Comparative Analysis between LDPC Code and Convolution Code

Harsh Yadav
Electronics and Electrical Engineering
Indian Institute of Technology
Guwahati Guwahati ,Assam 781039
Email: y.harsh@iitg.ac.in

***Abstract*— In this paper, we compare Convolution code and LDPC code by simulating both in python and analyzing the output error percentage, time and space complexity with respect to input error percentage. For both the codes we use generic and common variations to get a realistic picture.**

I. Introduction

Information theory key constituent include source and channel coding. Source coding focuses on mapping symbols from information source to a sequence of symbols(usually binary) and compression of data to save power, time and resources. Channel coding on the other hand focuses on adding redundant bits to the code before transmission and correcting error bits at receiving end (the process is shown in Fig 1). In this report we focus on channel coding and consider Channel coding as Forward error control coding(FECC) which is how it's mostly done practically as Backward error control coding requires return channel which is not as efficient. Channel coding is mostly done in two ways :

1) Convolutional Coding: A message is a data stream of arbitrary length and parity, generated by the sliding application of a Boolean function to a data stream.

2)Block Coding: Message is divided into blocks of fixed size to which redundant bits are added.

Some famous Block coding techniques are Hamming Codes, Reed Solomon code and Low-density parity check code(LDPC) , the last one being the half focus in this report.

Our goal is to construct both coding methods (convolution and LDPC) in python and send arbitrary messages , send over a channel(also coded) to be received, then we decode the message compile bit error rate with time and space complexity.

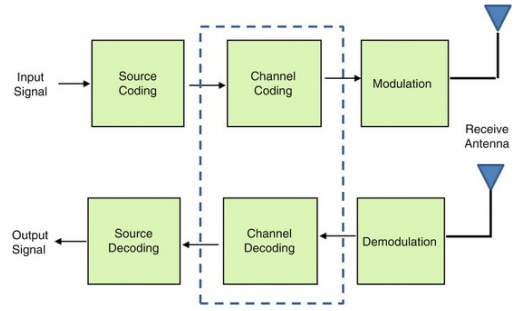

Fig. 1. Simplistic representation of how information is sent

II. Convolution Code

Convolution code simulation code is here. The code is on the simple variation where we have 3 registers, One for the current message bit(m) and the other two for storing previous bits(m1,m2).Then we send two interleaved bits(x1,x2) which represent m+m1+m2(x1) and m+m2(x2). The configuration is shown in Fig 2. Here k=3 and bit rate=½ , Note that this is not the most efficient or most used but a simpler version for analysis. Furthermore, technique with k=7 and r=½ is more commonly used and k=15 and r=1/6 are used in more precise ones like Mars rover. Taking a random input, for encoding, we simply code the finite state system diagram using if and else. Then we sent it over a binary symmetric channel with crossover probability p. Then we decode the obtained sequence using the Viterbi algorithm(this is chosen as it's easy to implement in VLSI). Then, we note the obtained sequence and compare it with the original sequence to find number of error bits and note the time and space complexity. Finally, we show these values on a graph by changing a variable(namely code length and p) and notice how others are affected by it. The obtained graphs after simulations are shown in Fig 3. Additionaly, for the given code the trellis diagram is shown in Fig 4.

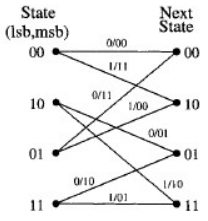

Fig. 4. Trellis diagram for the used convolution code

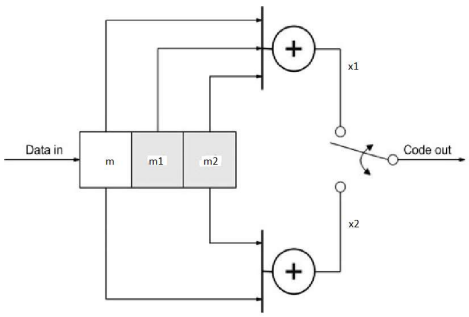

Fig. 2. Basic concept of convolution code encoding for simulation

This work is privately done and is unfunded.

