Dept. of Electronics and Electrical Communication Engineering

Indian Institute of Technology Kharagpur

COMMUNICATION -II LAB

Author: Arnepalli Jahnavi

Roll No.: 20EC10013

Group No.: 23

Experiment No: 7

Convolutional codes and Viterbi Decoding for Noisy Binary Channels

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.

Introduction:

Channel coding is the process of adding redundancy to a digital signal before transmission over a noisy channel, in order to increase the likelihood of error-free reception.

It is an important technique in modern communication systems because it enables reliable data transmission over unreliable channels.

There are a lot of channel coding techniques:

a) Repetition method.
b) Hamming codes
c) Convolutional codes.

In this experiment we will be learning briefly about the behaviour of convolutional codes and the decoding algorithm called ***Viterbi algorithm***

- Convolutional codes are a class of error-correcting codes used in digital communication systems to detect and correct errors that occur during data transmission.
- They are called "convolutional" because they are based on the mathematical operation of convolution.
- This is done by encoding the data using a shift register and a set of predefined generator polynomials.
- Contrary to block codes, which convert a constant stream of input bits into a constant stream of output bits using a linear feedback shift register (LFSR), convolutional codes convert a constant stream of input bits into a constant stream of output bits.
- The LFSR is a shift register that creates a new output bit based on a set of input bits and a set of previously created output bits using modulo-2 addition.

KEY COMPONENTS:

Math works – Mat lab and functions inbuilt

Objectives:

i. First of all, we need to generate a random sequence of length 256, this will be our input signal, which needed to be encoded.
ii. Generate a rate 1/3 convolutional code with generator polynomials
 a. g1= [1 1 0] b. g2= [1 1 1] c. g3= [1 0 1].
iii. Perform encoding as per standard operations for the convolutional code to generate the codeword.
iv. Now design a Viterbi decoding algorithm. Proceed with the Trellis design followed by the path optimization for Viterbi decoding.
v. Use a binary symmetric channel with a crossover probability p to perturb the codeword, consider at least 5 different values of p (low to high, between 0 to 1/2) in this experiment and determine the recovered or estimated message.
vi. Plot the bit error rate (fraction of bits is error in the data block) as a function of channel crossover probability p.
vii. Now do the same using a Binary asymmetric channel.

Brief Theory:

Encoding of the Convolutional codes:

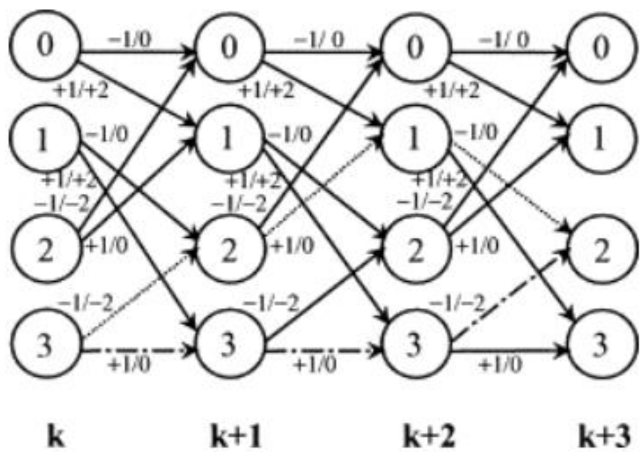
- The data input is divided into blocks of fixed length let’s say k i.e.k-bits
- A shift register is used of length n where one input is the first bit of data sequence and remaining all bits are initialized to zero.
- The input is given to shift register one at a time.
- For every clock cycle, the elements in the shift register are multiplied by the generator polynomial set to give m different output bits, where m is the length of the set of generator polynomials.
- Hence, the rate of encoding is said to be 1/m as we’re receiving m bits as output for 1 bit input.

Decoding of the Convolutional codes:

- The m output bits are compared with the received output bits to detect errors.
- At the receiver side, the decoder uses a maximum likelihood algorithm to find the most likely original sequence of bits that could have generated the received data.

