

**Travaux Pratiques Traitement Numérique du Signal**  
*FILIERE ELECTRONIQUE, 2<sup>E</sup> ANNEE*

2024-2025

Rémi Giraud  
remi.giraud@enseirb-matmeca.fr

Cornelia Vacar  
cornelia.vacar@netc.fr

## TABLE DES MATIÈRES

<b>1</b>	<b>RAPPELS : TRANSFORMÉE DE FOURIER DES SIGNAUX DISCRETS</b>	<b>4</b>
1.1	Notion d'échantillonnage . . . . .	4
1.1.1	Dirac . . . . .	4
1.1.2	Peigne de Dirac . . . . .	4
1.1.3	Échantillonnage discret d'un signal continu . . . . .	5
1.2	Transformée de Fourier Discrète et le théorème de Shannon . . . . .	5
1.2.1	Transformée de Fourier Continue d'un signal Discret (TFCD) . . . . .	5
1.2.2	Transformée de Fourier Discrète (TFD, ou DFT en anglais) . . . . .	7
1.2.3	Théorème d'échantillonnage de Shannon . . . . .	8
<b>2</b>	<b>UTILISATION DE LA TFD</b>	<b>8</b>
2.1	TFD et Fenêtrage . . . . .	8
2.1.1	TFD d'une sinusoïde . . . . .	9
2.1.2	Effet du fenêtrage . . . . .	9
2.2	Analyse de signaux sonores . . . . .	10
2.2.1	Synthèse de note . . . . .	10
2.2.2	Analyse fréquentielle . . . . .	10
2.2.3	Synthèse d'une mélodie . . . . .	11
2.2.4	Analyse temps/fréquence . . . . .	11
<b>3</b>	<b>FILTRAGE LINÉAIRE</b>	<b>12</b>
3.1	Rappels . . . . .	12
3.2	Effet fréquentiel du filtrage . . . . .	14
3.3	Signal de parole bruité . . . . .	15
3.3.1	Débruitage par filtrage sélectif . . . . .	15
<b>4</b>	<b>BONUS : POUR ALLER PLUS LOIN</b>	<b>16</b>
4.1	Sélectivité du filtrage . . . . .	16
4.2	Débruitage direct sur la transformée de Fourier . . . . .	16
4.3	Application : Débruitage d'un signal biomédical . . . . .	16
4.3.1	Objectifs . . . . .	16
4.3.2	Mise en pratique . . . . .	17
4.3.3	Déphasage lié au filtrage . . . . .	19
<b>5</b>	<b>ANNEXES</b>	<b>20</b>
5.1	Séparation de sources . . . . .	20
5.2	Notion de bruit blanc gaussien . . . . .	21
5.3	Débruitage d'un signal de parole . . . . .	22
5.3.1	Caractérisation des voyelles et notions de formants . . . . .	22
5.3.2	La fréquence fondamentale . . . . .	23
5.3.3	Bilan . . . . .	24
5.3.4	Sur le rehaussement monovoie par atténuation spectrale à court terme . . . . .	24
5.4	Formulaire . . . . .	26
5.5	Mémo Matlab . . . . .	27

## INTRODUCTION

---

Au travers de ces séances de Travaux Pratiques, nous traitons des bases du traitement numérique du signal et illustrons le cours de traitement numérique du signal.

Les notions suivantes seront abordées :

- ✓ Transformée de Fourier des signaux discrets
- ✓ Caractéristiques des systèmes à temps discret : linéarité et non linéarité, invariance du système, causalité et stabilité, fonction de transfert

En outre ces TPs reposent sur le logiciel Matlab.

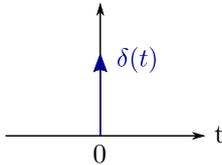
Un rappel de cours ou une introduction aux notions étudiées est proposé pour chaque thème.

# 1 RAPPELS : TRANSFORMÉE DE FOURIER DES SIGNAUX DISCRETS

## 1.1 Notion d'échantillonnage

### 1.1.1 Dirac

L'impulsion de Dirac peut-être définie comme :



$$\delta(t) = \begin{cases} \infty & \text{si } t = 0 \\ 0 & \text{sinon} \end{cases} \quad \text{avec } \int_{-\infty}^{\infty} \delta(t) dt = 1$$

En découle :

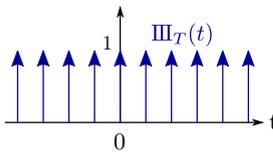
$$\delta(t - t_0) = \begin{cases} \infty & \text{si } t = t_0 \\ 0 & \text{sinon} \end{cases} \quad \text{et} \quad \int_a^b x(t) \delta(t - t_0) dt = \begin{cases} x(t_0) & \text{si } a < t < b \\ 0 & \text{sinon} \end{cases}$$

Côté transformée de Fourier :

$$\begin{aligned} TF(\delta(t - t_0))(f) &= \int_{-\infty}^{\infty} \delta(t - t_0) \exp(-j2\pi ft) dt \\ &= \exp(-j2\pi ft_0) \end{aligned}$$

### 1.1.2 Peigne de Dirac

Un peigne de Dirac de période  $T$  est composé d'une infinité d'impulsions de Dirac tel que :

$$\text{III}_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$


Ce peigne de Dirac peut également s'écrire comme :

$$\text{III}_T(t) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \exp(j2\pi nt/T)$$

Démonstration : utilisant l'expression en série de Fourier d'une fonction  $T$ -périodique :

$$\text{III}_T(t) = \sum_{n=-\infty}^{\infty} c_n \exp(j2\pi nt/T)$$

avec

$$\begin{aligned} c_n &= \frac{1}{T} \int_{-t_0}^{t_0} \text{III}_T(t) \exp(-j2\pi nt/T) dt \\ &= \frac{1}{T} \int_{-T/2}^{T/2} \text{III}_T(t) \exp(-j2\pi nt/T) dt \\ &= \frac{1}{T} \int_{-T/2}^{T/2} \delta(t) \exp(-j2\pi nt/T) dt = \frac{1}{T} \exp(-j2\pi n0/T) = \frac{1}{T} \end{aligned}$$

La transformée de Fourier du peigne de Dirac s'avère également être un peigne de Dirac période de  $1/T$  :

$$\begin{aligned} TF(\mathbb{I}_T(t))(f) &= \frac{1}{T} \sum_{n=-\infty}^{\infty} TF(\exp(j2\pi nt/T))(f) \\ &= \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta(f - n/T) \\ &= \frac{1}{T} \mathbb{I}_{1/T}(f) \end{aligned}$$

Démonstration :

$$\begin{aligned} TF^{-1}(\delta(f - f_0)) &= \int_{-\infty}^{\infty} \delta(f - f_0) \exp(j2\pi ft) df \\ &= \exp(j2\pi f_0 t) \\ &= TF^{-1}(TF(\exp(j2\pi f_0 t))(f)) \end{aligned}$$

Donc  $TF(\exp(j2\pi f_0 t))(f) = \delta(f - f_0)$

### 1.1.3 Échantillonnage discret d'un signal continu

L'échantillonnage d'un signal continu  $x(t)$  est souvent effectué de manière uniforme, à une période  $Te$ , pour obtenir un signal discret. En réalité cela revient à multiplier le signal  $x(t)$  par un peigne de Dirac de période  $Te$  :

$$x_e(t) = x(t) \cdot \mathbb{I}_{Te}(t) = \sum_{n=-\infty}^{+\infty} x(nTe) \delta(t - nTe)$$

Dans la suite, étant donné que l'on ne manipulera que le signal discret, on notera  $x[n]$  l'échantillon obtenu à l'instant  $t = nTe$  dans la mesure où cela ne prête pas à ambiguïté : les valeurs du signal continu d'origine  $x(t) = x(nTe)$  seront toujours désignées avec  $t$  ou  $Te$  alors que les valeurs du signal échantillonné  $x[n]$  se contenteront d'un indice entier  $n, m$  etc.

$$\{x[n]\}_{n=-\infty, \dots, +\infty} = \{x(nTe)\}_{n=-\infty, \dots, +\infty}$$

## 1.2 Transformée de Fourier Discrète et le théorème de Shannon

### 1.2.1 Transformée de Fourier Continue d'un signal Discret (TFCD)

L'opération d'échantillonnage par le peigne de Dirac transforme l'expression de la transformée de Fourier d'un signal continu en une expression qui dépend d'une somme discrète au lieu d'une intégrale continue et où la variable fréquentielle  $f$  reste continue :

$$X_e(f) = \sum_{n=-\infty}^{+\infty} x[n] \exp(-j2\pi f nTe) \quad (\text{TFCD})$$

$$x[n] = \frac{1}{Te} \int_{-f_e/2}^{+f_e/2} X_e(f) \exp(j2\pi f nTe) df \quad (\text{TFCDI})$$

Démonstration :

$$\begin{aligned}
 X_e(f) &= TFC(x_e(t))(f) = TFC(x(t) \cdot \mathbb{III}_{T_e}(t))(f) \\
 &= \int_{-\infty}^{\infty} x(t) \cdot \mathbb{III}_{T_e}(t) \exp(-j2\pi ft) dt \\
 &= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{+\infty} x(nT_e) \delta(t - nT_e) \exp(-j2\pi ft) dt \\
 &= \sum_{n=-\infty}^{+\infty} x(nT_e) \int_{-\infty}^{\infty} \delta(t - nT_e) \exp(-j2\pi ft) dt \\
 &= \sum_{n=-\infty}^{+\infty} x(nT_e) \exp(-j2\pi f n T_e) \\
 &= \sum_{n=-\infty}^{+\infty} x[n] \exp(-j2\pi f n T_e) \\
 &= \sum_{n=-\infty}^{+\infty} x[n] \exp(-j2\pi f n / f_e)
 \end{aligned}$$

Avec  $f_e$  la fréquence d'échantillonnage (ou  $f_s$  pour *sampling*) alors liée à cette période d'échantillonnage selon :  $f_e = 1/T_e$ .

