

```

clc;
close all;
clear all;
modulationTypes = categorical(["BPSK", "QPSK", "8PSK", "16QAM", "64QAM", "PAM4", "GFSK", "CPFSK", "B-FM", "DSB-AM", "SSB-AM"]);
load trainedModulationClassificationNetwork
trainedNet
% rand bits
for t= 1:100
N=1990;
message=randi([0 1],N,1);
% Signal noise
snr = 3 + [20 22 24 26 28 30];
%Encoded Data;
trellis = poly2trellis(3,[6 7])
codedData = convenc(message,trellis);
output = codedData';
guard1=[1 1 1 1 1 1];
guard2=[1 1 1 1 1 1];
UW=[1 0 1 0 0 0 1 0 ];
UW=[UW, UW, UW, UW UW, UW, UW, UW, UW, UW, UW, UW];
codedData=[guard1 UW output(1:940*2) guard2];
% modulation
r= 8;
y= modulate(codedData,8,7200000);
% upsampling by 8
z=upsample(y,8);
%pulse shaping filter
a = rcosine(0.35,1,'sqrt',r);
combine = conv(y,z);
leng = length(z);
%channel
sym_rate = 93600;
fd = [20,100,0];
k= [7 12 200];
[GI,GQ] = rician_func(fd,2048,sym_rate,k);
ricianch = GI + j*GQ;
ri_1 = ricianch*x;
% rician fading
K=12;
fd=100;
L=1024;
[RI,RQ]=rayleigh(fd,L,r);
% AWGN noise
q=awgn(a,snr);
% scattered power
m_path=10^(-K/10);
% normalize scattered power for the desired K
RI=RI*sqrt(m_path);
RQ=RQ*sqrt(m_path);

```