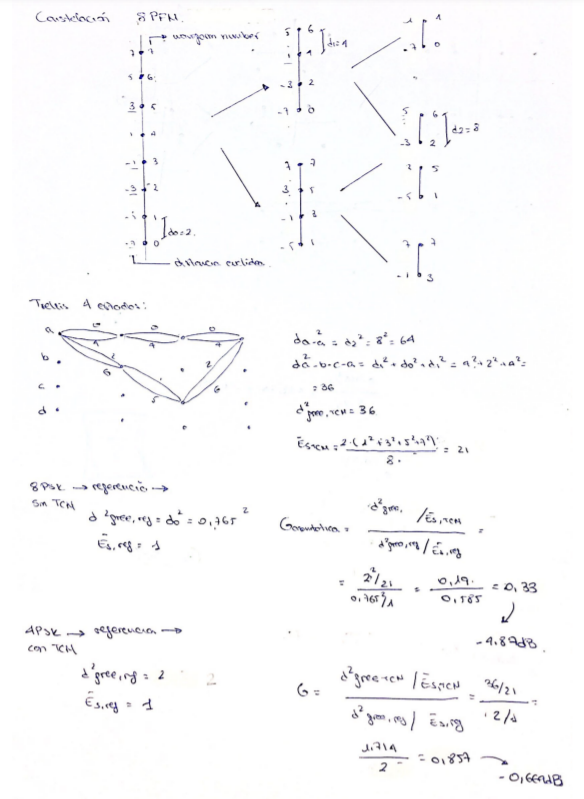
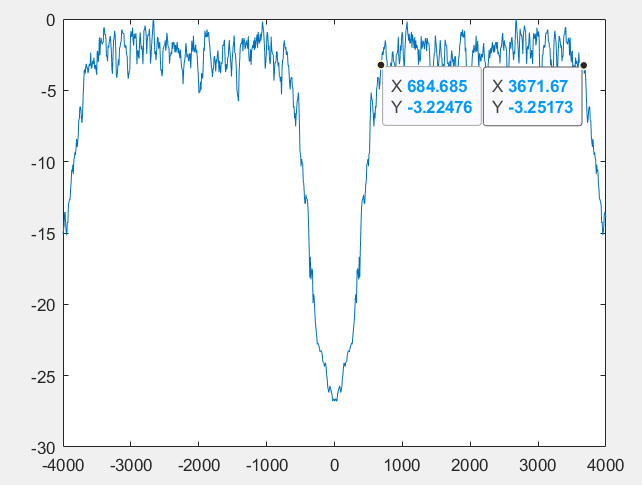
1. **Estudio previo**

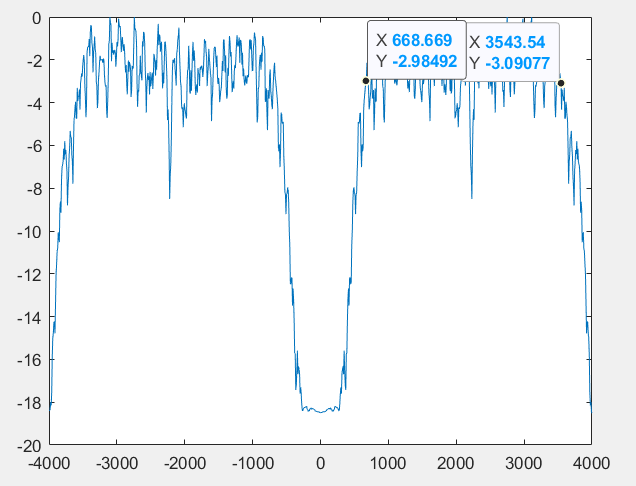
* **Calcular la ganancia con TCM de la constelación de 8 símbolos propuesta en la parte 1 respecto a un 4‐PSK sin codificar, usando el trellis del convolucional de cuatro estados visto en los ejemplos de clase.**
* **Caracterizar (obtener SNR, ancho de banda y capacidad) de los siguientes dos canales (canal1a.p y canal1b.p)**

Para la obtención del **ancho de banda**, se introduce una delta al canal y se debe calcular la caída a 3dB, al ser un paso banda se debe hacer en ambos lados, así obtendremos el ancho de banda.

