FYS2130 - Oblig 4

Aleksander Hansen

26. april 2013

Jeg manglet en oblig mellom 1-5. Arnt Inge sa det var greit at jeg leverte oblig 4 på nytt, selv om fristen har gått ut.

Oppgave 1

Vi kan lage en syntetisk lyd ved å spille av de viktigste, mest fremtredende frekvensene tatt fra frekvensspekteret til et instrument. Siden vi ikke tar med alle frekvensene vil ikke lyden være identisk med lyden fra instrumentet.

Oppgave 2

Shannon-Nyquists samplingsteorem sier at man må ha en samplingsfrekvens på minst 2 ganger den største frekvensen man måler hvis resultatet skal være entydig. Bruker man en samplingsfrekvens som er mindre risikerer man at avspillingen av lyden ikke høres ut som den man tok opp i utgangspunktet.

Oppgave 3

Ved fouriertransformasjon av et reelt signal trenger vi bare å plotte frekvensen som svarer til de første $N/2+1$ fourierkoeffisientene siden resten er bare redudant informasjon i form av de kompleks konjungerte av de første koeffisientene. Dette kalles speiling/folding.

Oppgave 5

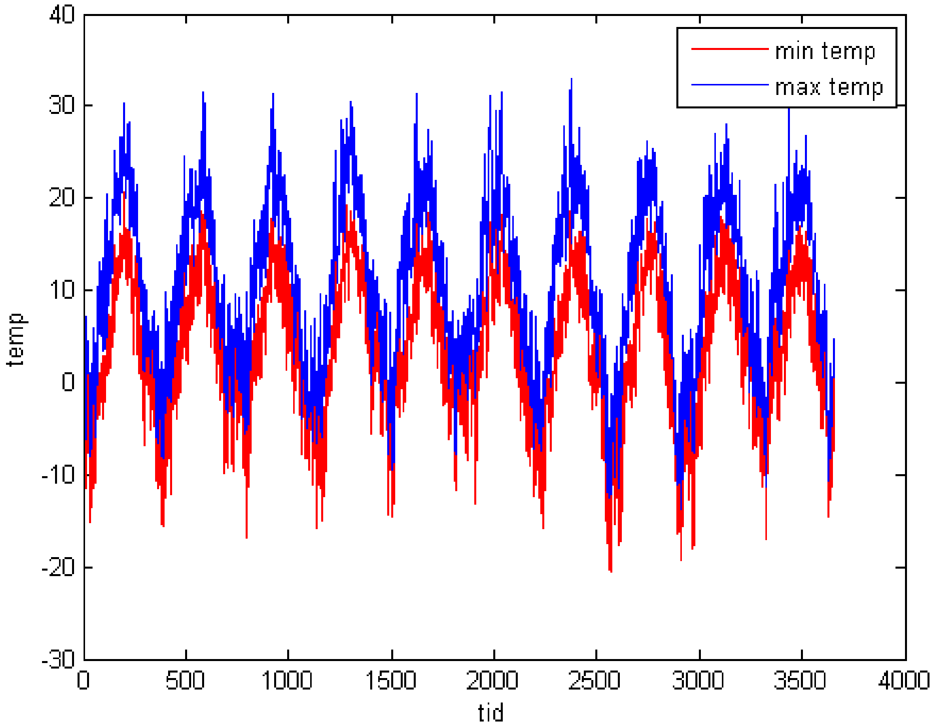
Linje (k=)	Samplinger pr. periode
0	0, 1, $\aleph_0$, $2^{\aleph_0}$?
1	8
2	4
3	$8/3 = 2.\bar{6}...$
N-3	$8/5 = 1.6$
N-2	$8/6 = 1.\bar{3}...$
N-1	$8/7 = 1.1428...$

Som sagt i oppgave 2, det må være over 2 samplinger pr. periode for å få et utvetydig resultat. Siden samplingene er uniformt fordelt over en periode, og cosinus og sinus er henholdsvis en like- og odde-funksjon, vil vi nødvendigvis få en speiling om midten.

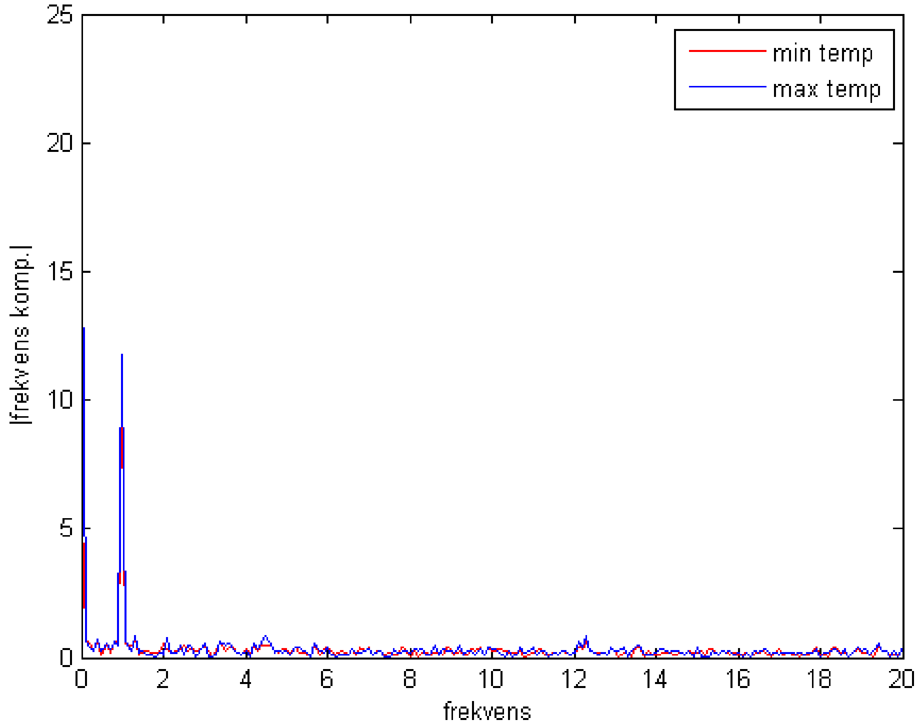
Oppgave 7

```
filnavn = 'tempBlindern10aar.txt';
fileID = fopen(filnavn, 'r');
A = fscanf(fileID, '%d %d %f %f %f', [5, inf]);
```

```
minT = A(4,:);
maxT = A(5,:);
plot(minT, '-r');
hold on;
plot(maxT, '-b');

Fs = 365.25;
N = length(maxT);
Xmax = fft(maxT,N)/N;
Xmin = fft(minT,N)/N;
f = (Fs/2)*linspace(0, 1, N/2 +1);

figure()
plot(f, 2*abs(Xmax(1:N/2+1)),'b-');
hold on;
plot(f, 2*abs(Xmin(1:N/2+1)),'r-');
xlabel('frekvens')
ylabel('|frekvens komp.|')
legend('max temp', 'min temp')
xlim([0,20])
```



Figur 1: Tidsbildet

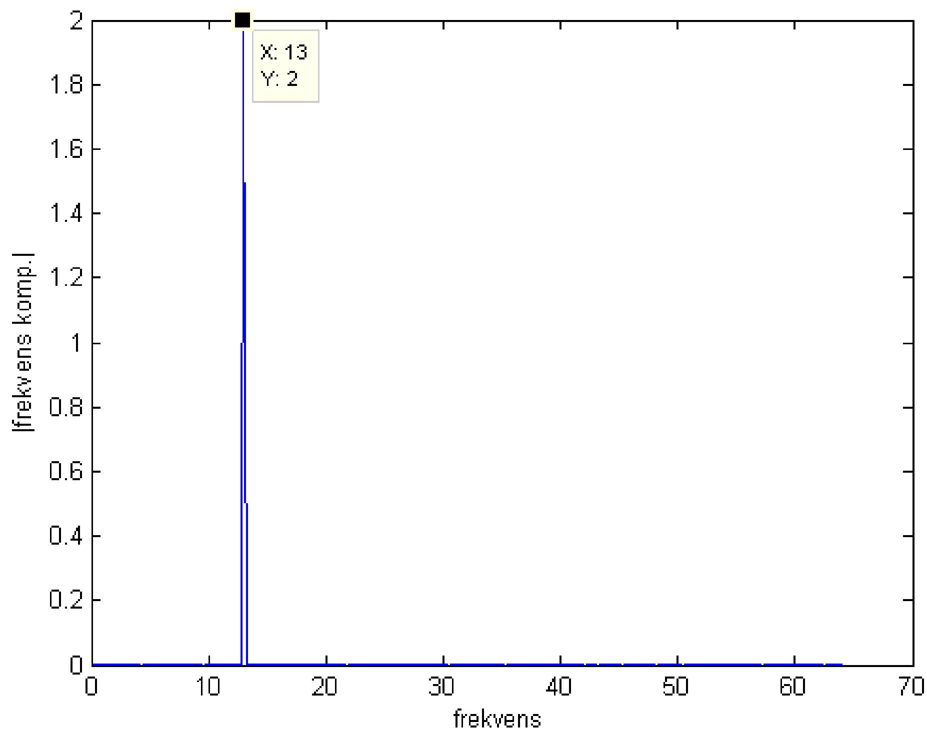


Figur 2: Frekvensbildet

Oppgave 11

```
Fs = 128;
dt = 1/Fs;
N = 512;
t = (0:N-1)*dt;

% rent sinussignal
y = 2.0*sin(13*2*pi*t);

figure
plot(t, y);

Y = fft(y, N)/N;
f = (Fs/2)*linspace(0, 1, N/2+1);

figure
```

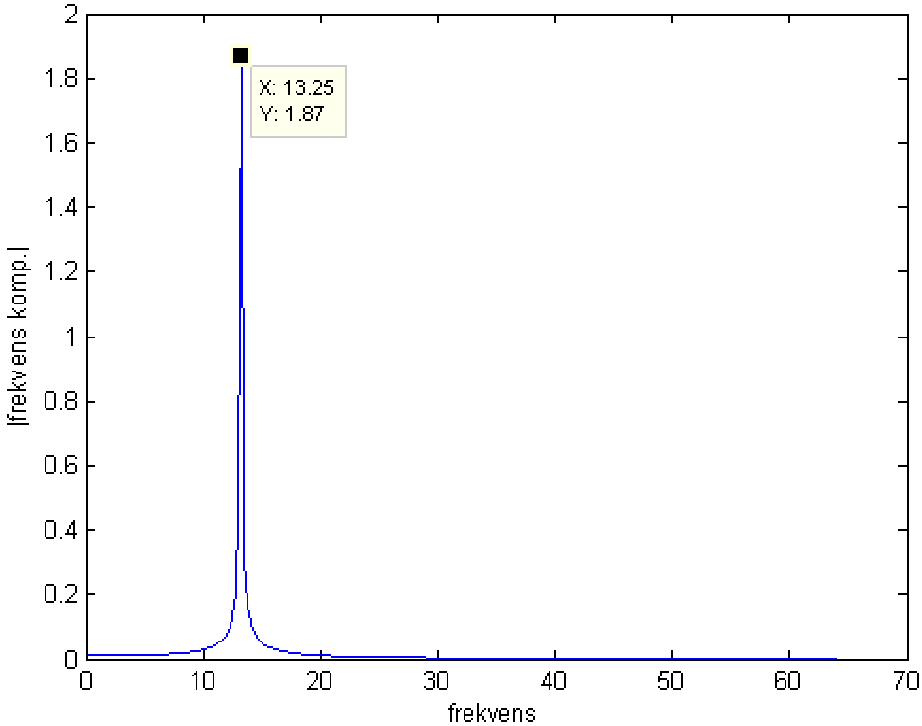
```
plot(f, 2*abs(Y(1:N/2+1)));
xlabel('frekvens')
ylabel('|frekvens komp.|')
```



Figur 3: Frekvensbildet til oppg 11

Oppgave 12

Frekvensspekteret er ikke like "definert" som i oppg 11.

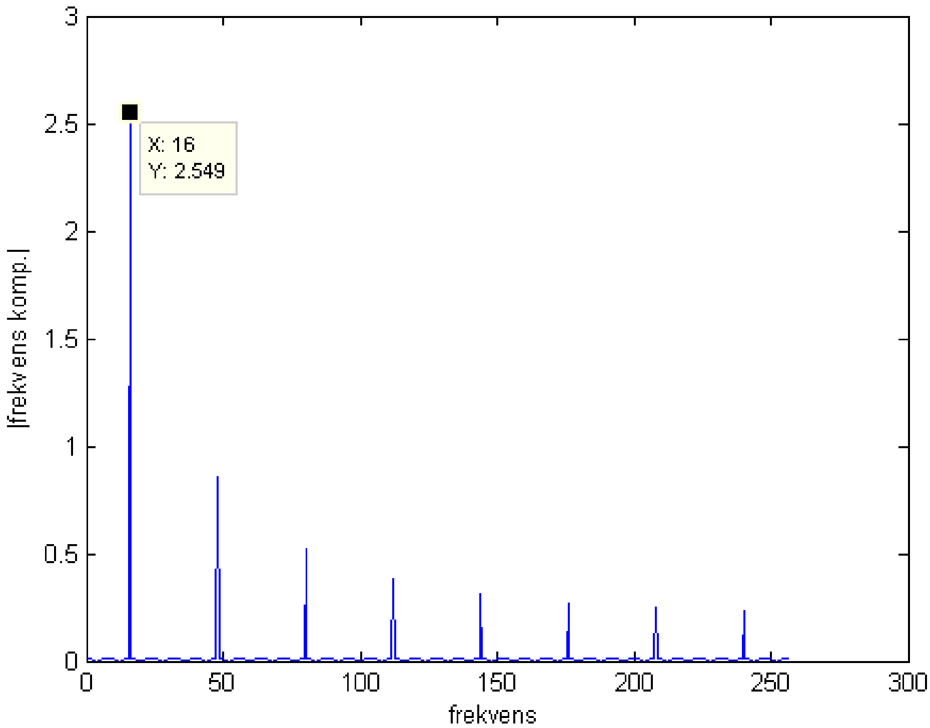


Figur 4: Frekvensbildet til oppg 12

Oppgave 13

Bytter ut sinussignalet med:

```
y = 2.0*square(16*2*pi*t);
```



Figur 5: Frekvensbildet til oppg 13

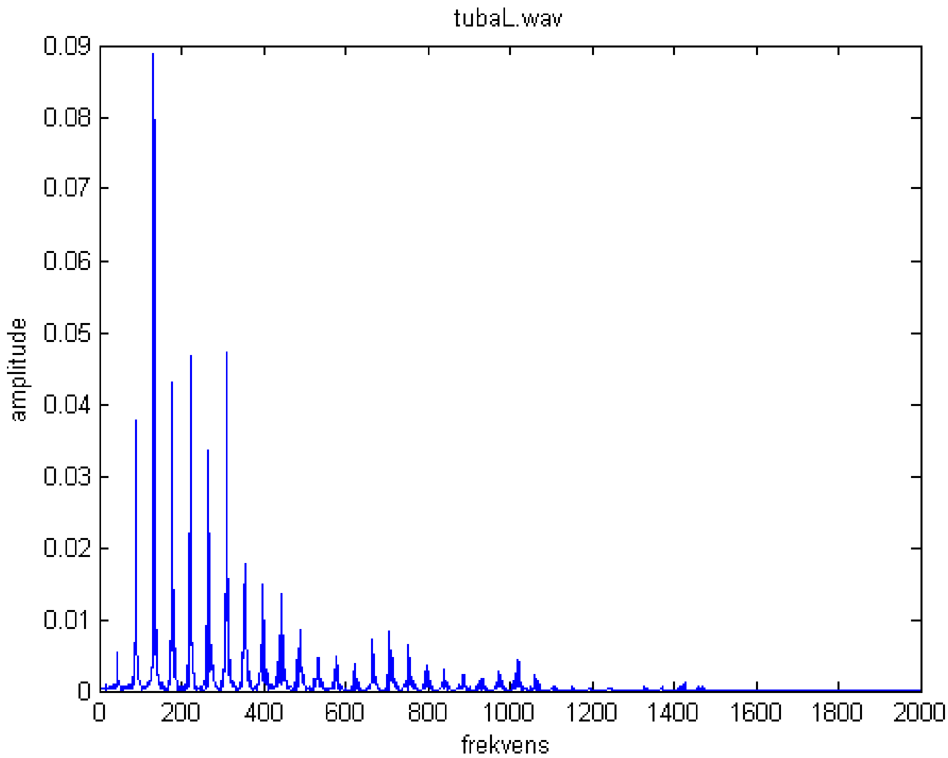
Den relative amplituden til frekvenskomponentene følger rekken $\{1, \frac{1}{3}, \frac{1}{5}, ...\}$ (hvis det lager mening). Som er tilnærmet det samme som jeg får, som er: $\{2.549, 0.8568, 0.5232, 0.3844, ...\}$

Oppgave 21

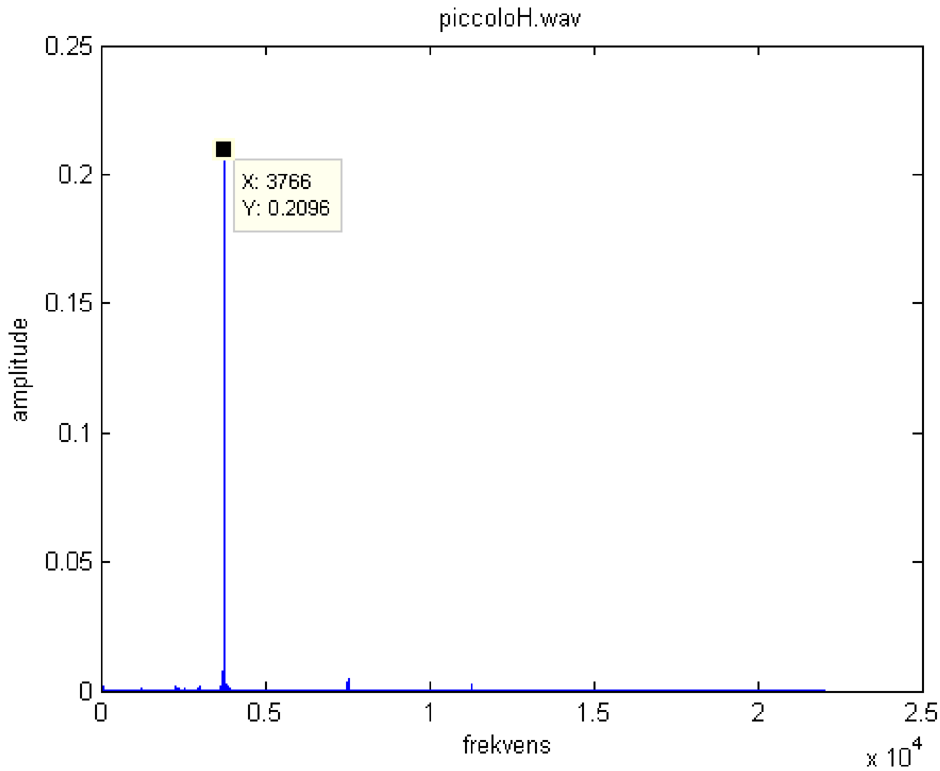
Jeg får en frekvens på 132 Hz for *tubaH.wav*. Da er det vel en C på den tempererte skalaen i figur 4.19. For *piccoloH.wav* får jeg 3766 Hz som er vel hva enn den svarte tasten mellom A og H på slutten er.

```
filnavn = 'tubaL.wav'
N = 2^16;
Nstart = 1;
[y, Fs, type] = wavread(filnavn, [Nstart, Nstart+N+1]);

wavplay(y, Fs);
g = y(:,1);
```

```
Y = fft(g, N)/N;
f = (Fs/2)*linspace(0, 1, N/2 + 1);

plot(f, 2*abs(Y(1:N/2 + 1)))
xlim([0, 2000])
xlabel('frekvens')
ylabel('amplitude')
title(filnavn)
```



Figur 6: Frekvensbildet til tubaL.wav



Figur 7: Frekvensbildet til piccoloH.wav