**Périodicité.** Cette transformée de Fourier s'avère d'ailleurs périodique de période  $f_e$ . On peut facilement le démontrer en utilisant le fait qu'une multiplication dans le domaine temporel correspond à une convolution dans le domaine fréquentiel. La transformée de Fourier d'un signal échantillonné à une période  $T_e$ , du fait de sa multiplication implicite par un peigne de Dirac est bien alors périodique de période  $1/T_e$  :

$$\begin{aligned}
 TFC(x_e)(f) &= TFC(x(t) \cdot \mathbb{III}_{T_e}(t))(f) \\
 &= TFC(x(t))(f) * TFC(\mathbb{III}_{T_e}(t))(f) \\
 &= TFC(x(t))(f) * \frac{1}{T_e} \mathbb{III}_{1/T_e}(f)
 \end{aligned}$$

En outre, la TFCD est proportionnelle au spectre du signal continu (condition de Shannon), et peut s'exprimer en fréquence réduite par :

$$\begin{aligned}
 X_e(f) &= \sum_{n=-\infty}^{+\infty} x[n] \exp(-j2\pi f_r n) \\
 \text{avec } f_r &= \frac{f}{f_e} \quad \text{et } f_r \in [-0.5, 0.5] \quad \text{ou } [0, 1]
 \end{aligned}$$

Le calcul sur ordinateur d'une telle quantité est bien évidemment exclu car la série comporte un nombre infini de termes et la fréquence varie continûment.

Concernant le premier point, on se restreint en pratique à un nombre d'échantillons fini  $N$  qui peut être stocké dans la machine. Tous les autres échantillons sont supposés nuls : on considère un signal à support temporel borné.

$$X_e(f) = \sum_{n=0}^{N-1} x[n] \exp(-j2\pi f_r n)$$

Concernant le second point, on va opérer une discrétisation dans le domaine fréquentiel, qui conduit à la Transformée de Fourier Discrète (TFD).

### 1.2.2 Transformée de Fourier Discrète (TFD, ou DFT en anglais)

Le principe de la TFD est le même que la TFCD mais on introduit un pas d'échantillonnage uniforme en fréquence pour discrétiser l'axe en fréquence. Le nombre de points d'échantillonnage du spectre est noté  $M$ . Le spectre est donc calculé sur le domaine fréquentiel  $[0, f_e]$  avec un pas de quantification de  $f_e/M$ . On note  $k$  l'indice entier permettant de déterminer la fréquence :

$$f = \frac{kf_e}{M} \quad \text{où} \quad \begin{cases} f_e & \text{est la fréquence d'échantillonnage,} \\ M & \text{est le nombre de points pour le calcul de la TFD} \end{cases}$$

La Transformée de Fourier Discrète s'exprime ainsi de façon indépendante de la fréquence d'échantillonnage :

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp\left(-j2\pi \frac{kn}{M}\right) \quad (\text{TFD})$$

Le spectre étant périodique, on prendra  $k = [0, \dots, M-1]$ . De la même façon que pour le signal temporel, on pourra noter  $X[k]$  la transformée discrète, alors que  $X(f)$  correspond à la transformée continue. On a  $X^{TFD}[k] = X^{TFC}(f)$  pour  $f = kf_e/M$ . Les noms de variables utilisés permettent de rendre non ambiguë la notation.

Si l'on attribue pour l'échantillonnage en fréquence autant de points que pour le signal temporel ( $N = M$ ), la TFD est réversible. On a dans ce cas :

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp\left(-j2\pi \frac{kn}{N}\right) \quad (\text{TFD réversible})$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \exp\left(j2\pi \frac{kn}{N}\right) \quad (\text{TFDI})$$

L'avantage de la TFD est que, de par sa structure, elle a permis de développer des algorithmes rapides appelés Transformée de Fourier Rapide (TFR) ou *Fast Fourier Transform* (FFT). La différence entre les deux est uniquement algorithmique ; la TFD et la TFR calculent la même chose. Sous Matlab, la fonction utilisée pour calculer la TFD s'appelle ainsi *fft*. Les algorithmes rapides utilisent les propriétés de afin d'éviter la redondance des calculs. La complexité calculatoire de la méthode directe est de  $\mathcal{O}(N^2)$  alors que l'approche rapide présente une complexité en  $\mathcal{O}(N \log_2(N))$ .

### 1.2.3 Théorème d'échantillonnage de Shannon

De nombreux paramètres ont une influence sur l'échantillonnage, notamment le pas de quantification et le temps de réponse du système numérique lors de l'acquisition et de la restitution. Cela dit, en augmentant la précision du convertisseur analogique-numérique (CAN) et en augmentant la rapidité des calculateurs, on peut pallier ces difficultés.

Le choix de la fréquence d'échantillonnage joue aussi un rôle. Plus la fréquence d'échantillonnage est grande, plus on dispose d'échantillons pour décrire le signal. En contrepartie, le traitement de ces données est plus lourd. Un compromis est à trouver.

Soit un signal  $s$  à spectre  $S(f)$  borné, *i.e.*,  $\forall f > f_{max}, |S(f)| = 0$ ; cette limitation du spectre est soit propre au signal traité, soit due à un pré-filtrage passe bas. Le théorème d'échantillonnage de Shannon fixe le choix de la fréquence d'échantillonnage  $f_e$  afin d'éviter les distorsions du spectre échantillonné, les phénomènes dits de repliement.

$$f_e \geq 2f_{max}$$

est appelée fréquence de Shannon, ou encore fréquence de Nyquist, ou de repliement.

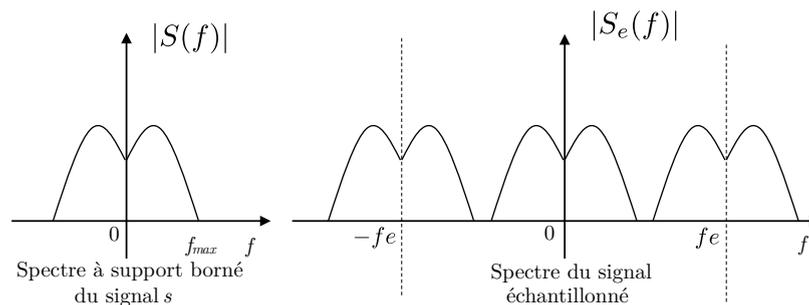


Figure 1 : Spectre du signal échantillonné

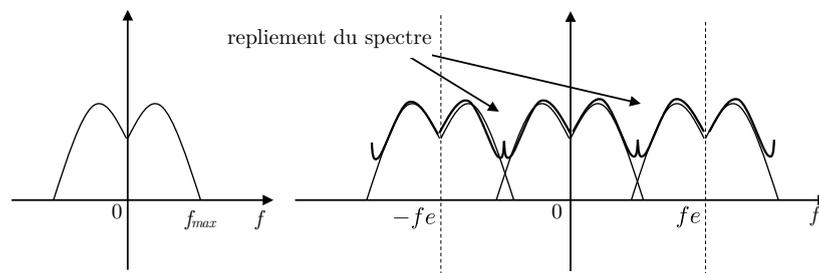


Figure 2 : Phénomène de repliement

## 2 UTILISATION DE LA TFD

### 2.1 TFD et Fenêtrage

1. Complétez le tableau suivant pour une fonction  $x(t)$  quelconque :

	Temps continu et fréquence continue	Temps discret et fréquence continue	Temps discret et fréquence discrète
Expression de la transformée de Fourier			
Domaine de fréquences observables			

### 2.1.1 TFD d'une sinusoïde

1. Créer un signal  $x[n]$  de  $N = 2048$  échantillons correspondant à une sinusoïde de fréquence porteuse  $f_0 = 300$  Hz et de fréquence d'échantillonnage  $f_e = 3000$  Hz :

$$x[n] = \cos\left(2\pi \frac{f_0}{f_e} n\right) \quad \text{avec} \quad n = \llbracket 0, N - 1 \rrbracket$$

Quelle est la durée de ce signal en secondes ?

2. Afficher ce signal. Le résultat paraît-il surprenant ? Réessayer avec  $N = 256$ ,  $N = 128$ .
3. Quelle serait la TFCD d'un tel signal ? Faire un schéma sur papier du résultat attendu, en faisant figurer  $f_e$  sur l'échelle fréquentielle.
4. Quelle est la relation qui existe entre la fréquence  $f$  (en Hz) et la fréquence discrète  $k$  de la TFD ?  
Compléter le schéma en exprimant l'échelle fréquentielle également en fréquence réduite  $f_r = f/f_e$ .
5. Calculer la TFD ( $fft$ ) sur  $M = 512$  points puis  $M \gg 512$ , et afficher son module ( $abs$ ).
6. Détailler le rôle précis des fonctions  $fft$  et  $fftshift$  sous Matlab.  
Modifier l'affichage sous Matlab pour tracer la TFD centrée autour de la fréquence 0.  
Légender l'axe des abscisses pour faire figurer les fréquences réelles en Hz.
7. Comment expliquer les différences entre le résultat théorique et observé ?

