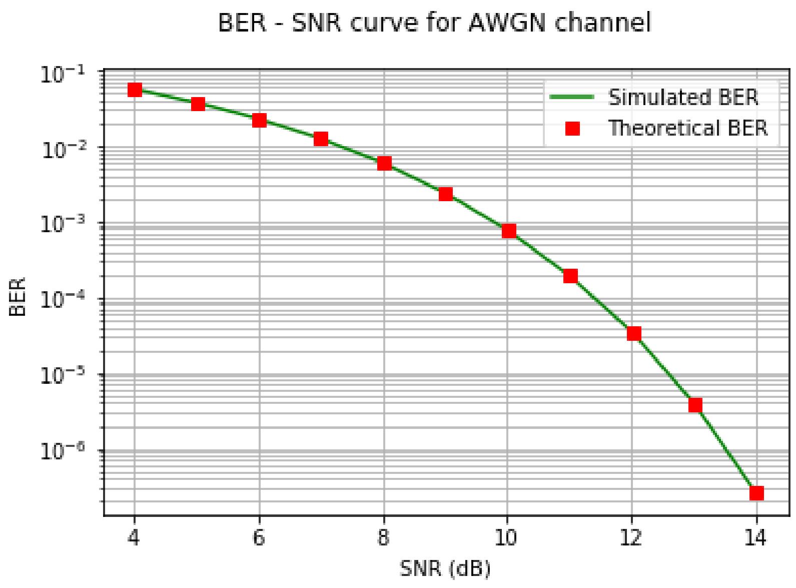
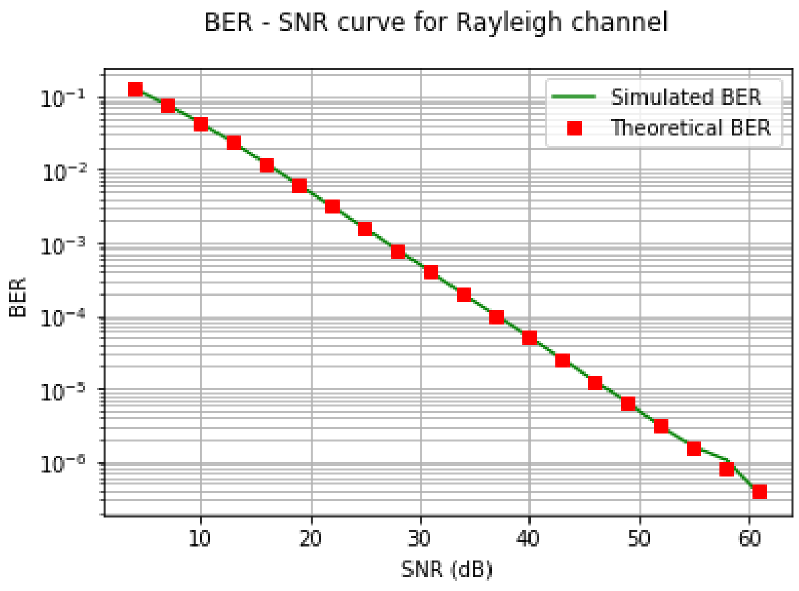
EE670A
PYTHON ASSIGNMENT #1
M. Aravind, 21104403

```python
#EE670A Python Assignment 1
#Prepared by M. Aravind (21104403)
#Aim: Compare BER vs SNR for empirical and analytical estimations (QPSK), for both AWGN and Rayleigh channels.

import numpy as np
import matplotlib.pyplot as plt
import numpy.random as nr
from scipy.stats import norm

#AWGN
blockLength = 10000 #Number of symbols per block
nBlocks = 1000 #Number of blocks
No=1 #Complex Gaussian Noise Variance
EbdB = np.arange(1,12,1)
Eb=10**(EbdB/10)
SNR = 2*Eb/No
SNRdB= 10*np.log10(SNR)
BER = np.zeros(len(EbdB))
BERt = np.zeros(len(EbdB))

#Rayleigh
rblockLength = 10000 # Separate configurable blocklengths and other parameters for Rayleigh channel
nBlocksr = 1000
No=1
EbdBr = np.arange(1,60,3)
Ebr=10**(EbdBr/10)
SNRr = 2*Ebr/No
SNRdBr= 10*np.log10(SNRr)
BERr = np.zeros(len(EbdBr))
BERrt = np.zeros(len(EbdBr))

#AWGN
for blk in range(nBlocks):
    BitsI = nr.randint(2,size=blockLength)
    BitsQ = nr.randint(2,size=blockLength)
    Sym = (2*BitsI-1)+1j*(2*BitsQ-1) #QPSK Symbols (BPSK as real and imaginary parts)
    noise = nr.normal(0,np.sqrt(No/2),blockLength)+1j*nr.normal(0,np.sqrt(No/2),blockLength)
    for k in range(len(EbdB)):
        TxSym = np.sqrt(Eb[k])*Sym  #Transmited symbol through AWGN Channel
        RxSym = TxSym + noise #Received symbol equals noise+Transmitted symbol
        DecBitsI = (np.real(RxSym)>0) #Decode 1 if Real part of Received symbol greater than 0. else -1. for both I and Q
        DecBitsQ = (np.imag(RxSym)>0) #Same as above logic but here we look at the imaginary part.
        BER[k] = BER[k]+ np.sum(DecBitsI != BitsI) + np.sum(DecBitsQ != BitsQ) #Adding up Bit Errors. It is a bit error, if decoded symbol doesn't equal the transmitted symbol.

#Rayleigh
for blk in range (nBlocksr):
    BitsIr = nr.randint(2,size=rblockLength)
    BitsQr = nr.randint(2,size=rblockLength)
    Sym_r = (2*BitsIr-1)+1j*(2*BitsQr-1)
    noise_r = nr.normal(0,np.sqrt(No/2),rblockLength)+1j*nr.normal(0,np.sqrt(No/2),rblockLength) #Separate noise variable for customized/changed blocklength for easy visualization  of Rayleigh case
    h = nr.normal(0,np.sqrt(1/2),rblockLength)+1j*nr.normal(0,np.sqrt(1/2),rblockLength) #Rayleigh Fading Channel
    for k in range(len(EbdBr)):
        TxSym_r = np.sqrt(Ebr[k])*Sym_r #Transmited symbol through Rayleigh Channel
        RxSym_r = h*TxSym_r + noise_r
        EqSym_r = 1/h*RxSym_r
        DecBitsIr = (np.real(EqSym_r)>0)
        DecBitsQr = (np.imag(EqSym_r)>0)
        BERr[k] = BERr[k]+ np.sum(DecBitsIr != BitsIr) + np.sum(DecBitsQr != BitsQr)

