Miniprojekt B

Mathias Vraa *au618687*

Indholdsfortegnelse

Formål

Følgene tekst er taget fra øvelsesvejledningen:

"Lav i Matlab et analysesystem, som baserer sig på Diskret Fourier Transformation (DFT). Systemet skal kunne lave DFT (vha. fft.m funktionen) på vilkårlige signaler og vise størrelsen af DFT'en med korrekte akser og skalering."

"Inkluder kortfattet teori, essentiel kode samt passende konklusioner."

Implementation

Hver opgave indeholder introduktion, implementation, resultater og diskussion. Resultater er vist under de grå kode blokke.

Opgave 1

"Find på nettet eksempler på disse to signaler: Vindmøllestøj (enten vingestøj eller generator/gear støj) og blæserstøj fra en computer. Der skal naturligvis haves kendskab til samplefrekvensen. Plot udvalgte 10 sekunder af begge signaler og beskriv ligheder og forskelle."

Med audioread indlæses hvert lydsignal ind på et variabel (s1, s2). Samplefrekvens indlæses på fs. Længden er nøjagtigt 10 sekunder for hvert klip:

```
[s1, fs1] = audioread("lyd_opg1-4/windmill.wav"); % windmill
[s2, fs2] = audioread("lyd_opg1-4/pc_fan.wav"); % pc fan
```

De har begge samme sample frekvens. Jeg laver derfor ét variabel, fs, til dem begge:

```
fs1
```

```
fs1 = 44100
```

```
fs2
```

```
fs2 = 44100
```

```
fs = fs1
```

```
fs = 44100
```

Jeg finder tid pr. sample(Ts) ligesom i projekt A. Spilletiden (Tn) er præcis 10 sekunder for begge klip. Mængden af sample(n) er også ens for de to klip, da de har samme sample rate:

```
Ts = 1/fs % sekunder
```

```
Ts = 2.2676e-05
```

```
Tn = 10 % sekunder
```

```
Tn = 10
```

```
n = fs * Tn % antal sample
```