1. Canal1a:



1. Canal1b:



En ambos casos se puede observar cómo se tiene un ancho de banda de 3kHz.

El código utilizado para esta parte es el siguiente:

N=1000;

fm=8e3;

delta=zeros(1,N);

delta(N/2)=1;

h=canal1a(delta,fm);

H=abs(fft(h));

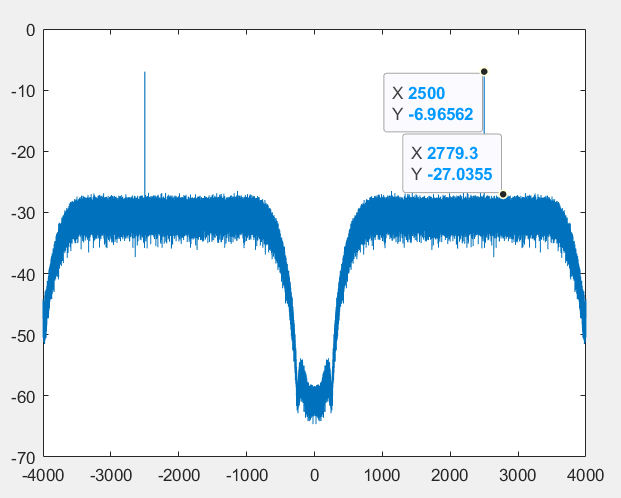
figure

f=linspace(-fm/2,fm/2,length(delta));

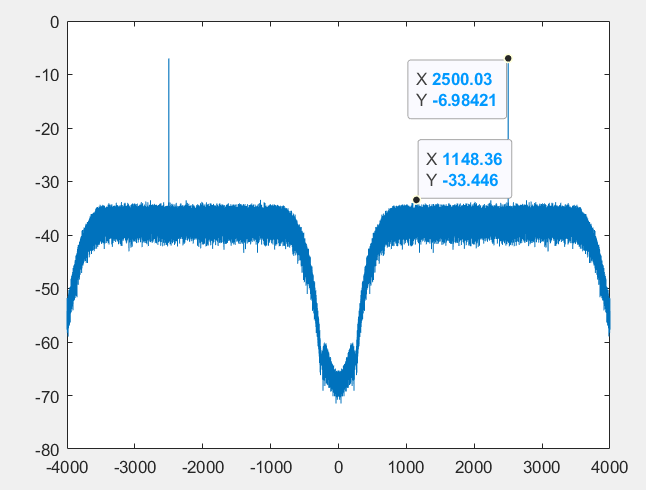
plot(f,10\*log10(H));

Para la obtención de la **relación señal a ruido**, se introduce un coseno por el canal y se ve la diferencia en espectro entre el pico de la frecuencia portadora y el resto del canal.

1. Canal1a:



1. Canal1b:



En el caso del canal 1a se obtiene una SNR de 20dB, y en el caso del canal 1b 27dB.

El código utilizado para esta parte es el siguiente:

fp=1500;

fm=8000;

mean\_window=5;

t\_sim= 100;

t=0:1/fm:t\_sim-1/fm;

x=cos(2\*pi\*fp\*t);

Y=abs(fft(canal1a(x,fm)));

Y\_mean=movmean(Y/max(Y),mean\_window);

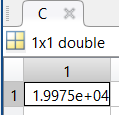
f=linspace(-fm/2,fm/2,length(x));

figure

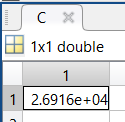
plot(f,10\*log10(Y\_mean));

Para obtener la **capacidad** del canal únicamente se aplica la fórmula vista en clase:

1. Canal1a:



1. Canal1b:



El código utilizado para esta parte es el siguiente:

B=3e3;

SNR\_dB=20;

SNR=10^(SNR\_dB/10);

C= B\*log2(1+SNR);

B=3e3;

