

# TS225 - Projet Signal

Maxime PETERLIN - Gabriel VERMEULEN

ENSEIRB-MATMECA, Bordeaux

12 janvier 2015

## Table des matières

<b>Introduction</b>	<b>2</b>
<b>1 Méthode par FFT</b>	<b>2</b>
1.1 Étude théorique . . . . .	2
1.2 Implémentation sur Matlab . . . . .	3
1.2.1 Signaux générés . . . . .	3
1.2.2 Signaux réels . . . . .	3
<b>2 Approche par méthode AR</b>	<b>3</b>
2.1 Étude théorique . . . . .	3
2.2 Implémentation sur Matlab . . . . .	8
2.2.1 Signaux générés . . . . .	8
2.2.2 Signaux réels . . . . .	9
<b>3 Approche par méthode de Capon</b>	<b>9</b>
3.1 Implémentation sur Matlab . . . . .	10
3.1.1 Signaux générés . . . . .	10
3.1.2 Signaux réels . . . . .	10
<b>4 Analyse temps-fréquence</b>	<b>11</b>
<b>5 Bilan de l'organisation</b>	<b>12</b>
<b>Conclusion</b>	<b>13</b>

# Introduction

Ce projet de traitement du signal a pour but l'analyse spectrale de signaux de paroles, et ce à l'aide de plusieurs méthodes. Pour se faire, nous étudierons théoriquement les différentes approches abordées par ce projet, en l'occurrence la FFT, le modèle autorégressif et la méthode de Capon, ainsi qu'une implémentation sur MATLAB pour chaque méthode.

## 1 Méthode par FFT

### 1.1 Étude théorique

La FFT (*Fast Fourier Transform*) est un algorithme de calcul de la transformée de Fourier discrète. Pour appliquer ce dernier, le signal que l'on cherche à étudier doit être constitué de  $2^N$  points, avec  $N \in \mathbb{N}$ . Si le signal analysé ne possède pas un nombre de points suffisant, on peut compléter ce dernier par des zéros, ce procédé est nommé zero-padding. Le nombre de point étant augmenté dans le domaine temporel, on gagne en précision dans le domaine fréquentiel.

Matlab nous fournit de nombreux outils implémentant cette algorithme. On peut ainsi s'aider des fonctions suivantes :

**fft** qui permet de calculer une transformée de Fourier discrète avec l'algorithme de la transformée de Fourier rapide ;

**fft2** qui permet de calculer une transformée de Fourier 2D discrète avec l'algorithme de la transformée de Fourier rapide ;

**fftshift** qui permet de recentrer la transformée de Fourier par rapport à la composante de fréquence nulle.

Lorsqu'on étudie des signaux, on ne peut le faire sur des temps infinis. Ainsi, afin d'avoir un support temporel qui soit fini, on fenêtrise les signaux étudiés. Nous allons, ici, analyser spectralement l'impact d'un tel fenêtrage sur un signal sinusoïdal, que l'on échantillonnera par la suite, représenté par la fonction suivante :  $s(x) = \sin(2\pi f_0 t)$ .

On fenêtrise ce signal rectangulairement par une porte de taille N, on obtient alors :

$$f(x) = \sin(2\pi f_0 t) \cdot \Pi_N(t)$$

Puis on échantillonne ce dernier à une fréquence  $f_{ech}$  :

$$\tilde{f}(x) = \sin(2\pi f_0 t) \cdot \Pi_N(t) \cdot \text{III}_{\frac{1}{f_{ech}}}(t)$$

On calcule alors la transformée de Fourier de ce signal :

$$\tilde{F}(v) = \frac{1}{2j} [\delta(v + f_0) - \delta(v - f_0)] \otimes N \cdot \text{sinc}(Nv) \otimes f_{ech} \text{III}_{f_{ech}}(v) \quad (1)$$

Finalement, on a :

$$\tilde{F}(v) = N \cdot f_{ech} \sum_{k=-\infty}^{+\infty} \text{sinc}(N \cdot k \cdot f_{ech} + N \cdot f_0) - \text{sinc}(N \cdot k \cdot f_{ech} - N \cdot f_0)$$

## 1.2 Implémentation sur Matlab

### 1.2.1 Signaux générés

Le signal que nous avons généré est une somme de sinusoides :  $s(t) = \sin(2\pi f_0 t) + 0.5\sin(2\pi f_1 t)$  avec  $f_0 = 30$  Hz et  $f_1 = 200$  Hz.

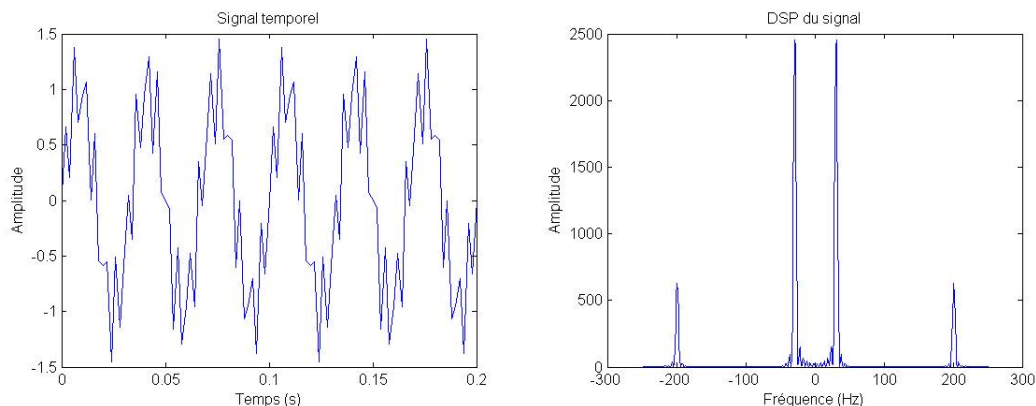


FIGURE 1 – Signal temporel et DSP

### 1.2.2 Signaux réels

Nous avons ensuite appliqué la FFT à un signal *réel*, qui est ici un signal voisé d'une durée de 50ms.

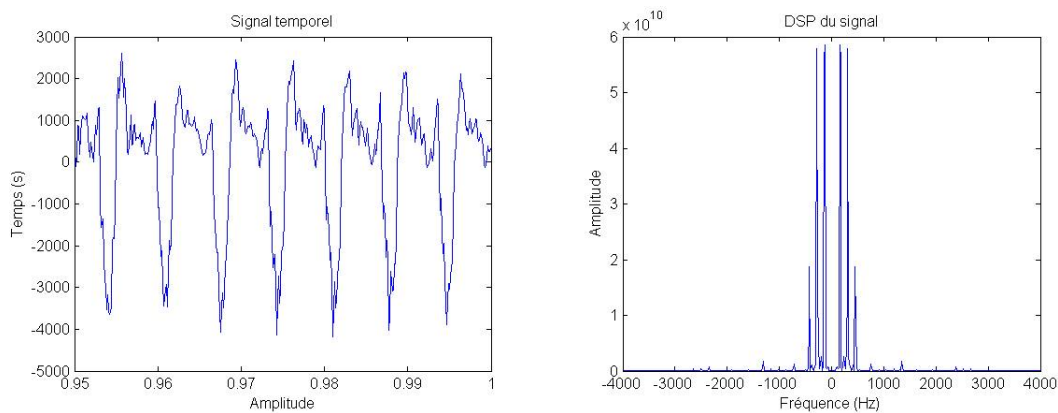


FIGURE 2 – Signal temporel et DSP

## 2 Approche par méthode AR

### 2.1 Étude théorique

Le but de cette partie est l'étude des processus auto-regressif AR tout en mettant en place une technique d'estimation de leurs paramètres en supposant connue leur fonction d'autocorrélation. Le modèle AR permet d'analyser spectralement un signal dans la mesure où les paramètres AR vont nous permettre de connaître les pôles du filtre qu'il représente et donc les fréquences de résonances : il peut ainsi être

vu comme un filtre RII.

Pour se faire, nous allons nous baser sur un processus AR supposé stationnaire au sens large et à temps discret défini de la manière suivante :

$$x(k) = - \sum_{i=1}^p a_i \cdot x(k-i) + u(k)$$

avec  $u(k)$  un bruit blanc gaussien centré de variance  $\sigma_u^2$ ,  $p$  l'ordre du processus et  $\{a_i\}_{i=1, \dots, p}$  les paramètres AR.

On veut un moyen d'estimer les coefficients AR à partir de la fonction d'autocorrélation, pour se faire nous allons établir une relation entre  $p+1$  valeur consécutives de la fonction d'autocorrélation de  $x(k)$ .

$$r_{xx}(\tau) = E[x(k)x(k-\tau)] \quad (2)$$

$$= E[(- \sum_{i=1}^p a_i \cdot x(k-i) + u(k-i))x(k-\tau)] \quad (3)$$

$$= - \sum_{i=1}^p a_i E[x(k-i)x(k-\tau)] + E[x(k-\tau)u(k-i)] \quad (4)$$

$$r_{xx}(\tau) = - \sum_{i=1}^p a_i E[x(k-i)x(k-\tau)] + E[x(k-\tau)]E[u(k-i)] \quad (5)$$

A présent, nous allons calculer la moyenne du processus AR en supposant que  $1 + \sum_{i=1}^p a_i \neq 0$

$$E[x(k)] = - \sum_{i=1}^p E[a_i \cdot x(k-i)] \quad (6)$$

$$= - \sum_{i=1}^p a_i E[x(k-i)] \quad (7)$$

$$= - \sum_{i=1}^p a_i E[x(k)] \quad (8)$$

$$E[x(k)](1 + \sum_{i=1}^p a_i) = 0 \quad (9)$$

On sait, par hypothèse, que  $1 + \sum_{i=1}^p a_i \neq 0$ , donc  $\boxed{E[x(k)] = 0}$

Ainsi,

$$r_{xx}(\tau) = - \sum_{i=1}^p a_i E[x(k-i)x(k-\tau)]$$

$$\boxed{r_{xx}(\tau) = - \sum_{i=1}^p a_i r_{xx}(\tau-i)}$$

On peut maintenant en déduire une relation entre les paramètres AR, défini par  $[a_1, \dots, a_p]^T$  et la matrice d'autocorrélation du processus.

On a alors :

$$\begin{pmatrix} r_{xx}(1) \\ \vdots \\ r_{xx}(p) \end{pmatrix} = - \begin{pmatrix} r_{xx}(0) & \cdots & r_{xx}(1-p) \\ \vdots & \ddots & \vdots \\ r_{xx}(p-1) & \cdots & r_{xx}(0) \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_p \end{pmatrix}$$

Maintenant que la relation est établie, nous allons étudier certains processus d'ordre 1 et 2 et analyser leur spectre de puissance en fonction des valeurs des paramètres AR, ainsi que leur domaine de stabilité.

- **Processus AR d'ordre 1**

- **Domaine de stabilité**

On a un processus de la forme suivante :

$$x(k) = -a_1 x(k-1) + u(k)$$

Ainsi, la fonction de transfert sera :

$$H(z) = \frac{1}{1 + a_1 z^{-1}}$$

On a

$$z^{-1}(a_1 + z) = 0 \iff z = -a_1$$

Il faut donc que  $|a_1| < 1$

On obtient alors le domaine de stabilité suivant :

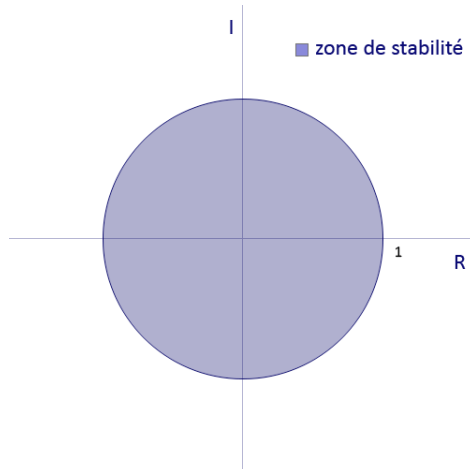


FIGURE 3 – Domaine de stabilité du processus AR d'ordre 1

- **Domaine de stabilité du processus d'ordre 1**

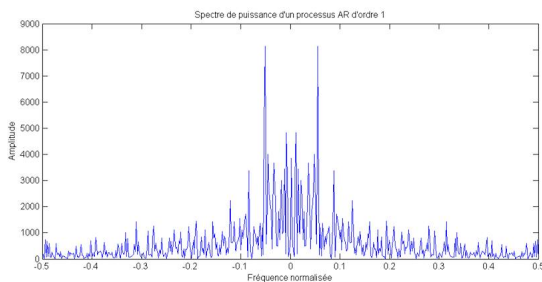


FIGURE 4 – Spectre de puissance pour  $a_1 = 0.5$

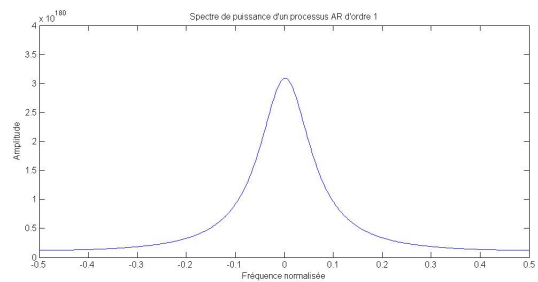


FIGURE 5 – Spectre de puissance de  $a_1 = 1.5$

On a bien une divergence pour  $a_1 = 1.5$ , lorsqu'on compare les valeurs du spectre de puissance par rapport à celles de  $a_1 = 0.5$ .

- **Processus AR d'ordre 2**

- **Domaine de stabilité**

On a un processus de la forme suivante :

$$x(k) = -a_2x(k-2) - a_1x(k-1) + u(k)$$

Ainsi, la fonction de transfert sera :

$$H(z) = \frac{1}{1 + a_1z^{-1} + a_2z^{-2}}$$

On a :

$$z^{-2}(a_2 + za_1 + z^2) = 0 \implies \Delta = \sqrt{a_1^2 - 4a_2}$$

\* **Cas réel :**

$$z_+ = \frac{-a_1 + \sqrt{\Delta}}{2}, z_- = \frac{-a_1 - \sqrt{\Delta}}{2}$$

$$|z_-|^2 < 1 \iff |2 \cdot z_-|^2 = |a_1^2 - a_1^2 - 4a_2^2| = 4a_2^2 < 4 \iff \boxed{|a_2| < 1}$$

\* **Cas complexe :**

$$z_+ = \frac{-a_1 + i\sqrt{\Delta}}{2}, z_- = \frac{-a_1 - i\sqrt{\Delta}}{2}$$

$$\begin{cases} |z_+| < 1 \\ |z_-| < 1 \end{cases} \iff \begin{cases} -2 < a_1 + \sqrt{a_1^2 - 4a_2} < 2 \\ -2 < a_1 - \sqrt{a_1^2 - 4a_2} < 2 \end{cases} \iff \begin{cases} -2 < a_1 < 2 \\ -2 + a_1 < \sqrt{a_1^2 - 4a_2} < 2 + a_1 \end{cases}$$

$$\iff \begin{cases} -2 < a_1 < 2 \\ -2 + a_1 < (-2 + a_1)^2 < a_1^2 - 4a_2 < (2 + a_1)^2 \end{cases} \iff \begin{cases} -2 < a_1 < 2 \\ -1 - a_1 < a_2 < -1 + a_1 \end{cases}$$

Ce qui nous donne  $\boxed{|a_1| < 2}$  et  $\boxed{|a_2 + 1| < a_1}$

On obtient alors le domaine de stabilité suivant :

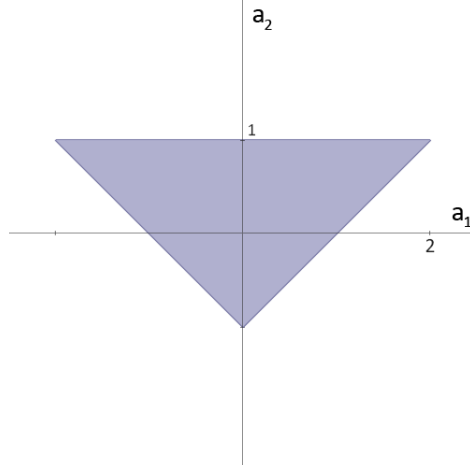


FIGURE 6 – Domaine de stabilité du processus AR d'ordre 2

- **Allure du spectre de puissance**

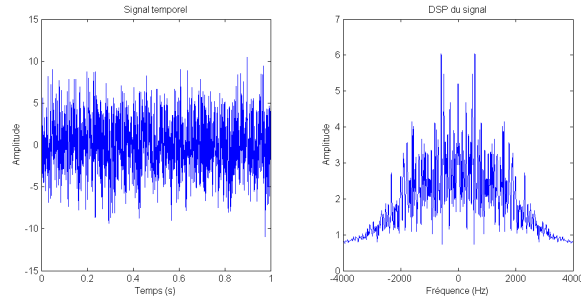


FIGURE 7 – Spectre de puissance pour  $a_1 = 0.5$  et  $a_2 = 0.5$

## 2.2 Implémentation sur Matlab

### 2.2.1 Signaux générés

Le but, à présent, est de retrouver l'allure de la DSP à partir de l'estimation des coefficients AR. On commence par générer un processus AR d'ordre 4. On a 4 pôles conjugués 2 à 2.

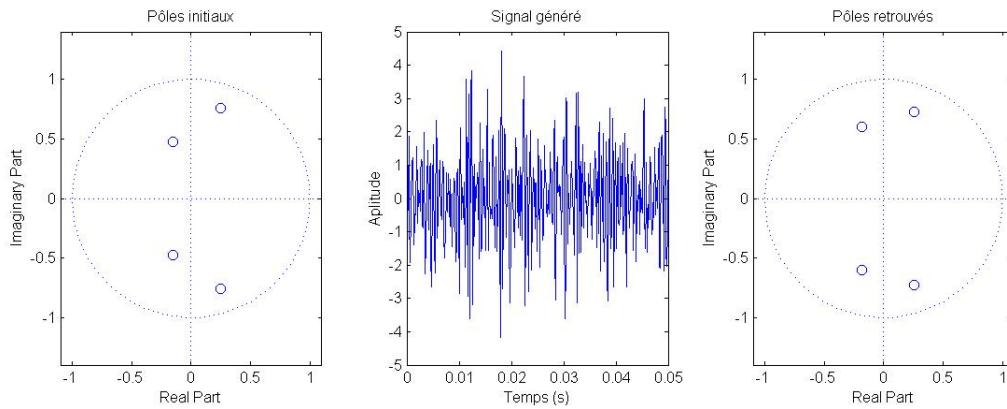


FIGURE 8 – Pôles générés et estimés

On remarque que les pôles estimés sont proches de ceux qui ont été générés.



### 2.2.2 Signaux réels

On étudie toujours le signal voisé de la première partie.

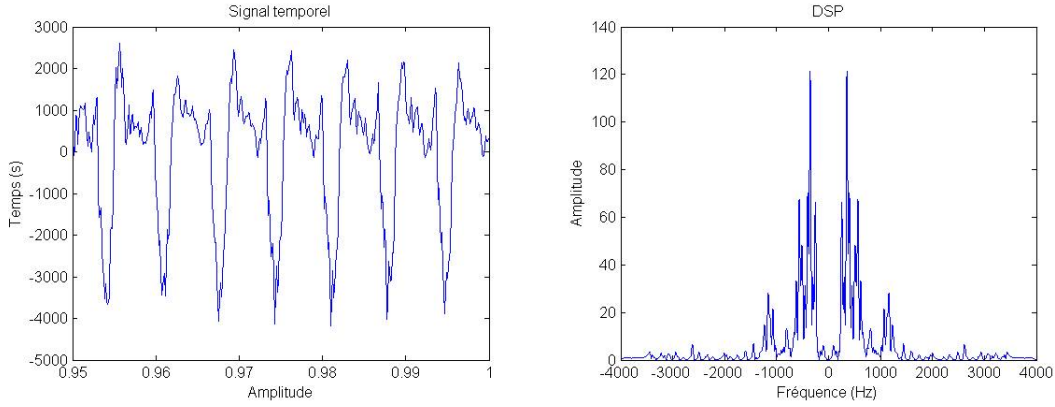


FIGURE 9 – Signal temporel et DSP

L'allure de la DSP estimée n'est pas aussi proche de celle calculée grâce à la FFT.

## 3 Approche par méthode de Capon

La méthode de Capon repose sur l'analyse de la puissance associée à chaque fréquence d'un signal donné à l'aide d'un filtre RIF ayant 1 comme gain à la fréquence qui nous intéresse, ce qui permet de détecter si la composante recherchée est présente ou non dans le signal.

Soit  $h(k)_{k=0,\dots,N-1}$  la réponse impulsionnelle du filtre causal,  $x(n)$  le signal d'entrée,  $y(n)$  le signal de sortie. On sait que  $y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$ , donc :

$$y(n) = \underline{h}x(n) = [h(0) \dots h(n-1)] \begin{bmatrix} x(n) \\ \vdots \\ x(n-N+1) \end{bmatrix}$$

On en déduit que :

$$E[y^2(n)] = E[y(n)y(n)^T] = E[\underline{h}x(n)^T \underline{h}^T x(n)] \quad (10)$$

Ainsi, la puissance instantanée du signal de sortie est :

$$E[y^2(n)] = \underline{h}R_x \underline{h}^T$$

Afin de trouver la contrainte sur le filtre dans le cadre de l'approche de Capon, nous commençons, tout d'abord, par calculer la transformée de Fourier de la réponse impulsionnelle du filtre RIF.

$$H(f) = \sum_{k=0}^{N-1} h(k)e^{-j2k\pi \frac{f}{f_{ech}}} \quad (11)$$

$$= \underline{h}a(f) \quad (12)$$

Ainsi, la contrainte en une fréquence d'intérêt donnée  $f_i$  est  $\underline{ha(f_i)} = 1$

### 3.1 Implémentation sur Matlab

#### 3.1.1 Signaux générés

L'étude se fait avec le signal généré dans la première partie.

