



OBJET CONNECTÉ
2 NOVEMBRE 2024

LICENCE 3 1EEEE OPTION ELECTRONIQUE

Moniteur Cardiaque

Élèves :

RAMOUL YASMINE
ANTOINE ROBIN
BOUDABOUS KHADIJA
BOUZID TAHA

Enseignants

O. BERNAL
E. PEUCH
H. KAOUACH
H.C. SEAT
G. PRIGENT



Table des matières

1	Introduction	2
2	Traitement Numérique du Signal	2
2.1	Observation des signaux	2
2.2	Analyse spectrale	3
2.3	Filtrage	4
2.3.1	Filtre à réponse impulsionnelle finie « RIF »	5
2.3.2	Filtre à réponse impulsionnelle infinie « RII »	5
3	Mesures et Analyses	8
3.1	Mesure du BPM	8
3.1.1	Méthode de mesure	8
3.1.2	Tests statistiques	9
3.2	Mesure de la vitesse d'onde de pouls	15
3.2.1	Méthode de mesure	15
3.2.2	Tests statistiques	17
3.3	Mesure de l'indice de perfusion	18
3.3.1	Méthode de mesure	18
4	Conclusion	20

1 Introduction

Ce second rapport concerne la dernière partie du projet liée au traitement numérique du signal (TNS). Maintenant que nous avons le moyen d'acquérir notre signal réel, nous devons relever les données essentiels qui nous intéressent. L'objectif de cette partie va être premièrement de procéder à une analyse spectrale et une visualisation de périodogrammes des signaux PPG pour ensuite construire un filtre adapté à ces signaux afin de supprimer du mieux possible le bruit et les fréquences parasites restantes, perturbant l'extraction des données. Une fois le signal filtré, nous pourrions nous concentrer sur la détermination du rythme cardiaque associé et d'autres indicateurs de santé, pour établir finalement l'état de santé du patient à l'aide de tests statistiques.

2 Traitement Numérique du Signal

2.1 Observation des signaux

La première étape consiste à visualiser les signaux. Dans un premier temps, nous avons commencé par la visualisation d'un signal PPG disponible sur MOODLE comme la montre la figure 1. Ensuite, nous avons observé un exemple de nos signaux réels.

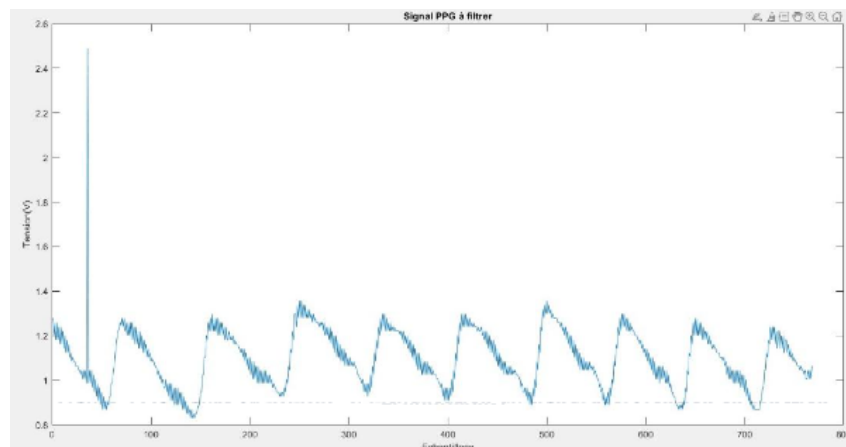


FIGURE 1 – Caption

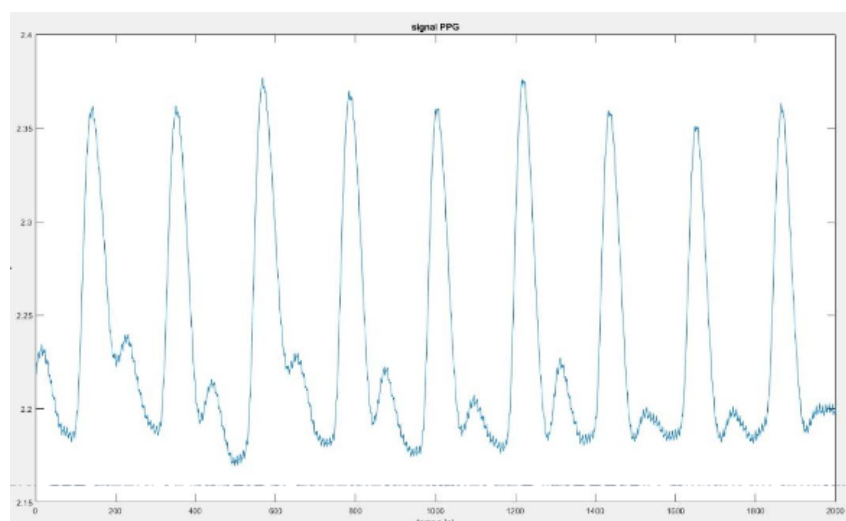


FIGURE 2 – Caption

À partir de ces deux figures, on constate que les deux signaux représentent du bruit. C'est ce que nous chercherons à éliminer grâce au filtrage, pour faciliter l'analyse des signaux ultérieurement.

2.2 Analyse spectrale

Nous disposons des signaux exploitables, nous pourrions alors entamer l'analyse. On a tracé tout d'abord à l'aide de la fonction FFT le Périodogramme qui est le moyen le plus facile d'évaluer la densité spectrale de puissance. (Programme MATLAB)

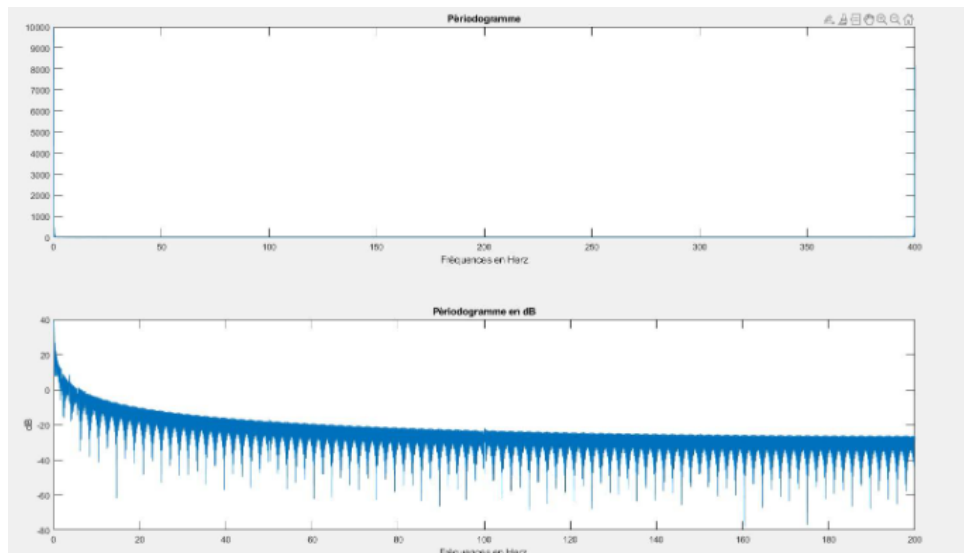


FIGURE 3 – Périodogramme via FFT

À partir de ce tracé, on observe que l'on a un grand pic en 0 Hz qui représente la composante continue (l'offset du signal) ainsi que les fréquences caractéristiques pour notre signal en basses fréquences, environ entre 1 et 10 Hz. Les pics de ces fréquences caractéristiques représentent les pics systoliques et diastoliques du rythme cardiaque.

