Prosjektoppgave FYS2130

Kandidatnummer: 10

Jeg har samarbeidet med kandidatnummer 68 gjennom hele arbeidet. Med mindre han/hun har kjørt noen av programmene på nytt for å endre på noe i løpet av helgen er alle figurer og programmer identiske med mine. Selve rapporten er kun skrevet av meg.

Innhold

1 Del 1

1.1 Beskrivelse

Den første delen av prosjektet går ut på å lage og teste en del programmer vi får bruk for senere.
Det er hovedsakelig tre programmer som skal lages:

HvitStoyGauss Dette programmet genererer en gausisk frekvens fordeling rundt et senter vi velger med en bredde vi også velger. Vi må også definerer hvor mange punkter denne lages for og hvor stor samplings frekvensen skal være. Vi kjører så en invers fourier transformasjon på disse frekvensene og får ut et singal.

WL1 Genererer en wavelet analyse mellom to frekvenser vi velger med K-parameter K. Dette programmet er en delvis redigert kopi fra kapittel 13 i læreboka.

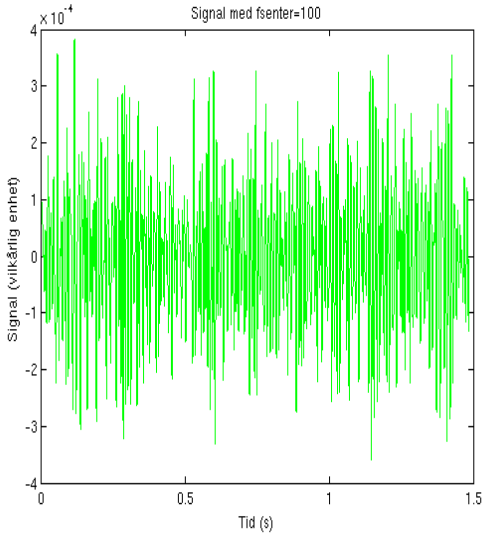
AutoCorr Beregner korrelasjonen mellom signalet på to forskjellige tidspunkter og plotter dette. Legger også på en linje for der korrelasjonen har falt til halvparten av startpunktet.

Alle disse programmene ligger vedlagt. Når disse programmene er laget skal vi så teste de for noen gitte parametre og studere resultatene.

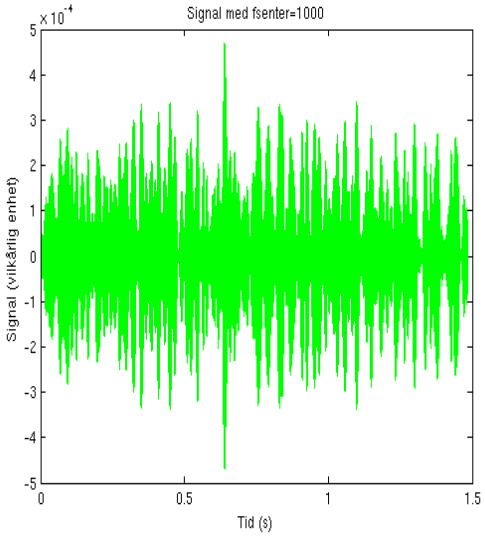
Vi velger å bruke følgende parametre:
Fullverdibredde for frekvensfordeling i signalet på 50 Hz med en senterfrekvens på 100, 1000 og 10000 Hz. Setter K-verdien til 48 i Wavelet analysen, og justerer frekvensområdet etter senterfrekvensen for hvert tilfelle.

For signalene forventer vi å få et signal som helt enkelt og greit består av støy. Denne støyen bør variere mer etterhvert som vi øker senterfrekvensen for signalet.

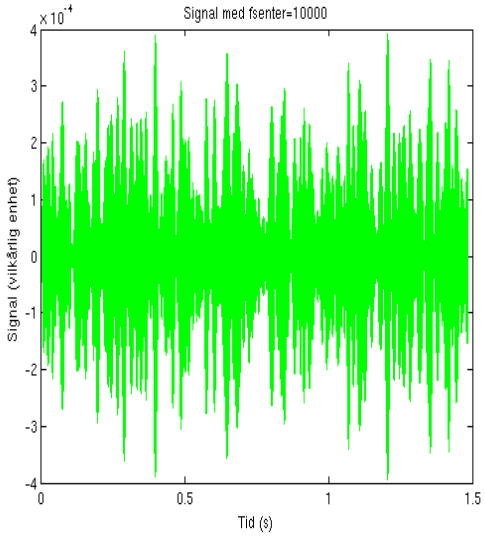
1.2 Signalene



(a) Vi ser at vi har fått et signal med støy.



(b) Ser her at signalet varierer raskere.

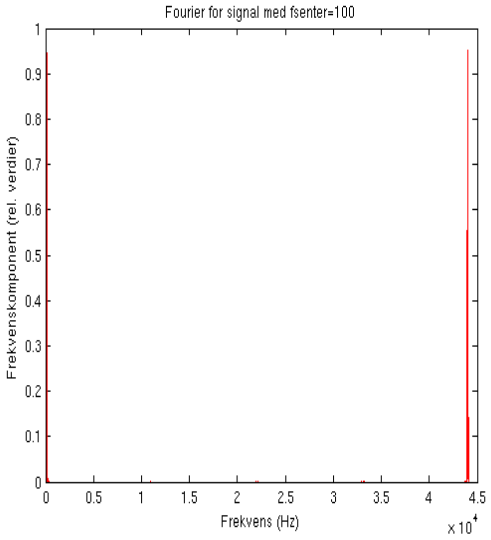


(c) Ser her at signalet varierer enda raskere.

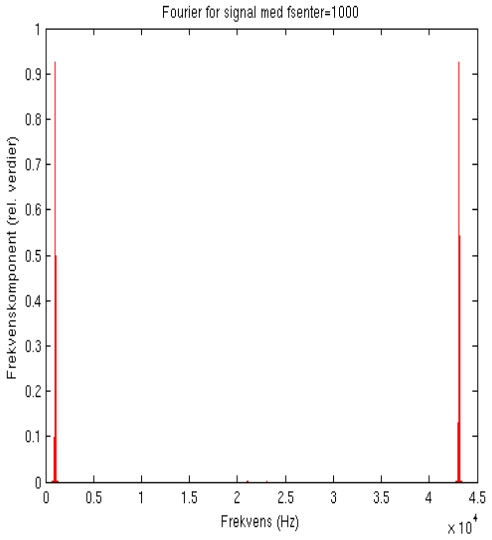
Figur 1: Signalet med forskjellige senterfrekvenser.

Vi får ut signaler som vi forventet å få. Programmet vårt har altså fungert som det skal.

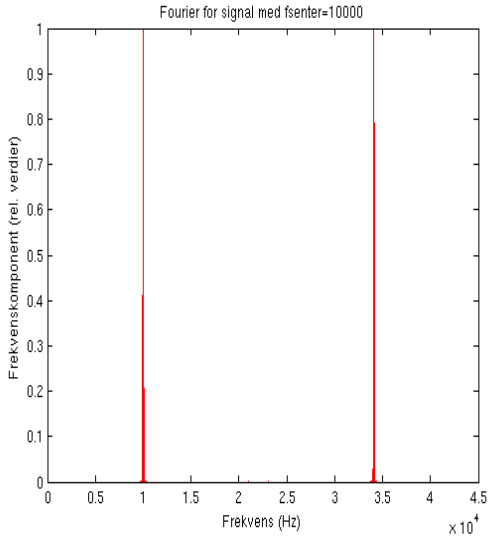
1.3 Frekvens spekterne



(a) En topp på 100 Hz og en topp på ca 44kHz.



(b) En topp på 1kHz og en topp litt under 44kHz.



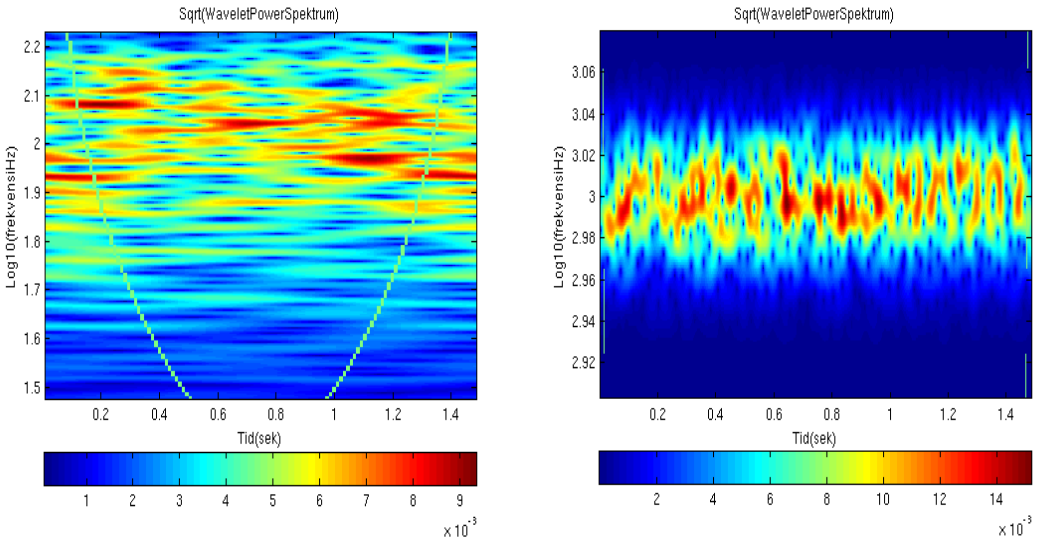
(c) En topp på 10kHz og en litt under 35kHz

Figur 2: Frekvensspekter for hvert av signalene.

Vi ser at frekvensene er der det skal være, men at vi har fått med en topp på en frekvens mye høyere enn den vi satt som senter!

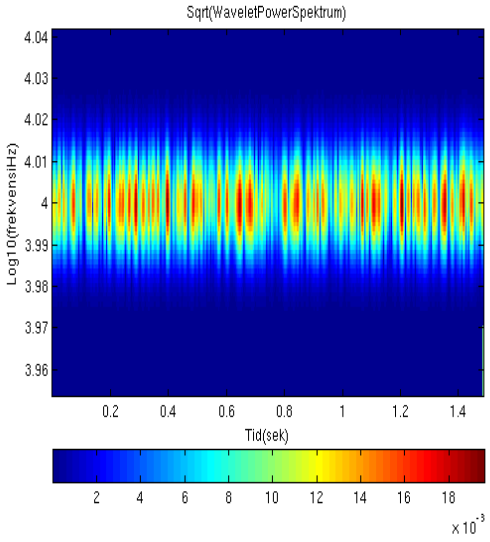
Dette skyldes måten Fourier transformasjon fungerer på, og er faktisk som forventet. Vi minner om at vi satt samplingsfrekvensen til å være 44100Hz, og at det da er ganske åpenbart at samlingsfrekvens minus senterfrekvens gir oss plasseringen til den første ekstratoppen. Regnestykket kan gjentas for å se sammenhengen for de to andre, om ønskelig.

1.4 Waveletdiagrammene



(a) Støy med varierende amplitude.

(b) Enda klarere.

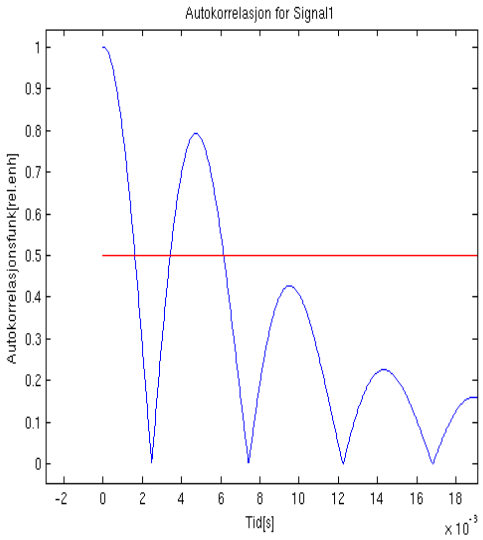


(c) Her får vi områder med nesten ingenting også!

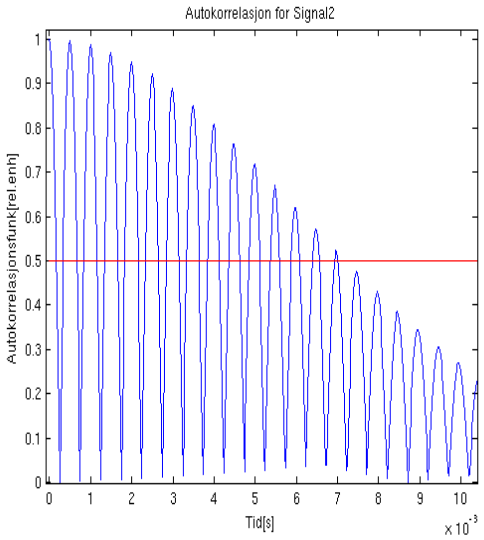
Figur 3: Waveletdiagrammer for hvert av signalene.

I det første diagrammet ser ikke ting så veldig spennende ut. Vi ser at frekvenser rundt senterfrekvensen er aktive som forventet og at amplituden varierer.
I det andre diagrammet gjentar dette seg rundt senterfrekvensen for dette signalet. Vi ser også at det begynner å komme frem områder der signalet nesten forsvinner i noen øyeblikk.
I det tredje diagrammet kommer det klart frem at signalet forsvinner og kommer tilbake igjen. Det kan virke som at signalet interferer konstruktivt og destruktivt med seg selv over tid.

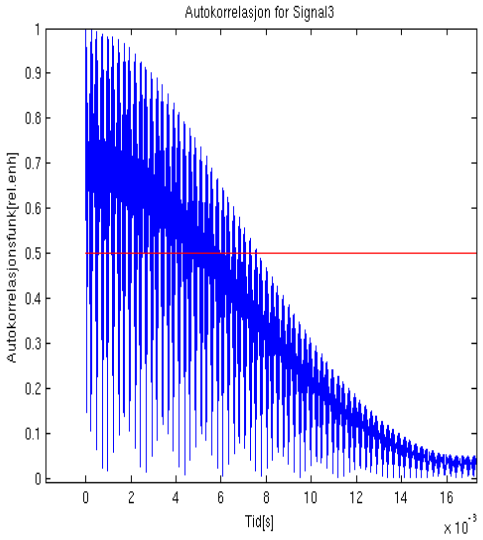
1.5 Autokorrelasjon



(a) Autokorrelasjon mefd fsenter = 100.



(b) Autokorrelasjon mefd fsenter = 1000



(c) Autokorrelasjon mefd fsenter = 10000

Figur 4: Autokorrelasjon for hver av signalene.

Avlesing av autokorrelasjon gir oss følgende målinger:

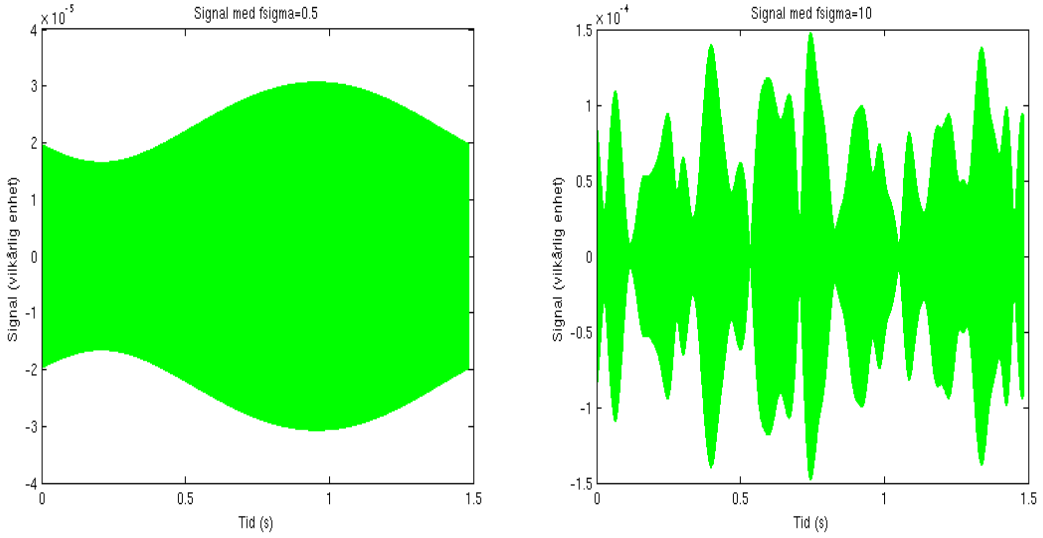
signal	fsenter	koherenstid
1	100	8.00 ms
2	1000	8.75 ms
3	10000	7.80 ms

Vi observerer at koherenstiden forblir omtrent den samme selv om vi endrer senterfrekvensen. Dette kan virke logisk hvis vi tenker på at selv om senterfrekvensen endrer seg så varierer signalet kun et lite område i frekvens rundt senterfrekvensen. Det er ikke da så merkelig at det å flytte rundt på signalet i spekteret ikke forandrer koherendstiden.
Så koherenstiden er altså uavhengig av senterfrekvensen for signalet, men hva med fullverdibredden? Hvis vi endrer på *den* må vel koherenstiden bli påvirket?

1.6 Beskrivelse

Vi skal nå sette senterfrekvensen fast på 10 kHz, og se hva som skjer med fullverdibredd på 0.5, 10, 100, 1000 og 5000 Hz. Dette er helt enkelt og greit. Vi setter inn parameterne i programmene vi allerede har laget og får ut resultatene på de neste sidene. jeg har valgt å ikke ta med resultatene for 5000Hz siden de ikke gir oss noen ny informasjon og dermed bare fyller opp sidene.

1.7 Signalene



(a) Liten variasjon

(b) Litt mer

(c) Bedre

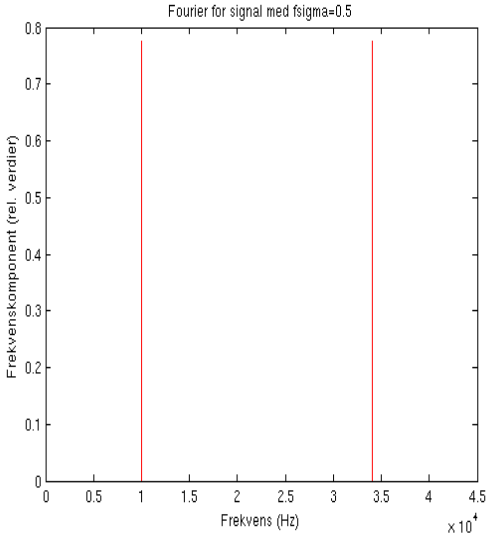
(d) Enda bedre

Figur 5: Signalet med forskjellige fullverdibredde.

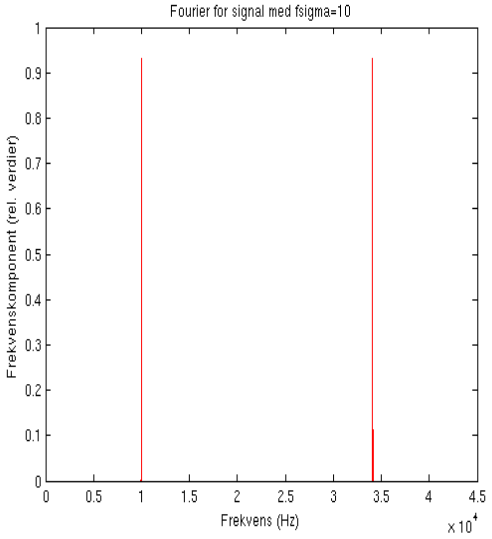
Vi ser at jo bredere vi sette fullverdibredden, jo større variasjon i signalet får vi. Dette er egentlig ganske åpenbart siden vi ved å øke bredden gir signalet flere

frekvenser å jobbe med, og dermed kan signalet variere på flere måter.

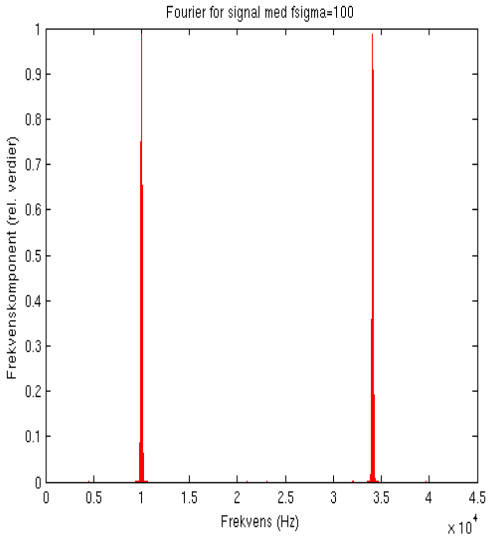
1.8 Frekvens spekterne



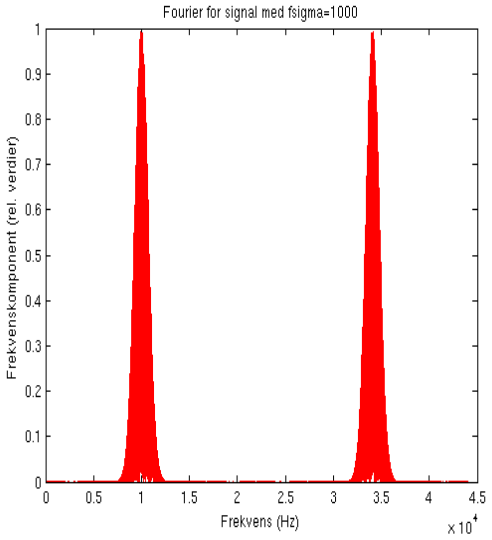
(a) Skarpe pigger .



(b) Fortsatt skarpe



(c) Litt bredere

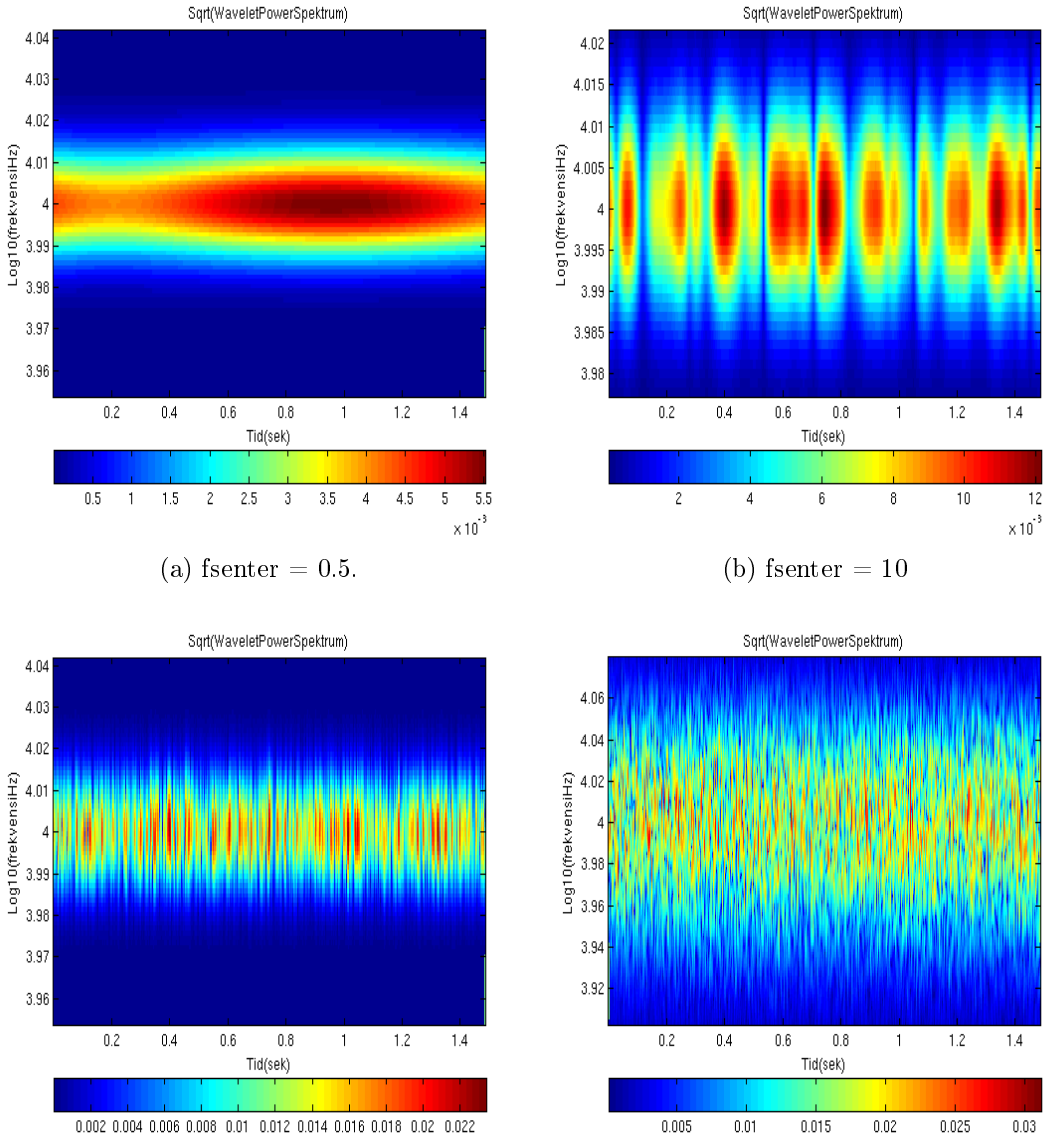


(d) Bredere

Figur 6: Frekvensspekter for hvert av signalene.

Vi ser at frekvensene oppfører som vi har satt de til å gjøre. Vi har to topper, akkurat som før, og de er der de skal være. Vi ser også at når vi endrer fullverdibredden, så blir faktisk piggene dratt utover også, akkurat som de skal.

1.9 Waveletdiagrammene



(a) fsenter = 0.5.

(b) fsenter = 10

(c) fsenter = 100

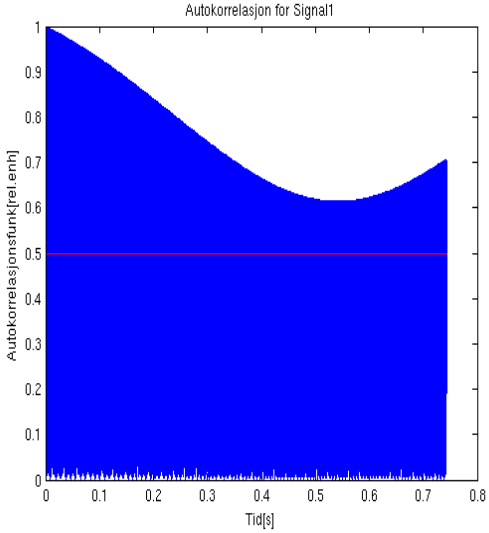
(d) fsenter = 1000

Figur 7: Waveletdiagrammer for hvert av signalene.

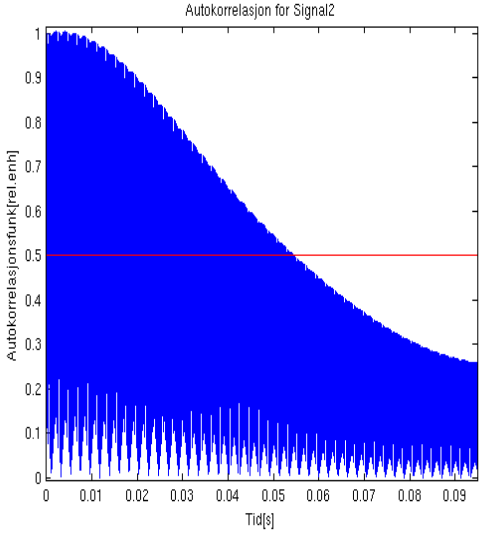
Vi ser at waveletdiagrammene har signaler sentrert rundt riktig frekvens. Vi ser også at fullverdibredden bestemmer bredden i frekvensspekteret som den skal. Vi

ser også at ved å øke bredden så øker variasjonen i signalet også.
I det første diagrammet ser vi liten variasjon. Vi har gitt signalet en liten bredde å jobbe med her, så dette er som forventet.
I det andre diagrammet ser vi et nydelig eksempel på at signalet interfererer med seg selv. Vi får sterke områder og striper der signalet slukker seg selv.
I det tredje diagrammet ser vi fortsatt en antydning til fenomenet, men variasjonen i signalet begynner å bli så stor at det er vanskelig å se det.
I det fjerde diagrammer har variasjonen tatt helt over, og vi kan ikke se noe til at vi får hverken sterke eller svake områder.

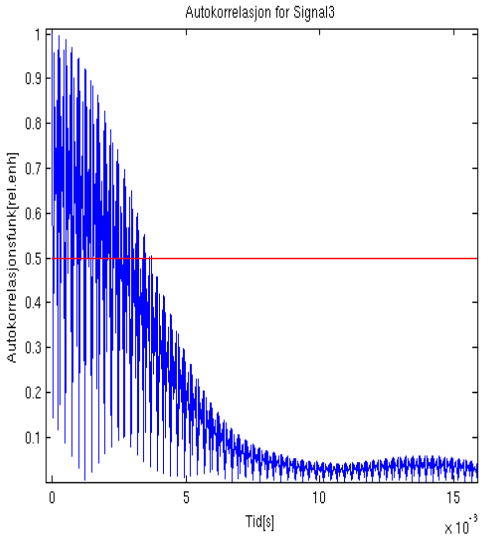
1.10 Autokorrelasjon



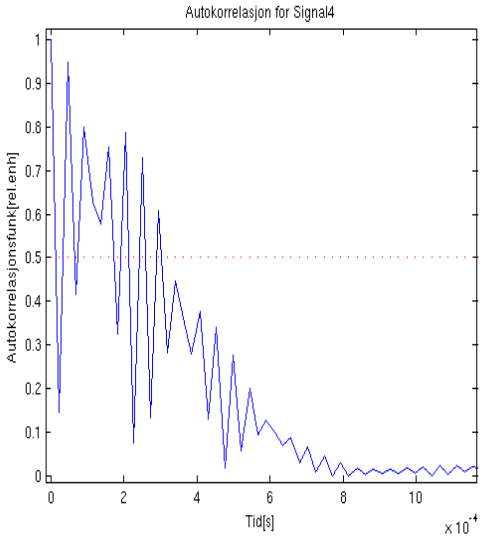
(a) Liten variasjon.



(b) Litt mer.



(c) Enda mer



(d) Stor variasjon

Figur 8: Autokorrelasjon for hver av signalene.

Avlesing av autokorrelasjon gir oss følgende målinger:

signal	fsigma	koherenstid
1	0.5	ikke nådd
2	10	0.05 s
3	100	3.70 ms
4	1000	0.35 ms

Vi ser at det er en klar sammenheng mellom koherenstid og fullverdibredde. Vi så i waveletdiagrammet for det første signalet at det var liten endring i signalet over tid og det første diagrammet her bekrefter dette. Det er faktisk så liten endring at koherenstiden ikke blir nådd!
I det andre diagrammet ser vi at vi når koherenstiden etter kort tid, og det to neste diagrammene viser at tiden blir kortere for høyere fullverdibredde.
En rask titt på resultatene i skjemaet kan tyde på at en tidobling i fullverdibredde gir oss en koherenstid på en tidel av det vi hadde.

1.11 Konklusjon

Jeg har valgt å sette koherenstiden lik halvparten av maksimum. På dette tidspunktet i oppgaven var dette bare et tilfeldig valg jeg gjorde. Vi kommer imidlertid i neste del til å se på krysskorrelasjon, og jeg ville ha samme linje for koherenstid i autokorrelasjon som i krysskorrelasjon. Det at jeg ville ha samme linje, gjorde at når jeg så på resultatene i neste oppgave, ikke ville gå lavere enn halvparten av maksimum, siden resultatet vi fikk for en verdi nær grensen, begynte å bli såpass dårlig at jeg ikke var komfortabel med å sette den ned på 1/e av maksimum. Jeg vil heller gå glipp av noen verdier enn å ta med verdier som ikke gir god korrelasjon når jeg tror at de gjør det fordi de er over linjen.

Vi har i del 1 sett at koherenstid er avhengig av fullverdibredden på signalet vi analyserer.
Vi har sett at vi ved liten fullverdibredde får liten variasjon i signalet, og at stor bredde gir stor variasjon.
Vi så også at vi ved å velge en mellomting kan få signalet til å interferere med seg selv, og at det på noen tidspunkter kan interferere nok til at det nesten blir borte.

2 Del 2

2.1 Beskrivelse

Vi skal først generere to ulike signaler, k1 og k2, ved hjelp av programmet vi laget i del 1. Disse signalene skal ha senterfrekvens 10kHz og fullverdibredd 1kHz. Vi skal så simulerer at vi har to detektorer som tar i mot disse signalene. For detektor A skal signalene komme frem på samme tidspunkt. For detektor B skal signalet

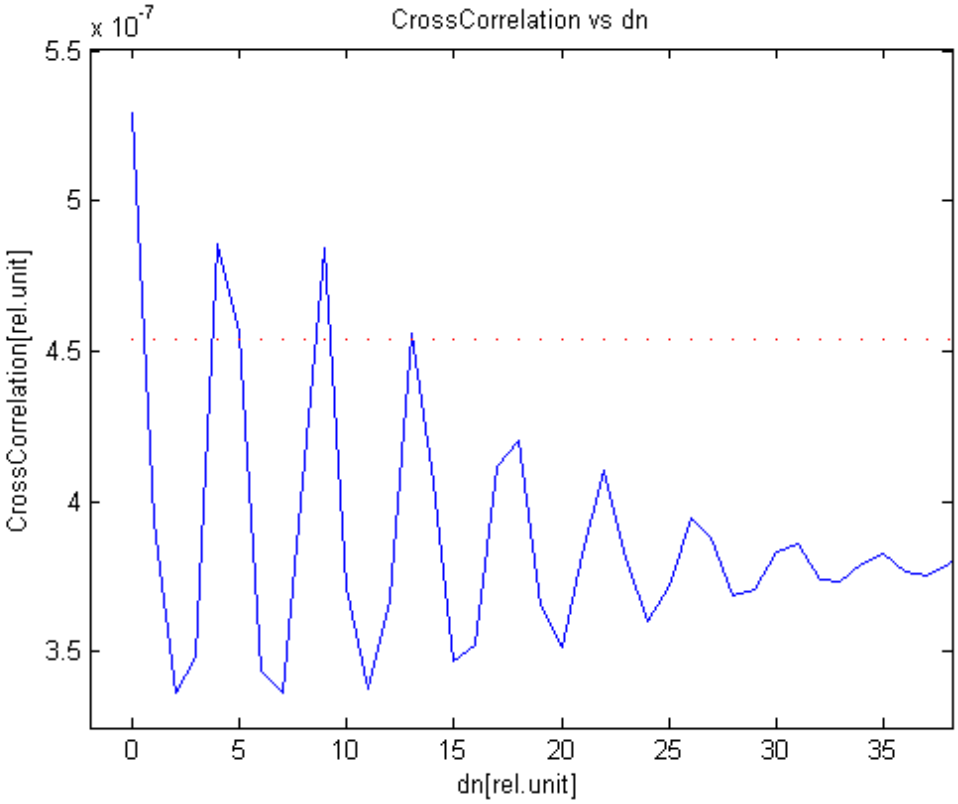
fra kilde 1 komme frem likt som på A, men signalet fra kilde 2 skal komme frem en tid senere. Vi modellerer dette ved å bruke et signal på et senere tidspunkt fra kilde 2, når vi summer signalene, når de kommer frem til B. Vi skal så plotte krysskorrelasjon som funksjon av forflyttningen

Først starter vi med å lage programmene vi trenger:

LowPass Dette programmet kjører en fourier transformasjon av et signal. Man setter så et maksimum for frekvensen. Programmet beregner så hvor midten av spekteret er og fjerner alt fra midten ned til maksimumet vi satt og opp til midten + maksimumet. Det blir så gjort en invers fourier transformasjon og vi får ut signalet igjen.

CrossCorr Dette programmet sjekker hvor like to signaler er på et tidspunkt. Ligner veldig på autokorrelasjon.

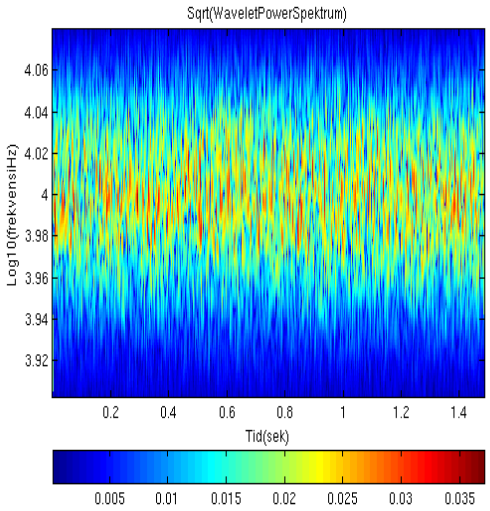
2.2 Plot av krysskorrelasjonen mellom A og B



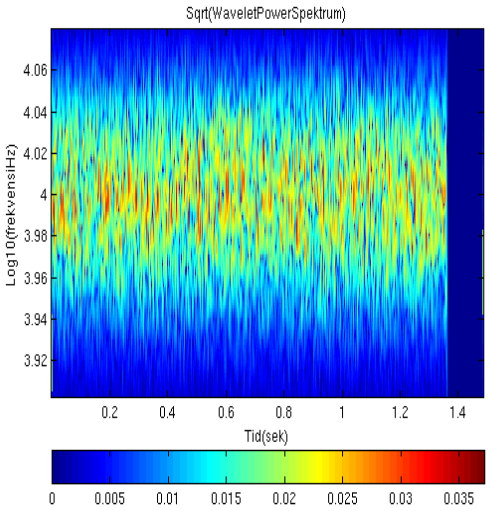
Figur 9: Vi ser har at det er noen topper med god korrelasjon

Det at vi ser på plottet at krysskorrelasjonen er god ved noen tidspunkter gjør at det er fristende å prøve å flytte mottaker B. Vi testet først med å sette plasseringen til A og B like, og vi fikk da like signaler på begge to. Dette plottet er ikke tatt med siden det er liten vits med to like plot i en rapport. Vi velger å flytte den tre forskjellige avstander. To av disse er de toppene vi så. Den siste er en dårlig en for å se om vi merker noen forskjell. Hvis vi så studerer waveletdiagrammene for hvert tilfelle bør vi se omtrent det samme signalet for de to gode avstandene og noe helt annerledes for den dårlige.

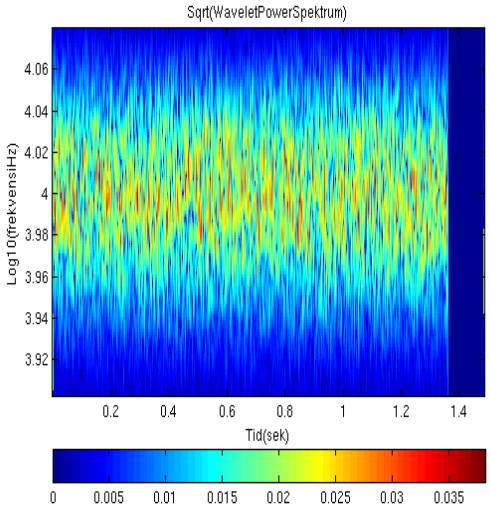
2.3 Waveletdiagrammene



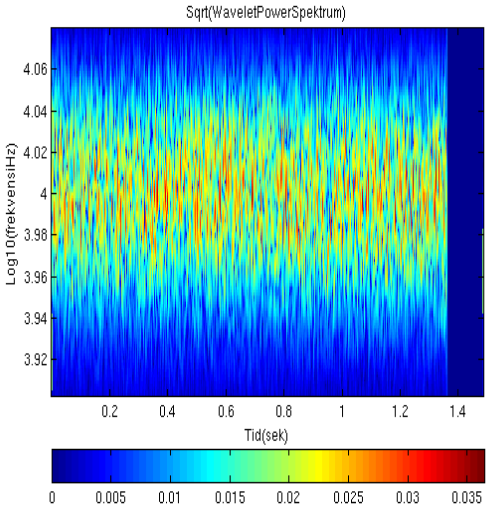
(a) Signalet mottatt hos detektor A.



(b) Signalet mottatt hos detektor B, dn = 4.



(c) Signalet mottatt hos detektor B, dn = 9.



(d) Signalet mottatt hos detektor B, dn = 82.

Figur 10: Waveletdiagrammer for hvert av signalene.

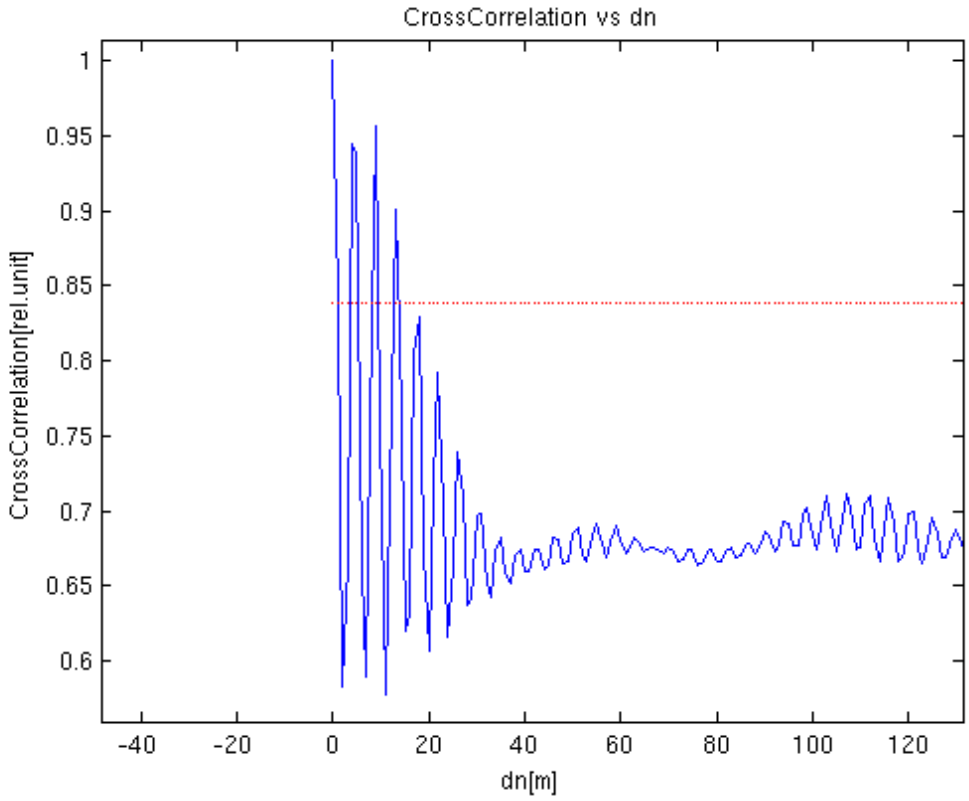
Hvis vi nå ser på det første signalet og merker oss plasseringen til noen røde områder og så ser etter de samme områdene på de neste diagrammene vil vi se

noe interessant. Vi ser at når vi flytter B at diagrammene ser ganske like ut for dn = 4 og 9. Når vi så flytter B den dårlige forflytningen, får vi noe helt annerledes. Forventningene våre blir altså bekreftet.

2.4 Beskrivelse

Vi får så beskjed om å gjøre akkurat det samme en gang til, men denne gangen skal vi kjøre signalene gjennom et lavpassfilter som kun slipper gjennom signaler under 1kHz. Vi minner om at signalet vi nå starter med har senterfrekvens 10kHz og fullverdibredde 5kHz. Det er altså et veldig svakt signal vi nå jobber med videre. Når vi gjør det, og så plotter krysskorrelasjonen, får vi ut følgende plot.

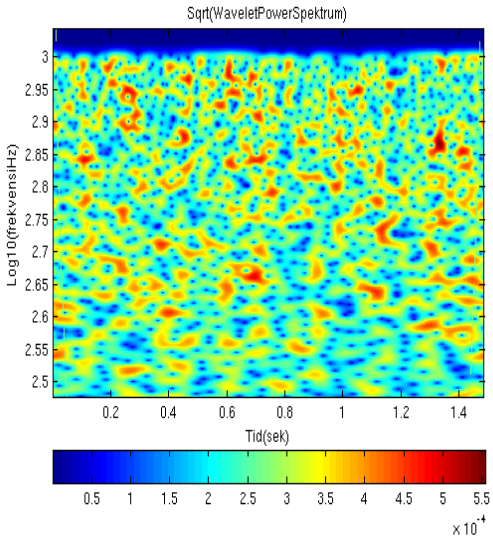
2.5 Krysskorrelasjon



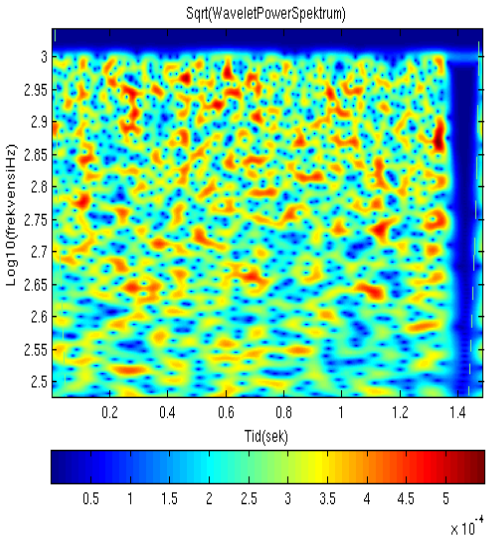
Figur 11: Plot av krysskorrelasjonen mellom A og B

Vi observerer her det samme som tidligere. Velger samme forflytninger denne gangen siden vi ser at toppene befinner seg på samme sted. Velger en ny dårlig forflytning tilfeldig et sted vi ser på plottet at er dårlig. Forventer å få ut noe som ligner på det vi hadde uten lavpass filter.

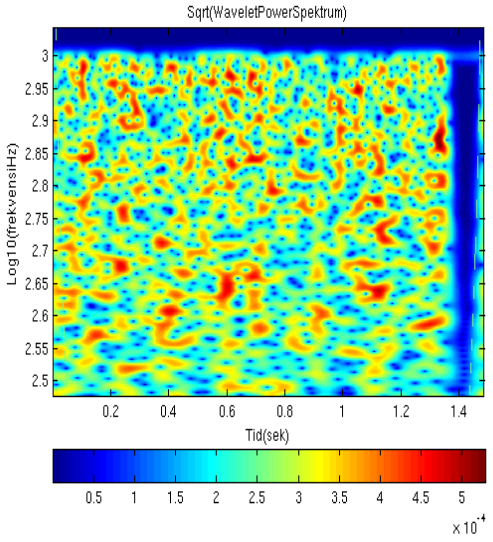
2.6 Waveletdiagrammene



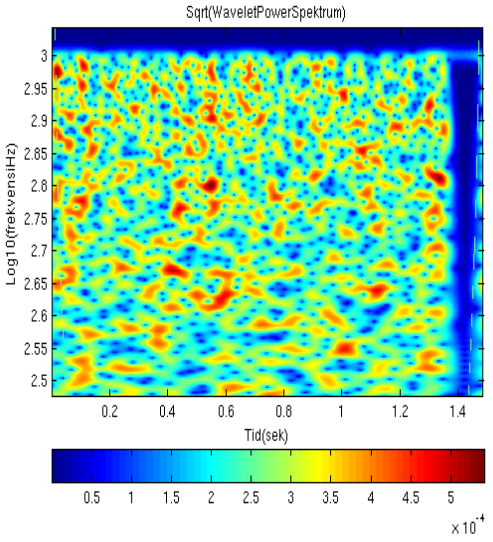
(a) Signalet mottatt hos detektor A.



(b) Signalet mottatt hos detektor B, dn = 4.



(c) Signalet mottatt hos detektor B, dn = 9.



(d) Signalet mottatt hos detektor B, dn = 26.

Figur 12: Waveletdiagrammer for hvert av signalene.

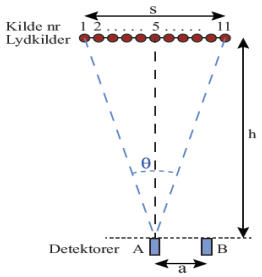
2.7 Konklusjon

Her fikk vi oss en liten overaskelse. Istedenfor å få kaotisk støy, som vi kunne se at lignet på hverandre til en viss grad, får vi nå ut et signal som er helt klart likt for de gode forflytningene, og helt annerledes for den dårlige! Hvis vi studerer hvert av plottene ser vi også at de er fulle av noe jeg bare kan kalle huller. Det virker som det er en hel haug med små hull i signalene der frekvenser blir borte. Siden det er forskjellig størrelse på disse, så antar jeg at dette ikke er en numerisk feil i matlab, men at det faktisk er slik. Om dette betyr noe eller ikke vet jeg ikke noe om. Det kan også nevnes at det er en antydning til striper der signalet interfererer med seg selv destruktivt her også, som tidligere.

3 Del 3

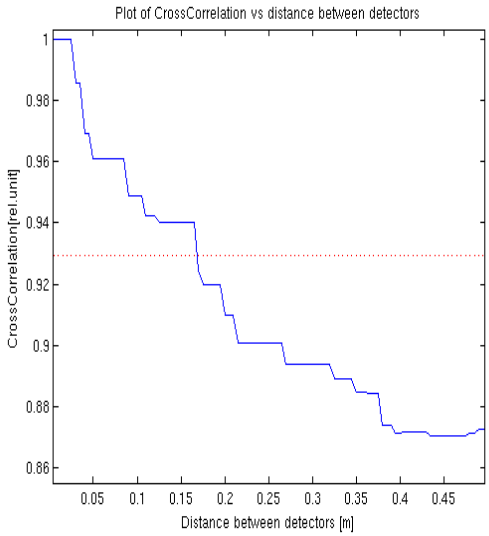
3.1 Beskrivelse

Her skal vi gjøre akkurat det samme som i del 2, men vi skal øke antallet kilder til 11! Vi velger å beregne intensiteten for innsignalene, og bruker lavpass filter på de mottatte signalene. Vi beregner så krysskorrelasjon for signalene. Vi skal gjøre dette for tre forskjellige avstander mellom kilder og detektorer. Vi velger å bruke avstandene h = 100, 200 og 400 m, og s = 10m. I tillegg skal vi for hver av disse avstandene endre avstanden mellom detektor A og B. Vi velger å sjekke avstandene mellom 0 og 5 m.
For å få gjort dette må vi lage et program som regner ut avstandene mellom A,B og alle de 11 kildene. I tillegg må vi beregne forskyvningen for alle signalene inn til B. For å beregne avstandene laget vi programmet ved navn Avstander. Programmet ligger vedlagt.
For å forstå oppsettet legger vi ved en kopi av figuren gitt i oppgaveteksten til prosjektet.

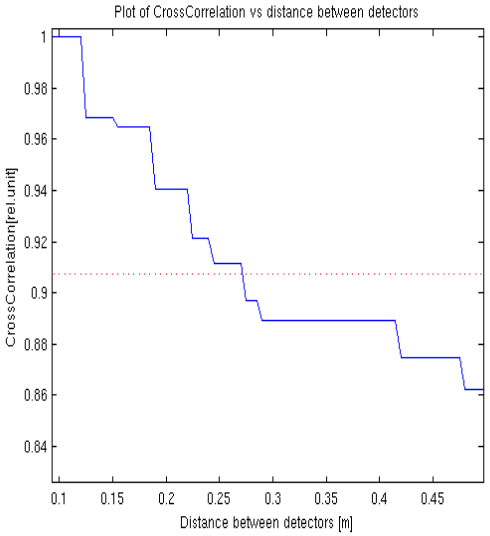


Figur 13: Tegning av oppsettet

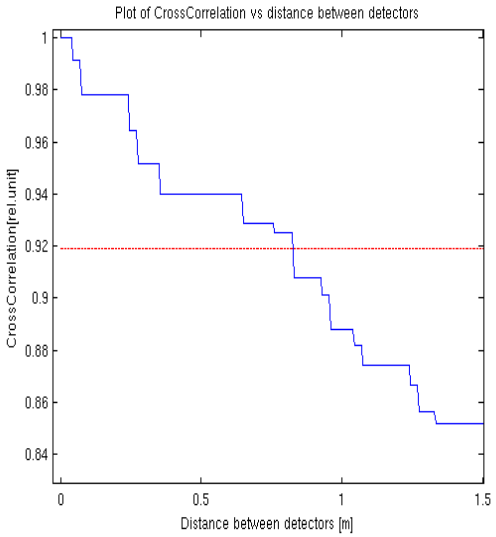
3.2 Krysskorrelasjon



(a) Krysskorrelasjon med h = 100.



(b) Krysskorrelasjon med h = 200.



(c) Krysskorrelasjon med h = 400.

Figur 14: Signalet med forskjellige senterfrekvenser

3.3 Konklusjon

Vi ser her at vi har et helt annet resultat enn for bare to kilder. Nå får vi en slags trapp med god korrelasjon på korte avstander! Vi kan altså flytte rundt på detektor B, og så lenge vi holder oss innenfor en viss avstand får vi god korrelasjon! Vi observerer også at denne avstanden øker hvis vi øker avstanden fra kildene. Dette gir mening med tanke på at vi i grensetilfellet, at denne avstanden blir stor, ikke bør kunne skille kildene fra hverandre. Det kan virke som at en dobling av avstanden gir dobbelt så stor koherenstid, men jeg ville gjort flere målinger før jeg sa noe sikkert angående dette.
Vi får så beskjed om at vi skal beregne størrelsen $\eta = \frac{\theta a}{\lambda}$ for hvert av tilfellene med a der koherensen har sunket til halvparten av maks.
Avlesning av plottene gir oss følgende:

h	a
100	0.2m
200	0.4m
400	0.8m

Dette viser oss at det er en klar sammenheng mellom avstanden h og avstanden vi kan flytte mottakerne fra hverandre. Dette bekrefter det vi allerede har funnet, altså at jo lenger unna kildene er jo vanskeligere blir det å skille signalene fra hverandre, og at tolleransen i avstand mellom A og B da øker.
Vi antar at vi kan si at $\lambda = v/f$, der v er lydhastigheten i luft og f er senterfrekvensen for signalene. Litt geometri forteller oss at $\theta = 2arctan(s/2h)$

h	a	θ	λ	η
100	0.2	0.1	0.05m	0.4
200	0.4	0.05	0.05m	0.4
400	0.8	0.025	0.05m	0.4

Vi får så beskjed om å vise at størrelsen η har et lignende opphav som $dsin(\theta) = \lambda$. I vårt tilfelle har vi at $d = a$ slik at vi skal vise at $asin(\theta) = \lambda$ har en sammenheng med $\lambda\eta = a\theta$. Vi vet at for små θ så er $sin(\theta) = \theta$ slik at $asin(\theta) = a\theta$. Problemet med denne fremgangsmetoden er at jeg ender opp med

$$\frac{a\theta}{\lambda} = 1$$

Dette hadde vært helt greit hvis jeg hadde kommet frem til at $\eta = 1$, men det har jeg jo ikke så jeg kommer egentlig ingen vei med dette.
Vi så i plottet av krysskorrelasjon for forskjellige lengder h, at for små endringer a, var korrelasjonen like god. Den er ikke nødvendigvis perfekt, siden jeg har

normalisert funksjonen min. På ett eller annet vis må dette bety at for å endre korrelasjonen mellom de må vi flytte B mer enn en viss avstand. Vi ser også at avstanden h har en betydning for dette. Det virker logisk at det er en sammenheng mellom denne minimumsavstanden, avstanden h og bølgelengden til bølgen. Hva denne sammenhengen er vet jeg ikke sikkert.
Dette tyder på at den romlige koherensen også bør være god for avstander opp til denne minimumsvstanden.

4 Del 4

4.1 Sirius

Grafen får omtrent samme form som vår. Og hvis vi justerer i forhold til avstander og intensitet ser det ut som at modellen vår passer bra til deres resultater.

4.2 Silvas Powerpoint

Kort forklart handler kontroversen om at eksperimentet tilsynelatende viser at bølgeteorien til lys er bedre enn partikkelteorien. Det blir lagt vekt på at det ikke var noen kontrovers ang. radiobølger, men med en gang det var snakk om synlig lys begynte folk å tvile på at interferometeret deres ville virke. Det var også veldig rart for mange at bølgeteorien skulle fungerer på dette fenomenet som ble sett på som et partikkelproblem.

5 Del 5

5.1 HBT effekten

I denne prosjektoppgaven har vi modellert HBT effekten med bølger. Vi har laget en enkel numerisk modell. Vi startet med å generere et signal med støy. Dette signalet ble så sendt ut fra en kilde et stykke unna. Signalet ble så mottatt av to detektorer. Vi har så analysert signalet på mange forskjellige måter. Vi fant underveis at koherenstiden til signalet var uavhengig av senterfrekvensen, men at det var avhengig av fullverdibredden til signalet. Vi fant også at det gikk ann å flytte detektor B noen spesifikke avstander, og fortsatt få god korrelasjon med A. Når vi så sendte signalet gjennom et lavpass filter fikk vi se at dette stemte og at vi hadde en stor del "klumper" som var fristende å tolke som "lydpakker". Vi fant så at hvis vi økte antall kilder, slik at signalet ligner mer på et signal fra den virkelige verden, fikk vi en krysskorrelasjonsfunksjon som holdt seg på samme nivå over små avstander. Vi fant også at hvis vi økte avstanden til kildene, kunne vi øke avstanden mellom detektorene våre uten å påvirke korrelasjonen mellom det mottatte signalet i detektorene.

Det virker utifra det vi har gjort som at hvis vi overfører dette til elektromagnetiske bølger, bør kunne oppdage disse klumpene"i signalet vi mottar, og at det da blir fristende å kalle disse "lyspakkene" for fotoner.

Hver av observasjonene som nevnes i denne delen er forklart bedre i tidligere deler.

Avslutningsvis kan det nevnes at Del 4 og 5 ikke har fått like mye oppmerksomhet som de burde. Jeg kan egentlig ikke skylde på noe annet enn at jeg har prioritert den delen av oppgaven som omhandler programmering og tolking av resultatene.

6 Programmer

6.1 Del1-1

```
function Del1a
% Lager et tilfeldig signal som ønsket
Fs = 44100;
N = 1024*64;
T = N/Fs;
t = linspace(0,T*(N-1)/N, N);
f = linspace(0,Fs*(N-1)/N,N);
fsigma = 50;
```

```
xx1 = HvitStoyGauss(Fs,N,100,fsigma);
xx2 = HvitStoyGauss(Fs,N,1000,fsigma);
xx3 = HvitStoyGauss(Fs,N,10000,fsigma);

% Velger å plotte tids- og frekvensbildet her
figure;
plot(t,xx1,'-g');
xlabel('Tid (s)');
ylabel('Signal (vilkårlig enhet)');
title('Signal med fsenter=100')

figure;
plot(f,abs(fft(xx1)),'-r');
xlabel('Frekvens (Hz)');
ylabel('Frekvenskomponent (rel. verdier)');
title('Fourier for signal med fsenter=100')

figure;
plot(t,xx2,'-g');
xlabel('Tid (s)');
ylabel('Signal (vilkårlig enhet)');
title('Signal med fsenter=1000')

figure;
plot(f,abs(fft(xx2)),'-r');
xlabel('Frekvens (Hz)');
ylabel('Frekvenskomponent (rel. verdier)');
title('Fourier for signal med fsenter=1000')

figure;
plot(t,xx3,'-g');
xlabel('Tid (s)');
ylabel('Signal (vilkårlig enhet)');
title('Signal med fsenter=10000')

figure;
plot(f,abs(fft(xx3)),'-r');
xlabel('Frekvens (Hz)');
ylabel('Frekvenskomponent (rel. verdier)');
title('Fourier for signal med fsenter=10000')
```

```
% Wavelet-analyse
K = 48;         % Morlet-wavelet-bredde (kan være 6 - 400+)

fmin = 30.0;    % Minimum frekvens i waveletanalysen (i Hz)
fmax = 170.0;   % Maximum frekvens
WL1(xx1,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

fmin = 800.0;   % Minimum frekvens i waveletanalysen (i Hz)
fmax = 1200.0;  % Maximum frekvens
WL1(xx2,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

fmin = 9000.0;  % Minimum frekvens i waveletanalysen (i Hz)
fmax = 11000.0; % Maximum frekvens
WL1(xx3,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

%Autokorrelasjons-analyse
tittel='Autokorrelasjon for Signal1';
AutoCorr(xx1,N,Fs,tittel);

tittel='Autokorrelasjon for Signal2';
AutoCorr(xx2,N,Fs,tittel);

tittel='Autokorrelasjon for Signal3';
AutoCorr(xx3,N,Fs,tittel);
end
```

6.2 Del1-2

```
function Del1b
% Lager et tilfeldig signal som ønsket
Fs = 44100;
N = 1024*64;
T = N/Fs;
t = linspace(0,T*(N-1)/N, N);
f = linspace(0,Fs*(N-1)/N,N);
fsenter = 10000;
xx1 = HvitStoyGauss(Fs,N,fsenter,0.5);
xx2 = HvitStoyGauss(Fs,N,fsenter,10);
xx3 = HvitStoyGauss(Fs,N,fsenter,100);
xx4 = HvitStoyGauss(Fs,N,fsenter,1000);
```

```
xx5 = HvitStoyGauss(Fs,N,fsenter,5000);

% Ploter frekvens- og tidsbildet
figure;
plot(t,xx1,'-g');
xlabel('Tid (s)');
ylabel('Signal (vilkårlig enhet)');
title('Signal med fsigma=0.5')

figure;
plot(f,abs(fft(xx1)),'-r');
xlabel('Frekvens (Hz)');
ylabel('Frekvenskomponent (rel. verdier)');
title('Fourier for signal med fsigma=0.5')

figure;
plot(t,xx2,'-g');
xlabel('Tid (s)');
ylabel('Signal (vilkårlig enhet)');
title('Signal med fsigma=10')

figure;
plot(f,abs(fft(xx2)),'-r');
xlabel('Frekvens (Hz)');
ylabel('Frekvenskomponent (rel. verdier)');
title('Fourier for signal med fsigma=10')

figure;
plot(t,xx3,'-g');
xlabel('Tid (s)');
ylabel('Signal (vilkårlig enhet)');
title('Signal med fsigma=100')

figure;
plot(f,abs(fft(xx3)),'-r');
xlabel('Frekvens (Hz)');
ylabel('Frekvenskomponent (rel. verdier)');
title('Fourier for signal med fsigma=100')

figure;
```

```
plot(t,xx4,'-g');
xlabel('Tid (s)');
ylabel('Signal (vilkårlig enhet)');
title('Signal med fsigma=1000')

figure;
plot(f,abs(fft(xx4)),'-r');
xlabel('Frekvens (Hz)');
ylabel('Frekvenskomponent (rel. verdier)');
title('Fourier for signal med fsigma=1000')

figure;
plot(t,xx5,'-g');
xlabel('Tid (s)');
ylabel('Signal (vilkårlig enhet)');
title('Signal med fsigma=5000')

figure;
plot(f,abs(fft(xx5)),'-r');
xlabel('Frekvens (Hz)');
ylabel('Frekvenskomponent (rel. verdier)');
title('Fourier for signal med fsigma=5000')



% Wavelet analyse
K = 48;             % Morlet-wavelet-bredde (kan være 6 - 400+)

fmin =  9000.0;     % Minimum frekvens i waveletanalysen (i Hz)
fmax = 11000.0;     % Maximum frekvens
WL1(xx1,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

fmin =  9500.0;     % Minimum frekvens i waveletanalysen (i Hz)
fmax = 10500.0;     % Maximum frekvens
WL1(xx2,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

fmin =  9000.0;     % Minimum frekvens i waveletanalysen (i Hz)
fmax = 11000.0;     % Maximum frekvens
WL1(xx3,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon
```

```
fmin =  8000.0;     % Minimum frekvens i waveletanalysen (i Hz)
fmax = 12000.0;     % Maximum frekvens
WL1(xx4,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

fmin =  4000.0;     % Minimum frekvens i waveletanalysen (i Hz)
fmax = 25000.0;     % Maximum frekvens
WL1(xx5,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

% Autokorrelasjons-analyse
tittel='Autokorrelasjon for Signal1';
AutoCorr(xx1,N,Fs,tittel);

tittel='Autokorrelasjon for Signal2';
AutoCorr(xx2,N,Fs,tittel);

tittel='Autokorrelasjon for Signal3';
AutoCorr(xx3,N,Fs,tittel);

tittel='Autokorrelasjon for Signal4';
AutoCorr(xx4,N,Fs,tittel);

tittel='Autokorrelasjon for Signal5';
AutoCorr(xx5,N,Fs,tittel);
end
```

6.3 Del2-1

```
function Del2a


%Genererer signalene k1 og k2
N  = 64*1024; % 2^6*2^10 = 2^16
Fs = 44100;
fsenter = 10000;
fsigma  = 1000;

k1 = HvitStoyGauss(Fs,N,fsenter,fsigma);
k2 = HvitStoyGauss(Fs,N,fsenter,fsigma);

%Genererer f og g for alle forskyvninger dn og
%regner ut krysskorrelasjon mellom f og g for forskjellige dn
```

```
f = k1+k2;
g = zeros(N,1);
CrossCorrFunk = zeros(2001, 1);

for dn=0:2000
    for i=1:60000
        g(i) = k1(i + dn) + k2(i);
    end
    CrossCorrFunk(dn+1) = 1/N.*CrossCorr(f,g);
end

dn = (0:2000);
snitt = mean(CrossCorrFunk( (3/4)*2001:2001) );
maks = max(CrossCorrFunk) - snitt;
linje = maks/2 + snitt;


figure;
plot(dn,CrossCorrFunk,'b-',dn,linje, 'r-')
title('CrossCorrelation vs dn')
xlabel('dn[rel.unit]')
ylabel('CrossCorrelation[rel.unit]')


% Wavelet analyse
K = 48;         % Morlet-wavelet-bredde (kan være 6 - 400+)
fmin =  8000.0;  % Minimum frekvens i waveletanalysen (i Hz)
fmax = 12000.0; % Maximum frekvens
WL1(f,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

%g1
dn = 0;
g = zeros(N,1);
for i=1:60000
    g(i) = k1(i + dn) + k2(i);
end
WL1(g,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

%g2
dn = 4;
```

```
g = zeros(N,1);
for i=1:60000
    g(i) = k1(i + dn) + k2(i);
end
WL1(g,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

%g3
dn = 9;
g = zeros(N,1);
for i=1:60000
    g(i) = k1(i + dn) + k2(i);
end
WL1(g,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

%g4
dn = 13;
g = zeros(N,1);
for i=1:60000
    g(i) = k1(i + dn) + k2(i);
end
WL1(g,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

%g5
dn = 82;
g = zeros(N,1);
for i=1:60000
    g(i) = k1(i + dn) + k2(i);
end
WL1(g,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

end
```

6.4 Del2-2

```
function Del2b

%Genererer signalene k1 og k2
N  = 64*1024; % 2^6*2^10 = 2^16
Fs = 44100;
%T = N/Fs;
%t = linspace(0,T*(N-1)/N,N);
```

```
Fmax = 1000;
fsenter = 10000;
fsigma  = 1000;

k1 = HvitStoyGauss(Fs,N,fsenter,fsigma);
k2 = HvitStoyGauss(Fs,N,fsenter,fsigma);

%Genererer f og g
f = k1+k2;
A2LP = LowPass(f.*f,N,Fs,Fmax);
CrossCorrFunk = zeros(2001,1);
g = zeros(N,1);

for dn=0:2000
    for i=1:60000
        g(i) = k1(i + dn) + k2(i);
    end
    B2LP = LowPass(g.*g,N,Fs,Fmax);
    CrossCorrFunk(dn+1) = CrossCorr(A2LP,B2LP);
end
CrossCorrFunk = 1/max(CrossCorrFunk) .*CrossCorrFunk;

%Regner ut krysskorrelasjon mellom f og g med forskjellige dn
dn = (0:2000);
snitt = mean(CrossCorrFunk((3/4)*2001:2001));
maks = max(CrossCorrFunk) - snitt;
linje = maks/2 + snitt;

%Plotting
figure;
plot(dn,CrossCorrFunk,'b-',dn,linje, 'r-')
title('CrossCorrelation vs dn')
xlabel('dn[m]')
ylabel('CrossCorrelation[rel.unit]')

%Wavelet for forskjellige dn
K    = 48;
fmin = 300;
fmax = 1100;
```

```
%f og g1
dn = 0;
g  = zeros(N,1);
for i=1:60000
    g(i) = k1(i + dn) + k2(i);
end
B2LP = LowPass(g.*g,N,Fs,Fmax);
WL1(A2LP,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon
WL1(B2LP,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

%g2
dn = 4;
g  = zeros(N,1);
for i=1:60000
    g(i) = k1(i + dn) + k2(i);
end
B2LP = LowPass(g.*g,N,Fs,Fmax);
WL1(B2LP,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

%g3
dn = 9;
g  = zeros(N,1);
for i=1:60000
    g(i) = k1(i + dn) + k2(i);
end
B2LP = LowPass(g.*g,N,Fs,Fmax);
WL1(B2LP,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

%g4
dn = 26;
g  = zeros(N,1);
for i=1:60000
    g(i) = k1(i + dn) + k2(i);
end
B2LP = LowPass(g.*g,N,Fs,Fmax);
WL1(B2LP,N,fmin,fmax,K,Fs); % Kaller på wavelettransformasjon

end
```

6.5 Del3

```
function Del3
%Konstanter
v       = 340.29; %Speed of sound at sea level [m/s]

%Genererer signalene
N       = 1024*64;
Fs      = 44100;
Fmax    = 1000; % Lowpass frequency max
fsenter = 7000;
fsigma  = 6000;

k01     = HvitStoyGauss(Fs,N,fsenter,fsigma);
k02     = HvitStoyGauss(Fs,N,fsenter,fsigma);
k03     = HvitStoyGauss(Fs,N,fsenter,fsigma);
k04     = HvitStoyGauss(Fs,N,fsenter,fsigma);
k05     = HvitStoyGauss(Fs,N,fsenter,fsigma);
k06     = HvitStoyGauss(Fs,N,fsenter,fsigma);
k07     = HvitStoyGauss(Fs,N,fsenter,fsigma);
k08     = HvitStoyGauss(Fs,N,fsenter,fsigma);
k09     = HvitStoyGauss(Fs,N,fsenter,fsigma);
k10     = HvitStoyGauss(Fs,N,fsenter,fsigma);
k11     = HvitStoyGauss(Fs,N,fsenter,fsigma);

a       = linspace(0,5,1001);
s       = 10;
h       = 400;
dr      = zeros(11,2);
dn      = zeros(11,2);

CCFunk  = zeros(numel(a),1);
%Beregner avstander fra kilder til mottakere

k = 1;
while k <= numel(a)
    AB = Avstander(a(k),s,h);
    for j=1:2
        for i=1:11
            dr(i,j) = AB(i,j)- h;
            dn(i,j) = round(Fs.*dr(i, j)./v);
```

```
        end
    end

    %Beregner signalene motatt i A og B
    MottattA = zeros(1,N-max(max(dn)));
    MottattB = zeros(1,N-max(max(dn)));
    for i=1:(N-max(max(dn)))
        MottattA(i) = k01(i+dn(1,1))  + k02(i+dn(2,1)) + k03(i+dn(3,1))...
                    + k04(i+dn(4,1))  + k05(i+dn(5,1)) + k06(i+dn(6,1))...
                    + k07(i+dn(7,1))  + k08(i+dn(8,1)) + k09(i+dn(9,1))...
                    + k10(i+dn(10,1)) + k11(i+dn(11,1));

        MottattB(i) = k01(i+dn(1,2))  + k02(i+dn(2,2)) + k03(i+dn(3,2))...
                    + k04(i+dn(4,2))  + k05(i+dn(5,2)) + k06(i+dn(6,2))...
                    + k07(i+dn(7,2))  + k08(i+dn(8,2)) + k09(i+dn(9,2))...
                    + k10(i+dn(10,2)) + k11(i+dn(11,2));
    end

    %Lowpass filters

    %Mottatt signal i A etter Lowpass
    A2LP = LowPass(MottattA.*MottattA,N,Fs,Fmax);
    %Mottatt signal i B etter Lowpass
    B2LP = LowPass(MottattB.*MottattB,N,Fs,Fmax);

    %Krysskorrelasjon
    CCFunk(k) = CrossCorr(A2LP,B2LP);
    %Counter
    k = k + 1;
end
CCFunk = 1/max(CCFunk).*CCFunk;

snitt = mean(CCFunk((3/4)*numel(a):numel(a)));
maks  = max(CCFunk) - snitt;
linje = maks/2 + snitt;

figure;
plot(a, CCFunk, 'b-', a, linje, 'r-')
title('Plot of CrossCorrelation vs distance between detectors')
xlabel('Distance between detectors [m]')
```

```
ylabel('CrossCorrelation[rel.unit]')
end
```

6.6 AutoCorr

```
function F = AutoCorr(xx,N,Fs,tittel)
M = N/2;
%Lager funksjonen
F = zeros(N,1);

for j=1:M
    F(j) = sum( xx(1:M).*xx(j:M+j-1) );

end
F = 1/F(1) .*F;
T = N/Fs;
tau = linspace(0,T*(N-1)/N, N);

%Plotting
F = abs(F);
Fmax = max(F);

figure;
plot(tau(1:N/2),F(1:N/2),'b-'...
    ,tau(1:N/2),Fmax/2  ,'r-') %Funksjonen

xlabel('Tid[s]')
ylabel('Autokorrelasjonsfunk[rel.enh]')
title(tittel)
hold off
```

6.7 Avstander

```
function AB = Avstander(a,s,h)

%Genererer en matrise AB,
%med avstanden fra punkt A til kilde 1-11 i rad 1,
% og avstanden fra punkt B til kilde 1-11 i rad 2

AB = zeros(11,2);
```

```
%A distances
for i=1:5
    AB(6-i,1) = sqrt((i/10*s)*(i/10*s)+(h)*(h));
end
AB(6,1)  = h;
AB(7,1)  = AB(5,1);
AB(8,1)  = AB(4,1);
AB(9,1)  = AB(3,1);
AB(10,1) = AB(2,1);
AB(11,1) = AB(1,1);

%B distances
for i=1:11
   b = s/2 - (i - 1)*s/10;
   c = a - b;
   AB(12-i,2) = sqrt((c)*(c)+(h)*(h));
end
end
```

6.8 CrossCorr

```
function CC = CrossCorr(f, g)
 % Give cross-correlation of two functions f and g.
 CC = sum(abs(f.*g));
end
```

6.9 HvitStoyGauss

```
function y = HvitStoyGauss(Fs,N,fsenter,fsigma)

%y = zeros(N,1);
%T = N/Fs;
%t = linspace(0,T*(N-1)/N,N);
f = linspace(0,Fs*(N-1)/N, N);
%nsenter = floor(N*fsenter/(Fs*(N-1)/N));
%nsigma = floor(N*fsigma/(Fs*(N-1)/N));
gauss = exp(-(f-fsenter).*(f-fsenter)/(fsigma*fsigma));
%figure;
%plot(f,gauss,'-k'); % For sjekking!

ampl = rand(N,1);
```

```
ampl = ampl.*transpose(gauss);
%figure;
%plot(f,ampl,'-g'); % For sjekk

faser = rand(N,1);
faser = faser*2*pi;
y = ampl.*(cos(faser) + 1i.*sin(faser));

%Fikser spekteret slik at kriterier for folding er ok
y(1) = 0 + 0i;
for i=1:N/2-1
    y(N/2+1+i) = conj(y(N/2+1-i));
end

%Plotter fourier spekteret til hvit stoy
% figure;
% plot(f,y)
% title('Fourier spekteret til hvit stoy')

y = ifft(y);

% %test
% figure;
% plot(f,real(y))
% title('Real del av signalet')
% figure;
% plot(f,imag(y))
% title('Imaginaer del av signalet')

y = real(y);
end
```

6.10 LowPass

```
function v = LowPass(u, N, Fs, Fmax)
 % Lowpass filter.
 % This program removes all frequencies above a frequency fmax in the
 % frequency spectrum of a signal u.
 % N is the length of u and Fs is the sampling frequency of u.
 %
 % Returns the real part of the signal after the frequencies have been
```

```
 % stripped. It should be noted that the imaginary part of all v are 0
 % before we remove the imaginary part. The only reason we bother to remove
 % it is in hopes of faster computation.
u = fft(u);
nmax =floor(Fmax/(Fs*(N-1))*N*N);
u(nmax+1:N-nmax+1) = 0;
v = real(ifft(u));
end
```

6.11 WL1

```
function WL1(h,N,fmin,fmax,K,fs)
% Funksjon for kontinuerlig diskret Morlet waveletanalyse av en array h.
% Array har N elementer samplet ved samplingsfrekvensen fs.
% Morlet waveletanalysen skal gjennomfres i frekvensintervallet
% [fmin,fmax] med K-parameteren K. Se detaljer i et kapittel i lreboka
% Svingninger og blger, Arnt Inge Vistnes, 2014.

FTsignal = fft(h);
T = N/fs; % Total tid lydutsnittet tar (i sek)
t = linspace(0,T*(N-1)/N,N);
f = linspace(0,fs*(N-1)/N, N);

% Beregner # analysefrekvenser, skriver til skjerm, klargjr frekvensene
M = floor(log(fmax/fmin) / log(1+(1/(8*K)))) + 1;
AntallFrekvenserIAnalyse = M
ftrinn = (fmax/fmin)^(1/(M-1));
f_analyse = fmin;

% Allokerer plass til waveletdiagrammet og array for lagring av frekvenser
WLdiagram = zeros(M,N);
fbrukt = zeros(1,M);

%Lkke over alle frekvenser som inngr i analysen
for jj=1:M
    faktor=(K/f_analyse)*(K/f_analyse);
    FTwl=exp(-faktor*(f-f_analyse).*(f-f_analyse));
    FTwl=FTwl-exp(-K*K)*exp(-faktor*(f.*f));%Litekorreksjonsledd
    FTwl=2.0*FTwl;      %Faktor(ulikevalg!)
    %Beregnersenhellinjeiwaveletdiagrammetinjafs!
    %WLdiagram(jj,:)=abs(ifft(FTwl.*transpose(FTsignal))); %Ettalternativ
```

```
    %WLdiagram(jj,:)=sqrt(abs(ifft(FTwl.*(FTsignal))));%Ettannet
    WLdiagram(jj,:)=sqrt(abs(ifft(FTwl.*transpose(FTsignal))));%Ett til

    %Brukerdensistevariantenforfsvakepartierbedresynlig
    fbrukt(jj)=f_analyse; %Lagrerfrekvensenesomfaktiskerbrukt
    f_analyse=f_analyse*ftrinn; %Beregnernestefrekvens
end;
%Reduserenfilstrrelsevedfjernemyeavoverfldiginformasjonitid.
%Dettegjreskunforatfilstrrelsenpplotteneskalblihndterbar.
P = floor((K*fs)/(24*fmax));%Tallet24kanendresvedbehov
TarBareMedHvertXITid = P
NP = floor(N/P);
AntallPktITid = NP
for jj=1:M
    for ii=1:NP
        WLdiagram2(jj,ii)=WLdiagram(jj,ii*P);
        tP(ii)=t(ii*P);
    end;
end;
%Foretaenmarkeringiplottetforviseomrdermedrandproblemer
maxverdi=max(WLdiagram2);
mxv=max(maxverdi);
for jj=1:M
    m=floor(K*fs/(P*pi*fbrukt(jj)));
    WLdiagram2(jj,m)=mxv/2;
    WLdiagram2(jj,NP-m)=mxv/2;
end;

%Plotterwaveletdiagrammet
figure;
imagesc(tP,log10(fbrukt),WLdiagram2,'YData',[1 size(WLdiagram2,1)]);
set(gca,'YDir','normal');
xlabel('Tid(sek)');
ylabel('Log10(frekvensiHz)');
%title('WaveletPowerSpektrum');%Velgdennenrdeteraktuelt,
title('Sqrt(WaveletPowerSpektrum)');%mendennenrsqrtblirbrukt
colorbar('location','southoutside');
end
```