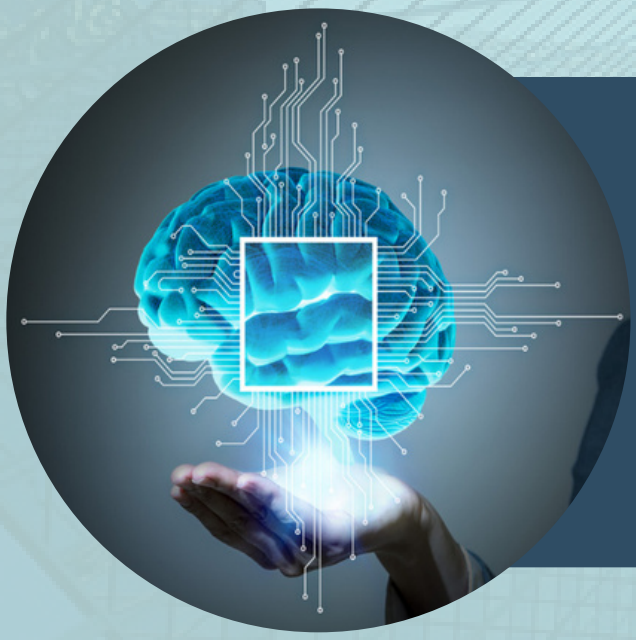
EIDIA
École d'Ingénierie Digitale et d'Intelligence Artificielle

Compte rendu

Traitement de signal

TP 2



Préparé par:

Ilias Manadir

Encadré par :

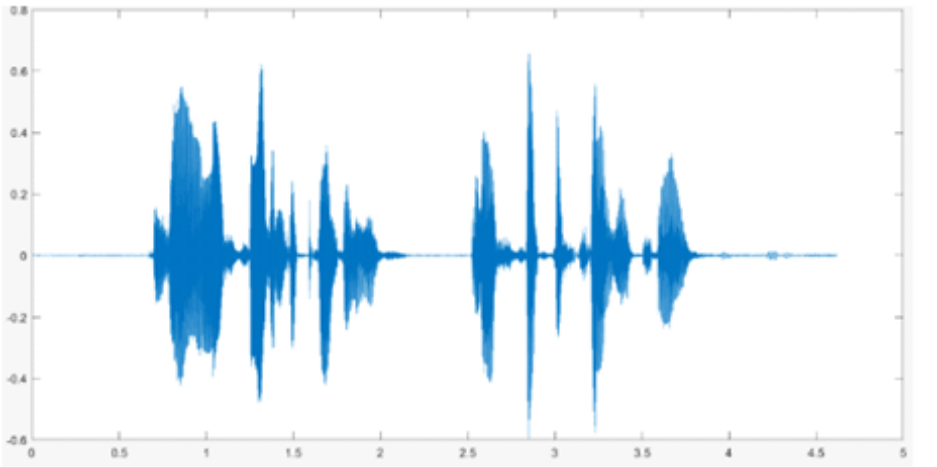
Mr Alaa

1-Sauvegardez ce fichier dans bureau,puis charger-le dans MATLAB

```
[y,fs]=audioread("audio.opus");
```

2- Tracez le signal enregistré en fonction du temps

```
[y,fs]=audioread("audio.opus");
N = length(y)
ts = 1/fs
t = (0:N-1)*ts;
plot(t,y)
```



puis en uitilisant la commande 'sound' on peut l'ecouter

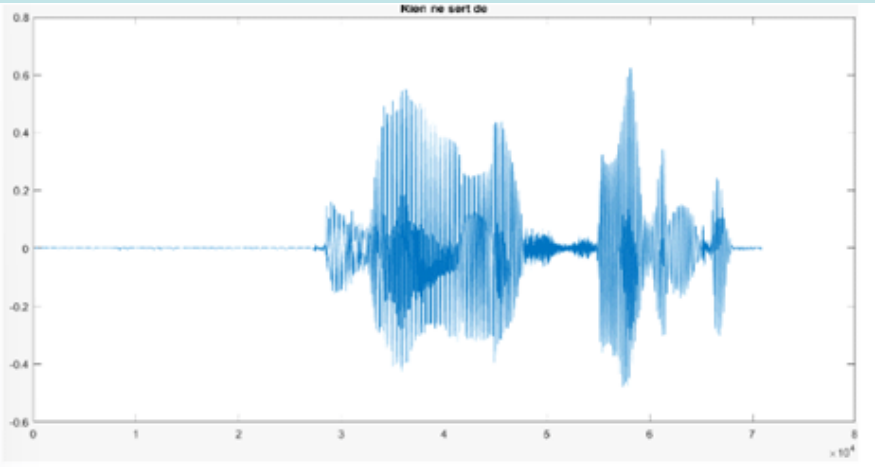
3-On peut également le compresser en multipliant la fréquence 2 ou le dilater en la divisant par 2

```
sound(x,2*fe);
sound(x,fe/2);
```

La compression du son a tendance à garder les pics à hautes fréquences, ce qui donne l'impression que le son est diminué et aigue.

