

TRAITEMENT NUMÉRIQUE DU SIGNAL

GROUPE

G9|A

PRÉSENTÉ PAR

MAISONNEUVE LE BREC Matthieu

MALTERRE-NGO Romain

MARTINEZ Christophe

MBONGO Noémie

MOREAU Lucie

LOISEL David

SOMMAIRE

1. Enoncé	3
2. Introduction	4
3. Méthode	5
4. Résultats expérimentaux	7
5. Conclusion	11

ÉNONCÉ

Problème I :

Proposer un algorithme qui permet de détecter la présence/absence d'un signal audio enregistrant le bruit ambiant. Le micro utilisé a une sensibilité S (dBV) et amplifie le signal électrique avec un gain G . On définit un « bruit pénible », appelé plus simplement « bruit », comme un signal sonore de durée supérieure à Dt (s) et de niveau supérieur ou égal à P_SPL (dB SPL - Sound Pressure Level).

A contrario, un bruit acceptable, qu'on appellera par abus de langage « silence », correspond à un signal sonore de niveau inférieur à P_SPL . Les plages de silence de durée inférieure à dt (s) seront intégrées aux bruits détectés. Caractériser chaque bruit détecté par :

- Sa durée
- Sa puissance moyenne en mW et en dBm
- Sa tension RMS en V
- Son coefficient d'autocorrélation $\max_{\tau>0} C_{xx}(\tau)$ et le délai correspondant $\arg \max_{\tau>0} C_{xx}(\tau)$.

Paramètres :

- $S = -48$ dBV
- $G = 30$ dB
- $P_SPL = 80$ dB SPL
- $Dt = 1$ s
- $dt = 0.5$ s

Les expérimentations seront menées sur les signaux environnementaux :

- MarteauPiqueur01.mp3
- Jardin01.mp3
- Jardin02.mp3
- Betonneuse.mp3

INTRODUCTION

Un signal audio est une représentation analogique d'un son. Elle est souvent représentée par une tension l'électricité et est synthétisée directement d'un transducteur (microphone). Un algorithme quant à lui est une suite d'instructions et d'opérations permettant de résoudre une classe de problème.

Dans notre cas, nous avons codé l'algorithme sous MATLAB. L'algorithme permettra de détecter la présence et l'absence de tout signal audio enregistrant le bruit ambiant. Pour cela, nous nous pencherons sur l'étude de signaux échantillonnés.

Un bruit est pénible si sa durée est supérieure à $Dt(s) = 1$ sec et si son niveau acoustique est supérieur ou égal à $P_SPL = 80$ dB SPL. Un bruit est défini comme acceptable si sa durée est inférieure à $dt(s) = 0.5$ sec et son niveau acoustique est inférieurs à $P_SPL = 80$ dB SPL.

Ainsi, dans nos travaux, chaque bruit sera défini par sa durée, sa puissance moyenne (mW et dBm), sa tension RMS (V) et son coefficient de corrélation.

MÉTHODE

Dans un premier temps, il nous faut déterminer le seuil de détection à partir de la sensibilité (S , une valeur donnée). Pour cela il est important de faire la distinction entre ces deux termes :

- Le **seuil de détection** est le niveau minimal d'une grandeur que doit atteindre une mesure pour être considérée comme significative : c'est la limite au-delà de laquelle une mesure est considérée comme "détectée".
- La **sensibilité**, désigne la capacité d'un système à détecter une variation dans une grandeur mesurée. Plus un système est sensible, plus il est capable de détecter des variations de faible amplitude.

Ainsi, nous avons calculé le seuil de détection :

$$S = 20 * \log\left(\frac{S_{pa}}{S_{ref}}\right) \quad \text{avec } S_{ref} = 1V/Pa$$

$$PSPL = 20 * \log\left(\frac{P_{pa}}{P_{ref}}\right) \quad \text{avec } P_{ref} = 20 * 10^{(-6)} Pa$$

Ainsi donc :

$$S_{vpa} = 10^{\frac{S}{20}} * S_{ref} \quad \text{en V/Pa}$$

$$P_{Pa} = 10^{\frac{PSPL}{20}} + P_{ref} \quad \text{en Pa}$$

De plus, le gain de l'énoncé est en dB, trouvons le gain sans unité :

$$G = 20 \log G2 \quad \text{donc } G2 = 10^{\frac{G}{20}}$$

Par analyse dimensionnelle on peut alors faire les rapports suivants :

$$RMS = S_{vpa} * P_{Pa} * G2$$

$$P_{bruit} = RMS^2$$

$$Seuil = 10 * \log(P_{bruit} * 10^3)$$

$$Seuil = -2dBm$$

Nous avons décidé de créer une fonction qui prendrait en argument le nom de l'audio sur lequel elle doit travailler. Nous appellerons cette fonction 'Analyse'.

