INFORME DE LA PRACTICA 5

Diego Ismael Antolinos García
Andrés Ruz Nieto

En esta práctica veremos cómo abordar con MATLAB los distintos procesos de modulación, demodulación, muestreo y reconstrucción de una señal en tiempo continuo. Partiendo de una señal pregenerada, veremos cómo conseguir esto pasando por varias etapas.

1. Producto de señales: Modulación

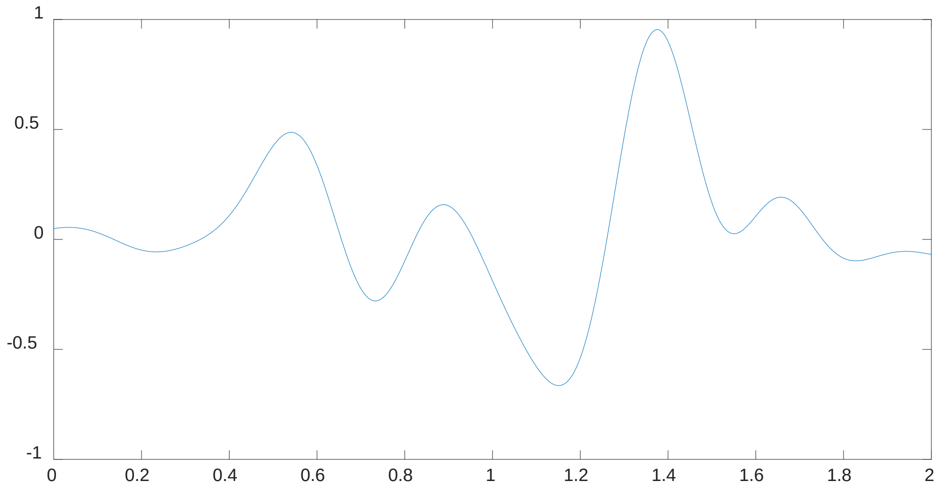
Como ya sabemos:

$$r(t) = s(t)p(t) \longleftrightarrow R(\omega) = \frac{1}{2\pi} S(\omega) * P(\omega)$$

- **Generación de las señales básicas**

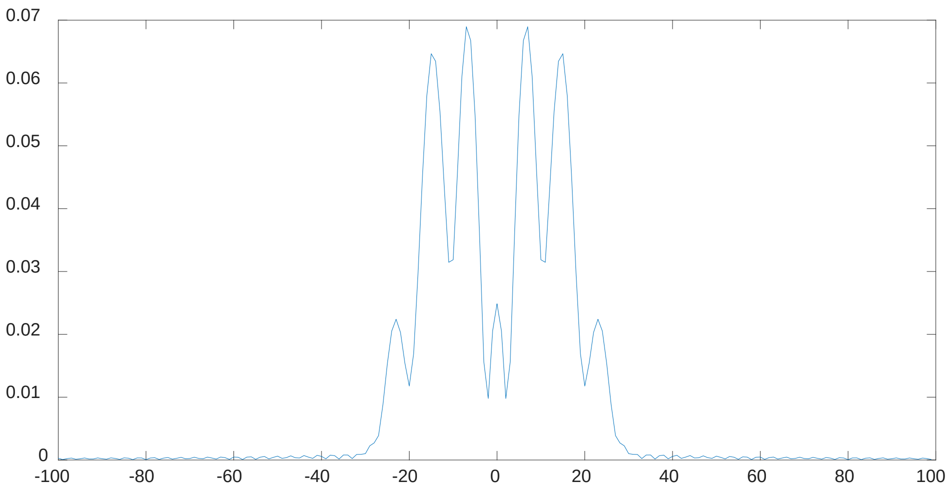
-Cargamos la señal s(t) almacenado en el disco duro (archivo *señal.mat*)

```
load senal
T=2; N=length(s);
t=linspace(0,T,N); dt = t(2) - t(1);
figure, plot (t,s)
```



-Calcular el espectro de la señal, generar el eje de frecuencias y mostrar en la pantalla el módulo del espectro en el rango de frecuencias comprendidas entre -100 y 100 rad/s (con precisión de 1 rad/s)

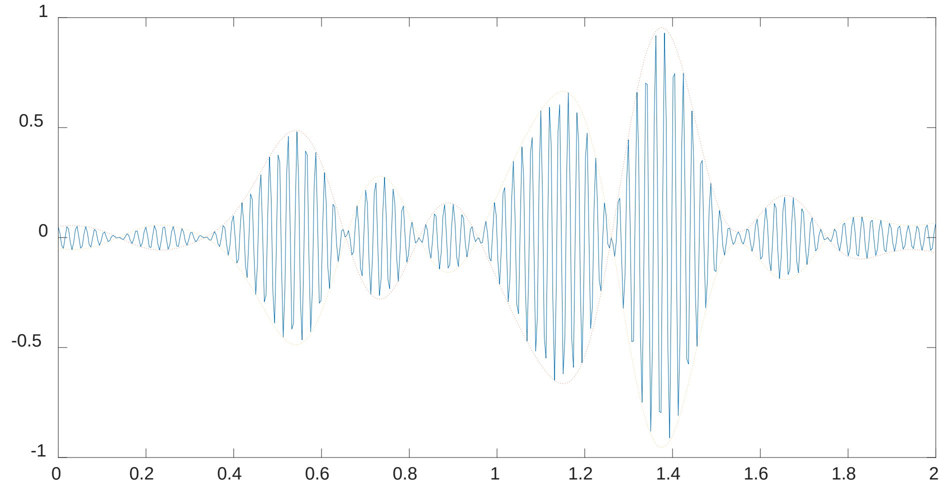
```
[S,w]=tfourier(s,t,1,100);
figure, plot(w,abs(S))
```



- **Modulación de la señal**

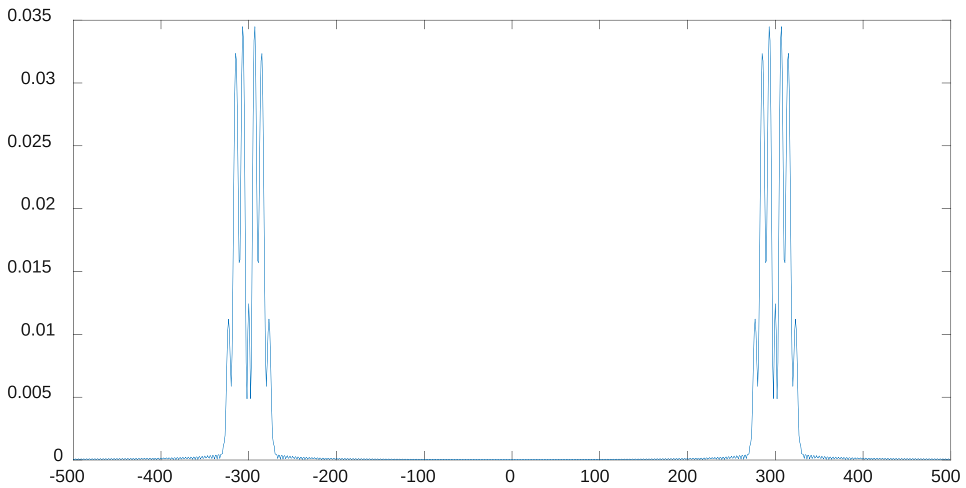
-Multiplicar la señal por una señal sinusoidal de frecuencia 300 rad/s

```
p=cos(300*t);
%portadora
x= s.*p;
%modulada
figure, plot(t,x,'-',t,s,':',t,-s,':');
```



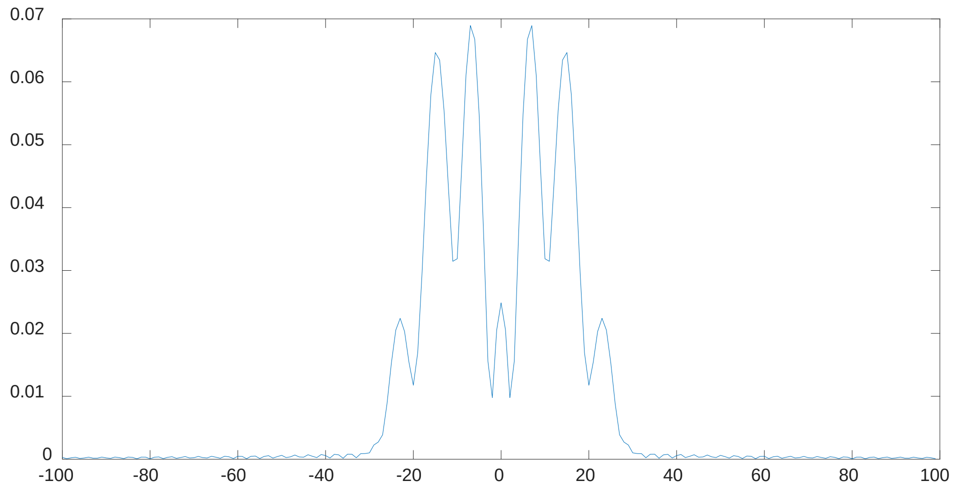
-Calcular el espectro de la señal producto (señal modulada) en un rango de frecuenciasentre -500 y 500 rad/s.

```
[X,w]=tfourier (x,t,1,500);
figure, plot (w,abs(X))
```

