



Maestría en Ingeniería Eléctrica especialización Telecomunicaciones

Comunicaciones Digitales

## Tarea #3

***Luis Emilio Tonix Gleason***

***Fernando Alberto Madera Torres***

*13/04/2022*

***Dr. Ramon Michel Parra***

## Tabla de contenido

Ejercicio 1 .....	3
Señal modulada ASK.....	3
Demodulacion ASK por envolvente.....	5
Ejercicio 2.....	10
Señal modulada BPSK .....	10
Filtro formador de Coseno Elevado.....	11
Demodulacion BPSK .....	13
Ejercicio 3.....	15
Ejercicio 4.....	19
Ejercicio 5.....	19

# Ejercicio 1

Realice una simulación de una señal modulada en ASK con filtro formador cuadrado. Usted especifique los valores de la frecuencia de los bits y la frecuencia portadora; considere una desmodulación por envolvente (pase la señal por un circuito recortador seguido de un filtro pasa bajas), grafique la señal demodulada.

## Señal modulada ASK

La señal por desplazamiento de amplitud también es conocida por cierre y apertura, consiste en activar y desactivar una señal portadora senoidal con una señal binaria unipolar.

Es una modulación binaria unipolar en una DSB-SC.

La señal ASK está dada por  $S(t)$

$$s(t) = A_c m(t) \cos \omega_c t$$

% The number of bits to send - Frame Length

N = 15;

time0 = 0:1:1514;

% Sampling rate - This will define the resolution

fs = 100;

% Generate a random bit stream

bit\_stream = round(rand(1,N));

% Enter the two Amplitudes

% Amplitude for 0 bit

A1 = 0;

% Amplitude for 1 bit  $A_c$  in the diagram is the  $A_2$  in the code

A2 = 5; % esta es la amplitud total, por lo cual será de -2.5v a 2.5v

% Frequency of Modulating Signal

f = 1;

% Time for one bit

% Cual es el tiempo de Signo 0.01 seg

t = 0: 1/fs : 1;

% This time variable is just for plot

time = [];

ASK\_signal = [];

Digital\_signal = [];

% Frequency of Carry Signal es 100 Hz

carry\_signal = cos(2\*pi\*f\*t);

% Ciclo para rellenar la señal ASK acorde al tamaño del bit stream

for ii = 1: 1: length(bit\_stream)

% The ASK Signal

ASK\_signal = [ASK\_signal (bit\_stream(ii)==0)\*A1/2\*carry\_signal+...  
(bit\_stream(ii)==1)\*A2/2\*carry\_signal];

% The Original Digital Signal

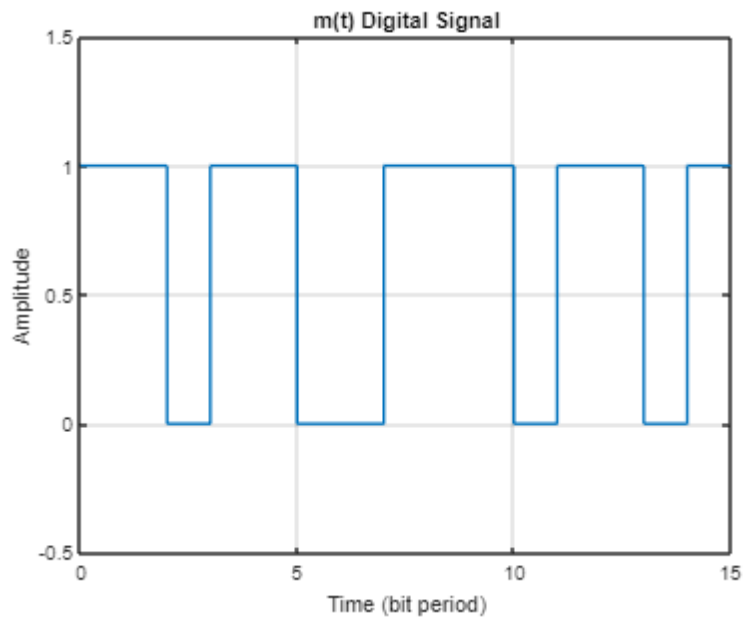
Digital\_signal = [Digital\_signal (bit\_stream(ii)==0)\*...  
zeros(1,length(t)) + (bit\_stream(ii)==1)\*ones(1,length(t))];

```

time = [time t];
t = t + 1;

end
%sorry for the inconvenience but time was defined in the for loop
%freq axis
time_res = 100/(length(time)-1);
% Plot the m(t) Digital Signal
plot(time,Digital_signal);
xlabel('Time (bit period)');
ylabel('Amplitude');
title('m(t) Digital Signal');
axis([0 time(end) -0.5 1.5]);
grid on;

