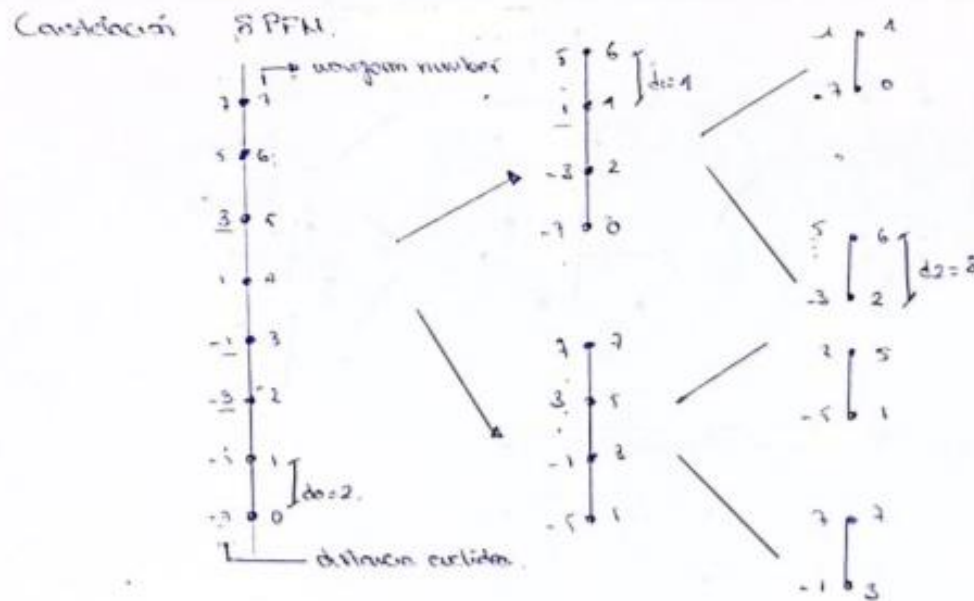
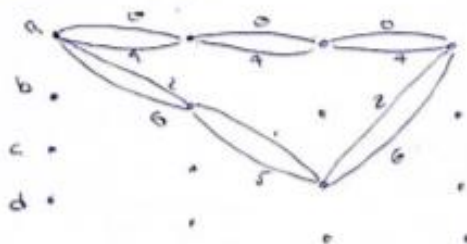


1. Estudio previo



Tickets 4 estados:



$$d_{a-a} = d_2^2 = 8^2 = 64$$

$$d_{a-b-c-a} = d_1^2 + d_0^2 + d_1^2 = 1^2 + 2^2 + 1^2 = 6$$

$$d_{free, reg}^2 = 36$$

$$\bar{E}_{s, reg} = \frac{2 \cdot (1^2 + 3^2 + 5^2 + 7^2)}{8} = 21$$

8 PSK → referencia →

sin TCM

$$d_{free, reg}^2 = d_0^2 = 0,765^2$$

$$\bar{E}_{s, reg} = 1$$

$$G_{constitucion} = \frac{d_{free, reg}^2 / \bar{E}_{s, reg}}{d_{free, reg}^2 / \bar{E}_{s, reg}}$$

