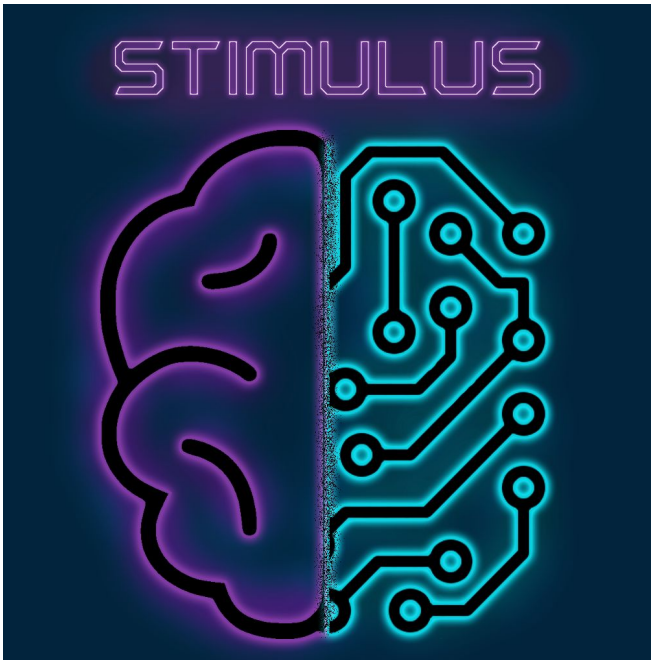
APP Signal
Livrable Semaine 1

Equipe G10B

JULIEN COLOMBAIN
CYPRIAN CUNIN
GABRIEL ENRIQUEZ
CELIA HOULETTE
CAROLINE YAN



isep
École d'ingénieurs du numérique

2019 - 2020

Introduction

Ce premier livrable concerne la résolution de deux problèmes de traitement de signal échantillonné. Pour chacun de ces problèmes, nous allons présenter l'algorithme que nous avons mis au point et implémenté sur Matlab. Le premier problème consiste à détecter la présence ou l'absence d'un signal audio. Le deuxième consiste à à comparer les fréquences de deux sons audio purs.

Résolution du problème I-A

Nous cherchons à déterminer la présence ou l'absence d'un signal audio. Nous supposerons que le signal audio et quasi-stationnaire et que les 50 premières millisecondes ne contiennent pas de signal utile. Il s'agira ensuite d'échantillonner ce signal et de regarder, pour chaque échantillon, s'il est associé à du bruit ou à du signal utile.

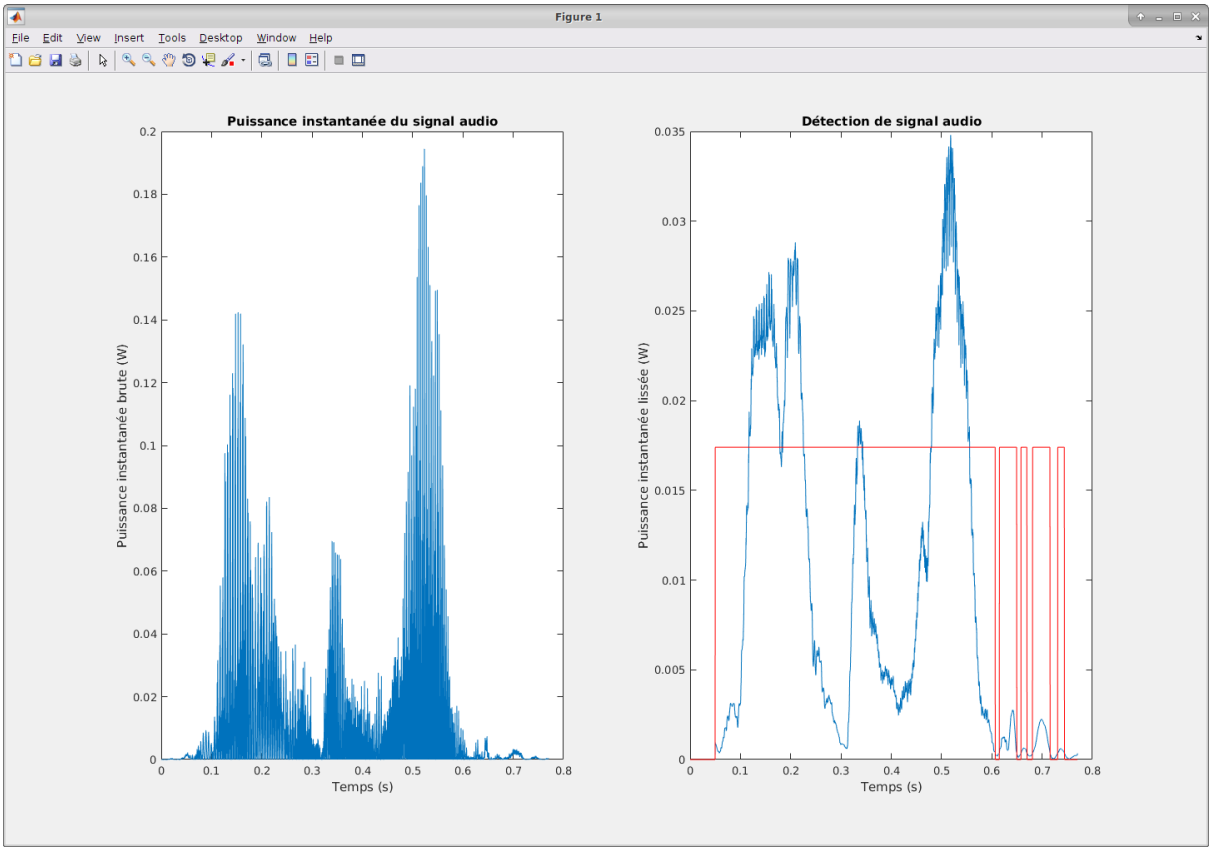
Pour échantillonner le signal, nous avons utilisé la fonction audioread() de Matlab qui permet de récupérer les données échantillonnées du fichier audio à l'intérieur d'un vecteur x et la fréquence d'échantillonnage fe.

Pour étudier le signal, nous nous sommes intéressés à sa puissance instantanée, définie par $p(t)=x(t)^2$ et représenté dans notre programme par la variable p, dans le but de travailler uniquement avec des valeurs positives. Nous avons ainsi fixé un seuil de puissance, représenté par la variable seuil, au-dessus duquel nous considérons qu'il y a bel et bien présence d'un signal audio.

Dans certains cas, le signal reçu présente une faible perturbation et certains points isolés pourraient se retrouver au-dessus du seuil et ainsi être considérés comme un signal audio sans que ce soit le cas. Pour palier ce problème, nous avons lisser la puissance instantanée du signal en utilisant le principe des fenêtre glissantes : nous avons attribué à chaque point la moyenne m de la puissance instantanée des n points le suivant, en mettant le point au milieu de la fenêtre afin de pouvoir détecter le bruit un juste avant qu'il se déclenche. Nous avons choisi la taille de la fenêtre glissante en fonction de la fréquence d'échantillonnage et de de la précision, dans le cadre de notre hypothèse de signal quasi-stationnaire. Nous avons choisi une taille de 1000 points pour la fenêtre. Etant donné que les 50 premières ms ne contiennent pas de signal utile, nous avons initialisé la puissance à 0W.

Ensuite, pour définir le seuil, nous avons pris la moyenne des puissances instantanées des points correspondant aux 50 ms du début que nous avons modulé par un facteur 2 afin de réduire le risque de détecter un signal audio par erreur. Plus ce facteur de modulation est grand, plus le risque de détecter un signal audio par erreur est faible et le risque de ne pas détecter un signal audio plus faible est grand.

Enfin, nous avons stocké dans un booléen over les puissances instantanées du vecteur vitesse qui dépassent le seuil établi pour valider la condition de présence / absence de signal sonore. Nous avons affiché ce signal booléen sur le même graphique que la puissance instantanée lissée. On remarque ainsi qu'un signal créneau rouge se superpose à la courbe de puissance, délimitant la présence de l'absence du signal audio.



Graphique de la puissance instantanée lissée en fonction du temps avec le signal créneau lorsque le son est détecté pour le fichier "BonneJournee.wav", réalisé sur Matlab

Résolution du problème I-B

Nous cherchons à comparer les fréquences de deux sons audio purs (donc de la forme $A\cos(2\pi ft+\varphi)$).

Pour cela, nous avons utilisé la notion d'intercorrélation vue en cours. En effet, calculer l'intercorrélation entre deux signaux nous donne une indication sur la similitude entre ces deux signaux, qu'ils soient décalés ou bruités. C'est lorsqu'elle est maximale que les deux signaux se ressemblent le plus.

Pour implémenter notre algorithme, nous avons utilisé les résultats démontrés dans l'exercice 2-6 :

On considère les deux signaux suivants :

$$x1\,(t) = A1cos\ (2\pi f1t) \quad et \quad x2\,(t) = A2cos(2\pi f2t + \varphi)$$

Calculons l'intercorrélation de ces deux signaux en 0 sur l'intervalle [0, D] :

$$C_{x1,x2}(0) = \int_0^D x1(t)\,x2(t)\,dt$$

$$C_{x1,x2}(0) = A1A2 \int_0^D cos\ (2\pi f1t)\ cos\ (2\pi f2t + \varphi)\ dt$$

$$C_{x1,x2}(0) = \frac{A1A2}{2} \int_0^D [cos(2\pi(f1 + f2)t + \varphi) + cos(2\pi(f1 - f2)t\ - \varphi)]\,dt$$

$En\ prenant\ f1 = f2,\ \ et\ D\ = \frac{n}{f1}\ on\ obtient\ :$

$$C_{x1,x2}(0) = \frac{A1A2}{2} \int_0^{\frac{n}{f1}} [cos\ (4\pi f1t + \varphi)\ + cos\ (-\varphi)\ ]\,dt$$

$$C_{x1,x2}(0) = \frac{A1A2}{2} \int_0^{\frac{n}{f1}} [cos\ (4\pi f1t + \varphi)\ + cos\ (\varphi)\ ]\,dt$$

$$C_{x1,x2}(0) = \frac{A1A2}{2} \left[\frac{sin\ (4\pi f1t+\varphi)}{4\pi f1} + cos\ (\varphi)\ t\right]_0^{\frac{n}{f1}}$$

$$C_{x1,x2}(0) = \frac{A1A2}{2} \left[\frac{sin\ (4\pi n+\varphi)}{4\pi f1} + \frac{cos\ (\varphi)\ n}{f1} - \frac{sin\ (\varphi)}{4\pi f1}\right]$$

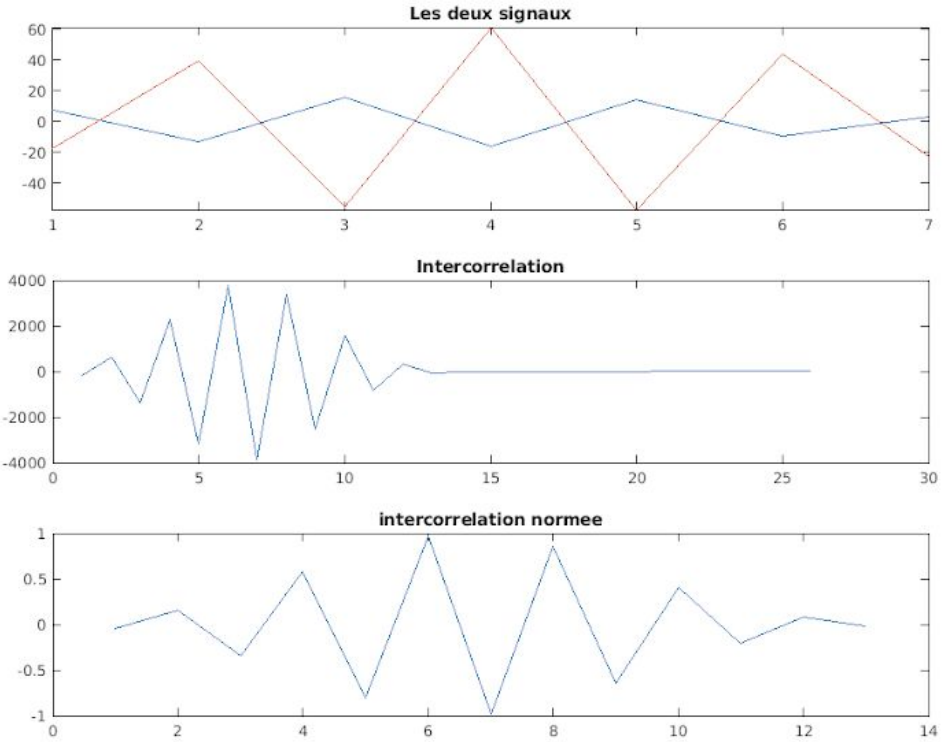
$$C_{x1,x2}(0) = \frac{A1A2}{2} \left[\frac{sin\ (\varphi)}{4\pi f1} + \frac{cos\ (\varphi)\ n}{f1} - \frac{sin\ (\varphi)}{4\pi f1}\right]$$

$$C_{x1,x2}(0) = \frac{A1A2}{2} \left[\frac{cos(\varphi)n}{f1}\right] = \frac{A1A2\ cos(\varphi)D}{2}$$