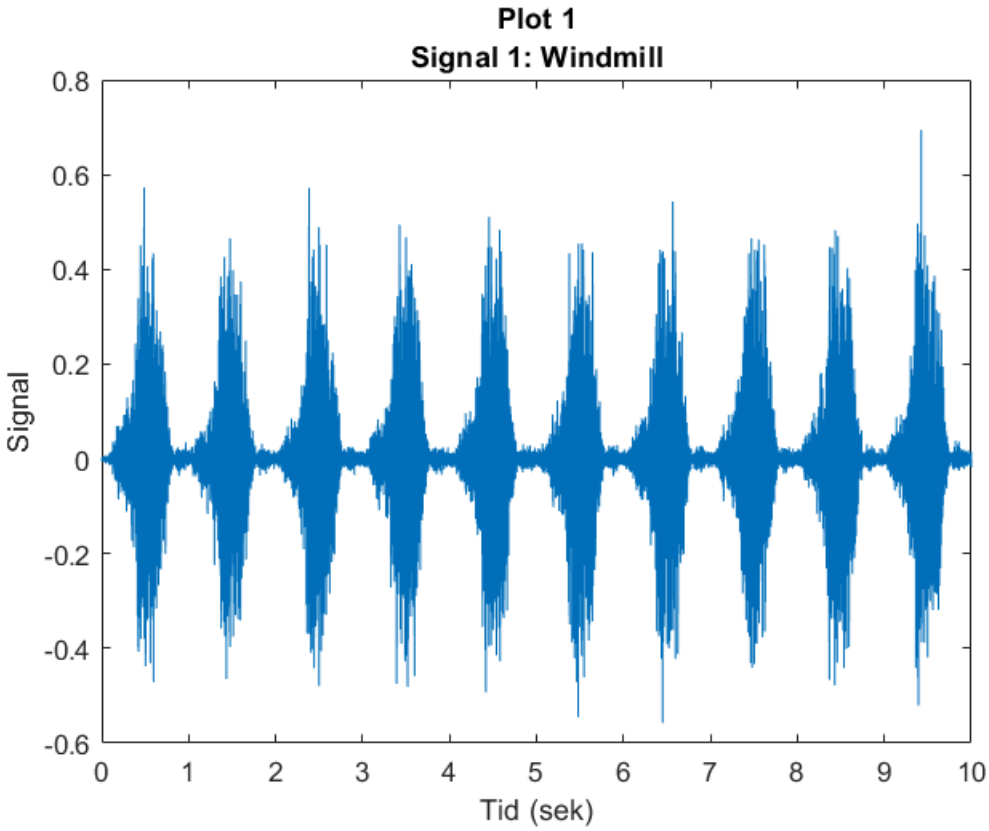
```
n = 441000
```

Tidaksen kan nu beregnes. Vi starter på 0, springer med Ts og stopper med at springe når Tn nåes:

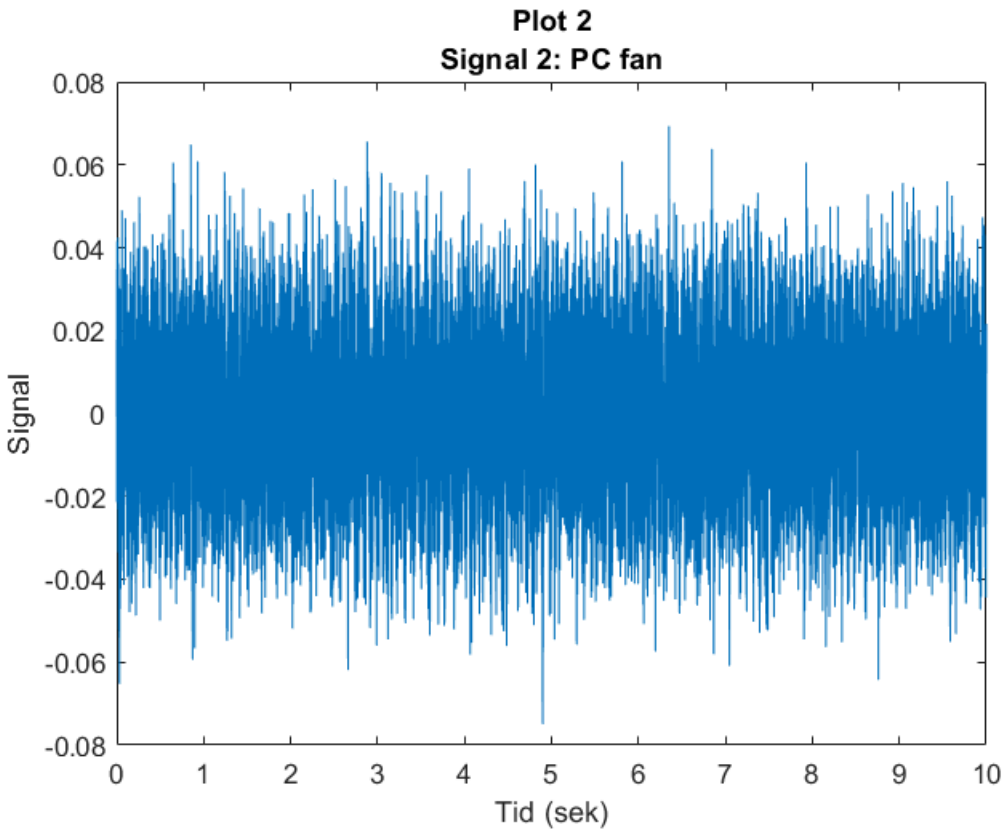
```
t = 0:Ts:Tn-Ts;
```

Signalerne kan nu plottes.

```
% windmill
plot(t,s1)
title("Plot 1" + newline + "Signal 1: Windmill");
ylabel("Signal");
xlabel("Tid (sek)");
```




```
% pc fan
plot(t,s2)
title("Plot 2" + newline + "Signal 2: PC fan");
ylabel("Signal");
xlabel("Tid (sek)");
```



En vindmølle og en pc blæser er teknisk set to ens komponenter - vindmøllen er dog meget større. Vindmøllen bevæger sig langt langsommere end en blæser - den har altså en lavere frekvens. Det kan man se på plottet, da hver spike svarer til en vinge der suser forbi mikrofonen. Det samme gælder for en blæser, vingerne bevæger sig bare så hurtigt at lyden/plottet kommer til at virke konstant - den har altså en højere frekvens.