$$= \frac{2/21}{0,765^2/1} = \frac{0,19}{0,585} = 0,33$$

$$-4,82dB$$

4 PSK → referencia →

con TCM

$$d_{free, reg}^2 = 2$$

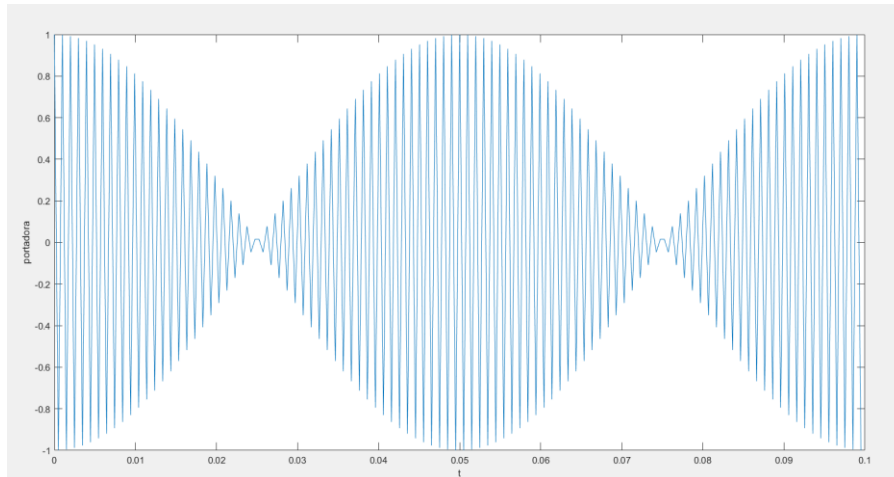
$$\bar{E}_{s, reg} = 1$$

$$G = \frac{d_{free, reg}^2 / \bar{E}_{s, reg}}{d_{free, reg}^2 / \bar{E}_{s, reg}} = \frac{2/21}{2/1} =$$

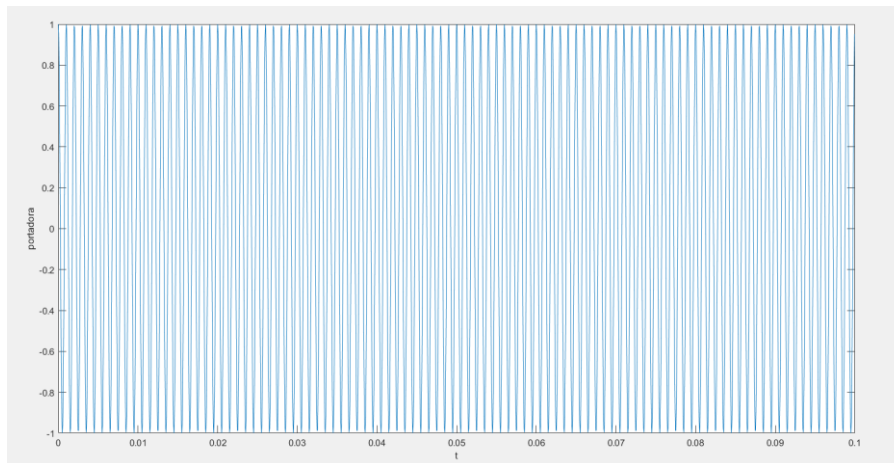
$$\frac{1,714}{2} = 0,857 \rightarrow -0,66dB$$

2. Portadora

En la figura que se muestra justo a continuación se muestra la portadora con una frecuencia de muestreo de 2020 Hz, y con valores de $\text{numero_de_bits} = 1$, $f_p = 1000$ y $f_b = 10$. Como se puede observar que se produce una traslación de la portadora por lo que se deduce que el periodo en ese momento de la señal no se encuentran en el máximo o mínimo. En la imagen se ve que conforme aumenta el tiempo los máximos y los mínimos de la señal se alejan.



