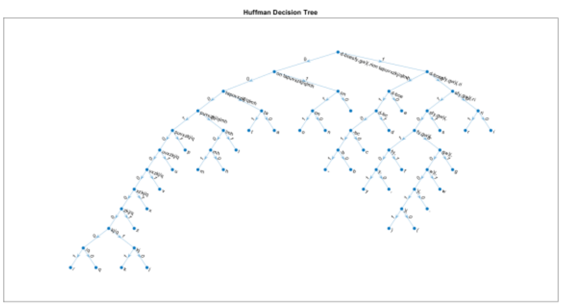
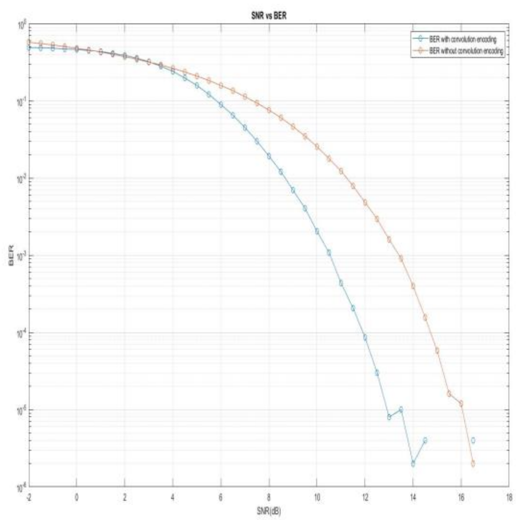




PROJECT 2 REPORT

Convolution coding integrated with source coding

ABSTRACT

Channel coding can be very challenging in the era of digital communication because of the many sources of noise and distortion. One of the most efficient ways to make the system immune to noise is convolution coding the Viterbi algorithm at the decoder. Also, the whole comminution system starting from Huffman source coding to Viterbi channel coding is implemented.

Ahmed Wael

Information Theory and Coding CIE425

Table of Contents

Introduction

After compressing the input at the transmitter for efficient communication, the channel coding is responsible for giving a reliable output as many sources of noise and interference are there by nature. While some coding techniques only detect the errors and do not correct it like Automatic-Repeat-Request (ARQ) protocol [1], many communications applications require detecting and correcting the error. One of the most famous techniques is Forward Error Correct (FEC) [2], where the encoder encodes with extra redundant bits in a way making the correcting possible. While there are some very simple error correction codes like the linear block codes [3], they have very low bit rate compared to other more complicated codes. Convolutional codes are one of the most famous examples for this type of codes, where the input slides into a memory of shift registers and a combination of modular-2 additions are outputted. The decoding process can be done in different way, one of them is by using the Viterbi algorithm [4].

Also, the integration between the source coding and the channel coding is crucial in any communication system. Using Huffman for its optimality as a source coding and Viterbi for its optimality as channel coding, reliable efficient communication can take place.

Theory

For the encoding part, the theory lies heavily on modular-2 addition with the idea of representing the generator functions in terms of polynomial for mathematical convenience. However, for the Viterbi decoder, the main theory is the maximum likelihood estimation which can be reduced into finding the minimum hamming distance [] between the input stream and the output of each branch.

ALGORITHM

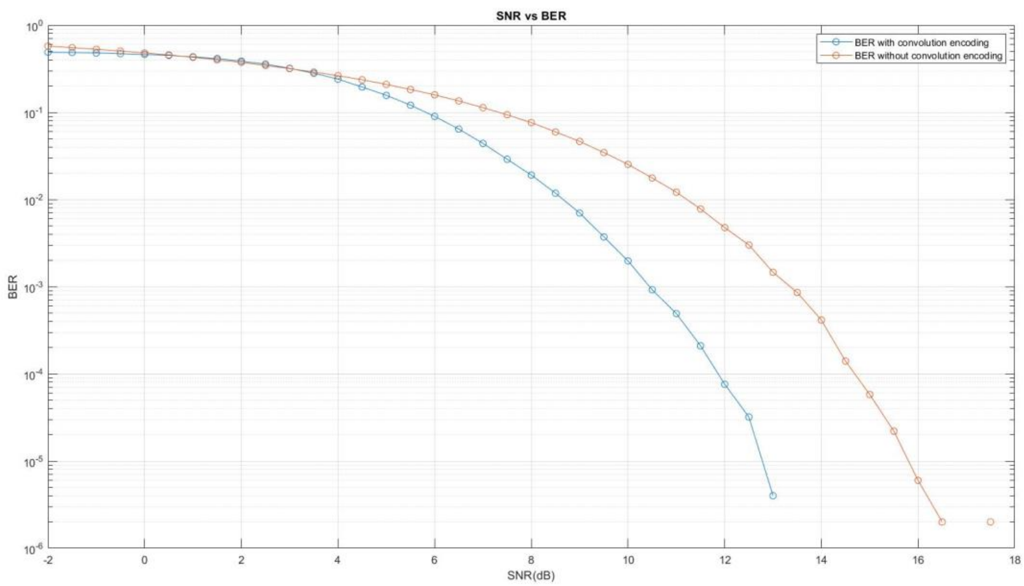
1. Encode the input stream block by block
   a. creating a temporary array and shift the elements right with each iteration a new input bit enters.
   b. Calculate the parity output bits based on the chosen generation functions ***K***.
2. Add the additive white gaussian noise (AWGN) to simulate the channel effect.
3. Decode the encoded stream block by block
   a. Calculate the output of each of the different branches are the are equal to 2^n where n is equal to the output size over the input size.

b. For each ***K*** bits, calculate 2^n hamming distances corresponding the different branches by summing the different bits.
c. Calculate the hamming distance for each node by comparing the accumulated distances of the branches leading to this node.
d. Loop from the end of the trellis to the beginning of it, and get the current state and the previous state, which can lead to finding the decoded bit.

4. Calculate the bit error rate by summing all the different bits of the decoded versus the input streams, and divide by the input size.
5. Compare the performance of the uncoded code versus the convolutional code.
6. All this is integrated with the source coding from part 1 of the project.

Results and Discussion

The obtained graph for the SNR vs. BER is given below, where it's obvious that the performance of both plots is nearly the same at very low SNRs. However, as the SNR increases, the convolutional code outperforms the uncoded sequence with a margin nearly 2 dB at BER of 10^{-2}.



Please also note the output file as there is no corrupted characters which confirms the optimality of both the source coding and the channel coding.

References

[1]"telecommunication | Technology, Examples, Devices, & Facts", *Encyclopedia Britannica*,2018.[Online].Available: https://www.britannica.com/technology/telecommunication#ref608198. [Accessed: 15- Dec- 2018].

[2] "Forward Error Correction (FEC) codes can detect and correct a limited number of errors without retransmitting the data stream", *Ecs.umass.edu*, 2018.[Online]. Available: http://www.ecs.umass.edu/ece/koren/FaultTolerantSystems/simulator/BCH/FEC.htm. [Accessed: 15- Dec- 2018].

[3]"The ABCs of linear block codes - IEEE Journals & Magazine", *Ieeexplore.ieee.org*, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/1311137/. [Accessed: 15- Dec- 2018].

[4]"The Viterbi Algorithm", *Cim.mcgill.ca*, 2018. [Online]. Available: http://www.cim.mcgill.ca/~latorres/Viterbi/va_alg.htm. [Accessed: 15- Dec- 2018].