Dept. of Electronics and Electrical Communication Engineering
Indian Institute of Technology Kharagpur

# DIGITAL COMMUNICATION LAB (EC39001)

Author: Aryan Satpathy
Roll Number: 20EC10014

Group Number: 10
Experiment Number: 7

TA Instructor:

Experiment Number: 7
Title Of Experiment: CONVOLUTIONAL CODES & VITERBI DECODING
Date Of Submission: April 2, 2023

# Introduction

This experiment in purely **MATLAB** based. The experiment intends to demonstrate Convolutional Codes and its Decoding(Viterbi Decoding Algorithm). Convolutional Codes are not linear codes. Neither are they instantaneous. Thus its decoding requires special technique.

## Key Objectives

For this experiment, out key objectives are:

- Generating a PN Sequence.
- Encoding the PN Sequence using Convolutional Codes.
- Decoding the Code using Viterbi Algorithm.
- Simulating a Binary Symmetric Channel(BSC) & Binary Asymmetric Channel(BASC).
- Encoding the PN Sequence after passing through BSC and BASC, and then Deocoding it using Viterbi's Algorithm.

# Results

We begin by first generating the PN Sequence of $6^{\text{th}}$ degree. This can be done by using the built in module in **Communications Toolbox**. We use the following Polynomial for this purpose:

$$x^6 + x^5 + 1 = 0$$

Let us have a look at how the PN Sequence looks.

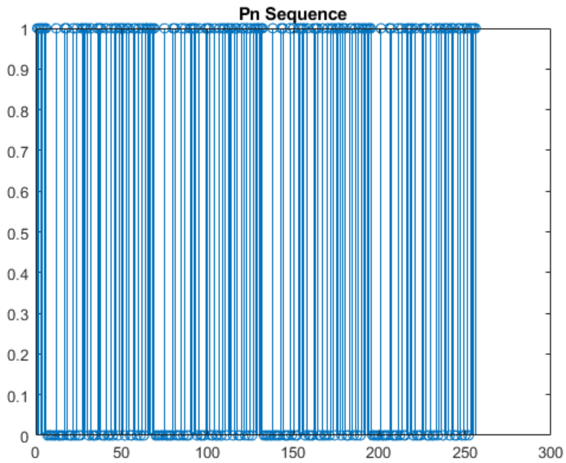


Figure 1: PN Sequence

Now we need to Encode the PN Sequence using Convolutional Codes. Their Generator Polynomials are given below:

$$g1 = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^T$$

$$g2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$$

$$g3 = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}^T$$

And this is how the Output Signal looks:

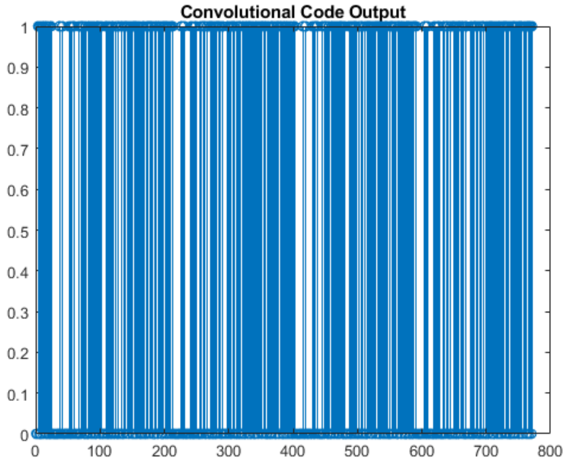


Figure 2: Convolution Code Output

Let us now Decode it using Viterbi's Decoding Algorithm. The result after Decoding the signal is as follows:

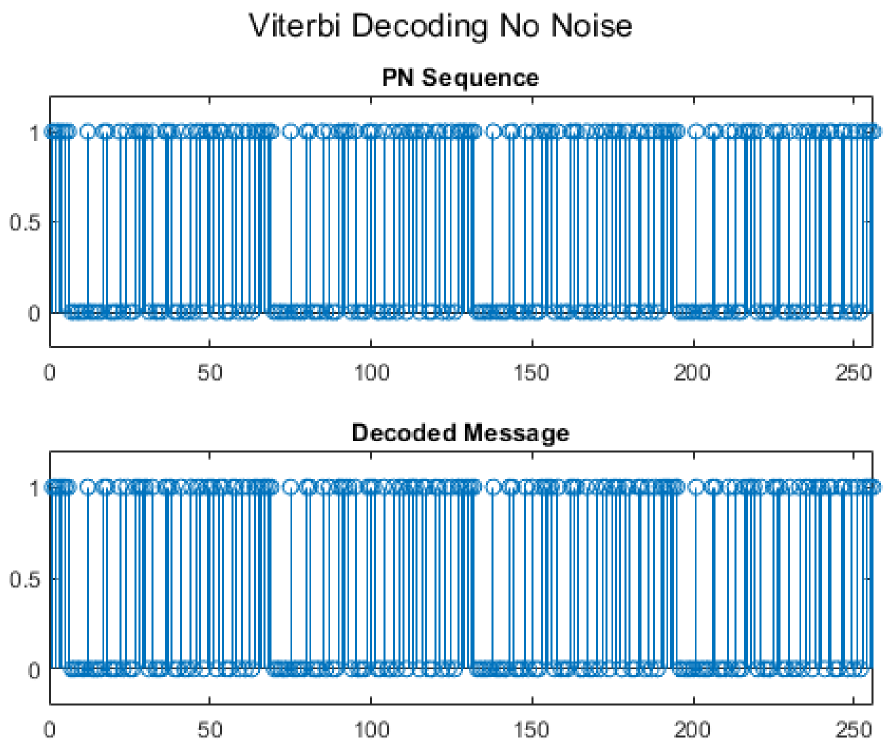


Figure 3: Viterbi Decoding

Decoding in the absence is rather easy. Let us now have a look at what happens when we add the Noise. Lets see how well the Viterbi Decoding Algorithm performs in that case. We will begin with the case with Binary Symmetric Channel(BSC). Let us have a look at the Decoded Result.

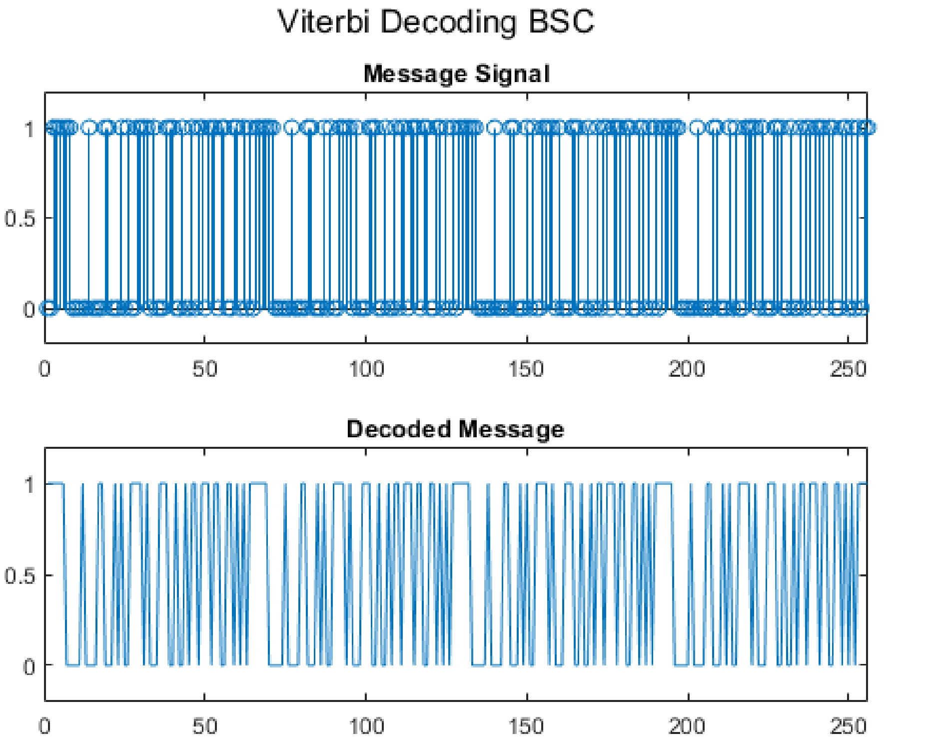


Figure 4: Viterbi Decoding with Flip Probability($p = 1/8$), Match Percentage = 100%

As we can see, Viterbi Decoding performs a great job in case of Binary Symmetric Channels. Let us now have a look at how the Algorithm performs in case of Binary Asymmetric Channels. We will have a look at the Decoded Result.

