

## Práctica 5: Diseño de Filtros IIR

1.1. Se desea diseñar, mediante el método de la **transformación invariante al impulso**, un **filtro paso bajo de Butterworth** que deje pasar todas aquellas señales cuya frecuencia sea inferior a  $\Omega_p = \pi/4$  rad y que atenúe las señales con frecuencias superiores a  $\Omega_a = \pi/2$  rad. Dentro de la banda de paso, la señal debe tomar un valor entre el 90 % y el 100 % del valor original, mientras que la amplitud de las señales en la banda de atenuación debe ser como mucho el 2 % de su valor original. Lleve a cabo el procedimiento de diseño con la ayuda de Matlab:

- Convierta las especificaciones del dominio discreto al continuo.
- Diseñe el filtro prototipo en tiempo continuo usando las funciones `buttord` y `butter`.
- Convierta el filtro prototipo del dominio continuo al discreto mediante el co-mando `impinvar`.
- Represente el módulo de su respuesta en frecuencia y compruebe que cumple la plantilla de especificaciones.
- Compruebe las prestaciones del filtro utilizando una señal de entrada formada por dos cosenos con frecuencias iguales a los límites de las bandas de paso y de atenuación.

```

Td = 1;
omegapasol = pi/4;
omegaatenuacion1 = pi/2;
rizpas1 = 0.1;
rizat1 = 0.02;

omegapasocont1 = omegapasol / Td;
omegaatcont1 = omegaatenuacion1 / Td;

limatpas1 = -20*log10(1-rizpas1);
limatat1 = -20*log10(rizat1);
[N, fcorte] = buttord(omegapasocont1, omegaatcont1, limatpas1, limatat1,
's');

[b, a] = butter(N, fcorte, 'low', 's');

[Bz, Az] =impinvar(b, a, 1/Td);

[h, ejec] = freqz(Bz, Az);

figure;
plot(ejec, abs(h), 'black');
axis ([0 pi 0 1]);

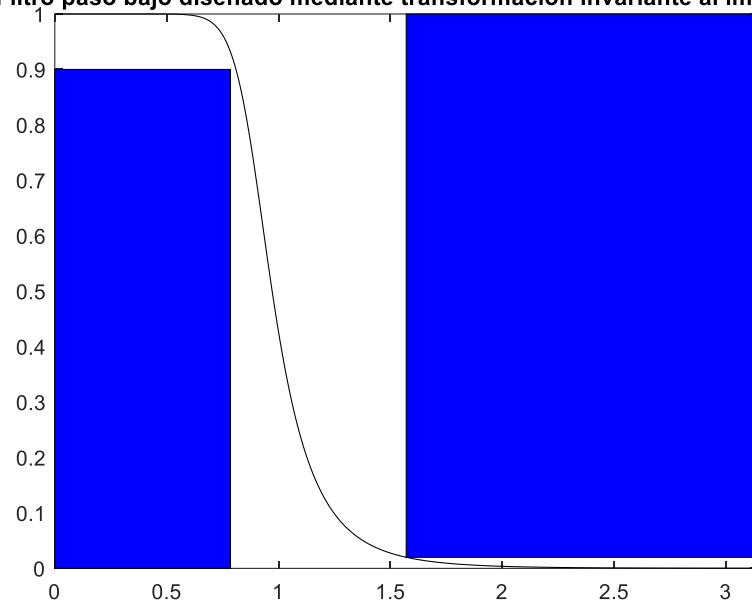
patch([0 0 omegapasol omegapasol], [0 (1-rizpas1) (1-rizpas1) 0], 'b')
patch([omegaatenuacion1 omegaatenuacion1 pi pi], [rizat1 1 1 rizat1],
'b');
title('Filtro paso bajo diseñado mediante transformación invariante al
impulso');

nc = 0:500;
xc = cos(omegaatenuacion1*nc)+cos(omegapasol*nc);
yc = filter(Bz,Az,xc);
Xc = fft(xc, 1024);
Yc = fft(yc, 1024);
ejec = linspace(0, 2*pi , 1024);

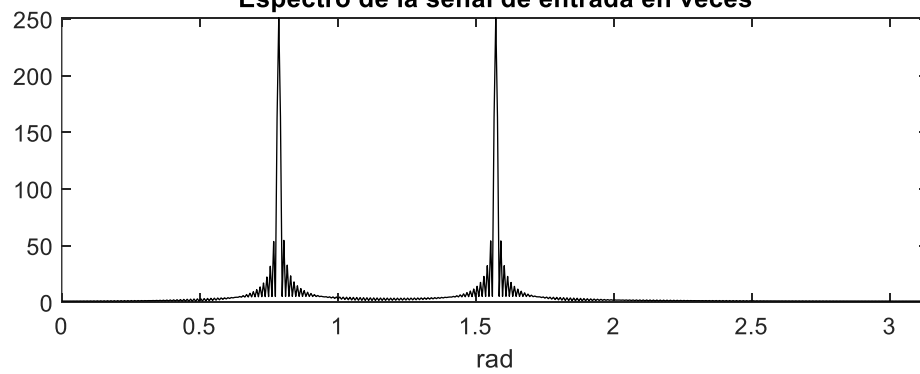
figure;
subplot(211); plot(ejec,abs(Xc), 'black');
axis([0 pi 0 max(abs(Xc))]); title('Espectro de la señal de entrada en
vecas');
xlabel('rad');
subplot(212); plot(ejec,abs(Yc), 'black');
axis([0 pi 0 max(abs(Yc))]); title('Espectro de la señal de salida en
vecas');
xlabel('rad');