La fonction 'Analyse' comprends plusieurs étapes :

- On charge le signal audio avec *audioread*, et on mets les valeurs récupérées (points, et fréquence d'échantillonnage) dans une liste qu'on pourra réutiliser par la suite ;
- S'il le signal audio est en stéréo, on fait la moyenne des deux pistes pour le convertir en mono afin de pouvoir travailler dessus ;
- On utilise le P_SPL (valeur fixée par l'énoncé) pour découper le signal en plusieurs bruits ;
- Pour chaque bruit, on note le temps de début et le temps de fin, qu'on stocke dans des listes ;
- On convertit les puissances du bruit de Watt en dBm ;
- On détermine la tension RMS à partir de cette puissance ;
- On calcule le coefficient d'autocorrélation à l'aide d'une boucle *for* et de la fonction *sum*, pour ne pas utiliser la fonction existante *xcorr* qui normalise de façon différente ;
- Matlab illustre ensuite à l'aide d'un *plot()* le tout, en superposant les courbes d'audio, avec des courbes, représentants par des créneaux, les bruits et silences détectés. Les bruits sont alors caractérisés par un créneau haut, tandis que les silences sont caractérisés par un créneau bas (c'est-à-dire que les points sont au niveau 0).

RÉSULTATS EXPÉRI- MENTAUX

La Figure 1 montre l'un des signaux qu'on étudie (Jardin01.mp3).

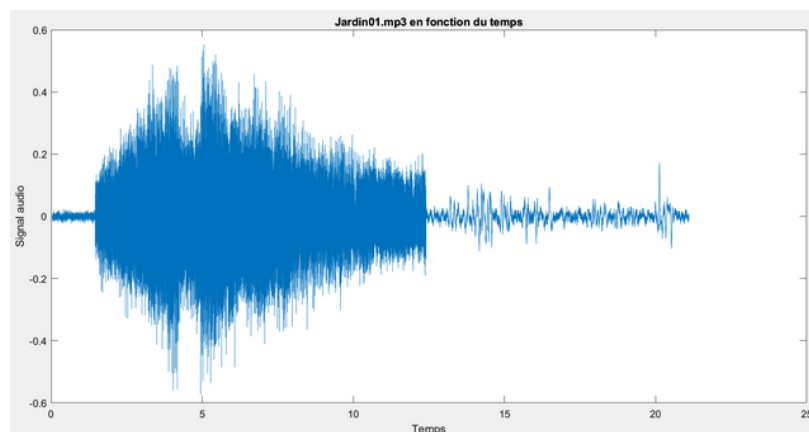
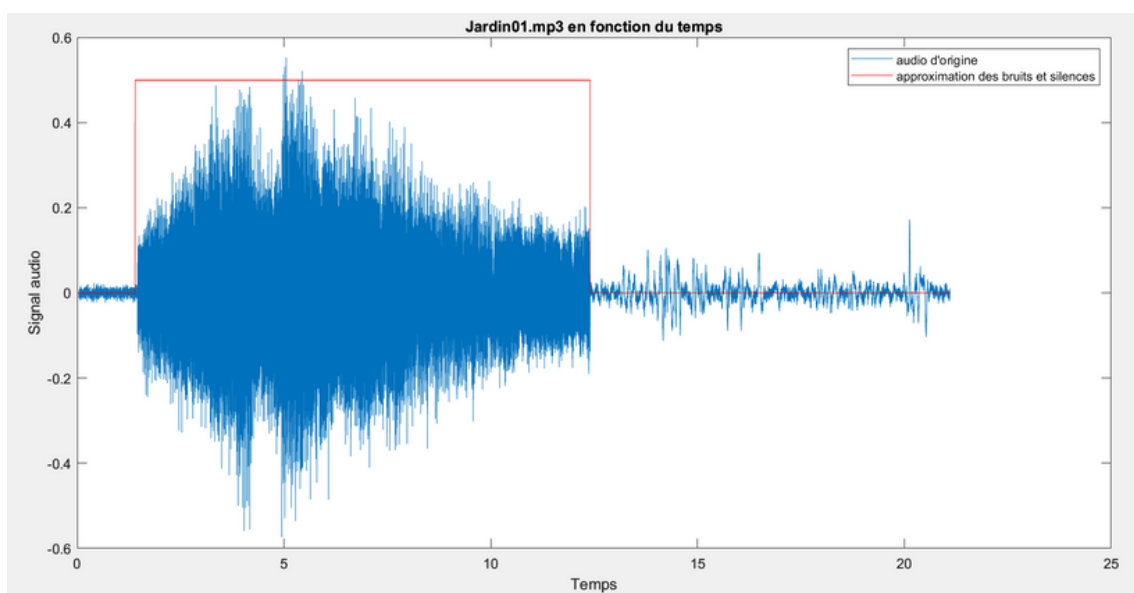


