

Prediksjon av romakustiske forhold i rom med ujevn absorpsjonsfordeling

Håvard Straum

Master i elektronikk

Innlevert: Juli 2012

Hovedveileder: Peter Svensson, IET

Norges teknisk-naturvitenskapelige universitet
Institutt for elektronikk og telekommunikasjon

Sammendrag

Denne oppgaven ser på metoder for databasert prediksjon av spredningsfaktorer. Teorien bak dette har blitt presentert. Det har blitt utarbeidet en metode for å lage måleserier med impulsresponser langs en linje i et rom. Disse måleseriene har blitt lastet inn i Matlab. Her har det har blitt utarbeidet en metode for etterbehandling av disse, med den hensikt å hente ut hver enkelt bølgefront fra den samlede matrisen med alle impulsresponsene. Dette har gjort det mulig å studere hver enkelt bølgefront i detalj og sammenligne de med simulerte verdier. Feilkilder i forbindelse med målingene og etterbehandlingen har også blitt vurdert. Det pekes til slutt på en del videre arbeid som kan være aktuelt å gjøre i framtiden for å utvikle og studere metoden enda nærmere.

Abstract

This paper looks at different methods of computer based prediction of scattering coefficients. The theory behind this have been presented. It has been developed a method to make measurement series with impulse responses along a line in a room. These measurement series has been loaded into Matlab. In Matlab there has been developed a method of post processing of these impulse responses. The goal with this has been to catch each and every wave front from the gathered matrix that holds all the impulse responses. This have made it possible to study each and every wave front in detalj and compare them with simulated values. Sources of error in connection with the measurements and the post processing has also been evaluated. In the end the paper points at different possibilities of further work on the subject, to develop and study the method further.

Forord

Denne oppgaven avslutter mine år ved NTNU Gløshaugen. Det har vært en spennende tid. Og spesielt spennende har det vært å jobbe med denne oppgaven. Jeg valgte denne oppgaven fordi jeg synes det er hørt ut som det kunne medføre en god miks av både praktiske målinger, utvikling av programkode i Matlab og fordypning i teoretisk stoff. Dette viste seg å slå til, og jeg har lært veldig mye i løpet av disse månedene.

Jeg har først lyst til å takke min veileder Professor Peter Svensson som har hjulpet meg veldig mye med de spørsmålene og utfordringene jeg har møtt på underveis. Det er veldig motiverende å jobbe når en har en så dyktig og positiv veileder. Jeg ønsker også å takke avdelingsingeniør Tim Cato Netland for masse kyndig hjelp og rådgiving innenfor fagfeltet måleteknikk. Han har i tillegg vist en fantastisk serviceinnstilling og har alltid stilt opp når jeg har hatt behov for hjelp eller råd. Jeg vil også benytte anledningen til å takke Arne Brevik og Sit Idrett for deres store samarbeidsvilje og positive innstilling i forbindelse med lån gymsal for å gjøre målinger.

Håvard Straum
Trondheim, juli 2012

Innholdsfortegnelse

1	Introduksjon	1
2	Teori	2
2.1	Lydens demningsstruktur.....	2
2.2	Absorpsjon.....	2
2.3	Spredning.....	3
2.4	Oktavbånd.....	5
2.5	Etterklangstid.....	5
3	Metode	7
3.1	Valg av rom til målinger.....	7
3.2	Utstyr benyttet.....	8
3.3	Målemetode.....	8
4	Testing og utvikling av måle- og etterbehandlingsmetode	10
4.1	Tidsforsinkelse.....	10
4.2	Oktavbåndsfiltrering.....	12
4.3	Oppsamling.....	13
4.4	Simulering av impulsresponsen.....	15
4.5	Kurvetilpasning.....	17
4.6	Oppløsning.....	19
5	Målinger i gymsal	21
5.1	Målerommet.....	21
5.2	Utstyr benyttet.....	22
5.3	Målinger.....	22
5.4	Simulering	
6	Resultat	25
6.1	Tom sal.....	25
6.2	Utplasserte matter.....	26
6.3	Utplasserte spredegjenstander.....	27
6.4	Simulering.....	28
6.5	Samlede resultater.....	28
6.6	Etterklangstid og demping.....	31
7	Vurdering av metoden	32
7.1	Diskusjon av feilkilder.....	32
7.1.1	Signal- Støyforhold.....	32
7.1.2	Høytalerens frekvensrespons.....	34
7.1.3	Høytalerens direktivitet.....	36
7.1.4	Mikrofonens direktivitet.....	37
7.2	Metodens bruksområder.....	38
7.3	Videre arbeid.....	38

8	Konklusjon	39
A	Vedlegg	41
	A.1 Matlabkode.....	41

1 Introduksjon

Databasert prediksjon av romakustiske forhold er et arbeidsområde innenfor akustikken hvor det drives mye forskning. For å kunne lage gode databaserte simuleringer av et rom er det først og fremst to størrelser en er avhengig av å kunne estimere nøyaktig for at resultatet av simuleringen skal bli god nok. Disse to størrelsene er absorpsjonsfaktoren og spredningsfaktoren for de ulike flatene i rommet. Når det gjelder absorpsjonsfaktoren har en kommet et godt stykke på vei når det gjelder å utvikle metoder for å beregne denne. Både når det gjelder in-situ-metoder der en beregner absorpsjonsfaktorer til flatene i allerede eksisterende rom og målinger på laboratorier har flere metoder vært utprøvd og det har blitt opparbeidet et relativt stort bibliotek som forteller hva slags absorpsjonsfaktor hvert enkelt materiale har i de ulike oktavnånder. Det er imidlertid fortsatt et langt stykke igjen før en klarer å estimere spredningsfaktorer på en nøyaktig måte.

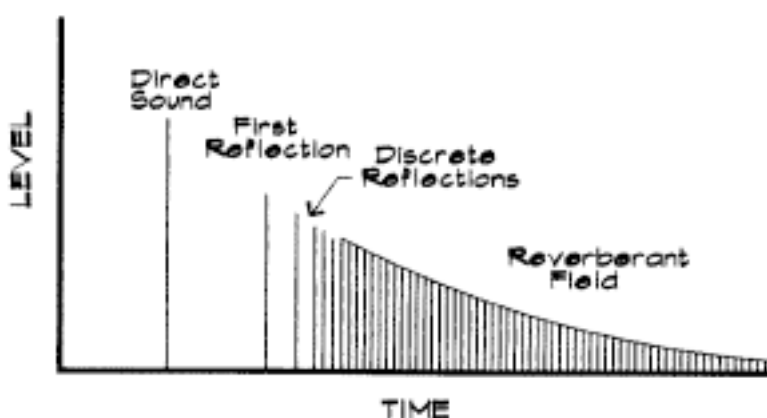
I denne oppgaven vil en metode for å estimere spredningsfaktorer bli forsøkt utarbeidet, og målet er at en skal prøve ut denne metoden i en gymsal. Målingene skal gi grunnlag for å vurdere feilkilder knyttet til metoden.

Det forutsettes at leseren har grunnleggende kjennskap til akustikk, men de fleste formler som blir brukt i denne oppgaven blir forklart.

2 Teori

2.1 Lydens dempningsstruktur

Når en impulsiv kilde som for eksempel et skudd eller en elektrisk induisert puls blir eksitert i et lukket rom kan rommets respons inneholde mye informasjon om de akustiske egenskapene ved rommet. Først er det en initiell lyddempning de første 10-20 millisekundene. Etter at direktelyden har nådd mottakeren følger en rekke med pulser som er de første refleksjonene fra de nærmeste overflatene. Disse refleksjonene er distinkte og identifiserbare. Deretter følger en komplisert rekke med de første få ordens refleksjoner fram rommets overflater. I denne regionen begynner de første akustiske mangler ved rommet å bli tydelige. Sene refleksjoner framstår som isolerte pulser og flutter-ekko dukker opp som repeterende refleksjoner som ikke dør ut like fort som resten av etterklangen. [1] Etterklangen dør etter hvert gradvis ut.



Figur 1: Figuren viser hvordan lyden fra en impulsiv kilde utvikler seg i et lukket rom per tidsenhet. Først kommer direktelyden, så de første refleksjonene, deretter en serie distinkte refleksjoner og til slutt etterklangen.

2.2 Absorpsjon

Når lydbølger treffer et materiale blir energien som finnes i den innkommende bølgen enten reflektert, absorbert i materialet eller transmittert gjennom materialet. Energibalansen kan skrives på følgende måte. [1]

$$E_i = E_r + E_a + E_t \quad (2.1)$$

Forholdet mellom den innkommende og den reflekterte trykkamplituden er definert som den komplekse trykkrefleksjonsfaktoren R_p [2]

$$R_p(\omega, \varphi) = \frac{\hat{p}_r}{\hat{p}_i} = |R_p| e^{j\delta} \quad (2.2)$$

En kan se at refleksjonsfaktoren er en funksjon av frekvensen og vinkelen på den innkommende bølgen, kalt φ . Siden intensiteten både i plane bølger og sfæriske bølger er proporsjonale med kvadratet av lydtrykket, vil en refleksjonsfaktor basert på intensitet være gitt ved $|R_p|^2$. Den delen av den innkommende energien som går tapt i refleksjonsprosessen kalles absorpsjonsfaktoren α og kan dermed skrives som

$$\alpha = 1 - |R_p|^2 \quad (2.3)$$

Transmisjon er ikke tatt hensyn til videre i denne oppgaven siden den analysen som vil bli gjort kun involverer interaksjon ved grensen mellom to medier. Forskjellen mellom absorpsjon, hvor energi blir omgjort til varme, og transmisjon, hvor energi passerer gjennom materialet er ikke relevant. Begge mekanismer er absorptive fra en innkommende bølges synspunkt siden energien ikke blir reflektert. [1] Videre så har gymsaler og generelt store boksformede rom veldig ofte tykke og solide vegger, og det er slike rom som skal analyseres i denne oppgaven. Den transmitterte energien antas derfor å være liten sammenlignet med den absorberte energien.

2.3 Spredning

Når en planbølge blir reflektert mot en ujevn overflate blir ikke all energien reflektert i den spekulære retningen. En del av energien blir reflektert diffust som diffus spredning. Andelen diffus spredning øker ved økende frekvens og er signifikant allerede når irregularitetene ved overflaten er $\gg 0,1 \lambda$. [3]

Ideell spekulær refleksjon



Ujevn overflate



Figur 2: Forskjellen mellom spekulær spredning og diffus spredning

Det har vært antatt at en diffuserende overflate kan karakteriseres gjennom en koeffisient som varierer mellom 0 for perfekt spekulær refleksjon og 1 for komplett diffusjon. Denne skal kunne måles ved hjelp av en standard målemetode. Denne vil variere etter om bølgene er i fritt felt, med en gitt innfallsvinkel, eller i diffust lydfelt med en vilkårlig innfallsvinkel. Spredningskoeffisienten er forholdet mellom den diffust reflekterte energien og den totale reflekterte energien. [1]

$$\delta = \frac{E_{diff}}{E_{tot}} = \frac{E_{diff}}{E_{spek} + E_{diff}} \quad (2.4)$$

Siden det å måle all diffus energi er vanskelig er det vanligvis lettere å måle den spekulært reflekterte energien og bruke den diffuse absorpsjonskoeffisienten til å kalkulere den. Det gir denne likningen

$$\delta = \frac{E_{diff}}{E_{tot}} = 1 - \frac{E_{spek}}{E_{tot}} \quad (2.5)$$

hvor $E_{spek} / E_{inn} = 1 - \alpha_{spek} =$ Normalisert spekulært reflektert energi

$E_{tot} / E_{inn} = 1 - \alpha =$ Normalisert totalt reflektert energi

det gir,

$$\delta = \frac{\alpha_{spek} - \alpha}{1 - \alpha} \quad (2.6)$$

Som gir oss en metode til å måle spredningskoeffisienten δ

2.4 Oktavbånd

Et bredbåndsspekter kan filtreres med standardiserte passbåndfilter, også kalt oktavbåndfilter. Disse har konstant relativ båndbredde. [3] Senterfrekvensene er gitt ved

$$f_{senter} = 10^{\frac{3N}{10}} \quad (2.7)$$

Videre gjelder

$$\begin{aligned} f_{nedre} &= f_{senter} \cdot \frac{1}{\sqrt{2}} \\ f_{\text{\o}vre} &= f_{senter} \cdot \sqrt{2} \\ \Delta f &\approx 0,71 \cdot f_{senter} \end{aligned} \quad (2.8, 2.9, 2.10)$$

2.5 Etterklangstid

Etterklangstiden T_{60} er definert som den tiden det tar før lydtrykksnivået i refleksjonene fra en direktelyd har blitt redusert med 60 dB i forhold til den opprinnelige direktelyden. [4] Sabines formel for etterklangstid er gitt ved

$$T_{60} = \frac{0,163 \cdot V}{A} \quad (2.11)$$

Hvor V er rommets volum i kubikkmeter og A er romabsorpsjonen. A er gitt ved

$$A = \sum_i \alpha_i \cdot S_i \quad (2.12)$$

Som er summen av alle absorpsjonsbidrag i rommet. S er hver enkelt flateenhet og α er dens tilhørende absorpsjonsfaktor. I store rom må en også ta hensyn til luftabsorpsjonen. Da blir den samlede etterklangstiden gitt ved

$$T_{60} = \frac{0,163 \cdot V}{A + 4mV} \quad (2.13)$$

Der m er svekningskonstanten for luftabsorpsjon i dB/m. En god tilnærmet verdi for m er gitt ved

$$m = \frac{0,074}{\phi} * f^2$$

(2.14)

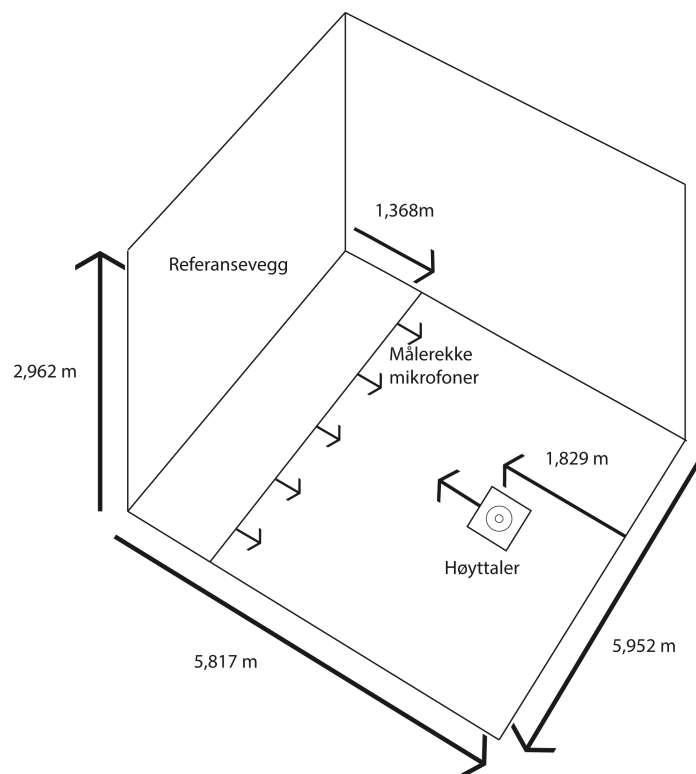
Hvor ϕ er relativ luftfuktighet i % og f er frekvens i kHz. Luftabsorpsjonen øker sterkt med økende frekvens. [4]

3 Metode

For å utvikle en metode til å hente ut hver enkelt spekulære refleksjon skapt av en impulskilde i et lukket rom er det mange faktorer en må ta hensyn til. Det ble først lagt vekt på å hente inn et så bra datagrunnlag som mulig. Målinger ble foretatt i et lite rom under mest mulig ideelle forhold. Det datagrunnlaget en hentet ut fra dette rommet ble deretter lastet inn i Matlab og bearbeidet der. Ved hjelp av et ferdig script i Matlab ble det laget en simulering av impulsresponsen i et rom som tilsvarte det rommet målingene ble gjort i. Denne simuleringen ble brukt til å sammenligne de målte resultatene med. Dermed kunne en studere feilkilder og det ble tatt tak i de ulike faktorene som hadde betydning for nøyaktigheten i måleresultatene. Etter at en hadde minimert feilkildene mest mulig ble det foretatt flere ulike måleserier i en middels stor gymsal.

3.1 Valg av rom til målinger

For å hente inn et best mulig datagrunnlag er det viktig å velge et rom som er mest mulig ideelt. Det er ønskelig at rommet har mest mulig harde overflater. For å hindre påvirkning fra takrefleksjoner er det ønskelig at det er så høyt som mulig under taket. For å gjøre simuleringen enklest mulig å implementere i ettertid er det en fordel om rommet er boksformet. Et lite forelesningsrom ble valgt til å gjøre de innledende målingene i. Sideveggene var av betong, mens de to andre veggene hadde en gipsoverflate. Alt av møblement og inventar ble fjernet fra rommet før en startet med målingene. Det eneste som var igjen i rommet da var en tavle og en vask, samt to dører og to vinduer med innfelte vindusposter. Gardinene ble trukket til side.



Figur 3: Rommet som ble brukt til datainnhenting

3.2 Utstyr benyttet

For å få en mest mulig omnidireksjonell spredning av lyden ble en spesialutviklet høyttaler benyttet i målingene. Denne var formet som en sylinder og kassen var laget av aluminium. Den var helt lukket i bakkant av sylinderelementet og plassert i front av sylinderelementet. Sylinderelementet var 40 cm lang og hadde en diameter på 5 cm. Høyttalerelementet hadde en diameter 4,8 cm. En omnidireksjonell mikrofon ble også benyttet. For å måle opp dimensjonene av flatene i rommet mest mulig nøyaktig ble det benyttet en laser avstandsmåler. Følgende utstyr ble benyttet i målingene:

Produkt	Produsent	Modell/Type
Bærbar datamaskin	Dell	
Effektforsterker	Utviklet ved NTNU	
Høyttaler	Utviklet ved NTNU med Aurasound fulltoneelement	Sylinderformet høyttaler, høyttalerelement av typen NSW2-326-8A
Mikrofonmembran	BSWA Technology Company Limited	Type MP 216 frittfelt (snr 4501095)
Mikrofonkropp	BSWA Technology Company Limited	SM 4000 phantom-drevet
Eksternt lydkort/phantom power forsterker	M-Audio	Mobile Pre USB (snr 071G159A12125)
Laser avstandsmåler	Bosch GmbH	PLR 30 (snr 887514578)

3.3 Målemetode

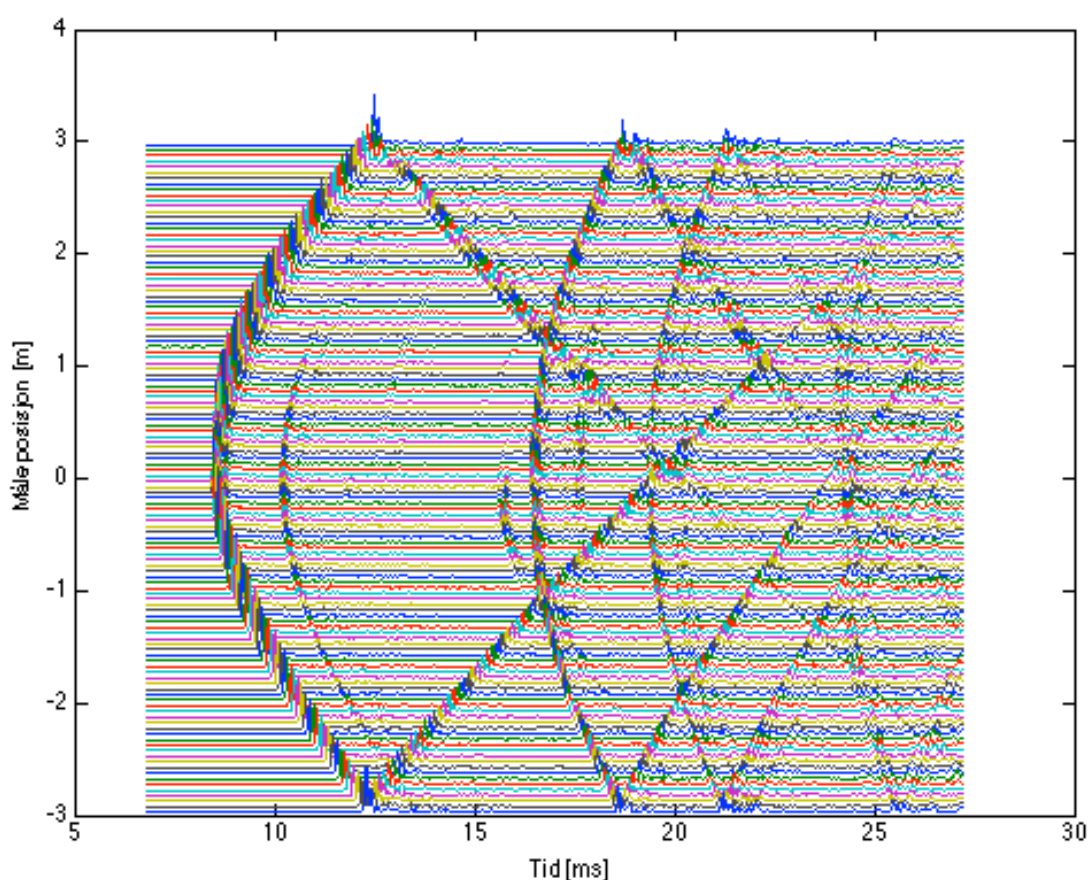
Mikrofonen og høyttaleren ble plassert så nært gulvet som mulig. Dette ble valgt for å hindre at takrefleksjoner skulle komme tidlig i de målte impulsresponsene og forstyrre de veggrefleksjonene en er ute etter å undersøke. Både mikrofon og høyttaler ble satt på et stativ med fronten pekende mot gulvet. Mikrofonen ble plassert 19 mm over gulvet, målt fra sentrum av mikrofonmembranen. Den stod i en vinkel på 60 grader i forhold til gulvet. Høyttaleren ble plassert 37 mm over gulvet, målt fra sentrum av høyttalerelementet. Referanseveggen ble valgt som den veggen der tavla er plassert. For å hindre at refleksjonen fra veggen bak høyttaleren og veggen bak mikrofonen, også kalt referanseveggen, skulle interferere med hverandre ble høyttaleren plassert ca. 2/3 romlengde unna

referanseveggen. Den linja det skal gjøres målinger fra ble plassert ca. 1/4 romlengde unna referanseveggen. Høytaleren ble plassert midt mellom de to sideveggene. Høytaleren peker mot referanseveggen mens mikrofonen peker bort fra referanseveggen. For å sørge for høyest mulig oppløsning i resultatene ble det gjort 120 impulsresponsmålinger fra den valgte målelinja, med 5 cm mellomrom mellom hver måleposisjon. Datamaskin ble plassert i rommet ved siden slik at en kunne gå ut av rommet under hver måling og dermed påvirke måleresultatet minst mulig.

For å måle impulsresponsen ble programmet WinMLS benyttet. Siden det er veldig viktig at målingene er tidssynkrone med hverandre ble det koblet en tilbakekobling mellom utgangssignalet på kanal 2 og inngangssignalet på kanal 2 på det eksterne lydkortet. Mikrofonen ble koblet til inngangssignalet på kanal 1 og en minijackplugg ble koblet fra utgangen på lydkortet og til inngangen på effektforsterkeren. Denne var koblet direkte til høytaleren. Det eksterne lydkortet var koblet til datamaskinen med en USB-kabel. Et sinus-sveip på 1,5 sekunder ble tatt opp med en punktprovingsfrekvens på 44100 Hz. Den forhåndsvalgte verdien på 16 bit ble beholdt som oppløsning. Ut fra sinus-sveipet kalkulerer WinMLS impulsresponsen. Denne blir så lagret på datamaskinen. Mellom hver måling ble mikrofonen flyttet 5 cm, hele tiden med samme avstand til referanseveggen. Prosedyren ble gjentatt for alle de 120 målingene. De 120 impulsresponsene blir så importert i Matlab. Resten av beregningene og presentasjon av figurer blir gjort i Matlab.

4 Testing og utvikling av måle- og etterbehandlingsmetode

Hensikten med den innledende målingen var å skaffe et datagrunnlag som kunne brukes til videre testing og bearbeiding. Dette ble foretatt i Matlab. Alle målingene ble først importert og samlet i en stor matrise. Disse ble så presentert i en samlet figur.



Figur 4: Impulsrespons ufiltrert

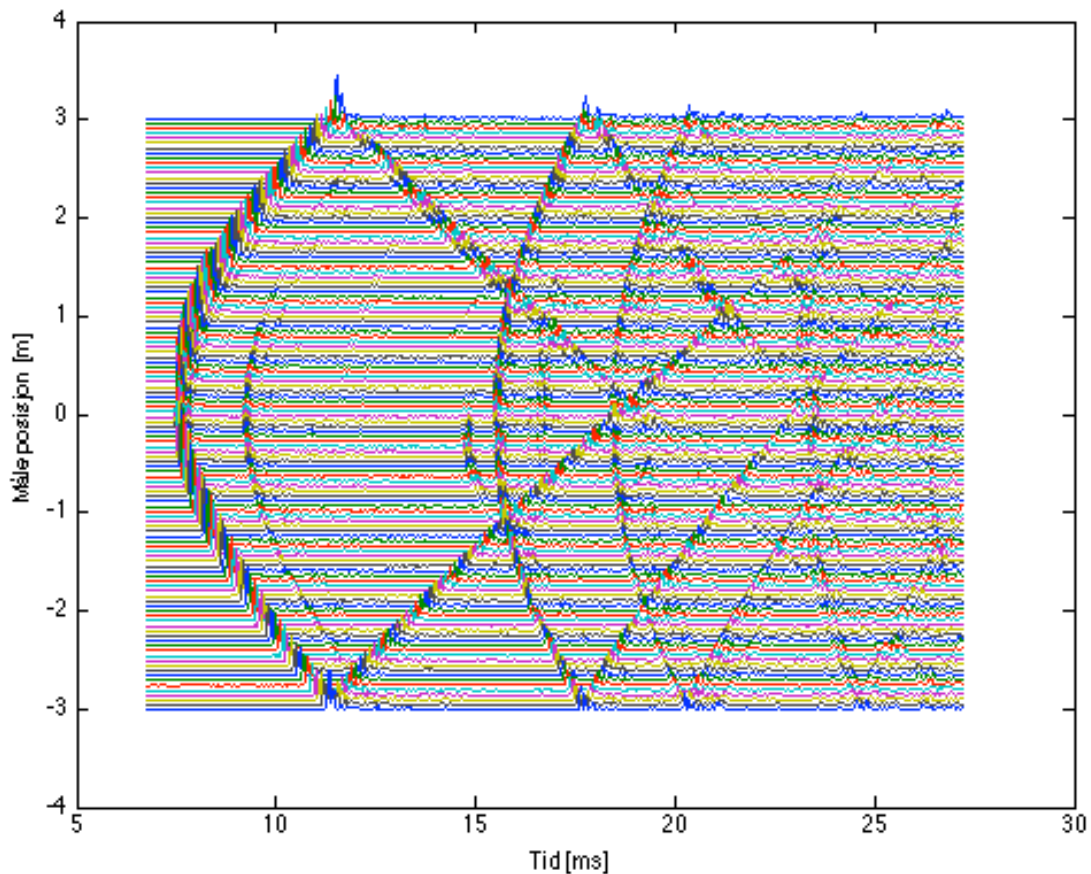
4.1 Tidsforsinkelse

Den første bølgefronten en ser er direktelyden. Siden lyden utbreder seg med hastigheten $c \approx 343$ m/s (Kinsler) ved 20 °C kan regne seg fram til når direktelyden skal nå fram til den midterste mikrofonen ved å bruke formelen:

$$t = \frac{d}{c}$$

(4.1)

Der t , er tiden i sekunder, d er avstanden i meter og c er lydhastigheten. Dette gir $t \approx 7,6$ ms. Som en kan se på figuren kommer direktelyden først litt seinere enn dette, etter over 8 ms. Dette skyldes i hovedsak at lydkortets digital til analog-konverter benytter sigma-delta konvertere, men kan også skyldes selve datamaskinen. [3] Dette gir en tidsforsinkelse, vanligvis på ca 1 ms. For å ta høyde for denne tidsforsinkelsen ble det gjort en måling i WinMLS der utsignalet ble koblet direkte til innsignalet. Den resulterende impulsresponsen ble deretter analysert for å finne ut på hvilket punktprøvningsnummer den første pulsen kom på. Alle punktprøvinger før denne posisjonen ble deretter slettet fra alle målingene.

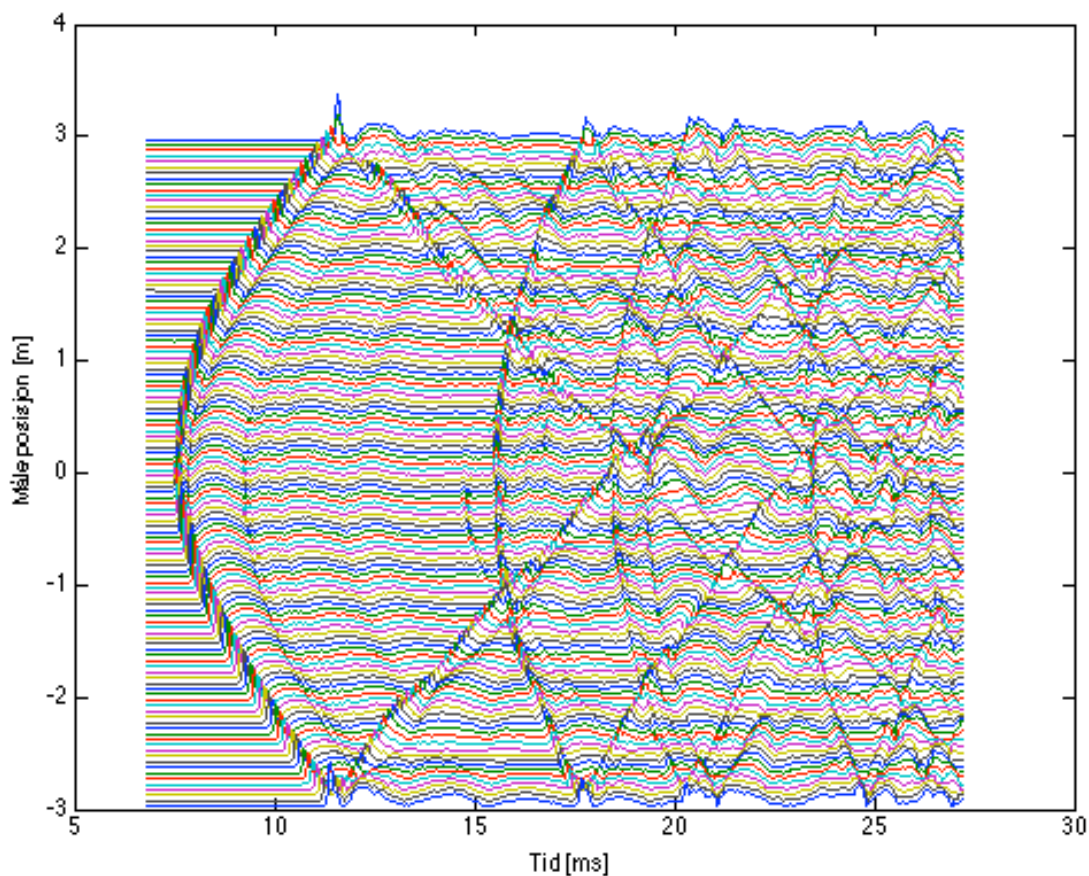


Figur 5: Impulsrespons, ufiltrert, etter at tidsforsinkelse i lydkort er tatt høyde for

Ved å bruke formelen 4.1 kan en finne ut hvilken refleksjon de ulike bølgefrontene i figur 5 representerer. Refleksjonen fra referanseveggen skal da komme etter ca. 15,6 ms, noe som stemmer bra med figuren. Refleksjonen fra veggen bak høyttaleren skal nå fram til sentermikrofonen etter ca 18,3 ms. Dette stemmer også bra med figuren.

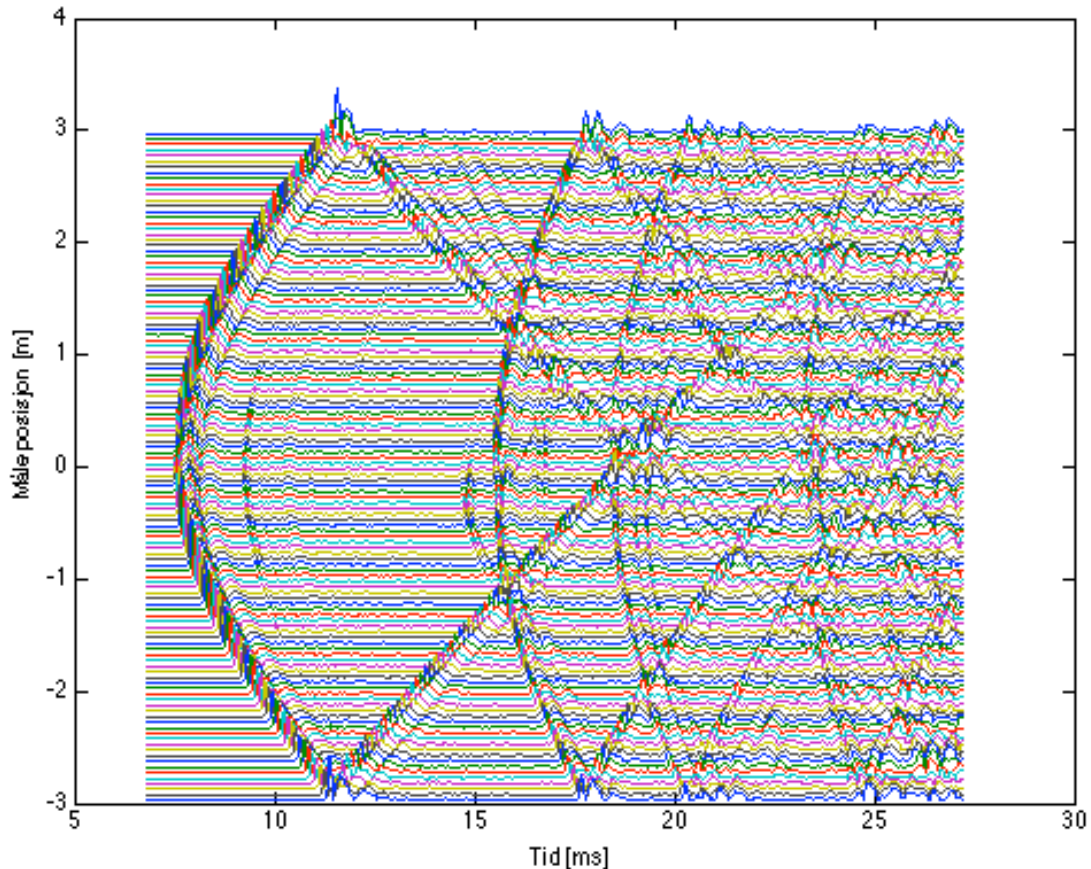
4.2 Oktavbåndsfiltrering

Siden spredningsfaktor og absorpsjonsfaktor er direkte avhengig av frekvensen er det ønskelig å filtrere impulsresponsen i oktavbånd. For å få minst mulig rippel i passbåndet ble et butterworth-filter valgt til dette formålet. Et førsteordens butterworthfilter ble implementert i Matlab ved hjelp av de innebygde funksjonene *butter* og *filter*. Rippel i passbåndet er meget uheldig, spesielt for de laveste oktavbåndene, siden rippelen kan gjøre at bølgefrontene interferer med hverandre.



