

PROSJEKTOPPGAVE I FYS2130
Måling og analysering av lyd

Kandidatnummer: 135

Sammendrag

Denne rapporten er todelt. Den studerer hvordan man kan sette opp en audiometritest og lage audiogrammer for å analysere hørsel og hørselsvekking, samt lage en decibelmåler for å analysere støyende omgivelser. I rapporten viser jeg hvordan man kan programmere både et audiometer og en decibelmåler, basert på matematiske modeller og fysikk. Resultatet er to praktisk anvendelige verktøy som blir brukt for å vise sammenheng mellom alder og hørselsvekking, og som viser at det er mye potensielt skadelig støy i bybildet og inne i bygninger. Rapporten konkluderer med at det er mange utfordringer knyttet til både audiometri og støymåling, og at det er særdeles viktig å kunne kalibrere verktøyene mot profesjonelt utstyr.

I. INTRODUKSJON

A. Audiometri og lyd

Lydenergi beveger seg i bølger, og som for andre bølger, er formen bestemt av bølgens *amplitude* A og *frekvens* f .

Amplituden er et mål på bølgens *intensitet*, og kommer av at bølgen lager et *lydtrykk* p i luften, angitt i enhet $[W\ m^{-2}]$. Forholdet mellom intensitet og trykk er

$$I = p^2 \quad (1)$$

Lydintensiteten påvirker ikke øret vårt lineært, så å måle lydtrykket i arbeid per kvadratmeter, er ofte lite ønskelig. Vi bruker derfor heller en logaritmisk *decibelskala*, som måler lydtrykket opp mot en referanseverdi. Decibelskalaen *SPL* (*sound pressure level*) har formen

$$L = [10\ \text{db(SPL)}] \log \frac{I}{I_{ref}} = [10\ \text{db(SPL)}] \log \frac{p^2}{p_{ref}^2} \quad (2)$$

der $I_{ref} \sim 10^{-12}\ W\ m^{-2}$, og er definert ut ifra lyd med frekvens 1000 Hz, og lydtrykk $p_{rms} = 20\ \mu\text{Pa}$ («root mean square»-verdi). Dette er en ofte brukt absolutt dB-skala. I dette prosjektet skal vi bruke *dB(A)-skalaen* (seksjon II. B.).

Lydens frekvens måles i antall svingninger per sekund. Menneskeøret er et sensitivt organ, og kan oppfatte lydbølger med frekvenser mellom 20 og 20 000 Hz, men det skiller best frekvenser mellom 2000 og 5000 Hz. Det betyr at når man måler lyd, vil man gjerne ta hensyn til dette, og derfor vekte målingene deretter.

For å teste hørsel, brukes et *audiometer*. Audiometeret spiller av korte toner med forskjellige frekvenser for en forsøksperson, og målet er å kartlegge ved hvilke lydintensiteter forsøkspersonen ikke lenger hører de forskjellige frekvensene. Deretter presenteres dataene fra forsøket i *audiogram*,

ett for hvert øre, og sammenlignes med hørselen til en «normalperson», representert ved en *minimum audibility curve*¹ for forskjellige frekvenser. Den laveste lydintensiteten et fullfungerende menneskeøre kan høre, er gitt ved 0-phon-kurven. 0 phon er definert ved I_{ref} i likning (2).

I slike tester brukes som oftest hodetelefoner som dekker hele øret. Utfordringen ved dette, er at 0-phon-kurven er definert fra tester der det brukes høytalere, plassert en bestemt avstand fra forsøkspersonen, i et lydtett rom. Vi nøyer oss derfor med å sammenligne absolutte forskjeller mellom forsøkspersonene.

Over tid kan man sammenligne resultatene for samme person, siden hørselen svekkes med alderen, samt påvirkning fra intens lyd over tid (se tabell 1).

Gjennomsnittlig lydintensitet	Tolerabel eksponeringstid
85 dB(A) – travel bytrafikk	8 timer
88 dB(A)	4 timer
91 dB(A)	2 timer
94 dB(A)	1 time
97 dB(A)	30 minutter
100 dB(A)	15 minutter

Tabell 1: For hver 3 dB økning i intensitet, halveres anbefalt eksponeringstid².

B. Lydmåling

I andre del av dette prosjektet ser vi på hvordan vi kan konstruere en lydmåler ved hjelp av en mikrofon og et data-program. En slik lydmåler kan brukes til å undersøke støy-

¹I. Khurana, Medical Physiology for Undergraduate Students, 2014, s. 931

²<http://www.cdc.gov/niosh/hhe/reports/pdfs/2002-0131-2898.pdf>

nivået i forskjellige situasjoner. I samfunnet i dag er det en nødvendighet å kartlegge støynivået ved for eksempel sterkt trafikkerte veier, jernbanelinjer eller byggeplasser, for å unngå at arbeidere skader hørselen ved lengre eksponering for høy lyd (se tabell 1).

