

1 Opgave 1

For at finde antallet af samples for alle 3 signaler, kan jeg ganske simpelt kalde funktionen **length()** på alle samples. **length()** returnerer antallet af elementer i matricen, hvilket dermed vil sige antallet af samples. Koden for at gøre dette kan ses i listing 1

Listing 1: Print af antal samples

```
1 length(y(s1).sample) % = 4213759
2 length(y(s2left).sample) % = 8753617
3 length(y(s2right).sample) % = 8753617
4 length(y(s3).sample) % = 1270957
```

Her fra kommer jeg frem til at s1 indeholder 4.213.759 samples, s2 (left og righth) indeholder 8.753.617 samples og s3 indeholder 1.270.957 samples. Dette lade til at være rigtigt, da jeg kan se s2 f.eks. varer meget længere end s3.

2 Opgave 2

For at plote alle 3 signaler med rigtige labels og titler, bruges funktionerne beskrevet i 2. For at gøre MATLAB koden mere overskuelig, er der brugt et for-loop på min struct, så jeg kan loope igennem hvert signal, og plottet dem med de allerede forudbestemte variable. Alle plots tegnes på figur f1, for overskuelighed.

Listing 2: Plot af s1, s2 og s3

```
1 %% Exercise 2
2 f1 = figure;
3
4 for i = 1:length(y)
5     subplot(2,2,i)
6     plot(y(i).time, y(i).sample)
7     title(y(i).name)
8     xlabel("Time(s)")
9     ylabel("Amplitude(~)")
10 end
```

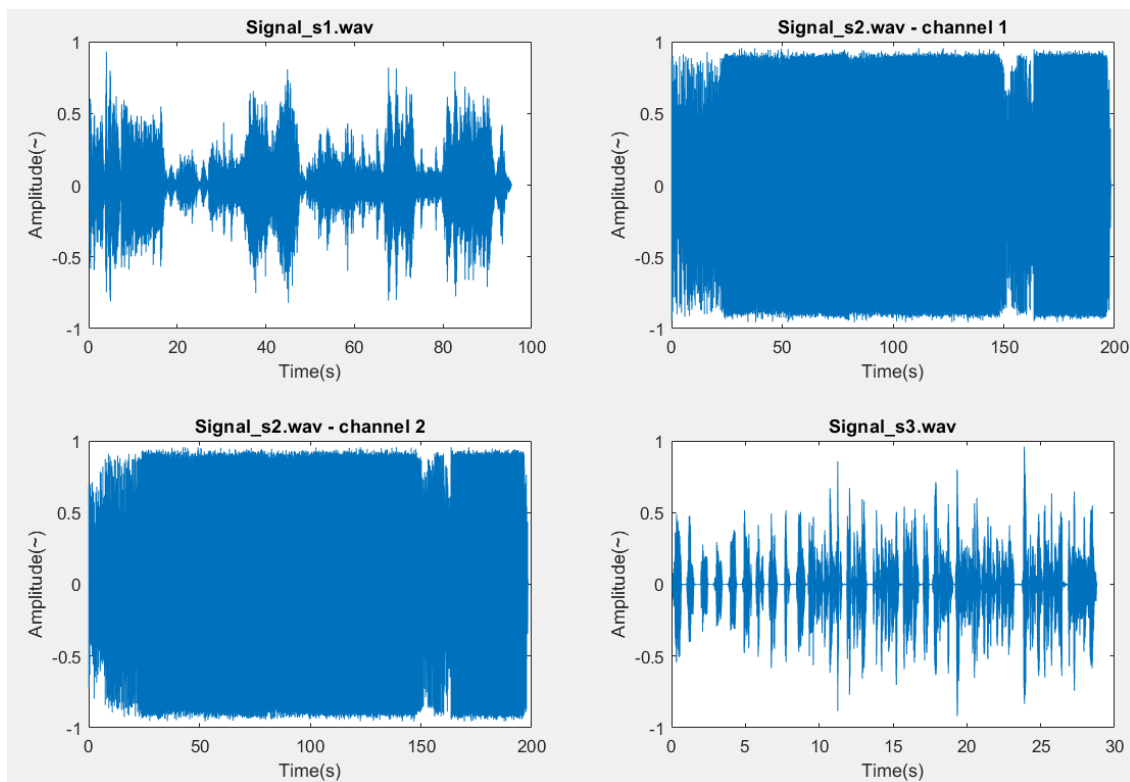


Figure 1: Plot af alle 3 signaler

3 Opgave 3

For at finde max, min, gennemsnit (mean), rms og effekt for alle 3 signaler, kan jeg kalde de forudbestemte værdier, jeg beregnede i listing 2. Et udsnit af hvordan de blev beregnet kan ses på listing 3.

Listing 3: Udsnit af beregning af max, min, gennemsnit (mean), rms og effekt

```

1 for i = 1:length(y)
2   y(i).max = max(y(i).sample);
3   y(i).min = min(y(i).sample);
4   y(i).mean = mean(y(i).sample);
5   y(i).rms = rms(y(i).sample);
6   y(i).effect = sum(y(i).sample.^2);
7 end

```

Table 1: Værdier for opgave 3

signal	max	min	mean	rms	effekt
s1	0.930	-0.820	≈ 0	0.119	60557.9
s2left	0.955	-0.955	≈ 0	0.338	1001787.6
s2right	0.955	-0.955	≈ 0	0.354	1102211.3
s3	0.960	-0.919	≈ 0	0.091	10738.9

4 Opgave 4

Ligesom i opgave 3, er crest allerede beregnet i listing 2. Til at beregne crest-faktoren, bruges formelen som beskrevet i ligning 1. Da der tages $20 \cdot \log_{10}$, kan det antages at crest-faktoren vil være i decibel. Det på listing 4 kan det også ses, hvordan crest-faktoren er beregnet i MATLAB.

$$20 \cdot \log_{10} \frac{x_{peak}}{x_{rms}} \quad (1)$$

Listing 4: Bergning af crest-faktor for alle 3 signaler

```

1 for i = 1:length(y)
2     y(i).crest = 20*log10(y(i).max/y(i).rms);
3 end

```

Table 2: Værdier af crest-faktor

signal	crest
s1	17.80 dB
s2left	9.02 dB
s2right	8.60 dB
s3	20.37 dB

I tabel 2 ses beregningen for de 3 signalers crest-faktor. Her ses der, at både s1 og s2 har en cirka dobbelt så stor crest-faktor, som s3 (både right og left). Svaret på hvorfor, kan findes i udregningen for crest-faktoren, da foruden omregningen til dB, tages x_{peak}/x_{rms} . Alle signalernes peak ligger omkring 0.95. Den store forskel er deres rms-værdi, hvor s2 har om rms-værdi på cirka 0.33, mens s1 og s3 begge ligger på en rms-værdi lige omkring 0.1. Herved har vi at x_{peak} divideres med et større tal for s2, end for s1 og s3. At s2 har en lavere crest-faktor giver meget god mening, når crest-faktoren beskriver hvor ekstrem forskellen er mellem peaks og rms af signalet. Da s2 er elektronisk musik, med masser af "gang i den", må vi dermed også kunne forvente en høj rms. Omvendt er der ikke lige så meget gang i den, i et signal med Mozart musik, eller tale.