



HJEMMEOPGAVE 1

34210 INTRODUCTION TO DIGITAL COMMUNICATION

Daniel Brasholt s214675

6. marts 2023

Indhold

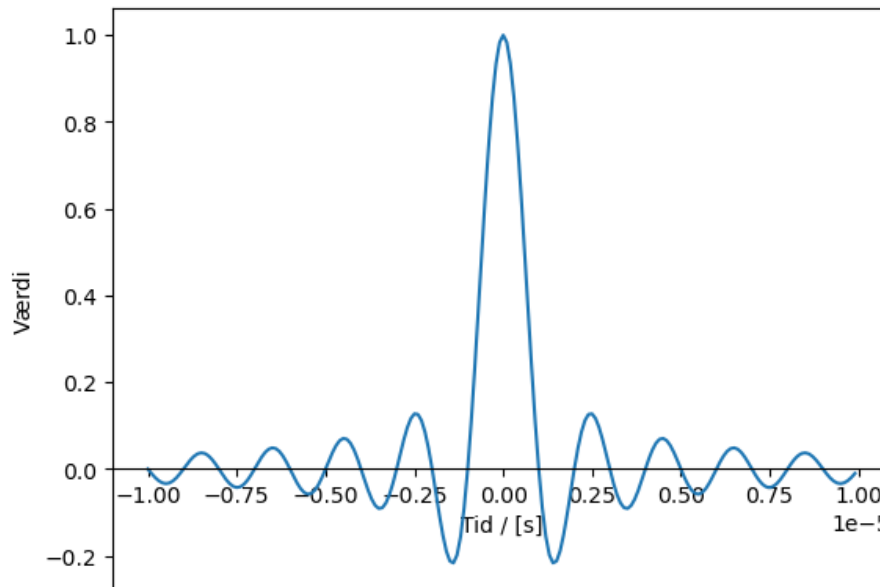
	Side
Spørgsmål 1	1
Spm 1.1 puls-signalet	1
Spm 1.2 Omordning	1
Spørgsmål 2	2
Spm 2.1 fft	2
Spm 2.2 fftshift	3
Spørgsmål 3	3
Spm 3.1 lpuls og puls	3
Spm 3.2 Sammenhængen mellem spektre	5
Spørgsmål 4	5
Spm 4.1 Energispektre	5
Spørgsmål 5	6
Spm 5.1 Energispektrum for tilfældigt signal	6

KODE

Koden kan findes som udskrift af en Jupyter-notebook nederst i dokumentet. Projektet er da udarbejdet i Python i stedet for MATLAB.

SPØRGSMÅL 1

Spm 1.1. `puls`-signalet



Figur 1: Puls-signal.

Ovenfor er `puls`-signalet plottet.

Længden af `puls` vil være 200, da den vil have samme længde som `ti` - denne går fra $-10 \cdot T$ til $10 \cdot T$ med mellemrum på $0.1T$, hvilket da giver 200 værdier. Periodelængden, $N_{puls}T_s$ er

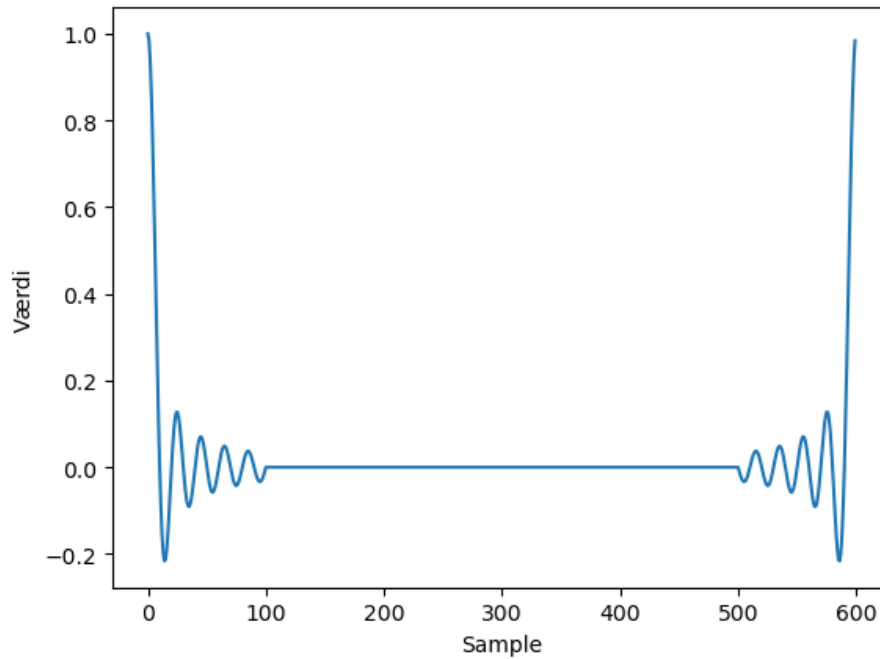
$$N_{puls} \cdot T_s = 200 \cdot 0.1\mu s = 20\mu s$$



Spm 1.2. Omordning

Max-værdien af `puls` er 1.0. For at MATLAB¹ forventer et lige signal, skal værdierne omordnes. Da vi gerne vil have en periodelængde $P = 600 \cdot T_s$, skal der indsættes 400 0'er. Disse indsættes da i midten og halvdelen af signalet flyttes til slutningen, hvilket giver nedenstående graf.

¹Jeg har selv brugt NumPy, men da funktionerne i dette hovedsageligt er lavet ud fra MATLAB, vil det fungere på samme måde.

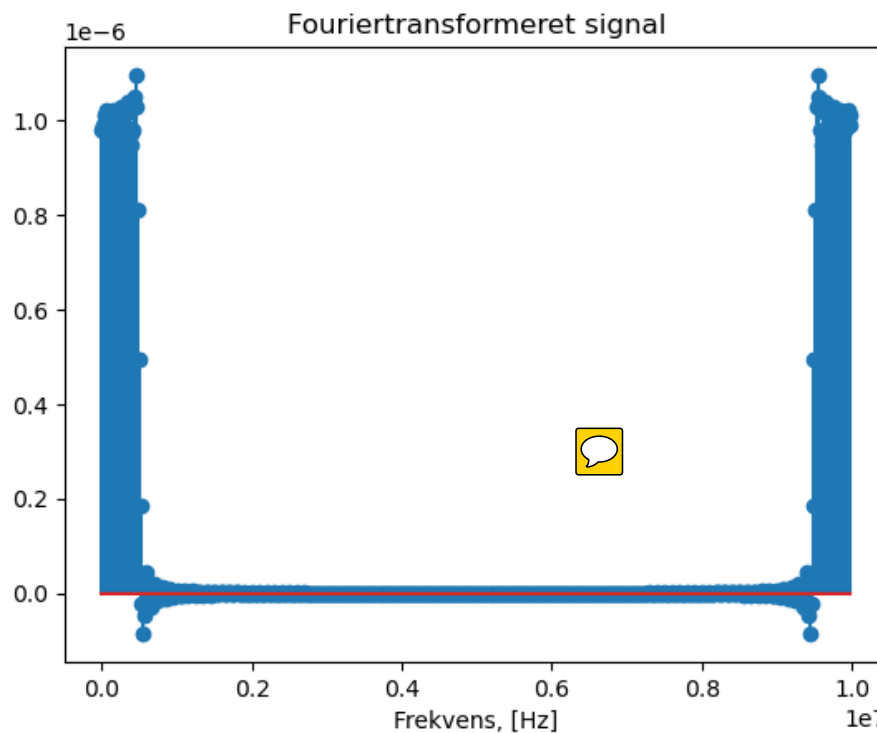


Figur 2: Omordnede pulsværdier, hvilket giver en lige graf.

