

Débruitage d'un signal sonore par transformée discrète en ondelettes

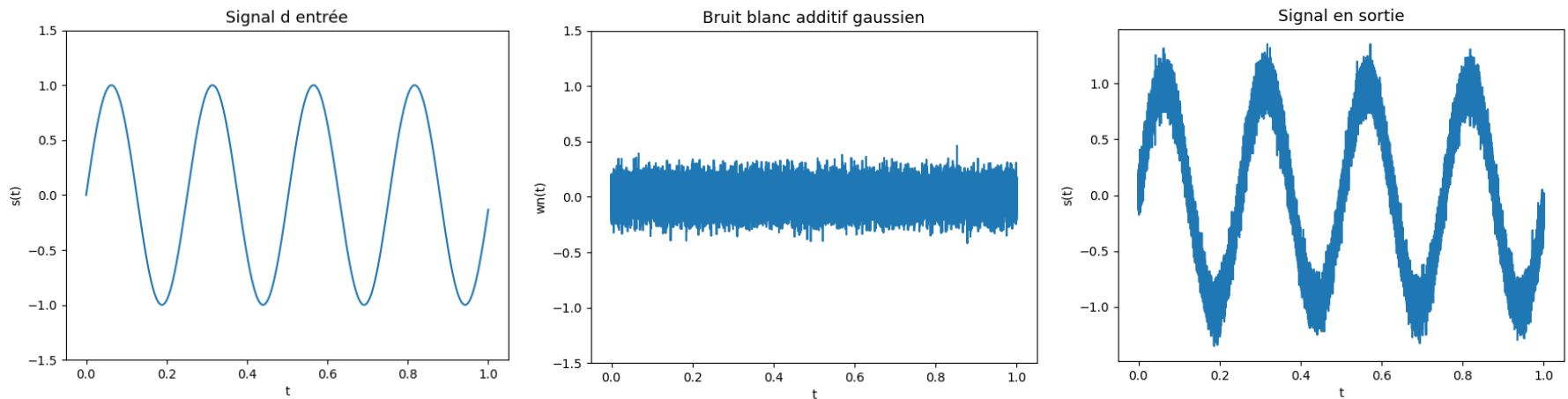
Épreuve du TIPE 2023

Martin DUFFAUD

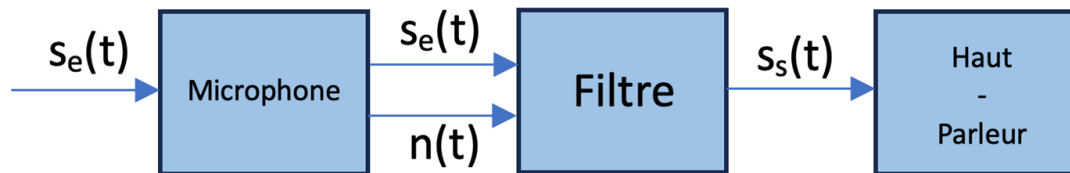
Numéro de candidat : 11524

Débruitage d'un signal : contexte d'utilisation et démarche employée

- a) Le débruitage d'un signal est l'élimination de tout résidu (appelé bruit) due à l'erreur dans l'enregistrement du signal.



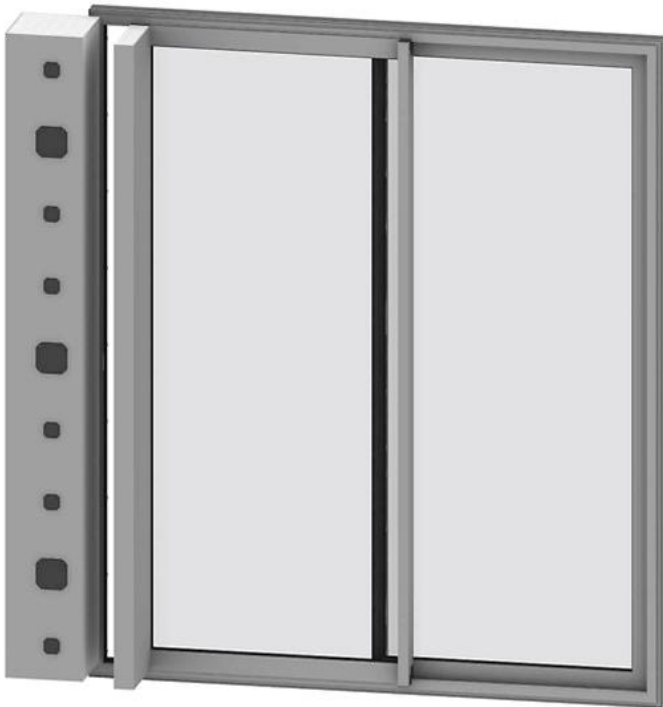
De gauche à droite : signal d'entrée, bruit additif (d'écart type 0.1) et signal en sortie d'un microphone