Utfordringene ved å lage en lydmåler er mange. Det er viktig å få til en god vektning av de ulike frekvensene, slik at resultatene samsvarer med profesjonelt utstyr. dB(A)-skalaen som vi skal jobbe med er mindre egnet for å ta opp lyd med høy intensitet, men fordi det er den ledende standarden innen lydmåling, velger vi å prøve å optimalisere måleren for denne skalaen.

Et annet spørsmål man må ta hensyn til, er hvordan lyden skal tas opp. Vi velger her å gjøre det ved å midle intensiteten over noen få sekunder, og deretter se på intensiteten i både en relativ, uvektet dB-skala, og en relativ, dB(A)-vektet skala.

Når man har fått målt lyden, er det fortsatt en stor utfordring som gjenstår. Hvordan velger vi referansepunktet slik at målingene våre gir mening? Dette diskuteres videre i II. B..

II. METODER

A. Oppsett av audiometer

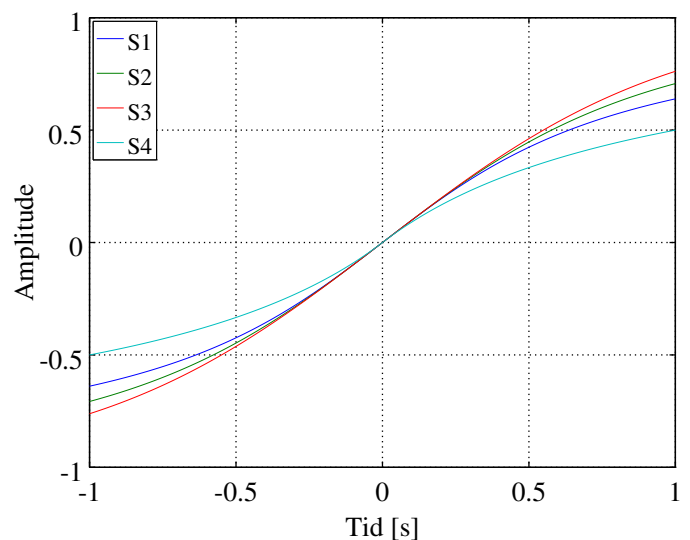
Som nevnt i seksjon I. A., består første del av dette prosjektet i å lage et fungerende audiometer som skal produsere audiogrammer for høyre og venstre øre til en gruppe forsøkspersoner.

Vi konstruerer en ren, tidsavhengig tone med en frekvens f og amplitude A på følgende måte

$$y = A \sin(2\pi ft). \quad (3)$$

Hvis man spiller av denne tonen uten noen modifiseringer, kan man risikere å høre en slags «klikkelyd». Dette er veldig ugunstig når man tester toner med lave intensiteter. Ved å øke amplituden gradvis ved inntoning, og tilsvarende reduksjon ved uttoning, unngår vi denne «klikkingen». Dette gjøres ved å multiplisere tonen med en omhyllingskurve, og i denne rapporten bruker vi en *sigmoid* funksjon (se figur 1 og 2).

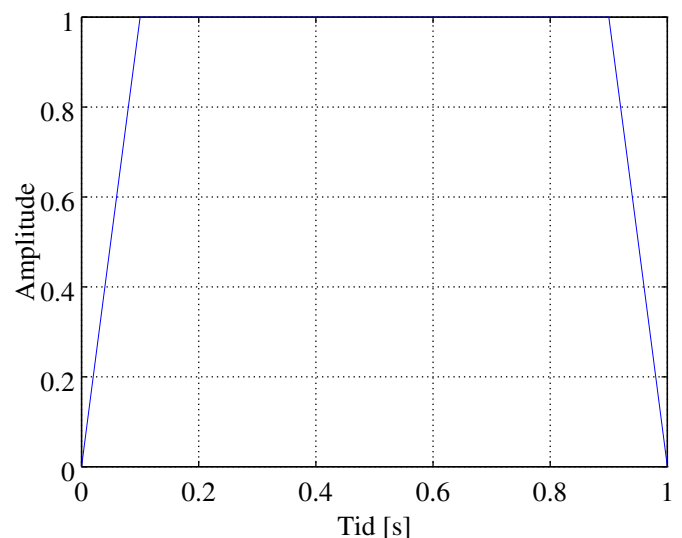
I audiometeret vårt bruker vi S_2 multiplisert med en konstant, slik at den har verdi 1 i $t = 0,1$ s, og vi får en fin inn- og uttoning.



Figur 1: Fire sigmoide funksjoner.

$$S_1 = (2/\pi) \arctan(\pi t/2), \quad S_2 = t/(1+t^2)$$

$$S_3 = \tanh(t), \quad S_4 = t/(1+|t|)$$



Figur 2: Omhyllingskurve gitt funksjonen S_2 multiplisert med en konstant.

Programmet som kjører audiometridelen heter *audiometri.m*, og er relativt enkelt å bruke. Når man kjører programmet dukker det opp en prompt som spør om du vil teste for høyre eller venstre øre, og deretter om du vil plote resultatet når testen er ferdig. Grafen lagres så til fil.

Selve testen utføres ved at brukeren utsettes for en lyd med én frekvens, og lyden varer i ett sekund. Hvis brukeren ikke hører lyden, kan den justere opp intensiteten med enten 1 eller 6 dB om gangen. Hvis lyden er høy, kan man justere ned

intensiteten med 1 dB. Når man har funnet den laveste intensiteten man kan høre, går turen videre til neste frekvens. Frekvensene som måles i dette forsøket er [60, 120, 250, 500, 1000, 2000, 4000, 8000] Hz.

Programmeringsdelen i dette prosjektet er gjort i *Octave/Matlab*, og programlistingen er i vedlegg ??

B. Uvektet og vektet decibelskala

For å kunne bruke en lydmåletest til noe nyttig, må vi ha en måte å sammenligne resultatene våre på. Det gjør vi ved å finne frekvensspekteret ved hjelp av den fouriertransformerte av lydvektoren y , og deretter regne ut en relativ intensitet ved å summere absoluttkvadratene av frekvensspekteret. Til slutt normerer vi resultatet med hensyn på antall punkter i lydsamplevektoren.

Menneskeøret er som kjent ikke like sensitivt³ for lyder ved alle frekvenser. Vi bruker derfor også en *A-vektning*⁴ for å tilpasse programmet vårt til den mye brukte db(A)-skalaen.

Vektfunksjonen kan tilnærmes matematisk for en gitt frekvens f på følgende måte

$$R_A(f) = \frac{12200^2 \cdot f^4 \cdot (f^2 + 12200^2)^{-1}}{(f^2 + 20,6^2) \sqrt{(f^2 + 107,7^2)(f^2 + 737,9^2)}}$$

En kan deretter regne ut relativ lydintensitet med db(A)-vektning slik

$$L_{dB(A)} = [10 \text{ dB(A)}] \cdot \log_{10} I_{dB(A)} + 2.0 - \text{dB}_{ref} \quad (4)$$

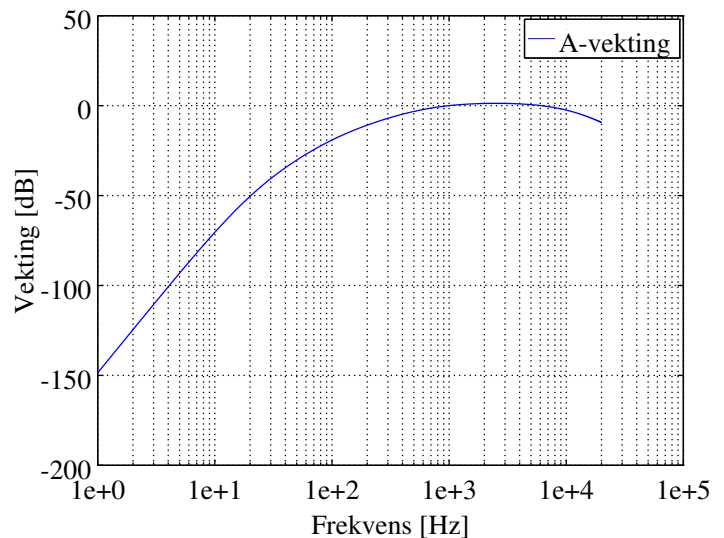
der vi finner den relative totale intensiteten ved

$$I_{dB(A)} = \frac{1}{N} \sum_{i=1}^N R_A(f_i) \cdot |g(i)|^2 \quad (5)$$

der $g = \text{fft}(y)/N$ er «fast fouriertransformasjonen» av lyden vår.

C. Lydmåling og -analyse

Algoritme 1 beskriver kort hvordan programmet *measureSound.m* virker. På grunn av spelling velger vi vekk halvdelen av spekteret, og samtidig fjerner vi det første elementet. Det er når « $f = 0$ », og der ligger gjennomsnittsverdien i spekteret, som vi ikke trenger.



Figur 3: A-vektning med funksjonen $20 \log_{10} R_A(f) + 2.0$. Samsvarer med A-kurven^{4,5}.

Algoritme 1 Lydopptak og fouriertransformasjon

```

1: Sett Fs = 44 100 Hz
2: Sett T = 3 s
3: Sett N = 217
4: Sett ch = 2
5: Sett M = N/2
6: Sett f = 1:M
7: y = recordSound(Fs, T, ch)
8: g = fft(y)/N
9: Initialiser vektorer F og FA
10: for For i = 2, ..., M do
11:     F(i) = |g(i)|2
12:     FA(i) = F(i) · RA(fi)
13: Iiv = sum(F(2:M))/N
14: IA = sum(FA(2:M))/N
15: dBiv = 10 log10 Iiv - dBref
16: dBA = 10 log10 IA + 2,0 - dBref

```

Hovedprogrammene er her *audiometer.m* og *measureSound.m*, som bruker de andre, større funksjonene på en oversiktlig måte. Filnavnene skal greit forklare hva funksjonen gjør, og funksjonene er av forskjellig størrelse, avhengig av hva de skal gjøre.

³<http://hyperphysics.phy-astr.gsu.edu/hbase/sound/earsens.html>

⁴<http://en.wikipedia.org/wiki/A-weighting>

⁵A. I. Vistnes, Svingninger og bølger, 2015, s. 191

III. TESTER

A. Audiometertesting

Audiometeret ble testet på tre (fire med meg selv) forskjellige forsøkspersoner. Aldrene på de involverte var [18, 22, 24, 35] år, og det var to kvinner og to menn. Alder og kjønn er også presentert i resultatdelen (står skrevet i figurbeskrivelsen).

Testen ble utført med øretelefoner som dekker hele øret⁶, i et stille rom uten andre mennesker. Forsøkspersonene fikk prøve programmet et par ganger før vi kjørte gjennom hele testen, og det er disse resultatene som presenteres i neste seksjon.

Jeg valgte å bruke min egen hørsel ved 1000 Hz som referanseverdi, siden jeg er relativt ung og har grei nok hørsel til at jeg burde ligge rundt normalen.

B. Lydmålingstest

Jeg kalibrerte lydmåleverktøyet i et delvis lyddempet rom, og jeg brukte en mobilapp kalt «Sound Meter»⁷ til å hjelpe meg.

Testingen av lydmåleren ble gjort i fire sammenhenger, og målingene er tidsmidlet over tre sekunder.

Den første målingen ble gjort på et kontor på Blindern på kveldstid, etter byggets stengetid. Den andre ble gjort inntil et litt støyende ventilasjonsanlegg. Den tredje målingen ble tatt ut fra vinduet i en blokk, pekende mot ringveien. Den fjerde målingen ble gjort i en park på kveldstid. Noen av de mer konsistente verdiene som ble målt på de ulike stedene er tabulert i 2.

En femte test for å se om det er samsvar mellom de vektete og uvektede decibelmålingene, er å teste for en tone på 1000 Hz, siden det er den frekvensen dB(A)-skalaen får størst utslag for. Ved å spille av en slik tone foran mikrofonen, fikk jeg mange målinger der differansen mellom relativ dB og relativ dB(A), var konsistent mellom 1 og 2, og målingene var jevnt over på samme nivå (~ 40 dB). Se tabell 2.

⁶<http://www.razerzone.com/gaming-audio/razer-electra>

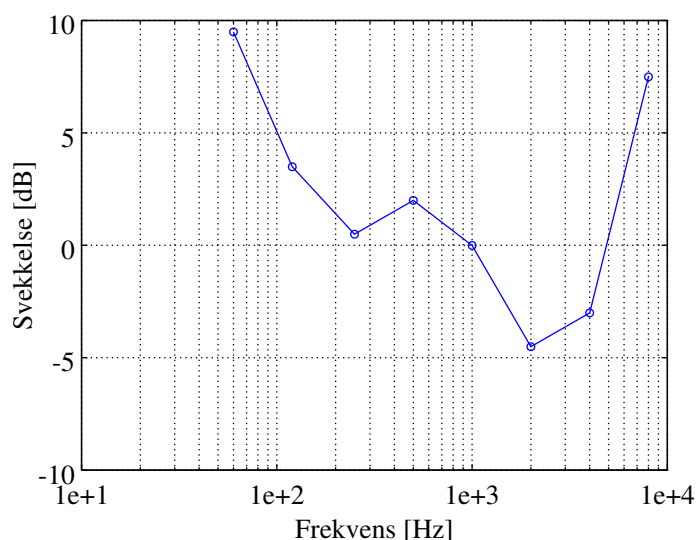
⁷Fra *Smart Tools* co.

IV. RESULTATER

A. Resultater fra audiometri

Figurene 4 til 11 (se Vedlegg B: for figurene 6 til 11) viser audiogrammer for ulike aldersgrupper og kjønn. Selv om det er et i overkant snevert utvalg, er det interessant å se at resultatene stemmer godt overens med det vi vet fra teorien: menn får dårligere hørsel tidligere enn kvinner, og unge under tretti år har gjerne relativt god hørsel.

Det kunne vært enda mer interessant og sett audiogrammene hvis man hadde gjort testene med høyttalere i stedet for øretelefoner.



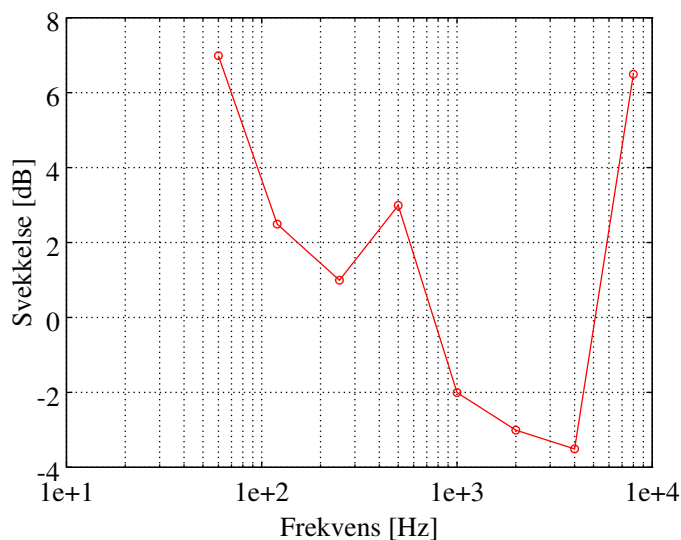
Figur 4: Audiogram for 24 år gammel mann. Høyre øre.

B. Resultater fra lydmålinger

I tabell 2 ser vi resultatet fra decibelmålinger diverse steder på og rundt Blindern.

Hvis vi ser på resultatene fra testen med den rene 1 kHz-tonen, ser vi at det er overraskende godt samsvar mellom vektet og uvektet dB-skala. Både det at differansen mellom målingene er små, og det at målingene holder seg jevnt på samme nivå, kan vise at lydmåleren gir gode resultater.

Siden vi jobber med relative lydintensiteter, er det veldig viktig at lydmåleren gir konsekvente målinger. Selv om den ikke nødvendigvis er kalibrert helt perfekt kan man i hvert fall måle forhold, samt måle forandringer i lydintensitet over tid, som man kan stole på.



så der er det også sannsynligvis forbedringspotensiale. Ved å teste i lydtette rom, kan man også knytte resultatene opp til minimum audibility curve, og få en bedre referanse til profesjonelle resultater på den måten også.

Gjennom å jobbe med denne rapporten ser jeg at det er flere store utfordringer knyttet opp til lydmåling og audiometri.

Figur 5: Audiogram for 24 år gammel mann. Venstre øre.

	dB_A	dB_{iv}
Kontor	30,9	33,1
	31,3	34,0
	31,2	32,8
Ventilasjon	66,6	76,7
	64,8	76,7
	51,7	76,6
Mot vei	59,0	75,4
	63,3	76,0
	64,3	76,5
Park	29,3	31,4
	31,4	33,2
	30,5	32,6
1000 Hz tone	40,6	41,7
	40,9	42,1
	41,2	42,4

Tabell 2: Relative intensiteter i decibel, A-vektet og uvektet. Basert på tresekunders lydopptak.

V. KONKLUSJON

I denne rapporten har jeg prøvd å programmere en totalløsning som er enkel å forstå, samt lett å bruke for noen som ikke nødvendigvis kan programmering.

Vi har sett at resultatene fra både audiometeret og lydmåleren stemmer greit overens med det vi forventet, selv om kalibreringen i lydmåleren sannsynligvis kan jobbes mer med. Referansen til audiometeret er tatt ut fra mine testresultater,

Vedlegg A: Programliste

audiometer.m:

```
Fs = 44100; % Sampling frequency
frequencies = [60,120,250,500,1000,2000,4000,8000]; % [Hertz]
T = 1; % Play time.
N = Fs*T; % Number of elements.
t = linspace(0,T,N);
tCut = 0.1; % For fading in/out.

ch = input("Which ear would you like to test? Press L or R then [Enter].\n","s");
if strcmp(ch,'L') % Test to choose channel.
    ch = 1;
elseif strcmp(ch,'R')
    ch = 2;
else
    Channel = 'Invalid. Exiting.'
    break;
end

plt = input("Would you like to plot the results? Press y or n then [Enter].\n","s");
if strcmp(plt,'n') % Plotting audiogram.
    plt = 0;
elseif strcmp(plt,'y')
    plt = 1;
else
    Plot = 'Invalid. Exiting.'
    break;
end

curve = createSigmoid(t,tCut,'alg',N,T,1); % "Omhyllingskurve" for fade.
A = 0.025; % Reference amplitude.
playSound(A,frequencies,curve,t,N,Fs,ch,plt);
```

measureSound.m:

```
Fs = 44100; % Sampling frequency.
recT = 3; % Record time.

y = recordSound(recT,Fs); % Record sound.

wavwrite(y,Fs,16,'test.wav'); % Save file in .wav format.

M = 2^17; % Number of samples we want.

analyzeSound(y,M) % Compute sound analysis.
```

recordSound.m:

```
function [y] = recordSound (T,Fs)
    y = record(T, Fs);
    'Sound_recording_successful'
endfunction
```

analyzeSound.m:

```
function [] = analyzeSound (y,M)
    y = y(1:M);                                % Cut sound sample vector to M points.
    g = fft(y)/M;                               % Fourier transform.

    absf2 = abs(g).*abs(g);                      % Absolute value of Fourier transform.
    absf2 = absf2(2:M/2);                      % Cut away folding and first element.

    f = 1:length(absf2);                       % Initiate frequencies.
    Aw = Ra(f);                               % Calculate A weighting.

    I = sum(absf2)/M                           % Unweighted intensity.
    IA = sum(Aw*absf2)/M                      % A weighted intensity.

    refdB = 155;                              % Reference dB for our device.

    reldB = 10*log10(I) refdB                 % Relative unweighted intensity.

    reldBA = 10*log10(IA)+2.0 refdB           % Relative A weighted intensity.
```

playSound.m:

```
function [] = playSound (A,frequencies ,curve ,t,N,Fs,ch ,plt)
    chL = 1;
    chR = 2;
    n = length(frequencies);
    Aref = A
    I = zeros(n,1);

    for i = 1:n
        fin = 0;
        A = Aref;
        while fin == 0
            if A > 1
                A = 1;
            end
            A                                     % Print current A for each run.
            freq = frequencies(i)

            oscillation = sin(2*pi*freq*t);      % Sound oscillation array.
```

```

    signal = A*oscillation.*curve;           % Modified sound array.

    stereosignal = zeros(N,2);
    stereosignal(:,ch) = signal(1:N);       % Determine which channel to use.
    sound(stereosignal,Fs);
    response = input("Press one of the following keys and then [Enter] \n...
        i: increase by 1 dB\n...
        I: increase by 6 dB\n...
        d: decrease by 1 dB\n...
        D: done\n","s");
    if strcmp(response,'D')                  % Controls for audiometry test.
        fin = 1;
        I(i) = A;
    elseif strcmp(response,'d')
        A *= 0.891;
    elseif strcmp(response,'i')
        A *= 1.122;
    elseif strcmp(response,'I')
        A *= 1.995;
    else
        'Error: Try again!'
    end
end
end
y = 10*log10(I/Aref);                      % Relative dB measure.

if ch == chR & plt == 1                    % Plot audiogram.
    h = figure;
    semilogx(frequencies, y, 'LineWidth', 1.0, 'o ');
    set(gca, 'LineWidth', 1.0, 'FontSize', 21, 'FontName', 'times');
    xlabel('Frekvens [Hz]', 'FontSize', 21, 'FontName', 'times');
    ylabel('Svekkelse [dB]', 'FontSize', 21, 'FontName', 'times');
    grid on;
    print color depsc '../doc/rightear61.eps'
elseif ch == chL & plt == 1
    h = figure;
    semilogx(frequencies, y, 'LineWidth', 1.0, 'ro ');
    set(gca, 'LineWidth', 1.0, 'FontSize', 21, 'FontName', 'times');
    xlabel('Frekvens [Hz]', 'FontSize', 21, 'FontName', 'times');
    ylabel('Svekkelse [dB]', 'FontSize', 21, 'FontName', 'times');
    grid on;
    print color depsc '../doc/leftear61.eps'
end
endfunction

```

createSigmoid.m:

```

function [curve] = createSigmoid (t,tCut,funcType,N,T,plt)
    tLow = tCut*(N/T)                      % Stop fading in.
    tHigh = N - tCut*(N/T)+1               % Start fading out.

```



```

top = sigmoid(funcType,tCut)           % Amplitude; determines the sound level.

t1 = t(1,1:tLow);
t2 = t(1,tHigh:N);

curve = top*ones(1,N);
curve(1,1:tLow) = sigmoid(funcType,t1);
curve(1,tHigh:N) = sigmoid(funcType,flip(t2) (T tCut));
if plt == 1
    plot(t,curve);
    set(gca,'LineWidth',1.0,'FontSize',21,'FontName','times');
    xlabel('Tid[s]','FontSize',21,'FontName','times');
    ylabel('Amplitude','FontSize',21,'FontName','times');
    grid on;
    print color depsc '../doc/omhyll.eps'
end
endfunction

```

sigmoid.m:

```

function [curve] = sigmoid(str,t)
% This function returns a sigmoid curve array
% for a given array t and a keyword str.

if strcmp(str,'arc')
    output = 'Used_arctan_sigmoid'
    curve = (2/pi)*atan((pi/2)*t);
elseif strcmp(str,'alg')
    output = 'Used_algebraic_sigmoid'
    curve = 10.049847242*t./sqrt(1+t.^2);
elseif strcmp(str,'alg3')
    output = 'Used_algebraic_sigmoid'
    curve = t./sqrt(1+t.^2);
elseif strcmp(str,'alg2')
    output = 'Used_algebraic_2_sigmoid'
    curve = t./(1+abs(t));
else strcmp(str,'tanh')
    output = 'Used_tanh_sigmoid'
    curve = tanh(t);
end
endfunction

```

Ra.m:

```

function [Aw] = Ra (frequency)
% This function returns a weight for the dB(A) scale.
f = frequency;

```

```

f2 = f.*f;          % Efficient vectorized calculations of f^2 and f^4.
f4 = f2.*f2;
Aw = (12200^2*f4)./( (f2+20.6^2).*sqrt((f2+107.7^2).*(f2+737.9^2)).*(f2+12200^2));
endfunction

```

dbaplot.m:

% Program that plots A weighting.

```

f = 1:20000;

Aweight = 20*log10(Ra(f(1:20000))) + 2.0;

h = figure;
semilogx(f(1:20000),Aweight, 'LineWidth', 1.0);
set(gca, 'LineWidth', 1.0, 'FontSize', 21, 'FontName', 'times');
xlabel('Frekvens_[Hz]', 'FontSize', 21, 'FontName', 'times');
ylabel('Vekting_[dB]', 'FontSize', 21, 'FontName', 'times');
h_legend = legend('A vekting');
set(h_legend, 'LineWidth', 1.0, 'FontSize', 21, 'FontName', 'times');
grid on;
print color depsc '../doc/A vekting.eps'

```

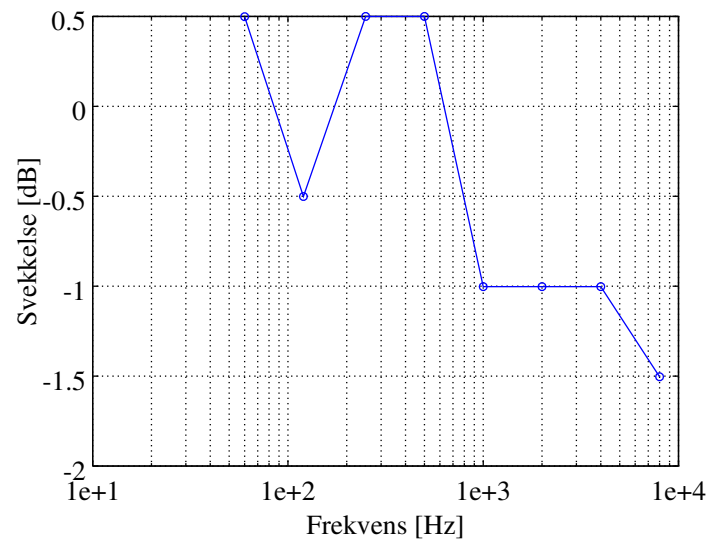
compareSigmoids.m:

```

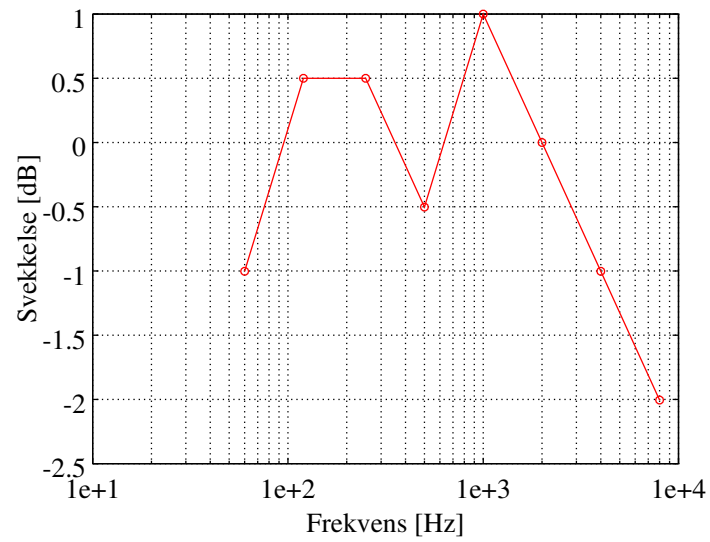
function [] = compareSigmoids (t)
% Function to compare different sigmoids.
plot(t, sigmoid('arc',t),t, sigmoid('alg3',t),t, sigmoid('tanh',t),t, sigmoid('alg2',t))
set(gca, 'LineWidth', 1.0, 'FontSize', 21, 'FontName', 'times');
xlabel('Tid_[s]', 'FontSize', 21, 'FontName', 'times');
ylabel('Amplitude', 'FontSize', 21, 'FontName', 'times');
grid on;
h_legend=legend('S1', 'S2', 'S3', 'S4');
set(h_legend, 'FontSize', 18, 'Location', 'northwest');
print color depsc '../doc/sigmoidsammenligning.eps'
endfunction

```

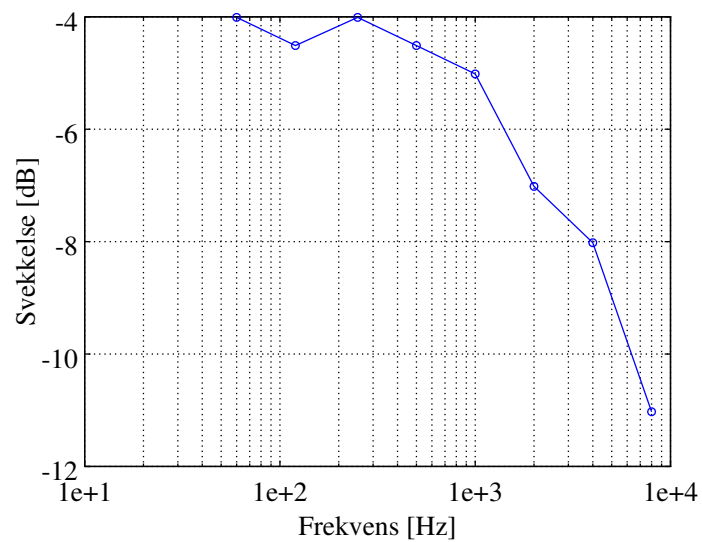
Vedlegg B: Flere figurer



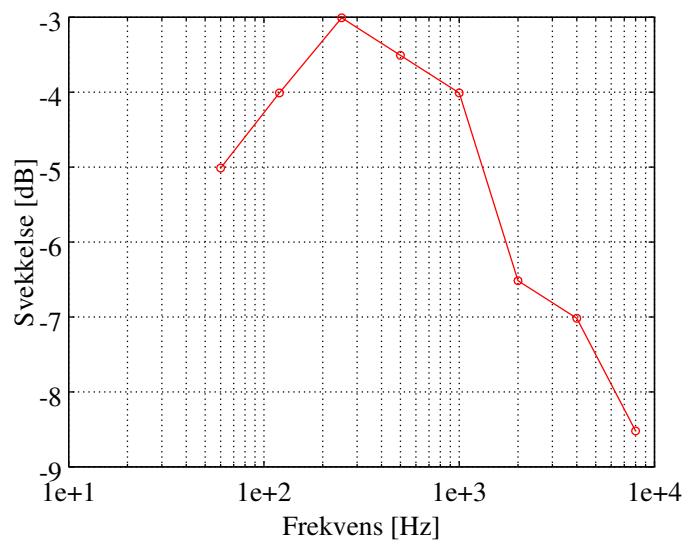
Figur 6: Audiogram for 22 år gammel kvinne. Høyre øre.



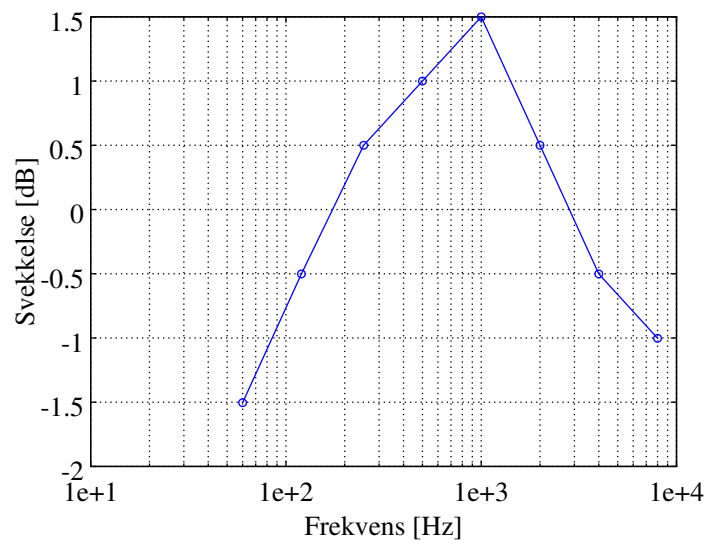
Figur 7: Audiogram for 22 år gammel kvinne. Venstre øre.



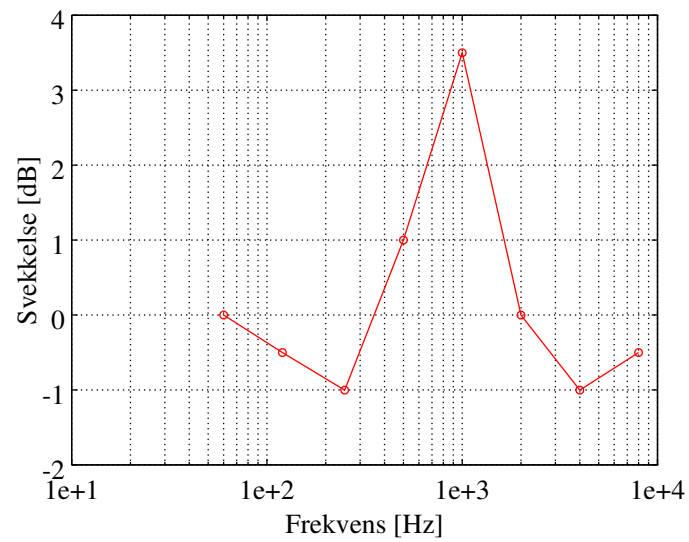
Figur 8: Audiogram for 35 år gammel mann. Høyre øre.



Figur 9: Audiogram for 35 år gammel mann. Venstre øre.



Figur 10: Audiogram for 18 år gammel kvinne. Høyre øre.



Figur 11: Audiogram for 18 år gammel kvinne. Venstre øre.