

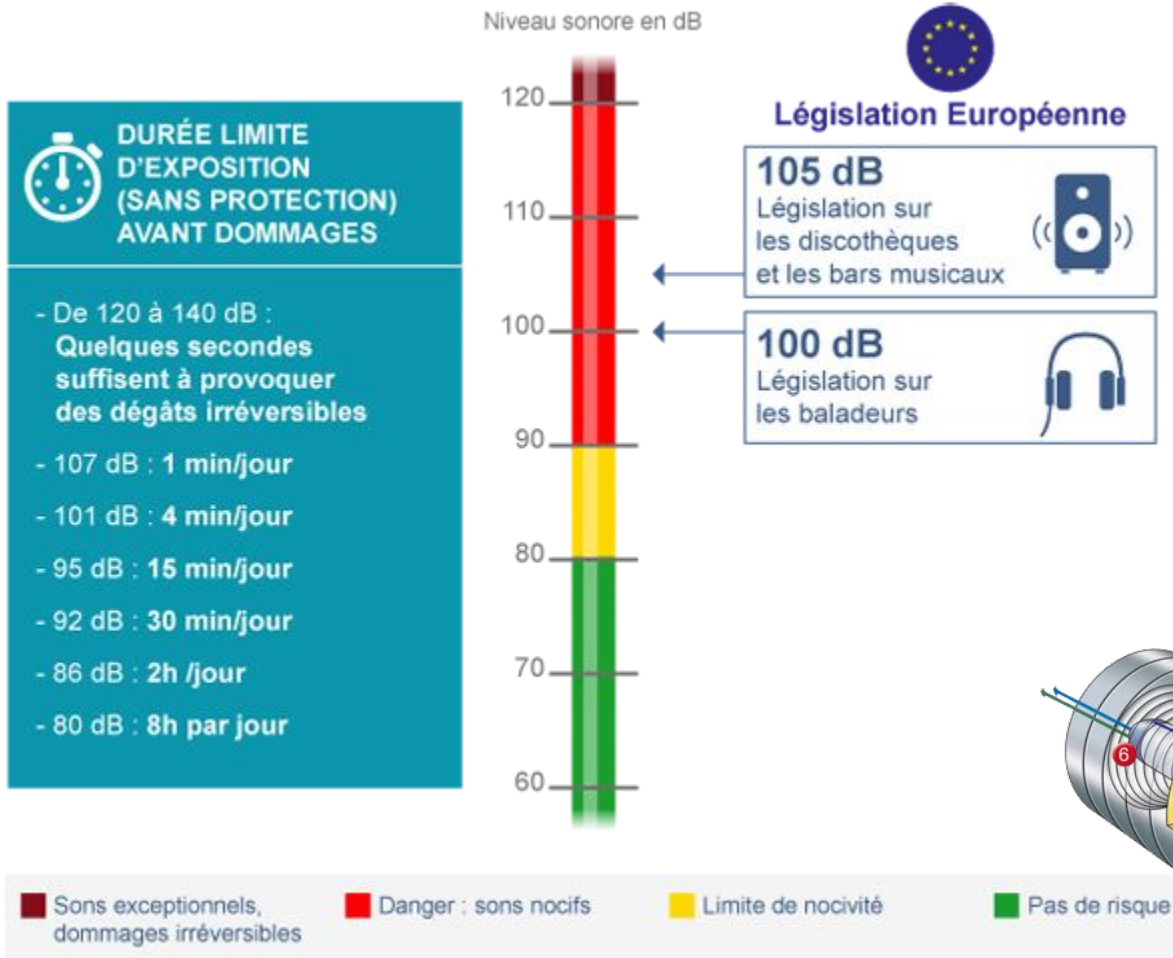


Le contrôle actif du bruit

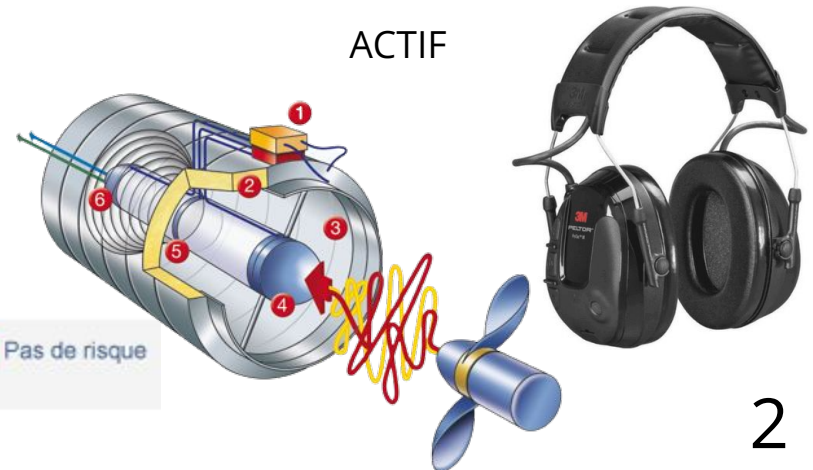


Contexte

maximum engin sur chantier : **90 dB**
(= puissance sonore)



ACTIF



Plan

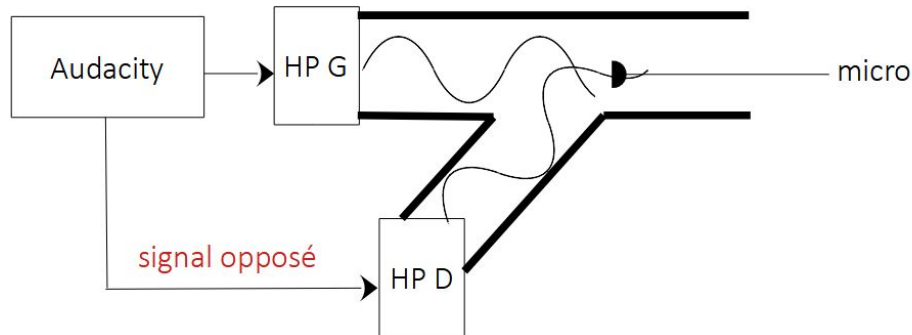
I/ Contrôle actif du bruit : principe, attentes et limites

II/ Paysage sonore et objectifs

III/ 3 processus, avantages et inconvénients

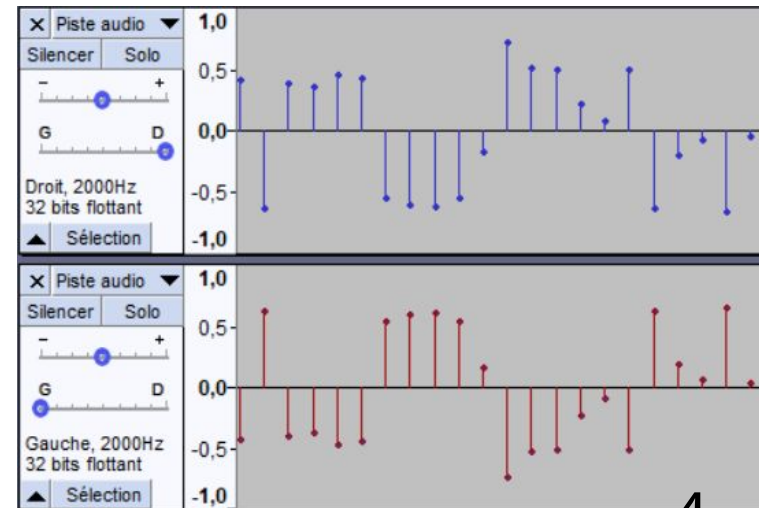
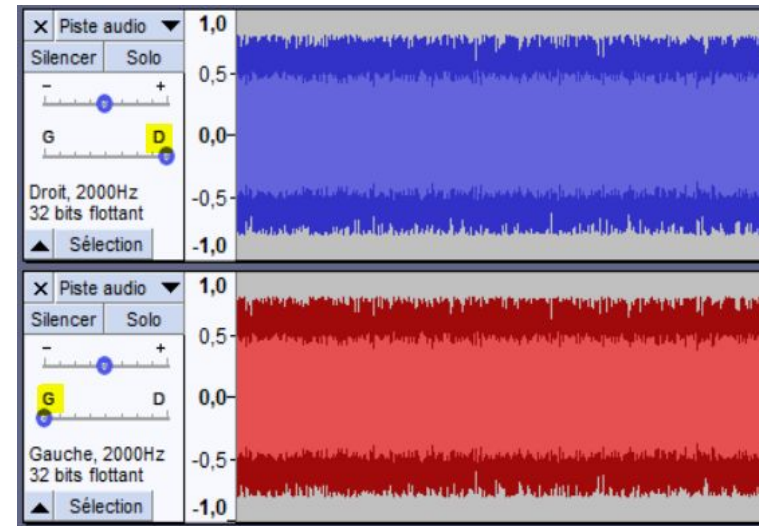
IV/ Conclusion et utilisation actuelle