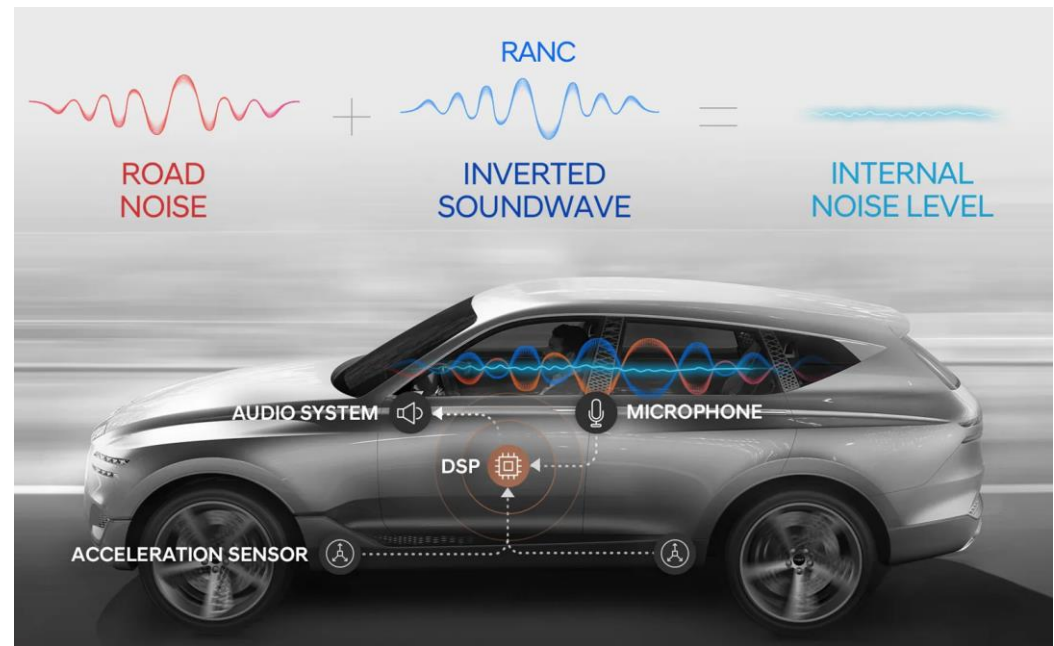
En général, le filtrage est réalisé numériquement ou par un montage électronique, dont les caractéristiques dépendent de l'utilisation du microphone

Débruitage d'un signal : contexte d'utilisation et démarche employée

b) Un objectif : la réduction des nuisances sonores



Prototype de fenêtre Anti-Bruit Active®, Technal



L'Active Noise Control pour les habitacles des voitures Genesis (Hyundai),
ici le modèle G70

Débruitage d'un signal : contexte d'utilisation et démarche employée

c) Fonctionnement et pertinence de l'utilisation du débruitage



Prototype de casque développé par l'IRSST, Québec

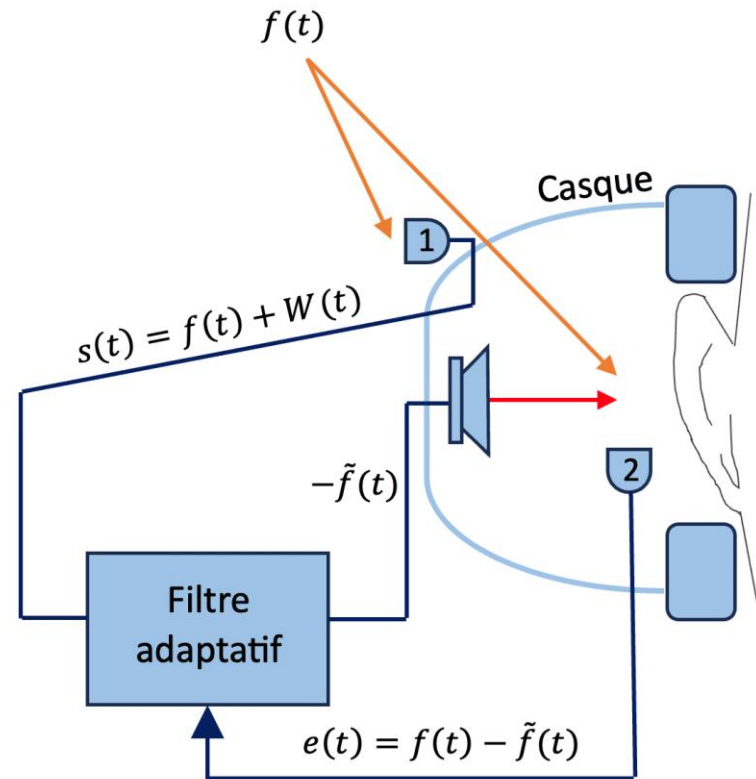


Schéma représentatif du système

Objectifs de ce TIPE

- Proposer une alternative à l'utilisation d'un microphone d'erreur pour débruiter un signal : les ondelettes.
- Évaluer la pertinence des résultats obtenus avec les ondelettes de Haar.
- Confronter les résultats avec les hypothèses.

Dans quelle mesure les ondelettes sont des outils qui permettent de débruiter un signal ?

Introduction à la théorie des ondelettes

a) Les ondelettes de Haar

$$\phi: x \mapsto \begin{cases} 1 & \text{si } x \in [0,1] \\ 0 & \text{sinon} \end{cases}$$

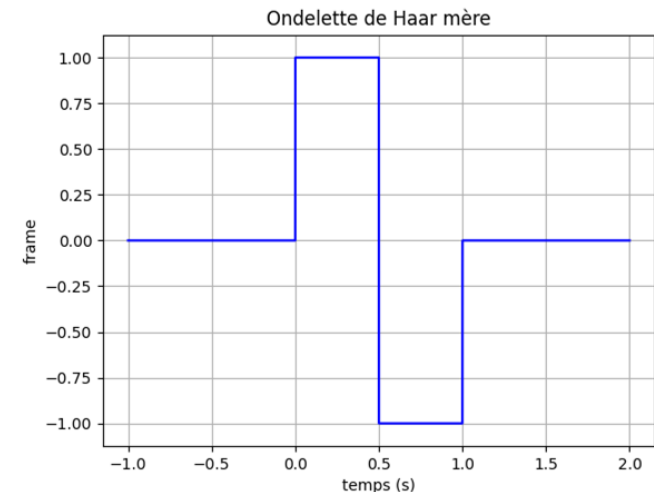
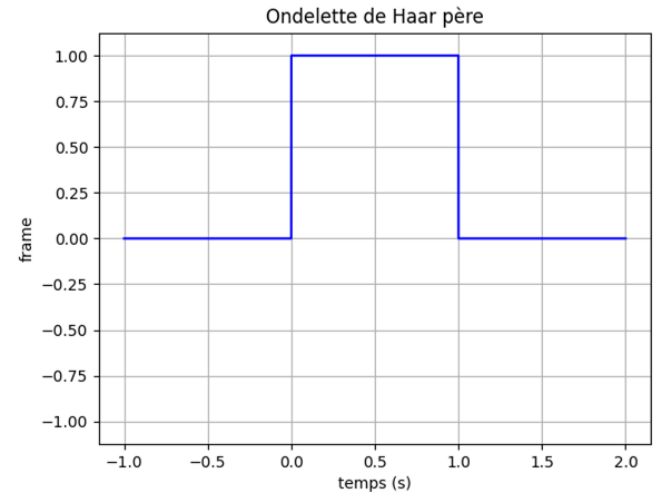
$$\forall p \in \mathbb{N}, \forall k \in \llbracket 0, 2^p - 1 \rrbracket, \quad \forall x \in \mathbb{R}, \phi_{p,k}(x) = \phi\left(2^p \left(x - \frac{k}{2^p}\right)\right)$$