Opgave 2

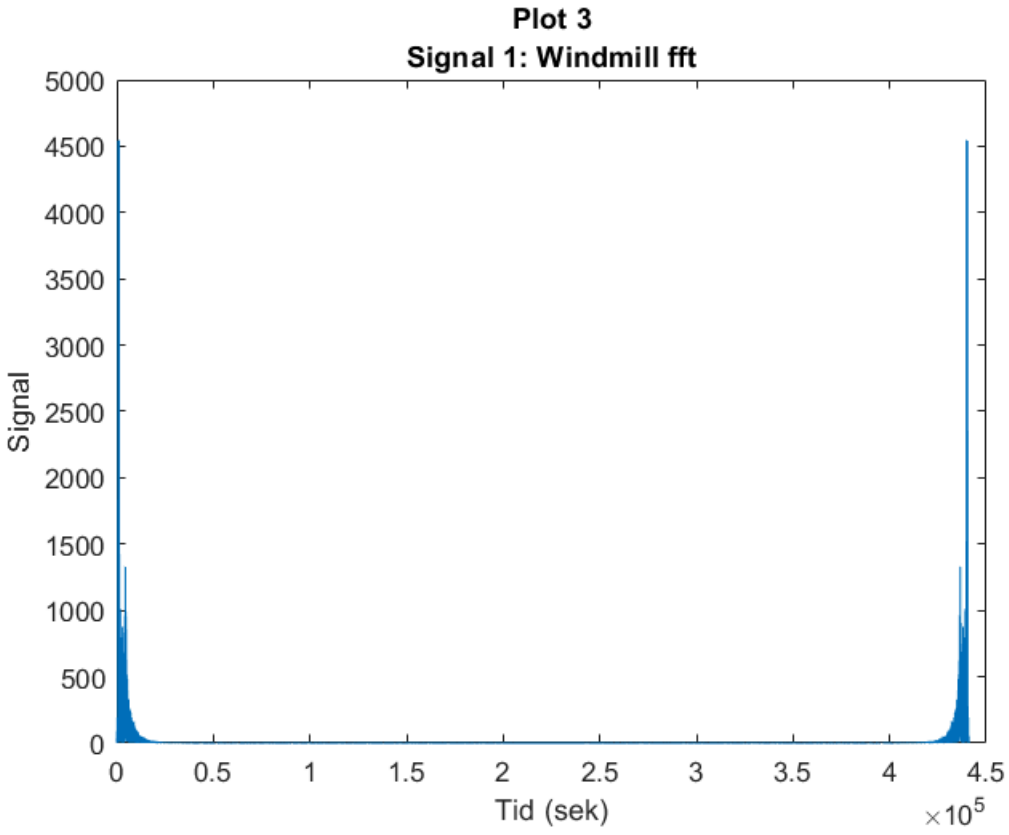
"Lav frekvenstransformation på de 10 sekunders udsnit af signalerne i 1) og vis frekvensspektrene. Beskriv igen ligheder og forskelle."

Jeg bruger funktionen fft til at finde "**f**ast **f**ourier **t**ransform" at mine signaler. Jeg får derfor en array med en række komplekse tal:

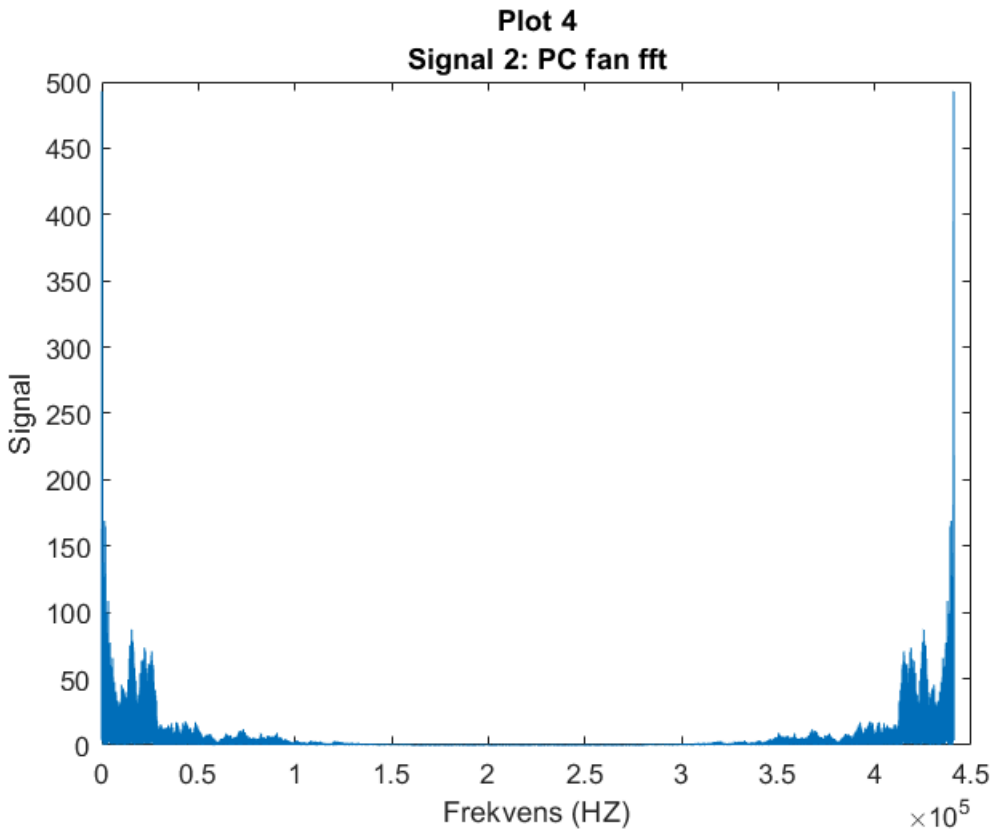
```
S1 = fft(s1);
S2 = fft(s2);
```

For at plotte signalerne tager jeg den absolutte værdi af det transformeret signal:

```
%windmill
plot(abs(S1))
title("Plot 3" + newline + "Signal 1: Windmill fft");
ylabel("Signal");
xlabel("Tid (sek)");
```




```
%pc fan
plot(abs(S2))
title("Plot 4" + newline + "Signal 2: PC fan fft");
ylabel("Signal");
xlabel("Frekvens (HZ)");
```



Fft deler signalet op efter frekvens. Frekvens ligger langs x-aksen og hyppighed for frekvens i siganlet er amplituden. For at kunne plotte bruges abs funktionen som finder den absolutte værdi. Plottet vil altid være spejlvendt - vi fokusere kun på det der ligger på den vestre side.

Som nævnt i opgave 1 er frekvensen for en vindmøllen langsommere end pc blæseren. Den har derfor flere spike i den lave ende af spektret og meget få spikes i den høje ende. PC blæseren har en hurtig frekvens, og den har derfor en del spikes i den høje ende af frekvens spektret.

Opgave 3

"Bestem lavfrekvent effekt Plow (under 60Hz) og højfrekvent effekt Phigh (over 60) for de to støjsignaler i 1). Udregn for de to signaler effektforholdet Plow/Phigh og diskuter. I hvilket er der mest lavfrekvent energi?"

For at udføre denne opgave skal de to signaler først deles op i to dele. Dem over 60Hz og dem under. Da fft har opdelt efter frekvens er dette meget nemt. Jeg tager bare de første 60 indexer for dem under, og de sidste indexer for dem over. Den spejlvendte del skal ikke med, så jeg dividere med 2:

```
S1_u60 = abs(S1(1:60));
S1_o60 = abs(S1(61:end/2));

S2_o60 = abs(S2(1:60));
S2_u60 = abs(S2(61:end/2));
```

