

IELET2110 Signalbehandling

Signalbehandling

Øving 7

Name: Taheera Ahmed

Submission date: 19. mars 2023



Norwegian University of
Science and Technology

All koden til denne øvingen kan finnes i dette [Github-repositoryet](#)

Oppgave 1

I [Listing 1](#) kan man se definisjonen av vektorene som videre brukes for å plotte i denne delen av oppgaven.

```
1 b_k = [1 1 1];  
2 [H, w] = freqz(b_k);
```

Listing 1: Vektorene som brukes for å plotte frekvens, amplitude og faseresponsen

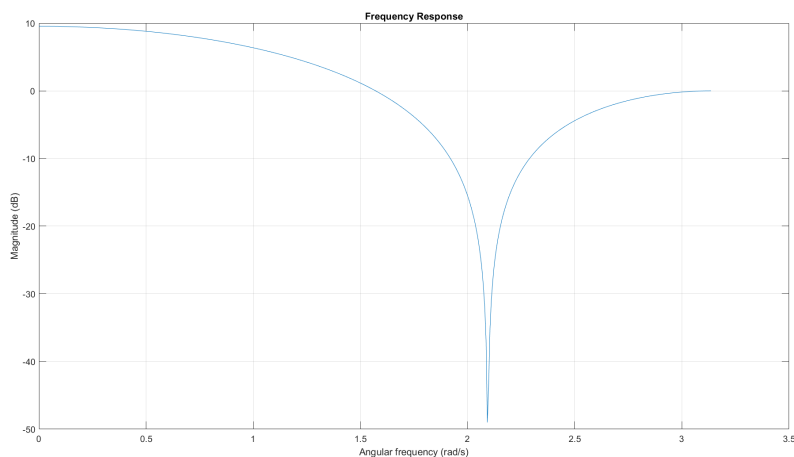
a

For å plotte frekvensresponsen til dette filteret har koden i [Listing 2](#) blitt brukt. Her kan man se at $20 \cdot \log_{10}(H)$ har blitt brukt for å plotte y-aksen. Dette har blitt brukt for å konvertere fra den lineære skalaen til desibel.

```
1 figure(1);  
2 plot(w, 20*log10(H));  
3 xlabel('Angular frequency (rad/s)');  
4 ylabel('Magnitude (dB)');  
5 title('Frequency Response');  
6 grid on;
```

Listing 2: Koden brukt for å plotte frekvensresponsen

Frekvensresponsen kan sees i [Figur 1](#), og den har blitt generert av [Listing 2](#).



Figur 1: Frekvensresponsen til filteret definert i [Listing 1](#)

b

For å plotte magnituderesponsen til filteret har koden i [Listing 3](#) blitt tatt i bruk. For å konvertere aksene fra et lineært til et logaritmisk forhold har denne koden blitt brukt `set(gca, 'XScale', 'log')`.

Her brukes også den innebygde MATLAB-funksjonen `abs()` for å plotte amplituderesponsen.

```
1 figure(2);
2 plot(w, abs(H));
3 xlabel('Frequency (\omega/\pi)');
4 ylabel('Magnitude');
5 title('Magnitude Response');
6 set(gca,'XScale','log')
7 set(gca,'YScale','log')
8 grid on;
```

Listing 3: Koden brukt for å plotte magnituderesponsen

Videre kan man se resultatet av plottet i [Figur 2a](#).

