

# FYS2130

Prosjektoppgave våren 2015

Kandidatnummer: **139**

---

# Innhold

<b>1</b>	<b>Lydmåleren</b>	<b>3</b>
1.1	Lydmåling . . . . .	3
1.1.1	Desibel . . . . .	3
1.1.2	Vektete desibelmål . . . . .	3
1.2	Programvare . . . . .	4
1.2.1	Utstyr . . . . .	4
1.2.2	Lydopptak og fouriertransformasjon . . . . .	4
1.2.3	Beregning av relativt støynivå . . . . .	5
1.2.4	Vekting og kalibrering . . . . .	6
1.3	Testresultater og kalibrering av Programvare . . . . .	7
1.4	Løsningsmetodens ulemper . . . . .	9
1.5	Ulemper med støymåling generelt . . . . .	10
1.6	Programkode . . . . .	11
<b>2</b>	<b>Audiometeret</b>	<b>13</b>
2.1	Hørseltesting . . . . .	13
2.1.1	Om øret . . . . .	13
2.1.2	Ulike typer hørselstester . . . . .	13
2.1.3	Rentoneaudiometri . . . . .	14
2.2	Programvare . . . . .	16
2.2.1	Utstyr . . . . .	16
2.2.2	Avspilling av lyd med én frekvens . . . . .	16
2.2.3	Valg av lavest hørbare amplitude . . . . .	17
2.2.4	Normalperson og valg av nullpunktsamplitude . . . . .	18
2.2.5	Håndtering av data og forbedringsalgoritme . . . . .	19
2.2.6	Plotting og informasjonsfremvisning . . . . .	19
2.3	Testing og initialisering av programvare . . . . .	20
2.4	Resultater . . . . .	20
2.4.1	Hørseltest nummer én . . . . .	21
2.4.2	Hørseltest nummer to . . . . .	22
2.4.3	Hørseltest nummer tre . . . . .	24
2.4.4	Hørseltest nummer fire . . . . .	25
2.4.5	Om resultater . . . . .	26
2.5	Ulemper ved løsningsmetoden . . . . .	26
2.6	Ulemper ved audiometriske målinger generelt . . . . .	27
2.7	Programkode . . . . .	29

# 1 Lydmåleren

## 1.1 Lydmåling

Man møter unektelig ulike støynivåer i hverdagen. Et stille kontor, et trafikert lyskryss eller et passasjerfly som tar av er alle opphav til ulike grader av støy, men hvordan kan vi kvantifisere forskjellen?

En lydmålers hensikt er å kvantifisere støynivåer. Ved å gjøre et opptak av lyden kan lydmåleren analysere den tilhørende amplituden til hver frekvens. Dette kan brukes til å produsere et bilde av opptakets lydintensitet, som gjerne oppgis i desibel.

### 1.1.1 Desibel

Desibel, eller én tidels bel, er et logaritmisk mål på relativ intensitet som er hyppig brukt i områder som akustikk og elektronikk. Konversjonen mellom en intensitet  $I$  og desibelmålet  $I_{\text{dB}}$  er definert relativt til en referanseintensitet  $I_0$  på følgende måte

$$I_{\text{dB}} = 10 \lg \left( \frac{I}{I_0} \right).$$

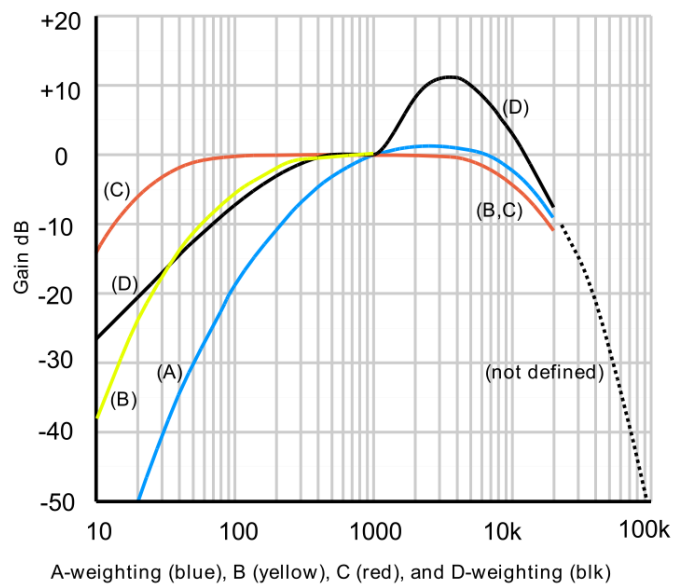
Det at desibel er definert i forhold til en referanseintensitet gjør at det fremstår som lite meningsfylt å oppgi verdier i desibel uten også å oppgi referansen som er brukt. For å kunne bruke desibel som en absolutt skala har det av den grunn blitt definert forskjellige referanseintensiteter. Blant disse er det trolig  $\text{dB}(\text{SPL})$  som er mest brukt. SPL står for *sound pressure level* og har en definert referanseverdi lik intensiteten til lyd med frekvens  $1\text{kHz}$  og lydtrykk  $p_{\text{rms}} = 20\mu\text{Pa}$ <sup>1</sup>. Med lydtrykket mener man her trykkbølgens såkalte *root mean square* verdi, definert som kvadratroten av gjennomsnittet av kvadratet av trykkets ulike verdier over en tidsperiode. Dette tilsvarer omtrent den lavest hørbare intensiteten til et lydsignal på  $1\text{kHz}$ .

### 1.1.2 Vektete desibelmål

Ettersom menneskers opplevelse av lyd varierer med frekvensen oppstår det iblant et behov for å vekte desibelverdiene slik at de bedre reflekterer lydintensiteten slik den oppleves. Her finnes det en rekke standardiserte vektfunksjoner. Blant disse er den såkalte *A*-vekting mest brukt.

---

<sup>1</sup>Kilde: Læreboken side 189.



Figur 1: Ulike vektfunksjoner som skal gjøre at desibelmålet bedre reflekterer menneskers oppfattelse av en lyd. Kilde: Wikipedia <http://en.wikipedia.org/wiki/A-weighting> 10. mai 2015.

A-vekting er imidlertid bare gyldig for lave lydintensiteter og rene toner<sup>2</sup>. Når desibelverdier som bruker referanseintensitetsstandarden *SPL* vektet med A-vekting, oppgis verdiene i enheten  $dB(A)$ . Denne enheten er blant annet mye brukt innen lydmåling.

## 1.2 Programvare

### 1.2.1 Utstyr

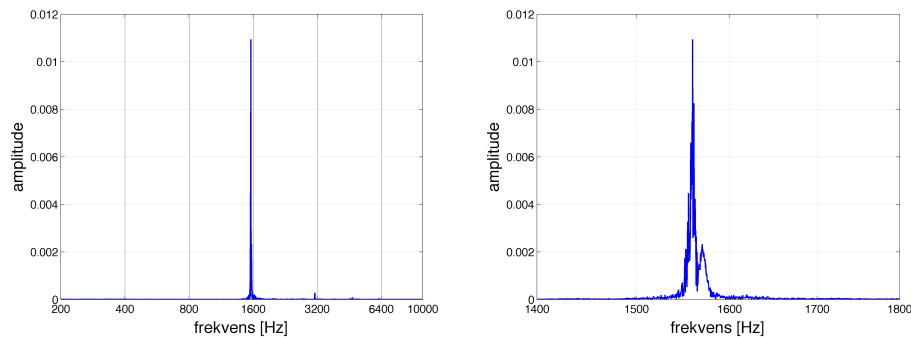
- Macbook Pro OSX Version 10.9.5 (MATLAB versjon R2013a, innebygd mikrofon)
- iPhone4s: (app: *Signal Gen*)
- Komersiell lydmåler RION NL-05

### 1.2.2 Lydopptak og fouriertransformasjon

Når lyd skal tas opp kan Nyquist–Shannons samplingsteorem benyttes til å velge samplingsfrekvens. Vi tar utgangspunkt i den menneskelige hørsel,

<sup>2</sup>Kilde: Wikipedia <http://en.wikipedia.org/wiki/A-weighting> 10. mai 2015.

dobler den maksimale hørbare frekvensen  $20kHz$ , og legger til litt mer for å være på den sikre siden. Da får vi, som tilfeldigvis også er konvensjonen brukt i blant annet CDer, en samplingsfrekvens lik  $F_s = 44100$ . For å få nok data lar vi opptaket vare i  $T = 3$  sekunder. Videre bruker vi 16bit-kvantisering av hver lydprøve og tar opp fra begge kanaler (høyre og venstre). Ved å benytte MATLABs implementering av Cooley-Tukey algoritmen for rask fouriertransformasjon (FFT), transformerer vi deretter lydets tidsbilde om til et frekvensbilde<sup>3</sup>. Vi fjerner så siste halvdel av den fouriertransformerte for å unngå folding. Deretter oppretter vi en tilhørende tallsekvens som skal representere frekvenser bestående av jevnt fordelte verdier fra 0 til  $F_s/2$ , og beregner lengden av hvert element i den fouriertransformerte.



Figur 2: Frekvensbildet av plystring i 3 sekunder ved  $1560Hz$ . Figuren til venstre viser frekvensbildet for hele det hørbare spekteret. I figuren til høyre viser det samme frekvensbildet, men nå bare i frekvensområdet  $1400Hz - 1800Hz$ , der plystelyden synes tydelig.