### 2.1.2 Effet du fenêtrage

On se trouve dans un contexte discret, avec des signaux de taille finie. Le signal manipulé  $x[n]$  est donc en réalité affecté par un fenêtrage et peut être défini comme le produit du signal et d'une fenêtre de pondération (apodisation) de la taille du signal :

$$x_w[n] = x[n] \cdot w[n]$$

avec  $w[n]$  une fenêtre de pondération (ou fenêtre temporelle) de valeur nulle en dehors de l'intervalle d'observation  $[0, \dots, N - 1]$ .

1. Indiquer quelle relation théorique existe entre les Transformées de Fourier Continues  $X(f)$ ,  $W(f)$  et  $X_w(f)$  des signaux discrets à support non bornés,  $x[n]$ ,  $w[n]$  et  $x_w[n]$ . En déduire la relation entre les Transformées de Fourier Discrètes calculées sur les  $N$  échantillons  $[0, \dots, N - 1]$  où la fenêtre est non nulle.
2. Définir un signal sinusoïdal de  $N = 256$  échantillons, et de période 32 échantillons (donc de fréquence réduite  $f_x = 1/32 = 8/256$ ). On supposera la fréquence d'échantillonnage unitaire  $f_e = 1$ , afin de travailler directement en fréquence réduite.  
Quelle serait la TFCD théorique d'un tel signal s'il n'était pas limité (fenêtré) à l'intervalle  $[0, \dots, N - 1]$  ?
3. Calculer et afficher la TFD du signal, calculé sur  $M = N = 256$  points, puis  $M$  points avec  $M \gg N$  (par ex.  $M = 2048$ ). Faire attention à l'échelle des fréquences.  
Rappeler quelle est la TFD d'une fenêtre carrée de largeur  $N$ .  
Expliquer l'allure du spectre dans chaque cas, à l'aide d'un graphique.
4. Modifier la fréquence de la sinusoïde à  $f_x = 8.5/256$ . Observer le signal fenêtré et sa TFD pour la fenêtre rectangulaire à  $M = N = 256$  points. Puis  $M = 2048$ . Que peut-on constater ? Expliquer le phénomène.

## 2.2 Analyse de signaux sonores

Cette partie a pour objectif l'application de la transformée de Fourier à des signaux sinusoïdaux mis bout à bout pour former une mélodie.<sup>1</sup>

### 2.2.1 Synthèse de note

Pour le moment nous avons considéré une sinusoïde de fréquence fondamentale  $f_0$  quelconque, sans savoir si cette fréquence correspondait à une note produite effectivement par un instrument de musique. En réalité, si l'on considère par exemple un piano, il y a seulement un ensemble fini de notes (donc de fréquences) qu'il peut produire. Dans notre cas, nous allons donner à chaque note du piano un numéro, et calculer la fréquence fondamentale associée grâce à la formule suivante :

$$f_0^{note} = 440 * 2^{\frac{note-69}{12}} \quad (1)$$

où *note* est le numéro de la note que l'on veut jouer. La Table 1 présente un extrait de la correspondance entre numéros de notes, notes musicales et fréquences fondamentales.

octave	do	do #	ré	ré #	mi	fa	fa #	sol	sol #	la	la #	si
1	24 32.7	25 34.65	26 36.71	27 38.89	28 41.2	29 43.65	30 46.25	31 49	32 51.91	33 55	34 58.27	35 61.74
2	36 65.41	37 69.3	38 73.42	39 77.78	40 82.41	41 87.31	42 92.5	43 98	44 103.83	45 110	46 116.54	47 123.47
3	48 130.81	49 138.59	50 146.83	51 155.56	52 164.81	53 174.61	54 185	55 196	56 207.65	57 220	58 233.08	59 246.94
4	60 261.63	61 277.18	62 293.66	63 311.13	64 329.63	65 349.23	66 369.99	67 392	68 415.3	69 440	70 466.16	71 493.88
5	72 523.25	73 554.37	74 587.33	75 622.25	76 659.26	77 698.46	78 739.99	79 783.99	80 830.61	81 880	82 932.33	83 987.77
6	84 1046.5	85 1108.73	86 1174.66	87 1244.51	88 1318.51	89 1396.91	90 1479.98	91 1567.98	92 1661.22	93 1760	94 1864.66	95 1975.53
7	96 2093	97 2217.46	98 2349.32	99 2489.02	100 2637.02	101 2793.83	102 2959.96	103 3135.96	104 3322.44	105 3520	106 3729.31	107 3951.07

Table 1 : Correspondance entre notes et fréquences fondamentales

### 2.2.2 Analyse fréquentielle

1. Charger le signal *note.wav* sous Matlab (commande *audioread*). Vérifier que la fréquence d'échantillonnage est bien 8kHz. Quelle est la durée du son en secondes ?

Calculer la TFD (*fft*) sur  $M = 512$  points puis  $M \gg 512$ , et afficher son module.

2. Quelles est la fréquence de la sinusoïde ? De quelle note s'agit-il ?
3. Construire le signal correspondant à la note *mi* de la 5e octave, sur le même nombre de points que le signal précédent *note.wav*.  
Quelle est la dimension du signal créé ? Du signal précédent ? Sont-elles strictement les mêmes ? Penser à vérifier le *workspace*.
4. Additionner les deux signaux. Vérifier la dimension du signal créé dans le *workspace*.  
Visualiser la représentation fréquentielle du signal obtenu.
5. Modifier manuellement la fréquence caractérisant la touche *mi* pour se rapprocher de la fréquence de la *note* de *note.wav*.

À un certain point, on ne distingue plus les réponses des deux notes. Relier cette observation aux paramètres  $M$ ,  $N$  et  $f_e$  à l'aide d'un schéma.

<sup>1</sup>inspiré des supports de Laurent Ourde

### 2.2.3 Synthèse d'une mélodie

1. Depuis le code utilisé pour créer la note *mi*, créer une fonction *create\_note.m* prenant en paramètres d'entrée une durée  $d$ , un numéro de note *note* et une fréquence d'échantillonnage  $f_e$  et renvoyant un signal  $x$  (correspondant à une sinusoïde de durée  $d$  et de fréquence fondamentale  $f_0^{note}$  (1).

Dans le cas où  $note = -1$ , on renverra un signal nul de durée  $d$  (cela revient à créer une sinusoïde de fréquence  $f_0 = 0$ ).

2. Tester la fonction en synthétisant et en écoutant plusieurs notes de différentes hauteurs et différentes durées. Tester le cas  $note = -1$ .

Pour écouter un signal audio, deux solutions : 1) utiliser *soundsc* ou 2) Enregistrer votre signal avec *audiowrite* et l'écouter depuis l'explorateur de fichiers.

3. Créer une nouvelle fonction *create\_melody.m* prenant en paramètres d'entrée un vecteur de durée  $d\_vect$ , un vecteur de numéros de note  $note\_vect$  et une fréquence d'échantillonnage  $f_e$  et renvoyant un signal  $x$ . Le signal  $x$  correspond à la concaténation de plusieurs sinusoïdes dont les durées et hauteurs sont respectivement définies dans les vecteurs  $d\_vect$  et  $note\_vect$ .

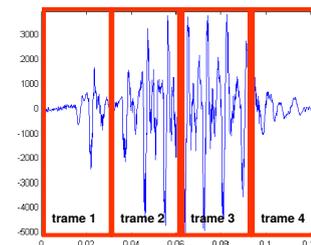
4. Tester la fonction en chargeant les vecteurs  $d\_vect$  et  $note\_vect$  stockés dans la variable *melody.mat*. Synthétiser et écouter la mélodie associée.

5. Comment modifier cette mélodie pour qu'elle soit jouée une octave en dessous ?

Et pour qu'elle aille 50% plus vite ?

### 2.2.4 Analyse temps/fréquence

En pratique, on ne manipule pas toujours des signaux sinusoïdaux parfaits, n'évoluant pas au cours du temps. Dans de nombreux cas en traitement du signal (notamment dans le cas du traitement de la parole, le débruitage, etc.), on doit donc se placer dans des conditions de quasi stationnarité et effectuer une analyse du signal sur des "trames" d'une durée donnée. La non stationnarité d'un signal impose de faire une analyse fréquentielle par trame.



Traitement par trame

Une telle analyse produit une TFD par trame, ce qui ne peut être facilement visualisé sous la forme d'un simple graphe. Une solution consiste à associer un code couleur aux amplitudes du module de la TFD, et à afficher l'ensemble des TFD sous la forme d'une image (*spectrogram*), dans laquelle chaque colonne correspond au spectre d'une fenêtre du signal.

1. Regarder l'aide de la fonction *spectrogram* pour comprendre son fonctionnement.
2. Charger le signal de parole *waziwaza.wav*, et afficher son spectrogramme :  
Indiquer sur le spectrogramme les zones fréquentielles qui caractérisent la prononciation des /z/. Indiquer approximativement les zones formantiques des voyelles prononcées. (voir annexe "Débruiteur un signal de parole").
3. Modifier le spectrogramme pour que la TFD soit calculée sur  $M = 128$  points.
4. Tester la fonction *spectrogram* sur des mélodies créées avec la fonction *create\_melody*.  
Qu'observe-t-on ?  
Charger la mélodie *melody\_easy.wav*, et afficher son spectrogramme. Détecter les notes et l'ordre joués.