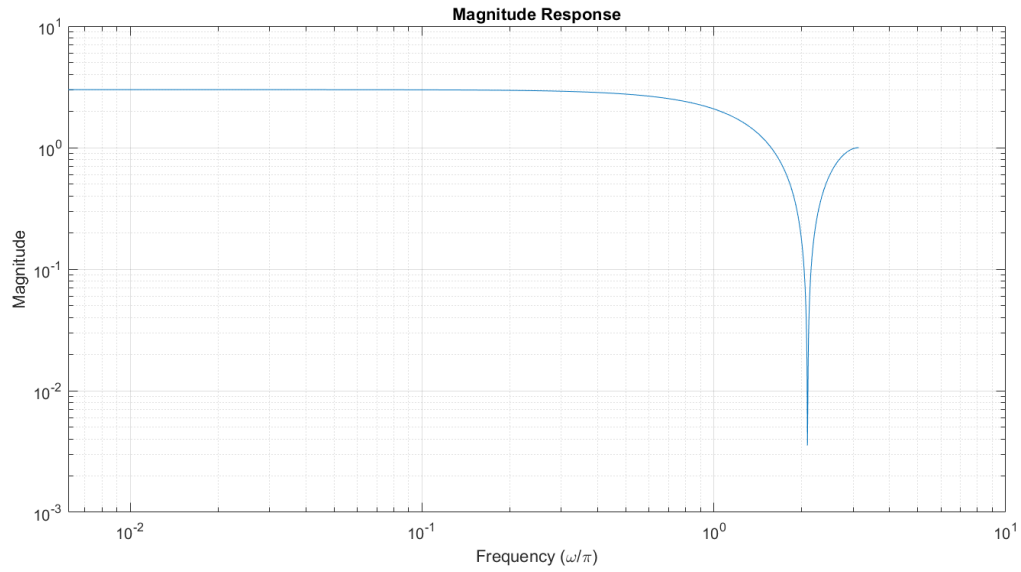
c

For å plotte faseresponsen har koden i [Listing 4](#) blitt brukt. Her brukes den innebygde MATLAB-funksjonen `angle()` for å finne faseresponsen til filteret.

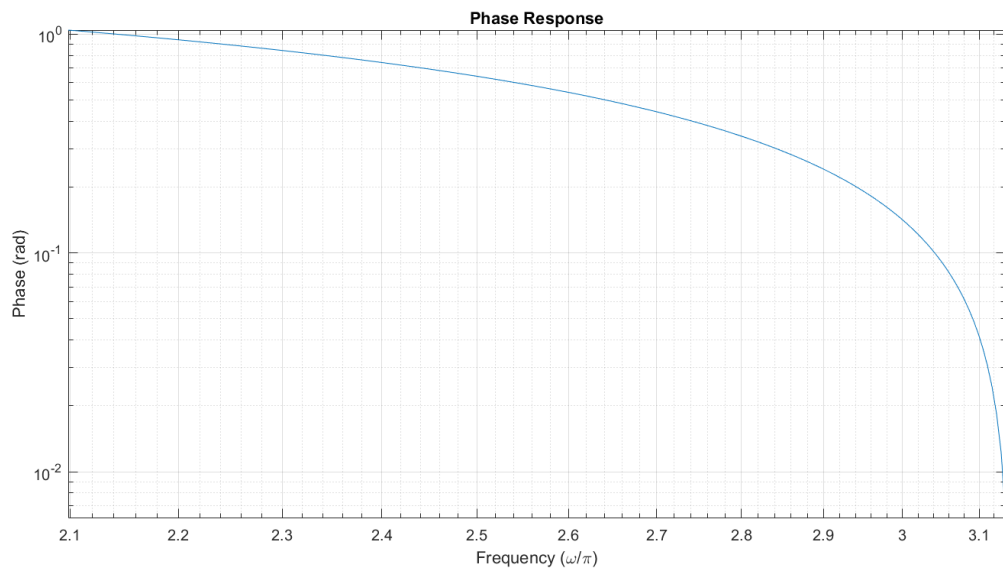
```
1 figure(3);
2 plot(w, angle(H));
3 xlabel('Frequency (\omega/\pi)');
4 ylabel('Phase (rad)');
5 title('Phase Response');
6 set(gca,'XScale','log')
7 set(gca,'YScale','log')
8 grid on;
```

Listing 4: Koden brukt for å plotte faseresponsen

Resultatet av plottet kan sees i [Figur 2b](#).



(a) Magnituderesponden til filteret definert i [Listing 1](#)



(b) Faseresponden til filteret definert i [Listing 1](#)

Figur 2: Magnituderesponden og faseresponden til filteret.

Oppgave 2

a

Koden til oppgave a kan finnes i [Listing 5](#).

```
1 [x,fs] = audioread('tale_pcm16000.wav');  
2 soundsc(x,fs,16);
```

Listing 5: Koden brukt for å lese inn fildata og lagre det som en vektor

b

Koden til oppgave b kan finnes i [Listing 6](#).

```
1 f1 = [0.02, 0.04, 0.08, 0.12, 0.15, 0.16, 0.15, 0.12, 0.08, 0.04, 0.02];  
2 y1 = filter(f1,1,x);  
3 soundsc(y1,fs,16);
```

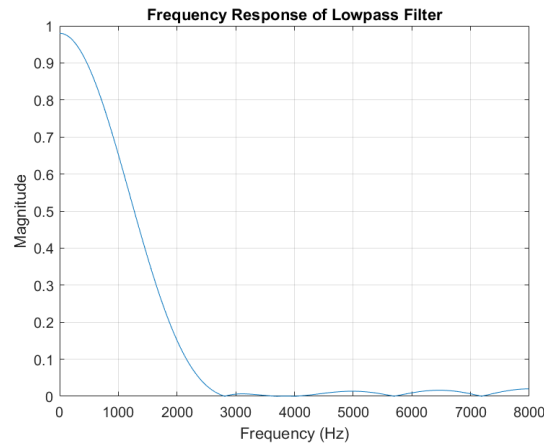
Listing 6: Koden brukt for å filtrere fildataen og vektoren x

Når man kjører koden blir den filtrerte lyden spilt av. Etter det har blitt filtert høres det litt ut som om mannen samt. mikrofonen hans er under vann. For å se nærmere på hvilke frekvenser som har blitt filtrert ut kan man se på frekvens responsen til filteret `f1` som blir gjort i [Listing 7](#)

```
1 fs = 16000; % Sampling frequency  
2 [H, f2] = freqz(f1, 1, 1024, fs);  
3  
4 figure(2);  
5 plot(f2, abs(H));  
6 xlabel('Frequency (Hz)');  
7 ylabel('Magnitude');  
8 title('Frequency Response of Lowpass Filter');  
9 grid on;
```

Listing 7: Koden brukt for å se på frekvensresponsen til filteret `f1`

[Listing 7](#) vil generere et plot som er vist i [Figur 3](#). Utifra denne figuren kan man si at filteret er et lavpass filter.



Figur 3: Frekvensresponsen til filteret definert i [Listing 6](#)

Man kan videre også se på frekvensspektrumet til lydene før og etter signalet har blitt filtrert. Koden til hvordan dette blir plottet kan sees i [Listing 8](#).

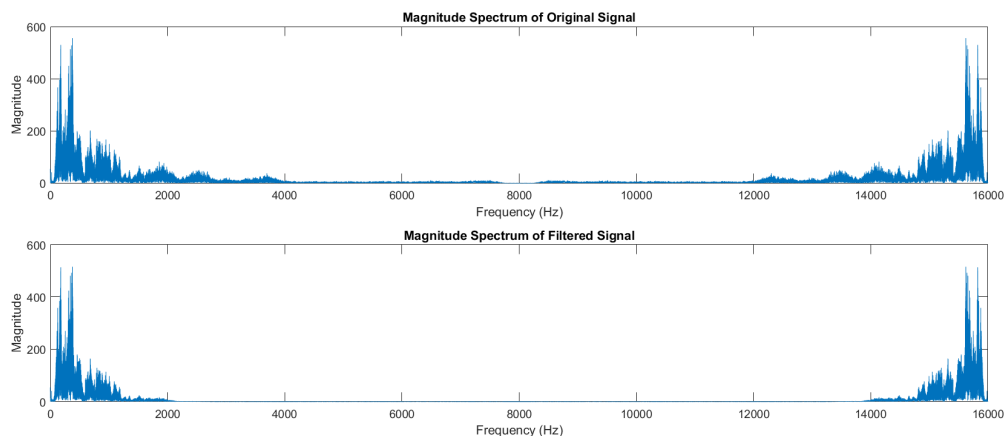
```

1  % Compute the Fourier Transform of the original signal
2  N = length(x);
3  X = fft(x);
4
5  % Compute the frequency axis
6  f2 = (0:N-1)*(fs/N);
7
8  % Compute the magnitude spectrum of the original signal
9  X_mag = abs(X);
10
11 % Repeat for the filtered signal
12 Y1 = fft(y1);
13 Y1_mag = abs(Y1);
14
15 % Plot the magnitude spectrum of the original signal and the filtered signal
16 figure(1);
17 subplot(2,1,1);
18 plot(f2, X_mag);
19 xlabel('Frequency (Hz)');
20 ylabel('Magnitude');
21 title('Magnitude Spectrum of Original Signal');
22 subplot(2,1,2);
23 plot(f2, Y1_mag);
24 xlabel('Frequency (Hz)');
25 ylabel('Magnitude');
26 title('Magnitude Spectrum of Filtered Signal');
```

Listing 8: Koden brukt for å se på frekvensspektrumet til den ufiltrerte lyden og den filtrerte lyden.

Koden i [Listing 8](#) generer figuren i [Figur 4](#). Utifra denne figuren kan det mer se ut som om det

er et båndpass-filter som filtrere signalet, ettersom det filtrerte signalet har en lavere magnitude på frekvensene mellom 2 kHz til 14 kHz. Dette synes jeg er litt rart. Hmm.



Figur 4: Frekvensspektrumet til den ufiltrerte og filtrerte lyden [Listing 8](#)

c

Nå skal det genereres et nytt filter `f2` hvor koeffesienten `b5` blir satt til `-0.84` også skal man se på hvordan dette påvirker lyden. For å kunne se nærmere på hvordan filteret har påvirket signalet kan man plote magnitudespektrumet til lydene. Dette kan gjøres ved å bruke `fft()`, som er MATLAB sin innebygde funksjon for å utgjøre Fast Fourier Transform. Videre kan man bruke `abs()` som vil ta absoluttverdien av denne vektoren som returneres fra `fft()`. Her har samme fremgangsmetode som i [Listing 8](#) blitt brukt.

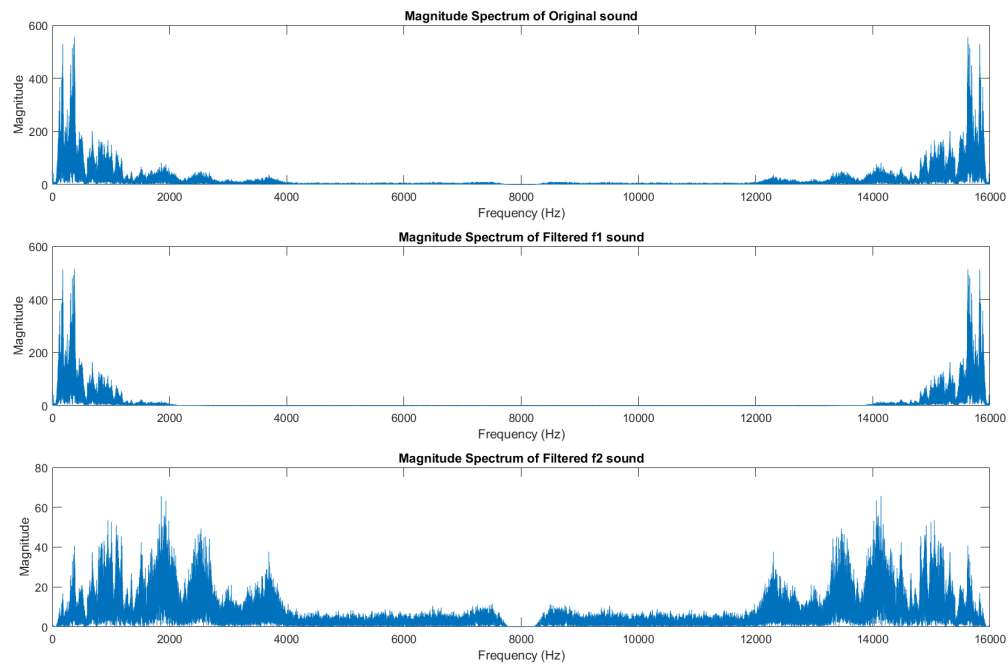
Plottet som genereres kan sees [Figur 5](#). Her kan man se at frekvenskomponentene mellom 2 kHz til 14 kHz har blitt forsterket i forhold til det originale signalet.

Videre kan man også se på frekvensresponsen til filtrene `f1` og `f2` for å se hva slags filter det er. Dette har blitt gjort på samme måte som i [Listing 7](#). Resultatet for frekvensresponsen til hvert filter kan sees i [Figur 6](#). Her kan man med andre ord se at `f2` er et høypass filter, i motsetning til `f1` som er et lavpass filter.

Når man spiller av lyden filtrert med filteret `f2` høres lyden mye klarere ut.

d

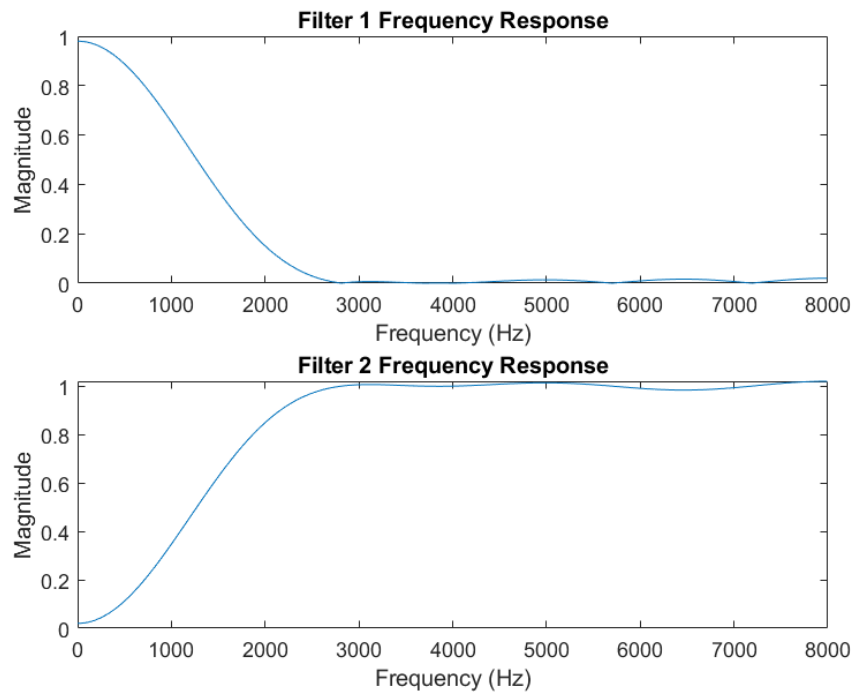
I denne oppgaven skal vi lage et nytt filter `filter3 = fir1(M,[300,3400]*2/fs);`. Dette skal plottes i både linær og logaritmisk skala. Resultatet kan sees i [Figur 7](#). Selv om de representerer lik informasjon vil en logaritmisk skala i dB være mer intuitiv for mennesker og en mer kompakt og klarere representasjon av magnituderesponsen. Poenget er at disse har forskjellige bruksområder.



