Implementation of polar codes as per 5G standard

Mannava Venkatasai

Contents

1. Introduction

In recently finalized release-15 5G new radio (NR) access technology standards by 3rd Generation Partnership Project (3GPP), two new physical-layer channel coding schemes of polar and low-density parity-check (LDPC) codes have been introduced to replace convolutional and Turbo codes.There are three main 5G usage scenarios of enhanced mobile broadband (eMBB), ultra-reliable and low latency communications (URLLC), and massive machine type communications (mMTC). These scenarios require improved throughput, latency, and reliability compared with a 4G system.

2. Encoding

A. *Algorithm for polar encoding process*

The polar encoder is the type of encoder which takes K bits as input and gives N bits as output.Where N≥K.As per the 5G standard The Value of N should be 1024.

While encoding the message bits in order to achieve efficient communication the K must be small compared to N.
The polar encoding process include Two steps.

1) Adding the frozen bits to the message signal as per the reliability sequence.
   a) As per the 5G standard the size of N must be 1024. So we need the convert the message of length K to N by adding the frozen bits at unreliable positions.
   b) As per the Bhattacharya bounds the reliability sequence for the 5G standard in the order from worst to best is given as :

   https://github.com/Mannava123455/module_2/blob/main/FWC_2_polarencoder/codes/reliability_sequence.dat

2) Apply polar transform for the manipulated message signal.
   a) The polar transform of the manipulated signal is done by the matrix multiplication of the meanipilated signal with the polar matrix.
   b) The basic kernel is given as:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \tag{2.1}$$

   c) The Generator matrix for higher values of N is obtained form the kronocker product with same matrix.
   d) For example in order to encode the message of length 8.We have to do the kronocker product of the base matrix 3 times.

$$\mathbf{G^1} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \tag{2.2}$$

   Perform the kronocker product with same.

$$\mathbf{G^2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{2.3}$$

   Again perform the kronocker product with same.

$$\mathbf{G^3} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{2.4}$$

   If manipulated signal came step 1 is :

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.5}$$

   e) The encoded signal is given as :

$$\mathbf{E} = \mathbf{M}\mathbf{G}^3 \tag{2.6}$$

B. *Performing BPSK modulation*

The encoded bits are modulated by using BPSK i.e By changing 0 to 1 and 1 to -1.
Now we have a signal that contains -1 and 1.We have an array of size 1024 that has 1 and -1.

C. *Adding the Noise to the modulated signal*

In order to analysed the real time communication we have to deal with noise.The modulated signal is being added with noise signal.The noise is analysed by the gaussian signal. The siulation is done by taking different values of SNR in dB.
Let us consider the SNR is 2dB.

$$SNR = 2dB \tag{2.7}$$

$$EBNo = 10^{\frac{SNR}{10}} \tag{2.8}$$

$$EBNo = 1.5848 \tag{2.9}$$

$$Rate = \frac{K}{N} \tag{2.10}$$

$$Rate = \frac{200}{1024} \tag{2.11}$$

$$Rate = 0.1953 \tag{2.12}$$

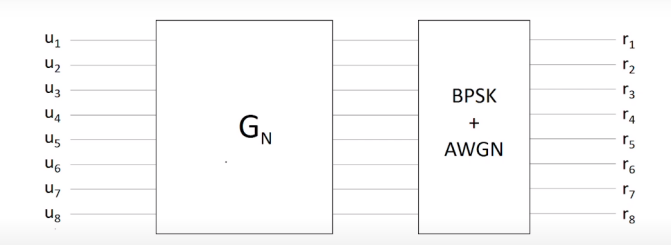
$$\sigma = \frac{1}{\sqrt{2 * Rate * EBNo}} \tag{2.13}$$

The above σ is multiplied with gaussian signal and add it to the BPSK signal. So that we are having the signal mixed with noise.

3. Decoding

A. *Successive Cancelation Decoding*

1) The decoding process includes both Demodulation and Decoding the message signal.
The block diagram of Polar encoder is :



I the above figure $u_1, u_2 ..... u_8$ are the manipulated message bits and $r_1, r_2, ..., r_8$ are the modulated and encoded bits which will be transmited to the air.
The main aim of decoder is to demodulate and decode the transmitted bits.
The decoding process is implemented step by step and it is achieved by the Binary search tree algorithm.

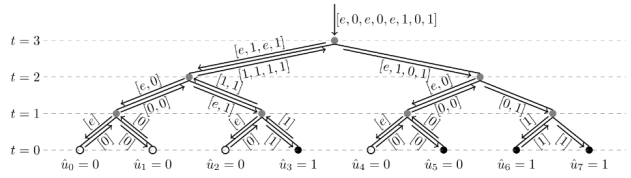


Fig. 4: SC decoding of an $(8, 4)$ polar code over a BEC; white and black dots represent frozen and information bits.