```

```
#AWGN
BER = BER/blockLength/2/nBlocks #Total number of bits is (2bits per symbol)*(blocklength
symbols)*(number of blocks)
BERt = 1-norm.cdf(np.sqrt(SNR)) #Theoretical BER Curve = Q(sqrt(SNR))
#Rayleigh
BERr = BERr/rblockLength/2/nBlocksr
BERrt = 0.5*(1-np.sqrt((SNRr/(2+SNRr)))) #Theoretical (Analytical) BER Curve = 0.5(1-
sqrt(SNR/(2+SNR)))

#Plots:

#AWGN
plt.figure()
plt.yscale('log');
plt.plot(SNRdB,BER,'g-')
plt.plot(SNRdB,BERt,'rs')
plt.grid(1,which='both')
plt.suptitle('BER - SNR curve for AWGN channel')
plt.xlabel('SNR (dB)')
plt.ylabel('BER')
plt.legend(["Simulated BER", "Theoretical BER"])

#Rayleigh
plt.figure()
plt.yscale('log');
plt.plot(SNRdBr,BERr,'g-')
plt.plot(SNRdBr,BERrt,'rs')
plt.grid(1,which='both')
plt.suptitle('BER - SNR curve for Rayleigh channel')
plt.xlabel('SNR (dB)')
plt.ylabel('BER')
plt.legend(["Simulated BER", "Theoretical BER"]

```

Output:





<u>Observation:</u>

1. SNR curves were computed and plotted approximately till the Bit Error rate of 10^{-6} (QPSK scheme).
2. Computed BER vs SNR Plot almost follows the standard analytical model corresponding to each channel as:

<u>AWGN Channel</u> ➔ $BER_{AWGN} = Q(\sqrt{SNR}) \ = \int_{\sqrt{SNR}}^{\infty} e^{\frac{-x^2}{2}} dx$

<u>Rayleigh Channel</u> ➔ $BER_{Rayleigh} = E\left(Q\sqrt{a^2SNR}\right) = \int_{-\infty}^{\infty} Q(\sqrt{a^2SNR}) f_A(a) da =$

$$\int_0^{\infty} Q\left(\sqrt{a^2SNR}\right) 2a\, e^{-a^2} da = \frac{1}{2}\left(-\sqrt{\frac{SNR}{2+SNR}}\right),$$

$$where\ 'a'\ follows\ rayleigh\ pdf\ distributon.$$

3. It is verified by superimposing the calculated plot and theoretical plot, that both are same.