4-on trace le signal , puis on essaye de repérer les indices du début et de fin de la phrase « Rien ne sert de ».

```
rien_ne_sert_de = y(5055:76000);
plot(rien_ne_sert_de);
title('Rien ne sert de');
```



5- Pour segmenter le premier mot, il faut par exemple créer un vecteur « riennesertde »contenant les n premières valeurs du signal enregistré x puis one cree le vecteur avant de l'écouter

```
rien_ne_sert_de = y(5055:76000);
sound(rien ne sert de,fs);
```

6- on segmente cette fois la phrase en créant des variables : riennesertde, courir, ilfaut, partir apoint.

```
rien_ne_sert_de = y(5055:76000);

courir  = y(76000:95395);

il_faut = y(95395:141652);

partir_a_point  = y(141652:198500);
```

7-Réarrangez ce vecteur pour écouter la phrase synthétisée « Rien ne sert de partir à point, il faut courir ».

```
vector =[rien_ne_sert_de ; partir_a_point ; il_faut ; courir];
sound(vector,fs);
```

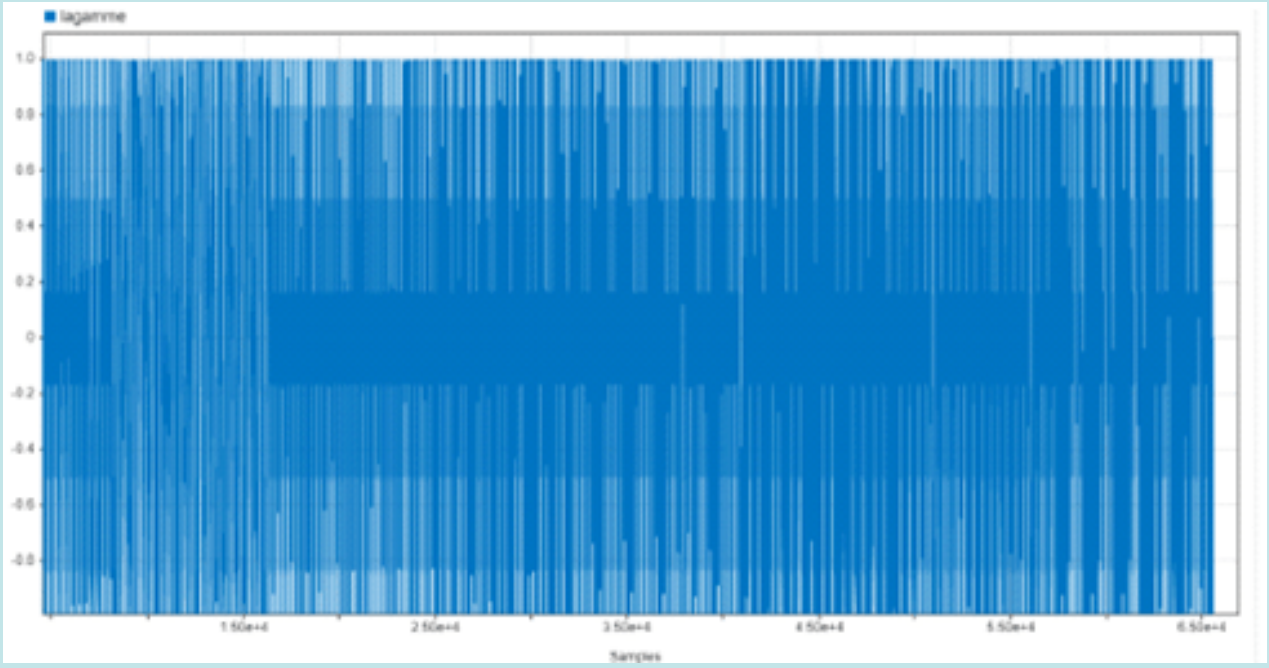
Synthèse et analyse spectrale d'une gamme de musique

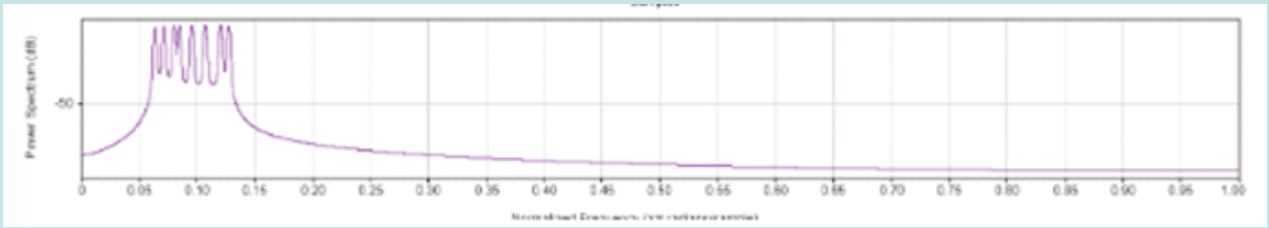
Do1	Ré	Mi	Fa	Sol	La	Si	Do2
262 Hz	294 Hz	330 Hz	349 Hz	392 Hz	440 Hz	494 Hz	523 Hz

1- on crée un programme qui permet de jouer une gamme de musique. La fréquence de chaque note est précisée dans le tableau ci-dessous. Chaque note aura une durée de 1s.