```

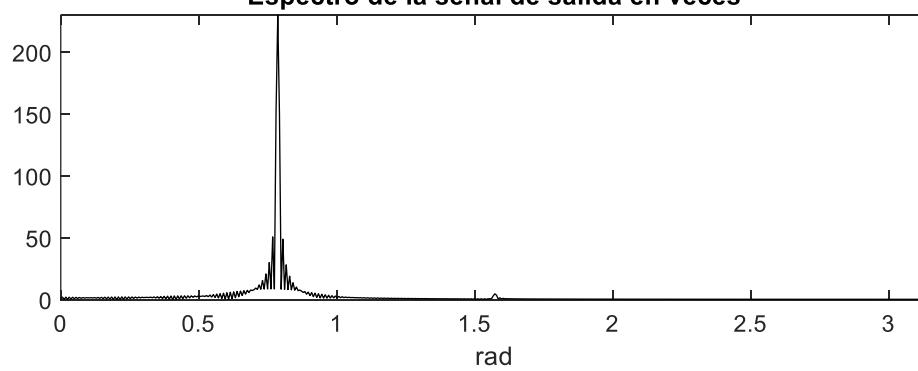
**Filtro paso bajo diseñado mediante transformación invariante al impulso**



**Espectro de la señal de entrada en veces**



**Espectro de la señal de salida en veces**



- 1.2. Repita el Ejercicio 3.1 utilizando el método de la **transformación bilineal**. Represente el módulo de la respuesta en frecuencia obtenida junto con la del apartado anterior. Comparando las características de ambos filtros, ¿cuál es mejor?

Nota: Tenga en cuenta que en este caso debe usar el comando bilinear para realizar la conversión del dominio continuo al discreto.

```
Td = 1;
omegapaso2 = pi/4;
omegaatenuacion2 = pi/2;
rizpas2 = 0.1;
rizat2 = 0.02;
nmuestras = 1024

limatpas2 = -20*log10(1-rizpas2);
limatat2 = -20*log10(rizat2);
for i = 1:2
    if( i == 1 )
        ompasocont2 = omegapaso2 / Td;
        omatcont2 = omegaatenuacion2 / Td;
    end
    if( i == 2 )
        ompasocont2 = (2/Td) *tan(omegapaso2 / 2);
        omatcont2 = (2/Td) *tan(omegaatenuacion2 / 2);
    end
    %filtro buttord orden mínimo del filtro y frecuencia de corte
    [N, fcorte] = buttord(ompasocont2, omatcont2, limatpas2, limatat2, 's');
    %comando butter hayaremos los coeficientes del filtro normalizado
    [b, a] = butter(N, fcorte, 'low', 's');

    if( i == 1 )
        [Bz, Az] =impinvar(b, a, 1/Td);
        [h, ejec2] = freqz(Bz, Az);
        figure;
        plot(ejec2, abs(h), 'r'); axis ([0 pi 0 1]);
    end
    if(i==2)
        [Bz, Az] = bilinear(b, a, 1/Td);

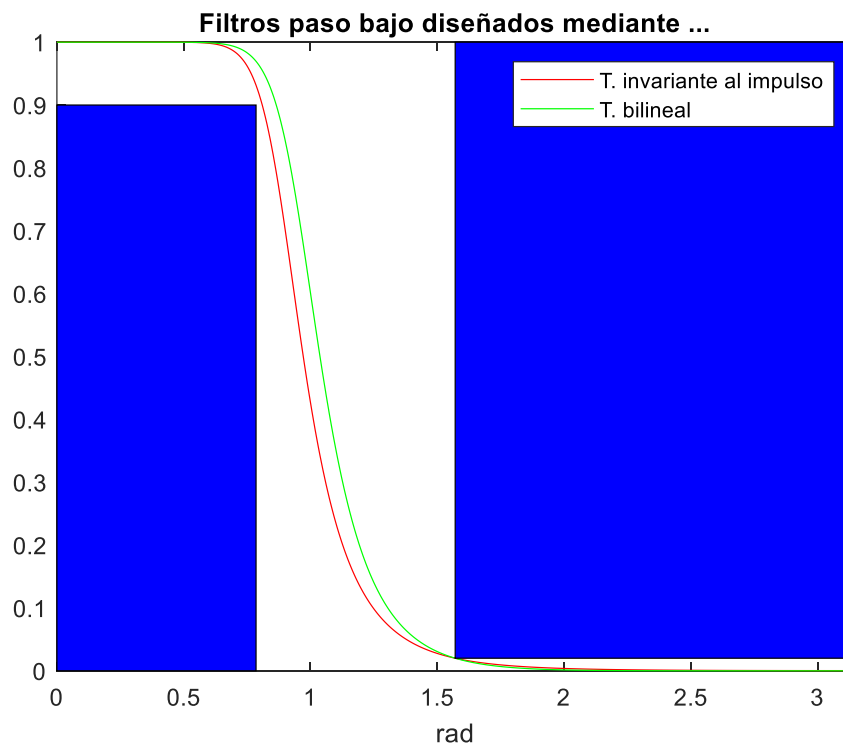
        [h, ejec2] = freqz(Bz, Az, nmuestras);
        hold on;
        plot(ejec2, abs(h), 'g');
        axis ([0 pi 0 1]);
        xlabel('rad');
```

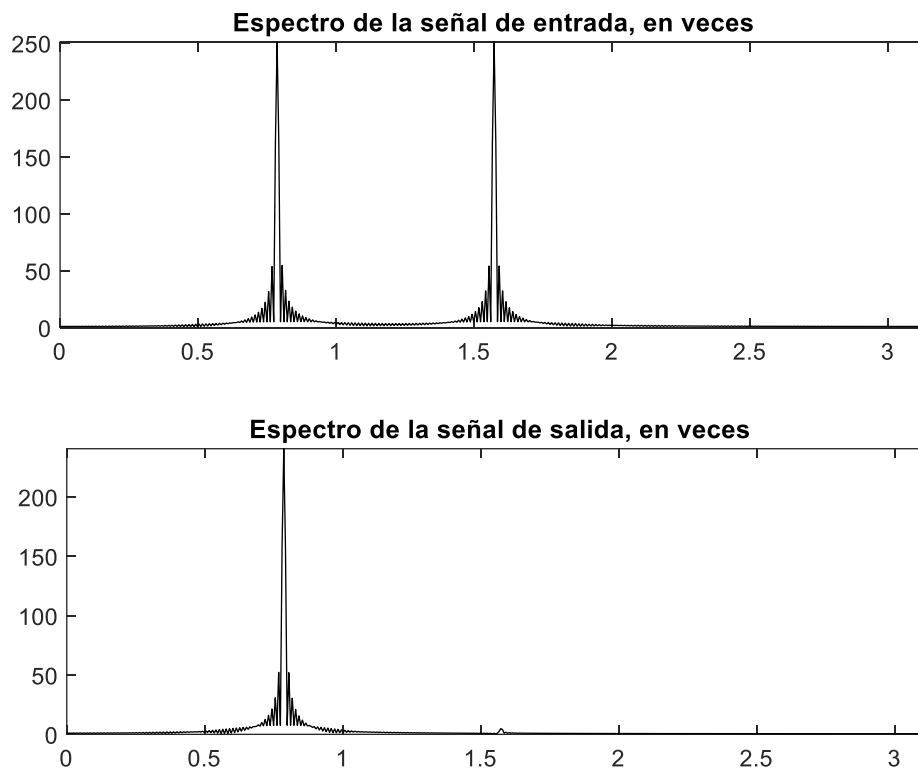
```

patch([0 0 omegapaso2 omegapaso2], [0 (1-rizpas2) (1-rizpas2) 0], 'b');
patch([omegaatenuacion2 omegaatenuacion2 pi pi], [rizat2 1 1 rizat2], 'b');
legend('T. invariante al impulso', 'T. bilineal');
hold off;
title('Filtros paso bajo diseñados mediante ...');
end
end
%COMPROBACION
nc2 = 0:500;
xc2 = cos(omegapaso2 *nc2) + cos(omegaatenuacion2 *nc2);
yc2 = filter(Bz, Az, xc2);
Xc2 = fft(xc2, nmuestras);
Yc2 = fft(yc2, nmuestras);
ejec2 = linspace(0, 2*pi, nmuestras);

figure;
subplot(211);
plot(ejec2, abs(Xc2), 'black');title('Espectro de la señal de entrada, en veces');
axis([0 pi 0 max(abs(Xc2))]);
subplot(212);
plot(ejec2, abs(Yc2), 'black');title('Espectro de la señal de salida, en veces');
axis([0 pi 0 max(abs(Yc2))]);

```





- 1.3. Repita los Ejercicios 3.1 y 3.2 utilizando los **distintos filtros prototipo** vistos: Chebyshev Tipo I y II, y Elíptico. En todos los casos, compare los filtros obtenidos mediante los métodos de la transformación invariante al impulso y de la transformación bilineal. ¿Qué conclusiones obtiene?

Nota: Tenga en cuenta que los parámetros que debe pasarle a las distintas funciones necesarias para calcular el orden del filtro (`cheb1ord`, `cheb2ord` y `ellipord`) y para encontrar sus coeficientes (`cheby1`, `cheby2` y `ellip`) pueden ser diferentes. Además, según el tipo de filtro deberá definir el rizado (esto es, el error) en la banda de paso y/o de atenuación. Vea la ayuda de Matlab de cada una de estas funciones para averiguar el modo de utilizarlas.

```

Td = 1;
omegapaso3 = pi/4;
omegaatenuacion3 = pi/2;
rizpas3 = 0.1;
rizat3 = 0.02;
muestras3 = 1024;

limatpas3 = -20*log10(1-rizpas3);
limatat3 = -20*log10(rizat3);

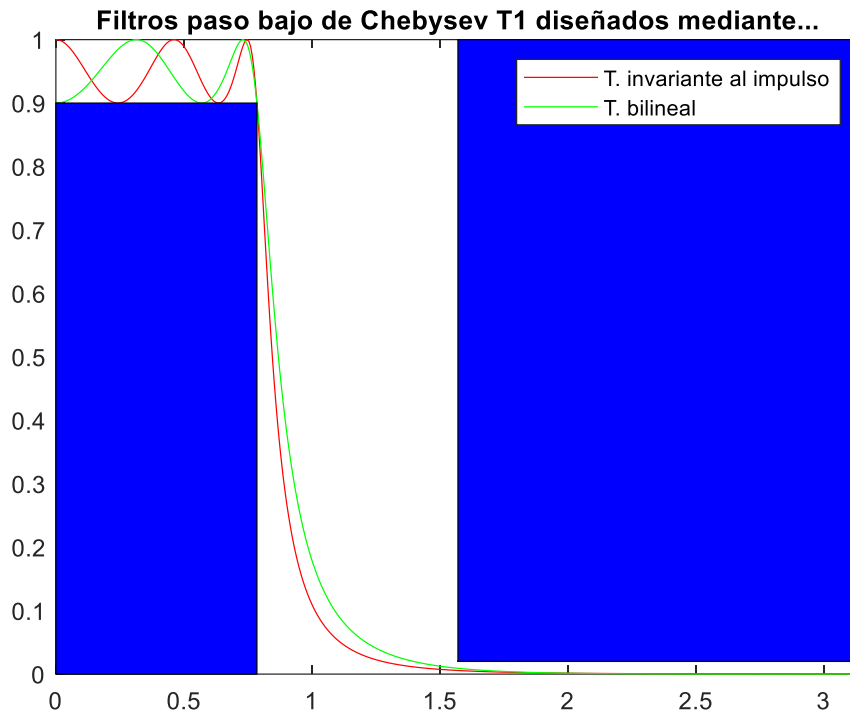
for i = 1:2
    if( i == 1 )
        ompasocont3 = omegapaso3 / Td;
        omatcont3 = omegaatenuacion3 / Td;
    end
    if( i == 2 )
        ompasocont3 = (2/Td) *tan(omegapaso3 / 2);
        omatcont3 = (2/Td) *tan(omegaatenuacion3 / 2);
    end

    [N, fcorte] = cheb1ord(ompasocont3, omatcont3, limatpas3, limatat3, 's');
    [b, a] = cheby1(N, limatpas3, ompasocont3, 'low', 's');
    if( i == 1 )
        [Bz, Az] =impinvar(b, a, 1/Td);
        [h, ejec] = freqz(Bz, Az);
        figure;
        plot(ejec, abs(h), 'r'); axis ([0 pi 0 1]);
    end
    if(i==2)
        [Bz, Az] = bilinear(b, a, 1/Td);

        [h, ejec] = freqz(Bz, Az, muestras3);
        hold on;
        plot(ejec, abs(h), 'g');
        axis ([0 pi 0 1]);

        patch([0 0 omegapaso3 omegapaso3], [0 (1-rizpas3) (1-rizpas3) 0], 'b');
        patch([omegaatenuacion3 omegaatenuacion3 pi pi], [rizat3 1 1 rizat3], 'b');
        legend('T. invariante al impulso', 'T. bilineal');
        hold off;
        title('Filtros paso bajo de Chebysev T1 diseñados mediante...');
    end
end
end

```



```
Td = 1;
omegapaso32 = pi/4;
omegaatenuacion32 = pi/2;
rizpas32 = 0.1;
rizat32 = 0.02;
muestras32 = 1024;

limatpas32 = -20*log10(1-rizpas32);
limatat32 = -20*log10(rizat32);
for i = 1:2
    if( i == 1 )
        ompasocont32 = omegapaso32 / Td;
        omatcont32 = omegaatenuacion32 / Td;
    end
    if( i == 2 )
        ompasocont32 = (2/Td) *tan(omegapaso32 / 2);
        omatcont32 = (2/Td) *tan(omegaatenuacion32 / 2) ;
    end

    [N, fcorte] = cheb2ord(ompasocont32, omatcont32, limatpas32, limatat32, 's');

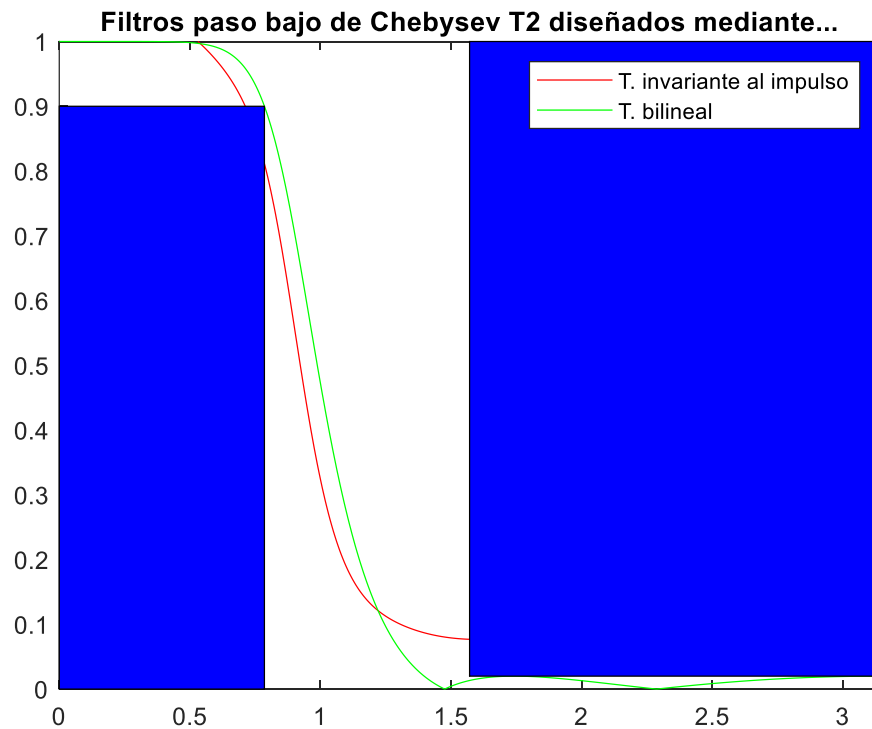
    [b, a] = cheby2(N, limatat32,fcorte,'low', 's');
    if( i == 1 )
        [Bz, Az] =impinvar(b, a, 1/Td);
        [h, ejec] = freqz(Bz, Az);
        figure;
        plot(ejec, abs(h), 'r'); axis ([0 pi 0 1]);
    end
    if(i==2)
        [Bz, Az] = bilinear(b, a, 1/Td);
        [h, ejec] = freqz(Bz, Az, muestras32);
        hold on;
        plot(ejec, abs(h), 'g');
        axis ([0 pi 0 1]);
    end
end
```



```

patch([0 0 omegapaso32 omegapaso32], [0 (1-rizpas32) (1-rizpas32) 0], 'b');
patch([omegaatenuacion32 omegaatenuacion32 pi pi], [rizat32 1 1 rizat32], 'b');
legend('T. invariante al impulso', 'T. bilineal');
hold off;
title('Filtros paso bajo de Chebysev T2 diseñados mediante...');
end
end

```



- 1.4. Se desea diseñar, utilizando la aproximación de Butterworth, un **filtro paso alto** que deje pasar todas aquellas señales cuya frecuencia sea superior a  $\Omega_p = \pi/2$  rad y que atenúe las señales con frecuencias inferiores a  $\Omega_a = \pi/4$  rad. Dentro de la banda de paso, la señal debe tomar un valor entre el 90 % y el 100 % del valor original, mientras que la amplitud de las señales en la banda de atenuación debe ser como mucho el 2 % de su valor original.

Nota: Para diseñar un filtro paso alto debe seleccionar la opción 'high' a la hora de llamar a la función butter.

```

Td = 1;
omegapaso3e = pi/4;
omegaatenuacio3e = pi/2;
rizpas3e = 0.1;
rizat3e = 0.02;
muestras3e = 1024;

limatpas3e = -20*log10(1-rizpas3e);
limatat3e = -20*log10(rizat3e);
for i = 1:2
    if( i == 1 )
        ompasocont3e = omegapaso3e / Td;
        omatcont3e = omegaatenuacio3e / Td;
    end
    if( i == 2 )
        ompasocont3e = (2/Td) *tan(omegapaso3e / 2);
        omatcont3e = (2/Td) *tan(omegaatenuacio3e / 2);
    end

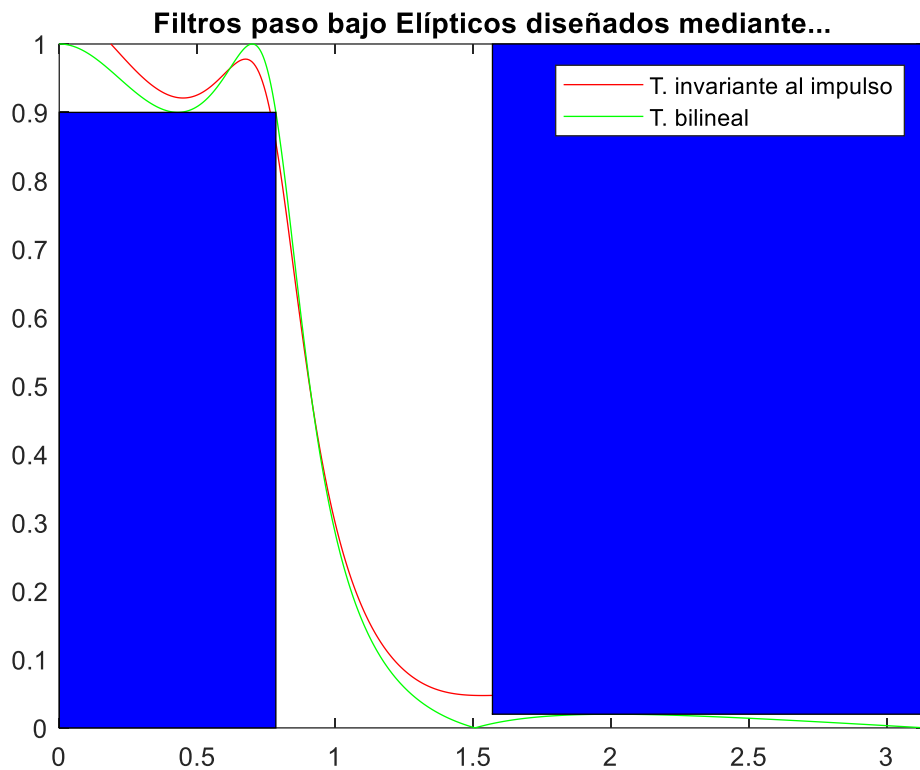
[N, fcorte] = ellipord(ompasocont3e, omatcont3e, limatpas3e, limatat3e, 's');

[b, a] = ellip(N,limatpas3e,limatat3e,ompasocont3e,'low', 's');
if( i == 1 )
    [Bz, Az] =impinvar(b, a, 1/Td);
    [h, ejec] = freqz(Bz, Az);
    figure;
    plot(ejec, abs(h), 'r'); axis ([0 pi 0 1]);
end
if(i==2)
    [Bz, Az] = bilinear(b, a, 1/Td);

[h, ejec] = freqz(Bz, Az, muestras3e);
hold on;
plot(ejec, abs(h), 'g');
axis ([0 pi 0 1]);

patch([0 0 omegapaso3e omegapaso3e], [0 (1-rizpas3e) (1-rizpas3e) 0], 'b');
patch([omegaatenuacio3e omegaatenuacio3e pi pi], [rizat3e 1 1 rizat3e], 'b');
legend('T. invariante al impulso', 'T. bilineal');
hold off;
title('Filtros paso bajo Elípticos diseñados mediante...');
end
end

```



1.4 Se desea diseñar, utilizando la aproximación de Butterworth, un **filtro paso alto** que deje pasar todas aquellas señales cuya frecuencia sea superior a  $\Omega_p = \pi/2$  rad y que atenúe las señales con frecuencias inferiores a  $\Omega_a = \pi/4$  rad. Dentro de la banda de paso, la señal debe tomar un valor entre el 90 % y el 100 % del valor original, mientras que la amplitud de las señales en la banda de atenuación debe ser como mucho el 2 % de su valor original.

Nota: Para diseñar un filtro paso alto debe seleccionar la opción 'high' a la hora de llamar a la función butter.

```
Td = 1;
omegaatenuacion4 = pi/4;
omegapaso4 = pi/2;
rizpas4 = 0.1;
rizat4 = 0.02;
muestras4 = 1024;

ompasocont4 = (2/Td) *tan(omegapaso4 / 2);
omatcont4 = (2/Td) *tan(omegaatenuacion4 / 2);

limatpas4 = -20*log10(1-rizpas4);
limatat4 = -20*log10(rizat4);
```

```

[N, fcorte] = buttord(ompasocont4, omatcont4, limatpas4, limatat4,
's');

[b, a] = butter(N, fcorte, 'high', 's');

[Bz, Az] = bilinear(b, a, 1/Td);

[h, ejec4] = freqz(Bz, Az, muestras4);

figure;
plot(ejec4, abs(h), 'black');

patch([0 0 omegaatenuacion4 omegaatenuacion4], [rizat4 1 1 rizat4],
'b')
patch([omegapaso4 omegapaso4 pi pi], [0 (1-rizpas4) (1-rizpas4) 0], 'b');
axis ([0 pi 0 1]);
title('Filtro paso alto diseñado mediante transformación bilineal');
xlabel('rad');

nc4 = 0:500;
xc4 = cos(omegaatenuacion4*nc4)+cos(omegapaso4*nc4);

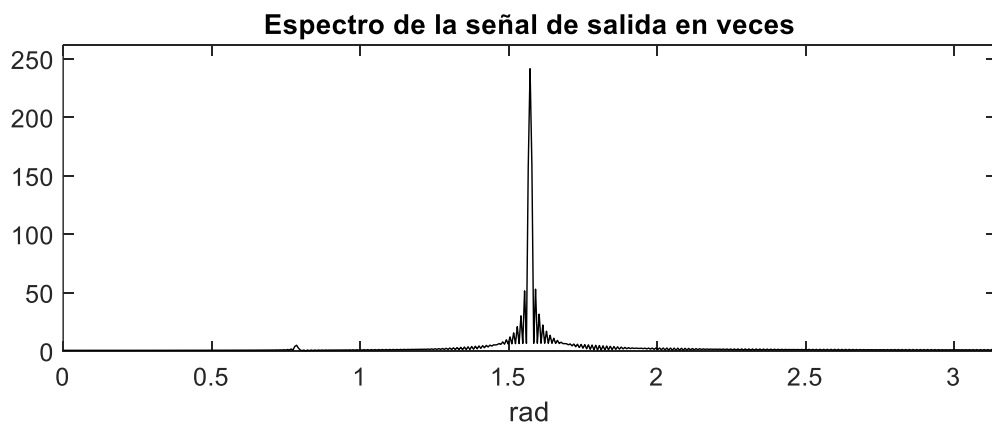
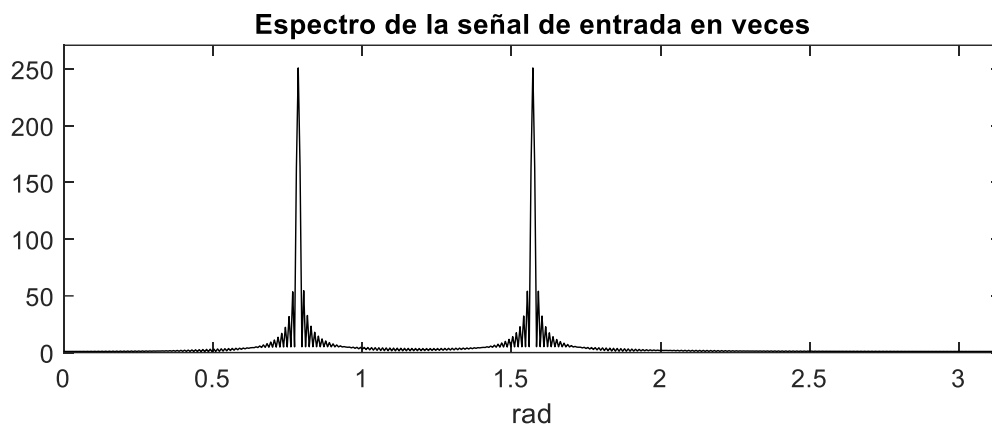
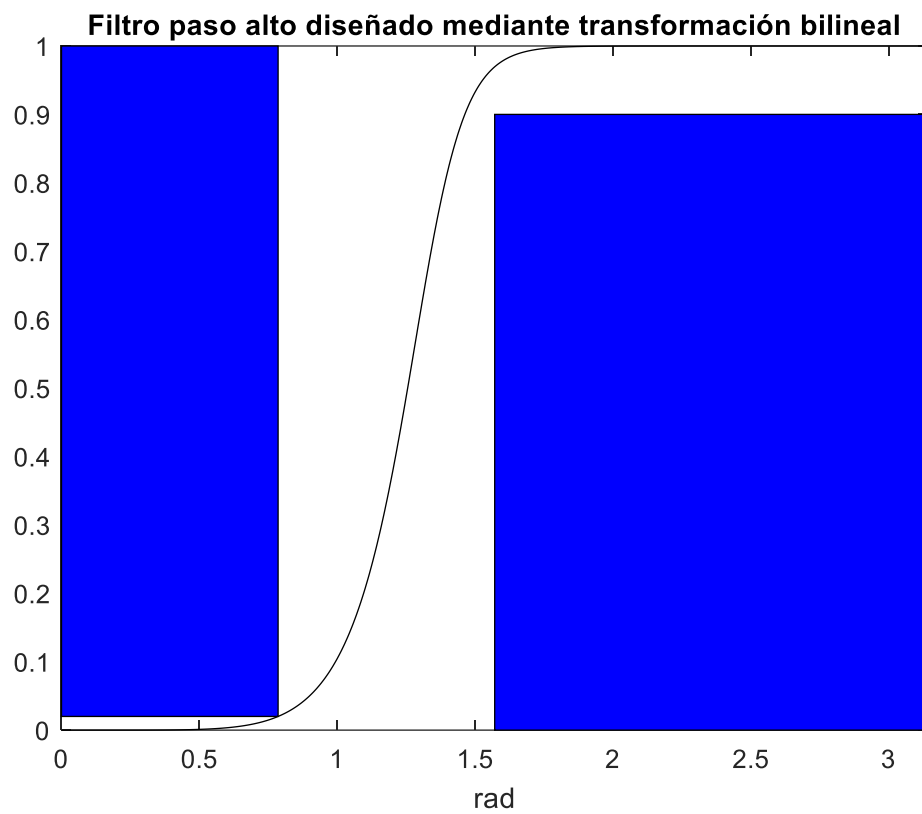
yc4 = filter(Bz,Az,xc4);

Xc4 = fft(xc4, 1024);
Yc4 = fft(yc4, 1024);
ejec4 = linspace(0, 2*pi, muestras4);

figure;
subplot(211); plot(ejec4,abs(Xc4), 'black');
axis([0 pi 0 (max(abs(Xc4))+20)]); title('Espectro de la señal de
entrada en veces');

xlabel('rad');
subplot(212); plot(ejec4,abs(Yc4), 'black');
axis([0 pi 0 (max(abs(Yc4))+20)]); title('Espectro de la señal de
salida en veces');
xlabel('rad');

```



1.5 Repita el Ejercicio 3.4 utilizando los **distintos filtros prototipo** vistos: ChebyshevTipo I y II, y Elíptico. ¿Qué conclusiones obtiene?

Nota: No olvide utilizar la opción 'high' a la hora de llamar a las funciones usadas para diseñar los distintos filtros (cheby1, cheby2 y ellip).

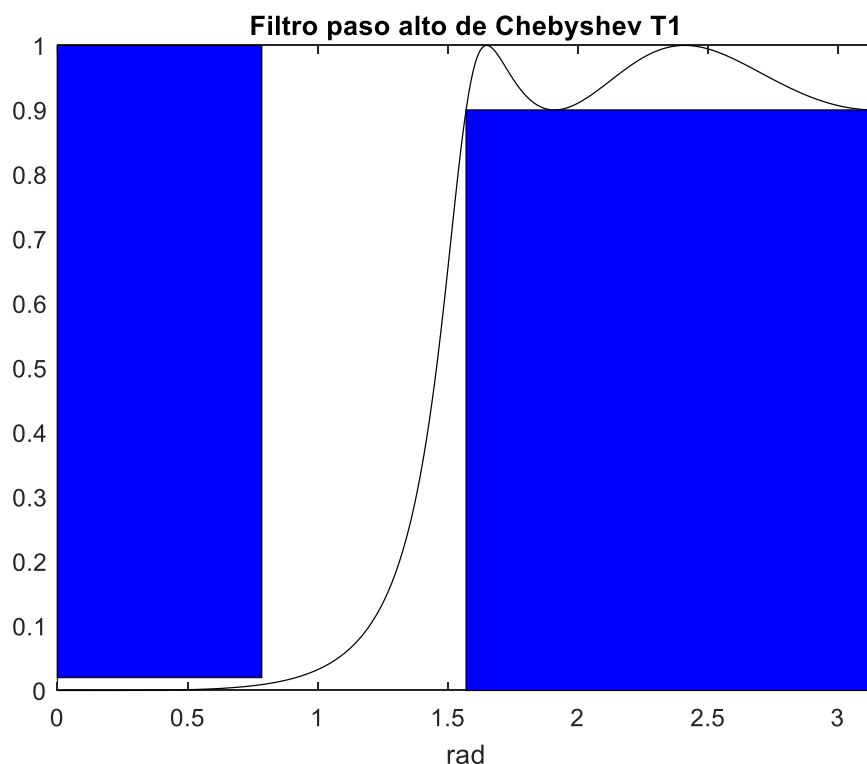
```
Td = 1;
omegaatenuacion5 = pi/4;
omegapaso5 = pi/2;
rizpas5 = 0.1;
rizat5 = 0.02;
muestras5 = 1024;

ompasocont5 = (2/Td) *tan(omegapaso5 / 2);
omatcont5 = (2/Td) *tan(omegaatenuacion5 / 2);

limatpas5 = -20*log10(1-rizpas5);
limatat5 = -20*log10(rizat5);

[N, fcorte] = cheblord(ompasocont5, omatcont5, limatpas5, limatat5, 's');
[b, a] = cheby1(N, limatpas5, ompasocont5, 'high', 's');
[Bz, Az] = bilinear(b, a, 1/Td);
[h, ejec] = freqz(Bz, Az, muestras5);

figure;
plot(ejec, abs(h), 'black');
axis ([0 pi 0 1]);
patch([0 0 omegaatenuacion5 omegaatenuacion5], [rizat5 1 1 rizat5], 'b');
patch([omegapaso5 omegapaso5 pi pi], [0 (1-rizpas5) (1-rizpas5) 0], 'b');
title('Filtro paso alto de Chebyshev T1');
xlabel('rad');
```



```

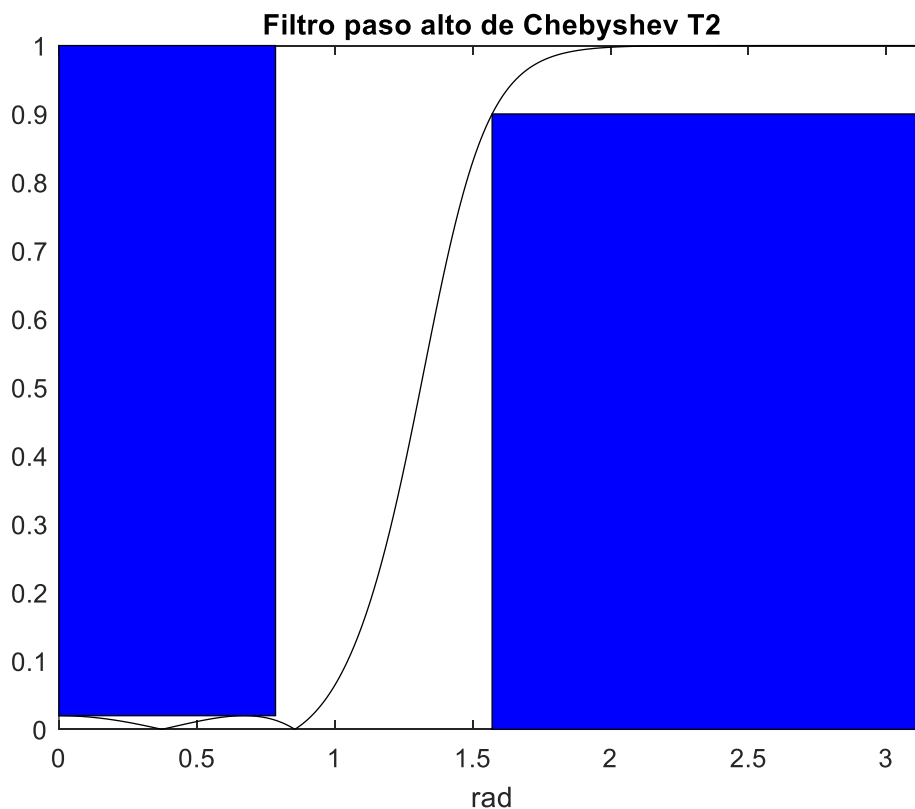
Td = 1;
omegaatenuacion5 = pi/4;
omegapaso5 = pi/2;
rizpas5 = 0.1;
rizat5 = 0.02;
muestras5 = 1024;

ompasocont5 = (2/Td) *tan(omegapaso5 / 2);
omatcont5 = (2/Td) *tan(omegaatenuacion5 / 2);

limatpas5 = -20*log10(1-rizpas5);
limatat5 = -20*log10(rizat5);

[N, fcorte] = cheb2ord(ompasocont5, omatcont5, limatpas5, limatat5, 's');
[b, a] = cheby2(N, limatat5, fcorte, 'high', 's');
[Bz, Az] = bilinear(b, a, 1/Td);
[h, ejec] = freqz(Bz, Az, muestras5);
%Representación
figure;
plot(ejec, abs(h), 'black');
axis ([0 pi 0 1]);
xlabel('rad');
patch([0 0 omegaatenuacion5 omegaatenuacion5], [rizat5 1 1 rizat5], 'b');
patch([omegapaso5 omegapaso5 pi pi], [0 (1-rizpas5) (1-rizpas5) 0], 'b');
title('Filtro paso alto de Chebyshev T2');
xlabel('rad');

```



```

Td = 1;
omegaatenuacion5 = pi/4;
omegapaso5 = pi/2;
rizpas5 = 0.1;
rizat5 = 0.02;
muestras5 = 1024;

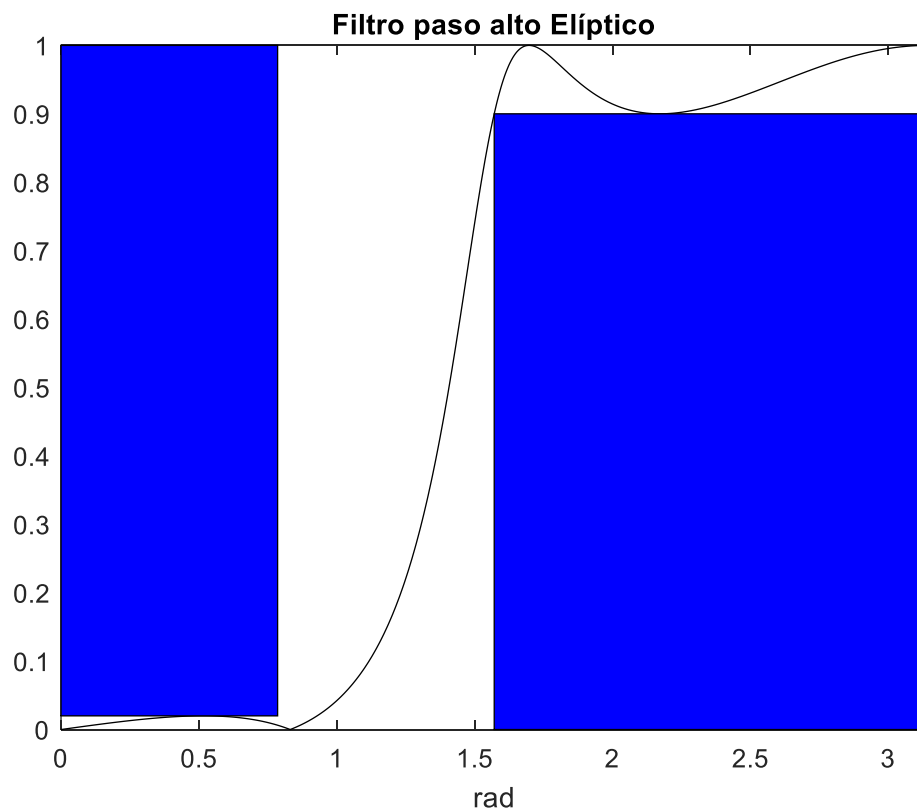
ompassocont5 = (2/Td) *tan(omegapaso5 / 2);
omatcont5 = (2/Td) *tan(omegaatenuacion5 / 2);

limatpas5 = -20*log10(1-rizpas5);
limatat5 = -20*log10(rizat5);

[N, fcorte] = ellipord(ompassocont5, omatcont5, limatpas5, limatat5, 's');
[b, a] = ellip(N, limatpas5, limatat5, ompasocont5, 'high', 's');
[Bz, Az] = bilinear(b, a, 1/Td);
[h, ejec] = freqz(Bz, Az, muestras5);

figure;
plot(ejec, abs(h), 'black');
axis ([0 pi 0 1]);
patch([0 0 omegaatenuacion5 omegaatenuacion5], [rizat5 1 1 rizat5], 'b');
patch([omegapaso5 omegapaso5 pi pi], [0 (1-rizpas5) (1-rizpas5) 0], 'b');
title('Filtro paso alto Elíptico');
xlabel('rad');

```





## 2 Diseño de un Filtro para una Aplicación Real

Tal y como se indicó en la práctica anterior, una de las principales aplicaciones de los filtros selectivos en frecuencia es la eliminación de ruido e interferencias de las señales de interés. En este último apartado vamos a retomar la aplicación que exploramos al final de la práctica anterior: la eliminación de la fluctuación de la línea de base (“baseline wander”), un ruido de muy baja frecuencia que aparece típicamente por debajo de 0,5 Hz, en señales de electrocardiograma (ECG). En lo sucesivo se le pide diseñar un filtro paso alto IIR para eliminar dicha interferencia, comparándolo con el filtro FIR diseñado en la práctica anterior.

- 2.4 Cargue la señal proporcionada (s0273lremV4T10s), que contiene 10 segundos de un solo canal de ECG (derivación pre-cordial V4) tomado con una frecuencia de muestreo  $f_s = 1000$  Hz. Dibuje la señal tanto en el dominio del tiempo como de la frecuencia (definiendo correctamente los correspondientes vectores de tiempo y frecuencia). Observe la repetición de picos en la señal temporal (correspondientes a los distintos latidos cardiacos) y la fluctuación de la línea de base de la señal. En el dominio frecuencial, confirme que el espectro está concentrado dentro de los límites esperados (haciendo un zoom de la figura o usando el comando xlim).

```
load s0273lremV4T10s.mat x

Tc = 10;
t = 1:length(x);
fs = 1000;
Ts = 1/fs;
n = 0:1:fs;
N = 2048;
subplot(2,1,1);
plot(t,x);

XF = fft(x,N);
XF = fftshift(XF);

k = 0:N-1;
Om = 2*pi*k/N;
Omc = Om/Tc;

subplot(2,1,2);
plot(Om,abs(XF));
axis([2 4 0 2e5]);
```

