FYS2130 regneoppgaver uke 06

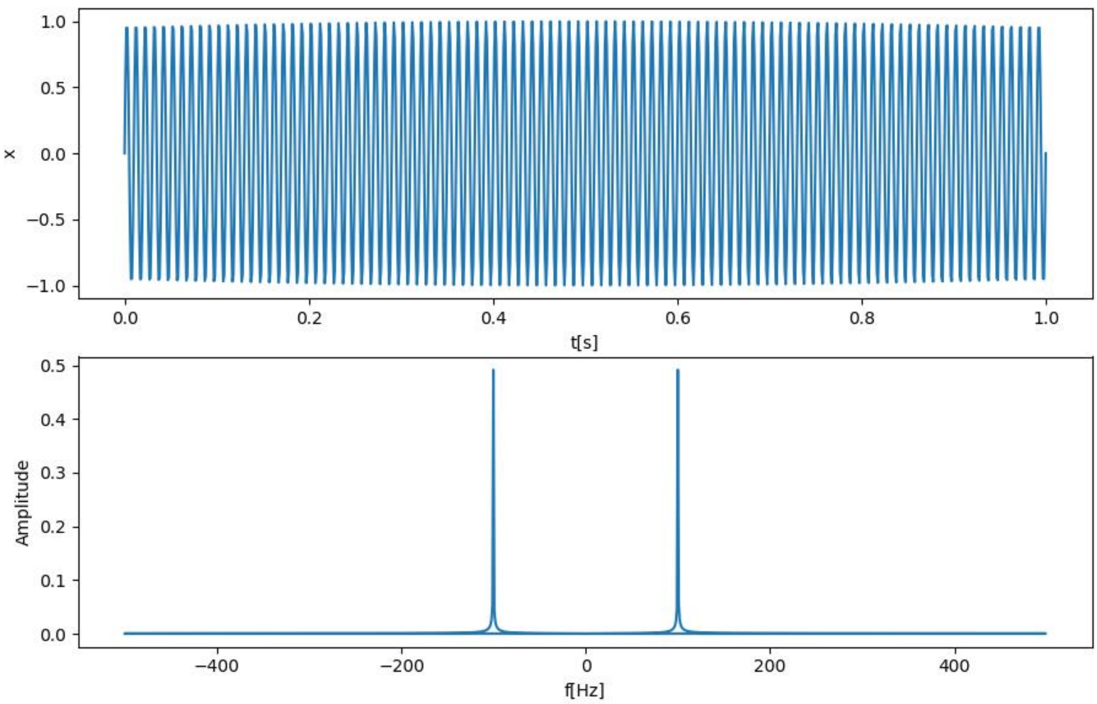
Klaudia Pawlak

klaudiap@student.matnat.uio.no

04. mars 2021

Oppgave 1. Aliasing

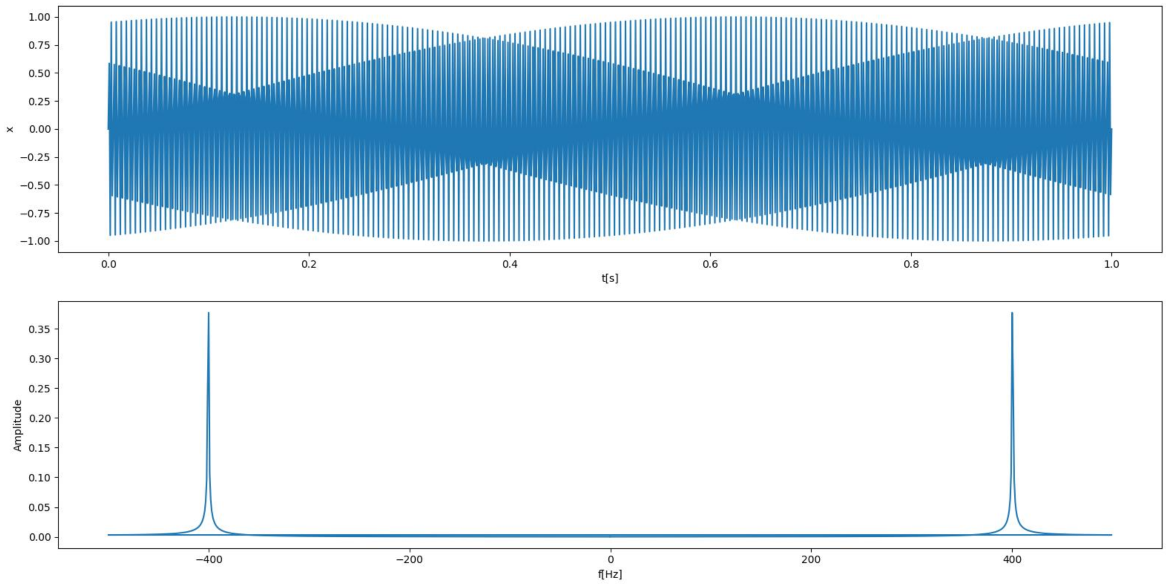
a. Vi bruker Python til å lage programmet. Vi bruker informasjon fra oppgaveteksten. Koden er vedlagt i appendiks med navnet *Kode 1a*. Kjøreeksempel gir



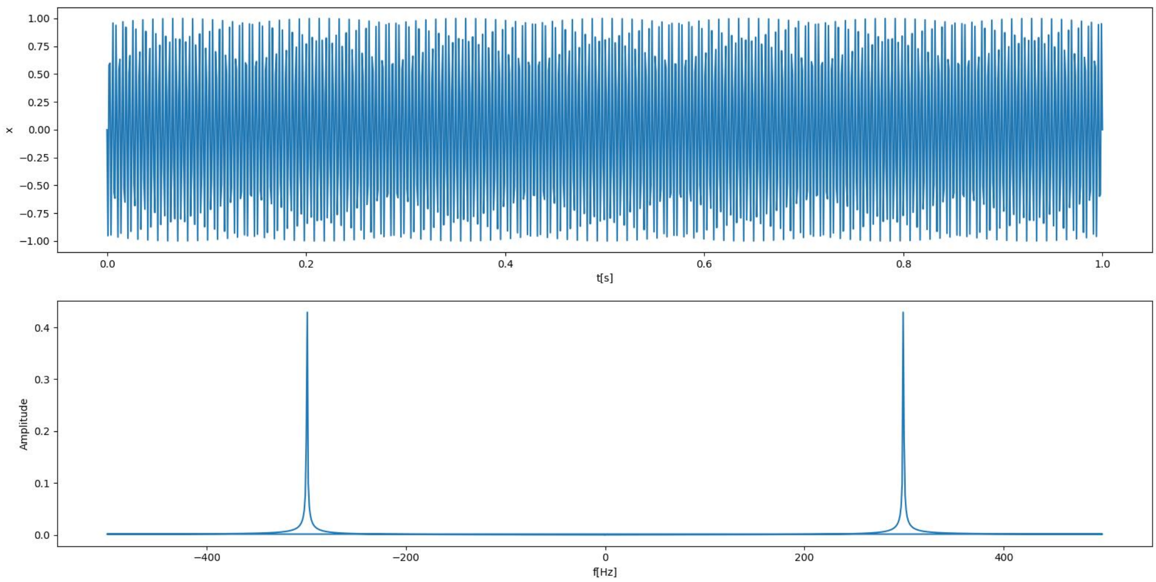
Figur 1. Kjøreeksempel av oppgave 1a

Vi ser at vi får en topp på frekvensspektret på $f = 100hz$. Amplituden er som forventet (summen av toppene fra frekvensspektret blir til amplituden).

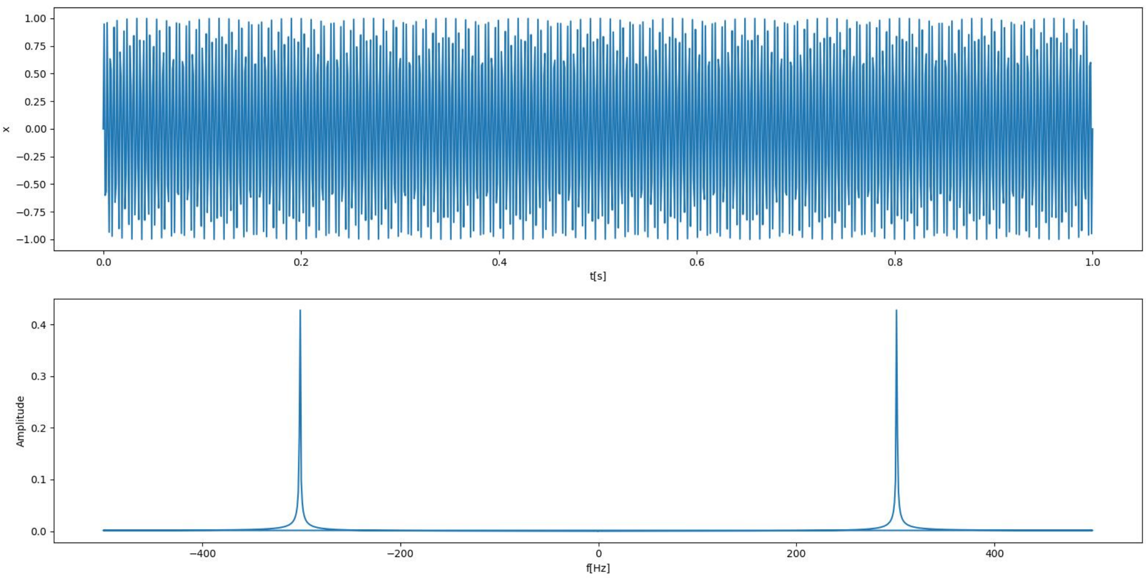
b. Vi bruker Python til å lage programmet. Vi bruker informasjon fra oppgaveteksten. Koden er vedlagt i appendiks med navnet *Kode 1b*. Kjøreeksempel gir



Figur 2. Kjøreeksempel av oppgave 1b der $f = 400hz$



Figur 3. Kjøreeksempel av oppgave 1b der $f = 700hz$



Figur 4. Kjøreeksempel av oppgave 1b der $f = 1300hz$

I tilfelle der $f = 700hz$ og $f = 1300hz$, er samplingsfrekvensen mindre enn det dobbelte av maksimale frekvensen, da har vi en Nyquist frekvens. Siden den punktprøves over tid, så får vi en foldning, som da fører til aliasing. Vi ser at i Figur 3 og i figur 4 er det vanskelig å skille ulike type signaler, dette er ikke en filfelle for der $f = 400hz$.

Oppgave 2. Første koeffisient i Fourierrekka

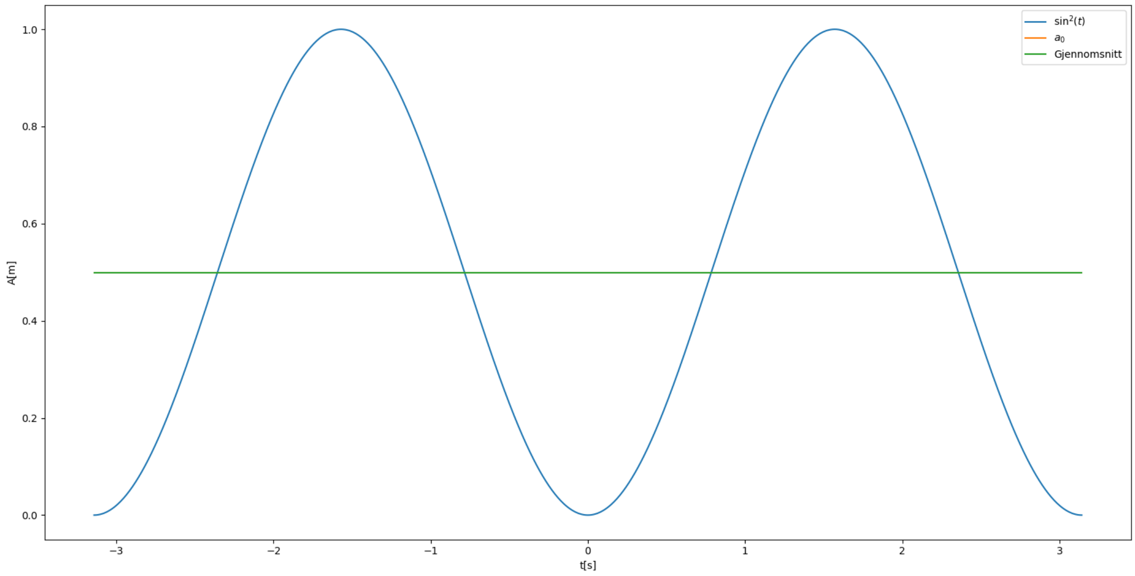
Vi skal først vise dette matematisk. Ut fra definisjon av den diskrete fouriertransformasjonen vet vi at

$$X_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi}{N}kn}$$

Vi setter at $k = 0$, da får vi

$$X_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n$$

Vi skal også vise dette ved en programmeringseksempel av funksjonen $sin^2(t)$. Vi bruker Python. Koden finner vi appendiks, og har navnet Kode 2. Kjøreeksempel gir

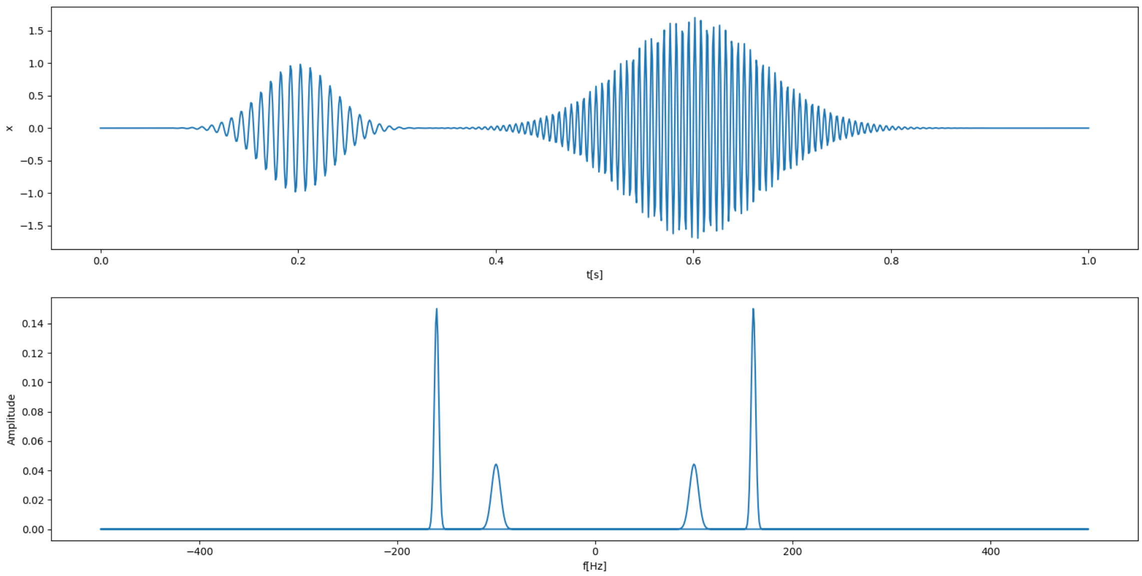


Figur 5. Kjøreeksempel av oppgave 2

Første komponenten og gjennomsnittsverdien av tidsserien har samme verdi, derfor er ikke a_0 synlig på plotet.

Oppgave 3. Waweletanalyse

a. Vi bruker Python. Vi bruker Kode 3a fra appendikset. Kjøreeksempel gir



Figur 6. Kjøreeksempel av oppgave 3a

Appendiks

Kode 1a:

```
import numpy as np
```

```
import matplotlib.pyplot as plt

def g(t, f): #Funksjon g
       return A*np.sin(2*np.pi*f*t)

A = 1.0             # [m]
f = 100             # [Hz]
f_s = 1E3           # [Hz]
T = 1               # Samplingstiden [s]

n = 1000                   # Antall sampler
dt = 1/f_s                 # Tidsintervallet
t = np.linspace(0, T, n)   # Tids array

G = g(t, f)
ft = (1/n)*np.fft.fft(G)        # Fouriertransform
ft_f = np.fft.fftfreq(n, dt)    # Frekvensene fra frekvensspektret

#Tidsserien
plt.figure()
plt.subplot(211)
plt.plot(t, G)
plt.xlabel('t[s]')
plt.ylabel('x')

#Fouriertransform
plt.subplot(212)
plt.plot(ft_f, np.abs(ft[:n]))
plt.xlabel('f[Hz]')
plt.ylabel('Amplitude')
plt.show()
```

Kode 1b:

```
import numpy as np
import matplotlib.pyplot as plt

def g(t, f): #Funksjon g
return A*np.sin(2*np.pi*f*t)

A = 1.0             # [m]
f = [400,700,1300]  # [Hz]
f_s = 1E3           # [Hz]
T = 1               # Samplingstiden [s]

n = 1000                   # Antall sampler
dt = 1/f_s                 # Tidsintervallet
t = np.linspace(0, T, n)   # Tids array
```

```

for i in f:
    G = g(t, i)
    ft = (1/n)*np.fft.fft(G)     # Fouriertransform
    ft_f = np.fft.fftfreq(n, dt)# Frekvensene fra frekvensspektret

    #Tidsserien
    plt.figure()
    plt.subplot(211)
    plt.plot(t, G)
    plt.xlabel('t[s]')
    plt.ylabel('x')

    #Fouriertransform
    plt.subplot(212)
    plt.plot(ft_f, np.abs(ft[:n]))
    plt.xlabel('f[Hz]')
    plt.ylabel('Amplitude')
    plt.show()
```

Kode 2:

```
import numpy as np
import matplotlib.pyplot as plt

def f(t):
return np.sin(t)**2

n = 1000
t = np.linspace(-np.pi, np.pi, n)

ft = (1/n)*np.fft.fft(f(t))

g = np.mean(f(t))
gt = np.ones_like(t)
gt = g*gt

a = np.abs(ft[0])
at = np.ones_like(t)
at = a*at

plt.plot(t, f(t), label='$\sin^2(t)$')
plt.plot(t, at, label='$a_0$')
plt.plot(t, gt, label='Gjennomsnitt')
plt.legend()
plt.xlabel('t[s]')
plt.ylabel('A[m]')
plt.show()
```

Kode 3a

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

#Variablene osv.
a = np.array([1,1.7])
f = np.array([100,160])
t = np.array([0.20,0.60])
a = np.array([0.05,0.10])

n = 1000
fs = 1e3
dt = 1/fs
T = 1
tt = np.linspace(0,T,n)

#Funksjon fra oppgave 1 (endret)
def f_t(tt,t,o,f,a):
    x = 0
    for i in range(2):
        x += a[i]*np.sin(2*np.pi*f[i]*tt)*np.exp(-((tt-t[i])/o[i])**2)
    return x

#Fouriertransformasjon
F = f_t(tt, t, o, f, a)
ft = (1/n)*np.fft.fft(F)
ft_f= np.fft.fftfreq(n, dt)

#Lage ploter
plt.figure()
plt.subplot(211)
plt.plot(tt, F)
plt.xlabel('t[s]')
plt.ylabel('$x$')

plt.subplot(212)
plt.plot(ft_f, np.abs(ft[:n]))
plt.xlabel('f[Hz]')
plt.ylabel('Amplitude')
plt.show()
```