Viterbi decoding algorithm:

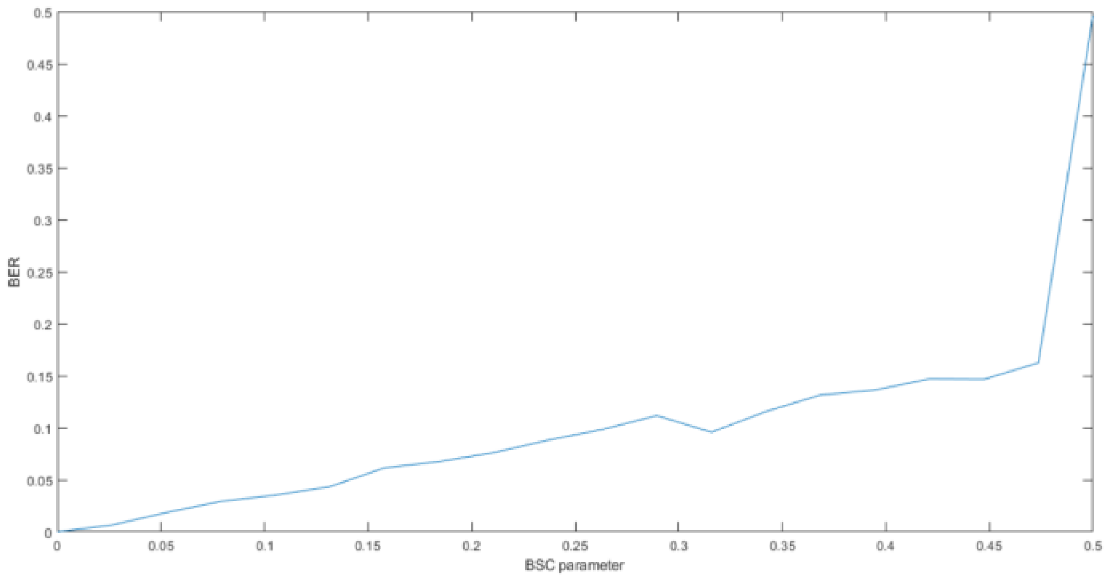
- A maximum-likelihood detector searches over all possible input sequences using an efficient recursive algorithm known as the Viterbi algorithm.
- A trellis is used to represent all possible input sequences. Each path through the trellis represents a different binary input sequence.
- A trellis that extends for *n* sampling times contains unique paths for all 2^n possible input sequences.



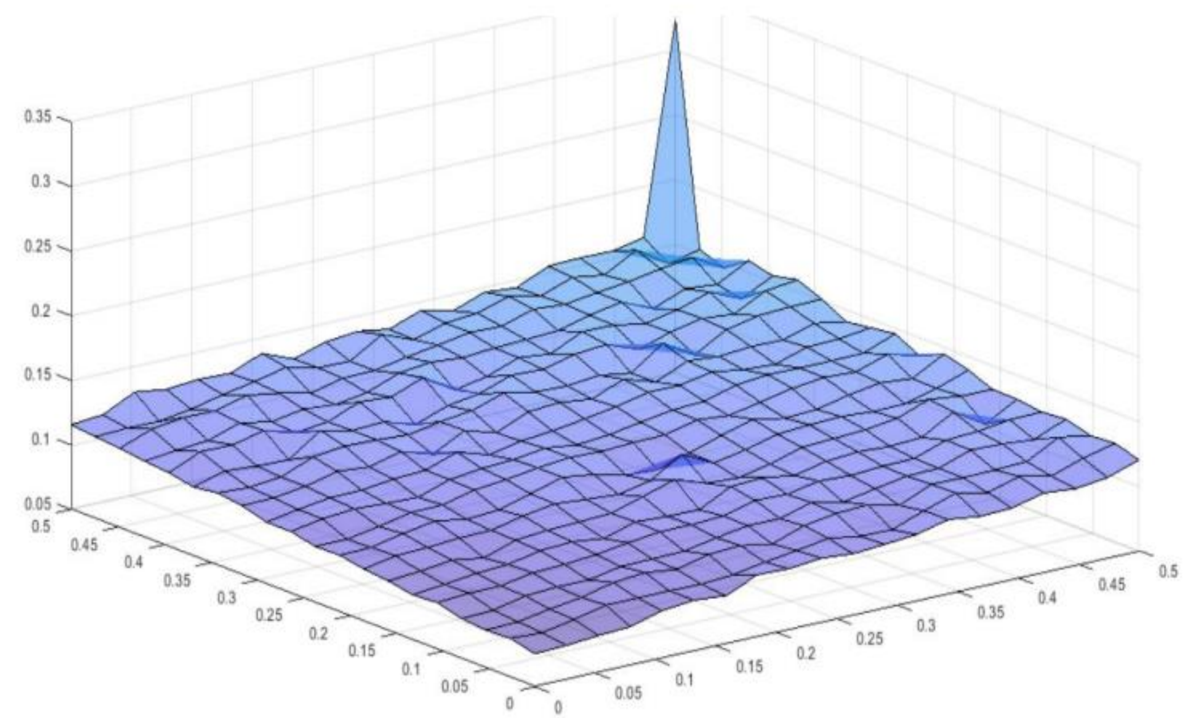
- Binary symmetric channel (BSC): In a BSC, each bit that is transferred has a chance of being flipped (or inverted) during transmission.
- The likelihood of a bit flipping is commonly represented by the letter p, and it is assumed that all bits broadcast across the channel have the same probability.
- Binary asymmetric channel (BAC): In this case, not every bit will necessarily have the same chance of being flipped during transmission.
- To put it another way, for each of the two possible bit values, the channel has a distinct probability of introducing mistakes (0 and 1).