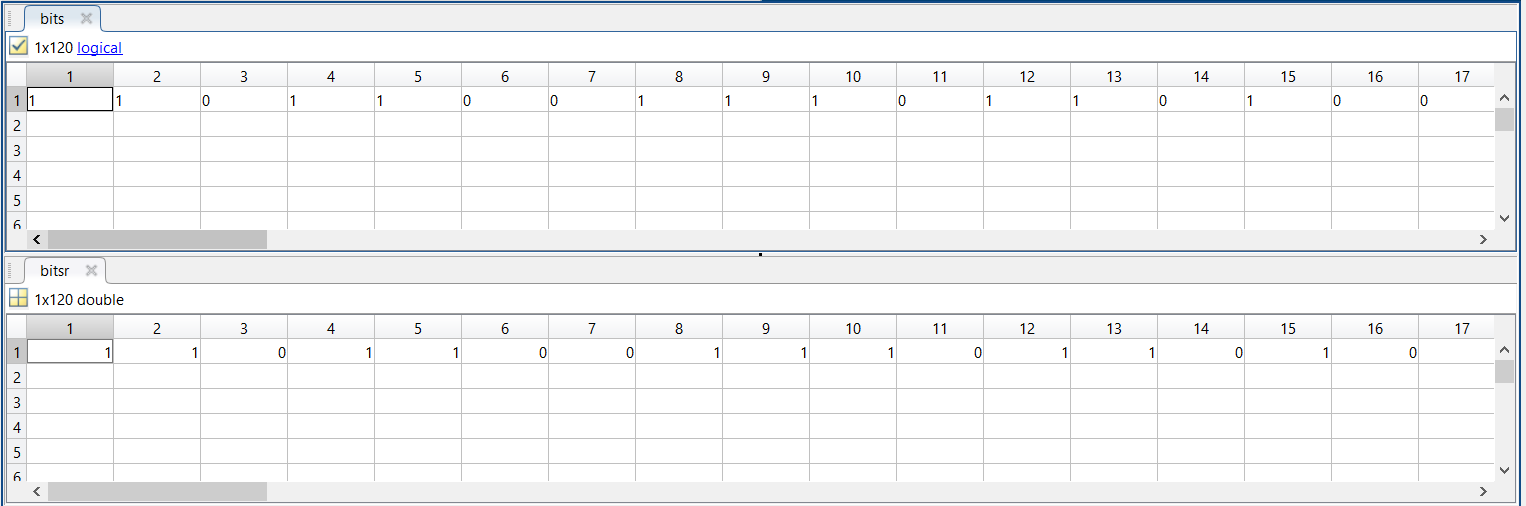
SNR\_dB=27;

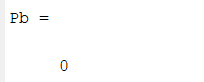
SNR=10^(SNR\_dB/10);

C= B\*log2(1+SNR);

1. **Sistema TCM**

Observando el código que se proporcionar para el sistema TCM, se comprueba que los bits originales y los recibidos no sufren ninguna modificación, y por consiguiente, la probabilidad de error es nula.

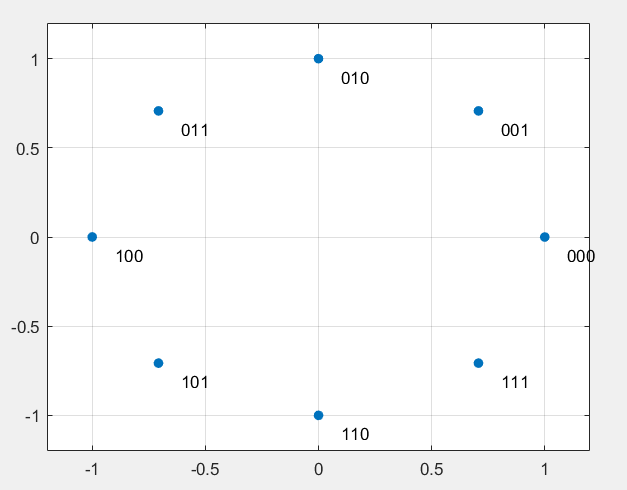
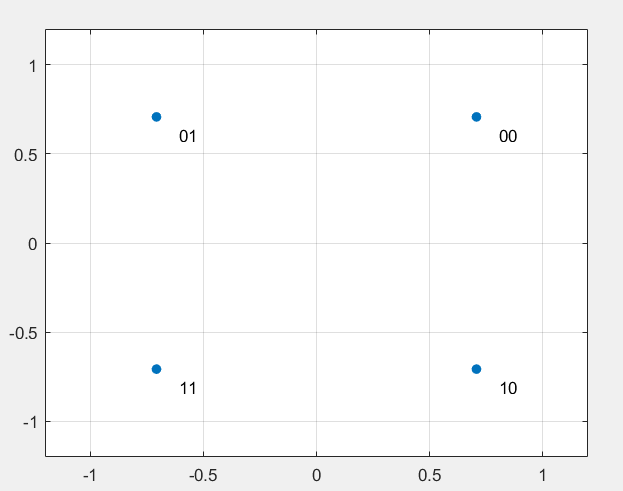




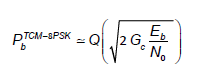
1. **Simulación Sistema TCM con canal AWGN**

* **Simulación 8PSK-TCM frente a 4PSK**

Las constelaciones que se van a comparar tienes la siguiente disposición:



De estas dos constelaciones se realiza el cálculo de la probabilidad de error y de la probabilidad de error teórica mediante la fórmula:



Donde Gc en este caso el ser 8PSK con TCM frente a 4PSK tiene un valor de 3dB.

Para realizar los cálculos se ha utilizado el siguiente código:

Pbtcm(i) = sum(bits ~= bitsr )/numero\_de\_bits; %tasa real de error

Pb4psk(i) = sum(bits ~= bitsr )/numero\_de\_bits; %tasa real de error

%--- Pb teorica

EbN0(i) = 10.^(EbN0dB(i)/10);

Pb\_teorica(i) = fQ(sqrt(2\*EbN0(i)\*(10^(3/10))));

figure,

semilogy(EbN0dB,Pbtcm)

hold on

semilogy(EbN0dB,Pb4psk)

hold on

semilogy(EbN0dB,Pb\_teorica)

legend("Pb 8PSK TCM en funcion de EbN0","Pb 4PSK en funcion de EbN0","Pb PSK teorica")

Donde los bits generados se crean de la siguiente manera:

%---8PSK

numero\_de\_bits = 30000;%multiplo de 3

bits = fuente(numero\_de\_bits);

%---4PSK

numero\_de\_bits=1000;%tiene que ser multiplo de 2

bits = fuente(numero\_de\_bits);

Y los bits recibidos son los conseguidos mediante las siguientes operaciones:

M=4;

constelacion = constelacion\_4psk();%constelacion

[Ik,Qk] = asignacion\_simbolos(bits,constelacion);

fp=1000;

fb=100;

fs = fb / log2(M);

fm = 2\*(fp + fb) \* 10;

% Pulso conformador rectangular

[pulso,retardo] = rcos(fm,fs,0.5,5);

[I,Q] = filtro\_tx(Ik,Qk,fm,fs,pulso);

[xI,xQ] = modulador(I,Q,fm,fp);

x=xI+xQ;

%%%% Canal gaussiano

Es = sum(pulso.^2);

Eb = Es/log2(M);

EbN0dB = 0:1:6;

for i=1:7

n = canal\_awgn(x,Eb,EbN0dB(i)); % Ruido generado

r=x+n;

%--- demodulacion

[xI,xQ] = demodulador(r,fm,fp);

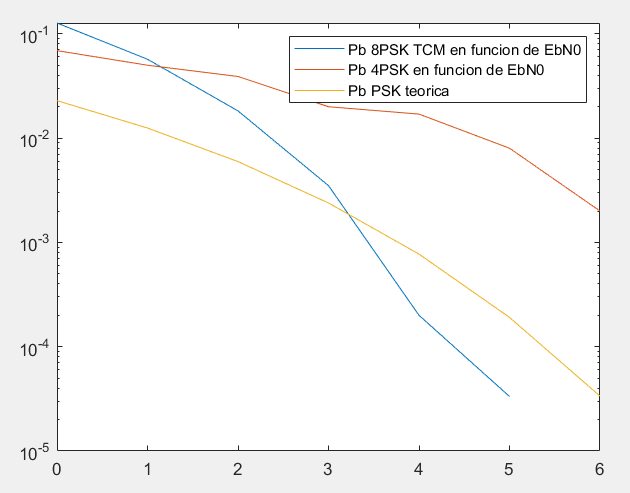
[I,Q] = filtro\_rx(xI,xQ,pulso);

[Ik,Qk] = muestreo(I,Q,fm,fs,retardo);

D = distancias(Ik,Qk,constelacion);

bitsr = decodificador\_map(D);

La representación de las 3 curvas queda de la siguiente manera:



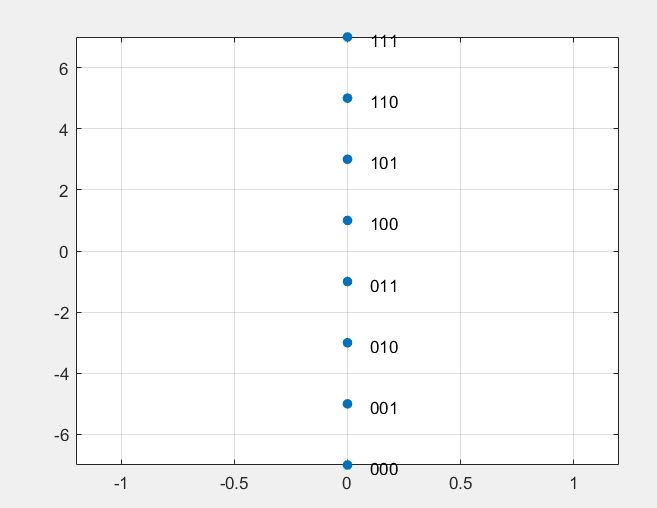
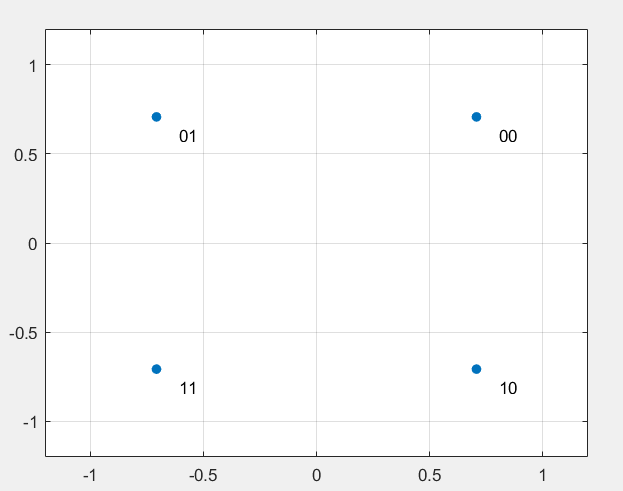
En esta figura se ve como la constelación 8PSK tiene una probabilidad de error menor a medida que aumenta la relación EbN0. Se observa como la probabilidad de error de la constelación 4PSK también disminuye según aumenta la relación EbN0, pero no cae tanto como con la 8PSK.

En cuanto a la probabilidad de error teórica se observa que cae también según aumenta la relación EbN0.

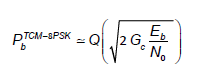
También se observa como la teórica tiene valores inferiores a la 4PSK, eso nos indica que la constelación 8PSK con TCM es mejor que la 4PSK.

* **Simulación constelación inventada frente a 4PSK**

Las constelaciones que se van a comparar tienes la siguiente disposición:



De estas dos constelaciones se realiza el cálculo de la probabilidad de error y de la probabilidad de error teórica mediante la fórmula:



Donde Gc en este caso el ser 8PFM con TCM frente a 4PSK tiene un valor de -0.7dB.

Para realizar los cálculos se ha utilizado el siguiente código:

Pbpfm(i) = sum(bits ~= bitsr )/numero\_de\_bits; %tasa real de error

Pb4psk(i) = sum(bits ~= bitsr )/numero\_de\_bits; %tasa real de error

%--- Pb teorica

EbN0(i) = 10.^(EbN0dB(i)/10);

Pb\_teorica(i) = fQ(sqrt(2\*EbN0(i)\*(10^(-0.7/10))));

figure,

semilogy(EbN0dB,Pbpfm)

hold on

semilogy(EbN0dB,Pb4psk)

hold on

semilogy(EbN0dB,Pb\_teorica)

legend("Pb 8PFM TCM en funcion de EbN0","Pb 4PSK en funcion de EbN0","Pb PSK teorica")

Donde los bits generados se crean de la siguiente manera:

%---8PSK

numero\_de\_bits = 30000;%multiplo de 3

bits = fuente(numero\_de\_bits);

%---4PSK

numero\_de\_bits=1000;%tiene que ser multiplo de 2

bits = fuente(numero\_de\_bits);

Y los bits recibidos son los conseguidos mediante las siguientes operaciones:

M=4;

constelacion = constelacion\_4psk();%constelacion

[Ik,Qk] = asignacion\_simbolos(bits,constelacion);

fp=1000;

fb=100;

fs = fb / log2(M);

fm = 2\*(fp + fb) \* 10;

% Pulso conformador rectangular

[pulso,retardo] = rcos(fm,fs,0.5,5);

[I,Q] = filtro\_tx(Ik,Qk,fm,fs,pulso);

[xI,xQ] = modulador(I,Q,fm,fp);

x=xI+xQ;

%%%% Canal gaussiano

Es = sum(pulso.^2);

Eb = Es/log2(M);

EbN0dB = 0:1:6;

for i=1:7

n = canal\_awgn(x,Eb,EbN0dB(i)); % Ruido generado

r=x+n;

%--- demodulacion

[xI,xQ] = demodulador(r,fm,fp);

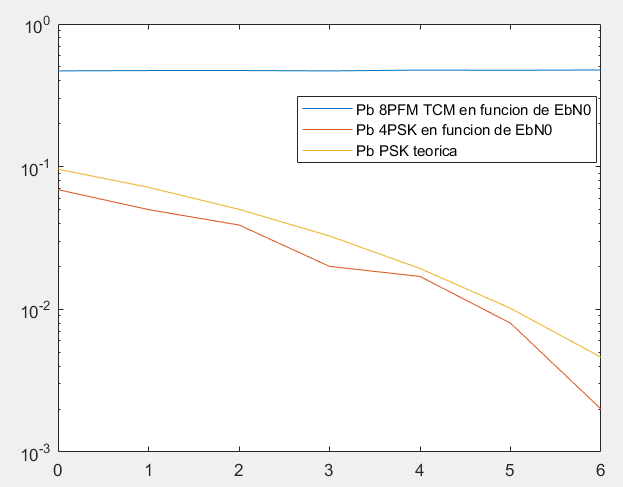
[I,Q] = filtro\_rx(xI,xQ,pulso);

[Ik,Qk] = muestreo(I,Q,fm,fs,retardo);

D = distancias(Ik,Qk,constelacion);

bitsr = decodificador\_map(D);

La representación de las 3 curvas queda de la siguiente manera:



En esta figura se ve como la constelación 8PFM tiene una probabilidad de error muy poco dependiente de la variación de EbN0, es decir, se mantiene prácticamente constante para todo el rango. Además se observa como el valor que toma es bastante elevado para ser una probabilidad de error, esto nos indica que la constelación inventada no es lo suficientemente buena.

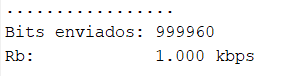
A su vez, se observa como la probabilidad de error de la constelación 4PSK disminuye según aumenta la relación EbN0, por lo tanto se deduce que la 4PSK es mejor que la inventada en el previo de la práctica 3.

En cuanto a la probabilidad de error teórica se observa que cae también según aumenta la relación EbN0.

En este caso, se observa como la teórica está por encima de la 4PSK, esto también es indicativo de que la constelación inventada es mala en comparación con la 4PSK.

1. **Estudio de canal1b**

Mediante el escenario dado se consigue el siguiente resultado:



Para ello, los parámetros utilizados han sido:

constelacion = constelacion\_4psk();

M = size(constelacion,1);

%--- fb, fp, fm

fp = 1350;

beta = 0.5;

ntaps = 10;

Rb = 1000;

fs = Rb/log2(M);

fm = fs\*20;

Variando los parámetros a los siguientes valores:

constelacion = constelacion\_tcm\_8psk();

M = size(constelacion,1);

%--- fb, fp, fm

fp = 1700;

beta = 0.15;

ntaps = 6;

Rb = 9250;

fs = Rb/log2(M);

fm = fs\*20;

Se ha conseguido la siguiente cantidad de bits enviados y por consiguiente, la siguiente Rb:

