

Tarea 8

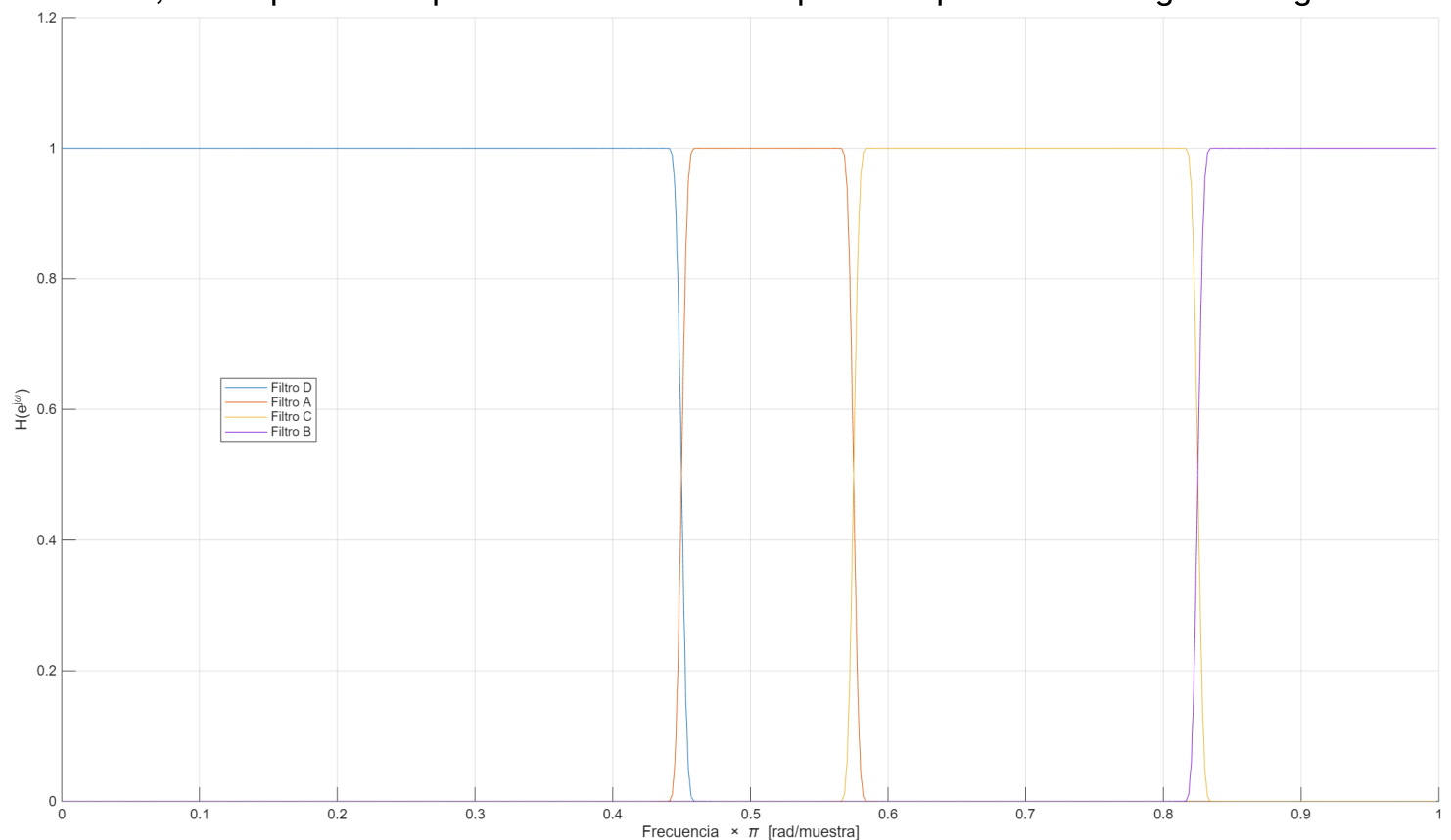
Falconieri - Kromberger

Diseño de filtros

Para diseñar los filtros, se decidió emplear la ventana de blackman sobre un filtro pasabandas ideal, cuya implementación se puede ver en el siguiente snippet:

```
function [hT] = BlackmanPBd(wc1,wc2,N)
% Muestras de la ventana
M = ceil(N/2);
n = -M:M;
%Ventana y respuesta ideal
hd = wc2/pi*sinc(wc2*n/pi)-wc1/pi*sinc(wc1*n/pi);
W = blackman(2*M+1)';
% Variable de retorno (rta impulsiva del filtro)
hT = W.*hd;
```

Debido a que la cantidad de muestras de la ventana debe mantenerse constante a lo largo de todos los filtros (en pos de mantener el retardo de la señal para todas las frecuencias), se desistió de definir la banda de paso, para en su lugar trabajar de forma directa con las frecuencias de corte. Con 601 muestras en la ventana y partiendo de las frecuencias de corte indicadas, las respuestas impulsivas de los filtros se pueden apreciar en la siguiente figura:



Inversión de tiempo y frecuencia de la señal encriptada

Para invertir en tiempo y frecuencia la señal de entrada se aplicaron los sistemas:

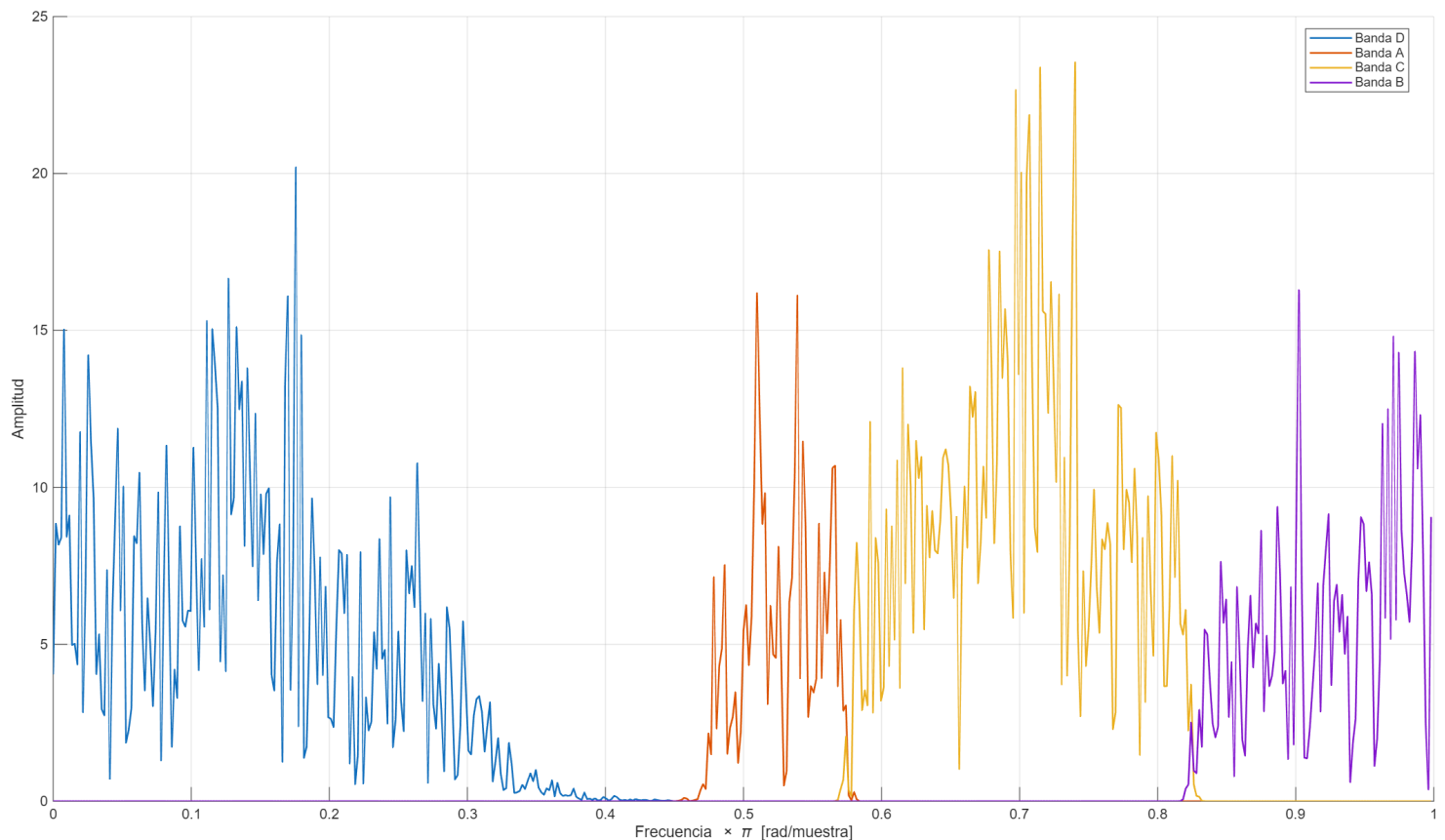
```
y[n] = x[-n] \\
y[n] = x[n](-1)^n
```

Cuya implementación en matlab termina siendo:

```
IN = open(fullfile('AudioIN',src.String{src.Value}));
IN.xe = flip(IN.xe);
IN.xe = invsors_freq(IN.xe);
function invfreq = invsors_freq(senal)
    N = length(senal);
    n = 0:N-1;

    invfreq = senal.*((-1).^n');
end
```

Luego se implementaron los filtros sobre la señal, lo cual resultó en el siguiente espectro en frecuencia:



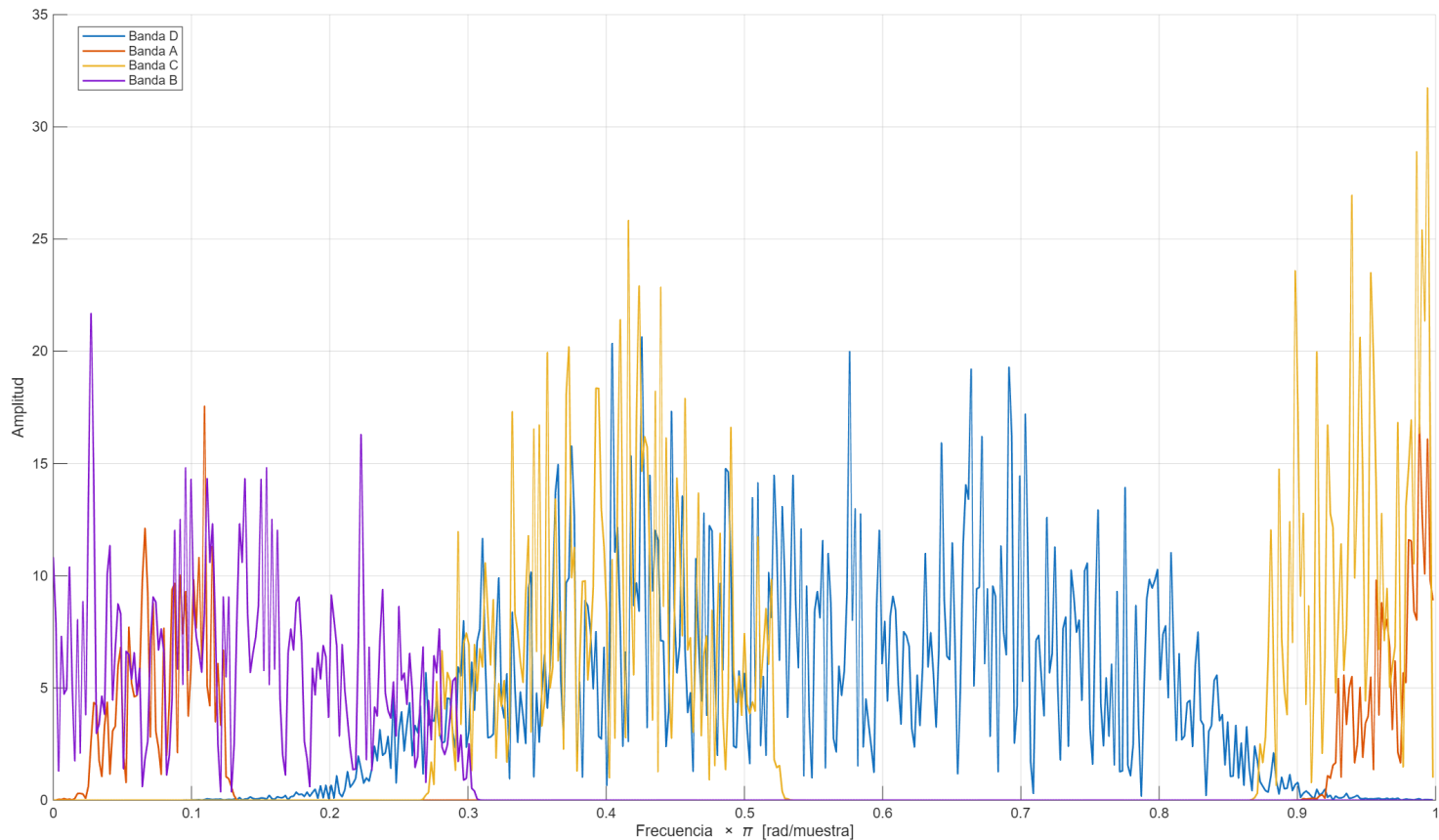
Desplazamiento de bandas

Para el siguiente paso se deben definir las frecuencias de desplazamiento de las señales, las cuales se pueden obtener mediante la ecuación:

```
\omega_0 = \omega_{destino} - \omega_{banda}
```

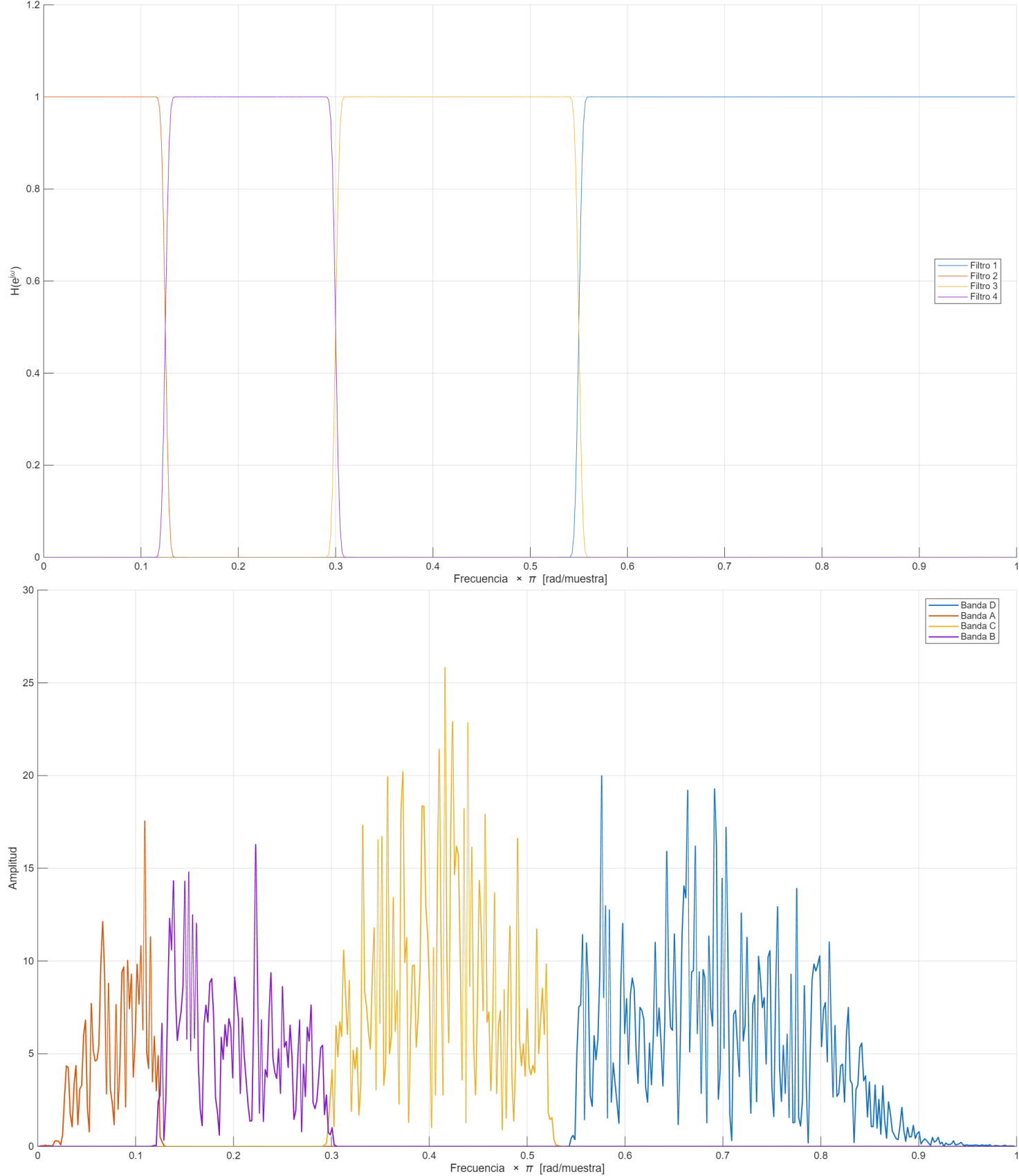
Al desplazar utilizando el coseno, el resultado genera un espectro y su reflejo, lo cual resulta en que la banda B presente su reflejo en la banda de destino. Por esto, se la desplaza hacia el origen y luego a su banda de destino, resultando en el espectro correcto, como se muestra a

continuación.



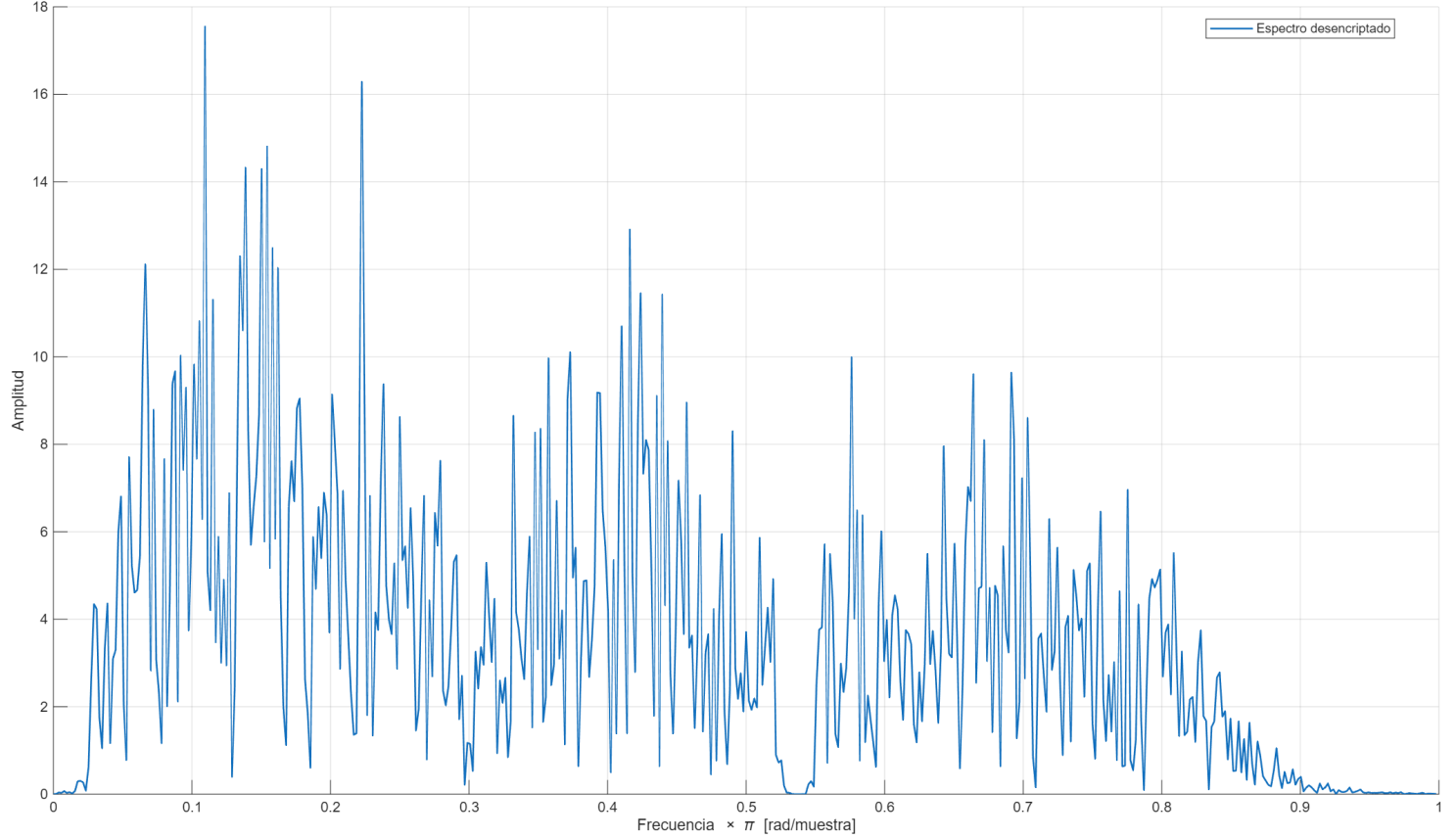
Filtrado de las señales superpuestas

Para reducir las respuestas a los espectros pertinentes, se generaron otros cuatro filtros ideales con ventana de blackman cuya respuesta en frecuencia se puede aplicar en la siguiente figura:



Resultado final

Para finalizar el descryptado se deben mitigar las bandas C y D a la mitad de su amplitud, para luego sumar las cuatro bandas y obtener la siguiente señal:Ç



Donde la implementación final del sistema para descriptar resulta ser:

```
function signalis = decrypter(Selectores, Desplazadores, Filtros, x)
    %La senal se invierte en tiempo y frecuencia al seleccionarla,
    %por eso se obvian los pasos correspondientes.
    bandas = freqsep(Selectores, x);
    for k = 1:numel(bandas)
        n = 0:numel(bandas{k})-1;
        bandas{k} = bandas{k} * 2 * cos(Desplazadores{k} * n');
        bandas{k} = filter(Filtros{k}, 1, bandas{k});
    end
    bandas{1} = bandas{1} / 2; bandas{3} = bandas{3} / 2;
    signalis = bandas{1} + bandas{2} + bandas{3} + bandas{4};
end
```