Mise en évidence du principe

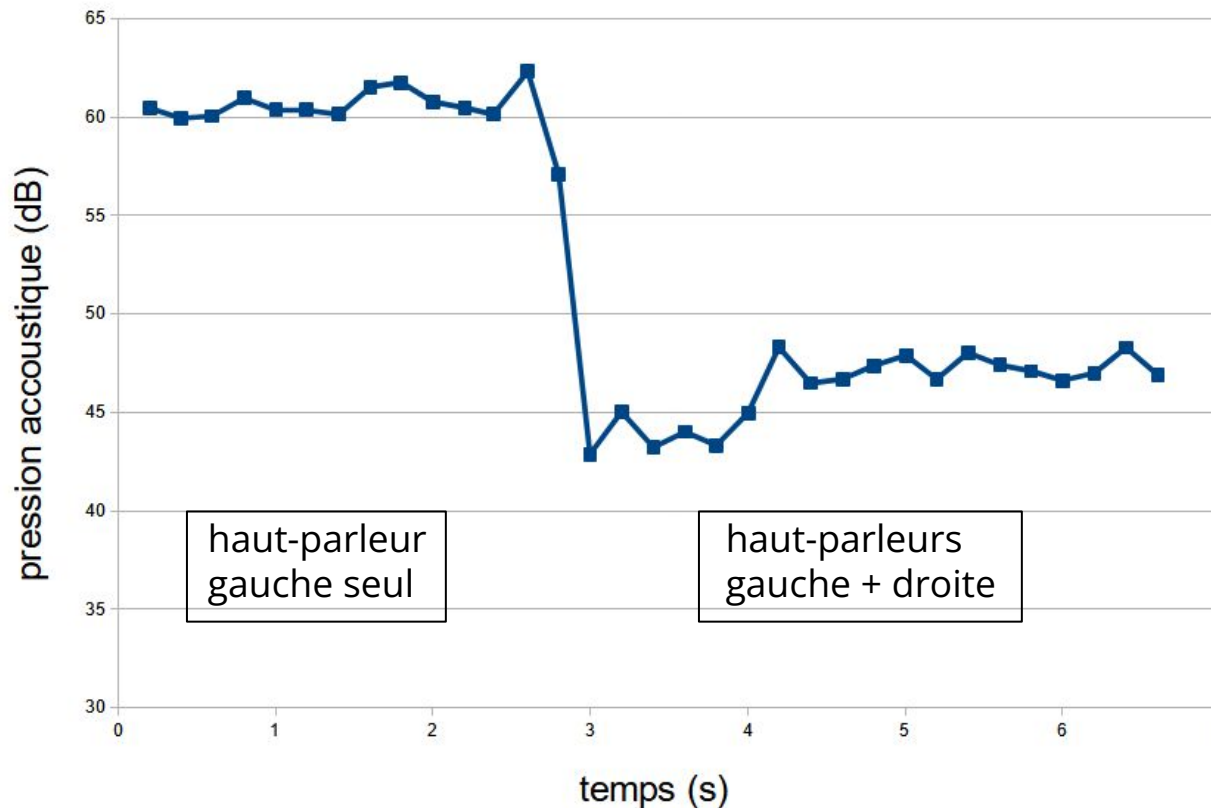


Atténuation en un point : angle d'incidence sans importance
Logiciels : Audacity et Phypox

Bruit blanc



Mise en évidence du principe



Remarques :

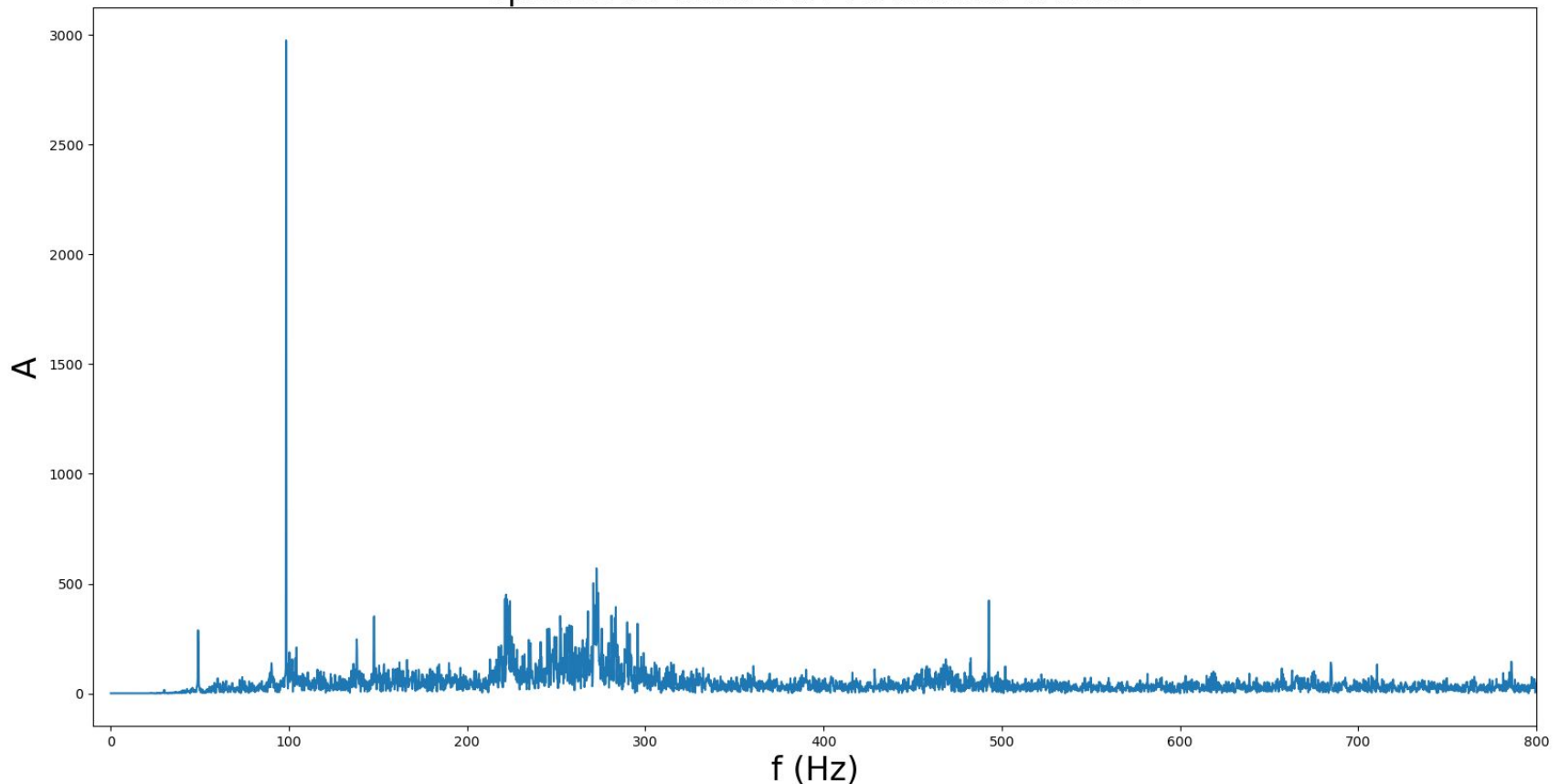
- atténuation totale impossible en réalité
- - 20dB au plus (grande atténuation: processus efficace)

Contraintes de l'application réelle:

- générer un contre-bruit exact (opposé d'un signal qu'on ne connaît pas à priori)
- fonctionner en **temps réel** (rapidité de l'ordre de la ms)

