4. Diseñar 4 filtros de 45 coeficientes, cada uno con las frecuencias positivas de corte ideales en

$$\text{filtro1}= \left[0, \frac{\pi}{4}\right]$$

$$\text{filtro2}= \left[\frac{\pi}{4}, \frac{\pi}{2}\right]$$

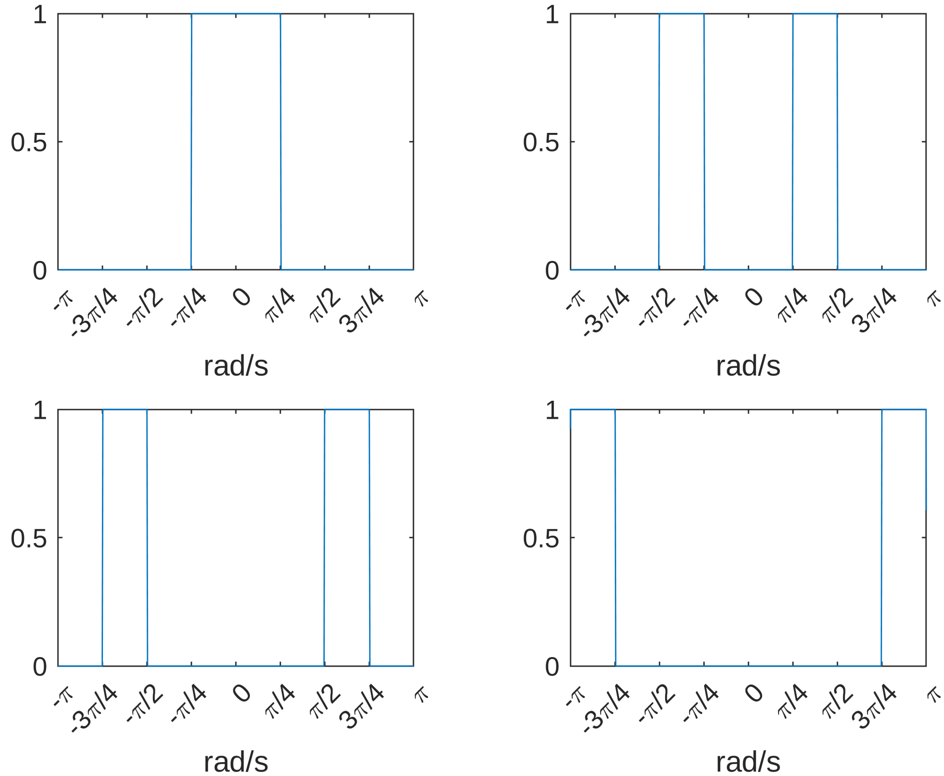
$$\text{filtro3}= \left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$$

$$\text{filtro4}= \left[\frac{3\pi}{4}, \pi\right]$$

Definimos un tiempo de observacion de 1 seg. Esto quiere decir que si muestreamos a 8000 muestras por segundo tendremos 1/8000 como tiempo de obserbacion.

Radian domain

```
bound =[-pi,pi];
space = bound(1)-1/100:1/100:bound(2)+1/100;
%generate 4 spaces
rad_space = zeros(4,length(space));
range = 0:pi/4:pi;
% 4 graphs to be ploted
figure(1)
t = tiledlayout(2,2);
side =[-1,1];
for n=2:1:5
    %index in space
    for m=side % negative, positve side
        start_filter = find(space == interp1(space,space,m*range(n-1),'nearest'));
        end_filter   = find(space == interp1(space,space,m*range(n),'nearest'));
        %4d tensor of filters, n-1 acces to layer of tensor.
        rad_space(n-1,min(start_filter,end_filter):max(start_filter,end_filter)) = 1;
    end
    %plotting stuff
    nexttile
    plot(space,rad_space(n-1,:));
    xlim([-pi pi])
    xticks([-pi -pi*3/4 -pi/2 -pi/4 0 pi/4 pi/2 pi*3/4 pi])
    xticklabels({'-\pi','-3\pi/4','-\pi/2','-\pi/4','0','\pi/4','\pi/2','3\pi/4','\pi'}
    xlabel('rad/s')
end
```



Sampling Frequency domain

Dividimos el eje por 2π. Esto para hacer la conversion de $\frac{\text{rad}}{s}$ a $\text{Hz} = \frac{1}{s}$

$$\omega = 2\pi f \ , \ f = \frac{\omega}{2\pi}$$

Posteriormente multiplicamos por la frequencia de muestreo $F_S = 8000\text{Hz}$ para obtener el **maximo espacio libre de alias .**

