

AARHUS UNIVERSITET

DSB

SEMESTER 3

Mini-projekt

Studerende:

Mette HAMMER NIELSEN-KUDSK

Martin BANASIK

Finja JETTE RALFS

October 1, 2015



Indholdsfortegnelse

1	Begreber	2
2	Aliasering	2
2.1	Aliasering	2
3	Envelopes	3
3.1	Envelopes	3
4	ADSR	3
4.1	ADSR	4
5	LFO	4
5.1	LFO	5
6	Fourier transformation	6
6.1	Fourier transformation	6
7	Analyse	6
7.1	Analyse af bilmotor	7
7.2	Analyse af vindmøllestøj	11
7.3	Analyse af EKG	13
7.4	Analyse af musik 1	13
7.5	Analyse af musik 2	16
7.6	Analyse af musik 3	19
7.7	Analyse af musik 4	22
7.8	Analyse af musik 5	25

1 Begreber

Når vi går fra analoge signaler til digitale, så finder vi repræsentationer af det kontinuerlige signal. Dette kalder vi samples og betegnes med N . Når vi har flere samples på et signal, betegnes intervallet i mellem samples som T_s , samplingstid. Når vi har samplingstid kan vi indføre samplingsfrekvens, det inverse af samplingstid.

$$f_s = 1/T_s$$

Så snart at vi har T_s , ved vi at vi har med et digitalt signal at gøre.

Ved opsætning af sampletidsaksen, definerer vi først vores sampletæller, n :

$$n = [0 : N - 1]$$

Hvor N er antal samples. Efterfølgende bestemmer vi vores sampletidspunkter, t :

$$t = n * T_s$$

Vi kan nu indføre:

$$x(t_s) = X(n * T_s) = X(n)$$

Vi har altid en grundfrekvens og den kalder vi altid f_0 .

iiiiiii HEAD

2 Aliasering

=====

2.1 Aliasering

~~~~~ origin/master

Alias = et andet navn for noget/tvetydighed. Vi har tre forskellige slags alias:

- Forkert samling - både for mange samples og for få
- Gentagelser
- Spejling (rundt om Nyquist-frekvensen)

Shannons sandheds sætning

$$f_s \geq 2 * f_{\text{øvre}}$$

I praksis er dette aldrig lig med, men skal altid overholdes. Nyquist-frekvens:

$$f_{\text{nyquist}} = \frac{f_s}{2}$$

Altså defineret som halvdelen af samplingsfrekvensen,

$$f_s$$

iiiiiii HEAD

## 3 Envelopes

=====

### 3.1 Envelopes

lllllll origin/master

Envelopes = Amplitude billede over et tidsinterval. Under emnet envelopes har vi to punkter:

- ADSR - Attack, Decay, Sustain, Release
- LFO - Low-Frequency Oscillation

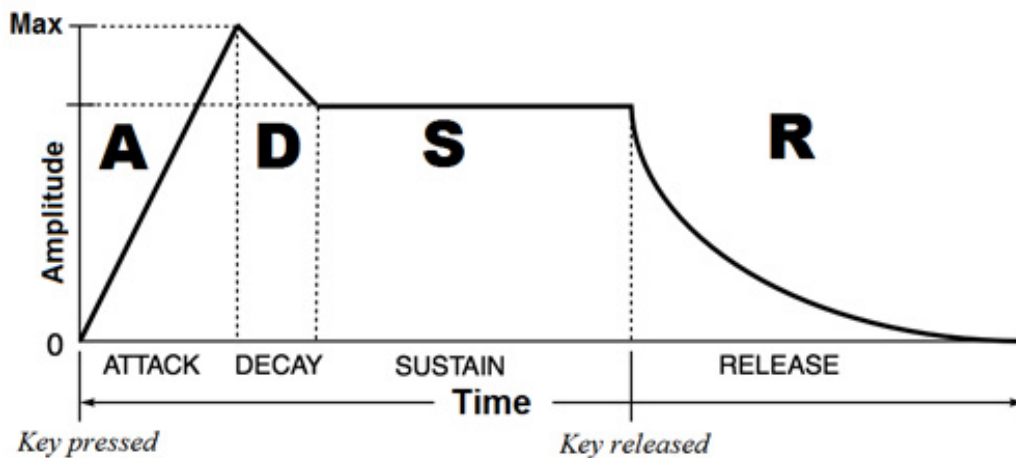
iiiiiii HEAD

## 4 ADSR

=====

## 4.1 ADSR

~~~~~ origin/master Vi starter med ADSR: Her har vi en figur over den basale ADSR:



Som det ses af figuren har vi fire forskellige stadier:

- Attack - Dette er i starten af signalet, hvor f.eks. en streng på en guitar bliver slået. Som vi kan se så stiger grafen kraftigt i Attack-stadiet.
- Decay - Her falder vores graf kraftigt, en smule, da vores streng på guitaren ikke kan holde den kraftige tone i så lang tid. Den skal falde ned til den stationære lyd, hvilket er det næste stadie.
- Sustain - Her holder vores graf et stationært niveau over noget tid. Dvs. at tonen, som vores guitar streng har givet, bliver holdt stabil over længere tid nu - Indtil tonen falder og dør ud (næste stadie).
- Release - Som vi kan se på vores graf dør vores signal ud her. Vores tone har altså holdt så længe den kan og dør nu ud efter det stabile-stadie. Så her falder og til sidst dør tonen ud.

~~~~~ HEAD

## 5 LFO

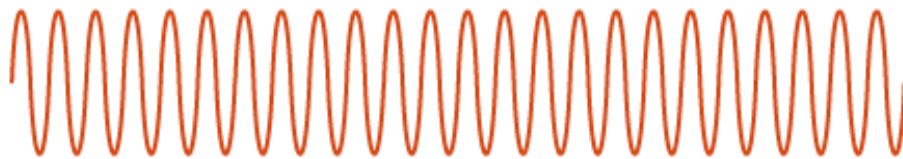
=====

## 5.1 LFO

~~~~~ origin/master

Hvis vi så går videre til LFO:

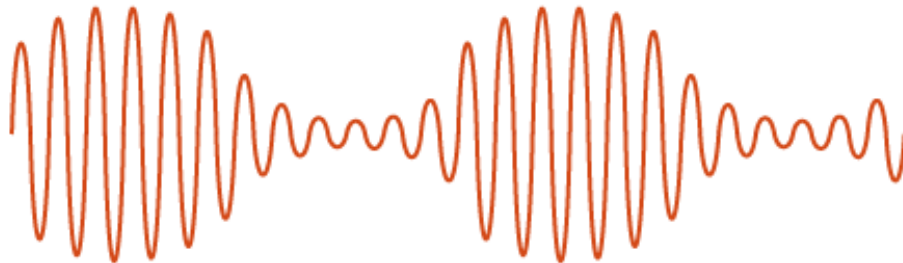
Oscillator's Sound Wave



Low Frequency Oscillator



Resulting Wave with LFO modulating Amp



©2011 ArtificialTunes.tumblr.com

Low-Frequency Oscillation er en anden måde at varierer amplituder på. Dette signal er for det meste under 20 Hz og bruges oftes til lydsignaler. Som navnet af denne metode siger, så bruger man altså kun dette når man har med lavere frekvenser at gøre. Frekvensen man benytter, når man skal have lavet sin sinusbølge, skal altid være lavere end tonen. Når man har fået lavet sin sinus kan vi beregne modeler sinus:

$$S_{mod}(n) = (A_{vo} + 1) * s(n)$$

,hvor $s(n)$ er vores sinus kurve.

Herefter kan vi så beregne modulations graden, som betegnes, f_L .

Det skal så også siges at der er mange forskellige LFO-typer, det er ikke kun sinusser. Der er også firkants-, trekants- og mange andre LFO-typer.

iiiiiii HEAD

6 Fourier transformation

=====

6.1 Fourier transformation

iiiiiii origin/master I DSB har vi nogle forskellige analyse værktøjer. Et af disse er Fourier transformation (DFT). Når vi benytter DFT regner vi med komplekse tal. Definitionen af DFT er betegnet:

$$X(m) = \sum_{n=0}^{N-1} x(n) * e^{-j * \frac{2 * \pi}{N} * m * n}$$

,hvor m = frekvens nummerering.

Det er altså sådan vi går fra tidsdomæne $x(n)$, til frekvensdomæne $X(m)$ - via Fourier transformation, DFT.

Hvis vi så gerne vil tilbage igen, altså fra frekvensdomæne $X(m)$, til tidsdomæne $x(n)$. Så laver vi Invers Fourier Transformation, IDFT. Definitionen af IDFT er betegnet:

$$x(n) = \frac{1}{N} \sum_{m=0}^{N-1} X(m) * e^{j * \frac{2 * \pi}{N} * m * n}$$

Længden er = 1. Herefter er vi altså tilbage i tidsdomænet $x(n)$.

7 Analyse

Når vi laver sådan en analyse her så opstille vi to grafer: En for længden og en for fasen (Bodeplot).

- X-aksen = Frekvens i Hz (opdelt i decader)

- Y-aksen på længde grafen = dB = $20 * \log(10) | X(m) |$
- Y-aksen på fasevinkel grafen = $\angle(X(m))^\circ$

Når vi har fået tegnet vores graf, så har vi det, der hedder Frekvensopløsningen, som er afstanden hver sample:

$$\Delta f = \frac{fs}{N}$$

Det næste vi kan tilføje er Analysefrekvensen:

$$f_{analysis}(m) = m * \Delta f$$

Til sidst har vi Parsevals sætning:

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{m=0}^{N-1} |X(m)|^2$$

Summen af kvadrerede samples i tidsdomænet er lige med summen af kvadrerede samples i frekvensdomænet.

7.1 Analyse af bilmotor

Efter at have fået styr på alle begreberne, går vi nu i gang med bilmotoren. Koden fra MatLab og graferne, der bliver tegnet klipper vi ind, her i dokumentet. Vi starter med at indsætte bilmotor lyden og derefter angiver vi alle de enheder, der skal angives, hvorefter vi opretter grafer, laver DFT og får vores signaler ind. Se nedenunder.

```

Vi indlæser bilmotor lyd (.wav:
[x, fsample] = wavread('ARv6');
N = length(x); —(Antal samples)
Tlength = N/fsample; —(Varighed i sek.)
X = fft(x,N); —( Vi laver DFT)
delta_f = fsample/N; —(Vi finder Δf)
f_axis = [0 : delta_f : fsample - delta_f]; — (Vi opretter en akse med interval

figure(1); clf — (Vores Længde graf)
semilogx(f_axis(1 : 0.5 * end), 20 * log10(abs((2/N) * X(1 : 0.5 * end))))
xlabel('Frekvens i Hz')
```



```

ylabel('Størrelse dB rel. 1 Volt')
title('DFT størrelse (magnitude)')
grid on — (Gøre vores akser pæne)
axis([10 1000 -95 -25])

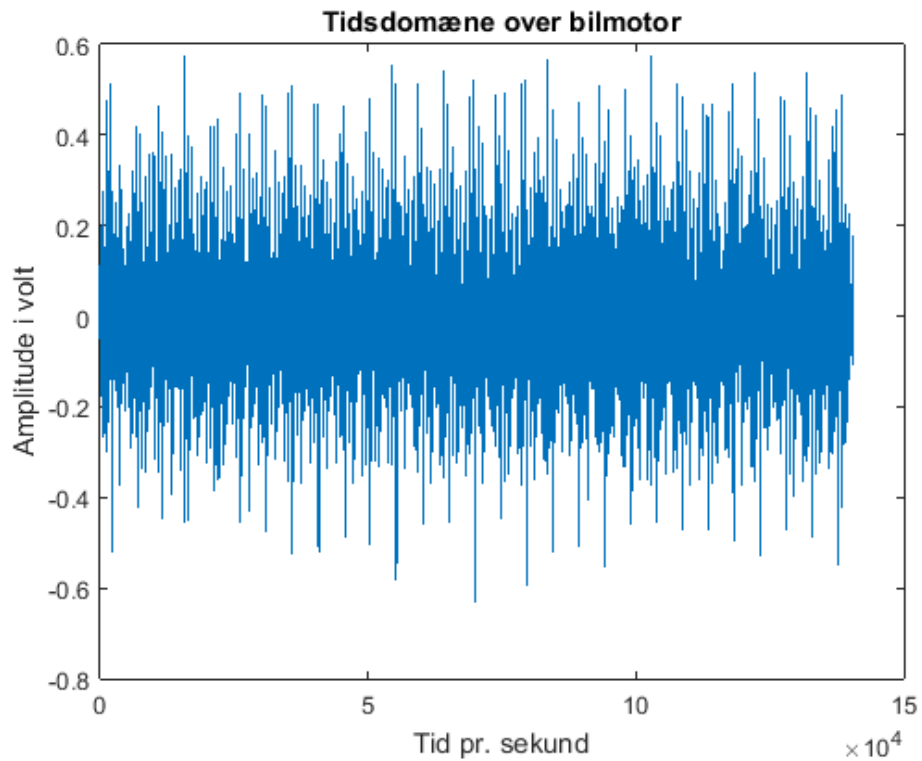
```

