Hjemmeopgave 2

34210 Introduktion til Digital Kommunikation

Daniel Brasholt s214675

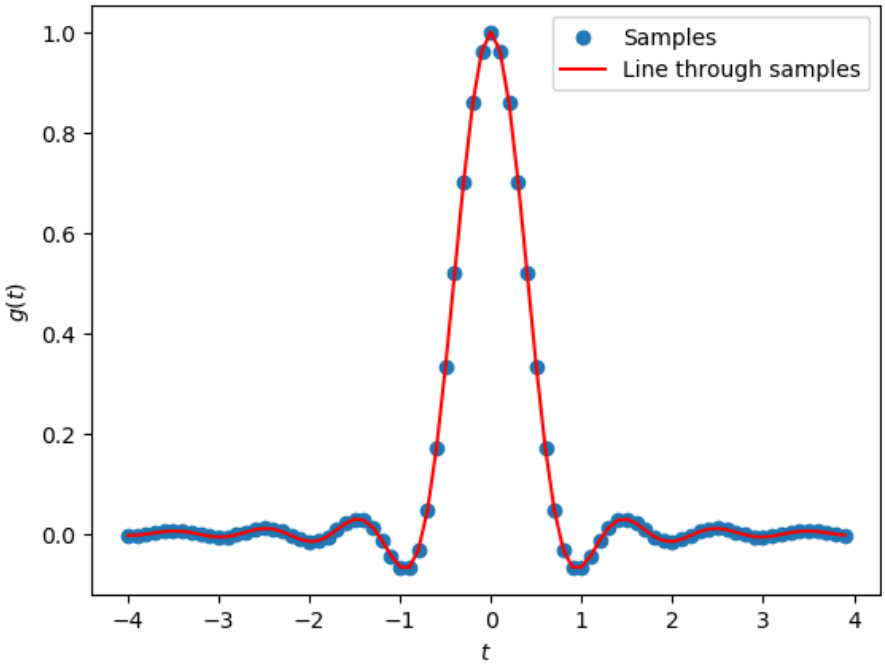
28. marts 2023

Indhold

Spørgsmål 1

Spm 1.1: Udsnit af pulsen

Med koden givet i opgaven fås følgende figur, der viser udsnittet af pulsen med længden $8T$:



Figur 1: *Udsnit af $g(t)$ med længden $8T = 8s$.*

Spm 1.2: Udtryk for filter, `g_M`

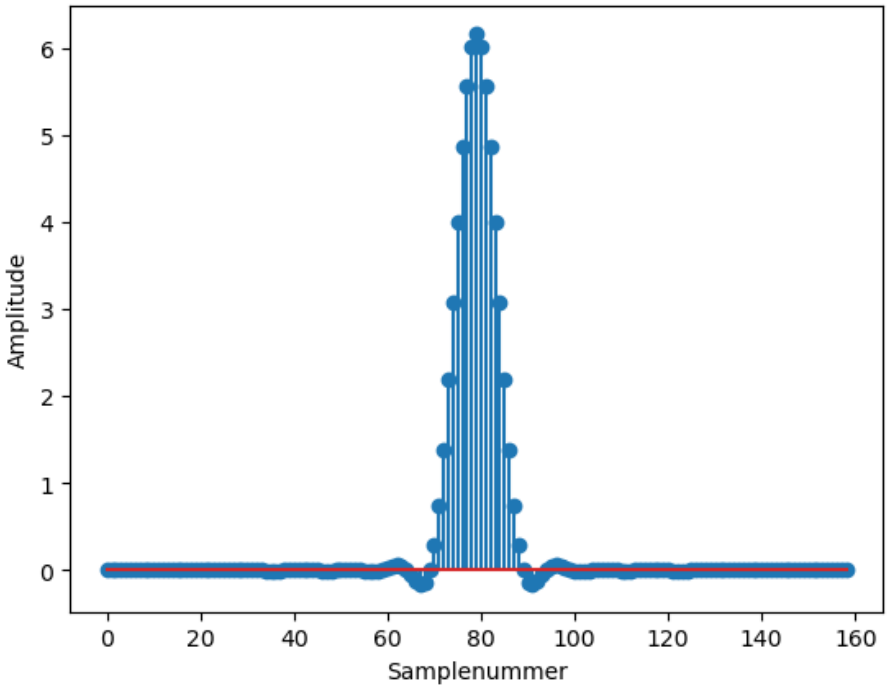
Påvirker den viste transmissionskanal ikke signalet, vil man kunne finde udtrykket for det tilpassede filter, `g_M`, ved at vende signalet om i tid, altså ved

$$g_M = \left(g_0^{(M)}, g_1^{(M)}, \ldots, g_{L-1}^{(M)},\right) = (g_{L-1}, g_{L-2}, \ldots, g_0)$$

Dette fås da blot ved at "flippe" signalet for en puls, `g_T`.

Spm 1.3: Start for sampling

Plotter man `x` fås:



Figur 2: *`x` plottet som funktion af samplenummer*

Det er tilmed fået at maksimum er ved sample 80[1] Da sættes

```
start=79
```

```
E=6.168297108214154
```

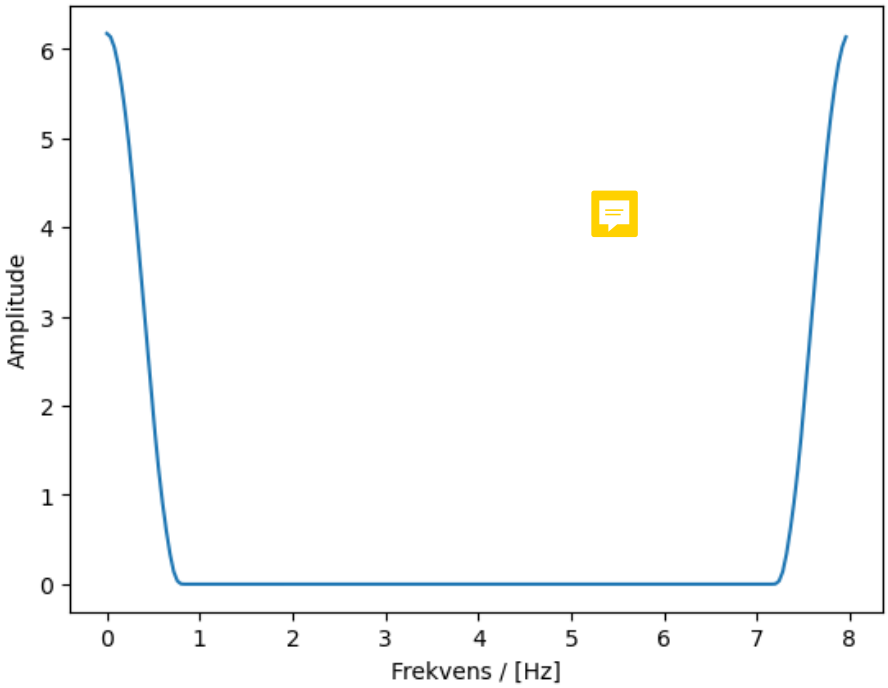
Spm 1.4: Analyse af `x`

Man kan se på `x`, at der ikke er nogen symbolinterferens tilsammen for de anvendte pulsformer, da resultatet af at plotte `x` (figur 2) giver det samme som $g(t)$ (figur 1). Der er stadig en "top" på signalet, hvilket bør give en klar transmission.

Spm 1.5: Spektret for x

Nedenfor er spektret for x plottet som en lige funktion. Da det skulle gøres med $N = 200$, er der indsat 0'er i midten af pulsen (midten efter pulsen er omrokeret, så den starter ved index `start`. Det kan ses på figuren, at toppen flader ud omkring $1Hz$, hvilket netop stemmer overens med figur `5.16` i noterne, eftersom $\alpha = 1$ i cosinus-roll-off-funktionen.

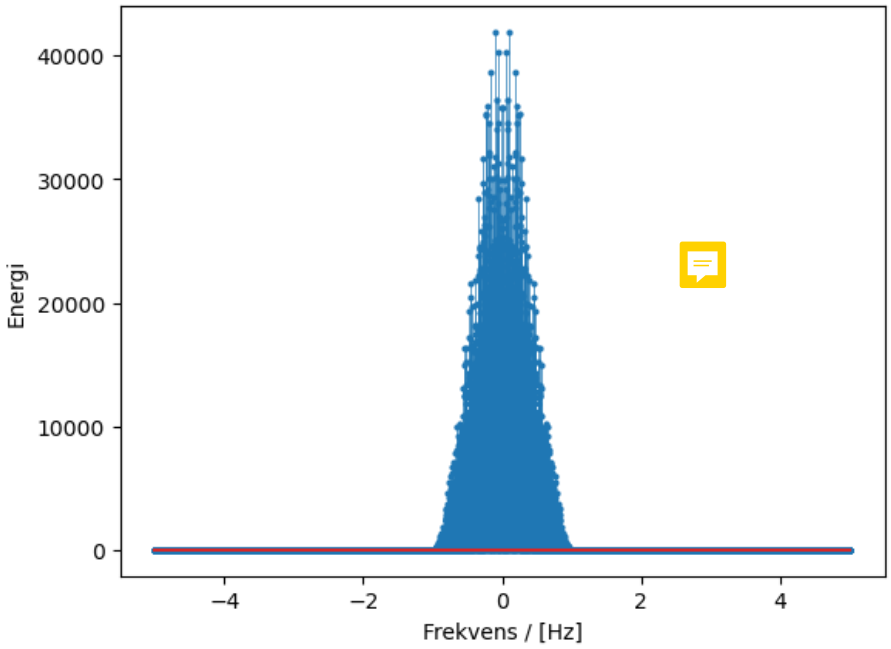
[1] I bilagene er det vist som 79, men i modsætning til MATLAB 0-indekserer `NumPy`.



Figur 3: *Spektrum for x placeret som en lige funktion*

Spørgsmål 2

Nedenfor ses energispektret for signalet v som funktion af frekvens. Det kan ses, at der er bidrag omkring $-1Hz$ til $+1Hz$, altså et frekvensbånd af bredden $2Hz$. Eftersom vi transmitterer med 1 baud, giver dette frekvensbånd mening.

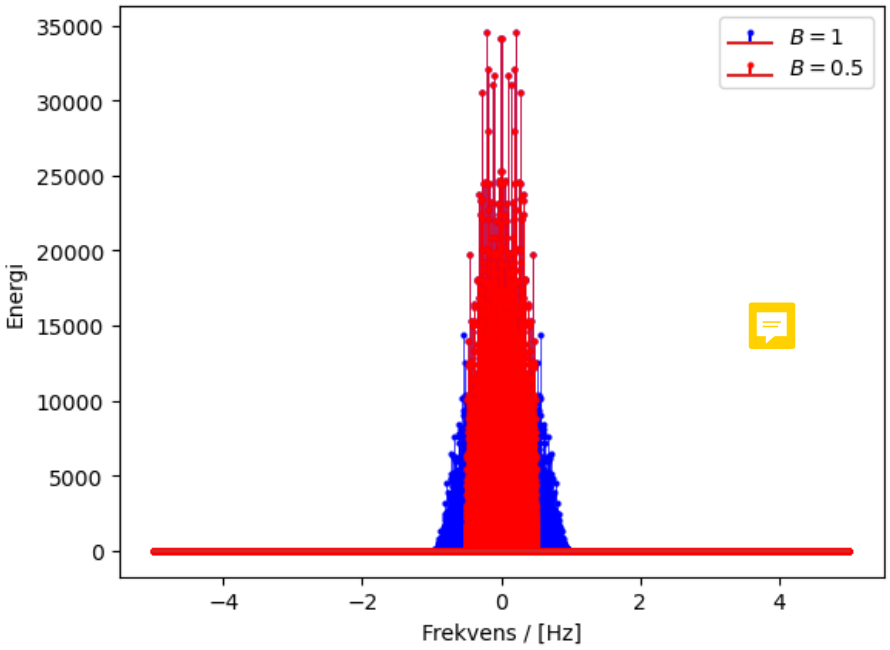


Figur 4: *Energispektrum for v som funktion af frekvens. Der ses bidrag i området $-1Hz$ til $+1Hz$*

Spørgsmål 3

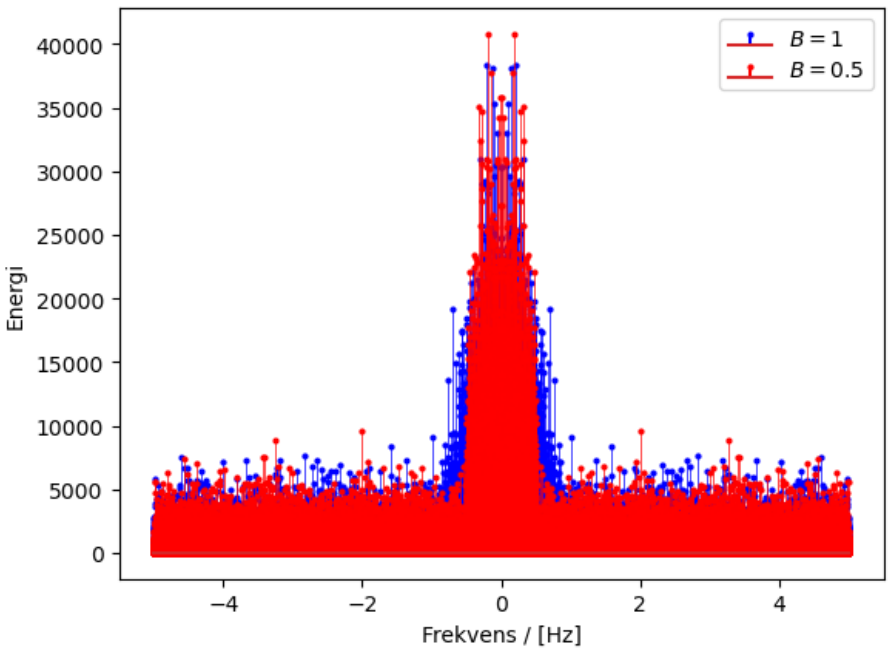
For at overføre med $B = 1/T$ baud kræver det en båndbredde på $B/2Hz = 1/(2T)$. For $B = 0.5$ giver det `Bf=5003`, hvilket betyder de første 5003 værdier samt de sidste 5003 værdier i spektret

for `V`. For $B = 1$ er det i stedet de første og sidste 10007 værdier. Laves da energispektret for `r`, når `sigma=0` og `B` er henholdsvis 0.5 og 1, fås følgende:



Figur 5: *Energispektrum for* `r`*, begrænset ved* 0.5 *og* 1

Det kan da ses på figur 5, at båndbegrænsningen virker, da den fjerner alle bidrag fra frekvenser over den givne `B`-værdi. Efterom `sigma` er sat til 0, vil der ikke være et bidrag af støj, hvorfor figuren med båndbegrænsning `B=1` er den samme som figur 4 (det modtagne signal vil være det samme som det sendte). Eventuelle forskelle ligger måske i indekseringen af omrokeringerne, men gør reelt set ikke en forskel for det endelige resultat. Sættes `sigma=1`, fås i stedet følgende spektrum for `r`:



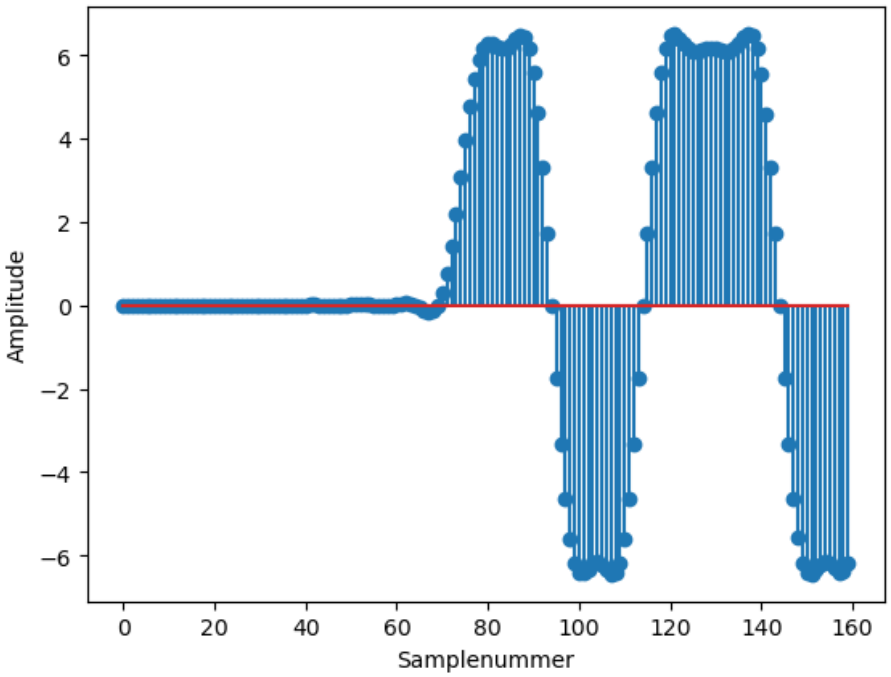
Figur 6: *Energispektrum for* `r`*, begrænset ved* 0.5 *og* 1*, denne gang med støj givet ved* `sigma=1`

På figur 6 kan det altså ses, at det modtagne signal er forskelligt fra det sendte, da der nu er kommet normalfordelt støj på `r`.

Spørgsmål 4

Spm 4.1: Begyndelsen af `s`

På nedenstående figur 7 kan det ses, at signalet skal samples fra position 79 og frem. Dette passer med den værdi, der blev fundet i Spm 1.3. Herfra skal der samples hver `m=10`. gang. Det kan ses, at der ikke er symbolinterferens, da disse samplingtidspunkter har klare fortegn og ikke ligger omkring 0.



Figur 7: *De første* 160 *samples af signalet* `s`

Spm 4.2: Antal fejl med `sigma=0`

Som det fremgår af bilagene, er antallet af fejl her fået til 0, hvilket er det, der forventes, eftersom båndbegrænsningen indeholder hele spektret og der ikke er nogen støj på linjnen - `sigma=0`.

Spm 4.3: Antal fejl med `sigma=1`

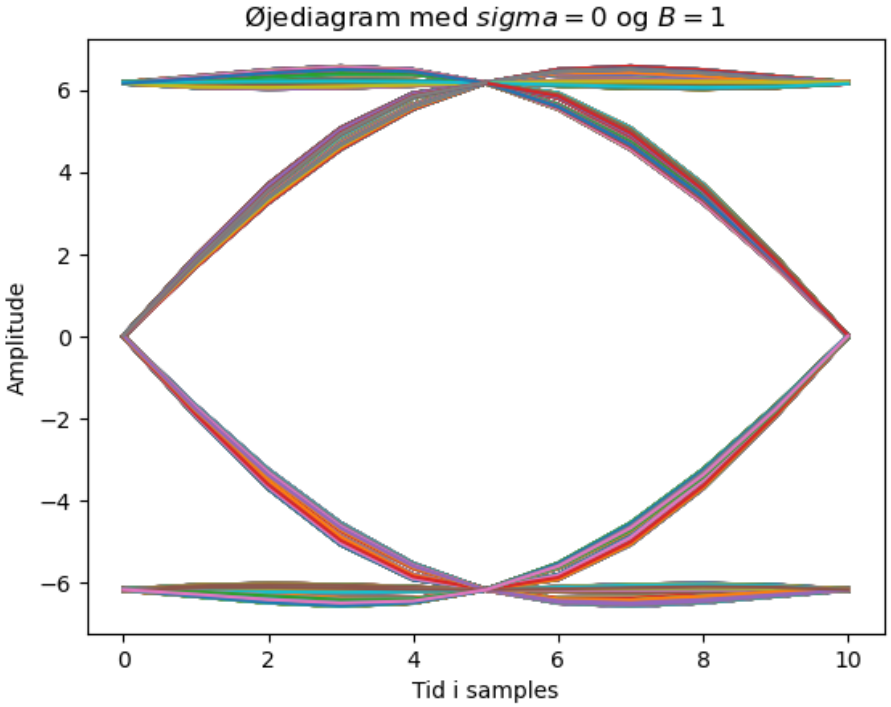
Teoretisk vil vi forvente, at antallet af fejl bliver:

$$Fejl = \left\lceil L \cdot \texttt{qfunc}\frac{\sqrt{E}}{\sigma} \right\rceil \tag{4.1}$$

I (4.1) er `L=10000`, `E=6.168297108214154` (som fundet i Spm 1.3) og $\sigma = 1$. Det teoretiske antal fejl er da fundet til at være 66, hvilket er tæt på det fundne antal, 56.

Spørgsmål 5

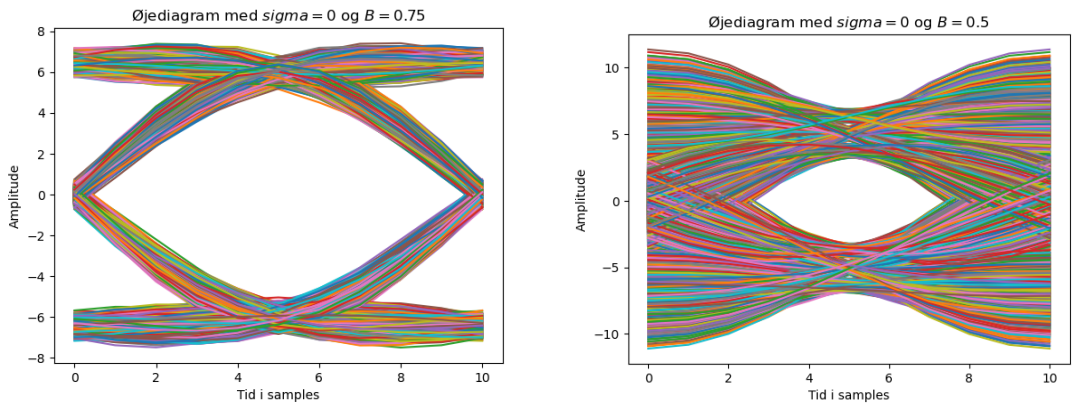
Spm 5.1: Øjediagram uden båndbegrænsning og støj



Figur 8: *Øjediagram for signalet uden støj og båndbegrænsning*

Ovenfor på figur 8 ses øjediagrammet for signalet uden støj og båndbegrænsning; dette vil være den "optimale" situation - den samme som Spm 4.2, hvor vi ikke kunne tælle nogen fejl.