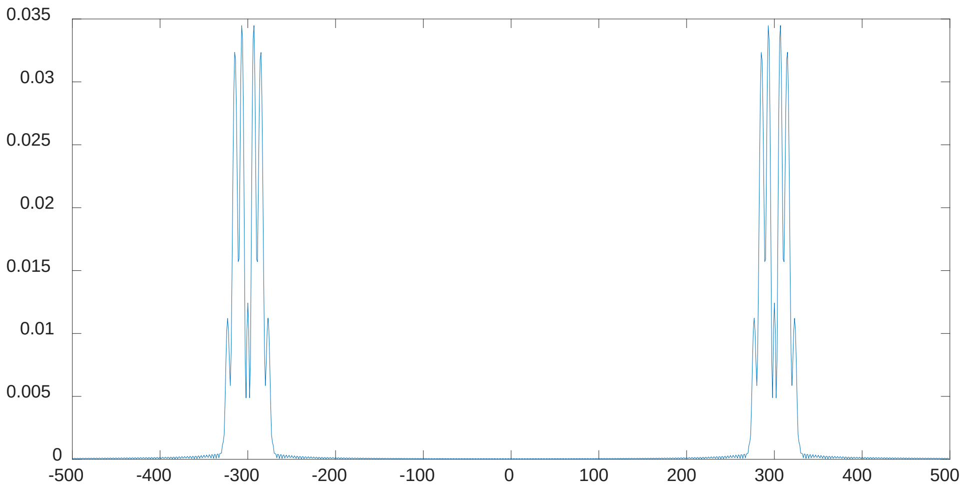


-Compare el espectro resultante con el de la señal original. Razonar el resultado.

Espectro señal original:



Espectro señal modulada:

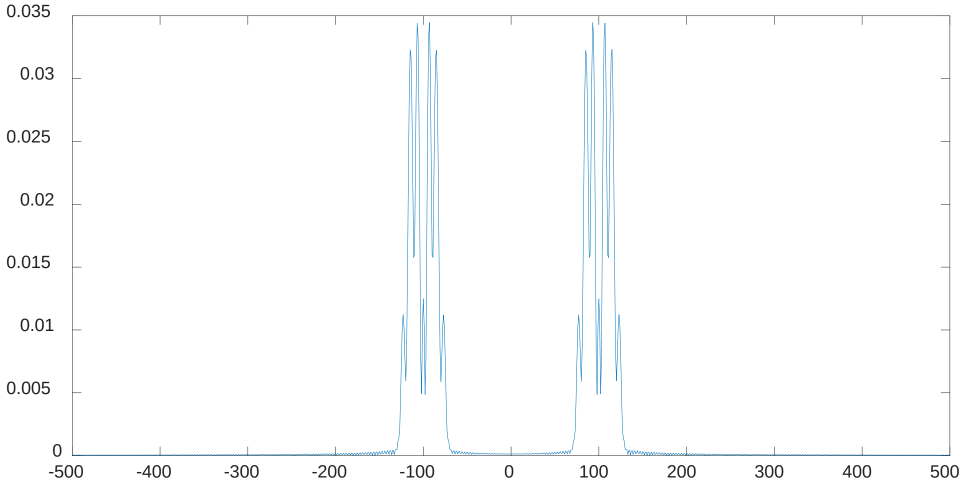


Como vimos en clase, al multiplicar la señal moduladora por una portadora en forma de coseno (que en frecuencia serán dos deltas) nos queda que (aplicando la propiedad de multiplicación): el espectro de la señal modulada queda como el espectro de la señal moduladora pero desplazado a ambos lados w_0 (w_0 =300) y de amplitud la mitad.

-Para asegurarnos de nuestras conclusiones, repita la modulación con tonos de 100 y de200 rad/s y compruebe de nuevo qué ocurre con el espectro. Razonar el resultado.

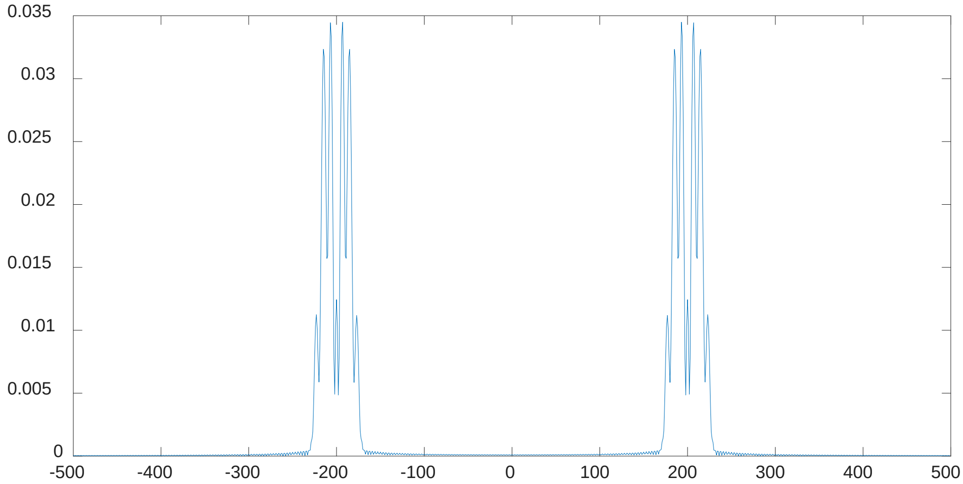
Con el tono (señal portadora) de 100 rad/s:

```
p=cos(100*t); x= s.*p;
[X,w]=tfourier (x,t,1,500);
figure, plot (w,abs(X))
```



Con el tono de 200 rad/s:

```
p=cos(200*t); x= s.*p;
[X,w]=tfourier (x,t,1,500);
figure, plot (w,abs(X))
```

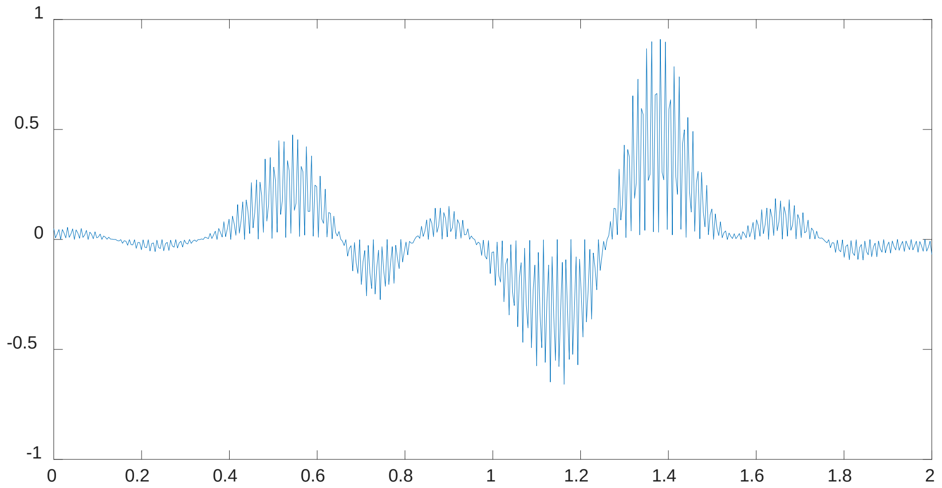


Con los tonos de 100 y 200 rad/s ocurre lo mismo que hemos comentado arriba pero ahora el desplazamiento a ambos lados cambia.

- **Demodulación de la señal**

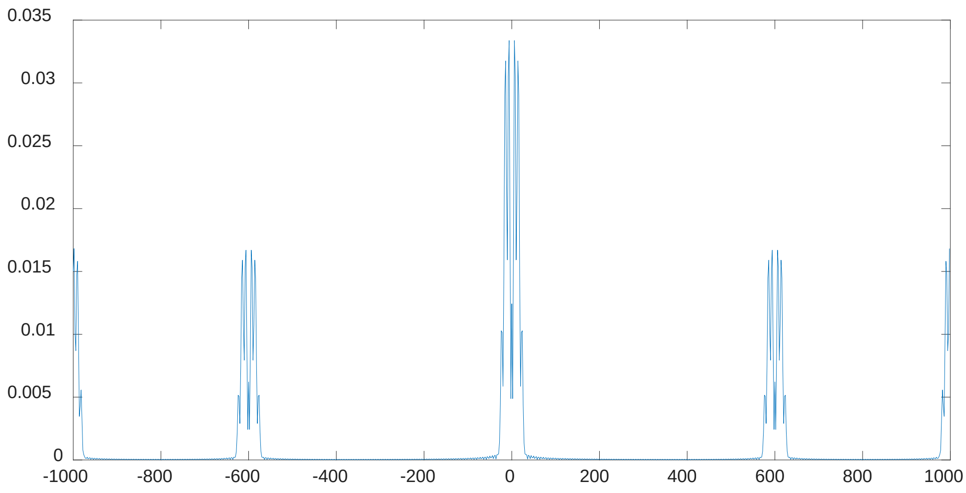
-Multiplicar la señal modulada por el mismo tono de 300 rad/s

```
p=cos(300*t);
x=s.*p;
%pongo estos dos comandos porque hemos modificado p y x anteriormente
y=x.*p;
 %demodulada
Figure, plot(t,y)
```



-Calcular el nuevo espectro en un rango de frecuencias entre -1000 y 1000 rad/s

```
[Y,w]=tfourier(y,t,2,1000);
figure,plot(w,abs(Y))
```

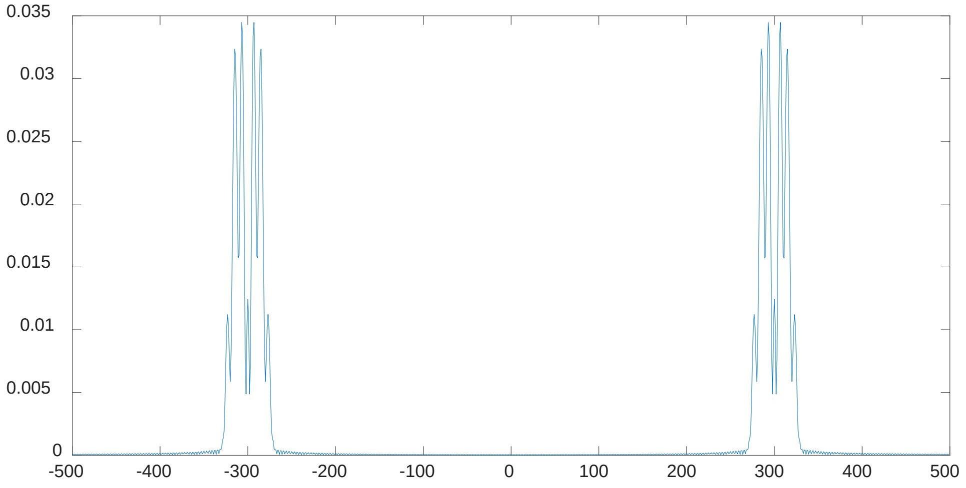


-Compare el espectro de la señal demodulada con el de la señal modulada. Razonar el resultado.

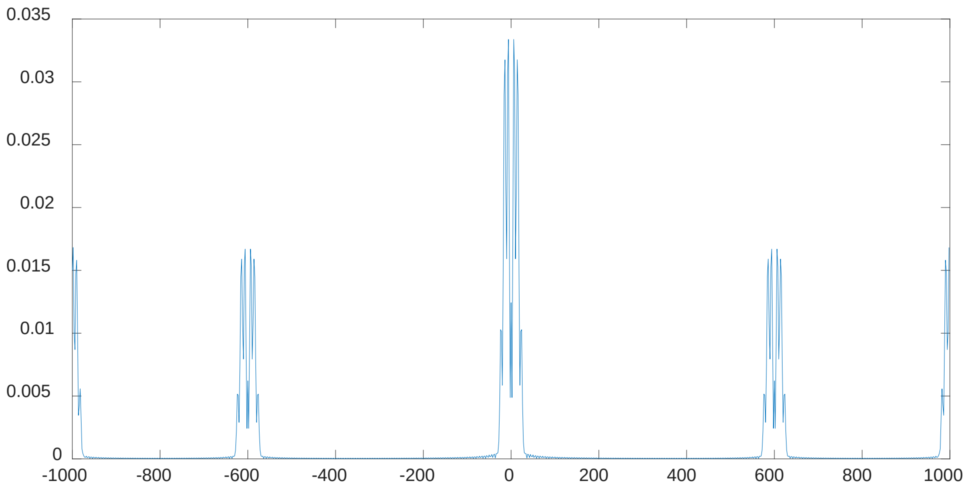
Vemos como ahora mismo la señal modulada y demodulada difieren, ya que para obtener la señal demodulada multiplicamos por la portadora (la señal modulada). La señal demodulada sufre el mismo efecto que hemos comentado antes.

Lo que nos faltaría para que la señal demodulada fuera igual a la señal moduladora

(original) es el filtro paso bajo, que seleccionaría solo las frecuencias por debajo de una frecuencia de corte establecida. La señal demodulada aparece con las réplicas.
Espectro señal modulada:



Espectro señal demodulada:



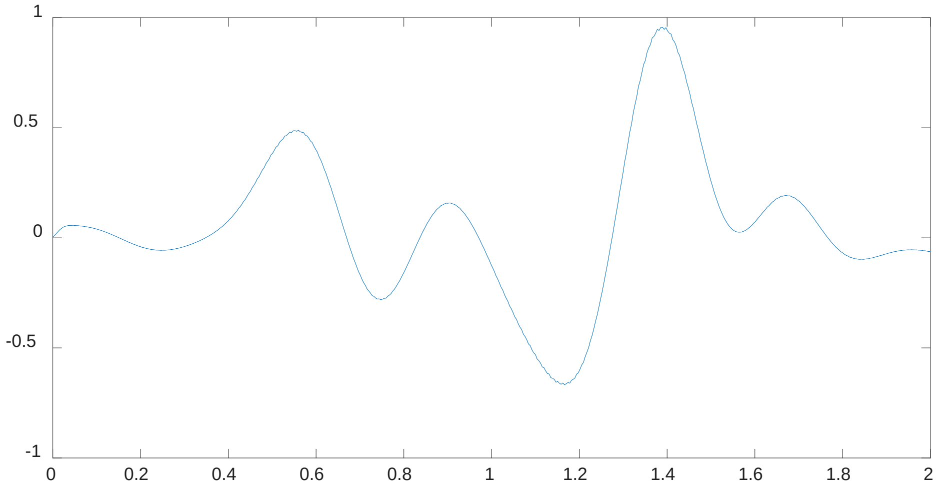
- **Recuperación de la señal original**

-La parte central del espectro corresponde al espectro de la señal de partida. Filtrando paso-bajo se consigue recuperar la señal original.

-Filtramos la señal con un filtro de Butterworth de segundo orden. Debemos elegir un valor para la frecuencia de corte wc.

Mirando la gráfica del espectro de la señal original, seleccionamos una wc = 100 rad/s.

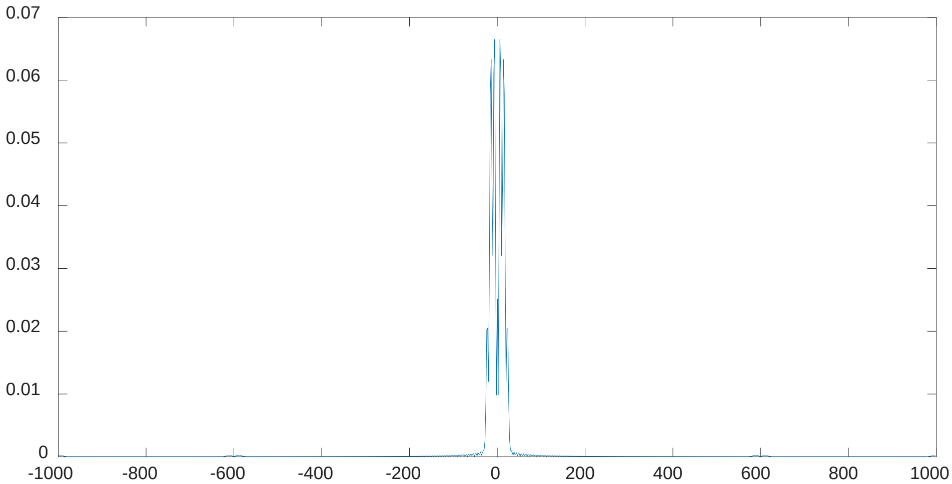
```
wc=100;
[b,a]=butter(2,wc/(pi/dt));
z=filter(2*b,a,y);
figure, plot(t,z)
```