Plots:

Binary Symmetric Channel : BER plotted against BSC parameter

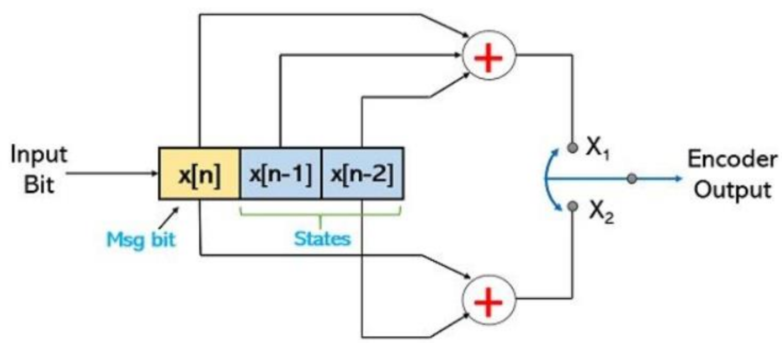


Binary Asymmetric Channel : 2D plot where two axes has varying probabilities o flipping zero and flipping one .



Discussions and Conclusion:

- Convolutional codes are characterized by two parameters: the constraint length, which determines the number of previous input bits that are used to generate each output bit, and the code rate, which determines the ratio of output bits to input bits.
- The higher the constraint length and code rate, the better the error-correction performance of the code.
- Code rate is the ratio of a number of bits shifted at once within the shift register (denoted by k) to the total number of bits in an encoded bitstream (denoted by n). Thus, it is given as k/n.



Block Diagram for Convolutional Encoder

- The bit error rate plot shows that when the likelihood of a BSC channel crossing over rises, bit error rate rises and peaks at 0.5 before falling once more.
- The worst channel behavior occurs when the crossover probability is 0.5.
- If we have n flipflops in the shift register then we will have 2^n states in the state transition diagram.
- The Viterbi algorithm works by computing the most likely sequence of states that could have produced the given sequence of observations.
- At each time step, the method computes the probability of being in each state, given the observation and the preceding state.
- The likelihood of existing in the prior state is multiplied by the probabilities of transitioning to the present state and of emitting the current observation from the current state.

- The algorithm then chooses the state with the highest probability as the most likely state at that time step.
- Once the algorithm has computed the probabilities for each time step, it backtracks through the trellis to find the most likely sequence of states that could have produced the observed sequence.
- When it comes to a BSC channel, the cross over probability (P) is the same for both 0 to 1 and 1 to 0.
- In order to determine which data sequence is most likely to be the source of y, let y be the received vector. We require this feature:

$$\max_{X} P_{Y|X}(\overline{y}|\overline{x})$$

- Hamming distance is our metric for a BSC channel.
- For a BAC channel, in which the cross over probability are different for 0 to 1 (P01) and 1 to 0 (P10).Here the objective remains the same as before, i.e.,

$$\max_{X} P_{Y|X}(\overline{y}|\overline{x})$$

- Here, we our metric will be (For a BAC channel):

$$metric \ = \ d_{01}.log\left(\frac{1-p_{10}}{p_{01}}\right) + d_{10}.log\left(\frac{1-p_{01}}{p_{10}}\right)$$

References:

- https://www.sciencedirect.com/topics/mathematics/viterbi-algorithm
- https://electronicsdesk.com/convolutional-code.html