Spm 5.2: $B = 0.75$ og $B = 0.5$

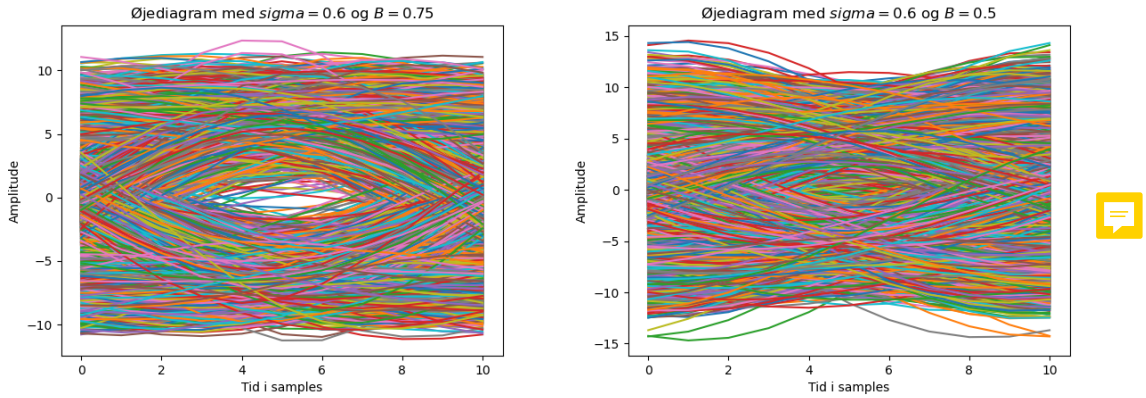


(a) $\sigma = 0$ *og* $B = 0.75$ ***(b)*** $\sigma = 0$ *og* $B = 0.5$

Figur 9: *Øjediagrammer med* $B = 0.75$ *og* $B = 0.5$ *- stadig uden støj*

På ovenstående figur 9 kan det ses, hvad resultatet af båndbegrænsning er på øjediagrammet. Det giver stadig ingen fejl på det endelige signal. Man kan dog se, at øjet er blevet tættere; altså at der er kommet mindre "afstand" mellem symbolerne, hvilket kan gøre det sværere at træffe en beslutning om, hvilket symbol, man har modtaget.

Spm 5.3: $\sigma = 0.6$



(a) $\sigma = 0.6$ *og* $B = 0.75$ ***(b)*** $\sigma = 0.6$ *og* $B = 0.5$

Figur 10: *Øjediagrammer med* $B = 0.75$ *og* $B = 0.5$ *- denne gang med støj med* $\sigma = 0.6$

Figur 10 viser øjediagrammerne for signalet, når der er støj på transmissionskanalen. Støjen er givet ved en normalfordeling med $\sigma = 0.6$. De to figurer viser også, at øjet lukker tættere sammen, når der kommer båndbegrænsning på. Dette betyder også, at der kommer flere reelle fejl i signalet end der teoretisk skulle forekomme; i begge scenarier (begge værdier af B) bør der være kun 1 fejl; dog kan man tælle, at der er 3 fejl ved $B = 0.75$ og 17 fejl ved $B = 0.5$. Man kan også se på delfigur 10b, at øjet nogle gange er helt "lukket", hvilket netop er der, man ville opleve fejl.

Hjemmeopgave 2

March 28, 2023

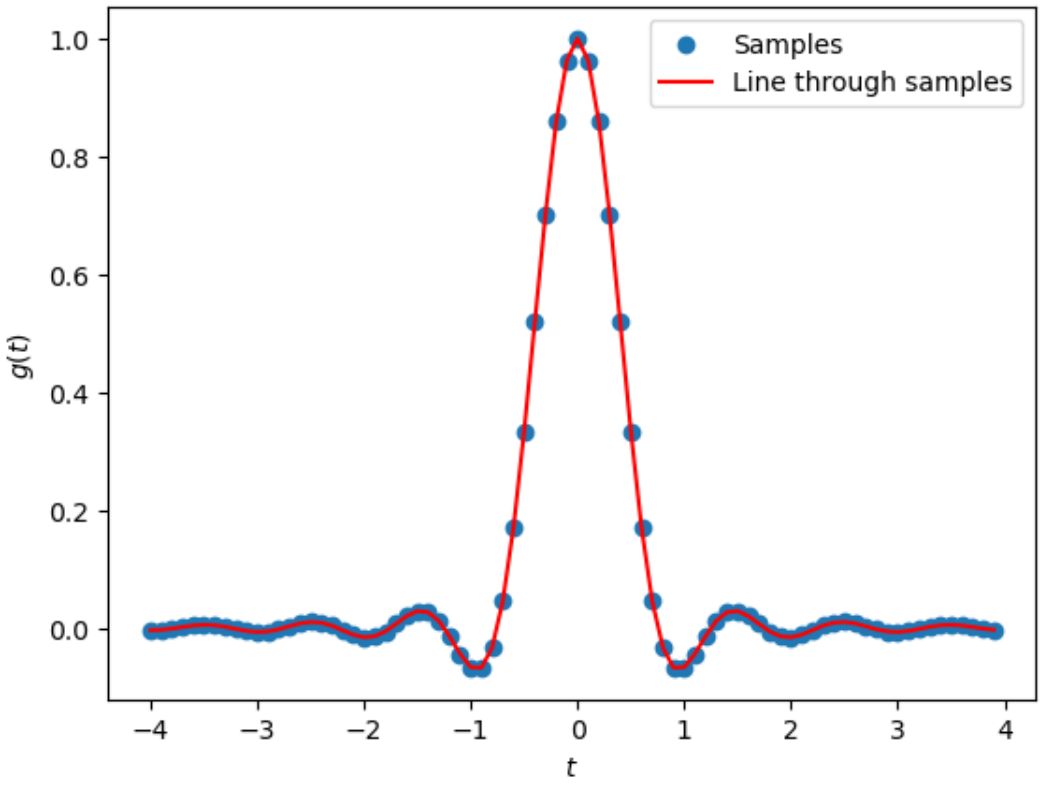
[53]:
```
import numpy as np
from matplotlib import pyplot as plt
from scipy.stats import norm
```

1 Spørgsmål 1

1.1 1.1

[2]:
```
T = 1 #second
m = 10
Ts = T/m
ti = np.arange(-4*T, 4*T, Ts)
g_T = (np.cos(2*np.pi * ti/T)) / (1- (4* ti/T)**2)
```

[3]:
```
plt.scatter(ti, g_T, label='Samples')
plt.plot(ti, g_T, 'r', label='Line through samples')
plt.xlabel('$t$')
plt.ylabel('$g(t)$')
plt.legend()
plt.show()
```



1.2 1.2

[4]:
```
g_M = np.flip(g_T)
```

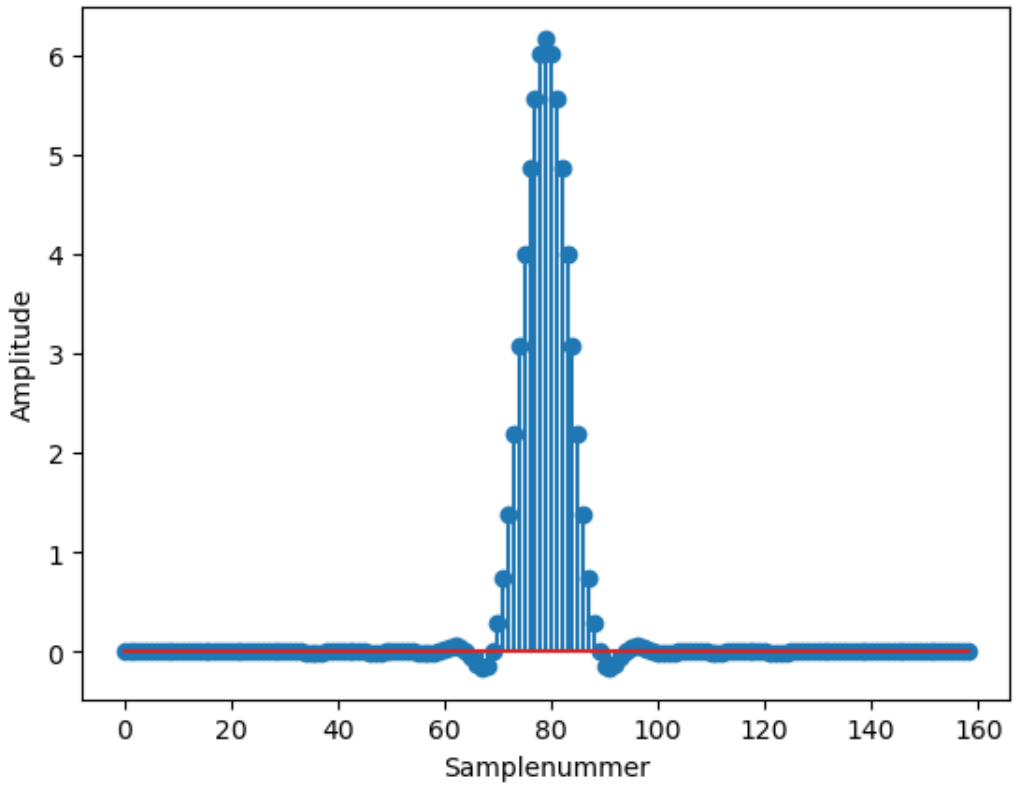
1.3 1.3

[5]:
```
x = np.convolve(g_T, g_M)
```

[51]:
```
start = x.argmax()
E = x[x.argmax()]
print(start, E)
```

```
79 6.168297108214154
```

[126]:
```
plt.stem(x)
plt.xlabel('Samplenummer')
plt.ylabel('Amplitude')
plt.show()
```



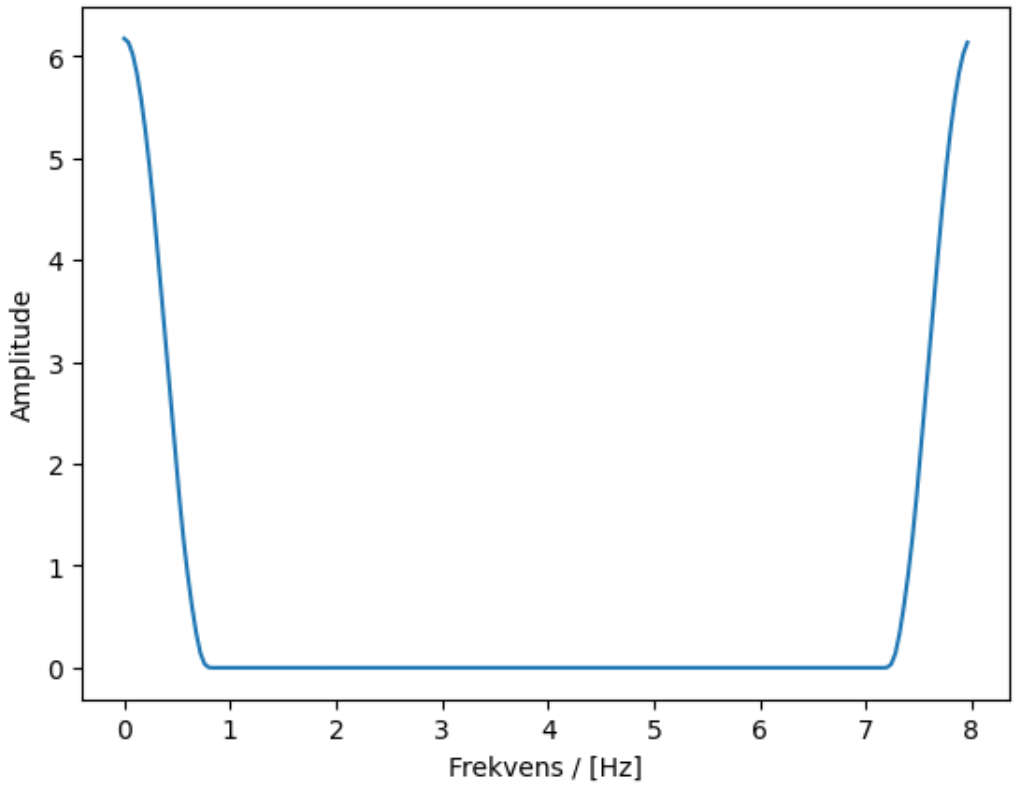
1.4 1.5

[128]:
```
x_shifted = np.concatenate((x[79:], np.zeros(200-x.shape[0]), x[:79]))
x_f = np.real(np.fft.fft(x_shifted) * Ts)
f = np.arange(0, 8*T, 8*T/200)
```

[9]:
```
f.shape
```

[9]: (200,)

[129]:
```
plt.plot(f,x_f)
plt.xlabel('Frekvens / [Hz]')
plt.ylabel('Amplitude')
plt.show()
```