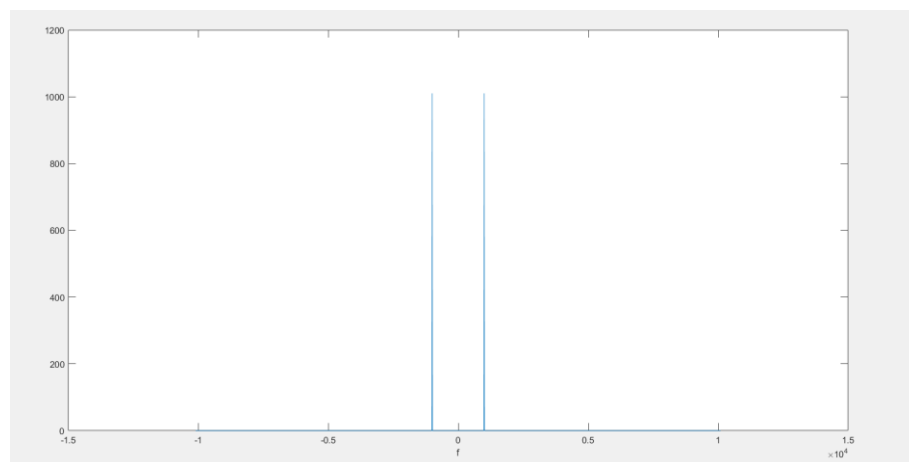
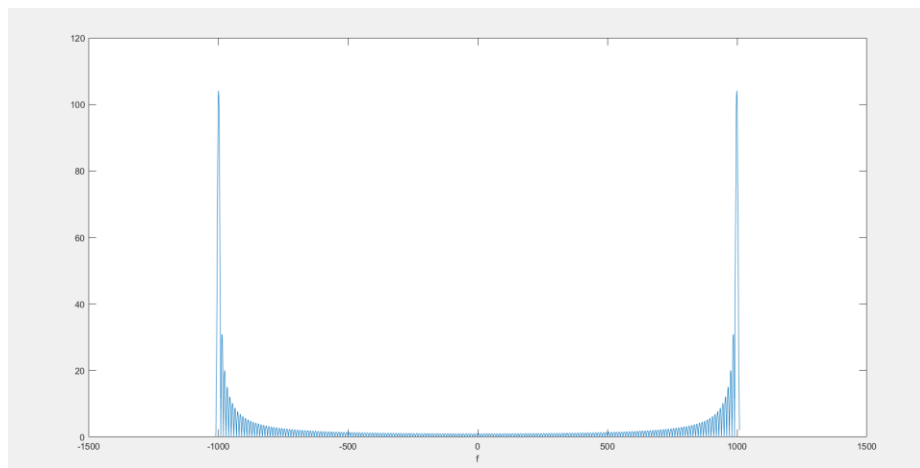
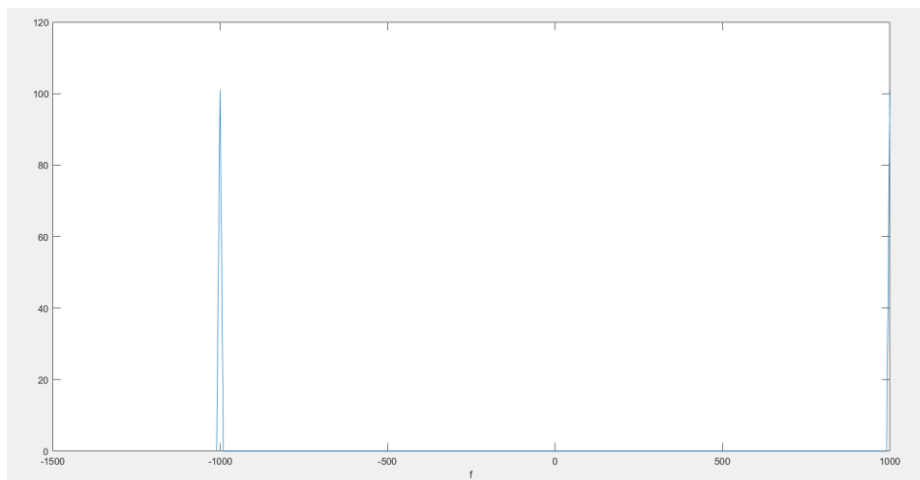
Para corregir el problema de que los máximos o los mínimos de la señal no se encuentren en los periodos, que debe multiplicar la frecuencia de muestreo (f_m) por un factor de 10, y así conseguiremos el resultado mostrado en la figura que se muestra a continuación. En ella, se observa como en los instantes de nuestros se localizan siempre los máximos o mínimos de la señal.