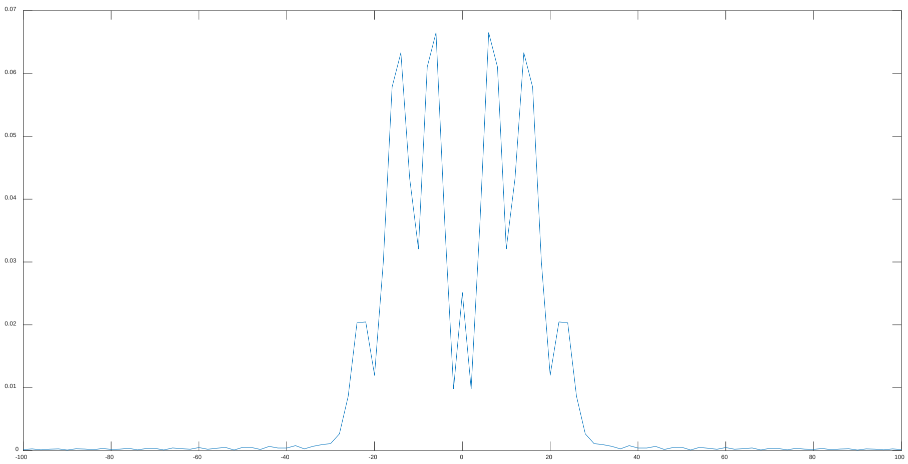
Que coincide con la señal original

-Calculamos su nuevo espectro

```
[Z,w]=tfourier(z,t,2,1000);
figure, plot(w,abs(Z));
```

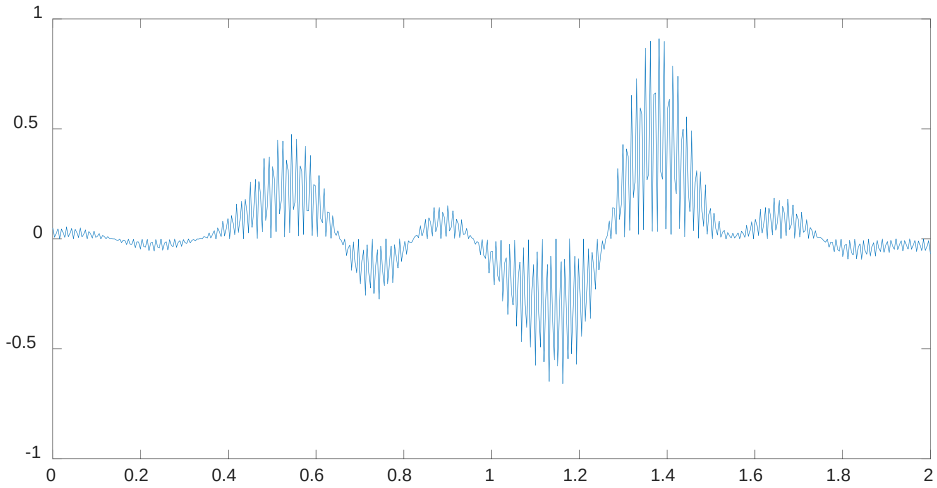


Con `figure,axis([-100 100 0 0.07])`obtenemos: (para ver que el espectro de la señal original coincide con el espectro de la señal obtenida tras el filtro)

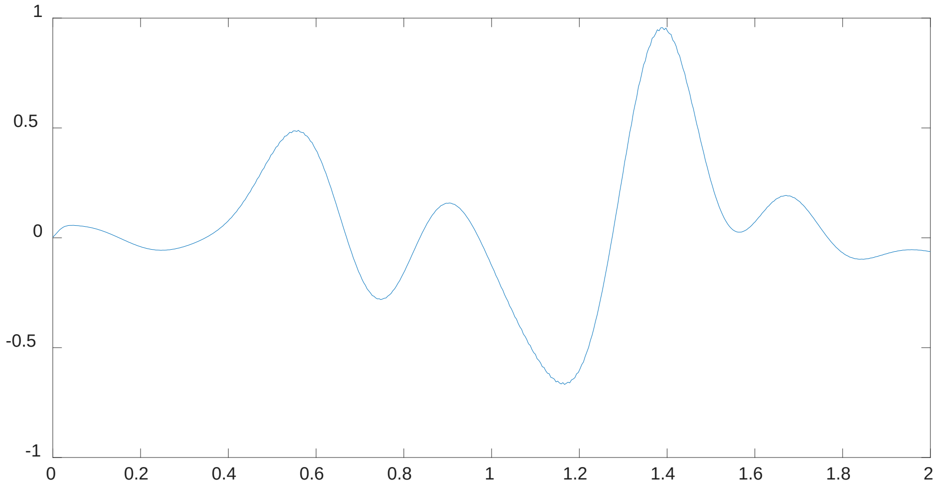


-Compare la señal demodulada con la original.

Señal demodulada:



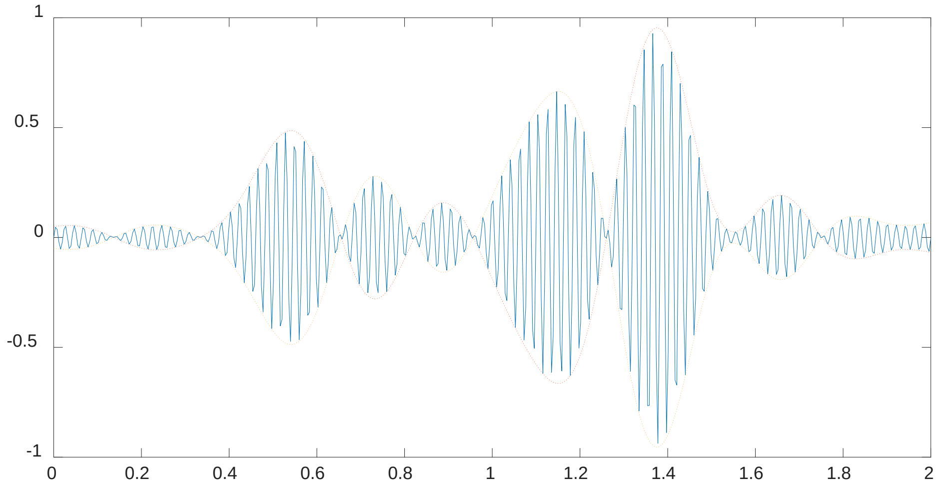
Señal original:



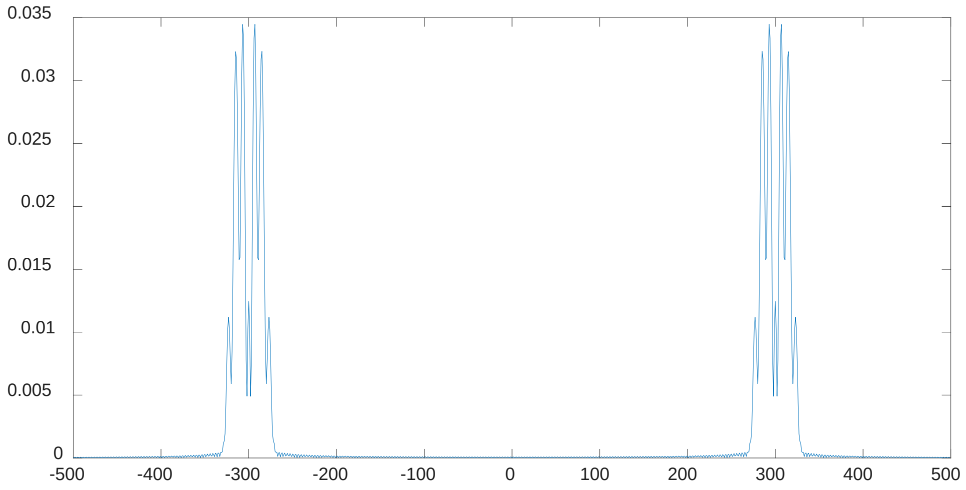
*-Repetir las etapas de demodulación y filtrado utilizando como señal demoduladora p =sin (300*t)*

Para repetir las etapas de demodulación y filtrado también debemos volver a repetir la modulación.

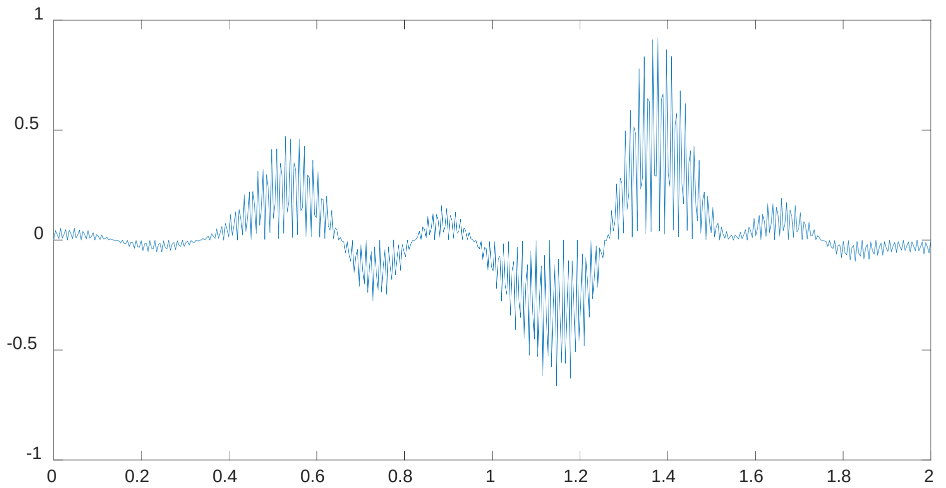
```
p=sin(300*t); x=s.*p;
figure, plot(t,x,'-',t,s,':',t,-s,':')
```



