

# laSalle

## LSTools

Pràctica PDS – Fase 1  
2023-24



## INDEX

<b>1. INTRODUCCIÓ .....</b>	<b>2</b>
<b>2. DESCRIPCIÓ DE LA PRÀCTICA.....</b>	<b>2</b>
<b>3. INTRODUCCIÓ A FILTER DESIGNER .....</b>	<b>3</b>
<b>4. FASE I.....</b>	<b>6</b>
4.1 Disseny de filtres.....	7
4.2 Disseny i ús dels filtres .....	8
4.3 Resposta en freqüència .....	10
4.4 Diagrama de pols i zeros .....	11
4.6 Reverberació .....	12
4.7 Test dels efectes .....	14
<b>5. ENTREGA .....</b>	<b>15</b>
<b>6. ANNEX .....</b>	<b>16</b>

## 1. INTRODUCCIÓ

L'estudi de gravació de La Salle Campus Barcelona vol crear un plugin nou per l'eina d'edició sonora LSTools que permeti equalitzar o aplicar reverberació al senyal a les produccions realitzades pels alumnes i es pugui utilitzar en temps real en les entrevistes del canal de televisió SalleVisión. Per aquest motiu, s'han posat en contacte amb els estudiants d'enginyeria que cursen Processament Digital del Senyal perquè els ajudin a aconseguir-ho fent el primer prototip en Matlab.



## 2. DESCRIPCIÓ DE LA PRÀCTICA

La pràctica es dividirà en dues fases que es s'han d'implementar de manera seqüencial pel correcte funcionament d'aquesta:

- **FASE I: Disseny dels diferents efectes (equalitzador basat en un banc de filtres IIR, i reverberació) i proves de test sobre senyals d'exemple.**
- **FASE II: Implementació del plugin d'efectes amb una Graphical User Interface (GUI) que permeti veure i escoltar a temps real el resultat.**

A l'hora de qualificar la pràctica es valorarà cada una de les fases per separat i la pràctica en conjunt, per tant, és molt important complir amb els requeriments de cada fase i els requeriments finals de la pràctica.

Fase I	5 punts
Fase II	5 punts

La entrega de la **primera fase** serà el dia **22/12/23 sobre 10**, o bé **14/1/23 sobre 7**.  
La **segona fase** s'entregarà el **dia 14/1/23**.

### 3. INTRODUCCIÓ A FILTER DESIGNER

Filter Designer és una potent interfície gràfica d'usuari (GUI) en el quadre d'eines de processament de senyals per dissenyar i analitzar filtres. Filter Designer permet dissenyar ràpidament filtres FIR o IIR digitals establint especificacions de rendiment de filtres, important filtres des del seu espai de treball MATLAB o afegint, movent, o eliminant pols i zeros. El disseny de filtres també proporciona eines per analitzar els filtres dissenyats, com gràfiques de resposta de magnitud i fase, o diagrames de pols i zeros.

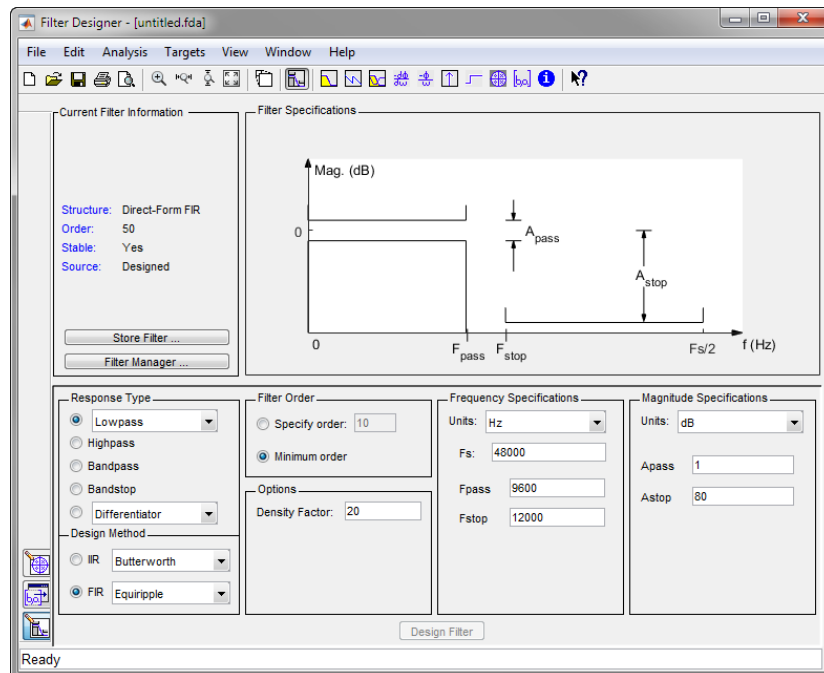


Figura 1: GUI filter designer.

La documentació corresponent al Filter Designer la podeu trobar en el següent enllaç. A continuació, passarem a explicar breument com treballar amb la interfície.

<https://es.mathworks.com/help/signal/examples/introduction-to-filter-designer.html>

#### Executar l' interfície

Per obrir la interfície, s'ha d'escriure la següent comanda en la línia de comandes de Matlab.

```
>>filterDesigner
```

## Configuració de paràmetres

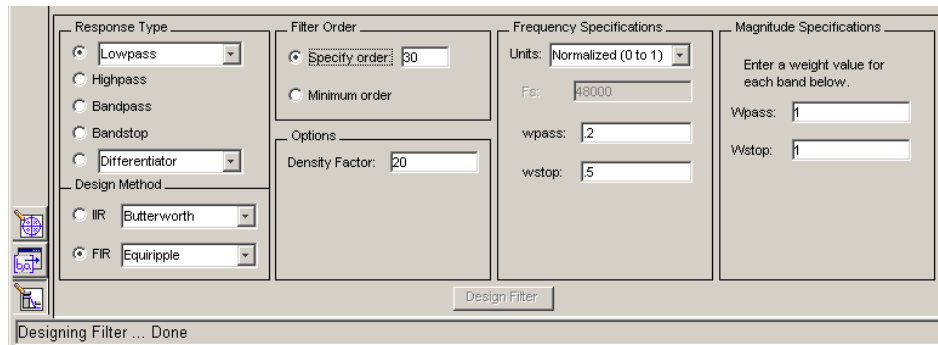


Figura 2: Configuració de paràmetres dins de Filter Designer.

