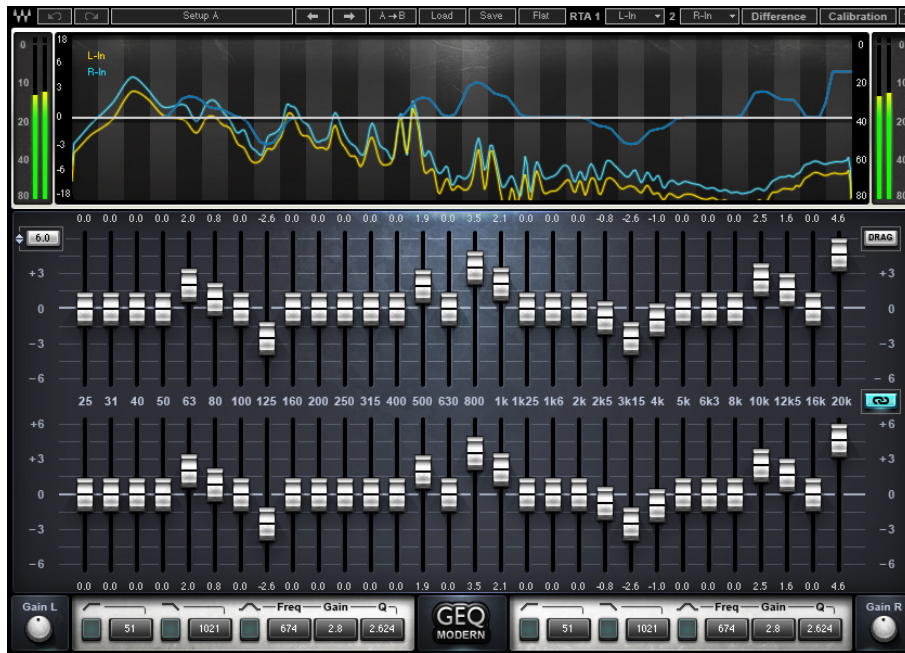


INGENIØRHØJSKOLEN AARHUS

DSB

E, IKT OG EP



Miniprojekt Lektion 10

Udarbejdet af:

Simon Thrane Hansen

Lars Hjerrild

Kasper Lauge Madsen

201500150

201409555

201409873

Underviser:

Lars G. Johansen

26. april 2016

Indhold

1	Indledning	2
2	Teori	3
2.0.1	Equalizer	3
2.0.2	Z-transformation	4
2.0.3	FIR-filtre	4
2.0.4	IIR-filtre	4
2.0.5	Hanning vindue	5
3	Audio Equalizer	6
3.1	Audio Equalizer	6
3.2	Matlab kode for de forskellige funktioner	6
3.2.1	FIR-Båndpass funktion	6
3.2.2	IIR-lowpass filter	7
3.2.3	FIR-weight funktion	8
3.3	Resultat	9
4	Resultat og Diskussion	10
5	Konklusion	11

1. Indledning

Denne opgave ophandler design og implementering af en Audio equalizer i Matlab. Der er i opgaveformuleringen blevet stillet følgende krav: "Equalizeren skal kunne justere niveauet på et indkommende lydssignal i fem forskellige frekvensbånd (LP, BP1, BP2, BP3, HP fordelt over det hørbare spektrum) med ± 12 dB. Der skal i opgaven indgå filtre af begge typer (FIR og IIR)."¹.

I opgaven er der blevet eksperimenteret med følgende filterbegreber:

- Filterorden
- Knækfrekvens

¹"Del 3- formål og indhold og opgaver"

2. Teori

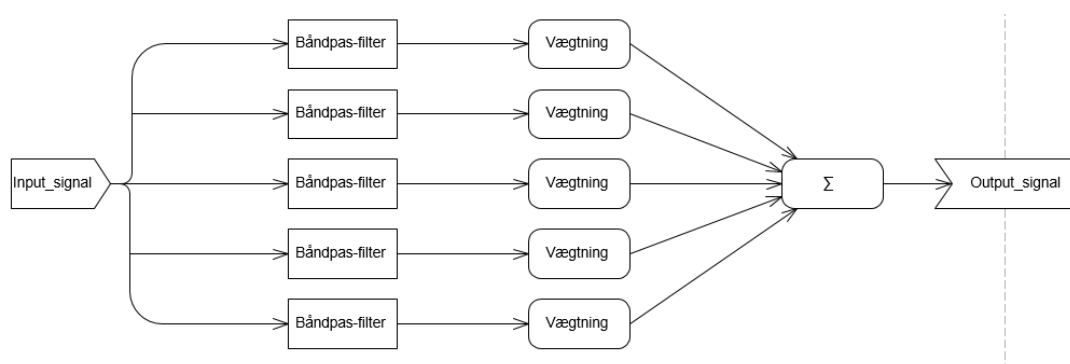
I denne opgave er der brugt en række forskellige digitale filtre og redskaber til at implementere en Audio equalizer. De forskellige filtre og redskaber, der er arbejdet med i opgaven er:

- FIR-filtre
- IIR-filtre
- Hanning funktionen
- Z-transformation

2.0.1 Equalizer

En Equalizer er en , den .

Equaliseren er blevet implementeret ved at lave 5 forskellige båndpas filtre, der opdeler Audio-signalet i 5 forskellige frekvens dele. Disse båndpas filtre er blevet designet ved brug af IIR- og FIR-filtre. Der er et blokdiagram for Equalizeren på figur 2.1.



Figur 2.1: Aktivitetsdiagram for Audio Equalizeren

Efter signalet er blevet opdelt i 5 forskellige frekvensbånd bliver hver enkelt af frekvens båndene vægtet på forskellige vis.

I det nedenstående afsnit følger en beskrivelse af de to forskellige filtre type. Man kan på nedenstående billede se den grundlæggende forskel på de to filtre.

2.0.2 Z-transformation

Z-transformationen er en tidsdiskret variant af Laplace Transformation, der transformere et tidsdiskret signal til frekvensdomænet.

Det matematiske udtryk for Z-transformation ses i ligning 2.1.

$$H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n} \quad (2.1)$$

2.0.3 FIR-filtre

FIR står for Finite Impulse Response, hvilket betyder, at der er et endeligt antal af impuls-svar. Dette kan man se, i ligning (2.2), der er differensligningen for et FIR filter.

$$y(n) = \sum_{k=0}^{M-1} b_k * x(n - k) \quad (2.2)$$

2.0.4 IIR-filtre

IIR står for Infinite Impulse Response, hvilket betyder, at det har et uendeligt antal output, da filteret benytter sig af feedback fra tidligere output. Hvis man kigger på ligning (2.3) kan man se den generelle differensligning for et IIR-filter.

$$y(n) = \sum_{k=0}^{M-1} b_k * x(n - k) + \sum_{l=1}^{N-1} a_l * y(n - l) \quad (2.3)$$

x(n): Inputsignal

y(n): Filterets output signal

b_k: Filterets feedforward koefficienter

a_l: Filterets feedback koefficienter

N-1: filterets orden

Fordelen ved et IIR-filter i forhold til et FIR-filter, at de kan bruges til at forstærke et signal, de benytter sig også af færre udregninger og dermed mindre hukommelse. Ulemperne ved IIR-filtre er, at de i modsætning til FIR-filtre godt kan være ustabile. Dette kan ses af ligning (2.4), der viser overføringsfunktionen for et IIR-filter i Z-domænet. Ligningen er fremkommet ved brug af Z-transformationen.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^N b(k) * z^{-k}}{1 - \sum_{k=1}^M a(k) * z^{-k}} \quad (2.4)$$

Som fremgår af ligning (2.4) indeholder overføringsfunktionen både poler og nulpunkter. Det gælder for IIR-filtre, at de er stabile, hvis alle deres poler har en magnitude, der er mindre end 1 (Polerne lægger i enhedscirklen).

2.0.5 Hanning vindue

Hanning vinduet er en typisk vinduesfunktion med den matematiske funktions foreskrift:

$$w(n) = 0.5 * (1 - (\cos \frac{2 * \pi * n}{N - 1})) \quad (2.5)$$

Vindues funktioner er en funktion (værdier), du ganger på dine frekvensbins. Man benytter vindues funktioner til at mindske mængden af lækage, men essensen af en vindues funktin er sådan set bare, at en funktion ganges på en anden funktion. I praksis inden for Digital Signal behandling anvendes stort set altid et Hanning vindue for at undgå lækage.

3. Audio Equalizer

3.1 Audio Equalizer

Audio Equalizeren er som beskrevet i afsnit 2 bestående af 5 båndpas-filtre. Der opdeler input-signalet i 5 frekvensbånd, der bliver vægtet forskelligt før de bliver sat sammen til et output-signal.

Der er blevet lavet et Matlab program, hvor der først bliver foretaget en Diskret Fourier Transformation på indgangssignalet for at finde ud af, hvilke frekvenser indgangssignalet indeholder. Efterfølgende bliver inputsignalet kørt igennem den fremstillede Audio Equalizer. Efter det er blevet kørt igennem Equalizeren bliver output-signalet også Diskret Fourier Transformeret, så der kan overskues, hvilke frekvenskarakteristika dette signal indeholder.

3.2 Matlab kode for de forskellige funktioner

3.2.1 FIR-Båndpass funktion

```
function [ output_signal, filter1] = FIR_bandpass_function(input_signal,fc1,fc2 )

len = length(input_signal);

fs = 48000;

N = 1000;

df = fs/N;

m1 = round(fc1/df); %bin nummer der passer bedst
m2 = round(fc2/df); %-||-

filter1 = [zeros(1,m1) ones(1,abs(m1-m2)) zeros(1,N/2-m1-abs(m1-m2))];
filter1 = [filter1 zeros(1,N/2-m1-abs(m1-m2)) ones(1,abs(m1-m2)) zeros(1,m1)];
```

```

filter_time = fftshift(real(ifft(filter1)));

x1 = filter(filter_time,1,input_signal);

hann = hanning(len)';

output_signal = x1.*hann;

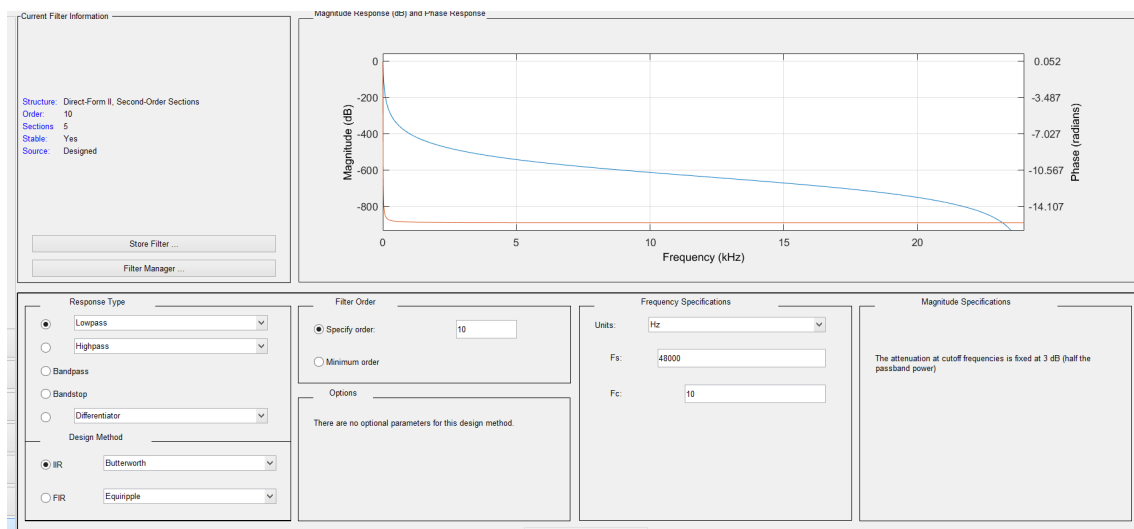
end

```

3.2.2 IIR-lowpass filter

Der blev designet et lavpas-filter i matlab ved brug af Matlabs Filter Design and Analysis Tool. Indstillingerne, der blev sat og det vedhæftede fase og amplitude response kan ses på billede

reffi:FDAtools.



Figur 3.1: Design af IIR lavpas filter i FDA tools i matlab

Filteret blev desginet med ønske i at skabe et IIR butterworth filter med en knækfrekvens på 10Hz og med en orden på 10. Da filteret var designet blev koefficienterne exporteret til et matlab-fil, der blev benyttet i funktionen IIR-lowpass filter.

Koden for IIR lavpas filter funktionen kan ses nedenuner:

```

function [ output_signal ] = IIR_Lowpass( input_signal )
%Load koefficienter for designet IIR-filter:
load LowpassIIR

```



```

b=[-1.99958882636935;-1.99881044943430;-1.99814879930370;
-1.99766835702012;-1.99741586621321];
a=[0.999590539491295;0.998812161889374;0.998150511191915;
0.997670068496728;0.997417577473500];

```

```

output_signal=filter(b,a,input_signal);

```

```

end

```

3.2.3 FIR-weight funktion

Equalizeren blev kørt med forskellige signaler, hvor der blev eksperimenteret med forskellige vægtning af de enkelte Båndpas-filtre samt frekvensbåndet de dækkede. Efter at havde eksperimenteret med de forskellige frekvensbånd blev det bestemt, at de enkelte frekvensbånd skulle dække følgende frekvensområder, der alle ligger i det hørbare frekvensområde:

- Bånd 1 -
- Bånd 2 -
- Bånd 3 -
- Bånd 4 -
- Bånd 5 -

```

%Give weights in dB

```

```

function [ output_signal ] = FIR_WEIGHT(input_signal,weight,weight2,weight3,weight4,weight5)

```

```

[LP] = IIR_Lowpass(input_signal);
[BP1, notused] = FIR_bandpass_function(input_signal, 8, 100);
[BP2, notused] = FIR_bandpass_function(input_signal, 80, 1000);
[BP3, notused] = FIR_bandpass_function(input_signal, 800, 10000);
[HP, notused] = FIR_bandpass_function(input_signal, 8000, 20000);

```

```

%weight = ln(10)*(x/(20*ln(10)));
%weight2 = ln(10)*(x/(20*ln(10)));
%weight3 = ln(10)*(x/(20*ln(10)));

```

```
%weight4 = ln(10)*(x/(20*ln(10)));  
%weight5 = ln(10)*(x/(20*ln(10)));  
  
LP = LP * weight;  
BP1 = BP1 * weight2;  
BP2 = BP2 * weight3;  
BP3 = BP3 * weight4;  
HP = HP * weight5;  
  
output_signal = LP + BP1 + BP2 + BP3 + HP;
```

3.3 Resultat

I dette afsnit fremstilles resultaterne af at køre Audio Equalizeren på en række forskellige signaler.

4. Resultat og Diskussion

Vores analyse af de forskellige signaler viste, hvilke frekvenser, der var dominerende i de enkelte signaler. Denne viden er vigtig at have, da den fortæller meget godt, hvad signalet indeholder.

Man kan f.eks. ved brug af analysen af symfoni-signalet (fig.??) se, at det indeholder stor aktivitet i frekvensområdet 100Hz-1000Hz, hvilket stemmer meget godt overens med, hvad man kunne forvente af frekvenser i et sådan signal.

Man kan, hvis man sammenligner de to signaler: Bas (fig.??) og musikboxen (fig.??). Se, at signalet fra musikboxen indeholder flere høje frekvenser end Bas-signalet, dette er helt forventelig. Man kan også se, at signalet fra vinglasset, der knipses på indeholder nogen meget præcise frekvenser modsat f.eks. signalet fra symfoni-orkesteret, der indeholder mange forskellige instrumenter med forskellige frekvenser.

Vores analysesystem har dermed givet os en bedre forståelse for, hvilke frekvenser, der er dominerende i de enkelte signaler.

Dette kan f.eks. bruges, i forbindelse med vindmøller, til at bestemme om, der er en bolt i motoren, der er slidt og skal skiftes. Hvis man har tilstrækkelig stor erfaring inde for dette område kan man nemlig sammenligne lyden fra et velfungerende system med et slidt eller fejlagtigt system.

I analysen er det valgt ikke at bruge vindues-funktioner, da det ikke gav mening i forhold til de signaler der er blevet arbejdet med. De anvendte signalers frekvensopløsning syntes at have været så god, at der ikke sås nogen nævneværdig forbedring af lækageproblemer, eller støjproblemer, ved brug af fx Hanning-vinduer. Dette er eksemplificeret ved figur: ??.

5. Konklusion

I denne opgave har vi arbejdet med at fremstille en Audio Equalizer i Matlab ved brug af forskellige filtre. Der er i opgaven blevet anvendt både IIR- og FIR-filtre.