

CAPTEUR FREQUENCE CARDIAQUE

SOMMAIRE

Enoncé	3
Introduction	4
Réduction du signal	
Méthode 1 : Temporelle	5
Méthode 2 : Puissance instantanée	6
Méthode 1 : Méthode temporelle	8
Méthode 2 : Méthode fréquentielle	12
Comparaison des méthodes	15

ÉNONCÉ

Problème II :

Proposer deux algorithmes permettant de détecter un signal ECG et d'estimer le rythme cardiaque correspondant.

On travaillera sur les signaux ECG stockés dans les fichiers 100.wav, 101.wav, ..., et 109.wav. Le résultat de ces algorithmes doit être une fréquence cardiaque mesurée au cours du temps, lorsqu'un signal ECG est présent, 0 sinon.

Appliquer les algorithmes proposés sur tous les signaux (une méthode qui fonctionne sur un seul cas n'a aucun intérêt !).

Discuter les différentes approches possibles en termes de précision de l'analyse, de robustesse et de complexité algorithmique. Indiquer quelle approche est à favoriser pour une implémentation sur le microcontrôleur et quelle approche conduit aux résultats les plus robustes et les plus précis.

INTRODUCTION

Un électrocardiogramme (ou ECG) est un test qui mesure l'activité électrique du cœur au cours du temps. Ceci est alors représenté sur un graphique :

Abscisse : temps (ms)
Ordonnée : amplitude (mV)



Figure 1. Signal ECG typique

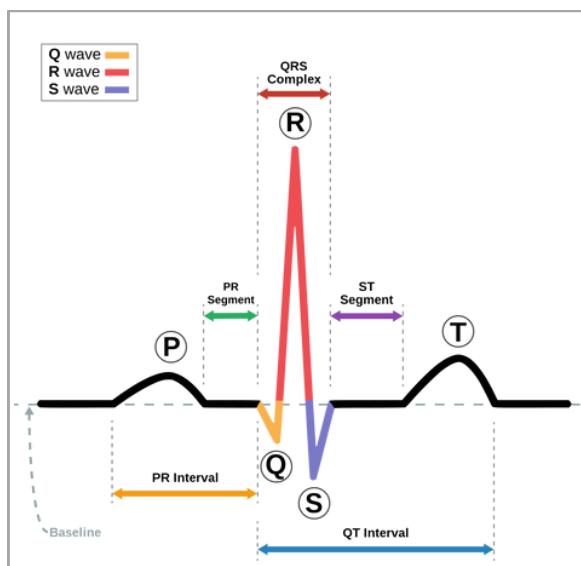


Figure 2. Un période d'un signal ECG

Un ECG est globalement périodique.

Chaque période est caractérisée par un pic (appelé complexe QRS) qui représente la contraction des ventricules (soit un battement de cœur).

Il est donc possible grâce à ces complexes QRS de calculer le rythme cardiaque d'un patient.

Pour le problème II, nous travaillons sur des signaux d'ECG (les fichiers 100.wav à 109.wav). Le principe était de récupérer la fréquence cardiaque de façon régulière dans chaque signal, pour pouvoir ensuite l'implémenter dans notre carte TIVA.

Pour commencer, nous récupérons le signal analogique à étudier de la même façon que pour tout signal audio : on utilise la fonction `audioread()`.

RÉDUCTION DU SIGNAL

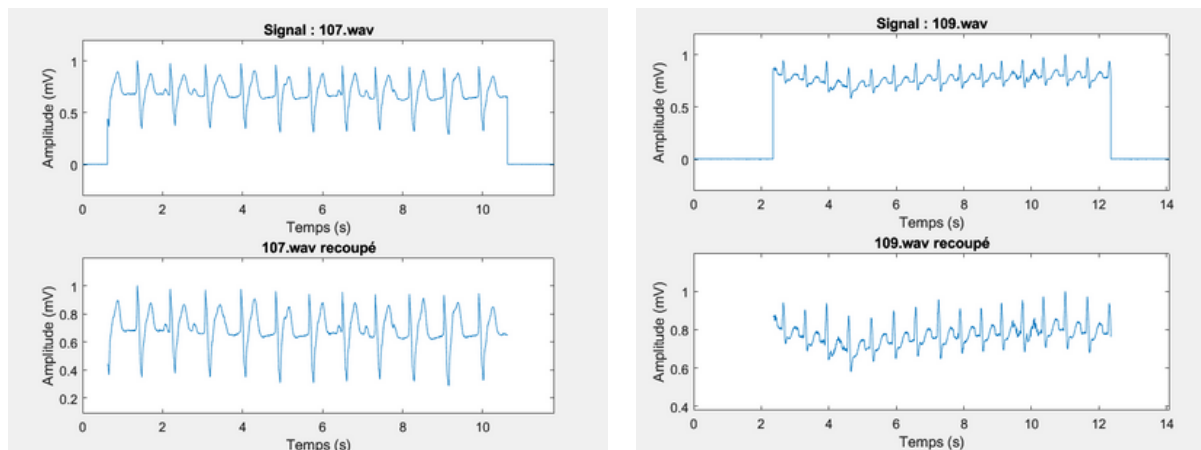
Méthode 1 : Temporelle

Afin d'afficher les ECG de façon plus visible sur *Matlab*, et de mieux visualiser les pics à détecter, nous avons décidé de réduire la taille de l'échantillon en enlevant les bouts d'ECG où le signal était nul.

Pour cela, nous fixons un seuil de 0.2 mV, et enlevons les signaux en dessous de ce seuil :

```
for i=1:n-1
    if x(i)>Seuil;
        y=[y x(i)];
    end
end
```

Quelques exemples d'affichage :



Très efficace sur tous les signaux étudiés, cette méthode n'est pourtant pas à privilégier. En effet, nous ne sommes pas à l'abri que les signaux à récupérer grâce à notre capteur n'aient pas le même offset, et donc que le seuil soit à modifier entre chaque signal, ou encore que aucun seuil ne soit définissable. On choisira plutôt une autre méthode.

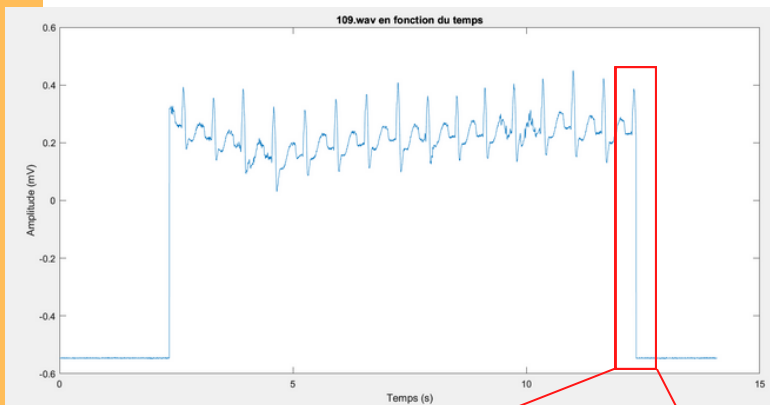
Méthode 2 : Puissance instantanée

Par le calcul de la puissance instantanée, il est possible de trouver une méthode qui semble plus adaptée à notre problème.

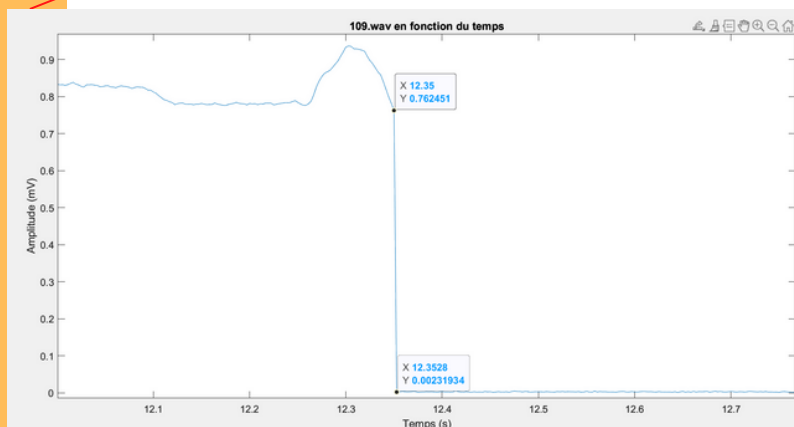
On rappelle la formule de la puissance instantanée :

$$P_i(t) = x(t)^2$$

On peut faire afficher les valeurs de puissance instantanée (PI) pour comparer les résultats obtenus avec le graphique d'ECG. Regardons par exemple un extrait du signal 109.wav :



PI : 0.7508 pour 12.3333 s
 PI : 0.73721 pour 12.3361 s
 PI : 0.7058 pour 12.3389 s
 PI : 0.67181 pour 12.3417 s
 PI : 0.63852 pour 12.3444 s
 PI : 0.61102 pour 12.3472 s
 PI : 0.58133 pour 12.35 s
 PI : 5.3793e-06 pour 12.3528 s
 PI : 6.7288e-06 pour 12.3556 s
 PI : 1.3862e-05 pour 12.3583 s
 PI : 1.0863e-05 pour 12.3611 s
 PI : 7.8827e-06 pour 12.3639 s
 PI : 9.3132e-06 pour 12.3667 s
 PI : 1.3411e-05 pour 12.3694 s

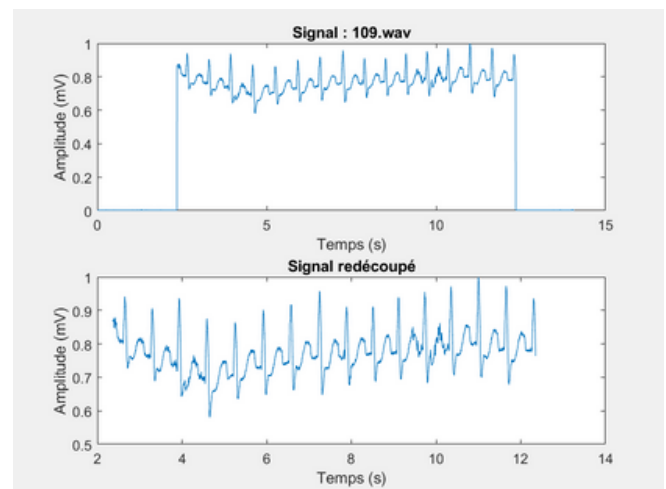
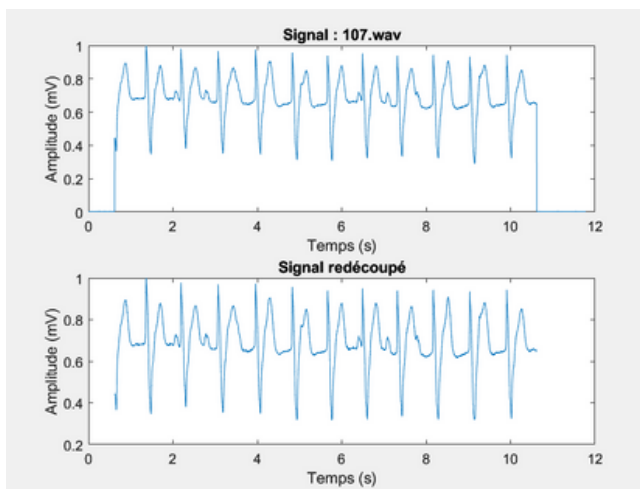


On trouve dans nos valeurs de puissance instantanée le saut représenté sur le graphique. En effet la puissance instantanée diminue drastiquement au moment où l'ECG passe à 0.

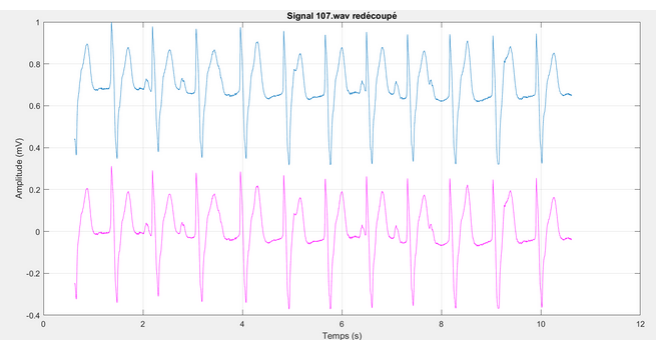
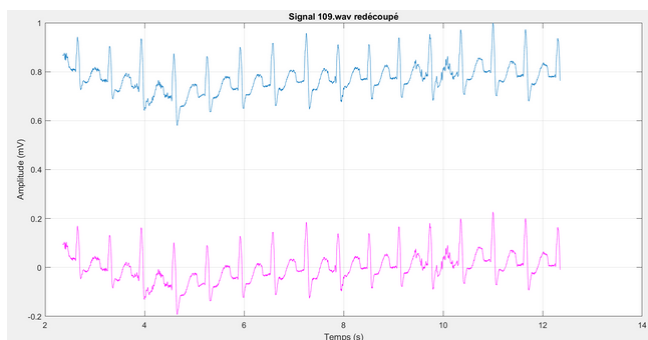
A l'aide de ces valeurs de puissance instantanée, on va chercher à déterminer quelles sont les données à garder pour l'analyse et celles à faire disparaître.

Pour cela on fixe un **Seuil = 0.1**. La valeur est prise arbitrairement, de sorte à ce que le programme accomplisse l'effet cherché.

On crée alors une nouvelle liste, dans laquelle on va venir stocker l'entièreté des points situés au dessus du seuil fixé, donc représentant le signal à garder. On obtient alors des courbes où l'on a pu enlever la plupart des points de valeurs inintéressantes pour notre analyse :



Pour finir, on enlèvera l'offset en soustrayant la moyenne du signal au signal lui même :



En **bleu** : avec offset

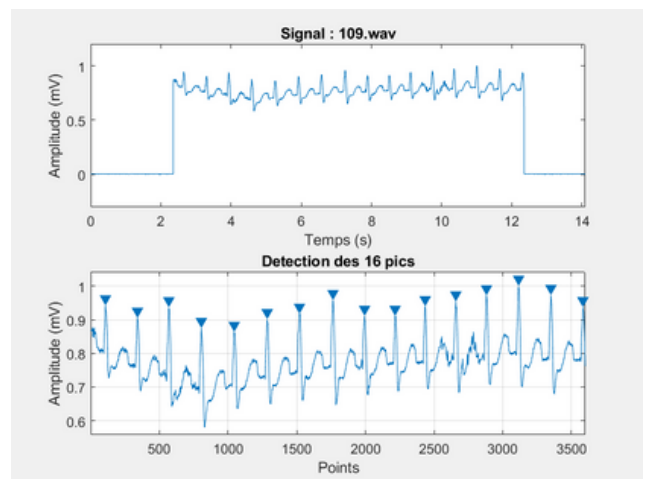
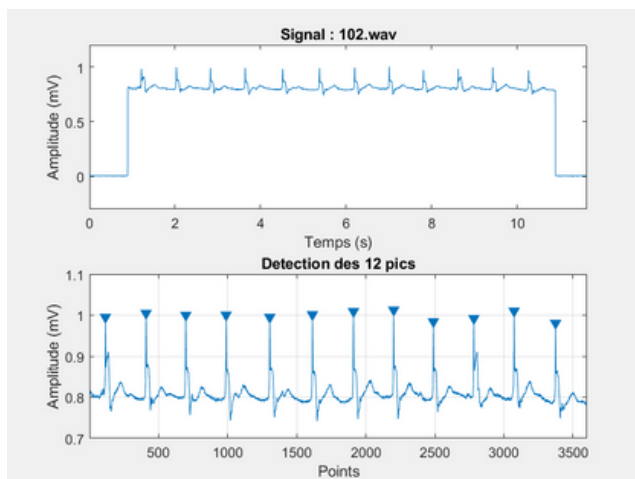
En **magenta** : sans offset

On obtient globalement les mêmes courbes qu'avec la méthode 1 ce qui est bon signe, quelques points sont conservés avec une méthode mais pas avec l'autre, ceci ne change rien pour la suite.

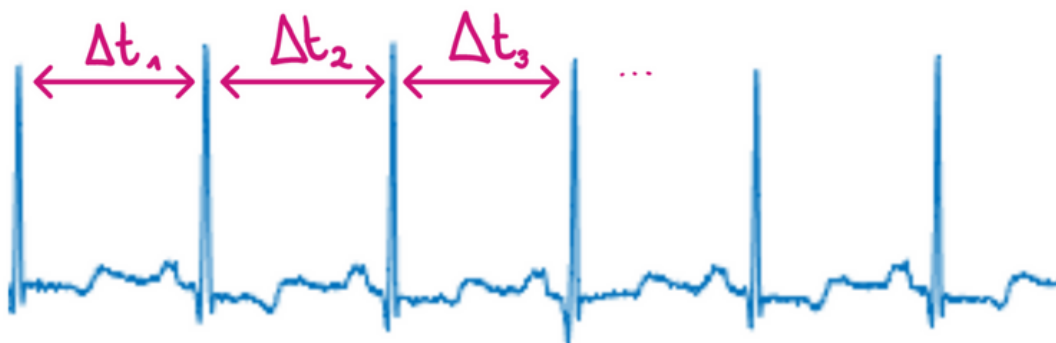
MÉTHODE 1

La première méthode est une méthode temporelle. Nous avons créé une fonction **ECG()** qui consiste à :

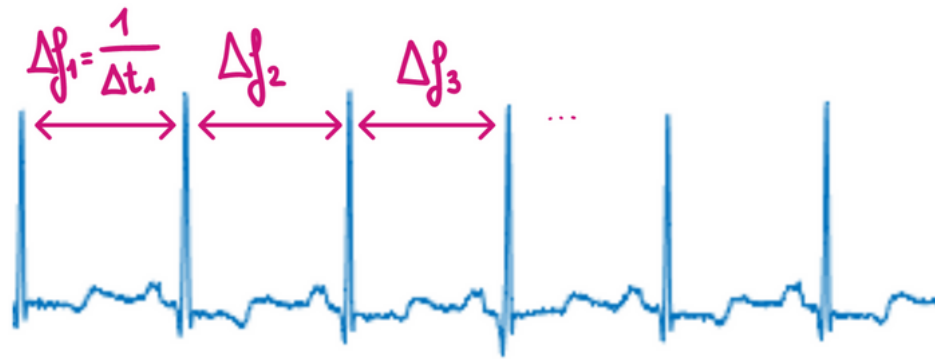
- Détecter les complexes QRS de l'ECG :
 - Afin de détecter chaque pic du signal et de récupérer leurs coordonnées. On les rangera dans une liste [x,y] ;
 - Utilisation de la fonction **findpeaks()** : prend en compte l'écart minimum entre 2 pics, ainsi que la hauteur minimale d'un complexe QRS pour éviter de compter les ondes P et T comme des contractions ventriculaires.



- Calculer l'écart de durée entre les pics :
 - Pour cela on récupère les valeurs de temps de deux pics consécutifs et on fait la soustraction de l'un moins l'autre ;
 - Création d'une variable $dt = x(i+1) - x(i)$.



- Transformer cette période en une fréquence :
 - On calcul la fréquence entre 2 pics ;
 - Création d'une variable $df=1/dt$
 - Cette étape sert principalement à pouvoir comparer les valeurs avec celles de la méthode 2.



- Calculer le nombre de battements par minutes (bpm) à chaque moment du signal :
 - On pose alors simplement $bpm=60*df$;
 - Pour éviter que la fréquence cardiaque affichée varie à chaque nouveau battement de cœur, on va calculer cette fréquence cardiaque tous les 3 à 5 pics.
- Calculer la moyenne de la fréquence cardiaque :
 - Nous permettra de comparer avec les valeurs obtenues pour la méthode 2
 - On pose :

$$bpm = \frac{n_i}{t} * 60$$

Avec n_i le nombre d'intervalles entre 2 pics non successifs

Avec t le temps entre ces 2 pics non successifs

La fréquence cardiaque mesurée au cours du temps, pris tous les 3 pics pour ne pas encombrer le patient de données :

Pour l'audio : 100.wav

Fréquence cardiaque : 74 battements par minutes
Fréquence cardiaque : 76 battements par minutes
Fréquence cardiaque : 92 battements par minutes
Fréquence cardiaque : 74 battements par minutes

Pour l'audio : 101.wav

Fréquence cardiaque : 69 battements par minutes
Fréquence cardiaque : 64 battements par minutes
Fréquence cardiaque : 69 battements par minutes
Fréquence cardiaque : 66 battements par minutes

Pour l'audio : 102.wav

Fréquence cardiaque : 73 battements par minutes
Fréquence cardiaque : 68 battements par minutes
Fréquence cardiaque : 74 battements par minutes
Fréquence cardiaque : 74 battements par minutes

Pour l'audio : 103.wav

Fréquence cardiaque : 69 battements par minutes
Fréquence cardiaque : 72 battements par minutes
Fréquence cardiaque : 68 battements par minutes
Fréquence cardiaque : 71 battements par minutes

Pour l'audio : 104.wav

Fréquence cardiaque : 73 battements par minutes
Fréquence cardiaque : 75 battements par minutes
Fréquence cardiaque : 69 battements par minutes
Fréquence cardiaque : 74 battements par minutes

Pour l'audio : 105.wav

Fréquence cardiaque : 83 battements par minutes
Fréquence cardiaque : 84 battements par minutes
Fréquence cardiaque : 79 battements par minutes
Fréquence cardiaque : 86 battements par minutes
Fréquence cardiaque : 83 battements par minutes

Pour l'audio : 106.wav

Fréquence cardiaque : 58 battements par minutes

Fréquence cardiaque : 57 battements par minutes

Fréquence cardiaque : 62 battements par minutes

Pour l'audio : 107.wav

Fréquence cardiaque : 73 battements par minutes

Fréquence cardiaque : 69 battements par minutes

Fréquence cardiaque : 73 battements par minutes

Fréquence cardiaque : 69 battements par minutes

Pour l'audio : 108.wav

Fréquence cardiaque : 61 battements par minutes

Fréquence cardiaque : 64 battements par minutes

Fréquence cardiaque : 61 battements par minutes

Fréquence cardiaque : 61 battements par minutes

Pour l'audio : 109.wav

Fréquence cardiaque : 93 battements par minutes

Fréquence cardiaque : 90 battements par minutes

Fréquence cardiaque : 89 battements par minutes

Fréquence cardiaque : 99 battements par minutes

Fréquence cardiaque : 92 battements par minutes

En moyenne, pour chaque signal, on obtient le nombre de battements par minute suivant :

Signal	100	101	102	103	104	105	106	107	108	109
Battements/ minutes	74	67	73	70	74	83	60	70	62	93