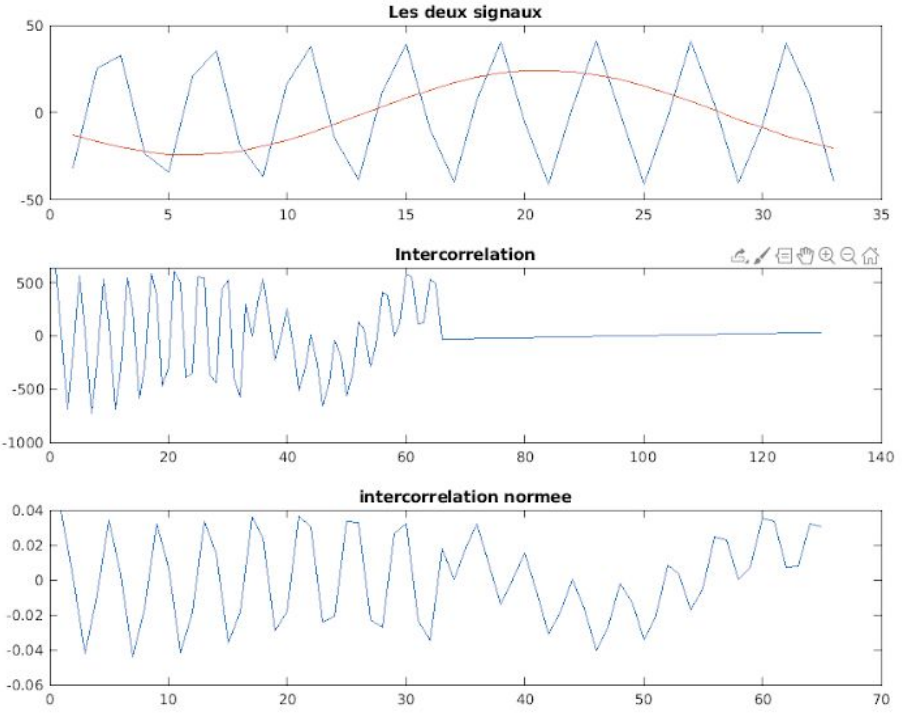
$$D'où\ \ \frac{C_{x1,x2}(0)}{D} = \frac{A1A2\ cos(\varphi)}{2}$$

Pour comparer la fréquence entre deux signaux, il nous faut donc calculer l'intercorrélation entre ces deux signaux, diviser le résultat par la durée, puis comparer le résultat obtenu avec la valeur attendue s'ils avaient exactement la même fréquence ($\frac{A1A2\ cos(\varphi)}{2}$) en se fixant une certaine marge d'erreur.

Pour cela, nous avons utilisé la fonction Matlab [corr decalage] = xcorr() qui calcule l'intercorrélation de deux signaux. Elle renvoie deux valeurs : l'intercorrélation de ces signaux (corr) et la valeur du décalage (decalage).
Pour pouvoir lire ces résultats nous divisons chaque signal par sa norme et nous cherchons l'intercorrélation entre les deux. Ainsi cela nous donne la courbe de l'intercorrélation normée avec des valeurs comprises entre -1 et 1. Si les signaux ont la même fréquence alors la valeur de l'intercorrélation normée sera plus proche de 1 sinon plus proche de 0. Nous avons donc choisis une marge d'erreur de 0,3 ce qui veut dire que s'il y a une intercorrélation de 0,7 ou plus, les signaux sont considérés de même fréquence.



Figures de deux signaux de même fréquence



Figures de deux signaux de fréquence différente

Conclusion

En conclusion, nous avons appris à implémenter des algorithmes sous Matlab afin de résoudre des problèmes de traitement de signal échantillonné. Nous avons été capables de résoudre les deux problèmes proposés. Cette première semaine nous a permis de nous initier au traitement de signal et à la programmation sous Matlab.