### 1.2.3 Beregning av relativt støynivå

Hvordan skal man gå fra et lydopptaks frekvensbilde til et tall på lydets støynivå? Den mest nærliggende løsningen på problemet er å integrere lyd-signalets amplituder over alle hørbare frekvenser. Diskretisering av en slik integrasjon gir

$$\text{støy} \stackrel{?}{=} \Delta f \sum_{f\text{hørbar}} A(f) = \frac{1}{T} \sum_{f\text{hørbar}} A(f) = F_s \frac{1}{N} \sum_{f\text{hørbar}} A(f),$$

der  $\Delta f$  er frekvensbildets oppløsning,  $A(f)$  er amplituden til frekvensen  $f$ ,  $T$  er lydopptakets lengde,  $F_s$  er samplingsfrekvensen og  $N$  er antallet datapunkter. Problemet med dette målet på støy er at et lydsignal med én frekvens på

<sup>3</sup>Før fouriertransformasjonen forkorter vi egentlig også lydsignalet slik at antall elementer kan skrives på formen  $2^n$ , som er mer egnet for en datamaskin.

1000 Hz og amplitude lik 1 vil gi samme støynivå som ti ulike frekvenser med amplitude 0.1 – det ønsker vi ikke. En måte å sno seg utenom problemet er heller å integrere over kvadratet av amplitudene. Da vil skarpe signaler med store amplituder vektet slik at de får høyere utslag på støymålet. Vi definerer altså støy ved

$$\text{støy} = \Delta f \sum_{f \text{ hørbar}} A^2(f) \propto \frac{1}{N} \sum_n A^2(f_n).$$

For enkelhets skyld lar vi summasjonsindeksen gå fra den nest første verdien<sup>4</sup> til  $N/2$  som svarer til omlag  $22k Hz$ .

Ved å bruke at intensiteten til en bølge er proporsjonal med kvadratet av amplituden, ser vi at kvantifiseringen av støy funnet ovenfor er et intensitetsmål. Dersom vi bestemmer oss for en referanseintensitet  $I_0$ , kan vi oppgi støyintensiteten i desibelskalaen

$$\begin{aligned} \text{Relativ støyintensitet} &= (10\text{dB}) \lg \left( \frac{I}{I_0} \right) = (10\text{dB}) \lg(I) - (10\text{dB}) \lg(I_0) \\ &= (10\text{dB}) \lg(I) - C_{\text{ref dB}}, \end{aligned}$$

der  $I$  er integralet over amplitudene kvadrert.

#### 1.2.4 Vekting og kalibrering

Den menneskelige opplevelse av lydintensitet avhenger av lydets frekvens. Et lydssignal med amplitude 1 i randen av frekvenser som kan oppfattes av mennesker vil oppfattes som lavere enn den ville dersom den hadde frekvens midt i frekvensspekteret, f.eks  $1500 Hz$ . Dette kan tas med i betraktningen ved å multiplisere amplitudene i frekvensbildet med en vektfunksjon. Vektfunksjonen bør representere opplevd lydstyrke som funksjon av frekvens når amplituden holdes konstant. Ved å undersøke en gruppe mennesker med ”normal” hørsel kunne vi konstruere en kurve som representerer lavest hørbar amplitude som funksjon av frekvens. Enkle manipulasjoner av denne kurven kunne gi oss en fornuftig vektfunksjon. Vi kan imidlertid også benytte oss av den såkalte A-vektingen omtalt i seksjon 1.1.2. På Wikipedia<sup>5</sup> oppgis et matematisk uttrykk for A-vekting slik

$$R_A(f) = \frac{12200^2 f^4}{(f^2 + 20.6^2) \sqrt{(f^2 + 107.7^2)(f^2 + 737.9^2)(f^2 + 12200^2)}},$$

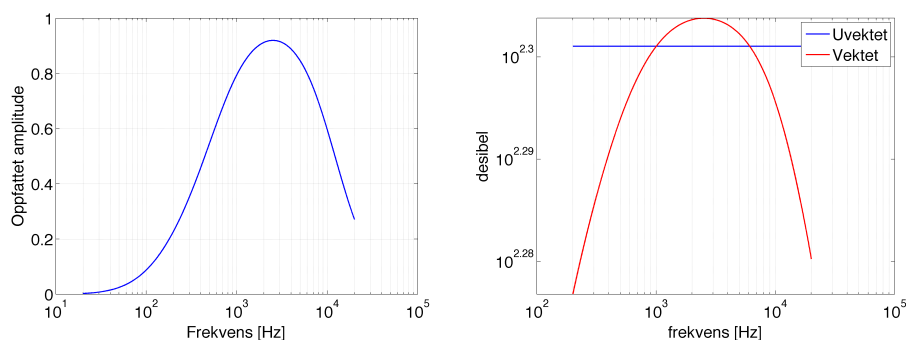
<sup>4</sup>Den første verdien representerer amplituden til frekvensen null. Ettersom trykkendringer med frekvens null slettes ikke er trykkendringer kan den første verdien ikke tolkes som lyd.

<sup>5</sup>Wikipedia, *A-weighting*: <http://en.wikipedia.org/wiki/A-weighting>

med tilleggsinformasjon om at vektfunksjonen skal virke på spekteret av amplituder og at bruken vil forskyve nullpunktet til lydstyrken to desibel nedover. Uttrykket for vektet relativ støyintensitet,  $I_{dB(A)}$ , blir altså

$$\begin{aligned} I_{dB(A)} &= (10\text{dB}) \lg(I) - C_{\text{ref dB}} + 2 \\ &= (10\text{dB}) \lg\left(\frac{1}{N} \sum_n A_{\text{vektet}}^2(f_n)\right) - C_{\text{ref dB}} + 2 \\ &= (10\text{dB}) \lg\left(\frac{1}{N} \sum_n [A(f_n)R_A(f_n)]^2\right) - C_{\text{ref dB}} + 2, \end{aligned}$$

der vi i prinsippet står fritt til å velge verdien  $C_{\text{ref dB}}$ . Det kan imidlertid fremstå som hensiktsmessig å velge  $C_{\text{ref dB}}$  slik at verdiene er i overensstemmelse med enheten  $dB(A)$ , som vi har sett baserer seg på den standardiserte referanseamplituden  $SPL$ . Enkleste måte å gjøre dette på er ved å sammenlikne verdier med en allerede fungerende lydmåler.



Figur 3: Til venstre er vektfunksjonen  $R_A(f)$  plottet over alle hørbare frekvenser. Til høyre er den vektete- og uvektede desibelintensiteten plottet sammen for en vilkårlig referansedesibel  $C_{\text{ref dB}} = -200\text{dB}$ .

### 1.3 Testresultater og kalibrering av Programvare

Testene som skal gi referanseverdien  $C_{\text{ref dB}}$  bør utføres omhyggelig. Hvordan lyden reiser fra en høyttaler til en mikrofon avhenger av orientering og rommets geometri. I tillegg vil det være ønskelig å sammenlikne verdier for rene frekvenser, noe som betyr at bakgrunnsstøy bør minimeres. Med dette i tankene utføres målingene i det lyddempede rommet i fysikkbyggets kjeller. Her utføres testene ved å plassere lydkilde<sup>6</sup> som spiller av lyd med frekvens  $1\text{kHz}$

<sup>6</sup>iPhone ved hjelp av appen Tone Gen

omlag 50cm fra lydmålerne, den kommersielle kalibrerte og den selvutviklede ukalibrerte. Alt plasseres midt i rommet og de to målerne plasseres med en avstand på omlag 20cm. Etter litt justeringer velges desibelreferansen lik  $C_{\text{ref dB}} = -157\text{dB}$ , da blir måleverdiene som følger

Tabell 1: Måleverdier i lyddempet rom for lydsignal på  $1\text{kHz}$ .

Kalibrert vektet dB(A)	Selvutviklet vektet dB(A)	Selvutviklet uvektet dB(SPL)
54.0	54.83	54.83
55.0	54.06	54.06
46.0	46.83	46.83

Frekvensen  $1\text{kHz}$  ble valgt ved å studere skjæringspunktet til den vektete og den uvektede lydstyrken til høyre i figur 3. På den måten kan vi samtidig med kalibreringen undersøke om vektfunksjonen er blitt riktig implementert. Måleverdiene for vektet og uvektet lydstyrke er identiske, noe som betyr at programvaren vektet intensitetene riktig. Etter et par forsøk viste altså  $C_{\text{ref dB}} = -157\text{dB}$  seg å være en kalibrering som gir god nok overensstemmelse med  $\text{dB}(A)$ -målet. Usikkerheten kan anslås til å være omlag  $\pm 3\text{dB}$ . Resultatene forteller med andre ord at den selvutviklede lydmåleren nå kan anses for å være kalibrert.

Den nylig kalibrerte lydmåleren ble siden testet i to situasjoner med følgende resultater:

Tabell 2: Måleverdier for lydmåler på tom lesesal med avskrudd vifte.

Vektet dB(A)	37.40	37.34
Uvektet dB(SPL)	40.01	40.10

Tabell 3: Måleverdier for lydmåler på lesesal i brukstiden med påskrudd vifte.