L'ondelette père de Haar, et les ondelettes fils

$$\psi: x \mapsto \begin{cases} 1 & \text{si } x \in \left[0; \frac{1}{2}\right[\\ -1 & \text{si } x \in \left[\frac{1}{2}; 1\right[\\ 0 & \text{sinon} \end{cases}$$

$$\forall p \in \mathbb{N}, \forall k \in \llbracket 0, 2^p - 1 \rrbracket, \quad \forall x \in \mathbb{R}, \psi_{p,k}(x) = \psi\left(2^p \left(x - \frac{k}{2^p}\right)\right)$$

L'ondelette mère de Haar, et les ondelettes filles



Introduction à la théorie des ondelettes

b) Quelques propriétés des ondelettes de Haar

$$\forall p \in \mathbb{N}^*,$$

$$\forall (k, k') \in \llbracket 0, 2^p - 1 \rrbracket, k \neq k' \Rightarrow \langle \phi_{p,k}, \phi_{p,k'} \rangle = 0$$

$$\forall (k, k') \in \llbracket 0, 2^p - 1 \rrbracket, k \neq k' \Rightarrow \langle \psi_{p,k}, \psi_{p,k'} \rangle = 0$$

$$\forall p \in \mathbb{N}, E_p := \text{Vect}(\phi_{p,k})_{k \in \llbracket 0, 2^p - 1 \rrbracket}$$

$$\forall p \in \mathbb{N}, F_p := \text{Vect}(\psi_{p,k})_{k \in \llbracket 0, 2^p - 1 \rrbracket}$$

Propriétés orthogonales des ondelettes fils et filles, espaces engendrés respectifs de dimensions 2^p

$$\forall p \in \mathbb{N}^*,$$

$$\forall (k, k') \in \llbracket 0, 2^{p-1} - 1 \rrbracket, \langle \phi_{p-1,k}, \psi_{p-1,k'} \rangle = 0$$

$$\forall p \in \mathbb{N}^*, E_p = E_{p-1} \oplus F_{p-1}$$

Ainsi, on pourra écrire tout signal appartenant à un espace E_p comme sommes d'ondelettes filles et de sa valeur moyenne, qui appartient à E_0

Application au traitement du signal

- a) Calcul des coefficients d'ondelettes d'un signal, Transformée Discrète

$$E_p = E_{p-1} \oplus F_{p-1} = \dots = E_0 \oplus F_0 \oplus \dots \oplus F_{p-1}$$

$$s = \langle s, \phi \rangle \phi + \sum_{j=0}^{p-1} \sum_{k=0}^{2^j-1} \frac{\langle s, \psi_{j,k} \rangle}{\|\psi_{j,k}\|^2} \psi_{j,k}$$

Le transformée en ondelette d'un signal est la donnée de sa valeur moyenne et de sa décomposition dans la base d'ondelettes filles des F_j .

$$\forall j \in \llbracket 0, p-1 \rrbracket, s_{(2^j)} := \sum_{k=0}^{2^j-1} s_{(2^j),k} \phi_{j,k} = \sum_{k=0}^{2^j-1} \frac{\langle s_{(2^{j+1})}, \phi_{j,k} \rangle}{\|\phi_{j,k}\|^2} \phi_{j,k}$$

$s_{(2^j)}$ est la projection orthogonale de $s_{(2^{j+1})}$ sur E_j , $s_{(2^{j+1})} - s_{(2^j)}$ celle sur F_j .
À chaque étape, on récupère les coordonnées (coefficients) de $s_{(2^{j+1})}$ dans F_j .

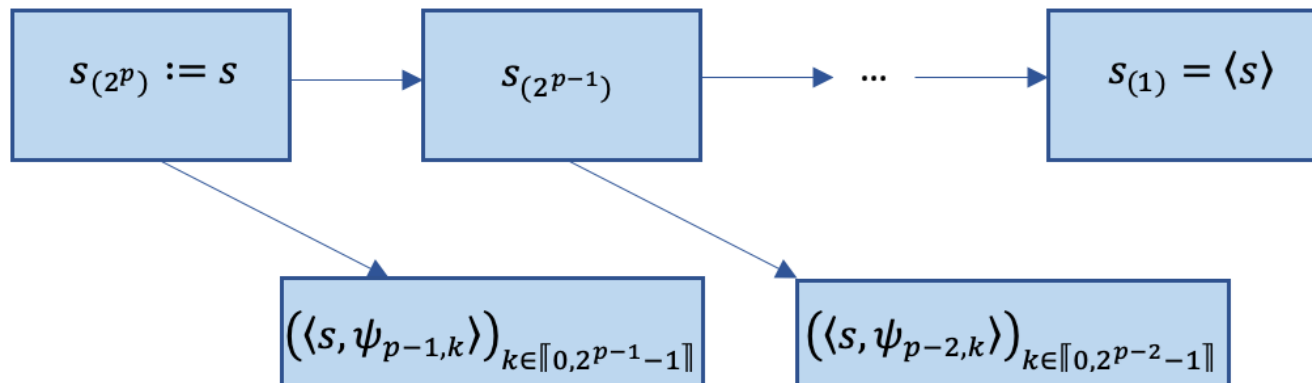
Application au traitement du signal

b) Transformée discrète en ondelette : stratégie de programmation dynamique

$$\forall j \in \llbracket 0, p-1 \rrbracket, \forall k \in \llbracket 1, 2^j - 1 \rrbracket, s_{(2^j),k} = \frac{\langle s_{(2^{j+1})}, \phi_{j,k} \rangle}{\|\phi_{j,k}\|^2} = \frac{s_{(2^{j+1}),2k} + s_{(2^{j+1}),2k-1}}{2}$$