En la part inferior de l'interfície trobareu tots els paràmetres a configurar per la creació de cada un dels filtres. Els paràmetres de cada filtre els trobareu en l'apartat de l'explicació de la Fase I.

## Disseny i guardat del filtre

Un cop configurats els paràmetres del filtre, podem dissenyar el filtre clicant el botó inferior de la interfície *Design Filter*. Si el resultat del filtre dissenyat es l'esperat, podem guardar el filtre clicant el botó del lateral esquerre de la interfície *Store Filter*.

## Exportar paràmetres al workspace

Finalment, per exportar els coeficients del filtre dissenyat a l'espai de treball de Matlab, polsarem en el menú *File > Export...* i ens apareixerà la següent finestra.

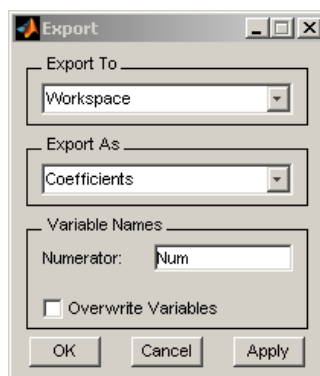


Figura 3: Exportar coeficients.

A la finestra, seleccionar exportar els coeficients al workspace i assignar el nom de la variable del workspace on es guardaran aquests coeficients.

En el cas dels filtres IIR, perquè apareguin els coeficients de numerador i denominador per separat, abans d'exportar el filtre s'ha d'anar a menú *Edit > Convert to Single Section*.

No obstant, a vegades aquesta conversió pot donar problemes per ordres elevats i és millor guardar el disseny en el format original.

### Guardar sessió de filtres

Per guardar una sessió de filtres dissenyats, abans de tancar el Filter Designer, seleccionar en el menú superior *File > Save Session*, i es generarà un arxiu amb extensió **.fda** on es guardaran tots els filtres dissenyats durant la sessió.

### Filter Visualization Tool

Per poder veure la resposta conjunta dels filtres dissenyats, utilitzarem l'eina [FVTool](#). Per fer-ho, pulsarem el botó de l'interfície *Filter Manager*, seleccionarem els filtres dissenyats i pulsarem el botó *FVTool*.

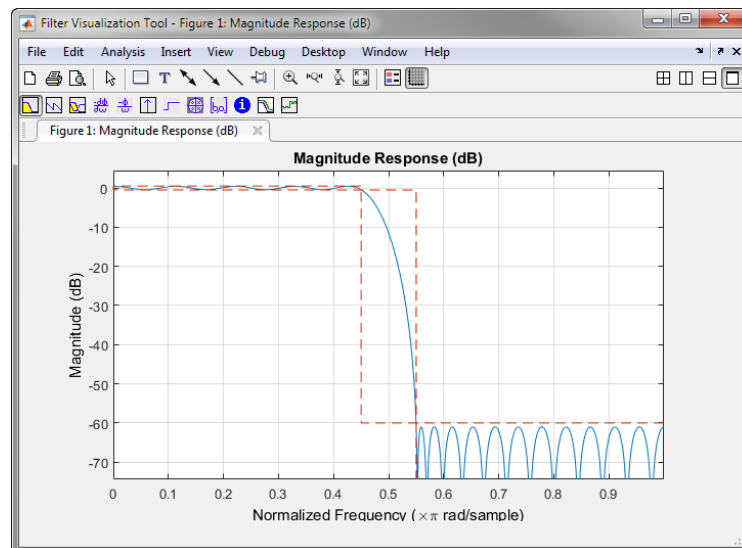
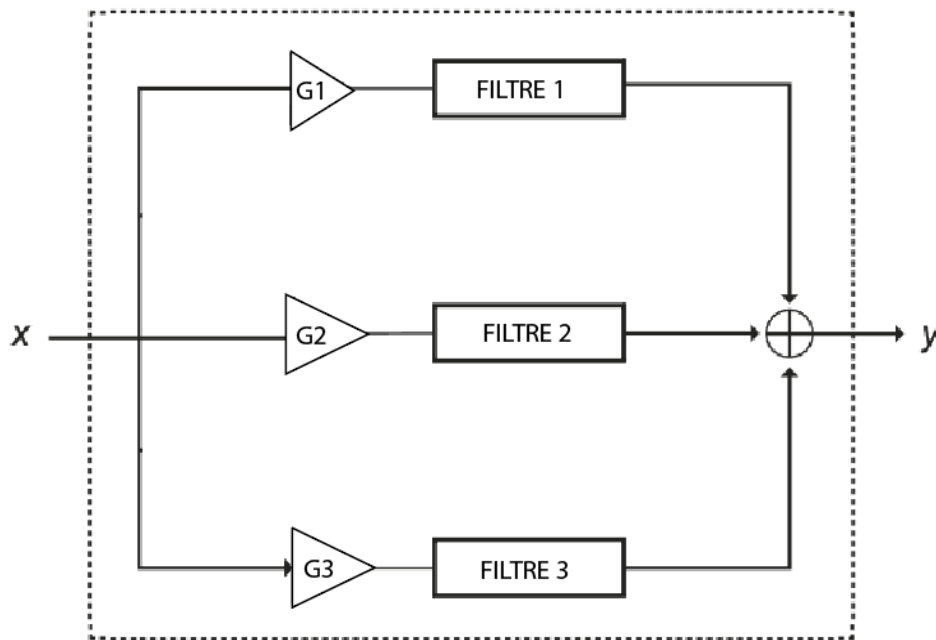


Figura 4: FVTool.