Nu bestemmes effekten for hver:

```
S1_plow = (1/length(S1_u60)) * sum(S1_u60.^2)
```

```
S1_plow = 131.1752
```

```
S1_phigh = (1/length(S1_o60)) * sum(S1_o60.^2)
```

```
S1_phigh = 3.9107e+03
```

```
S2_plow = (1/length(S2_u60)) * sum(S2_u60.^2)
```

```
S2_plow = 96.7319
```

```
S2_phigh = (1/length(S2_o60)) * sum(S2_o60.^2)
```

```
S2_phigh = 3.3067e+03
```

Effekt forholdet beregnes:

```
S1_ratio = S1_plow/S1_phigh
```

```
S1_ratio = 0.0335
```

```
S2_ratio = S2_plow/S2_phigh
```

```
S2_ratio = 0.0293
```

Opgave 4

"Udregn frekvensopløsningen for de to signaler."

Frekvensoplæsningen beregnes ved at dividere sample frekvensen med samples.

```
freq_res = fs/n % Hz
```

```
freq_res = 0.1000
```

Frekvens resolution er det spring vi tager langs x-aksen på et fft plot. Med andre ord, man tager sample frekvensen og opdeler den med mængden af samples.

Opgave 5

"Lav DFT på et telefon dial-up signal (otte cifre). Hvilke frekvenser (slå dem op) skal de enkelte cifre indeholde? Identificer disse på frekvensspektret."

Jeg indlæser først mit dial up signal på et variabel. Det dial up signal jeg har valgt, indeholder sekvensen "**50736433**". Sample frekvensen(fs_dial) er 8000Hz:

```
[s_dial, fs_dial] = audioread("8_digit_dial.wav"); % 8 digit dial tone
fs_dial % Hz
```

```
fs_dial = 8000
```

Før jeg fourier transformere, laver jeg min frekvens akse. Til det, skal jeg bruge sample længden (n_dial) og frekvens resolution (fRes_dial):

```
n_dial = length(s_dial)
```

```
n_dial = 7580
```

```
fRes_dial = fs_dial / n_dial
```

```
fRes_dial = 1.0554
```

Jeg laver nu min frekvens akse. Den skal starte på 0, springe med intervallet svarene til frekvens resolution og stoppe når den når enden af samples (resolution gange sample mængde):

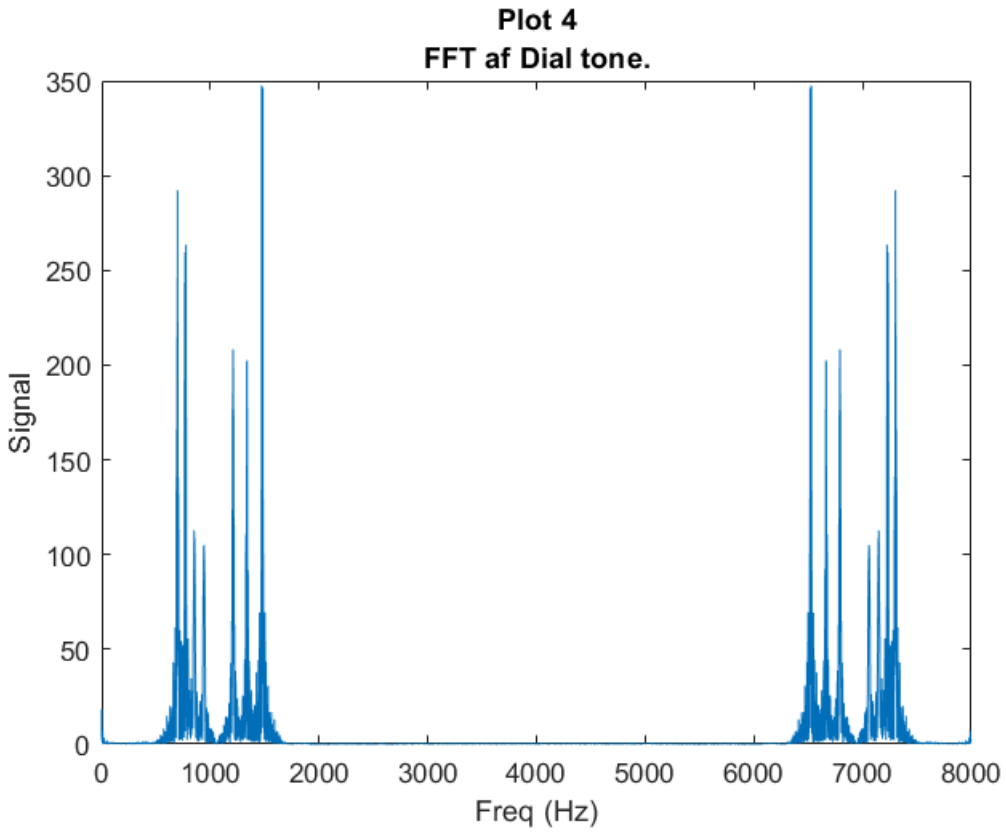
```
freqX = 0:fRes_dial:fRes_dial*(n_dial-1)
```

```
freqX = 1×7580
10^3 ×
         0    0.0011    0.0021    0.0032    0.0042    0.0053    0.0063    0.0074 ···
```