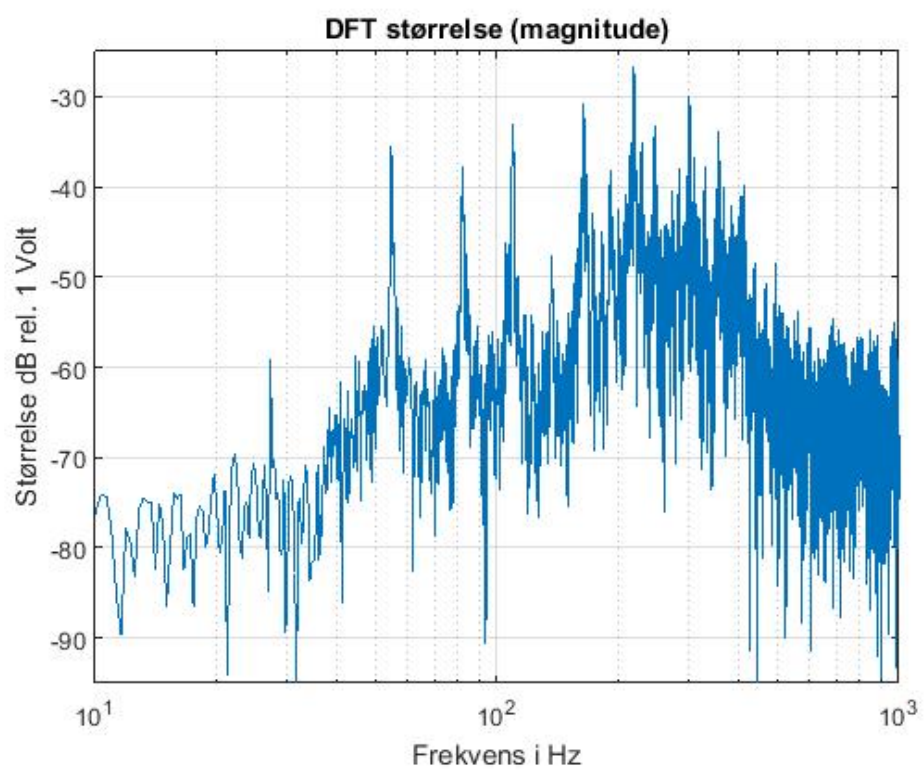
```

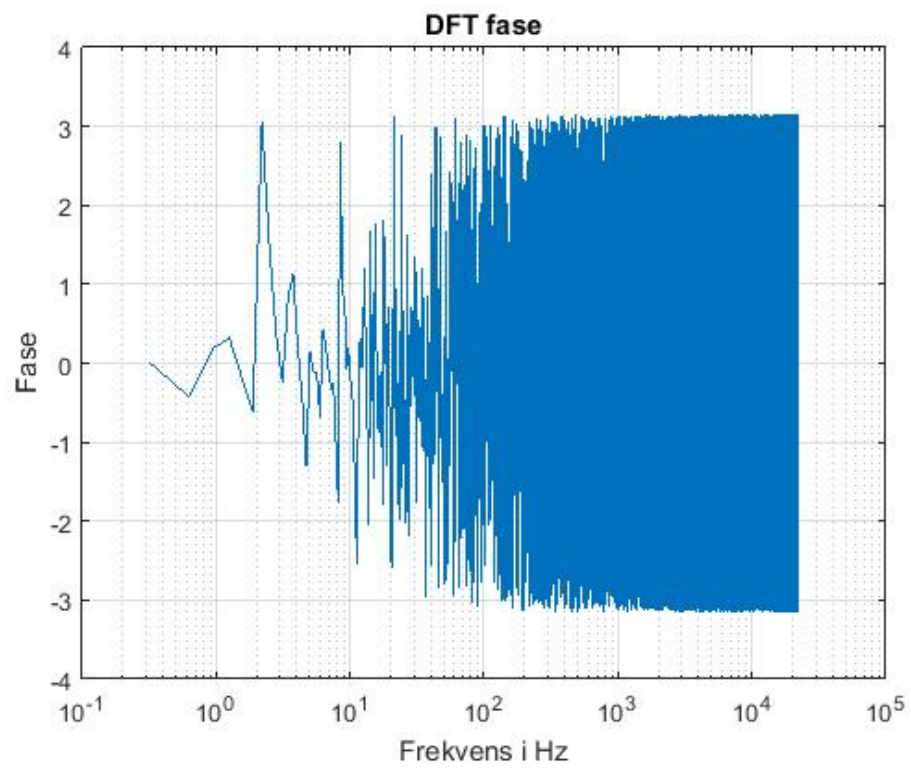
figure(2); clf — (Vores Fase vinkel graf)
semilogx( $f_{axis}(1 : 0.5 * end)$ , ( $angle((2/N) * X(1 : 0.5 * end))$ )))
xlabel('Frekvens i Hz')
ylabel('Fase')
title('DFT fase')
grid on

```

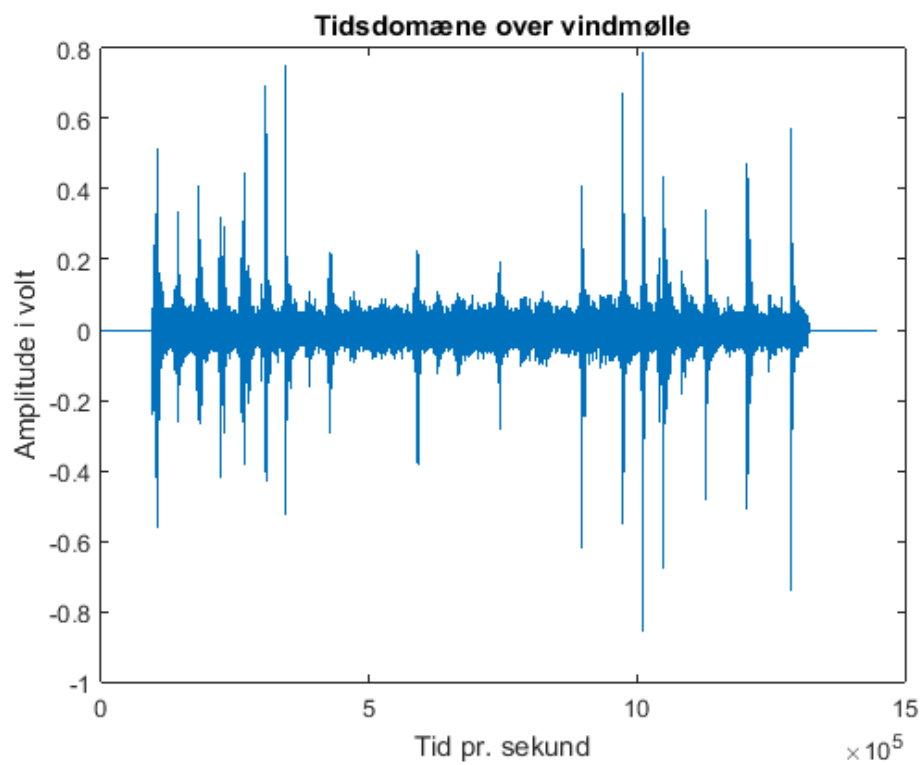
Når vi kører vores kode, udskrives disse grafer:

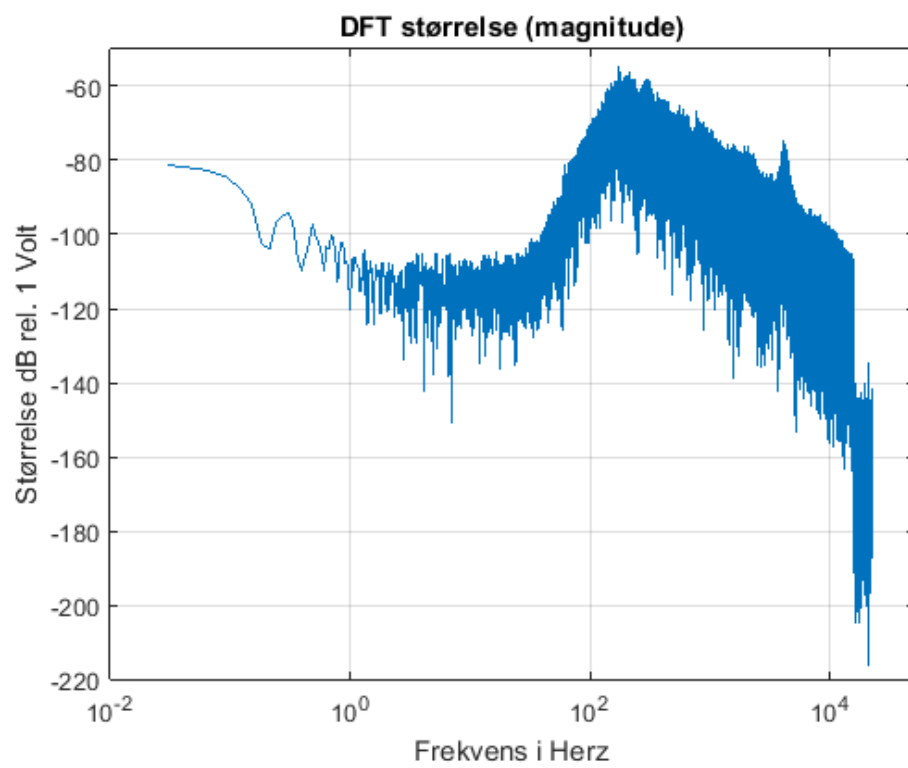


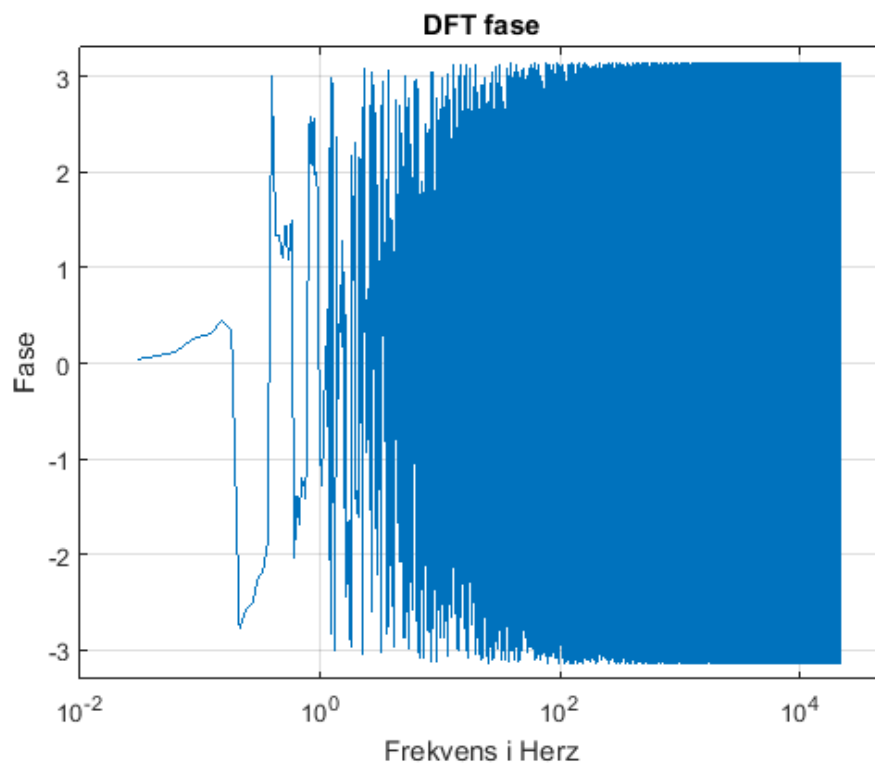




7.2 Analyse af vindmøllestøj





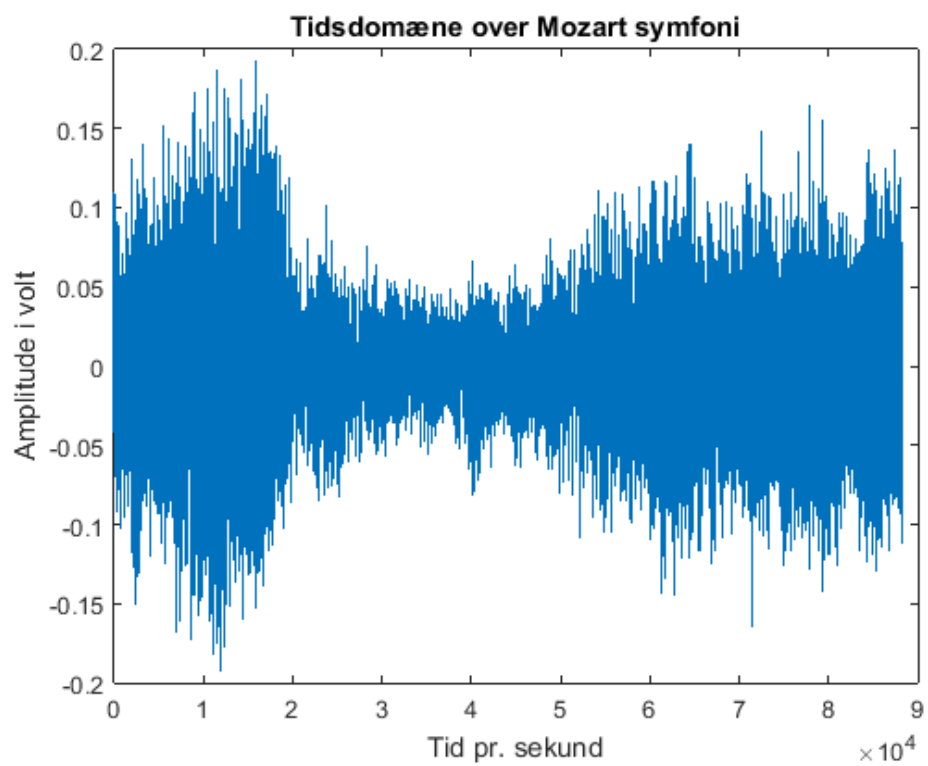


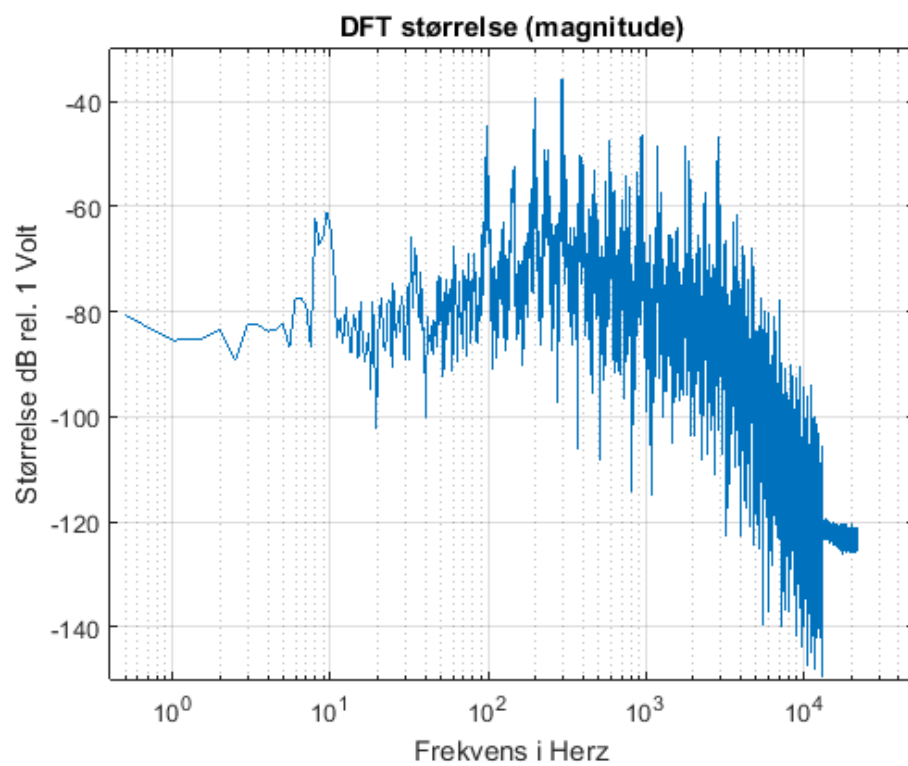
7.3 Analyse af EKG