A continuación, se muestran tres imágenes, donde en la primera se muestra la FFT de la señal portadora con unos valores de $f_p = 1000$ y $f_b = 10$. En ella se observa como los picos (deltas) se localizan en $-f_p$ y f_p con una anchura de f_b .

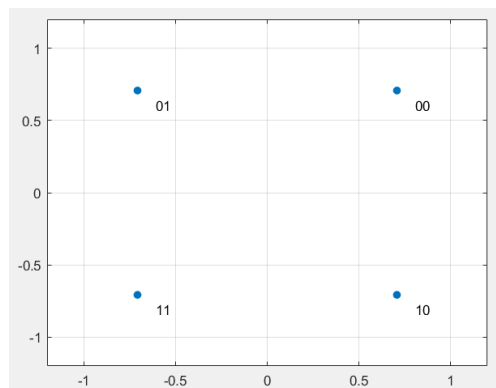
En la segunda imagen, se observa la FFT con un sobremuestreo de la señal en frecuencia, es decir, ya no se dispone solo de las dos deltas en $-f_p$ y f_p , si no que se dispone de más datos entre ellas.

Por último, en la tercera imagen se observa la FFT con un sobremuestreo de la señal en tiempo, en la cual se observa que aparecen dos deltas igual que en el primer caso, pero con un aumento en su amplitud con un factor 10



3. Modulador

En la siguiente figura se muestra una constelación de 4 símbolos. Hay que mencionar que se ha utilizado una asignación Gray, lo que implica que la distancia entre símbolos es la mínima.



4. Demodulador

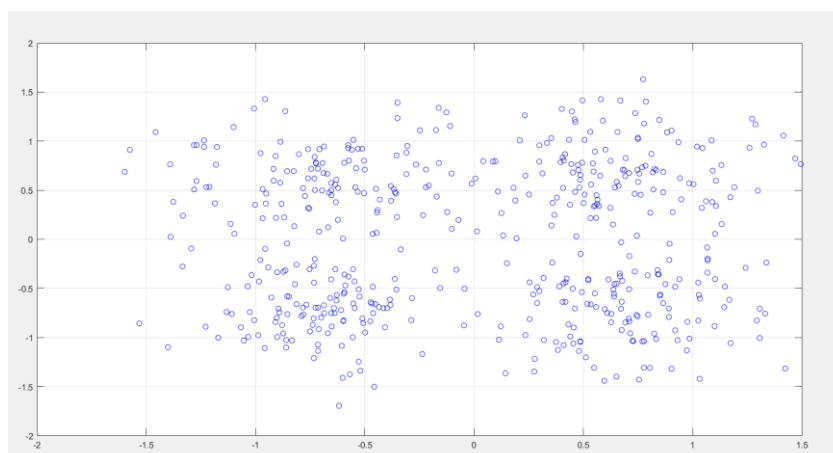
En la siguiente imagen se observa como al no tener canal, los bits que se reciben son igual a los originales, por lo que, en consecuencia, la probabilidad de error es nula.

bits												
1x12 logical												
1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	0	1	0	1	1	0	0	0	1	0	1
bitsr												
1x12 double												
1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	0	1	0	1	1	0	0	0	1	0	1
2												
3												
Pb												
1x1 double												
1	2	3	4	5	6	7	8	9	10	11	12	13
1	0											
2												
3												

5. Canal AWGN

Se observa como la probabilidad experimental es superior a la teórica, y además se muestra la constelación recibida en el canal AWGN

