

AARHUS UNIVERSITET

DSB

SEMESTER 3

Mini-projekt

Studerende:

Mette HAMMER NIELSEN-KUDSK - STUDIENR: 201408391

Martin BANASIK - STUDIENR: 201408398

Finja JETTE RALFS - STUDIENR: 201303659

Oktober 5, 2015



Indholdsfortegnelse

1	Begreber	2
2	Aliasering	2
3	Envelopes	2
4	ADSR	3
5	LFO	4
6	Fourier transformation	4
7	Zero Padding	5
8	Lækage	5
9	Vinduer (Hanning Vinduet)	5
10	Udglatning (Smoothing)	5
11	Analyse	5
11.1	Analyse af bilmotorstøj	6
11.2	Analyse af vindmøllestøj	8
11.3	Analyse af EKG	9
11.4	Analyse af Mozart	10
11.5	Analyse af The Weeknd	11
11.6	Analyse af Eminem	12
11.7	Analyse af Arctic Monkeys	13
11.8	Analyse af Rednex	14
12	Fremtidigt arbejde	15
13	Konklusion	15

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 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 .

2 Aliasering

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,

$$f_s$$

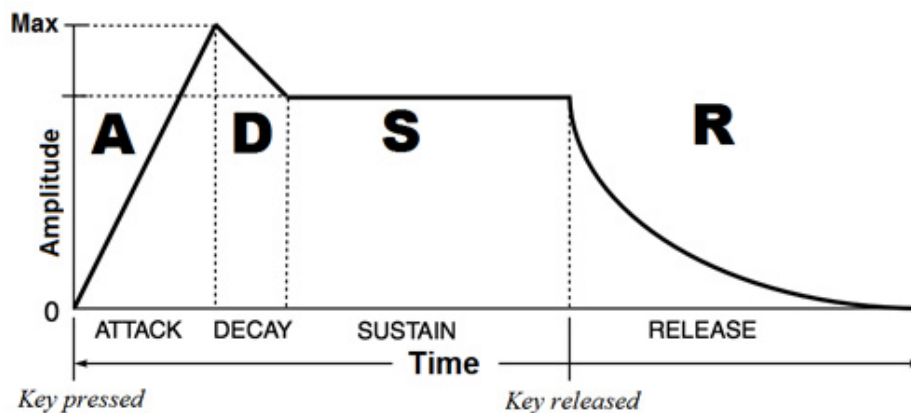
3 Envelopes

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

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

4 ADSR

Vi starter med ADSR: Her har vi en figur over den basale ADSR:

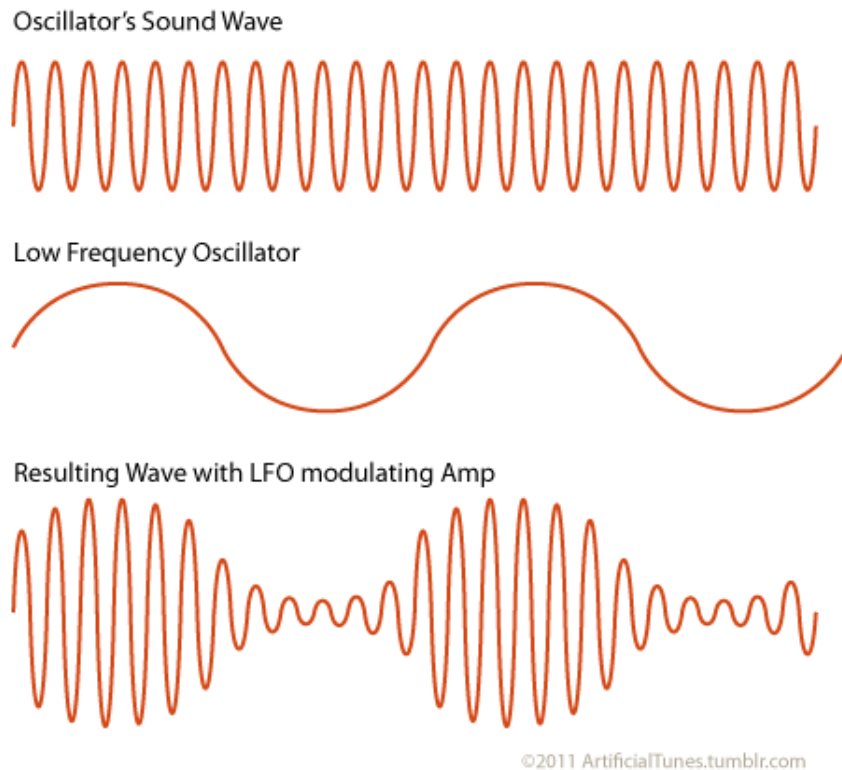


Som det ses af figuren har vi fire forskellige stadier:

- **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 i kort tid, 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 tonen og til sidst dør den ud.

5 LFO

Hvis vi så går videre til LFO:



Low-Frequency Oscillation er en anden måde at varierer amplituder på. Dette signal er for det meste under 20 Hz og bruges oftest 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 sinus-bølge, skal altid være lavere end tonen. Når man har fået lavet sin sinus kan vi beregne modeller 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.