```



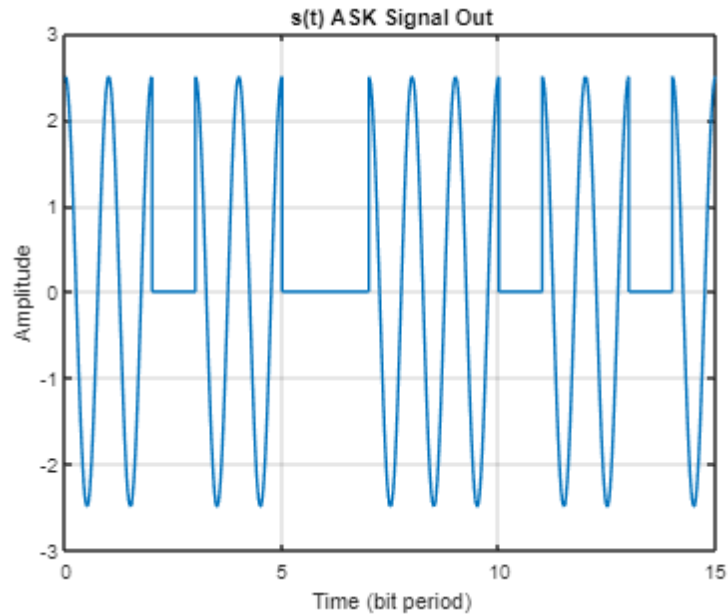
**%FALTA LA VENTANA DE INTERFERENCIA SIMBOLICA**

La señal ASK tendrá un rango de valores de -2.5V a 2.5V

```

% Plot the ASK Signal
plot(time,ASK_signal);
xlabel('Time (bit period)');
ylabel('Amplitude');
title('s(t) ASK Signal Out');
axis([0 time(end) -3 3]);
grid on;
hold off;

```



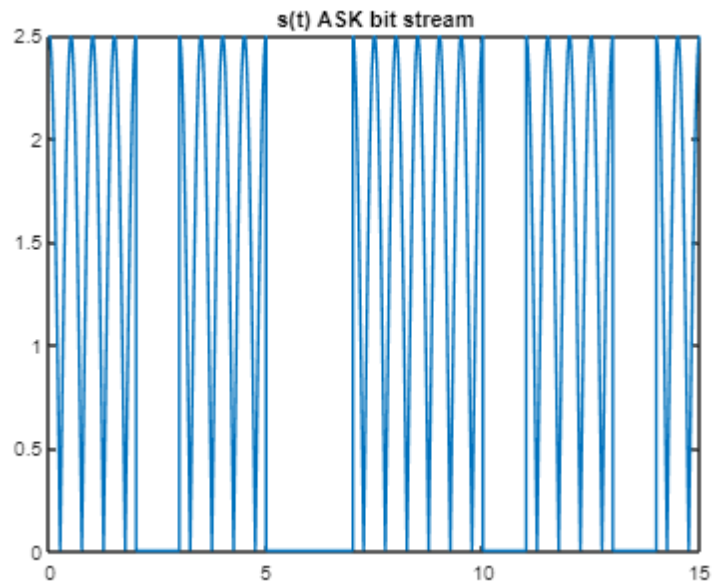
### Demodulacion ASK por envolvente

% Sacar el Valor Absoluto de l señal modulada

```
ask_signal_in = abs(ASK_signal);
```

```
plot(time,ask_signal_in);
```

```
title('s(t) ASK bit stream');
```



% convertir a frecuencia

```
freq0 = -fs/2:time_res:fs/2;
```

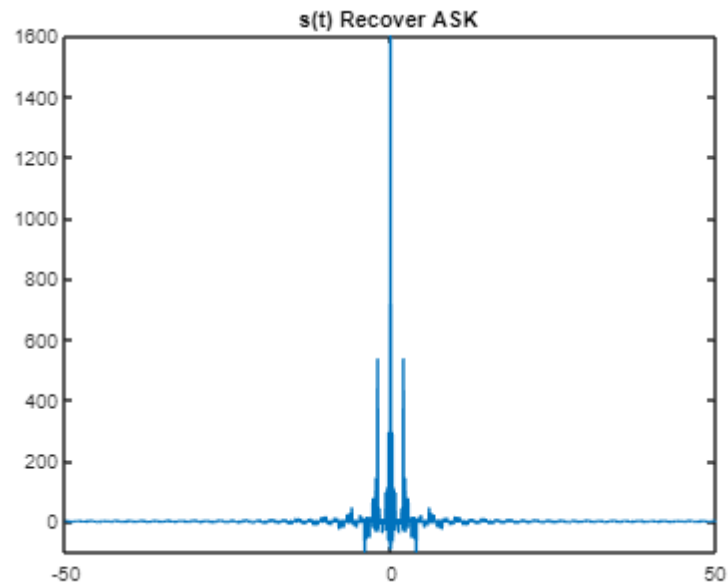
```
ask_signal_freq = real(ttof(ask_signal_in));
```

```
plot(freq0,ask_signal_freq);
```

```
title('s(t) Recover ASK ');
```