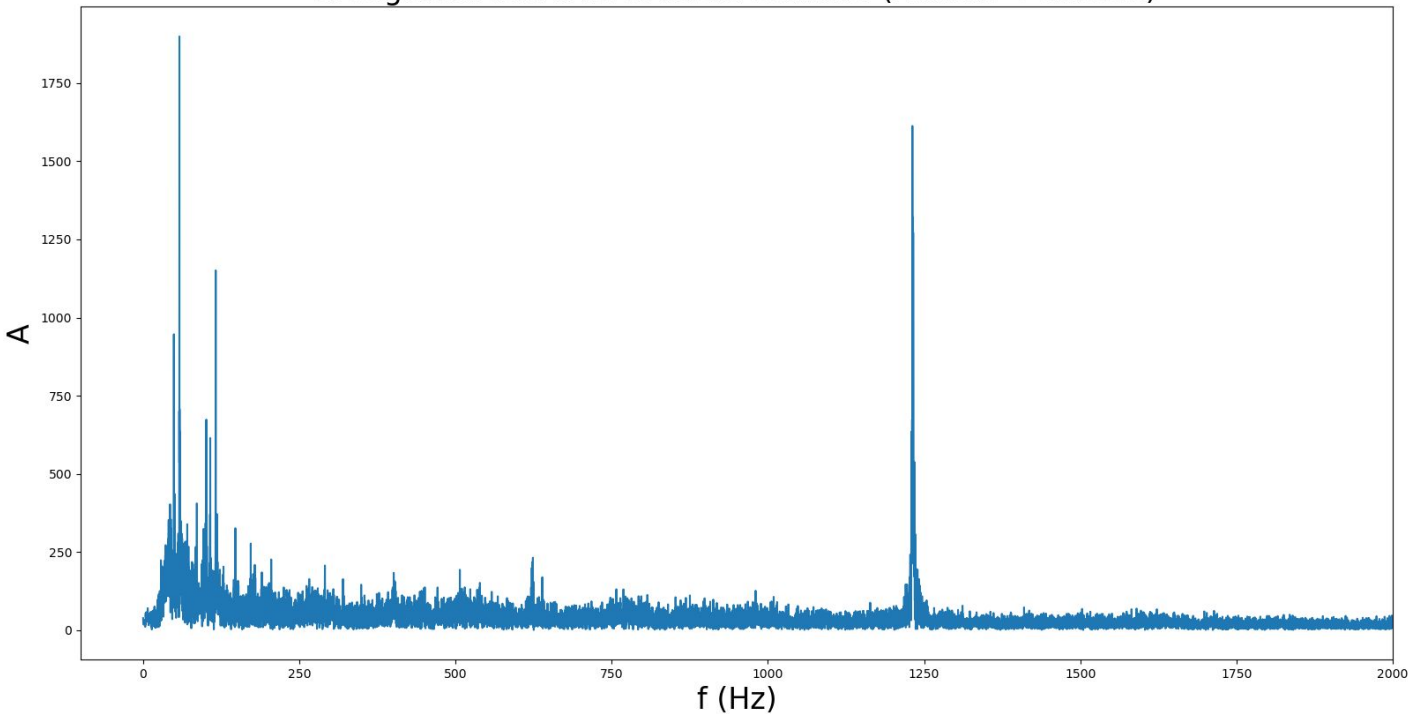
Identification des bruits indésirables

Spectre du bruit d'un ventilateur d'usine



Identification des bruits indésirables

Enregistrement d'un bruit de chantier (moteur + alarme)



Remarques :

- moteurs/ machines/ ventilateurs en basses fréquences (environ 100Hz)
- alarmes en plus hautes fréquences

Objectifs

- obtenir une atténuation de l'ordre de -20dB
- choisir le domaine de fréquences de l'atténuation (<250Hz)

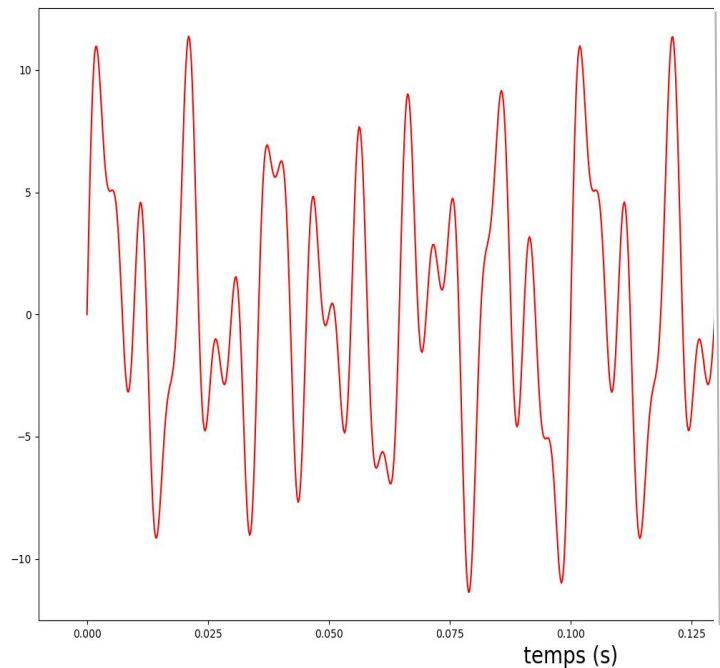
3 processus:

- analyse du son par transformée de Fourier
- circuit électrique
- filtre informatique

III/ 3 processus, avantages et inconvénients

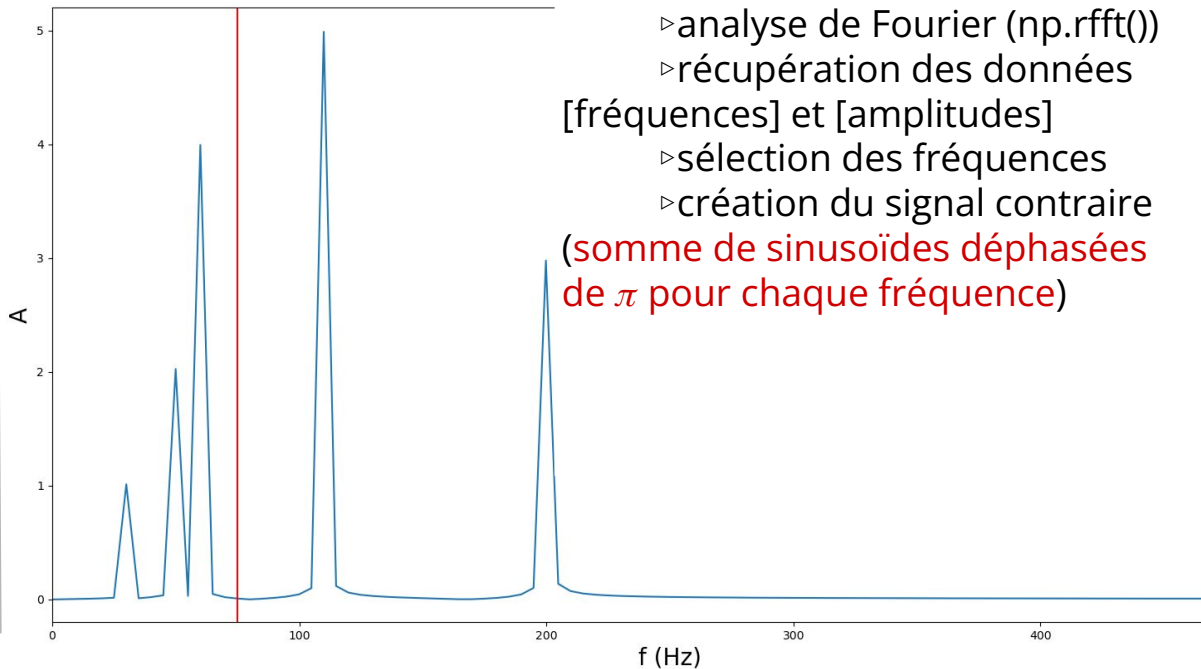
Processus 1 : analyse sonore (simulations informatiques)

↓ signal créé numériquement ↓



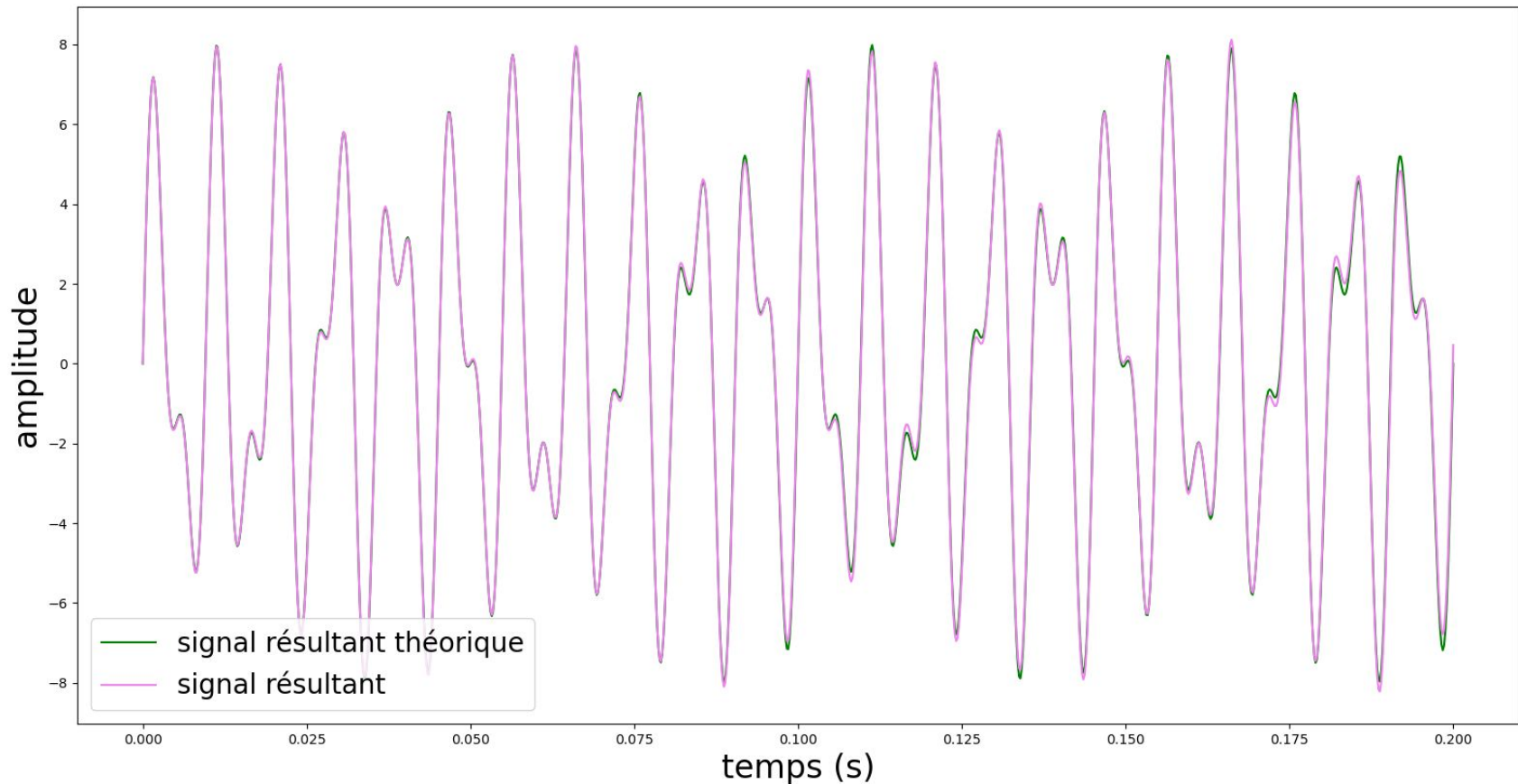
Programme informatique :

- ▷ analyse de Fourier (`np.rfft()`)
- ▷ récupération des données [fréquences] et [amplitudes]
- ▷ sélection des fréquences
- ▷ création du signal contraire (somme de sinusôides déphasées de π pour chaque fréquence)



III/ 3 processus, avantages et inconvénients

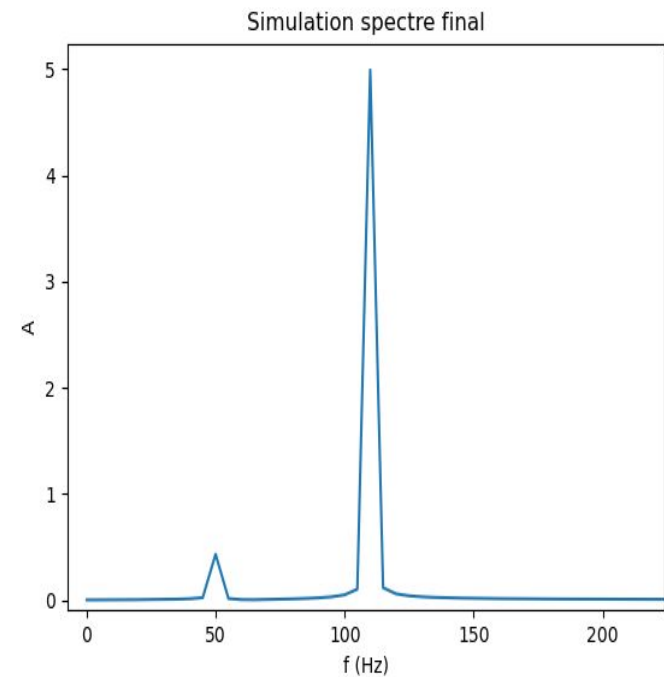
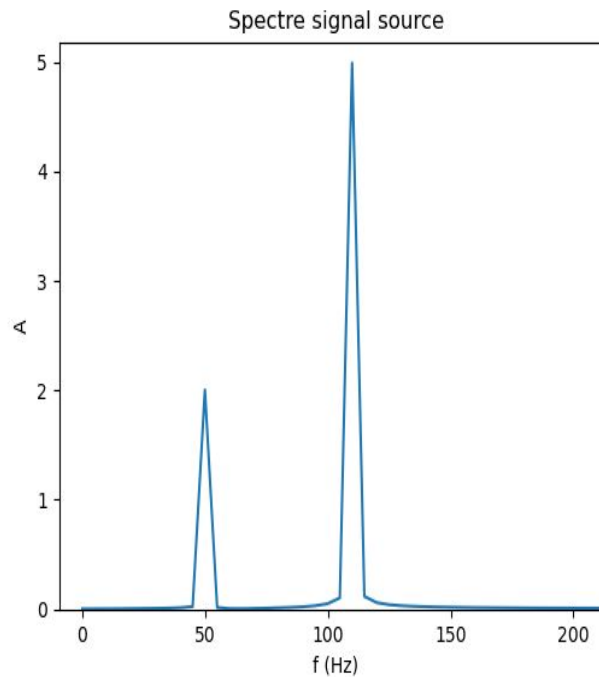
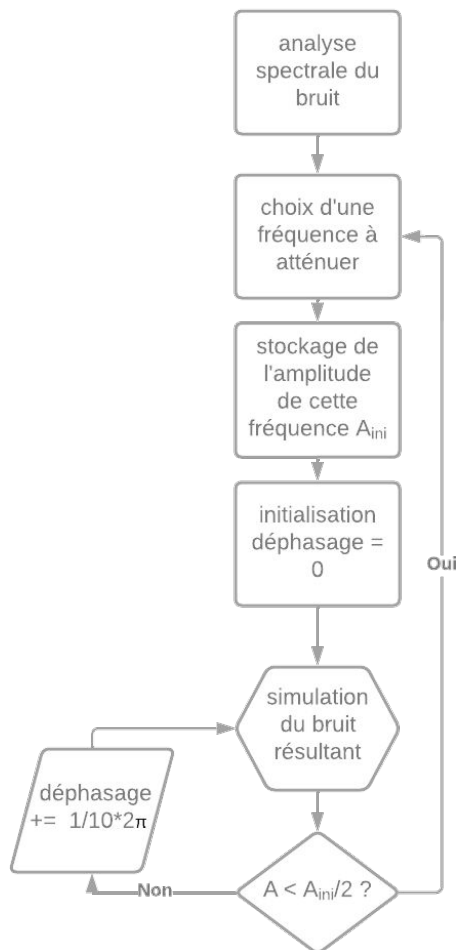
Processus 1 : analyse sonore (simulations informatiques)



III/ 3 processus, avantages et inconvénients

Processus 1 : déphasage

▷informatiquement : on modifie le déphasage (pas de $1/10 * 2\pi$) tant que l'amplitude est plus grande que la moitié de l'amplitude originelle



Inconvénient : complexité trop importante, **temps de calcul trop long** pour une utilisation en temps réel

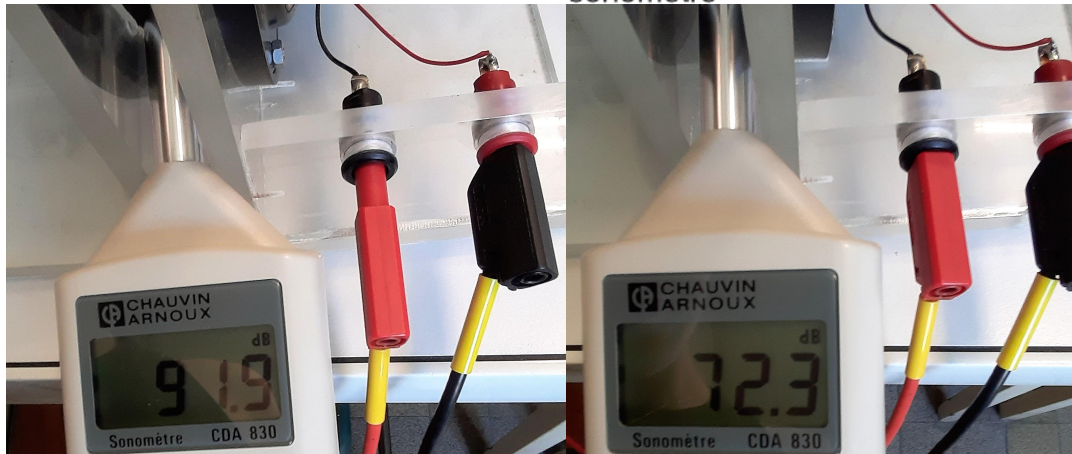
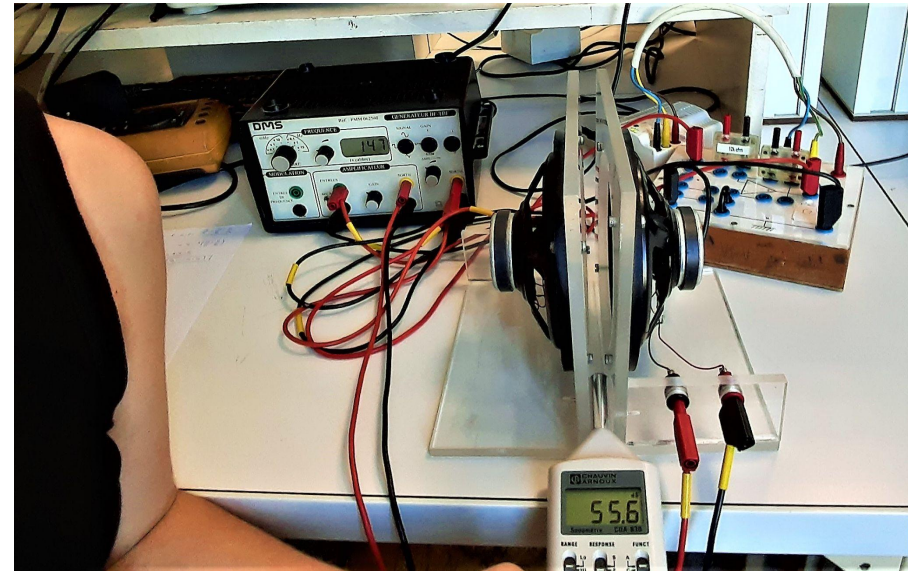
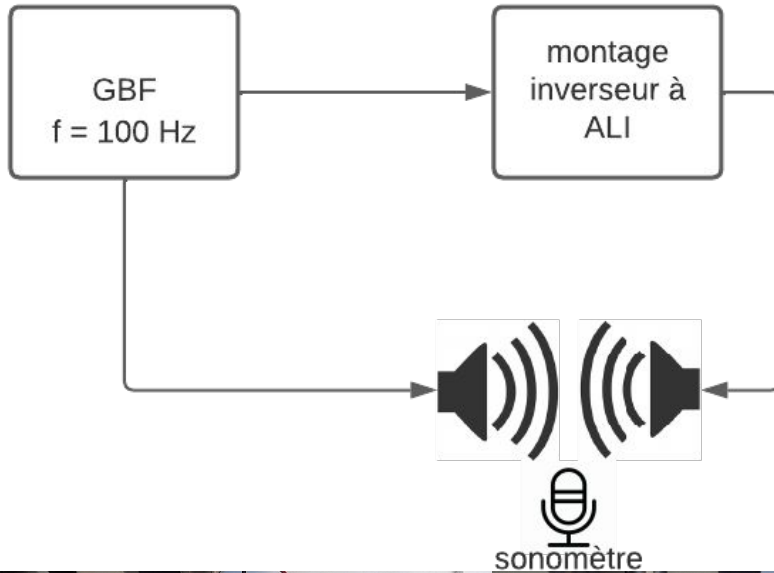
(essai sur bruit de chantier enregistré : plusieurs minutes pour le calcul total)

III/ 3 processus, avantages et inconvénients

Processus 2 : approche électrique

Remarques:

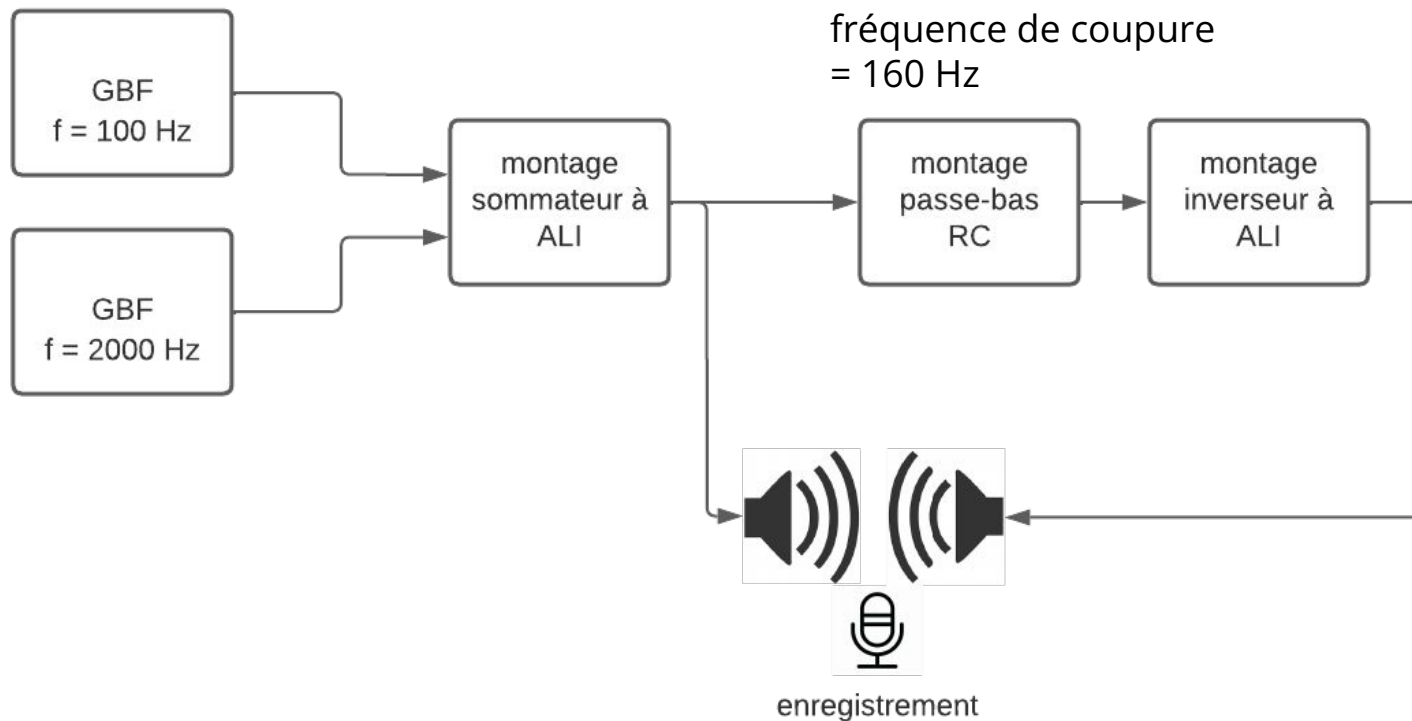
- choix de 100Hz: moteur
- $\lambda = c/f \approx 3\text{m}$



atténuation de -19.6dB

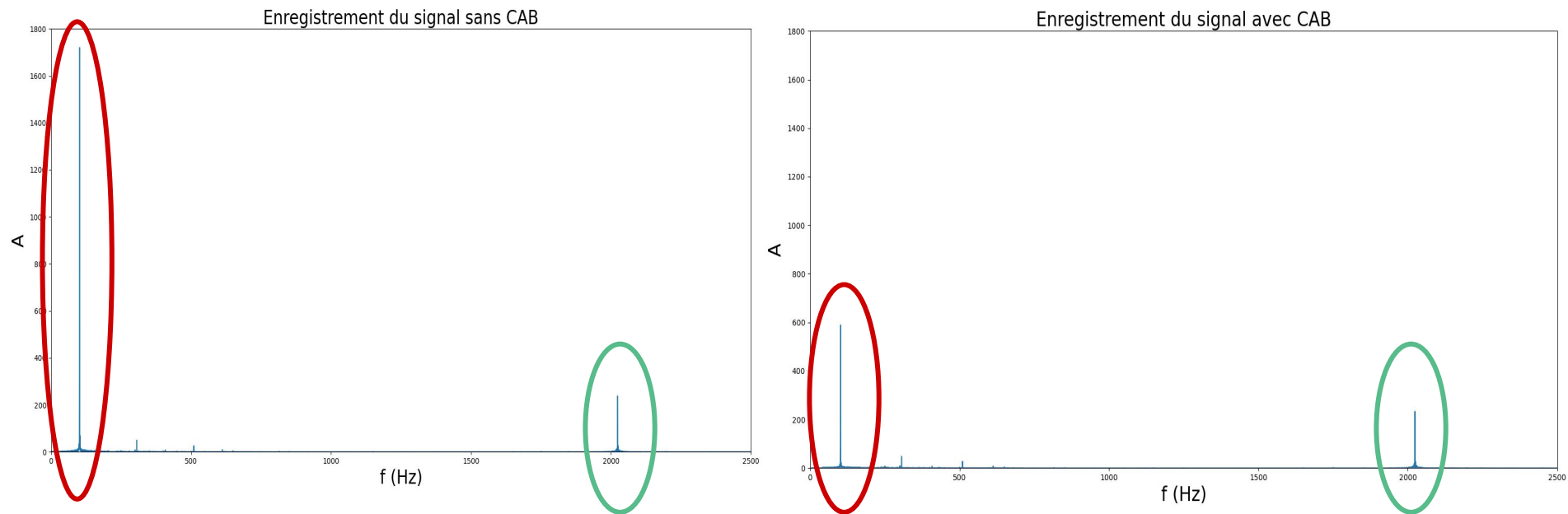


Processus 2 : approche électrique



III/ 3 processus, avantages et inconvénients

Processus 2 : approche électrique



III/ 3 processus, avantages et inconvénients

Processus 3 : filtres LMS (principe)

Formalisé en 1960 par Bernard Widrow, professeur de l'université de Stanford

Principe : **approcher $x(t)$ par $y(t)$**

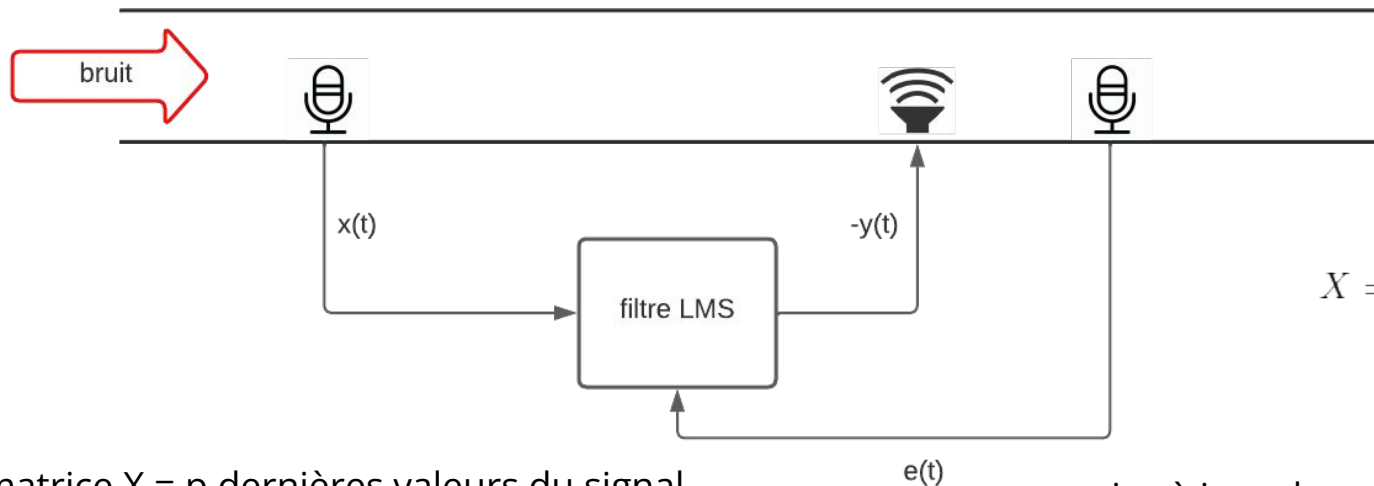
Légende :

$x(t)$ = signal d'entrée

$e(t)$ = erreur résultante (on veut $e(t) = 0$)

p = ordre du filtre

μ = pas



$$X = \begin{pmatrix} x[k] \\ \vdots \\ x[k - (p - 1)] \end{pmatrix}$$

matrice X = p dernières valeurs du signal d'origine

mise à jour des coefficients à chaque discrétisation de $x(t)$ selon :

$$W = W + \mu * e[k] * X$$

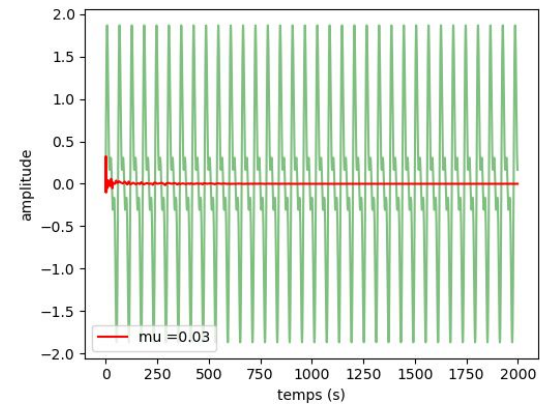
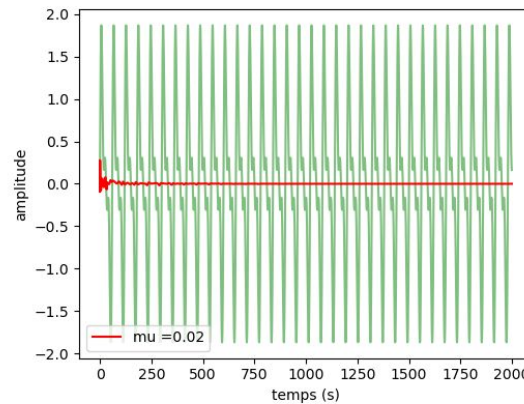
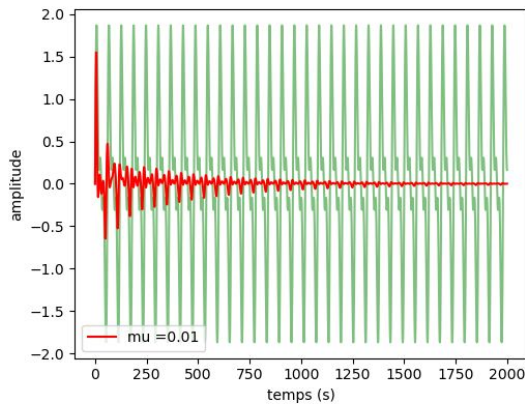
matrice W = matrice des coefficients de l'équation pondérée donnant la solution $y(t)$

$$y[k] = W[0] * x[k] + \dots + W[p - 1] * x[k - (p - 1)]$$

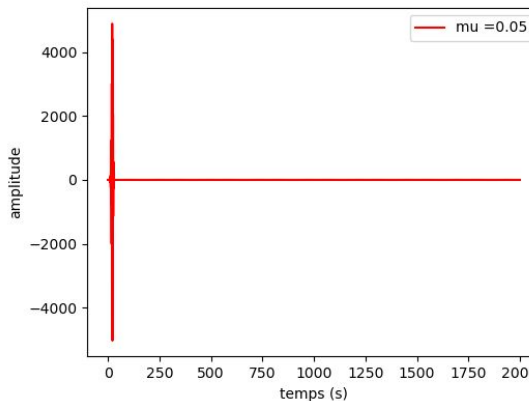
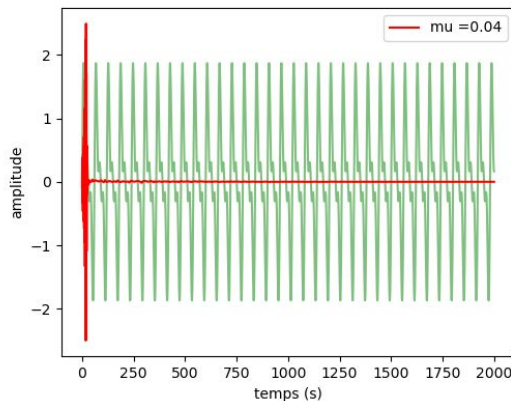
III/ 3 processus, avantages et inconvénients

Processus 3 : filtres LMS (compromis sur le coefficient μ)

Variation du pas μ entre 0.01 et 0.05 (signal originel en vert, bruit résiduel en rouge)

