Ejemplo de filtros sobre fragmento de música

Vídeo-presentación en: http://personales.upv.es/asala/YT/V/filtmlre.html

Objetivos: Comprender el funcionamiento de los filtros resonantes continuos, examinando su efecto sobre un fragmento musical.

Nota: los filtros continuos eran una opción de implementación usual en 1960-1980. Hoy en dia, excepto en electrónica de potencia (fuentes de alimentación, inversores, ...) para procesado de audio (como en este ejemplo), el procesado suele ser digital, usando filtros discretos [*https://en.wikipedia.org/wiki/Digital_filter*].

Música: Hep Cat Jive, Purple planet music, Geoff Harvey, Manchester, UK.

Licencia de música: Archivo libre para uso personal de http://www.purple-planet.com/jazz/4583971402

[https://api.soundcloud.com/tracks/190751014/download?secret_token=s-2Kn9H&client_id=cUa40O3Jg3Emvp6Tv4U6ymYYO50NUGpJ]

diseño de filtros

```
s=tf('s');%control system toolbox
```

Filtros resonantes

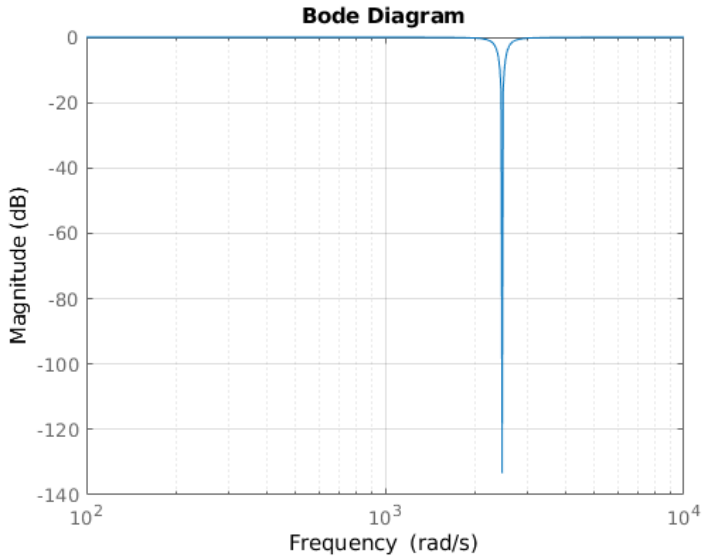
Indicados para dejar pasar / eliminar bandas muy estrechas.

Generamos escala de Do Mayor, y diseñamos un filtro ranura estrecha "notch" que elimina la nota "Sol".

```
Tiempopornota=0.5;
teclaspiano=[0 2 4 5 7 7 9 11 12];
escala=generamelodia(teclaspiano,Tiempopornota);

frecSol2=220*2*pi*2^(10/12); %frecuencia de la nota "Sol" de la escala
FiltroRanuraResonante=(s^2+2*0.00*frecSol2*s+frecSol2^2)/(s^2+2*0.05*frecSol2*s+frecSol
```

```
FiltroRanuraResonante =

       s^2 + 6.066e06
  ------------------------
  s^2 + 246.3 s + 6.066e06

Continuous-time transfer function.
```

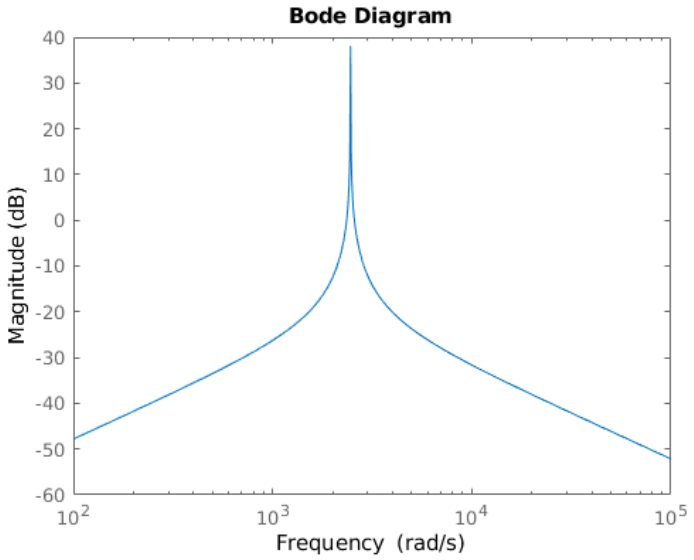
```
bodemag(FiltroRanuraResonante), grid on
```




```
reproduce(escala)
Filtra_y_reproduce(FiltroRanuraResonante,escala)
%filtro banda muy estrecha:
FiltroPasaResonante=(s^2+2*0.04*frecSol2*s+frecSol2^2)/(s^2+2*0.0005*frecSol2*s+frecSol
```

```
FiltroPasaResonante =

          246.3 s^3 + 4.853e04 s^2 + 1.494e09 s
  ------------------------------------------------------
  s^4 + 248.8 s^3 + 1.213e07 s^2 + 1.509e09 s + 3.68e13

Continuous-time transfer function.
```

```
bodemag(FiltroPasaResonante)
```



```
Filtra_y_reproduce(FiltroPasaResonante,escala)
```

Funciones auxiliares

```
function cancionFiltrada=Ecualizador(cancion,Filtro)
    numcanales=size(cancion.muestras,2);
    FiltroStereo=Filtro*eye(numcanales);
    Muestras=cancion.muestras;
    frecmuestreo=cancion.frecmuestreo;  PeriodoMuestreo=1/frecmuestreo;
    TiemposMuestras=(0:(size(Muestras,1)-1))*PeriodoMuestreo;
    MuestrasF=lsim(FiltroStereo,Muestras',TiemposMuestras);
    cancionFiltrada.muestras=MuestrasF;
    cancionFiltrada.frecmuestreo=frecmuestreo;
end

function reproduce(cancion)
    soundsc(cancion.muestras,cancion.frecmuestreo);
    pause(size(cancion.muestras,1)/cancion.frecmuestreo); %esperamos mientras escuchamo
    pause(0.5); %medio segundo extra de pausa entre canción y canción
end

function Filtra_y_reproduce(Filtro,cancion)
        reproduce(Ecualizador(cancion,Filtro))
end

function cancion=generamelodia(teclaspiano,Tiempopornota)
    Nnotas=length(teclaspiano);
    FrecMuestreo=44100;Ts=1/FrecMuestreo; %frec muestreo en Hz
    ccc=zeros(Nnotas*Tiempopornota*FrecMuestreo,1);
    FrecDo2=220*2^(3/12)*2*pi;
    for numnota=1:Nnotas
        Frecnota=FrecDo2*2^(teclaspiano(numnota)/12);
        muestras=(numnota-1)*FrecMuestreo*Tiempopornota+(1:Tiempopornota*FrecMuestreo);
        tiempomuestras=muestras*Ts;
        ccc(muestras)=sin(tiempomuestras*Frecnota);
    end
    cancion.muestras=ccc;
    cancion.frecmuestreo=FrecMuestreo;
end
```