WAVELET-ANALYSE

Ada Gjermundsen

FORORD

Slik denne oppgaven var lagt opp skapte den mer forvirring enn forståelse hos meg. Det ble mer fokus på tekniske detaljer, enn selve metoden. Jeg tolket denne oppgaven slik at det var lagt stor vekt på analysen av musikksignalene. Jeg mener at man først må forstå analyseverktøyet man skal bruke for å kunne trekke ut den viktige informasjonen. Derfor har jeg lagt hovedvekten av oppgaven på å forstå hva denne analysemetoden faktisk gjør og hvordan man kan bruke den, isteden for å analysere mange forskjellige lydfiler.

Jeg har valgt å skrive en oppgave som forklarer wavelet analyse på en vennelig og enkel måte. Dette er noe jeg generelt savner ved realfag. Jeg tror at om du klarer å forklare vanskelige ting på en lettfattelig måte har du en større forståelse av faget enn om du roter deg bort i vanskelige begreper og utydelige formuleringer. Jeg har m.a.o. forsøkt å skrive om wavelets slik jeg ville likt å lære om wavelets. Etter å ha vært frustrert nesten hele prosjektoppgaveuka over oppgaven fant jeg ut at jeg trengte enn annen måte å tilnærme meg wavelets enn det oppgaven la opp til. Dermed kastet jeg oppgaven og gjorde det på min egen måte. Dette har ført til at jeg har fått en god forståelse for metoden, og jeg har også fått lyst til å lære mer. Med dette som utgangspunkt kan jeg godt forstå om dere mener at jeg ikke svarer på oppgaven. Det synes jeg er helt greit. Det viktige for meg var å lære meg hva wavelets analyse var, og det føler jeg at jeg har gjort. Akkurat hvordan dere bedømmer om punkt 1 b er med eller ikke ser jeg på som mindre viktig.

Jeg har allikevel programmert alle de ulike programmene som vi skulle lage. Disse legger jeg ved, men jeg velger å ikke bruke tid på å kommentere dem.

I denne prosessen har jeg hatt et fantastisk samarbeid med Sigurd Elias Stette. Jeg bruker også en programbit som Jan Fredrik Kismul har programmert for å få logaritmeskalaen for frekvensen i scalogrammet riktig.

INNLEDNING

I denne oppgaven vil jeg først og fremst se på wavlet transform som et analyseverktøy. Først går jeg litt inn på problemene vi kan få når vi bruker fourier transform. Deretter går jeg grundig gjennom hvordan en wavelet funksjon er bygd opp og metoder vi har for å manipulere funksjonen. Jeg ser også på hvordan selve wavelet funksjonen brukes på et signal. Til slutt ser jeg på wavelet transform av to signaler, og sammenligner wavelet analyse og fourier analyse.

PROBLEMER KNYTTET TIL FOURIER TRANSFORM

"Because the FFT is very effective, people have used it in problems where it is not useful—the way Americans use cars to go half a block. Cars are very useful, but that's a misuse of a car. So the FFT has been misused, because it's so practical"

Yves Meyer

Fourier analyse kan gi oss informasjon om de ulike frekvensene som er inneholdt i et signal. Dette er en god metode for lineære problemer, dvs. problemer som kan uttrykkes ved lineære ligninger. Fourier analyse egner seg derimot dårlig for aperiodiske signaler, signaler med mye støy og/eller forstyrrelser, kortvarige signaler eller signaler som endrer seg raskt og uforutsigbart. I signalbehandlig er det ofte de raske endringene som gir informasjonen vi søker. Et problem som oppstår med fourier transformasjonen er en konsekvens av at fourier funksjonene er bygget opp av sinus og cosinus funksjoner. Disse funksjonene har uendelig rekkevidde og derfor kan vi ikke få informasjon om lokaliseringen i tid/rom av frekvensene i signalet vårt. Denne informasjonen ligger skjult i fasene til sinus og cosinus funksjonene. Dette gjør vi kan få igjen det opprinnelige signalet vårt, forutsatt at vi ikke har manipulert fasene. En fourier transform oppfører seg derfor som om signalet er identisk for hele tidsperioden vi ser på.

"A local characteristic of the signal becomes a global characteristic of the transform"

Barbara Burke Hubbard

Om vi f.eks. skal se på innholdet av forskjellige stoffer i en blodprøve gir fourier transformen oss tilstrekkelig informasjon, men skal vi lokalisere en kreftsvulst er ikke dette en god nok metode. I fourier transform blir diskontinuitet (f. eks en kreft svulst) representert ved en superposisjon av alle mulige frekvenser. Å tolke ut i fra en slik superposisjon av frekvenser at vi har en diskontinuitet er ikke alltid mulig – og å lokalisere diskontinuiteten er enda verre.

Fourier transform er også sårbar for feil. Om du har en liten feil et signal (som kan være langt) kan dette ødelegge hele transformen siden informasjonen i den "feile" delen av signalet kan fordeles ut i hele transformen.

Et tredje problem med fourier transform er at man må anta at signalet kan dekomponeres til sinus- og cosinusfunksjoner.

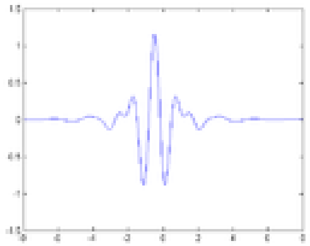
WINDOWED FOURIER TRANSFORM (WFT)

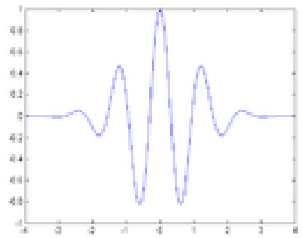
Problemene med å lokalisere de ulike frekvensene i tid/rom kan man delvis løse ved bruk av WFT. I denne metoden deler man signalet opp i intervaller og tar fourier transform for hvert intervall. På denne måten får man en viss informasjon om lokalisering/tid. Problemet med denne metoden er at informasjonen man får begrenses av lengden på intervallene. Opererer du med korte intervaller vil du få ganske god tids/steds informasjon for høye frekvenser, men du mister informasjon om de lave frekvensene da disse har for få svingninger innenfor intervallet. Velger du et lengre intervall får du med flere lave frekvenser, men informasjon om tid/sted tapes, spesielt for de høyeste frekvensene.

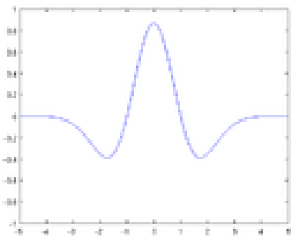
WAVELETS TRANSFORM

Wavelets transform ble utviklet for å løse tid/lokaliserings problemene tilknyttet fourier transform. Der du i fourier transform får ut verdier på frekvenser og energi, får du med wavelets transform ut informasjon om hvilke frekvenser signalet inneholder, hvor disse er lokalisert i signalet og hvilken intensitet hver frekvens har. Dette er med andre ord en god metode for å analysere diskontinuerlige signaler. Denne metoden brukes i dag innen en rekke fagfelt som seismologi, signalprosessering, bildebehandling, støyfjerning, kommunikasjon, informatikk, osv….

Wavelets transform bruker små bølgelignende funksjoner, der av navnet, som basisfunksjoner for transformen av signalet. Det er kanskje mer presist å kalle dem lokale bølgelignende funksjoner, i stedet for små. Når man skal benytte denne metoden tar man utgangspunkt i en Mother wavelet. En Mother wavelet gir grunnlaget for hvordan de bølgelignende funksjonene man benytter skal se ut. Hvilken Mother wavelet man velger avhenger av signalet man skal analysere og hvilken type informasjon man er interessert i. Det finnes mange typer Mother wavelets. Her er et eksempel på tre av dem som brukes mye:







Meyer Morlet Mexican Hat

(figurene har jeg hentet fra: http://en.wikipedia.org/wiki/Wavelet)

Mother waveleten man velger har altså en gitt form. Denne formen kan man manipulere videre og dermed lage uendelig mange basis wavelets funksjoner (Jeg velger i denne oppgaven å kalle wavelets funksjonene man får ved å manipulere Mother waveleten for basis wavelets.) Jeg tenker her at hver eneste wavelet funksjon av en gitt skala og tid vil være en basis som utspenner frekvensrommet). Siden fourier transformen bare har to basisfunksjoner, cosinus eller sinus, sier det seg selv at vi ved wavelets transformen har en kjempe fordel. Det matematiske uttrykket for Mother waveleten varierer avhengig av hvilken du velger. Jeg ser nærmere på det matematiske uttrykket for Morlet waveleten senere.

Basis wavelets

Mother waveleten kan manipuleres på to forskjellige måter. Man kan strekke den ut langs tids/posisjonsaksen og man kan trykke den sammen. På denne måten vil du endre frekvensen til waveleten. Den andre måten man kan manipulere en wavelet på er å flytte hele waveleten langs tids/posisjonsaksen. Dette kan man tenke på som om man holder et forstørrelsesglass over signalet. Du kan både flytte forstørrelsesglasset mot eller vekk fra signalet for å se på forskjellige detaljer(dilation), og du kan flytte det langs signalet for å se på ulike deler(translation). Waveleten gir altså muligheten til å se på funksjonens ulike frekvenskomponenter og studere frekvensene i ulik målestokk. Jeg skal se nærmere på hvordan disse manipulasjonene blir gjort litt senere.

Først litt matematikk

Wavelets transform er en matematisk metode, så vi har en del matematiske betingelser som må være oppfylt før vi kan bruke den.

Fourier transformen av waveleten kvadrert må være integrerbar:

$$C = \int_{-\infty}^{\infty} \frac{|\psi(\omega)|^2}{|\omega|} \, d\omega \qquad 0 < C < \infty$$

Denne betingelsen kalles *Admissibility condition*. Dette er en tilstrekkelig betingelse for at waveletene har en invers. Dette er nødvendig for at vi skal kunne få tilbake det opprinnelige signalet vårt. Wavelets som oppfyller dette kravet kalles for basic wavelets.
Eksistensen av C medfører at $\Psi(0) = 0$. Siden $1/\omega$ går mot uendelig når ω går mot null må absoluttverdien av Ψ kvadrert gå mot null ved denne grensen for at uttrykket ikke skal

divergere for ω. Det første punktet i en fourier transform angir gjennomsnittsverdien til funksjonen. Ved at denne er lik null sikrer vi oss at wavelet funksjonene har en oscilerende egenskap symmetrisk om tids/posisjonsaksen slik at arealet over og under aksen blir like stort. Derfor er det viktig at integralet av wavelets funksjonene blir null. Betingelsen at gjennomsnittet skal være lik null sikrer at vi får null i verdi for et kontinuerlig signal. Dette viser seg å være ganske så smart, men det kommer jeg tilbake til senere.

En annen betingelse er at wavelets funksjonene må ha begrenset energi.

$$E = \int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty$$

Dette sikrer oss at wavelets funksjonene er begrenset, i motsetning til cosinus- og sinusfunksjonene i fourier transformen. Dette er nødvendig for å tidsfeste/lokalisere hvor i signalet de ulike frekvensene befinner seg.

For komplekse wavelet funksjoner har vi også et krav om at fourier transformen må være reell og lik null for negative frekvenser.

Morlet

For å gå nærmere i detalj på hvordan selve manipulasjonen av wavelet funksjonene foregår er det lettere å se på en gitt form, altså en Mother wavelet. Morlet er en kompleks Mother wavelet. Den har en uendelig rekkevidde, men den avtar raskt pga e-faktoren i uttrykket. Vi kan derfor behandle den som en begrenset funksjon. Det matematiske uttrykket for Morlet wavelet funksjonen er:

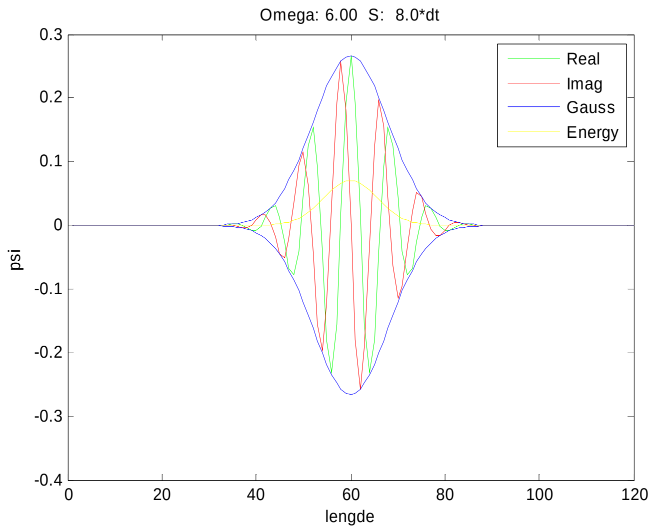
$$\Psi s, \tau \;=\; \pi^{\frac{-1}{4}} e^{i\omega\tau/s} * e^{-\tau^2/2s^2}$$

$\pi^{\frac{-1}{4}}$: normaliseringsfaktoren

$e^{i\omega\tau/s}$: kompleks sinuskurve

$e^{-\tau^2/2s^2}$: gaussisk omhyllingskurve

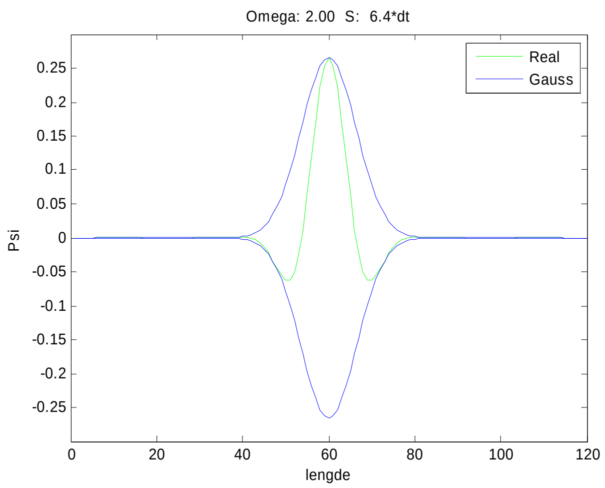
Den ser slik ut:



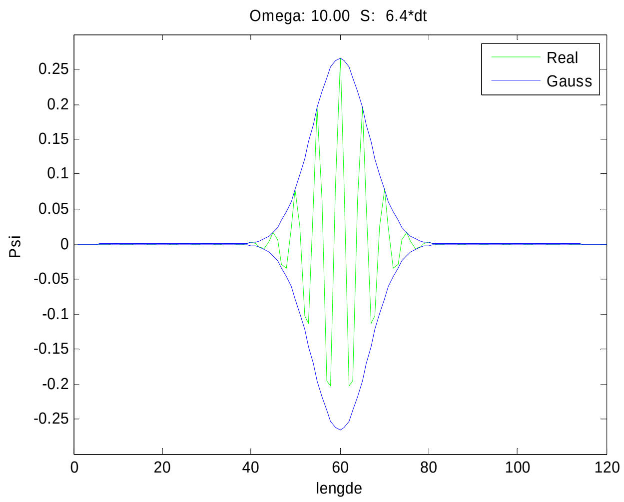
De ulike parameterne:

Bølgetallet, ω:
Når man har valgt Mother waveleten må man angi hvor mange oscilasjoner man vil ha i wavelet funksjonen.. Jo flere oscilasjoner vi har jo høyere frekvens vil vi ha inne i omhyllingskurven for en gitt størrelse av denne omhyllingskurven.

Denne figuren viser en wavelet funksjon med ω lik to.



Her har vi en figur med ω lik 10



De matematiske betingelsene for en wavelet viser at det er en stor fordel om integralet av waveleten er null. Det gjør at vi får en wavelet funksjon som er symmetrisk om tidsaksen. Når vi skal bestemme ω er de viktig at vi tester at denne betingelsen er oppfylt for waveleten vår. Setter vi ω lik 10 må vi strekke wavelet funksjonen mye mer ut for å fange opp lave frekvenser, enn om vi hadde satt ω lik 6. Dette gjør at vi får en dårligere tidsoppløsning. Når vi skal velge ω må vi derfor inngå et kompromiss. Waveleten vår må ha høy nok ω til å oppfylle integral-lik-null betingelsen, men ikke så høy at vi får en altfor dårlig tidsoppløsning. Dette testet jeg i ved å sette skalaen lik konstant, 8*dt, og variere ω:

Omega	Sum realdel	Sum imaginærdel
2.0	0.7207	0.0000
3.0	0.0592	0.0000
4.0	0.0018	0.0000
5.0	1.9846e-005	3.2074e-014
6.0	8.1105e-008	1.3788e-013
7.0	1.2218e-010	1.2766e-013
8.0	2.0358e-013	4.9426e-014
9.0	-1.3188e-013	1.6186e-013
10.0	-1.9200e-013	6.2816e-014

Integralet av den imaginære delen av wavelet funksjonen skal alltid være lik null siden denne er symmetrisk om et punkt. Vi ser at den er eksakt lik null for små verdier av omega for så å begynne å øke. Denne økningen kommer av avrundingsfeil hos datamaskinen. For den reelle delen av funksjonen ser vi at integrasjonsverdien blir mindre for høyere verdi av omega. Summen vil aldri bli null pga avrundingsfeil, så vi må selv sette en grense for hva vi mener er akseptabelt. For omega lik 6 er integalsummen mindre enn typiske avrundingsfeilverdier, derfor vil denne oppfylle integral-lik-null betingelsen. (Det kommer kanskje ikke helt frem av min tabell, men det er altså sånn:)

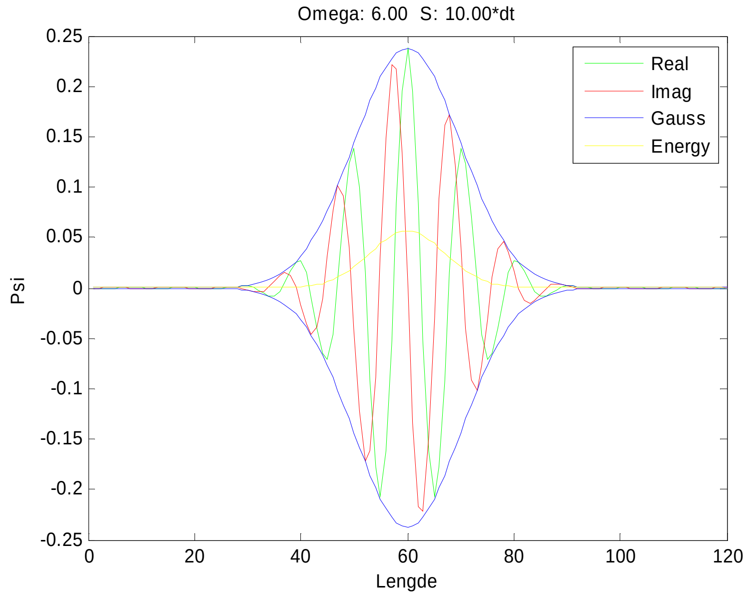
Skala:
Som jeg var inne på tidligere kan vi manipulere wavelet funksjonene ved å strekke dem ut langs tidsaksen eller presse dem sammen. Dette er det skaleringsfaktoren s som sørger for. Ved å endre s, endrer du omhyllingskurvens bredde og høyde. Siden antall oscillasjoner er fast vil endring av s gjøre at du endrer frekvensen til waveleten. Dette gjør at du automatisk tilpasser waveleten til de ulike signalkomponentene ved at den ved høye frekvenser er trykket sammen og bruker små intervaller slik at den får lokalisert de høye frekvensene. Ved lave frekvenser bruker waveleten store intervaller og får dermed også lokalisert lave frekvensene. Med andre ord så løser vi problemene vi får når vi bruker WFT!

Om vi velger en stor s, får vi en bred wavelet. En bred wavelet vil inkludere mer av tidsaksen, men vi mister informasjon om de små detaljene. Vi vil altså få en god frekvensoppløsning, men en dårligere tidsoppløsning. En smal wavelet vil gi oss en god tidsoppløsning, men en dårligere frekvensoppløsning. Dermed gjelder det å velge s slik at vi får tilstrekkelig sample av alle frekvensene i signalet.

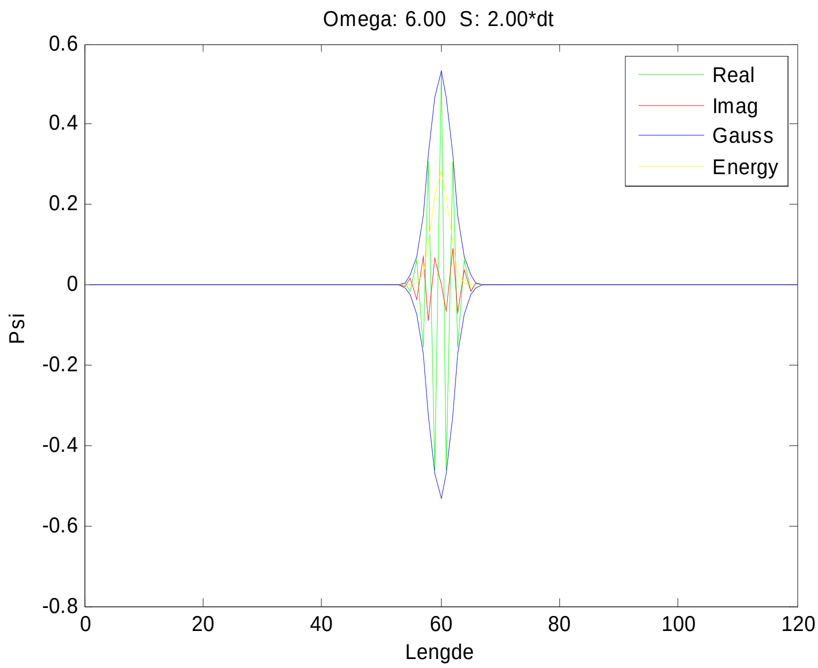
$S_j = S_o\, 2^{j*dj}$ for j = 0, 1, …, J,
S_o er startverdien for skalaen. $J = dj^{-1} \log_2(Ndt/S_o)$

(eksempler på neste side)

En stor skala gir en bred wavelet….



……. mens en liten skala gir en smal wavelet.



Forholdet mellom tidsoppløsningen og frekvensoppløsningen

Dette forholdet er enklest å løse analytisk. Dette blir ikke spesielt elegant når man skriver det i Word, men hovedtrekkene håper jeg å få frem. Først ser jeg på utrykket for den gaussiske omhyllingskurven for wavlet funksjonen, $e^{-\tau^2/2s^2}$. Dette er en gauss kurve med standardavvik, σ_τ lik s. Standardavviket er et mål for bredden på omhyllingskurven så jeg definerer denne til å være lik delta t. Så fourier omvender jeg wavelet funksjonen. Jeg bruker Ω isteden for ω for frekvens for ikke å blande det sammen med bølgetallet, ω.

$$F(\Omega)= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \Psi s,\tau \ * e^{-(i\Omega\tau)}\, d\tau$$
$$= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \pi^{\frac{-1}{4}} * e^{i\omega\tau/s} * e^{-\tau^2/2s^2} * e^{-i\Omega\tau}\, d\tau$$
$$= \frac{1}{\sqrt{2}}\frac{1}{\pi^{3/4}}\int_{-\infty}^{\infty} e^{((i\omega/s - i\Omega)\tau - \tau^2/2s^2)}\, d\tau$$

Dette er en eksponentialfunksjon på formen $e^{-a\tau^2 - 2b\tau}$. Der a = $1/2s^2$ og b = $i/2(\Omega - i\omega/s)$. Integralet fra ±∞ av denne funksjonen har løsningen $\sqrt{\pi/a}\ e^{b^2/a}$ (for a >0). (løsningen er hentet fra http://en.wikipedia.org/wiki/List_of_integrals#Exponential_functions)
Vi får at

$$F(\Omega)= \frac{s}{\sqrt[4]{\pi}} e^{-\frac{1}{2\frac{1}{s^2}}(\frac{\omega}{s}-\Omega)^2}$$

Dette er en gausiskfunksjon som gir omhyllingskurven for fourier funksjonen. Den har standardavvik σ_Ω lik 1/s. Ut i fra dette definerer jeg ΔΩ til å være lik $1/s$. Dette gir Δf lik $1/2\pi s$. Videre har vi fra Heisenbergs uskarphetsrelasjon at Δt * ΔE (energi) ≥ $h/4\pi$, der ΔE = Δf*h og h er Plancks konstant. Setter vi inn i Heisenbergs ligning får vi at

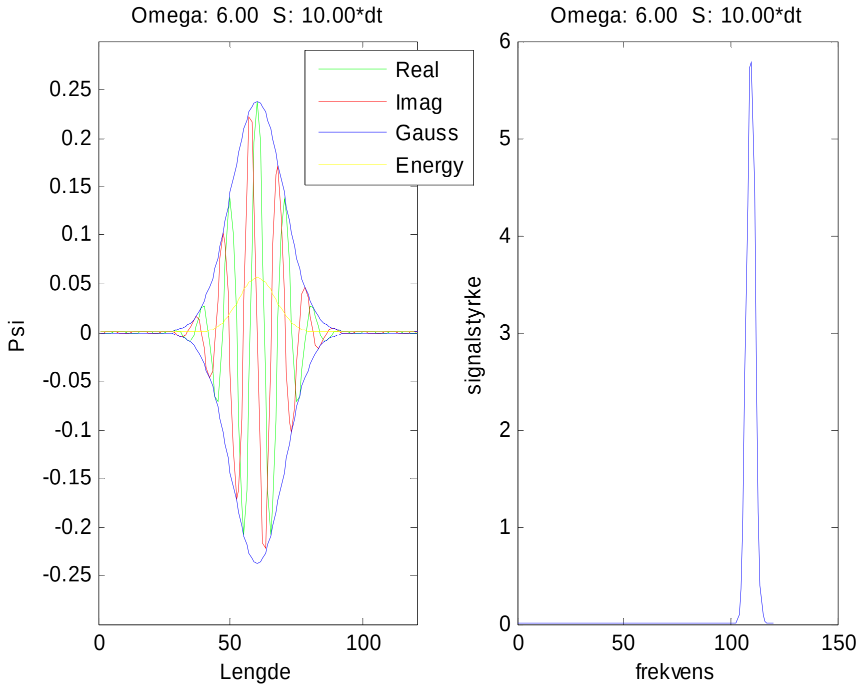
$$s * 1/2\pi s * h \geq h/4\pi$$

ut i fra dette ser vi at $\Delta t * \Delta f$ oppfyller Heisenbergs uskarpshetsrelasjon og at wavelets transformen begrenses av dette. Heisenbergs uskarphetsrelasjon sier at vi ikke kan bestemme to kvantemekaniske størrelser til en partikkel skarpt (presist) ved samme måling. En nøyaktig måling av den ene størrelsen vil medføre en mer unøyaktig måling av den andre størrelsen.

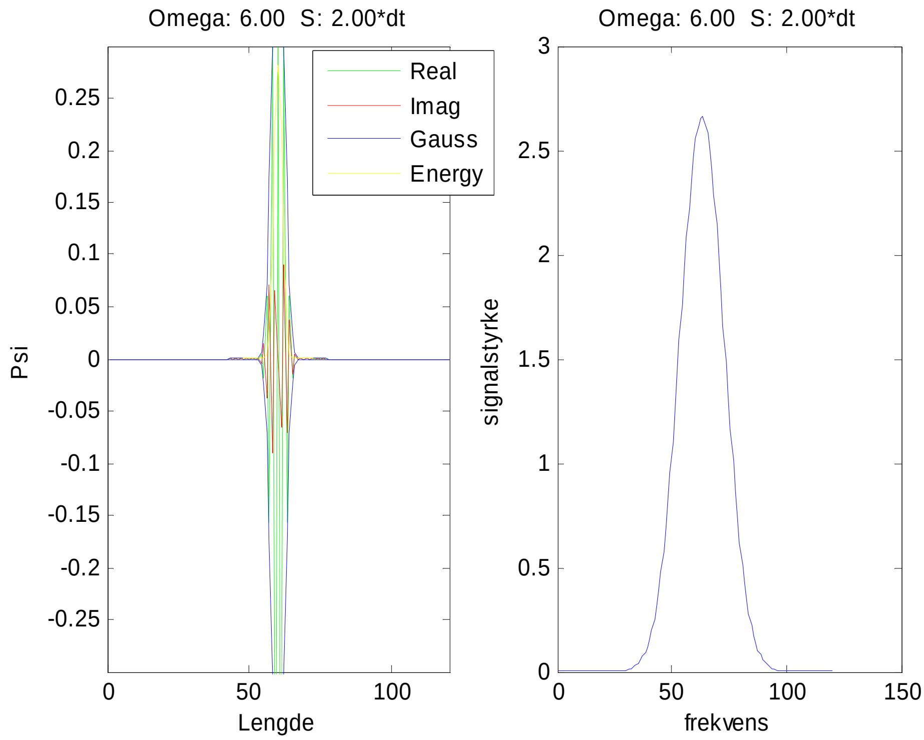
Her får vi klart ut forholdet mellom tidsbildet og frekvensbildet. Ved å øke tidsintervallet vi bruker får vi et mindre intervall til å bestemme signalets frekvenser. Slik at bedre tidsoppløsning gir dårligere frekvensoppløsning, og omvendt.

(eksempler på neste side)

Her ett eksampel med en bred skala. Dette gir dårlig tidsoppløsning, men god frekvensoppløsning noe man kan se av fourier transformen:



Hvis man istedenfor har en smal skala vil man få en god tidsoppløsning, men en mye dårligere frekvensoppløsning enn i eks over:

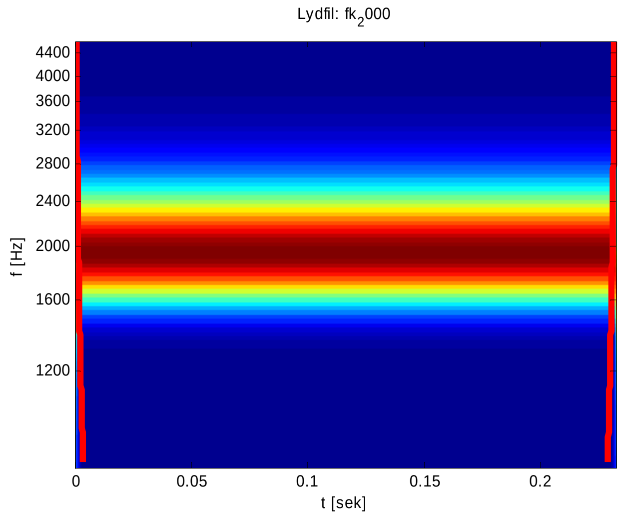


Men hva er så forholdet mellom skalafaktoren s og frekvensen f?

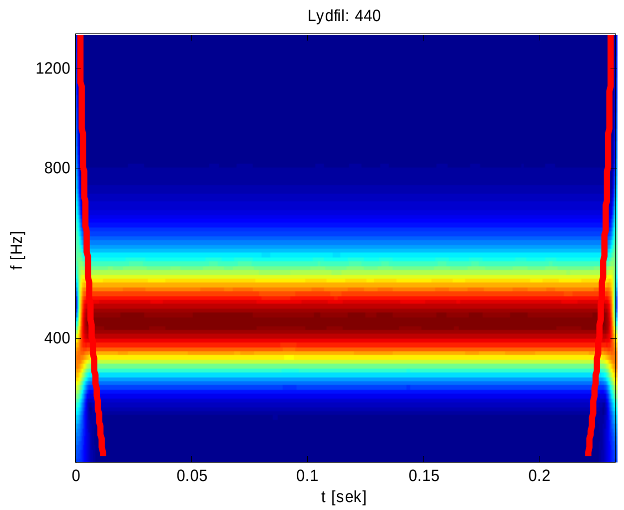
Vi ser av uttrykket for den fouieromvendte wavelet funksjonen at forventningsverdien til frekvensen, $\mu_f = \omega_o/s$ som gir oss at $f = {}^{\omega}/_{2\pi s}$

For å sjekke hvor godt dette stemte brukte jeg programmet jeg lagde oblig 1. Jeg lagde et signal med frekvens lik 1000Hz og et signal med frekvens lik 2000Hz. I tillegg fikk jeg et signal på 440Hz fra Sigurd Stette, i tilfelle programmet mitt ikke var riktig. Resultatet av den analytiske beregningen viste seg å stemme veldig godt med programmet for wavelet transform som jeg lagde:

I dette plottet brukte jeg signalet med frekvens lik 2000Hz. Vi ser at intensiteten er symmetrisk om linjen f = 2000 Hz.



Her brukte jeg signalet med frekvens lik 440Hz. Også her er intensiteten symmetrisk om frekvensverdien.



Av disse eksemplene mener jeg at det er grunnlag for å konkludere med at resultatene jeg kom frem til stemmer.

Wavelet transform i praksis

"You play with the width of the wavelet to catch the rythm of the signal. Strong correlation means that there is a little piece of the signal that looks like the wavelet"

Yves Meyer

Når man nå bruker disse ulike basis wavelets funksjonene på signalet vil man få en stor transform verdi, en koeffisient, når waveleten stemmer godt overens med signalet. En wavelet koeffisient er m.a.o. et mål på korrelasjonen mellom wavelet funksjonen og den korresponderende delen av signalet.. Høy korrelasjon gir store positive verdier. Når signalet og wavelet funksjonen ikke er i fase får vi negative koeffisientverdier. Wavlet funksjonen er var på endringer. Når du har et konstant signal får du koeffisienter lik null. Dette gjør at wavelet transform er helt overlegen fourier transform når det kommer til å komprimere signal siden man tar vare på endringene i signalet og ikke kontinuiteten.

"Wavelet analysis is a way of saying that one is sensitive to change. It's like our response to speed. The human body is only sensitive to accelerations, not to speed"

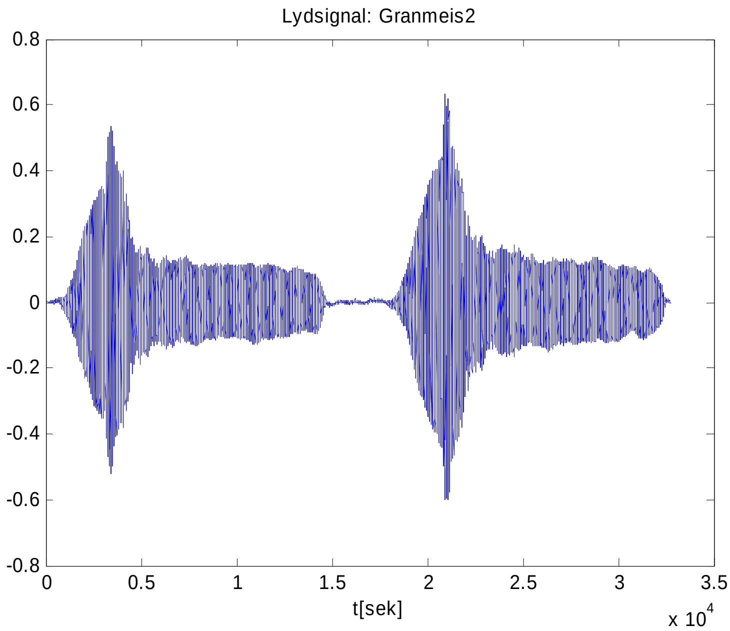
Yves Meyer

ANALYSEN

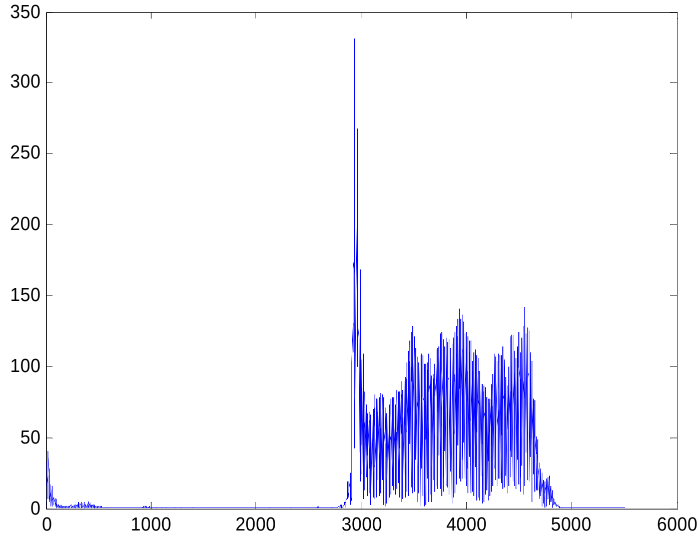
Siden jeg har valgt å legge hovedvekten av oppgaven på å forstå hva wavelet transform faktisk er, kommer jeg ikke til å legge like mye arbeid i denne delen. Som jeg nevnte i innledninger har jeg den oppfatningen at om du forstår verktøyet du bruker til å gjøre analysene, vil tolkningen av resultatene gå (i de fleste tilfeller) greit. Jeg tar derfor bare for meg to lydfiler.

<u>Granmeis:</u>

Lydsignalet av en granmeis som synger to toner:

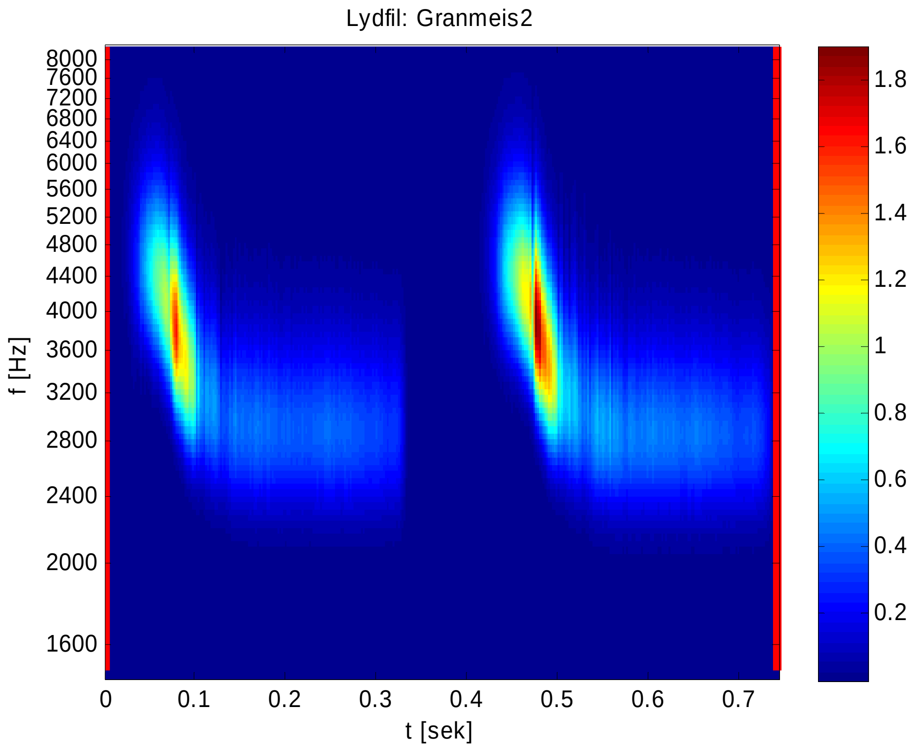


Fourier transform:

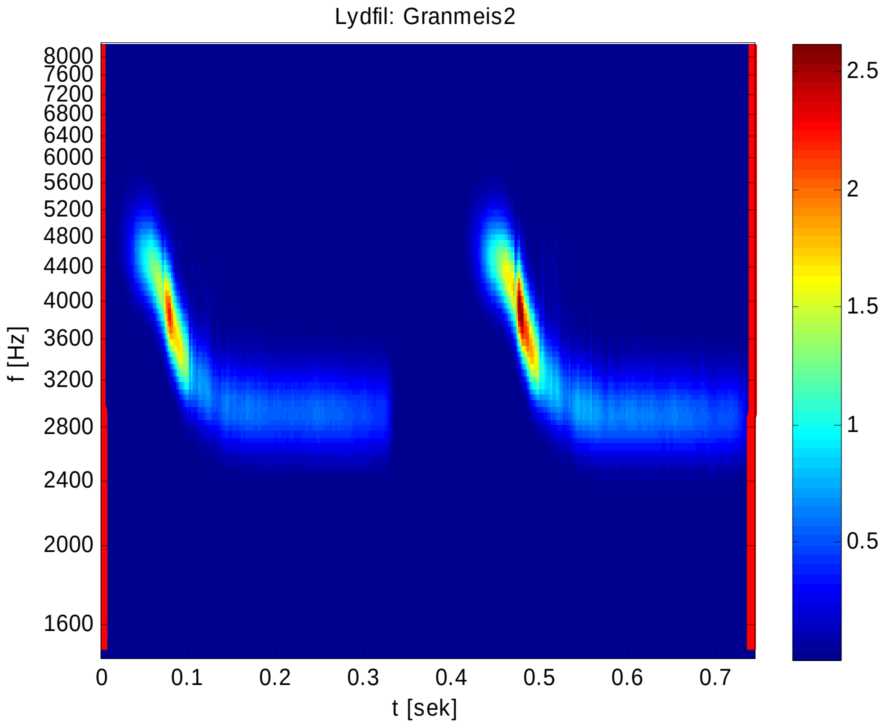


Her ser vi at granmeisens sang har frekvenser som ligger i intervallet mellom, grovt sagt, 3000Hz og 5000Hz. Dette er normale frekvenser for fuglesang. Vi ser også at tidsinformasjonen i signalet er borte siden de to tydelige tonene i signalet ikke kommer med i fourierspekteret.

Wavelet transform:



I dette skalogramet ser vi derimot at all tidsinformasjonen i granmeis-signalet vises. Her kommer de to tonene tydelig frem. Vi har høy intensitet i skalogrammet for de samme tidene som vi har høy intensitet i signalet vårt. I tillegg vises frekvensene som er inneholdt i tonene til meisen for enhver tid. Vi ser at han ikke synger en ren tone, men begynner med en ganske høy frekvens på mellom 4400Hz og 3600Hz før han glir ned til lavere frekvenser. Disse synger han også med svakere intensitet. Grunnen til at vi ved hvert tidsintervall får et såpass bredt utslag av frekvenser er at wavelet funksjonen registrerer grad av korrelasjon mellom funksjonen og signalet. Ved høy intensitet har vi sterkt samsvar mellom waveleten og signalet, ved lavere intensitet har vi også samsvar men ikke fullt så godt, men vi får også i det tilfellet positivt utslag for korrelasjon. Vi kunne fått bedre frekvensoppløsning ved å bruke en høyere omega, men dette går utover tidsoppløsningen. I det forrige skalogramet har jeg satt omega lik 6.0. Setter jeg istedenfor omega lik 12.0 får jeg frem dette skalogrammet:

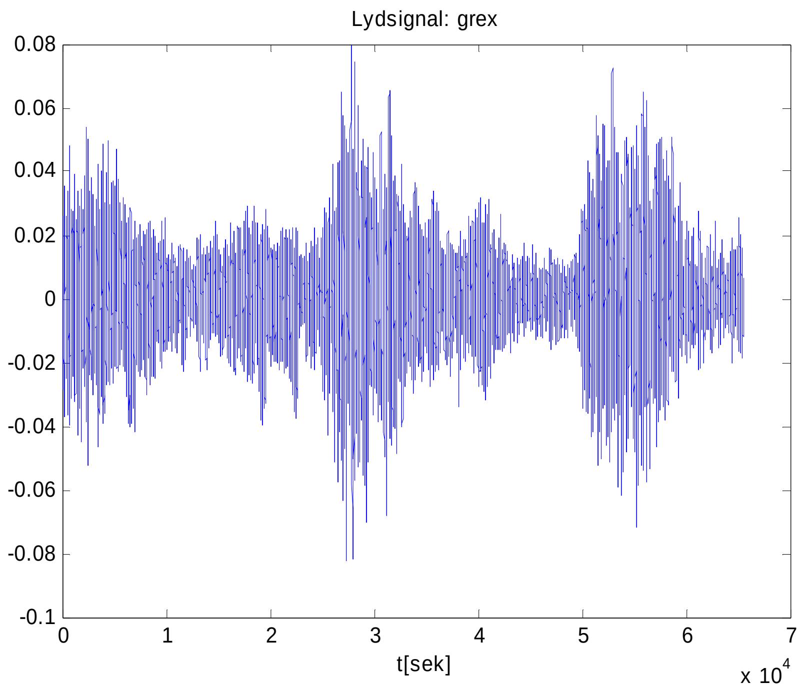


Her ser vi at frekvensoppløsningen er bedre ved at de ikke er så dratt ut i spekteret. Vi kan se at høyest intensitet ligger mellom 4000Hz og 3600Hz. Her kan vi altså forminske frekvensintervallet vårt med 400Hz. Tidsoppløsningen er dårligere, men i dette tilfellet er ikke det så avgjørende. Vi ser at vi allikevel får frem god tidsinformasjon. Dette er en av tingene som gjør wavelet transform til en så anvendelig analysemetode. Du kan tilpasse metoden etter hvilke data du selv er interessert i å analysere.

<u>Grex Vocalis:</u>

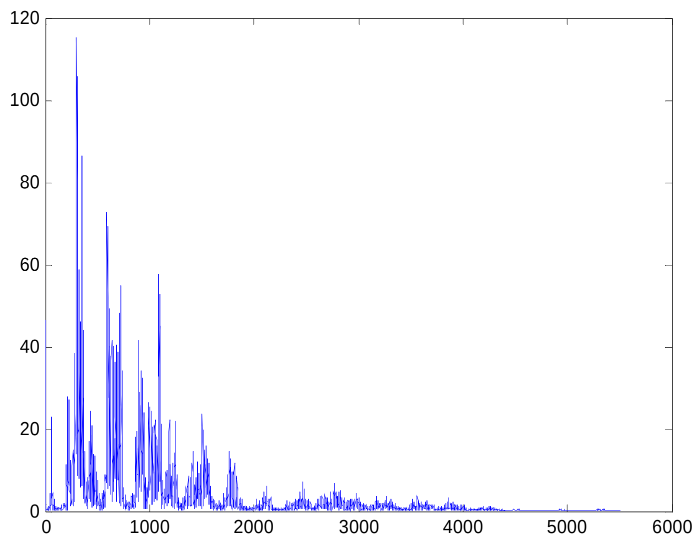
I dette eksempelet valgte jeg ut en del av Grex Vocalis der de synger "Revolusjonens Røst". I denne filen har vi et signal med et kor, noe som gjør at vi får ganske andre grafer enn i eksempelet over der vi kun hadde en ensom meis.

Lydsignalet:



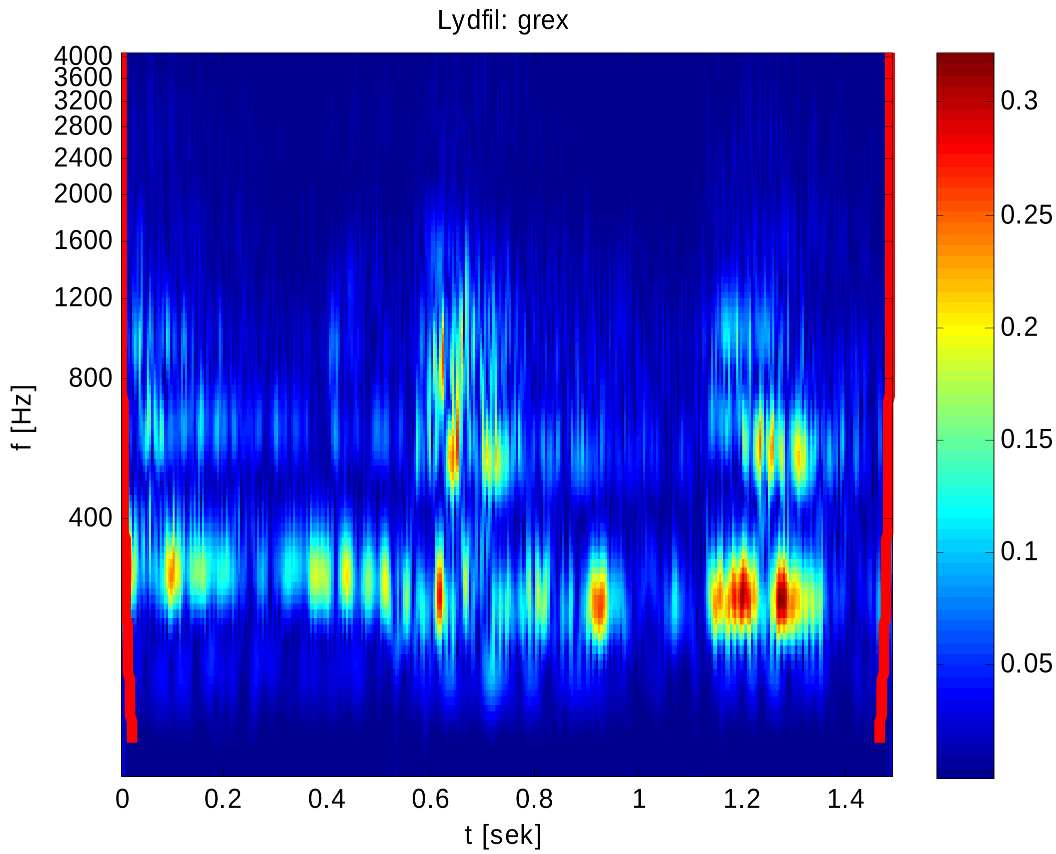
Lydsignalet gir oss informasjon om når det er høy intensitet og når det er lav, men vi kan ikke ut i fra dette si om det er en stemme som synger med stor intensitet eller om det er flere stemmer med lavere intensitet som synger samtidig.

Fourier transformen:



Her får vi en god oversikt over de ulike frekvensene som er inneholdt i signalet, men som vi har sett tidligere er all tidsinformasjon borte.

Wavelets transformen:



Her får vi virkelig frem fordelene ved wavelets transform. De to andre metodene, lydfilbildet og fourier transformen, kan ikke gi oss i nærheten den mengde informasjon som wavelets transformen kan. Ut i fra skalogrammet kan vi lese at vi har tre stemmer i kor arrangementet. Det er 1.-og 2. tenorstemme og en altstemme. Grex Vocalis synger a cappella, og vi kan se at den dypeste stemmen holder en jevn rytme. Har man litt mer kunnskap om musikk en det jeg har kan man sikkert lese ut informasjon om overtoner, kor arrangement osv.

Konklusjon:

Wavelets transform er helt overlegen fourier transform når det kommer til å behandle diskontinuerlige signaler eller om vi har med analyser der det er nødvendig å plassere frekvensene i tid/rom. Skal vi borre etter olje kan fourier analyse gi oss informasjon om det er mulighet for å finne olje, men vi vil ikke få noe informasjon om hvor dypt ned vi trenger å borre for å komme frem til oljen. Det kan vi derimot få informasjon om ved wavelet analyse. På den annen side kreves det mer ressurser for å utføre en wavelet transform enn en fourier transform og det er ikke alltid nødvendig å vite plassering i tid/rom. Er vi interessert i mengder ulike stoffer inneholdt i en væske er det sløsing av ressurser å starte med en wavelet analyse. Man må velge analysemetode etter hvilket type signal du jobber meg og etter hvilken informasjon du er interessert i å vite.

Ps I: Den røde streken i skalogrammene mine merker av de områdene der ikke hele waveleten var innenfor tidsignalet. Resultatene under denne streken kan man ikke ta altfor seriøst. Ingen av de signalene jeg brukte ga noe særlig utslag under denne så jeg har ikke kommentert den underveis.

Ps II: Jeg har programmert alle oppgavene dere ga i oppgaveteksten, og jeg har brukt noen av programmene, spesielt programmet fra oppgave 3, i oppgaven min. Jeg legger ved alle programmene men jeg har hverken tid eller plass til å kommentere dem.

KILDER

Bøker:

- "The World According to Wavelets: The Story of a Mathematical Technique in the Making" av Barbara Burke Hubbard, kap. 2
- "The Illustrated Wavelet Transform Handbook" avNapler Addison, kap 1 og 2
- "Wavelet transform", kompendium av Arnt Inge Vistnes

Nettsteder:

-http://paos.colorado.edu/research/wavelets/wavelet3.html
-http://fag.grm.hia.no/fagstoff/perhh/htm/fag/MATEM/datwwww/wavelet.htm
-http://matlab.izmiran.ru/help/toolbox/wavelet/scal2frq.html
-http://en.wikipedia.org/wiki/Wavelet_analysis
-http://en.wikipedia.org/wiki/Morlet_wavelet
-http://en.wikipedia.org/wiki/List_of_Integrals_(exponential_functions)
-http://en.wikipedia.org/wiki/Heisenberg_uncertainly_relation
-http://en.wikipedia.org/wiki/Normal_Distribution

Programfilene:

Program 1a:

```
function wavelet1a

omega = 6.0;
dt = 1.0;
s = 8*dt;
nw = 120;
nn = 120/2;
sum = 0;

for n=1:nw
    psi(n) = pi^(-1/4)*sqrt(dt/s)*exp(1i*omega*((nn-n)*dt/s))*exp(-(((nn-
n)/s)^2)/2);
    sum = sum + psi(n);
end

figure;
plot(real(psi), 'g')
hold on
plot(imag(psi), 'r')
hold on
plot(abs(psi), 'b')
hold on
plot(abs(psi).^2,'y')
hold on
plot(-abs(psi), 'b')
legend('Real','Imag','Gauss','Energy');
xlabel('lengde');
ylabel('psi');
tittel=sprintf('Omega: %4.2f  S: %4.1f*dt', omega,s );
title(tittel);

sum;
a=real(sum);
b=imag(sum);
```

Program 1 b:

```
function wavelet1b

close all;


dt = 0.8;
s = 8*dt;
nw = 120;
nn = 120/2;
frek = linspace(0,nw/dt, nw);
```

```
disp(' Sum:                Realdel:    Imaginærdel:')
for k=2:10
    omega = k;
    sum = 0;
for n=1:nw
    psi(n) = pi^(-1/4)*sqrt(dt/s)*exp(1i*omega*((nn-n)*dt/s))*exp(-(((nn-
n)/s)^2)/2);
    sum = sum + psi(n);
end

    %figure;
    %f = fft(psi);
    %plot(frek, abs(f))
    %xlabel('frekvens');
    %ylabel('signalstyrke');
    %title(sprintf('Omega: %4.1f', omega));
    %sprintf('Omega: %4.2f', omega)

    disp([sum' real(sum)' imag(sum)'])


    figure;
    plot(real(psi), 'g')
    hold on
    %plot(imag(psi), 'r')
    %hold on
    plot(abs(psi), 'b')
    hold on
    %plot(abs(psi).^2,'y')
    hold on
    plot( -abs(psi), 'b')
    legend('Real','Gauss');
    xlabel('lengde');
    ylabel('Psi')
    axis([0, nw, -0.3, 0.3]);
    tittel=sprintf('Omega: %4.2f  S: %4.1f*dt',omega, s);
    title(tittel);

end
```

<u>Program 1 c:</u>

```
function wavelet1c

clear all


dt = 1.0;
omega = 6.0;
nw = 120;
nn = 120/2;
frek = linspace(0,nw/dt, nw);


for k=2:10
```

```
    s = k*dt;
    sum = 0;
for n=1:nw
    psi(n) = pi^(-1/4)*sqrt(dt/s)*exp(1i*omega*((nn-n)*dt/s))*exp(-(((nn-
n)/s)^2)/2);
    sum = sum + psi(n);
end

sum
    figure;

    f = fft(psi);
    %subplot(1, 2, 2)
    plot(frek, abs(f))
    xlabel('frekvens');
    ylabel('signalstyrke');
    title(sprintf('Omega: %4.2f  S: %4.2f*dt', omega, s));
    sprintf('Omega: %4.2f', omega)
    %hold on
    %disp(' Sum:                Realdel:    Imaginærdel:')
    %disp([sum' real(sum)' imag(sum)'])
    figure;
    %subplot(1, 2,1)
    plot(real(psi), 'g')
    hold on
    plot(imag(psi), 'r')
    hold on
    plot(abs(psi), 'b')
    hold on
    plot(abs(psi).^2,'y')
    hold on
    plot( -abs(psi), 'b')
    legend('Real','Imag','Gauss','Energy');
    xlabel('Lengde');
    ylabel('Psi')
   % axis([0, nw, -0.6, 0.6]);
    tittel=sprintf('Omega: %4.2f  S: %4.2f*dt',omega, s);
    title(tittel);

end
```

Program 1 e_1:

```
function wavelet1e_1

dt = 0.8;
omega = 6.0*dt;
s = 8*dt;
nw = 120;
nn = 120/2;
frek = linspace(0,nw/dt, nw);
sum=0;


for n=1:nw
    psi(n) = pi^(-1/4)*sqrt(dt/s)*exp(1i*omega*((nn-n)*dt/s))*exp(-(((nn-
n)/s)^2)/2);
```

```
    sum = sum + psi(n);
end

G = fft(psi);
m = abs(G).^2;
plot(frek, m);
xlabel('Frekvens[Hz]');
ylabel('FFT psi')
tittel=sprintf('Skala: %4.2f og Omega: %4.2f*dt', s, omega);
title(tittel);
m_max = max(m);
m_2 = m_max/2;

n_1=0;

for j= 1:nw
   if(abs(m(j)-m_2)<dt) && (n_1 == 0)
       n_1 = j;
    end
    if(abs(m(j)-m_2)< dt)
        n_2 = j;
    end
    if(abs(m(j)-m_max))<dt
        n_3=j;
    end
end


halv_bredde= n_2 - n_1;
rel_bredde= halv_bredde/n_3;
sprintf('Relativ båndbredde: %e', rel_bredde)
```

Program 1 e_2:

```
function wavelet1e_2

close all

dt = 0.8;
omega = 6.0*dt;
s = 16*dt;
nw = 120;
nn = 120/2;
frek = linspace(0,nw/dt, nw);
sum=0;


for n=1:nw
    psi(n) = pi^(-1/4)*sqrt(dt/s)*exp(1i*omega*((nn-n)*dt/s))*exp(-(((nn-
n)/s)^2)/2);
    sum = sum + psi(n);
end

G = fft(psi);
m = abs(G).^2;
plot(frek,  m);
xlabel('Frekvens[Hz]');
```

```
ylabel('FFT psi')
tittel=sprintf('Skala: %4.2f og Omega: %4.2f*dt', s, omega);
title(tittel);
m_max = max(m);
m_2 = m_max/2;

n_1=0;

for j= 1:nw
   if(abs(m(j)-m_2)<dt) && (n_1 == 0)
       n_1 = j;
    end
    if(abs(m(j)-m_2)< 8.1)
        n_2 = j;
    end
    if(abs(m(j)-m_max))<dt
        n_3=j;
    end
end




n_1
n_2
n_3
halv_bredde= n_2 - n_1;
rel_bredde= halv_bredde/n_3;
sprintf('Relativ båndbredde: %e', rel_bredde)
```

Program 1 f:

```
unction wavelet1f
close all

dt = 1.0;
omega = 6.0*dt;
s = 8*dt;
nw = 120;
nn = 120/2;
sum = 0;

for n=1:nw
    psi(n) = pi^(-1/4)*sqrt(dt/s)*exp(1i*omega*((nn-n)*dt/s))*exp(-(((nn-
n)/s)^2)/2);
    sum = sum + psi(n);
end

figure;
plot(real(psi), 'g')
hold on
plot(imag(psi), 'r')
hold on
plot(abs(psi), 'b')
hold on
plot(abs(psi).^2,'y')
hold on
plot(-abs(psi), 'b')
legend('Real','Imag','Gauss','Energy');
```

```
xlabel('lengde');
ylabel('psi');
tittel=sprintf('Omega: %4.2f  S: %4.1f*dt', omega,s );
title(tittel);
gauss = abs(psi).^2;
m_2 = max(gauss)/2;
n_1=0;

for j= 1:nw
   if(abs(gauss(j)-m_2)<1e-2) && (n_1 == 0)
       n_1 = j;
    end
    if(abs(gauss(j)-m_2)< 1e-2)
        n_2 = j;
    end
end

n_1
n_2
halv_bredde= n_2 - n_1;
sprintf('Halvbreddeverdi gauss: %e', halv_bredde)
```

Program 2:

```
function wavelet2

close all;
m = 'Sound (WAV) file';
c = 'Granmeis';
N = 2^15;
nstart = 17000;
nslutt = nstart+N-1;
[y, Fs, type] = wavread(c, [nstart nslutt]);

plot(y)
figure

T=N/Fs;
dt = T*(N-1)/N;
nw=100;
nn=nw/2;
s0 = 2;
dd = 0.125;
wlsum = 0;
omega = 6.0;

for jj=1:40
    s = s0*2.0^(jj*dd)*dt;
    sca(jj)= s;
    %nw = floor(10*s);
    %nn = floor(nw/2);

    for n=1:nw
    psi(n) = pi^(-1/4)*sqrt(dt/s)*exp(i*omega*((nn-n)*dt/s))*exp(-(((nn-
n)/s)^2)/2);
    end

    for j=1:N
```

```
        lower = nn-1;
        upper = nn;
        if (j<nn)
            lower = j-1;
        end;
        if(j>N-nn)
            upper =N-j;
        end;
        wlsum = 0.0 + i*0.0;
        for k=nn-lower:nn+upper
            wlsum = wlsum + psi(k)*y(j+k-nn);
        end
        skalogram(jj,j) = abs(wlsum)^2;
    end;
end;

imagesc(skalogram)
```

Program 3:

```
function wavelet3_2

close all;
c = 'grex';
N =2^16;
nstart = 100000;
nslutt = nstart+N-1;
[f, Fs, type] = wavread(c, [nstart nslutt]);
wavwrite(f, Fs, 'grex_2');
h=f(:,1);
plot(h)
xlabel('t[sek]')
title(sprintf('Lydsignal: %s', c));
figure;


%%NB!!! FRESAG: SJEKK UT!!Fc*Fs/s

FFT_f = fft(h);
m = abs(FFT_f);
M = N/8;
fr=linspace(0,Fs,N);%Øvre grense for plot (hvor stor del som skal tas med)
plot(fr(1:M), m(1:M))

figure;

omega = 6.0;
n= 1:N;
kk = [ones(1, N/2+1) -(ones(1, N/2-1))];
T =N/Fs;
t = linspace(0,T*(N-1)/N, N);
s0 = 50.0;
dd = 0.03;
arg = 2*pi.*(n-1)./N;

for k=1:100
    ss = s0*2.0^(k*dd);
    sca(k)= ss;
```

```
    frek(k)=Fs./ss*(omega/(2*pi));
    argument =-0.5.*(ss.*arg-omega).^2;
    FFT_psi = sqrt(2.0*ss)*pi^(0.25).*exp(argument);
    FFT_psi(1)=0;
    FFT_psi = FFT_psi.*(kk>0);
    G = FFT_f.*FFT_psi';
    inv_G = ifft(G);
    abs_G(k,:) = abs(inv_G);

end

%Fc = centfrq('morl');
%F_rek = Fc./(sca.*(1./Fs));
%disp(F_rek')
figure;
p=1/Fs;
%frek = scal2frq(sca, 'morl',p )./(2*pi);
imagesc(t, frek, abs_G);
xlabel('t [sek]');
ylabel('f [Hz]');
title(sprintf('Lydfil: %s', c));
set(gca,'YDir','normal','Yscale','log',
'Ytick',0:400:max(frek),'YTickLabel',0:400:max(frek))
hold on
%disp('     Scale      Frequency')
%disp([sca' frek'])
%plot(sca, 1./frek)

Tp = 1./frek;
halv_bredde = 0.5*omega.*Tp;
halv_bredde_venstre= T-halv_bredde;
plot(halv_bredde, frek, 'r', 'LineWidth', 4)
hold on
plot(halv_bredde_venstre, frek, 'r', 'LineWidth', 4)
```