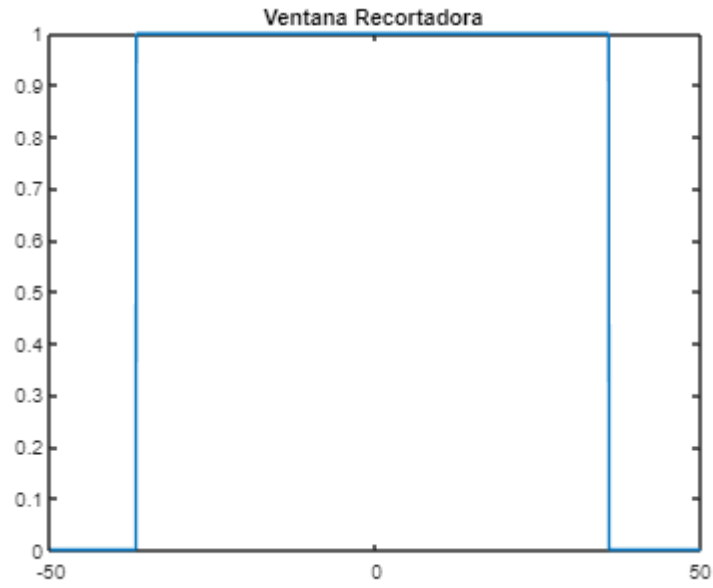
% Tiempos en espectro señal ASK recuperada

```
xlim([-fs/2 fs/2])
ylim([-100 1600])
```



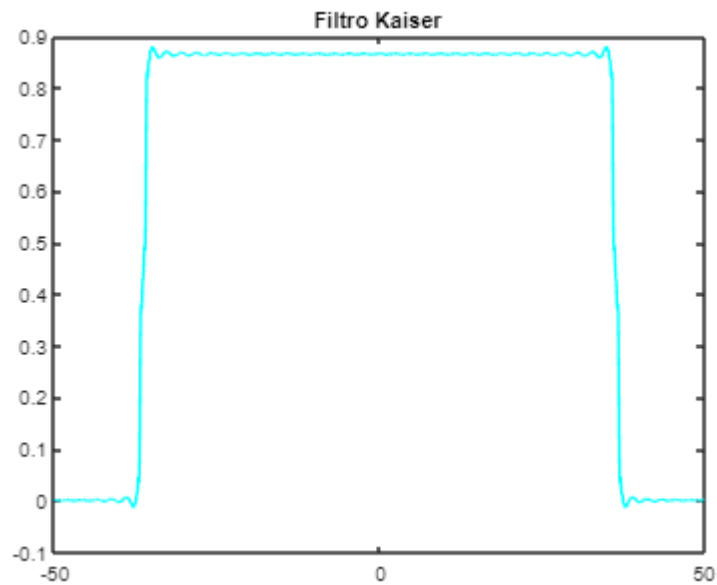
```
% para recortar la señal y solo agarrar DBLSC
time_filter = 0:1:300;
time_recover_signal = 0:1:1814;
% ask_rescaled = rescale(ask_signal_freq,-10,1);
% plot(freq0,ask_signal_freq);
% title('s(t) Recover ASK rescaled ');
```

```
% DEFINICION CIRCUITO RECORTADOR KAISER
%freq = -15:1/50:15;
frec = -fs/2:time_res:fs/2;
lenf = length(frec);
x = zeros(1,lenf);
x(201:1301) = 1;
plot(frec,x);
title('Ventana Recortadora ');
```