METHODE 2

La méthode fréquentielle consiste à travailler sur le spectre fréquentiel du signal, pour trouver la fréquence cardiaque du patient. Pour cela on rappelle la formule de la transformé de Fourier discrète d'une suite de N échantillons :

$$X(k) = \frac{1}{N} * \sum_{n=0}^{N-1} x(n) * e^{-\frac{2\pi jnk}{N}}$$

Avec $k \in [0, N - 1]$

TF à la fréquence $f = k \frac{F_e}{N}$

Plus la transformé de Fourier a de points, plus celle-ci est précise. C'est pour cela qu'on va agrandir la taille du vecteur de notre ECG en complétant par des zéros :

[... 0.1240 0.1161 0.1049 0.0937 0.0859 0.0674 0.0469 0.0263 0.0089 - 0.0103 0 0 0 0 0 0 0 ...]

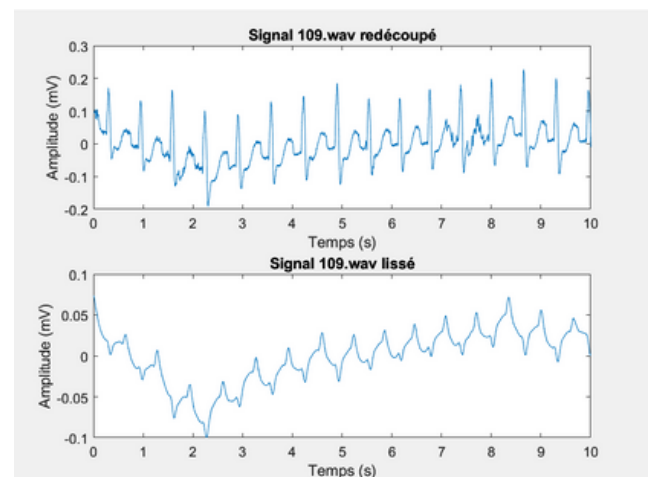
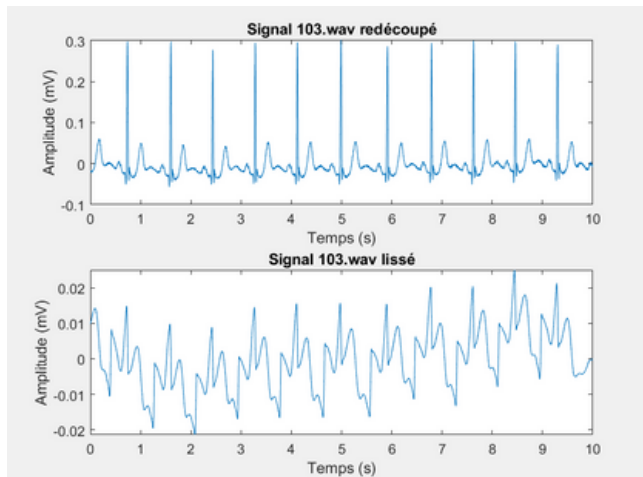
Signal recoupé
Liste de zéros

Le nombre de zéros ajoutés a lui aussi été fixé aléatoirement. Il s'agit simplement d'en mettre un grand nombre pour avoir un résultat plus précis derrière.

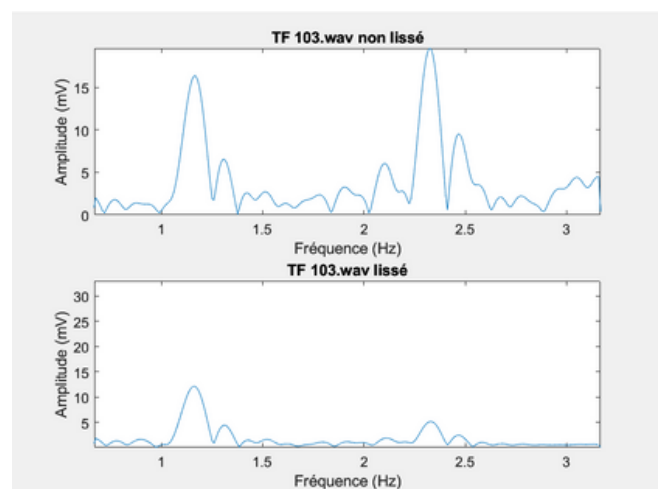
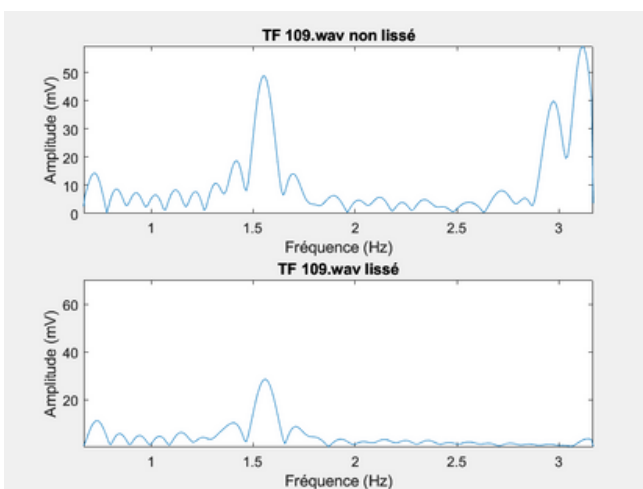
Il s'agit ensuite de lisser notre signal. Cela est utile car l'ECG est une fonction périodique mais qui n'apparaît pas comme telle dû à un trop grand nombre d'irrégularités, ce qui complique la lecture de sa transformé de Fourier.

On utilise alors la méthode de **moyenne mobile** pour lisser le signal :

- Moyenne car le principe est de faire des moyennes pour soulever les tendances de la courbe et enlever les fluctuations ;
- Mobile car celle-ci est recalculée de façon continue, pour chaque point.



Afin de visualiser l'effet du lissage sur la transformé de Fourier, traçons les transformé de Fourier des quatre courbes précédentes sur les signaux 103.wav et 109.wav :



La transformé de Fourier n'est ici représenté que de 0,67 à 3,167. La raison à cela, est que la fréquence cardiaque d'un être humain est compris entre 40 et 190 bpm.

$$40 \text{ bpm} < \text{Rythme cardiaque} < 190 \text{ bpm}$$

$$\frac{40}{60} \text{ Hz} < \text{Fréquence cardiaque} < \frac{190}{60} \text{ Hz}$$

$$0,67 \text{ Hz} < \text{Fréquence cardiaque} < 3,167 \text{ Hz}$$

Il ne reste plus qu'à récupérer l'indice du maximum de cette transformé de Fourier sur $[0.67, 3.167]$. Cet indice ensuite relié à sa valeur fréquentielle, puis multipliée par 60 pour obtenir le nombre de battements par minutes moyen.

En moyenne, pour chaque signal, on obtient le nombre de battements par minute suivant :

Signal	100	101	102	103	104	105	106	107	108	109
Battements/ minutes	75	68	73	70	73	83	59	71	62	93

COMPARAISON DES MÉTHODES

Comparons tout d'abord les moyennes de rythme cardiaque obtenus :

Signal	100	101	102	103	104	105	106	107	108	109
Méthode 1	74	67	73	70	74	83	60	70	62	93
Méthode 2	75	68	73	70	73	83	59	71	62	93



Identique

Diffère de 1 bpm

	Méthode 1 - Temporelle	Méthode 2 - Fréquentielle
Précision de l'analyse	+	—
Robustesse	+	—
Complexité algorithmique	+	++

Les deux méthodes sont aptes à fonctionner, mais il pourrait être préférable d'utiliser la méthode 1, temporelle.