Figure 1. Signal étudié (Jardin01.mp3)

Après l'avoir passé dans notre programme, on remarque une deuxième courbe (en rouge) qui permet de visualiser très simplement les bruits de cet audio. Il n'y en a ici qu'un seul, caractérisé cependant par sa longue durée (10,9 sec).



Bruit	Position	PDBM	V RMS	Coef autocorrélation
1	1.5 à 12.4 s	9.0339	0.089473	3848.1649

La Figure 2 montre l'un des signaux qu'on étudie (Jardin02).

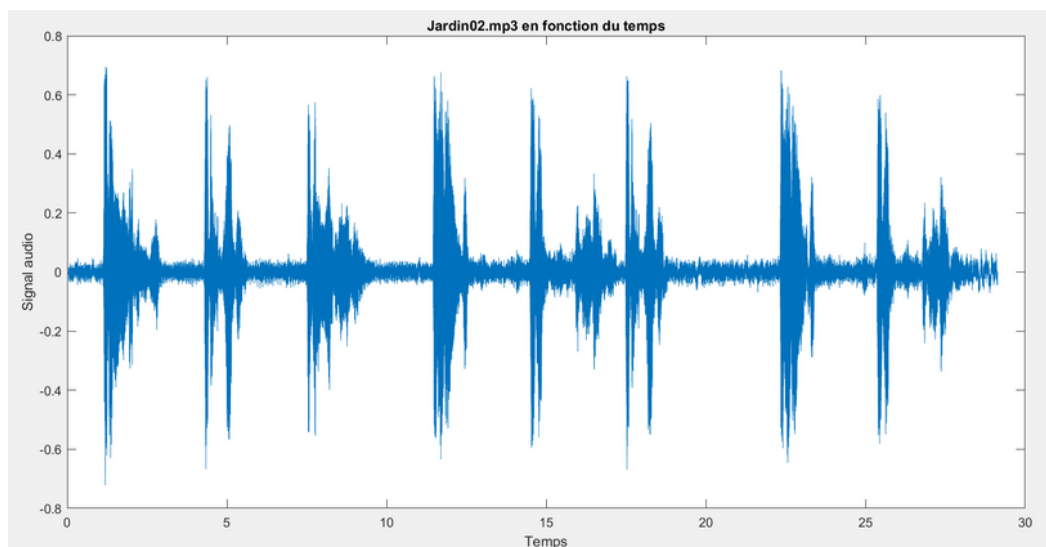
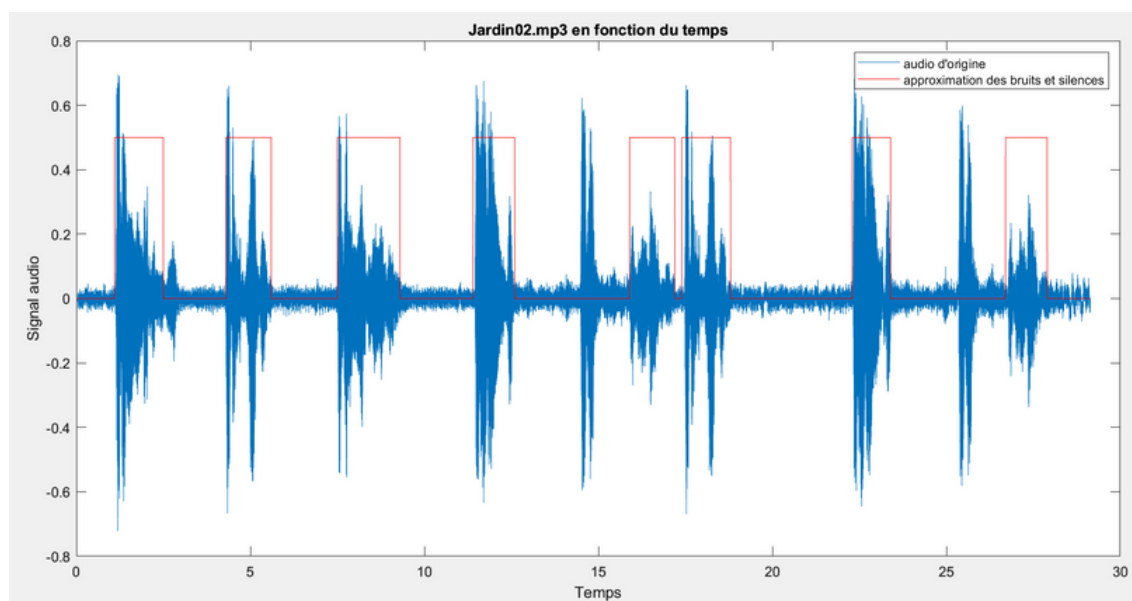


Figure 2. Signal étudié (Jardin02.mp3)

Après l'avoir passé dans notre programme, on remarque cette fois-ci que divers pics, plus courts, apparaissent : cet audio est une alternance de bruits et de silences, tous décrits dans le tableau ci-dessous.



Bruit	Position	PDBM	V RMS	Coef autocorrélation
1	1.1 à 2.5 s	13.009	0.1414	1343.6556
2	4.3 à 5.6 s	12.2867	0.13012	1056.4572
3	7.5 à 9.3 s	10.312	0.10366	928.3609
4	11.4 à 12.6 s	14.1802	0.16181	1508.1538
5	15.9 à 17.2 s	5.9509	0.06274	245.6283
6	17.4 à 18.8 s	12.0088	0.12602	1067.213
7	22.3 à 23.4 s	14.6508	0.17082	1540.6923
8	26.7 à 27.9 s	6.069	0.063599	232.982

La Figure 3 montre l'un des signaux qu'on étudie (MarteauPiqueur).

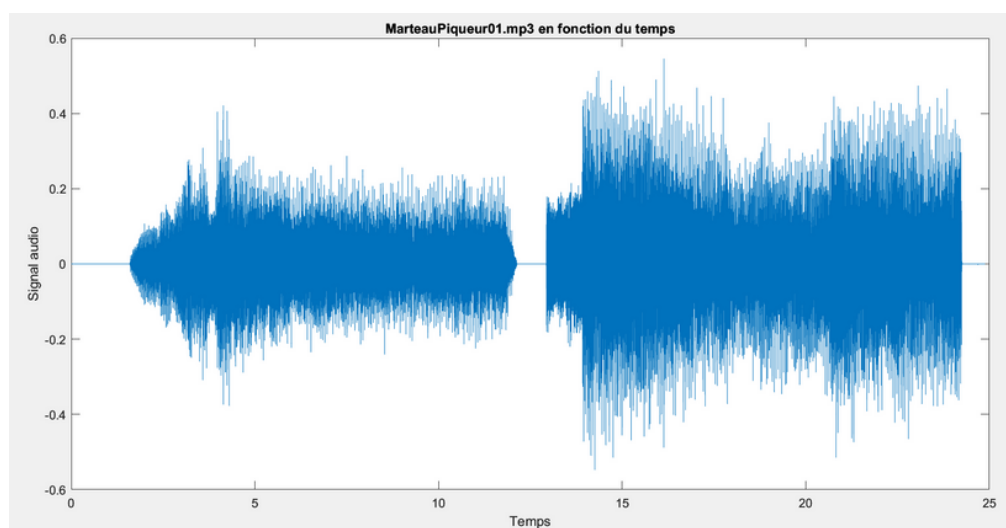
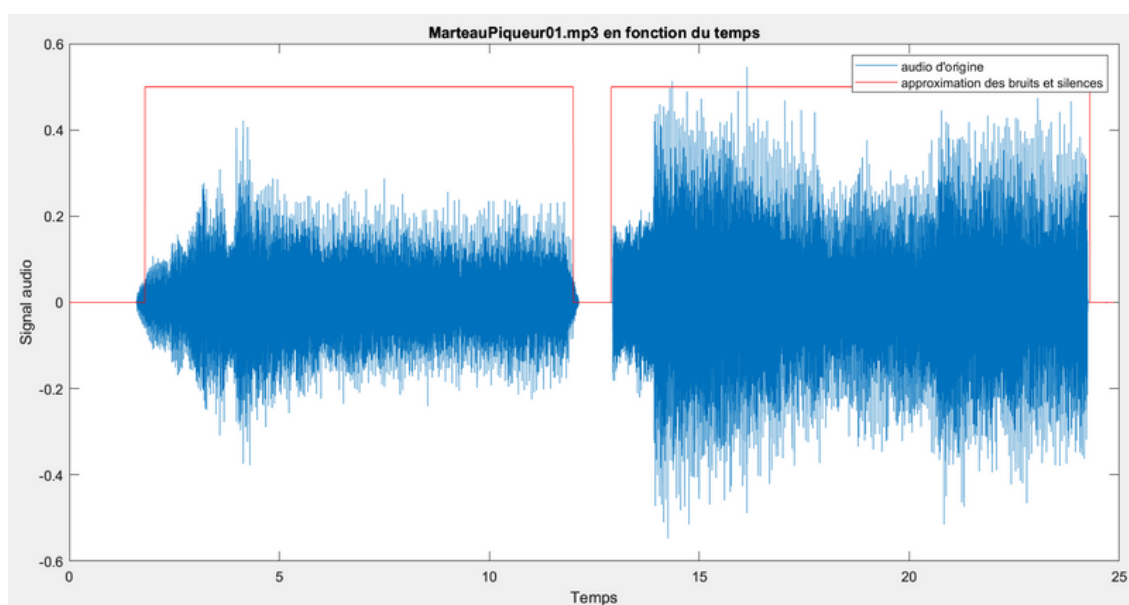


Figure 3. Signal étudié (MarteauPiqueur.mp3)

Après l'avoir passé dans notre programme, on remarque l'apparition de deux larges bruits entrecoupés par un silence de presque 1 seconde. Le deuxième bruit possède en revanche une puissance moyenne bien plus grande que celle du premier.



Bruit	Position	PDBM	V RMS	Coef autocorrélation
1	1.8 à 12 s	5.7206	0.061098	1679.183
2	12.9 à 24.3 s	9.1081	0.090242	4094.087

La Figure 4 montre l'un des signaux qu'on étudie (Betonneuse).

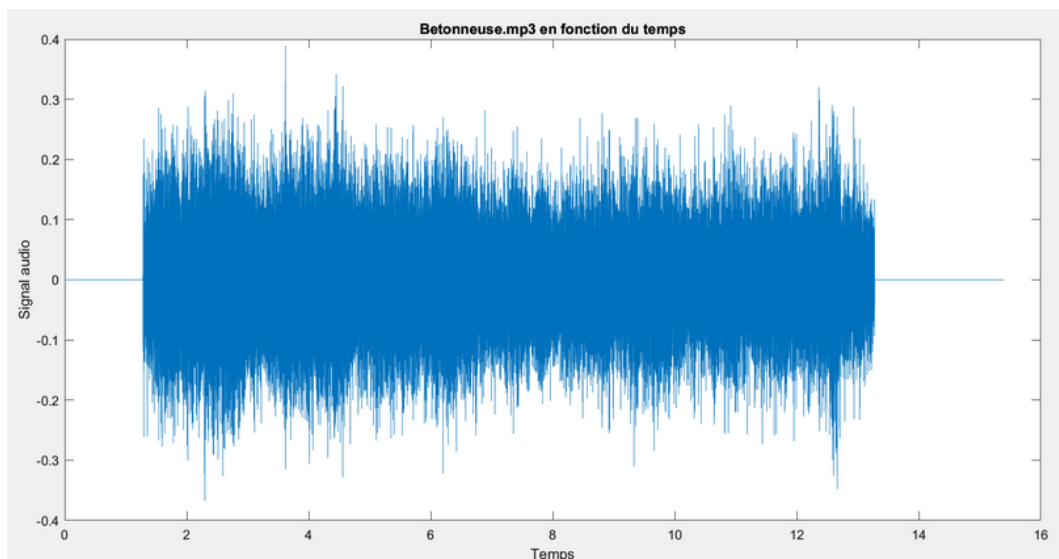
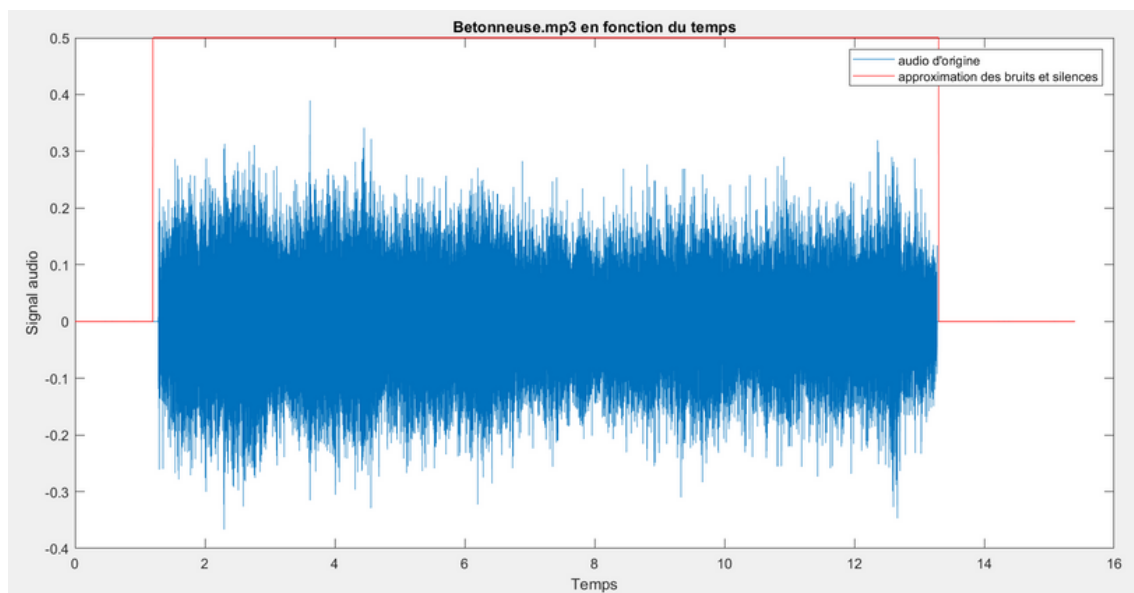


Figure 4. Signal étudié (Betonneuse.mp3)

Après l'avoir passé dans notre programme, 1 seul pic faisant pratiquement l'intégralité de l'audio apparaît.



Bruit	Position	PDBM	V RMS	Coef autocorrélation
1	1.2 à 13.3 s	7.3278	0.073517	2884.0341

CONCLUSION

Pour conclure, résoudre la problématique n°1 a permis à certains membres du groupe de revoir leurs bases en signaux et systèmes, notamment les notions clés de l'échantillonnage. Et pour d'autres d'apprendre de nouvelles facultés.

L'implémentation du code nous a donc mener à caractériser les différents sons en affichant leurs signaux et leurs caractéristiques, notamment la durée, la puissance moyenne ou encore la tension RMS et détection de bruit.

Nous avons toutefois rencontré quelques difficultés notamment afin d'avoir un programme qui tourne relativement rapidement, sur l'optimisation des boucles et du code. Nous avons aussi eu un soucis au niveau de l'implémentation de la méthode de calcul de l'autocorrélation. Autant notre code fonctionnait correctement lors des exercices réalisées dans la semaine, autant nous n'avons pas réussi à le faire fonctionner ici, sûrement par manque de temps.