Figur 6: Impulsrespons, oktavbåndsfiltrert med 250 Hz som senterfrekvens

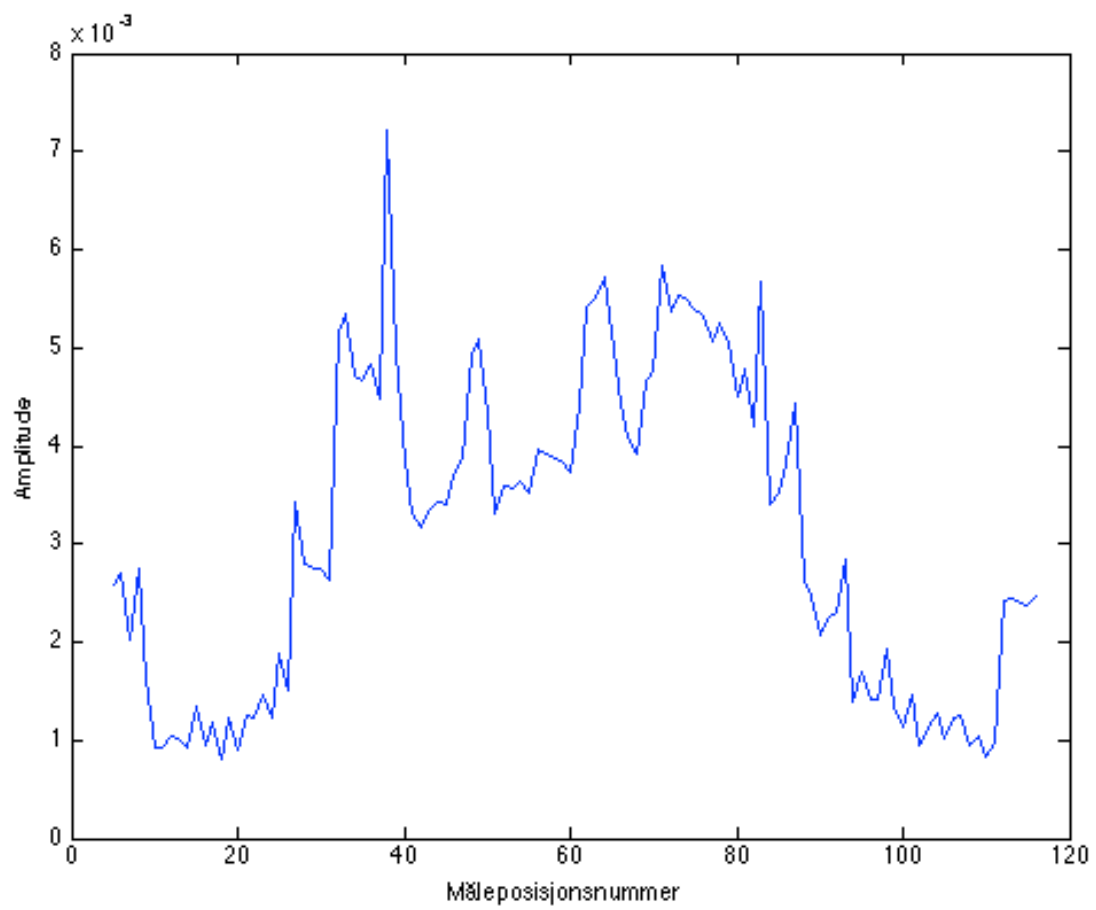
Oktavbåndet med 250 Hz som senterfrekvens er det oktavbåndet med den laveste senterfrekvensen som det er valgt å undersøke i denne oppgaven. En kan se at det oppstår litt rippel, men at bølgefrontene likevel står klart og tydelig fram. Filteret anses derfor som tilstrekkelig. Rippel avtar ved høyere senterfrekvens. For 1000 Hz i senterfrekvens er rippelen nesten ikke synlig lenger.



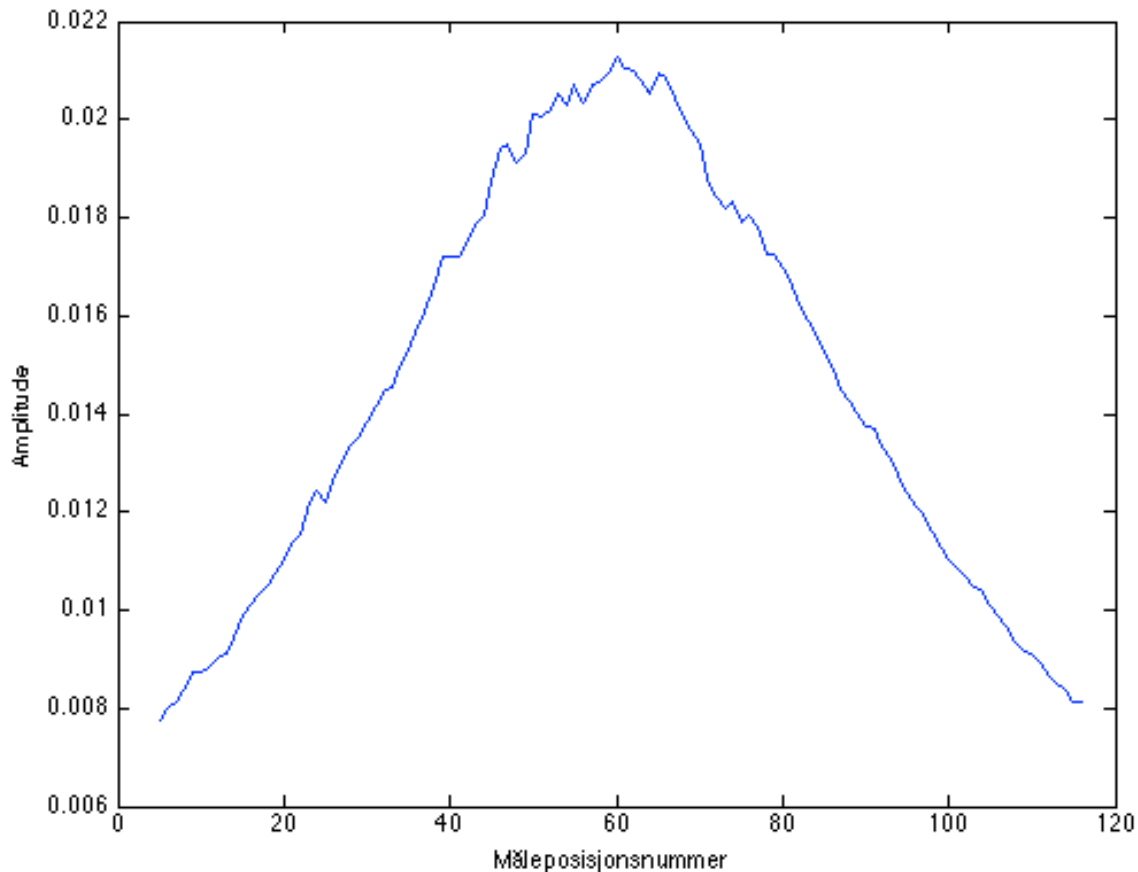
Figur 7: Impulsrespons, oktavbåndsfiltret med 1000 Hz som senterfrekvens

4.3 Oppsamling

En ønsker å hente ut amplituden for hver bølgefront langs hele målerekka slik at en kan studere hvordan amplituden for hver bølgefront varierer langs rommet målingen ble gjort i. Siden disse bølgetoppene kan være veldig bratte, spesielt for direktelyden og for oktavbånd med høy senterfrekvens, er det nødvendig å oppsamle de målte verdiene for at oppløsningen skal bli god nok. Hvis ikke vil en få veldig unøyaktige verdier for amplituden. Et skript for å hente ut hver enkelt bølgefront i Matlab ble laget, og en oppsamling med en faktor på 8 ble implementert. Denne oppsamlingen blir gjort gjennom interpolasjon ved at det settes inn nuller mellom punktprøvingene slik at vektoren får riktig lengde. Deretter blir nullerene interpolert med de originale punktprøvingene ved hjelp av et spesielt lavpassfilter slik at en ender opp med en vektor som er n ganger lenger og med n ganger større punktprøvingsfrekvens. Det finnes en innebygd funksjon i Matlab for dette som heter *interp*. Det ble også forsøkt med en oppsamplingsfaktor på 16, men dette gav ingen synlige forbedringer, så en oppsamplingsfaktor på 8 ble ansett som tilstrekkelig.



Figur 8: Direktelyden, oktavbåndsfilerert med 8 kHz som senterfrekvens, uten oppsamling

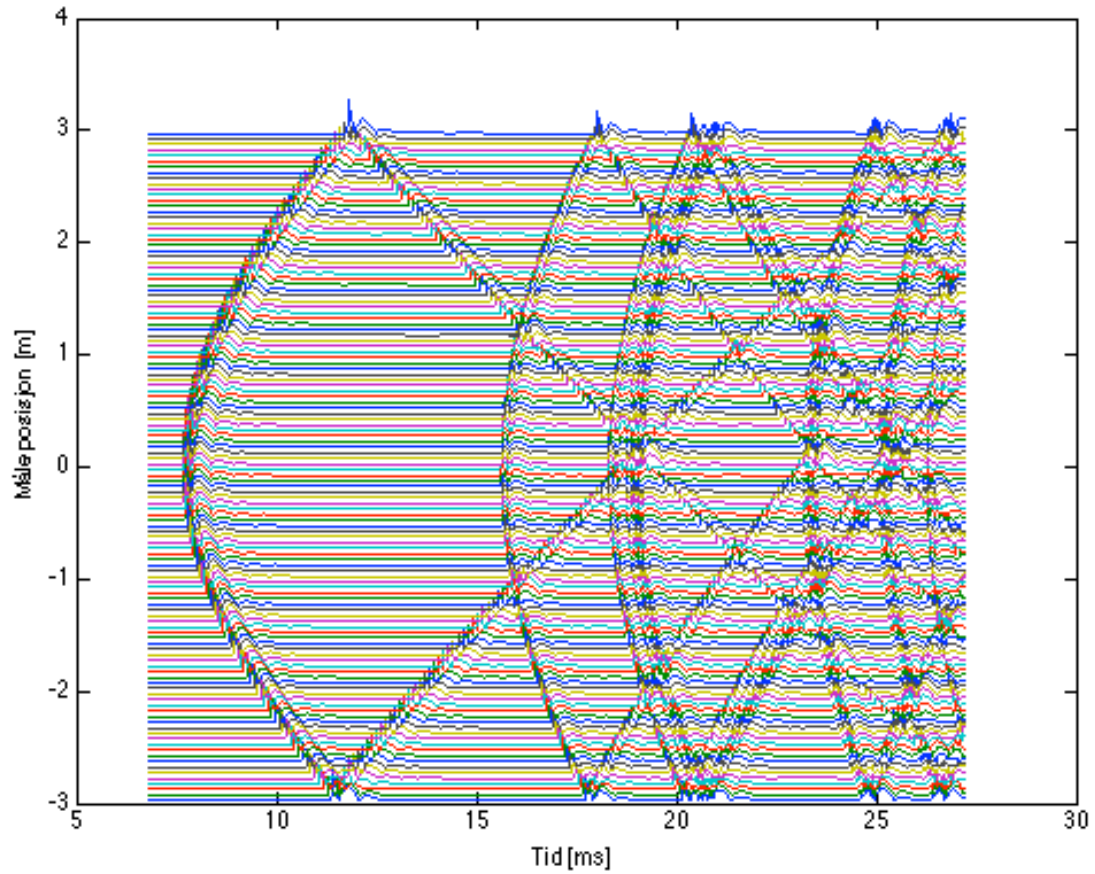


Figur 9: Direktelyden, oktavbåndsfiltret med 8 kHz som senterfrekvens, med enn oppsamplingsfaktor på 8

En kan se av figur 8 og 9 at oppsamlingen har veldig stor betydning for nøyaktigheten i avlesningen av amplituden for hver enkelt bølgefront.

4.4 Simulering av impulsresponsen

For å kunne sammenligne de målte verdiene med helt ideelle verdier ble det laget en simulering av impulsresponsen av et rom med like dimensjoner som det rommet målingene ble gjort i. Denne simuleringen ble laget ved hjelp av et ferdig skript som kan installeres i Matlab. Det er utviklet av Dr Emanuël Habets og kalles *Room Impulse Response Generator*. Den er basert på kilde-speilmetoden. I simuleringen ble alle overflater definert som perfekt reflekterende. Det vil si at alle overflater blir gitt en refleksjonsfaktor på 1.

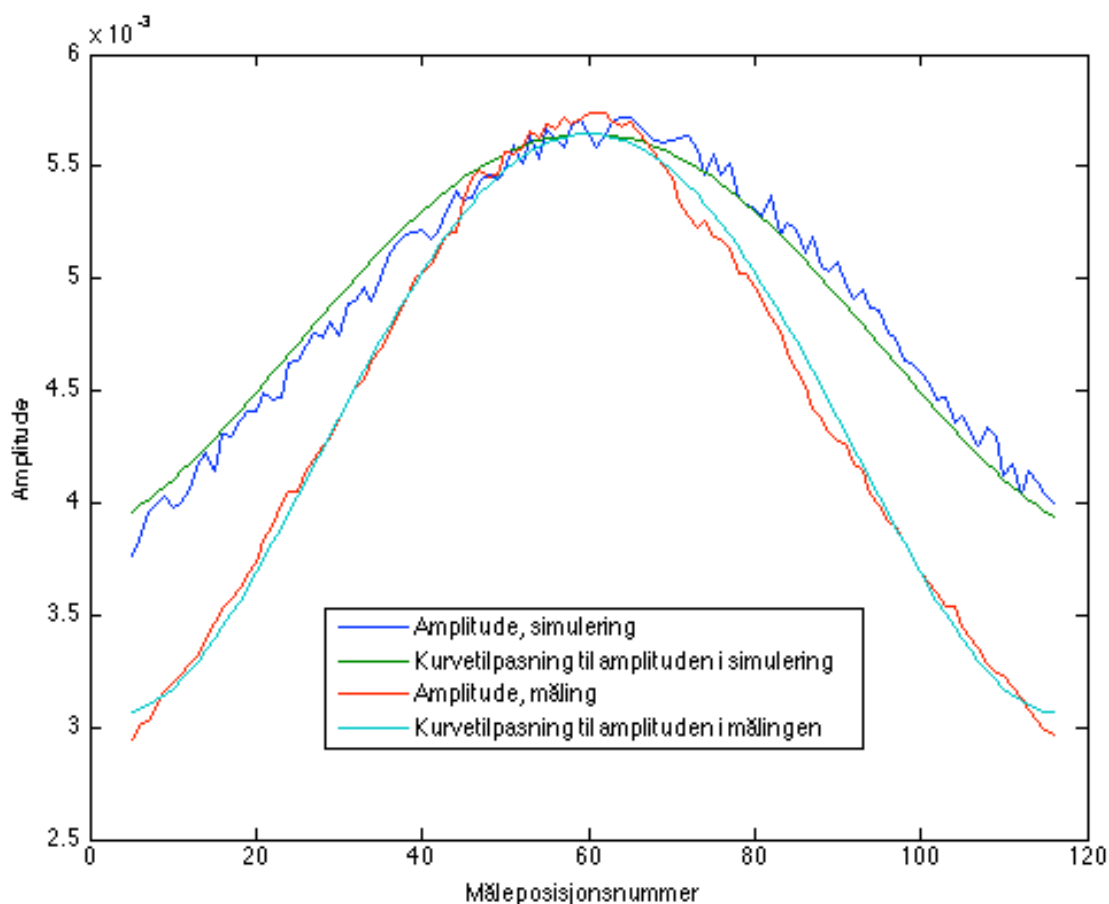


Figur 10: Simulert impulsrespons, oktavbåndsfiltret med 1000 Hz som senterfrekvens

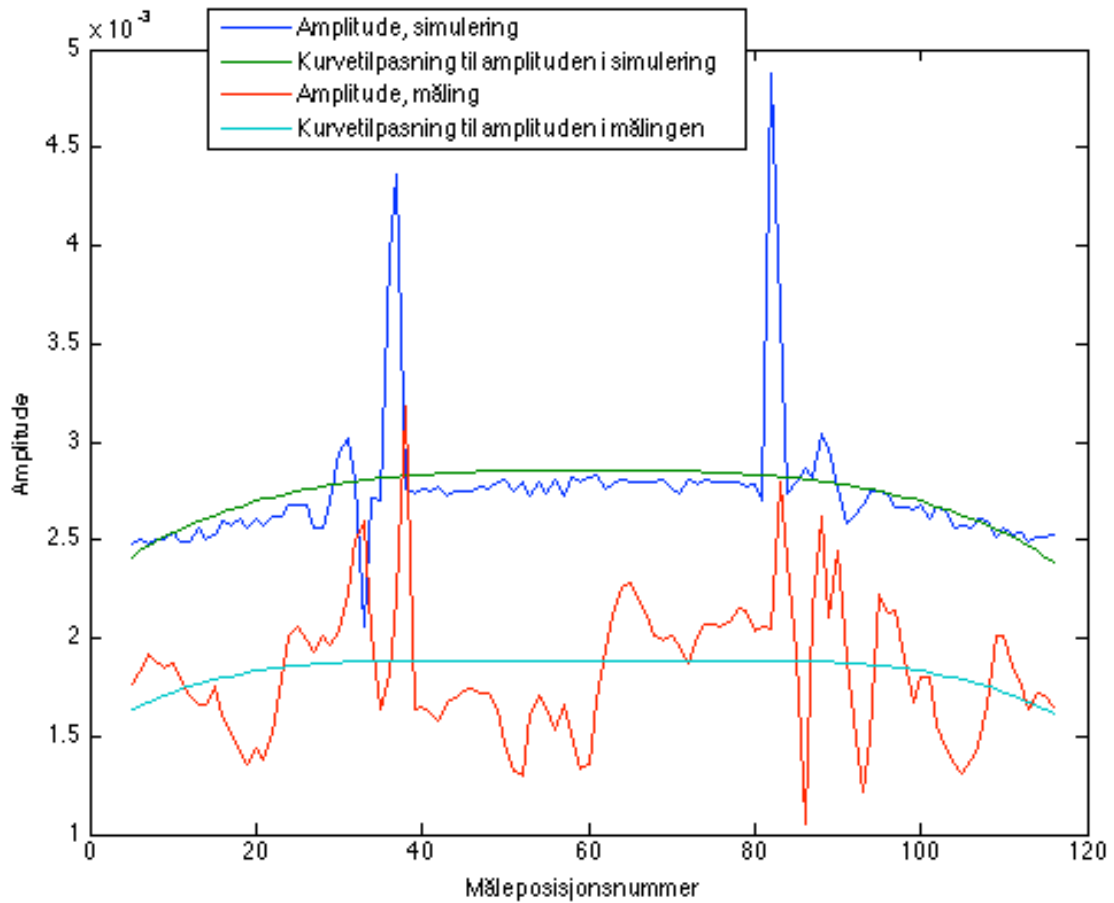
Når en sammenligner simulert impulsrespons med målt impulsrespons ser en at de stemmer veldig godt overrens med hverandre.