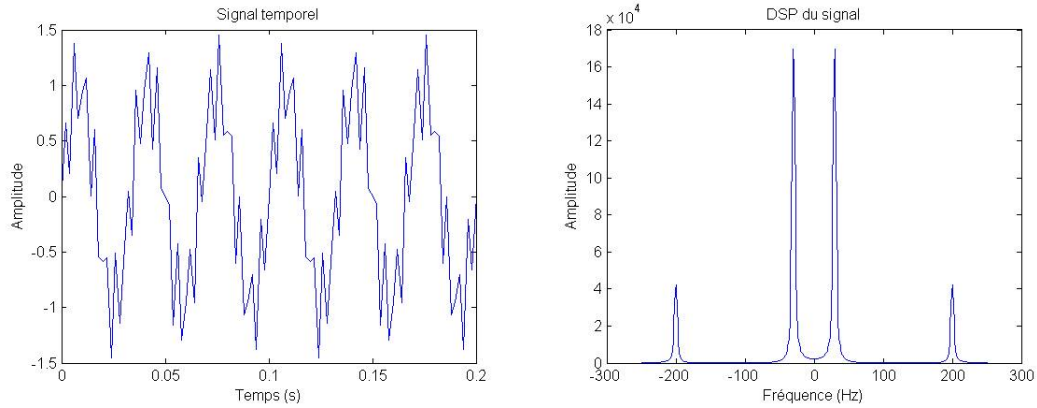


FIGURE 10 – Signal temporel et DSP

On remarque que la DSP calculée est en accord avec le signal généré.

#### 3.1.2 Signaux réels

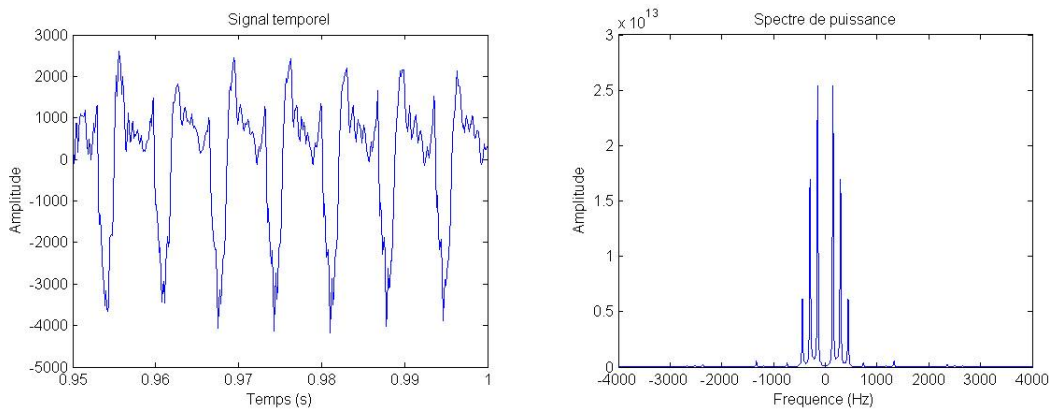


FIGURE 11 – Signal temporel et DSP

La DSP est cohérente avec celle trouvée grâce à la FFT, on retrouve bien les 3 pics de fréquences attendus.

## 4 Analyse temps-fréquence

Le problème avec un signal réel est la diversité des fréquences sur tout le long du signal. On ne peut en faire une analyse globale, car ils ne sont pas, en général, stationnaire au sens large. C'est pourquoi, afin d'analyser des signaux réels, on segmente temporellement ces derniers, pour se ramener à des trames stationnaires que l'on peut ensuite analyser fréquemment.

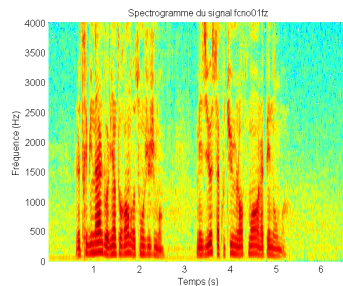


FIGURE 12 – Spectrogramme du signal de parole présenté dans la première partie

Cette figure a été générée avec la fonction *spectrogram* de MATLAB. Ensuite, nous avons essayé d'implémenter notre propre fonction permettant de faire du temps réel, c'est-à-dire un traitement de données arrivant en continue. Par conséquent, l'échelle des amplitudes est locale et l'amplitude maximale est calculée pour chaque fenêtre (la couleur de chaque maximum est la même pour chaque segment), elle n'est pas globale comme pour la fonction *spectrogram*.