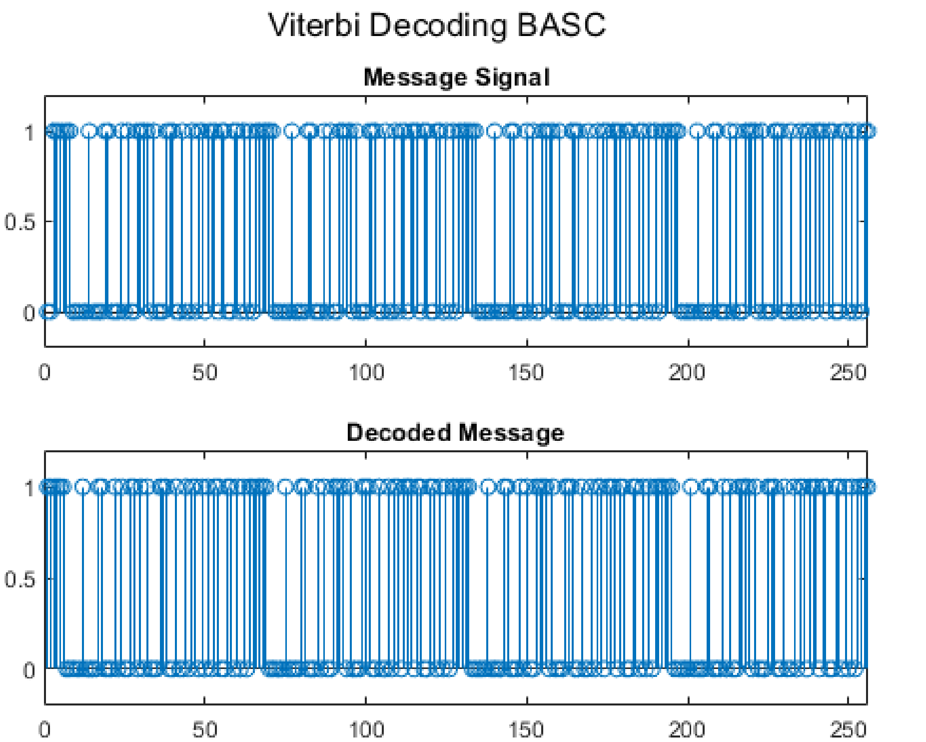


Figure 5: Flip Probabilitys($p_{0\to1} = 1/8$, $p_{1\to0} = 1/22$), Match Percentage = 98.44%

Convolution Codes along with Viterbi Decoding Algorithm are hence proved to be a very **good channel coding scheme** for both BSC and BASC.

## Discussion

Answering the questions given in Lab Manual:

- **Weighing for Binary Asymmetric Channel**

  The case of Binary Symmetric Channel is quite straightforward. We simply consider the **Hamming Distance** for BSC. However in case of Binary Asymmetric Channels, things change. Let the probability of flip from 0 to 1 is $p_{0\to1}$ and the probability of flip from 1 to 0 is $p_{1\to0}$. Now as $p_{0\to1}$ increases, the distance between 0 and 1 begins to matter less. Therefore, we need to weigh the distance between 0 and 1 with a function $f(p_{0\to1})$ such that $f$ decreases with $p_{0\to1}$. So we extend the idea to finally design a distance:

$$\text{distance} = \text{expected} \times xor(observed, expected) \times f(p_{1\to0}) + (1 - \text{expected}) \times xor(observed, expected) \times f(p_{0\to1})$$

- **Would the choice of $f$ matter?**

  Let us consider a time step $t$ of Viterbi Decoding.