Jeg laver DFT på signalet med fft funktionen og plotter langs frekvens aksen som jeg lige har lavet:

```
S_dial = fft(s_dial);

% dial tone plot
plot(freqX,abs(S_dial))
title("Plot 4" + newline + "FFT af Dial tone.");
ylabel("Signal");
xlabel("Freq (Hz)");
```



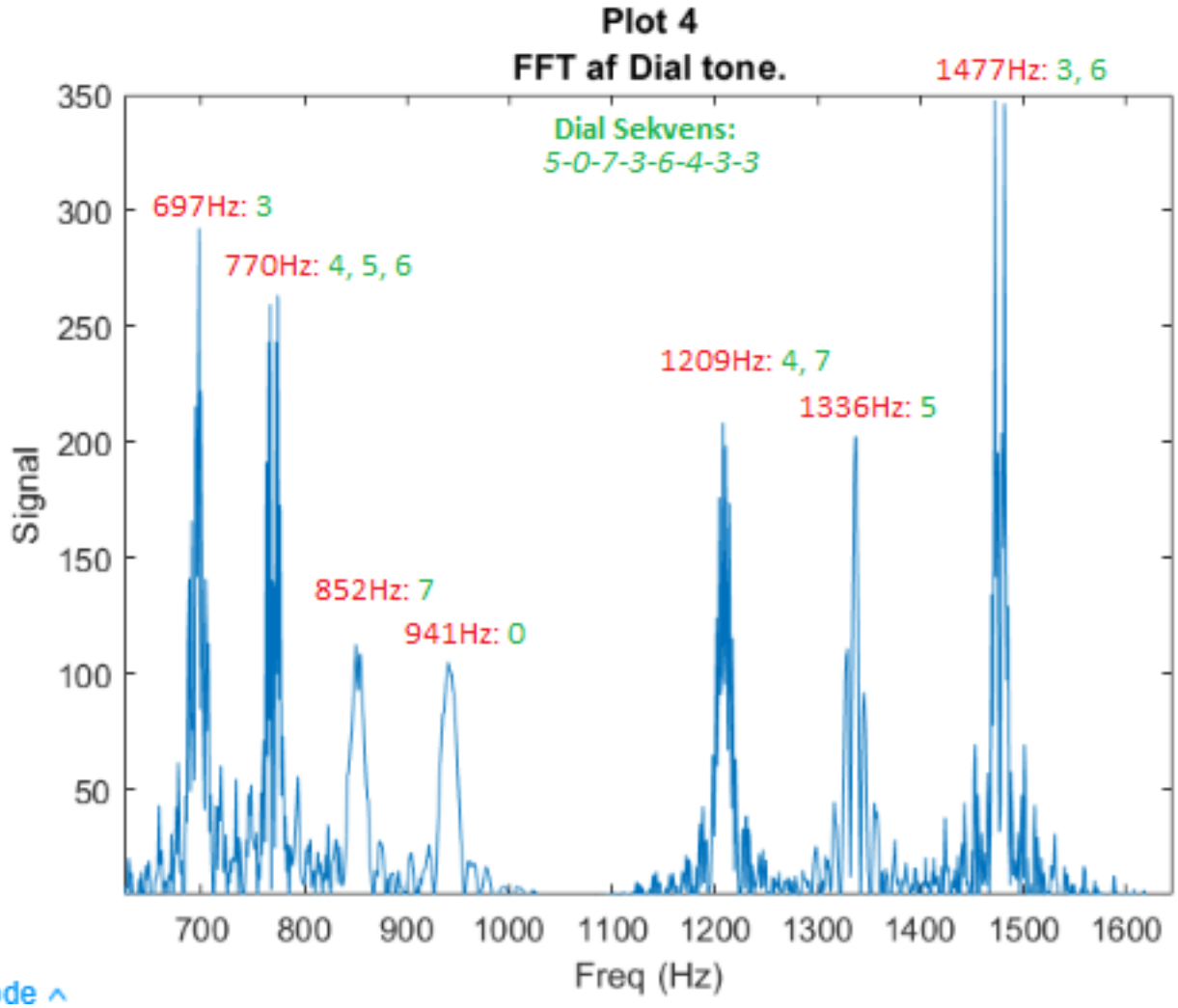
Frekvenser for hvert enkelt key er vist her. Som man kan se betår hvert key tryk af to sekvenser - dvs. 2 frekvenser overlapper og laver én tone:

DTMF keypad frequencies (with sound clips)

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Figur #: Table over keypad frekvenser fundet på Wikipedia.

Zoomes ind på højre side af plottet kan de forskellige frekvenser identificeres. Fft funktionen har delt frekvenserne op så de ligger langs x-aksen. Funktionen kan endda skille de overlappende toner fra hinanden. På figur # nedenunder kan man se hvor tonerne ligger - den røde tekst viser frekvensen for en spike og den grønne tekst viser den tilhørene key-tone.



Som man kan se på figur # er nogle spikes større end andre. Det har noget at sige med hvor meget tonen er repræsenteret i signalet. 1477Hz er f.eks. repræsenteret i signalet ved 4 keytryk. Dens amplitude er derfor meget stor. 941Hz er kun repræsenteret med ét tryk (key 0), så den er derfor ca. 4 gange mindre. Her er en rangering af key presses pr. frekvens:

1. **1477Hz:** 4 gange
2. **697Hz:** 3 gange
3. **770Hz:** 3 gange
4. **1209Hz:** 2 gange
5. **1336Hz:** 2 gange
6. **852Hz:** 1 gang
7. **941Hz:** 1 gang

Opgave 6

"Find følgende tre signaler: Vinglas der knipses på, en maskine der roterer, musik efter eget valg. Lav frekvenstransformation og vis amplitudespektrene – både med og uden udglatning. Diskuter de forskellige karakteristika i spektrene."

Jeg indlæser valgte signaler:

```
[s1, fs1] = audioread("lyd_opg6/vinglas.wav"); % vinglas der knipses på
[s2, fs2] = audioread("lyd_opg6/dc_motor.wav"); % roterende maskine
[s3, fs3] = audioread("lyd_opg6/yndigt_land.wav"); % musik: "Der er et yndigt land"
```

De har alle samme frekvens sample, så jeg laver en delt variabel, fs:

```
fs1
```

```
fs1 = 44100
```

```
fs2
```

```
fs2 = 44100
```

```
fs3
```

```
fs3 = 44100
```

```
fs = fs1;
```