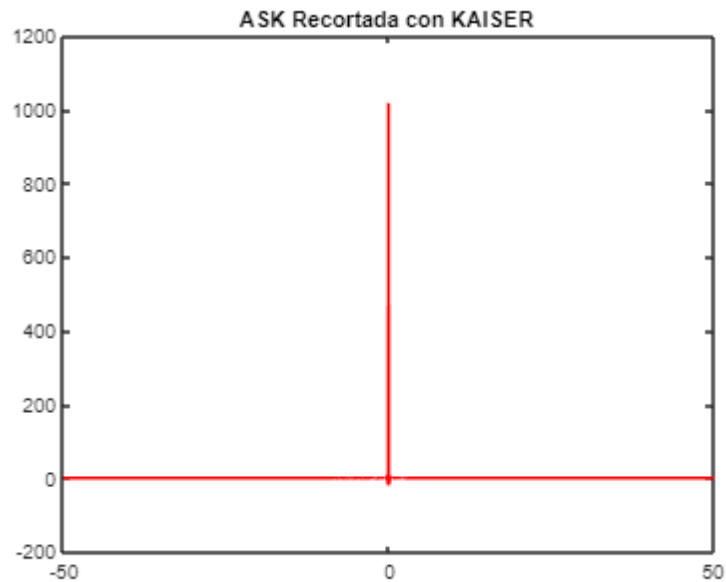


#### % FILTRO KAISER CONVOLUCION

```
y = real(ftot(x));
x2=zeros(1,lenf);
x2=y;
ventanakaiser = zeros(1,lenf);
%ventanakaiser (301:400) = kaiser(100,8.85); %ventana kaiser 100 coeficientes
ventanakaiser (401:800) = kaiser(400,1); %ventana kaiser 100 coeficientes
filtrokaiser = x2.*ventanakaiser;
plot(frec,real(ttof(filtrokaiser)),'c');
title('Filtro Kaiser');
```

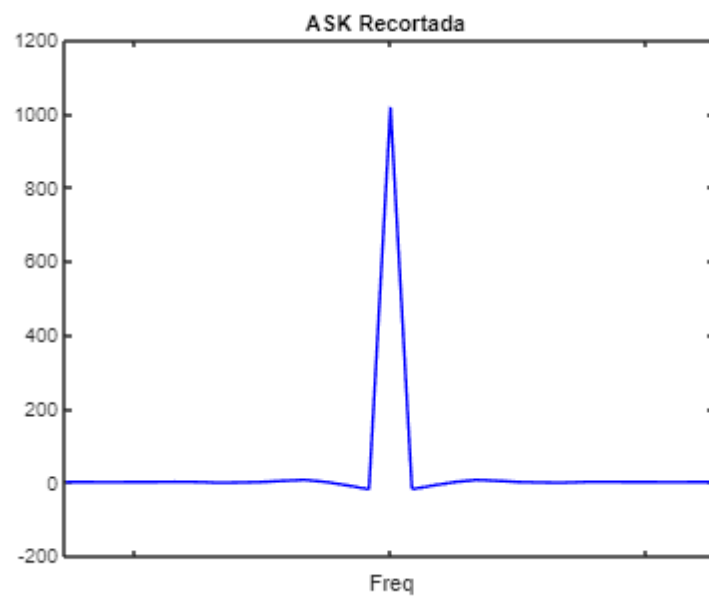


```
% convolucion con Filtro
ask_signal_filtered = filtrokaiser .* ask_signal_freq;
plot(frec,ask_signal_filtered,'r');
title('ASK Recortada con KAISER');
```



```
ask_rescaled = find(ask_signal_filtered ~= 0);
time_X = -200:1:199;
plot(frec,ask_signal_filtered,'b');
title('ASK Recortada ');
xlim([-1 1])
xticks([-pi -pi*3/4 -pi/2 -pi/4 0 pi/4 pi/2 pi*3/4 pi])
xticklabels({'-10','10'})
xlabel('Freq')
```





## Ejercicio 2

Dada una señal modulada en BPSK a 1 símbolo por segundo con filtro cuadrado, multiplíquela por una portadora de 25Hz para tener una señal modulada de amplitud.

- Realice el mismo procedimiento considerando un filtro formador de Coseno elevado con factor de banda de exceso=.5.
- Hacer demodulación coherente por multiplicación y filtrado.
- Hacer demodulación coherente considerando que la portadora tiene:
  - un corrimiento de fase de  $\pi/4$ ,  $\pi/2$ ,  $3\pi/4$  y  $\pi$
  - Una desviación en frecuencia de .1Hz. Comente sus observaciones.

### Señal modulada BPSK

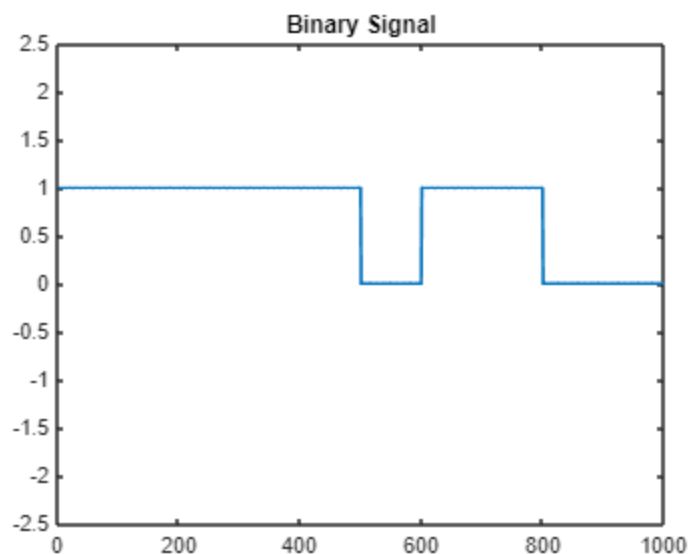
transmisión por desplazamiento de fase binaria, consiste en desplazar la fase de una portadora senoidal de 0 a 180° con una señal binaria unipolar, es una señalización PM con una forma de onda digital.

