



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

UNIVERSIDAD POLITÉCNICA DE
VALENCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE
TELECOMUNICACIÓN

Máster Universitario en Ingeniería de
Telecomunicación

Práctica 2 Técnicas Avanzadas de Codificación

PSCA

Autor:
Andrés Ruz Nieto

VALENCIA, 2021

— **TELECOM**
UPV VLC

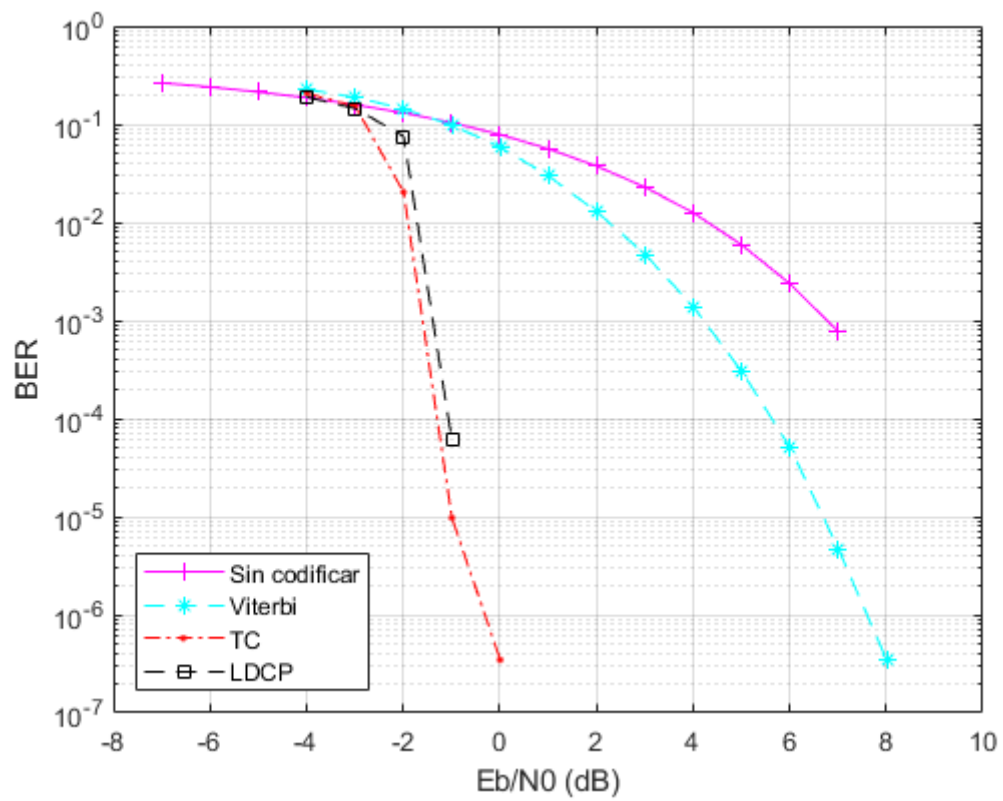


Figura 1: Probabilidad de error de bit

Para obtener más precisión en TC y LDPC se pueden realizar más transmisiones de los mensajes en los puntos de interés.

```

clc, clear all

Nmensajes = 5000;
tre = poly2trellis(4, [13, 15], 13);
k = 1152;
M = 2;
SNR = -7 : 1 : 7;
R = 1/2;
mm = randi( [0 1], k , 1);

BER_sincod = zeros(1,length(SNR));
BER_convoc = zeros(1,length(SNR));
BER_tc = zeros(1,length(SNR));
BER_ldcp = zeros(1,length(SNR));

%% SIN CODIFICAR
disp('SIN COD')
EbNo = SNR-10*log10(log2(M));
cc = mm;
hMod=comm.PSKModulator(M,0,'BitInput',true,'SymbolMapping','Gray');
hDemod=comm.PSKDemodulator(M,0,'SymbolMapping','Gray','BitOutput',true, 'DecisionMethod',
    'Hard decision');

for i = 1:length(SNR)
    hCalculoError=comm.ErrorRate;
    hcanal=comm.AWGNChannel('NoiseMethod','Signal to noise ratio (Eb/No)', 'EbNo',EbNo(i)
        ),'BitsPerSymbol',1);

    for j = 1:Nmensajes
        coordtx= step(hMod,cc);
        coordrx=step(hcanal,coordtx);
        rrdem=step(hDemod,coordrx);
        VectorErrores= step(hCalculoError, cc, rrdem);
    end

    BER_sincod(i) = VectorErrores(1);
end

semilogy( EbNo, BER_sincod, '-+m');
grid on
hold on

%% CONVOC
disp('CONVOC')
hEnc = comm.ConvolutionalEncoder( tre, 'TerminationMethod', 'Terminated');
hDec=comm.ViterbiDecoder(tre,'InputFormat','hard', 'TracebackDepth', M+1, '
    TerminationMethod', 'Terminated');
EbNo = SNR-10*log10(R*log2(M));
for i = 1:length(SNR)
    hCalculoError=comm.ErrorRate;
    hcanal=comm.AWGNChannel('NoiseMethod','Signal to noise ratio (Eb/No)', 'EbNo',EbNo(i)
        ),'BitsPerSymbol',1);

    for j = 1:Nmensajes
        cc = step(hEnc, mm);
        coordtx= step(hMod,cc);
        coordrx=step(hcanal,coordtx);
        rrdem = step(hDemod,coordrx);
        deco=step(hDec,rrdem);
    end
end

```

```

        mDec=deco(1:end-(3));
        VectorErrores= step(hCalculoError, mm, mDec);
    end

    BER_convoc(i) = VectorErrores(1);
end

semilogy( EbNo, BER_convoc, '--*c');

%% TC
disp('TC')
hDemod=comm.PSKDemodulator(M,0,'SymbolMapping','Gray','BitOutput',true, 'DecisionMethod',
    'log-likelihood ratio');

EbNo = SNR-10*log10(R*log2(M));
hEnc1 = comm.ConvolutionalEncoder('TrellisStructure', tre, 'TerminationMethod', '
    Terminated' );
hEnc2 = comm.ConvolutionalEncoder('TrellisStructure', tre, 'TerminationMethod', '
    Terminated' );
hDec1 = comm.APPDecoder('TrellisStructure', tre, 'TerminationMethod', 'Terminated', '
    Algorithm', 'True APP');
hDec2 = comm.APPDecoder('TrellisStructure',tre, 'TerminationMethod','Terminated', '
    Algorithm', 'True APP');
for i = 1:length(SNR)
    hCalculoError=comm.ErrorRate;
    hcanal=comm.AWGNChannel('NoiseMethod','Signal to noise ratio (Eb/No)', 'EbNo',EbNo(i)
        ),'BitsPerSymbol',1);

    for j = 1:Nmensajes
        cc = turbo_codif(hEnc1, hEnc2, mm);
        coordtx= step(hMod,cc);
        coordrx=step(hcanal,coordtx);
        rrdem = step(hDemod,coordrx);
        mDec=turbo_dec(rrdem,k,6,hDec1,hDec2);
        VectorErrores= step(hCalculoError, mm, mDec);
    end

    BER_tc(i) = VectorErrores(1);
end

semilogy( EbNo, BER_tc, '-.r');

%% LDPC
disp('LDPC')
EbNo = SNR-10*log10(R*log2(M));
load('matrixH.mat');
nmenosk=size(H,1);
n=size(H,2);
k=n-nmenosk;
mm = randi( [0 1], k , 1);
hEnc= comm.LDPCDecoder(H);
hDec= comm.LDPCDecoder(H);
for i = 1:length(SNR)
    hCalculoError=comm.ErrorRate;
    hcanal=comm.AWGNChannel('NoiseMethod','Signal to noise ratio (Eb/No)', 'EbNo',EbNo(i)
        ),'BitsPerSymbol',1);

    for j = 1:Nmensajes
        cc = step(hEnc, mm);

```

```

        coordtx= step(hMod,cc);
        coordrx=step(hcanal,coordtx);
        rrdem = step(hDemod,coordrx);
        deco=step(hDec,rrdem);
        mDec=deco;
        VectorErrores= step(hCalculoError, mm, cast(mDec,'double'));
    end

    BER_ldcp(i) = VectorErrores(1);
end

semilogy( EbNo, BER_ldcp, '--sk');

xlabel( 'Eb/NO (dB)' );
ylabel( 'BER' );
legend({'Sin codificar','Viterbi', 'TC', 'LDCP'},'Location','southwest','Orientation','vertical')

```

Script 1: Script de Matlab