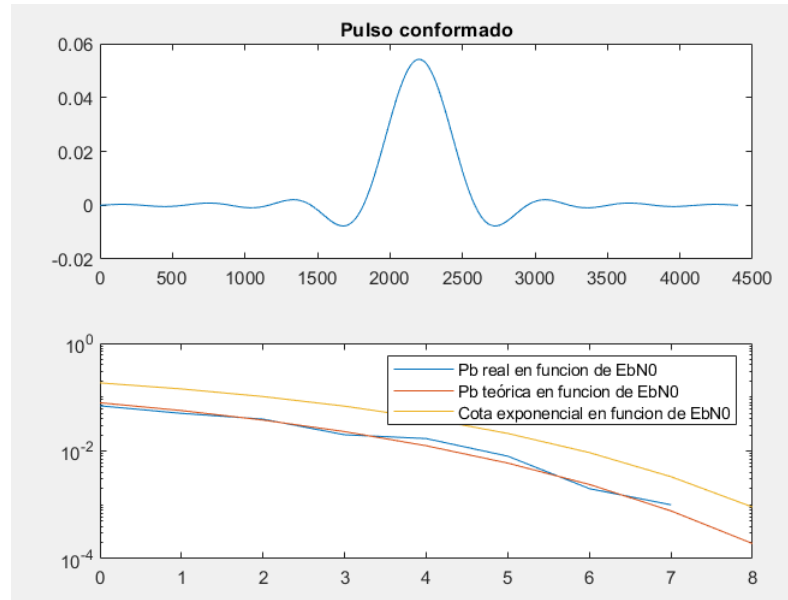
numero_de...	1000
Pb	0.0190
Pb_teorica	0.0072



6. Simulación 4PSK

En la imagen superior se puede observar el filtro de raíz de coseno realizado que se ha utilizado en el apartado. Dicho coseno genera un retardo de 4401 muestras.

Observando la gráfica restante se deduce que la probabilidad de error experimental oscila en torno a la probabilidad de error teórica y se encuentra por debajo de la cota exponencial.



El código utilizado para el cálculo de dichas curvas es el mostrado a continuación, donde se ha realizado un bucle para distintos valores de E_b/N_0 y así conseguir ver la tendencia de las distintas probabilidades de error.

```
numero_de_bits=1000;%tiene que ser multiplo de 2
bits = fuente(numero_de_bits);
M=4;
constelacion = constelacion_4psk();
Eav=mean( sqrt(sum(constelacion.^2,2)));
[Ik,Qk] = asignacion_simbolos(bits,constelacion);
ver_constelacion(constelacion);
fp=1000;
fb=100;
fs = fb / log2(M);
fm = 2*(fp + fb) * 10;
[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;
Es = sum(pulso.^2);
Eb = Es/log2(M);
EbNdB = 0:1:8;
for i=1:9
n = canal_awgn(x,Eb,EbNdB(i)); % Ruido generado
r=x+n;
[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);
bits;
Pb(i) = sum(bits ~= bitsr )/numero_de_bits;
EbN0(i) = 10.^(EbNdB(i)/10);
Pb_teorica(i) = fQ(sqrt(2*EbN0(i)));
Pb_cotaexp(i) = 1/2 * exp(-EbN0(i));
end
```

Y por último, para la representación se ha utilizado el siguiente código, donde en una misma gráfica de muestran tres curvas diferentes.

```
figure,  
subplot(2,1,1);plot(pulso), title("Pulso conformado")  
subplot(2,1,2);semilogy(EbN0dB,Pb)  
hold on  
semilogy(EbN0dB,Pb_teorica)  
hold on  
semilogy(EbN0dB,Pb_cotaexp)  
legend("Pb real en funcion de EbN0", "Pb teórica en funcion  
de EbN0", "Cota exponencial en funcion de EbN0")
```

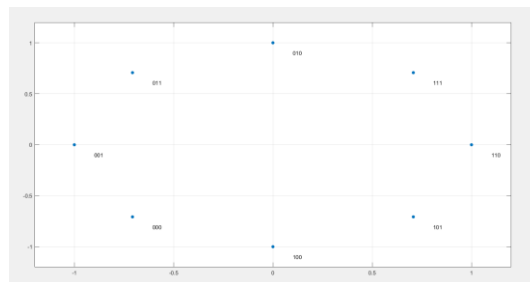
7. Simulación 8PSK

Para la generación de la constelación se ha utilizado el siguiente código:

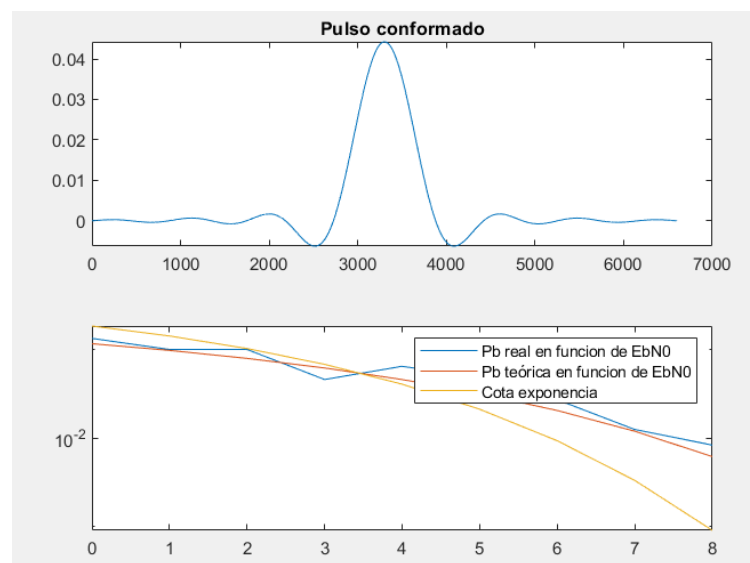
```
function constelacion = constelacion_8psk()  
theta = [5 4 2 3 6 7 0 1]*pi/4;  
constelacion = [cos(theta)' sin(theta)'];
```

A su vez, hay que tener en cuenta que en nº de bits utilizados deber ser múltiplo de 3 en este caso.

Teniendo en cuenta todo esto, se obtiene la siguiente constelación:



En la imagen superior se puede observar el filtro de raíz de coseno realizado que se ha utilizado en el apartado. Observando la gráfica restante se deduce que la probabilidad de error experimental oscila en torno a la probabilidad de error teórica.



En este caso, el código utilizado es el mismo exceptuando la expresión de la probabilidad de error teórica, que en este caso es la siguiente:

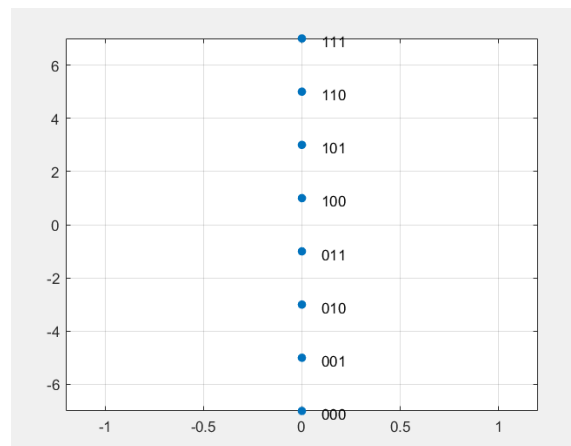
$$P_{b_teorica}(i) = (2/\log_2(M)) * f_Q(\sqrt{2 * E_b N_0(i) * \log_2(M)}) * \sin(\pi/M));$$

8. Simulación constelación previo

Para la generación de la constelación se ha utilizado el siguiente código:

```
function constelacion = constelacion_8pfm()  
x = [-7 -5 -3 -1 1 3 5 7];  
y = [0 0 0 0 0 0 0 0];  
constelacion = [y' x'];
```

A su vez, hay que tener en cuenta que en nº de bits utilizados deber ser múltiplo de 3 en este caso. Teniendo en cuenta todo esto, se obtiene la siguiente constelación:



Hay que mencionar que se ha utilizado el zoom out una vez se ha creado la figura ya que los pesos utilizados sobrepasan el rango de -1 a 1 y por lo tanto, no los muestra directamente.