Dept. of Electronics and Electrical Communication Engineering

Indian Institute of Technology Kharagpur

Communication-II Lab

Name: Dipak Bhandare (20EC10027)

Experiment No.: 7

Group Members:

1. Gaurav Barhoniya (20EC10045)
2. Shrija Karmakar (20EC30047)
3. Dipak Bhandare (20EC10027)

Date of Submission: 2– 04 – 2023

Problem Statement: In this software experiment, you need to devise an error correction code for communication over noisy binary channels. In particular construct a convolutional code, viz., encoder and decoder using Viterbi's algorithm.

Objective:

1) Generate a random bit string and encode it using convolutional codes with given polynomial.
2) design a Viterbi decoding algorithm and decode encoded data.
3) Construct the BSC with a crossover probability p and add randomly generated Noise to the transmitted data.
4) plot the bit error rate as a function of channel crossover probability p.

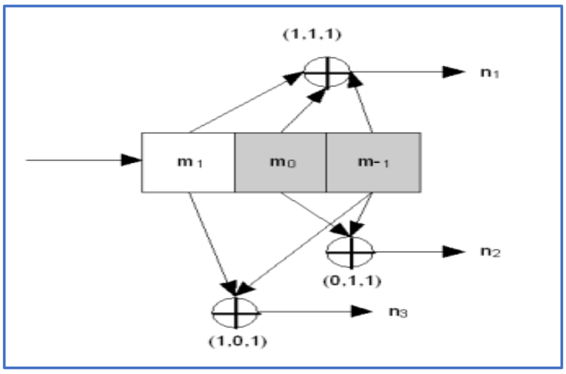
Theory:

Convolutional coding is a widely used coding method which is not based on blocks of bits but rather the output code bits are determined by logic operations on the present bit in a stream and a small number of previous bits. Convolutional codes work by adding redundancy to the transmitted data in a way that allows the receiver to detect and correct errors caused by noise or other forms of interference. This is done by encoding the data using a shift register and a set of predefined generator polynomials. The encoder takes in a stream of input bits and produces a corresponding stream of output bits, which are sent over the communication channel. Such that the encoded bits are the modulo 2 operation of previous bits and the present bit. As we can see that the single input bit can be represent using three **bits n1 n2 n3.** And encoding take places in following way.

$N1 = m1 + m0 + m_{-1}$

$N2 = m0 + m_{-1}$

$N3 = m1 + m_{-1}$



In out case the generator polynomials or encoder polynomials are:

g_1= [1 1 0]

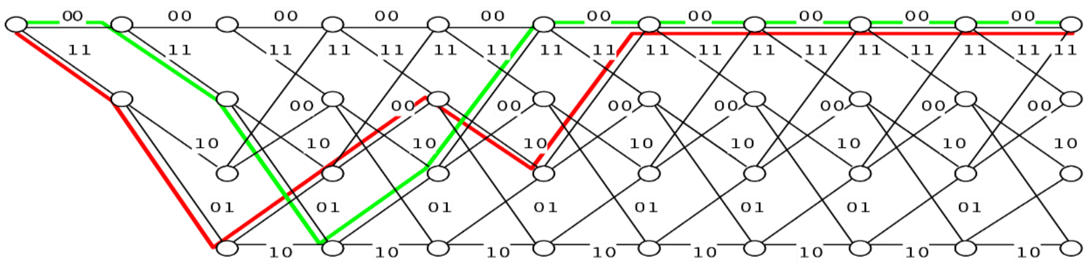
g_2= [1 1 1]

g_3= [1 0 1]

So after encoding the our message the encoded message is fed or transmitted over the different noisy channels in our case we are considering the **Binary Symmetric Channel BSC** with crossover probability of **P** which may introduce errors into the transmitted sequence. So it will become difficult for us to decode the our message bits correctly so we have to choose a proper decoding strategy to decode our message bits stream correctly.

We are using the well-known **Viterbi's algorithm** decoding to determine the most likely sequence of input bits that could have generated the observed output sequence (recall that noise flips bits, thereby causing the received sequence to look different from the transmitted one), given the encoder polynomials and the properties of the noisy channel.