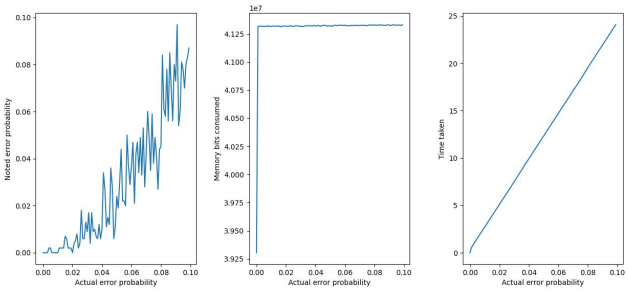

Fig. 3 . Graphs plotted using Convolution code

III. LDPC CODE

LDPC code is here. It is a more complex technique which aims at higher efficiency and reliability of information transferred. Thus, they are preferred when a noisy transmission channel is present. LDPC stand for Low Density Parity Check, so the parity check matrix is sparse in nature meaning there are more 0's than 1's. With a sparse parity check matrix(H) we use the equation $H*c^T=0$, here taking $c=[c_1,c_2\text{-----}c_n]$ we get $n_{equations}$ equations like $c_2+c_5=0$,etc. So we have $n_{equations}$ single parity code. This can be easily understood using tanner graph. Then we use some smart manipulations and conversions to scale up to our code. For decoding and encoding we are using pyldpc library .So the steps go as such:

Generate H and G based on n,d_c and d_v.*

Make a random message of length based on above parameters

Encode the message

Send the message over Binary Symmetric Channel

Decode the received codeword.

Coding all these steps and plotting the same graphs as for Convolutional code with a message length of 1002 we get Fig 5. Additionally, LDPC codes are about 66% more energy efficient than the best Convolutional code, but that can't be proved by this simulation.

An LDPC code is said to be regular in the case where parity check matrix H contains a constant number dc of 1s in each row, and a constant number dv of 1s in each column. Also, the dimensions of H are [nd_v/d_c,n] where n=n_{code}= length after encoding. As for length of message bit it can be found by the equation below

$$k = (d_v - 1) + \frac{n*(d_c - d_v)}{d_c} \quad (1)$$

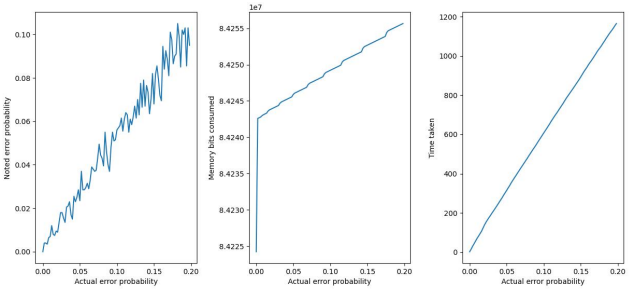

Fig. 5 . Graphs plotted using LDPC code

IV. CONCLUSION

Comparing the final graphs obtained by simulating Convolutional and LDPC codes we observe the following things:

1) In Convolution code the noted error probability was only smaller than actual error probability till about p=0.05. So, if p of the BSC<0.05 then and only then is Convolution code useful. With better codes this can be improved but not as much as in LDPC codes. On the other hand, LDPC code reduces error probability by at least half even at p=0.2 demonstrating that LDPC codes are better for higher noise channels to get a more reliable signal.

2)The memory consumed by the codes gives a picture of device capabilities needed in general(note that since we are using python there are many factors that are not considered thus this is only rough picture). The memory consumption for Convolutional code becomes constant and does not change with increasing error rate but for LDPC the memory usage is almost double, and it increases slightly with increasing p.

3) Finally, the time taken gives idea of summation of Encoding, sending on channel and decoding time. In both cases the time taken increases linearly with p indicating more time in decoding and transferring over channel. Also, the time taken is significantly more in LDPC being a result of its complexity and more operations done.

REFERENCES

1 Source and Channel Coding in Wireless Sensor Networks using LDPC codes,Mina Sartipi, Faramarz Fekri
2 Distributed Source coding in Wireless Sensor networks using LDPC coding: the entire Stepian-Wolf rate region,Mina Sartipi, Faramarz Fekri
3 Convolutional code, Wikipedia
4 Viterbi Algorithm, Wikipedia
5 LDPC, Wikipedia
6 Data Communication - Coding and error detection, NegarMirgati
7 Pyldpc by hicham janati, June 2017

*P.S. All code and images can be found here.