**Pour aller plus loin**, voir l'Annexe 5.1 Séparation des sources.

### 3 FILTRAGE LINÉAIRE

Ce TP a pour objectif d'étudier la notion de filtre linéaire d'un point de vue fréquentiel. En particulier, les notions suivantes sont abordées :

- Lien entre diagramme des pôles et zéros et réponse fréquentielle d'un filtre
- Caractérisation d'un filtre discret passe-haut, passe-bas
- Débruitage par élimination de fréquences

#### 3.1 Rappels

On peut définir l'entrée discrète  $x[n]$  d'un filtre à partir d'un signal échantillonné. Après une transformation  $T(\cdot)$ , représentant l'opération de filtrage, la sortie du filtrage est notée  $y[n] = T(x[n])$ .

- Rappel de quelques propriétés :

Si l'on considère deux entrées quelconques  $x_1[n]$  et  $x_2[n]$  et deux coefficients  $a_1$  et  $a_2$ , alors le système est dit **linéaire** si :

$$T(a_1x_1[n] + a_2x_2[n]) = a_1T(x_1[n]) + a_2T(x_2[n])$$

Le système est dit **invariant** si  $T$  ne change pas au cours du temps.

$$\text{Si } y[n] = T(x_n[n]) \text{ alors } y[n - k] = T(x[n - k])$$

Le système est dit **causal** si la sortie à l'instant ne dépend que du passé ou du présent :

$$y[n] = T(y_n[n - 1], \dots, y[1], x[n], \dots, x[1])$$

Dans le cas où la transformation dépend du futur, le système est dit **anticausal**.

- Systèmes linéaires invariants – présentation de deux types de filtres :

L'expression d'un système linéaire invariant peut être définie comme suit :

$$y[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n - k]$$

Ce qui correspond au produit de convolution  $y[n] = x[n] * h[n]$  avec  $h[n]$  la réponse impulsionnelle.

La réponse fréquentielle d'un filtre de fonction de transfert en  $Z$   $H(z)$  est la TFC de la réponse impulsionnelle  $h[n]$  considérée à la période d'échantillonnage  $T_e$  :

$$H(f) = \sum_{n=-\infty}^{+\infty} h[n] \exp(-j2\pi f n T_e)$$

Cette réponse  $H(f)$  est périodique de période fréquentielle  $f_e$  (à cause de l'échantillonnage à fréquence  $f_e$ ), et correspond exactement à la valeur de la fonction de transfert  $H(z)$  restreinte au cercle unité :

$$H(f) = H(z) \text{ pour } z = \exp(j2\pi f / f_e)$$

Aussi, l'expression générale du filtre peut également s'écrire à partir d'une équation aux différences :

$$\sum_{i=0}^p a_i y[n - i] = \sum_{j=0}^q b_j x[n - j]$$

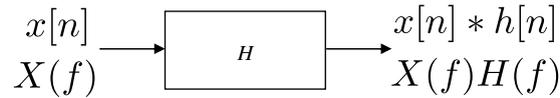


Figure 3 : Réponse d'un filtre en fréquence

Les expressions précédentes peuvent alors être exprimées dans le domaine fréquentiel en utilisant la transformée en  $Z$ . Ainsi on obtient :

$$Y(z) = X(z)H(z)$$

Remarque : un produit de convolution dans le domaine temporel est un produit dans le domaine fréquentiel et inversement. C'est une propriété de la transformée de Fourier et de la transformée en  $Z$ .

L'équation aux différences devient :

$$\sum_{i=0}^p a_i z^{-i} Y(z) = \sum_{j=0}^q b_j z^{-j} X(z)$$

En remplaçant  $Y(z)$  par  $H(z)X(z)$ , on obtient l'expression de  $H(z)$ , la fonction de transfert du filtre :

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_q z^{-q}}{a_0 + a_1 z^{-1} + \dots + a_p z^{-p}} = \frac{\sum_{j=0}^q b_j z^{-j}}{\sum_{i=0}^p a_i z^{-i}}$$

à noter que l'on considère généralement  $a_0 = 1$ .

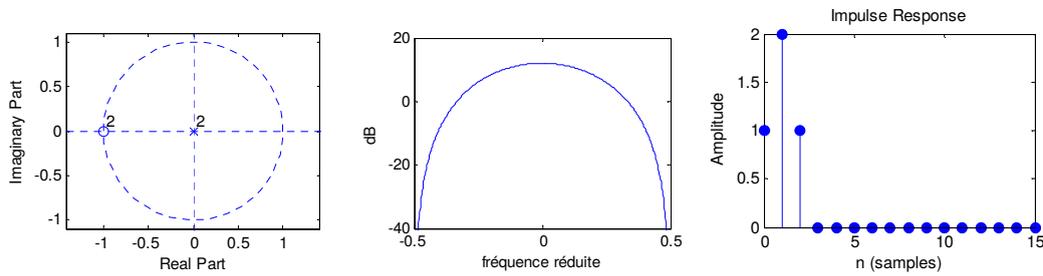
On peut distinguer deux polynômes dans l'expression de  $H(z)$  :  $B(z)$  et  $A(z)$ . Selon la valeur de leurs coefficients  $\{b_j\}_{j=0,\dots,q}$  et  $\{a_i\}_{i=0,\dots,p}$ , ces polynômes possèdent des racines particulières les caractérisant.

- Les zéros de la fonction de transfert : ce sont les racines de  $B(z)$ . Les zéros correspondent alors à des atténuations de fréquence en  $z = \exp(j2\pi f/f_e)$ . Si le module du zéro vaut 1, alors la réjection est totale.
- Les pôles de la fonction de transfert : ce sont les racines de  $A(z)$ . Les pôles correspondent à des amplifications de fréquence en  $z = \exp(j2\pi f/f_e)$ . On parle alors de fréquences de résonance. Cette fois, le module de la racine ne peut pas valoir 1 sinon  $H(z)$  n'est pas définie pour une valeur de  $z$  donnée.

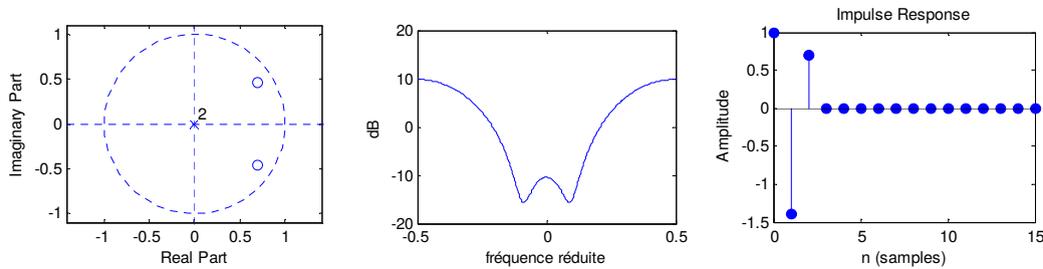
Lorsque  $A(z) = 1$ , le filtre correspondant est un filtre à réponse impulsionnelle finie (RIF) et il est toujours stable. Dans le cas contraire, il s'agit d'un filtre à réponse impulsionnelle infinie (RII) et sa stabilité est garantie par la position des pôles à l'intérieur du cercle unité.

Cette écriture permet de mettre en évidence la forme de  $H(z)$  en fonction des pôles et des zéros du filtre :  $|H(z)|$  est faible à proximité des zéros (racines de  $B(z)$ ), et  $|H(z)|$  est élevé à proximité des pôles (racines de  $A(z)$ ). En rapprochant ce fait du lien entre  $H(f)$  et  $H(z)$ , il est possible d'expliquer la forme générale de la réponse impulsionnelle en fonction de la position des pôles ( $\times$ ) et des zéros ( $\circ$ ).

Exemple d'un filtre RIF passe bas :  $y(n) = x(n) + 2x(n-1) + x(n-2)$

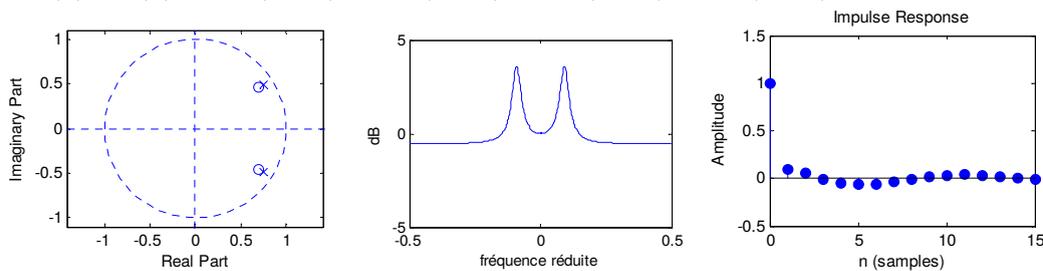


Exemple d'un filtre RIF passe-haut :  $y(n) = x(n) - 1.4x(n-1) + 0.7x(n-2)$



Exemple d'un filtre RII peigne (sélection de fréquence) :

$$y(n) = x(n) - 1.4x(n-1) + 0.7x(n-2) + 1.5y(n-1) - 0.8y(n-2)$$



Considérons les filtres suivants  $T_1$  et  $T_2$  :

$$T_1 : y[n] = \frac{1}{3}y[n-1] + \frac{1}{3}x[n] + \frac{1}{3}x[n-1] \quad (2)$$

$$T_2 : y[n] = \frac{1}{3}y[n-1] + \frac{2}{3}x[n] - \frac{2}{3}x[n-1] \quad (3)$$

### 3.2 Effet fréquentiel du filtrage

1. Créer un signal de  $N = 256$  échantillons correspondant à la superposition de trois sinusoides de fréquences différentes (une basse fréquence, une moyenne fréquence, une haute-fréquences) :

$$x(n) = \cos\left(2\pi\frac{4}{256}n\right) + \cos\left(2\pi\frac{32}{256}n\right) + \cos\left(2\pi\frac{124}{256}n\right)$$

2. Appliquer le filtre  $T_1$  au signal  $x$  suivant. Effectuer le filtrage dans le domaine temporel, c'est-à-dire en utilisant la fonction *filter*.