SPØRGSMÅL 2

Spm 2.1. fft

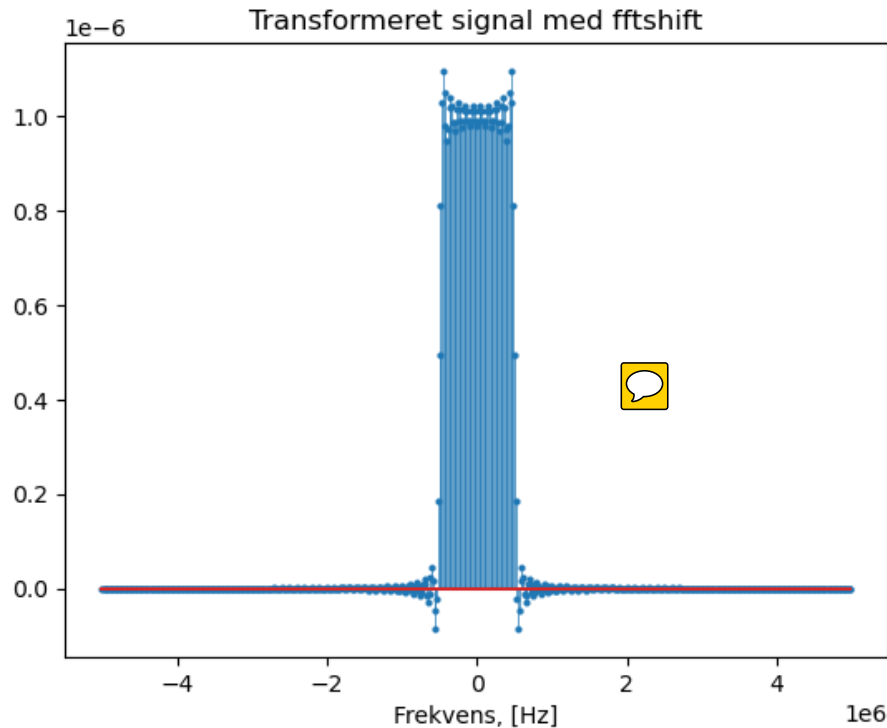
Transformationen vil dække frekvensområdet 0Hz til $1/T_s\text{Hz} = 1/0.1\mu\text{s} = 10\text{MHz}$. Afstanden mellem de enkelte værdier i spektret må være frekvensområdets størrelse delt med antallet af samples, hvilket da vil være $10\text{MHz}/600 \approx 16.7\text{kHz}$. Der deles med 600 i stedet for 200, da vi har indskudt værdier. Da bliver det udregnede spektrum dette:



Figur 3: Udregnet spektrum som funktion af frekvensen startende med 0 og sluttende med $1/T_s\text{Hz}$.

Spm 2.2. `fftshift`

Efter at have ændret frekvensaksen til at gå fra $-5MHz$ til $+5MHz$ og anvendt funktionen `fftshift`, fås følgende spektrum:



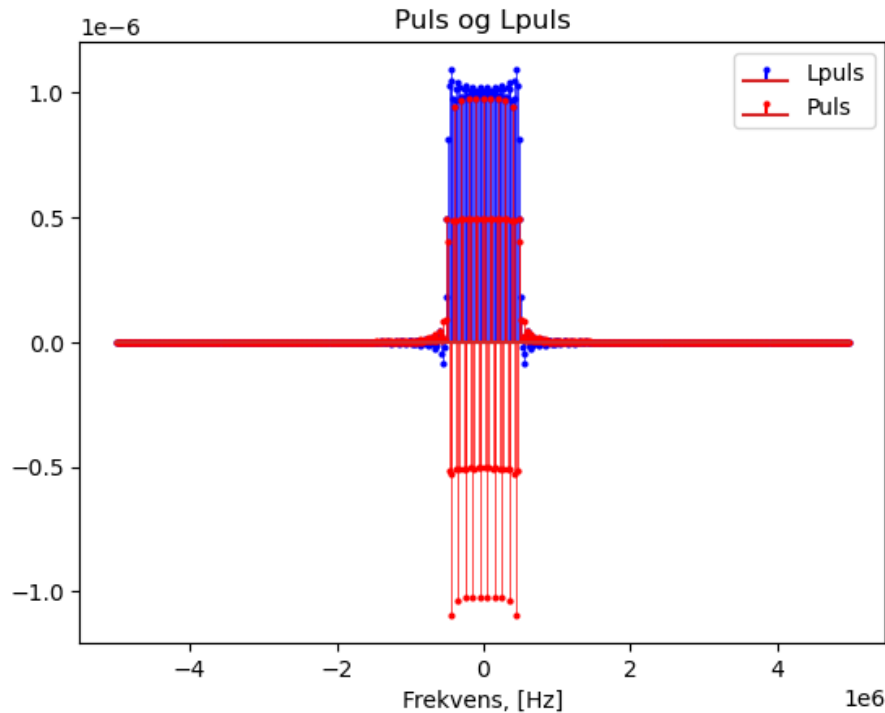
Figur 4: Spektrum med `fftshift`

Det kan dog ses på figur 4, at spektret ikke bliver udnyttet i betydelig grad så langt ud. Ligesom på noternes figur 3.4, bliver den uendeligt lange *sinc*-funktion til et endeligt langt spektrum i frekvens.

SPØRGSMÅL 3

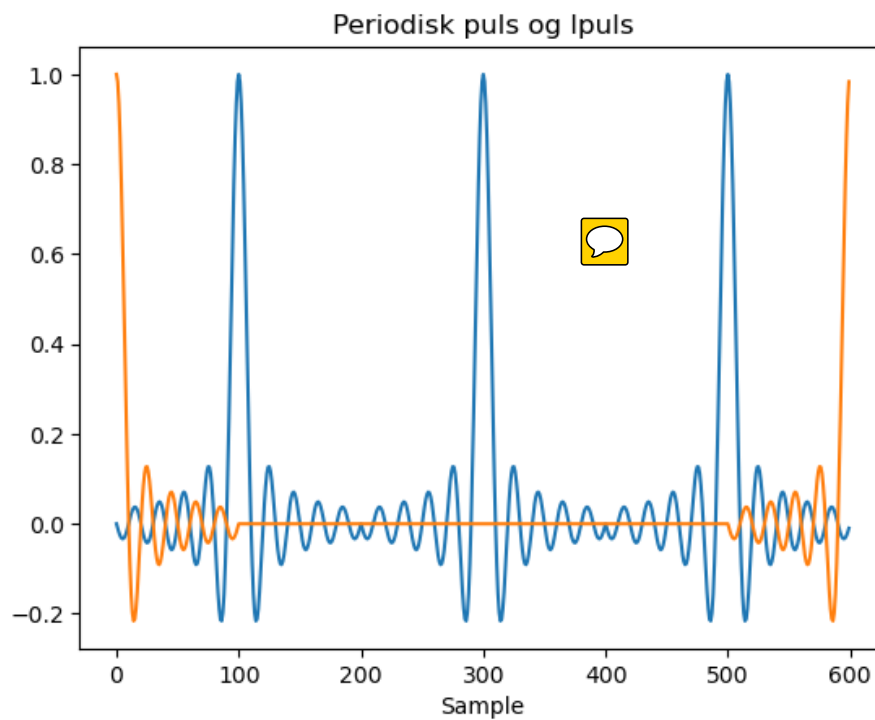
Spm 3.1. `lpuls` og `puls`

Efter at have transformeret `puls` og forlænget med den rette mængde 0'er, fås følgende spektrum (plottet med spektret fra 4):



Figur 5: Puls og Lpuls plottet sammen for at vise forskellene i spektrene.

Dernæst skulle det vises, at `puls` som periodisk signal kan ses som en forskydning af det periodiske signal for `lpuls` 100 pladser til højre. Plotter man flere perioder af disse ovenpå hinanden fås:



Figur 6: Flere perioder af puls og lpuls.

På ovenstående figur 6 kan det ses, at signalerne kan ses som de forventede forskydninger. Denne forskydning vil være 100 pladser. Da der er $T_s = 0.1\mu s$ mellem hver plads, vil det være en forskydning på $10\mu s$.



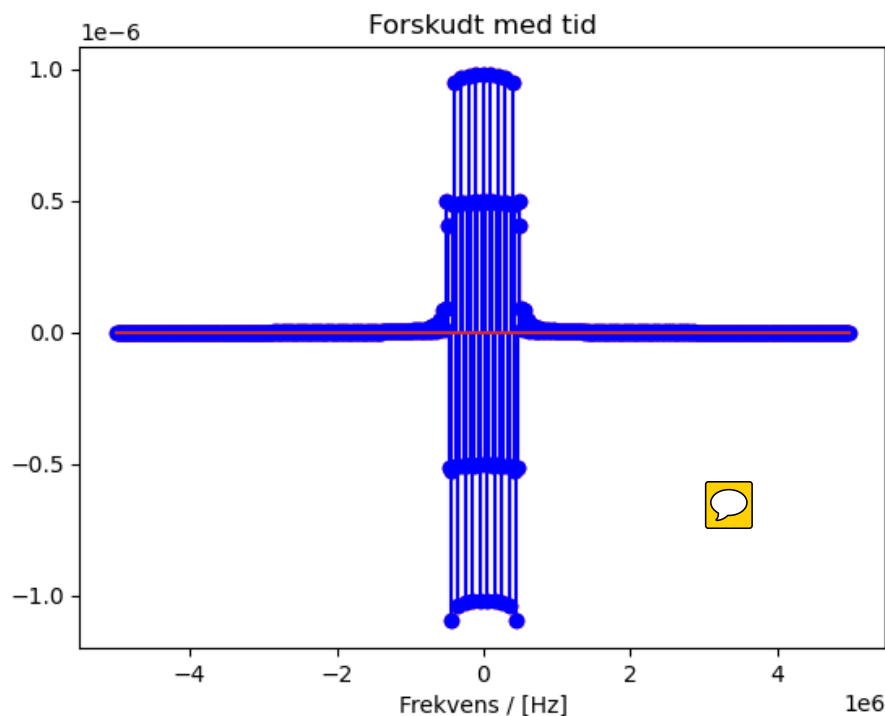
Figur 8: Energispektre for *puls* og *lpuls*, som viser, at de er ens.

Spm 3.2. Sammenhængen mellem spektre

Formlen på slide 19 siger dette om at forskyde med A:

$$W(f) = e^{-j2\pi f A T_s} V(f)$$

I dette tilfælde er $A = 100$. Anvendes denne formel på *Lpuls*, kan resultatet lægges oveni *Puls*, hvilket giver følgende:



Figur 7: Formel anvendt til at forskyde med tid.

Her kan det ses, at der er perfekt overlap mellem spektrene. Dette viser, at der er den forventede sammenhæng (der ligger også en rød graf bag den blå, hvilket fremgår af bilagene).

SPØRGSMÅL 4

Spm 4.1. Energispektre

Det kan ses på ovenstående figur 8, at de to energispetre er ens. Dette kan forklares ved at tidsforskydningen er uden betydning. Sætningen observeres:

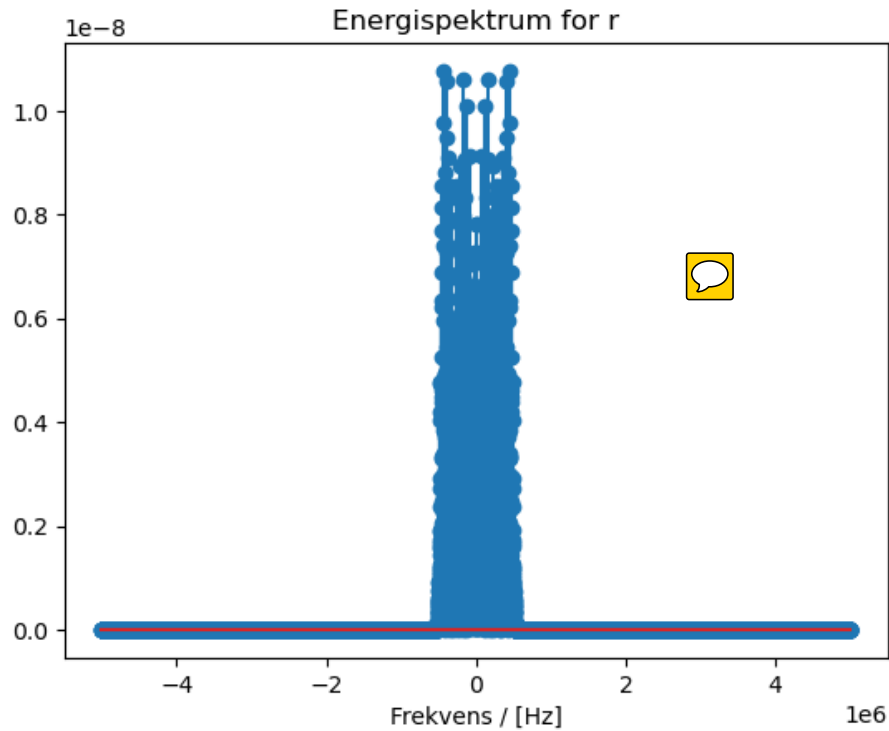
$$S_w(f) = |W(f)|^2 = |e^{-j2\pi f A} V(f)|^2 = |e^{-j2\pi f A}|^2 |V(f)|^2 = S_v(f)$$

Her kan man nemlig se, at amplituden af de komplekse tal skrevet på denne form vil være 1, hvorfor denne bortfalder og giver det samme energispektrum på trods af tidsforskydningen.

SPØRGSMÅL 5

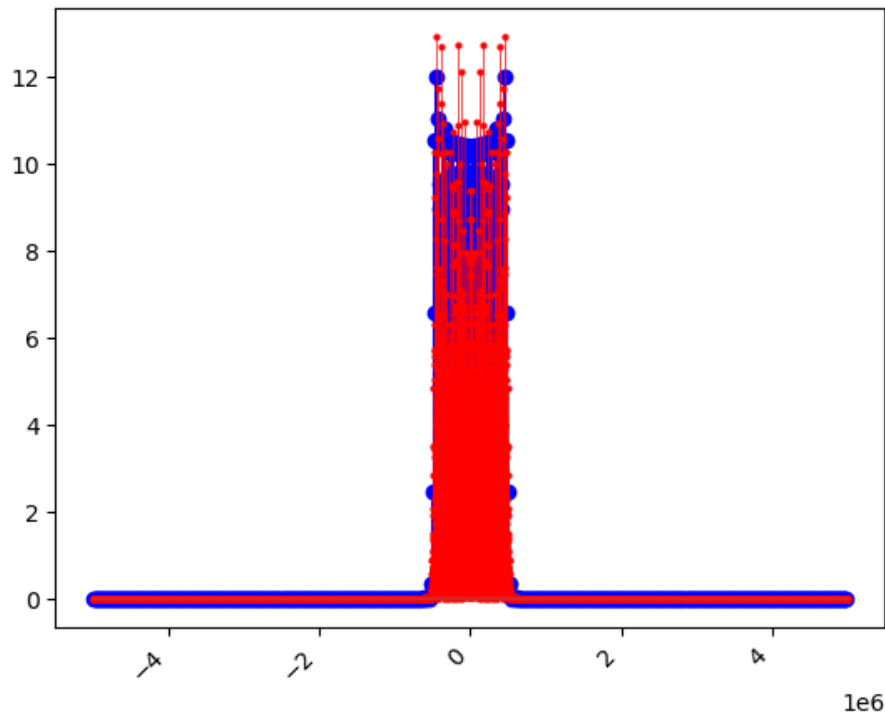
Spm 5.1. Energispektrum for tilfældigt signal

Efter den tilfældige sekvens er udregnet og anbragt, kan signalet udrenses. Energispektret for dette ser da således ud:



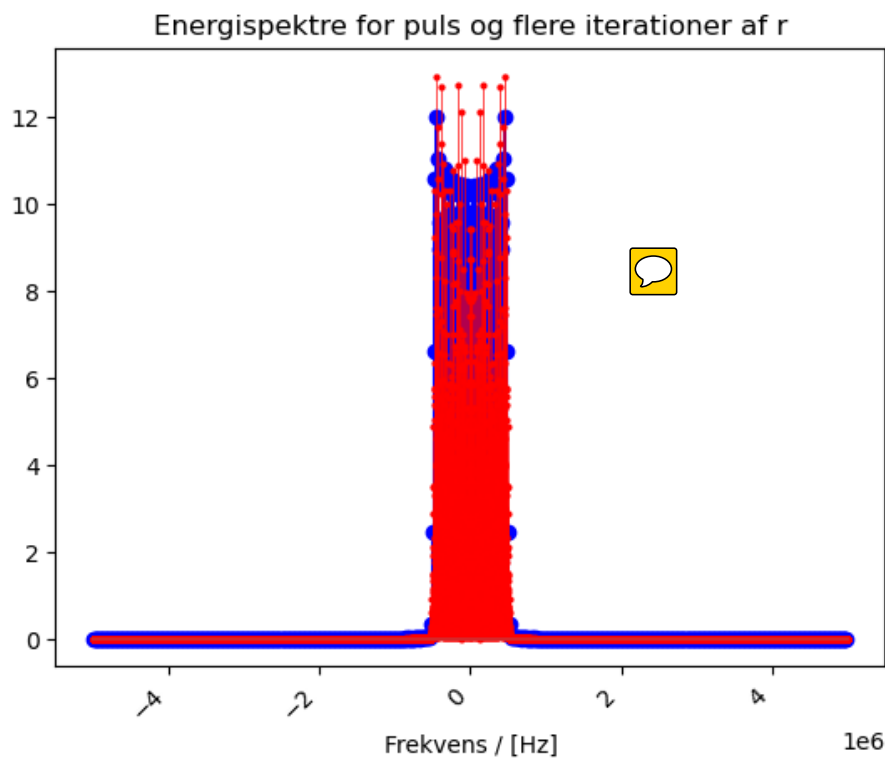
Figur 9: Energispektrum for r efter $fftshift$ er anvendt.

Siden puls-signalet er blevet brugt til at danne dette signal, må frekvensområdet igen være $1/T_s = 10MHz$. Dog vil opløsningen være $10MHz/20000 = 500Hz$, da der nu er flere værdier i spektret.



Figur 10: Energispektrum for Puls og r sammen (ganget med en konstant så de kan sammenlignes).

Herefter blev forsøget kørt 500 gange, hvilket gav følgende:



Figur 11: Flere iterationer plottet sammen med energispektret for puls

Figur 11 ligner meget figur 10, hvilket enten kan skyldes at der ikke er store variationer eller at der er en fejl i koden.

Hjemmeopgave 1

March 6, 2023

1 Imports

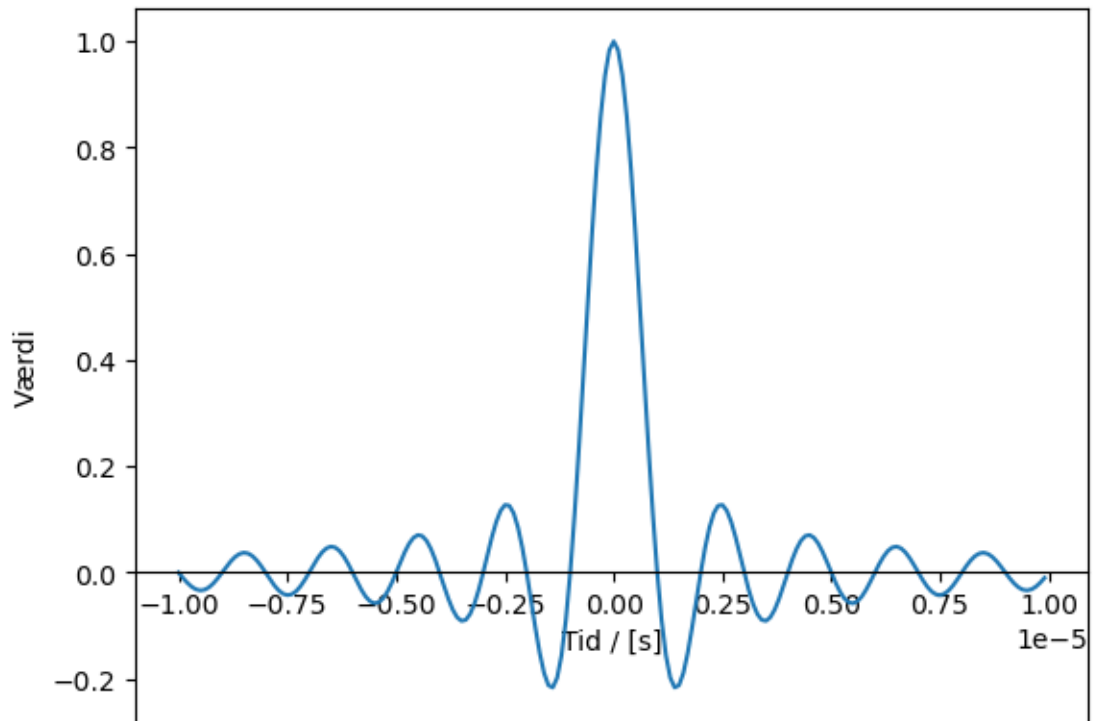
```
[3]: import numpy as np
from matplotlib import pyplot as plt
import random
```

2 Spørgsmål 1

2.1 1.

```
[43]: T = 1e-6
Ts = 0.1e-6
ti = np.arange(-10*T, 10*T, Ts)
puls = np.sinc(ti/T)
```

```
[44]: fig, ax = plt.subplots()
ax.plot(ti, puls)
plt.xlabel('Tid / [s]')
plt.ylabel('Værdi')
ax.spines['bottom'].set_position('zero')
```



```
[45]: Npuls = len(puls)
print(f'{Npuls=}')
print(f'{len(ti)=}')
print(f'{Npuls*Ts=}')

```

```
Npuls=200
len(ti)=200
Npuls*Ts=1.9999999999999998e-05

```

2.2 2.

```
[46]: print(f'{max(puls)=}')
```

```
max(puls)=1.0

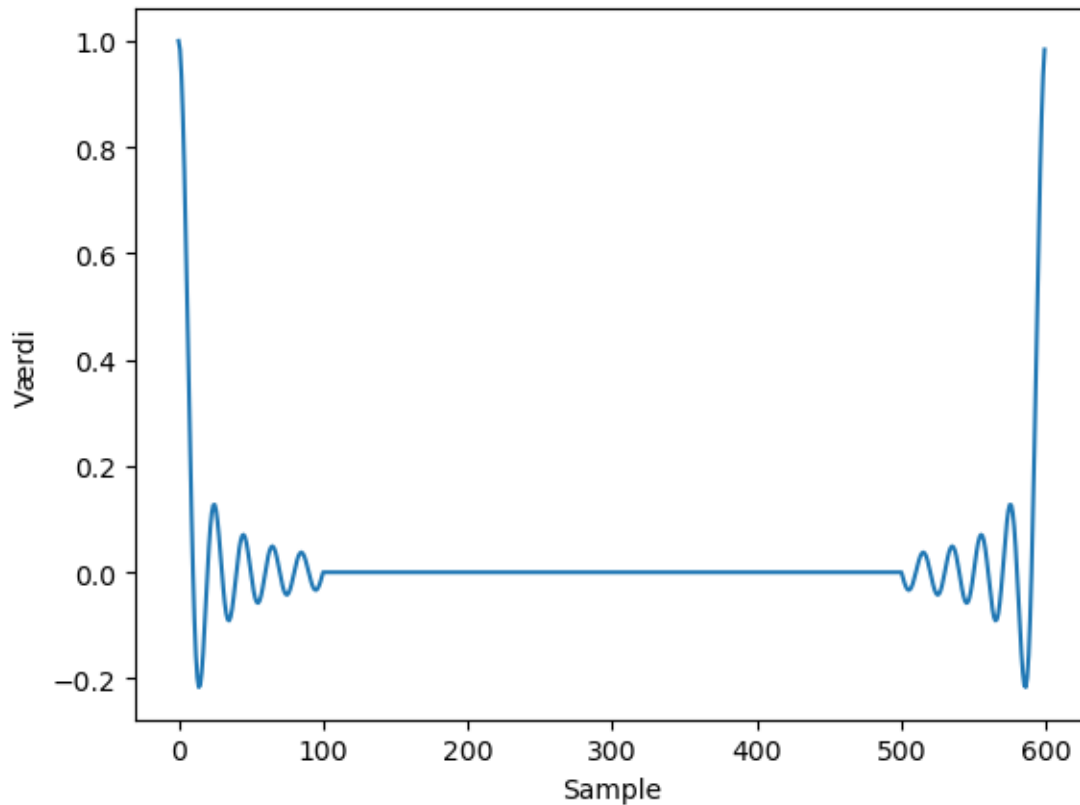
```

```
[47]: A = int(Npuls/2)
lpuls = np.concatenate((puls[A:], np.zeros(400), puls[:A]))

```

```
[48]: plt.plot(lpuls)
plt.xlabel('Sample')
plt.ylabel('Værdi')
plt.show()