[11]:
```
x_shifted.shape
```

[11]:
```
(200,)
```

2 Spørgsmål 2

2.1 2.1

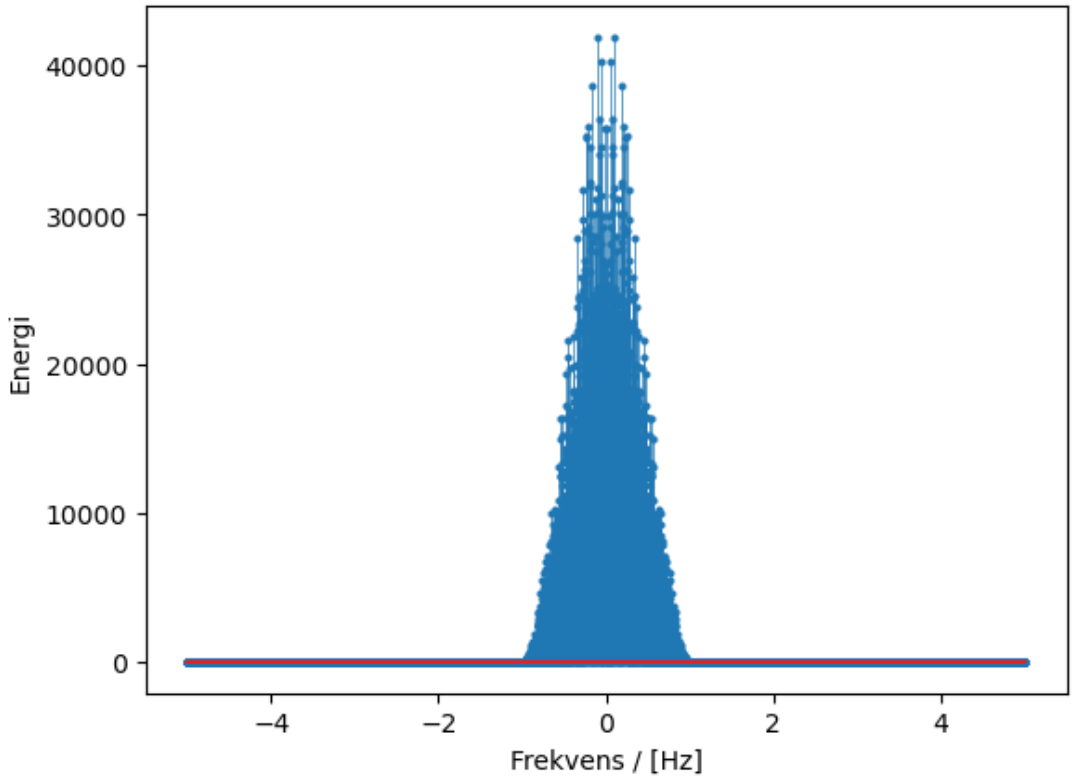
[143]:
```
L = 10000
a = 2*np.random.randint(0, 2, L)-1
```

[144]:
```
a0 = np.zeros(a.size * (m) - (m-1) , a.dtype)
a0[::m] = a
```

[145]:
```
v = np.convolve(g_T, a0)
```

[146]:
```
V = np.fft.fft(v) * Ts
V_energy = np.abs(V)**2
f_energy = np.arange(-1/(2*Ts), 1/(2*Ts), 1/Ts/V.shape[0])
markers, lines, base = plt.stem(f_energy, np.fft.fftshift(V_energy))
plt.setp(lines, linewidth=0.5)
```

```
plt.setp(markers, markersize=2)
plt.xlabel('Frekvens / [Hz]')
plt.ylabel('Energi')
plt.show()
```



3 Spøgsmål 3

[133]:
```
f = np.arange(0, 1/Ts, 1/Ts/V.shape[0])
```

[151]:
```
def index_of_best_fit(arr, B):
    i, = np.where(arr <= B)
    return np.max(i)

def band_limit(V, f, B):
    Bf = index_of_best_fit(f, B)
    VLP = np.concatenate( (V[:Bf], np.zeros(V.size-2*Bf), V[-Bf:]) )
    vlp = np.fft.ifft(VLP)/Ts
    return np.real(vlp)

def add_noise(V, f, B, sigma):
```

```
    vlp = band_limit(V, f, B)
    r = vlp + sigma*np.random.randn(vlp.size)
    return r
```

[134]:
```
print(index_of_best_fit(f, 0.5),
index_of_best_fit(f, 1))
```

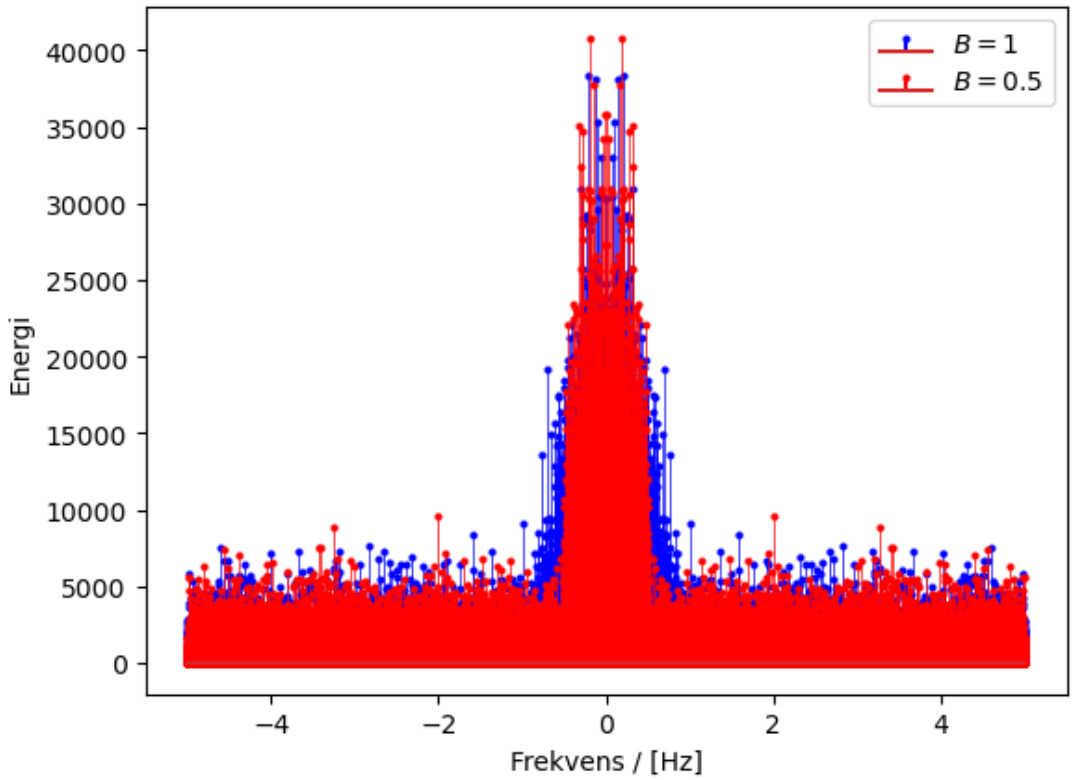
```
5003 10007
```

[148]:
```
sigma = 1 # sat til 0, når der ikke er støj

r = add_noise(V, f, 1, sigma)
R = np.abs(np.real(np.fft.fft(r)*Ts)) **2
markers, lines, base = plt.stem(f_energy,np.fft.fftshift(R), linefmt='b',␣
 ↪label='$B=1$')
plt.setp(lines, linewidth=0.5)
plt.setp(markers, markersize=2)

r = add_noise(V, f, 0.5, sigma)
R = np.abs(np.real(np.fft.fft(r)*Ts)) **2
markers, lines, base = plt.stem(f_energy,np.fft.fftshift(R), linefmt='r',␣
 ↪label='$B=0.5$')
plt.setp(lines, linewidth=0.5)
plt.setp(markers, markersize=2)

plt.xlabel('Frekvens / [Hz]')
plt.ylabel('Energi')
plt.legend()
plt.show()
```