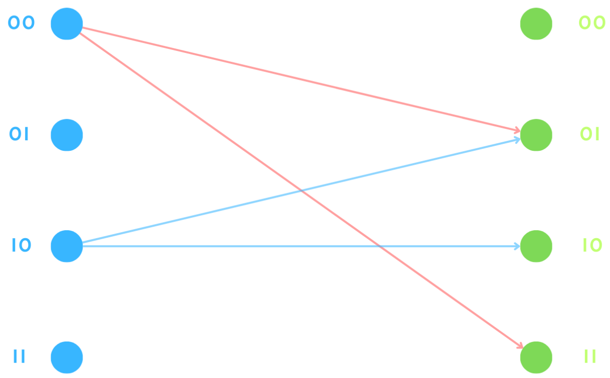


Figure 6: Viterbi Decoding

  We are only going to compare the distance, so the choice of the function $f$ wouldn't matter as long as we choose any Monotonically Decreasing Function $f$. To support this statement, I have also conducted experiment, please have a look at **"Argument at LAB Class"**. For simplicity sake, I went with 2 functions:

$$f(p) = \frac{1}{p}$$

$$f(p) = \text{sigmoid}(p)$$

  Both the choices gave me indistinguishable performance(Measured by Match%, same as Bit Error Rate).

- **Bit Error Rate vs Flip Probability**

  Although we have qualitatively observed that Convolutional Codes and Viterbi Decoding are a great Channel Coding Scheme, let us now have a quantitative look at Channel Code. A good way to do so is to plot Bit Error Rate v/s Probability of Flip. Let us have a look at those:

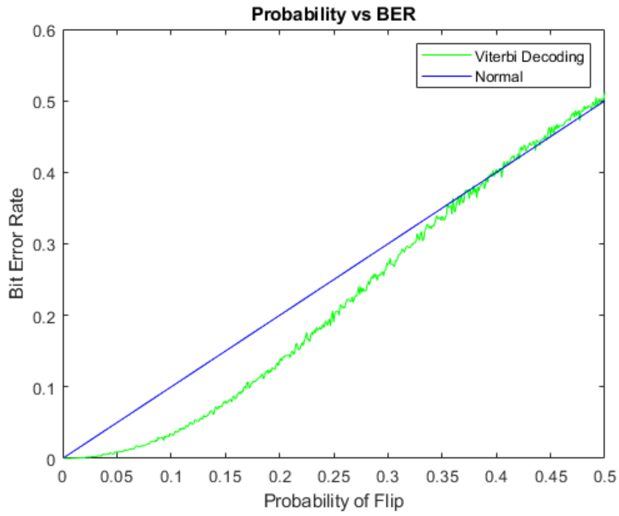


Figure 7: Binary Symmetric Channel

As we can see, the Channel Coding Scheme reduces Bit Error Rate, The thing to notice here is that there is an optimal range of Flip Probability where the Coding Scheme shines. We can always use Repetition Code which is simpler. Convolution Code performs better only in a particular range and this needs to be taken into account before choosing Channel Code scheme. Let us now have a look at Bit Error Rate for Binary Assymetric Channel.