Trellis Diagram

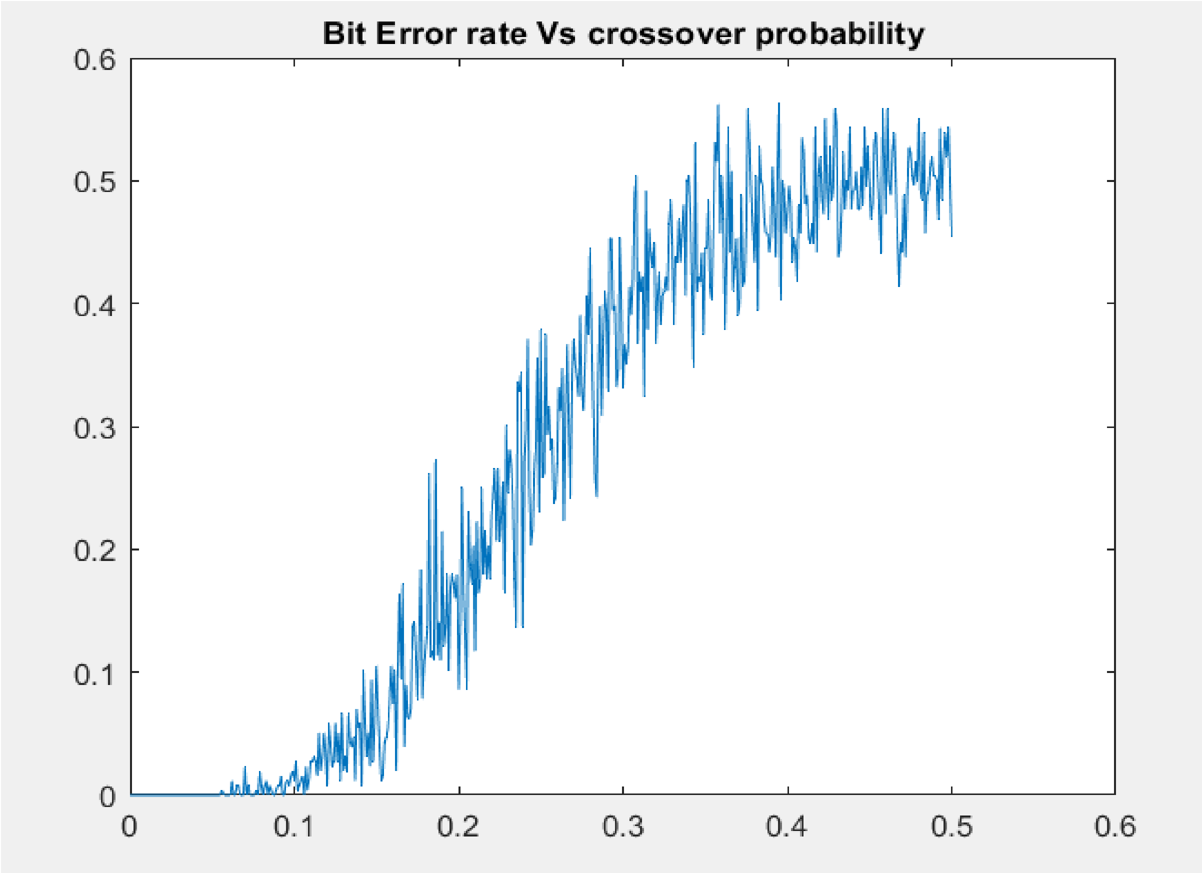


The key idea behind Viterbi's algorithm for decoding convolutional codes is to use dynamic programming to efficiently compute the most likely sequence of states in a bottom-up fashion, and then to perform a traceback operation to determine the most likely sequence of input bits that could have generated the observed output sequence.

Results:

As we can see the length of encoded bit is 768.

2) **bit error rate VS channel crossover probability:**



Discussion:

1) Convolutional codes' performance is influenced by the generator matrix and number of memory elements. Higher rate codes offer better error correction, but a more complex decoder may be needed.
2) Length of the "data" stream = 256 and the length of Encoded stream is 768 which is 3 times the data stream
3) One of the main strengths of Viterbi decoding is its ability to correct multiple errors in a single decoding step. This is particularly useful for channels that introduce errors in a bursty manner, as it can correct a group of errors that occur together in time Another advantage of Viterbi decoding is its high decoding speed. The algorithm has a linear computational complexity, making it suitable for real-time applications that require fast processing.

4) after passing through the channel the output become the **y=z+c** where the **c** is the code word and **z** is the random noise vector. And the addition is **modulo-2**. **The length of Z and C must be same**

5) So after noise get added to the c some bits goes into error so to calculate the no. of bits which is are in error we use the **hamming distance** (between y and c) . Which give the no. different of bits which are error. K is the hamming distance.

```
for  c=1:length(A)
     if(A(c)~=B(c))
         k=k+1;
     end
end
```

6) bit error rate (also BER) is the number of bit errors divided by the total number of transferred bits or the total bits having information.
   Bit error rate = $(\boldsymbol{no.bits\ in\ error})/(\boldsymbol{length}(\boldsymbol{c}))$

7) When the crossover probability is low the BER is also low and When the crossover probability is high(p= 0.5, maximum) the BER is also high.

   So as we can observe that tha BER rate increase with cross over probability increase from 0 to 0.5