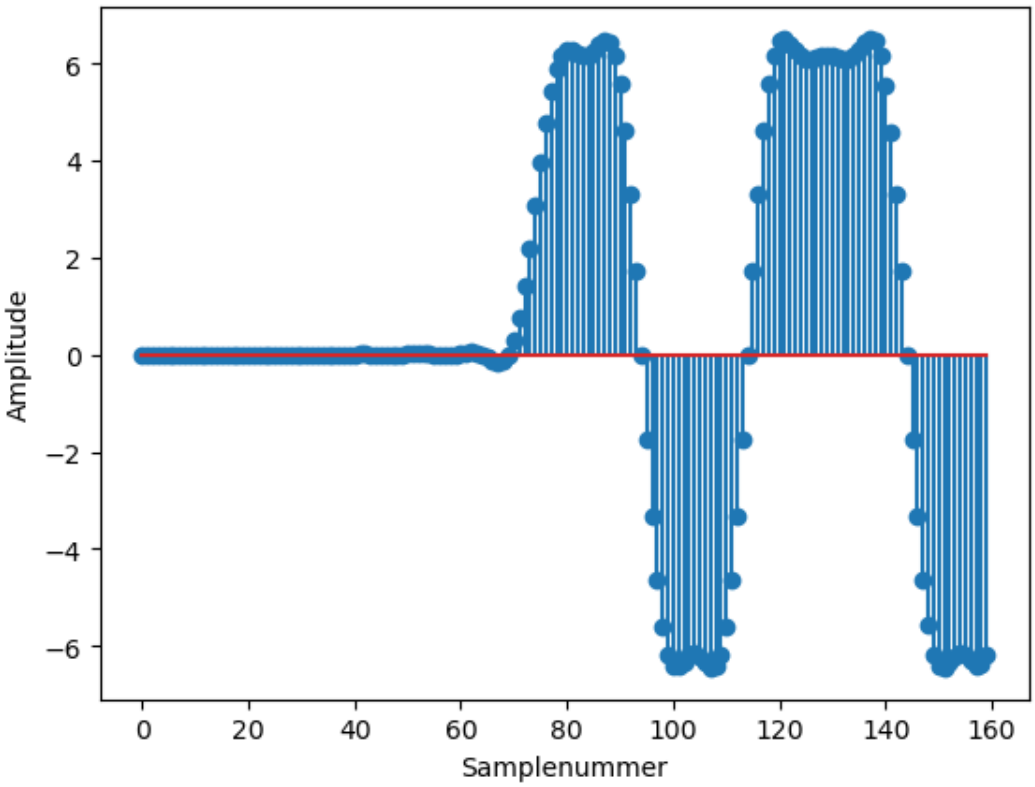


4 Spørgsmål 4

4.1 4.1

[149]:
```
sigma, B = 0, 1
r = add_noise(V, f, B, sigma)
s = np.convolve(g_M,r)
```

[152]:
```
plt.stem(s[:160])
plt.xlabel('Samplenummer')
plt.ylabel('Amplitude')
plt.show()
```



[153]:
```
a[:10]
```

[153]:
```
array([ 1,  1, -1, -1,  1,  1,  1, -1, -1,  1])
```

4.2 4.2

Nedenstående kode laver samplingen og tester om a (information) og a_hat (modtaget information) er ens. Funktionen `all()` giver True, hvis og kun hvis alle elementer i en liste er True - sagt med andre ord giver den True, hvis og kun hvis a_hat er den samme som a.

[154]:
```
a_hat = np.sign(s[start:-start:m])
all(a_hat == a)
```

[154]:
```
True
```

4.3 4.3

[155]:
```
sigma, B = 1, 1
r = add_noise(V, f, B, sigma)
s = np.convolve(g_M,r)
a_hat = np.sign(s[start:-start:m])
```

```
sum(a != a_hat)
```

[155]: 56

[54]:
```
def qfunc(x):
    return 1-norm.cdf(x)
```

[57]:
```
np.ceil(L * qfunc(np.sqrt(E) / sigma))
```

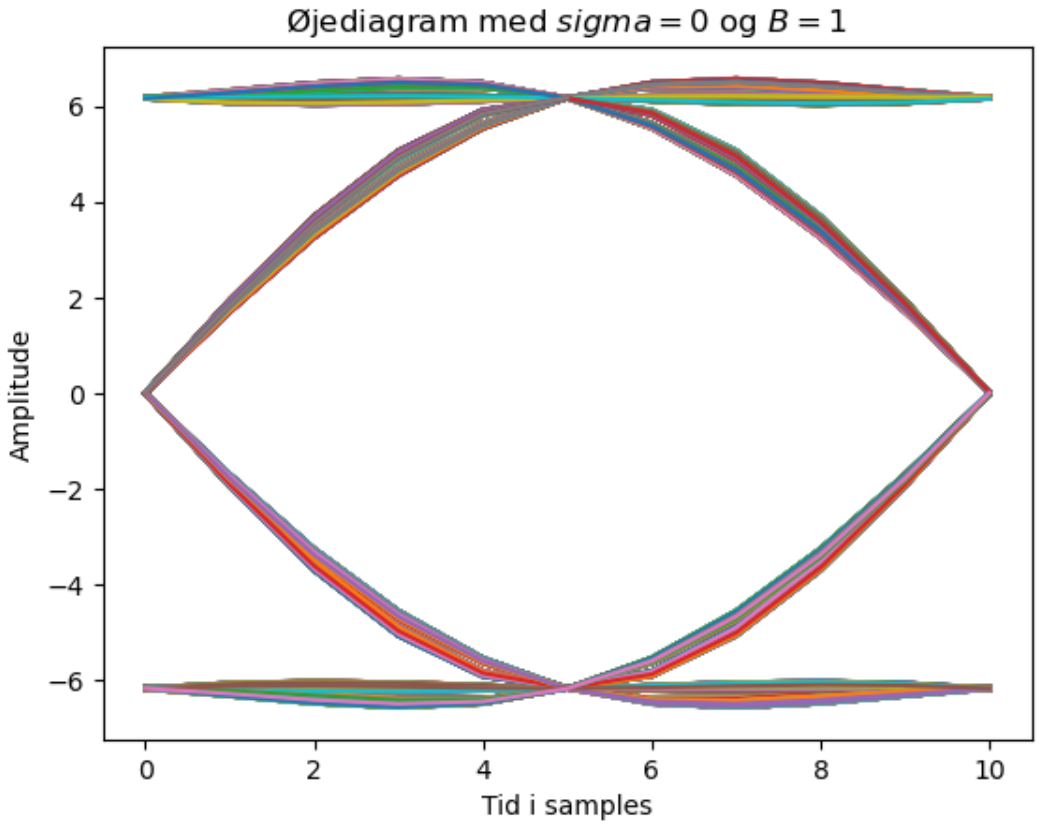
[57]: 66.0

5 Spørgsmål 5

5.1 5.1

[114]:
```
sigma, B = 0, 1
r = add_noise(V, f, B, sigma)
s = np.convolve(g_M,r)
s_cutoff = s[start+m-4:-start-m] # -4 for at få øjet til at være mere
 ↪"øjeformet"
```