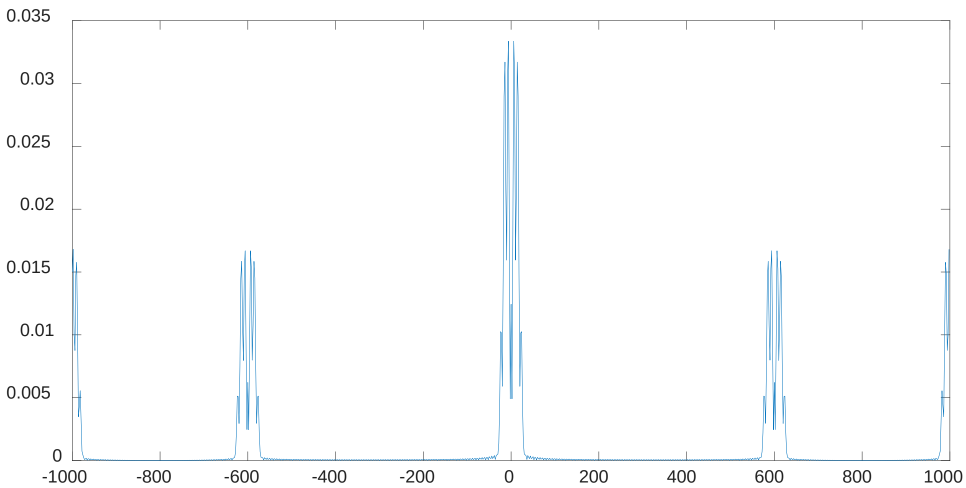

```
[X,w]=tfourier(x,t,1,500);
figure, plot(w,abs(X))
```



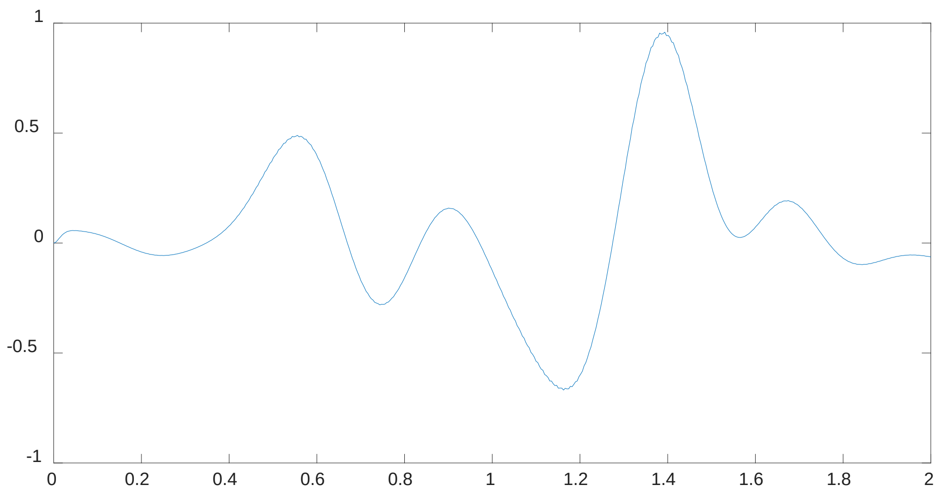

```
y=x.*p;
figure, plot(t,y)
```



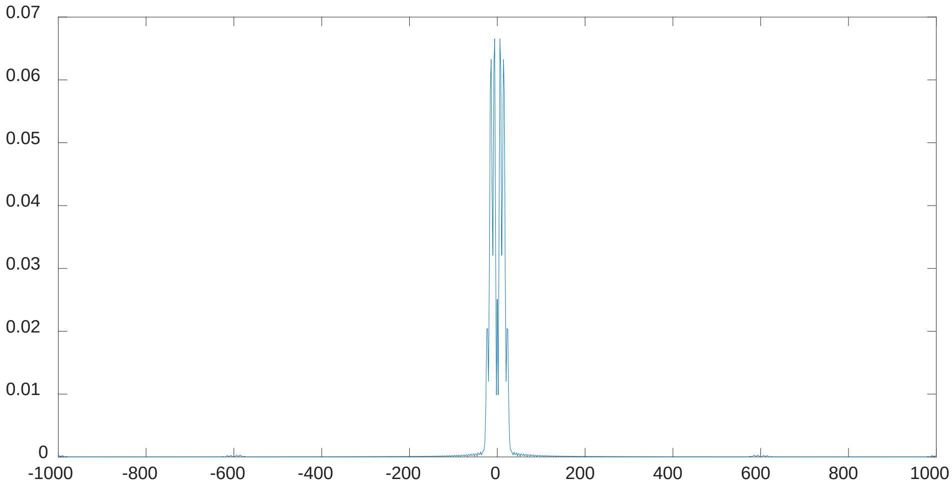

```
[Y,w]=tfourier(y,t,2,1000);
figure, plot(w,abs(Y))
```



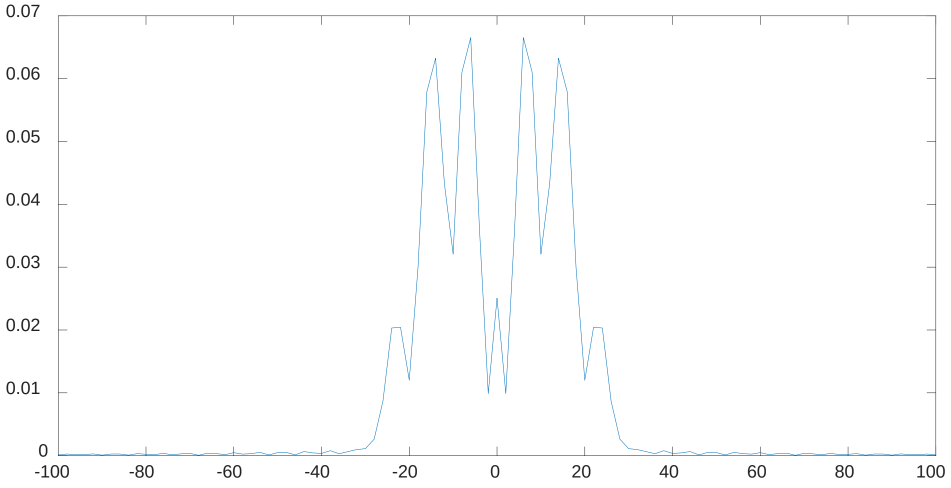

```
wc=100;
[b,a]=butter(2,wc/(pi/dt));
z=filter(2*b,a,y);
figure, plot(t,z)
```



```
[Z,w]=tfourier(z,t,2,1000);
figure, plot(w,abs(Z))
```



Con `figure,axis([-100 100 0 0.07])`obtenemos: (para ver que el espectro de la señal original coincide con el espectro de la señal obtenida tras el filtro)



2. Muestreo y Reconstrucción

-Se utilizará la señal inicial del ejercicio anterior, es decir

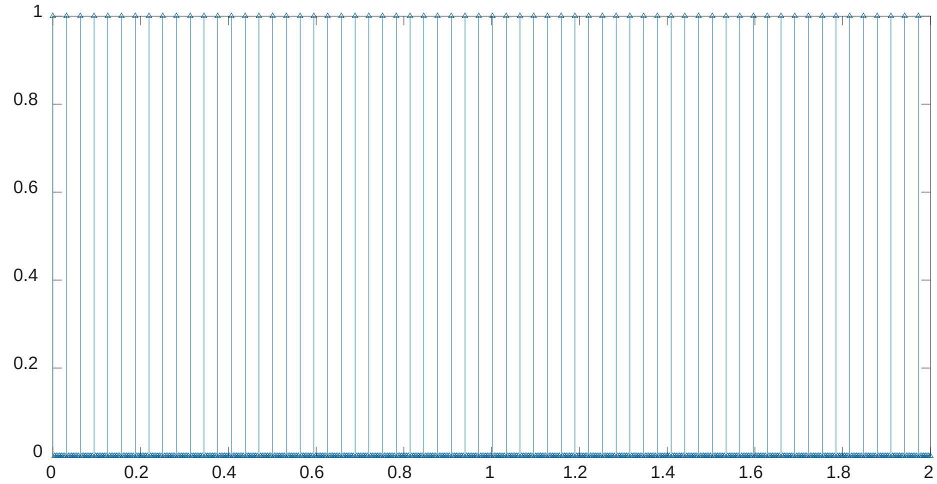
```
load senal
T=2; N=length(s);
t=linspace(0,T,N); dt=t(2)-t(1);
```

-Definir el periodo de muestreo, bien directamente o a partir de la frecuencia de muestreo, p. ej. ws=200 rad/s, Ts=2π/ws;

```
ws=200;
Ts=2*pi/ws;
```

-Generar un tren de impulsos, p(t), y mostrar el resultado por pantalla

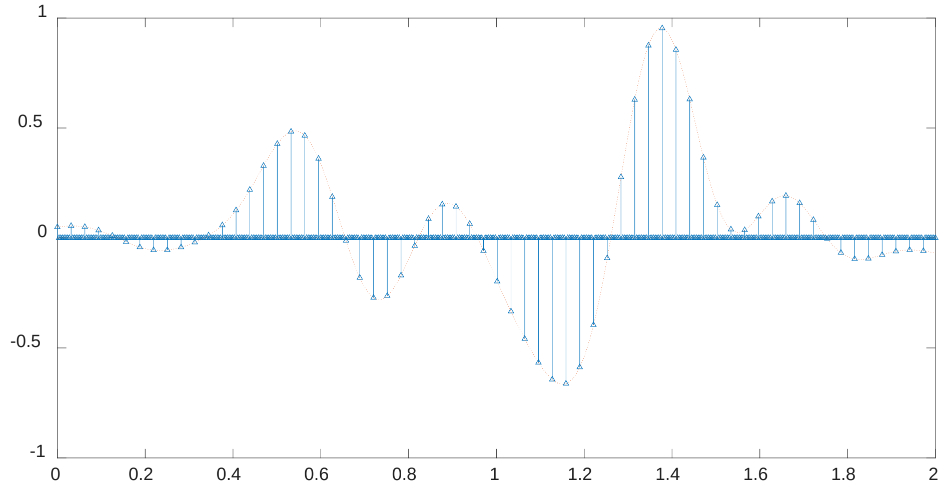
```
sc=N/T;
p=zeros(1,N);
p(1:round(Ts*sc):N)=1;
figure, stem(t,p,'^')
```



Muestreo de la señal

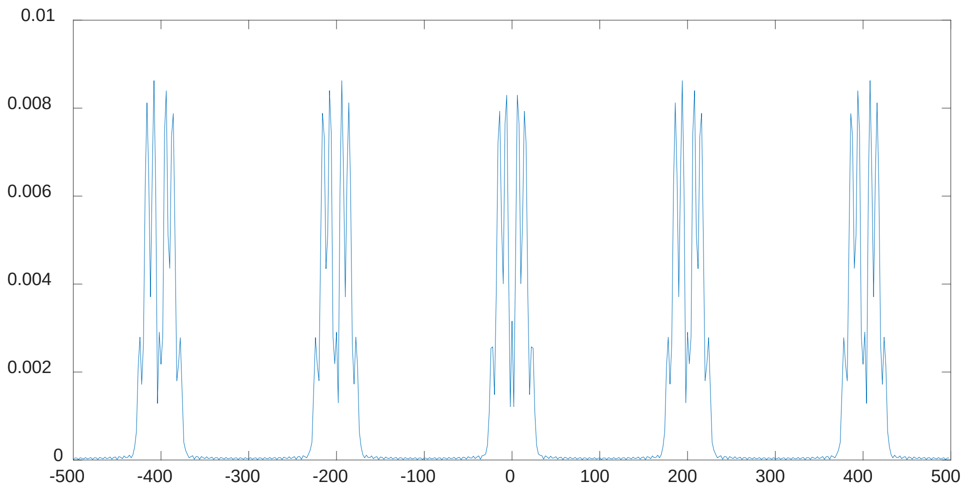
-Muestrear la señal s(t) con el tren de impulsos p(t) y mostrar el resultado en pantalla

```
y=s.*p;
figure, stem(t,y,'^')
hold on
figure, plot(t,s,':')
hold off
```



-Mostrar el espectro y(t)

```
[Y,w]=tfourier(y,t,2,500);
figure, plot(w,abs(Y))
```

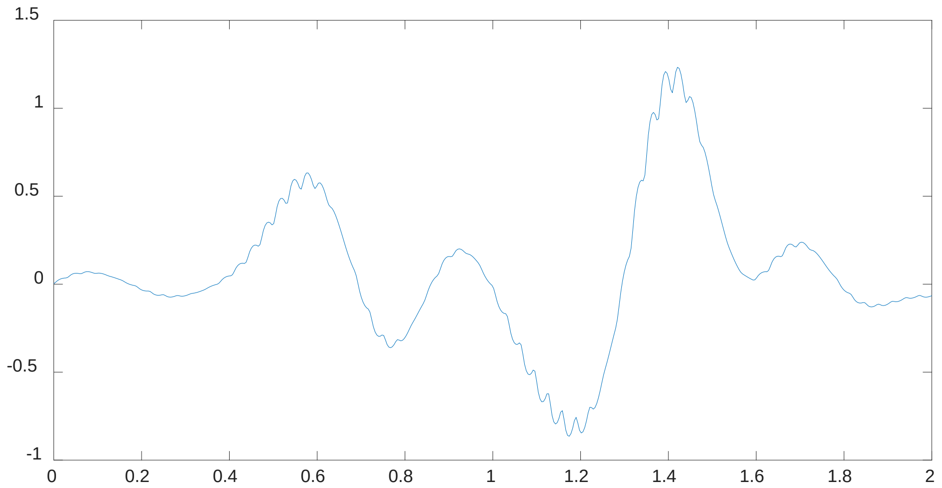


Recuperación de la señal

1. Con un filtro de Butterworth de segundo orden

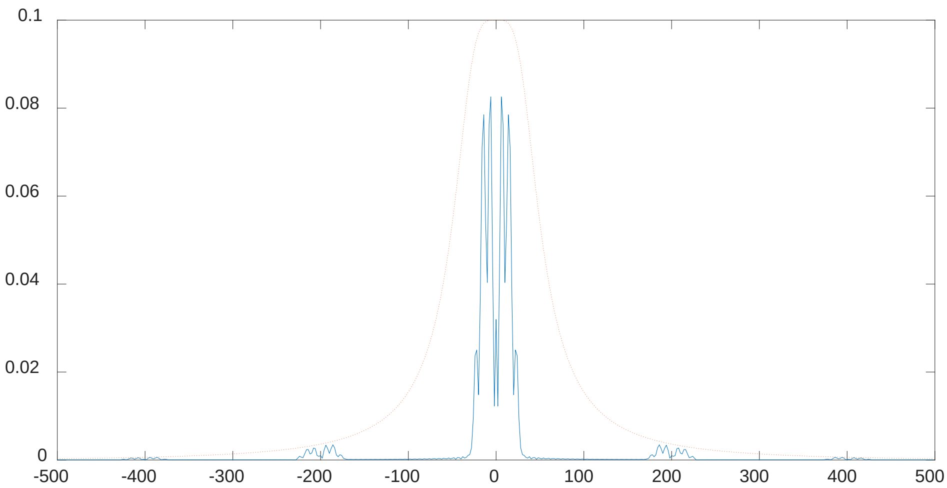
-Generar el filtro de reconstrucción $h_2(t)$ y filtrar la señal muestreada y(t) con dicho filtro

```
[b,a]=butter(2,0.2*ws/(pi/dt));
h2=filter(b,a,[1 zeros(1,N-1)]); z=filter(10*b,a,y);
figure, plot(t,z);
```



-Calcular el espectro de z(t) y mostrarlo. Razonar el resultado.

```
[Z,w]=tfourier(z,t,2,500);
[H2,w]=tfourier(h2,t,2,500);
figure, plot(w,abs(Z),'-',w,100*abs(H2),':')
```

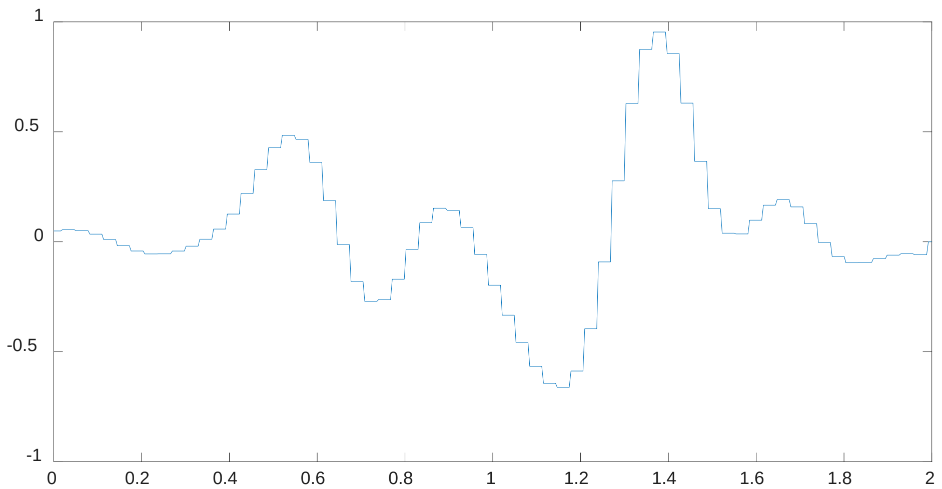


La señal recuperada difiere de la señal original, estas pequeñas variaciones se deben a los residuos (pequeñas réplicas) que aparecen en el espectro de la señal. Provocado por las “imperfecciones” que presenta el filtro (diferencias con el filtro ideal).

2. Con un filtro de orden 0 (pulso)

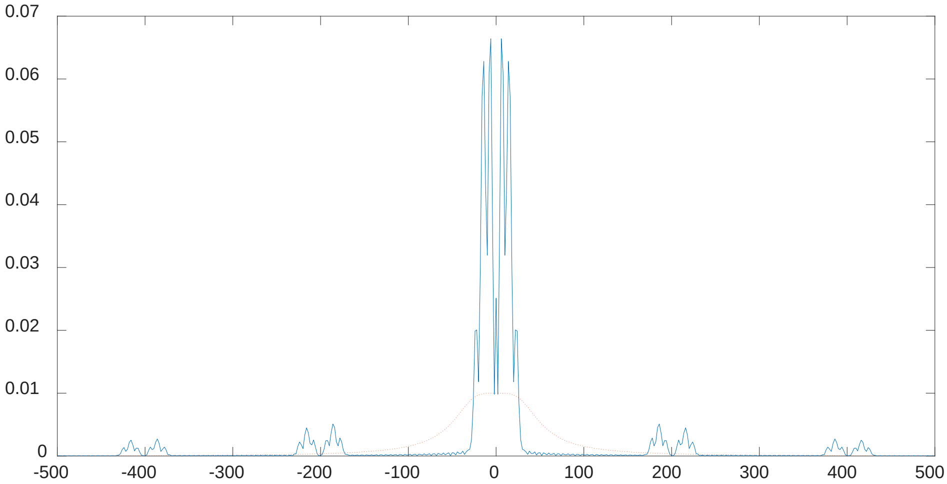
-Generar el filtro reconstructor ideal $h_0(t)$ y filtrar la señal muestreada y(t) con dicho filtro

```
L=floor(Ts*sc); h0=[ones(1,L) zeros(1,N-L)];
z=conv(y,h0); z=z(L/2:L/2+N-1);
figure, plot(t,z);
```



-Calcular el espectro de z(t) y mostrarlo. Razonar el resultado.

```
[Z,w]=tfourier(z,t,2,500);
[H0,w]=tfourier(h2,t,2,500);
figure, plot(w,abs(Z),'-',w,10*abs(H0),':')
```

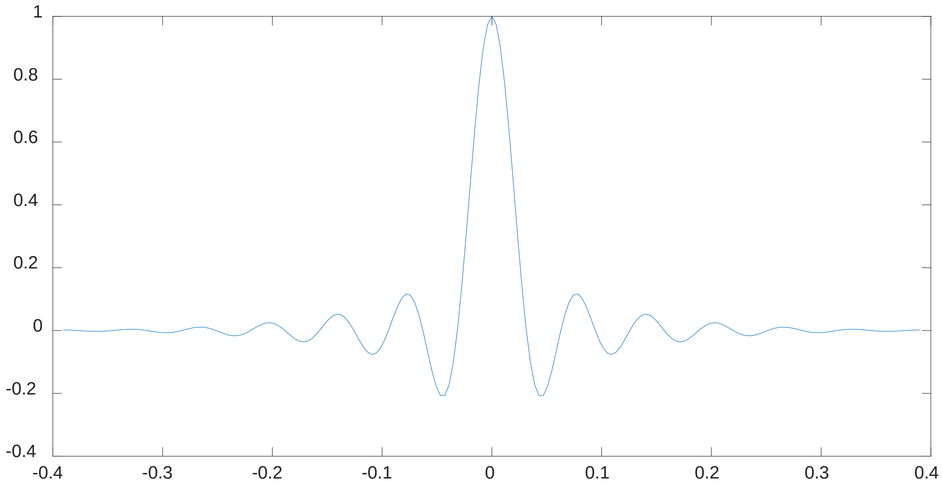


Como en el caso anterior, la señal recuperada difiere de la señal original, estas pequeñas
variaciones se deben a los residuos (pequeñas réplicas) que aparecen en el espectro de la señal.

3. *Con un filtro ideal*

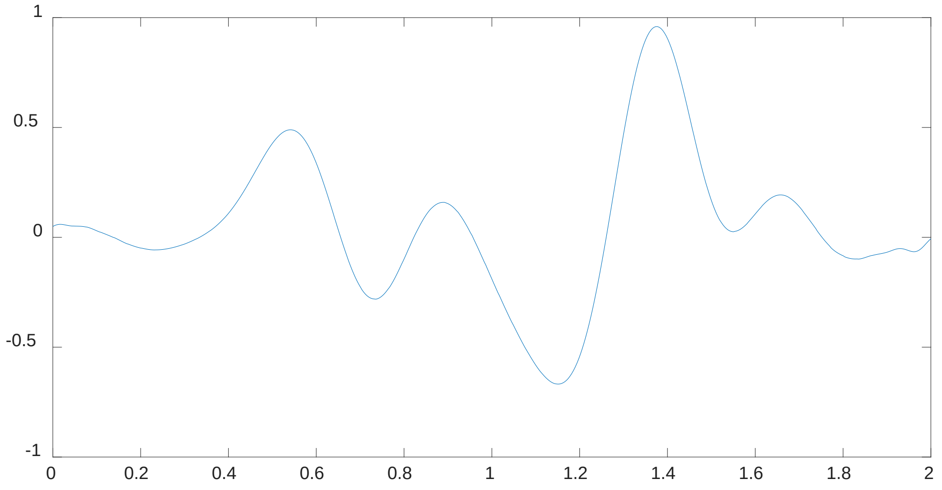
-Generar el filtro reconstructor ideal $h_i(t)$ y mostrar el resultado en pantalla

```
M=100; t1=[-M:M]/sc;
wc=ws/2;
hi=sinc(wc*t1/pi);
hi=hi.*hamming(2*M+1).';
figure; plot(t1,hi)
```



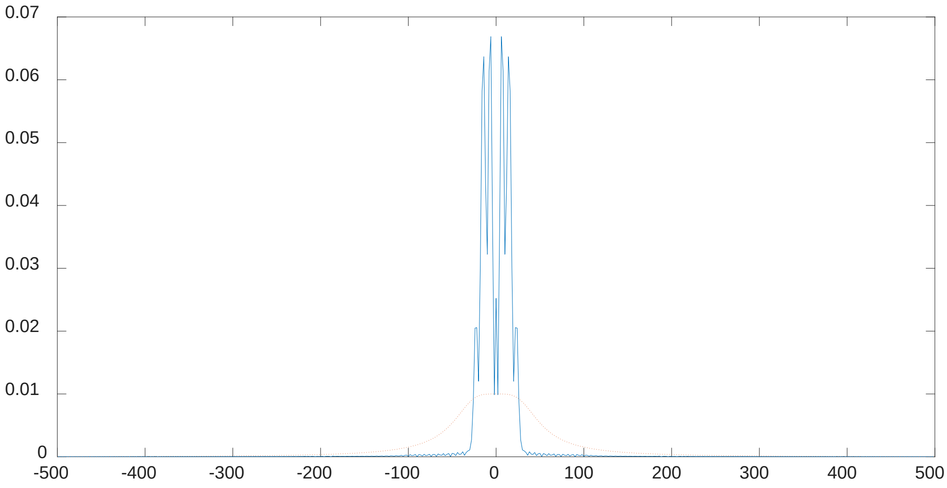
-Filtrar la señal muestreada y(t) con el filtro $h_i(t)$

```
z=conv(y,hi); z=z(M+1:N+M);
figure, plot(t,z)
```



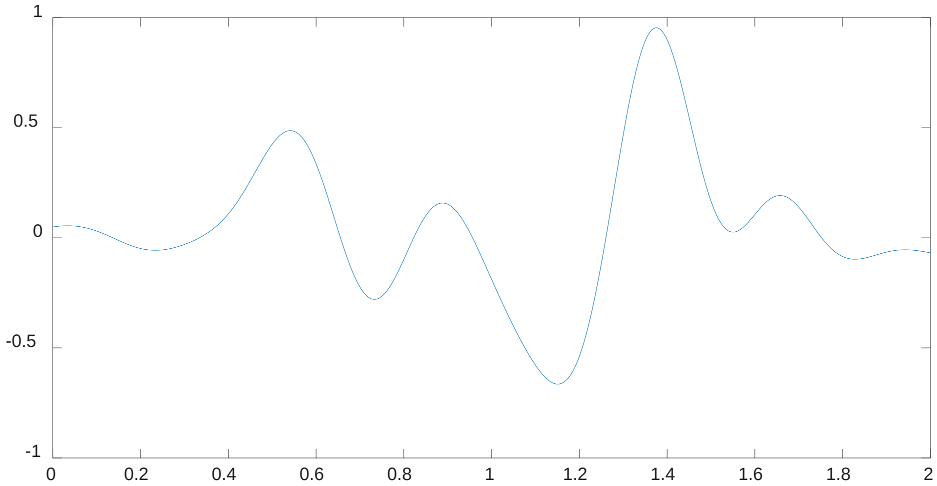
-Calcular el espectro de z(t) y mostrarlo. Razonar el resultado.

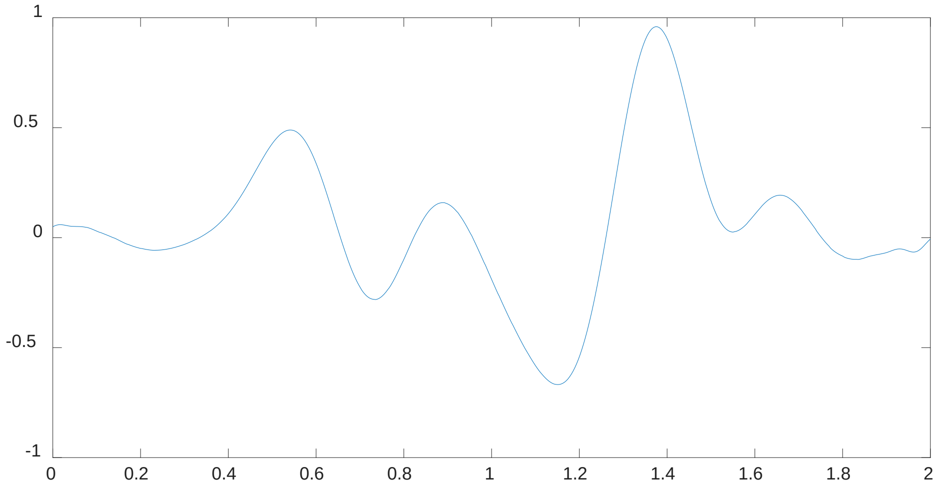
```
[Z,w]=tfourier(z,t,2,500);
[Hi,w]=tfourier(h2,t,2,500);
figure, plot(w,abs(Z),'-',w,10*abs(Hi),':')
```



Aquí podemos observar como ya no se producen réplicas en el espectro, debido a que el filtro es ideal (es capaz de descartar todas las réplicas). Permitiendo recuperar la señal original a la perfección.

-Comparar z(t) con la señal original s(t). ¿Se ha conseguido una reconstrucción perfecta?





Problemas

-Muestrear la señal a la frecuencia de Nyquist y recuperarla con el filtro ideal. Razonar lo que ocurre.

Sabemos que la frecuencia de muestreo es el doble de la frecuencia de Nyquist; por tanto, repetimos el proceso anterior para ws=100rad/s.

```
load senal
T=2; N=length(s);
t=linspace(0,T,N); dt=t(2)-t(1);
% Generación del tren de impulsos
ws=100;
Ts=2*pi/ws;
sc=N/T;
p=zeros(1,N);
p(1:round(Ts*sc):N)=1;
figure, stem(t,p,'^')
% Muestreo de la señal
```

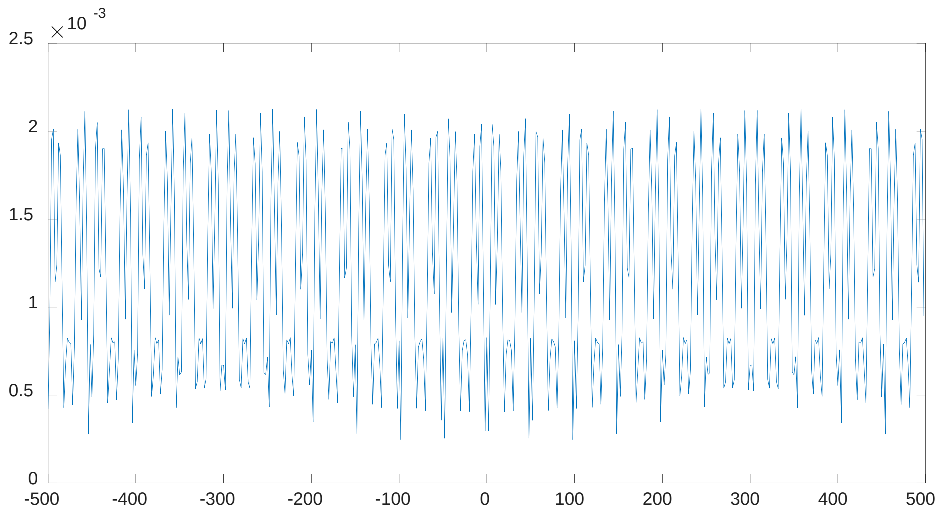
```
y=s.*p;
figure, stem(t,y,'^')
hold on
figure, plot(t,s,':')
hold off
[Y,w]=tfourier(y,t,2,500);
figure, plot(w,abs(Y))
% 3. Con un filtro ideal
M=100; t1=[-M:M]/sc;
wc=ws/2;
hi=sinc(wc*t1/pi);
hi=hi.*hamming(2*M+1).';
figure, plot(t1,hi)
z=conv(y,hi); z=z(M+1:N+M);
figure, plot(t,z)
[Z,w]=tfourier(z,t,2,500);
[Hi,w]=tfourier(h2,t,2,500);
figure, plot(w,abs(Z),'-',w,10*abs(Hi),':')
```

En este proceso hemos recuperado la señal perfectamente, debido a que la frecuencia de Nyquist es la frecuencia "límite" a partir de la cual se puede recuperar la señal original mediante muestreo.

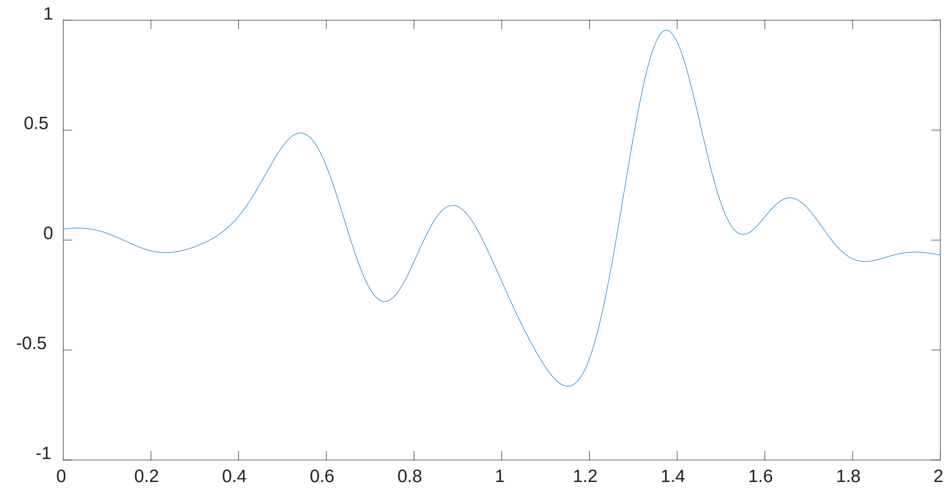
-Repetir el proceso anterior con una frecuencia menor a la de Nyquist. Razonar lo que ocurre

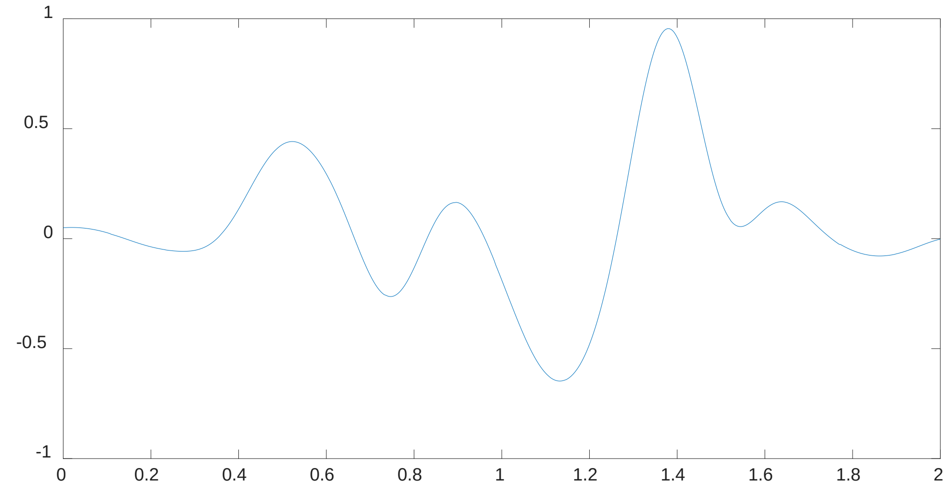
Si repetimos este proceso para una frecuencia inferior a la de Nyquist, por ejemplo, ws=50rad/s, observamos lo siguiente:

Se produce un solapamiento espectral (fenómeno conocido como aliasing), que impide que recuperemos la señal original.



Y vemos como la señal recuperada difiere de la señal original:





Cuanto más se aleje la frecuencia de muestreo de la frecuencia de Nyquist (hacia valores menores), la señal recuperada diferirá más de la original.
(Para una ws=10rad/s)