$$\forall j \in \llbracket 0, p-1 \rrbracket, \forall k \in \llbracket 1, 2^j - 1 \rrbracket, \frac{\langle s, \psi_{j,k} \rangle}{\|\psi_{j,k}\|^2} = \frac{\langle s_{(2^{j+1})}, \psi_{j,k} \rangle}{\|\psi_{j,k}\|^2} = \frac{s_{(2^{j+1}),2k} - s_{(2^{j+1}),2k-1}}{2}$$

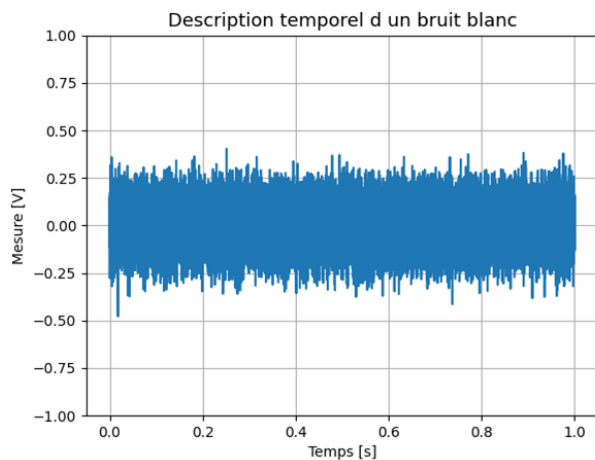
L'utilisation des ondelettes de Haar simplifie considérablement les calculs : plus besoin de calculer des produits scalaires, en connaissance des projections intermédiaires



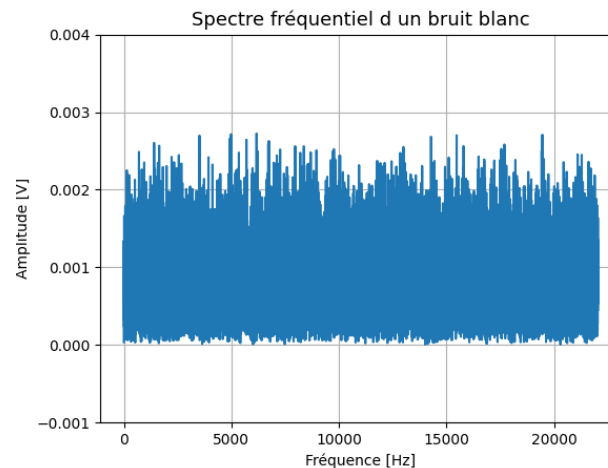
Stratégie schématique mise en place

Lien entre les coefficients d'ondelettes et le bruit

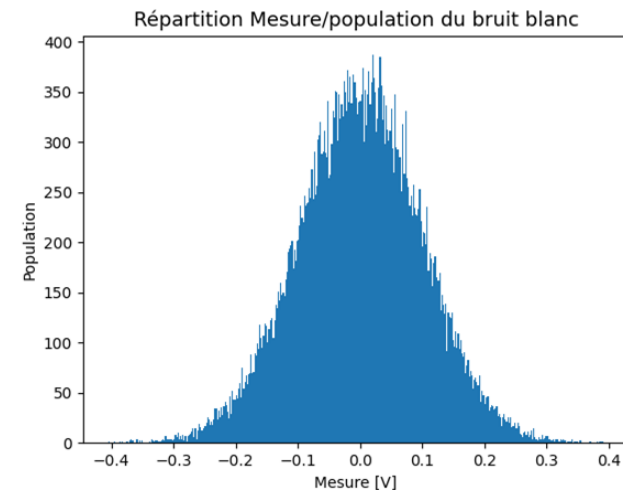
a) Une approche probabiliste : le bruit blanc gaussien



Moyenne nulle



Puissance uniforme sur toute la bande de fréquence



Répartition gaussienne des mesures

$$\forall \omega \in \Omega, W(\omega) : t \mapsto \sigma \sqrt{-\ln(U_{1,t}(\omega))} \cos(2\pi U_{2,t}(\omega))$$

*U1 et U2 sont des séquences de variables aléatoires suivant une loi uniforme sur [0,1]
Il s'agit de la formule de Rice, pour un bruit blanc gaussien*

Lien entre les coefficients d'ondelettes et le bruit

b) Méthode de seuillage de Donoho et Johnstone

$$T = \sigma\sqrt{2\log(2^p)}$$

$$\forall j \in \llbracket 0, p-1 \rrbracket, \forall k \in \llbracket 0, 2^j-1 \rrbracket, \theta_{j,k} = \begin{cases} 1 & \text{si } |\langle s, \psi_{j,k} \rangle| > T \\ 0 & \text{si } |\langle s, \psi_{j,k} \rangle| \leq T \end{cases}$$

Le seuil T dépend de la valeur de l'écart type du bruit blanc gaussien et de la taille du signal



Lien entre les coefficients d'ondelettes et le bruit

c) Calcul de l'écart-type : approximation de Mallat.

$$s = f + W$$

Sous l'hypothèse que f est un signal avec peu d'irrégularités, la donnée des coefficients d'ondelettes de s à l'échelle la plus fine donne des informations sur le bruit W :

$$\forall k \in \llbracket 0, 2^{p-1} - 1 \rrbracket, \langle f, \psi_{p-1,k} \rangle \ll \langle W, \psi_{p-1,k} \rangle$$

$$\forall k \in \llbracket 0, 2^{p-1} - 1 \rrbracket, \langle s, \psi_{p-1,k} \rangle \simeq \langle W, \psi_{p-1,k} \rangle$$

Or, si M est la médiane de la valeur absolue d'un bruit blanc à N variables aléatoires gaussiennes indépendantes de variance σ_0^2 , on peut montrer que :

$$\mathbb{E}(M) \simeq 0.6745\sigma_0$$

D'où :

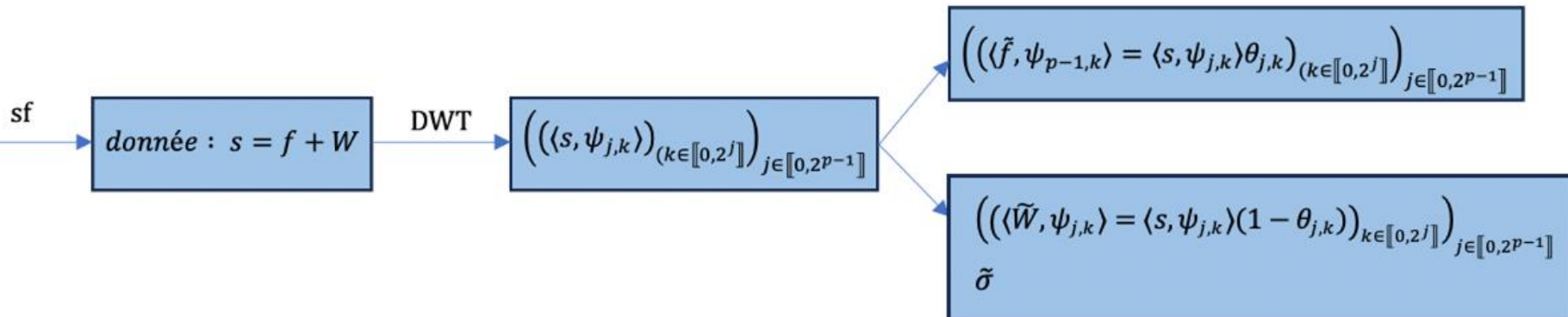
$$\sigma \simeq \frac{\text{Med} \left(|\langle s, \psi_{p-1,k} \rangle|_{k \in \llbracket 0, 2^{p-1} - 1 \rrbracket} \right)}{0.6745}$$

Approche numérique

a) Mise en place

Détection du bruit pour un enregistrement microphonique (bruit de tronçonneuse)

Nom du fichier	Format	Durée (s)	Nombre d'échantillons par seconde
Tronçonneuse	wav (mono)	09	48000



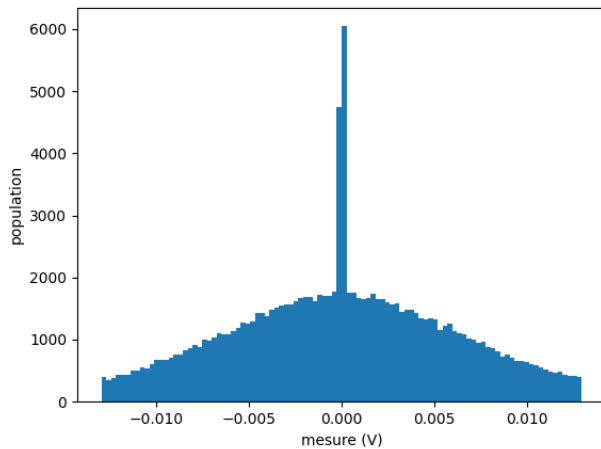
Stratégie schématique mise en place

Approche numérique

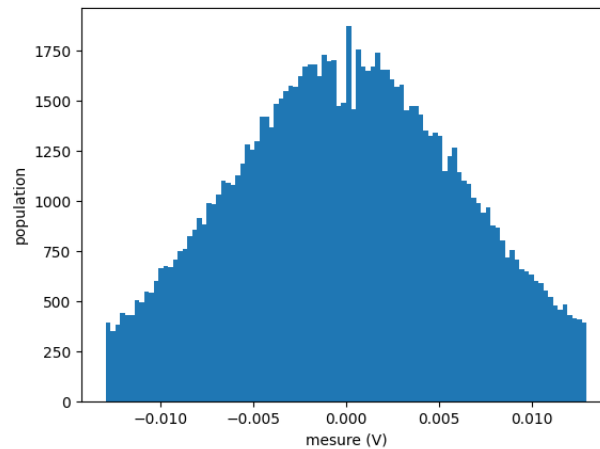
b) Résultats : test de normalité du bruit mesuré

	Piste mono	Référence (bruit blanc)
Moyenne	-0.0046	0.0039 ≈ 0

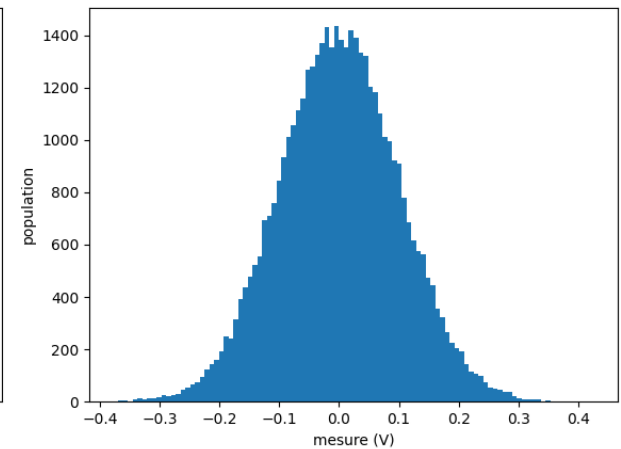
Dans les mêmes conditions de mesures (dilatation des valeurs pour faire coïncider les amplitudes maximales de chacun des signaux). le bruit mesuré a bien une moyenne approximativement nulle.



Cas pathologique : test faussé par une surpopulation anormale au voisinage de 0



Résultat après élimination partielle des valeurs presque nulles

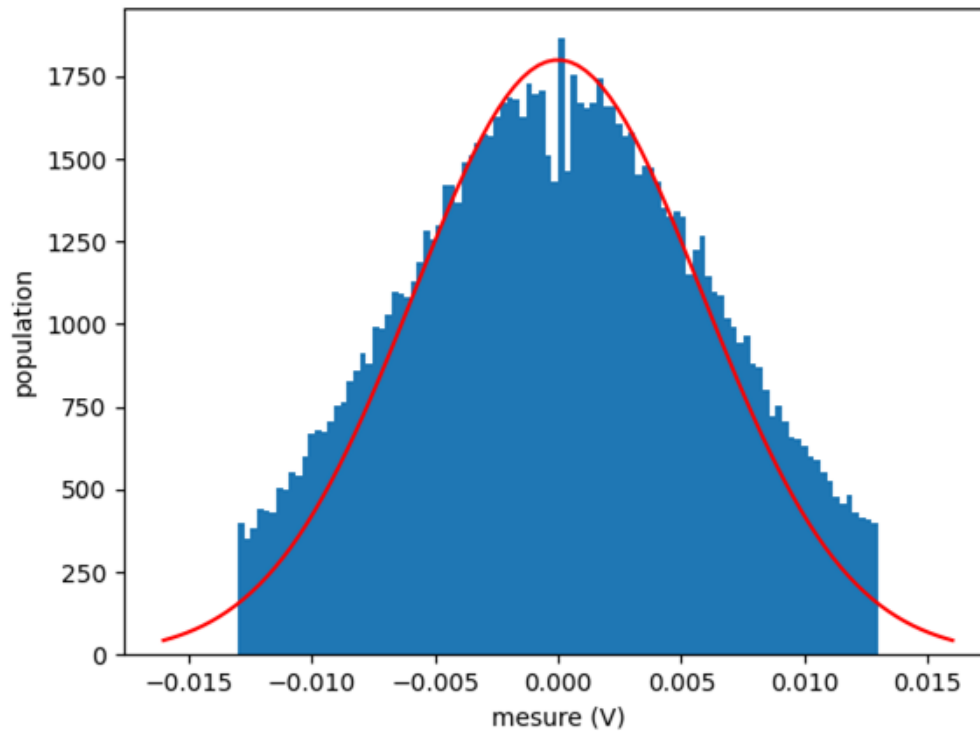


Histogramme de répartition des valeurs du bruit blanc (référence) généré numériquement

Approche numérique

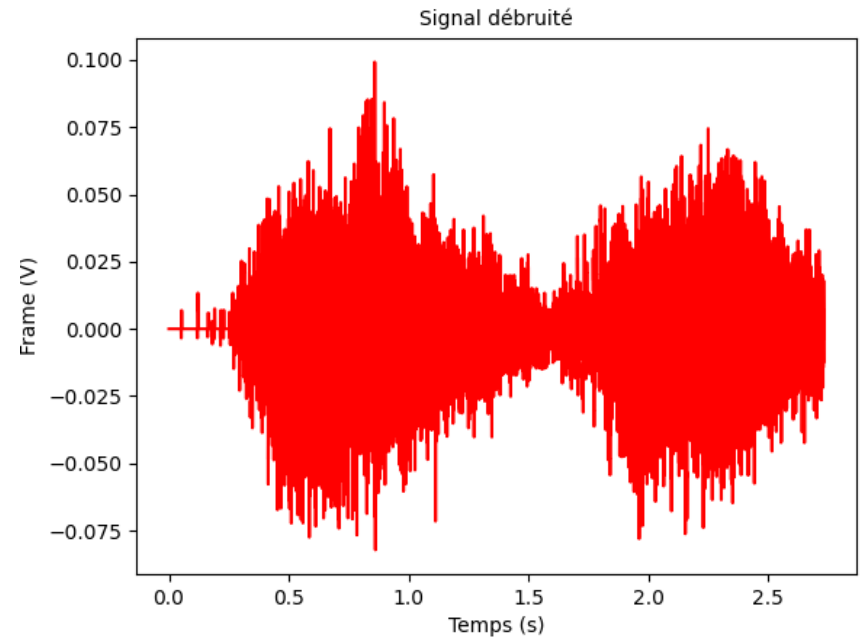
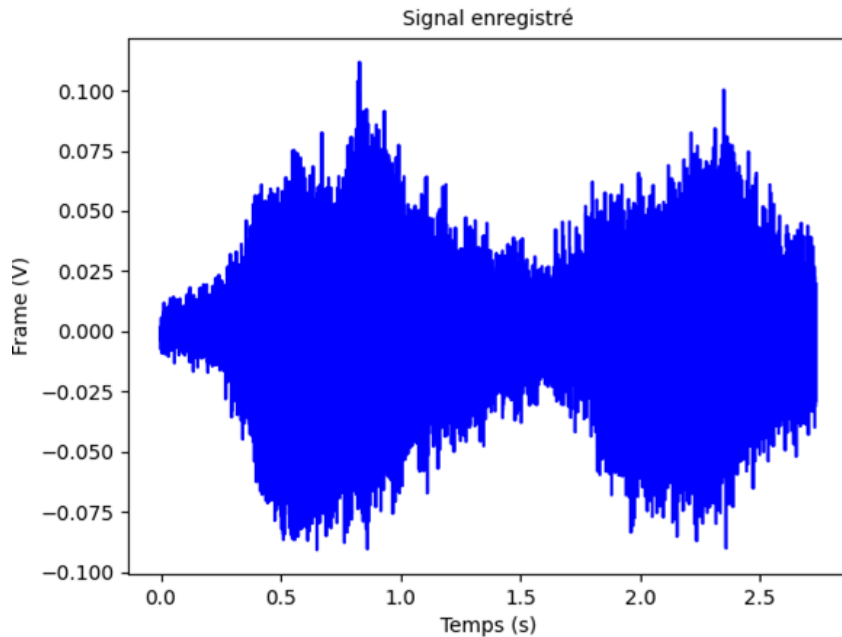
b) Résultats

Écart type calculé (approximation de Mallat) : $\sigma = 0.008279787689955524$



Superposition de la courbe population/mesure avec la Gaussienne théorique d'écart type σ

Conclusion et perspectives



Allure de l'enregistrement du son d'une tronçonneuse, avant et après le débruitage

$$s = f + W$$

$$\forall j \in \llbracket 0, p - 1 \rrbracket, \forall k \in \llbracket 0, 2^j - 1 \rrbracket, \quad \langle s, \psi_{j,k} \rangle \approx \langle f, \psi_{j,k} \rangle \text{ ou } \langle s, \psi_{j,k} \rangle \approx \langle W, \psi_{j,k} \rangle$$

Situation idéale : les coefficients d'ondelettes du signal bruité s ne sont pas simultanément défini par le bruit W et le signal authentique f

Questions et précisions

Annexes

1 - Compléments sur les ondelettes

1.1 - Justification du choix pour le seuil T (Donoho et Johnstone)

1.2 - Une ondelette continue : l'ondelette de Morlet

2 - Éléments de programmation, représentations graphiques

2.1 - Représentations graphiques de l'ondelette mère de Haar

2.2 - Représentations graphiques de l'ondelette de Morlet

2.3 - Histogramme du bruit [*matplotlib.pyplot*] et superposition avec la gaussienne théorique

3 - Éléments de programmation, implémentation des algorithmes présentés.

3.1 - Conversion d'un fichier wav (signal stéréo uniquement) en deux tableaux numpy [*soundfile*]

3.2 - Transformée discrète en ondelette (DWT) de Haar

3.3 - Transformée inverse en ondelette (IDWT) de Haar

3.4 - Calcul de l'écart-type du bruit blanc (Mallat)

3.5 - Algorithme de seuillage de Donoho et Johnstone

3.6 - Obtention des coefficients d'ondelettes du bruit W à l'échelle la plus fine

Annexe 1.1

Estimateur par seuillage de Donoho-Johnstone

Afin d'obtenir un estimateur de λ implémentable, Donoho et Johnstone ont suggéré un estimateur par seuillage fort où $\theta_{j,k}$ prend les valeurs suivantes :

$$\theta_{j,k} = \begin{cases} 1 & \text{si } |\langle X, \psi_{j,k} \rangle| > T \\ 0 & \text{si } |\langle X, \psi_{j,k} \rangle| \leq T \end{cases} \quad (17)$$

où le seuil T est égal à $\sqrt{2\log(n)} \sigma$.

Cette dernière valeur a été choisie pour T sur base de l'argument suivant. Sous l'hypothèse nulle $H_0 : \lambda = 0$, l'équation 12 devient $X = W$, qui implique à son tour que $\langle W, \psi_{j,k} \rangle = \langle X, \psi_{j,k} \rangle$. Afin d'obtenir $\lambda = 0$, nous devons dès lors avoir $T > |\langle W, \psi_{j,k} \rangle|$. Cependant, sous l'hypothèse alternative $H_a : \lambda \neq 0$, nous devons être prudents afin d'éviter des valeurs de seuil trop grandes qui mettraient tous les coefficients à zéro, spécialement dans les cas où $\langle X, \psi_{j,k} \rangle \neq \langle W, \psi_{j,k} \rangle$. Donoho et Johnstone (DONOHO et JOHNSTONE, 1994) ont suggéré une valeur de seuil ayant une grande probabilité d'être juste au-dessus de la valeur maximum prise par $|\langle W, \psi_{j,k} \rangle|$. Choisir $T = \sqrt{2\log(n)} \sigma$ permet de satisfaire cette contrainte puisque :

$$\lim_{n \rightarrow +\infty} P\left(T - \frac{\sigma \log(\log(n))}{\log(n)} \leq \max_{j,k} |\langle W, \psi_{j,k} \rangle| \leq T\right) = 1, \quad (18)$$

et

$$\lim_{n \rightarrow \infty} \frac{\log(\log(n))}{\log(n)} = 0. \quad (19)$$

Catherine Charles : Introduction aux applications des ondelettes : Unité de Statistique, Informatique et Mathématiques Appliquées (SIMa), Gembloux, Belgique. (2011)

Preuve (18) : Donoho D.L., Johnstone I.M. [1994]. Ideal spatial adaptation via wavelet shrinkage. Biometrika, 81 :425-455.

Annexe 1.2

$$\varphi: x \mapsto e^{-\frac{x^2}{2}} \cos(5x)$$

L'exemple de l'ondelette de Morlet, une ondelette continue.

$$\int_{-\infty}^{+\infty} \varphi(x) dx = 0$$

Comme toutes les ondelettes, il s'agit d'une fonction de moyenne nulle.



$$\exists M > 0, \text{supp}(\varphi) := \{x \in \mathbb{R} \mid \varphi(x) \neq 0\} \subset]-M, M[$$

En particulier, on parle d'ondelette discrète lorsque le support de celle-ci est fini. Ce n'est pas le cas de l'ondelette de Morlet.