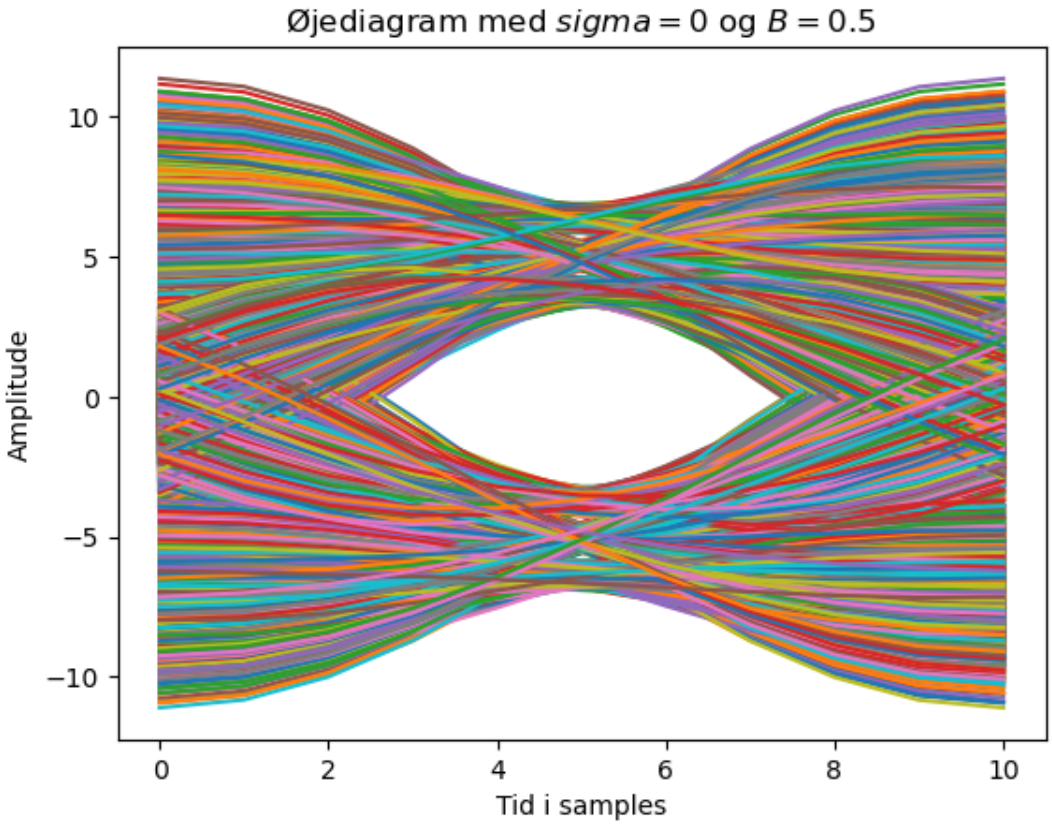
[118]:
```
for i in range(9999):
    plt.plot(s_cutoff[i*m-1:(i+1)*m])
plt.xlabel('Tid i samples')
plt.ylabel('Amplitude')
plt.title('Øjediagram med $sigma=0$ og $B=1$')
plt.show()
```



5.2 5.2

[123]:
```
sigma, B = 0, 0.5
r = add_noise(V, f, B, sigma)
s = np.convolve(g_M,r)
a_hat = np.sign(s[start:-start:m])
print('Antal fejl:', sum(a != a_hat))
s_cutoff = s[start+m-4:-start-m] # -4 for at få øjet til at være mere
 ↪"øjeformet"
for i in range(9999):
    plt.plot(s_cutoff[i*m-1:(i+1)*m])
plt.xlabel('Tid i samples')
plt.ylabel('Amplitude')
plt.title(f'Øjediagram med $sigma={sigma}$ og $B={B}$')
plt.show()
```

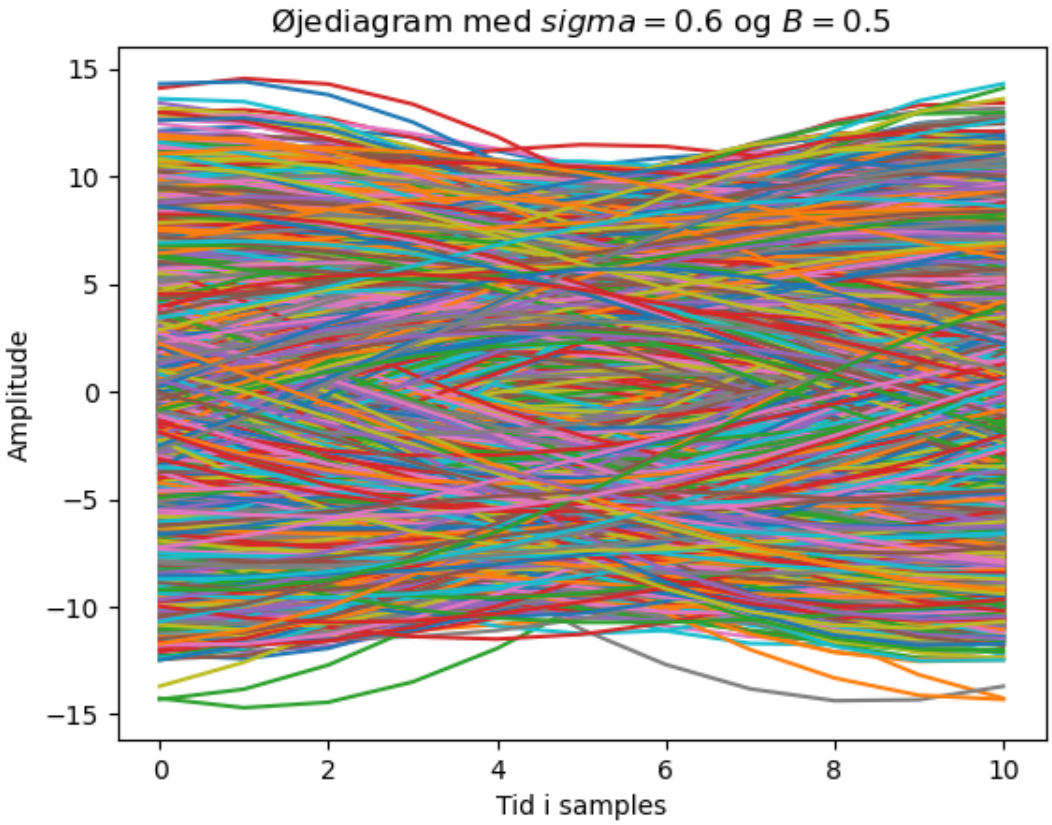
```
Antal fejl: 0
```



5.3 5.3

[125]:
```
sigma, B = 0.6, 0.5
r = add_noise(V, f, B, sigma)
s = np.convolve(g_M,r)
a_hat = np.sign(s[start:-start:m])
print('Antal fejl:', sum(a != a_hat))
print('Teoretisk antal fejl:', np.ceil(L * qfunc(np.sqrt(E) / sigma)))
s_cutoff = s[start+m-4:-start-m] # -4 for at få øjet til at være mere␣
 ↪"øjeformet"
for i in range(9999):
    plt.plot(s_cutoff[i*m-1:(i+1)*m])
plt.xlabel('Tid i samples')
plt.ylabel('Amplitude')
plt.title(f'Øjediagram med $sigma={sigma}$ og $B={B}$')
plt.show()
```

```
Antal fejl: 17
Teoretisk antal fejl: 1.0
```


Øjediagram med $sigma = 0.6$ og $B = 0.5$
Amplitude
Tid i samples
15
10
5
0
−5
−10
−15
0
2
4
6
8
10

[ ]: