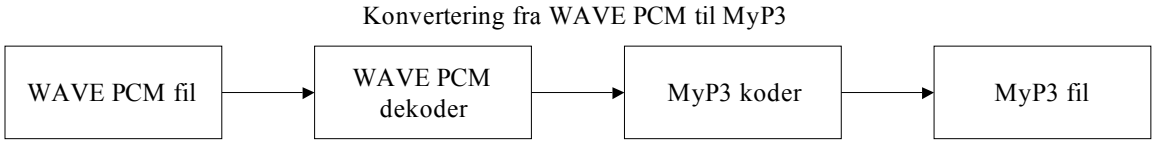
Indholdsfortegnelse

Indledning

Det menneskelige øre har begrænsninger, og det er fornuftigt at udnytte i den digitale verden for at begrænse datamængden. Øret kan f.eks. kun høre frekvenser i et bestemt interval, som bliver mindre med alderen. Dybe toner kan øret kun høre i mono, og toner kan overdøve andre, så de slet ikke registreres. Nogle af disse begrænsninger vil blive udnyttet i en standard kaldet MyP3 (se bilag 1 "MyP3-optager/-afspiller").
Denne rapport vil indeholde en analyse af ørets opfattelse af lyd, som tager udgangspunkt i information fra litteraturkilder. Med specifik viden om ørets opfattelse af lyd vil det være muligt at komprimere data fra en lydkilde. Det vil blive gjort ved at fjerne data uden der opstår et hørbart kvalitetstab.
MyP3 standarden vil blive implementeret i et kodek[1], så lydkilder, der er kvantiseret ved hjælp af Pulse Code Modulation (PCM), kan kodes til et MyP3 format og dekodes tilbage til et PCM signal.
For at kunne implementere MyP3 standarden skal signalbehandling bruges til at bestemme hvilke lyddata, der kan fjernes.

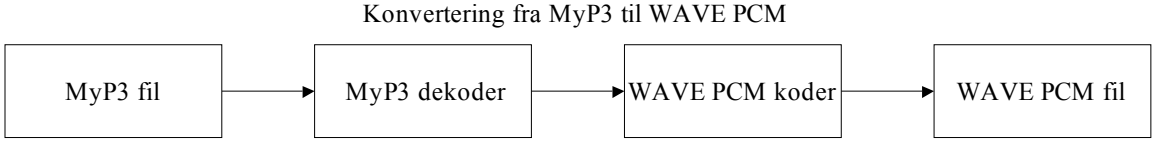
Beskrivelse

For at minimere projektets omfang vil analoge lydkilder på forhånd være kvantiseret, formateret til WAVE PCM (se bilag 2 "WAVE PCM format" og 3 "Microsofts RIFF Specifikation") og gemt i en fil. Det betyder, at der skal implementeres en WAVE PCM dekoder, så WAVE PCM formatet kan konverteres til MyP3 formatet.
Når en lydkilde af WAVE PCM formatet er konverteret til MyP3 formatet vil det blive gemt i en fil.



Figur 1. Illustration af konverteringen fra WAVE PCM til MyP3.

Efter lydkilden er formateret til MyP3 og gemt i en fil, vil det være hensigtsmæssigt at kunne afspille den. Formålet med projektet indebærer ikke, at der skal udvikles en MyP3 afspiller. Til gengæld findes et utal af WAVE PCM afspillere i stort set alle operativ systemer.
Derfor skal der også implementeres en WAVE PCM koder, så MyP3 formatet kan konverteres tilbage til WAVE PCM formatet.



Figur 2. Illustration af konverteringen fra MyP3 til WAVE PCM.

Det betyder overordnet, at følgende skal udvikles:

- Programmel til konvertering fra WAVE PCM format til MyP3 format.
- Programmel til konvertering fra MyP3 format til WAVE PCM format.

[1] Kodek er en forkortelse af koder og dekoder.

Projektets målsætninger

Krav til lydkilder

Alle lydkilder skal være af WAVE PCM formatet, men formatet er fleksibelt og lyddata i en WAVE PCM fil kan derfor have forskellige specifikationer.
Derfor skal en lydkilde i dette projekt opfylde følgende krav:

- 1 lydkanal (mono).
- Sample frekvens på 44100 Hz.
- Samples af typen: 2-kompliment 16-bit Little Endian.

Figur 3. Illustration af begrebet "Little Endian".

Arbejdsforløbet

Det er et faktum at et godt resultat opnås hurtigst, hvis det er muligt at teste tidligt i forløbet. Derfor er projektet opdelt i 3 iterationer:

- En 80 % løsning, hvor de grundlæggende målsætninger er implementeret.
- En 100 % løsning, hvor alle målsætninger er implementeret.
- En 120 % løsning, hvor mulige forbedringer, ekstra funktionaliteter og gode ideer, som projektgruppen har overvejet under forløbet, bliver implementeret. Det er derfor ikke et mål, at der arbejdes med denne løsning.

I alle iterationer bliver nye målsætninger implementeret, og derfor vil hver løsning have en tilhørende test.

Målsætninger i 80 % løsningen

I denne løsning skal 2 konverteringsmetoder implementeres. Den første skal kunne konvertere en fil af WAVE PCM formatet til en fil af MyP3 formatet. Den anden skal gøre det modsatte.
MyP3 formatet skal bestå af lyddata, som er transformeret til det frekvensbaseret område vha. Diskret Fourier Transformation (DFT). Det kvantiserede frekvensdata skal skaleres, så alle bits der er til rådighed bliver udnyttet bedst muligt. Allokeringen af bits til det kvantiserede frekvensdata skal være konstant.

Målsætninger i 100 % løsningen

I denne løsning skal en psykoakustisk model implementeres i MyP3 formatet, så uhørbar lyd bliver fjernet. Der skal også tages stilling til, hvor mange bits, der reelt er nødvendige for at gengive den kvantiserede frekvensdata, uden at der opstår et hørbart kvalitetstab.
Derudover skal Fourier transformationen forbedres, så MyP3's kodek hastighed øges.

Den psykoakustiske model skal baseres på ørets opfattelse af lyd. Frekvensernes amplitude skal undersøges, da amplitudens minimum skal have en bestemt størrelse for at øret kan opfatte dem. Frekvensernes amplitude kan også være så stor, at nærliggende frekvenser ikke bliver opfattet af øret. Denne viden skal implementeres i modellen, så uhørbare frekvenser kan blive fjernet.

Da amplituder ved forskellige frekvenser hele tiden varierer, er det ikke nødvendigt, at benytte flest mulige bits til at gemme de kvantiserede frekvensdata i MyP3 filen. Derfor skal det undersøges, hvor mange bits der reelt er nødvendige, så datamængden i MyP3 filen kan reduceres yderligere.
Fourier transformationen skal forbedres ved at DFT metoden udskiftes med Fast Fourier Transformation (FFT).

Målsætninger i 120 % løsningen

Følgende er forslag:

- 100 % løsningen bliver implementeret på en embedded computer.
- 100 % løsningen bliver implementeret, som en afspiller på et bestemt operativ system.
- 100 % løsningen bliver implementeret, som et plug-in til Winamp [2] eller en anden kendt afspiller.

Test af 80 % løsningen

En del af denne test skal udføres objektivt, da det er muligt at beskrive kvantiseringsstøjen[2] ved at udregne signal/støj forholdet.
Resten af testen skal være subjektiv, så det kan blive bekræftet af personer, at der ikke er et hørbart kvalitetstab i lyden.

Test af 100 % løsningen

Denne test kan kun udføres subjektivt, da den psykoakustiske model har kvantiseret lyddata, så det ikke længere er muligt at beskrive kvaliteten objektivt.
Derfor skal der udvikles en affektiv test, som betyder at en gruppe personers præferencer skal benyttes til at bedømme, om der er opstået et hørbart kvalitetstab ved komprimeringen.

Test af 120 % løsningen

Denne test er ikke bestemt, da en løsning først skal beskrives og implementeres.

Overordnet krav

Alt programmellet skal implementeres med C++:

- C++ standarder skal overholdes [3].
- Programmelkoden skal være uafhængigt af operativ system.
- Brugergrænsefladen skal implementeres, som et konsol program og være simpel.

[2] Kvantiseringsstøj, også kendt som afrundingsstøj, opstår ved multiplikation, hvis der ikke er tilstrækkeligt med bits til at beskrive resultatet.

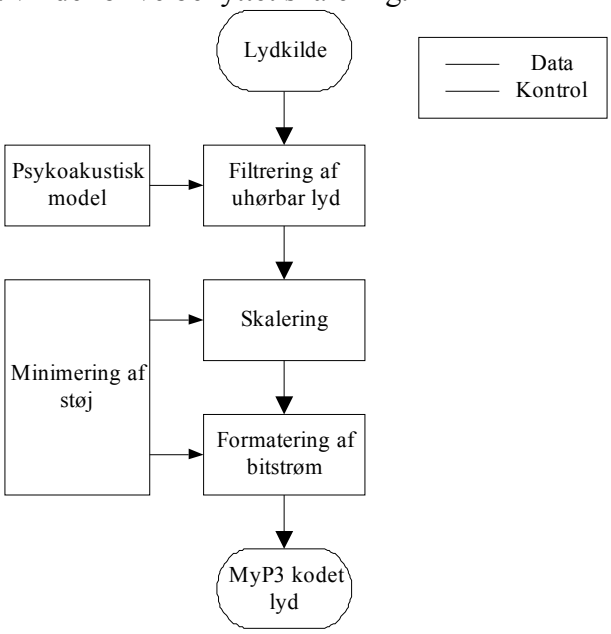
Analyse af MyP3

Motion Picture Expert Group (MPEG)

Siden MPEG organisationen blev grundlagt i 1988 er adskillige standarder blevet introduceret, og pga. MPEG's størrelse, som er i dag består af cirka 350 medlemmer fra forskellige industrier og universiteter, er de blevet meget kendt og brugt verden over. Nogle af de mest kendte standarder er MPEG-1, MPEG-2 og MPEG-4.
MPEG-1 video delen blev benyttet til Video CD, som var en af de første populære metoder til at lagre video på en Compact Disc (CD). Denne metode er i dag forældet, da MPEG-4 blev introduceret i 1998, og senere blev implementeret i kodek, som DivX [4] og XviD, der begge kan gengive video i langt bedre kvalitet i forhold til størrelsen af datamængden.
I 1994 blev MPEG-2 introduceret, og benyttes i dag til at gengive video i digitale netværk med stor datakapacitet. Nogle af de mere kendte er digital satellit og kabel TV. Men det mest populære brug af MPEG-2 er en mindre modificeret udgave, som benyttes i Digital Versatile Disc (DVD).
MPEG-1 audio delen findes i dag med 3 forskellige komprimeringsmetoder. De 3 metoder kendes bedre som lag. Lag 1 er den simpleste form for komprimering og lag 3 er den mest kompliceret (og bedste). Den mest populære metode er lag 3, der kendes under navnet MP3. Faktisk er MP3 en videreudvikling af metoder fra MPEG-1 og MPEG-2, så et mere korrekt navn ville være MPEG-1/2 lag 3 [6]. MP3 er i stand til at reducere størrelse på en lydkilde i CD kvalitet med 12:1 uden et hørbart kvalitetstab.

MyP3 standarden

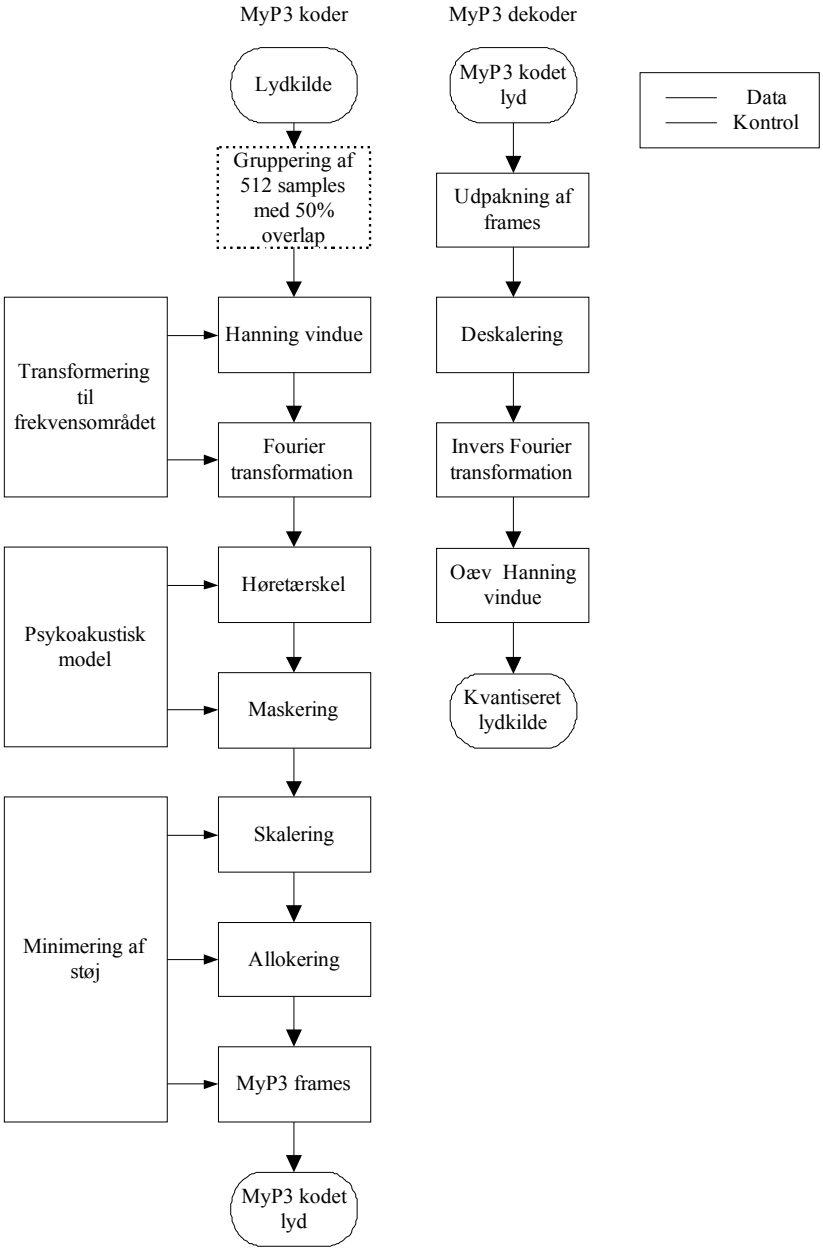
MyP3 er en standard, der minder om MPEG-1 lag 1. Komprimeringen vil udelukkede foregå i det frekvensbaserede område, og der vil kun blive benyttet lossy-metoder[3]. For at begrænse kvantiseringen af lyddata vil der blive benyttet skalering.



Figur 4. Illustration af MyP3 standarden

[3] Lossy komprimering fjerner data i modsætning til lossless komprimering, der tillader genskabelse af det oprindelige data.

MyP3 kodek



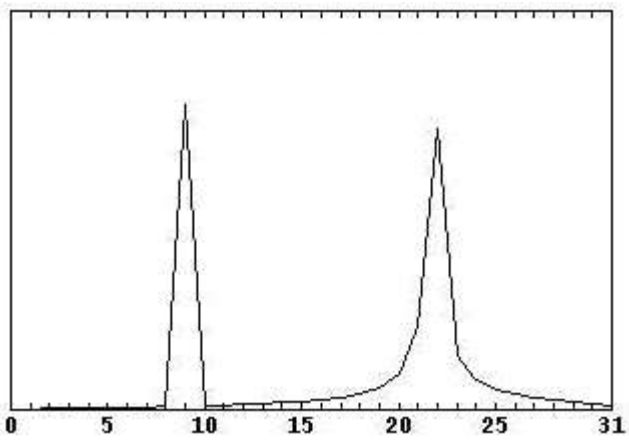
Figur 5. Illustration af MyP3 koder og dekoder

De følgende afsnit vil beskrive de forskellige dele af MyP3's kodek og give detaljeret forståelse af MyP3 standarden. Det vil give anledningen til implementering af MyP3's kodek senere i projektet.

Hanning vindue

Da kodningen af en lydkilde indebærer, at der benyttes Fourier transformation, og det sker på en gruppering af 512 samples med 50 % overlap (efterfølgende vil denne gruppering blive kaldt en blok), kan der opstå lækage [7] eller "tailing". Det sker i de respektive Barks (se bilag 1 "MyP3-optager/afspiller", side 6) efter Fourier transformationen, hvis en af spektralkomponenterne ligger mellem to basisfrekvenser.

Det skal bemærkes at 50 % overlap forårsager, at der sker en fordobling af datamængden.



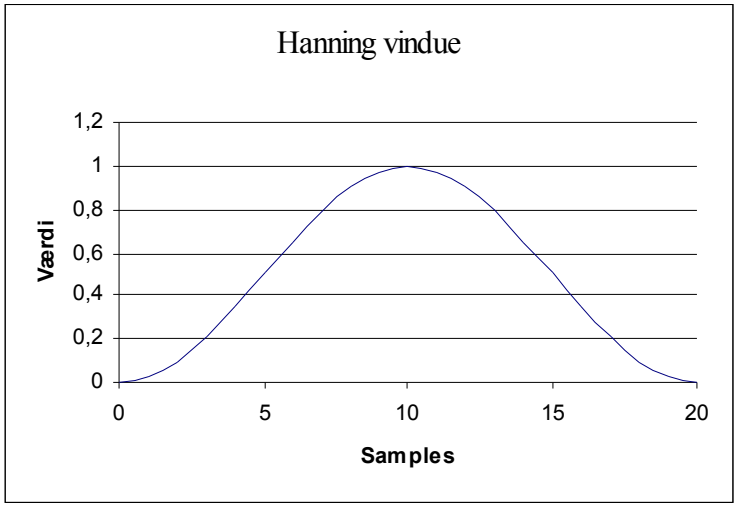
Figur 6. Et signal med en basisfrekvens (venstre) og en frekvens (højre) imellem 2 basisfrekvenser efter Fourier transformation, hvor der ikke er anvendt et Hanning vindue.

En basisfrekvens er en frekvens, der ligger præcis i en DFT bin. Det ses, at den højre frekvens, der ligger imellem to basisfrekvenser, har en betydelig"tailing". For at undgå dette anvendes et Hanning vindue, der multipliceres på alle blokke før Fourier transformationen.

Hanning vinduet ser således ud:

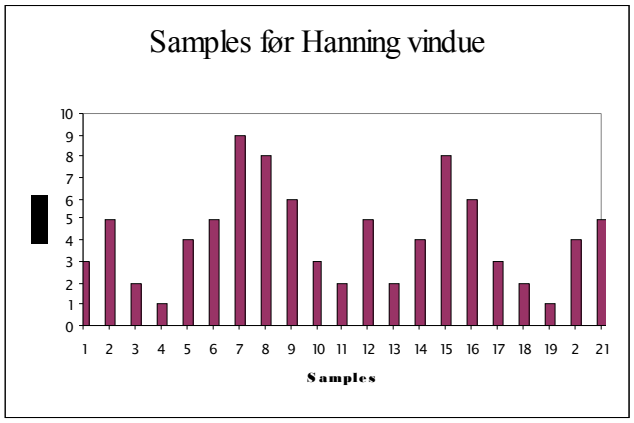
$$W[n] = \begin{cases} 0{,}5 - 0{,}5 \cdot \cos(2\pi n/M) \text{ for } 0 \le n \le M \\ 0 \text{ ellers} \end{cases}$$

Og er illustreret på en figur:

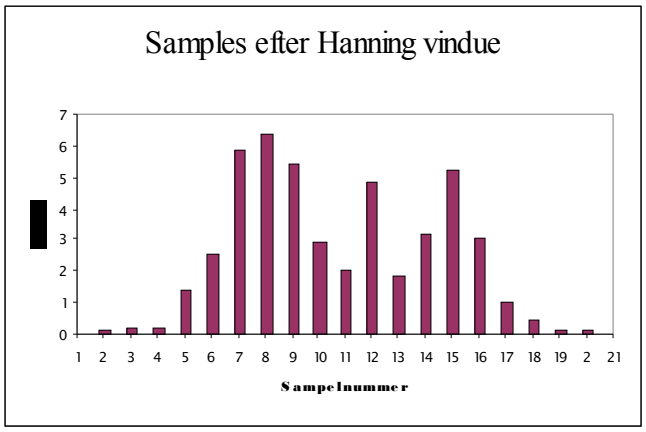


Figur 7. Illustration af et Hanning vindue.

Følgende figurer viser nogle samples før og efter Hanning vinduet bliver multipliceret på en blok.



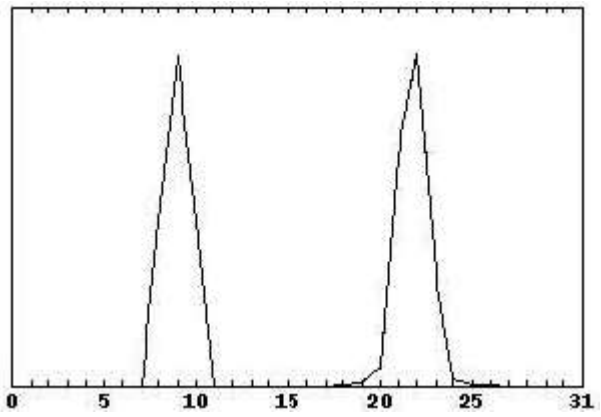
Figur 8. Illustration af en tilfældig blok før multiplikation med et Hanning vindue.



Figur 9. Illustration af en tilfældig blok efter multiplikation med et Hanning vindue.

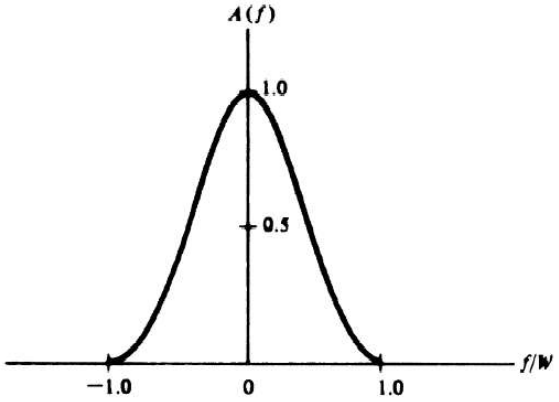
Det ses, at vinduet dæmper kraftigt i endepunkterne, og mindre mod midten af vinduet.

Ved anvendelse af et Hanning vindue før Fourier transformation begrænses den lækage eller "tailing", der opstår ved frekvenser udenfor basisfrekvenserne.

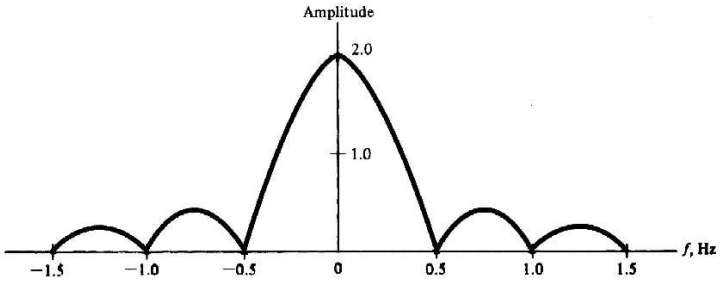


Figur 10. Et signal med en basisfrekvens (venstre) og en frekvens (højre) imellem 2 basisfrekvenser efter Fourier transformation, hvor der er anvendt et Hanning vindue.

Et argument for, hvorfor et Hanning vindue begrænser "tailing" i forhold til et rektangulært vindue, er at Fourier transformation af et rektangulært vindue, er en sinc funktion.
Fourier transformationen af et Hanning vindue er også en sinc funktion, men med mindre sidebånd. Når et Hanning vindue multipliceres på en blok i tidsområdet, svarer det til, at vinduerne foldes på signalet i frekvensområdet. Derved kan den øgede bredde af signaltoppen til højre på figurer 6 forklares.

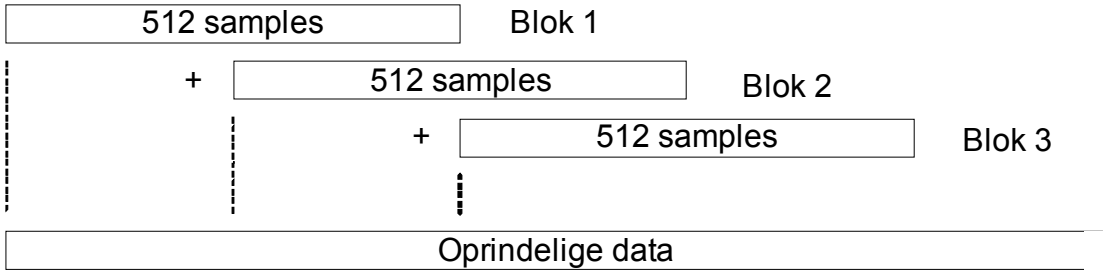


Figur 11. Fourier transformation, hvor et Hanning vindue er anvendt.



Figur 12. Fourier transformation, hvor et rektangulært vindue er anvendt.

Ved genskabning af samples efter invers Fourier transformation, skal en invers udgave af Hanning vinduet benyttes.
Blokkene med 50 % overlappene samples bevirker til, at Hanning vinduet kan ophæves ved at den sidste halvdel af hver blok adderes med den første halvdel af næste blok i den respektive rækkefølge.



Figur 13. Illustration af ophævelse af Hanning vindue, hvor det oprindelige data gendannes.

I det følgende vises nogle matematiske egenskaber ved Hanning vinduet.

Det ses på Figur 7, at Hanning vinduet har værdierne 0 i endepunkterne og 1 i midten.

Dette vises let matematisk (M = 20 på Figur 6):

$$W[0] = 0{,}5 - 0{,}5 \cdot \cos\left(\frac{2\pi\, 0}{M}\right) = 0{,}5 - 0{,}5 = 0$$

$$W[M/2] = 0{,}5 - 0{,}5 \cdot \cos\left(\frac{2\pi\, \frac{M}{2}}{M}\right) = 0{,}5 + 0{,}5 = 1$$

$$W[M] = 0{,}5 - 0{,}5 \cdot \cos\left(\frac{2\pi M}{M}\right) = 0{,}5 - 0{,}5 = 0$$

Og hvis 50 % overlap genskaber de oprindelige samples må der gælde at:

$$W[\frac{M}{2} + L] + W[L] = 1, \text{ for } 0 \le L \le \frac{M}{2}$$

W[L] findes:

$$W[L] = 0{,}5 - 0{,}5 \cdot \cos\left(\frac{2\pi L}{M}\right)$$

og

$$W[\frac{M}{2} + L] = 0{,}5 - 0{,}5 \cdot \cos\left(\frac{2\pi\left(\frac{M}{2} + L\right)}{M}\right)$$

$$\Updownarrow$$

$$W[\frac{M}{2} + L] = 0{,}5 - 0{,}5 \cdot \cos\left(\frac{\frac{2\pi M}{2} + 2\pi L}{M}\right)$$

$$\Updownarrow$$

$$W[\frac{M}{2} + L] = 0{,}5 - 0{,}5 \cdot \cos\left(\frac{2\pi M}{2M} + \frac{2\pi L}{M}\right)$$

$$\Updownarrow$$

$$W[\frac{M}{2} + L] = 0{,}5 - 0{,}5 \cdot \cos\left(\pi + \frac{2\pi L}{M}\right)$$

$$\Updownarrow$$

$$W[\frac{M}{2} + L] = 0{,}5 + 0{,}5 \cdot \cos\left(\frac{2\pi L}{M}\right), \text{da } \cos(x + \pi) = -\cos(x)$$

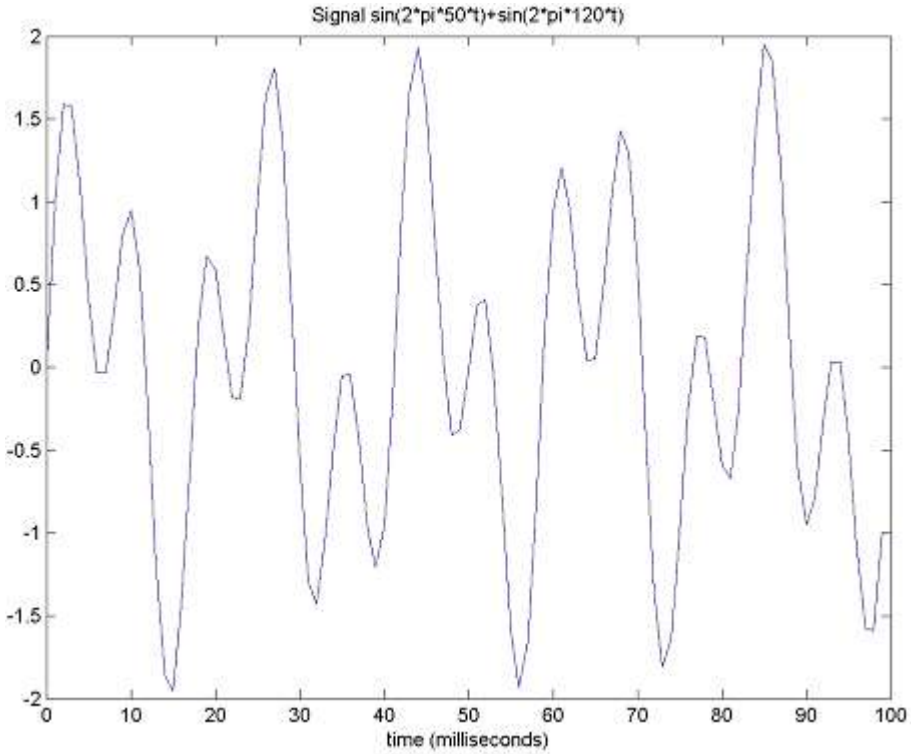
Derefter findes $W[\frac{M}{2} + L] + W[L]$:

$$0{,}5 + 0{,}5 \cdot \cos\left(\frac{2\pi L}{M}\right) + 0{,}5 - 0{,}5 \cdot \cos\left(\frac{2\pi L}{M}\right) = 0{,}5 + 0{,}5 = 1$$

Hvilket skulle vises.

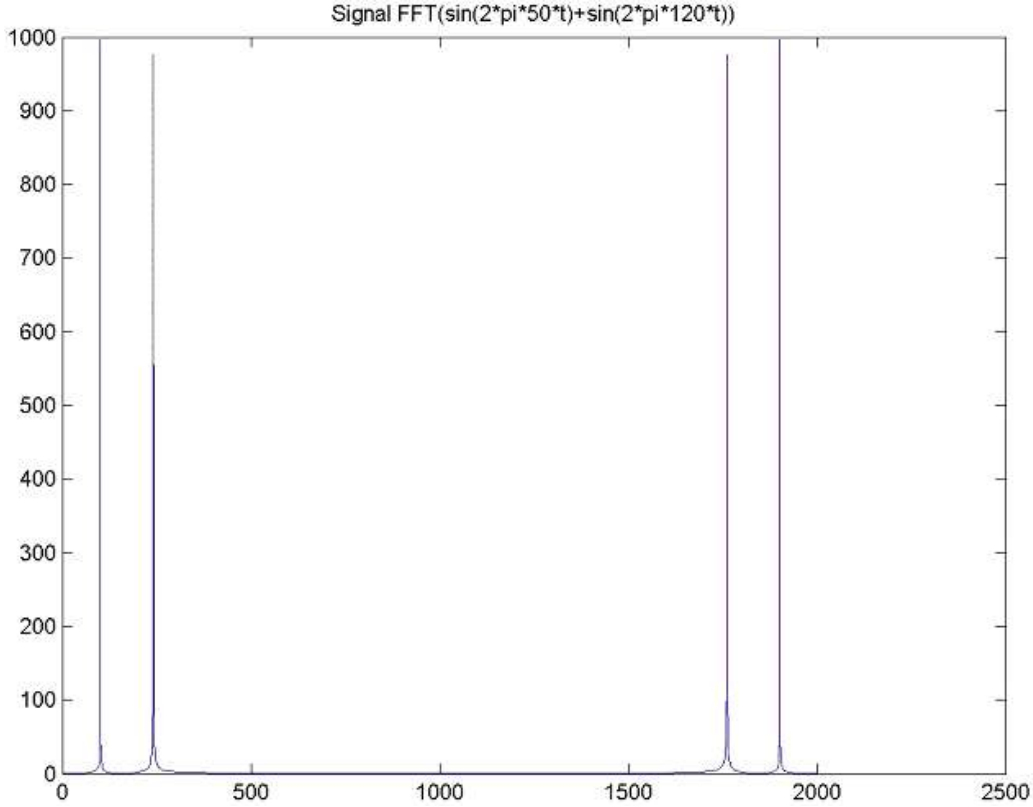
Fourier transformation

For at undersøge frekvensindholdet af den modtagne lydkilde, skal der udføres en transformation af signalet fra tidsområdet til frekvensområdet. Det skal ske ved hjælp af Diskret Fourier transformation (DFT) i 80 % løsningen. Som en illustration på dette betragt følgende figur, som er en afbildning af 2 sinussignaler med to forskellige frekvenser, men med samme amplitude. Det ses på figuren at amplitude er en funktion af tid.



Figur 14. Sinussignal bestående af 2 frekvenser.

Ved at Fourier transformere får man et udtryk for fasen og amplituden, som funktion af frekvensen. Følgende figur viser amplitude som funktion af frekvensen.



Figur 15. Fourier transformationens amplitudespektrum.

Det ses på figuren, at Fourier transformationens amplitudespektrum bliver symmetrisk omkring midten.
Der anvendes DFT, da det modtagne signal består af diskrete samples, og resultatet ønskes på diskret form. Ved DFT opnås ideelt et spektrum bestående af toppe, som hver repræsenterer en enkelt frekvens, hvor højden er amplituden. DFT kan beregnes ud fra følgende udtryk:

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n] \cdot e^{\frac{-j2\pi kn}{N}}$$

Ved beregning af DFT er det kun nødvendigt at beregne de første $\frac{N}{2}$ værdier, da spektret gentager sig selv. Det kan vises på følgende måde:

$$X[k] = X^*[N-k]$$

Ved at sætte $k = N - k$ i DFT udtrykket kan følgende gøres:

$$X[N-k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n] \cdot e^{\frac{-j2\pi k(N-k)n}{N}}$$

$$\Updownarrow$$

$$X[N-k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n] \cdot e^{\frac{-j2\pi Nn}{N}} \cdot e^{\frac{j2\pi kn}{N}}$$

$$\Updownarrow$$

$$X[N-k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi n} \cdot e^{\frac{j2\pi kn}{N}}$$

$e^{-j2\pi n}$ er altid 1

Og derfor kan udtrykket reduceres:

$$X[N-k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n] \cdot e^{\frac{j2\pi kn}{N}}$$

Det er netop den komplekst konjugerede af $X[k]$, $X[k] = X^*[N-.k]$.

Desuden kan følgende ses, når $k = \frac{N}{2}$:

$$X\left[\frac{N}{2}\right] = \frac{1}{N}\sum_{n=0}^{N-1} x[n] \cdot (-1)^n$$

Det vil sige, at $X\left[\frac{N}{2}\right]$ er et reelt tal med alternerende fortegn.

Det ses ydermere at når $k = 0$ fås et reelt tal:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{0} = \frac{1}{N} \sum_{n=0}^{N-1} x[n]$$

Den inverse Fourier transformation (iDFT) kan beregnes med følgende udtryk:

$$x[k] = \frac{1}{N} \sum_{n=0}^{N-1} X[n] \cdot e^{\frac{-j2\pi kn}{N}}$$

For at se $x[k]$ igen bliver et reelt tal indsættes iDFT i DFT udtrykket:

$$x[k] = \sum_{n=0}^{N-1} X[N] \cdot e^{\frac{j2\pi kn}{N}}$$

$$= \sum_{n=0}^{N-1} \frac{1}{N} \sum_{m=0}^{N-1} x[m] \cdot e^{\frac{-j2\pi km}{N}} \cdot e^{\frac{j2\pi kn}{N}}$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x[m] \cdot e^{\frac{j2n(k-m)n}{N}}$$

$$= \frac{1}{N} \sum_{m=0}^{N-1} x[m] \cdot \sum_{n=0}^{N-1} e^{\frac{j2\pi (k-m)n}{N}}$$

Den sidste sum er en kompleks eksponentiel summation, og det kan vises (se bilag 4 ”DFT note”) at den er 0 for alle $k \neq m$, og N for $k = m$.

Så til slut fås:

$$= \frac{1}{N} \sum_{m=0}^{N-1} x[m] \cdot (N \cdot \delta(m-k))$$

$$= x[k]$$

I 100 % løsningen skal DFT erstattes af Fast Fourier transformation (FFT). Når man undersøger DFT og iDFT algoritmen viser det sig, at antallet af basale udregninger er cirka N^2, hvilket i dette tilfælde er $(512)^2 = 262144$ pr. blok, men ved at udnytte ”divide and conquer” teknik, kan antallet af udregninger reduceres. Derfor bliver 2 algoritmer, der udregner FFT, undersøgt.

Følgende viser udledningen af DIT (decimation in time) fra DFT:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{nk}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} x[2n] \cdot W_N^{(2n)k} + \sum_{n=0}^{\frac{N}{2}-1} x[2n+1] \cdot W_N^{(2n+1)k}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} [2n] \cdot W_N^{n(2k)} + W_N^k \cdot \sum_{n=0}^{\frac{N}{2}-1} x[2n+1] \cdot W_N^{n(2k)}$$

$$= \frac{N}{2}\text{-point FFT (‘lige del’)} + \frac{N}{2}\text{-point FFT (‘ulige del’)}$$

De to $\frac{N}{2}$-point FFT’er finder kun værdier for $k = 1,..., \frac{N}{2} - 1$, men da FFT er periodisk, kan de resterende $\frac{N}{2}$ udregnes med:

$$W_N^{n\left(k+\frac{N}{2}\right)} = e^{\frac{-j2\pi\left(k+\frac{N}{2}\right)}{N}} = e^{-j\pi} \cdot e^{\frac{-j2\pi k}{N}} = -e^{\frac{-j2\pi k}{N}} = -W_N^{nk}$$

Så for $k = 1,..., \frac{N}{2} - 1$ er:

N-point $FFT(k) = \frac{N}{2}$-point $FFT(k) + W_N^k \cdot \frac{N}{2}$-point $FFT(k)$,

N-point $FFT\left(k + \frac{N}{2}\right) = \frac{N}{2}$-point $FFT(k) - W_N^k \cdot \frac{N}{2}$-point $FFT(k)$

Følgende viser udledningen af DIF (decimation in frequency) fra DFT

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{nk}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} x[n] \cdot W_N^{nk} + \sum_{n=\frac{N}{2}}^{N-1} x[n] \cdot W_N^{nk}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} \left(x[n] + x\left[n + \frac{N}{2}\right] \cdot e^{-j\pi \cdot k} \right) \cdot W_N^{nk}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} \left(x[n] + x\left[n + \frac{N}{2}\right] \cdot (-1)^k \right) \cdot W_N^{nk}$$

Summen splittes nu op i en ”lige” og ”ulige” del.

$$\sum_{n=0}^{\frac{N}{2}-1}\left(x[n]+x\left[n+\frac{N}{2}\right]\cdot(-1)^{(2k)}\right)\cdot W_{\frac{N}{2}}^{n(2k)}+\sum_{n=0}^{\frac{N}{2}-1}\left(x[n]+x\left[n+\frac{N}{2}\right]\cdot(-1)^{(2k+1)}\right)\cdot W_{\frac{N}{2}}^{n(2k+1)}=$$

$$\sum_{n=0}^{\frac{N}{2}-1}\left(x[n]+x\left[n+\frac{N}{2}\right]\right)\cdot W_{\frac{N}{2}}^{n(2k)}+\sum_{n=0}^{\frac{N}{2}-1}\left(x[n]-x\left[n+\frac{N}{2}\right]\right)\cdot W_{\frac{N}{2}}^{n(2k+1)}=$$

$$\sum_{n=0}^{\frac{N}{2}-1}\left(x[n]+x\left[n+\frac{N}{2}\right]\right)\cdot W_{\frac{N}{2}}^{n(2k)}+\sum_{n=0}^{\frac{N}{2}-1}\left(x[n]-x\left[n+\frac{N}{2}\right]\right)\cdot W_{\frac{N}{2}}^{n(2k)}\cdot W_{\frac{N}{2}}^{n}=$$

$\frac{N}{2}$-point FFT ('lige del') + $\frac{N}{2}$-point FFT ('ulige del')

De to algoritmer er næsten ens, forskellen ligger i hvornår $W_N^{nk}=e^{\frac{-j2\pi\cdot k}{N}}$ multipliceres på. Begge algoritmer kan implementeres med rekursive metoder og in-place udregninger, men DIT algoritmen vælges.

Antallet af udregninger er nu reduceret til $N\cdot\log_2 N$, i vores tilfælde $512\cdot\log_2(512)=512\cdot 9=4608$.

