**Université de Tunis El Manar**
**École Nationale d'Ingénieurs de Tunis**

# Debruitage par inversion de puissance

Mini-projet de traitement de signal

[Groupe 2]

**Elaboré par:**
**KHENISSI Nour**
**BELHAJ MESSAOUD Ines**
**AGREBI Afef**
**TELMINI mohamed Fyras**

**Année universitaire :2019/2020**
**Filière: MIndS**

# 1 Introduction

La discrétisation d'un signal naturel par échantillonnage permet de récupérer un signal numérique qu'on peut traiter pertinemment à l'aide des machines numériques. Mais il est fréquent que le signal récupéré soit dégradé par des fluctuations parasites ou des éléments perturbateurs qu'on appelle du bruit, ce qui le rend parfois inintelligible. Ainsi pour améliorer la perception du signal, on a recours aux algorithmes de débruitage permettant de réduire le bruit.

On s'intéresse dans ce projet à un signal audio qu'on cherche à débruiter. Ainsi la technique de débruitage qu'on met en œuvre, se fait à partir de la méthode d'inversion de puissance. Pour ce faire, on récupère deux enregistrements audio $x_1(n)$ et $x_2(n)$ qui sont fortement bruités, présentant un très faible rapport signal sur bruit et qu'on exprime de la manière suivante :

$$\begin{cases} x_1(n) = a_1 x(n) + b_1 w(n) \\ x_2(n) = a_2 x(n) + b_2 w(n) \end{cases}$$

où $x(n)$ est le signal d'origine qu'on cherche à débruiter, $w(n)$ est un bruit additif idépendant de $x(n)$ et $a_1$,$a_2$,$b_1$ et $b_2$ sont des constantes positives inconnues, créant un SNR très défavorable qui rend l'audio imperceptible. Et à partir de ces deux enregistrements, on applique notre méthode de traitement de signal, qu'on explicitera étape par étape, afin d'avoir, au final, un signal audio débruité.

# 2 Partie théorique

## 2.1 Minimisation de la puissance du signal $y(n)$

Comme première étape, on construit le signal $y(n)$ à partir des deux signaux dont on dispose $x_1(n)$ et $x_2(n)$, tel que $y(n) = x_1(n) - \rho x_2(n)$

Le but est de déterminer l'expression de $\rho$, sachant que $\rho$ est un réel qui minimise la puissance de $y(n)$. Pour ce faire on a calculé la puissance de $y(n)$ :

$E[y(n)^2] = E[(x_1(n) - \rho x_2(n))^2]$

$= E[x_1(n)^2] - 2\rho E[x_1(n).x_2(n)] + \rho^2 E[x_2(n)^2]$

$= E[x_1(n)^2] - \rho(2E[x_1(n).x_2(n)] - \rho E[x_2(n)^2])$

On obtient alors une fontion de $\rho$ qu'on note $f(\rho)$ dont on exprime la dérivée première par rapport à $\rho$, on aura :

$f'(\rho) = -2E[x_1(n).x_2(n)] + 2\rho E[x_2(n)^2]$

Pour déterminer le minimum de la puissance de $y(n)$ on prend $f'(\rho) = 0$, puis on calculera la dérivée seconde par rapport à $\rho$.

$f'(\rho) = 0$

$\Longrightarrow -2E[x_1(n).x_2(n)] + 2\rho E[x_2(n)^2] = 0$

$\Longrightarrow 2E[x_1(n).x_2(n)] = 2\rho E[x_2(n)^2]$

Donc la dérivée s'annule en :

$\Longrightarrow \rho = \frac{E[x_1(n).x_2(n)]}{E[x_2(n)^2]}$

On étudie par suite le signe de la dérivée seconde :

$f''(\rho) = 2 \cdot E[x_2(n)^2] > 0 \Longrightarrow$ Alors $\rho$ minimise la puissance de y(n).

On explicite finalement l'expression de $\rho$ :

$\rho = \frac{E[x_1(n).x_2(n)]}{E[x_2(n)^2]} = \frac{E[(a_1x(n)+b_1w(n)).(a_2x(n)+b_2w(n))]}{E[a_2x(n)+b_2w(n)]} = \frac{E[a_1a_2x(n)^2+a_1b_2w(n)x(n)+b_1a_2w(n)x(n)+b_1b_2w(n)^2]}{E[a_2^2x(n)^2+2a_2b_2x(n)w(n)+b_2^2w(n)^2]}$

Puisque x(n) et w(n) sont indépendants, on se permet d'écrire $E[w(n)x(n)] = E[w(n)].E[x(n)] \Longrightarrow$

$\rho = \frac{a_1a_2E[x(n)^2]+a_1b_2E[w(n)]E[x(n)]+b_1a_2E[w(n)]E[x(n)]+b_1b_2E[w(n)^2]}{a_2^2E[x(n)^2]+b_2^2E[w(n)^2]+2a_2b_2E[w(n))E(x(n)]}$

En supposant que $E[w(n)] = 0$, on aura :

$\rho = \frac{a_1a_2E[x(n)^2]+b_1b_2E[w(n)^2]}{a_2^2E[x(n)^2]+b_2^2E[w(n)^2]}$

## 2.2 Inversion de la puissance

Le rapport signal sur bruit est un nombre sans unité qui caractérise la qualité avec laquelle une information est transmise, elle est d'autant plus importante que le bruit est faible. C'est le rapport des puissances $\frac{P_x}{P_{bruit}}$.

On calcule SNR sur $x_2$ :

$SNR_{x_2} = \frac{E[a_2^2x(n)^2]}{E[b_2^2w(n)^2]} = \frac{a_2^2}{b_2^2}\frac{E[x(n)^2]}{E[w(n)^2]}$

On calcule SNR sur y suivant l'expression $\rho$ qu'on a déterminé :

$SNR_y = \frac{E[(a_1-\rho a_2)^2x(n)^2)]}{E[(b_1-\rho b_2)^2w(n)^2)]}$ , le calcul donne d'une part:

$(a_1 - \rho a_2)^2 = (a_1 - \frac{a_1a_2E[x^2]+b_1b_2E[w^2]}{a_2^2E[x^2]+b_2^2E[w^2]}a_2)^2 = (\frac{a_1a_2^2E[x^2]+a_1b_2^2E[w^2]-a_1a_2^2E[x^2]-b_1b_2a_2E[w^2]}{a_2^2E[x^2]+b_2^2E[w^2]})^2$

$= (\frac{(a_1b_2^2-b_1b_2a_2)E[w^2]}{a_2^2E[x^2]+b_2^2E[w^2]})^2$

D'autre part on :

$(b_1 - \rho b_2)^2 = (b_1 - \frac{a_1a_2E[x^2]+b_1b_2E[w^2]}{a_2^2E[x^2]+b_2^2E[w^2]}b_2)^2 = (\frac{b_1a_2^2E[x^2]+b_1b_2^2E[w^2]-b_1b_2^2E[w^2]-a_1a_2b_2E[x^2]}{a_2^2E[x^2]+b_2^2E[w^2]})^2 = (\frac{(b_1a_2^2-a_1a_2b_2)E[x^2]}{a_2^2E[x^2]+b_2^2E[w^2]})^2$

On obtient alors:

$SNR_y = \frac{(a_1b_2^2-a_2b_1b_2)^2E[w^2]^2}{(b_1a_2^2-a_1a_2b_2)^2E[x^2]^2}\frac{E[x^2]}{E[w^2]} = \frac{b_2^2(a_1b_2-a_2b_1)^2E[w^2]}{a_2^2(a_2b_1-a_1b_2)^2E[x^2]} = \frac{b_2^2E[w^2]}{a_2^2E[x^2]} = \frac{1}{SNR_{x_2}}$

On conclut que la valeur de $\rho$ inverse le rapport de puissance sur $y(n)$. Sachant que SNR sur $x_2$ est très faible, son inverse est donc d'une très grande valeur, signifiant que le bruit est négligeable. On a parvenu donc par cette inversion à récuperer un signal débruité.

# 3 Partie pratique

On explicite dans cette partie le code matlab qu'on a implémenté tout en expliquant son fonctionnement.

## 3.1 Lecture des fichiers audio

La fonction audioread() nous permet de récupérer un signal audio échantillonné à une fréquence f et de le sauvegarder dans sa variable correspendante. Nous sauvegardons aussi les fréquences d'échantillonnage f1 et f2 de $x_1$ et $x_2$ respectivement, mais comme ces fréquences sont égales on peut utiliser l'une ou l'autre arbitrairement par la suite.

## 3.2 calcul de $\rho$

Puisqu'il s'agit d'un signal stationnaire et érgodique, on peut confondre les moyennes statistiques et les moyennes temporelles.
En utilisant la formule déterminée dans 2.1 ($\rho$=$\frac{E(x_1x_2)}{E(x_2^2)}$) On peut donc estimer $E[x_2^2]$ par $\frac{1}{N}\sum_1^N x_2(n)^2$ et $E[x_1x_2]$ par $\frac{1}{N}\sum_1^N x_1(n)x_2(n)$, la fonction "mean" prédéfinie en matlab fera l'affaire.

## 3.3 Signal débruité

Le signal y(n) est obtenu d'après la formule de l'énoncé $y(n) = x_1(n) - \rho x_2(n))$

**Enregistrement du signal et écoute** la fonction soundsc() nous permet d'écouter le signal obtenu. En l'écoutant, nous nous sommes rendus compte que le signal x(n) contenu dans $x_1(n)$ et $x_2(n)$, qui étaient très bruités, était en réalité une partie de la chanson "Senza una donna" de Zucchero & Paul Young!. On l'a par la suite sauvegardé dans un fichier nommé "y.wav" par moyen de la fonction audiowrite(), cette fonction prend en paramètres le signal y, le nom du fichier de destination et la fréquence par laquelle on l'a échantilloné (dans notre cas c'est f1 ou f2 puisqu'elles sont égales).

# 4 Analyse des spectres et conclusion

L'observation des spectres de $x_1$,$x_2$ et y nous renseignent sur les effets du débruitege qu'on a appliqué. La création de ces spectres se fait en calculant la transformée de fourrier centrée de chaque signal et de son échelle de fréquence correspendante.

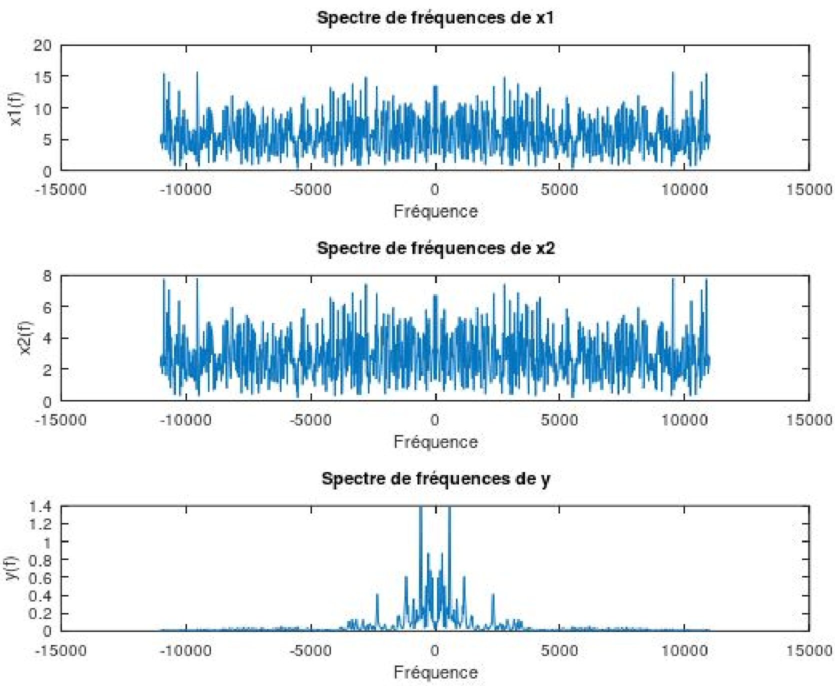


**Figure 1:** Spectres des signaux

En observant les spectres obtenus, on remarque que x1 et x2 ont des allures très similaires mais diffèrents par leurs amplitudes, cela est dû aux facteurs $a_1, a_2, b_1 et b_2$ qui diffèrent pour les deux signaux. Ces deux spectres sont également très semblabes à celui d'un bruit blanc. Cela s'explique par le fait qu'ils sont très bruités et donc contiennent le signal bruit additif $w(n)$ avec un rapport signal sur bruit très faible, reflètant le fait que le signal original $x(n)$ est dominé par le bruit additif $w(n)$. Le spectre de y est remarquablement moins bruité et semble contenir, théoriquement, le signal $x(n)$. Cela reflète l'inversion du SNR sur x2 donc du fait que dans $y(n)$, c'est le signal original $x(n)$ qui domine le bruit. Le bruit semble persister mais à des amplitudes extrèmement faibles donc il est presque inaudible.

On conclut donc que le débruitage par inversion de puissance nous a permis de récupérer un signal quasiment indiscernable du signal original $x(n)$. Cependant, cette technique de débruitage nécessite qu'on aie au moins deux signaux qui contiennent le signal original et le bruit additif avec des coefficients différents pour pouvoir restituer le signal original, cette méthode n'est donc pas applicable si on ne dispose que d'un seul signal bruité $x_1$. D'autres techniques existent comme par exemple le débruitage par filtre de Wiener.