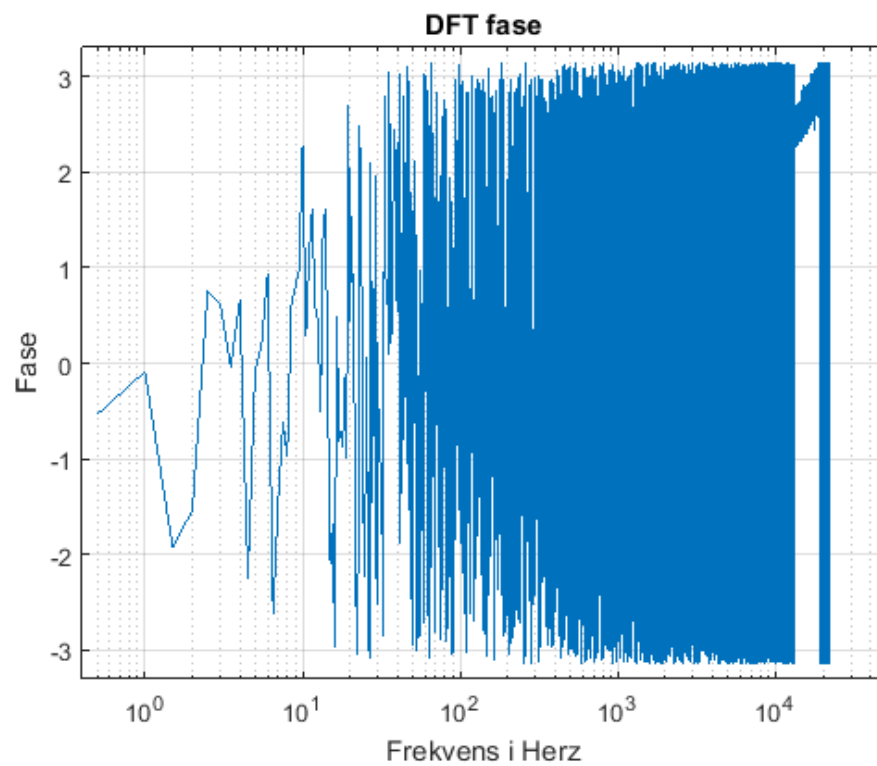
Da vi her ikke har med et lydsignal at gøre, skal vi ikke længe benytte funktionen "semilogx", men blot funktionen "plot".

7.4 Analyse af musik 1

Mozart - 39's Symphony no. 40 - 1st movement.