#### 4. FASE I

L'equalitzador estarà format per 3 filtres. Cada un d'ells tindrà associat un guany independent dels altres. No obstant, els 3 filtres treballen de forma paral·lela, tal i com es mostra en la següent figura, és a dir, la sortida de cada filtre se suma amb la dels altres per tal de generar la sortida final. Per tant, el disseny dels diversos filtres s'ha de realitzar de forma que permeti obtenir també el senyal original (p.ex. si tots els guanys es fixen a 0 dB).



*Figura 5. Equalitzador basat en un banc de 3 filtres.*

L'objectiu de l'equalitzador és filtrar i modificar senyals d'àudio, pel que utilitzarem una freqüència de mostreig molt típica en l'àmbit de la producció d'àudio: 44100 Hz.

A continuació, passem a analitzar els paràmetres de cadascun dels filtres, que dissenyarem amb aquesta freqüència de mostreig ( $f_s$ ) esmentada. D'aquesta forma, quan vulguem fer alguna prova amb àudios que estiguin mostrejats a freqüències diferents a la de disseny, aquests senyals s'hauran de remostrejar prèviament al seu filtratge.

***NOTA: És important comprovar que la  $f_s$  de l'àudio que esteu utilitzant es 44100 Hz, en el cas que la  $f_s$  no sigui 44100 Hz, aquest s'ha de remostrejar, ja que els filtres s'hauran dissenyat per treballar a aquesta freqüència.***

**PREGUNTA 1.** Fent ús de les funcions `rat.m` per aproximar qualsevol número  $X$  per un quocient de números enters  $N/D$  (o número racional) i `interp1.m` per remostrejar un senyal  $x$  qualsevol interpolant en factor  $L$  i delmant en factor  $M$ , genera les línies de codi Matlab que permeten passar d'un senyal audioIN mostrejat a una freqüència qualsevol  $f_s$  a la freqüència de 44100 Hz. En l'aproximació de la funció `rat` utilitza una tolerància màxima de  $10^{-4}$ .

Per l'equalitzador basat en banc de 3 filtres usarem la següent configuració:

Filtre	Tipus de filtre	Fc1	Fc2
LP	<i>Lowpass</i>	-	250 Hz
BP	<i>Bandpass</i>	250 Hz	2250 Hz
HP	<i>Highpass</i>	2250 Hz	-

Com podem veure, el filtre passa-baixes (LP) és el que té una banda de pas més petita, ja que deixa passar els senyals entre 0 i 250 Hz. En canvi el filtre passa-banda té un amplada de banda de 2200 Hz entre 250 i 2250 Hz. Finalment, el filtre passa-altes conté la banda més ampla de totes, entre 2250 i 22050 Hz (la meitat de la freqüència de mostratge). Aquesta configuració obeeix a una de les característiques de la percepció humana del so, que dona major rellevància a la baixa freqüència (on l'oïda té més resolució) mentre que menys a altes freqüències (on la resolució és molt menor).

#### 4.1 Disseny de filtres

##### Filtres IIR

Pel disseny dels filtres IIR, s'ha de seleccionar filtres de tipus Butterworth, que tenen la característica de presentar un guany amb una trajectòria màximament suau i sense arrissats<sup>1</sup>. Els filtres han de ser d'ordre 12 i els valors de les freqüències de tall són els que apareixen a la taula anterior. Per tant, s'ha de dissenyar i exportar 3 filtres IIR que compleixin els requisits que s'han especificat. Un cop dissenyats els filtres, comprovar visualment mitjançant l'eina *FVTool*<sup>2</sup> la resposta de guany en dB del filtre, comprovant que les freqüències de tall corresponen a les especificades i la visualització de les 3 bandes dona un resultat similar al de la figura següent:

<sup>1</sup> L'arrissat és una característica d'alguns filtres, que es presenta en forma d'oscil·lació en el domini de l'amplitud de la seva funció de transferència en alguna de les bandes estipulades (banda de pas, i/o banda de rebuig)

<sup>2</sup> La podeu trobar en el del botó Filter Manager de l'eina *filterDesigner*. Si seleccioneu els tres filtres dissenyats podreu visualitzar les respostes dels 3 a l'hora.



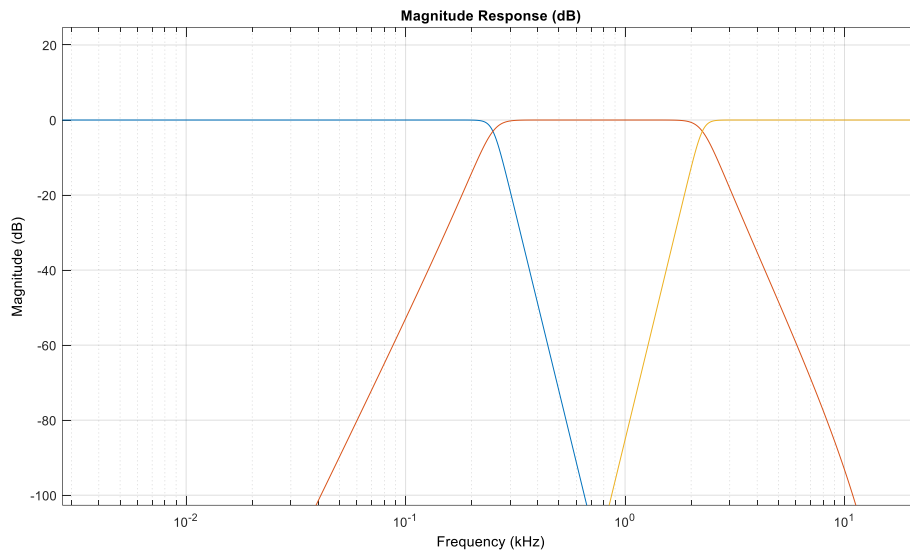


Figura 6: Filtres IIR de tipus Butterworth i ordre 12.

