Indian Institute of Technology, Kanpur**  
**Module 04**  
**Lecture Number 15**  
**Introduction to convolutional codes-I: Encoding**

(Refer Slide Time 00:18)



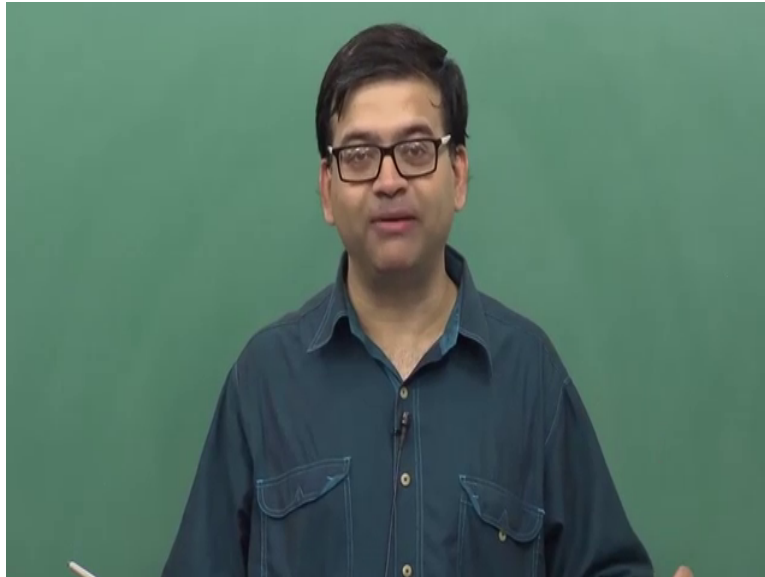
So we will start with introduction of convolutional code and today we are going to discuss how we can encode an information sequence using convolutional code. So

(Refer Slide Time 00:26)



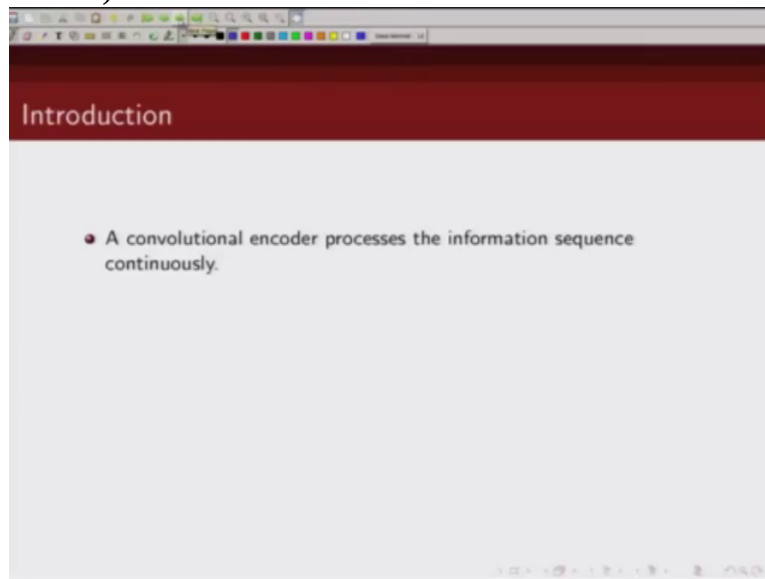
today's topic of discussion is encoding of convolutional code and we will take a very simple

(Refer Slide Time 00:31)



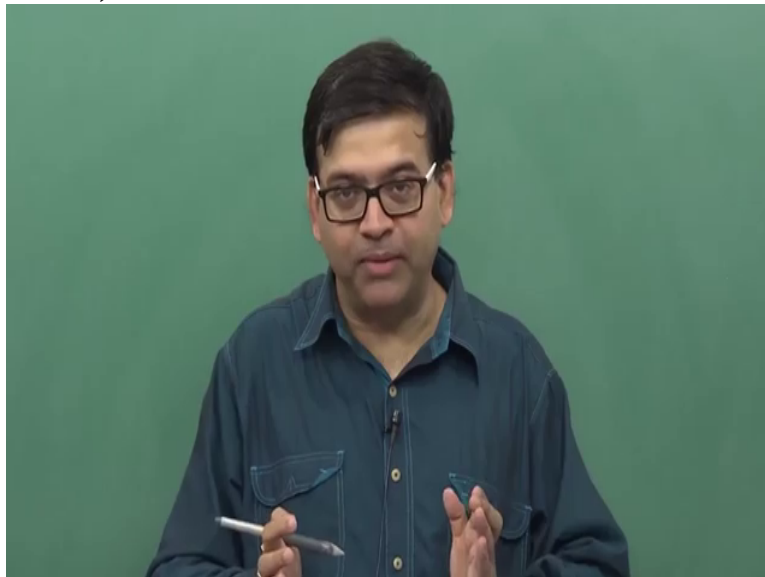
example of a rate  $1/n$  convolutional code As you know

(Refer Slide Time 00:38)



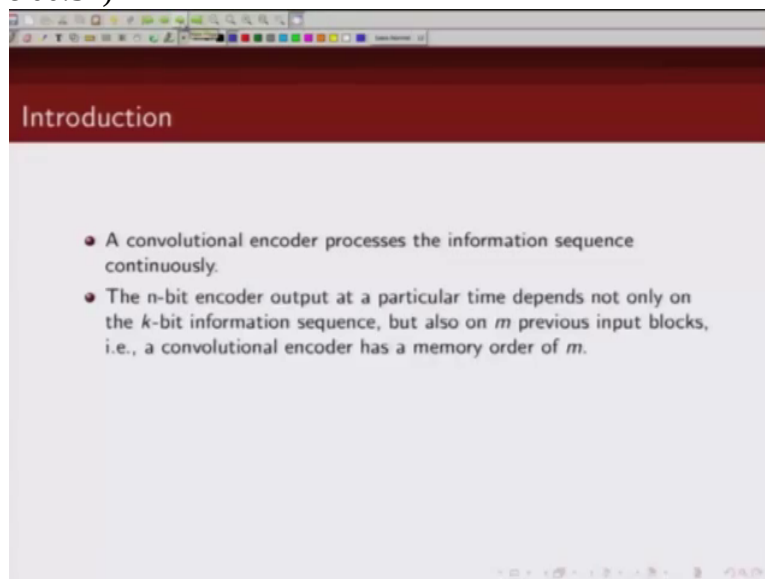
a convolutional code processes information sequence in a continuous fashion. So information bits come in and

(Refer Slide Time 00:46)



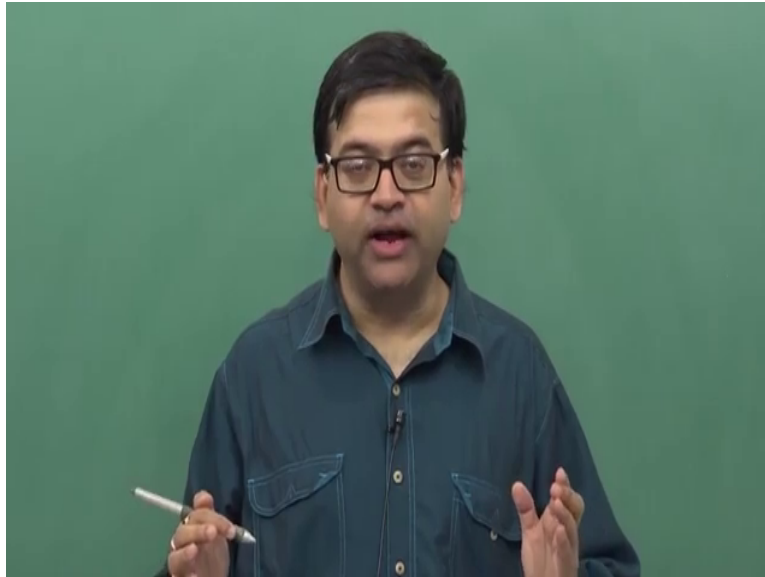
we can get continuous output from a convolutional encoder.

(Refer Slide Time 00:51)



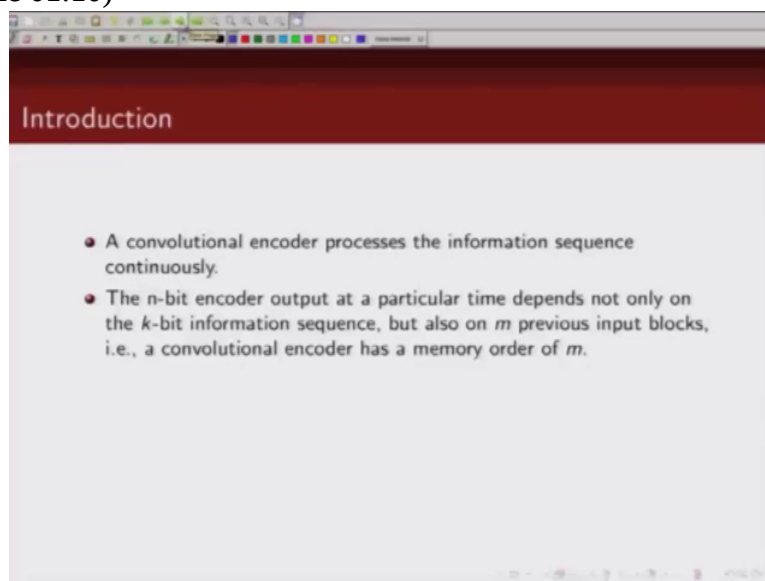
We also know that the output of convolutional encoder depends

(Refer Slide Time 00:58)



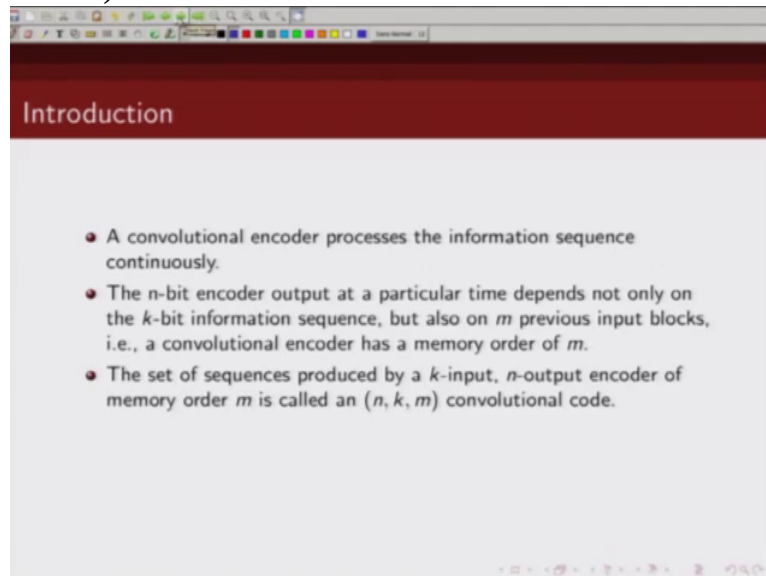
not only on the current input but it also depends on the past inputs and past outputs depending upon the memory of the convolutional encoder. And as we have seen

(Refer Slide Time 01:10)



we can realize the convolutional code using shift registers.

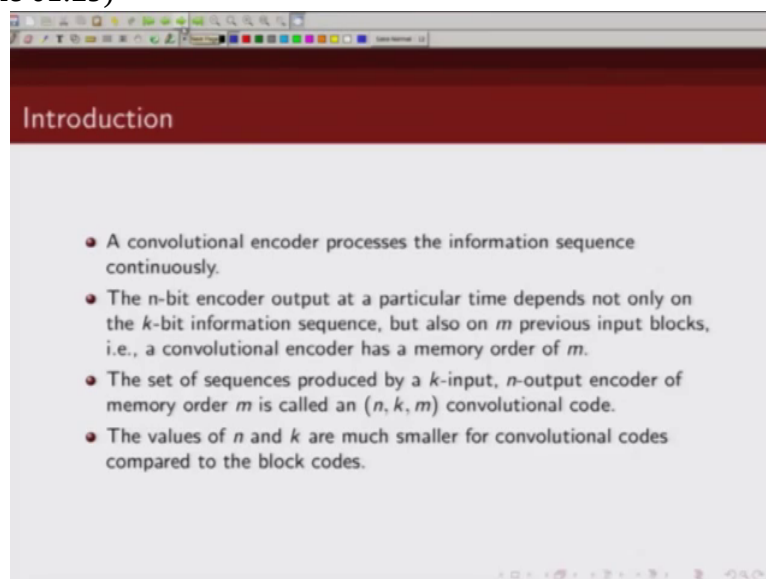
(Refer Slide Time 01:16)



So we describe a convolutional code basically as an  $n, k$  convolutional code with memory  $m$ .

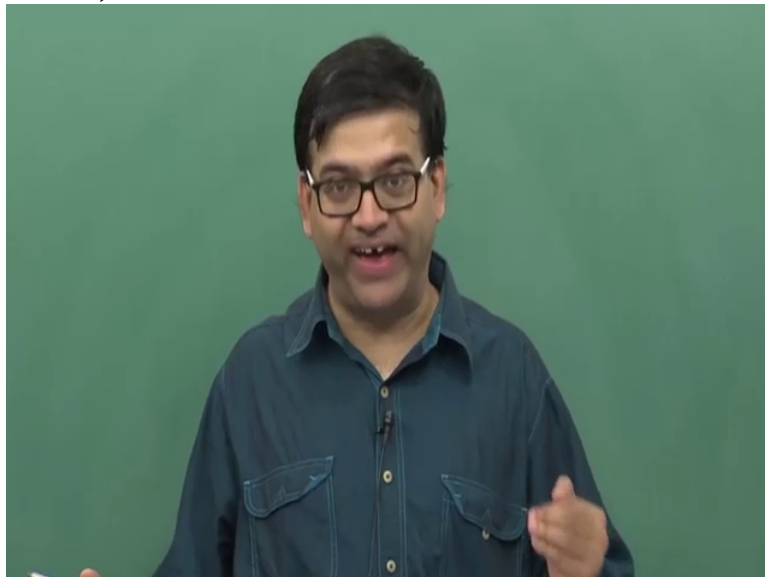
Now as we

(Refer Slide Time 01:29)



have said before, as opposed to the block codes, typically the value of  $n$

(Refer Slide Time 01:35)



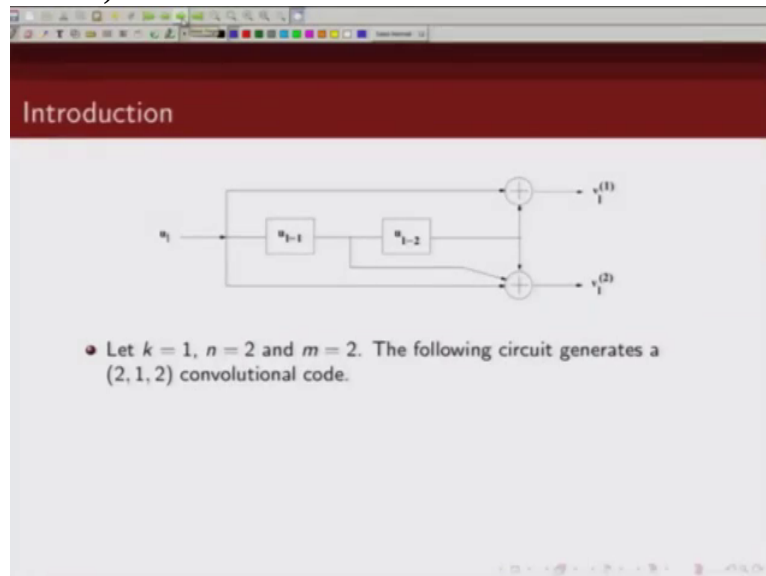
and  $k$  for a convolutional code is much smaller, like  $k$  may be 1, 2, 3 and similarly  $n$  will be, may be 2, 3, 4 like that.

(Refer Slide Time 01:44)

A presentation slide with a dark red header containing the word "Introduction" in white. The main content area is white and contains four bullet points. The slide is displayed within a window frame with a standard operating system taskbar at the top.

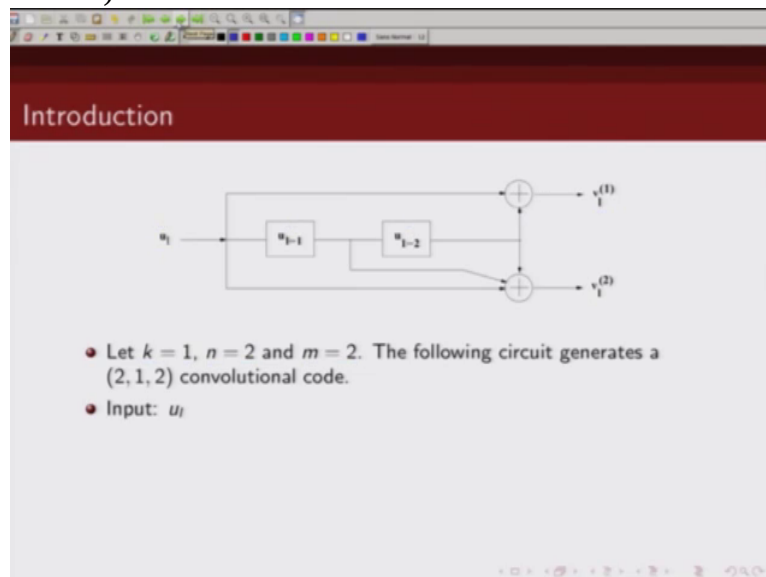
- A convolutional encoder processes the information sequence continuously.
- The  $n$ -bit encoder output at a particular time depends not only on the  $k$ -bit information sequence, but also on  $m$  previous input blocks, i.e., a convolutional encoder has a memory order of  $m$ .
- The set of sequences produced by a  $k$ -input,  $n$ -output encoder of memory order  $m$  is called an  $(n, k, m)$  convolutional code.
- The values of  $n$  and  $k$  are much smaller for convolutional codes compared to the block codes.

(Refer Slide Time 01:45)



So this is one example of a memory 2 convolutional encoder. We can see this is our input,  $u$  of  $l$  and output  $v$  of  $l$  1, and  $v$  of  $l$  2. Now note that the output  $v$  of  $l$  1 and  $v$  of  $l$  2 depends not only on the current input but also depends on the past inputs as indicated by content here and content here. And this is  $k$  equal to 1 because there is only one  $l$  input here. That's why  $k$  is 1. There are 2 outputs this one and this one, so that's why  $n$  is 2 and since the output depends on 2 memory elements, this and this,  $m$  is 2. So this is a 2 1 2 convolutional code. So as we

(Refer Slide Time 02:47)



said the input is  $u$  of  $l$  and output is  $v$  of  $l$  1 and

(Refer Slide Time 02:52)

**Introduction**

Let  $k = 1$ ,  $n = 2$  and  $m = 2$ . The following circuit generates a  $(2, 1, 2)$  convolutional code.

Input:  $u_i$

Outputs:

$$v_i^{(1)} = u_i + u_{i-2}$$

$$v_i^{(2)} = u_i + u_{i-1} + u_{i-2}$$

$v_i^{(1)}$  and how is  $v_i^{(1)}$  and  $v_i^{(2)}$  depend on the  $u_i$  and the past values? This is given by the interconnection. So we can see for  $v_i^{(1)}$  it depends on input  $u_i$  as given by this and it depends on  $u_{i-2}$  as given by this link. So  $v_i^{(1)}$  is given by

(Refer Slide Time 03:17)

**Introduction**

Let  $k = 1$ ,  $n = 2$  and  $m = 2$ . The following circuit generates a  $(2, 1, 2)$  convolutional code.

Input:  $u_i$

Outputs:

$$v_i^{(1)} = u_i + u_{i-2}$$

$$v_i^{(2)} = u_i + u_{i-1} + u_{i-2}$$

$u_i$  and  $u_{i-2}$ . So it depends on the current input and the input which was there two time instances earlier. Similarly  $v_i^{(2)}$  depends on  $u_i$  as given by this interconnection,  $u_{i-1}$  as given by this interconnection and  $u_{i-2}$  as given by this interconnection.



(Refer Slide Time 03:42)

Introduction

Let  $k = 1$ ,  $n = 2$  and  $m = 2$ . The following circuit generates a  $(2, 1, 2)$  convolutional code.

Input:  $u_i$

Outputs:

$$\begin{aligned} v_i^{(1)} &= u_i + u_{i-2} \quad \checkmark \\ v_i^{(2)} &= u_i + u_{i-1} + u_{i-2} \end{aligned}$$

So this is our  $v_i$ . So these are two outputs and this is how they are related to the input. So we can say that whether a particular input appears in the output, that is basically given by these interconnections. These interconnections tell us whether that particular bit is taking part in the output or not. So if

(Refer Slide Time 04:11)

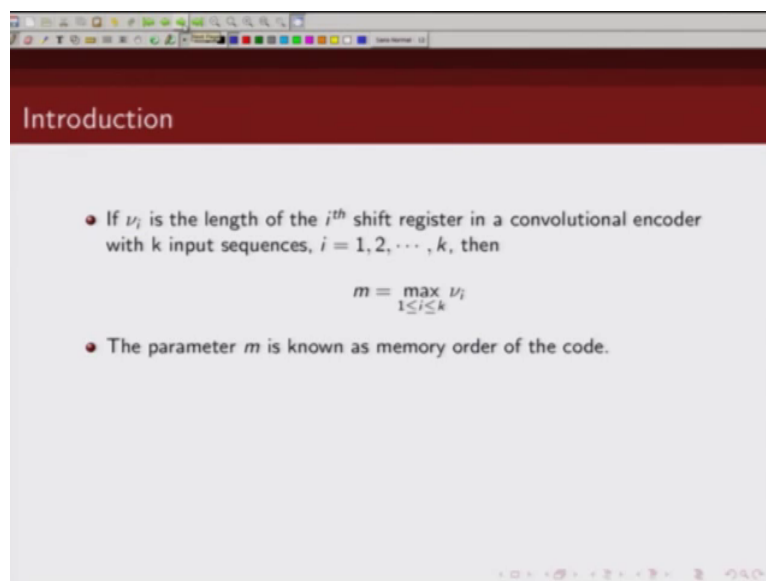
Introduction

If  $\nu_i$  is the length of the  $i^{\text{th}}$  shift register in a convolutional encoder with  $k$  input sequences,  $i = 1, 2, \dots, k$ , then

$$m = \max_{1 \leq i \leq k} \nu_i$$

we denote the  $\mu_i$  the length of the  $i^{\text{th}}$  shift register in the convolutional encoder then we define the memory order as the maximum of, maximum length of the shift register among the  $k$  shift registers used to represent the convolutional encoder.

(Refer Slide Time 04:33)



Introduction

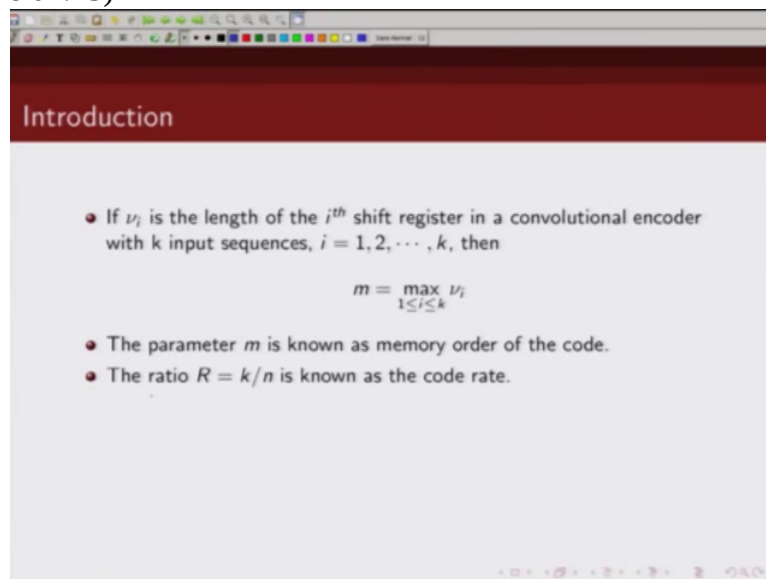
- If  $\nu_i$  is the length of the  $i^{\text{th}}$  shift register in a convolutional encoder with  $k$  input sequences,  $i = 1, 2, \dots, k$ , then

$$m = \max_{1 \leq i \leq k} \nu_i$$

- The parameter  $m$  is known as memory order of the code.

And this parameter  $m$  is also known as memory order of the convolutional code. As we know this ratio

(Refer Slide Time 04:45)



Introduction

- If  $\nu_i$  is the length of the  $i^{\text{th}}$  shift register in a convolutional encoder with  $k$  input sequences,  $i = 1, 2, \dots, k$ , then

$$m = \max_{1 \leq i \leq k} \nu_i$$

- The parameter  $m$  is known as memory order of the code.
- The ratio  $R = k/n$  is known as the code rate.

of information bits to coded bits  $k$  by  $n$  is known as code rate which we denote by capital  $R$ ,

(Refer Slide Time 04:57)

Introduction

- If  $\nu_i$  is the length of the  $i^{\text{th}}$  shift register in a convolutional encoder with  $k$  input sequences,  $i = 1, 2, \dots, k$ , then
$$m = \max_{1 \leq i \leq k} \nu_i$$
- The parameter  $m$  is known as memory order of the code.
- The ratio  $R = k/n$  is known as the code rate.
- The overall constraint length  $\nu$  of the encoder is defined as
$$\nu = \sum_{1 \leq i \leq k} \nu_i$$

and the overall constraint length is defined as sum of length of all the  $k$  shift registers. That's the overall constraint length.

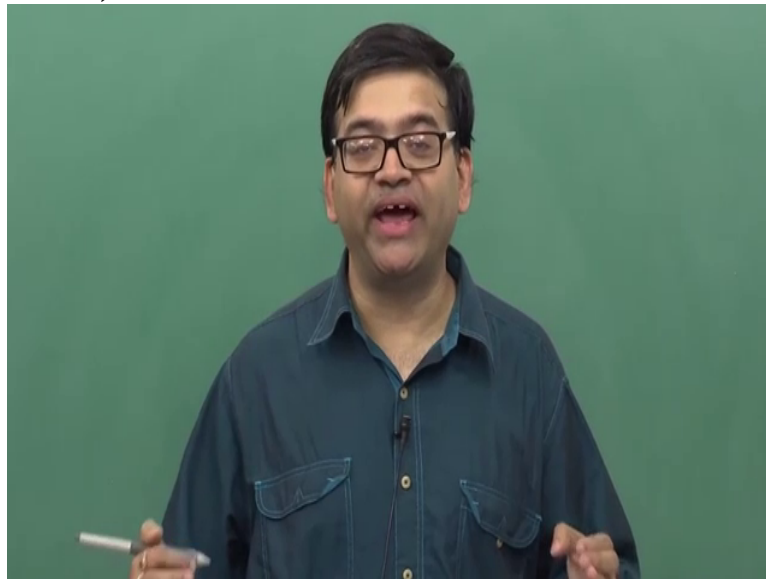
(Refer Slide Time 05:10)

Encoding of  $(n, 1, m)$  convolutional code

- In this case, a single information sequence
$$\mathbf{u} = (u_0, u_1, \dots, u_i, \dots)$$
is encoded into  $n$  output sequences.
$$\begin{aligned} \mathbf{v}^{(1)} &= (v_0^{(1)}, v_1^{(1)}, \dots, v_i^{(1)}, \dots) \\ \mathbf{v}^{(2)} &= (v_0^{(2)}, v_1^{(2)}, \dots, v_i^{(2)}, \dots) \\ &\vdots \\ \mathbf{v}^{(n)} &= (v_0^{(n)}, v_1^{(n)}, \dots, v_i^{(n)}, \dots) \end{aligned}$$

Now we are going to show how we are going to encode

(Refer Slide Time 05:17)



a convolutional or information sequence using a rate 1 by n convolutional encoder. Since it is a rate

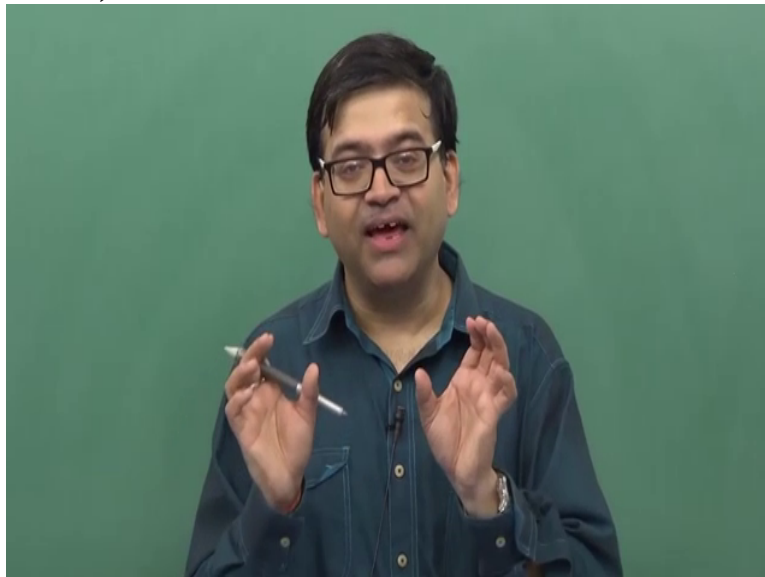
(Refer Slide Time 05:25)

Encoding of  $(n, 1, m)$  convolutional code

- In this case, a single information sequence  $\mathbf{u} = (u_0, u_1, \dots, u_i, \dots)$  is encoded into  $n$  output sequences.  
 $\mathbf{v}^{(1)} = (v_0^{(1)}, v_1^{(1)}, \dots, v_i^{(1)}, \dots)$   
 $\mathbf{v}^{(2)} = (v_0^{(2)}, v_1^{(2)}, \dots, v_i^{(2)}, \dots)$   
 $\vdots$   
 $\mathbf{v}^{(n)} = (v_0^{(n)}, v_1^{(n)}, \dots, v_i^{(n)}, \dots)$

1 by n convolutional code so k is 1 and number of coded bits is n. So there is 1 input coming in and there are n outputs and the maximum length of 1 shift register used to represent this rate 1 by n code is m. So this shift register has m

(Refer Slide Time 05:47)



memory elements. So let us take our

(Refer Slide Time 05:51)

Encoding of  $(n, 1, m)$  convolutional code

- In this case, a single information sequence

$$\mathbf{u} = (u_0, u_1, \dots, u_l, \dots)$$

is encoded into  $n$  output sequences.

$$\begin{aligned} \mathbf{v}^{(1)} &= (v_0^{(1)}, v_1^{(1)}, \dots, v_l^{(1)}, \dots) \\ \mathbf{v}^{(2)} &= (v_0^{(2)}, v_1^{(2)}, \dots, v_l^{(2)}, \dots) \\ &\vdots \\ \mathbf{v}^{(n)} &= (v_0^{(n)}, v_1^{(n)}, \dots, v_l^{(n)}, \dots) \end{aligned}$$

input which we denote by  $\mathbf{u}$  to be  $u_0, u_1, u_2, \dots, u_{l-1}$ . Since it is a rate  $1/n$  convolutional code so what we would get is corresponding to 1 input we are going to get  $n$  outputs and we denote these  $n$  outputs by  $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \mathbf{v}^{(3)}, \dots, \mathbf{v}^{(n)}$  where each of these  $\mathbf{v}^{(i)}$ 's can be written like this. So the

(Refer Slide Time 06:22)

Encoding of  $(n, 1, m)$  convolutional code

- In this case, a single information sequence  $\mathbf{u} = (u_0, u_1, \dots, u_i, \dots)$  is encoded into  $n$  output sequences.  
 $\mathbf{v}^{(1)} = (v_0^{(1)}, v_1^{(1)}, \dots, v_i^{(1)}, \dots)$   
 $\mathbf{v}^{(2)} = (v_0^{(2)}, v_1^{(2)}, \dots, v_i^{(2)}, \dots)$   
 $\vdots$   
 $\mathbf{v}^{(n)} = (v_0^{(n)}, v_1^{(n)}, \dots, v_i^{(n)}, \dots)$
- The  $n$  output sequences are interleaved to form a single code sequence.  
 $\mathbf{v} = (\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_i, \dots)$ , where  $\mathbf{v}_i = (v_i^{(1)}, v_i^{(2)}, \dots, v_i^{(n)})$

output at a particular instance then is, so corresponding to  $u_0$  then what's the output? These are the  $n$  bits

(Refer Slide Time 06:33)

Encoding of  $(n, 1, m)$  convolutional code

- In this case, a single information sequence  $\mathbf{u} = (u_0, u_1, \dots, u_i, \dots)$  is encoded into  $n$  output sequences.  
 $\mathbf{v}^{(1)} = (v_0^{(1)}, v_1^{(1)}, \dots, v_i^{(1)}, \dots)$   
 $\mathbf{v}^{(2)} = (v_0^{(2)}, v_1^{(2)}, \dots, v_i^{(2)}, \dots)$   
 $\vdots$   
 $\mathbf{v}^{(n)} = (v_0^{(n)}, v_1^{(n)}, \dots, v_i^{(n)}, \dots)$
- The  $n$  output sequences are interleaved to form a single code sequence.  
 $\mathbf{v} = (\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_i, \dots)$ , where  $\mathbf{v}_i = (v_i^{(1)}, v_i^{(2)}, \dots, v_i^{(n)})$

output corresponding to this input  $u_0$ . Similarly corresponding to  $u_1$ , my output is this.

(Refer Slide Time 06:44)

Encoding of  $(n, 1, m)$  convolutional code

- In this case, a single information sequence
 
$$\mathbf{u} = (u_0, u_1, \dots, u_i, \dots)$$
 is encoded into  $n$  output sequences,
 
$$\mathbf{v}^{(1)} = (v_0^{(1)}, v_1^{(1)}, \dots, v_i^{(1)}, \dots)$$

$$\mathbf{v}^{(2)} = (v_0^{(2)}, v_1^{(2)}, \dots, v_i^{(2)}, \dots)$$

$$\vdots$$

$$\mathbf{v}^{(n)} = (v_0^{(n)}, v_1^{(n)}, \dots, v_i^{(n)}, \dots)$$
- The  $n$  output sequences are interleaved to form a single code sequence.
 
$$\mathbf{v} = (\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_i, \dots), \text{ where } \mathbf{v}_i = (v_i^{(1)}, v_i^{(2)}, \dots, v_i^{(n)})$$

Corresponding to  $u_i$  my output is this  $n$  bit output

(Refer Slide Time 06:50)

Encoding of  $(n, 1, m)$  convolutional code

- In this case, a single information sequence
 
$$\mathbf{u} = (u_0, u_1, \dots, u_i, \dots)$$
 is encoded into  $n$  output sequences,
 
$$\mathbf{v}^{(1)} = (v_0^{(1)}, v_1^{(1)}, \dots, v_i^{(1)}, \dots)$$

$$\mathbf{v}^{(2)} = (v_0^{(2)}, v_1^{(2)}, \dots, v_i^{(2)}, \dots)$$

$$\vdots$$

$$\mathbf{v}^{(n)} = (v_0^{(n)}, v_1^{(n)}, \dots, v_i^{(n)}, \dots)$$
- The  $n$  output sequences are interleaved to form a single code sequence.
 
$$\mathbf{v} = (\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_i, \dots), \text{ where } \mathbf{v}_i = (v_i^{(1)}, v_i^{(2)}, \dots, v_i^{(n)})$$

Ok. So I write the output by  $v_0, v_1, v_2, v_l$  where this  $v_l$  is an  $n$  bit vector. Now

(Refer Slide Time 07:04)

Encoding of  $(n, 1, m)$  convolutional code

- The code is specified by a set of  $n$  generator sequences of length  $m + 1$ .

$$\mathbf{g}^{(1)} = (g_0^{(1)}, g_1^{(1)}, \dots, g_m^{(1)})$$
$$\mathbf{g}^{(2)} = (g_0^{(2)}, g_1^{(2)}, \dots, g_m^{(2)})$$
$$\vdots$$
$$\mathbf{g}^{(n)} = (g_0^{(n)}, g_1^{(n)}, \dots, g_m^{(n)})$$

how do we generate these  $n$  bit vector from this 1 input and if we just go back to our example

(Refer Slide Time 07:14)

Encoding of  $(n, 1, m)$  convolutional code

- In this case, a single information sequence

$$\mathbf{u} = (u_0, u_1, \dots, u_l, \dots)$$

is encoded into  $n$  output sequences.

$$\mathbf{v}^{(1)} = (v_0^{(1)}, v_1^{(1)}, \dots, v_l^{(1)}, \dots)$$
$$\mathbf{v}^{(2)} = (v_0^{(2)}, v_1^{(2)}, \dots, v_l^{(2)}, \dots)$$
$$\vdots$$
$$\mathbf{v}^{(n)} = (v_0^{(n)}, v_1^{(n)}, \dots, v_l^{(n)}, \dots)$$

that we had shown, look at this example, how did we generate



(Refer Slide Time 07:20)

Introduction

Let  $k = 1$ ,  $n = 2$  and  $m = 2$ . The following circuit generates a  $(2, 1, 2)$  convolutional code.

Input:  $u_i$

Outputs:

$$\underline{v_i^{(1)}} = u_i + u_{i-2} \quad \checkmark$$
$$\underline{v_i^{(2)}} = u_i + u_{i-1} + u_{i-2} \quad \checkmark$$

two coded bits corresponding to one information sequence. How did we generate these two coded bits? These coded bits were generated by various combination of input and these past inputs. And whether a particular bit appears in the output that is governed by these interconnections, these interconnections. Whether there is a line connecting this part to the output or not, that determines whether that particular bit, it participating in the output bit. So what we can

(Refer Slide Time 08:02)

Introduction

If  $\nu_i$  is the length of the  $i^{\text{th}}$  shift register in a convolutional encoder with  $k$  input sequences,  $i = 1, 2, \dots, k$ , then

$$m = \max_{1 \leq i \leq k} \nu_i$$

The parameter  $m$  is known as memory order of the code.

conclude from here is basically we can

(Refer Slide Time 08:07)

Encoding of  $(n, 1, m)$  convolutional code

- The code is specified by a set of  $n$  generator sequences of length  $m + 1$ .

$$\mathbf{g}^{(1)} = (g_0^{(1)}, g_1^{(1)}, \dots, g_m^{(1)})$$
$$\mathbf{g}^{(2)} = (g_0^{(2)}, g_1^{(2)}, \dots, g_m^{(2)})$$
$$\vdots$$
$$\mathbf{g}^{(n)} = (g_0^{(n)}, g_1^{(n)}, \dots, g_m^{(n)})$$

completely specify a code by this set of  $n$  generator sequence of length  $m$  plus 1 where each of these generated sequence is basically of length  $m$  plus 1 and what are these  $g_0, g_1, g_2, \dots, g_m$ ? So you can see, so this superscript that you see, 1, 2, 3 and  $n$ ; this corresponds

(Refer Slide Time 08:38)

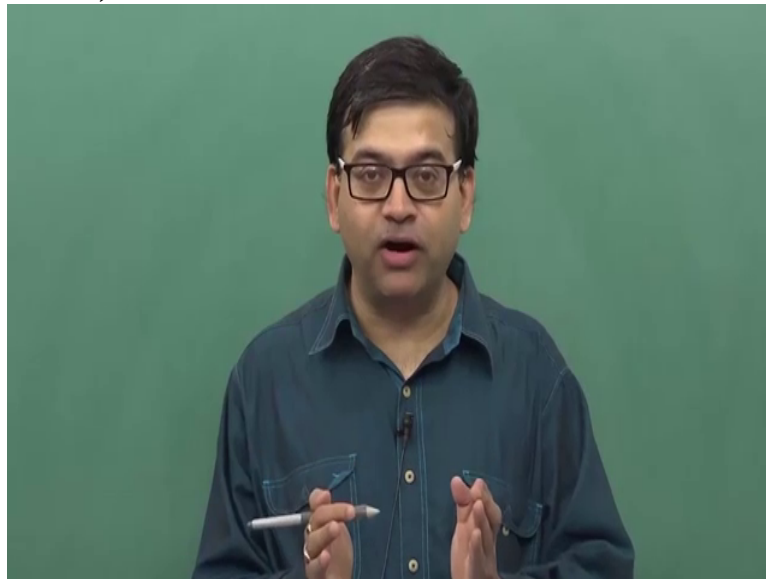
Encoding of  $(n, 1, m)$  convolutional code

- The code is specified by a set of  $n$  generator sequences of length  $m + 1$ .

$$\mathbf{g}^{(1)} = (g_0^{(1)}, g_1^{(1)}, \dots, g_m^{(1)})$$
$$\mathbf{g}^{(2)} = (g_0^{(2)}, g_1^{(2)}, \dots, g_m^{(2)})$$
$$\vdots$$
$$\mathbf{g}^{(n)} = (g_0^{(n)}, g_1^{(n)}, \dots, g_m^{(n)})$$

to each of the output sequence. So the first output sequence is specified by this generator sequence  $g_1$ . The second output sequence is specified by this generator sequence  $g_2$  and the  $n$ th output sequence is specified by this output sequence  $g_n$ . And what are these  $g_i$ 's. Now note that the memory order of our convolutional encoder

(Refer Slide Time 09:11)



is  $m$ . So there are, so if let's say, just take an example

(Refer Slide Time 09:16)

A slide titled "Encoding of  $(n, 1, m)$  convolutional code". The slide contains a bullet point: "The code is specified by a set of  $n$  generator sequences of length  $m + 1$ ,". Below this, there are three equations for generator sequences  $g^{(1)}$ ,  $g^{(2)}$ , and  $g^{(n)}$ , each represented as a vector of  $m+1$  elements:  $g^{(1)} = (g_0^{(1)}, g_1^{(1)}, \dots, g_m^{(1)})$ ,  $g^{(2)} = (g_0^{(2)}, g_1^{(2)}, \dots, g_m^{(2)})$ , and  $g^{(n)} = (g_0^{(n)}, g_1^{(n)}, \dots, g_m^{(n)})$ . The slide also shows a vertical ellipsis between  $g^{(2)}$  and  $g^{(n)}$ .

$m$  equal to 2, if we take  $m$  equal to 2, let's say 2 memory order, so then basically this is my input  $u$ . And my output  $I$  can take from some interconnections from this, let us say this is my example that I had. This was my  $v$  1; this was my  $v$  2. Now note

(Refer Slide Time 09:43)

The slide is titled "Encoding of  $(n, 1, m)$  convolutional code". It contains the following text and diagram:

- The code is specified by a set of  $n$  generator sequences of length  $m + 1$ .
- Generator sequences are listed as:
 
$$g^{(1)} = (g_0^{(1)}, g_1^{(1)}, \dots, g_m^{(1)})$$

$$g^{(2)} = (g_0^{(2)}, g_1^{(2)}, \dots, g_m^{(2)})$$

$$\vdots$$

$$g^{(n)} = (g_0^{(n)}, g_1^{(n)}, \dots, g_m^{(n)})$$
- A block diagram shows an input  $u_x$  entering a series of  $m$  shift registers. The outputs of these registers are combined with the input  $u_x$  through a set of  $n$  adders to produce  $n$  output sequences  $v_x^1, v_x^2, \dots, v_x^n$ .

that these interconnections are specifying whether a particular bit is participating in the output code sequence or not. So if we look at the first coded bit  $u_1$ , now this has memory order  $m$  so there are possible  $m + 1$  connection. What are those possible  $m + 1$  connection? One, first one is corresponding to whether  $u_1$  is participating in the output bit or not. Second one is corresponding to whether  $u_{1-1}$  is participating or not, this is this point. Third one is this point, whether  $u_{1-2}$  is participating or not. Similarly the second coded sequence, whether  $u_1$  is participating or not, whether  $u_{1-1}$  participating or not, whether  $u_{1-2}$  is participating or not. So we can see

(Refer Slide Time 10:38)

This slide is identical to the previous one, but with red annotations on the block diagram. Red dots are placed at the input  $u_x$  and at the outputs of the shift registers, indicating the specific connections that determine the output bits.

that the output here, let's take the first output sequence, that is completely specified by whether  $u_1$  is participating, where  $u_{1-1}$  is participating, whether  $u_{1-2}$  is

participating. So in this example  $v_0 = 1$ ,  $v_1 = 1$  and  $g_2 = m$ , completely specifies what inputs are participating in generating our code sequence.

(Refer Slide Time 11:09)

The slide is titled "Encoding of  $(n, 1, m)$  convolutional code". It contains the following text and diagram:

- The code is specified by a set of  $n$  generator sequences of length  $m + 1$ .
- $$g^{(1)} = (g_0^{(1)}, g_1^{(1)}, \dots, g_m^{(1)})$$

$$g^{(2)} = (g_0^{(2)}, g_1^{(2)}, \dots, g_m^{(2)})$$

$$\vdots$$

$$g^{(n)} = (g_0^{(n)}, g_1^{(n)}, \dots, g_m^{(n)})$$
- A block diagram of an encoder with  $n$  parallel paths. Each path  $i$  consists of a shift register of length  $m$  and a multiplier  $g_i^{(i)}$ . The outputs of all paths are summed to produce the code bit  $v_k$ . The input  $u_k$  is fed into all shift registers.

Similarly look at the second bit. Here also these  $m$

(Refer Slide Time 11:14)

This slide is identical to the one above, showing the same text and diagram for the encoding of  $(n, 1, m)$  convolutional code.

plus 1 connections will completely specify whether the particular bit or the past bits are taking part in the output coded bit. So you can see, if you have a rate  $1/n$  code whose memory is  $m$  then we can completely specify that code using a set of  $n$  generator sequence where each of these  $n$  generator

(Refer Slide Time 11:43)



sequence correspond to one of the output sequence and each of the generator sequence

(Refer Slide Time 11:50)

Encoding of  $(n, 1, m)$  convolutional code

- The code is specified by a set of  $n$  generator sequences of length  $m + 1$ ,

$$g^{(1)} = (g_0^{(1)}, g_1^{(1)}, \dots, g_m^{(1)})$$

$$g^{(2)} = (g_0^{(2)}, g_1^{(2)}, \dots, g_m^{(2)})$$

$$\vdots$$

$$g^{(n)} = (g_0^{(n)}, g_1^{(n)}, \dots, g_m^{(n)})$$

is of length  $m + 1$  specifying the interconnections of  $u_l$ ,  $u_{l-1}$ ,  $u_{l-2}$  up to  $u_{l-m}$ . So then what are these  $g_0$  and  $g_1$ ? If  $g_0$  and  $g_1$  are either 1 or 0, 1 means they are participating, 0 means it does not participate. For example, in this example, what is  $g_1$ ? Is  $u_l$  participating in the output sequence of  $v_l$ ? Yes it is. So then  $g_0$  will be 1. Is  $u_{l-1}$  participating in the output sequence  $v_l$ ? No, so then this will be zero. What about  $u_{l-2}$ ? It is participating in the output sequence. So it will be 1. So  $g_1$  is 1 0 1.

(Refer Slide Time 12:47)

Encoding of  $(n, 1, m)$  convolutional code

- The code is specified by a set of  $n$  generator sequences of length  $m + 1$ .

$$g^{(1)} = (g_0^{(1)}, g_1^{(1)}, \dots, g_m^{(1)})$$

$$g^{(2)} = (g_0^{(2)}, g_1^{(2)}, \dots, g_m^{(2)})$$

$$\vdots$$

$$g^{(n)} = (g_0^{(n)}, g_1^{(n)}, \dots, g_m^{(n)})$$

The diagram shows an encoder with an input  $u_2$  and  $n$  outputs  $v_2^{(1)}, v_2^{(2)}, \dots, v_2^{(n)}$ . Handwritten notes indicate  $g^{(1)} = (101)$ .

Similarly  $g_2$  will be 1 1 1 because  $u_1$ ,

(Refer Slide Time 12:54)

Encoding of  $(n, 1, m)$  convolutional code

- The code is specified by a set of  $n$  generator sequences of length  $m + 1$ .

$$g^{(1)} = (g_0^{(1)}, g_1^{(1)}, \dots, g_m^{(1)})$$

$$g^{(2)} = (g_0^{(2)}, g_1^{(2)}, \dots, g_m^{(2)})$$

$$\vdots$$

$$g^{(n)} = (g_0^{(n)}, g_1^{(n)}, \dots, g_m^{(n)})$$

The diagram shows an encoder with an input  $u_2$  and  $n$  outputs  $v_2^{(1)}, v_2^{(2)}, \dots, v_2^{(n)}$ . Handwritten notes indicate  $g^{(1)} = (101)$  and  $g^{(2)} = (111)$ .

$u_1$  minus 1,  $u_1$  minus 2 they are all participating in the output coded sequence, Ok. So if I specify these generator sequences then my convolutional code is completely specified.

(Refer Slide Time 13:12)

Encoding of  $(n, 1, m)$  convolutional code

- The code is specified by a set of  $n$  generator sequences of length  $m + 1$ ,  
$$\mathbf{g}^{(1)} = (g_0^{(1)}, g_1^{(1)}, \dots, g_m^{(1)})$$
$$\mathbf{g}^{(2)} = (g_0^{(2)}, g_1^{(2)}, \dots, g_m^{(2)})$$
$$\vdots$$
$$\mathbf{g}^{(n)} = (g_0^{(n)}, g_1^{(n)}, \dots, g_m^{(n)})$$
- The output sequence is the discrete convolution of the information sequence  $\mathbf{u}$  and the generator sequence  $\mathbf{g}^{(i)}$ , i.e.  
$$\mathbf{v}^{(i)} = \mathbf{u} * \mathbf{g}^{(i)}, \quad 1 \leq i \leq n,$$

and

$$\mathbf{v}_i^{(i)} = u_i g_0^{(i)} + u_{i-1} g_1^{(i)} + \dots + u_{i-m} g_m^{(i)}$$
$$= \sum_{j=0}^m u_{i-j} g_j^{(i)}$$

And what is my output then? My output is nothing but it is a discrete convolution of the information sequence with this generator sequence. So if my generator sequence, if my code has memory  $m$ , then basically I can write this discrete convolution in this particular fashion.

(Refer Slide Time 13:37)

Encoding of  $(n, 1, m)$  convolutional code

- The code is specified by a set of  $n$  generator sequences of length  $m + 1$ ,  
$$\mathbf{g}^{(1)} = (g_0^{(1)}, g_1^{(1)}, \dots, g_m^{(1)})$$
$$\mathbf{g}^{(2)} = (g_0^{(2)}, g_1^{(2)}, \dots, g_m^{(2)})$$
$$\vdots$$
$$\mathbf{g}^{(n)} = (g_0^{(n)}, g_1^{(n)}, \dots, g_m^{(n)})$$
- The output sequence is the discrete convolution of the information sequence  $\mathbf{u}$  and the generator sequence  $\mathbf{g}^{(i)}$ , i.e.  
$$\mathbf{v}^{(i)} = \mathbf{u} * \mathbf{g}^{(i)}, \quad 1 \leq i \leq n,$$

and

$$\mathbf{v}_i^{(i)} = u_i g_0^{(i)} + u_{i-1} g_1^{(i)} + \dots + u_{i-m} g_m^{(i)}$$
$$= \sum_{j=0}^m u_{i-j} g_j^{(i)}$$

And that's basically my output sequence, which is discrete convolution of the input sequence with this generator sequence. Now let us take an example.



(Refer Slide Time 13:54)

Encoding of  $(n, 1, m)$  convolutional code

Example:

- Consider a rate  $R = 1/2$ ,  $(2, 1, 2)$  convolutional code specified by the following generator sequences

$$g^{(1)} = (1 \ 0 \ 1),$$

$$g^{(2)} = (1 \ 1 \ 1),$$

This is the same example that we are considering, this rate one half code with memory 2 so you can see  $v_1^{(1)}$ , this is basically again discrete convolution of input with this generator sequence which we can write as  $u_i + u_{i-2}$  and this

(Refer Slide Time 14:20)

Encoding of  $(n, 1, m)$  convolutional code

Example:

- Consider a rate  $R = 1/2$ ,  $(2, 1, 2)$  convolutional code specified by the following generator sequences

$$g^{(1)} = (1 \ 0 \ 1),$$

$$g^{(2)} = (1 \ 1 \ 1),$$

$v_i^{(1)} = u_i + u_{i-2}$

$v_1^{(2)}$  can be written as  $u_i + u_{i-1} + u_{i-2}$ . If we go back,

(Refer Slide Time 14:30)

Encoding of  $(n, 1, m)$  convolutional code

Example:

- Consider a rate  $R = 1/2$ ,  $(2, 1, 2)$  convolutional code specified by the following generator sequences

$$\mathbf{g}^{(1)} = (1 \ 0 \ 1),$$

$$\mathbf{g}^{(2)} = (1 \ 1 \ 1),$$

Handwritten notes:

$$V_l^{(1)} = U_l + U_{l-2}$$

$$V_l^{(2)} = U_l + U_{l-1} + U_{l-2}$$

(Refer Slide Time 14:32)

Encoding of  $(n, 1, m)$  convolutional code

- The code is specified by a set of  $n$  generator sequences of length  $m + 1$ ,

$$\mathbf{g}^{(1)} = (g_0^{(1)}, g_1^{(1)}, \dots, g_m^{(1)})$$

$$\mathbf{g}^{(2)} = (g_0^{(2)}, g_1^{(2)}, \dots, g_m^{(2)})$$

$$\vdots$$

$$\mathbf{g}^{(n)} = (g_0^{(n)}, g_1^{(n)}, \dots, g_m^{(n)})$$

- The output sequence is the discrete convolution of the information sequence  $\mathbf{u}$  and the generator sequence  $\mathbf{g}^{(i)}$ , i.e.

$$\mathbf{v}^{(i)} = \mathbf{u} * \mathbf{g}^{(i)}, \quad 1 \leq i \leq n,$$

and

$$\mathbf{v}_l^{(i)} = u_l g_0^{(i)} + u_{l-1} g_1^{(i)} + \dots + u_{l-m} g_m^{(i)}$$

$$= \sum_{j=0}^m u_{l-j} g_j^{(i)}$$

our output is this if we can expand it for this particular example this will be  $u_l g_0^{(1)} + u_{l-1} g_1^{(1)} + u_{l-2} g_2^{(1)}$  plus  $u_l g_0^{(2)} + u_{l-1} g_1^{(2)} + u_{l-2} g_2^{(2)}$ .

(Refer Slide Time 15:05)

**Encoding of  $(n, 1, m)$  convolutional code**

- The code is specified by a set of  $n$  generator sequences of length  $m + 1$ ,
 
$$\mathbf{g}^{(1)} = (g_0^{(1)}, g_1^{(1)}, \dots, g_m^{(1)})$$

$$\mathbf{g}^{(2)} = (g_0^{(2)}, g_1^{(2)}, \dots, g_m^{(2)})$$

$$\vdots$$

$$\mathbf{g}^{(n)} = (g_0^{(n)}, g_1^{(n)}, \dots, g_m^{(n)})$$
- The output sequence is the discrete convolution of the information sequence  $\mathbf{u}$  and the generator sequence  $\mathbf{g}^{(i)}$ , i.e.
 
$$\mathbf{v}^{(i)} = \mathbf{u} * \mathbf{g}^{(i)}, \quad 1 \leq i \leq n,$$
 and
 
$$\mathbf{v}_i^{(i)} = u_i g_0^{(i)} + u_{i-1} g_1^{(i)} + \dots + u_{i-m} g_m^{(i)}$$

$$= \sum_{j=0}^m u_{i-j} g_j^{(i)} = u_x g_0^{(i)} + u_{x-1} g_1^{(i)} + u_{x-2} g_2^{(i)}$$

And for the first coded sequence, this  $g$  is,  $g_0 g_1 g_2$  was  $1 0 1$  and the second sequence was  $1 1 1$ , that's why the first

(Refer Slide Time 15:18)

**Encoding of  $(n, 1, m)$  convolutional code**

**Example:**

- Consider a rate  $R = 1/2$ ,  $(2, 1, 2)$  convolutional code specified by the following generator sequences
 
$$\mathbf{g}^{(1)} = (1 0 1),$$

$$\mathbf{g}^{(2)} = (1 1 1),$$

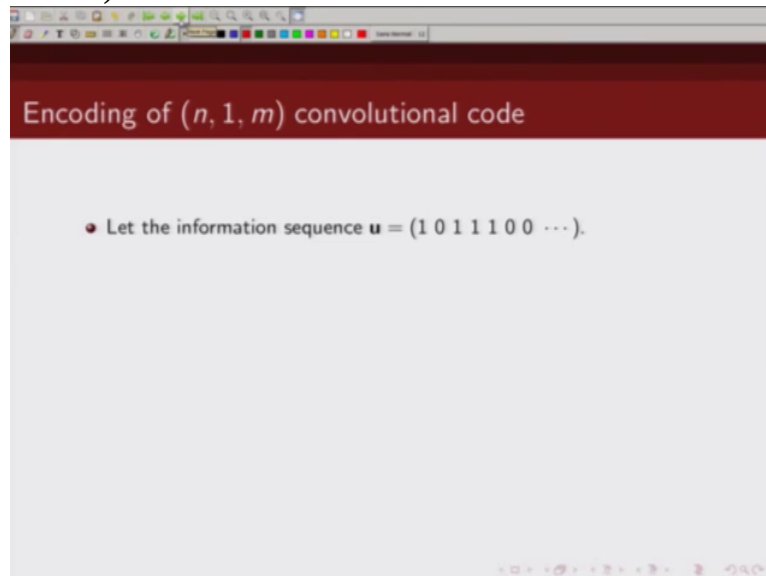
Block diagram showing the encoding process with delay elements  $u_{l-1}$  and  $u_{l-2}$ . Handwritten red notes show the equations for the output sequences:

$$V_l^{(1)} = U_l + U_{l-2}$$

$$V_l^{(2)} = U_l + U_{l-1} + U_{l-2}$$

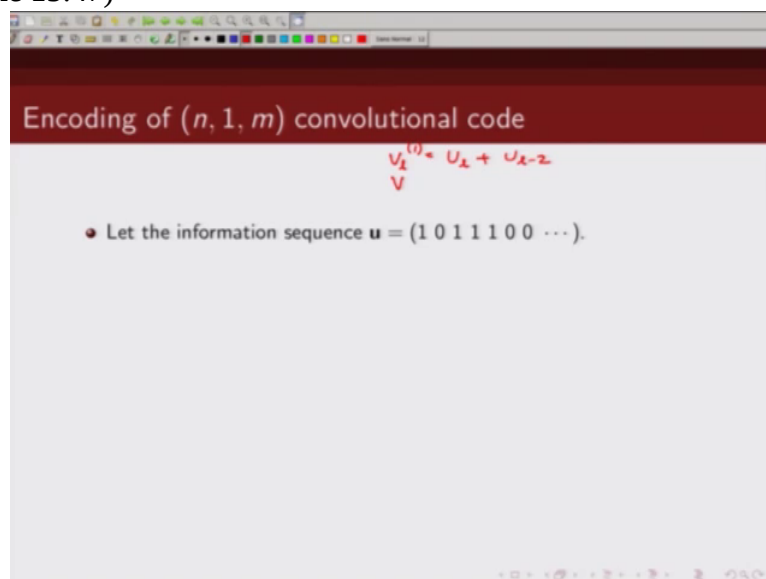
coded sequence is  $u_l$  plus  $u_{l-2}$  and the second coded sequence is  $u_l$  plus  $u_{l-1}$  plus  $u_{l-2}$ . So we

(Refer Slide Time 15:31)



have information sequence this. What was our output sequence? We had  $v_1$  is  $u_1$  plus  $u_1$  minus 2 and

(Refer Slide Time 15:47)



$v_2$  is  $u_1$  plus  $u_1$  minus 1 plus  $u_1$  minus 2. Now

(Refer Slide Time 15:56)

Encoding of  $(n, 1, m)$  convolutional code

$$v_l^{(1)} = u_l + u_{l-2}$$
$$v_l^{(2)} = u_l + u_{l-1} + u_{l-2}$$

- Let the information sequence  $\mathbf{u} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ \dots)$ .

we can show that

(Refer Slide Time 15:58)

Encoding of  $(n, 1, m)$  convolutional code

- Let the information sequence  $\mathbf{u} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ \dots)$ .
- The output sequences are given by

$$\mathbf{v}^{(1)} = (1\ 0\ 0\ 1\ 0\ 1\ 1\ \dots)$$
$$\mathbf{v}^{(2)} = (1\ 1\ 0\ 0\ 1\ 0\ 1\ \dots)$$

our output coded sequence will be given by this. Now this can be easily verified. So let's say what was our output coded sequence?  $v_1$  was  $u_1$  plus  $u_{1-2}$  and  $v_2$  is  $u_2$  plus  $u_{2-1}$  plus  $u_{2-2}$ . Now note

(Refer Slide Time 16:24)

Encoding of  $(n, 1, m)$  convolutional code

$v_x^{(1)} = u_x + u_{x-2}$   
 $v_x^{(2)} = u_x + u_{x-1} + u_{x-2}$

- Let the information sequence  $\mathbf{u} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ \dots)$ .
- The output sequences are given by

$$\mathbf{v}^{(1)} = (1\ 0\ 0\ 1\ 0\ 1\ 1\ \dots)$$
$$\mathbf{v}^{(2)} = (1\ 1\ 0\ 0\ 1\ 0\ 1\ \dots)$$

when the first input  $u_1$  which is 1 comes, what is the output? Now to specify the output we need to specify what the initial contents of  $u_1$  minus 1 and  $u_1$  minus 2. So initially we will assume that the

(Refer Slide Time 16:44)



convolutional encoder was in all zero state. What do we mean by all zero state? So we are assuming that initially the contents of the shift register were all zero. In other words  $u_1$  minus 2 and  $u_1$  minus 1, they were both zero, Ok. If both were zero initially and if  $u_1$  is 1, then what will be  $v_1$ ?

(Refer Slide Time 17:09)

Encoding of  $(n, 1, m)$  convolutional code

$v_x^{(1)} = u_x + u_{x-2}$   
 $v_x^{(2)} = u_x + u_{x-1} + u_{x-2}$

- Let the information sequence  $\mathbf{u} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ \dots)$ .
- The output sequences are given by
 
$$\mathbf{v}^{(1)} = (1\ 0\ 0\ 1\ 0\ 1\ 1\ \dots)$$

$$\mathbf{v}^{(2)} = (1\ 1\ 0\ 0\ 1\ 0\ 1\ \dots)$$

This is 1 plus 0 which is 1, so you can see this is 1. And what is  $v_1^{(2)}$ ? This is 1 plus 0 plus 0, so that's also 1. Next what happens, next if you go back this 1 which was here,

(Refer Slide Time 17:33)

Encoding of  $(n, 1, m)$  convolutional code

Example:

- Consider a rate  $R = 1/2$ ,  $(2, 1, 2)$  convolutional code specified by the following generator sequences
 
$$\mathbf{g}^{(1)} = (1\ 0\ 1),$$

$$\mathbf{g}^{(2)} = (1\ 1\ 1),$$

$v_l^{(1)} = u_l + u_{l-2}$   
 $v_l^{(2)} = u_l + u_{l-1} + u_{l-2}$

when you apply a clock this 1 moves here and a new bit comes here. So now the next time instance  $u_{l-1}$  becomes 1 and what is  $u_{l-2}$ ? Since  $u_{l-1}$  initially was zero so this zero will come here. So the new contents of the shift register will be now 1 and 0. So

(Refer Slide Time 17:58)

Encoding of  $(n, 1, m)$  convolutional code

$v_x^{(1)} = u_x + u_{x-2}$   
 $v_x^{(2)} = u_x + u_{x-1} + u_{x-2}$

- Let the information sequence  $\mathbf{u} = (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ \dots)$ .
- The output sequences are given by
 
$$\mathbf{v}^{(1)} = (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ \dots)$$

$$\mathbf{v}^{(2)} = (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ \dots)$$

what we have is now  $u_{l-1}$  is 1 and  $u_{l-2}$  is 0. Now the next input is zero. The next input is zero so what's the next output? This is zero and  $u_{l-2}$  is zero so this will be zero. You can see this is zero. What about this? Now  $u_l$  is zero,  $u_{l-1}$  is 1, and  $u_{l-2}$  is zero. So zero plus 1 plus 0, that will be 1 and that's given by this, Ok. Next what happens? Again go back to

(Refer Slide Time 18:39)

Encoding of  $(n, 1, m)$  convolutional code

Example:

- Consider a rate  $R = 1/2$ ,  $(2, 1, 2)$  convolutional code specified by the following generator sequences
 
$$\mathbf{g}^{(1)} = (1 \ 0 \ 1),$$

$$\mathbf{g}^{(2)} = (1 \ 1 \ 1),$$

$v_l^{(1)} = u_l + u_{l-2}$   
 $v_l^{(2)} = u_l + u_{l-1} + u_{l-2}$

this diagram. You had input zero here. So now this zero will move here and you had a 1 here. So this 1 will move here. So the new content of the shift register will be zero and 1, Ok. If that happens then next



(Refer Slide Time 18:58)

Encoding of  $(n, 1, m)$  convolutional code

$v_x^{(1)} = u_x + u_{x-2}$   
 $v_x^{(2)} = u_x + u_{x-1} + u_{x-2}$

- Let the information sequence  $\mathbf{u} = (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ \dots)$ .
- The output sequences are given by
 
$$\mathbf{v}^{(1)} = (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ \dots)$$

$$\mathbf{v}^{(2)} = (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ \dots)$$

input is 1. So this is 1. What is  $u_{l-2}$ ?  $u_{l-2}$  was 1 so  $1 + 1$  that's zero. And  $u_{l-1}$  is 1,  $u_{l-2}$  is 1 and  $u_{l-1}$  is zero. So  $1 + 0 + 1$  that's zero. So like that, you can basically write down the output coded sequence. So then what is my final output? So corresponding to this 1 and what is my coded sequence? That's given by this.

(Refer Slide Time 19:39)

Encoding of  $(n, 1, m)$  convolutional code

$v_x^{(1)} = u_x + u_{x-2}$   
 $v_x^{(2)} = u_x + u_{x-1} + u_{x-2}$

- Let the information sequence  $\mathbf{u} = (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ \dots)$ .
- The output sequences are given by
 
$$\mathbf{v}^{(1)} = (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ \dots)$$

$$\mathbf{v}^{(2)} = (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ \dots)$$

Corresponding to this zero, my coded sequence is given by this,

(Refer Slide Time 19:45)

Encoding of  $(n, 1, m)$  convolutional code

$v_x^{(1)} = u_x + u_{x-2}$   
 $v_x^{(2)} = u_x + u_{x-1} + u_{x-2}$

- Let the information sequence  $\mathbf{u} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ \dots)$ .
- The output sequences are given by

$$\mathbf{v}^{(1)} = (1\ 0\ 0\ 1\ 0\ 1\ 1\ \dots)$$
$$\mathbf{v}^{(2)} = (1\ 1\ 0\ 0\ 1\ 0\ 1\ \dots)$$

Ok. So then I can write my

(Refer Slide Time 19:48)

Encoding of  $(n, 1, m)$  convolutional code

- Let the information sequence  $\mathbf{u} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ \dots)$ .
- The output sequences are given by

$$\mathbf{v}^{(1)} = (1\ 0\ 0\ 1\ 0\ 1\ 1\ \dots)$$
$$\mathbf{v}^{(2)} = (1\ 1\ 0\ 0\ 1\ 0\ 1\ \dots)$$

- The code sequence can be written as

$$\mathbf{v} = (11, 01, 00, 10, 01, 10, 11, \dots)$$

final output as, corresponding to input 1, I get 1 1. That's given by this. Corresponding to zero I get 0 1, that's given by this. Corresponding to 1, I get 0 0, that's given by this. So this is how I can write my output coded sequence.

(Refer Slide Time 20:12)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix

- Matrix form

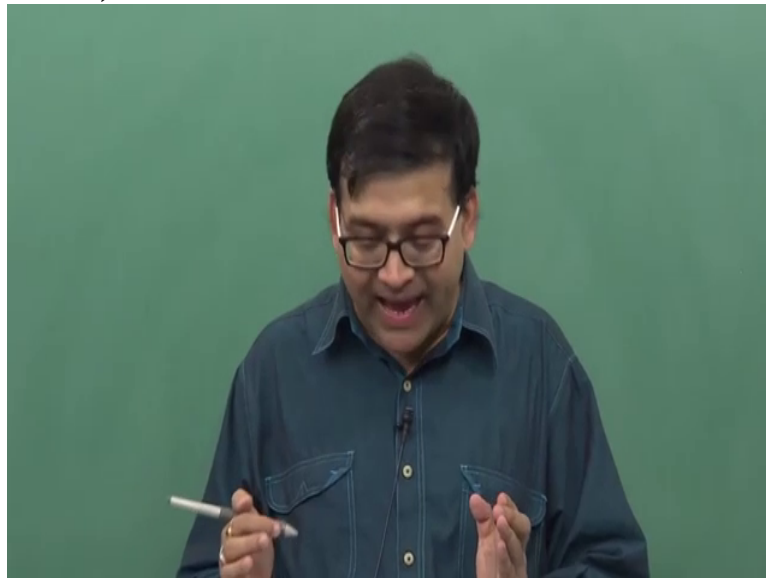
$$\mathbf{v} = \mathbf{uG}$$
$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 & \mathbf{g}_1 & \cdots & \cdots & \mathbf{g}_m \\ & \mathbf{g}_0 & \mathbf{g}_1 & \cdots & \cdots & \mathbf{g}_m \\ & & \ddots & & & \ddots \\ & & & & & & \ddots \end{bmatrix}$$

where

$$\mathbf{g}_i = (g_i^{(1)} \ g_i^{(2)} \ \cdots \ g_i^{(n)}), \quad 0 \leq i \leq m$$

Now the same thing I can write in the

(Refer Slide Time 20:15)



matrix form. So I define this generator

(Refer Slide Time 20:19)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix

- Matrix form

$$\mathbf{v} = \mathbf{uG}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 & \mathbf{g}_1 & \cdots & \cdots & \mathbf{g}_m & & \\ & \mathbf{g}_0 & \mathbf{g}_1 & \cdots & \cdots & \mathbf{g}_m & \\ & & & \ddots & & & \ddots \\ & & & & & & & \ddots \end{bmatrix}$$

where

$$\mathbf{g}_i = (g_i^{(1)} \ g_i^{(2)} \ \cdots \ g_i^{(n)}), \ 0 \leq i \leq m$$

matrix G which generates this codeword. So the output codeword can be written as input times this generator matrix G, Ok and this generator matrix is of the form like this. So let's just expand it and may be try to explain why the generator form has this semi infinite kind of form for a convolutional code. So let's say u is  $u_0, u_1, u_2$  dah dah dah it's continuing set of

(Refer Slide Time 20:58)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix

- Matrix form

$$\mathbf{v} = \mathbf{uG}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 & \mathbf{g}_1 & \cdots & \cdots & \mathbf{g}_m & & \\ & \mathbf{g}_0 & \mathbf{g}_1 & \cdots & \cdots & \mathbf{g}_m & \\ & & & \ddots & & & \ddots \\ & & & & & & & \ddots \end{bmatrix}$$

where

$$\mathbf{g}_i = (g_i^{(1)} \ g_i^{(2)} \ \cdots \ g_i^{(n)}), \ 0 \leq i \leq m$$

$\mathbf{u} = [u_0 \ u_1 \ u_2 \ \dots]$

sequence like this right now what is your output sequence? Output sequence so initially what happens, if you go back to this diagram,

(Refer Slide Time 21:12)

Encoding of  $(n, 1, m)$  convolutional code

- Let the information sequence  $\mathbf{u} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ \dots)$ .
- The output sequences are given by
 
$$\mathbf{v}^{(1)} = (1\ 0\ 0\ 1\ 0\ 1\ 1\ \dots)$$

$$\mathbf{v}^{(2)} = (1\ 1\ 0\ 0\ 1\ 0\ 1\ \dots)$$
- The code sequence can be written as
 
$$\mathbf{v} = (\underline{11}, \underline{01}, \underline{00}, 10, 01, 10, 11, \dots)$$

initially you are assuming

(Refer Slide Time 21:17)

Encoding of  $(n, 1, m)$  convolutional code

Example:

- Consider a rate  $R = 1/2$ ,  $(2, 1, 2)$  convolutional code specified by the following generator sequences
 
$$\mathbf{g}^{(1)} = (1\ 0\ 1),$$

$$\mathbf{g}^{(2)} = (1\ 1\ 1),$$

$$V_i^{(1)} = U_i + U_{i-2}$$

$$V_i^{(2)} = U_i + U_{i-1} + U_{i-2}$$

that the encoder is in all zero state

(Refer Slide Time 21:26)

Encoding of  $(n, 1, m)$  convolutional code

Example:

- Consider a rate  $R = 1/2$ ,  $(2, 1, 2)$  convolutional code specified by the following generator sequences

$$g^{(1)} = (1 \ 0 \ 1),$$

$$g^{(2)} = (1 \ 1 \ 1),$$

Handwritten notes:

$$V_i^{(1)} = U_i + U_{i-2}$$

$$V_i^{(2)} = U_i + U_{i-1} + U_{i-2}$$

correct? So what will be the first output that you will get here? That is nothing but  $u_0$  times  $g_0$ . What is  $g_0$ ?  $g_0^{(1)}$  is this,  $g_0^{(2)}$  is this, this interconnection, which is connecting  $u_i$  to the output. So at first time instance what you would get

(Refer Slide Time 21:51)

Encoding of  $(n, 1, m)$  convolutional code

$$V_i^{(1)} = U_i + U_{i-2}$$

$$V_i^{(2)} = U_i + U_{i-1} + U_{i-2}$$

- Let the information sequence  $\mathbf{u} = (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ \dots)$ .

is the output is nothing

(Refer Slide Time 21:53)

Encoding of  $(n, 1, m)$  convolutional code

- Let the information sequence  $\mathbf{u} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ \dots)$ .
- The output sequences are given by

$$\mathbf{v}^{(1)} = (1\ 0\ 0\ 1\ 0\ 1\ 1\ \dots)$$
$$\mathbf{v}^{(2)} = (1\ 1\ 0\ 0\ 1\ 0\ 1\ \dots)$$

but

(Refer Slide Time 21:55)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix

- Matrix form

$$\mathbf{v} = \mathbf{uG}$$
$$\mathbf{G} = \begin{bmatrix} g_0 & g_1 & \dots & \dots & g_m \\ & g_0 & g_1 & \dots & \dots & g_m \\ & & \ddots & & & \ddots \end{bmatrix}$$

where

$$\mathbf{g}_i = (g_i^{(1)}\ g_i^{(2)}\ \dots\ g_i^{(n)}),\ 0 \leq i \leq m$$

$\mathbf{u} = [u_0\ u_1\ u_2\ \dots]$

but  $u_0$  times  $g_0$ . This is the output that you will get at first time instance.

(Refer Slide Time 22:01)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix

- Matrix form

$$\mathbf{v} = \mathbf{uG}$$

$$\mathbf{G} = \begin{bmatrix} g_0 & g_1 & \dots & g_m \\ & g_0 & g_1 & \dots & g_m \\ & & \ddots & \ddots & \ddots \\ & & & & g_m \end{bmatrix}$$

where

$$g_i = (g_i^{(1)} \ g_i^{(2)} \ \dots \ g_i^{(n)}), \quad 0 \leq i \leq m$$

Handwritten notes:  $\mathbf{v} = [v_0 \ v_1 \ v_2 \ \dots]$  and  $u_0 g_0$

What is the output you will get at the second time instance? Now whatever  $u_0$  you had, now that  $u_0$  has moved here,

(Refer Slide Time 22:12)

Encoding of  $(n, 1, m)$  convolutional code

Example:

- Consider a rate  $R = 1/2$ ,  $(2, 1, 2)$  convolutional code specified by the following generator sequences

$$g^{(1)} = (1 \ 0 \ 1),$$

$$g^{(2)} = (1 \ 1 \ 1),$$

Handwritten notes:  $v_1^{(1)} = u_x + u_{x-2}$  and  $v_1^{(2)} = u_x + u_{x-1} + u_{x-2}$

correct and a new bit which is  $u_1$  has come here,  $u_1$ . So what is the output at this time? It is  $u_1$  times  $g_0$  plus  $u_0$  times  $g_1$ .



(Refer Slide Time 22:46)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix

- Matrix form

$$\mathbf{v} = \mathbf{uG}$$

$$\mathbf{G} = \begin{bmatrix} g_0 & g_1 & \dots & \dots & g_m \\ & g_0 & g_1 & \dots & \dots & g_m \\ & & \ddots & \ddots & \ddots & \ddots \\ & & & & & \ddots \end{bmatrix}$$

where

$$g_i = (g_i^{(1)} \ g_i^{(2)} \ \dots \ g_i^{(n)}), \quad 0 \leq i \leq m$$

*Handwritten notes:*  
 $\mathbf{v} = [v_0 \ v_1 \ v_2 \ \dots \ ]$   
 $v_0 = u_0 g_0$   
 $v_1 = u_1 g_0 + u_0 g_1$

So I can write, and the second time instance my output is given by  $u_1$  times  $g_0$  plus  $u_0$  times  $g_1$ , fine. Next time instance, what is my output? Again go back, now what is going to happen is,

(Refer Slide Time 22:58)

Encoding of  $(n, 1, m)$  convolutional code

Example:

- Consider a rate  $R = 1/2$ ,  $(2, 1, 2)$  convolutional code specified by the following generator sequences

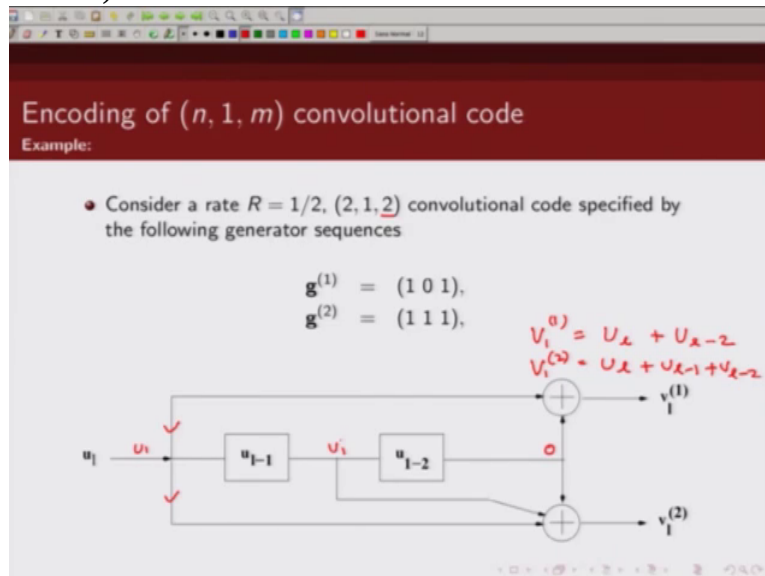
$$g^{(1)} = (1 \ 0 \ 1),$$

$$g^{(2)} = (1 \ 1 \ 1),$$

*Handwritten notes:*  
 $v_i^{(1)} = u_i + u_{i-2}$   
 $v_i^{(2)} = u_i + u_{i-1} + u_{i-2}$

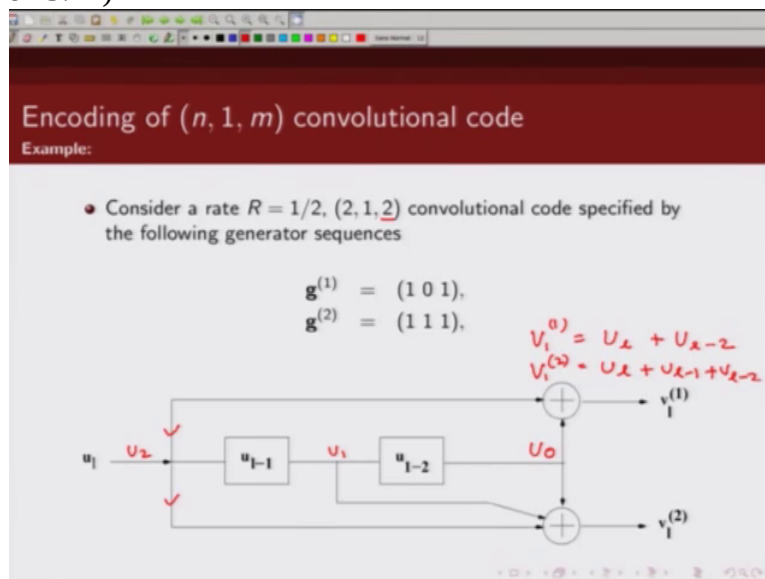
this  $u_1$  will move here. So this will be now  $u_1$ .

(Refer Slide Time 23:07)



This will become u 2 and this will become u 0. So what is my output now?

(Refer Slide Time 23:21)



It is u 2 times v 0 plus u 1 times v 1 plus u 0 times g 2. So go back, so what would

(Refer Slide Time 23:40)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix

- Matrix form

$$\underline{v} = \underline{uG}$$

$$G = \begin{bmatrix} g_0 & g_1 & \cdots & \cdots & g_m \\ & g_0 & g_1 & \cdots & \cdots & g_m \\ & & \ddots & & & \ddots \\ & & & & & \ddots \end{bmatrix}$$

where

$$g_i = (g_i^{(1)} \ g_i^{(2)} \ \cdots \ g_i^{(n)}), \quad 0 \leq i \leq m$$

$v = [v_0 \ v_1 \ v_2 \ \dots \ ]$

$v_0 g_0$

$v_1 g_0 + v_0 g_1$

be my output here? It is  $v_0$  times  $g_0$  plus  $v_1$  times  $g_1$  plus  $v_2$  times  $g_2$ .

(Refer Slide Time 23:54)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix

- Matrix form

$$\underline{v} = \underline{uG}$$

$$G = \begin{bmatrix} g_0 & g_1 & \cdots & \cdots & g_m \\ & g_0 & g_1 & \cdots & \cdots & g_m \\ & & \ddots & & & \ddots \\ & & & & & \ddots \end{bmatrix}$$

where

$$g_i = (g_i^{(1)} \ g_i^{(2)} \ \cdots \ g_i^{(n)}), \quad 0 \leq i \leq m$$

$v = [v_0 \ v_1 \ v_2 \ \dots \ ]$

$v_0 g_0$

$v_1 g_0 + v_0 g_1$

$v_2 g_0 + v_1 g_1 + v_0 g_2$

What happens next?

(Refer Slide Time 24:00)

Encoding of  $(n, 1, m)$  convolutional code

Example:

- Consider a rate  $R = 1/2$ ,  $(2, 1, 2)$  convolutional code specified by the following generator sequences

$$g^{(1)} = (1\ 0\ 1),$$

$$g^{(2)} = (1\ 1\ 1),$$

Handwritten notes in red:

$$V_i^{(1)} = U_i + U_{i-2}$$

$$V_i^{(2)} = U_i + U_{i-1} + U_{i-2}$$

This  $u_0$  moves out. Here what we will get is  $u_1$ , this will be  $u_1$ . What about this,

(Refer Slide Time 24:14)

Encoding of  $(n, 1, m)$  convolutional code

Example:

- Consider a rate  $R = 1/2$ ,  $(2, 1, 2)$  convolutional code specified by the following generator sequences

$$g^{(1)} = (1\ 0\ 1),$$

$$g^{(2)} = (1\ 1\ 1),$$

Handwritten notes in red:

$$V_i^{(1)} = U_i + U_{i-2}$$

$$V_i^{(2)} = U_i + U_{i-1} + U_{i-2}$$

this will become  $u_2$ , so this is  $u_2$  and this will

(Refer Slide Time 24:21)

Encoding of  $(n, 1, m)$  convolutional code

Example:

- Consider a rate  $R = 1/2$ ,  $(2, 1, 2)$  convolutional code specified by the following generator sequences

$$g^{(1)} = (1 \ 0 \ 1),$$

$$g^{(2)} = (1 \ 1 \ 1),$$

Handwritten notes on the diagram:

$$V_i^{(1)} = U_i + U_{i-2}$$

$$V_i^{(2)} = U_i + U_{i-1} + U_{i-2}$$

become  $u_3$ . So this is  $u_3$ . This is  $u_3$ . So what will be the output now? It's  $u_3$  times  $g_0$  plus  $u_2$  times  $g_1$  plus  $u_1$  times  $g_2$ . And  $u_0$  does not appear because the memory order of this code was 2. So what is the output in this case? Third instance, this will be

(Refer Slide Time 24:48)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix

- Matrix form

$$\underline{v} = \underline{uG}$$

$$G = \begin{bmatrix} g_0 & g_1 & \dots & \dots & g_m \\ & g_0 & g_1 & \dots & \dots & g_m \\ & & \dots & \dots & \dots & \dots \\ & & & \dots & \dots & \dots \end{bmatrix}$$

where

$$g_i = (g_i^{(1)} \ g_i^{(2)} \ \dots \ g_i^{(n)}), \quad 0 \leq i \leq m$$

Handwritten notes on the slide:

$$v = [v_0 \ v_1 \ v_2 \ \dots]$$

$$v_0 = u_0 g_0$$

$$v_1 = u_1 g_0 + u_0 g_1$$

$$v_2 = u_2 g_0 + u_1 g_1 + u_0 g_2$$

$u_3$  times  $g_0$  plus  $u_2$  times  $g_1$  plus  $u_1$  times  $g_2$ . Now if we write the same thing in a matrix form, so what is  $v$ ?  $v$  is

(Refer Slide Time 25:07)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix

- Matrix form  $\underline{v} = \underline{u}G$

$$G = \begin{bmatrix} g_0 & g_1 & \dots & \dots & g_m \\ & g_0 & g_1 & \dots & g_m \\ & & \ddots & \ddots & \ddots \\ & & & & \ddots \end{bmatrix}$$

where  $g_i = (g_i^{(1)} \ g_i^{(2)} \ \dots \ g_i^{(n)}), 0 \leq i \leq m$

Handwritten notes:

$$\underline{v} = [v_0 \ v_1 \ v_2 \ \dots \ ]$$

$$v_0 = u_0 g_0$$

$$v_1 = u_1 g_0 + u_0 g_1$$

$$v_2 = u_2 g_0 + u_1 g_1 + u_0 g_2$$

$$v_3 = u_3 g_0 + u_2 g_1 + u_1 g_2$$

basically, v at times zero, time 1, time 2, if we write this in this particular form, is equal to u times this matrix G.

(Refer Slide Time 25:17)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix

- Matrix form  $[v_0 \ v_1 \ v_2 \ \dots \ ] = \underline{v} = \underline{u}G$

$$G = \begin{bmatrix} g_0 & g_1 & \dots & \dots & g_m \\ & g_0 & g_1 & \dots & g_m \\ & & \ddots & \ddots & \ddots \\ & & & & \ddots \end{bmatrix}$$

where  $g_i = (g_i^{(1)} \ g_i^{(2)} \ \dots \ g_i^{(n)}), 0 \leq i \leq m$

Handwritten notes:

$$\underline{v} = [v_0 \ v_1 \ v_2 \ \dots \ ]G$$

$$v_0 = u_0 g_0$$

$$v_1 = u_1 g_0 + u_0 g_1$$

$$v_2 = u_2 g_0 + u_1 g_1 + u_0 g_2$$

$$v_3 = u_3 g_0 + u_2 g_1 + u_1 g_2$$

Now you compare this equation with this equation. So at first time instance, output is  $u_0 g_0$ . So that's what, this is  $u_0$  times  $g_0$ . So this is  $g_0$ . Second instance, what is my output? My output is  $u_0$  times  $g_1$ , this term,  $u_0$  times  $g_1$  is this term and then  $g_0$  times  $u_1$ , which is this term. So the second entry of this generator matrix is this, Ok. Now what's the third entry you can see?  $u_0$  times  $g_2$ , so  $u_0$  times this is  $g_2$ , plus  $u_1$  times  $g_1$ , so this is  $g_1$  and then this is  $u_2$  times  $g_0$ . So you can see. Then further if we look at this, what we get here is, so  $u_0$  times zero will get here and then and we will get  $u_1$  times  $g_2$ ,  $u_3$  times so this will be like

zero and  $g_2, g_1, g_0$  so in this case the memory order was  $m$ . That's why we are getting like this.

(Refer Slide Time 26:37)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix

- Matrix form  $[v_0 v_1 v_2 \dots] = v = [u_0 u_1 u_2 \dots] G$

$$v = uG$$

$$G = \begin{bmatrix} \underline{g_0} & g_1 & \dots & 0 & \dots & g_m & & \\ & g_0 & g_1 & g_2 & \dots & g_m & & \\ & & & g_1 & & & & \\ & & & & g_0 & & & \\ & & & & & & & \dots \end{bmatrix}$$

where  $g_i = (g_i^{(1)} g_i^{(1)} \dots g_i^{(n)}), 0 \leq i \leq m$

Handwritten calculations for  $v$ :

- $v_0 = u_0 g_0$
- $v_1 = u_1 g_0 + u_0 g_1$
- $v_2 = u_2 g_0 + u_1 g_1 + u_0 g_2$
- $v_3 = u_3 g_0 + u_2 g_1 + u_1 g_2$

So you can see here, our generator matrix is of the form, of semi infinite form where basically our  $G$  is something like this, so we have  $g_0$  to  $g_m$ , now this becomes 0, now this is  $u g_0$  and this is  $g_m$  and this 0 0 and this is  $g_0$ , so it is like in this way diagonally my

(Refer Slide Time 27:01)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix

- Matrix form  $[v_0 v_1 v_2 \dots] = v = [u_0 u_1 u_2 \dots] G$

$$v = uG$$

$$G = \begin{bmatrix} \underline{g_0} & g_1 & \dots & 0 & \dots & g_m & & \\ & g_0 & g_1 & g_2 & \dots & g_m & & \\ & & & g_1 & & & & \\ & & & & g_0 & & & \\ & & & & & & & \dots \end{bmatrix}$$

where  $g_i = (g_i^{(1)} g_i^{(1)} \dots g_i^{(n)}), 0 \leq i \leq m$

Handwritten calculations for  $v$ :

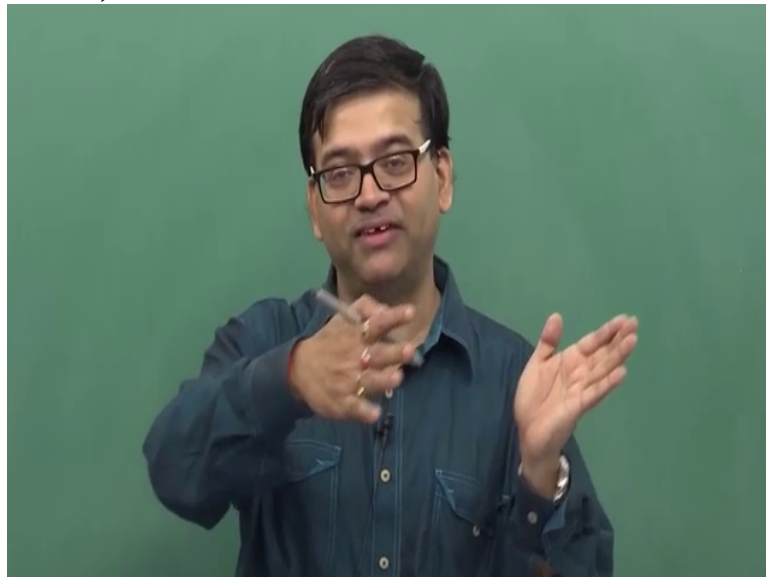
- $v_0 = u_0 g_0$
- $v_1 = u_1 g_0 + u_0 g_1$
- $v_2 = u_2 g_0 + u_1 g_1 + u_0 g_2$
- $v_3 = u_3 g_0 + u_2 g_1 + u_1 g_2$

Diagram showing the shifting of generator sequences:

$$G = \begin{bmatrix} g_0 & - & - & g_m \\ & g_0 & - & - & g_m \\ & & & & & - & - & g_m \\ & & & & & & & & - & - & g_m \\ & & & & & & & & & & & - & - & g_m \end{bmatrix}$$

generator sequence is moving. And that's what I have written here. So if I try to write it in the form of generator matrix then I can, my generator matrix in this case is a semi infinite

(Refer Slide Time 27:14)



form and through this example for a memory 2 code

(Refer Slide Time 27:18)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix

- Matrix form  $[v_0 v_1 v_2 \dots] = \mathbf{v} = \mathbf{uG}$  where  $\mathbf{u} = [u_0 u_1 u_2 \dots]$

$$\mathbf{G} = \begin{bmatrix} \underline{g_0} & g_1 & \dots & 0 & \dots & g_m \\ & g_0 & g_1 & g_2 & \dots & g_m \\ & & \ddots & g_1 & \dots & \dots \\ & & & g_0 & \dots & \dots \end{bmatrix}$$

where  $g_i = (g_i^{(1)} g_i^{(2)} \dots g_i^{(n)}), 0 \leq i \leq m$

Handwritten notes on the slide show the calculation of the output vector  $\mathbf{v}$  as a linear combination of the input vector  $\mathbf{u}$  and the generator matrix  $\mathbf{G}$ :

$$v_0 = u_0 g_0$$

$$v_1 = u_1 g_0 + u_0 g_1$$

$$v_2 = u_2 g_0 + u_1 g_1 + u_0 g_2$$

$$v_3 = u_3 g_0 + u_2 g_1 + u_1 g_2 + u_0 g_3$$

The handwritten  $\mathbf{G}$  matrix is shown with blue annotations indicating the shift of the generator polynomial components.

we showed that this specifically G is of the form like this, Ok. And where each of these g 0's are basically these, will represent what are these n bit output.

So let us





(Refer Slide Time 27:47)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix: Example

- For  $(2, 1, 2)$  convolutional code with  $\mathbf{g}^{(1)} = (1\ 0\ 1)$ , and  $\mathbf{g}^{(2)} = (1\ 1\ 1)$ , the generator matrix is given by

$$\mathbf{G} = \begin{bmatrix} 1\ 1 & 0\ 1 & 1\ 1 & & & \\ & 1\ 1 & 0\ 1 & 1\ 1 & & \\ & & 1\ 1 & 0\ 1 & 1\ 1 & \\ & & & 1\ 1 & 0\ 1 & 1\ 1 \\ & & & & 1\ 1 & 0\ 1 & 1\ 1 \\ & & & & & \ddots & \ddots \end{bmatrix}$$

sequence. For the first code, the sequence is given by 1 0 1, because my output  $v_1$  is  $u_1$  plus  $u_1$  minus 2. Similarly the generator sequence for the second codeword is given by 1 1, Ok. Then can I write basically what is my  $g_0$ ,  $g_1$  and  $g_2$ . So  $g_0$  is given by, now there are 2 outputs so  $g_0$  will have 2 terms, the first term corresponding to the first coded sequence so here this is 1, and what about the second coded sequence that is 1, so  $g_0$  is 1 1,  $g_1$  is this is 0,

(Refer Slide Time 28:37)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix: Example

- For  $(2, 1, 2)$  convolutional code with  $\mathbf{g}^{(1)} = (1\ 0\ 1)$ , and  $\mathbf{g}^{(2)} = (1\ 1\ 1)$ , the generator matrix is given by

$$\mathbf{G} = \begin{bmatrix} 1\ 1 & 0\ 1 & 1\ 1 & & & \\ & 1\ 1 & 0\ 1 & 1\ 1 & & \\ & & 1\ 1 & 0\ 1 & 1\ 1 & \\ & & & 1\ 1 & 0\ 1 & 1\ 1 \\ & & & & 1\ 1 & 0\ 1 & 1\ 1 \\ & & & & & \ddots & \ddots \end{bmatrix}$$

$g_0 = (1\ 1)$   
 $g_1 = 0$   
 $g_2 = 0$

so this is 0, this is 0 and this is 1. So  $g_1$  is 0 1.

(Refer Slide Time 28:44)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix: Example

- For  $(2, 1, 2)$  convolutional code with  $\mathbf{g}^{(1)} = (1 \ 0 \ 1)$ , and  $\mathbf{g}^{(2)} = (1 \ 1 \ 1)$ , the generator matrix is given by

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & & & \\ & 1 & 1 & 0 & 1 & 1 & & & \\ & & 1 & 1 & 0 & 1 & 1 & & \\ & & & 1 & 1 & 0 & 1 & 1 & \\ & & & & 1 & 1 & 0 & 1 & 1 \\ & & & & & 1 & 1 & 0 & 1 & 1 \\ & & & & & & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

Handwritten notes:  $g_0 = (1 \ 1)$ ,  $g_1 = (0 \ 1)$ ,  $g_2 = (1 \ 1)$

What about  $g_2$ ?  $g_2$  is this is 1 and this is 1. So  $g_2$  is 1 1.

(Refer Slide Time 28:53)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix: Example

- For  $(2, 1, 2)$  convolutional code with  $\mathbf{g}^{(1)} = (1 \ 0 \ 1)$ , and  $\mathbf{g}^{(2)} = (1 \ 1 \ 1)$ , the generator matrix is given by

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & & & \\ & 1 & 1 & 0 & 1 & 1 & & & \\ & & 1 & 1 & 0 & 1 & 1 & & \\ & & & 1 & 1 & 0 & 1 & 1 & \\ & & & & 1 & 1 & 0 & 1 & 1 \\ & & & & & 1 & 1 & 0 & 1 & 1 \\ & & & & & & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

Handwritten notes:  $g_0 = (1 \ 1)$ ,  $g_1 = (0 \ 1)$ ,  $g_2 = (1 \ 1)$

So I can then write my generator matrix which is of the form  $e$  is the form  $g_0 \ g_1 \ g_2$  and the rest of all of these are basically zero. These are zero, these are all zero. This is  $g_0$ ,  $g_1$ ,  $g_2$  and then these are all 0s. So what is  $g_0$ ?  $g_0$  is 1 1, that's what I have written here.  $g_1$  is 0 1 that's what I have written here. And  $g_2$  is 1 1, rest all these entries are 0. Similarly this is 0 0, and then I have  $g_0$ ,  $g_1$ ,  $g_2$  and then these are all 0's, Ok. So this is how I can write a

(Refer Slide Time 29:38)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix: Example

- For  $(2, 1, 2)$  convolutional code with  $\mathbf{g}^{(1)} = (1 \ 0 \ 1)$ , and  $\mathbf{g}^{(2)} = (1 \ 1 \ 1)$ , the generator matrix is given by

$$G = \begin{bmatrix} \underline{11} & \underline{01} & \underline{11} & & & & \\ \underline{00} & 11 & 01 & 11 & 00 & \dots & \\ & & 11 & 01 & 11 & & \\ & & & 11 & 01 & 11 & \\ & & & & 11 & 01 & 11 \\ & & & & & \dots & \dots \end{bmatrix}$$

Handwritten notes on the slide:

- $g_0 = (1 \ 1)$
- $g_1 = (0 \ 1)$
- $g_2 = (1 \ 1)$
- Handwritten  $G = \begin{bmatrix} g_0 & g_1 & g_2 & 0 & 0 & \dots \\ g_0 & g_1 & g_2 & 0 & \dots \end{bmatrix}$

generator matrix. Now let's verify

(Refer Slide Time 29:42)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix: Example

- For  $(2, 1, 2)$  convolutional code with  $\mathbf{g}^{(1)} = (1 \ 0 \ 1)$ , and  $\mathbf{g}^{(2)} = (1 \ 1 \ 1)$ , the generator matrix is given by

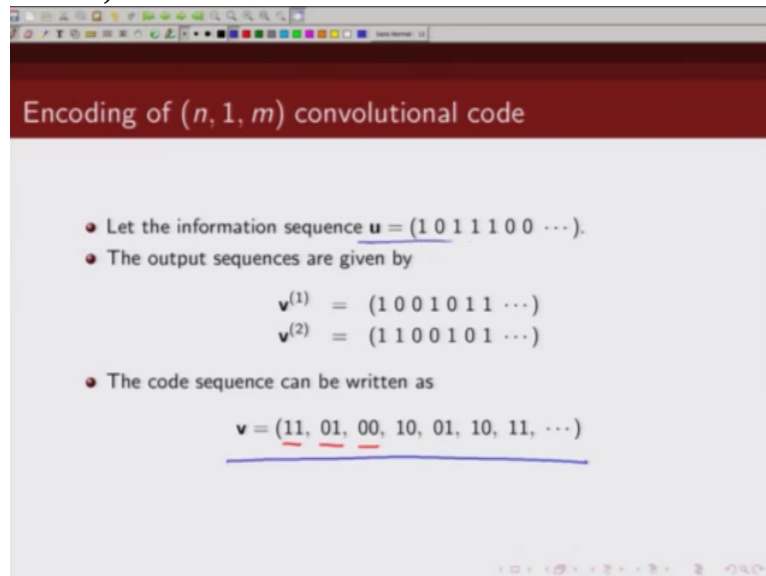
$$G = \begin{bmatrix} 11 & 01 & 11 & & & & \\ & 11 & 01 & 11 & & & \\ & & 11 & 01 & 11 & & \\ & & & 11 & 01 & 11 & \\ & & & & 11 & 01 & 11 \\ & & & & & \dots & \dots \end{bmatrix}$$

- For  $\mathbf{u} = (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ \dots)$ ,

$$\mathbf{v} = \mathbf{uG} = (11, 01, 00, 10, 01, 10, 11, 00, 00, \dots)$$

our coded sequence that we calculated in the last time. Coded sequence corresponding to this information sequence was given by this. This was our coded sequence corresponding to this

(Refer Slide Time 29:57)

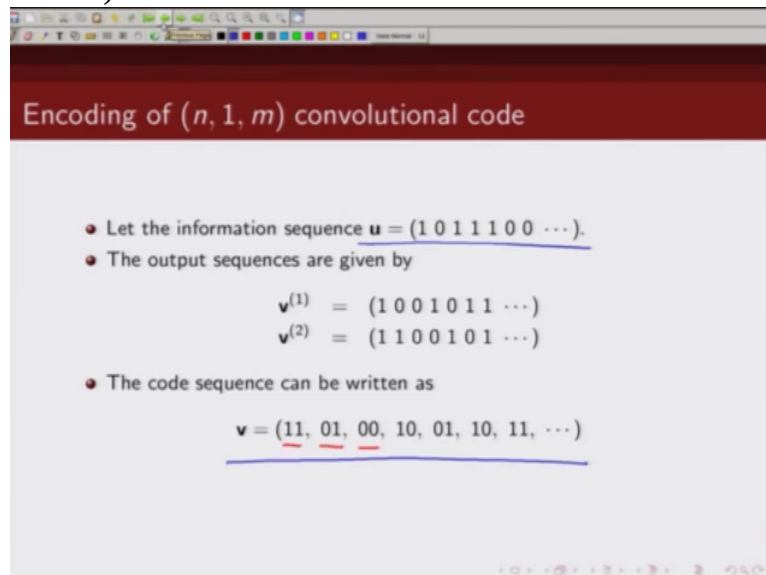


Encoding of  $(n, 1, m)$  convolutional code

- Let the information sequence  $\mathbf{u} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ \dots)$ .
- The output sequences are given by
$$\mathbf{v}^{(1)} = (1\ 0\ 0\ 1\ 0\ 1\ 1\ \dots)$$
$$\mathbf{v}^{(2)} = (1\ 1\ 0\ 0\ 1\ 0\ 1\ \dots)$$
- The code sequence can be written as
$$\mathbf{v} = (\underline{11}, \underline{01}, \underline{00}, 10, 01, 10, 11, \dots)$$

information sequence. Now let's try using this generator matrix. So if we use the generator matrix then our first input is 1 so 1 times  $g_0$ , that is this. Then next is input is zero so 1 times this plus zero times  $g_1$  that will be 0 1. Next 1 0 1 so 1 times  $g_1$ , 0 times this, 1 times this so that is 0 0. So we can see basically we are getting the same output sequence. We can verify 1 0 1 0 0, 1 1 0 1 0 0 so we are getting the same output sequence

(Refer Slide Time 30:39)



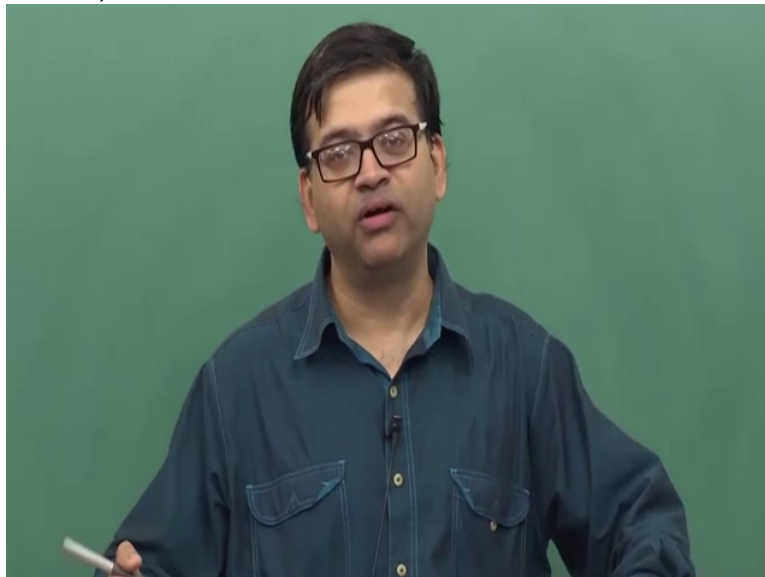
Encoding of  $(n, 1, m)$  convolutional code

- Let the information sequence  $\mathbf{u} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ \dots)$ .
- The output sequences are given by
$$\mathbf{v}^{(1)} = (1\ 0\ 0\ 1\ 0\ 1\ 1\ \dots)$$
$$\mathbf{v}^{(2)} = (1\ 1\ 0\ 0\ 1\ 0\ 1\ \dots)$$
- The code sequence can be written as
$$\mathbf{v} = (\underline{11}, \underline{01}, \underline{00}, 10, 01, 10, 11, \dots)$$

as before.



(Refer Slide Time 30:53)



this generator sequence which is very convenient in case of convolutional codes. So I am introducing a delay operator

(Refer Slide Time 31:02)

Encoding of  $(n, 1, m)$  convolutional code

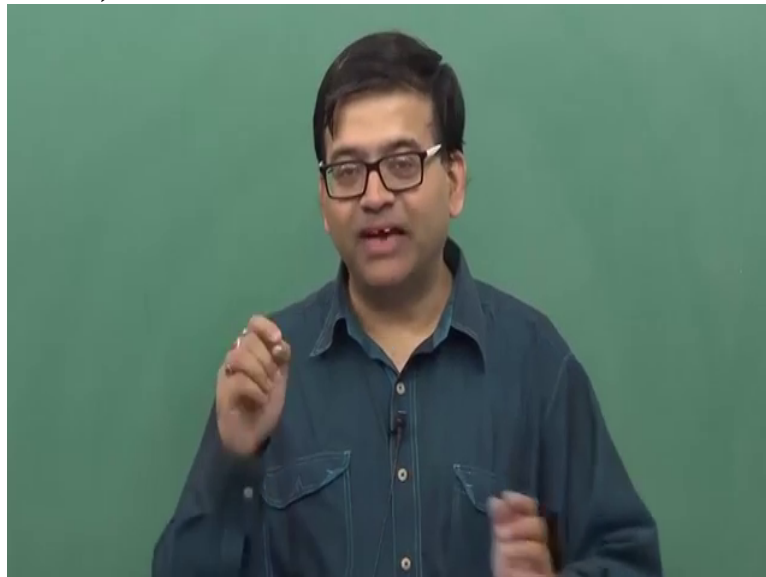
- Polynomial representation:
 
$$\mathbf{v}^{(i)}(D) = \mathbf{u}(D)\mathbf{g}^{(i)}(D), \quad 1 \leq i \leq n$$

$$\mathbf{v}(D) = \mathbf{v}^{(1)}(D^n) + D\mathbf{v}^{(2)}(D^n) + \dots + D^{n-1}\mathbf{v}^{(n)}(D^n)$$

Time Domain	Transform Domain
$\mathbf{g}^{(1)} = (1 \ 0 \ 1)$	$\mathbf{g}^{(1)}(D) = 1 + D^2$
$\mathbf{g}^{(2)} = (1 \ 1 \ 1)$	$\mathbf{g}^{(2)}(D) = 1 + D + D^2$
$\mathbf{u} = (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ \dots)$	$\mathbf{u}(D) = 1 + D^2 + D^3 + D^4$
$\mathbf{v}^{(1)} = \mathbf{u} * \mathbf{g}^{(1)}$	$\mathbf{v}^{(1)}(D) = \mathbf{u}(D)\mathbf{g}^{(1)}(D)$
$\quad = (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ \dots)$	$\quad = 1 + D^3 + D^5 + D^6$
$\mathbf{v}^{(2)} = \mathbf{u} * \mathbf{g}^{(2)}$	$\mathbf{v}^{(2)}(D) = \mathbf{u}(D)\mathbf{g}^{(2)}(D)$
$\quad = (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ \dots)$	$\quad = 1 + D + D^4 + D^6$

D. So if you have 1 memory element delay that will be D.

(Refer Slide Time 31:07)



If you have delay of 2,

(Refer Slide Time 31:10)

Encoding of  $(n, 1, m)$  convolutional code

- Polynomial representation:
 
$$\mathbf{v}^{(i)}(D) = \mathbf{u}(D)\mathbf{g}^{(i)}(D), \quad 1 \leq i \leq n$$

$$\mathbf{v}(D) = \mathbf{v}^{(1)}(D^n) + D\mathbf{v}^{(2)}(D^n) + \dots + D^{n-1}\mathbf{v}^{(n)}(D^n)$$

Time Domain	Transform Domain
$\mathbf{g}^{(1)} = (1 \ 0 \ 1)$	$\mathbf{g}^{(1)}(D) = 1 + D^2$
$\mathbf{g}^{(2)} = (1 \ 1 \ 1)$	$\mathbf{g}^{(2)}(D) = 1 + D + D^2$
$\mathbf{u} = (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ \dots)$	$\mathbf{u}(D) = 1 + D^2 + D^3 + D^4$
$\mathbf{v}^{(1)} = \mathbf{u} * \mathbf{g}^{(1)}$	$\mathbf{v}^{(1)}(D) = \mathbf{u}(D)\mathbf{g}^{(1)}(D)$
$\quad = (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ \dots)$	$\quad = 1 + D^3 + D^5 + D^6$
$\mathbf{v}^{(2)} = \mathbf{u} * \mathbf{g}^{(2)}$	$\mathbf{v}^{(2)}(D) = \mathbf{u}(D)\mathbf{g}^{(2)}(D)$
$\quad = (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ \dots)$	$\quad = 1 + D + D^4 + D^6$

it will be D square. If you have delay of 3, it will be D cube. So the exponent of D is going to specify how much delay, Ok. so what I am going to show you



(Refer Slide Time 31:23)

**Encoding of  $(n, 1, m)$  convolutional code**

- Polynomial representation:
 
$$\mathbf{v}^{(i)}(D) = \mathbf{u}(D)\mathbf{g}^{(i)}(D), \quad 1 \leq i \leq n$$

$$\mathbf{v}(D) = \mathbf{v}^{(1)}(D^n) + D\mathbf{v}^{(2)}(D^n) + \dots + D^{n-1}\mathbf{v}^{(n)}(D^n)$$

Time Domain	Transform Domain
$\mathbf{g}^{(1)} = (1 \ 0 \ 1)$	$\mathbf{g}^{(1)}(D) = 1 + D^2$
$\mathbf{g}^{(2)} = (1 \ 1 \ 1)$	$\mathbf{g}^{(2)}(D) = 1 + D + D^2$
$\mathbf{u} = (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ \dots)$	$\mathbf{u}(D) = 1 + D^2 + D^3 + D^4$
$\mathbf{v}^{(1)} = \mathbf{u} * \mathbf{g}^{(1)}$	$\mathbf{v}^{(1)}(D) = \mathbf{u}(D)\mathbf{g}^{(1)}(D)$
$= (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ \dots)$	$= 1 + D^3 + D^5 + D^6$
$\mathbf{v}^{(2)} = \mathbf{u} * \mathbf{g}^{(2)}$	$\mathbf{v}^{(2)}(D) = \mathbf{u}(D)\mathbf{g}^{(2)}(D)$
$= (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ \dots)$	$= 1 + D + D^4 + D^6$

is that I can write my output sequence in this polynomial notation as  $\mathbf{u}$  times  $D$  into  $g_i$  times  $D$ . So every output code sequence can be represented as product of this information sequence using this delay operator multiplied by this generator sequence in this delay operator framework and the overall code sequence when we have rate 1 by  $n$  code can be given by this expression. So let's first try to write each of these terms in terms of delay operator polynomial representation and then we will show that this time domain representation where we were computing the output during convolution, discrete convolution can be similarly obtained using just this operation in the delay domain which we are calling transform domain operation. So we will take the same example that we were considering. So I will just go back and show you again the convolutional encoder. So this is the convolutional encoder that we are considering.

(Refer Slide Time 32:31)

Encoding of  $(n, 1, m)$  convolutional code

Example:

- Consider a rate  $R = 1/2$ ,  $(2, 1, 2)$  convolutional code specified by the following generator sequences

$$\mathbf{g}^{(1)} = (1 \ 0 \ 1),$$

$$\mathbf{g}^{(2)} = (1 \ 1 \ 1),$$

Handwritten notes:  $v_i^{(1)} = u_i + u_{i-2}$ ,  $v_i^{(2)} = u_i + u_{i-1} + u_{i-2}$

We have 1 input, we have 2 outputs. Output depends on past 2 inputs, so basically memory order is 2.  $g_1$  is given by this;  $g_2$  is given by this. These are my output,  $v_1^{(1)}$ ,  $v_1^{(2)}$  these are my output sequences, Ok. So let's look at

(Refer Slide Time 32:51)

Encoding of  $(n, 1, m)$  convolutional code

- Polynomial representation:

$$\mathbf{v}^{(i)}(D) = \mathbf{u}(D)\mathbf{g}^{(i)}(D), \quad 1 \leq i \leq n$$

$$\mathbf{v}(D) = \mathbf{v}^{(1)}(D^n) + D\mathbf{v}^{(2)}(D^n) + \dots + D^{n-1}\mathbf{v}^{(n)}(D^n)$$

Time Domain	Transform Domain
$\mathbf{g}^{(1)} = (1 \ 0 \ 1)$	$\mathbf{g}^{(1)}(D) = 1 + D^2$
$\mathbf{g}^{(2)} = (1 \ 1 \ 1)$	$\mathbf{g}^{(2)}(D) = 1 + D + D^2$
$\mathbf{u} = (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ \dots)$	$\mathbf{u}(D) = 1 + D^2 + D^3 + D^4$
$\mathbf{v}^{(1)} = \mathbf{u} * \mathbf{g}^{(1)}$	$\mathbf{v}^{(1)}(D) = \mathbf{u}(D)\mathbf{g}^{(1)}(D)$
$= (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ \dots)$	$= 1 + D^3 + D^5 + D^6$
$\mathbf{v}^{(2)} = \mathbf{u} * \mathbf{g}^{(2)}$	$\mathbf{v}^{(2)}(D) = \mathbf{u}(D)\mathbf{g}^{(2)}(D)$
$= (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ \dots)$	$= 1 + D + D^4 + D^6$

Handwritten notes:  $D^2$ ,  $D^3$

this. So  $g_1$  is 1 0 1. Now what does this 1 corresponds to? 1 corresponds to this connection  $g_0$  which is linking my input  $u_1$ , so that would be  $u_1$  without any delay. So that would be 1. What was this? This corresponds to  $g_1$ , that is input delayed by 1. So this will be represented using  $D$  so  $D$  times 0 will be 0 and this will be, this will correspond to  $g_2$  basically and

(Refer Slide Time 33:29)

**Encoding of  $(n, 1, m)$  convolutional code**

- Polynomial representation:  $D^2 \quad D^3$

$$\underline{v^{(i)}(D)} = \underline{u(D)g^{(i)}(D)}, \quad 1 \leq i \leq n$$

$$\underline{v(D)} = \underline{v^{(1)}(D^n) + Dv^{(2)}(D^n) + \dots + D^{n-1}v^{(n)}(D^n)}$$

Time Domain	Transform Domain
$\underline{g^{(1)}} = (1 \ 0 \ 1)$	$\underline{g^{(1)}(D)} = 1 + D^2$
$\underline{g^{(2)}} = (1 \ 1 \ 1)$	$\underline{g^{(2)}(D)} = 1 + D + D^2$
$\underline{u} = (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ \dots)$	$\underline{u(D)} = 1 + D^2 + D^3 + D^4$
$\underline{v^{(1)}} = \underline{u * g^{(1)}}$	$\underline{v^{(1)}(D)} = \underline{u(D)g^{(1)}(D)}$
$\quad = (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ \dots)$	$\quad = 1 + D^3 + D^5 + D^6$
$\underline{v^{(2)}} = \underline{u * g^{(2)}}$	$\underline{v^{(2)}(D)} = \underline{u(D)g^{(2)}(D)}$
$\quad = (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ \dots)$	$\quad = 1 + D + D^4 + D^6$

this is delay of 2, so this will be represented using D square. So this g 1 in this transformed domain using this delay operator can be written as 1 plus D square. Similarly this g 2 which is 1 1 1 can be written as 1 plus D plus D square. So this is my g 2 of D. Now information sequence also I can write in this delay notation, it is u 0, u 1, u 2, u 3. So this is information sequence I am getting at this time, this after 1 delay element, 2, 3, 4 so this would be 1 plus D square plus D cube plus D four and that's basically my information sequence. Now the discrete convolution of information sequence is g 1 is basically given by this and this if I write in uh delay operator form, will be what, 1 plus D D square D cube, D cube plus D four D five plus D six. And what is u D? u D is given by this;

(Refer Slide Time 34:52)

**Encoding of  $(n, 1, m)$  convolutional code**

- Polynomial representation:  $D^2 \quad D^3$

$$\underline{v^{(i)}(D)} = \underline{u(D)g^{(i)}(D)}, \quad 1 \leq i \leq n$$

$$\underline{v(D)} = \underline{v^{(1)}(D^n) + Dv^{(2)}(D^n) + \dots + D^{n-1}v^{(n)}(D^n)}$$

Time Domain	Transform Domain
$\underline{g^{(1)}} = (1 \ 0 \ 1)$	$\underline{g^{(1)}(D)} = 1 + D^2$
$\underline{g^{(2)}} = (1 \ 1 \ 1)$	$\underline{g^{(2)}(D)} = 1 + D + D^2$
$\underline{u} = (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ \dots)$	$\underline{u(D)} = 1 + D^2 + D^3 + D^4$
$\underline{v^{(1)}} = \underline{u * g^{(1)}}$	$\underline{v^{(1)}(D)} = \underline{u(D)g^{(1)}(D)}$
$\quad = (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ \dots)$	$\quad = 1 + D^3 + D^5 + D^6$
$\underline{v^{(2)}} = \underline{u * g^{(2)}}$	$\underline{v^{(2)}(D)} = \underline{u(D)g^{(2)}(D)}$
$\quad = (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ \dots)$	$\quad = 1 + D + D^4 + D^6$

*Handwritten annotations:*  
 - In Time Domain:  $1 + D^3 + D^5 + D^6$  is written next to  $v^{(1)}$ .  
 - In Transform Domain:  $1 + D^3 + D^5 + D^6$  is written next to  $v^{(1)}(D)$ .

g 1 D is given by this. So let's multiply these 2. So what do we get? So if you multiply u D by g 1 of D so one times this will be 1 plus D square plus D cube plus D four plus D square times 1 is D square, this will be D four, this will be D five and this will be D six. D square plus D square is zero, D four plus D four is zero, so what we are left with, 1 plus D cube plus D five

(Refer Slide Time 35:29)

**Encoding of  $(n, 1, m)$  convolutional code**

- Polynomial representation:
 
$$\mathbf{v}^{(i)}(D) = \mathbf{u}(D)\mathbf{g}^{(i)}(D), \quad 1 \leq i \leq n$$

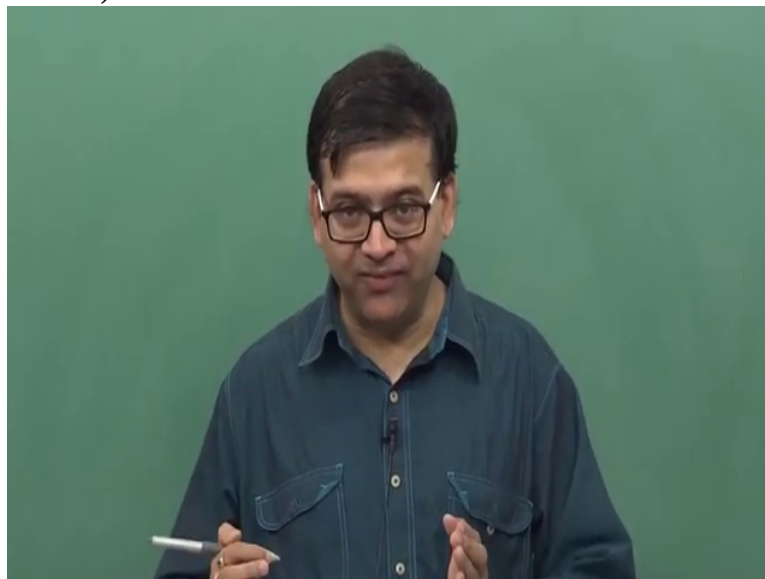
$$\mathbf{v}(D) = \mathbf{v}^{(1)}(D^n) + D\mathbf{v}^{(2)}(D^n) + \dots + D^{n-1}\mathbf{v}^{(n)}(D^n)$$

Time Domain	Transform Domain
$\mathbf{g}^{(1)} = (1 \ 0 \ 1)$	$\mathbf{g}^{(1)}(D) = 1 + D^2$
$\mathbf{g}^{(2)} = (1 \ 1 \ 1)$	$\mathbf{g}^{(2)}(D) = 1 + D + D^2$
$\mathbf{u} = (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ \dots)$	$\mathbf{u}(D) = 1 + D^2 + D^3 + D^4$
$\mathbf{v}^{(1)} = \mathbf{u} * \mathbf{g}^{(1)}$	$\mathbf{v}^{(1)}(D) = \mathbf{u}(D)\mathbf{g}^{(1)}(D)$
$= (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ \dots)$	$= 1 + D^3 + D^5 + D^6$
$\mathbf{v}^{(2)} = \mathbf{u} * \mathbf{g}^{(2)}$	$\mathbf{v}^{(2)}(D) = \mathbf{u}(D)\mathbf{g}^{(2)}(D)$
$= (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ \dots)$	$= 1 + D + D^4 + D^6$

plus D six. This is precisely what you get here, Ok. So you can see basically these 2 representations is equivalent. Similarly we can write u 2 which is given by this and you can verify for yourself that u 2 D is given by this.

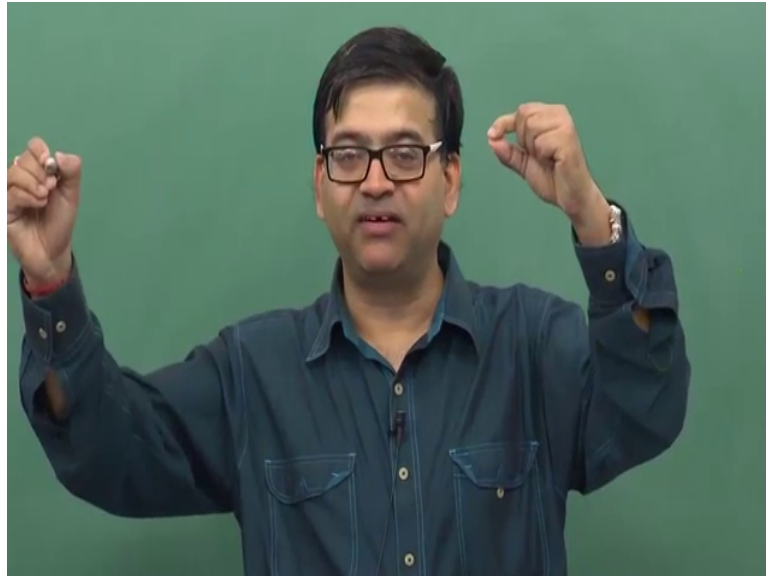
Now once you have these individual sequence,

(Refer Slide Time 35:53)



how do you write the overall output sequence? So note that for 1 input sequence you are getting  $n$  outputs, Ok. So is taken care by this. So if  $v_i D$  is going to give me output sequence corresponding to each of these output,  $n$  output outputs, now I can combine  $n$  outputs in this particular fashion. So I take the first output, note that I have made here  $D$  to the power  $n$  because if it is a rate  $1/n$  code the first output will appear after every  $n$  bits,

(Refer Slide Time 36:33)



the first bit is from the first coded sequence. Then after  $n$  bits, it will again repeat. It will come, meaning. So that's why I have made it  $D^n$ .

Now how do I combine these  $n$  sequences? So note, this is the output,  $v_1 D^n$  is the output from the first coded sequence. This is the output from the second coded sequence. That's delayed by  $D$ . The output from third sequence is delayed by  $D^2$ . Similarly the output from the  $n$ th sequence will be delayed by  $n-1$ . Go back here. These are the 2 individual outputs. How are we getting

(Refer Slide Time 37:15)

Encoding of  $(n, 1, m)$  convolutional code

- Let the information sequence  $\mathbf{u} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ \dots)$ .
- The output sequences are given by
$$\mathbf{v}^{(1)} = (1\ 0\ 0\ 1\ 0\ 1\ 1\ \dots) \quad \checkmark$$
$$\mathbf{v}^{(2)} = (1\ 1\ 0\ 0\ 1\ 0\ 1\ \dots) \quad \checkmark$$
- The code sequence can be written as
$$\mathbf{v} = (\underline{11}, \underline{01}, \underline{00}, 10, 01, 10, 11, \dots)$$

the final output? So note here. I am taking first from here. That is 1; second bit I am taking from here,

(Refer Slide Time 37:24)

Encoding of  $(n, 1, m)$  convolutional code

- Let the information sequence  $\mathbf{u} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ \dots)$ .
- The output sequences are given by
$$\mathbf{v}^{(1)} = (1\ 0\ 0\ 1\ 0\ 1\ 1\ \dots) \quad \checkmark$$
$$\mathbf{v}^{(2)} = (1\ 1\ 0\ 0\ 1\ 0\ 1\ \dots) \quad \checkmark$$
- The code sequence can be written as
$$\mathbf{v} = (\underline{11}, \underline{01}, \underline{00}, 10, 01, 10, 11, \dots)$$

that is this. The third bit is this, which is this. Fourth bit is this which is this, so what

(Refer Slide Time 37:33)

Encoding of  $(n, 1, m)$  convolutional code

- Let the information sequence  $\mathbf{u} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ \dots)$ .
- The output sequences are given by
 
$$\mathbf{v}^{(1)} = (1\ 0\ 0\ 1\ 0\ 1\ 1\ \dots) \quad \checkmark$$

$$\mathbf{v}^{(2)} = (1\ 1\ 0\ 0\ 1\ 0\ 1\ \dots) \quad \checkmark$$
- The code sequence can be written as
 
$$\mathbf{v} = (11\ 01\ 00, 10, 01, 10, 11, \dots)$$

am I doing? In this case rate was one half so after every, you can see

(Refer Slide Time 37:40)

Encoding of  $(n, 1, m)$  convolutional code

- Let the information sequence  $\mathbf{u} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ \dots)$ .
- The output sequences are given by
 
$$\mathbf{v}^{(1)} = (1\ 0\ 0\ 1\ 0\ 1\ 1\ \dots) \quad \checkmark$$

$$\mathbf{v}^{(2)} = (1\ 1\ 0\ 0\ 1\ 0\ 1\ \dots) \quad \checkmark$$
- The code sequence can be written as
 
$$\mathbf{v} = (11\ 01\ 00, 10, 01, 10, 11, \dots)$$

$R = \frac{1}{2}$

the output; every second bit is coming from this. So this is my 1 which is appearing here. This is my zero which is this. This is my zero which is this. This is my 1 which is appearing this. So note this is appearing every second bit and that's why what we did was, when we combined we made it, each of these coded bit, we made it  $D$  raised to power  $n$ . Next

(Refer Slide Time 38:08)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix: Example

- For  $(2, 1, 2)$  convolutional code with  $\mathbf{g}^{(1)} = (1\ 0\ 1)$ , and  $\mathbf{g}^{(2)} = (1\ 1\ 1)$ , the generator matrix is given by

$$\mathbf{G} = \begin{bmatrix} 1\ 1 & 0\ 1 & 1\ 1 & & & \\ & 1\ 1 & 0\ 1 & 1\ 1 & & \\ & & 1\ 1 & 0\ 1 & 1\ 1 & \\ & & & 1\ 1 & 0\ 1 & 1\ 1 \\ & & & & 1\ 1 & 0\ 1 & 1\ 1 \\ & & & & & \ddots & \ddots \end{bmatrix}$$

- For  $\mathbf{u} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ \dots)$ ,

$$\mathbf{v} = \mathbf{uG} = (\underline{11}, \underline{01}, \underline{00}, 10, 01, 10, 11, 00, 00, \dots)$$

(Refer Slide Time 38:11)

Encoding of  $(n, 1, m)$  convolutional code

Representation of encoding matrix: Example

- For  $(2, 1, 2)$  convolutional code with  $\mathbf{g}^{(1)} = (1\ 0\ 1)$ , and  $\mathbf{g}^{(2)} = (1\ 1\ 1)$ , the generator matrix is given by

$$\mathbf{G} = \begin{bmatrix} \underline{1\ 1} & \underline{0\ 1} & \underline{1\ 1} & & & \\ \underline{0\ 0} & 1\ 1 & 0\ 1 & 1\ 1 & 0\ 0 & \dots \\ & & 1\ 1 & 0\ 1 & 1\ 1 & \\ & & & 1\ 1 & 0\ 1 & 1\ 1 \\ & & & & 1\ 1 & 0\ 1 & 1\ 1 \\ & & & & & \ddots & \ddots \end{bmatrix}$$

Handwritten notes:

- $g_0 = (1\ 1)$
- $g_1 = (0\ 1)$
- $g_2 = (1\ 1)$

$$\mathbf{G} = \begin{bmatrix} g_0 & g_1 & g_2 & 0\ 0 & \dots \\ g_0 & g_1 & g_2 & 0 & \dots \end{bmatrix}$$

if you look here, the first coded sequence is this one. This is the output from the first coded sequence. And what is the output from second sequence, which is this one. So what are you doing?



(Refer Slide Time 38:30)

Encoding of  $(n, 1, m)$  convolutional code

- Let the information sequence  $u = (1 0 1 1 1 0 0 \dots)$ .
- The output sequences are given by  $R = \frac{1}{2}$ 

$$v^{(1)} = (1 0 0 1 0 1 1 \dots)$$

$$v^{(2)} = (1 1 0 0 1 0 1 \dots)$$
- The code sequence can be written as
 
$$v = (11 01 00 10 01 10 11, \dots)$$

When you are combining these output sequence which is  $v_1$  and  $v_2$ , so you are taking  $v_1$  like as it is, only thing is it is spread out after every second bit and  $v_2$  is delayed by 1 and it is also spread out, this is 1, this 1 is appearing here, this zero is appearing here, this zero is appearing here. So every second bit is also

(Refer Slide Time 38:56)

Encoding of  $(n, 1, m)$  convolutional code

- Let the information sequence  $u = (1 0 1 1 1 0 0 \dots)$ .
- The output sequences are given by  $R = \frac{1}{2}$ 

$$v^{(1)} = (1 0 0 1 0 1 1 \dots)$$

$$v^{(2)} = (1 1 0 0 1 0 1 \dots)$$
- The code sequence can be written as
 
$$v = (11 01 00 10 01 10 11, \dots)$$

from this encoded sequence and note that this is delayed by one corresponding to  $v_1$ . So that's what we are doing here.

(Refer Slide Time 39:06)

**Encoding of  $(n, 1, m)$  convolutional code**

- Polynomial representation:
 
$$v^{(i)}(D) = u(D)g^{(i)}(D), \quad 1 \leq i \leq n$$

$$v(D) = v^{(1)}(D^n) + Dv^{(2)}(D^n) + \dots + D^{n-1}v^{(n)}(D^n)$$

<p>Time Domain</p> $g^{(1)} = (1 \ 0 \ 1)$ $g^{(2)} = (1 \ 1 \ 1)$ $u = (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ \dots)$ $v^{(1)} = u * g^{(1)}$ $= (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ \dots)$ $v^{(2)} = u * g^{(2)}$ $= (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ \dots)$	<p>Transform Domain</p> $g^{(1)}(D) = 1 + D^2$ $g^{(2)}(D) = 1 + D + D^2$ $u(D) = 1 + D^2 + D^3 + D^4$ $v^{(1)}(D) = u(D)g^{(1)}(D)$ $= 1 + D^3 + D^5 + D^6$ $v^{(2)}(D) = u(D)g^{(2)}(D)$ $= 1 + D + D^4 + D^6$
--	--

*Handwritten annotations:  $D^2 \ D^3$  at top right;  $1+D^3+D^5+D^6$  next to  $v^{(1)}$ ;  $1+D+D^4+D^6$  next to  $v^{(2)}$ ; arrows pointing to  $D^n$  terms in the polynomial representation.*

If you combine this, consider this combined output sequence there are n coded sequence, v 1, v 2, v 3, v n. Now first sequence we just take v 1 D to power n, second is v 2 D to power n, third is v 3 D to power n and then each one of them are delayed by 1 1. So this no delay, this is delay of 1, delay of 2, this is delay of n minus 1. So overall code sequence will be given by this expression. So I hope I made it clear

(Refer Slide Time 39:41)

**Encoding of  $(n, 1, m)$  convolutional code**

- Polynomial representation:
 
$$v^{(i)}(D) = u(D)g^{(i)}(D), \quad 1 \leq i \leq n$$

$$v(D) = v^{(1)}(D^n) + Dv^{(2)}(D^n) + \dots + D^{n-1}v^{(n)}(D^n)$$

<p>Time Domain</p> $g^{(1)} = (1 \ 0 \ 1)$ $g^{(2)} = (1 \ 1 \ 1)$ $u = (1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ \dots)$ $v^{(1)} = u * g^{(1)}$ $= (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ \dots)$ $v^{(2)} = u * g^{(2)}$ $= (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ \dots)$	<p>Transform Domain</p> $g^{(1)}(D) = 1 + D^2$ $g^{(2)}(D) = 1 + D + D^2$ $u(D) = 1 + D^2 + D^3 + D^4$ $v^{(1)}(D) = u(D)g^{(1)}(D)$ $= 1 + D^3 + D^5 + D^6$ $v^{(2)}(D) = u(D)g^{(2)}(D)$ $= 1 + D + D^4 + D^6$
--	--

*Handwritten annotations:  $D^2 \ D^3$  at top right;  $1+D^3+D^5+D^6$  next to  $v^{(1)}$ ;  $1+D+D^4+D^6$  next to  $v^{(2)}$ ; a red box around the  $v(D)$  equation; arrows pointing to  $D^n$  terms.*

why this is D raised to power n and why each of the parity bits are delayed by 1 power like D, D square, D cube

(Refer Slide Time 39:51)

**Encoding of  $(n, 1, m)$  convolutional code**

- Polynomial representation:
 
$$v^{(i)}(D) = u(D)g^{(i)}(D), \quad 1 \leq i \leq n$$

$$v(D) = v^{(1)}(D^n) + Dv^{(2)}(D^n) + \dots + D^{n-1}v^{(n)}(D^n)$$

<p>Time Domain</p> $g^{(1)} = (1 \ 0 \ 1)$ $g^{(2)} = (1 \ 1 \ 1)$ $u = (1 \ 0 \ 1 \ 1 \ 0 \ 0 \ \dots)$ $v^{(1)} = u * g^{(1)}$ $= (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ \dots)$ $v^{(2)} = u * g^{(2)}$ $= (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ \dots)$	<p>Transform Domain</p> $g^{(1)}(D) = 1 + D^2$ $g^{(2)}(D) = 1 + D + D^2$ $u(D) = 1 + D^2 + D^3 + D^4$ $v^{(1)}(D) = u(D)g^{(1)}(D)$ $= 1 + D^3 + D^5 + D^6$ $v^{(2)}(D) = u(D)g^{(2)}(D)$ $= 1 + D + D^4 + D^6$
--	--

$D^{n-1}$ . So following this basically we can

(Refer Slide Time 39:55)

**Encoding of  $(n, 1, m)$  convolutional code**

- The encoding equations can alternately written as,
 
$$v(D) = u(D^n)g(D)$$
- where
 
$$g(D) \triangleq g^{(1)}(D^n) + Dg^{(2)}(D^n) + \dots + D^{n-1}g^{(n)}(D^n)$$

also write our encoding sequence in this particular form where output sequence is given by  $u(D^n)g(D)$  where  $g(D)$  is, this is generator sequence for the first coded sequence, the generator sequence for the second delayed by 1, generated sequence of the third delayed by 2, generated sequence of the  $n$  delayed by  $D^{n-1}$ . So the overall encoding sequence can be equivalently written like this.

(Refer Slide Time 40:26)

Encoding of  $(n, 1, m)$  convolutional code

- The encoding equations can alternately written as,
 
$$\mathbf{v}(D) = \mathbf{u}(D^n)\mathbf{g}(D)$$
- where
 
$$\mathbf{g}(D) \triangleq \mathbf{g}^{(1)}(D^n) + D\mathbf{g}^{(2)}(D^n) + \dots + D^{n-1}\mathbf{g}^{(n)}(D^n)$$
- Example: Time Domain
 
$$\mathbf{v} = \mathbf{u}\mathbf{G} = (11, 01, 00, 10, 01, 10, 11, 00, 00, \dots)$$

And we can again go back to the same example. Our output sequence in the time domain was given by this. And if we follow the same procedure,

(Refer Slide Time 40:36)

Encoding of  $(n, 1, m)$  convolutional code

- The encoding equations can alternately written as,
 
$$\mathbf{v}(D) = \mathbf{u}(D^n)\mathbf{g}(D)$$
- where
 
$$\mathbf{g}(D) \triangleq \mathbf{g}^{(1)}(D^n) + D\mathbf{g}^{(2)}(D^n) + \dots + D^{n-1}\mathbf{g}^{(n)}(D^n)$$
- Example: Time Domain
 
$$\mathbf{v} = \mathbf{u}\mathbf{G} = (11, 01, 00, 10, 01, 10, 11, 00, 00, \dots)$$
- Example: Transform Domain
 
$$\begin{aligned} \mathbf{v}(D) &= \mathbf{u}(D^2)\mathbf{g}(D) = (1 + D^4 + D^6 + D^8)(1 + D + D^3 + D^4 + D^5) \\ &= 1 + D + D^3 + D^6 + D^9 + D^{10} + D^{12} + D^{13} \end{aligned}$$

$\mathbf{v}$  of  $D$  should be  $\mathbf{u}$  of  $D$  square times  $\mathbf{g}$   $D$  where  $\mathbf{g}$   $D$  is  $\mathbf{g}$   $1$   $D$  square plus  $\mathbf{g}$   $D$   $2$   $D$  square. So  $\mathbf{u}$   $D$   $D$  square is, what is  $\mathbf{u}$   $D$ ? What is  $\mathbf{u}$   $D$ ? Go back to the example.  $\mathbf{u}$   $D$  is  $1$  plus  $D$  square plus  $D$  cube plus  $D$  four.

(Refer Slide Time 41:01)

**Encoding of  $(n, 1, m)$  convolutional code**

- Polynomial representation:
 
$$\mathbf{v}^{(i)}(D) = \mathbf{u}(D)\mathbf{g}^{(i)}(D), \quad 1 \leq i \leq n$$

$$\mathbf{v}(D) = \mathbf{v}^{(1)}(D^n) + D\mathbf{v}^{(2)}(D^n) + \dots + D^{n-1}\mathbf{v}^{(n)}(D^n)$$

<p>Time Domain</p> $\mathbf{g}^{(1)} = (1 \ 0 \ 1)$ $\mathbf{g}^{(2)} = (1 \ 1 \ 1)$ $\mathbf{u} = (1 \ 0 \ 1 \ 1 \ 0 \ 0 \ \dots)$ $\mathbf{v}^{(1)} = \mathbf{u} * \mathbf{g}^{(1)}$ $= (1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ \dots)$ $\mathbf{v}^{(2)} = \mathbf{u} * \mathbf{g}^{(2)}$ $= (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ \dots)$	<p>Transform Domain</p> $\mathbf{g}^{(1)}(D) = 1 + D^2$ $\mathbf{g}^{(2)}(D) = 1 + D + D^2$ $\mathbf{u}(D) = 1 + D^2 + D^3 + D^4$ $\mathbf{v}^{(1)}(D) = \mathbf{u}(D)\mathbf{g}^{(1)}(D)$ $= 1 + D^3 + D^5 + D^6$ $\mathbf{v}^{(2)}(D) = \mathbf{u}(D)\mathbf{g}^{(2)}(D)$ $= 1 + D + D^4 + D^6$
---	---

So this is,

(Refer Slide Time 41:06)

**Encoding of  $(n, 1, m)$  convolutional code**

- The encoding equations can alternately written as,
 
$$\mathbf{v}(D) = \mathbf{u}(D^n)\mathbf{g}(D)$$
- where
 
$$\mathbf{g}(D) \triangleq \mathbf{g}^{(1)}(D^n) + D\mathbf{g}^{(2)}(D^n) + \dots + D^{n-1}\mathbf{g}^{(n)}(D^n)$$
- Example: Time Domain
 
$$\mathbf{v} = \mathbf{u}\mathbf{G} = (11, 01, 00, 10, 01, 10, 11, 00, 00, \dots)$$
- Example: Transform Domain
 
$$\mathbf{v}(D) = \mathbf{u}(D^2)\mathbf{g}(D) = (1 + D^4 + D^6 + D^8)(1 + D + D^3 + D^4 + D^5)$$

$$= 1 + D + D^3 + D^6 + D^9 + D^{10} + D^{12} + D^{13}$$

u D is 1 plus D square plus D cube plus D four. So u D square will be 1 plus D four plus D six plus D eight, so that's what we have written

(Refer Slide Time 41:22)

Encoding of  $(n, 1, m)$  convolutional code

- The encoding equations can alternately written as,  $v(D) = u(D^n)g(D)$ 

$$v(D) = u(D^n)g(D)$$

where

$$g(D) \triangleq g^{(1)}(D^n) + Dg^{(2)}(D^n) + \dots + D^{n-1}g^{(n)}(D^n)$$

- Example: Time Domain
 
$$\mathbf{v} = \mathbf{uG} = (11, 01, 00, 10, 01, 10, 11, 00, 00, \dots)$$
- Example: Transform Domain
 
$$v(D) = u(D^2)g(D) = (1 + D^4 + D^6 + D^8)(1 + D + D^3 + D^4 + D^5)$$

$$= 1 + D + D^3 + D^6 + D^9 + D^{10} + D^{12} + D^{13}$$

here, Ok and what is  $g(D)$ ?  $g(D)$  is  $g^{(1)}(D^2) + Dg^{(2)}(D^2) + \dots + D^{n-1}g^{(n)}(D^2)$  and what is

(Refer Slide Time 41:40)

Encoding of  $(n, 1, m)$  convolutional code

- The encoding equations can alternately written as,  $v(D) = u(D^n)g(D)$ 

$$v(D) = u(D^n)g(D)$$

where

$$g(D) \triangleq g^{(1)}(D^n) + Dg^{(2)}(D^n) + \dots + D^{n-1}g^{(n)}(D^n)$$

- Example: Time Domain
 
$$\mathbf{v} = \mathbf{uG} = (11, 01, 00, 10, 01, 10, 11, 00, 00, \dots)$$
- Example: Transform Domain
 
$$v(D) = u(D^2)g(D) = (1 + D^4 + D^6 + D^8)(1 + D + D^3 + D^4 + D^5)$$

$$= 1 + D + D^3 + D^6 + D^9 + D^{10} + D^{12} + D^{13}$$

$g^{(1)}(D^2)$  and  $g^{(2)}(D^2)$ ?  $g^{(1)}(D^2)$  is  $1 + D^2 + D^4 + D^6 + D^8$  and this was  $1 + D + D^2 + D^3 + D^4 + D^5$ . So  $g^{(1)}(D^2)$  square will be  $1 + D^4 + D^6 + D^8 + D^{10} + D^{12} + D^{13}$

(Refer Slide Time 41:53)

**Encoding of  $(n, 1, m)$  convolutional code**

- The encoding equations can alternately written as,
 
$$\mathbf{v}(D) = \mathbf{u}(D^n)\mathbf{g}(D)$$

$$\mathbf{v}(D) = g^{(1)}(D)u(D) + Dg^{(2)}(D)u(D)$$
 where
 
$$\mathbf{g}(D) \triangleq \mathbf{g}^{(1)}(D^n) + D\mathbf{g}^{(2)}(D^n) + \dots + D^{n-1}\mathbf{g}^{(n)}(D^n)$$

$$g^{(1)}(D) = 1 + D^2 + D^4$$

$$g^{(2)}(D) = 1 + D + D^2$$
- Example: Time Domain
 
$$\mathbf{v} = \mathbf{u}\mathbf{G} = (11, 01, 00, 10, 01, 10, 11, 00, 00, \dots)$$
- Example: Transform Domain
 
$$\mathbf{v}(D) = \mathbf{u}(D^2)\mathbf{g}(D) = (1 + D^4 + D^6 + D^8)(1 + D + D^3 + D^4 + D^5)$$

$$= 1 + D + D^3 + D^6 + D^9 + D^{10} + D^{12} + D^{13}$$

and this will be, so this will be 1 plus D four, 1 plus D four and g 2 D will be, g 2 D square, this is,

(Refer Slide Time 42:08)

**Encoding of  $(n, 1, m)$  convolutional code**

- The encoding equations can alternately written as,
 
$$\mathbf{v}(D) = \mathbf{u}(D^n)\mathbf{g}(D)$$

$$\mathbf{v}(D) = g^{(1)}(D)u(D) + Dg^{(2)}(D)u(D)$$
 where
 
$$\mathbf{g}(D) \triangleq \mathbf{g}^{(1)}(D^n) + D\mathbf{g}^{(2)}(D^n) + \dots + D^{n-1}\mathbf{g}^{(n)}(D^n)$$

$$g^{(1)}(D) = 1 + D^2 + D^4$$

$$g^{(2)}(D) = 1 + D + D^2$$
- Example: Time Domain
 
$$\mathbf{v} = \mathbf{u}\mathbf{G} = (11, 01, 00, 10, 01, 10, 11, 00, 00, \dots)$$
- Example: Transform Domain
 
$$\mathbf{v}(D) = \mathbf{u}(D^2)\mathbf{g}(D) = (1 + D^4 + D^6 + D^8)(1 + D + D^3 + D^4 + D^5)$$

$$= 1 + D + D^3 + D^6 + D^9 + D^{10} + D^{12} + D^{13}$$

so this, this term is given by this. I hope this is clear. So g 1 D is given by this. What we are interested is g 1 D square, so g 1 D square will be given by this expression and we are interested in g 2 D square. So this will be given by 1 plus D square plus D four. Now what is our overall g D? This is given by g 1 D square plus D times g 2 D square. So then this will be g 1 D square is 1 plus D four plus D times this, Ok. So this can

(Refer Slide Time 42:55)

**Encoding of  $(n, 1, m)$  convolutional code**

- The encoding equations can alternately written as,  $v(D) = u(D^n)g(D)$ 

$$v(D) = u(D^n)g(D) = g^{(0)}(D)u(D^n) + Dg^{(1)}(D)u(D^n) + \dots + D^{n-1}g^{(n-1)}(D)u(D^n)$$
- where  $g(D) \triangleq g^{(1)}(D^n) + Dg^{(2)}(D^n) + \dots + D^{n-1}g^{(n)}(D^n)$
- Example: Time Domain  $v = uG = (11, 01, 00, 10, 01, 10, 11, 00, 00, \dots)$
- Example: Transform Domain  $v(D) = u(D^2)g(D) = (1 + D^4 + D^6 + D^8)(1 + D + D^3 + D^4 + D^5)$   
 $= 1 + D + D^3 + D^6 + D^9 + D^{10} + D^{12} + D^{13}$

*Handwritten notes:*  
 $u(D) = 1 + D^2 + D^3 + D^4$   
 $u(D^2) = 1 + D^4 + D^6 + D^8$   
 $g^{(0)}(D) = 1 + D^2$   
 $g^{(1)}(D) = 1 + D + D^2$   
 $1 + D^4 + D(1 + D^2 + D^4)$

be written as 1 plus D four plus D plus D cube plus D five. So this is one plus D, this is 1,

(Refer Slide Time 43:06)

**Encoding of  $(n, 1, m)$  convolutional code**

- The encoding equations can alternately written as,  $v(D) = u(D^n)g(D)$ 

$$v(D) = u(D^n)g(D) = g^{(0)}(D)u(D^n) + Dg^{(1)}(D)u(D^n) + \dots + D^{n-1}g^{(n-1)}(D)u(D^n)$$
- where  $g(D) \triangleq g^{(1)}(D^n) + Dg^{(2)}(D^n) + \dots + D^{n-1}g^{(n)}(D^n)$
- Example: Time Domain  $v = uG = (11, 01, 00, 10, 01, 10, 11, 00, 00, \dots)$
- Example: Transform Domain  $v(D) = u(D^2)g(D) = (1 + D^4 + D^6 + D^8)(1 + D + D^3 + D^4 + D^5)$   
 $= 1 + D + D^3 + D^6 + D^9 + D^{10} + D^{12} + D^{13}$

*Handwritten notes:*  
 $u(D) = 1 + D^2 + D^3 + D^4$   
 $u(D^2) = 1 + D^4 + D^6 + D^8$   
 $g^{(0)}(D) = 1 + D^2$   
 $g^{(1)}(D) = 1 + D + D^2$   
 $1 + D^4 + D(1 + D^2 + D^4)$   
 $= 1 + D^4 + D + D^3 + D^5$

this is 1,D D, D cube D cube, D four D four, D five D five, Ok. So this is our g of D. Now if you multiply all of them, what we get is this. And we can write this, what is 1, 1 is 1, d is this, d square is 0, d cube is 1, d four is 0, d five is 0, d six is 1, d seven is 0, d eight 0, d nine 1, d ten 1, d eleven 0, d twelve 1, d thirteen 1 so this is our output sequence. Now compare with this, what we got in time domain,



(Refer Slide Time 43:50)

**Encoding of  $(n, 1, m)$  convolutional code**

- The encoding equations can alternately written as,  $v(D) = u(D)g(D)$ 

$$v(D) = u(D)g(D) = g^{(0)}(D)u(D) + Dg^{(1)}(D)u(D) + \dots + D^{m-1}g^{(m-1)}(D)u(D)$$

where

$$g(D) \triangleq g^{(1)}(D) + Dg^{(2)}(D) + \dots + D^{m-1}g^{(m)}(D)$$

Example: Time Domain

$$v = uG = (11, 01, 00, 10, 01, 10, 11, 00, 00, \dots)$$

Example: Transform Domain

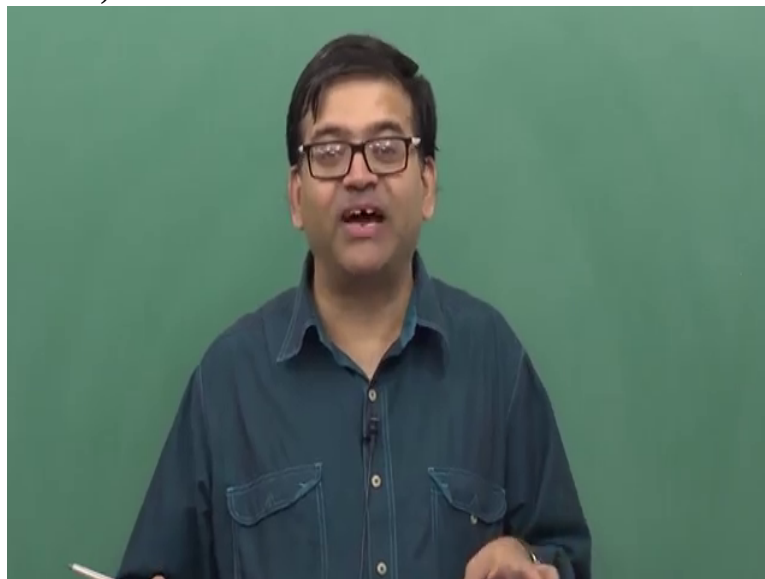
$$v(D) = u(D^2)g(D) = (1 + D^4 + D^6 + D^8)(1 + D + D^3 + D^4 + D^5)$$

$$= 1 + D + D^3 + D^6 + D^9 + D^{10} + D^{12} + D^{13}$$

**11010010011**

1 1, 1 1, 0 1, 0 1, 0 0, 0 0, 1 0, 1 0, 0 1, 0 1. So you can see we are basically getting the same sequence, 1 0 1 0, 1 1 1 1 and rest are all zeroes, here also we are getting all zeroes. So the point to take is, these generator sequence that we

(Refer Slide Time 44:17)



wrote using this time domain representation, we can similarly represent them using this display domain representation and it is lot more convenient to write in this particular notation because then the output sequence is just product of the input sequence and this generator sequence in this domain. So with this I am

(Refer Slide Time 44:44)

Encoding of  $(n, 1, m)$  convolutional code

- The encoding equations can alternately written as,  
$$\mathbf{v}(D) = \mathbf{u}(D^n)\mathbf{g}(D)$$

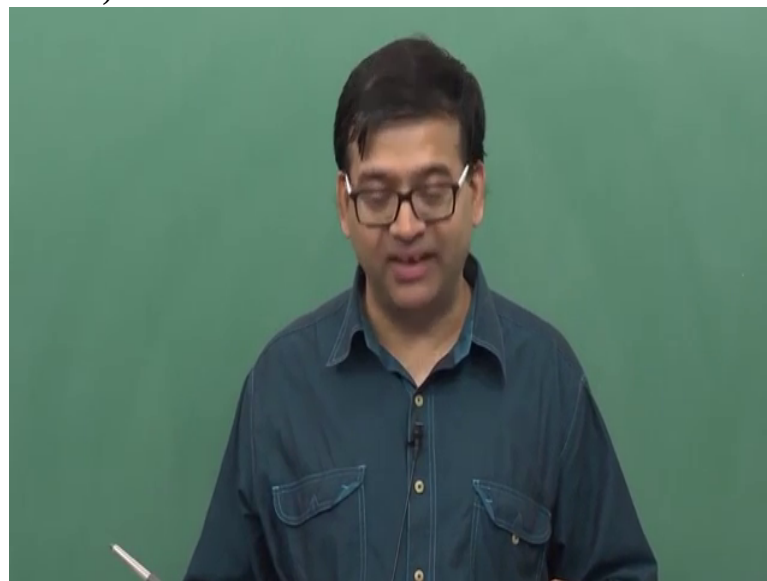
where

$$\mathbf{g}(D) \triangleq \mathbf{g}^{(1)}(D^n) + D\mathbf{g}^{(2)}(D^n) + \dots + D^{n-1}\mathbf{g}^{(n)}(D^n)$$

- Example: Time Domain  
$$\mathbf{v} = \mathbf{u}\mathbf{G} = (11, 01, 00, 10, 01, 10, 11, 00, 00, \dots)$$
- Example: Transform Domain  
$$\begin{aligned}\mathbf{v}(D) &= \mathbf{u}(D^2)\mathbf{g}(D) = (1 + D^4 + D^6 + D^8)(1 + D + D^3 + D^4 + D^5) \\ &= 1 + D + D^3 + D^6 + D^9 + D^{10} + D^{12} + D^{13}\end{aligned}$$

going to conclude this lecture. I just want

(Refer Slide Time 44:46)



to make another point, that these generator matrix that we

(Refer Slide Time 44:50)

Encoding of  $(n, 1, m)$  convolutional code

- The encoding equations can alternately written as,
 
$$\mathbf{v}(D) = \mathbf{u}(D^n)\mathbf{g}(D)$$
- where
 
$$\mathbf{g}(D) \triangleq \mathbf{g}^{(1)}(D^n) + D\mathbf{g}^{(2)}(D^n) + \dots + D^{n-1}\mathbf{g}^{(n)}(D^n)$$
- Example: Time Domain
 
$$\mathbf{v} = \mathbf{u}\mathbf{G} = (11, 01, 00, 10, 01, 10, 11, 00, 00, \dots)$$
- Example: Transform Domain
 
$$\begin{aligned} \mathbf{v}(D) &= \mathbf{u}(D^2)\mathbf{g}(D) = (1 + D^4 + D^6 + D^8)(1 + D + D^3 + D^4 + D^5) \\ &= 1 + D + D^3 + D^6 + D^9 + D^{10} + D^{12} + D^{13} \end{aligned}$$

saw, for example  $g_1$  of  $D$  which we wrote as 1 0 1 and  $g_2$  of  $D$

(Refer Slide Time 44:58)

Encoding of  $(n, 1, m)$  convolutional code

- The encoding equations can alternately written as,
 
$$\mathbf{v}(D) = \mathbf{u}(D^n)\mathbf{g}(D)$$
- where
 
$$\mathbf{g}(D) \triangleq \mathbf{g}^{(1)}(D^n) + D\mathbf{g}^{(2)}(D^n) + \dots + D^{n-1}\mathbf{g}^{(n)}(D^n)$$
- Example: Time Domain
 
$$\mathbf{v} = \mathbf{u}\mathbf{G} = (11, 01, 00, 10, 01, 10, 11, 00, 00, \dots)$$
- Example: Transform Domain
 
$$\begin{aligned} \mathbf{v}(D) &= \mathbf{u}(D^2)\mathbf{g}(D) = (1 + D^4 + D^6 + D^8)(1 + D + D^3 + D^4 + D^5) \\ &= 1 + D + D^3 + D^6 + D^9 + D^{10} + D^{12} + D^{13} \end{aligned}$$

*Handwritten red annotations:*  
 $g^{(1)}(D) = (1 \ 0 \ 1)$   
 $g^{(2)}$

which is 1 1 1, it is typically represented using octal notation. So in many books when they will describe the convolutional encoder for this, they will write it as 5 7 code because octal notation of this is 5 and octal notation of this is 7, so in many places they will say it is a rate 1 by 2 5 7 code and what it means is, they are specifying the generator sequence using this octal notation,

(Refer Slide Time 45:29)

Encoding of  $(n, 1, m)$  convolutional code

- The encoding equations can alternately written as,  $v(D) = u(D^n)g(D)$  where  $g(D) \triangleq g^{(1)}(D^n) + Dg^{(2)}(D^n) + \dots + D^{n-1}g^{(n)}(D^n)$

where

$$g(D) \triangleq g^{(1)}(D^n) + Dg^{(2)}(D^n) + \dots + D^{n-1}g^{(n)}(D^n)$$

- Example: Time Domain  
 $v = uG = (11, 01, 00, 10, 01, 10, 11, 00, 00, \dots)$
- Example: Transform Domain  
$$v(D) = u(D^2)g(D) = (1 + D^4 + D^6 + D^8)(1 + D + D^3 + D^4 + D^5)$$
$$= 1 + D + D^3 + D^6 + D^9 + D^{10} + D^{12} + D^{13}$$

Handwritten notes in red:  
 $g^u(D) = (1 \ 0 \ 1)$   
 $g^o(D) = (1 \ 1 \ 1)$   
 $R = \frac{1}{2} \ (5, 7)$

thank you

(Refer Slide Time 45:31)