4.5 Kurvetilpasning

Når en skal sammenligne amplituden i de simulerte bølgefrontene med amplituden i de målte bølgefrontene vil en vanligvis få veldig store lokale variasjoner langs målelinja. Dette kommer blant annet av at bølgefrontene interfererer med sideveggsrefleksjoner, rippel fra oktavbåndsfileret og forhold ved rommets beskaftenhet. For å kunne sammenligne de simulerte bølgefrontene med de målte bølgefrontene ble det derfor laget en kurvetilpasning til de amplitudekurvene en hentet ut. Denne kurvetilpasningen blir gjort både for simuleringen og den målte amplitudekurven. Siden amplituden i simuleringen gir ut en vilkårlig verdi for amplituden må denne tilpasses til målingene ved hjelp av et forholdstall. Dette forholdstallet finner en ved å hente ut verdien i kurvetilpasningen til simuleringen for den midterste mikrofonplasseringen for direktelyden og tilsvarende verdi for kurvetilpasningen til målingen. Hele matrisen til simuleringen blir deretter multiplisert med det forholdstallet som gjør at kurvetilpasningen for simuleringen av direktelyden og kurvetilpasningen til direktelyden for målingen får lik verdi for den midterste mikrofonposisjonen. Dette kan man se et eksempel på i figur 11.



Figur 11: Direktelyden for måling og simulering, oktavbåndsfileret med 1000 Hz som senterfrekvens med kurvetilpasning.

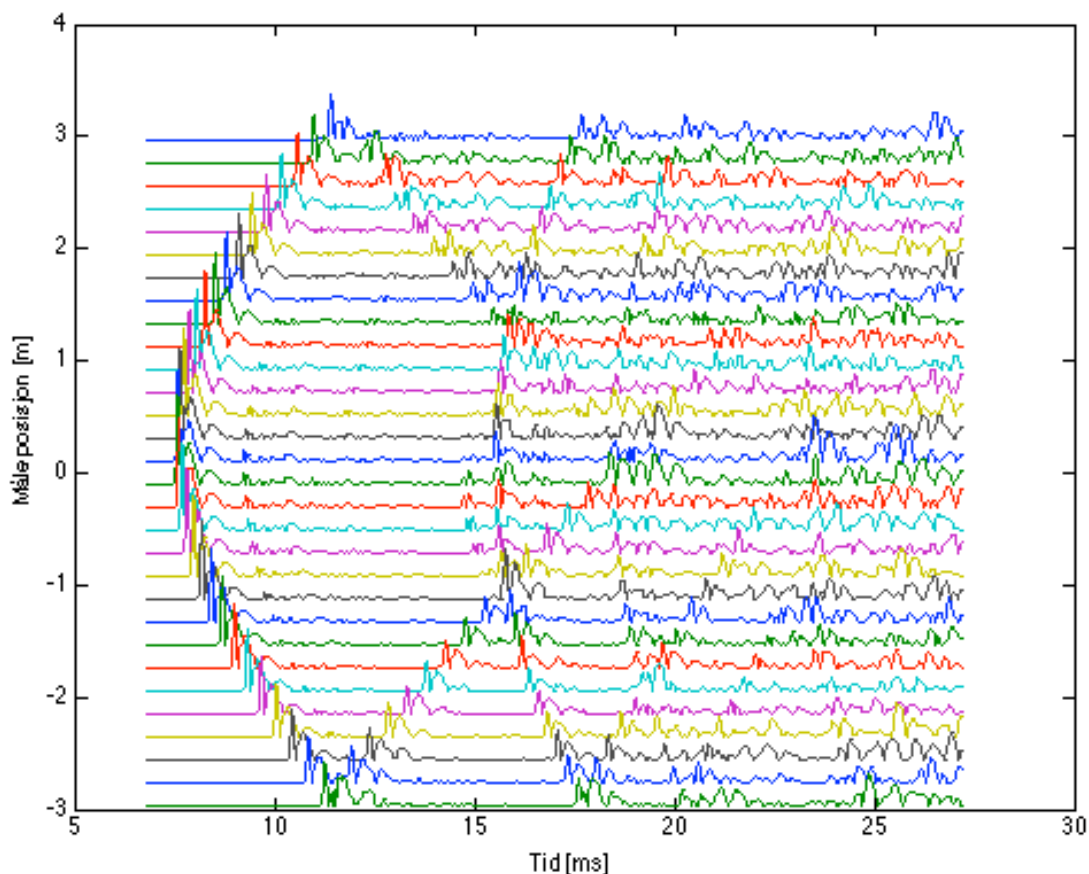


Figur 12: Spekulær refleksjon fra referanseveggen for måling og simulering, oktavgbåndsfiltrert med 1000 Hz som senterfrekvens med kurvetilpasning.

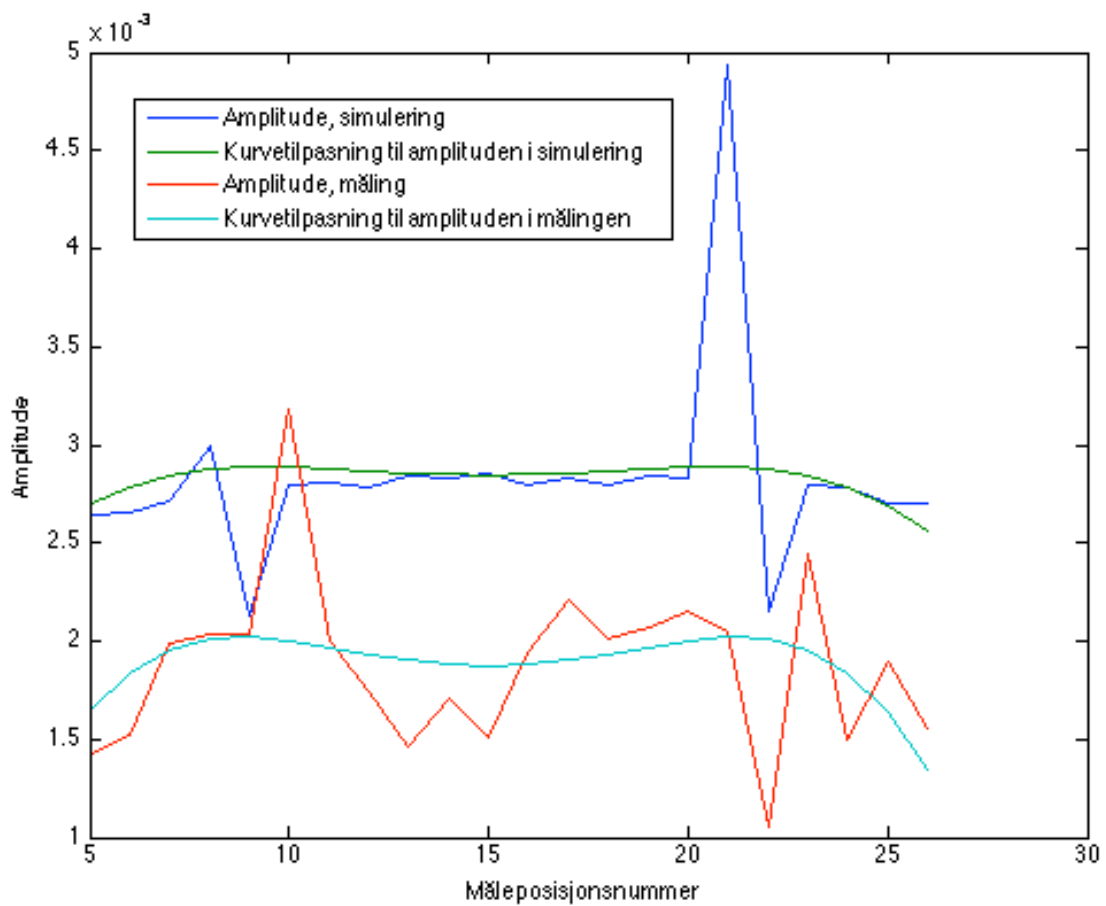
Kurvetilpasningen er gjort ved å benytte den innebygde funksjonen *polyfit* i Matlab.

4.6 Oppløsning

Det ble også sett på hvor stor oppløsning i form av avstand mellom hver måling det var nødvendig å ha for å få et bra resultat. En prøvde først å se hvilken innvirkning det fikk hvis en doblet avstanden mellom hver måling, slik at det i dette tilfellet ble 60 målinger i stedet for 120. Dette hadde veldig begrenset betydning både for opptegningen av bølgefrontene og for den resulterende kurven for amplituden langs bølgefronten for refleksjonen. Hvis en derimot gikk ned til kun 30 målinger og 20 cm mellom hver måling begynte oppløsningen å bli merkbart dårligere. Det ble da vanskeligere å se detaljer i opptegningen av bølgefrontene og den lave oppløsningen fikk også innvirkning på kurven for amplituden til refleksjonen.



Figur 13: Impulsrespons, oktavbåndsfiltret med 1000 Hz som senterfrekvens med kun 30 måleposisjoner

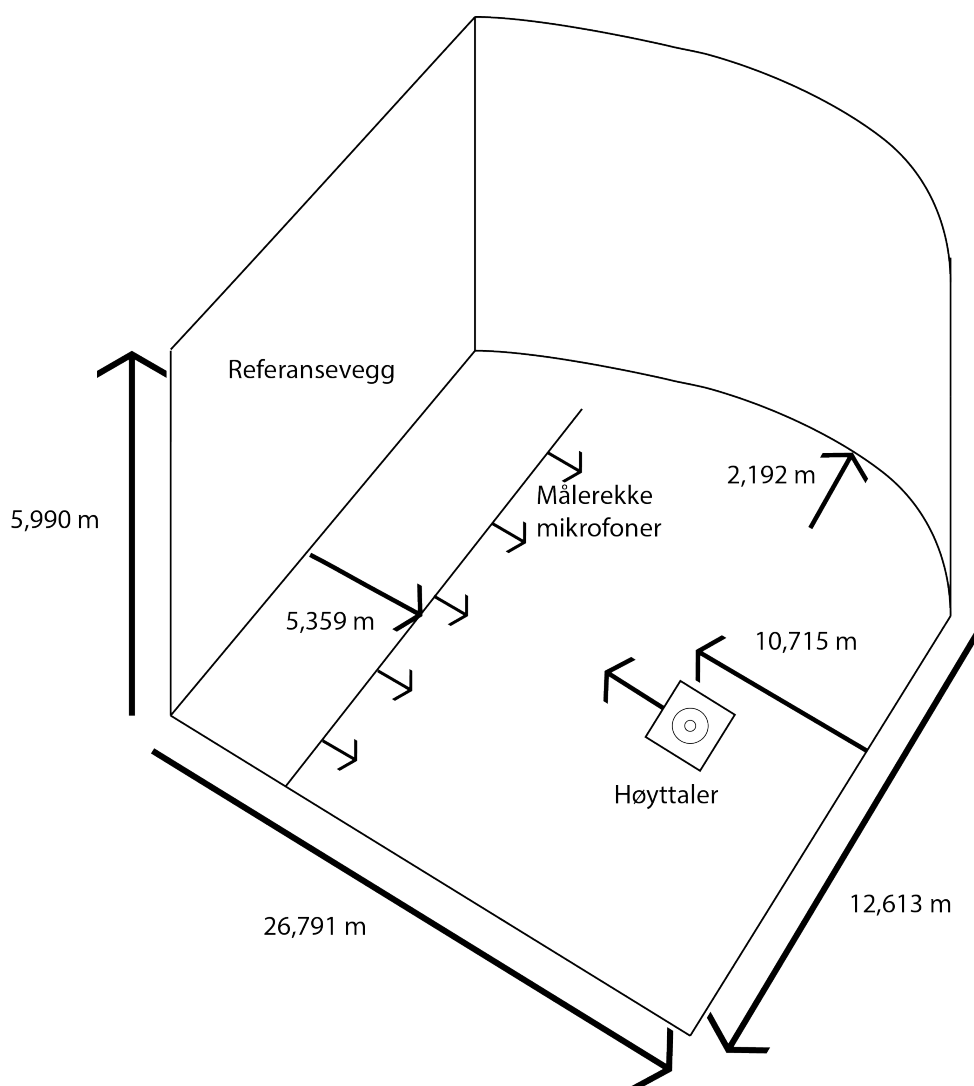


Figur 14: Spekulær refleksjon fra referanseveggen for måling og simulering, oktavbåndsfiltret med 1000 Hz som senterfrekvens med kurvetilpasning med kun 30 måleposisjoner.

5 Målinger i gymsal

5.1 Målerommet

Etter å ha utviklet en metode for behandling av impulsresponsdata ble det gjort nye målinger i et mye større rom for å benytte metoden i praksis og undersøke feilkilder og muligheter ved metoden. Til dette formålet ble Armfjeldsalen i idrettsbygget på Gløshaugen valgt. Dette er en middels stor gymsal med en stor grad av glatte og harde overflater. Tre av veggene er bekledd med glatte treplater, den kurvede veggen er bekledd med et vertikalt trepanel med små glipper mellom hvert panel, på gulvet er det parkett og taket har treplater med masse små runde hull.



Figur 15: Måleoppsett i Armfjeldsalen i idrettsbygget på Gløshaugen

5.2 Utstyr benyttet

For å kunne spille litt høyere enn i forrige målerom ble det benyttet en større høyttaler. Det ble benyttet en aktiv monitorhøyttaler produsert av Genelec Oy av typen 1029A. Siden dette er en aktiv høyttaler trengte en ikke lenger noen effektførsterker. Det gir denne utstyslista:

Produkt	Produsent	Modell/Type
Bærbar datamaskin	Asus	
Høyttaler	Genelec Oy	1029A Active Monitor (snr 029A039532)
Mikrofonmembran	BSWA Technology Company Limited	Type MP 216 frittfelt (snr 4501095)
Mikrofonkropp	BSWA Technology Company Limited	SM 4000 phantom-drevet
Eksternt lyd kort/phantom power forforsterker	M-Audio	Mobile Pre USB (snr 071G159A12125)
Laser avstandsmåler	Bosch GmbH	PLR 30 (snr 887514578)

5.3 Målinger

Det ble gjort tre måleserier totalt. En måleserie ble gjort med helt tom sal. Dette ble gjort for å ha en referansemåleserie som en kan sammenligne de to andre måleseriene med.

I den andre måleserien ble det plassert ut gymnastikkmatter langs referanseveggen. Disse var 1,88 m lange, 1,0 m brede og 5 cm tykke. De ble støttet opp mot veggen med den korteste siden pekende oppover og satt inntil hverandre langs veggen midt på referanseveggen. Det ble en 2 cm glippe mellom hver matte. For å begrense at mattene vinkler lyden oppover på grunn av vinkelen de står i ble de forsøkt plassert så langt inntil veggen som det var mulig uten å risikere at de falt ned.

Vinkelen de sto i i forhold til veggen ble beregnet til 10,5 grader.



Figur 16: Armfjeldsalen med matter plassert mot referanseveggen

I den siste måleserien ble det satt ut ulike gjenstander med den hensikt å prøve å spre lyden i forskjellige retninger. Det var kun harde gjenstander som ble satt ut.



