

Tips til å ekstrahere data ut fra et waveletdiagram for lage syntetisk lyd

Vi kan bruke kontinuerlig wavelettransformasjon med Morlet-wavelets som utgangspunkt for å lage god syntetisk lyd. I dette skrevet er det gitt ett eksempel på dette. Det er en detalj i bokfinksang.

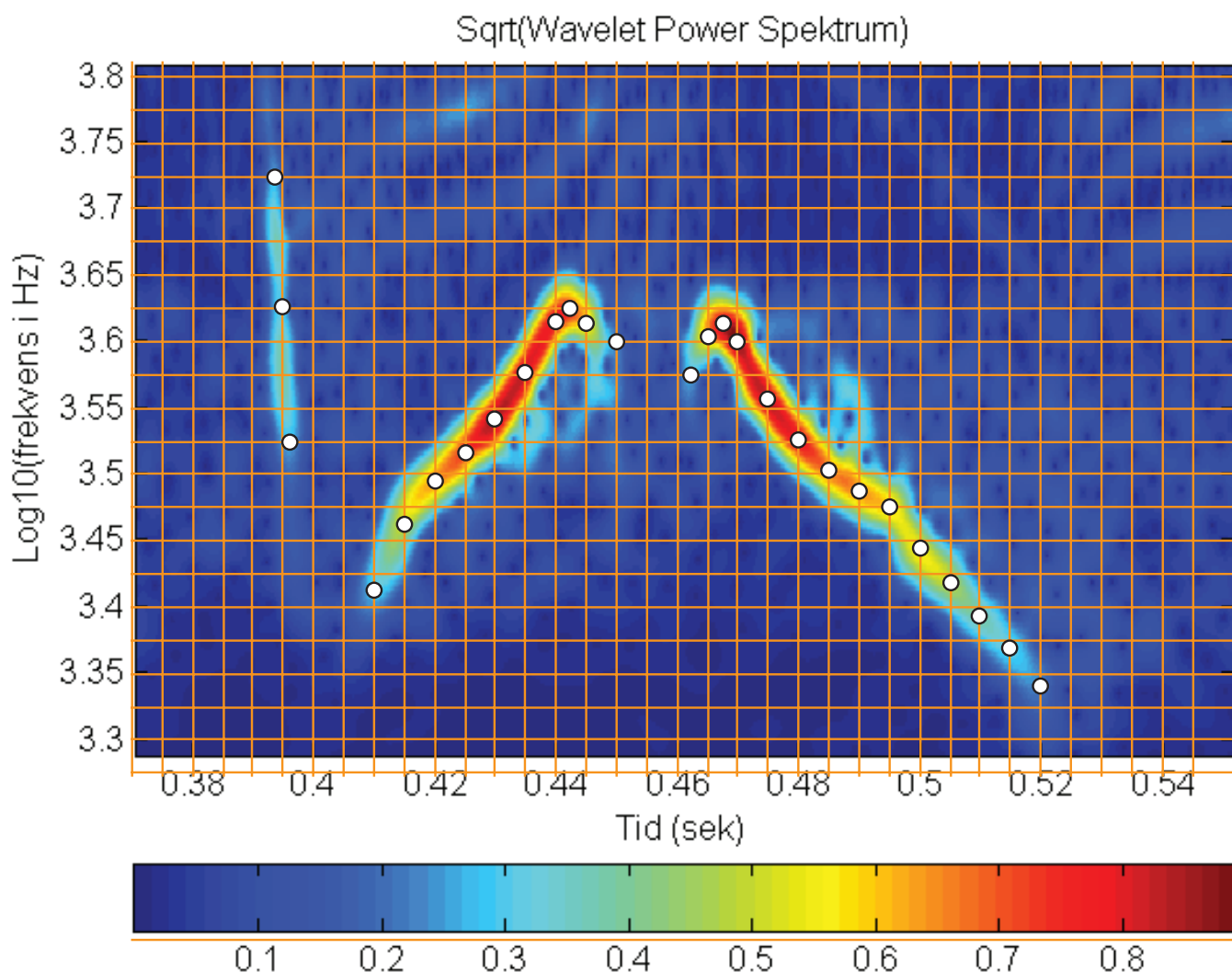
For syntesen av lyd må vi kjenne “momentan frekvens” og “momentan amplitude” for alle tider syntesen skal strekke seg over. For lyd samlet med 44100 Hz, har lydfilen 44100 verdier hvert sekund. Vi ønsker å klare oss med langt mindre, ja for syntesen av den lyden vi bruker i dette eksemplet, klarer vi oss med 90 reelle tall, og kunne nok uten særlig tap av kvalitet klart oss med enda litt færre tall.

Poenget er at vi tilnærmer variasjonene ved å bruke interpolasjon mellom de punktene vi har valgt ut. Det enkleste er å bruke en lineær interpolasjon, og det er det vi har brukt i vårt eksempel.

Først vil jeg fortelle hvordan jeg opprinnelig gikk fram. Det er en langt enklere måte som duger i alle fall for Matlab-brukere. Den gjengir jeg til sist i dette skrevet. Til tross for at den opprinnelige prosedyren er langt mer tidkrevende enn den nye, tar jeg med den opprinnelige for at tankegangen skal komme fram. Tankegangen er nemlig den samme for den nye metoden som den gamle, bare at prosedyren går raskere.

1) Start med et utsnitt i waveletdiagrammet av lyden vi ønsker å syntetisere.

2) Skriv ut waveletdiagrammet og tegn inn et rutenett på utskriften. Alternativt kan vi legge over et gjennomsiktig rutenett i “papirformat” over utskriften. Enda et alternativt er å tegne inn et rutenett i et tegneprogram (slik det er gjort her, se figuren nedenfor). Har vi mulighet for å justere rutestørrelsen etter tallene langs aksene, letter det avlesningen mye.



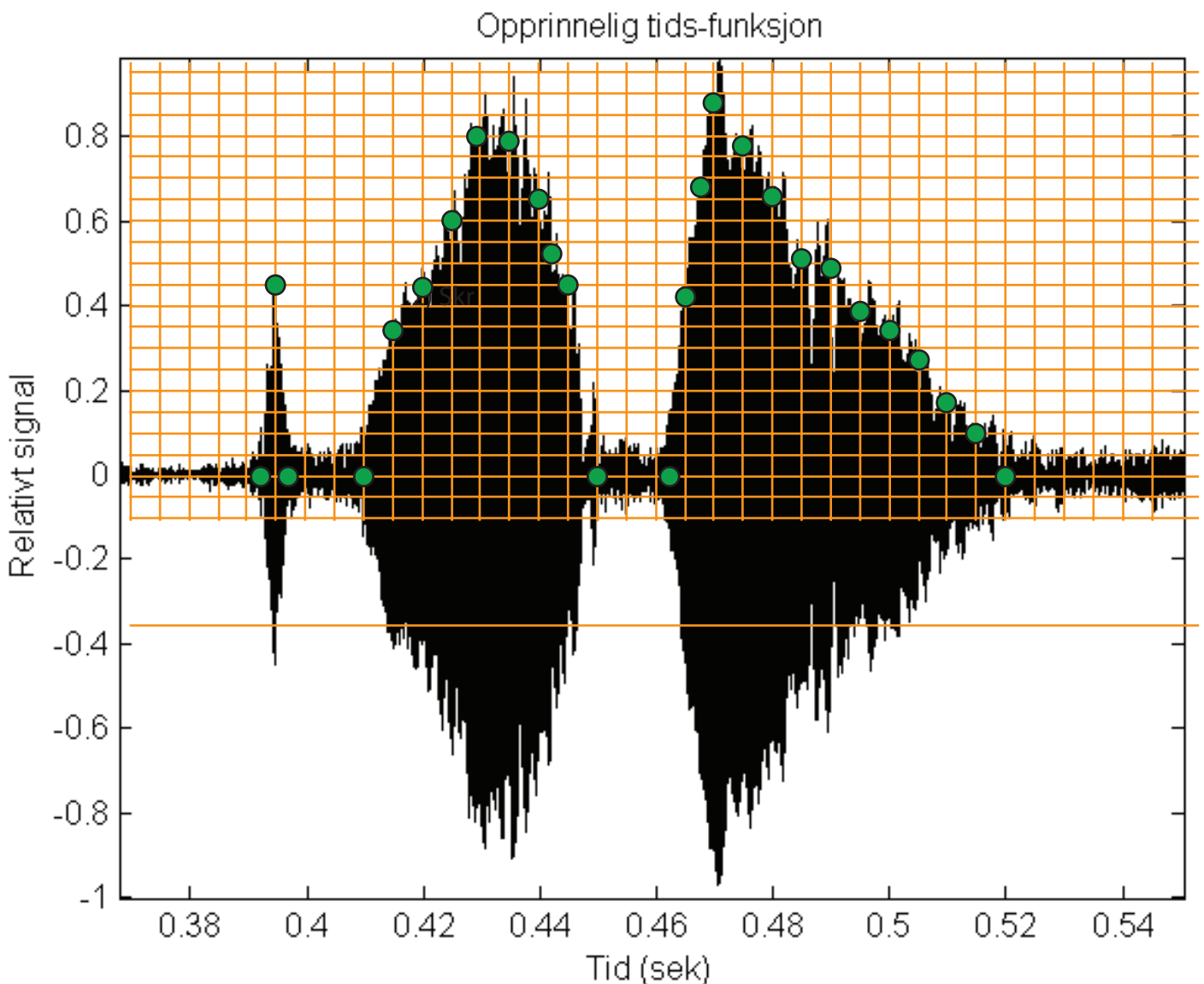
3) Marker punkter i diagrammet som lett lar seg avlese (se figur forrige side). Jeg valgte å stort sett bruke punkter som lå på de vertikale linjene i rutenettet slik at avlesningen av tidene for disse punktene gikk meget raskt. I partier hvor frekvensen skifter mye innen kort tid, eller at amplituden på lyden endrer seg mye innen kort tid, valgte jeg å bruke noen få punkter som ikke ligger på de vertikale strekene i tillegg til de andre.

4) Vi leser så av frekvensen (eller her \log_{10} av frekvensen) for hver av de valgte punktene. Resultatet er en todimensjonell matrise der den ene kolumnen angir valgte tidspunkt for angivelsene og den andre kolumnen angir frekvensen (eller \log_{10} av frekvensen).

Vi trenger også info om amplituder. Disse kan vi hente ut av waveletdiagrammet gjennom fargekodingen, men det blir ganske upresist. I vårt tilfelle var det (tilnærmet) ingen andre lyder ved siden av den vi ønsket å syntetisere. Da kan vi bruke et plot av lyden vs tid (amplitude vs tid) for å lese av amplituden med rimelig god nøyaktighet. I så fall blir prosedyren videre slik:

5) Tegn inn et rutenett i amplitude vs tid utskriften slik at de vertikale linjene ligger på samme sted i dette diagrammet som for waveletdiagrammet (se figuren nederst denne siden).

6) Les av relative amplituder ved de samme tidspunktene som ble valgt ut for beskrivelsen i punkt 3. Vi ender da opp med en tredimensjonal matrise hvor den tredje kolonnen nå blir relative amplituder. I figuren nedenfor er de valgte amplitudene angitt med symbol. Det er absolutt rom for forbedringer, men selv når vi valgte omtrent som vist i figuren, fikk vi slett ikke verst syntetisk lyd ut!



7) Når matrisen skal lages er det en siste vesentlig detalj som krever litt oppmerksomhet. I tiden før lyden for alvor starter (i perioden der det bare er litt svak bakgrunnstøy) må vi bruke interpolasjonsrutiner for å beregne frekvens og amplitude. Det samme gjelder i tiden etter at lyden har dødd ut. Dette krever at vi setter inn et “datapunkt” i vår matrise både før og etter de dataene vi allerede har fått. Tiden for disse punktene må ligge utenfor den tiden vi kommer til å bruke i interpolasjonen når lyden skal syntetiseres. Amplituden må være null, mens frekvensen ikke er kritisk. Jeg har valgt en verdi for frekvensen lik nabo-punktet. De to ekstra punktene (radene) er markert med gult i tabellen nedenfor.

Her er da matrisen som jeg fikk ut fra denne prosedyren da jeg laget den aktuelle bokfink-detalj som syntetisk lyd:

0.360	3.725	0.00
0.393	3.725	0.00
0.395	3.625	0.45
0.397	3.525	0.00
0.410	3.413	0.00
0.415	3.462	0.33
0.420	3.490	0.45
0.425	3.518	0.60
0.430	3.537	0.82
0.435	3.575	0.80
0.440	3.618	0.65
0.442	3.625	0.57
0.445	3.617	0.45
0.450	3.600	0.00
0.460	3.575	0.00
0.462	3.575	0.15
0.465	3.605	0.47
0.468	3.612	0.65
0.470	3.600	0.95
0.475	3.560	0.77
0.480	3.527	0.67
0.485	3.504	0.54
0.490	3.482	0.50
0.495	3.475	0.40
0.500	3.442	0.35
0.505	3.420	0.27
0.510	3.395	0.18
0.515	3.371	0.11
0.520	3.341	0.00
0.999	3.341	0.00

Se så på neste side hvor elegant disse dataene kan hentes inn når vi først kjenner tankegangen og utnytter Matlabs fasiliteter!

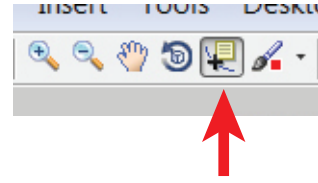
Bruk gjerne denne raske metoden for å syntetisere den lyden som opprinnelig er gitt som “lydForSynthese.wav” i deloppgave 5 i prosjektoppgaven i FYS2130 våren 2012. Ta gjerne tiden du trenger for å lage en datafil a la den gjengitt i tabellen ovenfor, og gi tiden i rapporten. Jeg vil tro dette kan gjøres ganske raskt!

Lykke til!!!

Raskere metode for å ekstrahere data

I Matlab kan vi lese av verdier i et plot ved å bruke:

Denne funksjonaliteten fungerer også bra når vi plotter ved hjelp av `imagesc()`-funksjonen, slik vi bruker i waveletdiagrammene våre.



Det betyr at vi rett og slett kan klikke oss gjennom et waveletdiagram og notere tider (X), frekvenser eller \log_{10} av frekvenser (Y) og waveletamplituder (Index) så enkelt som bare det. Pass på å notere hvorvidt du har valgt å bruke kvadratrotten av den waveltransformerte, eller den transformerte eller kvadratet av den transformerte.

Du kan så bruke et program liknende `interpolf7.m` (gitt separat i prosjektoppgaveopplegget) for å justere hvilket tidsrom du vil la den syntetiske lyden gå over, og du kan regne om fra \log_{10} frekvenser til frekvenser i en lineær skala, og du kan finne fram til amplituder ut fra "index"-verdier lest ut av diagrammet.

Du bør nok bruke en relativt lav K-verdi i waveletanalysen dersom "index"-verdier skal gi et godt mål for amplituder ved ulike tider. Jeg har ikke selv testet ut dette i detalj, så du får være litt obs. Husk at amplitudeverdiene i programmet må normeres slik at den høyeste verdien aldri er større enn 1 (absoluttverdi) før lyden kan spilles av f.eks. vha funksjonen `playblocking()` i Matlab.