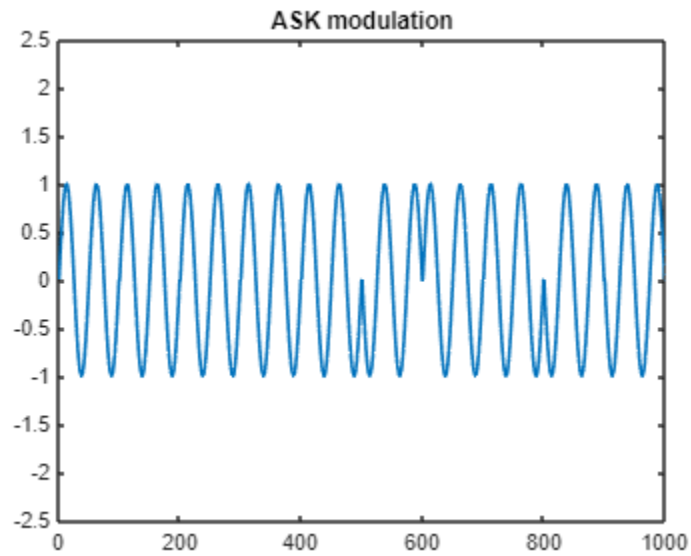
es una Señal DSB-SC con una forma de onda digital polar.

La señal BPSK está dada por  $S(t)$

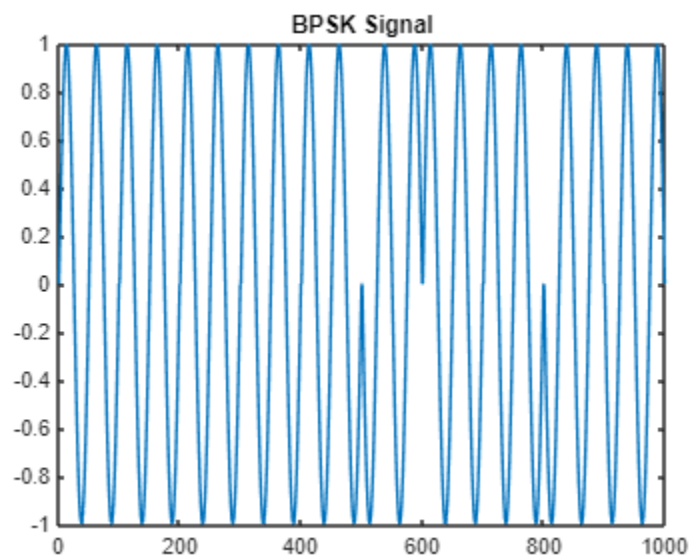
$$s(t) = A_c \cos [\omega_c t + D_p m(t)]$$

```
% The number of bits to send - Frame Length
N      =      10;
time0   =      1:1:1000;
% Sampling rate - This will define the resolution
fs      =      90;
% Generate a random bit stream
bit_stream = round(rand(1,N));
bpsk_signal = BPSK(bit_stream,2);
```





```
figure()  
plot(time0, bpsk_signal);  
title(['BPSK Signal']);
```



### Filtro formador de Coseno Elevado

% para recortar la señal y solo agarrar DBLSC

```
time_filter = 0:1:500;
```

```
time_filter_2 = 0:1:300;
```

```
filtro_rsine = rcosine(1,50);
```