Annexe 2.1

```
def Haar(p,k):
    debut = k-1
    lim0 = k * 2**p
    lim1 = (2 ** (-p-1) + lim0)
    lim2 = (2 ** (-p-1) + lim1)
    fin = 2+k
    nbe_sequences = (fin -debut)*44100
    #fs = 44100
    Temps = np.linspace(debut, fin, nbe_sequences)
    Valeurs_prises = np.zeros(nbe_sequences)
    for T in range(nbe_sequences):
        t = Temps[T]
        if t >= lim0 and t <= lim1:
            Valeurs_prises[T] = 1
        if t > lim1 and t <= lim2:
            Valeurs_prises[T] = -1
    plt.plot(Temps, Valeurs_prises, color="blue")
    plt.xlabel('temps (s)')
    plt.ylabel('frame')
    plt.ylim([-1.125, 1.125])
    plt.title('Ondelette de Haar mère')
    plt.grid()
    plt.show()
```

Annexe 2.2

```
def Morlet ():
    fin = 30
    nbe_sequences = fin * 44100
    #fs = 44100
    Temps = np.linspace(-30, fin, nbe_sequences)
    Valeurs_prises = np.zeros(nbe_sequences)
    for t in range(nbe_sequences):
        T = Temps[t] / 2**3
        Valeurs_prises[t] = np.cos(5*T) * np.exp(- T**2 /
2)
    plt.plot(Temps, Valeurs_prises, color="blue")
    plt.xlabel('x')
    plt.ylabel('y')
    plt.title('Ondelette de Morlet')
    plt.grid()
    plt.show()
```

Annexe 2.3

```
def white_noise (s):
    Coefficients = coeffHaar(s)
    T, ecart_type = Mallat(s, Coefficients[-1])
    p = np.shape(Coefficients)[0]
    wn = []
    # On calcule alors le seuil proposé par Donoho et Johnstone
    for j in range(p):
        for k in range(2 ** j):
            if abs(Coefficients[j][k]) <= T:
                wn.append(Coefficients[j][k])
    return wn, ecart_type

import soundfile as sf
filename1 = '/Volumes/MARTIN/TIPE principal/tipe/TIPE/Benny.wav'
data1, fs = sf.read(filename1, dtype='float32')

chan1, _ = data_to_function(data1, 1)
(wn1, ec1) = white_noise(chan1)
```

Annexe 3.1

```
import soundfile as sf
filename = '/Volumes/MARTIN/TIPE/tronçonneuse.wav'
data, fs = sf.read(filename, dtype='float32')

def data_to_function(sound_data):
    N = np.shape(sound_data)[0]
    chan1 = np.zeros(N)
    chan2 = np.zeros(N)
    for k in range(N):
        chan1[k] = sound_data[k][0]
        chan2[k] = sound_data[k][1]
    return chan1, chan2

#Mise en route
chan1, chan2 = data_to_function(data)
```

*Le signal stéréo possède deux pistes (channel 1 et channel 2). Soundfile fournit, à partir de l'emplacement du fichier dans le stockage, un tableau de couples correspondant aux valeurs prises par les pistes 1 et 2 et une fréquence d'échantillonnage fs : la n -ième valeur du tableau `data` correspond aux valeurs des pistes 1 et 2 à l'instant $t = n*fs$.*

Annexe 3.2

```
def dwtHaar (s):
    p = int(math.log2(len(s)))
    Coefficients = [[] for _ in range(p)]
    si = [[] for _ in range(p+1)]
    si[p] = s
    for j in range(p-1, -1, -1):
        for k in range(2**j):
            si[j].append((si[j+1][2*k] + si[j+1][2*k+1]) / 2)
            Coefficients[j].append((si[j+1][2*k] - si[j+1][2*k+1]) / 2)
    return Coefficients
```

Annexe 3.3

```
def idwtHaar (coeffs) :
    p = len(coeffs)
    N = 2**len(coeffs)
    s = [[] for _ in range (p)]
    s[0] = coeffs[0]
    for i in range(1, p):
        for k in range(len(coeffs[i])):
            if k//2 < len(coeffs[i]) :
                if k % 2 == 0 :
                    s[i].append(s[i - 1][k // 2] + coeffs[i - 1][k // 2] * 2)
                else:
                    s[i].append(s[i - 1][k // 2] - coeffs[i - 1][k // 2] * 2)
    return s[-1]
```

Annexe 3.4

```
def Mallat(s, coeffs):
    n = len(coeffs)
    p = int(math.log2(len(s)))
    abs_coefs_triees = np.zeros(n)
    for k in range(n):
        abs_coefs_triees[k] = abs(coeffs[k])
    np.sort(abs_coefs_triees)
    # tri rapide
    if n % 2 == 0:
        M = (abs_coefs_triees[n//2] + abs_coefs_triees[n//2 + 1]) / 2
        ecart_type = Mx/0.6745
    else:
        M = abs_coefs_triees[n//2 + 1]
        ecart_type = M/0.6745
    # calcul de la médiane de abs_coefs_triees
    T = math.sqrt(2*math.log10(2**p))*ecart_type
    return T, ecart_type
```

Annexe 3.5

```
def DonohoJohnstone (coeffs, T):
    # Ici, coeffs correspond uniquement au coefficients de l'échelle la plus
    fine
    thresholded = np.copy(coeffs)
    # deep copy
    p = np.shape(thresholded)[0]
    for j in range(p):
        for k in range(2**j):
            if abs(thresholded[j][k]) <= T:
                thresholded[j][k] = 0
    return thresholded

def denoising (s):
    Coefficients = coeffHaar(s)
    T = Mallat(s, Coefficients[-1])[0]
    thresholded = DonohoJohnstone(Coefficients, T)
    return thresholded
```

Annexe 3.6

```
def Gauss(sigma):
    U1 = random.random()
    U2 = random.random()
    return sigma*np.sqrt(-2*np.log(U1))*np.cos(2*np.pi*U2)

G = np.zeros(fs)
# fs = 48000 ici
for t in range(fs):
    G[t] = Gauss(0.1)

def gaussienne (x):
    return 1800*np.exp(-(x/ec1)**2)

T = np.linspace(-0.016,0.016, 100)
plt.hist(wnl, bins = 100)
plt.plot(T, gaussienne(T), color = 'r')
plt.xlabel('mesure (V)')
plt.ylabel('population')
plt.show()
```