Fig. 1: Check node operation

2) The above decoding will be achieved by xoring the bits and by the implementing the to two functions.
   a) Step 1:Minsum

$$L(u_1) = f(r_1, r_2) = sgn(r_1)sgn(r_2)min(|r_1|, |r_2|) \tag{3.1}$$

   i) u_1=0
   if L(u_1) $\geq$ 0
   ii) u_1=1
   if L(u_1) $\leq$ 0

   b) Step2:

$$g(r_1, r_2, b) = r_2 + (1 - 2b)r_1 \tag{3.2}$$

   i) r_2+r_1
   if b=0
   ii) r_2-r_1
   if b=1

By computing the transmited data with the above Steps we will be recovering the message signal from the transmitted signal.

4. Implementing the polar encoder in python

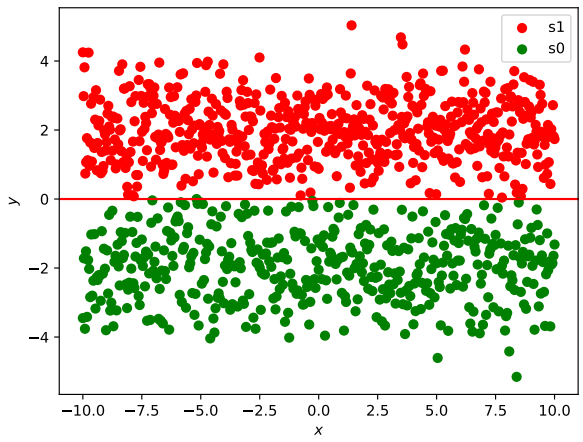
The polar encoder is implemented in python by using the above algorithm.
The code is uploaded in the below link for polar encoder.

https://github.com/Mannava123455/module_2/blob/main/FWC_2_polarencoder/codes/polar_python/polarencoder.py

Analysis of encoded signal with the addition of noise is shown in the figure for SNR = 2dB.
Python code for analysis of BPSK is given as :

```
import matplotlib.pyplot as plt
import numpy as np
mean = [0, 0]
cov = [[1, 0], [0, 1]] # diagonal covariance
A = 2
A1=10**(0.1*A);
simlen = int(1024)
n12 = np.random.multivariate_normal(mean, cov, simlen).T
values = [-1, 1]
probabilities = [0.4, 0.6]
X=np.random.choice(values, simlen, p=probabilities)
y_var = A*X + n12
y1=[]
y2=[]
for i in range(0,len(y_var)):
    if (y_var[i]>0):
        y1.append(y_var[i])

for i in range(0,len(y_var)):
    if (y_var[i]<0):
        y2.append(y_var[i])
print(len(y1))
x=np.linspace(-10,10,len(y1))
x1=np.linspace(-10,10,len(y2))
plt.scatter(x,y1,color=['red'])
plt.scatter(x1,y2,color=['green'])
plt.axhline(y = 0, color = 'r', linestyle = '-')
plt.xlabel("$x$")
plt.ylabel("$y$")
plt.legend(["s1","s0"])
plt.savefig('/home/mannava/latex/snr.pdf')
plt.show()
```



5. Implementing the polar decoder in python

1) The python code for polar decoder is given in the link.

https://github.com/Mannava123455/module_2/blob/main/FWC_2_polarencoder/codes/polar_python/polar_encoder_and_decoder.py

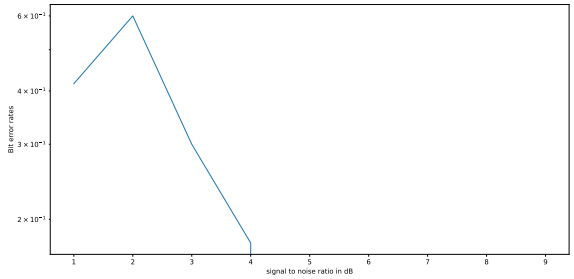
2) The python code for polar encoder and decoder which is implemented in noisy channel is given in followed link.

https://github.com/Mannava123455/module_2/blob/main/FWC_2_polarencoder/codes/polar_python/polarencoder_in_noisychannel.py

3) Bit error rate analysis
BER is given as :

$$BER = \frac{e}{K} \tag{5.1}$$

Where K is no bits transmitted and e is No of errors in recovered signal. The BER graph is plotted for 10 values of SNR for transmission of 1000 bits :



6. Implementing the polar encoder in C language

The polar encoder is implemented in C by using the above algorithm.
The code is uploaded in the below link for polar encoder.

https://github.com/Mannava123455/module_2/blob/main/FWC_2_polarencoder/codes/polar_in_c_language/main.c

7. Implementing the polar decoder in C language

The polar decoder is implemented in C by using the above algorithm.
The code is uploaded in the below link for polar decoder.

https://github.com/Mannava123455/module_2/blob/main/FWC_2_polarencoder/codes/polar_decoder_c/decoder.c

8. Implementing the polar encoder in ESP32 in Serial communication manner

In this implementation the Arduino board is the transmitter which transmit random bits to Esp32 through UART protocol.
There is a FIFO buffer in esp32 of size 200 which continously encode the incoming bits of size 200.
The out put is displayed in Serial monitor of ESP32 board.
The project files for implementing polar encoder in esp32 is given in followed link:

https://github.com/Mannava123455/module_2/tree/main/FWC_2_polarencoder/codes/polar_encoder_esp32