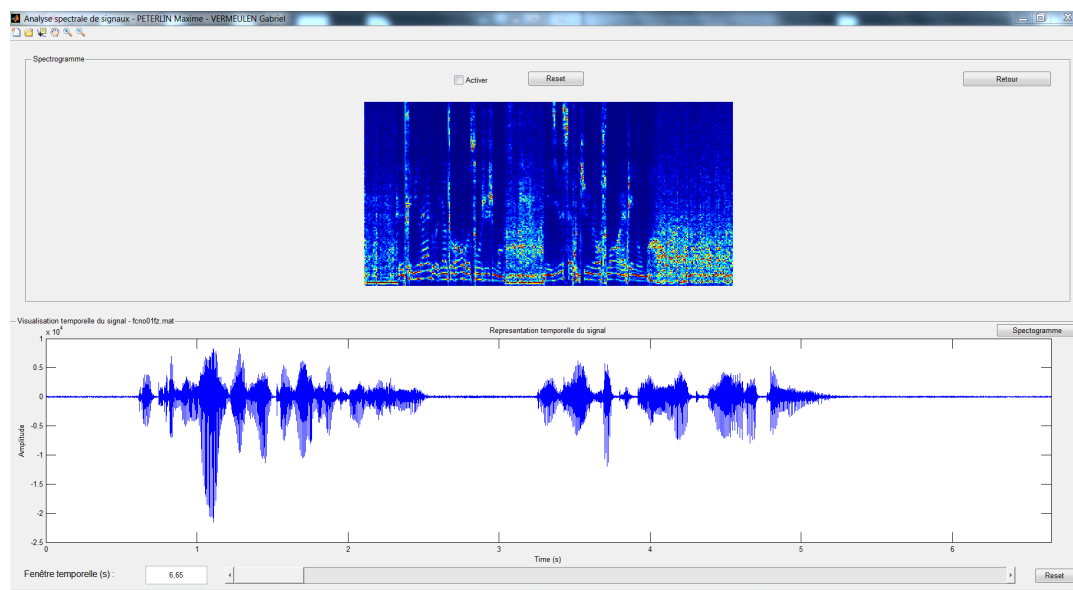


FIGURE 13 – Spectrogramme du signal de parole présenté dans la première partie

Sur ce spectrogramme, on remarque dans les intervalles sans signaux de voix, la présence d'un bruit. A l'inverse lorsque qu'il y a un signal de voix, le bruit est masqué et les caractéristiques fréquentielles, telles que les harmoniques, du signal sont affichés.

## 5 Bilan de l'organisation

- **Première séance** : Théorie et pratique de la partie FFT.
- **Deuxième séance** : Théorie et pratique de la partie AR.
- **Entre la deuxième et troisième séance ( 3h )** : Mise en place de la GUI.
- **Troisième séance** : Théorie et pratique de la partie Capon.
- **Quatrième séance** : Ajout de fonctionnalités à la GUI et spectrogramme.
- **Après la dernière séance ( 10h )** : Réglages des derniers bugs et écriture du rapport.

## Conclusion

Nous avons vu, grâce à ce projet, différentes méthodes permettant d'analyser spectralement un signal. La FFT qui est un algorithme facile à mettre en oeuvre et efficace d'un point de vue complexité, reste sujet à des problèmes de résolution survenant lorsque le nombre de points le composant est insuffisant ou encore à des valeurs indésirables fréquemment lorsqu'on fenêtré le signal.

Pour le modèle AR, le problème a été le choix de l'ordre, il faut trouver la bonne valeur permettant d'avoir une estimation correcte, mais lorsqu'on l'a, les résultats obtenus sont plus que satisfaisant.

Enfin, pour la méthode de Capon, le résultat que l'on obtient est lui aussi tout à fait satisfaisant, cependant, vu que l'on doit balayer toutes les fréquences, le temps de calcul est relativement plus long par rapport aux autres méthodes.