```



3 Spørgsmål 2

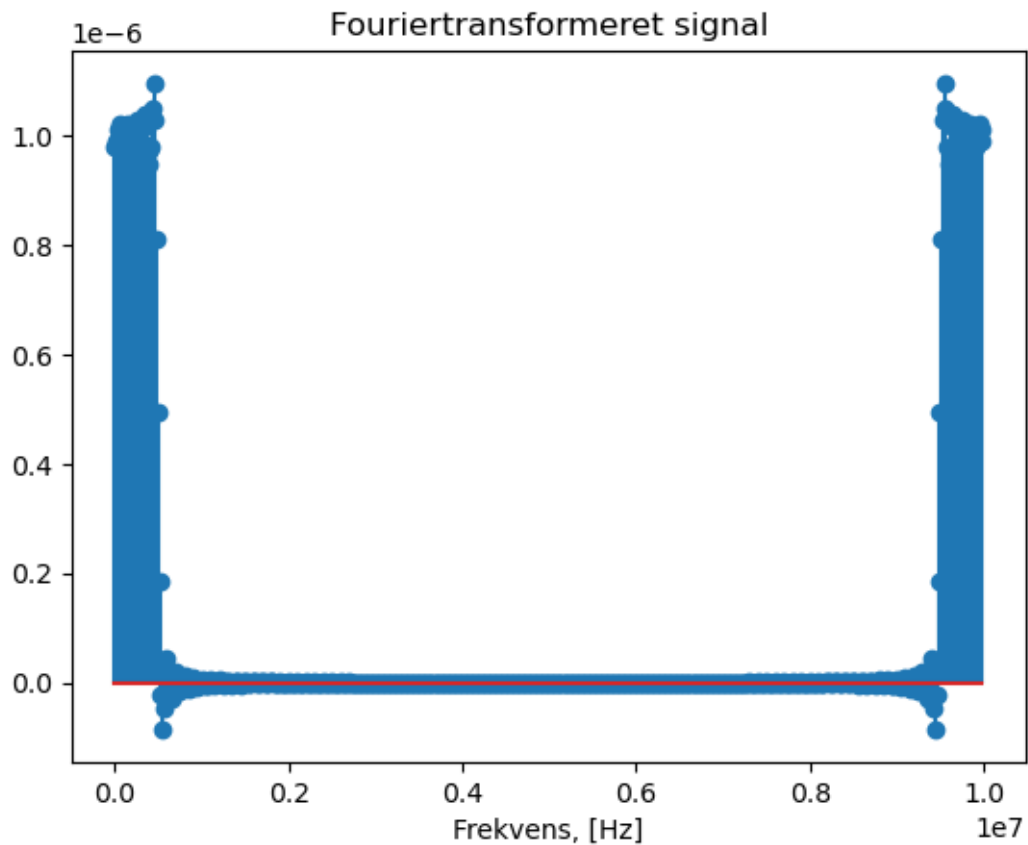
3.1 1.

```
[49]: Lpuls = np.real(np.fft.fft(lpuls) * Ts)
```

```
[50]: print(f'Sampling frequency: {1/Ts}Hz')
      print(f'Resolution: {1/Ts / len(Lpuls)}Hz')
```

```
Sampling frequency: 10000000.0Hz
Resolution: 16666.666666666668Hz
```

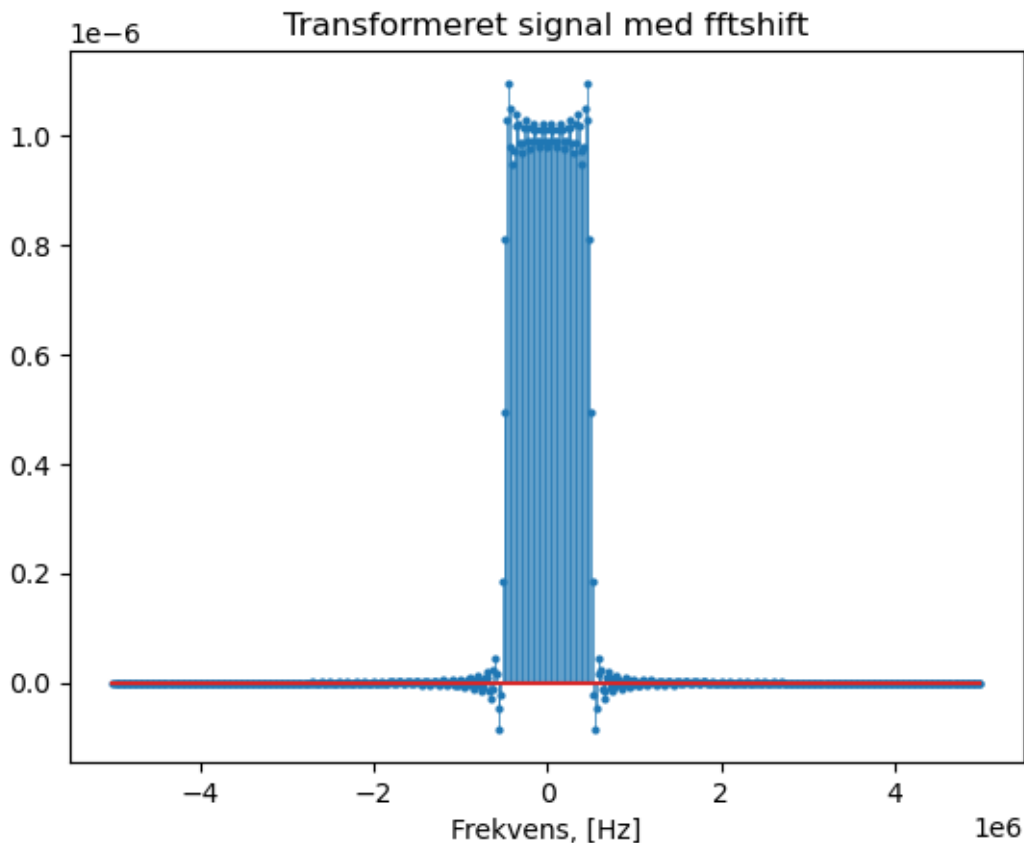
```
[51]: f = np.arange(0, 600) / (600*Ts)
      plt.stem(f, Lpuls)
      plt.xlabel('Frekvens, [Hz]')
      plt.title('Fouriertransformeret signal')
      plt.show()
```



3.2 2.

```
[52]: Lpuls_shifted = np.fft.fftshift(Lpuls)
```

```
[53]: f = np.arange(-300, 300) / (600*Ts)
markers, stems, base = plt.stem(f, Lpuls_shifted)
plt.setp(stems, 'linewidth', 0.5)
plt.setp(markers, markersize=2)
plt.xlabel('Frekvens, [Hz]')
plt.title('Transformeret signal med fftshift')
plt.show()
```

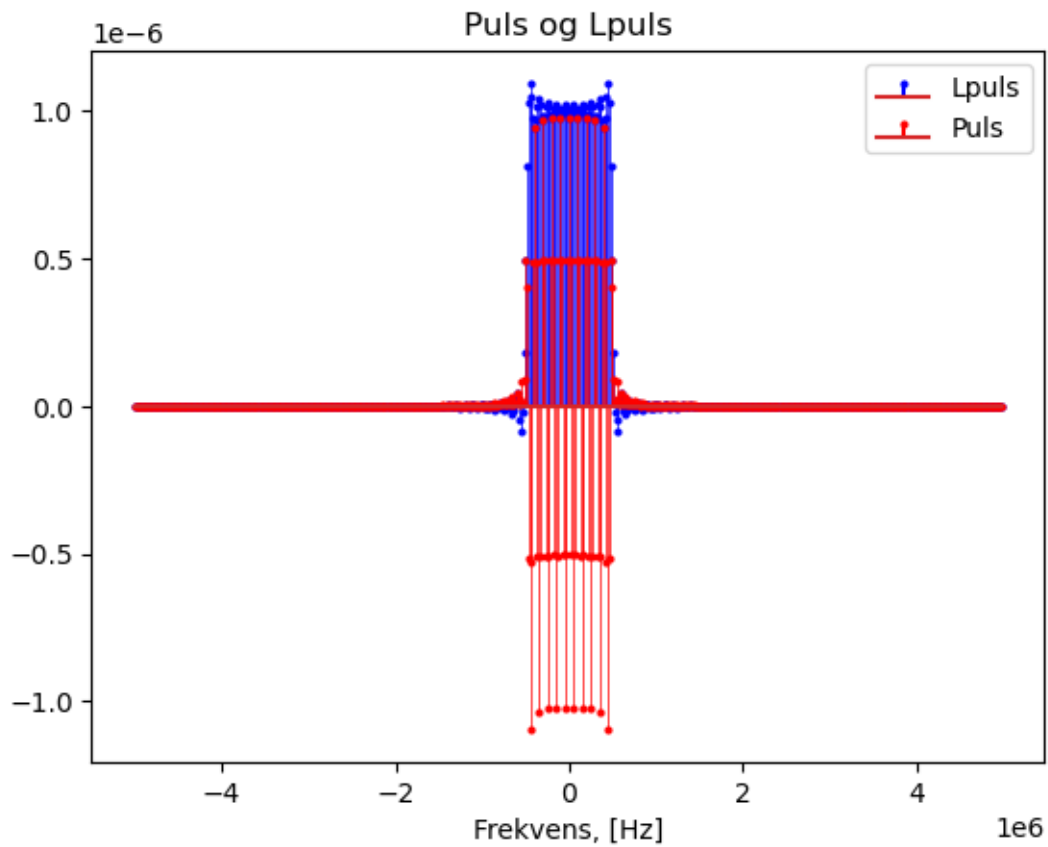


4 Spørgsmål 3

4.1 1.

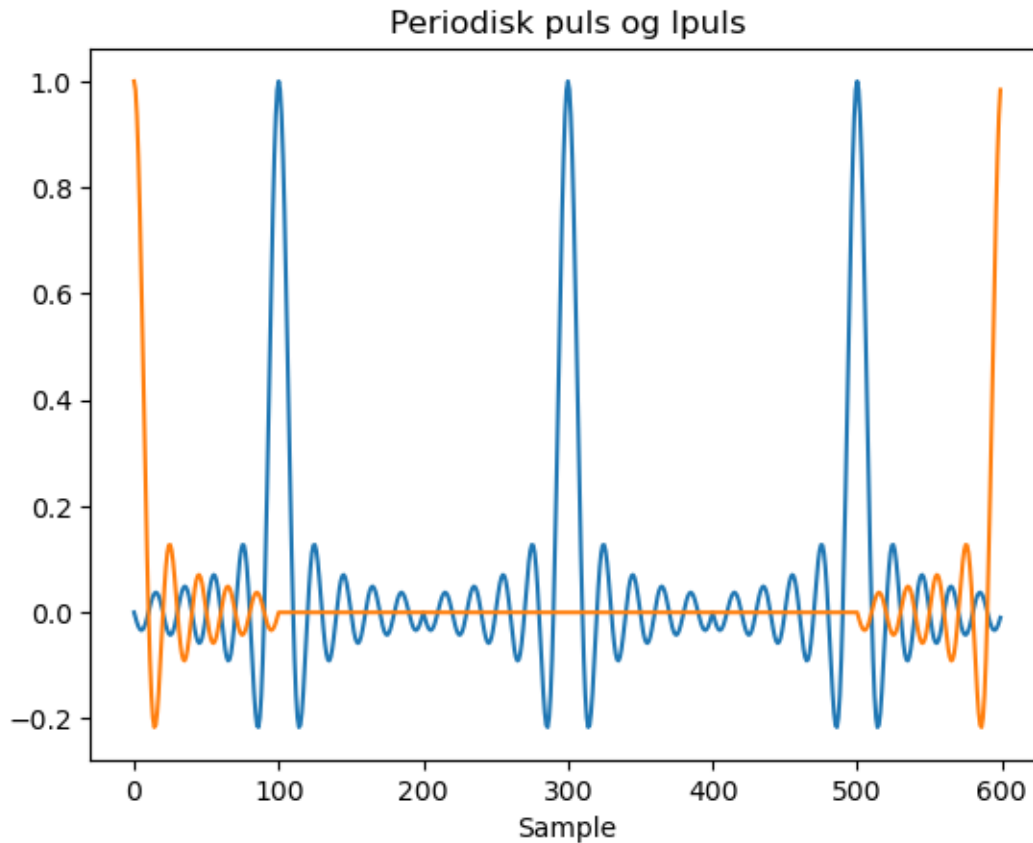
```
[54]: Puls = Ts * np.fft.fft(puls, 600)
```

```
[55]: f = np.arange(-300, 300) / (600*Ts)
markers, stems, base = plt.stem(f, Lpuls_shifted, label='Lpuls', linefmt='blue')
markers2, stems2, base2 = plt.stem(f, np.real(np.fft.fftshift(Puls)),
    ↪ linefmt='red', label='Puls')
plt.setp(stems, 'linewidth', 0.5)
plt.setp(markers, markersize=2)
plt.setp(stems2, 'linewidth', 0.5)
plt.setp(markers2, markersize=2)
plt.xlabel('Frekvens, [Hz]')
plt.title('Puls og Lpuls')
plt.legend()
plt.show()
```



4.2 2.

```
[56]: plt.plot(np.tile(puls, 3))
plt.plot(lpuls)
plt.xlabel('Sample')
plt.title('Periodisk puls og lpuls')
plt.show()
```



```
[57]: m = np.arange(0,600)
after_shift = np.exp(-1j*2*np.pi*m/(600*Ts)*A*Ts) * Lpuls
```

```
[58]: f = np.arange(-300, 300) / (600*Ts)
plt.stem(f,np.real(np.fft.fftshift(after_shift)), linefmt='red')
plt.stem(f,np.fft.fftshift(Puls), linefmt='blue')
plt.xlabel('Frekvens / [Hz]')
plt.title('Forskudt med tid')
plt.show()
```

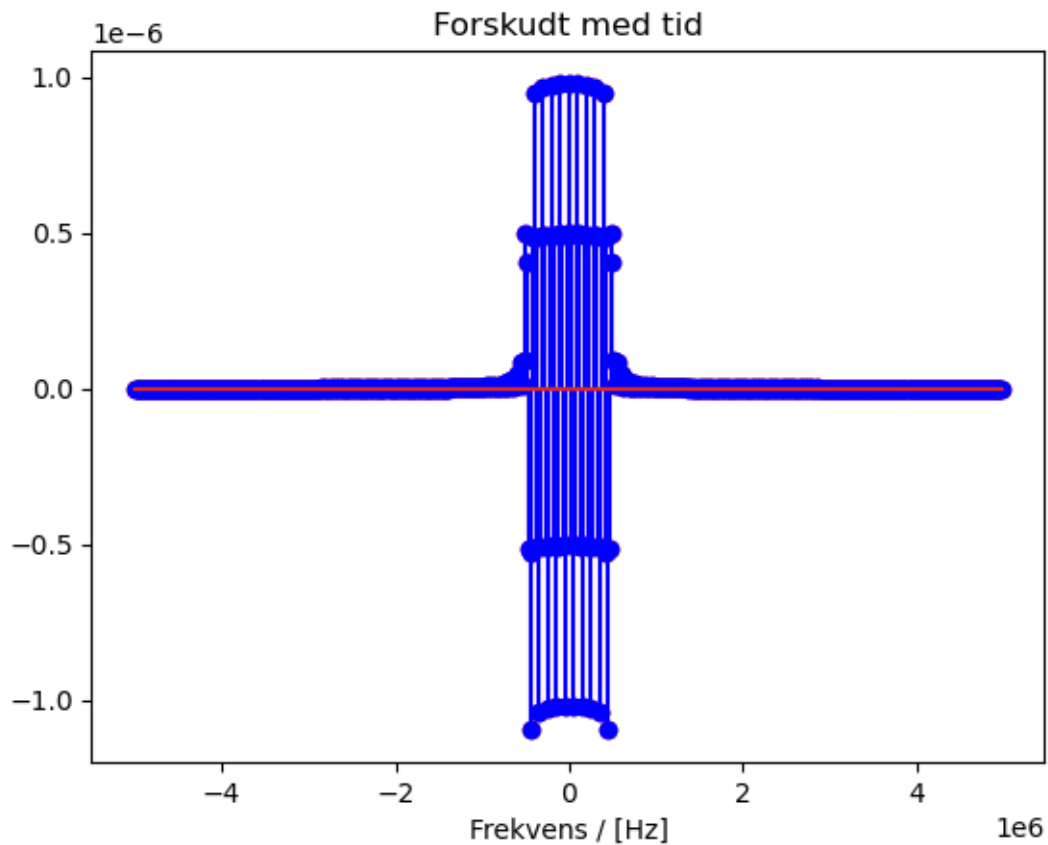
```
/usr/lib/python3.10/site-packages/numpy/ma/core.py:3371: ComplexWarning: Casting
complex values to real discards the imaginary part
```

```
_data[indx] = dval
```

```
/usr/lib/python3.10/site-packages/matplotlib/cbook/__init__.py:1369:
```

```
ComplexWarning: Casting complex values to real discards the imaginary part
```

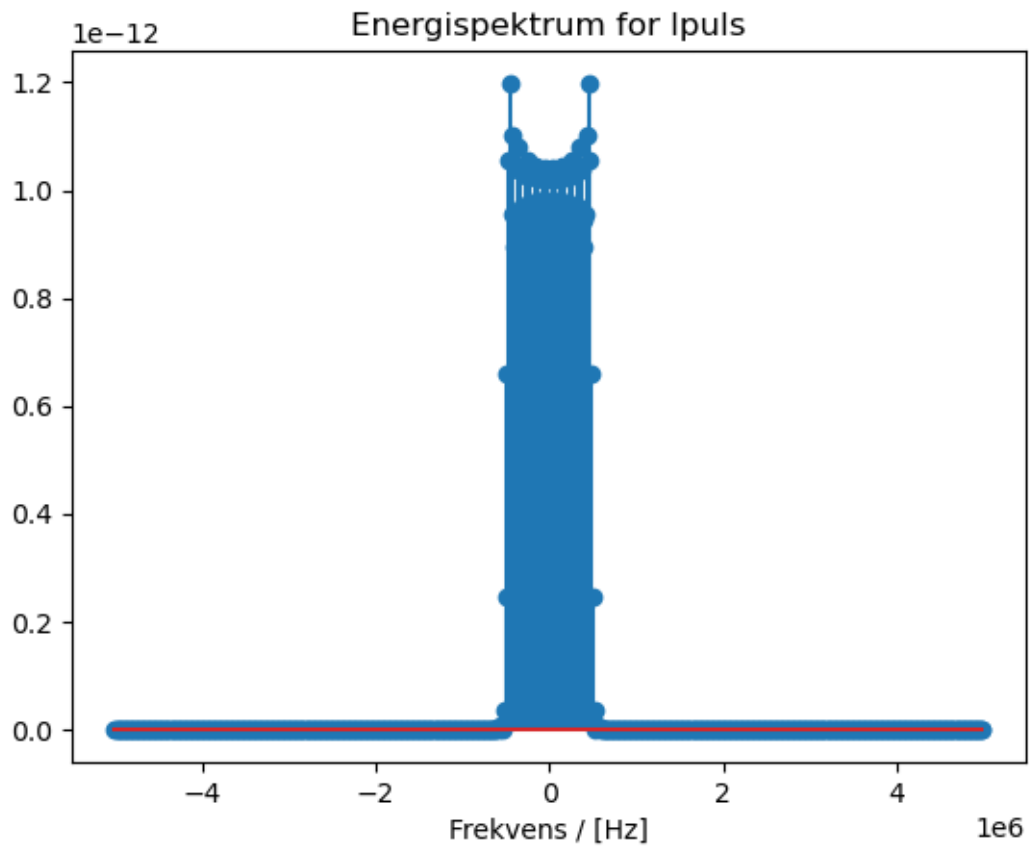
```
return np.asarray(x, float)
```



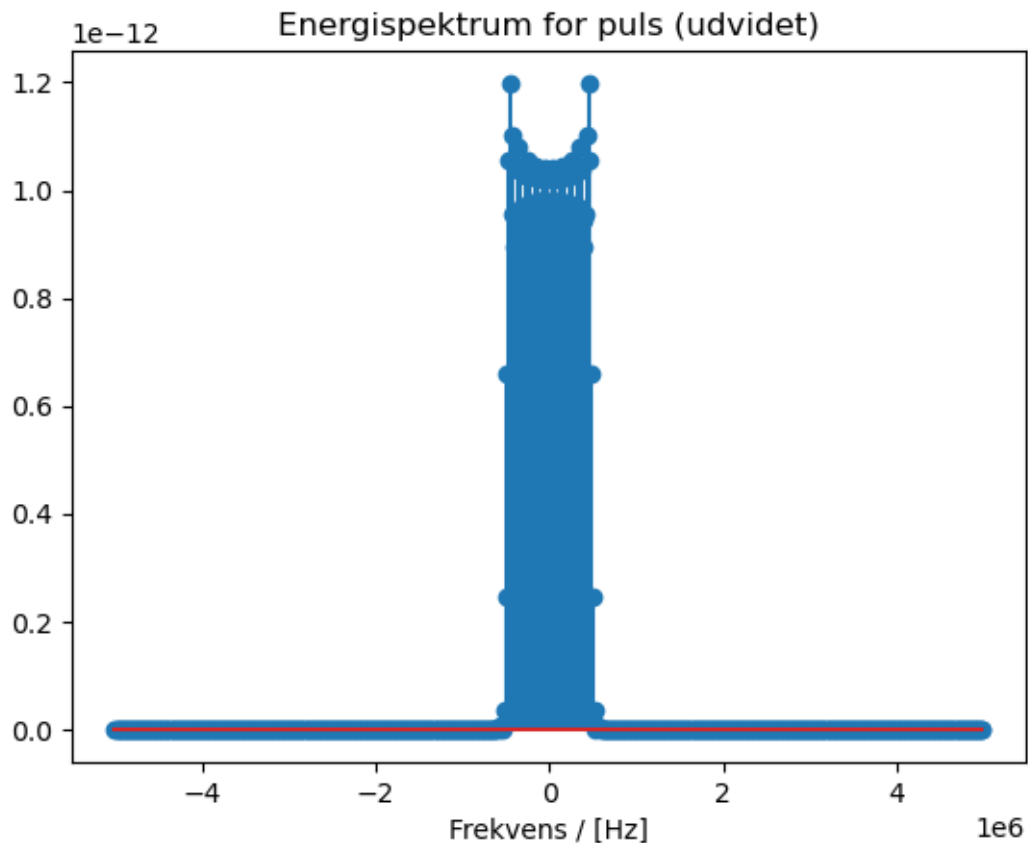
5 Spørgsmål 4

5.1 1.

```
[59]: lpuls_energy = np.abs(lpuls) ** 2
plt.stem(f, np.fft.fftshift(lpuls_energy))
plt.xlabel('Frekvens / [Hz]')
plt.title('Energispektrum for lpuls')
plt.show()
```

```
[60]: puls_energy = np.abs(Puls) ** 2
plt.stem(f, np.fft.fftshift(puls_energy))
plt.xlabel('Frekvens / [Hz]')
plt.title('Energispektrum for puls (udvidet)')
plt.show()
```



6 Spørgsmål 5

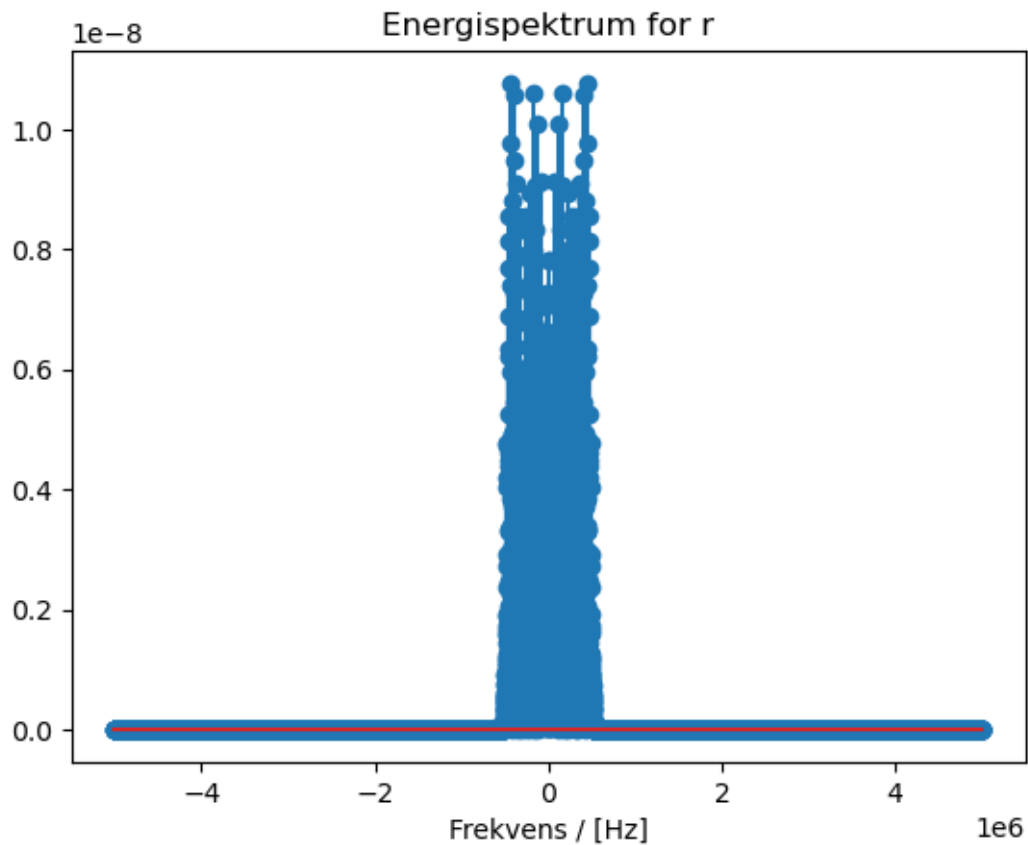
```
[119]: N = 20000
a = np.zeros(20000)
a[1::10] = [2*random.randint(0,1)-1 for i in range(2000)]
```

```
[120]: r = np.convolve(puls,a)
print(f'{r.shape=}')
r = r[100:-99]
print(f'{r.shape=}')

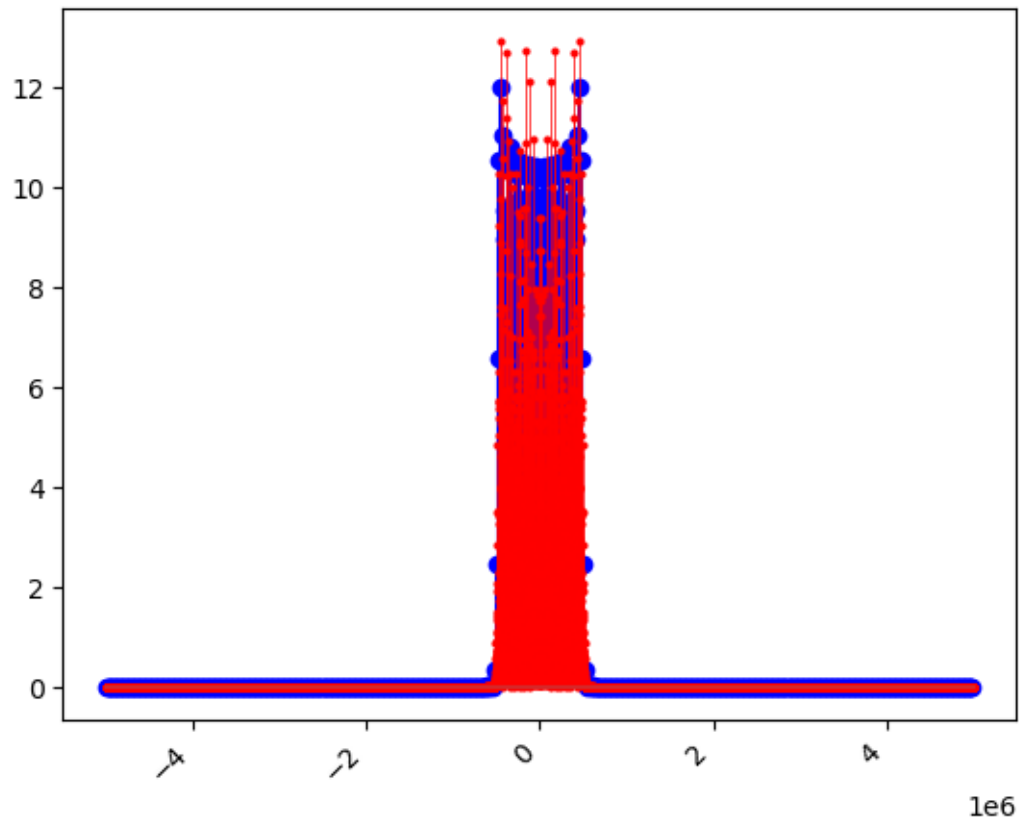
r.shape=(20199,)
r.shape=(20000,)
```

```
[121]: f = np.arange(-N/2, N/2) / (N*Ts)
r_fft = np.fft.fft(r) * Ts
r_energy = np.abs(r_fft)**2
```

```
[123]: plt.stem(f,np.fft.fftshift(r_energy))
plt.xlabel('Frekvens / [Hz]')
plt.title('Energispektrum for r')
plt.show()
```



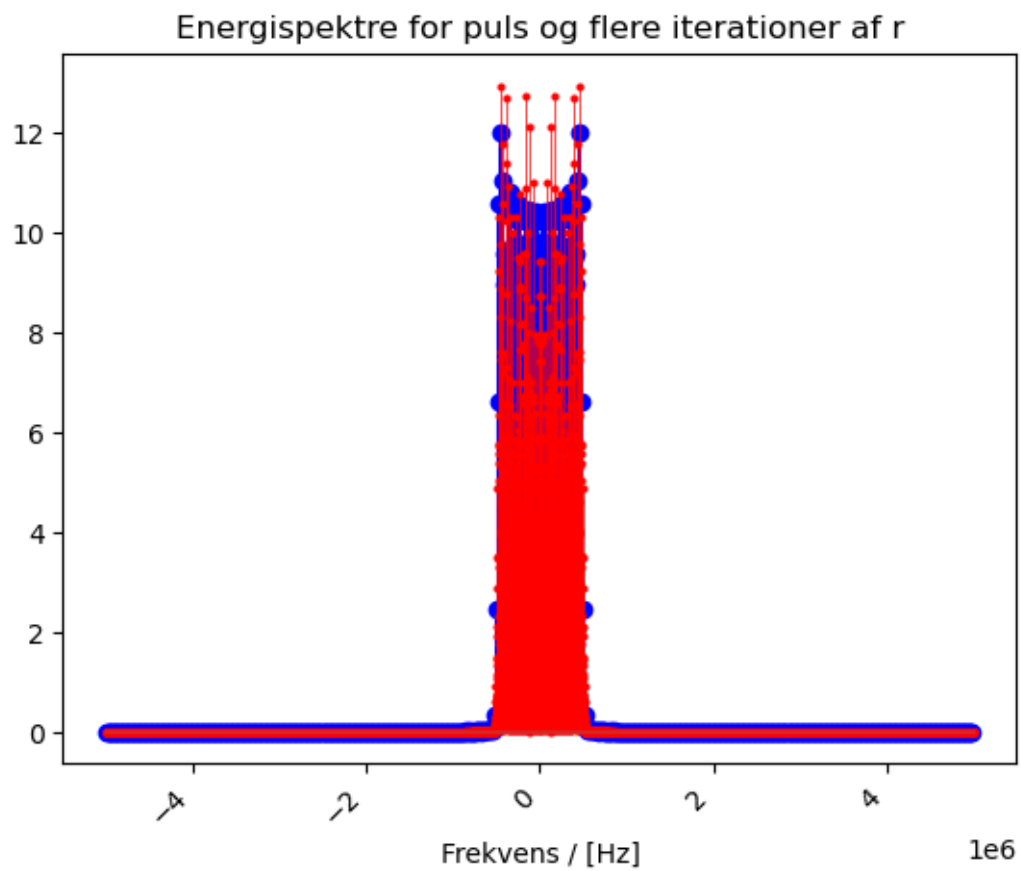
```
[129]: f2 = np.arange(-300, 300) / (600*Ts)
puls_energy = np.abs(Puls) ** 2
plt.stem(f2, np.fft.fftshift(puls_energy)*1e13, linefmt='blue')
markers, lines, base = plt.stem(f, np.fft.fftshift(r_energy)*1.2e9,
↪ linefmt='red')
plt.setp(lines, linewidth=0.5)
plt.setp(markers, markersize=2)
plt.xticks(rotation=45, ha='right')
plt.show()
```



```
[136]: N = 20000
f = np.arange(-N/2, N/2) / (N*Ts)
spectrum = np.zeros(20000)
for i in range(500):
    a = np.zeros(20000)
    a[1::10] = [2*random.randint(0,1)-1 for i in range(2000)]
    r_fft = np.fft.fft(r) * Ts
    r_energy = np.abs(r_fft)**2
    spectrum += r_energy
spectrum = spectrum / 500
```

```
[138]: f2 = np.arange(-300, 300) / (600*Ts)
puls_energy = np.abs(Puls) ** 2
plt.stem(f2, np.fft.fftshift(puls_energy)*1e13, linefmt='blue')
markers, lines, base = plt.stem(f, np.fft.fftshift(spectrum)*1.2e9,
    linefmt='red')
plt.setp(lines, linewidth=0.5)
plt.setp(markers, markersize=2)
plt.xticks(rotation=45, ha='right')
plt.xlabel('Frekvens / [Hz]')
```

```
plt.title('Energispektre for puls og flere iterationer af r')  
plt.show()
```



[]: