**U19EC045 | DCOM | LAB 6**

**AIM**

To study and simulate Hamming and Convolutional code as a channel coding

technique.

**THEORY**

All real-life channels are affected by **noise** and this causes discrepancies in input and output of digital communication system.

Therefore, to increase the resistance of digital communication to channel noise, **Channel coding** is performed.

The **Channel encoder,** maps the incoming data sequence to a channel input sequence at transmitter and it is transmitted over the noisy channel.

The **Channel decoder** reverse maps the channel output sequence at receiver on to an output data’s sequence.

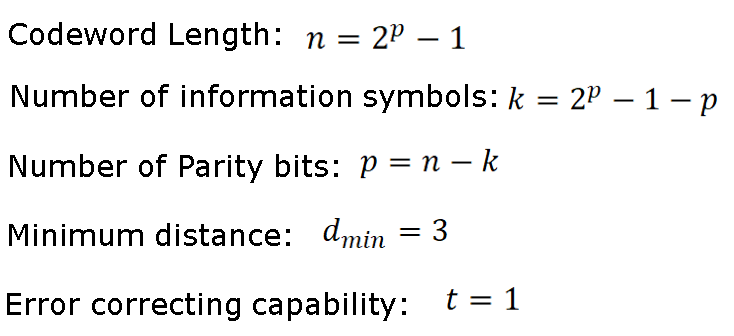
### *HAMMING CODE:*

Linear binary Hamming code falls under the category of linear block codes that can detect two-bit errors and correct single bit error.

The hamming code have the property that,

Here, ***2p-1*** is the number of symbols in the encoded codeword and ***2p-1-p*** is the number of information symbols the encoder can accept at a time

### Characteristics Of (n, k) Hamming Code:



Hamming codes are also classified into two categories that differ in terms of structure of the encoder output:

* Systematic encoding and
* Non-systematic encoding

In **systematic encoding**, just by seeing the output of an encoder, we can separate the data and the redundant bits (also called parity bits).

In the **non-systematic encoding**, the redundant bits and data bits are interspersed.

### *CONVOLUTION CODE:*

The encoder of an (n, k, m) Convolutional code accepts k-bit blocks of input sequence and produce n-bit blocks of encoded sequence.

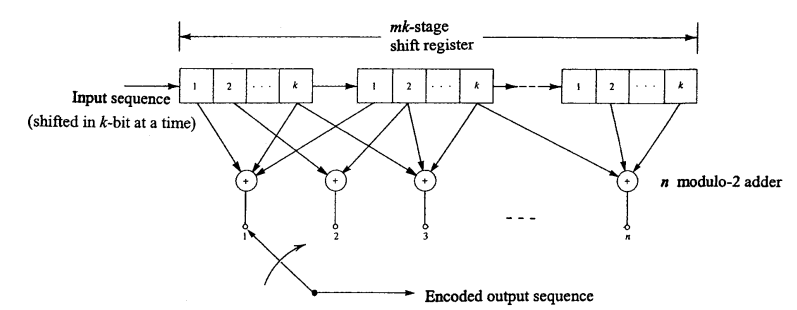
However n-bit blocks of encoded sequence depends not only on the corresponding k-bit input block at same time unit but also on previous (m-1)k bit input blocks.

The integer m is known as constraint length of Convolutional encoder.

A general convolutional encoder consists of an m k-stage shift register and n modulo-2 adders.

At each unit of time, k bits are shifted into the first k-stage of register, all bits in the register are shifted to the right.

The outputs of n modulo-2 adders are sequentially sampled to generate encoded output sequence.



**MATLAB CODE**

1. Hamming Code

|  |
| --- |
| **clc; clear all;**  ***%% Hamming Coding***  **data=input('Enter data bits : ')**  **k = length(data) ; *% no. of information symbols***  **n = 2^3 - 1 ; *% codeword length***  **p = n - k ; *% no.of parity bits***  **disp('parity check matrix');**  **H = hammgen(p) *% parity check matrix***  **parity = H(1:p,p+1:n);**  **disp('generator matrix');**  **G = [eye(k),parity'] *% generator matrix***  **c = data \* G ; *% codeword***  **disp('codeword');**  **c = mod(c,2)** |

1. Convolution code

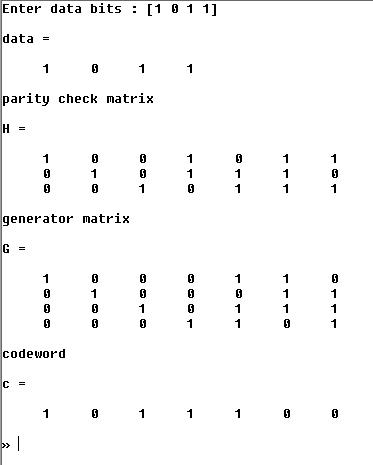
|  |
| --- |
| **d = input('Enter message signal : ');**  **g1 = input('Enter generator 1 matrix : ');**  **g2 = input('Enter generator 2 matrix : ');**  **c1 = conv(d, g1);**  **c2 = conv(d, g2);**  **code = zeros(1, length(c1)+length(c2));**  **for i = 1:length(c1)+length(c2)**  **if mod(i, 2) == 0**  **code(i) = c1(i/2);**  **else**  **code(i) = c2((i+1)/2);**  **end**  **end**  **code = mod(code, 2);**  **disp(['codeword : ',num2str(code)]);** |

1. Cyclic Code

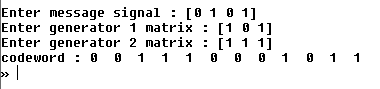
|  |
| --- |
| **clc;**  **close all;**  ***%% Taking Input***  ***% data***  **d = input('Enter data bits : ');**  ***%Shift Register initially set to 000.***  **shift\_initial=[0 0 0];**  ***%creating a shift vector***  **shift = [d,shift\_initial];**  ***%a stores the state of shift register table in transpose form***  **a=zeros(3,length(shift));**  ***%initialising the column of D0.***  **for i=1:length(shift)**  **a(1,i)=shift(i);**  **end**  ***%shift once to get D1,D2.***  **for i=2:3**  **for j=2:length(shift)**  **if j==length(shift)**  **a(i,1)=a(i-1,j);**  **else**  **a(i,j)=a(i-1,j-1);**  **end**  **end**  **end**  ***%% calculating outputs v1,v2 based on logic***  **v1=zeros(1,length(shift));**  **v2=v1;**  **for i=1:length(shift)**  **v1(i)=mod(a(1,i)+a(3,i),2);**  **v2(i)=mod(a(1,i)+a(2,i)+a(3,i),2);**  **end**  ***%displaying outputs state matrix, v1,v2.***  **disp([' D0' ' D1' ' D2' ' v1' ' v2'])**  **for i=1:length(D)**  **disp([' ' num2str(D(i,:)) '  ' num2str(v1(i)) '  ' num2str(v2(i))]);**  **end**  ***%% Concatenating v1,v2 to get code word***  ***%neglection last row of v1,v2.***  **len = length(v1)+length(v2)-2;**  **c=zeros(1,len);**  **for i=1:len**  **if mod(i,2)==0**  **c(i)=v2(i/2);**  **else**  **c(i)=v1((i+1)/2);**  **end**  **end**  **disp('code word');**  **disp(c);** |

**OUTPUT**

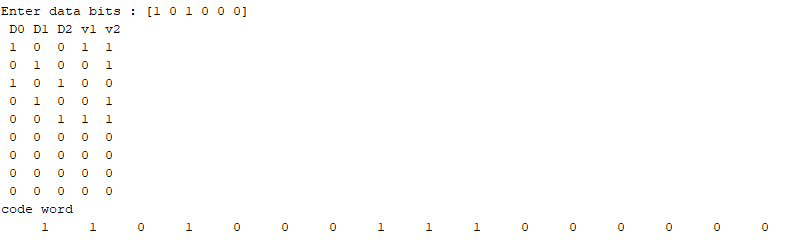
1. Hamming Code



1. Convolution code



1. Cyclic Code



**CONCLUSION**

In this experiment we have learned certain source coding and channel coding techniques like convolutional coding, hamming coding, cyclic codes and we also simulated and checked for various properties of the techniques using MATLAB.