6 Fourier transformation

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 Zero Padding

Ved Zero Padding ligger vi x antal nuller ind i enden af vore signal. På den måde bliver længden af vores signal længere. Vi vinder altså ikke mere information omkring signalet, vi får blot flere ”nul-samples” hæftet på i enden. Dette gør at vi lettere kan fange frekvenskomponenternes opførsel (faktisk dobbelt så god frekvensopløsning) og at længderne på de forskellige signaler bliver ens. I dette miniprojekt synes vi ikke det gav mening, at lave Zero Padding på nogle af vores signaler og derfor er dette ikke blevet gjort.

8 Lækage

Hvis man ikke rammer den rigtige frekvens ville alle ens samples fordele sig ud over hele x -aksen. Dette duer ikke, det bliver udoverskueligt og ulæseligt. Dette kaldes lækage. Lækage er noget vi kan få nedsat ved hjælp af vinduer. (se næste afsnit).

9 Vinduer (Hanning Vinduet)

Indenfor digital signal behandling har vi mange forskellige vinduer, der nedsætter lækage, ved hjælp af forskellige egenskaber. Hanning Vinduet er det vindue som vi benytter os mest af. Hanning Vinduet er energimæssigt mindre, hvilket ikke gør så meget, da vi let kan skrue op for strømmen. Vi mister frekvensopløsning - bliver faktisk næsten halveret, når vi benytter Hanning vinduet. Så vi betaler altså en pris, for at benytte os af Hanning vinduet, for at få vores signal til at se pænt og læseligt ud.

10 Udglatning (Smoothing)

Udglatning fungerer lidt ligesom et lavpasfilter - det fjerner nemlig højfrekvenserne. Ved dette får vi altså pænere grafer, som er lettere læselige. Det er altså en udglatning af vores frekvensspektre, der er tale om.

11 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:

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

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.

11.1 Analyse af bilmotorstøj

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), unwrap(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 — (Tilføjer gridlines)
axis([10 1000 -95 -25])

figure(2); clf — (Vores Fase vinkel graf)
semilogx(f_axis(1 : 0.5 * end), unwrap((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:

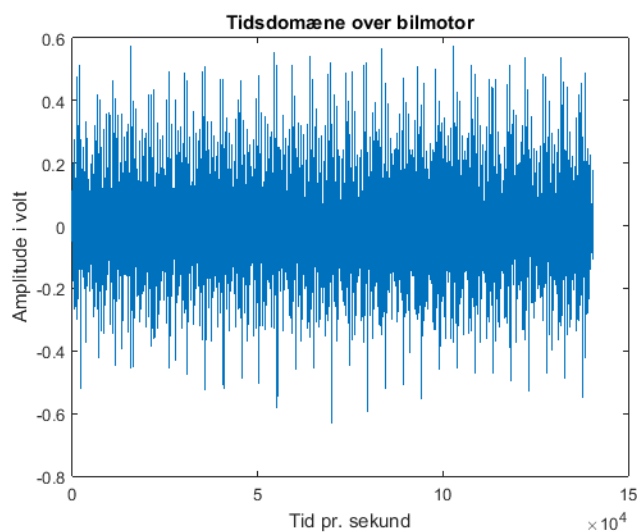


Figure 1: Tidsdomæne over bilmotor

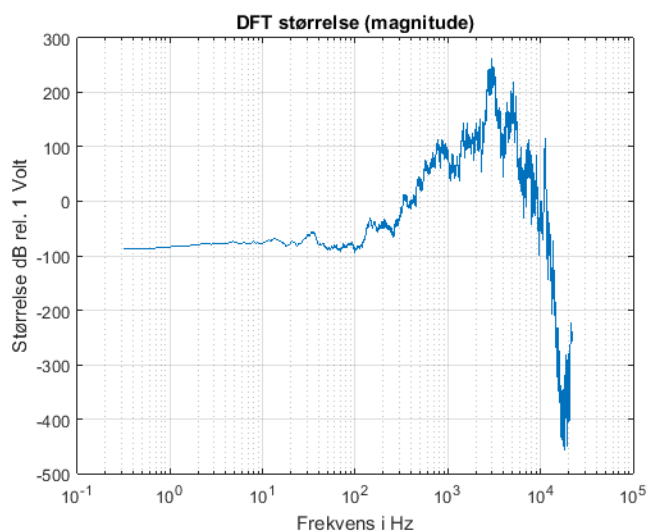


Figure 2: DFT størrelse (magnitude)