Visualiser le signal d'origine  $x$ , et le signal filtré  $y$ , ainsi que leurs TFD respectives  $X[k]$  et  $Y[k]$  (on prendra une TFD à  $M = N = 256$  points).

S'agit-il d'un filtre qui conserve les basses-fréquences (passe-bas) ou les hautes-fréquences (passe-haut) ?

Comment cela se traduit-il dans le domaine temporel ?

- Afficher la réponse impulsionnelle du filtre (*impz*) puis fréquentielle du filtre (on fera attention à prendre un nombre suffisant d'échantillons ( $M \geq 256$ )).

```
%réponse fréquentielle
[H,W]=freqz(b,a,M,'whole');
plot(W,abs(H));
```

- $Y[k]$  peut être également obtenu par la relation suivante :

$$Y[k] = X[k]H[k]$$

Le montrer en calculant  $Y[k]$  de cette façon et en le superposant au résultat précédent.

- Mêmes questions avec  $T_2$ .

### 3.3 Signal de parole bruité

Le rapport signal sur bruit (Signal to Noise Ratio, SNR) d'un signal bruité  $x[n] = s[n] + e[n]$  est défini comme le rapport entre la puissance du signal non bruité  $s[n]$  et la puissance du bruit  $e[n]$  :

$$RSB(x) = \frac{P_s}{P_e} = \frac{\sum_n s[n]^2}{\sum_n e[n]^2}$$

$$RSB_{dB}(x) = 10 \log_{10}(RSB(x)) = P_s^{dB} - P_e^{dB}$$

Dans cette partie, nous nous intéressons au cas d'un signal de parole échantillonné à  $f_e = 16000\text{Hz}$ . Ce signal sera bruité par un signal sinusoïdal additif de fréquence  $f_b = 0.2 \times f_e$  et d'amplitude maximale notée  $G$ , considérée à 1 dans un premier temps.

- Charger le signal *parole.wav* noté  $s[n]$  et rajouter le bruit. L'écouter en utilisant *soundsc*. (Si *soundsc* fait buguer Matlab, écouter directement le signal depuis un lecteur multimédia après l'avoir enregistré avec *audiowrite*).
- Calculer la puissance de  $s[n]$  et ajuster l'amplitude du bruit, c.à.d. la valeur de  $G$ , pour obtenir un  $RSB_{dB}$  de  $15\text{dB}$ . Visualiser le spectrogramme en utilisant *spectrogram*.

#### 3.3.1 Débruitage par filtrage sélectif

Nous considérons à présent le filtre suivant :

$$H_1(z) = 1 - 2r \cos\left(2\pi \frac{f_0}{f_s}\right) z^{-1} + r^2 z^{-2} \quad \text{pour } 0 \leq f_0 \leq \frac{f_s}{2}, 0.9 \leq r \leq 1$$

- De quel type est ce filtre ? Faire varier la valeur de  $r$  et visualiser la transformée de Fourier des différentes réponses impulsionnelles.
- Filtrer le signal de parole bruité en utilisant le filtre  $H_1$  en fixant  $r$  à 1, le reconstituer puis écouter le résultat. Comparer le spectrogramme du signal initial et celui du signal filtré.

## 4 BONUS : POUR ALLER PLUS LOIN

### 4.1 Sélectivité du filtrage

9. Observer la réponse fréquentielle du filtre précédent en amplitude et en phase (*figure, freqz(b,a)*).
10. En représentant les pôles et les zéros (*zplane*), expliquer l'allure du filtre.
11. Le filtre  $H_1(z)$  admet donc comme zéros  $z_0 = r \exp(j2\pi f_0/f_e)$  et  $\bar{z}_0$ . On peut le vérifier en appelant la fonction *roots*.  
On considère alors le filtre suivant  $H_2 = \frac{B(z)}{A(z)}$  avec  $B(z) = H_1(z)$  pour  $r = 1$ , et  $A(z)$  ayant pour zéros  $z_1 = r * z_0$  et  $\bar{z}_1$ , avec  $0 < r < 1$ . On pourra utiliser la fonction *poly* pour calculer les coefficients d'un polynôme depuis ses racines.  
Représenter la réponse fréquentielle du filtre pour plusieurs valeurs de  $r$ .  
Représenter les pôles et zéros du filtre pour  $r \approx 1$  et faire le lien avec l'allure de la réponse fréquentielle du filtre.
12. Appliquer ce filtrage au signal bruité et observer le résultat.

### 4.2 Débruitage direct sur la transformée de Fourier

Dans cette partie, on va tenter de débruiter le signal directement dans l'espace des fréquences, puis de le reconstruire par transformée inverse. Ce débruitage peut s'apparenter à de la séparation de sources (signal, bruit).

13. Quelle hypothèse faut-il sur  $M$  pour pouvoir effectuer cette transformée inverse ?
14. Visualiser la transformée de Fourier du signal bruité. Constaté la présence du bruit aux fréquences fixées.  
Afficher directement les indices de la transformée ( $[1, \dots, M]$ ) sur l'axe des fréquences.  
Utiliser ces indices pour annuler directement la valeur de la transformée de Fourier du signal aux indices correspondant aux fréquences du bruit.
15. Reconstruire le signal filtré par transformée inverse (*real(iff(...))*) et l'écouter en utilisant *soundsc*.

### 4.3 Application : Débruitage d'un signal biomédical

#### 4.3.1 Objectifs

Les signaux biomédicaux sont exploités dans des contextes tels que la médecine pour la détection de pathologies, dans des applications connectées pour le suivi de constantes physiologiques (tension artérielle, rythme cardiaque, indice de masse corporelle etc.) ou encore le contrôle de la concentration, de l'attention et de la fatigue (voitures intelligentes).

En médecine, les outils exploitant le domaine du traitement du signal et de l'image sont de plus en plus nombreux. Cela va du célèbre électrocardiogramme pour le suivi du rythme cardiaque à l'électromyogramme pour la détection de l'activité musculaire.

Dans le cas d'un électrocardiogramme par exemple, une forme particulière du signal est attendue. Cette forme est issue de l'étude préalable sur des sujets ne présentant pas de pathologie ou au contraire présentant une maladie connue. Par comparaison, le signal du patient peut alors être analysé et reconnu comme étant celui d'un patient sain ou non.



Figure 4 : Exemple d'électrocardiogramme

Dans le cas d'un électromyogramme, la détection de pathologie n'est pas le seul but recherché. Ces signaux peuvent être utilisés dans un contexte de rééducation ou encore pour piloter des prothèses intelligentes.

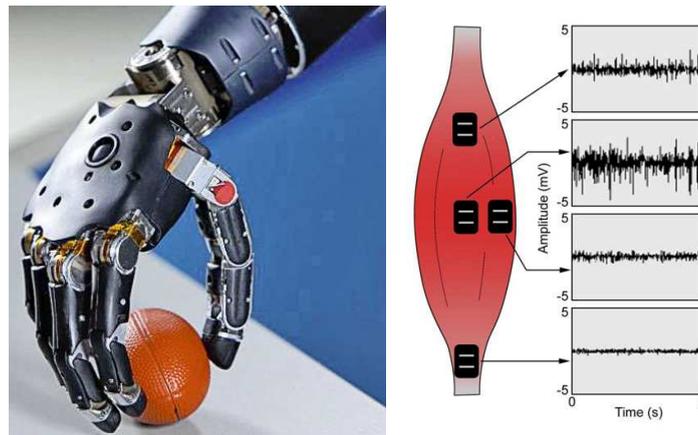


Figure 5 : Prothèse de main intelligente et électromyogramme

Ce genre de techniques peut être mis en défaut si le signal récupéré est perturbé et que sa forme s'en trouve modifiée. C'est dans ce contexte qu'intervient un ensemble de traitements permettant d'éliminer ou d'atténuer les bruits. Ce sont des perturbations dues à l'environnement (perturbations électromagnétiques du réseau électrique, bruit d'acquisition, etc.) ou provenant du matériel de mesure lui-même (bruits thermiques).

De nombreux capteurs sont aujourd'hui dotés de sorties numériques et présentent alors des problématiques de débruitage spécifiques. Parmi ces problématiques, nous proposons de nous intéresser à l'influence de composantes de nature haute fréquence sur un signal échantillonné à basse fréquence.

#### 4.3.2 Mise en pratique

1. Récupérer avec `load` les échantillons du signal biomédical `biomed.mat` et sa fréquence d'échantillonnage. Quelle est la durée de ce signal ? Tracer la représentation de ce signal en fonction du temps.
2. Quelle est la fréquence maximale pouvant être représentée ? Tracer le module de la TFD en utilisant les fonctions `fft`, `abs` et `fftshift`.
3. Récupérer les échantillons du signal biomédical `bruité` et sa fréquence d'échantillonnage. Représenter ce signal en fonction du temps ainsi que le module de sa TFD. Reconnaît-on facilement ce signal ?

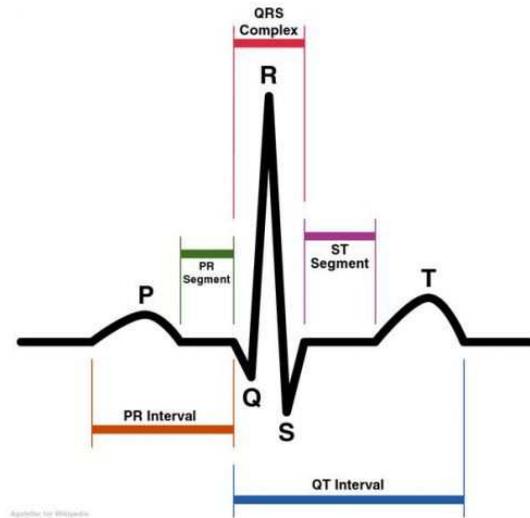


Figure 6 : Électrocardiogramme schématique

On peut définir une entrée discrète  $x[n]$  à partir du signal biomédical échantillonné. Après une transformation  $T(\cdot)$ , représentant l'opération de filtrage, le but est de récupérer le signal avec le moins de bruit possible. La sortie du filtrage est notée  $y[n] = T(x[n])$ .

4. D'après les représentations fréquentielles du signal biomédical, dans quel domaine de fréquence se trouve le bruit ? Combien de fréquences doivent-êtré éliminées ? Idéalement, quelle transformation souhaiteriez-vous appliquer au reste du spectre ? Quelle solution de filtrage envisagez-vous, RII ou RIF ?

En déduire les polynômes  $B(z)$  et  $A(z)$  du filtre à construire.

Tracer les représentations du signal bruité et débruité en fonction du temps ainsi que les spectres associés. Que constate-t-on ?

### 4.3.3 Déphasage lié au filtrage

L'opération de filtrage envisagée pour débruiter le signal biomédical consiste à sélectionner des coefficients tels que  $|H(z)| = 0$  pour  $z = e^{j2\pi f_b/f_e}$ ,  $f_b$  étant la fréquence que l'on souhaite retirer. Partout ailleurs, on souhaiterait que  $|H(z)| = 1$ . Cependant, une autre condition concerne le déphasage du filtre. En effet, si l'on souhaite conserver la forme de notre signal d'origine il est important que le retard de groupe soit constant. Pour l'étude et la classification de notre signal médical, il s'agit d'un impératif.

- Lien entre déphasage constant et filtre à phase linéaire :

Ainsi on pose :  $H(z) = |H(z)|e^{j\phi(f)}$ , on a donc  $Y\left(e^{j\frac{2\pi f}{f_e}}\right) = \left|H\left(e^{j\frac{2\pi f}{f_e}}\right)\right| e^{j\phi(f)} X\left(e^{j\frac{2\pi f}{f_e}}\right)$ .

Si on considère  $y[n]$  en prenant la transformée inverse sur  $Y\left(e^{j\frac{2\pi f}{f_e}}\right)$  :

$$y[n] = \frac{1}{f_e} \int_{-f_e/2}^{f_e/2} \left|H\left(e^{j\frac{2\pi f}{f_e}}\right)\right| e^{j\phi(f)} X\left(e^{j\frac{2\pi f}{f_e}}\right) e^{j\frac{2\pi f n}{f_e}} df$$

Si on se concentre sur  $\phi(f)$  lorsque  $f$  est proche d'une fréquence  $f_0$ , on peut donner son développement limité à l'ordre 1 :

$$\phi(f) = \phi(f_0) + \frac{2\pi}{f_e} (f - f_0) \left( \frac{d\phi(f)}{df} \right)_{|f=f_0}$$

En remplaçant  $\phi(f)$  par son développement limité :

$$y[n] = \frac{1}{f_e} e^{j\phi(f_0)} \int_{-f_e/2}^{f_e/2} \left|H\left(e^{j\frac{2\pi f}{f_e}}\right)\right| X\left(e^{j\frac{2\pi f}{f_e}}\right) e^{j\frac{2\pi}{f_e}(f-f_0)\tau(f_0)} e^{j\frac{2\pi f n}{f_e}} df$$

Avec  $\tau(f_0) = \frac{d\phi(f)}{df} \Big|_{f=f_0}$  le retard temporel lié au filtrage (pour des fréquences proches de  $f_0$ ).

Ainsi  $\tau(f_0) = cste$  signifie que  $\phi(f) = \phi_0 + \frac{\alpha 2\pi f}{f_e}$ . Le déphasage est alors une fonction linéaire de la fréquence.

- Rappels sur les différents types de filtres à phase linéaire :

En considérant que le filtrage s'effectue en utilisant  $N$  échantillons, on a :

$$H(z) = \sum_{k=0}^N h[k] z^{-k}$$

Plusieurs types de filtres sont à phase linéaire :

$$h[k] = h[N - k] \quad \text{ou} \quad h[k] = -h[N - k]$$

avec  $h[N/2] = 0$  dans le cas où  $N$  est impair.

5. À l'aide de la fonction *filter* sous Matlab, effectuer le filtrage du signal bruité en utilisant le filtre :

$$H(z) = 1 - 2 \cos\left(\frac{2\pi f}{f_e}\right) z^{-1} + z^{-2}$$

Observer son amplitude et sa phase avec *freqz*.

La comparer avec celle du filtre conçu dans la partie 3.3 Sélectivité du filtrage.

## 5 ANNEXES

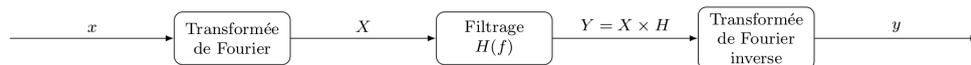
### 5.1 Séparation de sources

Un des grands domaines du traitement du signal est la séparation de sources. L'idée étant, à partir d'un mélange de différents signaux provenant de plusieurs sources (par exemple plusieurs personnes qui parlent en même temps), de parvenir à reconstruire le message associé à chacune des sources (par exemple retrouver ce qu'a dit chaque personne). C'est un problème courant en acoustique donc, mais aussi en télécommunications, en bio ingénierie, etc.

Dans notre cas, nous allons considérer la somme de deux signaux sonores (mélodies), et tenter de reconstruire chacun d'entre eux grâce à un filtrage adapté, directement dans l'espace des fréquences. Pour cela, nous allons utiliser au choix un filtre passe-bas idéal (si l'on veut supprimer les notes aiguës), passe-haut idéal (si l'on veut supprimer les notes graves). On peut aussi considérer un passe-bande (et ainsi choisir de supprimer toutes les notes aiguës et graves et ne garder que les médiums). Dans cette partie, le signal est composé de 2 mélodies sur des gammes de fréquences différentes. En pratique, la séparation de sources sur des mêmes espaces fréquentiels peut s'avérer beaucoup plus complexe. Les fonctions de transfert de ces filtres sont définies de la façon suivante et dépendent d'une ou plusieurs fréquences de coupure (selon le filtrage que l'on souhaite effectuer) :

$$\begin{array}{ccc}
 \text{Passe-bas} & \text{Passe-haut} & \text{Passe-bande} \\
 H_{LP}(f) = \begin{cases} 1 & \text{si } |f| < f_c \\ 0 & \text{sinon} \end{cases} & H_{HP}(f) = \begin{cases} 0 & \text{si } |f| < f_c \\ 1 & \text{sinon} \end{cases} & H_{BP}(f) = \begin{cases} 1 & \text{si } f_{c1} < |f| < f_{c2} \\ 0 & \text{sinon} \end{cases}
 \end{array}$$

La suite des opérations à réaliser est présentée ci-dessous :



Il s'agit ici d'un filtrage linéaire vu dans le domaine fréquentiel, correspondant donc à une multiplication de la transformée du signal avec la fonction de transfert du filtre.

1. Avec la fonction *create\_melody*, créer deux mélodies  $x_1$  et  $x_2$  d'une durée de 10 secondes, échantillonnés à  $f_e = 8000\text{Hz}$  et contenant des notes (au moins 4), pour  $x_1$  comprises entre 50 et 71 et pour  $x_2$  comprises entre 72 et 90.

Écouter les signaux et visualiser leurs transformées de Fourier.

2. Créer le signal  $x = x_1 + x_2$ .

Écouter le signal et visualiser sa transformée de Fourier.

Quelle fréquence de coupure  $f_c$  peut-on utiliser pour séparer les deux signaux  $x_1$  et  $x_2$  ?

3. Créer un filtre adapté (vecteur de 1 et de 0, de la taille de la transformée de Fourier), multiplier ce vecteur à la transformée du signal et utiliser la transformée inverse (*ifft*) pour reconstruire à partir de  $x$  les signaux  $y_1$  et  $y_2$  estimant respectivement  $x_1$  et  $x_2$ .

Quelle hypothèse faut-il sur  $M$  pour pouvoir effectuer cette transformée inverse ?

4. Écouter les signaux obtenus et les comparer aux sources réelles.

Calculer la distance euclidienne entre  $x_1$  et  $y_1$  et celle entre  $x_2$  et  $y_2$ .

5. Charger le signal *melody\_multiple.wav*. En utilisant l'information du spectrogramme, essayer d'appliquer la stratégie précédente pour séparer la mélodie du fond.

On pourra calculer la distance euclidienne entre le signal bruité et l'original contenu dans *melody\_single.wav*.

## 5.2 Notion de bruit blanc gaussien

Une séquence Gaussienne de Bruit Blanc Gaussien Centré a trois propriétés caractéristiques :

### Blanc

La notion de blancheur implique que tous ses échantillons sont décorrélés, *i.e.*,

$$R_{BB}(t_i, t_j) = \sigma^2 \delta_{t_i, t_j} \quad (4)$$

où  $\delta_{t_i, t_j}$ , le symbole de Kronecker, est tel que  $\delta_{t_i, t_j} = 0$  si  $t_i \neq t_j$  et  $\delta_{t_i, t_j} = 1$  si  $t_i = t_j$ . De manière général, un processus blanc est stationnaire. On a donc  $R_{BB}(l) = \sigma^2 \delta_l$ .