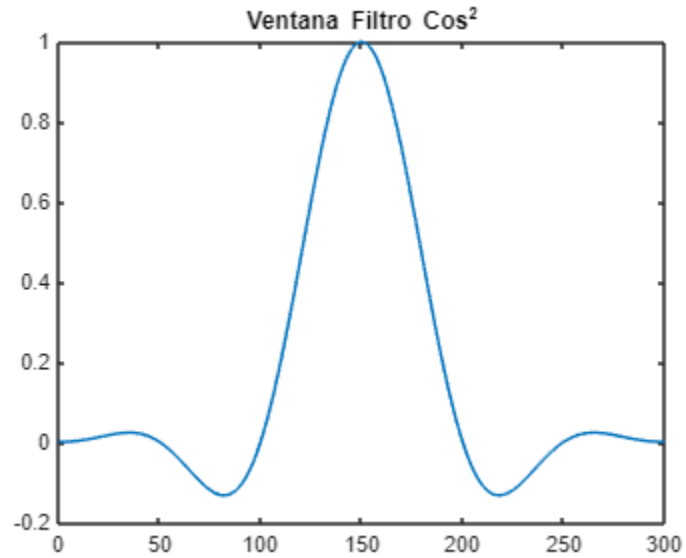
Warning: rcosine will be removed in a future release. Use rcosdesign instead.

Warning: rcosfir will be removed in a future release. Use rcosdesign instead.

%VENTANA FILTRO

```
plot(time_filter_2, filtro_rsine);
```

```
title('Ventana Filtro Cos^2');
```



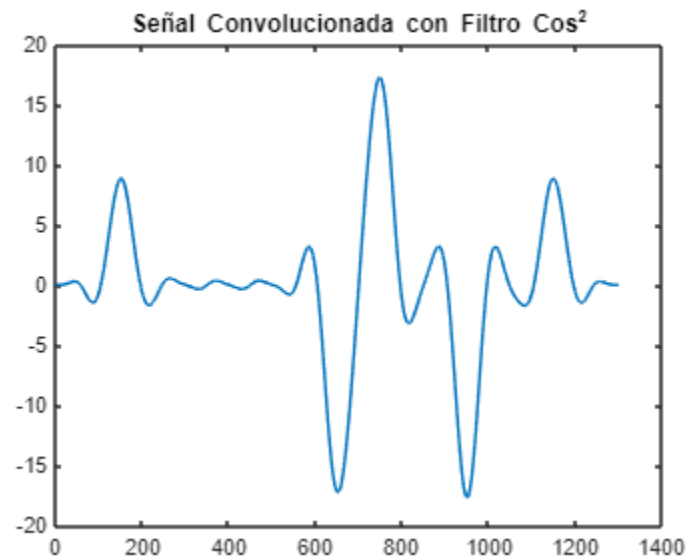
```
% CONVOLUTION WITH THE WINDOW
```

```
time_recover_signal = 1:1:1300;
```

```
res = conv(bpsk_signal,filtro_rsine);
```

```
plot(time_recover_signal, res);
```

```
title('Señal Convolucionada con Filtro Cos2');
```



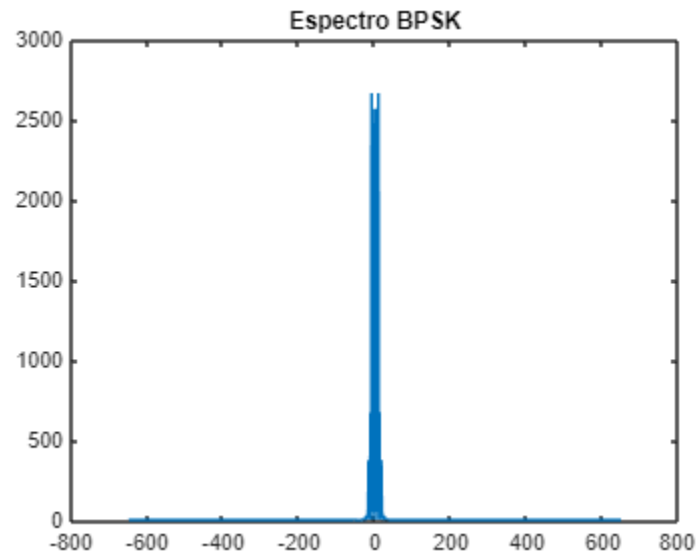
```
% graficado en frecuencia
```

```
freq = -649 : 1 : 650;
```

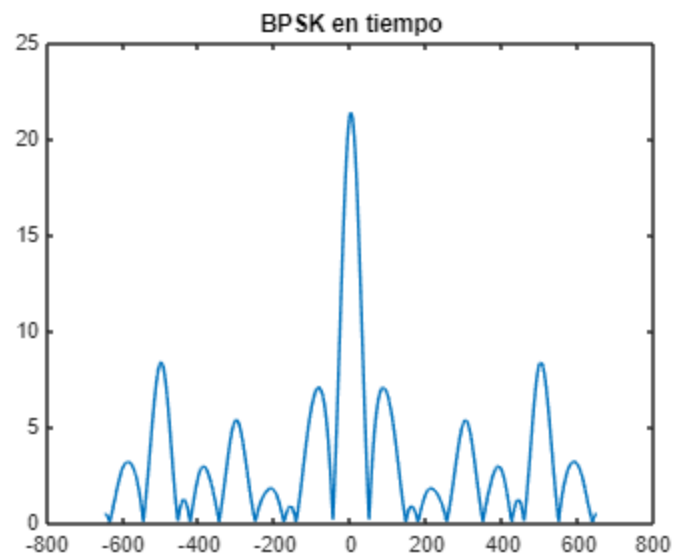
```
bpsk_recover_sig = abs(real(ttoft(res)));
```

```
plot(freq, bpsk_recover_sig);
```

```
title('Espectro BPSK');
```