Vektet dB(A)	40.70	45.28
Uvektet dB(SPL)	40.02	62.48



Måleresultatene fra lesesalen i tabell 2 og 3 har en gjennomsnittlig differanse på cirka  $5dB$ . Trolig er det den påskrudde viften som er den utslagsgivende effekten. En såpass liten differanse i opplevd støynivå kan bære preg av at lydmåleren er defekt. Lyden fra viften er påfallende høy og burde trolig utgjort en større forskjell. Det er også verdt å merke seg den store forskjellen i måleverdiene fra tabell 3. Differansen i uvektet støyintensitet er  $22.46dB$ , mens differansen i vektet støyintensitet er bare  $5.26dB$ . Denne effekten har trolig sin forklaring i en tabbe gjort under målingene. Dersom testpersonen puster dypt imens lydopptaket pågår vil pustelyden kunne avleses som en drastisk amplitudeøkning i lavfrekvensområdet. Et forsøk av denne typen ble utført og ga uvektet støyintensitet lik  $74.10 dB(SPL)$  mens den vektete støyintensiteten viste seg å være bare  $48.71 dB(A)$ . Altså virker "tung pust" som en forklaringsmodell for differansen i måleverdier i tabell 3.

Ettersom alle målingene hittil er gjort i relativt stille omgivelser ble en siste måling utført sent på fysikkforeningens "Vårfest" fredag 8. mai. Lydmåleren ble plassert midt i rommet i rimelig avstand fra både ropende mennesker og høyttalere; verdiene ble som følger:

Tabell 4: Måleverdier for lydmåler på fysikkforeningens Vårfest natt til lørdag 9.mai (kl 01:40)

Vektet dB(A)	90.01	90.32
Uvektet dB(SPL)	90.02	90.32

Målingene ble gjort da lydnivået på festen var på et høydepunkt, noe resultatene klart formidler. Ifølge arbeidstilsynets faktasider for støy og helse<sup>7</sup> regnes alt over  $80dB(A)$  for å være skadelig støynivå – det er  $10dB$  lavere enn de målte verdiene. Pussig nok viser måleresultatene nesten lik verdi for vektet og uvektet støynivå. Det bærer preg av at lydets frekvensspekter var fordelt rundt  $1kHz$  eller  $6kHz$ , som er overraskende høye frekvenser.

## 1.4 Løsningsmetodens ulemper

Prosessen forklart over viser hvordan man lett kan lage en lydmåler på egen datamaskin. Løsningsmetoden er rask, men ikke veldig pålitelig. Under følger en punktvis diskusjon av ulemper ved metoden

- Amplituder til frekvenser som ikke er i det hørbare spekteret tas med i beregningen.

<sup>7</sup>Arbeidstilsynet, *Faktasider for støy og helse*: <http://www.arbeidstilsynet.no/fakta.html?tid=78245>

*I diskretisering av integrasjonen av kvadrerte amplituder forenklet vi algoritmen ved å la summasjonsindeksen gå fra den nest første verdien til den siste,  $N/2$ . Da tar vi med frekvenser som ikke kan oppfattes av det menneskelige øre, noe som til en viss grad vil forkrøple troverdigheten til de uvektede støyintensitetene. Den vektete støyintensiteten er imidlertid fremdeles til å stole på ettersom bidraget fra lave frekvenser er forsvinnende etter vektingen.*

- Mikrofonens plassering på datamaskinen gjør at orientering er utslagsgivende for måleresultater.

*Hvilken retning lyden kommer fra når den treffer mikrofonen vil utgjøre store forskjeller i måleresultatene. Lydbølger vil reflekteres i datamaskinens skjerm og skape et kaotisk nærfelt for lyden rundt mikrofonen. Dette kan løses ved å benytte en ekstern mikrofon. Da må imidlertid kalibreringen repeteres.*

## 1.5 Ulemper med støymåling generelt

Utviklingen av programvaren har reist endel spørsmål rundt støymåling generelt. Det synes å være problemer som ikke er unike for løsningsmetoden omtalt i denne prosjektoppgaven. Under følger en punktvis diskusjon av generelle ulemper ved støymåling

- Støymåling gir dårlig representasjon av menneskers oppfattelse av lyd *Selv når støynivåer måles ved å vekte amplituder i frekvensspekteret vil ikke måleverdiene reflektere menneskers oppfattelse av lyden spesielt godt. Blant de faktorene som er oversett i støymåling er først og fremst lydets varighet. Kortvarige lydsignaler med høy intensitet, såkalte "impulslyder", vil fremstå som mye høyere lyd for en menneskelig lytter enn for en lydmåler.*
- Måleverdier vil i svært stor grad avhengige av mikrofonens posisjon i rommet. *Dersom støymåling skal benyttes som en kvantifisering av oppfattet støy gir måleresultater kun mening i det ene punktet når rommet er i nøyaktig samme orientering som det var da testene ble utført. Mikrofonens avstand fra lydreflekterende objekter vil ha meget stor innvirkning på måleresultater. Det gjør resultater av støymåling vanskelige å tolke.*
- Desibel er forvirrende og krever en mer konsekvent nullpunktsdefinisjon.

*Det relative lydintensitetsmålet Desibel bærer mer preg av å være et kulturminne fra regnestavens tid enn et nyttig intensitetsmål. Mens den logaritmiske skalaen gjør aritmetisk argumentasjon mindre intuitiv, gjør skalaens mangel på en konsekvent nullpunktsdefinisjon det vanskelig å overføre målet til fysiske enheter. Dette gjør at desibel, i motsetning til de fleste andre kvantitative mål, står alene og vanskelig lar seg sammenlikne med noe annet enn andre desibelverdier med samme nullpunktsdefinisjon.*

## 1.6 Programkode

- Hovedprogrammet

```
%Desibelreferanse (kalibreringsfaktor)
referanse_dB = -157;

%Fouriertransformasjonen
N = 2^17;
[F,f] = fourier();

%Relativ lydintensitet uten vekting:
absF = abs(F);
absF2 = absF.*absF; %Intensitetstetthet
totI_iv = sum(absF2(2:N/2))/N; %integrasjonen
rel_dB_unWeighed = 10*log10(totI_iv) - referanse_dB;
%Skriv verdier til skjerm med to siffer
fprintf('Intensitet: %.2f dB(SPL) \n',rel_dB_unWeighed);

%Relativ lydintensitet med vekting:
absF = abs(F).*AWeighting(f)';
absF2 = absF.*absF; %Intensitetstetthet
totI_iv = sum(absF2(2:N/2))/N; %integrasjonen
%korreksjon: 2dB (fra vekting)
rel_dB_weighed = 10*log10(totI_iv) - referanse_dB + 2.0;
%Skriv verdier til skjerm med to siffer
fprintf('Intensitet: %.2f dB(A) \n',rel_dB_weighed);
```

- Funksjonen som utfører fouriertransformasjonen

```
function [transform,frequencies] = fourier()
    Fs = 44100; %samplingsfrekvens
    T = 3.0; %Lydopptakets lengde [sek]

    %Lydopptaket
    recObj = audiorecorder(Fs, 16, 2); %bits:16, No
    channels:2
```

```

recordblocking(recObj, T); %Utfoor opptak
minLyd = getaudiodata(recObj);
y = minLyd(:,2); % Plukker ut hooyre kanal
N = 2^17; % 131072 datapunkter i stedet for
3*44100=132300
y = y(1:N); %Initialisering
g = fft(y)/N; %fast fourier transform
freqs = Fs/2*linspace(0,1,N/2); %Tilhoorende
frekvensverdier
transform = g(1:1:0.5*N); %send ut
frequencies = freqs; %send ut
plot(frequencies,abs(transform)); %Plot frekvensbilde
end

```

- Vektfunksjonen

```

function weight = AWeighting( f )
f2 = f.*f;
f4 = f2.*f2;
numerator = 12200*12200 * f4;
denominator = (f2+20.6*20.6).*sqrt((f2+107.7*107.7)
.*(f2+737.9*737.9)).*(f2+12200*12200);
weight = numerator./denominator;
end

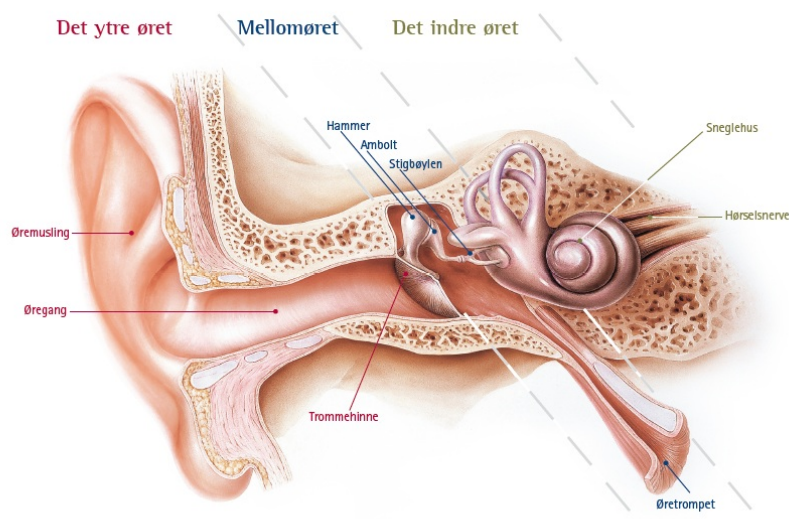
```

## 2 Audiometeret

### 2.1 Hørseltesting

#### 2.1.1 Om øret

Veien fra oscillasjoner i lufttrykket til menneskelig persepsjon av lyd begynner når trykkbølger treffer øret og beveger seg innover i øregangen. Inne i øret stopper trykkbølgen i møte med en hinne, kalt *trommehinnen*. Dette får trommehinnen til å vibrere. Vibrasjonen får en rekke med små ben, kalt *hammer*, *ambolt* og *stigbøyle*, til å bevege seg. Det innerste benet, stigbøylen, banker da på utsiden av membranen til et sneglehusformet instrument, kalt *sneglehuset*, og oomgjør lyden til trykkbølger i den indre væsken.



Figur 4: kilde: <http://trimo.gratisnettside.no/?mid=3271>

Trykkbølgene i vesken inne i sneglehuset forplanter seg innover i forskjellige kanaler og får flimmerhår langs veggene til å bevege seg. Bevegelsen av disse flimmerhårene er koblet til nerver, som formidler de signalene hjernen oppfatter som lyd.

#### 2.1.2 Ulike typer hørselstester

Det eksisterer mange ulike typer hørselstester. Noen er rent objektive, andre måler forståelse av tale og persepsjon av ulike frekvenser. Under følger en

kort forklaring av enkelte slike tester<sup>8</sup>

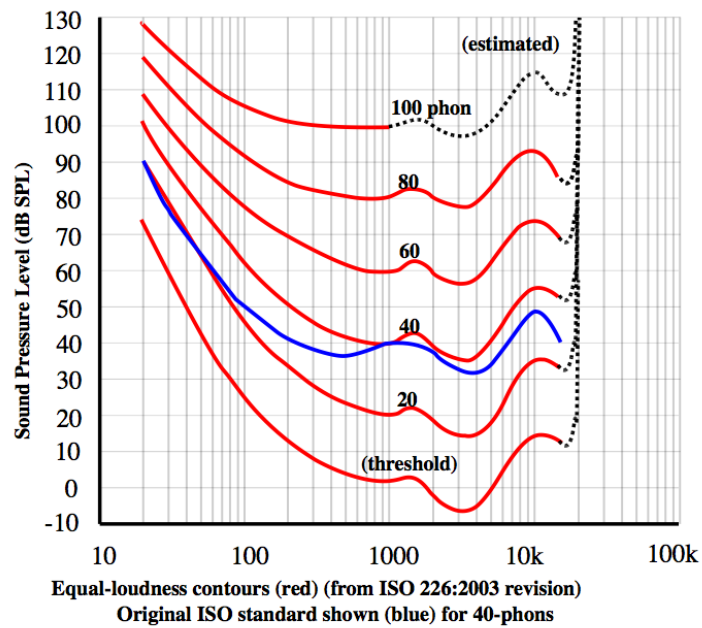
- OAE - Otoakustiske emisjoner  
*Ved å plassere en liten mikrofon inne i øregangen kan det tilbakevendende ekkoet registreres når en klikkelyd sendes inn i øret. Fra mikrofonens opptak kan den lyden som ble utsendt fra de ytre hårcellene i sneglehuset bli analysert. Dersom dette signalet er normalt er det sannsynlig at også hørselen er normal.*
- ABR - Auditory Brainstem Response  
*Øret til en sovende testperson stimuleres med klikkelyder. Fire elektroder festet på personens hode registrerer deretter hjernestammens svar på klikkelydene. Testen kan i enkelte tilfeller si noe om formen for hørselstap og hørselen i frekvensområdet 1-4kHz.*
- Hint - Hearing in noise test  
*Testpersonen blir bedt om å gjenta setninger som spilles i et lydregulert rom. Prosessen gjentas med økende bakgrunnsstøy og danner derfor et bilde av personens evne til å skille tale fra bakgrunnsstøy.*

### 2.1.3 Rentoneaudiometri

Instrumentet vi skal utvikle er av den typen man bruker i den kanskje mest populære formen for hørselstesting, kalt *rentoneaudiometri*. I testen settes personen i et lyddempet rom med hodetelefoner, som spiller av rene frekvenser med variabel intensitet. Ved å registrere amplituder i grenseområdet for hørbarhet av lydsignalene kan det dannes et bilde over hvordan minste hørbare lydintensitet endrer seg som funksjon av frekvens.

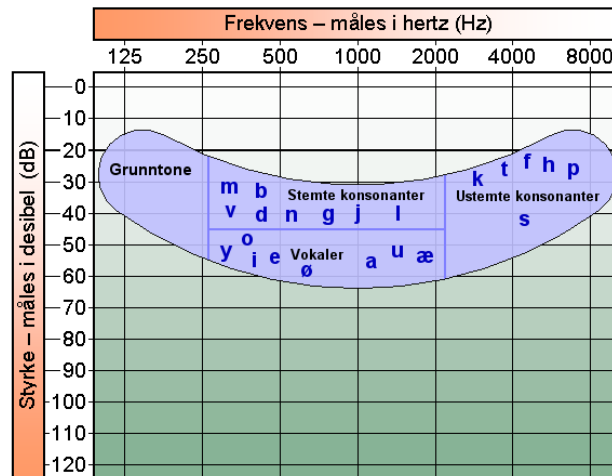
---

<sup>8</sup>Informasjon om ulike former for hørselstester er hentet fra nettsiden <http://www.sansetap.no/voksne-horsel/om/nedsatt-horsel/horselsmalinger-og-tester/>



Figur 5: Konturer som viser lik-opplevd-lydstyrke for en gjennomsnittlig ung person uten alvorlig hørselssvekkelse. Kurvene er i den internasjonale standarden ISO 226:2003. Kilde: [http://en.wikipedia.org/wiki/Equal-loudness\\_contour](http://en.wikipedia.org/wiki/Equal-loudness_contour) den 10. mai 2015.

Den resulterende kurven kan deretter sammenliknes med kurven til en "normal" person ved å plote testpersonens avvik fra normalkurven i det som utgjør et såkalt *audiogram*. Audiogrammet analyseres deretter i forhold til språklydenes plassering i audiogrammet, den såkalte *talebananen*.



En talebanan er en figur som framkommer når vi legger talespråklydene inn i et audiogram. De ligger spredt utover et område som blir seende ut som en banan. Framstillingen er forenklet, for hver enkelt språklyd er sammensatt av flere ulike frekvenser. Språklyder er ikke rene toner.

© sansetap.no

Figur 6: Illustrasjon av språklydenes posisjon i frekvensspekteret – den såkalte "talebananen". Kilde: <http://www.sansetap.no/voksne-horsel/om/nedsatt-horsel/horselsmalinger-og-tester/subjektive-tester/>

## 2.2 Programvare

### 2.2.1 Utstyr

- Macbook Pro OSX Version 10.9.5 (MATLAB versjon R2013a)
- Bang & Olufsen Headphones, modell H6

### 2.2.2 Avspilling av lyd med én frekvens

Alt man trenger for å skape et lydssignal på datamaskin er en sekvens av flyttall. Når flyttallene sendes til en høyttaler, sammen med informasjon om den ønskede tidsforskjellen mellom verdiene, bit-kvantisering og kanal (venstre eller høyre øre), skaper høyttaleren trykkforskjeller i luften som av et menneske vil kunne oppfattes som lyd. Ved å la sekvensen av flyttall svare til jevnt utplukkede verdier fra et harmonisk signal,  $\sin(2\pi fn\Delta t)$ , kan vi få høyttaleren til å produsere et lydssignal med en ren frekvens  $f$ . Etersom menneskeøret kan oppfatte frekvenser mellom  $20\text{Hz}$  og  $20\text{kHz}$  har det blitt en konvensjon å velge en samplingsfrekvens,  $F_s = 1/\Delta t$ , slik at en eventuell fouriertransformasjon vil gi informasjon om alle frekvenser i det hørbare spekteret. Ifølge Nyquist–Shannons samplingsteorem bør vi altså velge en samplingsfrekvens



minst lik  $40000\text{Hz}$ . Sonys PCM-adaptorer fra slutten på 1970-tallet innførte valget  $F_s = 44100\text{Hz}$ , som brukes i CDer den dag i dag<sup>9</sup>. Det er ingen grunn til å vike fra denne trenden.

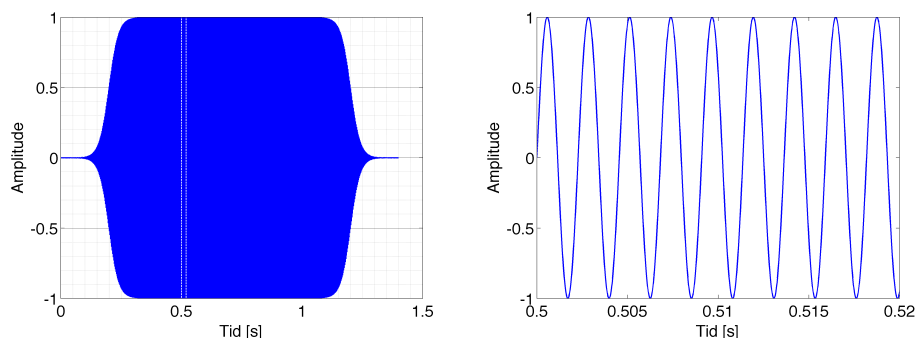
Vi lar altså  $\Delta t = \frac{1}{44100}$  sekunder, og definerer de diskrete tidene ved  $t_n = n\Delta t$  med  $n = 0, 1, \dots, N$  slik at  $N\Delta t \leq T \leq (N + 1)\Delta t$ , der  $T$  er den totale spilletiden.

For å unngå brå endringer i lydsignalet, som kan skape et "lite smell", kan det være lurt å benytte seg av en dobbel sigmoid-funksjon,

$$S(t) = \frac{1}{1 + e^{-(t-a)s}} - \frac{1}{1 + e^{-(t-b)s}},$$

som omhyllingskurve. Vi velger sigmoid-funksjonen slik at  $b - a = 1\text{sek}$ , med en passende  $s$ , og får et lydsignal gitt som

$$L_{f,A}(t_n) = A \sin(2\pi f t_n) S(t_n).$$



Figur 7: Til venstre vises tidsbildet til et 1400 millisekunders lydsignal med frekvens  $440\text{Hz}$  og en dobbel sigmoid funksjon som omhyllingskurve. Til høyre vises det samme lydsignalet for tidsrommet 500 ms til 520 ms. Dette tidsområdet er merket i venstre figur med to hvite, vertikale, stiplede linjer.

### 2.2.3 Valg av lavest hørbare amplitude

Vi ønsker at programvaren på en brukervennlig og rask måte skal kunne finne den lavest hørbare amplituden et lydsignal med en bestemt frekvens må ha for å bli oppfattet av en testperson. Det mest nærliggende er da å begynne på en amplitude som ingen vil høre, for deretter å gå passende steg oppover til testpersonen oppfatter lyden. For å få ekstra nøyaktighet kan man

<sup>9</sup>Wikipedia,  $44100\text{Hz}$ : [http://en.wikipedia.org/wiki/44,100\\_Hz](http://en.wikipedia.org/wiki/44,100_Hz)

etter lyden er erklært oppfattet gå små steg nedover i amplitude til lyden ikke lenger høres. Ettersom sluttresultatet skal oppgis i desibel, bør stegene i amplitude også uttrykkes desibel, slik at steglengden forblir konstant i det ferdige audiogrammet. Lydintensiteten  $I_{\text{dB}}$ , oppgitt i desibel, til en amplitude  $A$  er definert relativt til en referanseamplitude  $A_0$  som følger<sup>10</sup>

$$I_{\text{dB}} = 20 \lg \left( \frac{A}{A_0} \right).$$

Ved å snu om på formelen får vi følgende uttrykk for amplituden

$$A = A_0 10^{\frac{I_{\text{dB}}}{20}}.$$

I programmet begynner vi med  $I_{\text{dB}} = 0$  og går i steg oppover på  $10\text{dB}$ . Det betyr at hver gang testpersonen ikke hører lyden transformerer vi amplituden ifølge

$$A \mapsto A 10^{\frac{1}{2}}. \quad (1)$$

Dersom personen hører lyden går programmet ned i steg på  $2\text{dB}$ . Det betyr at når lyden ikke høres transformeres amplituden ved

$$A \mapsto A 10^{\frac{-1}{10}}. \quad (2)$$

For enkelhets skyld går programmet gjennom 16 jevnt, logaritmisk fordelte frekvenser i intervallet fra  $200\text{Hz}$  til  $10\text{kHz}$  i rekkefølge. For hver frekvens bestemmes først lavest hørbare amplitude for venstre øret, deretter for høyre øret. Implementeringen av dette er gjort ved å opprette en funksjon, `AudioTest`, som spiller av lyd med valgfri frekvens og amplitude til en valgfri lydkanal. Deretter blir testpersonen spurt om lyden ble oppfattet eller ikke. Funksjonen returnerer 1 hvis lyden ikke ble oppfattet og 0 hvis den ble oppfattet. Denne funksjonen kjøres av nok en funksjon, som har for hensikt å bestemme amplituden dersom frekvens og kanal er kjent. Dette gjør den ved å utføre transformasjonen (1) så lenge funksjonen `AudioTest` returnerer 1. Deretter utfører den transformasjonen (2) så lenge `AudioTest` returnerer 0. Når dette er gjort returneres amplituden og hovedprogrammet bytter kanal og, dersom den returnerte verdien var fra høyre øre, øker frekvens. Slik fortsetter programmet til alle amplitudene er samlet.

## 2.2.4 Normalperson og valg av nullpunktsamplitude

For å måle en persons hørsel holder det ikke å vite hvilke amplituder som må til for at personen skal høre et lydsignal. Vi er også nødt til å vite hvilke amplituder som må til for at en "normal" person skal høre det samme

<sup>10</sup>Se seksjon 1.2.3 for rettferdiggjøring av dette.

lydsignalet. Ettersom en testpersons opplevelse av lyd er avhengig av hode-telefonenes utforming bør normalpersonens hørsel være beskrevet i forhold til hodetelefonene brukt i denne programvaren. Den mest nærliggende løsningen er da å bruke testene ovenfor til å samle data fra en gruppe mennesker slik at gjennomsnittet av målingene kan settes til å representere normalpersonens hørsel. Graden av et eventuelt avvik fra den gjennomsnittlige hørselskurven blir da et omtrentlig mål på testpersonens hørselssvekkelse.

Desibel er et mål på relativ intensitet og krever derfor at man definerer en intensitet som kan fungere som et nullpunkt i skalaen. Ettersom programvaren kun benytter seg av relasjoner mellom programvarens tidligere målinger er imidlertid valget av nullpunkt uviktig. Likevel kan vi gjøre en grovt anslag av  $dB(SPL)$ -målets nullpunkt ved å finne den lavest hørbare amplituden for et lydsignal på  $1kHz$ , og velge dette som referansepunkt. Ved å benytte programvaren omtalt i forrige seksjon anslås denne nullpunktsamplituden til å være  $A_0 = 10^{-5}$ .

### 2.2.5 Håndtering av data og forbedringsalgoritme

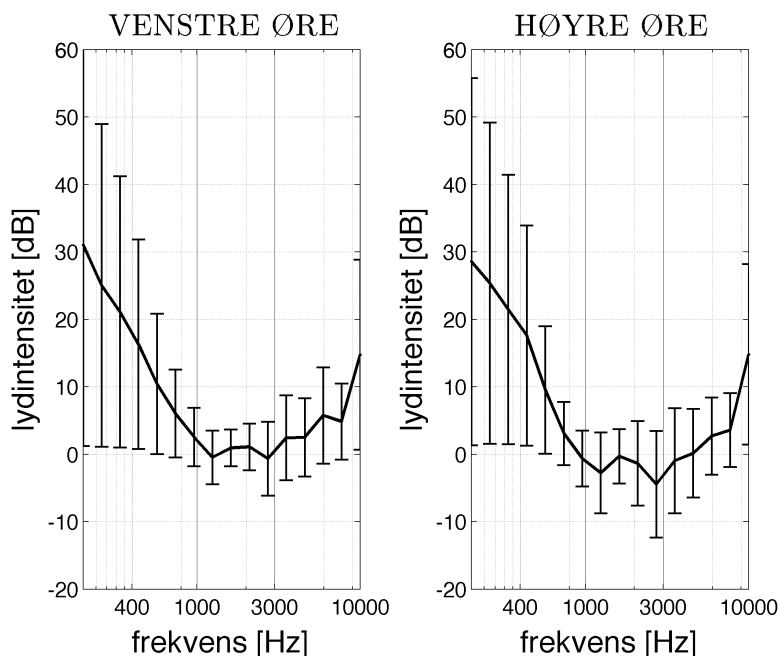
For å hele tiden forbedre konseptet om en "normalpersons" hørsel bør data lagres til fremtidig bruk hver gang hørselstesten utføres på en person som ikke lider av hørselssvekkelse. Da vil bildet av den gjennomsnittlige hørsel bli tydeligere for hver gang testen benyttes. Måten dette er implementert på er ved å sende inn et argument til testen under oppstart som bestemmer om resultatene av testen skal brukes i fremtiden. Testen leser først fra datafilen, som er forskjellig fra øre til øre, og beregner gjennomsnitt og standardavvik for hver frekvens. Dersom testpersonens data skal brukes i fremtiden legges disse til i verdiene hentet fra datafilen. Deretter skrives verdiene tilbake i datafilen.

### 2.2.6 Plotting og informasjonsfremvisning

Når all data er hentet bør den vises frem på en slik måte at man tydelig ser testpersonens hørselstilstand. Dette gjøres ved å returnere et audiogram, ved å plote avvik fra gjennomsnitt, samt en figur der gjennomsnittlig hørsel er plottet sammen med testresultatene. Før å holde et øye med uskarpheten i gjennomsnittskurven plasseres det feilmarginbarer på gjennomsnittskurven som viser datasettets standardavvik for den frekvensen.

## 2.3 Testing og initialisering av programvare

Skrivingen av programvaren krevde svært mye tid til testing av programvare. Finjusteringer på valg av steglengde og frekvensverdier krevde at testen ble gjort mange ganger. Etterhvert som valg av denne typen ble gjort rettet fokus seg mot innhenting av data. Etersom audiometerets programvare blir bedre desto flere testpersoner som har fullført testen måtte mye data samles før hørselstesten kunne anses for å være pålitelig. Testen tok fra 9 til 15min og ble fullført av 15 personer i aldersgruppen 19 – 25år. Selv om dette er lite data, danner gjennomsnittet en vagt bilde av kurven som representerer normalpersonens hørsel.

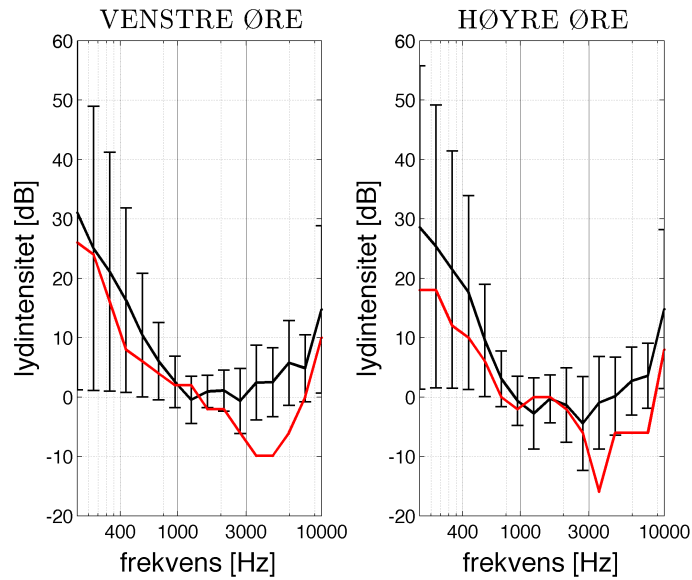


Figur 8: Gjennomsnittlig hørsel estimert ved hjelp av datapunkter fra 15 ulike personer. På hvert målepunkt er datasettets standardavvik illustrert med vertikale errorbarer.

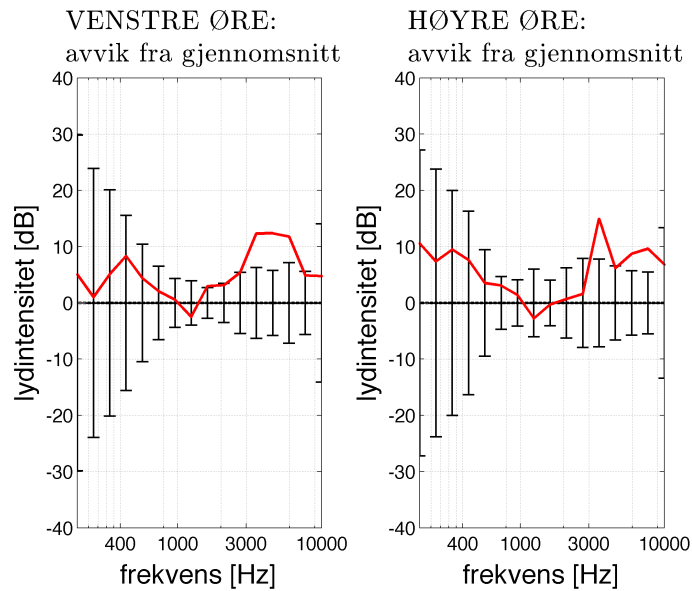
## 2.4 Resultater

Etter de 15 ulike personene tok testen for å initialisere bildet av normalpersonen ble det gjort 4 hørselstester. Data fra testene ble ikke lagret for fremtidig bruk for å holde normalperson-kurven konstant. Resultatene var som følger:

### 2.4.1 Hørseltest nummer én



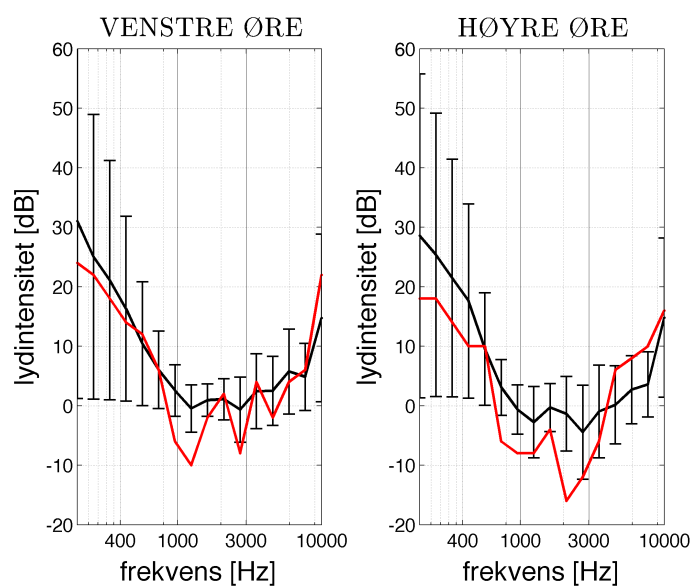
Figur 9: Plot av resultatene fra person 1 sammen med gjennomsnittshørsel. De vertikale barene på datapunktene er ment å illustrere standardavviket i hørselsdata.



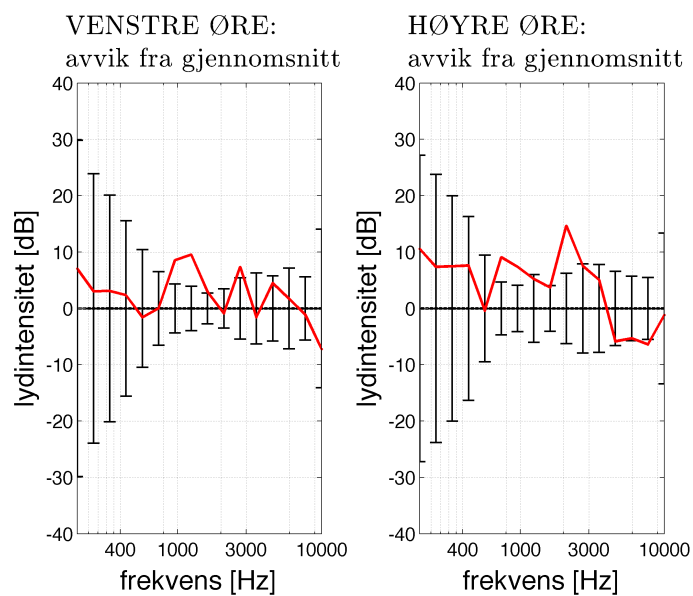
Figur 10: Audiogram for testperson nummer én.

Resultatene fra testperson nummer én viser at personen i snitt har god hørsel. Personen hører, ifølge audiogrammet, frekvenser i ytterkanten av det hørbare spekteret spesielt godt. For frekvenser i intervallet  $3\text{-}7\text{kHz}$  er hørselen over standardavvik-barrieren, noe som vitner om spesielt god hørsel.

#### 2.4.2 Hørseltest nummer to



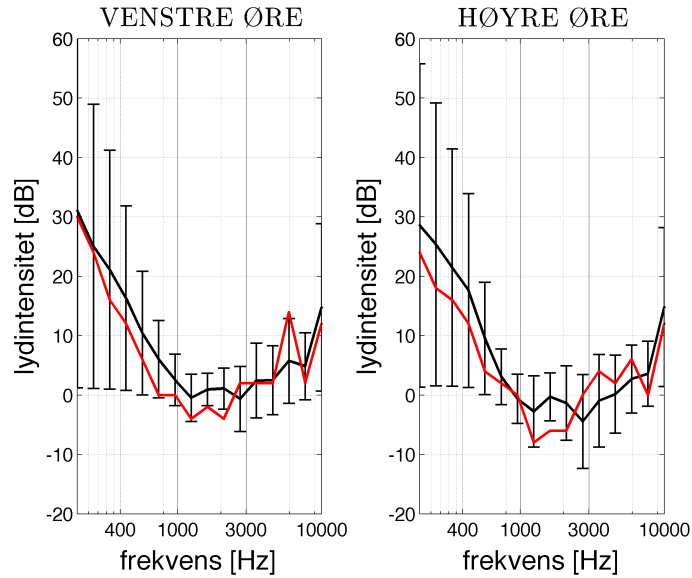
Figur 11: Plot av resultatene fra person 2 sammen med gjennomsnittshørsel. De vertikale barene på datapunktene er ment å illustrere standardavviket i hørselsdata.



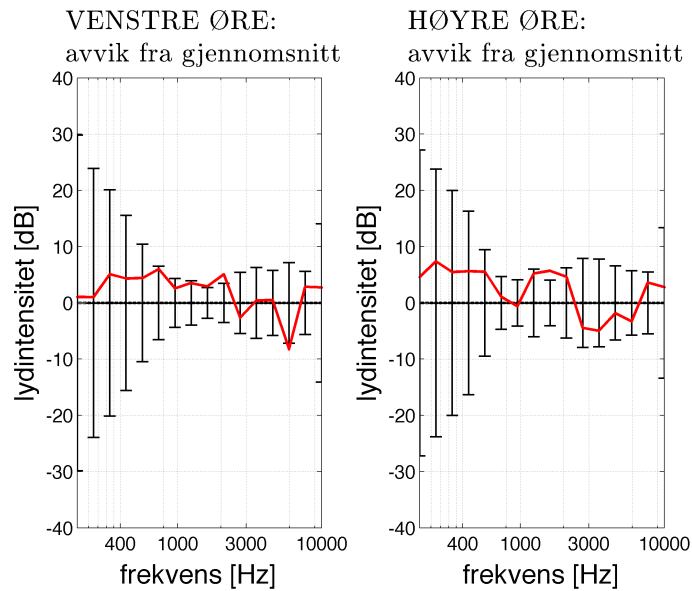
Figur 12: Audiogram for testperson nummer to.

Resultatene fra testperson nummer to vitner om en generelt sett god hørsel. Personen hører frekvenser som er midt i det hørbare frekvensspekteret godt og har bedre hørsel på høyre øre enn på venstre.

### 2.4.3 Hørseltest nummer tre



Figur 13: Plot av resultatene fra person 3 sammen med gjennomsnittshørsel. De vertikale barene på datapunktene er ment å illustrere standardavviket i hørselsdata.

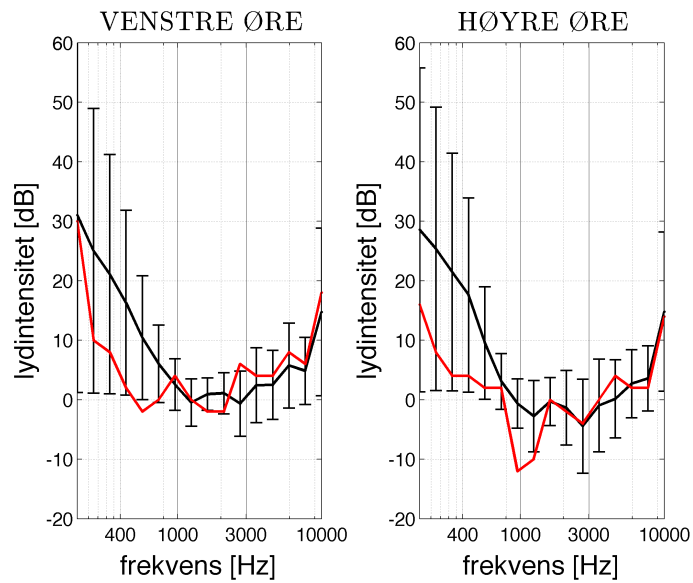


Figur 14: Audiogram for testperson nummer tre.

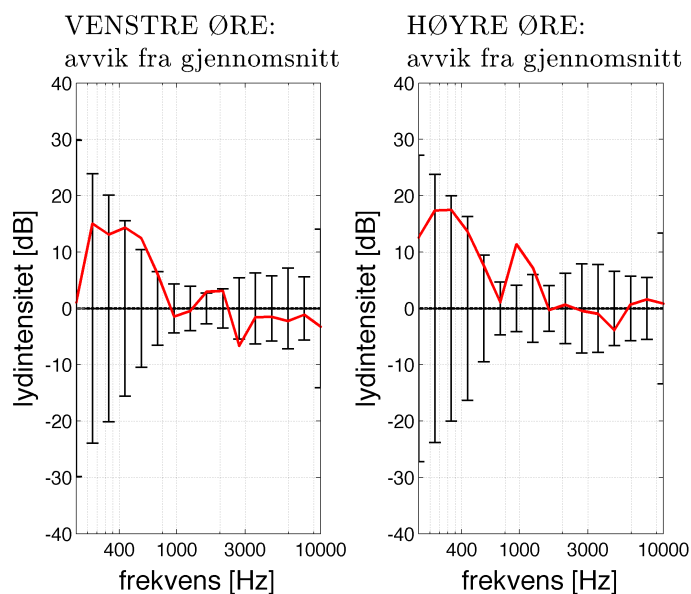


Resultatene fra hørselstest av person nummer tre viter om at personen har alminnelig hørsel.

#### 2.4.4 Hørselstest nummer fire



Figur 15: Plot av resultatene fra person 4 sammen med gjennomsnittshørsel. De vertikale barene på datapunktene er ment å illustrere standardavviket i hørselsdata.



Figur 16: Audiogram for testperson nummer fire.

Resultatene fra hørselstest nummer fire vitner om en person som godt hører lave frekvenser. Ovenfor nedre halvdel av frekvensspekteret har personen alminnelig hørsel. Personen har i gjennomsnitt bedre hørsel på høyre øre enn på venstre.

#### 2.4.5 Om resultater

Selv om det tilsynelatende er stor spredning i resultatene fra hørselstestene over, led ingen av forsøkspersonene av ualminnelig hørselssvekkelse. Personene som tok testen likte den og syntes den virket fullstendig, men at varigheten var litt for lang. Trenden viste at høyre øre hadde litt bedre hørsel enn venstre, men dette kommer trolig av at høyre øre alltid ble testet rett etter det venstre. Standardavvikene i dataverdier vitner om meget store forskjeller i evnen til å høre lave frekvenser, mens det var liten varians i lavest hørbar lydintensitet for frekvenser rundt  $1\text{kHz}$ . Frekvensene som ligger mellom  $1\text{kHz}$  og  $3\text{kHz}$  viste seg å være de som generelt sett oppfattes best. Dette er i tråd med vektingen av hørbar frekvenser i utviklingen av lydmåleren, som er omtalt over.

### 2.5 Ulemper ved løsningsmetoden

Løsningsmetoden for utvikling av et audiometer beskrevet over synes å fungere godt. Den vil forbedres etterhvert som den brukes, testen har rimelig

varighet og den fremstår som brukervennlig. Likevel er det et par ulemper som bør nevnes:

- Nedgang i amplitude fører til psykologisk effekt.  
*Når lavest hørbare amplitude skulle bestemmes ble nesten alle påvirket av den psykologiske effekten som fremkommer av å gå nedover i intensitet. Alle forsøkspersonene gikk på minst én frekvens over fem steg nedover etter lyden var erklært oppfattet. Ettersom stegene nedover har lengde 2dB og stegene oppover har lengde 10dB mente altså alle forsøkspersonene å ha hørt et lydsignal de for bare noen sekunder siden svarte at de ikke oppfattet. Dette kunne vært løst ved heller å hoppe 10dB ned, for deretter å gå oppover igjen i mindre steg til lyden oppfattes for annen gang. Ettersom effekten synes å være svært konsekvent og uavhengig av lydsignalets frekvens vil denne effekten imidlertid ha liten innvirkning på hørseltesten.*
- Frekvenser og kanaler i rekkefølge skaper får hørselen til å virke bedre enn den er.  
*Ettersom testpersonene ble utsatt for lydsignal med samme frekvens flere ganger på rad skaper personen en forventning til lydsignalets tonehøyde. Det gjør at personen lettere hører amplituder den ellers ikke ville oppfattet. Ettersom hver frekvens først testes på venstre øre, og deretter på høyre, vil den samme effekten være gjeldende for hørselssvekkelse til de forskjellige ørene. Dette er trolig mye av årsaken til at mange av hørselstestene vitnet om bedre hørsel på høyre øre enn på venstre. Dette kunne vært løst ved å teste ørene hver for seg. Det tilsvarende problemet med frekvensene kunne vært løst ved å stokke om på frekvensene etter hver lydavspilling. Dette ville trolig gått utover testens varighet.*
- Startamplituden burde variert som funksjon av frekvens.  
*For veldig lave og veldig høye frekvenser var det nødvendig å gå mange steg oppover i amplitude før lyden ble oppfattet, mens for frekvenser mellom 1kHz og 3kHz hørtes lyden med en gang. Dette skaper en inkonsekvent behandling av frekvenser og øker testens varighet. En enkel løsning på dette ville være å for eksempel alltid begynne på den amplituden som svarer til gjennomsnittets minus standardavviket.*

## 2.6 Ulemper ved audiometriske målinger generelt

Fremgangsmåten for selvutvikling av et audiometer beskrevet på de foregående sidene har mange mangler og dårlige løsninger, men hva kan sies om

ulempene ved denne typen audiometriske målinger generelt? Under følger en punktvis diskusjon av dette

- **Problematisk definisjon av normal hørsel.**  
*Selv med kolossale mengder data er definisjonen av en normalperson vanskelig. For eksempel vil hørselskurven til en stor grad avhenge av utstyret som brukes i testen. Problematikken som oppstår ved valget av testpersoner som skal representere normalpersonens hørsel er også viktig. Skal normal hørsel defineres som gjennomsnittet av hørselen til alle mennesker, eller bare til mennesker som ikke har hørselssvekkelse? I såfall, hvor skal man sette grensen mellom en normal hørsel og svekket hørsel? Selv dersom valget av normalperson gjøres konsekvent av alle institusjoner som tilbyr hørselstester, vil dette valget være vanskelig å rettferdiggjøre.*
- **Bakgrunnsstøy.**  
*Uansett hvilken løsning som benyttes for avspilling av lyd til testpersonens øre ikke være fullstendig skjermet fra bakgrunnsstøy. Ettersom bakgrunnsstøyet varierer vil også resultatene av hørselstesten avhenge lydnivået i omgivelsene. I de verste tilfellene kan en testperson med normal hørsel da være så uheldig å bli erklært hørselssvekket på grunn av bakgrunnsstøy.*
- **Hørselstesten er personavhengig.**  
*Dersom hørselen skal testes på hvert øre separat må en eller annen form for hodetelefoner benyttes. Hvordan hodetelefonene passer på testpersonens hode vil imidlertid variere i stor grad. Ettersom lydintensitet er svært avhengig av omgivelsenes geometri betyr dette at selv om lydsignalet som sendes ut er det samme, vil det lydsignalet som treffer testpersonens øre variere.*
- **Det er vanskelig å avgjøre hvilke lyder som oppfattes.**  
*Selv en perfekt utformet hørselstest vil ha stor usikkerhet. Man skulle tro det var en lett avgjørelse å si hvorvidt et lydsignal ble oppfattet eller ikke. Realiteten er at når en testperson utsettes for en knapt hørbar lyd, vil det være krevende å avgjøre om lydsignalet ble hørt eller om det bare "føltes" som det. Beslutningen av en slik avgjøringsprosess vil variere fra person til person og derfor skape usikkerhet i de målte verdiene. Denne effekten forkrøpler imidlertid ikke hørselstester, men skaper en uunngåelig feil i enhver test.*

## 2.7 Programkode

- Hovedprogrammet

```
function CollectData( name, useDataInFuture )
    dataPoints = 16;
    frequencies = logspace(log10(200),log10(10000),
dataPoints);
    amplitudes = zeros(2,length(frequencies));
    i=1;
    zeroDb = 1e-5;
    for freq=frequencies
        amplitudes(1,i) = findAmpForGivenFreq(freq,1); %
venstre
        amplitudes(2,i) = findAmpForGivenFreq(freq,2); %
hooyre
        i=i+1;
    end

    %Hent lagrede data og skriv derom oppgitt
    [VL VR] = handleDataFiles(...
                                'audioDataLeft.txt',...
                                'audioDataRight.txt',...
                                amplitudes(1,:),...
                                amplitudes(2,:),...
                                useDataInFuture...
                                );

    %VENTRE OORE
    meanA_L = [];
    stDevA_L = [];
    for n=1:length(VL(:,1))
        allOfThisFreq = VL(n,2:length(VL(1,:)));
        %gjennomsnitt
        mean = sum(allOfThisFreq)/length(allOfThisFreq);
        meanA_L = [meanA_L mean];
        %standardavvik
        stDev = sqrt(sum((fromAmpToDb(allOfThisFreq,
zeroDb)-mean).^2)/length(allOfThisFreq));
        stDevA_L = [stDevA_L stDev];
    end

    %HOOYRE OORE
    meanA_R = [];
    stDevA_R = [];
    for n=1:length(VR(:,1))
        allOfThisFreq = VR(n,2:length(VR(1,:)));
        %gjennomsnitt
        mean = sum(allOfThisFreq)/length(allOfThisFreq);
        meanA_R = [meanA_R mean];
        %standardavvik
```

```

        stDev = sqrt(sum((fromAmpToDb(allOfThisFreq,
zeroDb)-mean).^2)/length(allOfThisFreq));
        stDevA_R = [stDevA_R stDev];
    end

% plot data
xTickValues = [400 1000 3000 10000];

intensitiesMeanLeft = fromAmpToDb(meanA_L,zeroDb);
intensitiesLeft = fromAmpToDb(amplitudes(1,:),zeroDb
);
intensitiesMeanRight = fromAmpToDb(meanA_R,zeroDb);
intensitiesRight = fromAmpToDb(amplitudes(2,:),zeroDb
);

% MAALEVERDIER SAMMEN MED GJENNOMSNIITT
figure
axes('LineWidth',1.5,'FontSize',20,'FontName','Arial'
);

subplot(1,2,1)
semilogx(frequencies,intensitiesMeanLeft,'Marker','.',
,'Color',[0,0,0],'LineWidth',2); hold on
errorbar(frequencies,intensitiesMeanLeft,stDevA_L,'
Color',[0,0,0],'LineWidth',1.3); hold on
semilogx(frequencies,intensitiesLeft,'Marker','.',
'Color',[1,0,0],'LineWidth',2)
ylabel('lydintensitet [dB]','FontSize',20)
xlabel('frekvens [Hz]','FontSize',20)
grid on
title('VENSTRE {\0}RE','interpreter','latex','
FontSize',20)
set(gca,'fontsize',14)
set(gca,'XTick', xTickValues)
xlim([frequencies(1),frequencies(dataPoints)]);
ylim([-20,60]);

subplot(1,2,2)
semilogx(frequencies,intensitiesMeanRight,'Marker','.',
,'Color',[0,0,0],'LineWidth',2); hold on
errorbar(frequencies,intensitiesMeanRight,stDevA_R,'
Color',[0,0,0],'LineWidth',1.3); hold on
semilogx(frequencies,intensitiesRight,'Marker','.',
'Color',[1,0,0],'LineWidth',2)
ylabel('lydintensitet [dB]','FontSize',20)
xlabel('frekvens [Hz]','FontSize',20)
grid on
title('H{\0}YRE {\0}RE','interpreter','latex','
FontSize',20);

```

```

set(gca,'fontSize',14)
set(gca,'XTick', xTickValues)
xlim([frequencies(1),frequencies(dataPoints)]);
ylim([-20,60]);

plotName1 = sprintf('EarFile_%s',name);
print(plotName1,'-dpng','-r300') % lagre figur

% AUDIOGRAM
figure
axes('LineWidth',1.5,'FontSize',20,'FontName','Arial'
);

subplot(1,2,1);
semilogx(frequencies,0*intensitiesLeft,'--','Color',
,[0,0,0],'LineWidth',2); hold on;
errorbar(frequencies,0*intensitiesLeft,stDevA_L,'
Color',[0,0,0],'LineWidth',1.3); hold on;
semilogx(frequencies,intensitiesMeanLeft -
intensitiesLeft,'Marker','.', 'Color',[1,0,0], '
LineWidth',2);
ylabel('lydintensitet [dB]','FontSize',20);
xlabel('frekvens [Hz]','FontSize',20);
grid on;
title(strvcat('VENSTRE {\0}RE:', 'avvik fra
gjennomsnitt'),'interpreter','latex','FontSize',20);
set(gca,'fontSize',14)
set(gca,'XTick', xTickValues)
xlim([frequencies(1),frequencies(dataPoints)]);
ylim([-40,40]);

subplot(1,2,2);
semilogx(frequencies,0*intensitiesRight,'--','Color',
,[0,0,0],'LineWidth',2); hold on
errorbar(frequencies,0*intensitiesRight,stDevA_R,'
Color',[0,0,0],'LineWidth',1.3); hold on
semilogx(frequencies,intensitiesMeanRight -
intensitiesRight,'Marker','.', 'Color',[1,0,0], '
LineWidth',2)
ylabel('lydintensitet [dB]','FontSize',20)
xlabel('frekvens [Hz]','FontSize',20)
grid on;
title(strvcat('H{\0}YRE {\0}RE:', 'avvik fra
gjennomsnitt'),'interpreter','latex','FontSize',20)
set(gca,'fontSize',14)
set(gca,'XTick', xTickValues)
xlim([frequencies(1),frequencies(dataPoints)]);
ylim([-40,40]);

```

```

plotName2 = sprintf('EarFile_dev_%s',name);
print(plotName2,'-dpng','-r300') % lagre figur

end

```

- Funksjonen for frekvensavspilling

```

function notDetected = AudioTest( freq,ampl,kanal )
    Fs = 44100; %samplingsfrekvens
    T = 1.4; %varighet
    t = 0:1./Fs:T; %tidsverdier
    omhyll = DoubleSigmoid( 0.2,1.2,50.,t );
    oscillasjon = sin(2*pi*freq.*t); %harmonisk signal
    signal = ampl * oscillasjon .* omhyll;
    stereosignal = zeros(length(t),2);
    if kanal==1 %venstre kanal
        EAR = 'LEFT';
    else %hooyre kanal
        EAR = 'RIGHT';
    end
    stereosignal(:,kanal) = signal(1:length(t));
    nBits = 24;
    player = audioplayer(stereosignal,Fs,nBits);
    playblocking(player);

    % spoor om lyden ble oppfattet og returner svar
    fprintf('freq: %d, amp: %d, ear: %s \n',freq, ampl,
    EAR);
    heard = input(...
    'Did you hear it?[ ENTER:no other:yes ] answer:', 's
    ');
    notDetected = strcmpi(heard,''); %1 hvis ikke hoort

end

```

- Den doble sigmoid-funksjonen

```

function out = DoubleSigmoid( pointA,pointB,sharpness,t )
    S1 = 1./(1+exp(-(t-pointA)*sharpness));
    S2 = 1./(1+exp(-(t-pointB)*sharpness));

    out = S1-S2;

end

```

- Funksjonen som finner amplituder for en gitt frekvens

```

function amp = findAmpForGivenFreq( frequency, kanal )
    zeroDbAmp = 1e-5; % startamplitude
    amplitude = zeroDbAmp;

```



```

% gaa opp naar lyden ikke hoores
while AudioTest(frequency,amplitude,kanal)
    amplitude = amplitude*(10^(0.5));
end

% gaa ned naar lyden hoores
while ~AudioTest(frequency,amplitude,kanal)
    amplitude = amplitude*(10^(-1.0/10.0));
end

amp = amplitude;
end

```

- Konversjonsfunksjon fra amplitude til desibel

```

function desibel = fromAmpToDb( amp, ampReference )
    desibel = 10*log10(amp.^2/(ampReference.^2));
end

```

- Funksjonen som håndterer lagrede data

```

function [out1,out2] = handleDataFiles( file1,file2,
newData1,newData2,useDataInFuture )
% LES FRA FIL
dL = importdata(file1,'\t');
VL = [];
for n=1:length(dL(:,1))
    lineCell = dL(n,1);
    lineStr = lineCell{1};
    valuesCell = textscan(lineStr,'%f');
    values = valuesCell{:};
    VL = [VL values]; %legg til nye verdier
end
out1 = VL;

dR = importdata(file2,'\t');
VR = [];
for n=1:length(dR(:,1))
    lineCell = dR(n,1);
    lineStr = lineCell{1};
    valuesCell = textscan(lineStr,'%f');
    values = valuesCell{:};
    VR = [VR values]; %legg til nye verdier
end
out2 = VR;

% SKRIV TIL FIL
if useDataInFuture

```

```

fileID = fopen(file1,'w');
% skriv det som stod der foor
for j=1:length(VL(1,:))
    for i=1:length(VL(:,1))
        fprintf(fileID,'%2.4d ',VL(i,j));
    end
    fprintf(fileID,'\n');
end
% skriv ny informasjon
for i=1:length(newData1)
    fprintf(fileID,'%2.4d ',newData1(i));
end
fclose(fileID);

fileID = fopen(file2,'w');
% skriv det som stod der foor
for j=1:length(VR(1,:))
    for i=1:length(VR(:,1))
        fprintf(fileID,'%2.4d ',VR(i,j));
    end
    fprintf(fileID,'\n');
end
% skriv ny informasjon
for i=1:length(newData2)
    fprintf(fileID,'%2.4d ',newData2(i));
end
fclose(fileID);
end
end
end

```