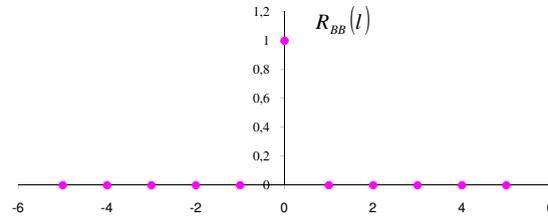


Figure 7 : Fonction d'autocorrélation du bruit blanc

### Centré

Sa moyenne est nulle.

### Gaussien

Un processus aléatoire est un processus gaussien si pour tout ensemble d'instantanés  $\{t_i\}$ , le vecteur ligne aléatoire correspondant  $X = [X_1, \dots, X_n]$ , a une densité de probabilité conjointe multidimensionnelle gaussienne  $X_i = X(t_i)$ . Chaque  $X_i$  possède une densité de probabilité marginale :

$$p_{X_i}(x_i) = \frac{1}{\sqrt{2\pi}\sigma_{X_i}} \exp\left(-\frac{[x_i - m_{x_i}]^2}{2\sigma_{X_i}^2}\right). \quad (5)$$

Si le processus est réel et  $N = 1$  (cas mono-dimensionnel), sa densité de probabilité est définie par :

$$p(x; t) = \frac{1}{\sqrt{2\pi}\sigma_x^2(t)} \exp\left(-\frac{[x - m_x(t)]^2}{2\sigma_x^2(t)}\right). \quad (6)$$

Si le processus est complexe et  $N = 1$ , sa densité de probabilité est définie par :

$$p(x; t) = \frac{1}{\pi\sigma_x^2(t)} \exp\left(-\frac{[x - m_x(t)]^2}{\sigma_x^2(t)}\right). \quad (7)$$

Si le processus est réel et de dimension  $N$ , sa densité de probabilité est définie par :

$$p(x; t) = \frac{1}{(\sqrt{2\pi})^N |C_{XX}|^{1/2}} \exp\left(-\frac{1}{2} [x - m_x(t)] C_{XX}^{-1} [x - m_x(t)]^T\right). \quad (8)$$

Si le processus est complexe et de dimension  $N$ , sa densité de probabilité est définie par :

$$p(x; t) = \frac{1}{\pi^N |C_{XX}|} \exp\left(-[x - m_x(t)] C_{XX}^{-1} [x - m_x(t)]^T\right). \quad (9)$$

$C_{XX}$  est la matrice d'autocovariance.

### 5.3 Débruitage d'un signal de parole

Le rehaussement du signal de parole consiste en l'amélioration d'un ou de plusieurs aspects perceptifs du signal vocal quand ce dernier est perturbé par un bruit dû à l'environnement et/ou à la transmission. Un traitement de débruitage est ainsi souhaitable dans une voiture, lors de conversations main libre, dégradées par le bruit du moteur et les conditions du trajet (pluie, fenêtre ouverte, etc.). La situation est comparable dans le domaine de l'aéronautique, lors de communications entre le pilote et la tour de contrôle. C'est aussi le cas dans les situations suivantes :

- les postes de téléphone situés dans des environnements où les bruits des conversations et des équipements informatiques perturbent les communications ;
- les vidéophones et systèmes de téléconférence ;
- la restauration d'enregistrements anciens ;
- les systèmes de reconnaissance utilisés dans des environnements bruités.

Nous supposons ici qu'un seul microphone est utilisé pour la prise de son et nous cherchons à réduire le bruit sans introduire de distorsion trop importante sur le signal utile.

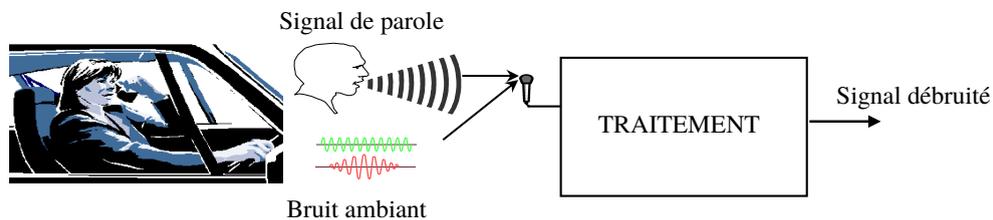


Figure 8 : Débruitage du signal de parole

Nous nous placerons dans le cas simple où le bruit additif est un bruit blanc.

Ce document s'organise de la manière suivante : dans un premier temps, quelques notions sur la parole sont données telles que les notions de voisement/non voisement, fréquence fondamentale, etc. Puis, une présentation théorique de l'approche de rehaussement est réalisée.

#### 5.3.1 Caractérisation des voyelles et notions de formants

Par essence, le signal de parole est *non stationnaire* ; cependant, on peut faire une hypothèse de quasi-stationnarité, notamment pour des voyelles, sur une durée de 40 ms. En fenêtrant le signal de parole en trames de 10 à 40 ms, on peut faire apparaître un caractère pseudo-périodique ou/et aléatoire. Plusieurs cas peuvent être envisagés :

- les trames **voisées** qui présentent un caractère périodique ou pseudo périodique ;
- les trames non voisées ou aléatoires ;
- les "bruits voisés" qui sont une combinaison simultanée des deux cas précédents ;
- les trames **mixtes** représentant les zones de transition entre des sons voisé et non voisé.

Le spectre d'un signal voisé est riche en fréquences harmoniques de la fréquence fondamentale. Si l'on regarde la figure 2, on constate en outre que l'enveloppe spectrale du signal analysé décroît doucement vers les hautes fréquences. Ses résonances sont appelées *formants*

et notées  $F1$ ,  $F2$ , etc. dans l'ordre des fréquences croissantes. Les deux premiers caractérisent en général la voyelle prononcée alors que les deux suivants sont plutôt liés au locuteur. Pour un locuteur masculin, tous les formants sont inférieurs à 5 kHz.

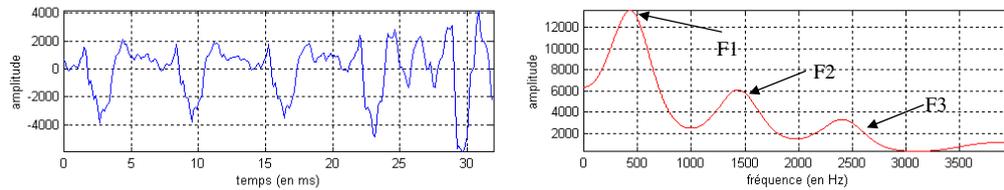


Figure 9 : Mise en évidence des trois premiers formants pour le segment voisé analysé

On peut aussi observer les formants sur un spectrogramme. Le spectrogramme est une représentation graphique fréquence $\times$ temps, qui traduit l'évolution de la richesse spectrale du signal au cours du temps ; la présence de formants correspond alors à ces gammes de fréquence dont l'énergie est particulièrement élevée et qui apparaissent sous la forme de bandes sensiblement parallèles à l'axe des abscisses. Quand le spectrogramme est en couleur, on adopte généralement une palette allant du rouge pour représenter une forte énergie au bleu pour représenter une faible énergie. C'est notamment le cas à la Figure 10.

### 5.3.2 La fréquence fondamentale

Lorsque les cordes vocales oscillent de façon quasi-périodique, le signal de parole émis est lui aussi quasi-périodique ; il est dit voisé, c'est notamment le cas lorsque l'on prononce des voyelles telles que le /a/. La période du signal est alors appelée période fondamentale. On nomme l'inverse de cette période, la fréquence fondamentale ou pitch. La valeur moyenne du pitch est variable selon les locuteurs. Basse et de l'ordre de 80 à 200 Hz pour les hommes, elle peut atteindre 600 Hz pour les enfants, la voix féminine ayant une fréquence fondamentale comprise de manière générale entre 150 et 450 Hz. Un exemple de spectrogramme d'un signal de parole<sup>2</sup> est donné en Figure 10.

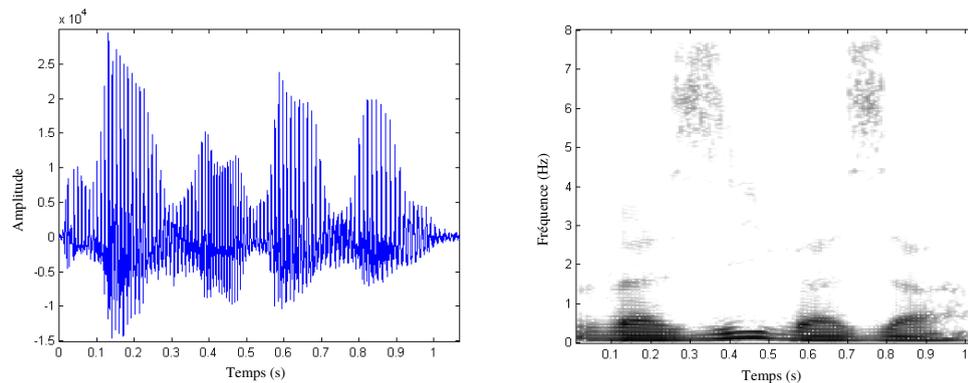


Figure 10 : Spectrogramme d'un signal "waziwaza"<sup>1</sup> mettant en évidence la présence de [z]

<sup>2</sup>Nous remercions Matra et l'ENST pour la base de données de signaux audio qu'elles ont bien voulu nous fournir.