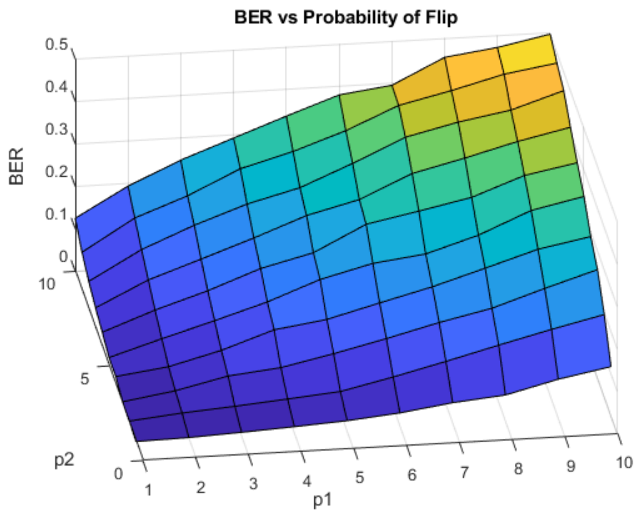


Figure 8: Binary Asymmetric Channel

- **Argument at LAB Class**

  At the Lab Class, Prof. Jithin and Vinay were claiming that I should take log over the weights so that they kind of get Multiplied as time proceeds. Although they were correct, **not taking log in this case would still work**. We need to take the log and multiply the probability if the distances/flips at time instant $t$ is related to the distances/flips at time instant $t+1$. However, here the input sequence is PN Sequence. It is **Memory-Less/Uncorrelated**, therefore the name **Pseudo Random Noise**. Therefore, one can get away without using log over weights. However, when I explained this to Prof. Jithin, he said "Then why are you decoding at the end of block. If it is Memory Less, you could in theory stop and backtrack at any time instant and get the same result". Even though the input sequence is Memory Less, Decoding Algorithm has sequential properties. There are restrictions on states based on the State Transition Diagrams, which reduce the Bit errors as some of the Erroneous Sequences get restricted by the State Transition Diagram given enough time steps. To support the argument, let me just plot the BERs vs Probability with and without taking Logarithm on weight.

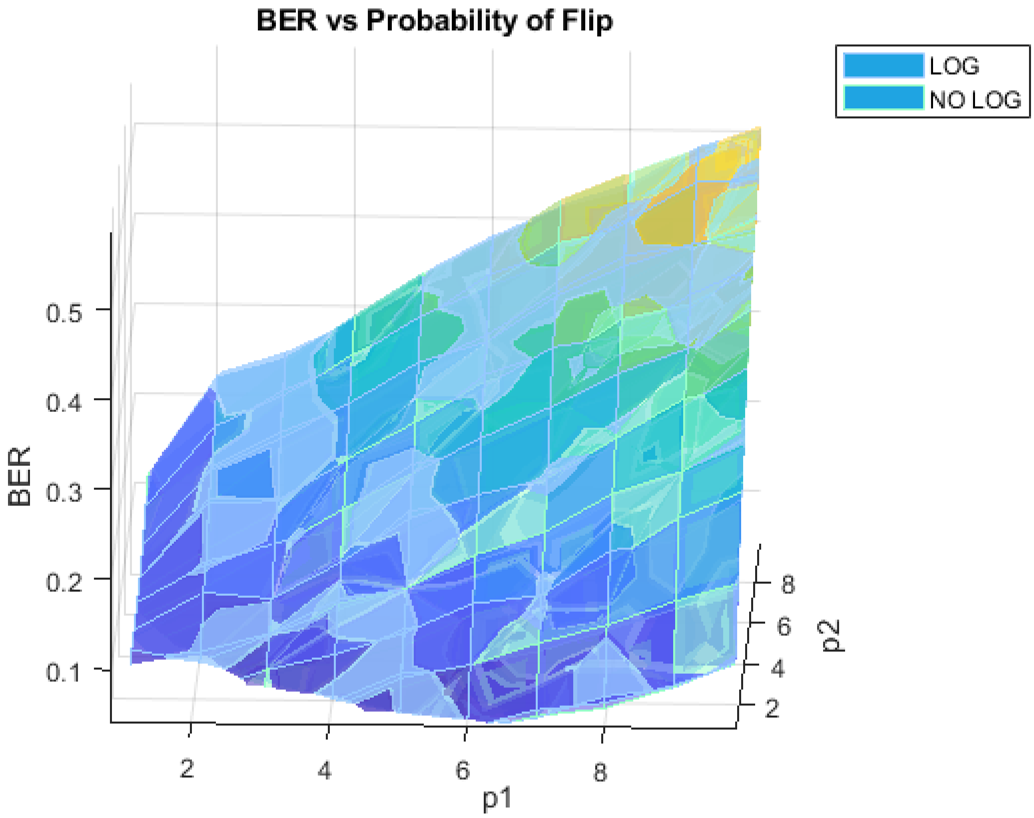


Figure 9: It may not be clear but the Green Surface is without Logarithm and the Blue Surface is with Logarithm. They mostly overlap with a little differences here and there.

Please generate all the 3D plots in order to get the full view. Link to the code: Drive Link to the Code

## Conclusion

In this experiment, we coded Viterbi's Algorithm and observed the Bit Error Rates for Binary Symmetric and Asymmetric Channels.

## References


- MATLAB Official Documentation
- Wikipedia
- ChatGPT
- Lab Manual