E3DSB miniprojekt 1 - Tidsdomæneanalyse

Janus Bo Andersen [1]

15. september 2019

[1] ja67494@post.au.dk

Indhold

1. Indledning

Dette første miniprojekt i E3DSB behandler tre lydsignaler med analyser i tidsdomænet. Opgaven er løst individuelt. Dette dokument er genereret i Matlab med en XSL-template. Matlab-kode og template findes på `https://github.com/janusboandersen/E3DSB`. Følgende lydklip benyttes

Signal	Skæring	Genre	Samplingsfrekv.
s_1	Spit Out the Bone	Thrash-metal	44.1 kHz
s_2	The Wayfaring Stranger	Bluegrass	96 kHz
s_3	Svanesøen	Klassisk	44.1 kHz

Tabel 1.1: 3 signaler behandlet i analysen

2. Analyser

Før analyser ryddes der op i `Workspace`.

```
clc; clear all; close all;
```

2.1 Afspilning af lydklip

Filen med signaler åbnes med `load`. Signaler kan afspilles med `soundsc(signal, fs)`. Samplingsfrekvensen f_s sættes efter værdi i tabel 1.1. Samplingfrekvenser for de tre signaler er inkluderet i `.mat`-filen.

```
load('miniprojekt1_lydklip.mat');    % åbn .mat-fil
soundsc(s1, fs_s1);                  % playback startes sådan her
clear('sound');                      % stop playback
```

2.2 Bestemmelse af antal samples

Et sample er en værdi, eller sæt af værdier, fra et givent punkt i tid. Alle tre signaler er i stereo, så hver sample har to værdier.

Signalerne er repræsenteret som $N \times K$-matricer. Antallet af rækker, N, repræsenterer antallet af samples. N kan findes med `length(matrix)`. Antallet af søljer, K, er antallet af kanaler (værdier per sample). Samlet antal af værdier i matricen er NK, antaget at ingen er `NaN`.

N og K kan bestemmes på en gang via `[N, K] = size(matrix)`. Vi kan også bare benytte, at vi ved, at der er to kanaler, så $K = 2$.

Data samles i en tabel. Den kan udvides med signalernes afspilningstider.

Der er altså fx 1,323 millioner samples i signal s_1. Signal s_2, som dog har højere samplingsfrekvens, har 2,5 gange flere samples. De tre lydklip har afspilningstider på mellem 30 og 35 sek.

```
signaler = {'s1'; 's2'; 's3'};
N = [length(s1); length(s2); length(s3)];          % antal samples
K = [2; 2; 2];                                     % antal kanaler
M = N.*K;                                          % antal værdier
samplingsfrek = [fs_s1; fs_s2; fs_s3];             % f_s fra .mat-fil
tid = N./samplingsfrek;                            % spilletid i sek.
T = table(signaler, N, K, M, samplingsfrek, tid)   % vis en datatabel
```

```
T =

  3×6 table

    signaler        N         K        M         samplingsfrek    tid
    --------    ----------    -    ----------    -------------    ---

      's1'       1.323e+06    2     2.646e+06        44100        30
      's2'        3.36e+06    2      6.72e+06        96000        35
      's3'      1.4112e+06    2    2.8224e+06        44100        32
```

2.3 Plot af signal

Når vi skal plotte signalerne med en tidsakse i sekunder, bruges det at $t = nT_s = \frac{n}{f_s}$. Man bør plotte et diskret signal i et stem-diagram, dvs. `stem`-funktionen, men for at få noget mindre gnidret at se på, bruges `plot`. Til at danne akserne bruges Matlabs :-operator.

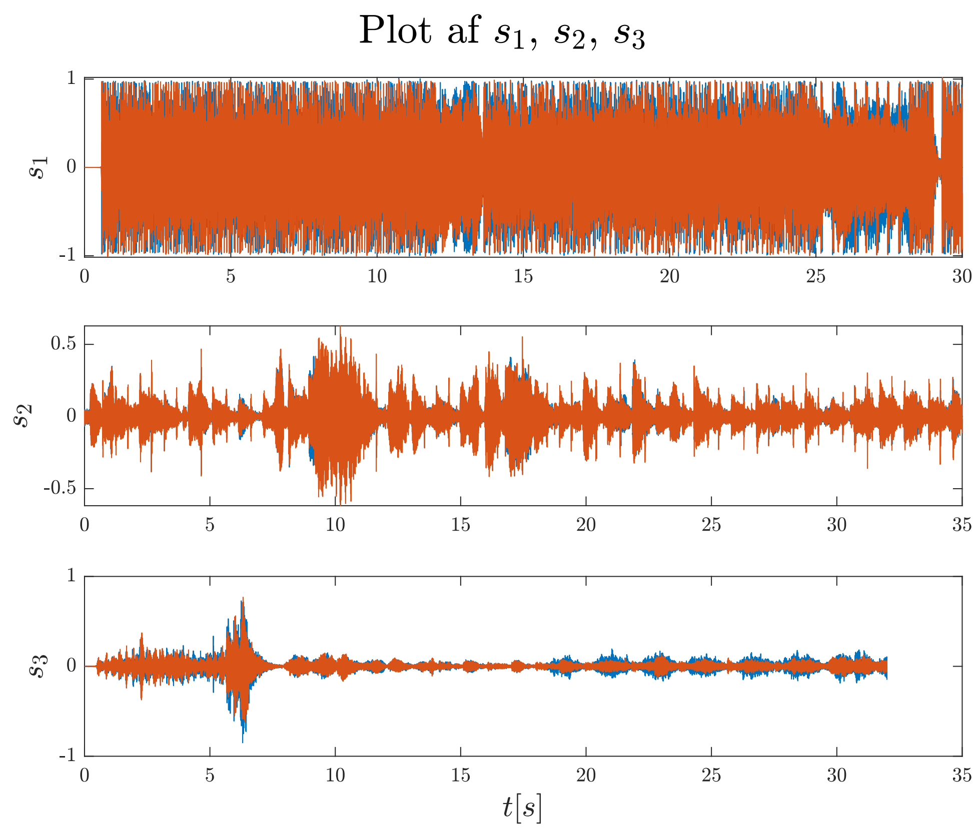
```
t1 = [0:1:N(1)-1]'/fs_s1;                      % søjlevektor, dog ej vigtigt
t2 = [0:1:N(2)-1]'/fs_s2;
t3 = [0:1:N(3)-1]'/fs_s3;

% der gøres lidt arbejde for at få et rent latex layout
set(groot, 'defaultAxesTickLabelInterpreter','Latex');
set(groot, 'defaultLegendInterpreter','Latex');
set(groot, 'defaultTextInterpreter','Latex');

figure(1)                                      % figur med 3 stablede subplots
```

```
subplot(3,1,1);
plot(t1,s1);                                  % signal 1
ylabel('$s_1$','Interpreter','Latex', 'FontSize', 15);
subplot(3,1,2);
plot(t2,s2);                                  % signal 2
ylabel('$s_2$','Interpreter','Latex', 'FontSize', 15);
subplot(3,1,3);
plot(t3,s3);                                  % signal 3
ylabel('$s_3$','Interpreter','Latex', 'FontSize', 15);
xlabel('$t [s]$','Interpreter','Latex', 'FontSize', 15);

% og en titel for hele diagrammet
sgtitle('Plot af $s_1$, $s_2$, $s_3$', 'Interpreter', 'Latex', 'FontSize',
    20);
```



Plots viser ret tydeligt store forskelle i lydklippenes “intensitet”. Forstået på den måde, at lydklippet med thrash-metal har en gennemgående høj amplitude (opleves som “højt”), i modsætning til fx det klassiske stykke. Nogle ville nok bare mene, at plottet over Metallicas nummer ligner “støj” :-).

Næste analyse kan måske give numeriske mål på disse visuelle observationer.

2.4 Min, max, energi og RMS

I dette afsnit beregnes forskellige mål på signalernes lydmæssige "karakter".

Overvejelser: Signalerne er i stereo (2 kanaler / søjler). Hvis vi har et system med to højttalere, giver det mening at betragte kanalerne separat (ikke sammenlagt). Altså, jeg analyserer kanalerne i forlængelse, som en mono serie med $M = 2N$ samples. Denne løsning bruges, fordi det er sådan et menneske med to ører og sæt hovedtelefoner ville opleve signalet :-). Det er også proportionalt til effekt og energiafsættelse i et system med to højttalere.

En sum eller et gennemsnit på tværs af kanalerne ville betyde, at kanaler ude af fase kunne cancellere/eliminere hinanden. Dette ville måske give mening som en simpel konvertering til mono, dvs. vi kunne beregne mål på hvad der ville ske i et simpelt mono-system.

Beregning: Minimum og maksimum findes med hhv. `min()` og `max()`. I tidsdomænet er effekten af et signal proportionalt til kvadratet på amplituden. For en sekvens $x(n) \in \mathbb{R}$, $n = 0, \dots, N-1$ defineres effekten som $x_{pwr}(n) = |x(n)|^2 = x(n)^2$. I diskret tid er energien i signalet summen af "effekterne", dvs. $E_x = \sum_{n=0}^{N-1} |x(n)|^2$. Dette er også det indre produkt $\langle x(n), x(n) \rangle$. RMS-værdien kan beregnes som kvadratroden af middeleffekten, dvs. $x_{RMS} = \sqrt{\frac{1}{N} E_x}$. Nu regnes alle serier så blot over $n = 0, \dots, 2(N-1)$ jf. overvejelserne ovenfor.

```
s1_vec = reshape(s1,[],1);      % Reshape matricer til søjlevektorer:
s2_vec = reshape(s2,[],1);      % De har nu hver M = 2N rækker og 1 søjle
s3_vec = reshape(s3,[],1);      % N, M er selvfølgelig forskellige for hver

minima = [min(s1_vec); min(s2_vec); min(s3_vec)];
maxima = [max(s1_vec); max(s2_vec); max(s3_vec)];
energi = [sum(s1_vec.^2); sum(s2_vec.^2); sum(s3_vec.^2)];      % kvadratsum
rms = [energi(1)/M(1); energi(2)/M(2); energi(3)/M(3)].^(1/2); % kv.rod

T = table(signaler, N, M, minima, maxima, energi, rms)          % resultater
```

```
T =

  3×7 table

    signaler        N              M            minima      maxima       energi         rms
    --------    ----------    ----------    --------    -------    ----------    --------

      's1'       1.323e+06     2.646e+06     -1.0166     1.0191    2.5336e+05     0.30944
      's2'        3.36e+06      6.72e+06    -0.61796    0.62791         34641    0.071797
      's3'      1.4112e+06    2.8224e+06    -0.85016    0.76907        5662.2     0.04479
```

Resultaterne (i tabellen) viser det, som plots også illustrerede: Der er mere energi i metal end i klassisk og bluegrass :-) Og højttalerne bliver varmere af at spille Metallica end af Tchaikovsky.

2.5 Venstre vs. højre kanal (for s_1)

2.6 Nedsampling af signal (for s_1)

2.7 Fade-out med envelopes (for s_2)

3. Konklusion