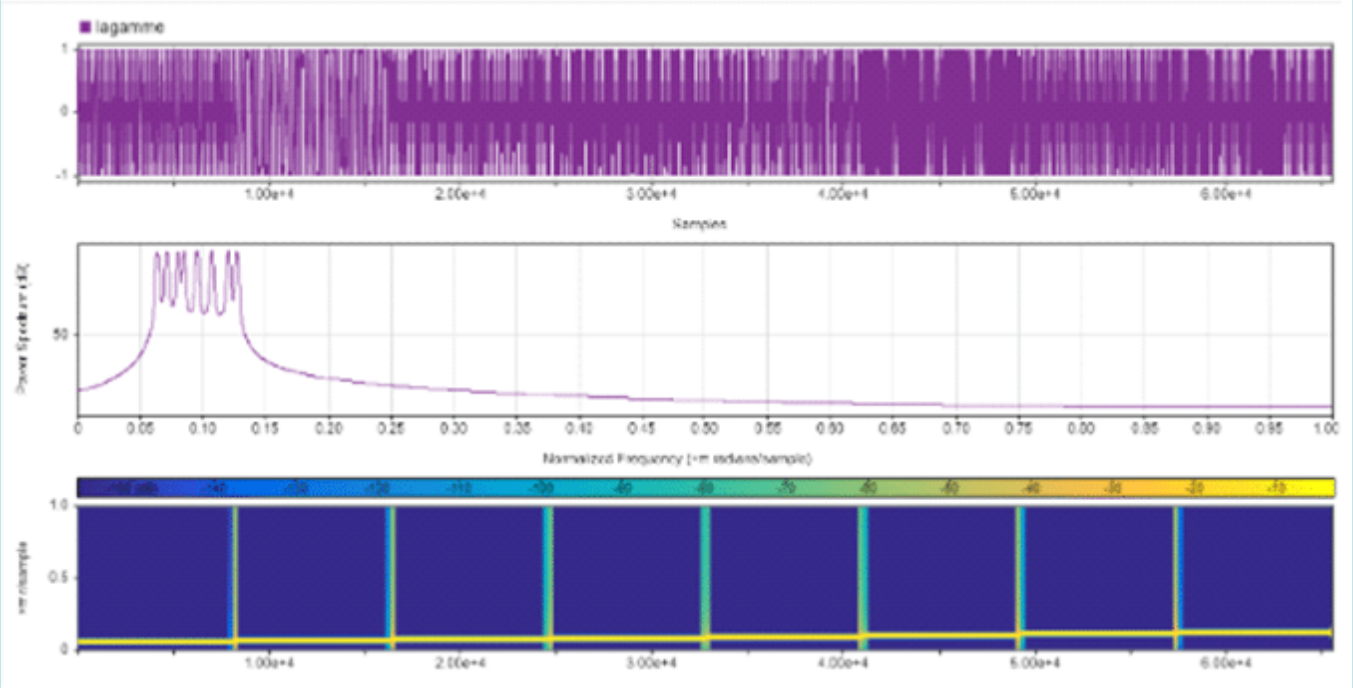
```
clear all
  close all
  clc

  fe=8192;
  te=1/fe;
  ts=0:te:1;


  fdol=262;
  fRe=294;
  fMi=330;
  fFa=349;
  fSol=392;
  fLa=440;
  fSi=494;
  fDo2=523;

  DOl=sin(2*fdol*pi*ts);
  RE=sin(2*fRe*pi*ts);
  MI=sin(2*fMi*pi*ts);
  FA=sin(2*fFa*pi*ts);
  SOL=sin(2*fSol*pi*ts);
  LA=sin(2*fLa*pi*ts);
  SI=sin(2*fSi*pi*ts);
  DO2=sin(2*fDo2*pi*ts);
  lagamme=[DOl RE MI FA SOL LA SI DO2];

  sound (lagamme, fe ) ;
```

2- Utilisez l'outil graphique d'analyse de signaux signalAnalyzer pour visualiser le spectre de votre gamme. Observez les 8 fréquences contenues dans la gamme et vérifiez leur valeur numérique à l'aide des curseurs.





3- Tracez le spectrogramme qui permet de visualiser le contenu fréquentiel du signal au cours du temps (comme le fait une partition de musique) mais la précision sur l'axe des fréquences n'est pas suffisante pour relever précisément les 8 fréquences.



4- Le spectre d'un signal à temps continu peut être approché par transformée de Fourier discrète (TFD) ou sa version rapide (Fast Fourier Transform (FFT). Afficher le spectre de fréquence de la gamme musicale crée en échelle linéaire, puis avec une échelle en décibels.

```
DS=abs(fft(gamme));
N=length(gamme)
subplot(2,1,1)
plot(DS);
title('Fft de la gamme');
k=mag2db(DS);
subplot(2,1,2)

%avec une échelle en décibels.
fshift=(-N/2:N/2 -1 )*fe/N;
plot(fshift,fftshift(k));
```