Par la suite, nous allons essayer d'autres types de périodogrammes pour avoir un résultat plus précis sur les fréquences caractéristiques du signal car à partir de ce périodogramme, on ne peut pas distinguer entre les différents pics. Nous allons alors introduire les périodogrammes de Bartlett et de Welch comme les montrent les figures suivantes. (Programme MATLAB)

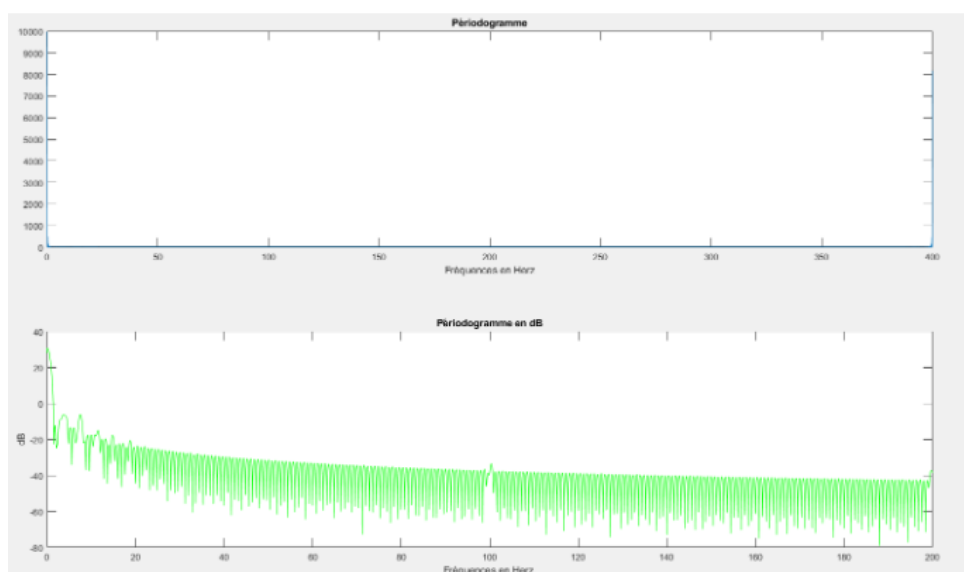


FIGURE 4 – Périodogrammes de Bartlett

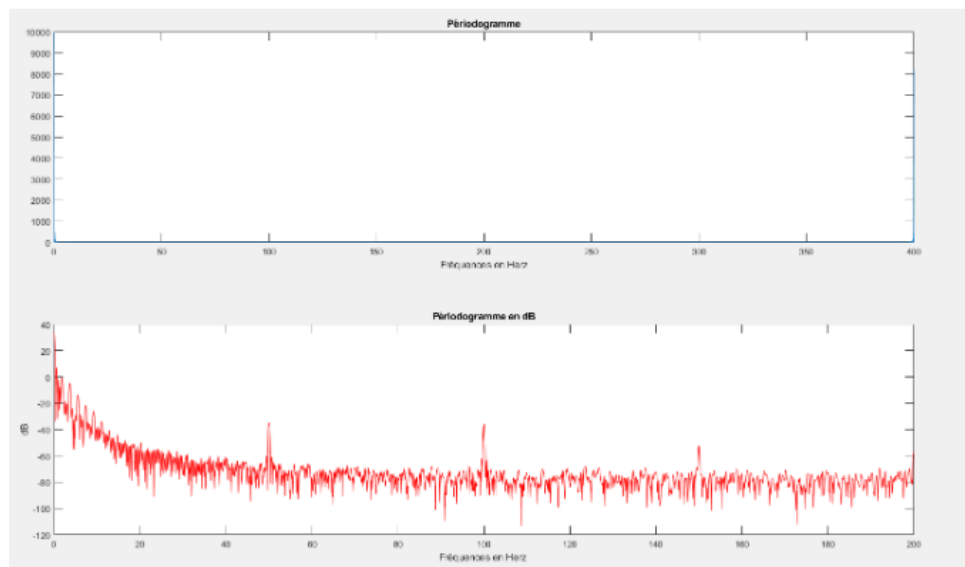


FIGURE 5 – Périodogrammes de Welch

En comparant ces deux types de périodogrammes, on remarque que Bartlett ne nous présente pas beaucoup d'informations comme celui de Welch qui nous permet de déterminer les différents pics plus précisément. C'est pour cette raison que ce périodogramme répond le mieux à nos attentes et c'est celui que nous conserverons.

2.3 Filtrage

Dans cette partie, on s'intéresse à éliminer le bruit présent dans les signaux observés à l'aide des filtres numériques. De ce fait, nous allons mettre en place un filtre passe-bas vu que les fréquences les plus utiles sont dans la bande passante [1 ; 10 Hz]. Le gabarit de notre filtre est le suivant :

- $F_e = 400 \text{ Hz}$
- $F_1 = 10 \text{ Hz}$
- $F_2 = 40 \text{ Hz}$

Nous avons essayé différents types de filtres en les générant avec la fonction `filtnum.m` étudiée en BE Traitement Numérique de Signal dans le but d'obtenir les coefficients de ces filtres.

2.3.1 Filtre à réponse impulsionnelle finie « RIF »

Nous avons mis en place un filtre RIF avec la fenêtre Hamming comme la montre la figure suivante.

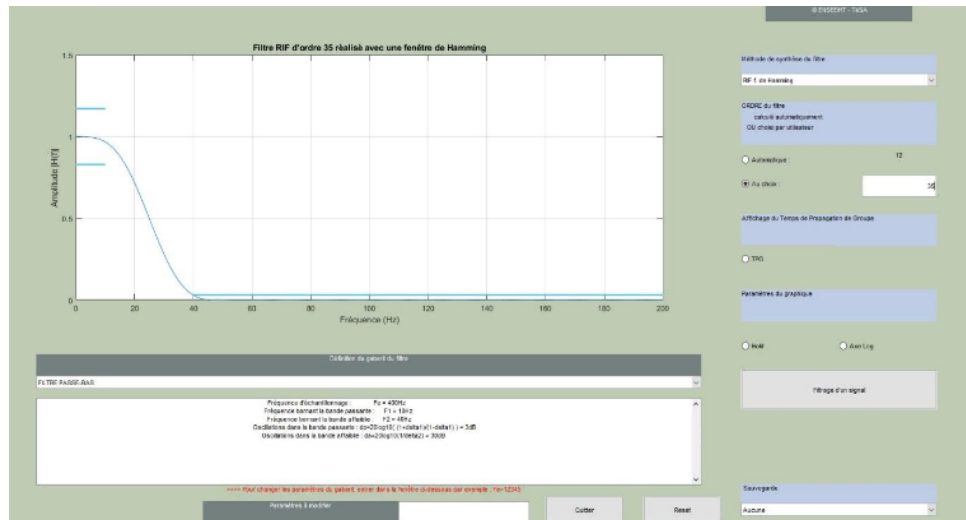


FIGURE 6 – Filtre RIF avec fenêtre Hamming

En mettant en œuvre ce filtre sur le signal PPG, on a obtenu le tracé suivant :

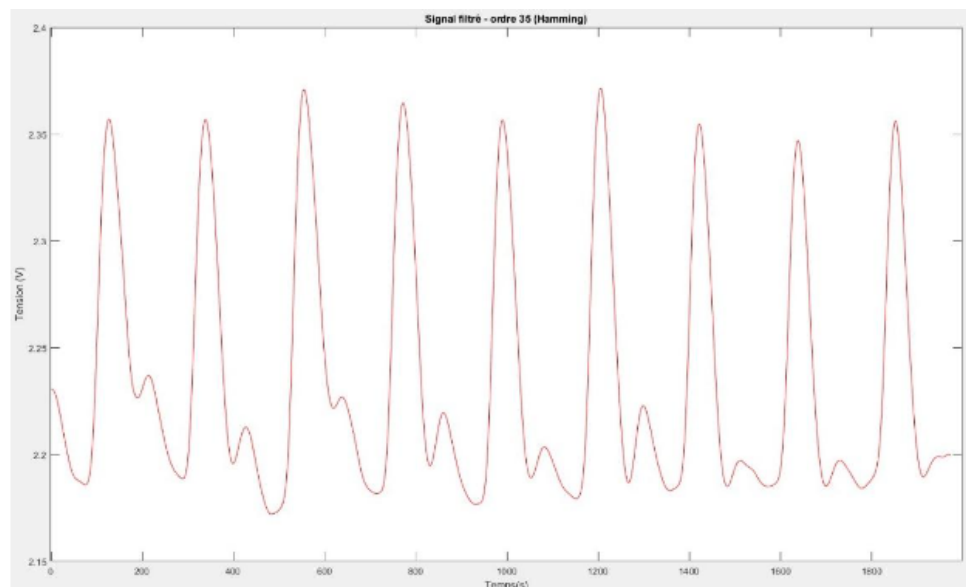


FIGURE 7 – Signal PPG filtré avec RIF

Il est évident que malgré l'intérêt du signal obtenu, notre filtre présente un ordre élevé de 35. L'inconvénient de ce filtre est d'avoir un coût de calcul élevé. Pour cela, nous allons essayer avec un filtre à réponse impulsionnelle infinie « RII ».

2.3.2 Filtre à réponse impulsionnelle infinie « RII »

Nous avons mis en place un filtre RII en utilisant les fenêtres Butterworth et Chebyshev (2).
Commençant par la fenêtre Butterworth :

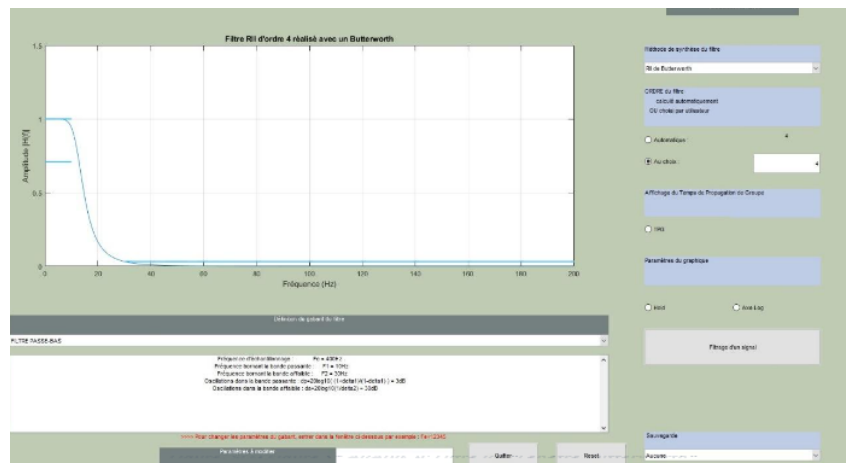


FIGURE 8 – Filtre RII avec fenêtre Butterworth

En mettant en œuvre ce filtre sur le signal PPG, on a obtenu le tracé suivant :

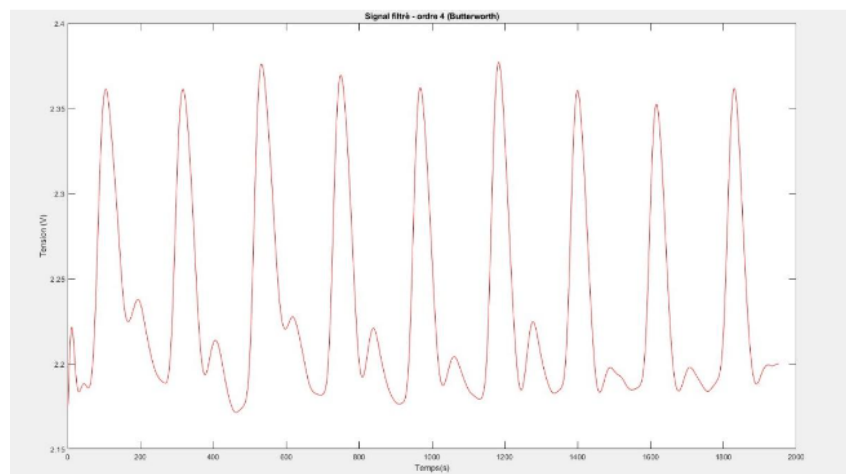


FIGURE 9 – Signal PPG filtré avec Butterworth

Ensuite, on a essayé avec fenêtre Chebyshev (2) :

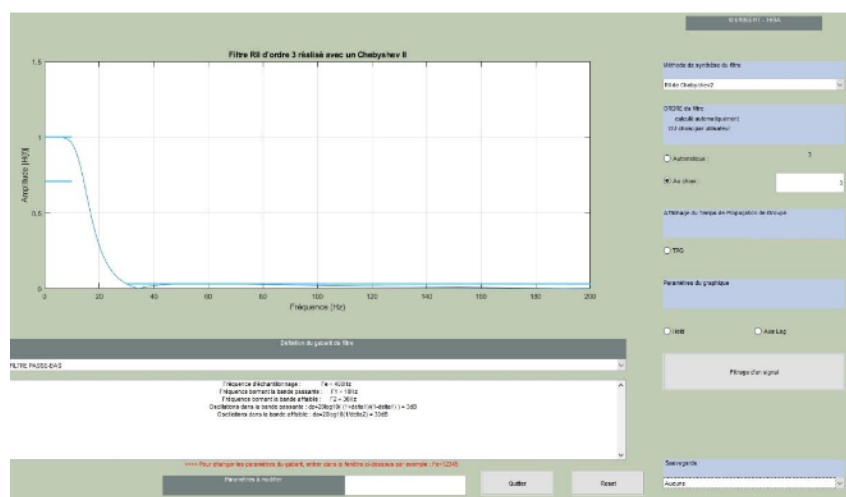


FIGURE 10 – Filtre RII avec fenêtre Chebyshev

En mettant en œuvre ce filtre sur le signal PPG, on a obtenu le tracé suivant :

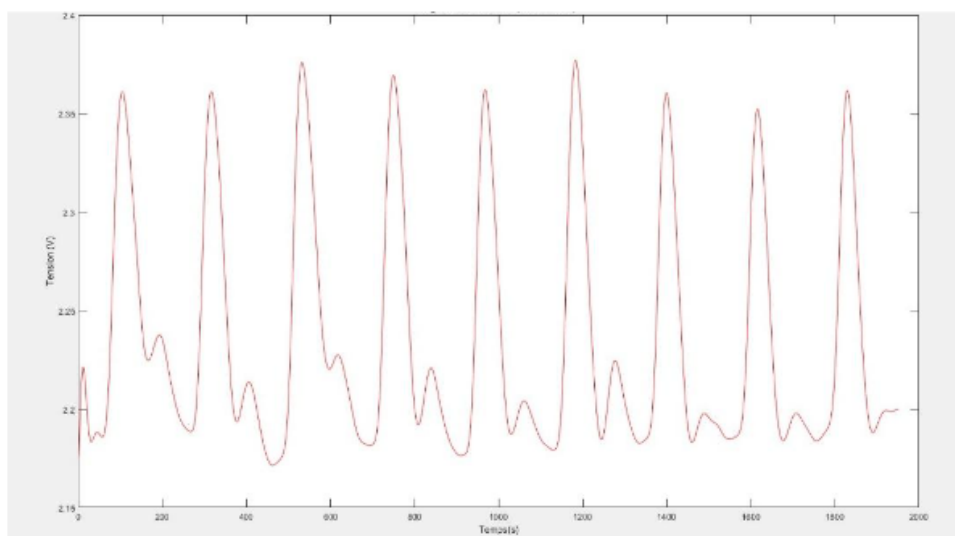


FIGURE 11 – Signal PPG filtré avec Chebyshev

On constate que l'ordre de ces filtres est plus faible par rapport au filtre RIF, on a obtenu un filtre d'ordre 4 pour la fenêtre Butterworth et un filtre d'ordre 3 pour la fenêtre Chebyshev (2). De plus, ces filtres RII ont un coût de calcul plus faible, donc ils sont plus rapides par rapport aux filtres RIF. De ce fait, on peut conclure que les filtres RII sont les plus adaptés pour un filtrage en temps réel. L'inconvénient de ce type de filtre est que sa réponse de phase n'est pas linéaire en fonction de la fréquence, ce qui peut engendrer des distorsions de phase, d'où le déphasage du signal.

Nous allons par la suite utiliser le filtre RII Chebyshev (2) car il possède l'ordre le plus faible (ordre 3) en le comparant avec le filtre RII Butterworth (ordre 4).

Lorsqu'on applique ce type de filtre, on remarque bien la réduction du bruit sur le signal.

3 Mesures et Analyses

3.1 Mesure du BPM

Maintenant que nous avons en notre possession un signal PPG filtré, nous pouvons désormais avoir accès proprement aux informations sur le flux sanguin de l'utilisateur. Dans un premier temps nous allons porter notre attention sur la détermination du battement par minute (BPM) Pour cela nous avons besoin de calculer la fréquence du signal.

3.1.1 Méthode de mesure

Une méthode simple pour déterminer la fréquence du signal consiste dans un premier temps à relever sa période moyenne. Pour cela nous devons réussir à visualiser une période du signal entre deux points clairement distinguables. Nous choisissons donc de relever l'écart de temps entre deux pics systoliques, correspondants aux maximums de chaque périodes. Une fois chaque écart relevé dans une liste nous pouvons en déduire la période moyenne qui nous servira à calculer la fréquence cardiaque moyenne par seconde, que nous convertiront en battements par minutes. En moyenne, un homme dans des conditions physiques ordinaires se situe entre 60 et 100 bpm.

Nous avons le choix entre deux méthodes pour créer la liste des pics systoliques du signal. nous pouvions utilisé la fonction `findpeaks` de matlab ou bien construire le code de détection des pics à la main. Les deux options ont été testées et conduisent au même résultat. La fonction créée par nous-même consiste à vérifier pour chaque point du signal (avec la boucle `for`) si il est un maximum local, c'est à dire supérieur au point précédent et au suivant, en vérifiant également que celui ci dépasse un certain seuil pour ne pas être confondu avec un pic dystolique, maximum local également mais de plus faible amplitude.

```
130
131
132     seuil= (max(signal)+min(signal)+std(signal))/2;    %Seuil
133     compteur = 0;                                     %compteur pour ajouter les pics à la liste
134     tp_maxs=zeros(1,5);                               %Initialisation du vecteurs temps des pics systoliques
135     val_maxs=zeros(1,5);                             %Initialisation du vecteur des valeurs des pics systoliques
136     for i =2:(length(signal)-1)
137         if (signal(i)>=signal(i+1)) &&(signal(i)>=signal(i-1)) && (signal(i)>seuil)
138             compteur=compteur+1;
139             tp_maxs(compteur)=i;
140             val_maxs(compteur)=signal(i);
141         end
142     end
143
```

FIGURE 12 – Programme de détection des pics systoliques

Maintenant que nous avons la liste `tps_maxs` qui contient la position de chaque pics systolique du signal, nous créons une nouvelle liste `periodepics` qui contient l'écart entre chaque pics successifs. A partir de cette liste et de la fréquence d'échantillonnage, nous pouvons déterminer la liste des fréquences associées en Herz, puis en déduire la liste des bpm pour en déduire le bpm moyen du signal.

```
148     %Détermination du bpm
149     periodepics=diff(tp_maxs)/Fe;    %Liste des périodes inter-pics
150     bpms=60*1./periodepics;          %Liste des bpms associés
151     bpm=mean(bpms)                   %Calcul du bpm moyen
152     ecart_type_bpm=std(bpms);        %Ecart type de la liste des bpms
```

FIGURE 13 – Programme de calcul du BPM

Pour le signal réel que nous avons extrait, nous trouvons en sortie du programme un bpm de 73.3862, ce qui est dans la moyenne, et cohérent avec ce qu'on aurait pu s'attendre.

3.1.2 Tests statistiques

Test Arythmie

Le test d'arythmie s'appuie sur l'écart-type de la période du signal. L'énoncé nous donne :

- Une personne malade si l'écart-type dépasse 15 bpm
- Une personne saine si l'écart-type est inférieur à 15 bpm

Nous fixons quelques hypothèses permettant notre étude. Nous supposons que la période suit une loi normale. Bien sûr, c'est une hypothèse de simplification et nous ne faisons aucune revendication sur la nature exacte de la distribution.

Dans l'idéal, nous aurions pu prendre plusieurs périodes dans notre acquisition ($n > 30$). Les échantillons suivront ainsi la même loi que T et leur somme suivra une loi normale. Nous pourrions aussi estimer l'écart-type par S^* . Cependant, vu la difficulté de prendre un nombre aussi élevé de périodes pendant notre acquisition, nous considérons dans toute la suite que T suit une loi normale.

L'hypothèse de maladie du patient se ramène alors au test statistique suivant :

- $H_0 : \sigma \leq 15$ bpm (sain)
- $H_1 : \sigma > 15$ bpm (malade)

Il s'agit d'un test composite. Le rapport de vraisemblance sera alors difficile à calculer puisqu'il va comprendre des bornes sup qui ne sont pas si évidentes à calculer. On pourra par contre simplifier ce test sous la forme suivante :

- $H_0 : \sigma = \sigma_0 = 15$ bpm (sain)
- $H_1 : \sigma = \sigma_1 > \sigma_0$ (malade)

Nous effectuons alors le test de Neyman-Pearson.

Nous acceptons H_0 si $RV = \frac{L(X, \sigma_1)}{L(X, \sigma_0)} < C'_\alpha$, avec $C'_\alpha = f(\alpha)$.

Ainsi,

$$RV = \frac{L(X, \sigma_1)}{L(X, \sigma_0)} = \left(\frac{\sigma_0}{\sigma_1}\right)^n \exp\left(\left(\frac{1}{2\sigma_0^2} - \frac{1}{2\sigma_1^2}\right) \sum_{i=1}^n (t_i - m)^2\right)$$

Pour un patient sain, cela équivaut à

$$RV < C'_\alpha \Rightarrow \frac{1}{2} \left(\frac{1}{\sigma_0^2} - \frac{1}{\sigma_1^2}\right) \sum_{i=1}^n (t_i - m)^2 < \ln(C'_\alpha) - n \ln\left(\frac{\sigma_0}{\sigma_1}\right)$$

D'où, puisque $\sigma_1 > \sigma_0$,

$$\sum_{i=1}^n (t_i - m)^2 \times \frac{1}{\sigma_0^2} < \left(\ln(C'_\alpha) - n \ln\left(\frac{\sigma_0}{\sigma_1}\right)\right) \times \frac{2}{\frac{1}{\sigma_0^2} - \frac{1}{\sigma_1^2}} \times \frac{1}{\sigma_0^2} = C_\alpha$$

Nous prenons $X = \sum_{i=1}^n \frac{(t_i - m)^2}{\sigma_0^2}$ comme notre statistique de test.

Ainsi, nous rejetons H_1 si $X < C_\alpha$.

Sous H_0 , nous avons $X \sim \chi_{n-1}^2$, puisque pour i allant de 1 à n , on a $t_i \sim N(\mu, \sigma_0^2)$. Ainsi,

$$\frac{(t_i - \mu)^2}{\sigma_0^2} \sim (0, 1)$$

Alors,

$$X = \sum_{i=1}^n \frac{(t_i - \mu)^2}{\sigma_0^2} \sim \chi^2(n)$$

Cependant, un degré de liberté est perdu puisque m est connue. Donc,

$$X \sim \chi_{n-1}^2$$

Nous fixons $\alpha = P(\text{prendre } H_1 \mid H_0 \text{ est vraie}) = P(X > C_\alpha \mid \sigma = \sigma_0)$.

D'où $\alpha = 1 - P(X \leq C_\alpha \mid \sigma = \sigma_0) = 1 - P(X \leq C_\alpha \mid X \approx \chi_{n-1}^2)$.

Nous avons donc $\alpha = 1 - F_{\chi_{n-1}^2}(C_\alpha)$.

Nous pouvons donc trouver C_α en calculant : $C_\alpha = \text{chi2inv}(1 - \alpha, n - 1)$. Dans notre code, nous fixons α à 0.05.

Nous mettons en place ce test sur Matlab, en posant $\sigma_0 = 100$ ms. Cette valeur d'écart-type correspond à la limite entre une personne en bonne santé et une personne en arythmie (arythmie si $\sigma_1 > \sigma_0$). le code matlab est alors le suivant

```
function [] = testArythmie(T)

% Calcul de la moyenne
n = length(bpms);
alpha = 0.05;
sigma0=15
moy = mean(bpms);
X = ( bpms- moy)/sigma0;
stat_test = sum(X.^2);
calpha = chi2inv(1 - alpha, n - 1);

if (stat_test < calpha)
    disp('malade')
else
    disp('sain')
end
end
```

On peut également calculer $\beta = P(T_n < C_\alpha \mid H_1)$:

$$\beta = P(T_n < F_{\chi_{n-1}^2}^{-1}(1 - \alpha) \mid H_1)$$

$$\beta = P(T_n < F_{\chi_{n-1}^2}^{-1}(1 - \alpha) \mid T_n \sim \chi_{n-1}^2)$$

$$\beta = F_{\chi_{n-1}^2}[F_{\chi_{n-1}^2}^{-1}(1 - \alpha)]$$

On détermine le test de puissance avec :

$$\Pi = 1 - \beta$$

$$\Pi = 1 - F_{\chi_{n-1}^2}[F_{\chi_{n-1}^2}^{-1}(1 - \alpha)]$$

on trace alors la puissance en fonction de alpha pour différentes valeurs de σ_1 le code utilise est le suivant

```
alpha = linspace(0, 1, 100);
sigma0 = 15;
figure()
n = length(bpms);
hold on
plotHandles = [];
labels = cell(1, 5);

for i = 1:5
    sigma1 = 15 + 3* i;
    pi = 1-chi2cdf(chi2inv(1-alpha, n - 1) * (sigma1 / sigma0), n - 1);
    h = plot(pi, alpha);
    plotHandles = [plotHandles h];
    labels{i} = ['\sigma_1 = ' num2str(sigma1)];
end
legend(plotHandles, labels, 'Location', 'Best')
```

```
xlabel('\alpha');
ylabel('\pi');
title('puissance de test arythmie en fonction de alpha');
hold off
```

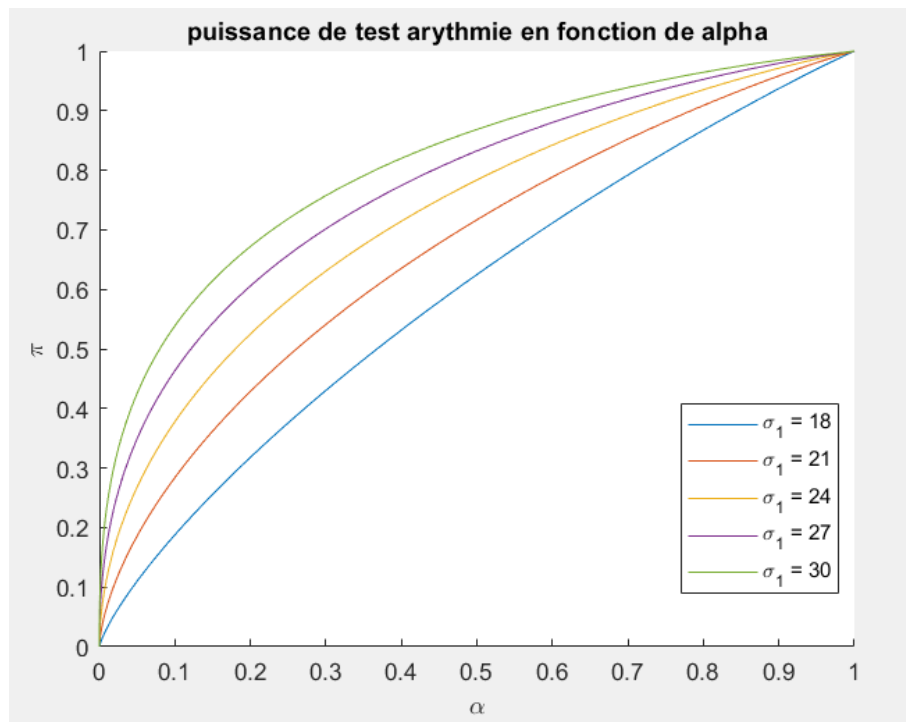


FIGURE 14 – Puissance de test arythmie en fonction de alpha

On remarque alors que plus σ_1 se rapproche de σ_0 plus la puissance de test diminue pour des mêmes valeurs de α . Il est aussi à noter que plus la période de l'acquisition de signal est longue, plus notre test sera puissant.

Test tachycardie

Pour les tests de bradycardie ou tachycardie l'hypothèse se fait sur la moyenne **tachycardie**

- H_1 : Le patient est atteint de tachycardie si $m_1 = 100$
- H_0 : Le patient est sain si $m = m_0 < m_1$

Nous acceptons l'hypothèse H_0 si

$$\frac{L(T_1, \dots, T_n \mid m_1)}{L(T_1, \dots, T_n \mid m_0)} < C_\alpha$$

$$\prod_{i=1}^n \frac{\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t_i - m_1}{\sigma}\right)^2}}{\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t_i - m_0}{\sigma}\right)^2}} < C_\alpha$$

$$\prod_{i=1}^n e^{-\frac{1}{2}\left[\left(\frac{t_i - m_1}{\sigma}\right)^2 - \left(\frac{t_i - m_0}{\sigma}\right)^2\right]} < C_\alpha$$

$$-\frac{1}{2\sigma^2} \sum_{i=1}^n [(t_i - m_1)^2 - (t_i - m_0)^2] < \ln(C'_\alpha)$$

$$\begin{aligned}
 -\frac{1}{2\sigma^2} \sum_{i=1}^n [(t_i - m_1 + t_i - m_0)(t_i - m_1 - t_i + m_0)] &< \ln(C'_\alpha) \\
 \sum_{i=1}^n [(m_0 - m_1)(2t_i - (m_0 + m_1))] &> -2\sigma^2 \ln(C'_\alpha) \\
 -2(m_1 - m_0) \sum_{i=1}^n t_i &> -2\sigma^2 \ln(C'_\alpha) - m_1^2 + m_0^2 \\
 \sum_{i=1}^n t_i &< \frac{\sigma^2 \ln(C'_\alpha) + \frac{m_1^2 - m_0^2}{2}}{m_1 - m_0} = C_\alpha
 \end{aligned}$$

on pose $X = \sum_{i=1}^n t_i$ Sous l'hypothèse H_0 , $X \sim N(m_0, \sigma^2)$ donc

$$\sum_{i=1}^n \frac{t_i - m_0}{\sigma} \sim N(0, 1)$$

$$\alpha = P(X > C_\alpha \mid H_0)$$

$$\alpha = P(X > C_\alpha \mid T_n \sim N(m_0, \sigma^2))$$

$$\alpha = P\left(\frac{X - m_0}{\sigma^2} > \frac{C_\alpha - m_0}{\sigma^2} \mid \frac{X - m_0}{\sigma^2} \sim N(0, 1)\right)$$

$$\alpha = 1 - P\left(\frac{X - m_0}{\sigma^2} < \frac{C_\alpha - m_0}{\sigma^2} \mid \frac{X - m_0}{\sigma^2} \sim N(0, 1)\right)$$

$$\alpha = 1 - F_{0,1}\left(\frac{C_\alpha - m_0}{\sigma^2}\right)$$

$$\frac{C_\alpha - m_0}{\sigma^2} = F_{0,1}^{-1}(1 - \alpha)$$

$$C_\alpha = \sigma^2 F_{0,1}^{-1}(1 - \alpha) + m_0$$

On implémente ce test sur matlab grâce au code suivant :

```

function [] = testTachycardie(T)

% Calcul de la moyenne

alpha = 0.05;
moy = mean(bpms);
sigma = std(bpms); % Estimation de l'écart-type

% Calcul de la statistique de test
stadetest = sum(bpms);
m0 = 75; % Moyenne sous H0
m1 = 100; % Moyenne sous H1

% Calcul de C_alpha
calpha = sigma * norminv(1 - alpha) + m0;

% Décision du test
if (stadetest > calpha)
    disp('tachycardie')

```

```
else
    disp('sain')
end
end
```

On peut également calculer $\beta = P(X < C_\alpha \mid H_1)$

$$\beta = P(X < C_\alpha \mid X \sim \mathcal{N}(m_1, \sigma^2))$$

$$\beta = P\left(\frac{X - m_1}{\sigma} < \frac{C_\alpha - m_1}{\sigma^2} \mid Z \sim \mathcal{N}(0, 1)\right)$$

$$\beta = F_{0,1}\left(\frac{C_\alpha - m_1}{\sigma^2}\right)$$

$$\beta = F_{0,1}\left[F_{0,1}^{-1}(1 - \alpha) + \frac{m_0 - m_1}{\sigma}\right]$$

On détermine le test de puissance avec :

$$\Pi = 1 - \beta$$

$$\Pi = 1 - F_{0,1}\left[F_{0,1}^{-1}(1 - \alpha) + \frac{m_0 - m_1}{\sigma}\right]$$

On pourra alors tracer la puissance de test en fonction de alpha comme précédemment grâce au code matlab suivant :

```
alpha = linspace(0,1,100);
sigma =std(bpms);
figure()
hold on
m1=100
plotHandles = [];
labels = cell(1, 5);
for i = 1:5
    m0 = m1-8 * i;
    C_alpha = sigma * norminv(1- alpha) + m0;
    pi = 1-normcdf((C_alpha - m1) / sigma);
    h = plot(pi, alpha);
    plotHandles = [plotHandles h];
    labels{i} = ['m1 = ' num2str(m0)];
end
legend(plotHandles, labels, 'Location', 'Best')
xlabel('\alpha');
ylabel('\pi');
title('puissance de test tachycardie en fonction de alpha');
hold off
```

On trace alors la courbe ROC de ce test permettant de visualiser la puissance du test en fonction de alpha pour différents valeurs de 1. Nous obtenons

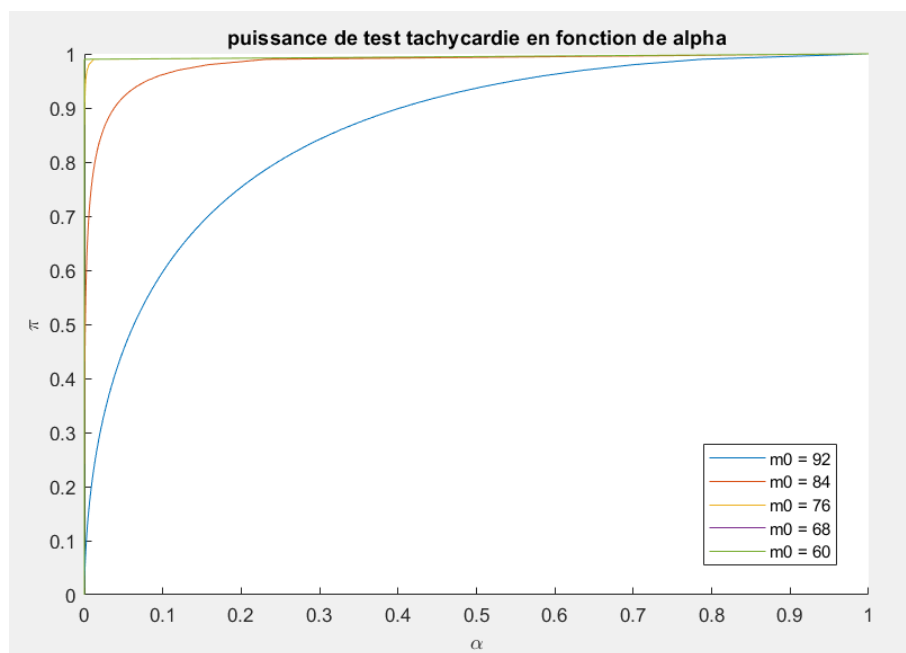


FIGURE 15 – Puissance de test tachycardie en fonction de alpha pour différentes valeurs de m_0

test bradychardie

De même, puisque ce test s'appuie sur la moyenne et que T suit toujours la même loi, on aura la même statistique de test, c'est la décision qui changera. En effet, pour la bradycardie, on a :

- H_1 : Le patient est atteint de bradycardie si $m_1 = 50$ bpm
- H_0 : Le patient est sain si $m = m_0 > m_1$

Par analogie avec le test précédent, on aura qu'un patient est sain si et seulement si :

$$\sum_{i=1}^n t_i > \frac{\sigma^2 \ln(C'_\alpha) + \frac{m_1^2 - m_0^2}{2}}{m_1 - m_0} = C_\alpha$$

car maintenant on a $m_0 > m_1$. On calcule alors C_α , on a :

$$\alpha = P(X < C_\alpha \mid T_n \sim \mathcal{N}(m_0, \sigma^2))$$

donc :

$$\alpha = P\left(\frac{X - m_0}{\sigma^2} < \frac{C_\alpha - m_0}{\sigma^2} \mid \frac{X - m_0}{\sigma^2} \sim \mathcal{N}(0, 1)\right)$$

Alors :

$$\alpha = F_{0,1}\left(\frac{C_\alpha - m_0}{\sigma^2}\right)$$

ou encore :

$$C_\alpha = F_{0,1}^{-1}(\alpha)\sigma^2 + m_0$$

de meme

$$\beta = 1 - F_{0,1}\left[\sigma F_{0,1}^{-1}(\alpha) + \frac{m_0 - m_1}{\sigma}\right]$$

et

$$\Pi = F_{0,1}\left[\sigma F_{0,1}^{-1}(\alpha) + \frac{m_0 - m_1}{\sigma}\right]$$

On trace alors la courbe ROC du test bradycardie on modifiant un peu celui de la tachycardie(valeur de m_1 et on fait augmenter m_0 au lieu de la diminuer) et on obtient alors :

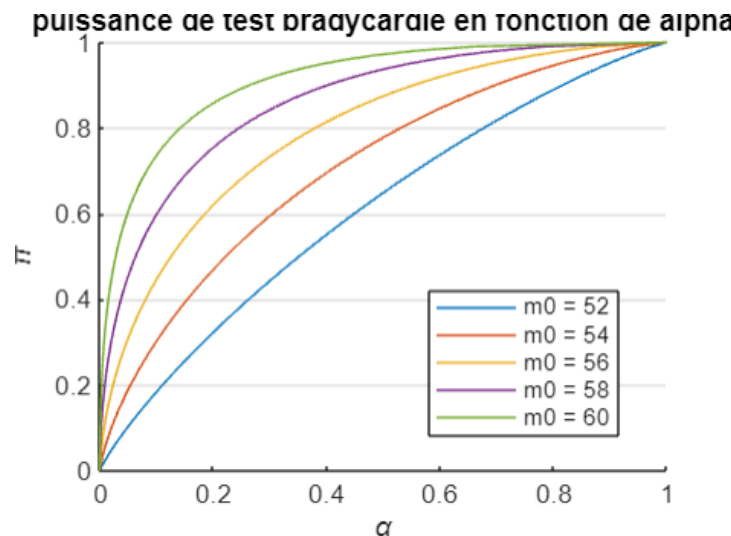


FIGURE 16 – Puissance de test bradycardie en fonction de alpha pour différentes valeurs de m_0

On remarquera alors que les deux test sont très similaires en effet en fixant une valeur m_1 limite qui détermine la maladie, plus on s'éloigne de cette valeur et plus le test sera puissant.

3.2 Mesure de la vitesse d'onde de pouls

Une autre mesure possible permettant de caractériser la rigidité des artères est la vitesse de propagation de l'onde de pouls. En effet, plus les artères sont rigides, plus la vitesse de propagation de l'onde de pouls sera élevée.

3.2.1 Méthode de mesure

La vitesse d'onde de pouls s'exprime de la manière suivante :

$$PWV = \frac{2 \cdot d_{\text{coeur} \rightarrow \text{doigt}}}{\overline{\Delta T}}$$

avec $d_{\text{coeur} \rightarrow \text{doigt}}$ la distance entre le coeur d'une personne et point d'acquisition du signal PPG (dans notre cas c'est l'extrémité de son doigt) et ΔT le délai entre le pic systolique et le pic diastolique dans chaque période du signal et on note $\overline{\Delta T}$ sa moyenne empirique. En revanche, comme notre signal PPG est de type A, donc le pic diastolique est caractérisé par une baisse de décroissance (donc de dérivée proche de zéro) qui complique sa détection directe .Pour cette raison, on dérive notre signal :

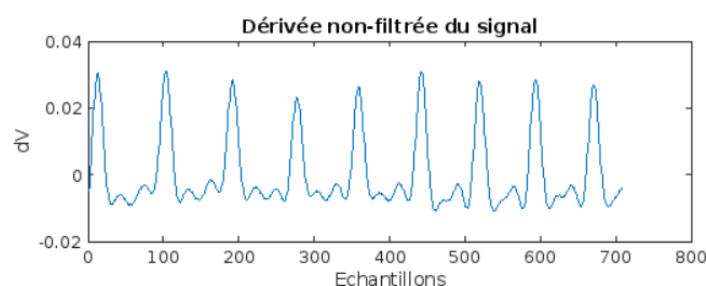


FIGURE 17 – Dérivée non filtrée du signal réel

On remarque que la dérivée première présente du bruit ce qui rend difficile la distinction des diastoles (pics proches de zéro). Il sera utile de filtrer dérivée de signal par un filtrage de Savitzky-Golay : c'est une

technique de lissage du signal en concernant les caractéristiques importantes comme les pics et les creux (et non pas comme les autres filtres qui peuvent atténuer ou déformer ces caractéristiques), en plus qu'il s'applique directement dans le domaine temporel.

Le signal obtenu après filtrage est le suivant :

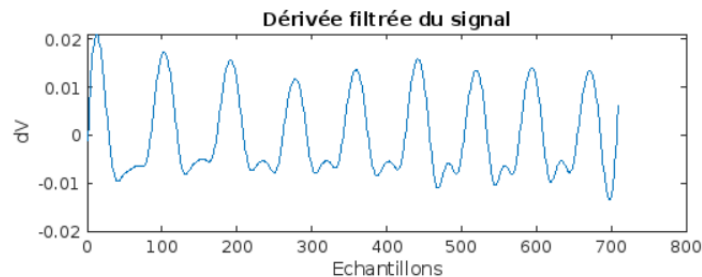


FIGURE 18 – Dérivée filtrée du signal réel par un filtrage de Savitzky-Golay

Ensuite, afin de détecter les diastoles, on procède par la même méthode de détection des pics systoliques pour calculer le BPM :

```
% On identifie l'endroit où la dérivée est presque nulle (en dessous d'un certain seuil)
% et où elle présente un maximum local.
compteur =0;
seuil_derivee=std(dsF);
for i = 2:length(signal)-1
    if (dsF(i) >= dsF(i-1)) && (dsF(i) >= dsF(i+1)) && (dsF(i)<= seuil_derivee)
        compteur=compteur+1;
        tps_dystoles(compteur) = i;
        val_dystoles(compteur) = signal(i);
    end
end
% Affichage des dystoles sur le signal filtré
plot(tps_dystoles, val_dystoles,'o');

%Pour la suite des calculs, si le signal filtré fait apparaître un dystole avant un systole, alors
if tps_dystoles(1,1)<tp_maxs(1,1)
    tps_dystoles=tps_dystoles(2:end);
end

%pour réaliser cette opération on met les deux listes à la même taille,
%quitte à perdre les dernières valeurs (on part du principe que l'on aura au moins 7 périodes complètes)
deltaTs=(tps_dystoles(1:7)-tp_maxs(1:7))/Fe;
deltaTm=mean(deltaTs);

%distance bras coeur
d=1;
% Calcul du PMW moyen
PMW=2*d/deltaTm
```

FIGURE 19 – Programme de détection des diastoles et du calcul du PWV

On trouve alors une valeur $PWV = 5.8824$, ce qui est cohérent en ordre de grandeur avec les valeurs que l'on peut retrouver dans le tableau de la figure 20.

3.2.2 Tests statistiques

	Catégorie de pression artérielle				
	Optimale	Normale	Élevée	HT Niveau 1	HT Niveaux 2/3
PWV < 5,6	110	33	7	1	0
5,6 ≤ PWV < 6	85	32	8	2	1
6 ≤ PWV < 6,4	91	46	12	2	0
6,4 ≤ PWV < 6,8	74	51	15	3	1
6,8 ≤ PWV < 7,2	46	48	15	4	1
7,2 ≤ PWV < 7,6	22	33	13	3	1
7,6 ≤ PWV	10	35	14	10	2

FIGURE 20 – Tableau des catégories de pression artérielle en fonction du PWV

On peut s'appuyer sur le tableau pour déterminer les probabilités d'appartenance à chaque classe de pression. En sommant les valeurs de chaque ligne et en divisant le nombre associé à chaque classe de pression par cette somme, on obtient alors :

PWV	Optimale	Normale	Élevée	HT Niveau 1	HT Niveau 2/3
PWV < 5,6	$0,728 = \frac{110}{151}$	$0,219 = \frac{33}{151}$	0,046	0,007	0
5,6 ≤ PWV < 6	0,664	0,25	0,062	0,016	0,007
6 ≤ PWV < 6,4	0,603	0,305	0,079	0,013	0
6,4 ≤ PWV < 6,8	0,528	0,354	0,104	0,021	0,007
6,8 ≤ PWV < 7,2	0,404	0,421	0,132	0,035	0,009
7,2 ≤ PWV < 7,6	0,306	0,458	0,181	0,042	0,014
7,6 ≤ PWV	0,141	0,493	0,197	0,141	0,028

TABLE 1 – Probabilité d'appartenance à une classe artérielle selon le PWV

Ensuite, on peut calculer la probabilité que le PWV soit dans l'un de ces intervalles en utilisant la loi des probabilités conditionnelles sur chacune des lignes. On obtient :

- $P(\text{PWV} < 5,6) = 0,182$
- $P(5,6 \leq \text{PWV} < 6) = 0,154$
- $P(6 \leq \text{PWV} < 6,4) = 0,182$
- $P(6,4 \leq \text{PWV} < 6,8) = 0,173$
- $P(6,8 \leq \text{PWV} < 7,2) = 0,137$
- $P(7,2 \leq \text{PWV} < 7,6) = 0,087$
- $P(\text{PWV} \geq 7,6) = 0,085$

Puis, on désire calculer la probabilité d'appartenance à chaque classe de pression. Par exemple, pour la classe optimale, on a :

$$P(\text{Optimale}) = \sum P(\text{Optimale et PWV} \in N_i) = \sum P(\text{PWV} \in N_i) \times P(\text{Optimale} | \text{PWV} \in N_i)$$

avec N_i étant les différents intervalles pris par PWV.

En procédant d'une manière similaire pour les autres classes de pression, on trouve :

Classe	Optimale	Normale	Élevée	HT Niveau 1	HT Niveau 2/3
Proba d'apparition 0,007	0,529	0,334	0,100	0,030	

TABLE 2 – Classe et probabilité d'appartenance

3.3 Mesure de l'indice de perfusion

La dernière mesure effectuée est celle de l'indice de perfusion. Celui-ci donne une information sur la puissance du signal de pouls.

3.3.1 Méthode de mesure

On reprend d'abord l'expression de l'indice de perfusion :

$$PI = \frac{AC_{TOP} - AC_{LOW}}{DC} \times 100\% \quad (3.1)$$

avec DC la moyenne empirique :

$$DC = \frac{1}{N} \sum_{n=1}^N s(n), \quad (3.2)$$

où s désigne le signal acquis et N sa longueur (en échantillons). On commence d'abord par déterminer les différents paramètres de l'équation et les tracer sur Matlab :

```
% Détection des pics systoliques
[pks, locs] = findpeaks(signal, 'MinPeakHeight', 0.5, 'MinPeakDistance', Fe/2);

% Calcul des composantes DC et AC
N = length(signal);
DC = mean(signal) * ones(size(signal)); % Moyenne du signal pour la composante DC

% Interpolation spline sur les pics systoliques pour ACTOP
x = 1:length(signal);
ACTOP = spline(locs, pks, x); % Interpolation spline pour ACTOP

% Détection des minima pour ACLOW

[pks_min, locs_min] = findpeaks(-signal, 'MinPeakDistance', Fe/2); % Utilisation de -signal pour trouver les minima
pks_min = -pks_min; % Inversion des valeurs pour retrouver les valeurs originales

ACLOW = spline(locs_min, pks_min, x); % Interpolation spline pour ACLOW
```

FIGURE 21 – Programme de détermination de ACTOP, ACLOW et DC

En traçant chacun de ces paramètres sur notre signal PPG, on obtient la figure suivante :

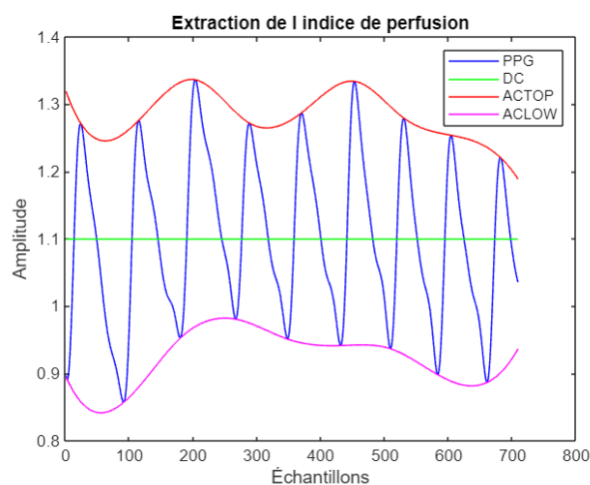


FIGURE 22 – Extraction de l'indice de perfusion

On trace enfin la courbe de l'indice de perfusion en implémentant les lignes de codes suivantes :

```
% Calcul de l'indice de perfusion
PI = (ACTOP - ACLOW) ./ DC * 100;
% Affichage de l'indice de perfusion
figure;
plot(x, PI, 'k');
xlabel('Échantillons');
ylabel('Indice de perfusion (%)');
title('Indice de perfusion');
```

FIGURE 23 – Programme de détermination de l'indice de perfusion

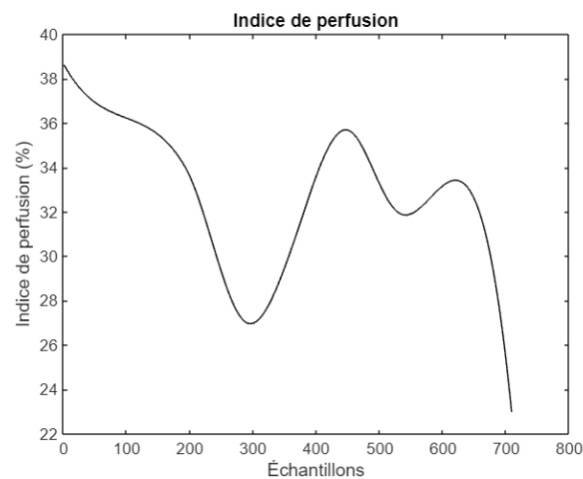


FIGURE 24 – Tracé de l'indice de perfusion

4 Conclusion

Après avoir défini un filtre efficace nous permettant d'atténuer les composantes parasites du signal originel, nous avons pu réaliser l'extraction des données d'un signal PPG réel de type A exploitable. De ce signal nous avons extrait le rythme cardiaque (bpm), la vitesse d'onde de pouls ainsi que l'indice de perfusion. L'extraction des données nous aura forcé à traiter le signal sous différents aspects, sur lui même ou ses dérivées et grâce à des tests statistiques, nous pouvons dorénavant déterminer l'état de santé de l'utilisateur.

Cette seconde partie du projet nous aura donc permis de développer nos compétences à analyser et identifier les caractéristiques d'un signal numérique. Nous avons dû pour cela apprendre à se séparer les tâches et à regrouper le travail de chacun dans un même code Matlab. Malgré tout, même si le projet fonctionne, il reste quelques pistes d'améliorations à exploiter pour la suite, notamment en ce qui concerne le code de la partie TNS qui peut grandement être optimisé et peut-être mieux organisé avec moins de bugs. A l'avenir, il aurait été judicieux de rendre également le code plus automatique, afin qu'il y ait moins de manipulations laborieuses à faire pour extraire les données d'un signal que l'on aurait récolté à partir de l'oscilloscope. Enfin une amélioration serait de rendre le code plus adaptatif en fonction du signal, qu'il puisse fonctionner avec des signaux qui ont une allure moins claire et plus bruités.

Malgré les nombreuses difficultés que nous avons rencontrées tout au long de la conception, nous tenons à remercier l'ensemble des professeurs encadrant qui nous ont accompagnés et apporter leur aide et leur soutien durant tout le projet.