

FYS2130 Svingninger & Bølger

Obligatorisk oppgave 3

Kenneth Ramos Eikrehagen

22. februar 2018

Innhold

| | | |
|----------|--|----------|
| 1 | Forståelses- og diskusjonsspørsmål | 4 |
| 1.1 | Samplingsfrekvensen og lavpassfilter | 4 |
| 1.2 | Forskjell mellom fourier transformasjon og waveletanalyse . . . | 5 |
| 1.3 | Situasjoner fouriertransformasjon gir ubrukelig resultat | 5 |
| 1.4 | Ulemper med wavelettransformasjon sammenlignet med fouriertransformasjon | 6 |
| 1.5 | Wavelettransformasjon sitt “randproblem” | 6 |
| 2 | Regneoppgaver | 7 |
| 2.1 | Første punkt i en digital FT av et signal er lik gjennomsnittsverdien til signalet | 7 |
| 2.2 | Kommer linjene ut i frekvensspekteret? | 7 |
| 2.3 | Solflekke | 9 |
| 2.4 | Harmonisk signal | 10 |
| 2.5 | Wavelettransformasjon og Heisenbergs uskarphetsrelasjon . . . | 12 |

Figurer

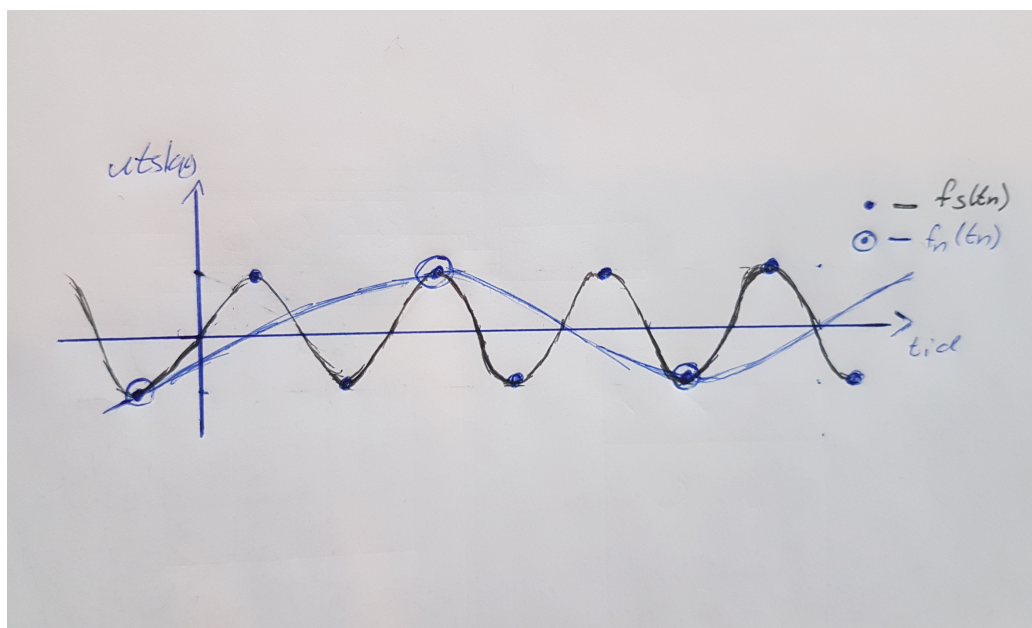
| | | |
|----|--|----|
| 1 | Samplingfrekvens | 4 |
| 2 | Første punkt i en digital fouriertransformasjon numerisk . . . | 7 |
| 3 | Frekvens = 100 | 8 |
| 4 | Frekvens = 200 | 8 |
| 5 | Frekvens = 400 | 8 |
| 6 | Frekvens = 700 | 8 |
| 7 | Frekvens = 950 | 8 |
| 8 | Frekvens = 1300 | 8 |
| 9 | Solflekker | 9 |
| 10 | Kode: Solflekker | 9 |
| 11 | Kode: Signal | 10 |
| 12 | Signal med 13 perioder | 11 |
| 13 | Signal med 13.2 perioder | 11 |
| 14 | Firkantbølge | 14 |
| 15 | Analysisk løsning av amplitude av en firkant-bølge | 14 |
| 16 | Zoom av tidsbildet til signalet | 15 |
| 17 | Frekvensspekter ved FFT | 15 |
| 18 | Wavelet spektrogram | 15 |
| 19 | Tidsbilde 14.d) | 16 |
| 20 | Frekvensspekter 14.d) | 16 |
| 21 | Wavelet spektrogram 14.d) | 16 |
| 22 | Kode: Wavelettransformasjon | 17 |

1 Forståelses- og diskusjonsspørsmål

1.1 Samplingsfrekvensen og lavpassfilter

For CD lyd er samplingsfrekvensen (f_s) 44.1 kHz. Hvorfor må vi ha et lavpassfilter mellom mikrofonforsterker og samplingskretsene som fjerner frekvenser over 22 kHz?

Hvis man bruker en fast fourier transformasjon (FFT) er den høyeste frekvensen vi kan oppløse $\frac{1}{2}f_s$. Hvis vi ignorerer dette kan det medføre at vi får høyere frekvens f_n enn den vi maksimalt kan oppløse og lyden som spilles av på CD'en vil være i form av støy. Jeg har prøvd å illustrer hva som skjer om $f_n > \frac{1}{2}f_s$ på figur 1. Som vi ser på figuren vil dette resultere i at vi ikke får tilstrekkelige punkter til å kunne evaluere det riktige signalet. Jeg har tegnet et signal med liten frekvens men i prinsippet kan det være uendelig mange frekvenser mellom disse punktene.



Figur 1: Den sorte linjen er samplingsfrekvensen $f_s(t_n)$ og $f_n(t_n)$ er en illustrasjon av at $f_n(t_n) > \frac{1}{2}f_s(t_n)$. De blå prikkene er punktene vi får hvis $f_n = \frac{1}{2}f_s$ og de blå prikkene med sirkel er eksempler på hva som kan skje hvis $f_n(t_n) > \frac{1}{2}f_s(t_n)$

1.2 Forskjell mellom fourier transformasjon og wavelet-analyse

| | Fouriertransformasjon | Wavelettransformasjon |
|----|--|--|
| 1) | Spektogrammet bygges opp linje for linje med <i>vertikale</i> linjer | Spektogrammet bygges opp linje for linje med <i>horisontale</i> linjer |
| 2) | Behøver ikke velge hele tidsintervallet | Må velge hele tidsintervallet før analysen |
| 3) | Kan ikke velge frekvensområde kjører gjennom hele signalet og finner alle frekvenser | Kan velge hvilket frekvensområdet man vil studere |
| 4) | God til stasjonære signaler | God til signaler som varierer over tid |
| 5) | „Oppdaget” for ca 200 år siden | „Oppdaget” for ca 100 år siden |
| 6) | 2D beskrivelse | 3D beskrivelse |
| 7) | $X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-i\omega t}dt$ | $\gamma k(\omega_a, t) = \int_{-\infty}^{\infty} x(t + \tau)\Psi_{\omega_a, k}(\tau)d\tau$ |

Slenger til et sitat fra [7]:

„ In layman’s terms: A fourier transform (FT) will tell you what frequencies are present in your signal. A wavelet transform (WT) will tell you what frequencies are present and where (or at what scale). If you had a signal that was changing in time, the FT wouldn’t tell you when (time) this has occurred. ”

Fouriertransformasjon blir begrenset av noe som kan minne om Heisenberg uskarphetsrelasjon. Hvis vi velger å måle høyest presisjon i tiden til signalet mister vi all kontroll over frekvensen, og vice versa. Wavelettransformasjon bruker dette til sin fordel ved at for hver måling av en wavelet forteller den oss noe om tids utstrekningen og frekvensen til signalet. ([4])

1.3 Situasjoner fouriertransformasjon gir ubrukelig resultat

Fast fouriertransformasjon (FFT) er ubrukelig hvis vi skal analysere signaler som endrer seg over tid. Da kan det heller lønne seg å bruke en „korttids-fouriertransformasjon” , det er altså en stykkevis fouriertransformasjon (STFT). Fourertransformasjon klarer ikke å måle presisjon i tid og frekvens

samtidig.

1.4 Ulemper med wavelettransformasjon sammenlignet med fouriertransformasjon

Ulempene med en wavelettransformasjon:

1. må velge hele tidsintervallet før analysen
2. hvis vi velger å gjøre en wavelettransformasjon av flere frekvenser, si 100, så vil den bruke omtrent 100 ganger så lang tid som en enkel fouriertransformasjon
3. fordi det er ulike start punkt og lengder underveis i beregningene gjør at det er vanskelig å holde orden på faser (fasorer)
4. i en waveletanalyse gir ikke bare utslag for frekvenser som tilsvarer signalets men også nærliggende frekvenser.
5. har et randproblem
6. må velge om vi vil a et skarpt frekvens-bilde eller et skarpt tidsbilde. Kan ikke få begge deler samtidig.

1.5 Wavelettransformasjon sitt “randproblem”

For å forklarer randproblemet til en wavelettransformasjon må jeg nevne hoved prinsippet med en slik transformasjon. Prinsippet i en wavelettransformasjon er å gange sammen et signal med en wavelet punkt for punkt og summere sammen alle produktene. Vi flytter så waveleten og gjør det samme på nytt, dette gjør vi om igjen og om igjen. Wavelettransformasjon bruker tiden den tar inn i midten av waveleten. Dette medfører at ved starten og slutten av perioden så vil kun halve waveleten være med i dataområdet som gjør at en del av begge endene i spekteret vil gi unyttige data. Hvor mye av endene som vi må forkaste kommer an på bredden vi velger waveleten til å være (stor eller liten k)

2 Regneoppgaver

2.1 Første punkt i en digital FT av et signal er lik gjennomsnittsverdien til signalet

Dette skal jeg vise matematisk og numerisk. For å vise dette matematisk bruker jeg ligning 5.18 i boka [1]. Jeg bruker at $k = 0$

$$\begin{aligned} X_{k=0} &= \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi}{N}(k=0)} \\ &= \frac{\sum_{n=0}^{N-1} x_n}{N} \end{aligned}$$

Første punktet da k er null blir gjennomsnittet til signalet, numerisk gjorde jeg det i python som du kan se i figur 2.

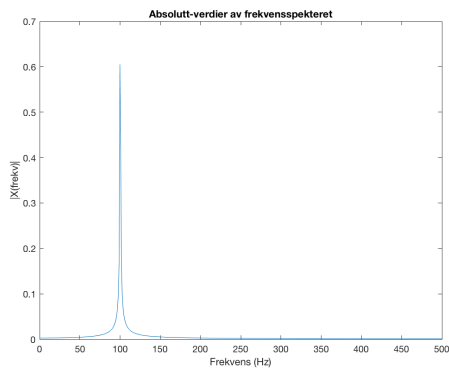
```
1 N = 500.0
2 x = 2*np.pi
3 t = np.linspace(0,10,N)
4 signal = np.sin(x*t)+np.cos(x*t)
5 FFT = fft(signal)/N
6 forste = FFT[0]
7 gjen = np.sum(signal)/N
8 diff = gjen - forste
9 print gjen, forste
10 """
11 terminal >>
12 0.002 (0.002+0j)
13 """
```

Figur 2: Første punkt i en digital fouriertransformasjon numerisk

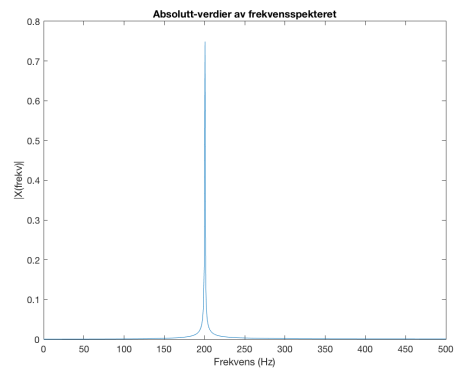
Vi har nå sett at første punkt i en digital fouriertransformasjon er gjennomsnittet til signalet, analytisk og numersik.

2.2 Kommer linjene ut i frekvensspekteret?

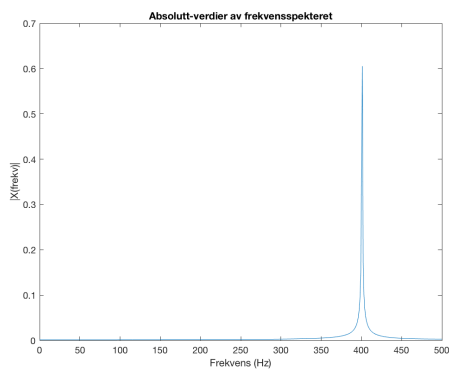
Figurene 3,4,5,6,7, 8 på neste side er de plotene jeg fikk da jeg satte inn for de forskjellige frekvensene. Frekvensene fram til 400 vet vi skal finne, men frekvenser fra 700 og oppover hadde vi ønsket at ikke ble synlig her. Det jeg observerer når jeg studerer frekvensspektrene er at når frekvensen blir for høy virker det som om den blir reflektert tilbake fra „veggen”. Figur 6 på neste side ser vi at den har gått ca 200 „skritt” tilbake fra veggen. Dette er akkurat differansen mellom frekvensen vi har og maks frekvens vi kan måle. Det samme fenomenet ser vi på figur 8 der den har blitt reflektert 2 ganger.



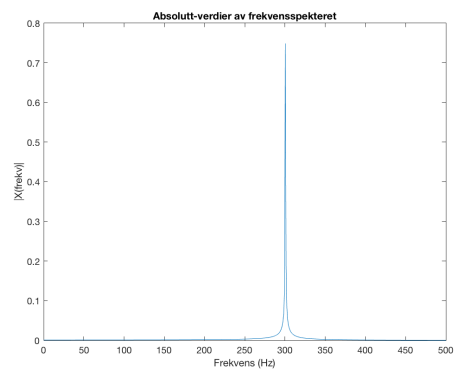
Figur 3: Frekvens = 100



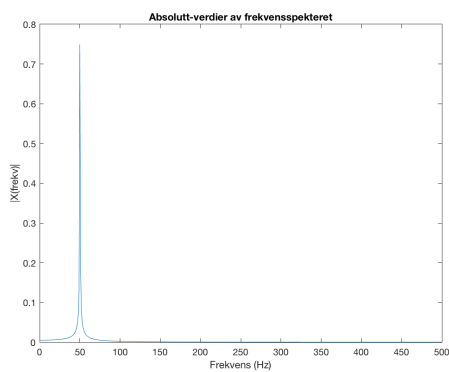
Figur 4: Frekvens = 200



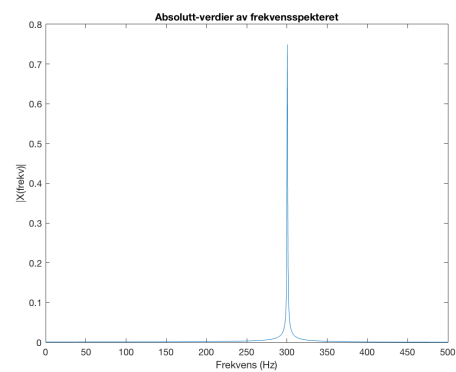
Figur 5: Frekvens = 400



Figur 6: Frekvens = 700



Figur 7: Frekvens = 950



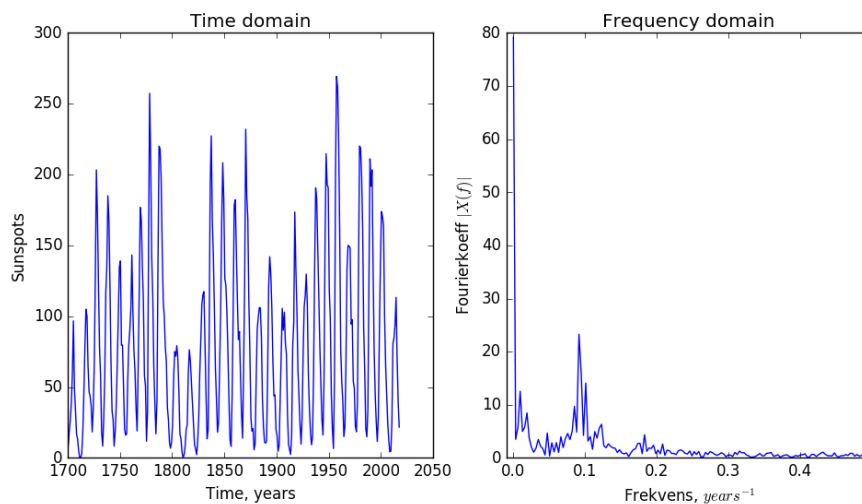
Figur 8: Frekvens = 1300

2.3 Solflekker

Plottet jeg fikk kan du se på figur 9. Som vi ser på frekvensbilde (frequency domain) er det en topp som kan være pga de høye amplitudene i tidsbilde

(time domain), den neste tydelige toppen er nok der fordi vi ser at tidsbilde har en del topper rundt 100. Dette kan bety at det er en periode på ca 10 år hvor det forekommer 100 solflekker.

Hvordan jeg løste oppgaven numerisk kan du se i figur 10.



Figur 9: Til venstre er tidsbilde til solflekkenene. Til høyre er frekvensbilde

Figur 10: Kode: Solflekker

```

1 data = urllib2.urlopen("http://www.sidc.be/silso/DATA/SN_y_tot_V2.0.txt")
2 times = []
3 sunspots = []
4 for line in data:
5     cols = line.split()
6     times.append( float(cols[0]) )
7     sunspots.append( float(cols[1]) )
8 #Array
9 tid = np.array(times)
10 solflekker = np.array(sunspots)
11 N = float(len(solflekker))
12 #Fast fourier transformasjon
13 fourier = fft(solflekker)/N
14 fs = 1.0/(tid[1]-tid[0])
15 freq = np.linspace(0,fs,N)

```

2.4 Harmonisk signal

Koden jeg laget med signalet jeg valgte ser du i figur 11. Tid- og frekvensbilde av det første signalet med akkurat 13 perioder ser du på figur 12 på

neste side. Tidsbildet viser et harmonisk signal med 13 perioder og frekvensspekteret viser en skarp og fin kurve på 1[Hz] akkurat som forventet. Hvis vi ser på figur 13 på neste side ser vi på tidsbildet et harmonisk signal på 13.2 perioder. Jeg observerer derimot at frekvensspekteret har endret seg. Den skarpe kurven som vi hadde ved forrige signal er nå blitt avrundet i bunn, kurven er også fått et mindre utslag. En fourier transformasjon antar at signalet er periodisk og stasjonært gjennom hele intervallet ([2]), noe den ikke lenger er i dette tilfellet. Det er nok forklaringen på forskjellen jeg observerer i frekvensspektrene.

Figur 11: Kode: Signal

```

1 def FFT(p,n): #p = perioder , n = samplinger
2     t = np.linspace(0,p,n)
3     signal = np.sin((2*np.pi)*t)
4     FFT = fft(signal)/float(n)
5     fs = 1.0/(t[1]-t[0])
6     freq = np.linspace(0,fs,n)
7     return signal,FFT,freq,t
8 def square(p,n): #p = perioder , n = samplinger
9     t = np.linspace(0,p,n)
10    square = signal.square((2*np.pi)*t)
11    fs = 1.0/(t[1]-t[0])
12    freq = np.linspace(0,fs,n)
13    FFT = fft(square)/n
14    return square,FFT,freq,t
15 #konstanter
16 N = 512.0
17 n = 2**(14)
18 #
19 signal_a,FFT_a,freq_a,t_a = FFT(13,N)
20 signal_b,FFT_b,freq_b,t_b = FFT(13.2,N)
21 square,FFT_square,freq_square,t_s = square(16,n)

```

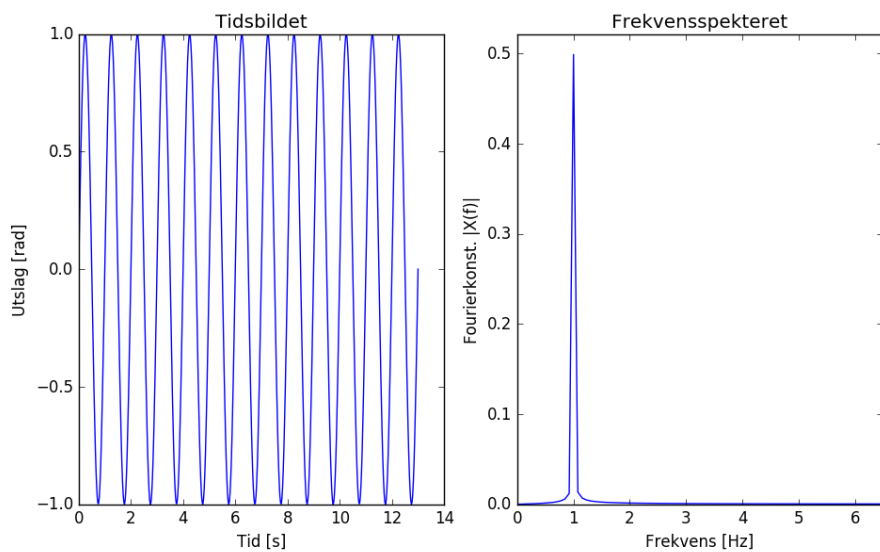
Jeg brukte `scipy.signal.square()` for å konstruere firkant-bølgen, som du ser du i koden (figur 11) under `def square(p,n)`. Firkant-bølgen med tidsbildet og frekvensspekter ser du i figur 14 på side 14. På internett ([6]) fant jeg et uttrykk for amplituden for en firkant-bølge.

$$A_n = \frac{2}{\pi(2n-1)}$$

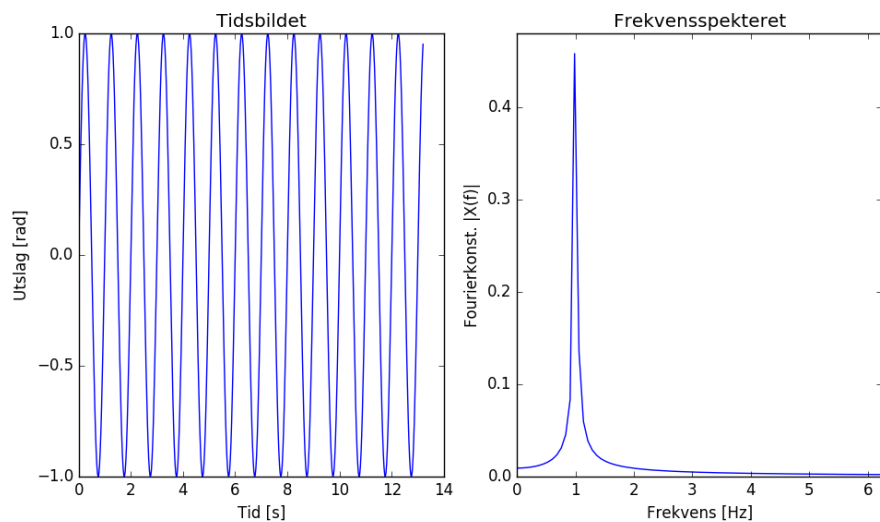
Når lar jeg n gå mot ∞

$$A_n = \frac{2}{\pi(2n-1)} = \lim_{n \rightarrow \infty} \frac{2}{\pi(2n-1)} = \frac{1}{n}$$

Dette resultatet samsvarer med det vi ser i frekvensspekteret til firkant-bølgen. $\frac{1}{n}$ gir oss nettopp en slik graf. Hvis vi sammen ligger frekvensspekteret



Figur 12: Signal med 13 perioder, Til venstre er tidsbilde og til høyre er frekvensspekteret



Figur 13: Signal med 13.2 perioder, Til venstre er tidsbilde og til høyre er frekvensspekteret

i figur 14 og grafen i figur 15 på side 14 ser vi at jeg får omtrent det samme fra mine numeriske beregninger.

2.5 Wavelettransformasjon og Heisenbergs uskarphetsrelasjon

På figur 16 og 17 på side 15 ser du tidsbildet til signalet og frekvensspekteret jeg fikk med FFT. Figur 18 på side 15 viser spektogrammet jeg får med en wavelettransformasjon. Jeg ser her at ved en lav K får vi et bra tidsbilde, og jeg ser godt at signalene varer igjennom hele intervallet. Med en stor K får jeg et skarpt bilde av frekvensene til dette signalet, men det minste signalet blir nesten borte. Ser også at randproblemet kommer litt frem her, ser det på det første og siste 0.1[s] på frekvensen ved 1[kHz].

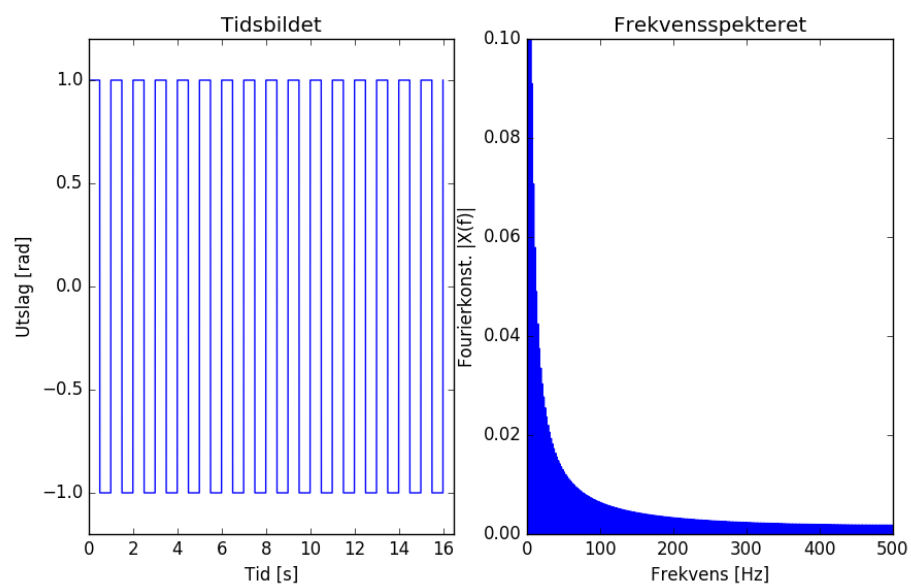
Neste jeg gjør er å konvultere signalet med en gaussisk funksjon slik at jeg får 2 „bølgepakker” som kommer tydelig frem i tidsbildet du ser i figur 19 på side 16. Jeg brukte deretter en FFT for å finne frekvensspekteret (figur 20 på side 16) til disse „bølgepakkene”, og som forventet ble dette likt som i figur 17 på side 15. Waveletspektogrammet ble annerledes, med å konvultere signalet har jeg fått frem at det kun er 2 frekvenser i dette signalet, spektogrammet ser du på figur 21 på side 16. På siste bilde der $K = 124$ har amplituden til det minste frekvensen forsvunnet! Dette viser at vi må være fornuftige av valg av bredden til waveleten. For stor kan medføre at informasjon forsvinner. Med en liten bredde ser vi derimot hvor lenge signalene varer, men på bekostning av at frekvensen blir mer utydelig. Jeg synes at ved $K = 100$ får vi et godt bilde av både tid og frekvens, ser tydelig at frekvensene er på 1.6 [kHz] og 1[kHz]. Og at den minste frekvensen varer i omlag 0.05[s] og den største frekvensen i omlag 0.4[s] som samsvarer med tidsbildet til signalet.

Koden jeg brukte ser du på figur 22 på side 17

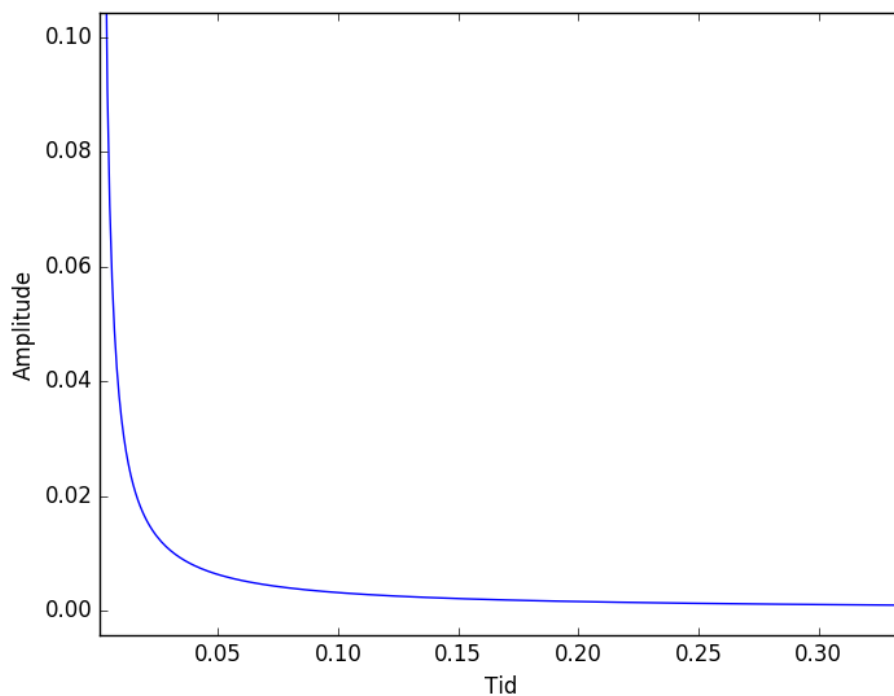
Referanser

- [1] Arnt Inge Vistnes *Svingninger og bølgers fysikk* first edition Desember 2016
- [2] Lasse Clausen *Forelesninger* vår 2018
- [3] Stor takk til gruppelærere
- [4] <https://no.wikipedia.org>
- [5] <https://snl.no>
- [6] <http://www.informit.com/articles/article.aspx?p=1374896&seqNum=7>

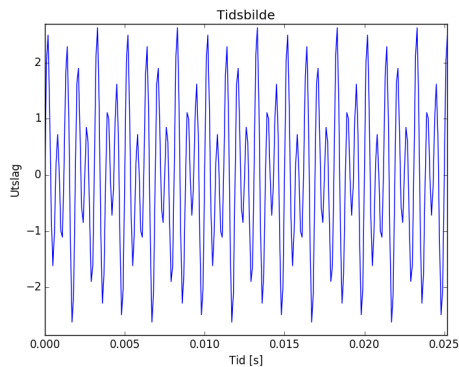
[7] <http://bootmath.com>



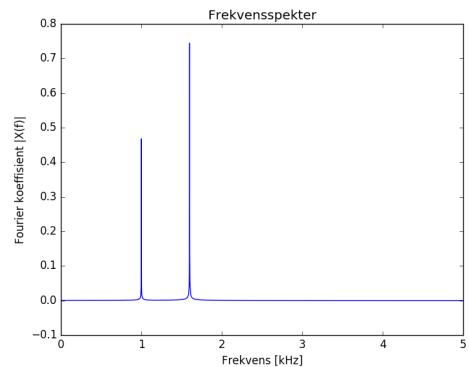
Figur 14: Til venstre er tidsbilde av en firkantbølge og til høyre er frekvensspekteret



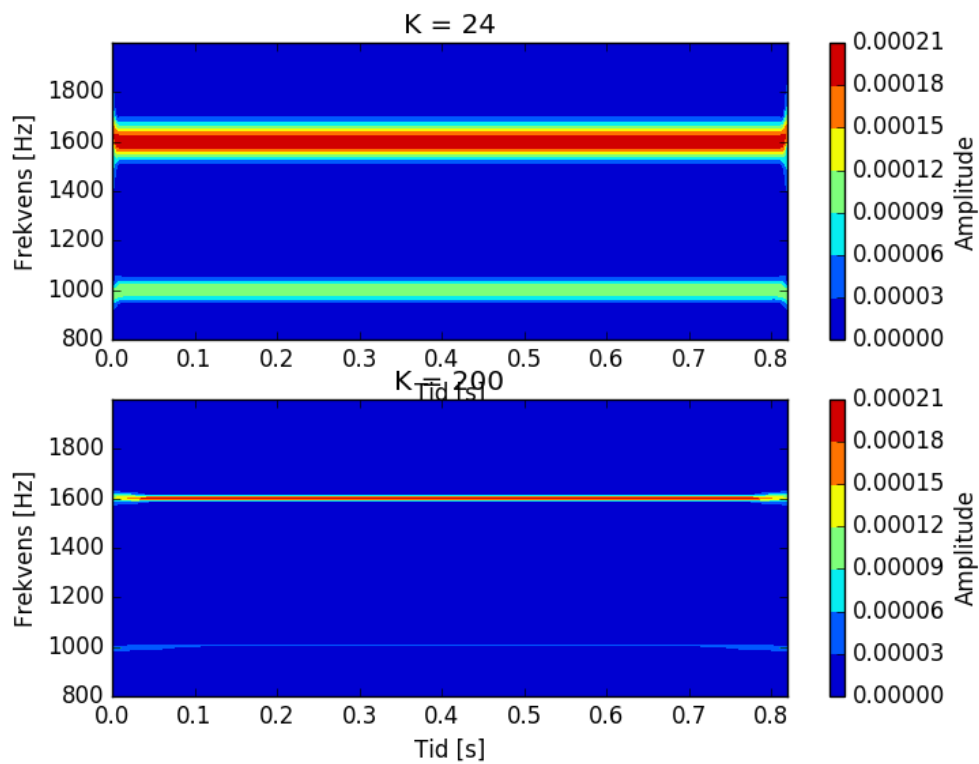
Figur 15: Analytisk løsning av amplitude av en firkant-bølge



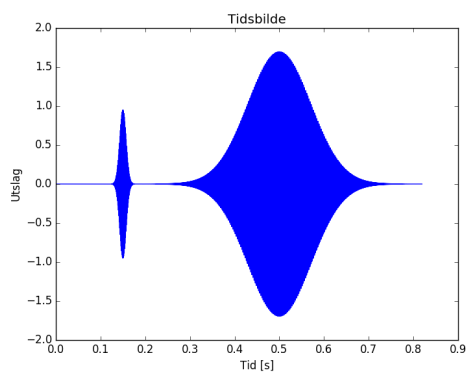
Figur 16: Zoom av tidsbildet til signalet



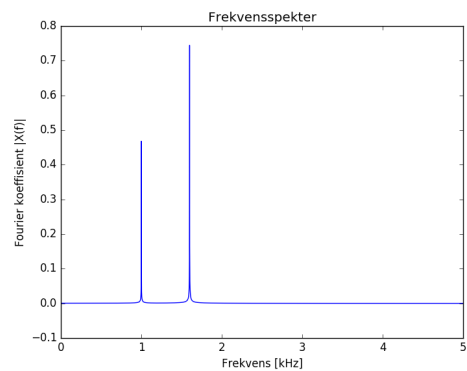
Figur 17: Frekvensspekteret til signalet



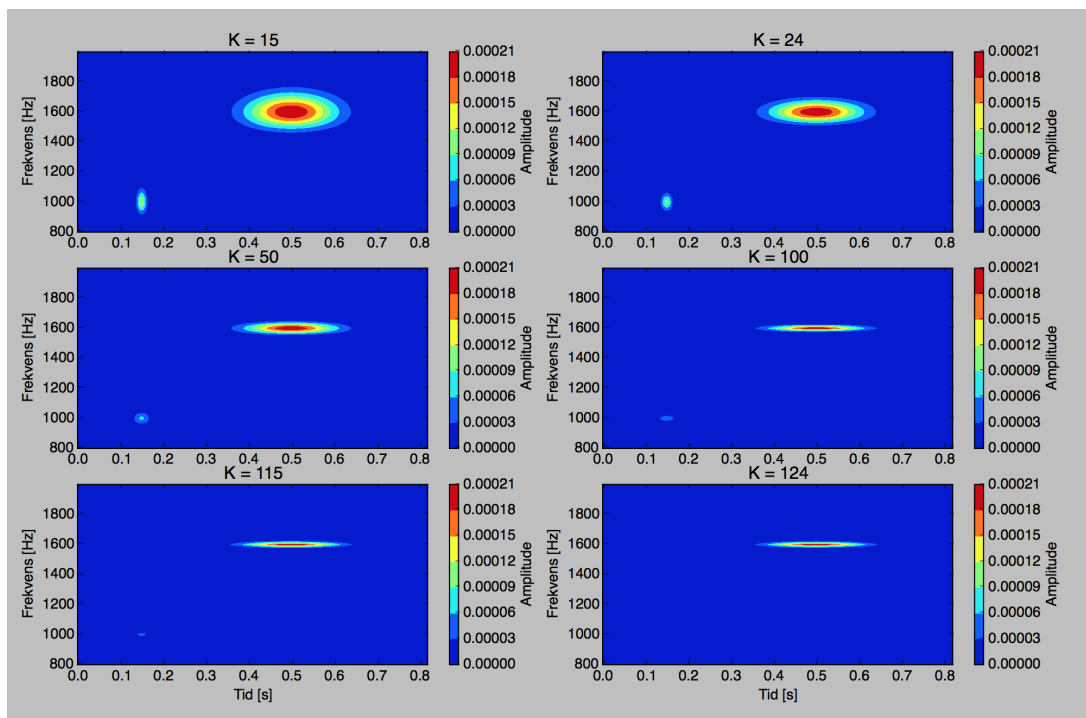
Figur 18: Øverst er wavelettransformasjonen med en bredde $K = 24$, nederst er wavelettransformasjonen med en bredde $K = 200$. Fargekartet helt til høyre i begge figurene viser amplituden. Mørk rød = mange amplituder, mørk blå = ingen amplituder.



Figur 19: Tidsbilde av signalet



Figur 20: Frekvensspekteret til signalet ved FFT



Figur 21: Waveletspektrogram med forskjellig K verdier $\in [15, 124]$

Figur 22: Kode: Wavelettransformasjon

```
1  #konstanter
2  fs = 10**4; f1 = 10**3; f2 = 1.6*10**3
3  c1 = 1.0; c2 = 1.7
4  t1 = 0.15; t2 = 0.5
5  sigma1 = 0.01; sigma2 = 0.10
6  dt = 1.0/fs
7  N = 8192.0
8  tid = np.arange(N)*dt
9  #signal
10 sig = c1*np.sin(2*np.pi*f1*tid)*np.exp(-((tid-t1)/sigma1)**2)+\
11      c2*np.sin(2*np.pi*f2*tid)*np.exp(-((tid-t2)/sigma2)**2)
12 #FFT
13 FFTsig = fft(sig)/N
14 freq_FFT = np.linspace(0,fs,N)/10.0**3
15 #wavelet
16 n = 100
17 K = [15,24,50,100,115,124]
18 omegaa = np.linspace(800,2000,n)*(2*np.pi)
19 omega = np.linspace(0,fs,N)*(2*np.pi)
20
21 for j in xrange(len(K)):
22     wave = []
23     for i in xrange(len(omegaa)):
24         solver = Wavelet(omega)
25         psi = solver(omegaa[i],K[j])
26         knall = ifft(FFTsig*psi)
27         wave.append(knall)
28
29     wave = np.array(np.abs(wave))
30     times, freq = np.meshgrid(tid,omegaa)
31     freq = freq/(2*np.pi)
```