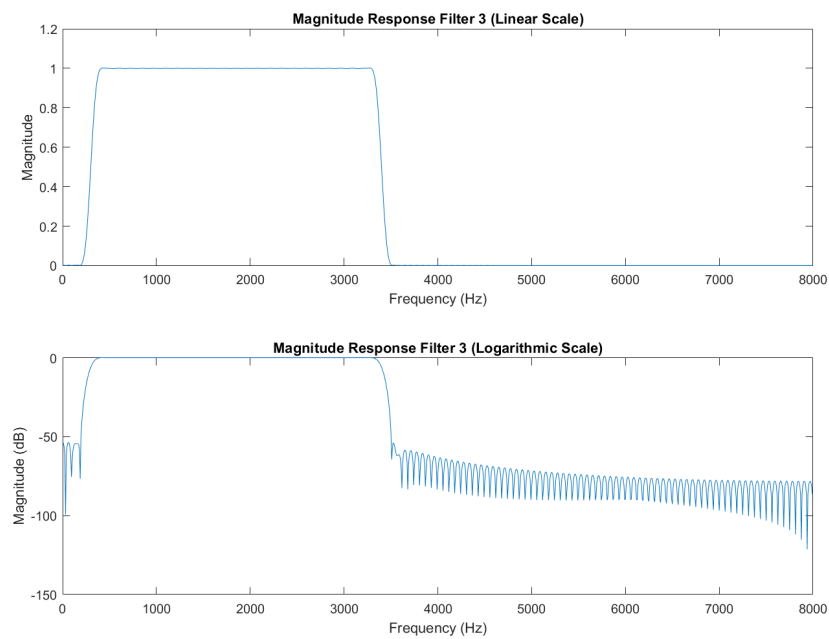
Figur 5: Frekvensspektrumene til alle de tre forskjellige signalene. Det ufiltrerte, signalet filtrert med filter 1 og signalet filtrert med filter 2.

e

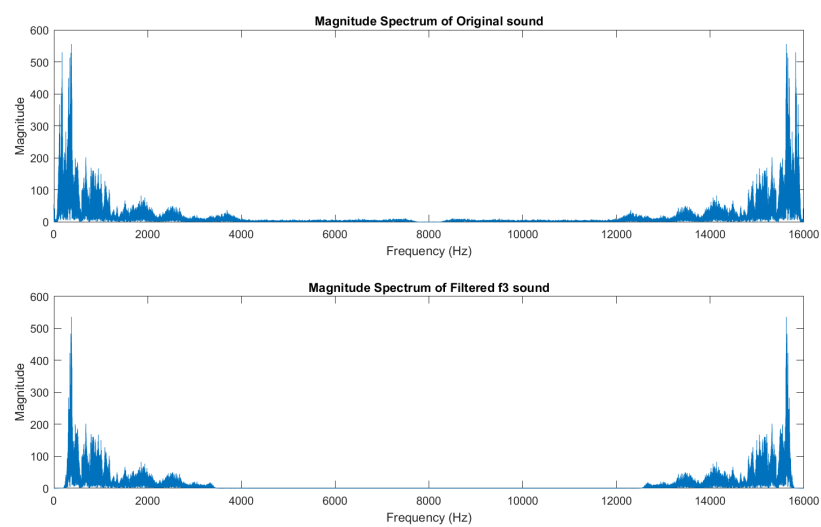
Det høres egentlig ganske likt ut som det originale signalet, kanskje litt mer dempet. Dersom man ser på frekvenskomponentene i signalet kanskje man kan se større forskjeller. Samme fremgangsmetode som ble brukt i oppgavene over er brukt for å lage disse frekvensspektrumet til signalet i [Figur 8](#).



Figur 6: Frekvensresponsen til filter 1 og filter 2.



Figur 7: Filteret representert med både linær og logaritmisk skala



Figur 8: Frekvenskomponentene av lyden før og etter det er filtrert med filter 3.