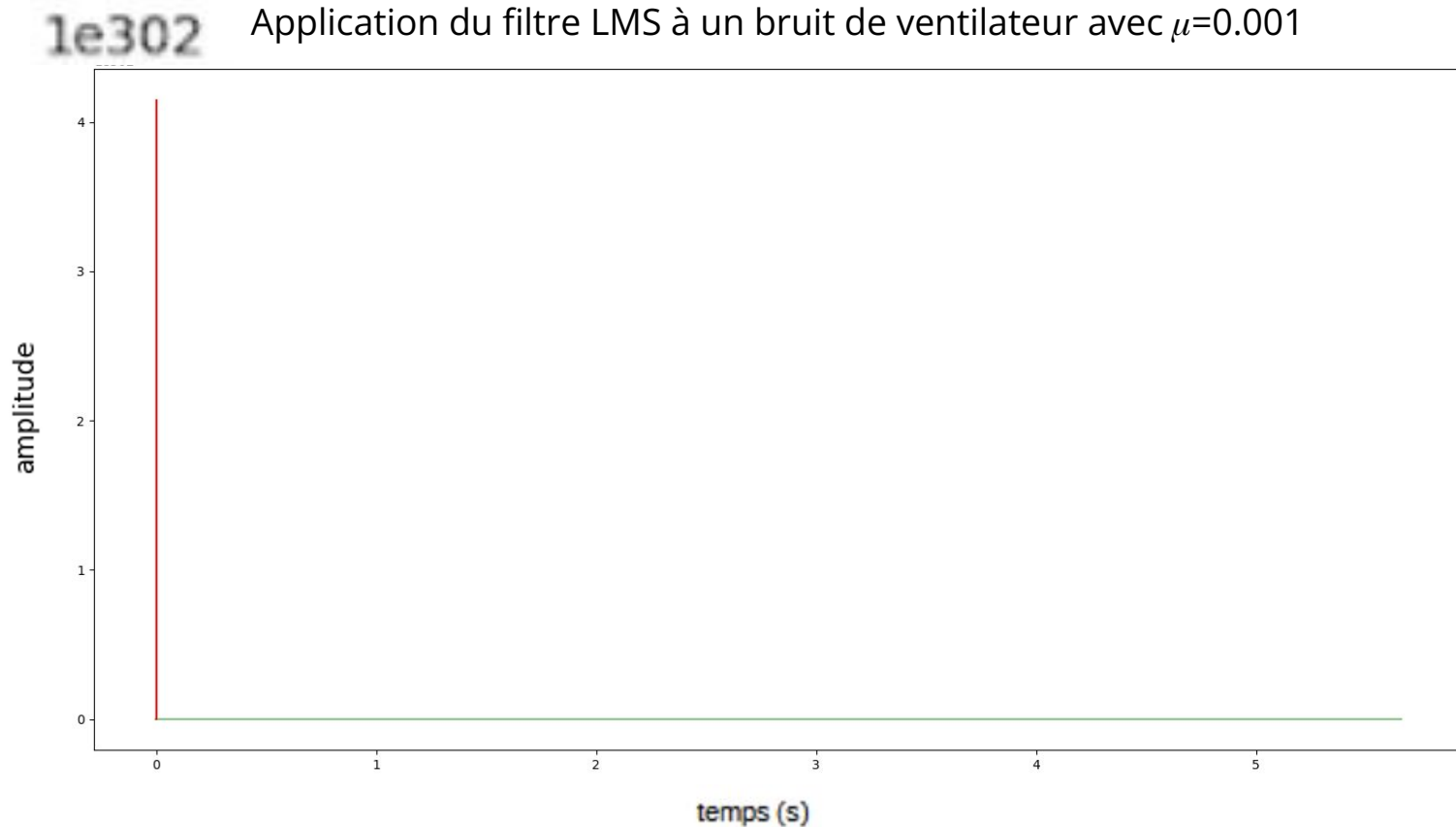


Plus μ est grand, plus la convergence est rapide.



Un trop grand μ peut causer une réponse divergente.

Processus 3 : filtres LMS (application à un bruit réel)

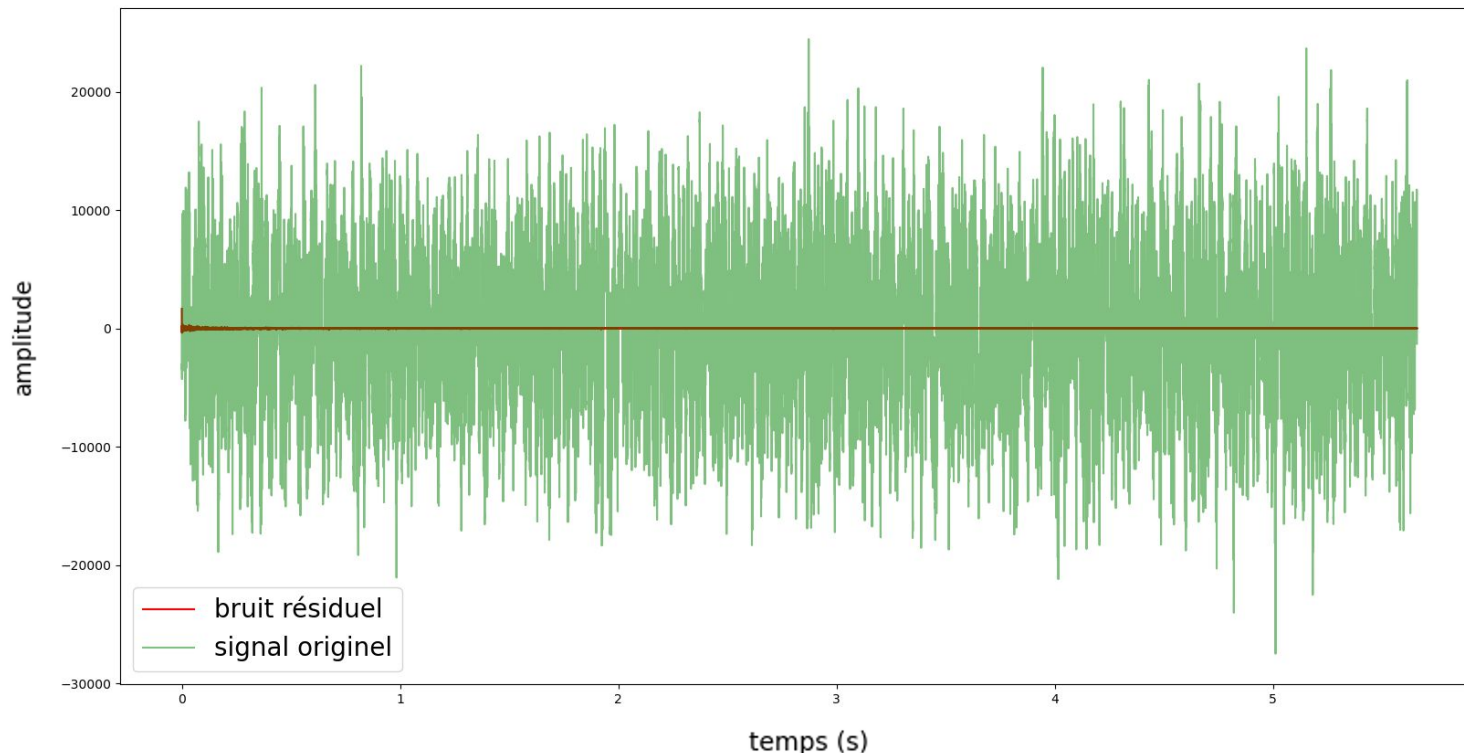


✗ Divergence même avec un μ faible

Processus 3 : filtres LMS (Normalized LMS)

Variation du pas selon :
$$\mu(k) = \frac{\mu_{lim}}{\sum_{i=k-(p-1)}^{i=k} x^2[i]}$$
 Avec $\mu_{lim} < 2$.