**NOTA:** Utilitza una escala logarítmica per representar l'eix freqüencial (la eina FVTool permet fer-ho en la secció Analysis parameters).

## 4.2 Disseny i ús dels filtres

Un cop dissenyats els filtres, exportar els coeficients al workspace i generar l'arxiu *.mat* (*filtersSOS.mat* amb les variables del workspace dels filtres IIR). Els filtres, al ser d'ordre elevat (12), és recomanable representar-los amb la mateixa forma en què apareixen després del seu disseny: amb seccions d'ordre 2 en cascada (en el nostre cas, seran 6 seccions per a cada banda de l'equalitzador). Això dona major estabilitat numèrica, ja que si es fa la conversió al filtre equivalent d'ordre 12, el guany podria tenir distorsions no desitjades. Així doncs, quan exportem els coeficients d'un dels filtres, ens trobarem que aquest està format per una matriu SOS (*Second Order Sections*) i un vector de guanys *G* (*Scale values*). La matriu de SOS té tantes columnes com seccions i tantes files com coeficients ocupen els numeradors i els denominadors de cada secció. Com són seccions d'ordre 2, aquests ocuparan 3 coeficients, per tant tindrem que cada fila de la matriu SOS de 6x6 té la forma següent:

$$\begin{array}{c} \underbrace{[b_0 \ b_1 \ b_2]}_{\text{Coeficients del denominador}} \ \underbrace{[1 \ a_1 \ a_2]}_{\text{Coeficients del numerador}} \end{array}$$

D'altra banda, el vector de guanys *G* conté els factors de guany que multipliquen a cada secció, de forma que el seu producte és igual al guany total del filtre. Aquest guany el podem aplicar, per exemple, multiplicant-lo en el numerador del primer subfiltre, de forma que el guany estarà ja incorporat als coeficients de les seccions d'ordre 2:

$$SOS(1,1:3) = SOS(1,1:3) * prod(G);$$

Finalment, per a filtrar un senyal  $x$  amb un filtre dissenyat amb seccions d'ordre 2 ho podem fer amb la crida següent:

$$\text{audioOUT} = \text{sosfilt}(\text{SOS}, \text{audioIN});$$

La funció *sosfilt.m* ja s'encarrega de fer la crida a l'estructura formada per 6 filtres en cascada, tot evitant els errors de precisió numèrica que apareixerien si el filtre s'hagués convertit a una única secció d'ordre 12.

Per generar l'arxiu *filtersSOS.mat*, cal utilitzar la funció *save.m*. Els arxius generats haurien de tenir emmagatzemats les següents variables (un parell per a cada subbanda):

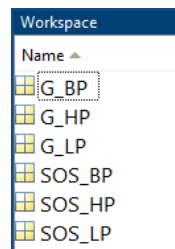


Figura 7: Variables workspace de l'arxiu *filterSOS.mat*.

Pel disseny del banc de 3 filtres en paral·lel (com es mostra a la figura 5), considerarem inicialment cada filtre per separat, tal i com es mostra en la següent imatge:



Figura 8: Procés filtrat.

D'aquesta manera posarem un valor de guany indicat en dB, però convertit a lineal per a aplicar el factor multiplicatiu  $G_x$ . Aquest valor de guany lineal, el podem aplicar multiplicant els coeficients del numerador del primer subfiltre d'ordre 2, tal i com s'ha fet amb el vector de guanys del disseny.

Un cop aplicat el guany als coeficients SOS del filtre, filtrarem la senyal d'entrada amb la funció *sosfilt.m* i obtindrem la sortida per aquest filtre en concret. Per obtenir la sortida final del banc de filtres, sumarem les sortides de cadascun dels tres filtres del banc.

### 4.3 Resposta en freqüència

Una funció molt comú en Matlab per obtenir la resposta en freqüència d'un filtre digital es la funció *freqz.m*. Per comprovar que el disseny dels filtres és correcte, podem assignar diferents guanys a cada filtre i veure la seva resposta en freqüència (veure apartat 4.7).

#### Filtres IIR

Pels filtres IIR, primer necessitem calcular la resposta impulsional  $h[n]$  de cada filtre. Per a això, s'ha de filtrar una delta amb zeros al final (els suficients zeros per a que  $h[n]$  convergeixi a un valor proper a zero, assumint que es tracta d'un filtre estable). El codi a fer hauria ser semblant al següent:

```
N = numero de zeros; % es recomanable posar un numero prou elevat
x=[1;zeros(N-1,1)];
h_i = sosfilt(SOS_i,x);
```

essent  $SOS_i$  la matriu SOS del filtre i-èssim del banc de filtres, i  $h_i$  el vector temporal que conté la resposta impulsional associada.

Verifiqueu que el número de zeros escollit permet veure una  $h[n]$  que acaba convergint gairebé a zero per a tots els filtres del banc, així com pel filtre global.

Pel filtre global, el qual tindrà uns guanys concrets fixats per a cada banda en dB, recordeu que les matrius SOS han d'incorporar el guany de cada banda convenientment passat a lineal en els coeficients del numerador del primer subfiltre.

Una cop calculada  $h[n]$  de cada filtre individual i del banc de filtres global per a uns guanys concrets fixats per cada banda, podem mostrar les corresponents respostes en freqüència. Per a fer-ho, abans usarem la crida a la funció *freqz.m* per obtenir aquestes respostes en freqüència, passant-li la corresponent resposta impulsional com a primer argument:

$$[H,f] = \text{freqz}(h,1,\text{nfft},fs)$$

escollint un número de punts en freqüència de 2048.