Det tal forma que la venta queda de

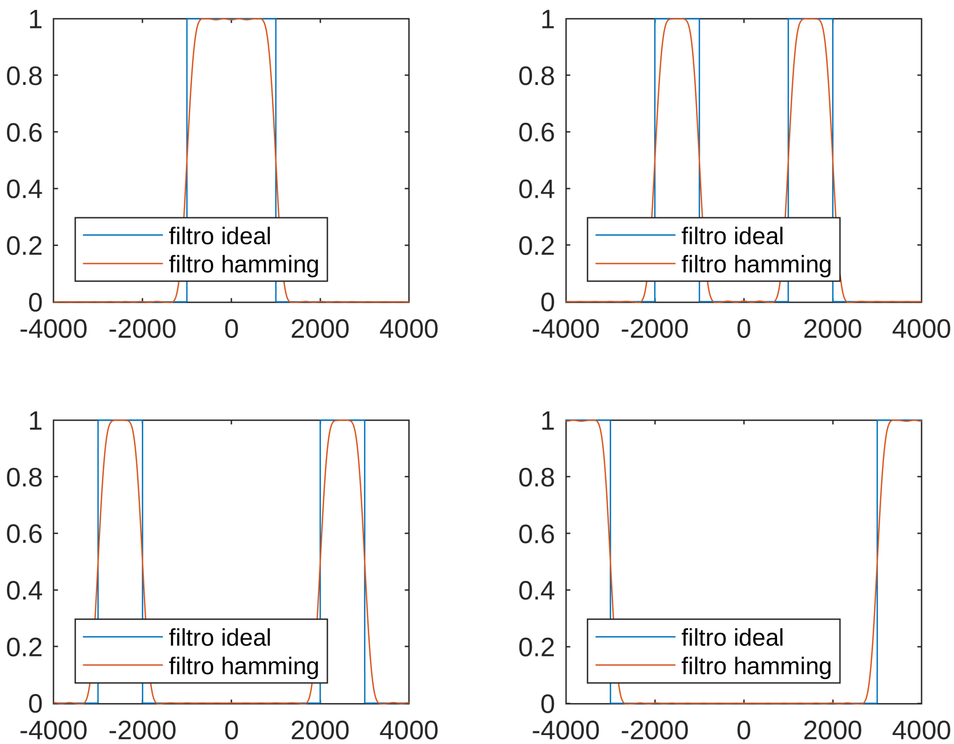
$$\left[-\frac{F_s}{2}, \frac{F_s}{2}\right]$$

```
Fs    = 8000; % sampling frequency
Coeff = 45; % num of coeffiientes

bound_fs = (bound./(2*pi)).*(Fs);
space_fs = bound_fs(1):1:bound_fs(2); % 1 second
%tensors space
filter_fs   = zeros(4,length(space_fs));
filter_time = zeros(4,length(space_fs));
windows     = zeros(4,length(space_fs));
```

```
range_fs = (range./(2*pi)).*Fs;

% 4 graphs to be ploted
figure(2)
dt_fs = tiledlayout(2,2);
side =[-1,1];
for n=2:1:5
    %index in space
    for m=side % negative, positve side
        start_filter = find(space_fs == interp1(space_fs,space_fs,m*range_fs(n-1),'near
        end_filter   = find(space_fs == interp1(space_fs,space_fs,m*range_fs(n),'neares
        %4d tensor of filters, n-1 acces to layer of tensor.
        filter_fs(n-1,min(start_filter,end_filter):max(start_filter,end_filter)) = 1;
    end
    %time signal
    filter_time(n-1,:) = real(ftot(filter_fs(n-1,:)));
    %in this case time signal is centered at 0 so 45 coeff means -22,0,22
    middle      = (length(space_fs))/2;
    left        = int32((middle-Coeff/2));
    right       = int32((middle+Coeff/2));
    windows(n-1,left:right) = hamming(Coeff+1)';
    windows(n-1,:) = windows(n-1,:).*filter_time(n-1,:);%convolution
    %normalization
    windows(n-1,:) = real(ttof(windows(n-1,:)))/max(real(ttof(windows(n-1,:))));
    %plotting stuff
    nexttile
    plot(space_fs,filter_fs(n-1,:));
    hold on;
    plot(space_fs,windows(n-1,:));
    legend({'filtro ideal','filtro hamming'},'Location','southwest')
    hold off;
end
```


1
0.8
0.6
0.4
0.2
0
-4000
-2000
0
2000
4000
filtro ideal
filtro hamming
filtro ideal
filtro hamming
filtro ideal
filtro hamming
filtro ideal
filtro hamming