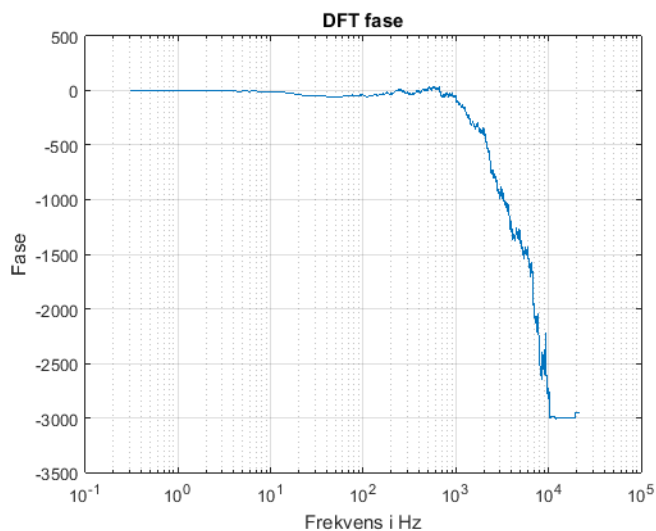


Figure 3: DFT fase

Figur 1 er en graf over tidsdomænet. Her ses hvordan amplituden forholder sig i forhold til tiden. Herefter laver vi DFT og transformerer vores signal over i frekvensdomænet. Her laver vi BodePlot - amplitude- og fasekarakteristik over signalet. Figur 2 er vores amplitudespektre - som det kan ses på figur 2 har vi i bilmotor-støjen mest med lavfrekvenser at gøre. Figur 3 er fasekarakteristikken, der er ikke det store fasedrej her, så i princippet er figure 3 ikke så interessant.

Det menneskelige øre kan høre frekvenser fra 20-20.000 Hz og derfor vil der i vores lydsignal i hvert fald ikke være frekvenser, hverken over eller under. Dette kan ses på alle amplitude-spektrene.

11.2 Analyse af vindmøllestøj

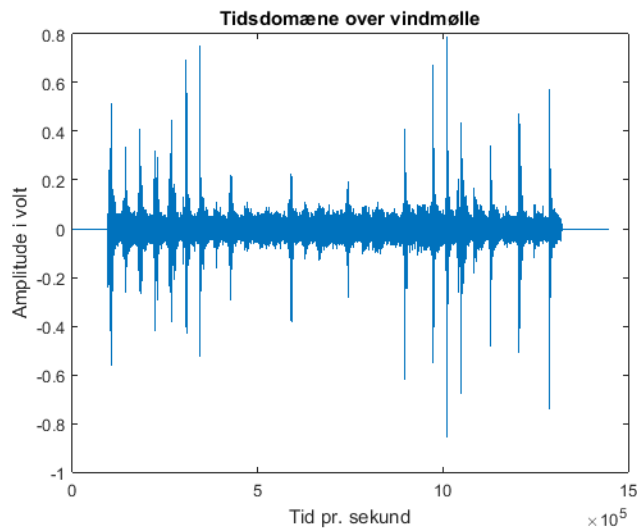


Figure 4: Tidsdomæne over vindmølle

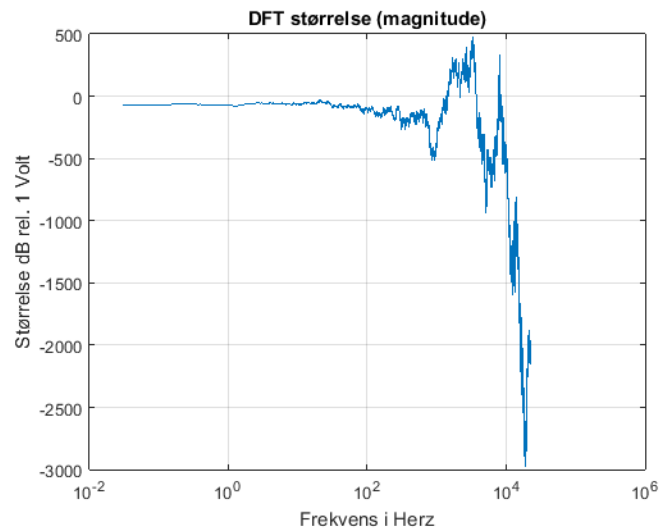


Figure 5: DFT størrelse (magnitude)

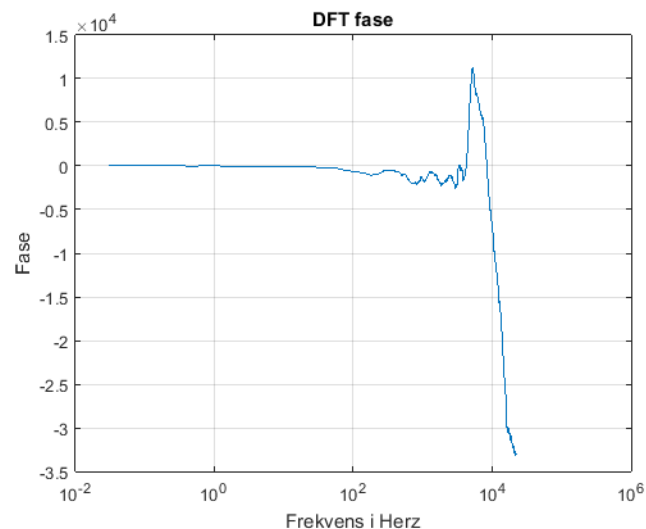


Figure 6: DFT fase

Da vindmøllestøj er meget lave frekvenser, er det her smart at køre vores signal igennem et lavpasfilter, så vi netop får filtreret alle de høje frekvenser, der ikke burde være der, fra. Det kunne f.eks. være snak og fuglekvidder. På figur 6 ses vores fasespektre og her kan vi igen ikke rigtig sige noget om fasen herudfra. Derfor vil vi fra nu og resten af miniprojektet ikke kommentere på fasespektrene - det giver nemlig ingen mening i disse sammenhæng.

11.3 Analyse af EKG

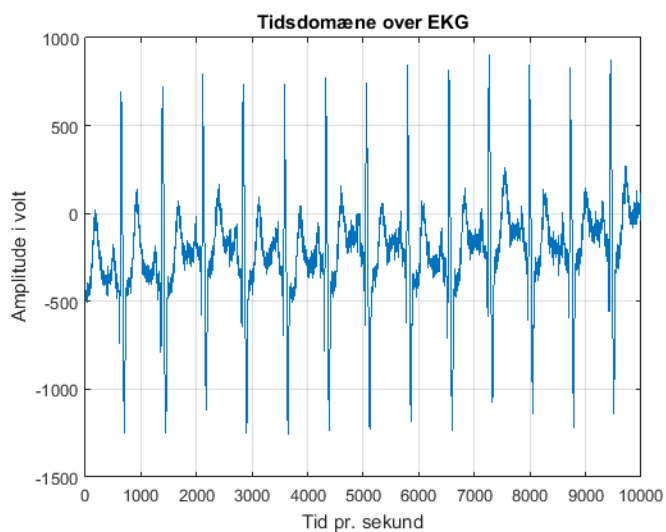


Figure 7: EKG signal i tidsdomænet

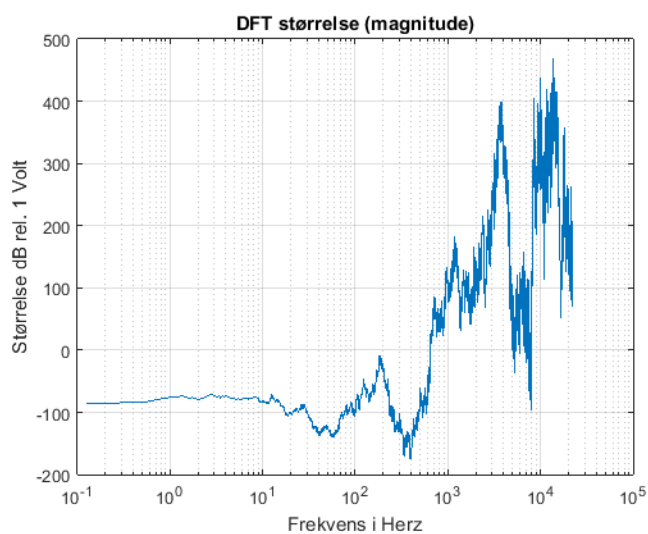


Figure 8: DFT størrelse (magnitude)

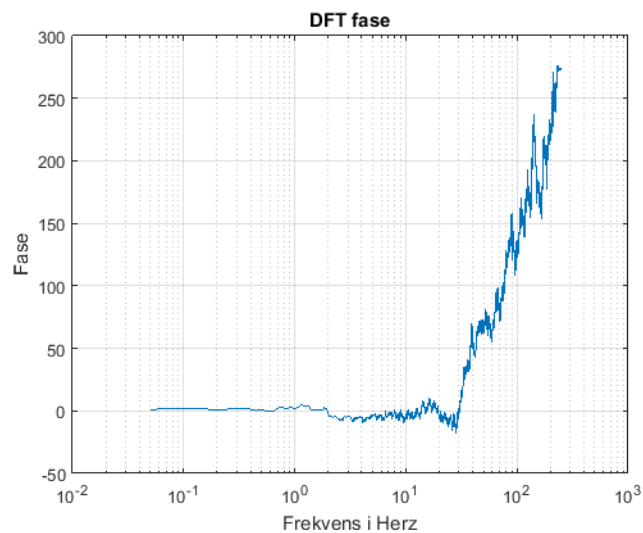


Figure 9: DFT fase

På figur 7 ses tidsdomænet for et EKG signal. Vi kan på dette signal analysere på en hel masse forskellige ting ift. anatomi og patofysiologi. Vi har alle de forskelle segmenter og takker. Lægerne benytter sig af EKG'er, fordi de fleste hjertesygdomme afspejles i ændringer i EKG'et.

11.4 Analyse af Mozart

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

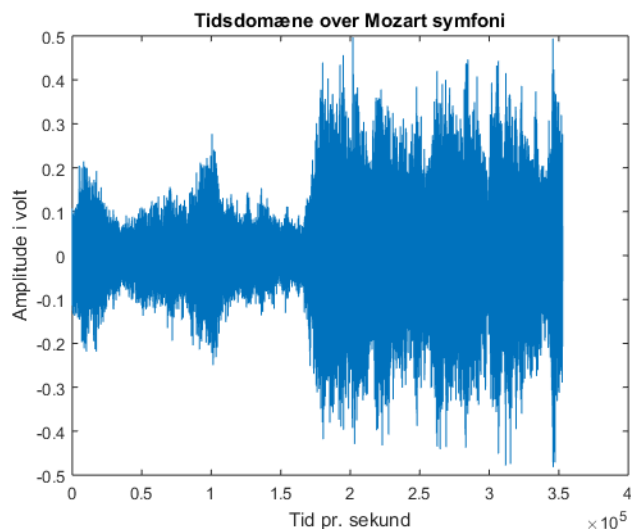


Figure 10: Tidsdomæne over Mozart symfoni

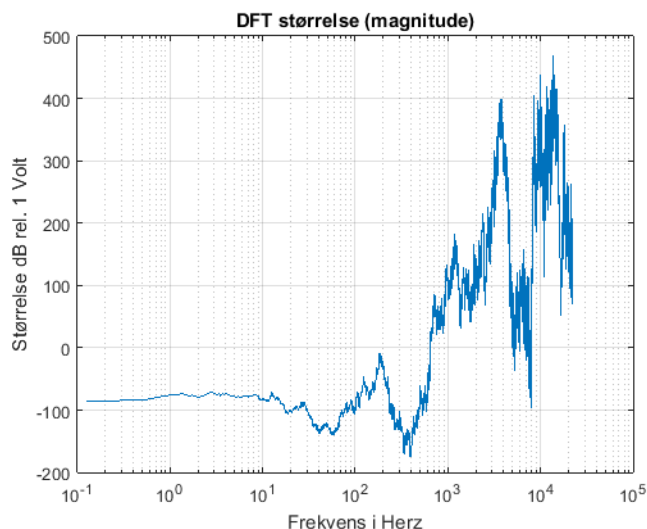


Figure 11: DFT størrelse (magnitude)

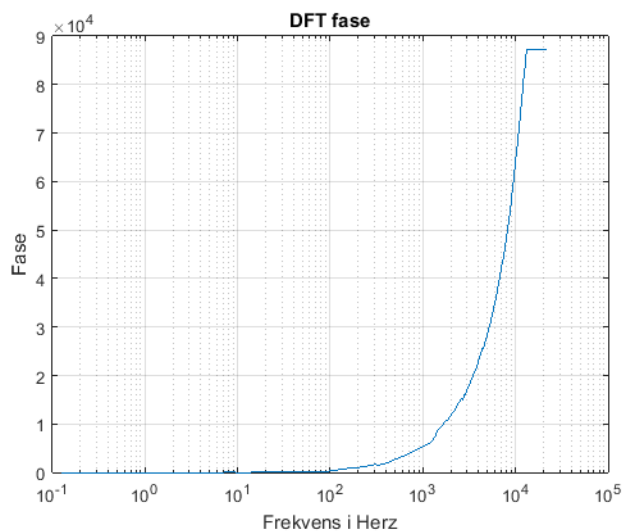


Figure 12: DFT fase

På figur 6 ses Mozarts 39's symfoni i tidsdomænet. Det ses hvordan symfonien udvikler sig over tid. I begyndelsen af udklipet af symfonien er der få instrumenter, der spiller mezzo-piano, så amplituden bliver ikke særlig høj, men efter ca. 1.8 sekunder kommer der flere instrumenter på og de spiller pludselig forte, derfor bliver amplituden højere, hvilket kan ses på figur 6. På figur 7 ses amplitude-spektret i frekvensdomænet. Figur 8 er fasespektret.

11.5 Analyse af The Weeknd

The Weeknd - Can't feel my face

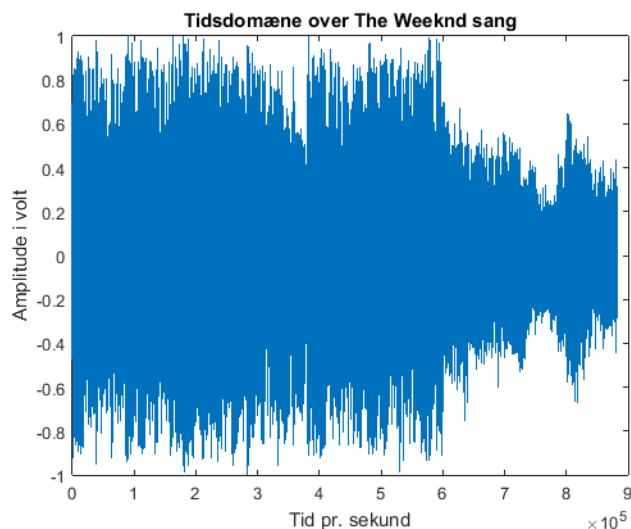


Figure 13: Tidsdomæne over The Weeknd sang

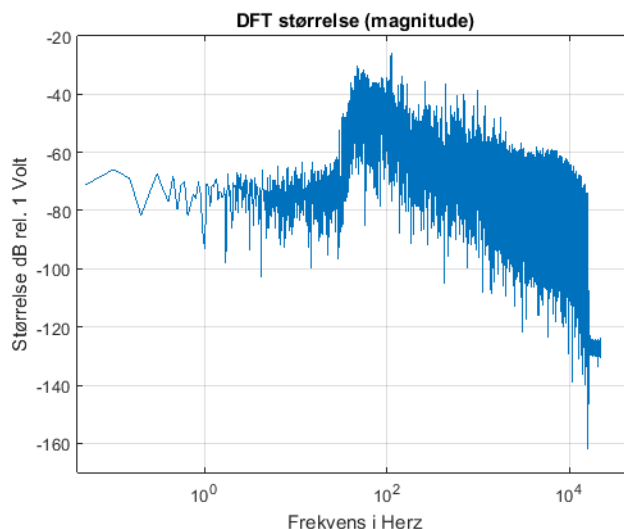


Figure 14: DFT størrelse (magnitude)

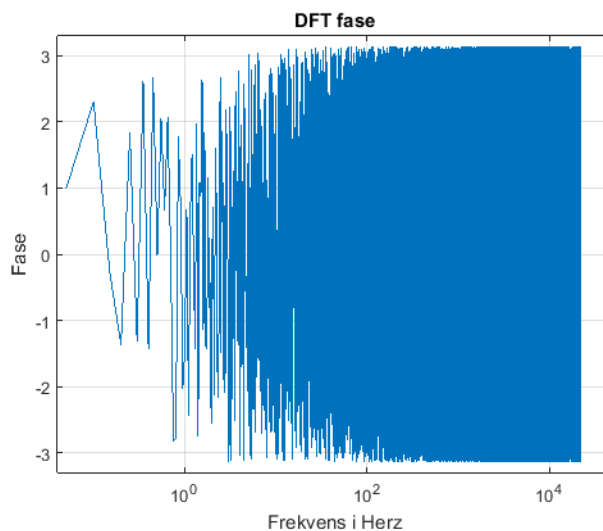


Figure 15: DFT fase

For at se funktionen af "unwrap" i MatLab har vi i denne sang valgt ikke at unwrappe. Så her ser vores signaler meget mere forvirrende ud og det er ikke ligeså let at læse. Vi har lige ramt ind i omkvædet af sangen, hvor der er rigtig meget gang i den - efter omkvædet stilner det lidt af igen. Dette kan ses på figur 9, som er vores signal i tidsdomænet. Denne sang er rigtig sjov, for The Weeknd kan komme rigtig højt op og det giver os mulighed for at se en hel del høje frekvenser på vores figur 10.

11.6 Analyse af Eminem

Eminem - Lose yourself

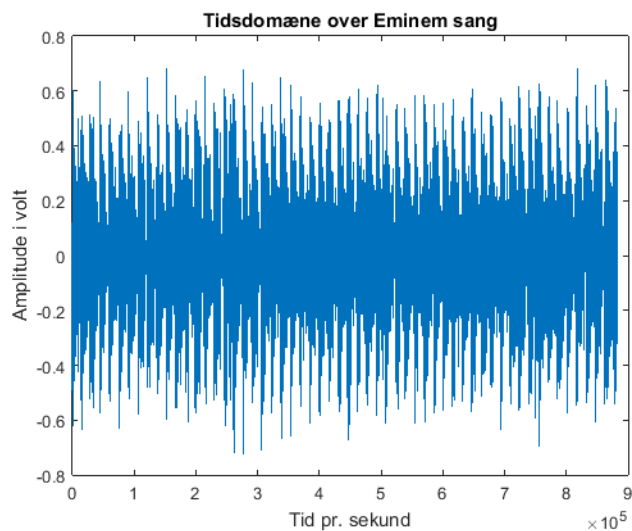


Figure 16: Tidsdomæne over Eminem sang

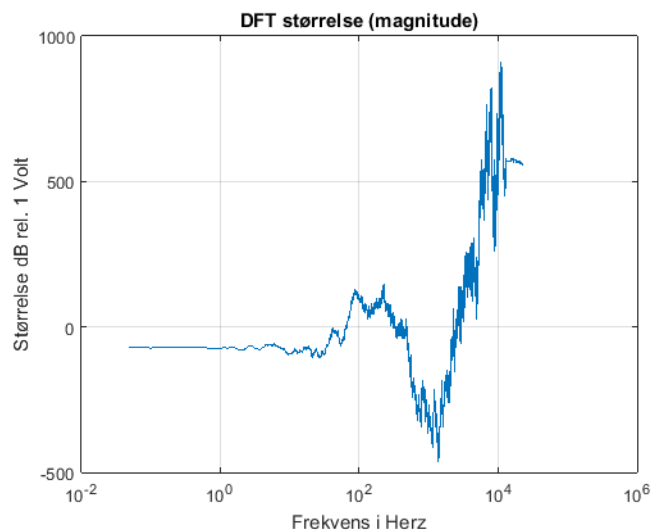


Figure 17: DFT størrelse (magnitude)

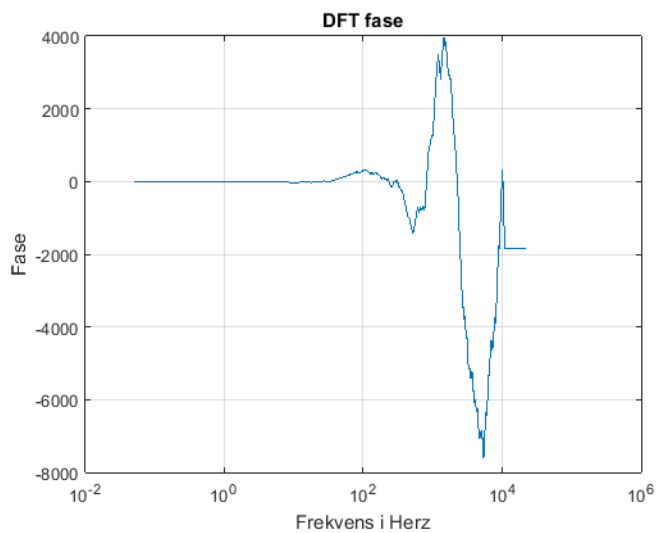


Figure 18: DFT fase

11.7 Analyse af Arctic Monkeys

Arctic Monkeys - R U Mine

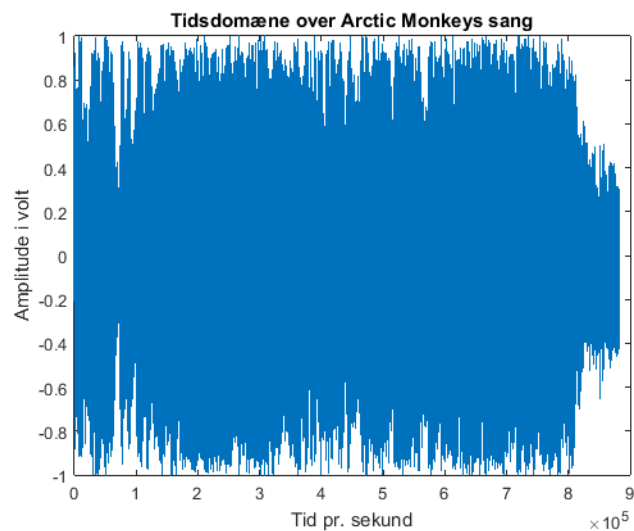


Figure 19: Tidsdomaæne over Artic Monkeys

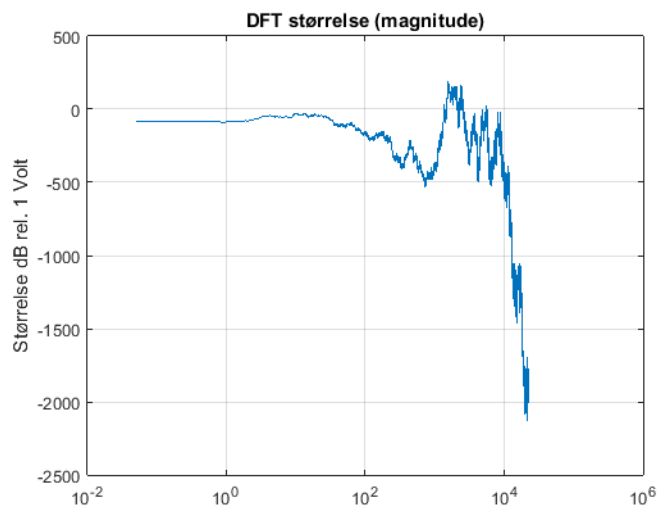


Figure 20: DFT størrelse (magnitude)

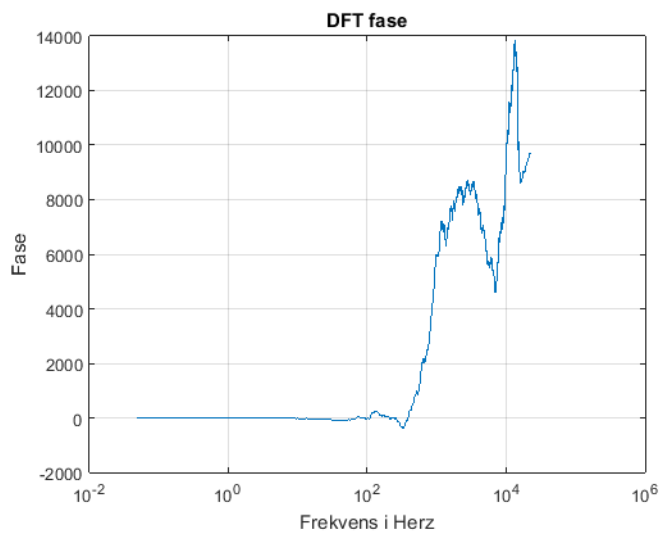


Figure 21: DFT fase

11.8 Analyse af Rednex

Rednex - Cotton Eye Joe

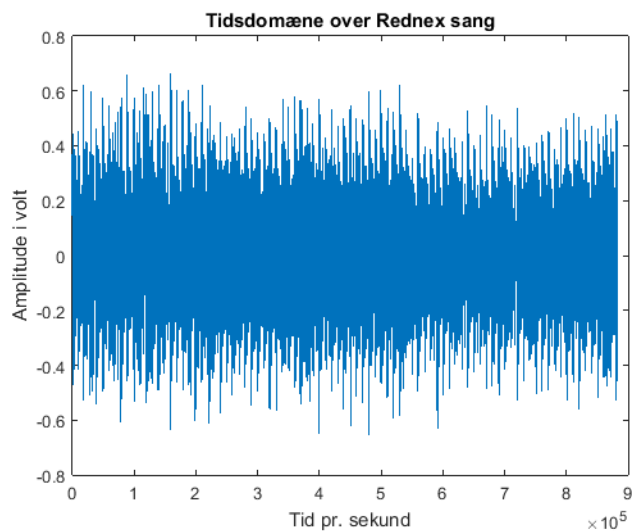


Figure 22: Tidsdomæne over Rednex

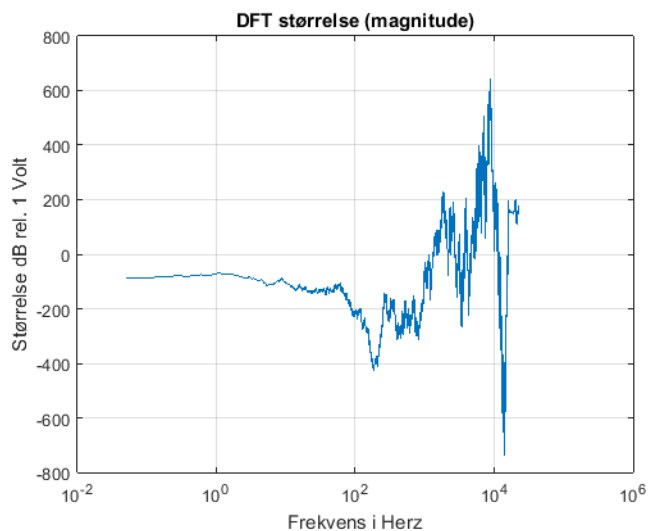


Figure 23: DFT størrelse (magnitude)

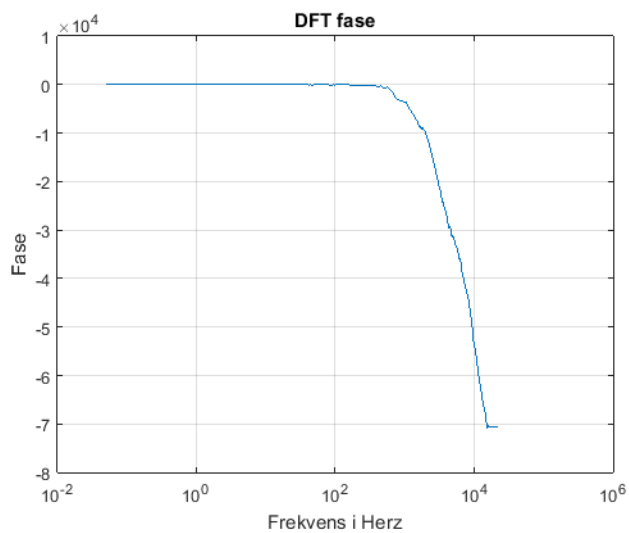


Figure 24: DFT fase

12 Fremtidigt arbejde

Vi ville rigtig gerne havde lavet Hanning vindue på vores vindmøllesignal, da vi her kunne havde fået skåret alle de baggrundsfrekvenser, der er, væk. Dette kunne vi desværre ikke finde ud af, da længderne ikke passede sammen. Vi kunne simpelthen ikke finde ud af at få længderne til at fungere, derfor fik vi ikke lavet vores Hanning vindue - det vil vi arbejde videre på og lære til næste miniprojekt. Vi ville enormt gerne havde lavet udglatning op opløsning, så vi havde kunnet få vores grafer til at se rigtig flotte ud. Igen havde vi problemer med at få længderne til at passe sammen, så det fungerede ikke. Vi har da lært de to MatLab funktioner "Smooth" og "HANN" at kende og ved hvad de ville havde gjort ved vores graf osv. - det var bare en skam at vi ikke kunne få det til at fungere (finder vi ud af til næste miniprojekt). Vi ville med en graf også havde vist at vi har forstået Zero Padding, men det samme, med uoverensstemmelse af længderne, skete her. Vi mente ikke at det gav nogle mening, at lave Zero Padding i dette projekt og tænkte ikke videre over det. Selvfølgelig skal vi også sætte os ind i det og kan til næste gang også benytte os af MatLab funktionen " $S1 = [S \text{ Zeros}(1, P)]$ (P = antal nuller, der ønskes)". Dette er fremtidigt arbejde.

13 Konklusion

Igennem dette miniprojekt har vi lært alle de forskellige MatLab funktioner meget bedre at kende. F.eks. er vi nu helt sikre på hvordan man kommer fra tidsdomænet til frekvensdomænet og i hvilket domæne man laver de forskellige funktioner. Ved er nu klare over hvordan vi laver Zero Padding, Smoothing, Hanning vinduet osv. Vi har lært alle begreberne fra forelæsningerne bedre at kende - f.eks. ved vi nu hvad Aliasering, Envelopes og Lækage betyder i et signal. Vi kan konkluderer at vi ud fra amplitude-spektret for et signal kan analysere utrolig meget omkring en lyd, dog siger fasespektret ikke noget vi kan bruge.