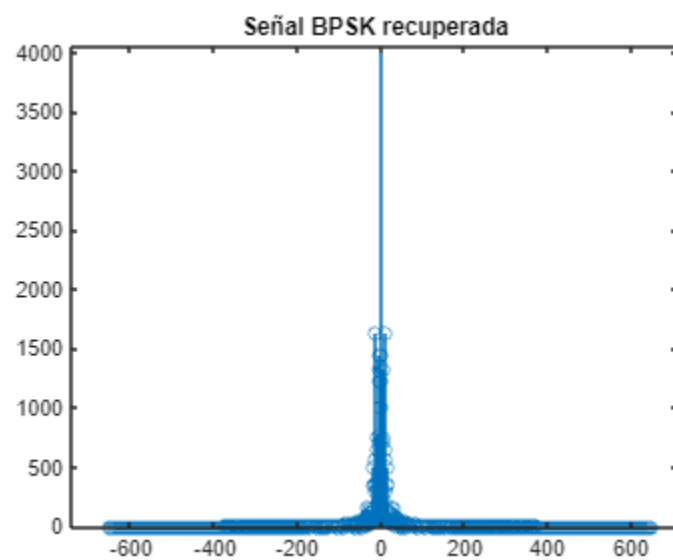


```
% freq to time
bpsk_signal_time = abs(real(ftot(bpsk_recover_sig)));
plot(freq, bpsk_signal_time);
title('BPSK en tiempo');
```



#### Demodulacion BPSK

```
time_recover_signal_0 = -649:1:650;
stem(time_recover_signal_0, abs(real(ttof(bpsk_signal_time))));
title('Señal BPSK recuperada');
```



### Ejercicio 3

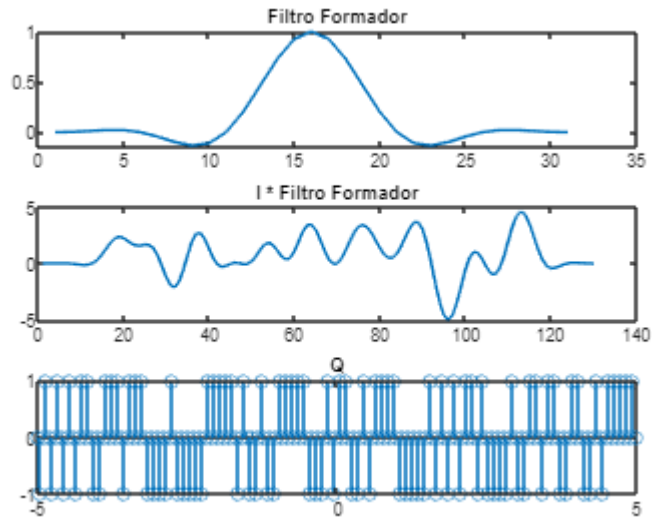
Genere una señal 4QAM con filtros formadores de coseno elevado con las condiciones señaladas en la pregunta 2 (Usted elija la constelación que prefiera, pero justifique su elección). Envíe al menos 100 símbolos.

- Presente el diagrama esquemático del modulador y del demodulador.
- Presente las formas de onda obtenidas en puntos relevantes del sistema descrito.
- Realice demodulación coherente y después del muestreo y normalización de los símbolos recibidos (muestreados a los instantes óptimos), grafique el diagrama de constelación de la señal recibida (todos los puntos sobrepuestos mediante scatterplot.m) y compárela con la constelación de la señal transmitida (multiplique la señal en cuadratura por  $j$  y súmela a la señal en fase antes de usar scatterplot.m).