### 5.3.3 Bilan

On retiendra de cette brève présentation du signal de parole deux de ses caractéristiques majeures :

- son caractère non-stationnaire (nécessitant un traitement par frames),
- et son éventuelle nature quasi-périodique, à laquelle se rapportent les notions de voisement et de non voisement.

La parole est un moyen de communication au même titre que l'écriture. On attend donc d'un message vocal qu'il soit intelligible, c'est à dire facile à comprendre.

Mais le signal de parole peut être dégradé par un bruit additif, notamment lors de communications longue-distance sur des lignes bruitées, ou lorsque l'on utilise un téléphone portable dans une voiture. Des traitements de débruitage du signal peuvent alors être mis en œuvre. Dans le chapitre suivant, nous nous proposons de faire l'état de l'art des méthodes de rehaussement qui utilisent un unique microphone.

### 5.3.4 Sur le rehaussement monovoie par atténuation spectrale à court terme

Les solutions élaborées pour le débruitage sont multiples. Certaines emploient plusieurs microphones ; on peut effectuer un rehaussement bivoie, en plaçant un premier microphone directionnel à proximité du locuteur alors qu'un second microphone, omnidirectionnel, est positionné le plus près possible de la source de bruit. Ici, nous focalisons notre attention sur une approche de rehaussement de type monovoie. Une des difficultés dans ce contexte est d'obtenir une bonne approximation des données statistiques du bruit perturbateur.

Les techniques de rehaussement par atténuation spectrale à court terme consistent à appliquer une atténuation spectrale, dépendant de l'estimation du bruit additif, à chaque composante fréquentielle. Plus précisément, ces méthodes nécessitent les opérations suivantes :

1. On effectue un traitement tous les  $N$  échantillons. La durée des tranches d'analyse est comprise entre 20 et 40 ms. Les blocs que l'on traite successivement peuvent être en outre disjoints ou se recouvrir partiellement ; on adopte d'ailleurs couramment un recouvrement de 50%. Dans le contexte de la téléphonie, où le signal est échantillonné à 8 kHz, on peut travailler sur des blocs de 256 échantillons dont le recouvrement s'élève à 128 échantillons.
2. Puis, on applique une règle de suppression de bruit au spectre du signal bruité fenêtré  $Y_w(f)$ , que l'on nomme aussi spectre du signal à court terme, pour en atténuer chaque canal fréquentiel et obtenir ainsi  $\hat{S}(f)$ , l'estimation du spectre à court terme du signal de parole original :

$$\hat{S}_w(f) = H_w(f, B(f))Y_w(f) \quad (10)$$

Cette règle varie selon les méthodes utilisées et dépend de manière générale de la variance du bruit additif dont on fait une estimation préalable ; on peut tirer avantage des instants de "silence", où seul le bruit additif est présent, pour évaluer et/ou actualiser la variance du bruit. On en fait alors une approximation long terme, en supposant le bruit ergodique<sup>3</sup>.

3. Une transformation de Fourier inverse permet enfin d'avoir une estimation du signal fenêtré. Puis on procède à la reconstruction du signal, en appliquant une méthode dite d'addition-recouvrement (addition de l'ensemble des signaux fenêtrés débruités).

Le schéma de principe de rehaussement par atténuation spectrale est donné à la Figure 12.

Parmi les méthodes qui obéissent à ce type de traitement, la soustraction spectrale est l'une des techniques les plus employées par les traiteurs de signal de parole.

---

<sup>3</sup>ergodique au sens de la moyenne" revient à dire que l'on peut approximer la moyenne d'ensemble du processus, c'est à dire l'espérance mathématique par la moyenne temporelle.

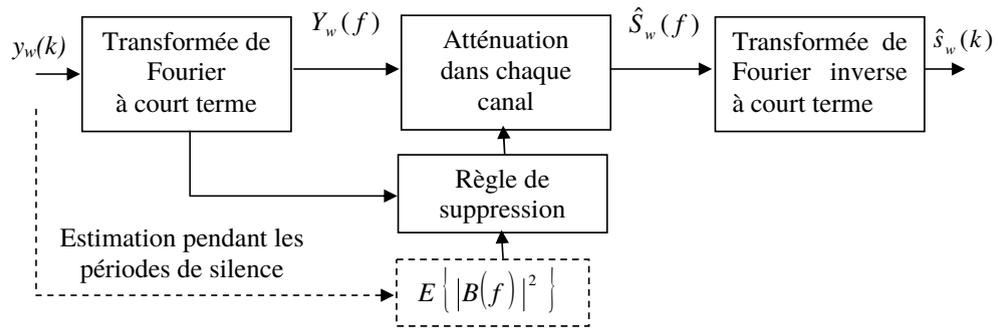


Figure 11 : Schéma de principe de rehaussement par atténuation spectrale

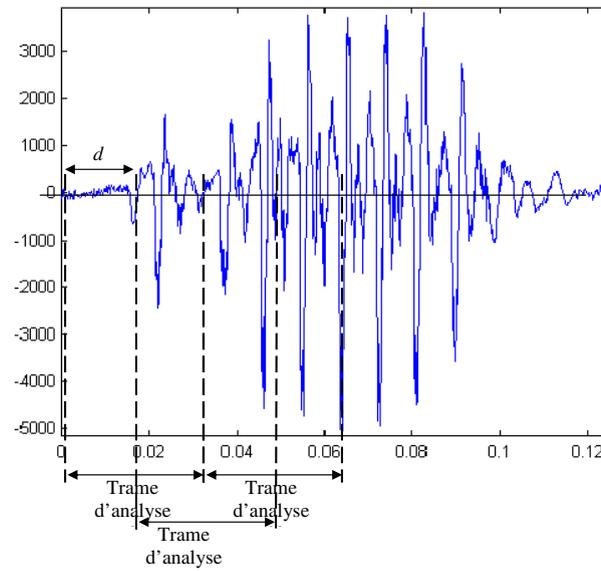


Figure 12 : Traitement par trame avec recouvrement d'un signal

## 5.4 Formulaire

Domaine continu	Domaine discret
$X(p) = \int_{-\infty}^{+\infty} x(t) \exp(-pt) dt$	$X(z) = \sum_{n=-\infty}^{+\infty} x(n) z^{-n}$
Équation différentielle à coefficients constants $\sum_{k=0}^{K-1} C_k \frac{d^k y(t)}{dt^k} = \sum_{l=0}^{L-1} D_l \frac{d^l x(t)}{dt^l}$	Équation récurrentes $\sum_{k=0}^{K-1} a(k) y(n-k) = \sum_{l=0}^{L-1} b(l) x(n-l)$
fréquence : $p = j\omega = j2\pi f$	fréquence : $z = \exp(j\omega T_e) = \exp(j2\pi \frac{f}{f_e})$
Fonction de Transfert $H(p) = \frac{\sum_{l=0}^{L-1} D_l p^l}{\sum_{k=0}^{K-1} C_k p^k}$	Fonction de Transfert $H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{l=0}^{L-1} b(l) z^{-l}}{\sum_{k=0}^{K-1} a(k) z^{-k}}$
Stabilité (système causaux) : pôles à partie réelle négatives	Stabilité (système causaux) : pôles à l'intérieure du cercle unité

On rappelle en outre que :

$$\sum_{n=0}^{N-1} a^n = \frac{1-a^N}{1-a} \quad \text{d'où} \quad \sum_{n=0}^{+\infty} a^n = \frac{1}{1-a} \quad \text{pour} \quad |a| < 1$$

## 5.5 Mémo Matlab

```

%Chargement d'un signal 'audio'
[s, fe] = audioread('signal.wav');
N = length(s)

%Chargement d'une archive matlab
load('data.mat'); %Données chargées directement dans le workspace
X = load('data.mat'); %Données dans X
s = X.s; %Accès aux champs de X

%Valeur maximale
max_s = max(s);
%Accès valeurs
s_10_20 = s(10:20); valeurs 10 à 20

%Mise à zéro
s = zeros(N,1); %ones() existe aussi
s = s*0;
s(5:20) = 0; %mise à 0 des coeff 5 à 20

%Vecteurs/matrices
vect = (1:2:11); %(début:(pas):fin) [1 3 5 7 9 11] (1x6)
vect = linspace(1,11,6); %(début,fin,N) [1 3 5 7 9 11] (1x6)
mat = [1 1; 2 2; 3 3]; %matrice (3x2)
vect_t = vect'; %transposée (6x1)

%Attention aux * de vecteurs et matrices !!!
vect_x = vect*vect; %ERREUR DE DIMENSION
vect_1x6 = vect.*vect; %terme à terme
vect_6x6 = vect'*vect; %produit matriciel
vect_1 = vect*vect'; %produit matriciel

%Sous-échantillonnage
s_ech = s(1:2:end); %une valeur sur deux

%Somme des termes au carrés
sum_s = sum(s.^2); %équivalent à sum(s.*s);

%Création d'un cosinus
N = 1024; t = 0:N-1; %de longueur N
fe = 256; f0 = 64; %freq ech fe, porteuse f0
s = cos(2*pi*f0/fe*t);

%Affichage
figure, %Création d'une nouvelle figure
subplot(121) %Affichages multiples 1x2
plot(1:length(s), s) %première figure
title('Signal s');
subplot(122) %deuxième figure
plot(s); %indices 1 à length(s) par défaut
hold on; superposition
plot(1:length(s2), s2, 'Color', [0 0 1]); title('Comparaison de s et s2');
xlabel('Signal'); ylabel('Intensité');

```