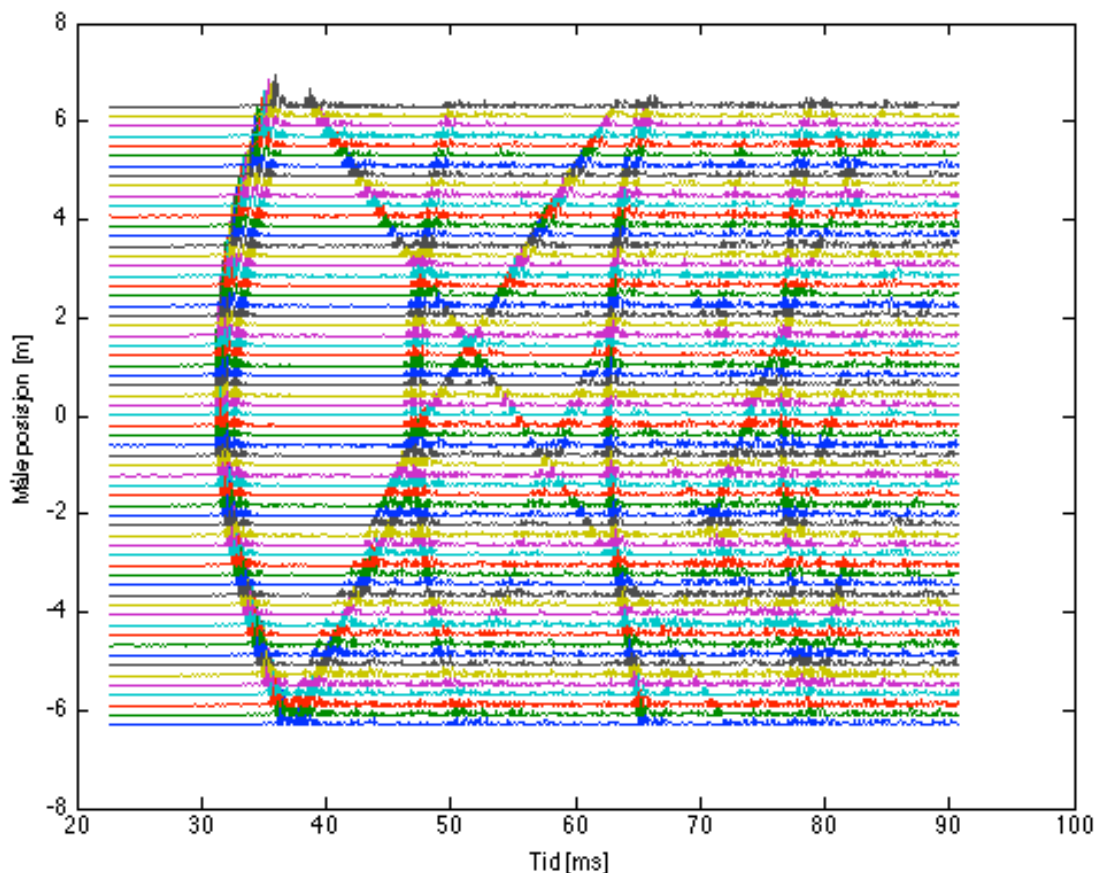
Figur 17: Armfeldsalen med spredegjenstander plassert langs referanseveggen

Impulsresponsmålingene ble foretatt ved hjelp av WinMLS på samme måte som beskrevet i avsnitt 3.3. Målelinja for mikrofonen ble lagt ca $1/5$ romlengde fra referanseveggen. Høytaleren ble plassert ca $3/5$ romlengde fra referanseveggen. For å sørge for god nok oppløsning for bølgefrontene ble det foretatt 63 impulsresponsmålinger per måleserie. Sinus-sveipet ble satt til 15 sekunder per impulsresponsmåling og nedre grense for sinus-sveipet ble satt til 125 Hz. Punktprøvingsfrekvensen var 44100 Hz og oppløsningen var på 24 bit.

6 Resultat

6.1 Tom sal

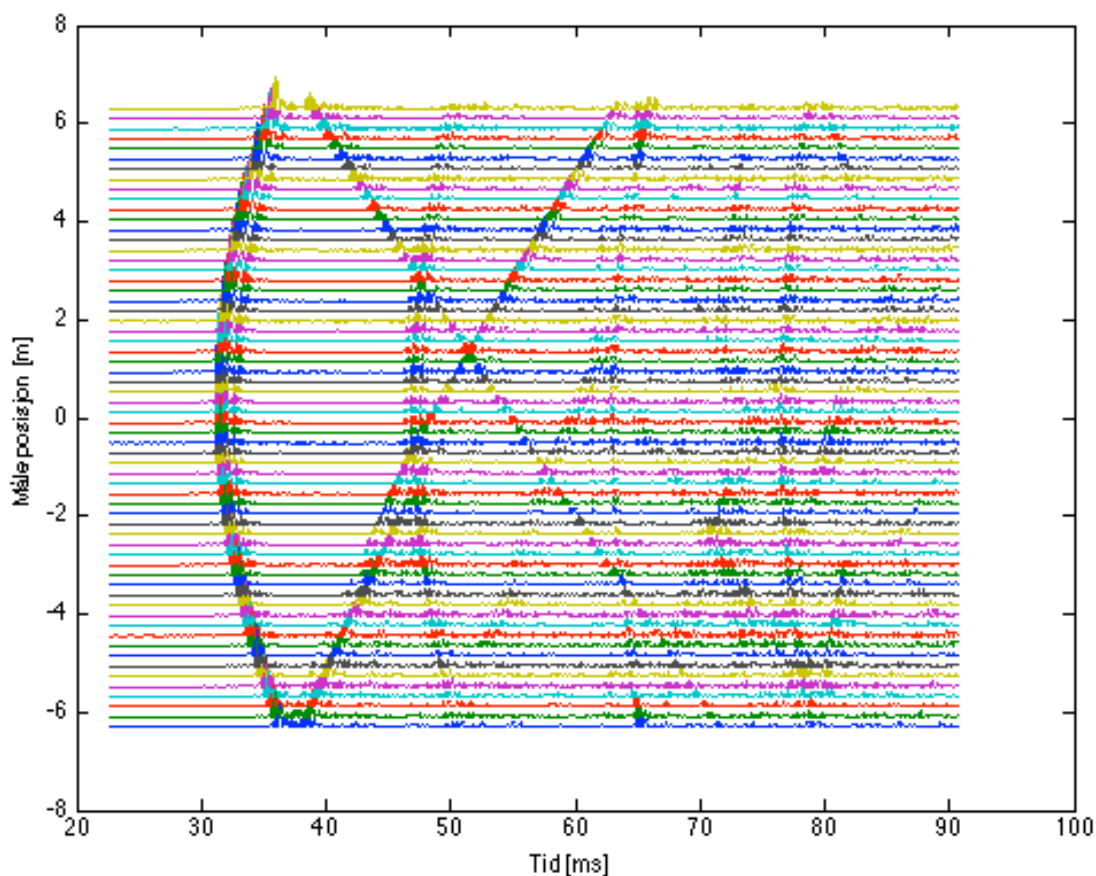
Ettersom det ble gjort 3 forskjellige måleserier som alle ble filtrert i 6 ulike oktavbånd kommer ikke alle figurer over impulsresponsen i rommet til å bli presentert her. En figur fra hver måleserie som er oktavbåndsfiltrert med 1000 Hz som senterfrekvens kommer til å bli presentert, samt at den simulerte impulsresponsen for det tilsvarende oktavbåndet kommer til å bli presentert. Videre kommer amplituden langs rommet for den reflekterte bølgen fra referanseveggen til å bli presentert for samtlige oktavbånd. Amplitudekurvene med kurvetilpasning for de ulike måleseriene kommer til å bli presentert i en samlet figur for hvert oktavbånd slik at de kan sammenlignes direkte.



Figur 18: Impulsrespons i tom sal, oktavbåndsfiltrert med 1000 Hz som senterfrekvens

Likning 4.1 gir at direktelyden skal ankomme etter $\approx 31,2$ ms. Dette stemmer bra med figuren. Refleksjonen fra referanseveggen skal komme etter $\approx 62,5$ ms. Dette stemmer også bra med figuren. En ser dermed at den reflekterte bølgen fra referanseveggen vil studere som refleksjon nummer to. Den første refleksjonen er refleksjonen fra taket.

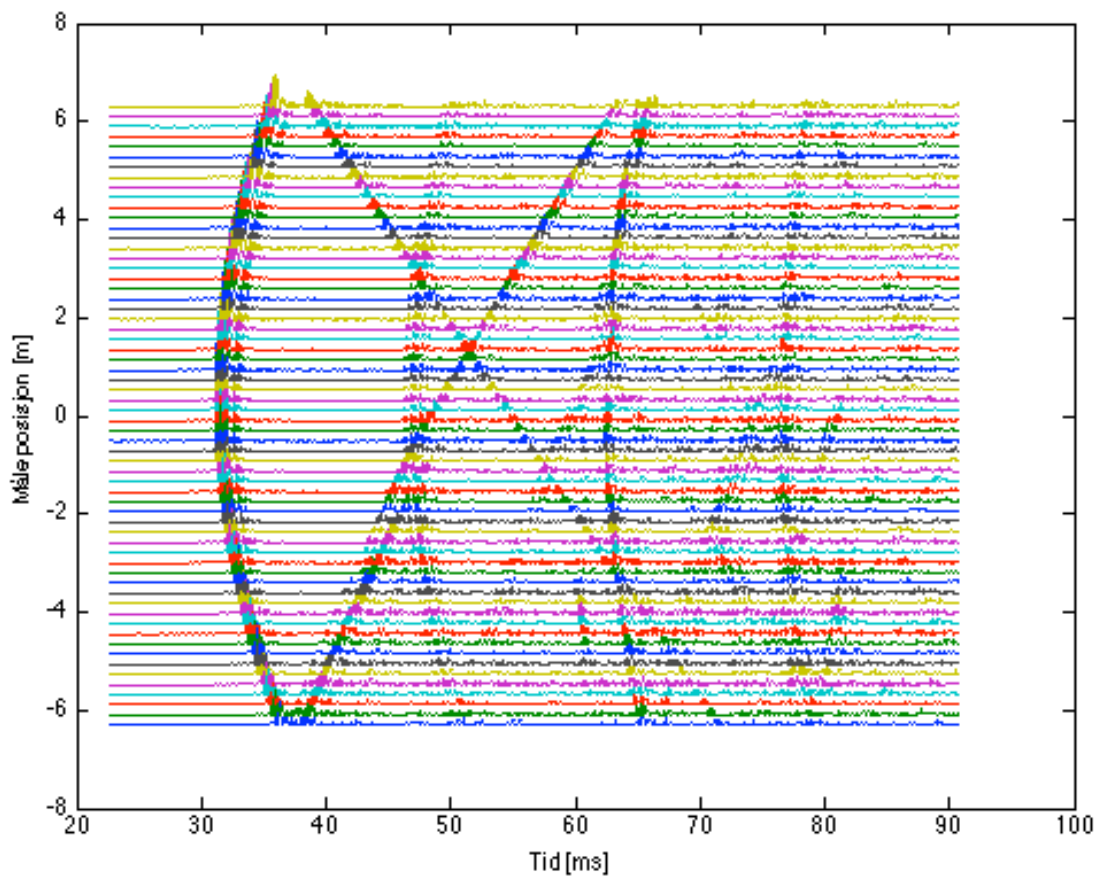
6.2 Ut plasserte matter



Figur 19: Impulsrespons med matter plassert mot referanseveggen, oktavbåndfiltrert med 1000 Hz som senterfrekvens

En ser her at energien for refleksjonen fra referanseveggen har blitt kraftig redusert, sammenlignet med figur 18. Det kan også se ut som om det har kommet to refleksjoner rett etter hverandre fra denne veggen. Dette kan skyldes at noe av energien blir spredt på grunn av vinkelen mattene er plassert i. Ute ved kantene av salen har ikke energien blitt dempet i like stor grad som på midten, der mattene er plassert.

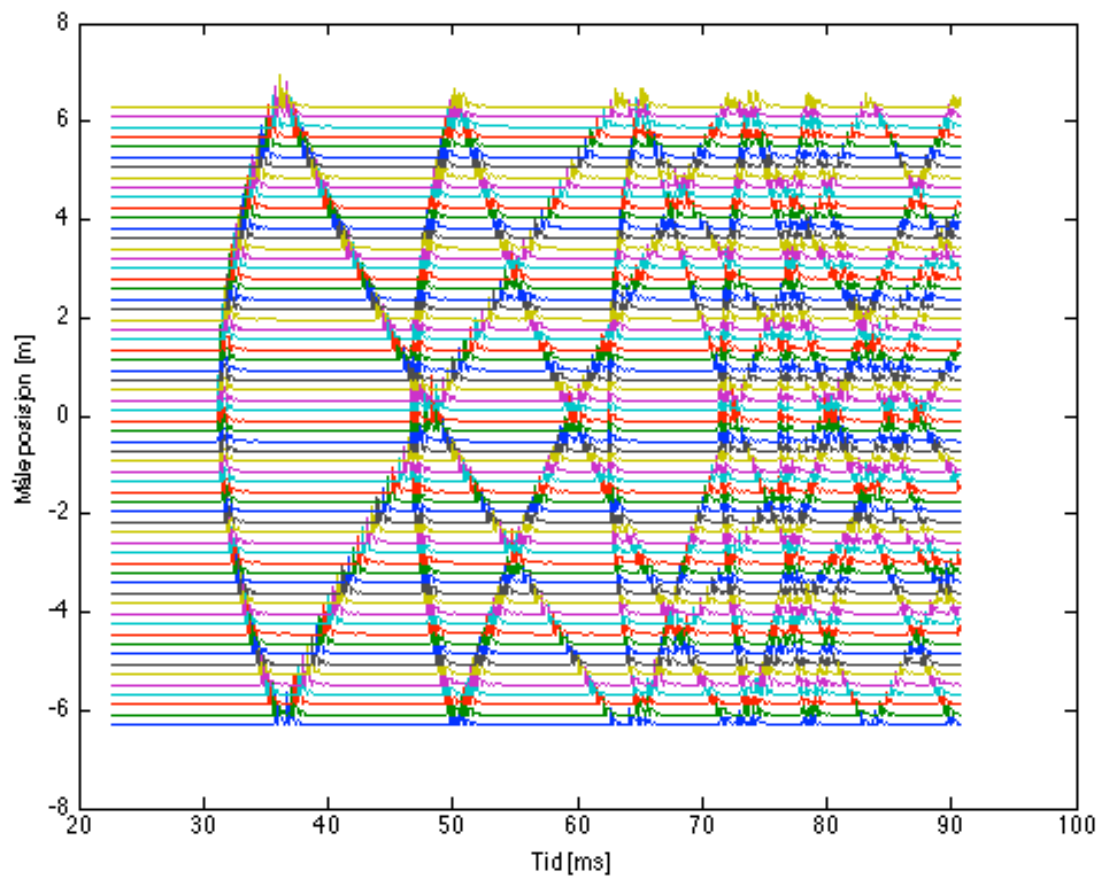
6.3 Ut plasserte spredegenstander



Figur 20: Impulsresponsen med ulike gjenstander plassert langs referanseveggen, oktavbåndfiltrert med 1000 Hz som senterfrekvens.

En ser her at energien i refleksjonen fra referanseveggen er redusert i noen grad, men ikke i så stor grad som det som var tilfelle når mattene stod utplassert.

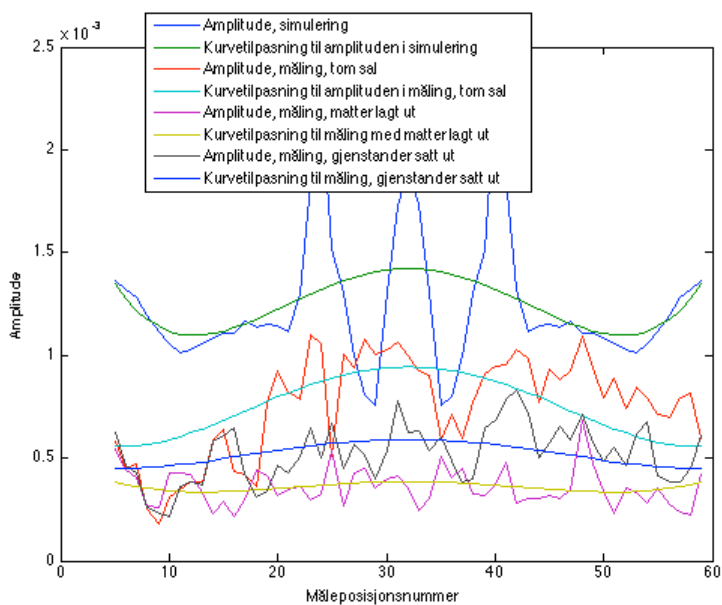
6.4 Simulering



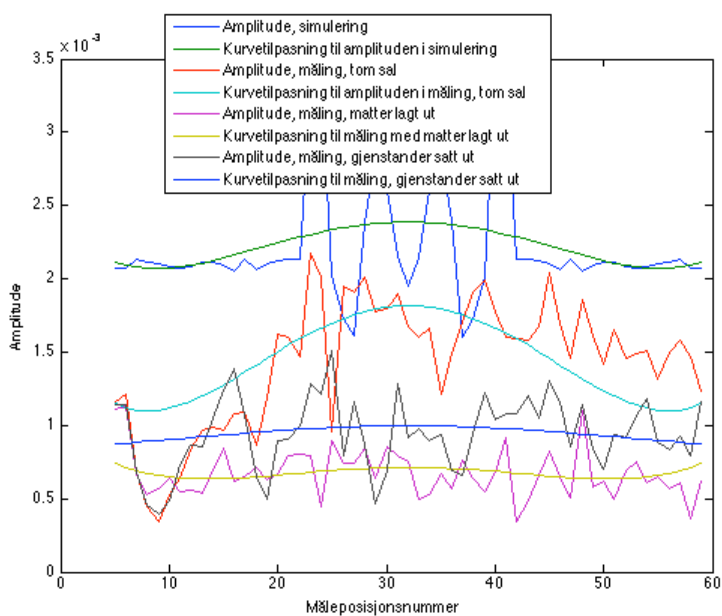
Figur 21: Simulering av impulsresponsen, oktavbåndsfiltrert med 1000 Hz som senterfrekvens.

I simuleringen er det gjort en tilnærming der hvor den kurvede veggen har blitt simulert som en rett vegg på lik linje med den andre sideveggen. Dette ble gjort for enkelthets skyld siden det ikke har noen betydning for resultatet siden det er kun direkte lyden og refleksjonen fra referanseveggen som skal analyseres. En kan se at simuleringen stemmer godt overrens med figurene fra målingene.

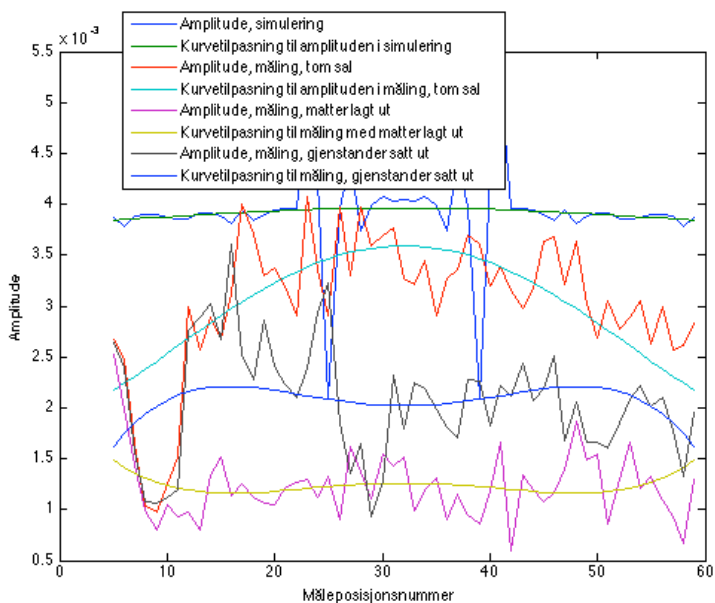
6.5 Samlede resultater



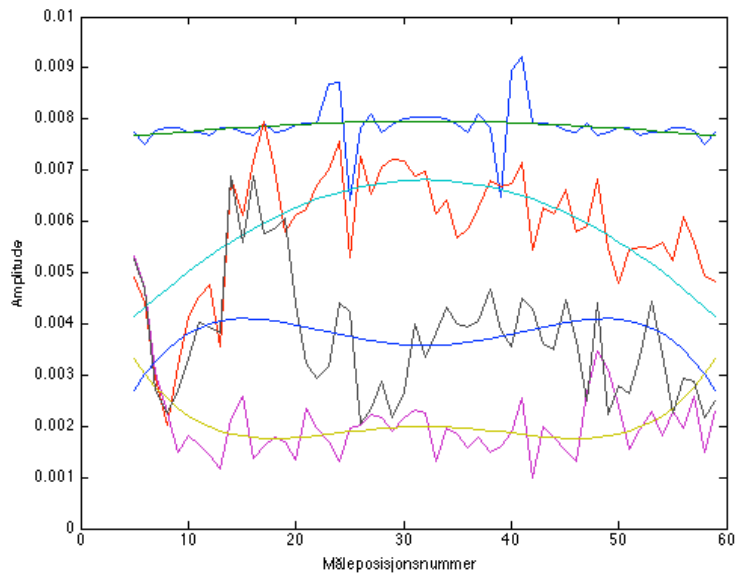
Figur 22: 250 Hz senterfrekvens



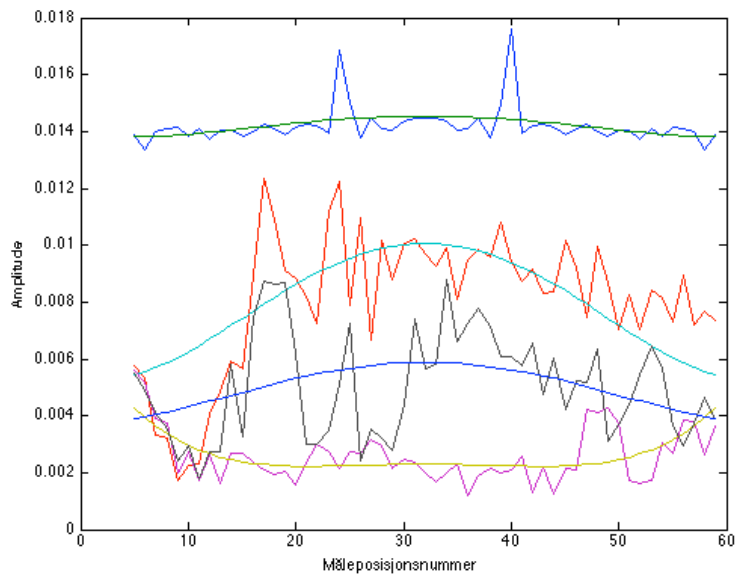
Figur 23: 500 Hz senterfrekvens



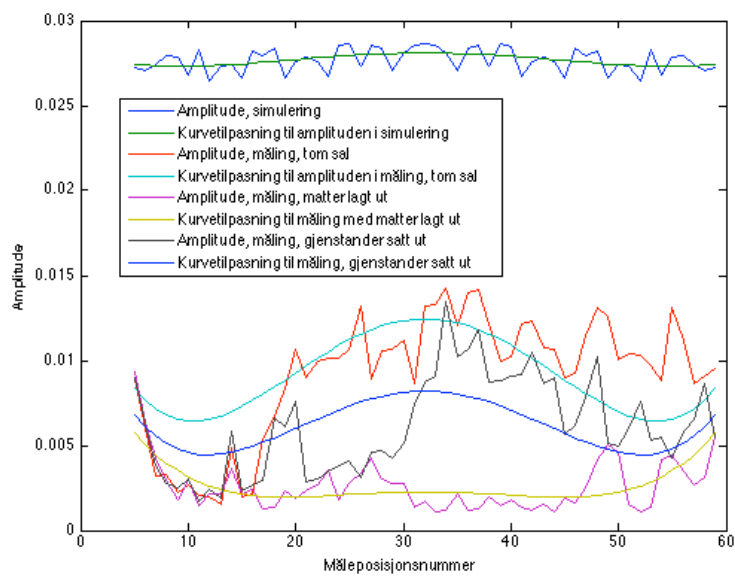
Figur 24: 1 kHz senterfrekvens



Figur 25: 2 kHz senterfrekvens



Figur 26: 4 kHz senterfrekvens



Figur 27: 8 kHz senterfrekvens

6.6 Etterklangstid og dempning

Etterklangstiden ble beregnet ut fra et utvalg av de impulsresponsene som ble målt i hver måleserie. Det ble hentet ut 13 impulsrespons med en meters mellomrom mellom hver måleposisjon. Etterklangstiden ble beregnet for disse 13 ved hjelp av WinMLS. Man beregnet deretter den gjennomsnittlige etterklangstiden for hvert oktavbånd ut i fra de 13 ulike verdiene for hvert oktavbånd.

	250 Hz	500 Hz	1 kHz	2 kHz	4 kHz	8 kHz
Tom sal	1,32	1,65	1,78	1,94	1,56	0,89
Matter	1,21	1,30	1,51	1,65	1,27	0,88
Gjenstander	1,30	1,46	1,66	1,78	1,39	0,86

Figur 28: Etterklangstid for hver av måleseriene

Ved å finne den gjennomsnittlige verdien for hver av kurvetilpasningene til amplitudekurvene for hver måleserie kan en finne et prosentvis tall for hvor mye dempning av den spekulære refleksjonen det er for henholdsvis den måleserien hvor matter er lagt ut og den måleserien hvor det er satt ut gjenstander. En ser da hvor stor prosentvis dempning det er i forhold til måleserien med tom sal.

	250 Hz	500 Hz	1 kHz	2 kHz	4 kHz	8 kHz
Matter	62	65	67	75	75	71
Gjenstander	31	33	32	37	34	19

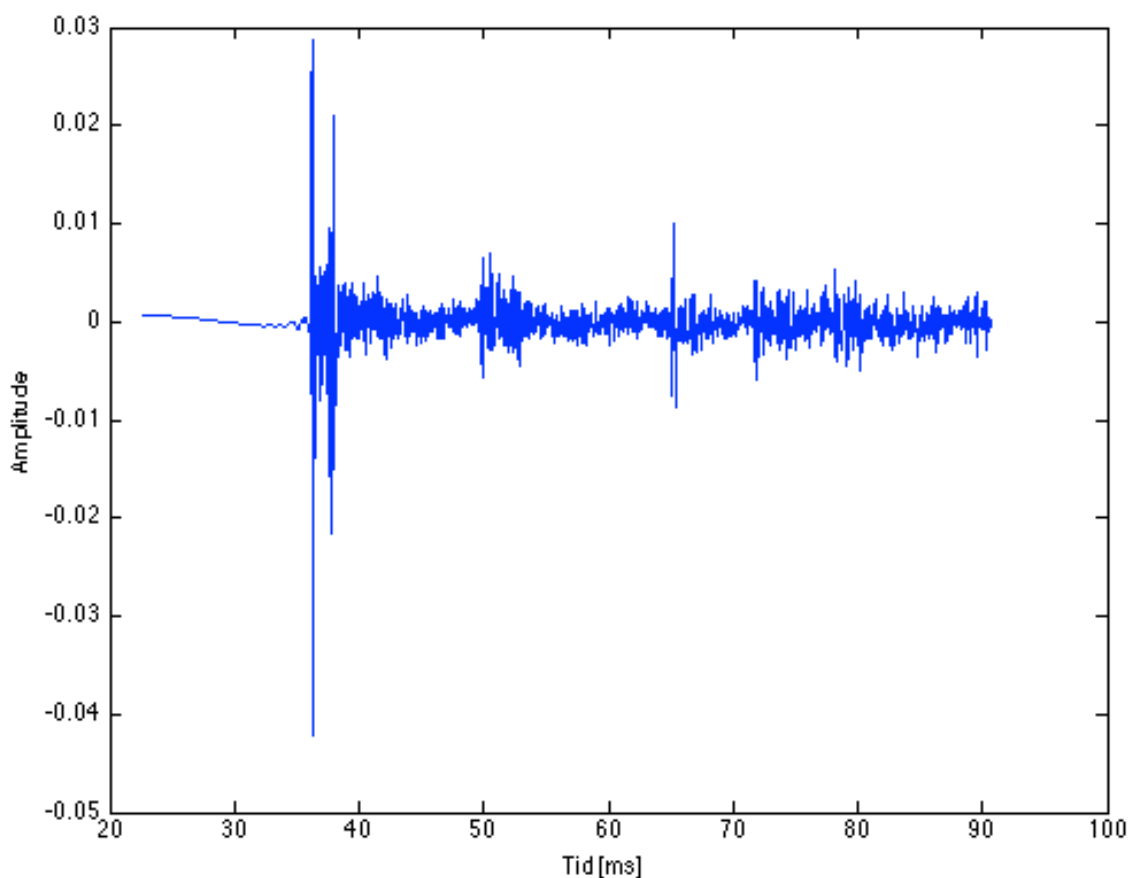
Figur 29: Prosentvis dempning av spekulær refleksjon

7 Vurdering av metoden

7.1 Diskusjon av feilkilder

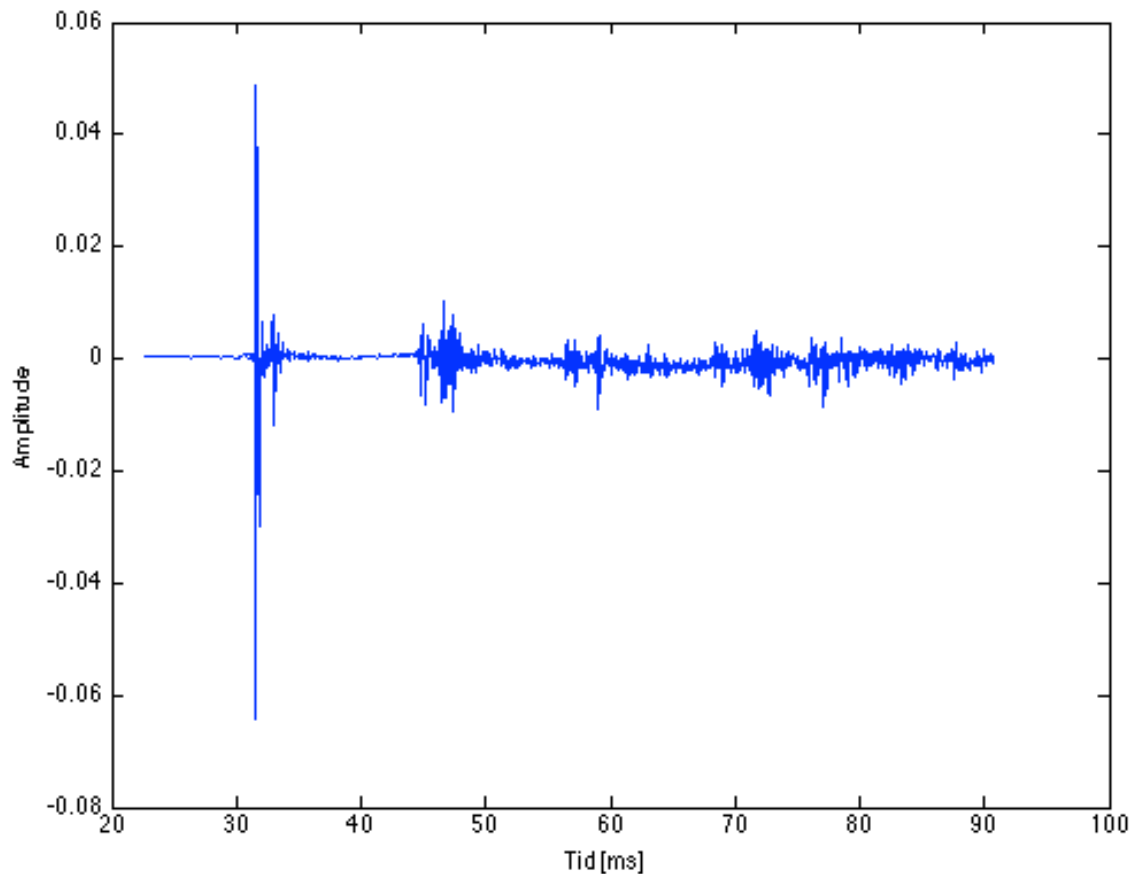
7.1.1 Signal- støyforhold

I en såpass stor sal må en gjøre noen tiltak for å sørge for godt nok signal- støyforhold. Lengden på sinus-sveipet ble øket til 15 sekunder per impulsresponsmåling. For å få mest mulig ut av disse 15 sekundene ble også sinus-sveipet kuttet ved 125 Hz slik at sveipet startet først ved denne frekvensen. Dette gjorde at høytaleren kunne spille høyere uten å risikere at en fikk forvrengning samt at en fikk ut mer energi i de oktavbåndene en var ute etter å undersøke. På grunn av salens størrelse ble det også valgt å benytte en større høytaler. Denne kunne spille vesentlig høyere enn det som var tilfelle for den lille sylinder-høytaleren. Ulempen med denne høytaleren er at den ikke er like omnidireksjonell som den som ble brukt i testingen av metoden. Sensitiviteten til mikrofonen ble også skrudd opp så høyt som mulig uten å risikere forvrengning eller klipping.



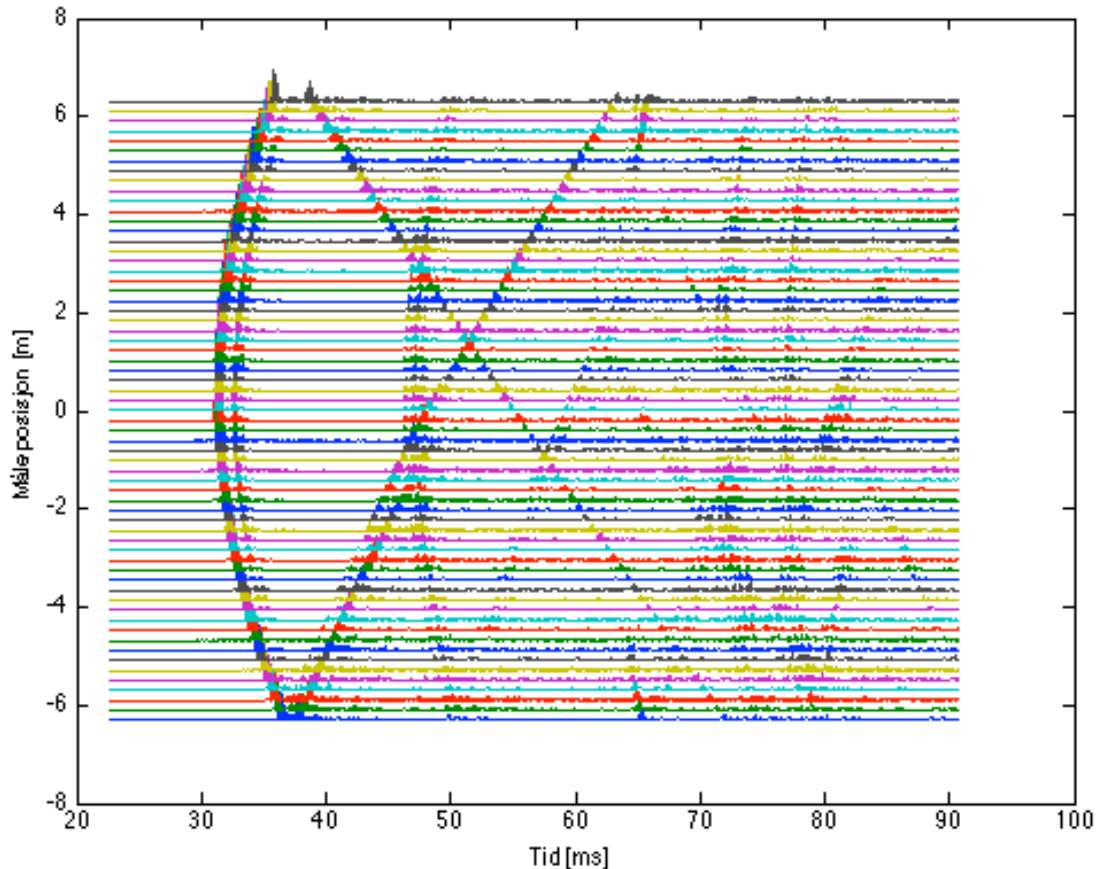
Figur 30: Impulsrespons for den ytterste mikrofonplasseringen (nr. 63), tom sal, oktavbåndsfiltret med 1000 Hz som senterfrekvens

Den refleksjonen en er ute etter å analysere i figuren over er den som kommer etter ca 65 ms. En kan se at denne skiller seg klart ut fra støyen. Utfordringen oppstår når en skal lese av en refleksjon som er sterkt diffusjonert.



Figur 31: Impulsrespons for den 40. mikrofonplasseringen, diffusjonerende matter er lagt ut, oktavgbåndsfiltrert med 2000 Hz som senterfrekvens

I figuren over kan en se at den refleksjonen en skal analysere omtrent forsvinner helt i støyen. Den kan nå bare så vidt skimtes ved tiden 63 ms. Hvis en analyserer 4 kHz-oktavgbåndet kan se dette.



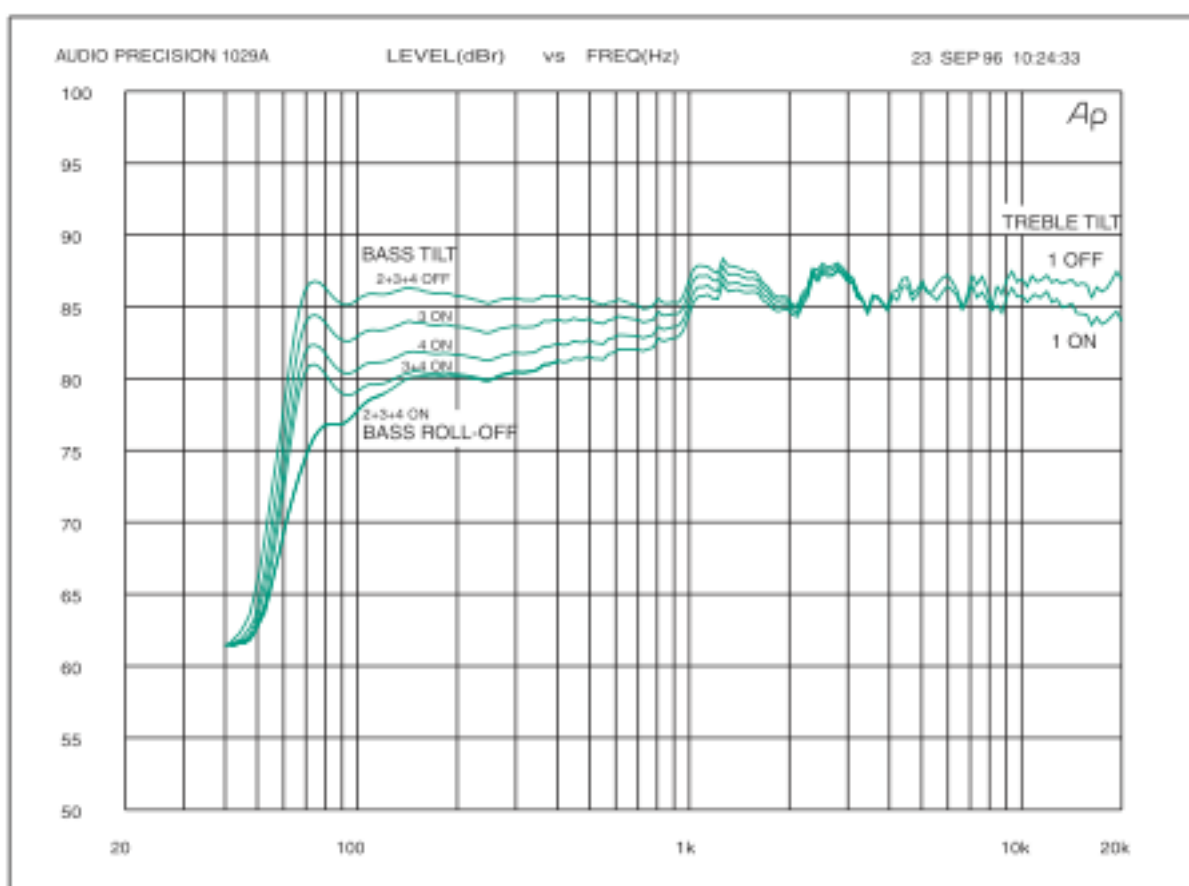
Figur 32: Impulsrespons, matter lagt ut langs veggen, oktavbåndfiltrert med 4000 Hz som senterfrekvens

I figuren over ser en at bølgefronten som ankommer ved ca. 65 ms forsvinner helt i støyen for alle måleposisjoner i midten, der hvor mattene har innvirkning gjennom spredning av lydbølgene. Konklusjonen er at det å få godt nok signal- støyforhold kan være vanskelig i noen situasjoner, spesielt for høye frekvenser og for veldig spredende elementer. Det hadde kanskje bedret situasjonen hvis en satte høyttaleren nærmere mikrofonene, men hvis en setter den for nært vil en kunne få en utfordring ved at høyttalerens direktivitet gjør at en vil slite med signal- støyforhold i de ytterste posisjonene.

7.1.2 Høyttalerens frekvensrespons

Høyttalerens frekvensrespons kan være en feilkilde. Dette kan medføre at enkelte oktavbånd blir forsterket eller dempet som følge av dette. Et ekstra stort problem vil det kunne medføre hvis en høyttaler har store utsving i frekvensresponsen innenfor et og samme oktavbånd. Da kan enkelte egenskaper ved et diffusjonerende materiale forsterkes mens andre andre igjen kan dempes. Dette kan gi merkbare feil i målingene. Siden den høyttaleren som ble brukt under målingene i denne oppgaven har en relativt flat frekvensrespons ble det ikke lagt vekt på å gjøre noen tiltak i forbindelse med dette. Men et inversfilter som retter ut frekvensresponsen kan relativt enkelt

implementeres i Matlab ved hjelp av den innebygde funksjonen *fir2*. Da må en først måle frekvensresponsen til høyttaleren i et ekkofritt rom. Dette kan en gjøre med WinMLS. Den impulsresponsen en får kan en da fouriertransformere ved hjelp av funksjonen *fft*. Den frekvensresponsen en får kan en da invertere. Det lønner seg kun å invertere det frekvensområdet som er relativt flatt. Dette er fordi de fleste høyttalere har en veldig stor demping i de laveste frekvensene og kan også ha det i de høyeste frekvensene. For å få et effektivt inversfilter bør en utelate å invertere disse frekvensområdene. Når en har invertert ønsket frekvensområde kan en finne filterkoeffisienter som implementerer dette filteret ved hjelp av funksjonen *fir2*. Det siste en da må gjøre er å konvolvare disse filterkoeffisientene med den impulsresponsen en vil inversfiltrere. Et slikt inversfilter ble laget av Peter Svensson og brukt under uttestingen av metoden, da den høyttaleren som ble brukt da ikke hadde like jevn frekvensrespons som den som ble brukt i målingene.

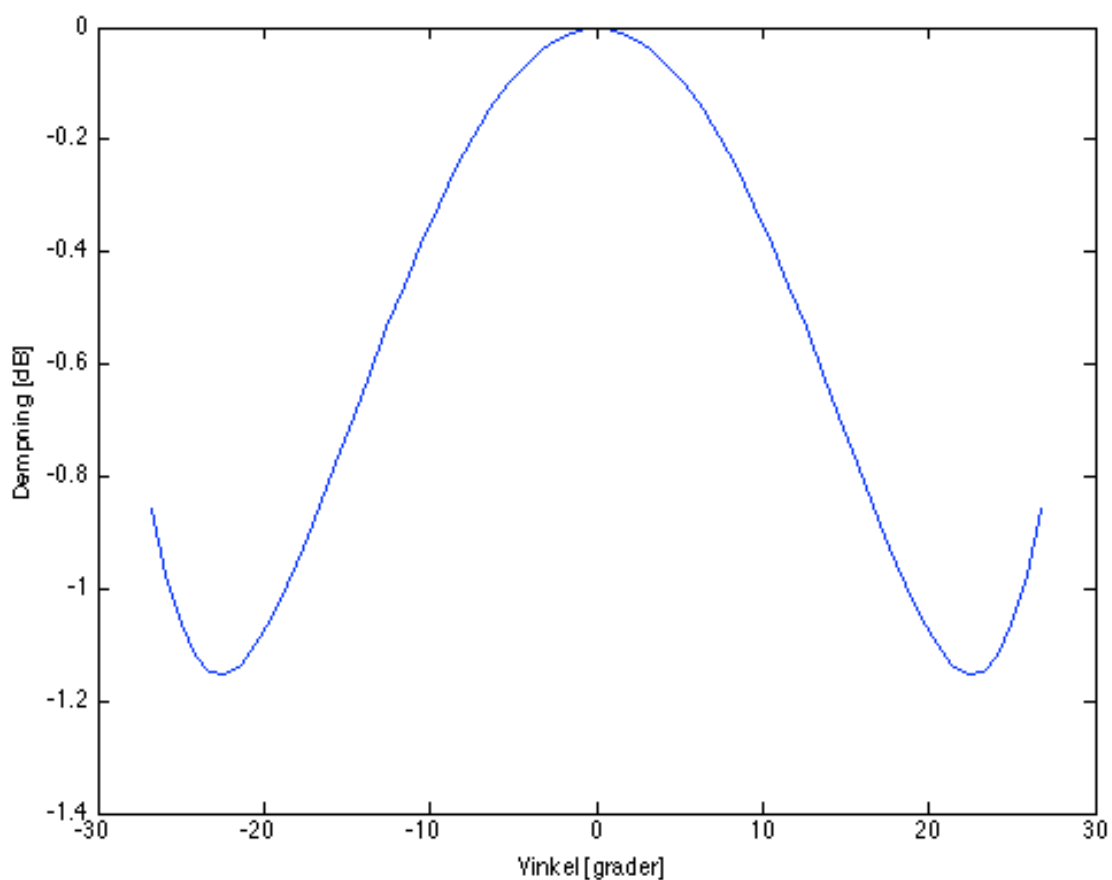


Figur 33: Frekvensrespons. Utdrag fra datablad for Genelec Active Monitor 1029A [9]

I figuren over ser en at en har et utslag på maks 3 dB i frekvensresponsen for frekvensområdet mellom 70 Hz og 20 kHz. Det er den øverste kurven som er gjeldende hvis en ikke har slått på noen av høyttalerens innebygde filtre. Siden alle høyttalere har individuelle forskjeller kan det likevel være en fordel å kontrollere hvilken frekvensrespons den aktuelle høyttaleren har. Dette ble ikke lagt vekt på i denne oppgaven.

7.1.3 Høytalerens direktivitet

Høytalere stråler ikke like kraftig i alle retninger. Så godt som alle høyttalere stråler mest rett fram. Intensiteten høyttaleren avgir avtar dess større vinkel en får mellom lytteposisjon og senterlinja, som er den linja som går rett framover den veien høyttaleren peker. For å få en tilnærmet verdi på hvor stor innvirkning høyttalerens direktivitet har på resultatet ble dette undersøkt nærmere i Matlab. Ved hjelp av funksjonen *atan* ble det laget en vektor som regnet ut hvor stor vinkelen er mellom senterlinja og mottakerposisjonen langs hele den linja hvor en har mikrofonposisjoner. Ved å finne forholdstallet mellom kurvetilpasningen til den målte direktelyden og kurvetilpasningen til den simulerte direktelyden, og konvertere dette til dB fant man en tilnærmet verdi for hvor stor dempningen er i dB langs hele målelinja som følge av høyttalerens direktivitet.



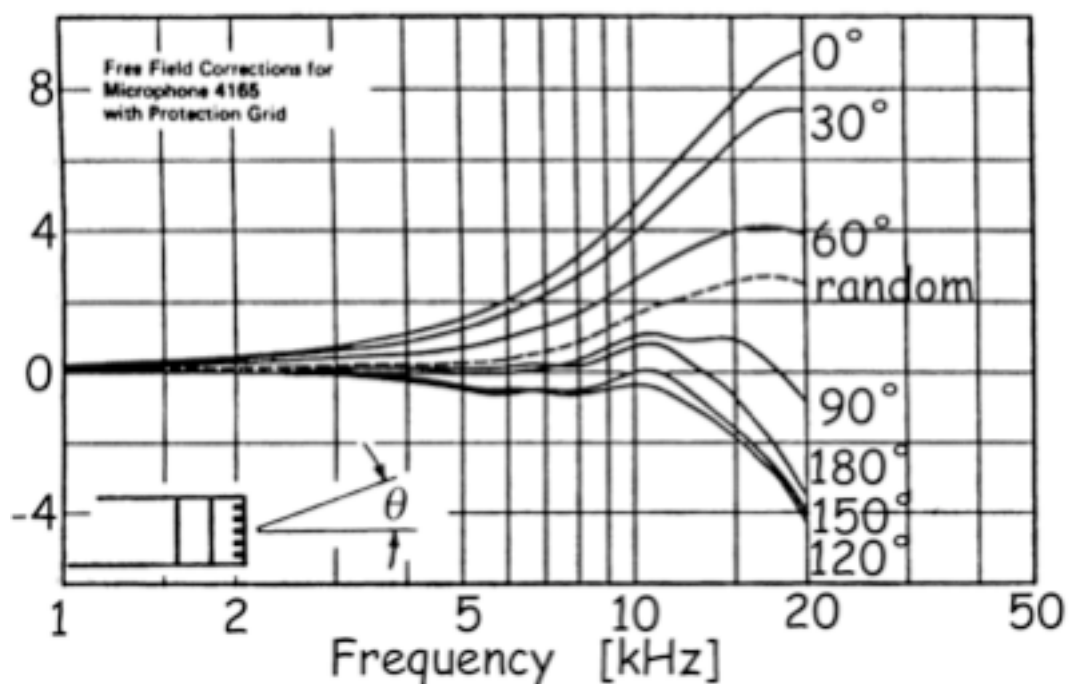
Figur 34: Dempningen som følge av høytalerens direktivitet i oktavgbåndet med 250 Hz som senterfrekvens

En ser fra figuren over at dempningen i de ytterste måleposisjonene er tilnærmet lik 1,18 dB. Dette er ikke så veldig mye. Dette skyldes i stor grad at siden høyttaleren sto relativt langt unna mikrofonrekka, så ble den maksimale vinkelen ganske liten. Dette har begrenset denne feilkilden i stor grad. Dempningen øker noe for oktavgbånd lenger opp i frekvens-spekteret. For 8 KHz-oktavgbåndet er dempningen maksimalt 2,3 dB i den ytterste posisjonen. Dette er heller ikke så ille. Likevel kan en sannsynligvis forklare hvorfor alle målinger har en dempning av amplituden i de ytterste måleposisjonene. Noe av årsaken er sannsynligvis at det er lengre vei å gå for lydbølgen og

dermed også større luftabsorpsjon, men sannsynligvis kan noe av dempningen også tilskrives høytalerens direktivitet. Det er derimot veldig usannsynlig at dempningen blir mindre helt ytterst slik figuren viser. Dette skjer sannsynligvis som følge av at kurvetilpasningen er basert på et polynom. Det kan gi slike effekter.

7.1.4 Mikrofonens direktivitet

Alle mikrofoner med sirkulær membran har den samme høyfrekvensdirektiviteten. [3]



Figur 35: Direktiviteten til mikrofoner med sirkulær membran gitt i dB som funksjon av frekvens.

Av figuren over ser en at direktiviteten øker relativt kraftig for enkelte infallsvinkler ved høye frekvenser. Under målingene sto mikrofonen i en vinkel på 60 grader ned mot gulvet. Det vil si at det er den kurven for 60 grader som gjelder for målingene som ble gjort. Det viser at direktiviteten er relativt lav opp til 5 kHz, men øker monotont opp til ca 4 dB forsterkning ved 15 kHz. Det er dermed først og fremst 8 kHz-oktavbåndet som blir berørt av denne direktiviteten under målingene.

7.2 Metodens bruksområder

Siden en allerede har funnet et forholdstall over hvor stor dempning en har av den spekulære refleksjonen for hver av måleseriene har en funnet et tilnærmet tall for spredningsfaktoren. Problemstillingen ligger i å finne ut hvor stor del av denne dempningen som eventuelt er absorpsjon. En mulig mulighet er å benytte etterklangstidsmålingene for tom gymsal til å beregne hva den gjennomsnittlige absorpsjonsfaktoren er for rommet i de ulike oktavbåndene ved hjelp av formel 2.13. Det er viktig at en tar hensyn til luftabsorpsjonen når en har å gjøre med et så stort rom som dette. I hvert fall for høye frekvenser. En burde kanskje ha gjort etterklangstidsmålinger jevnt fordelt i hele salen og ikke bare langs den linja hvor det har blitt målt i denne oppgaven. Det er også en mulighet for at det er en sen flutter-ekko-effekt som er grunnen til at en først får så lang etterklangtid. Dette kan være forklaringen på at etterklangstiden blir så mye kortere bare ved å sette opp 4 gymnastikkmatter langs veggen.

7.3 Videre arbeid

Dette kunne vært interessant å kunne kombinere denne metoden med en metode for å in-situ beregne absorpsjonsfaktor. En kunne også effektivisert metoden kraftig hvis en hadde tilgang på for eksempel 5 relativt like mikrofoner som en kunne kalibrert mot hverandre. Da hadde målingene gått mye raskere, og en kunne foretatt mange flere målinger. Det er også mange feilkilder som en må ta hensyn til og prøve å begrense hvis en skal jobbe videre med dette. En må få målt direktivitet for høyttaler og kontrollere dette opp mot den beregnede verdien en fikk nå i Matlab. Kanskje kunne en ha etterbehandlet den impulsresponsen en måler ut i fra hva vinkelen til høyttaleren er i hver måleposisjon. Tilsvarende burde en legge inn en dempning for å kompensere for mikrofonens direktivitet når en kommer opp i høye frekvensbånd. En må også vurdere tiltak for å bedre signalstøyforhold, siden det viste seg nå at dette ble akkurat litt for dårlig innenfor de to høyeste oktavbåndene. Du kunne også vært interessant å studert i detalj hva som skjer helt inne ved veggen akkurat når bølgen blir reflektert og eventuelt spredd.

8 Konklusjon

Fokuset til denne oppgaven har vært å undersøke metoder for å estimere absorpsjonsfaktorer og spredningsfaktorer og da spesielt spredningsfaktorer. Det har også vært fokusert på å undersøke hvilke feilkilder en må ta hensyn til når en gjennomfører slike målinger. Det har vist seg å være mange feilkilder som en må ta hensyn til.

Oppgaven har gitt flere spørsmål en svar. Det er tydelig at dette er et arbeidsområde hvor det er grunnlag for mye forskning. Men ved å benytte seg av de erfaringene som har blitt gjort i denne oppgaven vil en forhåpentligvis komme et skritt videre neste gang.

Bibliografi

- [1] Marshall Long (2006). *Architectural Acoustics*. Elsevier Academic Press.
- [2] Tor Erik Vigran (2008). *Building Acoustics*. Taylor & Francis.
- [3] Peter Svensson (2010). *Lecture Notes, Audio Technology TTT 4170*. It's Learning NTNU.
- [4] Asbjørn Krokstad (1999). *Akustikk for Ingeniører*. Institutt for Telematikk, NTNU.
- [5] Lawrence E. Kinsler; Austin R. Frey et. al. (1982). *Fundamentals of Acoustics, Third Edition*. John Wiley & Sons, Inc.
- [6] John G. Proakis; Dimitris G. Monolakis (1996). *Digital Signal Processing. Principles, Algorithms, and Applications. Third Edition*. Prentice-Hall, Inc.
- [7] Emanuël Habets (2010). *Room Impulse Response Generator for Matlab, version 2.0*. http://home.tiscali.nl/ehabets/rir_generator.html
- [8] Karl Rottmann (2003). *Matematisk Formelsamling*. Spektrum Forlag. 7. opplag.
- [9] Genelec Oy (2003). *Data Sheet Genelec 1029A. Bi-amplified Monitoring System*

A Vedlegg

A.1 Matlabkode

```
%Input values:
measurements = ('measurements/armfeld/armf');
%input the folder that holds the measurements. Do not include the '/' in
%the end of the string

rootname = 'ir';
%Set the root name of the files in the measurements folder

extension = '.wmb';
%Set the extension name of the files in the measurements folder

delayfile = ('measurements/forsinkelse.wmb');
%input where the measurement of the delay due to the sound card is placed.
%If such a delay is not measured, set the input to 0

Fs = 44100;
%input the sample frequency for the measurements

intpol = 8;
%input the order of the interpolation wanted

a = 1000;
%input the desired starting sample for the samples to be viewed in the
%diagrams

b = 4000;
%input the desired ending sample for the samples to be viewed in the
%diagrams

octavebandfreq = 250;
%input the center frequency of the desired octaveband in Hz

wavefrontnumber = 0;
%input the wavefront which is to be examined. 0 gives the direct sound,
%1 gives the first reflection, and so on

readwmbfiles = 1;
%input whether the wmb-files is supposed to be read or not. Is initially to
%be set to 1. After having first executed this file once, this varibale can
%be set to 0 to save the memory of the computer

saveclickpos_meas = 1;
%Saves the clickdata if = 1. It's supposed to be given the value of 1 the
%first time you execute the given wavefront. Then it should be set to 0
%so you don't have to click the clickdata one more time

%input values for the simulation:
dosimulation = 1;
```

```



```

```

    octavebandname = [num2str(octavebandfreq), ' Hz'];
else
    octavebandname = [num2str(octavebandfreq/1000), ' kHz'];
end
% determines how the name of the octaveband is to be displayed in the
% diagrams of this program

filelength=length(loadimp([measurements,'/',rootname,'1',extension]));
%calculates the number of samples that are in a single wmb-file

T = 1/Fs;
%calculates the sample time

t = ((0:nsample-1)*T).*1000;
%calculates the time vector in milliseconds

directory = ([measurements,'/']);
%sets the name for the directory that holds the measurements

[delaypos] = finddelay(delayfile);
%finds how many samples the delay consists of

%load inversfilter/inversfilter.mat

inverse = 0;
filtercoeff = 0;

[big_ir,counter] =
wmbfiles_reader(readwmbfiles,directory,rootname,extension,filtercoeff,inverse,del
aypos,amount,filelength);
%reads the wmb-files

[octavebandfiltered_big_ir] = ovtavebandfilter(octavebandfreq,Fs,big_ir );
%filters the measurements in the desired octave band

[upsampled_big_ir] = upsampler(intpol,octavebandfiltered_big_ir,a,b,counter);
%upsamples the measurements according to the desired upsampling factor and
%gets the desired part of the upsampled measurements to be used in the
%diagrams

[octavebandfiltered_sim] = ovtavebandfilter(octavebandfreq,Fs_sim,simulation );
%filters the simulation in the desired octave band

octavebandfiltered_sim = octavebandfiltered_sim((a*intpol):
(b*intpol)-1,1:amount);
%gets the desired part of the octavebandfiltered simulations to be used in
%the diagrams

[plotable_upsampled_big_ir,shiftmatrix,scalingfactor] =
makeplotable_big_ir(upsampled_big_ir,L);
%gets a plotable big_ir and a shiftmatrix

figure(1)
plot(linspace(t(a),t(b),size(plotable_upsampled_big_ir,
1)),abs(plotable_upsampled_big_ir) + shiftmatrix);
title(['Impulserespons, ',octavebandname,' octaveband']);
xlabel('Time [ms]');
ylabel('Measurement position [m]');

```

```

[clickpos] = getclickpos(0,saveclickpos_meas,wavefrontnumber,upsampled_big_ir);
%gets the desired clickpos according to the wavefrontnumber and if clickpos
%is not present the clickpos is saved

[xx,wavefrontpeaklocation_meas,wavefrontamplitude_meas] =
getwavefrontamplitude(0,upsampled_big_ir,clickpos);
% gets the loacation of the peak amplitude in the chosen wavefront for each
% measurement and the peak amplitude in the chosen wavefront in each
% measurement and xx vector

[plotable_octavebandfiltered_sim,shiftmatrix] =
makeplotable_big_ir(octavebandfiltered_sim,L);
%gets a plotable big_ir and a shiftmatrix

figure(4)
plot(linspace(t(a),t(b),size(plotable_octavebandfiltered_sim,
1)),abs(plotable_octavebandfiltered_sim) + shiftmatrix);
title(['Impulseresponse for the simulation in the ',octavename,' octaveband']);
xlabel('Time [ms]');
ylabel('Measurement position [m]');

[clickpos] =
getclickpos(1,saveclickpos_sim,wavefrontnumber,octavebandfiltered_sim);
%gets the desired clickpos according to the wavefrontnumber and if clickpos
%is not present the clickpos is saved

[xx,wavefrontpeaklocation_sim,wavefrontamplitude_sim] =
getwavefrontamplitude(0,octavebandfiltered_sim,clickpos);
% gets the loacation of the peak amplitude in the chosen wavefront for each
% measurement and the peak amplitude in the chosen wavefront in each
% measurement and xx vector

[curvefit_sim,curvefit_meas,relationnumber] =
makecurvefit(octavebandfiltered_sim,wavefrontamplitude_sim,wavefrontamplitude_meas,
wavefrontnumber,xx,octavebandfreq);

curvefit_measured = curvefit_meas;

curvefit_sim = curvefit_sim*relationnumber;
wavefrontamplitude_sim = wavefrontamplitude_sim*relationnumber;

figure(7)
plot(xx,wavefrontamplitude_sim,xx,curvefit_sim,xx,wavefrontamplitude_meas,xx,curvefit_meas);

if wavefrontnumber == 0
    title({'Maximum amplitude for the direct sound at the ';[octavename,'
octaveband with curve fit']});
else
    title({'Maximum amplitude for the front wall reflection at the ';
[octavename,' octaveband with curve fit']});
end
xlabel('Measurement position number');
ylabel('Amplitude');

figure(8)

[anglevector] = calcanglevector(xx,s,microphoneplacement,microphoneseparation);

plot(anglevector,20*log10(curvefit_meas./curvefit_sim));

```

```

    xlabel('Angle [degrees]');
    ylabel('Damping [dB]')
if wavefrontnumber == 0
    %title(['Directivity of loudspeaker for the direct sound at the
',octavename,' octaveband with curve fit']);
else
    %title(['Directivity for the front wall reflection at the ',octavename,'
octaveband with curve fit']);
end

figure(9)
centerindex = ceil(size(upsampled_big_ir,2)/2);
pr_sim = polyfit((xx-centerindex).^2,(wavefrontpeaklocation_sim+
(a-1)*intpol)*343.6/(Fs*intpol),2);
pr_meas = polyfit((xx-centerindex).^2,(wavefrontpeaklocation_meas+
(a-1)*intpol)*343.6/(Fs*intpol),2);
curvefitr_sim = pr_sim(3)+pr_sim(2)*(xx-centerindex).^2+pr_sim(1)*(xx-
centerindex).^4;
curvefitr_meas = pr_meas(3)+pr_meas(2)*(xx-centerindex).^2+pr_meas(1)*(xx-
centerindex).^4;

plot(xx,[(wavefrontpeaklocation_sim.'+(a-1)*intpol)*343.6/(Fs*intpol)
(wavefrontpeaklocation_meas.'+(a-1)*intpol)*343.6/
(Fs*intpol)],xx,curvefitr_sim,xx,curvefitr_meas);

```

```

function [ big_ir,counter ] =
wmbfiles_reader(readwmbfiles,directory,rootname,extension,filtercoeff,inverse,del
aypos,amount,filelength)
%WMBFILES_READER Makes the matrix that contains all the measurements
% The desired files from the folder that contains the measurements is
% collected in one single matrix.Each measurement is then convolved with
% the filtercoefficients of the inverse filter of the loudspeaker if such
% a filter is available. The delaytime of the soundcard is taken into
% account and therefore the first n samples is discarded.

if readwmbfiles == 1

    big_ir=zeros(filelength-delaypos+1,amount);
    counter = 1;

    for irnr = 1:amount
        filename = [directory,rootname,num2str(irnr), extension];
        [variable] = loadimp(filename);

        if irnr == 1
            desiredlength = length(variable);
        end

        if inverse == 1
            variable = conv(variable,filtercoeff);
            variable(1:floor(length(filtercoeff)/2)) = [];
        end

        variable = variable(delaypos:desiredlength);
        big_ir(:,counter) = variable;
        clear variable
        counter = counter+1;
    end

    save armf.mat
else
    load armf.mat
end
end

```



```

function [upsampled] = upsampler(intpol,big_ir,a,b,counter)
%UPSAMPLER Upsamples the desired big_ir matrix with the desired upsampling
%factor. This is done using an interpolation. The desired part of the
%upsampled measurements is also picked out according to the values of a and
%b

a = a*intpol;
b = b*intpol;

upsampled_z = zeros((size(big_ir,1))*intpol,counter-1);

for i = 1:size(big_ir,2)
    upsampled_z(:,i) = interp(big_ir(:,i),intpol);
end

upsampled = upsampled_z(a:b-1,1:counter-1);

end

```

```

function [simulation] = simulation_ir_exe(dosimulation, c, Fs_sim, L, s,
startingposition, nsample, beta, amount, microphoneplacement,
microphone separation)

if dosimulation == 1

    r = zeros(amount,3);
    for i = 1:amount;
        r(i,:) = [startingposition microphoneplacement(1)
microphoneplacement(2)];
        startingposition = startingposition+microphone separation;
    end

    [simulation] = rir_generator(c, Fs_sim, r, s, L, beta, nsample);
    simulation = simulation';

    save simulation_done.mat;

else
    load simulation_done.mat;
end

end

```

```
function [ octavebandfiltered ] = octavebandfilter(octavebandfreq,Fs,big_ir )
%OCTAVEBANDFILTER Filters each measurement in the desired octave band
% Each and every measurements in the big matrix containing all
% measurements is filtered in the desired octave band using a butterworth
% filter.

[Bbp,Abp] = butter(1,octavebandfreq*[1/sqrt(2) sqrt(2)]/(Fs/2));
octavebandfiltered = filter(Bbp,Abp,big_ir);

end
```

```

function [plotable_big_ir,shiftmatrix,scalingfactor] =
makeplotable_big_ir(big_ir,L)
%MAKEPLOTABLE_BIG_IR makes y-vector display distance in meter and
%minipulates big_ir to make the direct sound alike for every octaveband
%using a scalingfactor
% By using the width of the room contained in vector L, a shiftmatrix is
% made that makes it possible to plot each measurement in big_ir with an
% average amplitude that corresponds to the distance in meters from the
% centerposition. Also a scalingfactor is made in such a way that the
% direct sound wave is always displayed with the same amplitude whatever
% the octaveband that is used. Finally the big_ir is flipped in the
% left-right position as to display the first measurement on top and the
% last measurement in the buttom. Returns the manipulated big_ir and the
% shiftmatrix

y = zeros(1,size(big_ir,2));
y = linspace(-(L(1)/2),L(1)/2,size(big_ir,2));
shiftmatrix = zeros(size(big_ir));
for i = 1:size(big_ir,2)
    shiftmatrix(:,i)=shiftmatrix(:,i)+y(i);
end

%scalingfactor = (size(big_ir,2)*0.01)/max(abs(big_ir(:,(floor(size(big_ir,2)/
2))))))
scalingfactor = (L(1)/20)/max(abs(big_ir(:,1))%floor(size(big_ir,2))));
plotable_big_ir = zeros(size(big_ir));
for i = 1:size(big_ir,2)
    plotable_big_ir(:,i) = big_ir(:,i)*scalingfactor;
end

plotable_big_ir = fliplr(plotable_big_ir);

end

```

```

function [curvefit_sim,curvefit_meas,relationnumber] =
makecurvefit(big_ir,wavefrontamplitude_sim,wavefrontamplitude_meas,wavefrontnumber,xx,octavebandfreq)

centerindex = ceil(size(big_ir,2)/2);
p_sim = polyfit((xx-centerindex).^2,wavefrontamplitude_sim,2);
p_meas = polyfit((xx-centerindex).^2,wavefrontamplitude_meas,2);
curvefit_sim = p_sim(3)+p_sim(2)*(xx-centerindex).^2+p_sim(1)*(xx-centerindex).^4;
curvefit_meas = p_meas(3)+p_meas(2)*(xx-centerindex).^2+p_meas(1)*(xx-centerindex).^4;

if wavefrontnumber == 0
    relationnumber = curvefit_meas(ceil(length(xx)/2))/
curvefit_sim(ceil(length(xx)/2));
    savefile = ['relationnumber',num2str(octavebandfreq),'.mat'];
    save(savefile,'relationnumber');
else
    load(['relationnumber',num2str(octavebandfreq),'.mat']);
end

end

```

```

function [xx,wavefrontpeaklocation,wavefrontamplitude] =
getwavefrontamplitude(is_sim,big_ir,clickpos)
%GETWAVEFRONTAMPLITUDE finds the wavefront amplitude for the chosen
%wavefront along the line of microphones

% The xx and yy vectors are made from the big_ir and clickpos. A
% windowlength is then chosen and the clickpos is made to be in the
% center of this windowlength. The peak amplitude for each of this
% windows is then calculated and also the position of this peak value.
% The number of the start of the window minus 1 is then added to the peak
% position as to find where in the big_ir this position is found for each
% measurement. The xx and yy vectors is then returned along with the
% array holding the peak amplitudes and peak positions.

xx = [5:size(big_ir,2)-4];
yy = round(spline(clickpos(:,1),clickpos(:,2),xx));

windowlength = 400;
wavefrontpeaklocation = zeros(size(xx));
clickwindowmin = zeros(size(xx));
wavefrontamplitude = zeros(size(xx));
for ii = 1:length(xx)
    clickwindow = [yy(ii)-windowlength/2:yy(ii)+windowlength/2];
    clickwindowmin(ii) = yy(ii)-windowlength/2;
    [maxvalue,maxpos] = max(abs(big_ir(clickwindow,xx(ii))));
    wavefrontpeaklocation(ii) = maxpos + clickwindowmin(ii)-1;
    wavefrontamplitude(ii) = abs(maxvalue);
end

if is_sim == 0
    x = 3;
else
    x = 6;
end
figure(x);
plot(xx,yy);

end

```

```

function [clickpos] = getclickpos(is_sim,saveclickpos,wavefrontnumber,big_ir)

format long;
if saveclickpos == 1
    if is_sim == 0
        x = 2;
    else
        x = 5;
    end
    figure(x)
    mesh(abs(big_ir));
    view(45,90);
    disp(['Mark one wavefront with 7 clicks']);
    clickpos = ginput(7);
    if is_sim == 1
        name = ('sim');
    else
        name = ('meas');
    end
    savefile = ['clickpos_',name,num2str(wavefrontnumber),'.mat']
    save(savefile,'clickpos')
else
    if is_sim == 1
        name = ('sim');
    else
        name = ('meas');
    end
    load(['clickpos_',name,num2str(wavefrontnumber),'.mat'])
end

end

```

```
function [delaypos] = finddelay(file)
%FINNDELAY Determines whether there is done a measurement of the delay and
%if it is the function calculates the sample number where the delay of the
%soundcard finishes

if file == 0
    delaypos = 1;
else
    delay=loadimp(file);
    [top,delaypos] = max(abs(delay));
end

end
```



```

function [anglevector] =
calcanglevector(xx,s,microphoneplacement,microphoneseparation)
%CALCANGLEVECTOR Calculates the anglevector
% Calculates the anglevector by using arctan. The opposite is then the
% distance from to mid microphone to the microphone beeing measured and
% the adjacent is the distance between the source and the mid placed
% microphone in the line of microphones. The calculation is done for all
% the microphones in the line. The result is then converted to degrees.

anglevector = zeros(size(xx));

left_boarder = -floor(length(xx)/2)*microphoneseparation;
position = left_boarder;
mid_distance = s(2)-microphoneplacement(1);

for i = 1:length(xx)
    anglevector(i) = atan(position/mid_distance);
    anglevector(i) = anglevector(i)*(360/(2*pi));
    position = position+microphoneseparation;
end

end

```