#### Señal modulada 4QAM

```
%% Manejo de Bits (Q & I)
figure(1)
S = 100;
BpS = 2;
Bits = 2*(round(rand(1, BpS*S)) - 0.5);
stem(Bits)
title('Bits');
I = zeros(1, S);
Q = zeros(1, S);
j = 1;
for i = 1 : S*BpS
    if(mod(i,2))
        I(j) = Bits(i);
    else
        Q(j) = Bits(i);
        j = j + 1;
    end
end
f = 10;
tc = -5 : 1/(2*f) : 5;

Idis = zeros(1, length(tc));
Qdis = zeros(1, length(tc));
cero = find(tc == 0);
Idis(cero - 10*f + 1 : 2 : cero + 10*f - 1) = I;
Qdis(cero - 10*f + 1 : 2 : cero + 10*f - 1) = Q;
stem(tc, Idis)
title('I');
stem(tc, Qdis)
title('Q');
```



### Filtro formador Coseno Elevado

% 1 simbolo por segundo

% Multiplicacion señal modulada amplitud

% portadora 25 Hz

%% Filtro Formador

figure(2)

filfor = rcosine(1,5);

Warning: rcosine will be removed in a future release. Use rcosdesign instead.

Warning: rcosfir will be removed in a future release. Use rcosdesign instead.

plot(filfor);

title('Filtro Formador');

Ifilfor = conv(I, filfor);

Qfilfor = conv(Q, filfor);

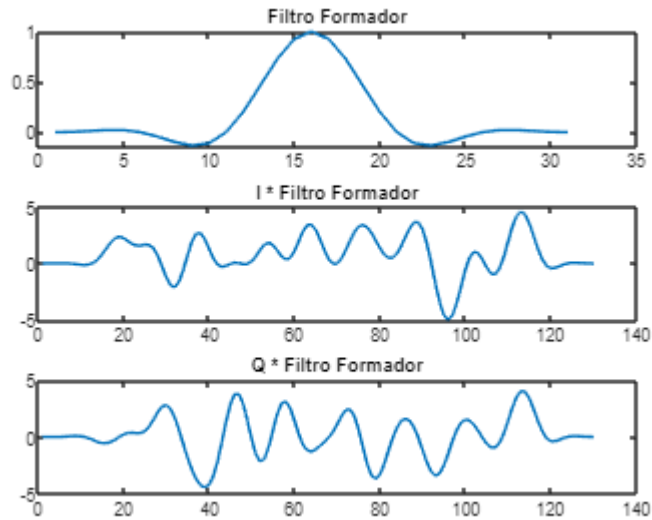
plot(Ifilfor);

title('I \* Filtro Formador');

plot(Qfilfor);

title('Q \* Filtro Formador');





%% Modulación

```
figure(3)
tr = (length(Qfilfor)/2)/f;
t = -tr : 1/f : tr - 1/f;
A = 1;
portI = A*cos(2*pi*f*t);
portQ = A*-cos(2*pi*f*t);
Imod = Ifilfor.*portI;
Qmod = Qfilfor.*portQ;
plot(t, Imod);
title('Portadora X I * F');
plot(t, Qmod);
title('Portadora X Q * F');
mod4QAM = Imod + Qmod;
plot(t, mod4QAM);
title('4QAM');
```

### Demodulacion Coherente 4QAM

```
% multiplicacion y filtrado
% corrimiento de fase portadora pi/4, pi/2, 3pi/4 y pi
% desviacion frecuencia .1Hz
% observaciones
%% Demodulación
filpb = zeros(1, length(mod4QAM));
filpb(find(filpb <= f) : find(filpb >= -f)) = 1;
Ires = mod4QAM.*portI;
Qres = mod4QAM.*portQ;
Iresf = ifftshift(Ires);
Iresf = fft(Iresf);
Iresf = fftshift(Iresf);
Ifilpbf = Iresf.*filpb;
```

```
Ifilpb = ifftshift(Ifilpbf);  
Ifilpb = fft(Ifilpb);  
Ifilpb = fftshift(Ifilpb);  
Qresf = ifftshift(Qres);  
Qresf = fft(Qresf);  
Qresf = fftshift(Qresf);  
Qfilpbf = Qresf.*filpb;  
Qfilpb = ifftshift(Qfilpbf);  
Qfilpb = fft(Qfilpb);  
Qfilpb = fftshift(Qfilpb);
```

## **Ejercicio 4**

Considere para la señal QAM generada en la pregunta anterior e introduzca:

- a. Corrimientos de fase (error de sincronización de fase de portadora) de 0, 45, 90, 135 y 180 grados.
- b. Corrimientos de frecuencia de portadora de .01Hz, .1Hz, 1Hz.
- c. Para cada uno de los casos anteriores grafique el diagrama de constelación de la señal recibida (todos los puntos sobrepuestos) y compárela con la constelación de la señal transmitida.

## **Ejercicio 5**

Considere para la señal QAM generada en la pregunta 3 y repítala agregando ruido con al menos dos diferentes relaciones señal a ruido (SNR) dentro de  $0 < \text{SNR} < 10\text{dB}$ . Grafique las formas de onda obtenida y comente los resultados.