***NOTA: Mostrar les gràfiques del guany de la resposta freqüencial en l'eix x logarítmic usant la crida a la funció semilogx.m. Recordar que el guany en dB d'un valor de guany lineal  $K$  s'obté com  $10 \cdot \log_{10}(K)$  essent  $K$  el valor absolut o mòdul de la resposta en freqüència.***

A la figura següent podem veure un exemple de visualització de les respostes en freqüència escollint uns guanys de -20 dB a la banda de baixa freqüència, 25 dB a la de freqüències mitges i 0 dB a les altes freqüències.

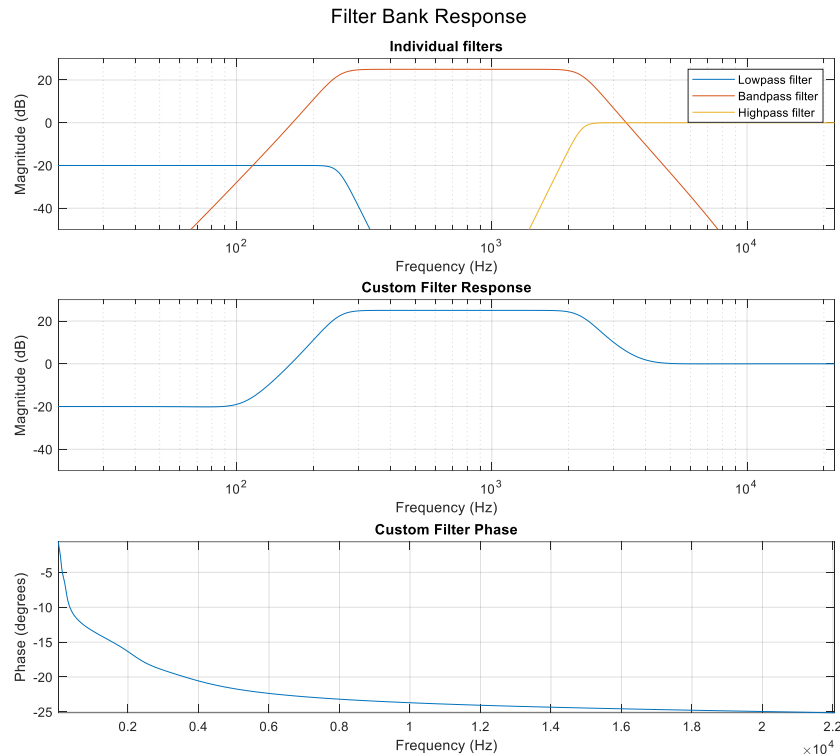


Figura 9. Resposta en freqüència del banc de filtres per a uns guanys de -20 dB (baixa freq.), 25 dB (fre. mitges) i 0 dB (altes freq.): guany final dels filtres individuals, guany final de la resposta global, i fase de la resposta global.

#### 4.4 Diagrama de pols i zeros

Des de l'eina *filterDesigner* podem observar el diagrama de pols i zeros que es genera de cada filtre dissenyat. Això es pot fer tant des de la mateixa finestra que s'obre quan es crida a l'eina com seleccionant un dels dissenys des del botó Filter Manager, i pitjant a sobre de FVTool (Filter Visualization Tool).

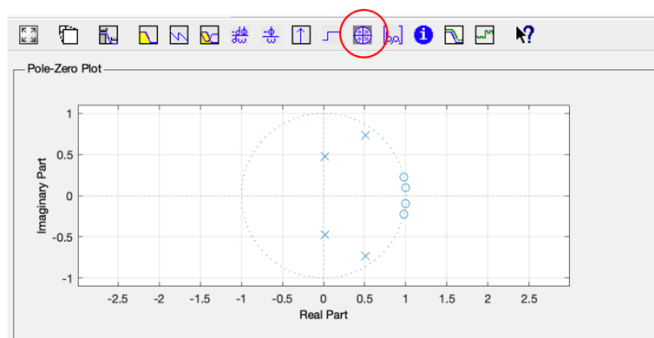
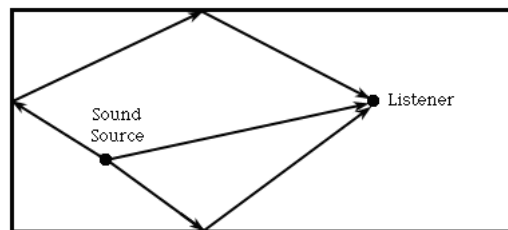


Figura 10: Diagrama de pols i zeros.

**PREGUNTA 2.** Mostrar el diagrama de pols i zeros dels tres filtres IIR dissenyats del banc de filtres i justificar les seves diferències. Observeu la posició dels pols i els zeros fent zoom a les zones d'interès, i relacioneu-la amb la resposta freqüencial dels filtres.

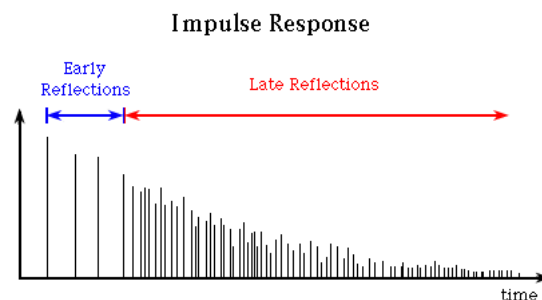
#### 4.6 Reverberació

L'acústica d'una sala es pot representar a partir de l'anàlisi de la seva resposta impulsional, la qual s'obté en excitar-la amb senyals amb espectre constant o pla i s'analitza la relació entre el senyal enregistrat (sortida) i aquesta excitació. El procés físic que es produeix quan una font sonora emet un so dins d'una sala amb una certa geometria es pot entendre com una propagació de rajos acústics que viatgen tot propagant-se per l'espai i canviant la seva direcció cada cop que la ona topa amb una superfície de la sala, com es pot veure en l'exemple de la figura següent.



*Figura 11: Exemple de geometria de sala rectangular i primeres reflexions que es produeixen entre una Font de soroll i un oient.*

En la següent figura podem veure la resposta impulsional típica d'una sala, a on es distingeixen les primeres reflexions (Early reflections) de la part difusa o reverberació (Late reflections), perquè mentre les primeres són les degudes als primer rebots dels rajos acústics, la segona conté una major proporció d'ones que es generen després d'aquestes primeres reflexions, obtenint una major densitat de rajos.



*Figura 12. Exemple de resposta impulsional d'una sala*

Es pot implementar l'efecte de reverberació tot usant com a resposta impulsional un soroll blanc Gaussià modulad amb una funció exponencial decreixent en el temps. L'expressió d'aquesta resposta impulsional és la següent:

$$h_R(t) = A(t)s(t)$$

$$A(t) = c^t u(t)$$

a on  $A(t)$  és l'envolupant exponencial decreixent de la resposta impulsional, i  $s(t)$  és un soroll blanc i Gaussià de mitja 0 i variància 1. La constant real  $c$  ( $c < 1$ ) determina el grau de decreixement de l'energia de la resposta impulsional.

Un paràmetre que permet entendre millor el comportament de la reverberació és el *temps de reverberació*  $T_r$ , que es defineix com el temps que triga en decaure l'energia de la resposta impulsional 60 dB respecte del valor inicial, és a dir:

$$10 \log_{10}|A(T_r)| = 10 \log_{10}|A(0)| - 60$$

**PREGUNTA 3: Calculeu el paràmetre  $c$  en funció de  $T_r$**

Trobar el valor del paràmetre  $c$  en funció del temps de reverberació  $T_r$ , a partir de la condició anterior.

**PREGUNTA 4: Exercici**

Expressa el càlcul de la resposta impulsional d'un sistema reverberador en Matlab, si treballem a una certa freqüència de mostratge  $F_s$ , i agafant mostres fins el valor del temps de reverberació  $T_r$ .

**NOTA:** La funció `randn.m` permet generar mostres d'un soroll Gaussià amb mitja 0 i variància 1.

Per aplicar l'efecte de reverberació sobre un senyal donat, usarem la funció `fftfilt.m`, que usa el mètode basat en la FFT (transformant el senyal i la resposta i fent la convolució circular amb el producte de FFTs), més eficient que la funcions `filter.m` o `conv.m` quan el filtre té una resposta impulsional molt llarga. Donat que la funció `fftfilt.m` genera a la sortida un senyal de la mateixa durada que l'entrada, quan apliquem l'efecte ho farem afegint al final del senyal d'entrada una cadena de zeros que tingui una durada igual a la durada de la resposta impulsional menys 1. D'aquesta forma, ens assegurem que no quedi tallada la cua de la reverberació si el senyal d'entrada és massa curt. Teniu en compte que per a concatenar dos vectors ho podeu fer tant amb vectors fila com amb columna (vegeu [aquest](#) link).

Per acabar de generar l'efecte de reverberació, definirem també un paràmetre d'intensitat d'efecte entre 0 i 100, i denotarem com a  $M$  (de Mix), de forma que per  $M = 100$  l'efecte sigui total, mentre que si  $M = 50$ , aquest només es mescli amb el senyal original en una proporció meitat. Per a fer-ho, primer hem de normalitzar la resposta impulsional calculada dividint-la per la seva norma Euclidiana (en Matlab, usem la funció `norm.m` per a calcular-la). En segon lloc, apliquem el coeficient de mescla  $M$  de la següent forma:

$$y_e = x*(1-M/100) + y*(M/100);$$



## 4.7 Test dels efectes

### Música amb banc de filtres

Per comprovar el funcionament del banc de filtres haurem de filtrar un senyal de música (*music.wav*) que llegirem amb la funció *audioread.m* de Matlab. Per a testear el funcionament correcte, proveu d'emfatitzar exageradament certes bandes sobre les altres (p.ex. posant un guany a 0 dB en una banda, i un de molt negatiu a les altres, p.ex. -100 dB), i finalment fent alguna equalització ad-hoc. Modificant el guany de les diferents bandes es podrà comprovar el resultat auditiu del efecte del filtre. A l'Annex teniu alguns exemples a títol de verificació del codi. Per escoltar l'àudio filtrat utilitza la funció *soundsc.m*.

### Reverberació

En el cas de la reverberació useu senyals curts com el de veu de *hola\_retallada.wav*, que, al ser un senyal més curt, us permetrà percebre millor aquest efecte. Proveu d'aplicar una reverberació amb diferents temps de reverberació (p.ex. des de 0.1 a 2 segons) i factors de mescla M (p.ex. des de 0 fins a 100).

### Gràfiques

Finalment, comprovareu la correcta implementació dels filtres a través de gràfiques i sentint els resultats. S'haurà de mostrar una figura com la següent on es mostri la senyal d'entrada (àudio input) i la senyal filtrada (àudio output) en els dominis temporal (eix lineal en segons) i en freqüència (eix logarítmic en Hz). A part, pel cas del banc de filtres, es mostrarà també la gràfica que inclou la resposta de guany per cada filtre i la global en guany i fase (vegeu exemple de la figura 9).

Al modificar els guanys, podem observar l'efecte de la equalització en las gràfiques. En la memòria s'haurà d'adjuntar al menys dos modificacions del guany i el seu corresponent efecte en las gràfiques.

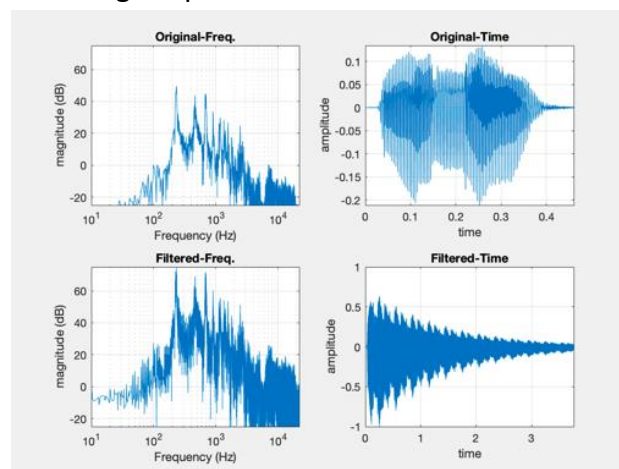


Figura 13: Gràfiques corresponent a entrades i sortides

**NOTA:** Utilitzar una escala del eix en freqüències en logarítmic amb la funció *semilogx.m*.

## 5. ENTREGA

Com a resultat d'aquesta primera fase de la pràctica heu d'adjuntar:

- **Memòria.** S'haurà d'entregar una memòria en format .pdf per cada fase per separat (si la primera fase s'entrega el 22/12/23 sobre 10). En el cas d'entregar la fase 1 sobre 7 , s'haurà d'entregar una única memòria que contingui la fase 1 i la fase 2. En la memòria s'han de veure TOTS els processos realitzats para complementar aquesta primera fase.

Elements essencials:

- Captures de tots els filtres dissenyats amb Filter Designer.
  - Resposta global dels filtres IIR del FVTool.
  - Feedback i explicació dels test realitzats.
  - Respostes de las preguntes o exercicis proposats.
- **Filters.fda.** Arxiu de la sessió de treball del disseny del filtres de l'eina Filter Designer.
- **filtersSOS.mat** Arxius amb els coeficients dels filtres IIR amb el format de les seccions d'ordre 2.
- **processaudio.m** Script de Matlab amb la implementació del codi, que generi tant el senyal amb l'efecte demanat com les gràfiques associades (veure annex per veure l'explicació dels paràmetres). A més, la funció ha de enviar ala tarja de so el senyal d'entrada amb l'efecte demanat.

Alhora de realitzar la pràctica haureu de tenir en conte las següents consideracions:

- S'han de complir totes las restriccions que marca cada apartat del enunciat.
- El codi de MATLAB s'ha de executar correctament amb els paràmetres que passeu. No s'acceptaran codis que donin errors d'execució.
- S'han de mostrar tantes figures simultàniament com es mostra a l'annex.
- Totes las preguntes que s'adjunten al document PDF han d'estar correctament justificades.

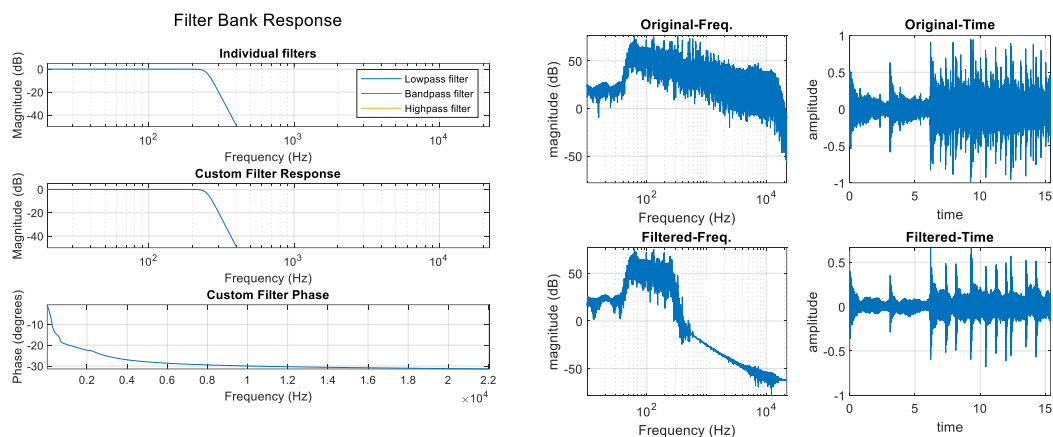
## 6. ANNEX

El codi s'haurà d'implementar dins d'una sola funció amb nom *processaudio.m*, i la capçalera de la qual serà la següent:

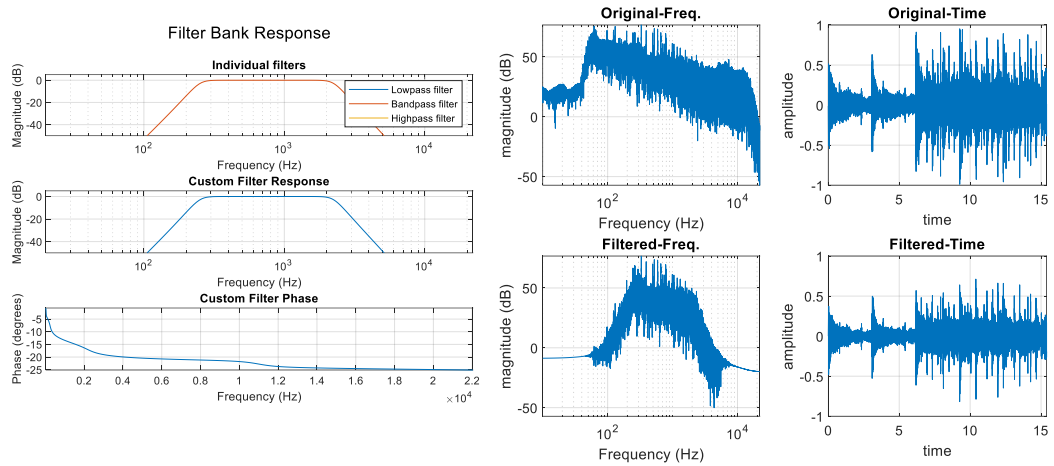
```
function [audioOUT, audioIN] = processaudio(audioINfilename, effect, param)
% Els paràmetres d'entrada són:
% audioINfilename: string amb el nom del fitxer d'àudio a processar
% effect: string que indica el tipus d'efecte ('equalizer' o 'reverb')
% param: vector numèric amb els paràmetres de l'efecte
% si effect = 'equalizer' --> param és un vector de 3 components
% amb els guanys en dBs dels 3 filtres LPF, BPF i HPF,
% respectivament.
% si effect = 'reverb' --> param(1) és el temps de reverberació en
% segons i param(2) és el coeficient de mescla (M) en %
```

La funció retornarà els senyals d'entrada (audioIN, corresponent al fitxer audioINfilename, però resmostrejat a 44100 Hz si la seva freqüència original no és aquesta) i sortida (audioOUT), i quan es cridi s'escoltarà per la tarja de sortida el senyal de sortida (amb l'efecte).

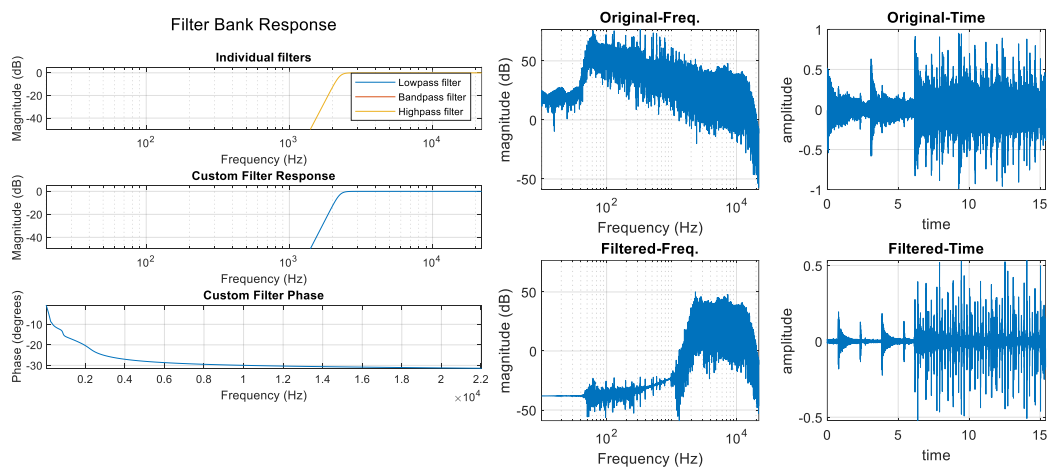
Us adjuntem unes captures de les figures que s'han de generar amb les crides a la funció perquè pugueu comprovar el correcte funcionament de la vostra fase I. En el cas de l'efecte 'equalizer':



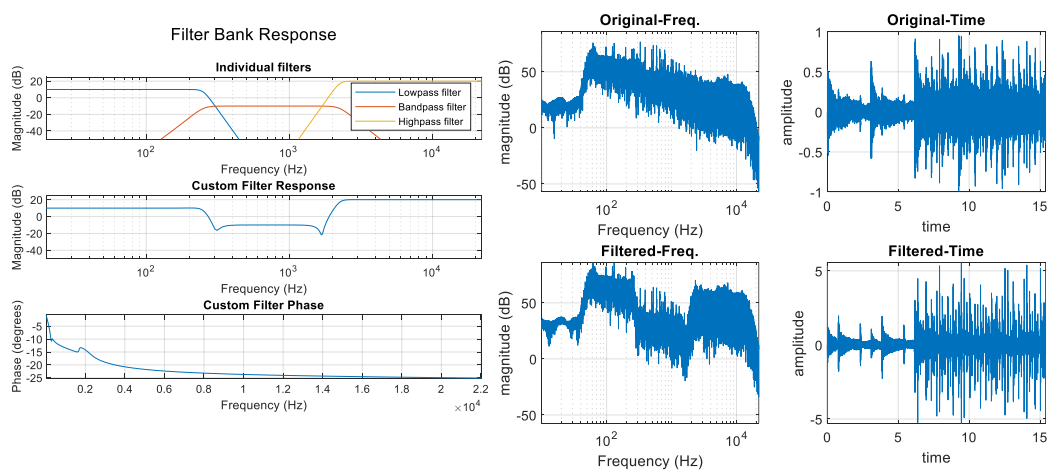
Comanda 1: *processaudio ('musica.wav', 'equalizer', [0 -100 -100])*



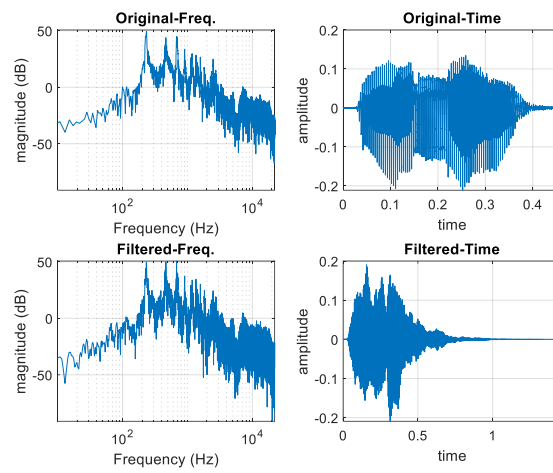
Comanda 2: `processaudio('musica.wav','equalizer',[-100 0 -100]);`



Comanda 3: `processaudio('musica.wav','equalizer',[-100 -100 0]);`



Comanda 4: `processaudio('musica.wav','equalizer',[10 -10 20])`



Comanda 4: `processaudio('hola_retallada.wav','reverb',[1 100]);`