P3.15

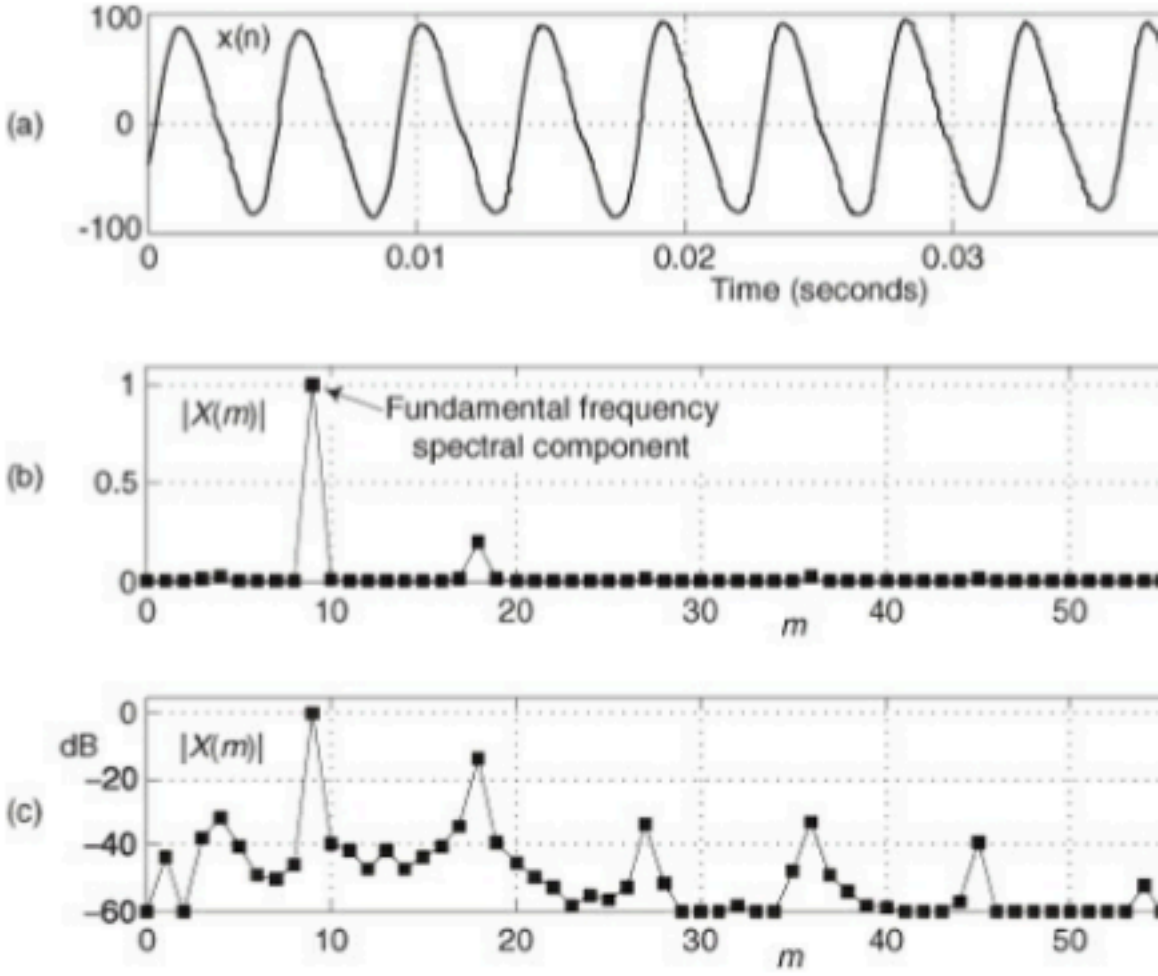
Opgave 3.15 fra ***"Understanding Digital Signal Processing"*** af ***"Richard D. Lyons"***:

3.15 Here is a real-world spectrum analysis problem. Figure P3-15(a) shows time sequence. (For clarity, we do not show the $x(n)$ samples as individual the sound of the "A3" note ("A" below middle "C") from an acoustic guitar kHz. Figure P3-15(b) shows the $X(m)$ spectral magnitude samples, the D scale for the frequency index range of $0 \leq m \leq 59$.

(a) Based on the $X(m)$ samples, what is the *fundamental* frequency, in H note?

(b) When we plot the DFT magnitude samples on a logarithmic scale, as see spectral *harmonics* and learn that the guitar note is rich in spectral are integer multiples of the fundamental frequency.) That's why guitars ha depending on the guitarist's skill, of course. What is the frequency of the component of the guitar's "A3" note?

Figure P3-15



P3.19

Opgave 3.19 fra ***"Understanding Digital Signal Processing"*** af ***"Richard D. Lyons"***:

3.19 Assume that an N-point DFT, performed on an N-sample $x(n)$ time-doma
DFT frequency-domain sample spacing of 100 Hz. What would be the
sample spacing in Hz if the N-sample $x(n)$ time sequence was padde
samples and we performed a DFT on that extended-time sequence?

Konklusion