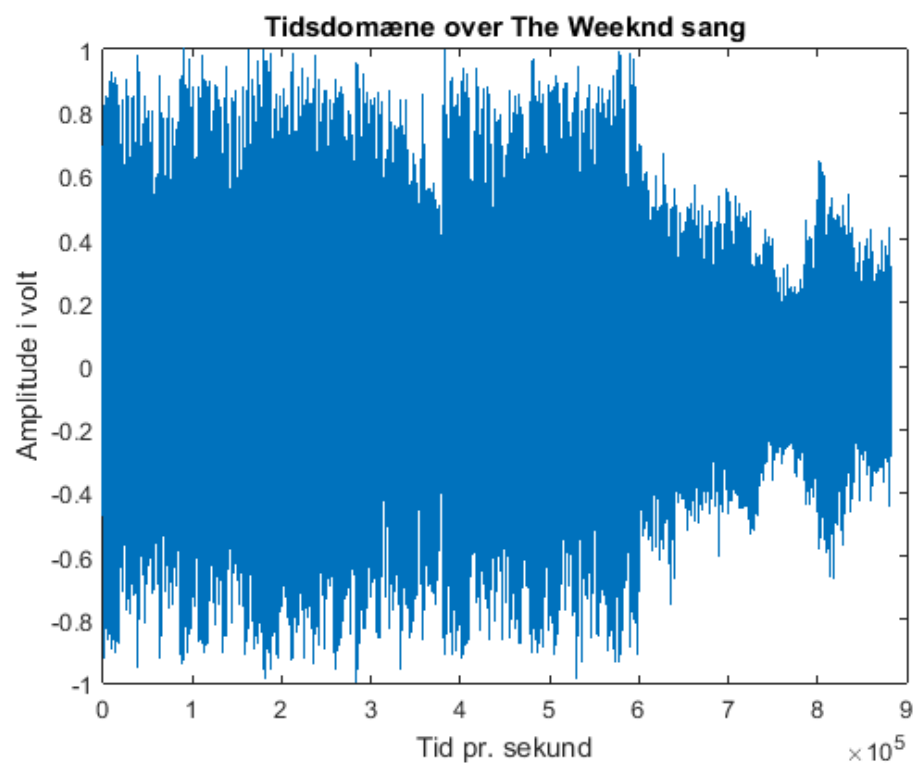


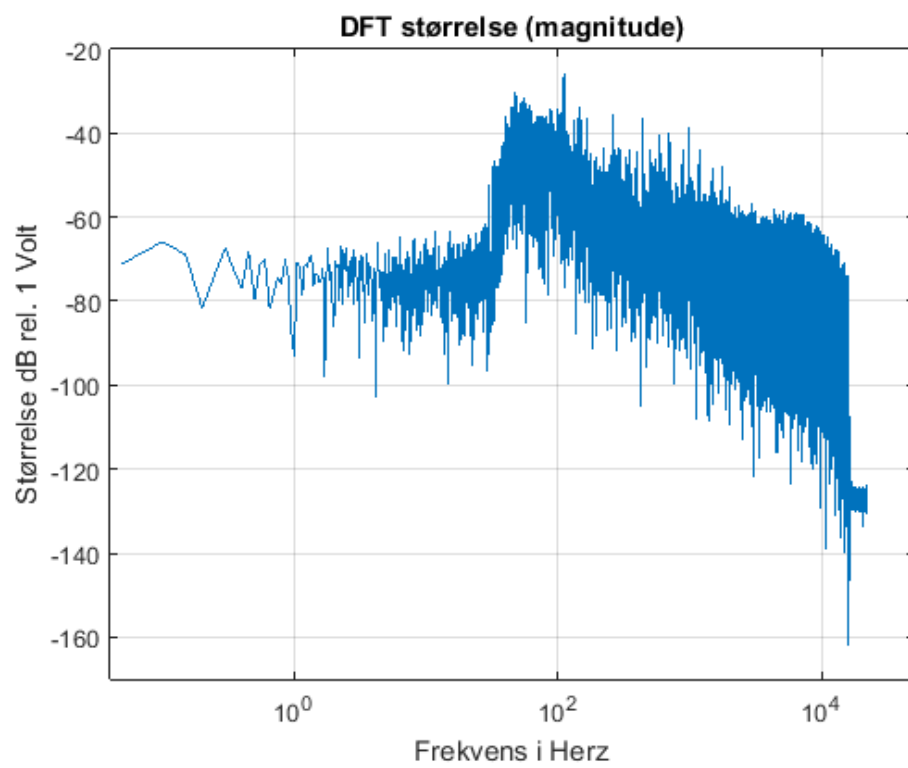


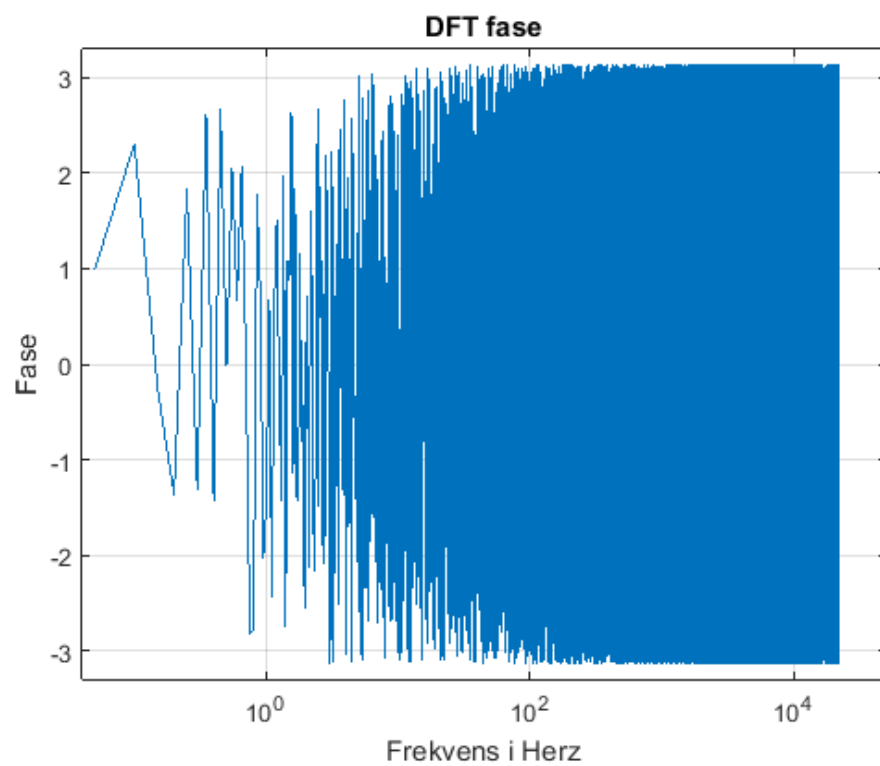


7.5 Analyse af musik 2

The Weeknd - Can't feel my face