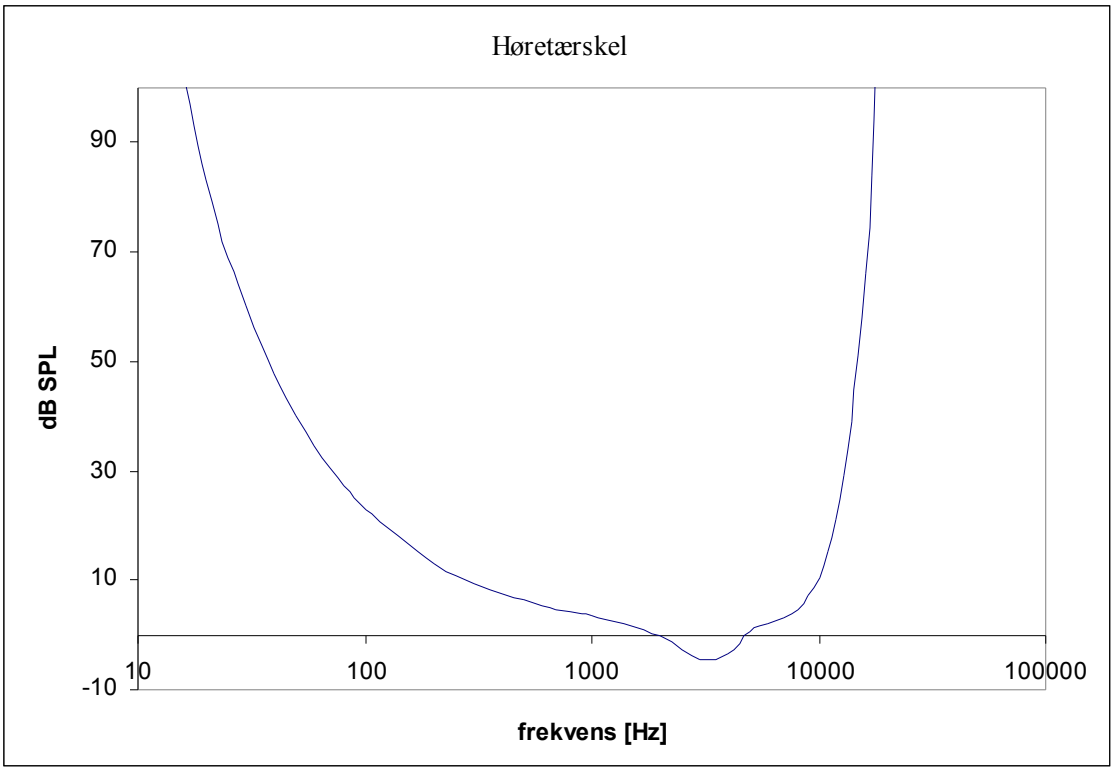
Hvilket giver en teoretisk maksimal forbedring på $\frac{512\cdot 512}{512\cdot\log_2(512)}=\frac{512}{9}\approx 57$.

Det ses tydeligt at jo større blokke der benyttes, jo hurtigere bliver FFT algoritmen i forhold til DFT algoritmen.

Høretærskel

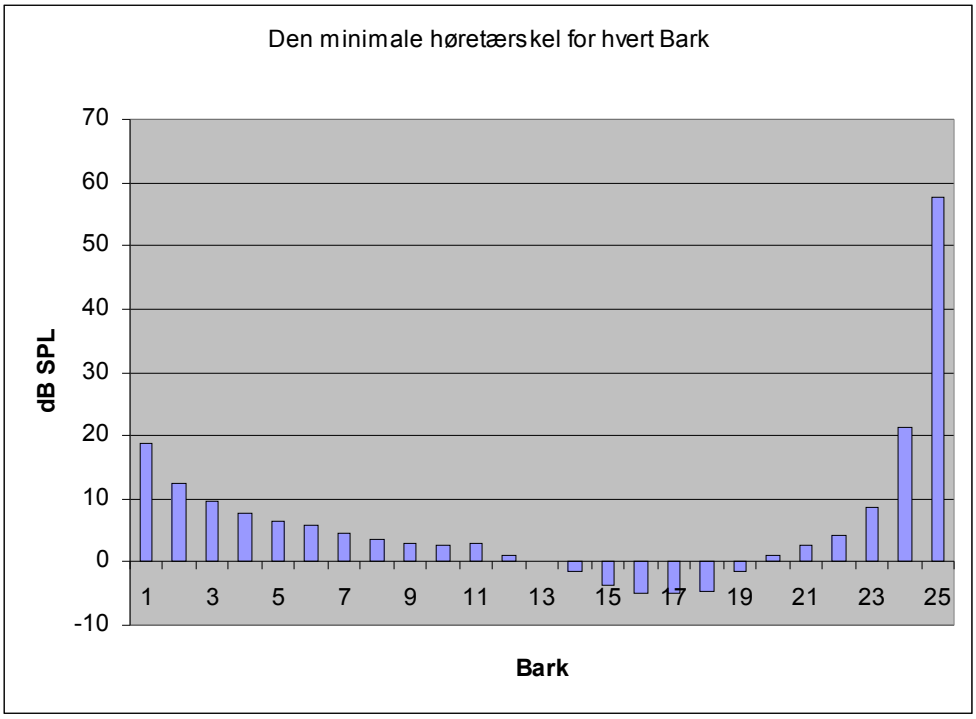
Analyser af menneskets øre har vist at det ikke har samme følsomhed ved alle frekvenser. Høretærskelkurven angiver den netop hørbare grænse ved en given frekvens.
Høretærsklen lader sig approximere ved følgende udtryk [1]:

$$T_q(f)=3{,}64\left(\frac{f}{1000}\right)^{-0{,}8}-6{,}5e^{-0{,}6\left(\frac{f}{1000}\right)^2}+10^{-3}\left(\frac{f}{1000}\right)^4,0Hz<f<22050Hz\quad\text{(dB)}$$



Figur 16. Graf af høretærsklen

Som det ses på grafen er øret ikke særligt følsomt i bas området (ca. 16-300 Hz), men til gengæld er det meget følsomt i sidste del af mellemtone området (ca. 300-3000 Hz), og i starten af diskant området (ca. 3000-20000 Hz).
Da analysen af lyd foregår i det frekvensbaseret området skal der findes en passende dB værdi for hvert Bark.
I dette projekt ønskes der den mindste dB værdi for hvert Bark, som reelt svarer til at høretærsklen findes for den frekvens, som øret er mest følsomt overfor i hvert Bark.
Grunden til at denne frekvens vælges, skyldes at en højere høretærskel ville forårsage, at netop denne frekvens ville blive fjernet fra lyddata, og det ville danne et hørbart kvalitetstab i komprimeringen.



Figur 17. Graf med de frekvenser, som øret er mest følsomt overfor i hvert Bark (se bilag 5)

Barkskalaen lader sig approximere ved følgende udtryk:

$$z(f) = 13\arctan(0{,}00076 f) + 3{,}5\arctan\left[\left(\frac{f}{7500}\right)^2\right] \quad \text{(Bark)}$$

Da den minimale høretærskel nu kendes for hvert Bark, skal amplituden findes for de spektralkomponenter der modtages fra Fourier transformationen.
Amplitude kan findes med følgende formel:

$$|a + j \cdot b| = \sqrt{a^2 + b^2}$$

Og kan omregnes til dB med følgende formel:

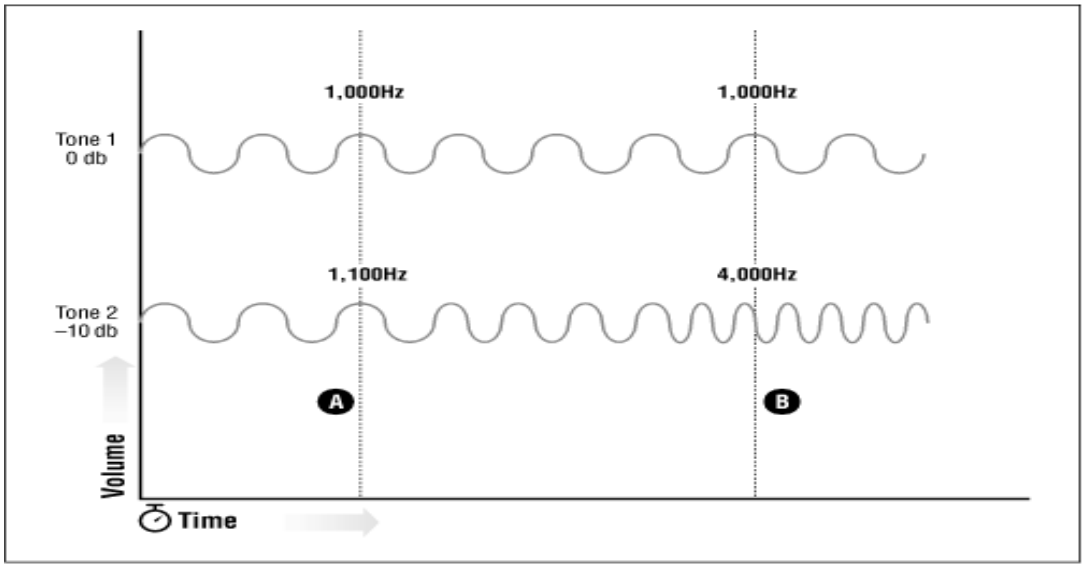
$$amp = 20 \cdot \log_{10}\left(\sqrt{a^2 + b^2}\right) \quad \text{(dB)}$$

Nu kan alle spektralkomponenter sammenlignes med den minimale høretærskel for deres respektive Bark. Hvis amplitude er mindre end den minimale høretærskel, skal spektralkomponenten nulstilles.
Med denne metode kan lossy komprimering opnås ved at tage ørets høretærskel i betragtning.

Maskering

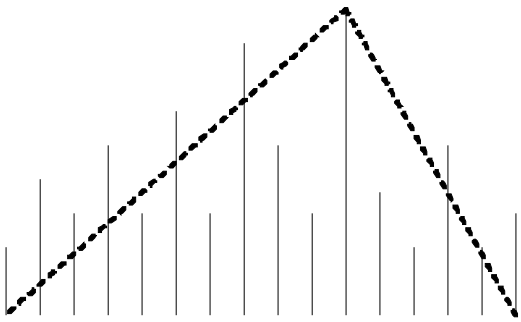
Følgende er eksempler på maskering. Hvis man betragter en fugl i luften, der flyver ind foran solen, vil fuglen være synlig før den når ind foran solen, og synlig igen efter den kommer forbi solen. Den tid fuglen er foran solen, er den ikke synlig da baggrundslyset er for skarpt – fuglen er

maskeret væk af solen. Et andet eksempel vil være den pibende lyd fra fingrene, der glider på strengene på en guitar, som lader sig høre ved et stille musikstykke. Er det et Metallica nummer man hører, kan den pibende lyd ikke høres, da den bliver maskeret væk af de andre kraftige toner i musikken [8].
For at gøre princippet mere konkret forestiller vi os, at vi har et 1000 Hz signal sammen med et 1100 Hz signal, der har en meget mindre amplitude f.eks. –10dB. De fleste vil ikke være i stand til at høre 1100 Hz signalet. Ikke kun fordi amplituden er så lille, men også fordi frekvenserne ligger så tæt. Den kraftige tone maskerer den svage. Hvis vi ændrer det svage signal til højere frekvenser, uden at ændre styrken på signalet, vil man på et vist tidspunkt kunne høre begge signaler, når deres frekvens er tilstrækkeligt langt fra hinanden [8].



Figur 18. Eksempel på simultan maskering [8].

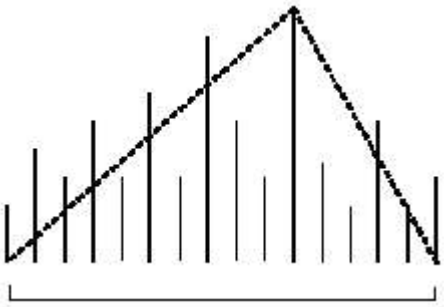
Hvis man i et givet signal betragter frekvens fordelingen, kan man inden for hvert Bark, identificere frekvensen med størst effekt, og tegne en maskeringskurve uden om denne. Princippet er så, at alle frekvenser, som ligger under maskeringskurven ikke er hørbare, og kan derfor fjernes uden det generelle lydbillede ændres.
Følgende figur er et eksempel på et tilfældigt Bark. Der ses adskillige spektralkomponenters amplitude. De amplituder, der ligger under den stiplede maskeringskurve kan ikke høres, og kan derfor fjernes.
På de illustrerende figurer vil de frekvenser, som er angivet med en tynd linie være ikke hørbare, og derfor blive fjernet.



Figur 19. Amplituder under maskeringskurven kan ikke høres.

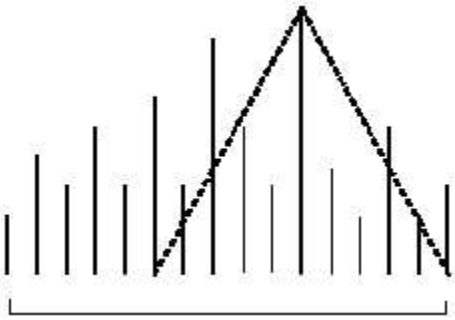
I dette projekt vil der blive arbejdet med 4 forskellige metoder med lineære maskeringskurver. De 4 forskellige metoder er som følger:

Metode 0: Tegner en ret linie fra frekvensen med højest amplitude til de enkelte Barks ender:



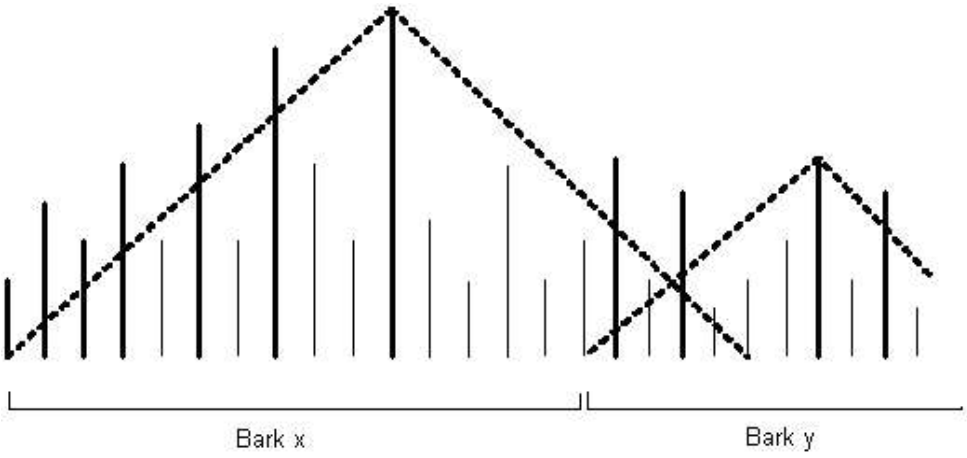
Figur 20. Makseringsmetode 1.

Metode 1: Tegner en ret linie fra frekvensen med højest amplitude til de enkelte Barks ender, men gør det symmetrisk, så Bark grænserne ikke overlappes:

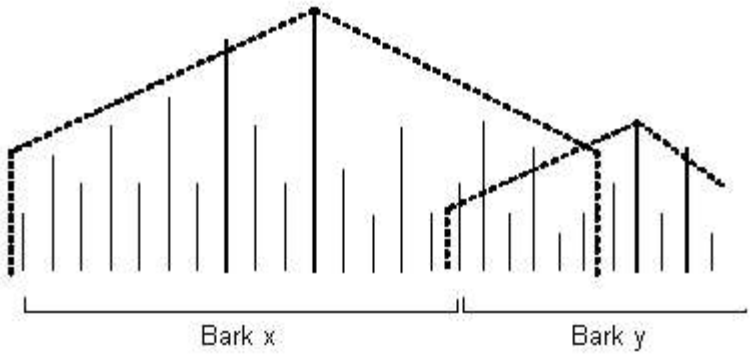


Figur 21. Maskeringsmetode 2.

Metode 2: Tegner en ret linie fra frekvensen med højest amplitude til de enkelte Barks ende længst fra den højeste amplitude, den gør det symmetrisk, så Bark grænserne overlappes:



Metode 3: Tegner en ret linie fra frekvensen med højest amplitude til de enkelte Barks ender med halv så stor hældning, som i udgave 2, den gør det symmetrisk, så Bark grænserne overlappes:



Det ses på figurerne, at metode 1 udmasker færrest frekvenser og udgave 3 udmasker flest frekvenser. Hvis rækkefølgen af udgaver efter hvor mange frekvenser de udmasker skal opstilles, ser de således ud (startende med den der udmasker mindst):
Metode 1 < Metode 0 < Metode 2 < Metode 3. Dvs. at det forventes at metode 1 komprimerer mindst og metode 3 komprimerer mest.

Skalering

For at minimere kvantiseringsstøjen i MyP3's kodek, er det fornuftigt at benytte skalering. Skalering er en metode, som drager bedre udnyttelse af de bits, der er stillet til rådighed i et system. Det betyder at efter skalering vil der være flere betydende bits til at beskrive et tal og derved mindske kvantiseringsstøjen.
Først skal det størst mulige tal, som kan beskrives med 2-kompliment 16-bit (1.15), findes.

$$Q[x] = -b_0 + \sum_{n=1}^{B} b_n \cdot 2^{-n}$$

Det største tal findes:

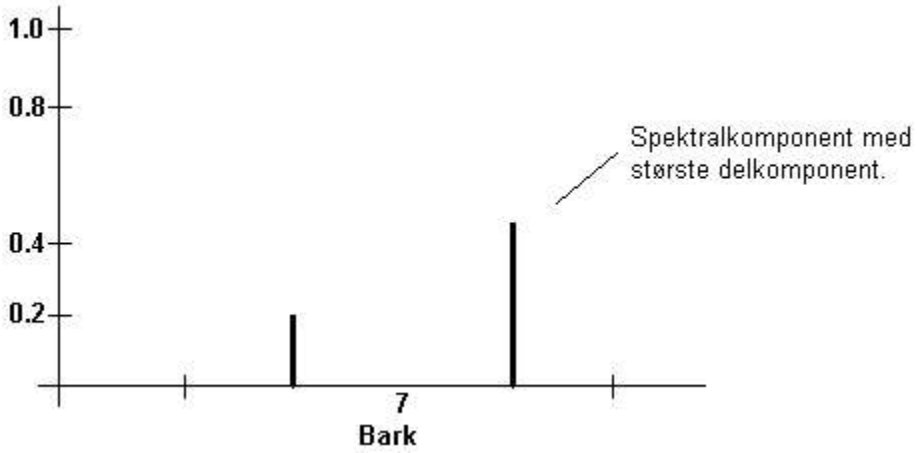
$$Q[\max] = -b_0 + \sum_{n=1}^{B} b_n \cdot 2^{-n}$$, $b_0 = 0$, b_n er alle 1 og B = 16

$$Q[\max] = 0.999969482421875 \approx 1$$

Derefter skal største delkomponent (a eller b i: $a + j \cdot b$) i hvert Bark findes. Når denne delkomponent er fundet, kan en skaleringsfaktor udregnes.
Skaleringsfaktorer er beskrevet med 2^{potens}, hvor potensen kan beskrives med 4 bits:

Bit-mønster	Skaleringsfaktor
0000	2^0
0001	2^1
0010	2^2
0011	2^3
...	...
1110	2^{14}
1111	2^{15}

Figur 22. Mulige skaleringsfaktorer.



Figur 23. Det ses at den største delkomponent i Bark 7 har størrelsen 0,4.

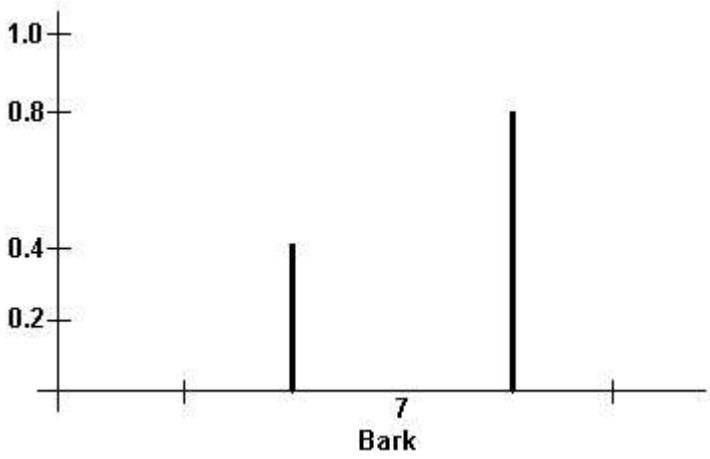
Det kan nu udregnes hvad skaleringsfaktoren's potens skal være:

$$\frac{\ln\left(\frac{Q[\max]}{amplitude}\right)}{\ln(2)} \approx potens$$

Potensen skal trunkeres til et heltal.

$$\frac{\ln\left(\frac{Q[\max]}{0{,}4}\right)}{\ln(2)} \approx 2$$

Hvis spektralkomponenterne fra Figur 23 bliver skaleret vil det se ud som følgende:



Figur 24. Skaleret spektralkomponenter.

Allokering

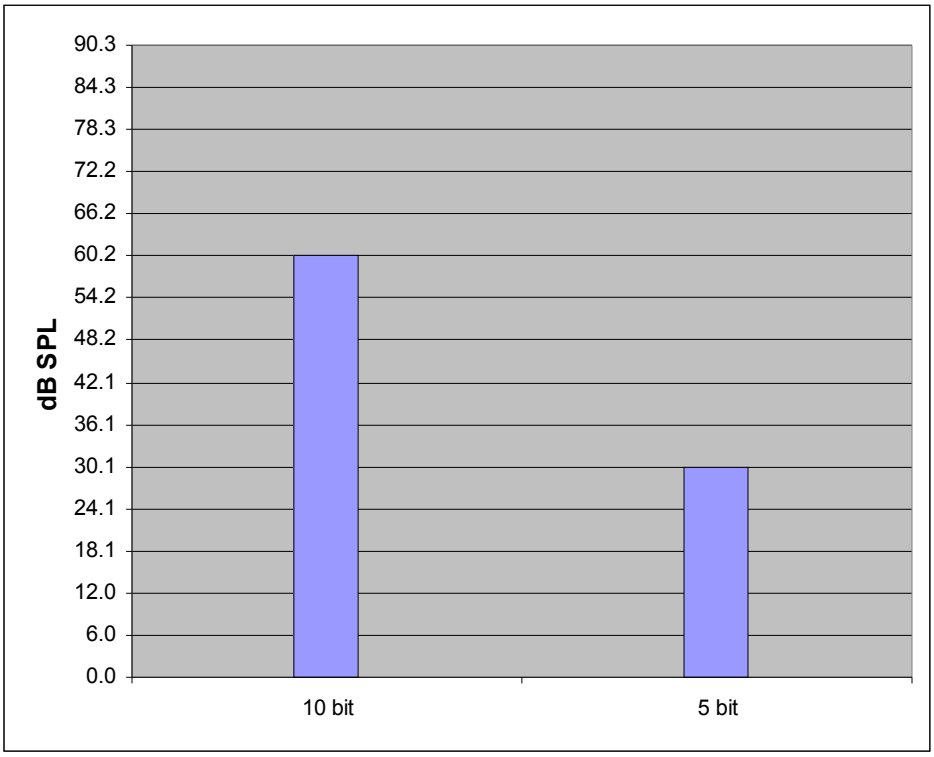
I 80 % løsningens skal allokeringen være konstant. Det vil sige at alle mulige bits skal anvendes ved hver allokering.

I 100 % løsningen skal allokeringen være dynamisk, og derfor er det relevant at undersøge hvad hver bit kan repræsentere.
Det vides at hver bit er en fordobling, og derved kan følgende udregnes:

$$20 \cdot \log_{10}(2) = 6{,}0206 \quad \text{(dB/bit)}$$

Det er også interessant at vide hvor stor en amplitude et 2-kompliment 16-bit tal kan repræsentere:

$$15 \cdot 20 \cdot \log_{10}(2) = 90{,}309 \quad \text{(dB)}$$



Figur 25. Illustration af allokering

Den største spektralkomponents amplitude i hvert Bark skal findes. Amplituden udregnes med følgende udtryk:

$$|a + j \cdot b| = \sqrt{a^2 + b^2}$$

Og kan omregnes til dB med følgende formel:

$$amp = 20 \cdot \log_{10}\left(\sqrt{a^2 + b^2}\right) \quad \text{(dB)}$$

Nu kan det udregnes hvor mange bits der er nødvendige til hvert Bark.

$$\text{antal bits} = \frac{\log_{10}\left(\sqrt{a^2 + b^2}\right)}{\log_{10}(2)} \quad \text{(bit)}$$

Resultatet skal altid rundes op til nærmeste heltal, så hørbar lyddata ikke bliver fjernet. Det skal dog bemærkes det maksimalt kan rundes op til 15.

MyP3 frames

Før der kan skrives frekvensdata til en MyP3 fil, skal det formateres til en bitstrøm. Alle værdier, som beskriver spektralkomponenter, skal afrundes til 2-kompliment 16 bit, og derefter indsættes i en frame.

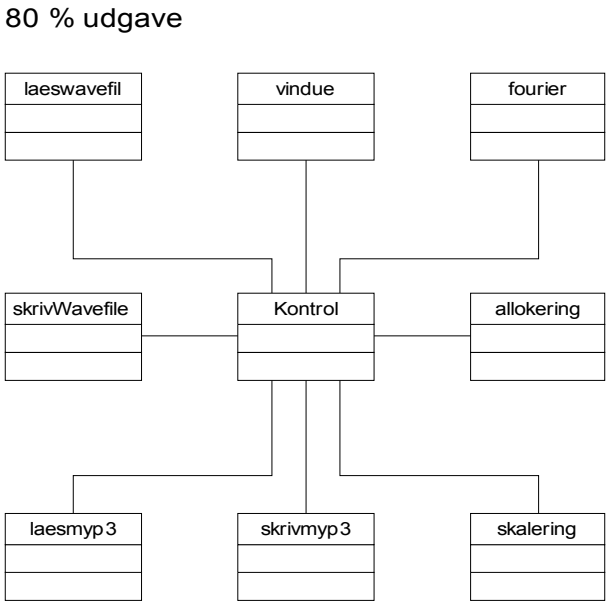
Header	Allokationsbit	Skaleringsfaktor	Kvantiseret frekvensdata	Zero-padding
0xFFF	$25 \cdot 4$ bits	$25 \cdot 4$ bits	….	0…

Tabel 1. MyP3 frame

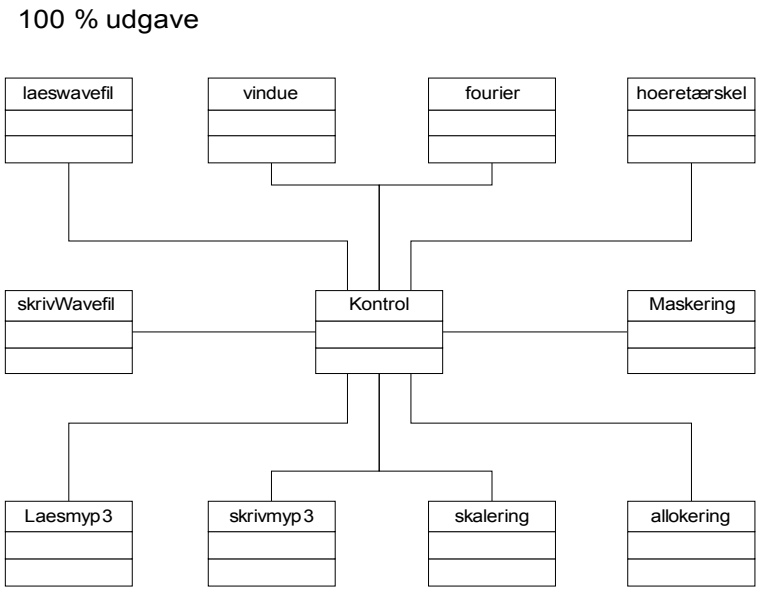
Zero-padding er nødvendigt, da summen af bits i hver frame, skal være et multiplum af 16.
Ifølge afsnittet, der omhandler Fourier transformation, er den første og sidste komponent i serien af spektralkomponenter altid et reelt tal, og derfor er den imaginære del 0, og skal ikke skrives i MyP3 framen.

Implementering

Koden til projektet implementeres med objekter tilsvarende de enkelte moduler i MyP3's kodek. Det vil sige, at der vil være et Hanning objekt, Fourier objekt m.m. Opbygningen af 100 % løsningen er magen til 80 % løsningen, dog bliver Fourier og allokering metoderne ændret til deres respektive funktionaliteter, og et høretærskel- og maskeringsobjekt bliver tilføjet.
Modulerne i MyP3's kodek vil blive implementeret som utilityklasser, der styres af en kontrolklasse. Følgende 2 diagrammer viser klassernes sammenhæng.



Figur 26. Diagram af 80 % løsningen.



Figur 27. Diagram af 100 % løsningen.

En mere detaljeret beskrivelse findes i bilag 6.

Lyttetest

For at få en fornemmelse for, hvor godt MyP3's kodek virker, er der blevet udviklet en lyttetest. Der foretages en lyttetest af både 80 % og 100 % løsningen, og testen vil blive gennemført i et lydisoleret rum. Lyttetesten udføres af et testpanel, der får udleveret et testskema, som skal udfyldes sideløbende med at der lyttes. Efterfølgende vil der drages en konklusion ud fra testresultaterne.

I 80 % løsningen anvendes ikke komprimering. Derfor vil den eneste forskel på den oprindelige WAVE PCM fil og WAVE PCM filen, der fremkommer ved dekodning af MyP3 filen, være kvantiseringsstøj. Kvantiseringsstøj opstår, da der under kodningen til en MyP3 fil, bliver foretaget en del beregninger, hvor resultatet ikke kan beskrives med det antal bits der er til rådighed, så der bliver foretaget afrunding. Da WAVE PCM filen kan genskabes næsten korrekt, bortset fra kvantiseringsstøj, bør der ikke være væsentlig hørbar forskel på de to filer.

For alligevel at kunne teste lidt på hvad der sker med den kvantiserede WAVE PCM fil, bliver signal/støj forhold (SNR) beregnet ved at sammenligne den oprindelige WAVE PCM fil med den kvantiserede. SNR udregnes således:

Samples fra oprindelig fil svarer til: $in_n[x]$
Samples fra kvantiseret fil svarer til: $out_n[x]$

$$\sigma_e^2 = \frac{1}{N}\sum_{n=0}^{N}\left(out_n[x] - in_n[x]\right)^2$$

$$\sigma_{out}^2 = \frac{1}{N}\sum_{n=0}^{N} out_n[x]^2$$

$$SNR = 20 \cdot \log_{10}\left(\frac{\sigma_{out}^2}{\sigma_e^2}\right)$$

Ved at dividere SNR med $20 \cdot \log_{10}(2)$ (dB), der svarer til en fordobling af lydstyrken, fås forholdet mellem musikken og støjen.

Det kan umiddelbart være svært at vurdere om tallet for SNR har noget at gøre med en god lydgengivelse eller ej. Det skyldes, at der i musik forekommer mange forskellige frekvenser, hvorved noget støj maskeres væk, og dermed ikke kan høres. Tallet kan dog sammenlignes med SNR for HiFi lyd, der ligger i niveauet 90-96 dB. Tallet kan sagtens være meget lavere, uden at støjen er hørbar, medmindre man har et professionelt lydanlæg og man er en trænet lytter.

De forskellige SNR værdier vil blive sammenlignet, for at undersøge om der er væsentlig forskel ved forskellige typer af musik.

I 100 % løsningen kan SNR ikke beregnes, da den kvantiserede WAVE PCM fil er blevet lossy komprimeret, og de to filer kan dermed ikke sammenlignes. Her er det til gengæld mere spændende at se på selve lyttetesten, da der er foretaget væsentlige ændringer i WAVE PCM filen. Ved anvendelse af maskering og viden om høretærsklen, er der nemlig fjernet lyddata, som øret alligevel ikke er i stand til at opfatte. Der vil derfor være større sandsynlighed for at testpanelet kan høre forskel, ved sammenligning af to filer.

Udførelse

Lyttetesten udføres af et testpanel på tre personer.

Der anvendes forskelligt slags lyd til testen, da der kan være forskel på støjniveauet i disse.

Der testes først på MyP3 formatet i forhold til den oprindelige WAVE PCM fil. Ved sammenligning af den oprindelige og kvantiserede lyd, anvendes udsnit på ca. 5-15 sekunder, der afspilles skiftevis gentagne gange. Da der anvendes så korte lydstykker, vurderes det at testpanelet godt kan huske kvaliteten af det ene stykke, når der skiftes til et nyt, og dermed kommentere en eventuel forskel. Alternativt skulle de to lydstykker mixes, så de skifter f.eks. efter hvert sekund, men det er ikke fundet nødvendigt.

Til 100 % løsningen sammenlignes der også imellem MyP3 og MP3 (bitrate 192), da det anses for at være CD-kvalitet.

Spørgsmålene kan variere afhængigt af lydstykket, da lyden kan være meget forskellig. Eksempler på spørgsmål kan være:

Spørgsmål:

- Læg mærke til om der er forskel på bass'en
- Er lyden "skinger"?
- Kan der høres støj?
- Lyt efter "clicking noises"

- Er der ”dropouts”?

Definitioner:

- ”clicking noises”: lyder som nålen på en grammofonpladespiller
- ”dropouts”: tabt lydinformation, lyder som afspilningen af en fedtet CD eller som om der spoles frem

Til besvarelse af hvert enkelt spørgsmål foretages en vurdering ud fra følgende skala:

- 5 - Ingen hørbar forskel
- 4 - Hørbar, men ikke irriterende
- 3 - Lidt irriterende
- 2 - Irriterende
- 1 - Voldsomt irriterende

Der kan eventuelt knyttes kommentarer til testen.

Testeksempler der anvendes til testen og deres kendetegn:

- Aerosmith Rockmusik, indeholder blandt andet trommer og elguitar
- hvidstoej Indeholder en kort støjsekvens
- Kastagnetter Kendetegnende lyd som ikke er blandet med baggrundsmusik
- Klaver Klassisk musik der indeholder en del klaver
- Sinus En klar tone der løbende stiger i frekvens

De anvendte testeksempler er valgt, fordi de indeholder en bred vifte af forskellige lyde og musik, så der kan sammenlignes om støj og komprimering varierer i forhold til dette. Sinustonen er valgt for at teste om kvaliteten afhænger af, hvor høje eller lave frekvenser der indgår i lyden, da der jo fjernes mere lydinformation ved høje og lave frekvenser.

Testskema

Eksempel på testskema ses nedenfor. Skemaer for hver lydfil til testen, findes på bilag 7.

Filnavn

Spørgsmål:	Kan der høres støj?				
Karakter, sæt X:	5	4	3 X	2	1
Kommentar:	Der opstår tydelig støj en enkelt gang				

DFT / FFT

Der er opnået en tidsforbedring ved ændring af Fourier transformationen fra DFT til FFT på 52 gange. Og ved ændring af Fourier transformationen fra invers DFT til invers FFT på 105 gange. Der er opnået en tidsforbedring fra 80 % løsningens DFT og iDFT til 100 % løsningens FFT og iFFT på 78 gange.

Den teoretisk bedst mulige tidsforbedring er på 57 gange, altså er der opnået en hastighedsforøgelse der er ca. 37 % bedre end det teoretisk mulige. En forklaring på dette er at FFT løsningen er bedre implementeret end DFT løsningen. For en beskrivelse og forklaring på hastighedsforøgelsen se bilag 8.

Høretærskel

Der er fra 80 % løsningen til 100 % løsningen opnået en reduktion af datamængden uden at der opstår hørbart støj. Fx er Klaver reduceret fra 2240 kbyte til 631 kbyte.
På bilag 9 kan det ses at datamængden reduceres mindst i Aerosmith og mest ved Sinus. Begrundelsen må være at Aerosmith indeholder frekvenser med kraftige amplituder og at Sinus signalet indeholder en del uhørbar støj, der bliver fjernet.

Maskering

Det ses i lyttetesten på bilag 10, at maskeringsmetode 1 giver den bedste gengivelse af lyden. Dog kan man se på bilag 9, at maskeringsmetoden ikke komprimerer datamængden væsentligt, når lydsignalet består af mange frekvenser.
Da gengivelsens kvalitet prioriteres højere end komprimering, vil maskeringsmetode 1 blive anvendt i 100 % løsningen.

Lyttetest af 80 % løsningen

Lyttetesten viser, at der ikke er hørbar forskel på MyP3 filerne og WAVE PCM filerne. Dog viser det sig, at støjen i hvidstoej.wav bliver kraftigere, hvilket er en indikation for, at der adderes støj under processen.
For testresultater se bilag 10.
Testdataene på bilag 9 viser, at filstørrelserne er 2 gange større end original filen, hvilket er forventet.

Følgende viser SNR for kodning og dekodning i 80 % løsningen.

Aerosmith		43 dB	
Hvidstoej	32 dB		
Kastagnetter		80 dB	
Klaver	52 dB		
Sinus	42 dB		

Det ses, at SNR varierer meget alt efter hvor mange frekvenser, hvor intensiv og hvor stor effekten er i lyden. Kastagnetter er derfor ret ubetydelig, da den indeholder mange pauser i lyden, der giver et forkert billede af SNR. Med et gennemsnitlig SNR på ca. 42 dB kan man konkludere at støjbidraget generelt er stort.

Lyttetest af 100 % løsningen

Lyttetesten på bilag 10 viser, at der ikke er hørbar forskel på lydkilden og den kvantiserede.
Der er følgende sammenlignet komprimeringen af med WAVE PCM til MyP3 og WAVE PCM til MP3.

	Aerosmith	Hvidstoej	Kastagnetter	Klaver	sinus
Original:	861	344	611	1120	1206
100 % løsning	862	506	598	631	360
MP3 bitrate192	235	95.5	167	306	329
Original / 100%	1	0.68	1.02	1.77	3.35
Original / MP3	3.66	3.6	3.66	3.66	3.67

Tabel 2. Viser filstørrelser i kilobytes og forholdet imellem den original, 100 % løsnings- og MP3 filen.

Forholdet imellem den original lydfil og MyP3 filen er i gennemsnit 1,56. Det vil sige, at MyP3 filen i gennemsnit fylder mindre end originalen, men som det også kan ses af tabellen, så fylder MyP3 filen også i nogle tilfælde mere end den originale lydfil. Til sammenligning er tallet for MP3 3,66.
Det ses, at der komprimeres godt ved sinus signalet, det matcher næsten MP3. Det kan begrundes med at en sinus er et simpelt signal, der svarer til en enkel spektralkomponent i et Bark. De andre Barks er tomme og derfor bliver der ikke allokeret bits til dem.
Derimod er komprimeringen af hvidstøj meget lille, fordi der er spektralkomponenter i alle Barks med en stor amplitude, og derved bliver der allokeret en meget større datamængde.

Under testforløbet er der blevet udvekslet MyP3 filer med en anden gruppe med det formål at dekode dem. Men efter analyse af de modtagne MyP3 filer har det vist sig, at de indeholdt fejl i frame header.

Konklusion

Til 80 % løsningen er konverteringsmetoderne implementeret vellykket. Desuden har vi med succes implementeret et Hanning vindue, som dog har den ulempe, at det forøger datamængden til det dobbelte. Vi har også implementeret DFT og iDFT til Fourier transformationen.
DFT og iDFT bruger en betragtelig mængde tid til beregningerne. Lyden er ikke komprimeret, og lyttetesten viser at der ikke er hørbart tab i lydkvaliteten. SNR er generelt dårligt, men støjen kan ikke høres. Der er implementeret skalering og allokering succesfuldt. Det er således lykkedes at fremstille et kodek i overensstemmelse med den 80 % løsning, der er beskrevet i målsætningen.

Til 100 % løsningen er det lykkedes, at implementere FFT og iFFT, der virker efter hensigten. Det har vist sig, at hastigheden, hvormed FFT og især iFFT beregnes, er hurtigere end forventet. Der er desuden implementeret maskering, der ikke komprimerer i forbindelse med anvendelse af høretærskel. Den manglende komprimering fra maskering er et tegn på, at maskeringen ikke har fjernet alle uhørbare frekvenser.
Høretærsklen er blevet implementeret uden, at der er opstået hørbart støj i processen. Det har også givet en god komprimering, som var fordelagtigt, da maskeringen ikke gav en nævneværdig komprimering.
Allokeringsmetoden i 100 % løsningen virkede efter hensigten.
Det er således lykkedes at fremstille et kodek i overensstemmelse med den 100 % løsning, der er beskrevet i målsætningen. Der er i alt opnået nogen grad af komprimering, afhængig af lydens karakteristik. Lyttetesten viste kun et hørbart tab i lydkvaliteten ved Sinus' diskantområde.

Det er ikke lykkedes os, inden for den givne tidsramme, at implementere den 120 % løsning, der er beskrevet i målsætningen.

Da 100 % løsningen var vores egentlige mål med projektet konkluderer vi, at vi har opnået at analysere og implementere det ønskelige.

Desværre løb tiden fra os til sidst så vores tidsplan blev forskudt med 3 uger hen mod slutningen af projektet.

Projektet har givet indblik i de problemstillinger man kommer ud for, når man ønsker at komprimere et lydsignal med en frekvensbaseret lossy metode, uden der må være hørbar ændring af det samlede lydbillede. Det har været lærerigt og inspirerende, at beskæftige sig med emner inden for lydbehandling.

Forbedringer til MyP3 standarden

For at reducere datamængde ville det være ideelt at bruge en anden vindue type end Hanning, idet Hanning vinduet fordobler den oprindelige datamængde. Datamængden kunne også blive reduceret yderligere, hvis diskret cosinus transformation blev anvendt i stedet for Fourier transformation, da resultatet kun vil være reelle tal, og derved halvere datamængden.
Temporal maskering ville også være interessant, da det foregår i tidsområdet, og derved er det muligt at vurdere hvor lang tid en eventuel maskering varer.
Desuden ville datamængden blive reduceret yderligere, hvis det kvantiserede frekvensdata blev komprimeret med en lossless algoritme, inden det bliver skrevet til en frame.

Referencer

[1] Ted Painter et.al: *Perceptual Coding of Digital Audio*, Department of Electrical Engineering, Arizona, USA.
[2] http://www.winamp.com
[3] http://www.research.okokatt.com/~bs/C++.html
[4] http://www.divx.com
[5] http://www.xivd.org
[6] http://encyclopedia.thefreedictionary.com/MP3
[7] http://www.parasitaere-kapazitaeten.net/Pd/hannings.htm
[8] http://www.mp3-converter.com/mp3codec/maskingeffects.htm