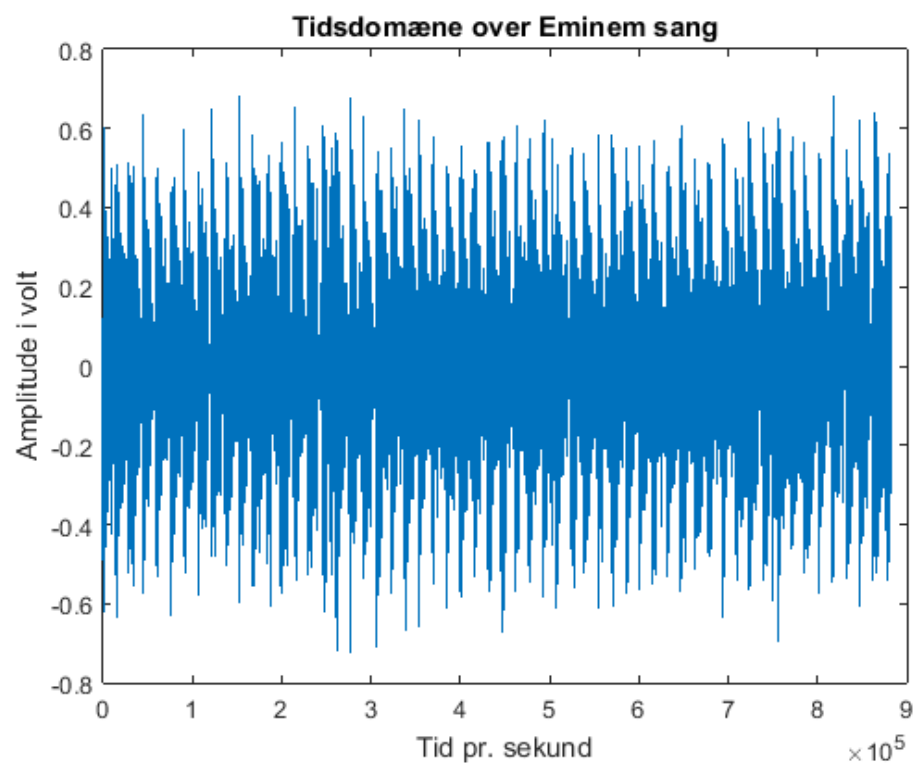


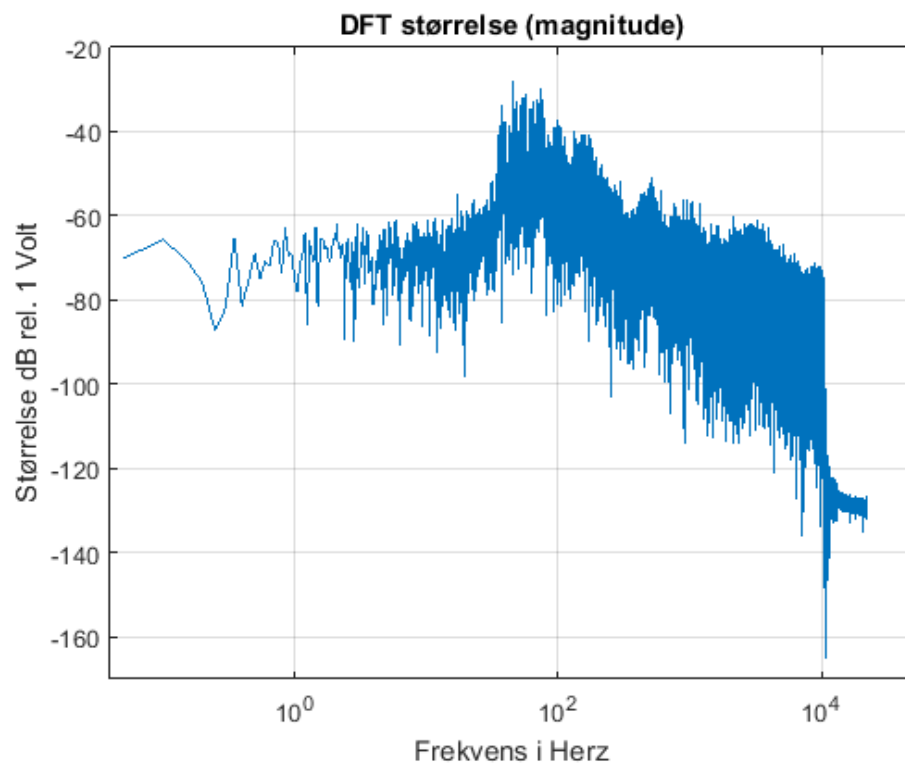


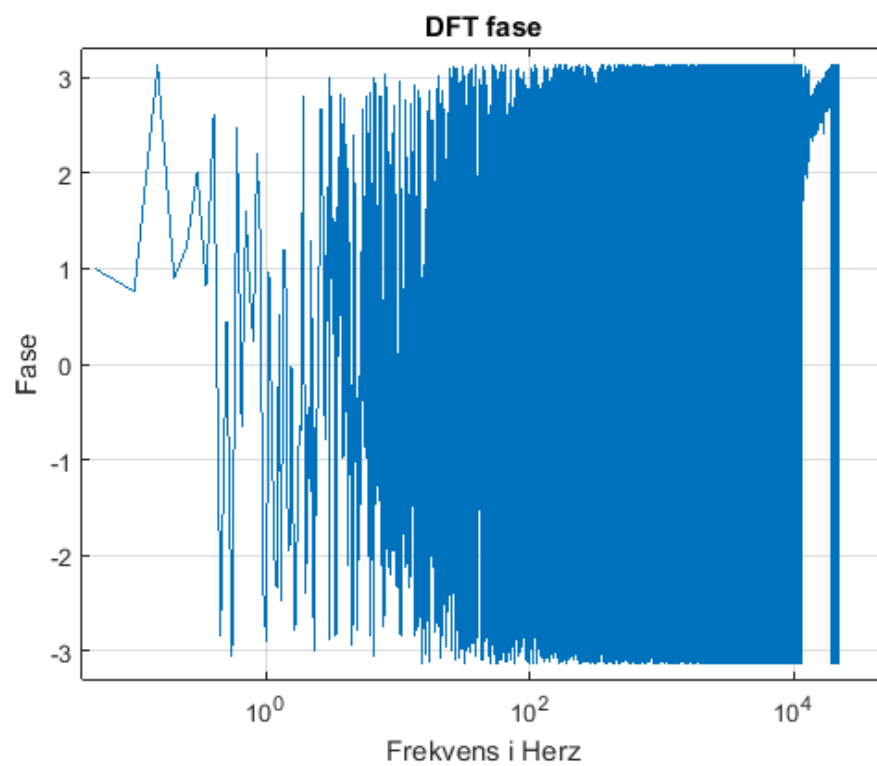


7.6 Analyse af musik 3

Eminem - Lose yourself

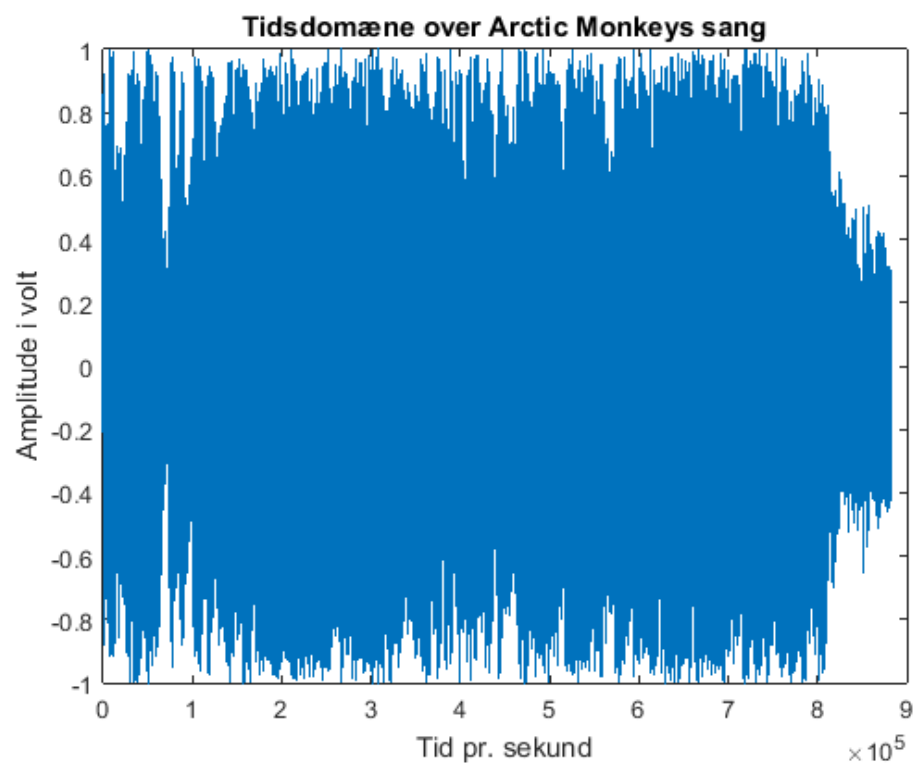


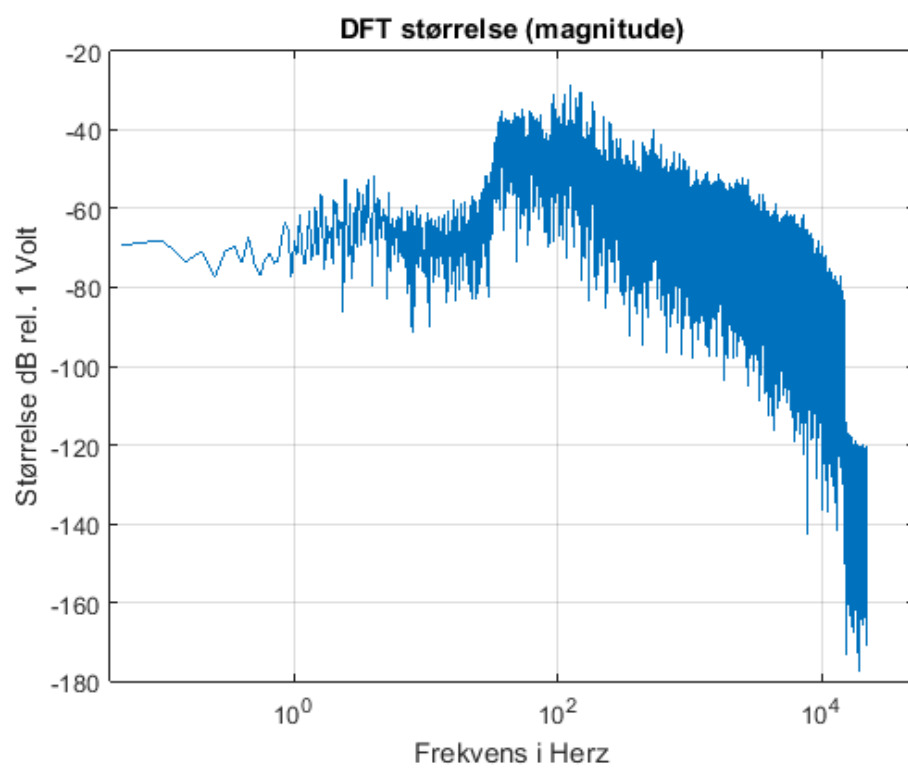


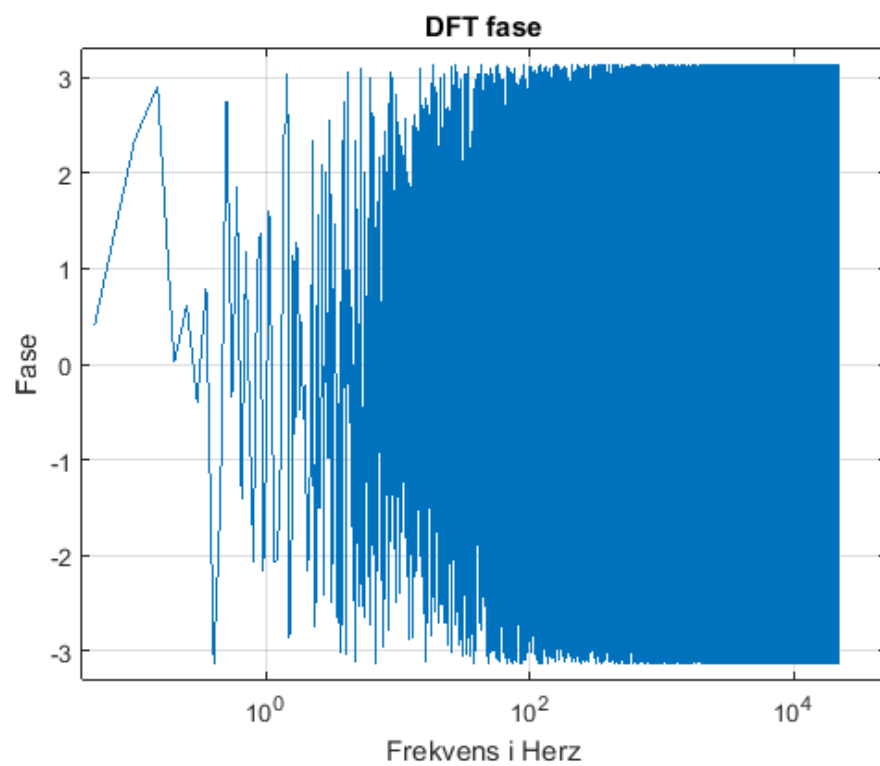


7.7 Analyse af musik 4

Arctic Monkeys - R U Mine







7.8 Analyse af musik 5

Rednex - Cotton Eye Joe