Application du filtre NLMS à un bruit de ventilateur avec $\mu_{lim}=0.5$



Comparaison

	Avantages	Inconvénients
1/ analyse sonore	permet de choisir précisément les fréquences à atténuer	temps de calcul trop long
2/ approche électrique	traitement presque immédiat	
3/ filtre informatique	efficacité	ne permet pas de choisir les fréquences atténuées

Remarque : communications possibles avec les filtres LMS (on connaît le signal à laisser passer)

Utilisation actuelle :

filtres LMS et NLMS par des circuits électroniques

Annexes

Détails microphone

Marque	AGPTEK
Couleur	AC02 Microphone 3.5mm
Distance focale	Jack 3,5 mm
Technologie de connectivité	Auxiliaire
Source d'alimentation	Alimenté par pile
Poids de l'article	0.05 Livres
Diagramme polaire	Omnidirectionnel
Plateforme matérielle	Ordinateur
Plage de fréquence	20Hz-20kHz
Réponse en fréquence	20 KHz

Méthode 1

création du spectre en fréquence

#Ne est le nombre d'échantillons, Te le pas d'échantillonnage

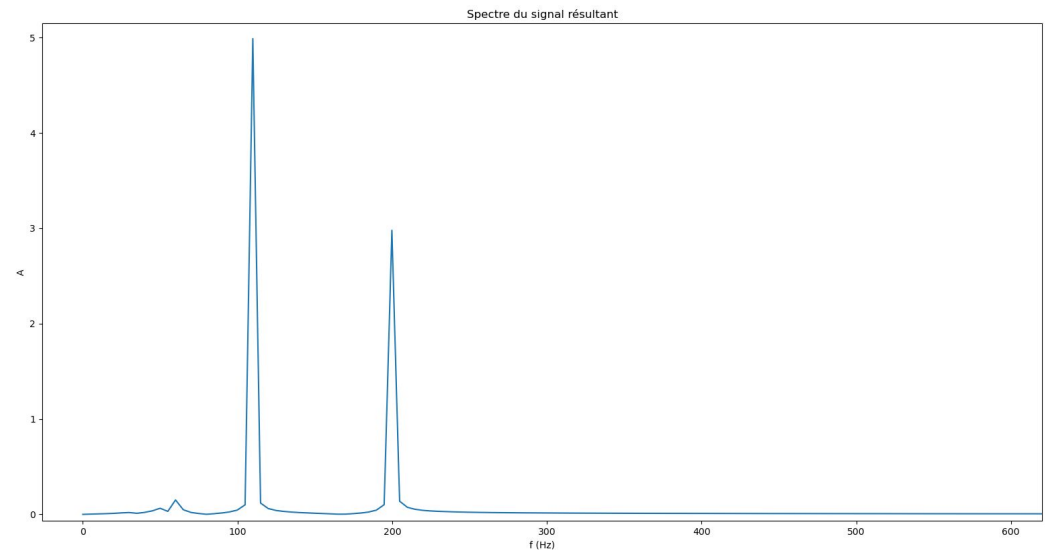
```
Ne = len(signal)
Te = temps[1]-temps[0]
spectre = 2*np.absolute(np.fft.rfft(signal))/Ne
frequences = np.arange(Ne/2+1,)*1.0/(Te*Ne)
```

création du signal en opposition de phase

```
def fonctionsindecallee(L,A,t):
    res = 0

    for i in range (len(L)):
        res += A[i]*np.sin(2*np.pi*L[i]*t + np.pi)
    return res

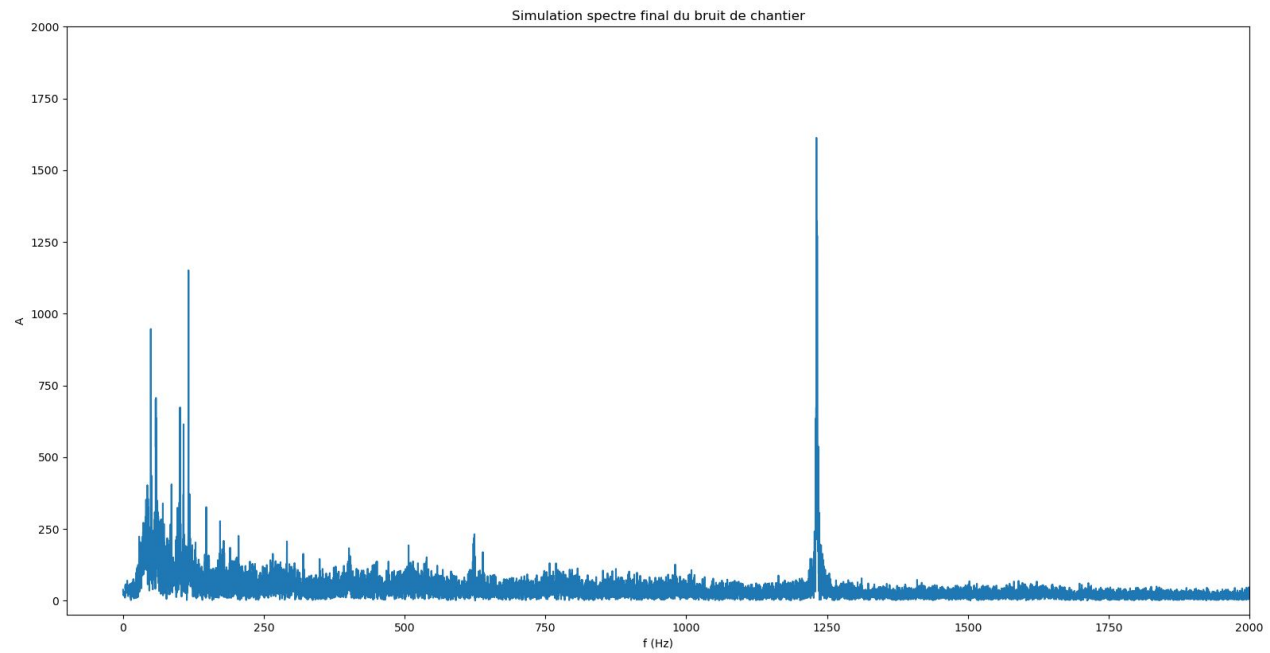
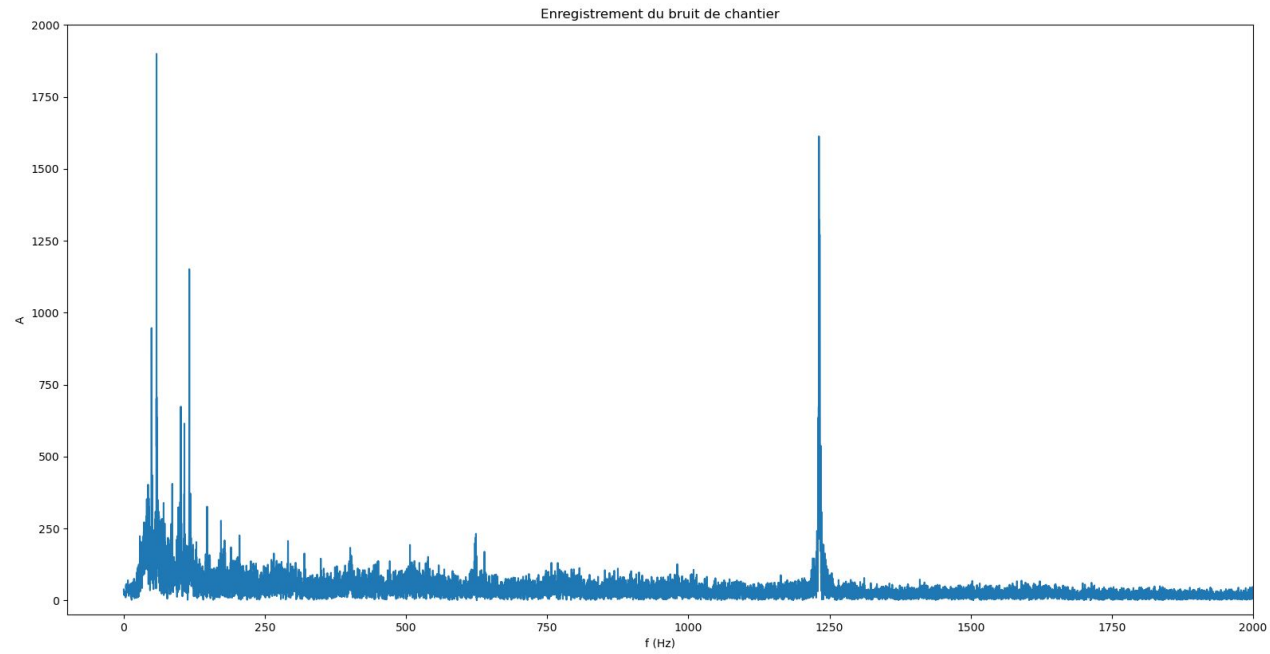
def creationsignalcontraire(liste,amplitudes,duree):
    L=[]
    for i in range (len(liste)):
        if liste[i]<75:
            L.append(liste[i])
    temps = np.linspace(0,duree,1000)
    signaldesortie = [fonctionsindecallee(L,amplitudes,t) + f(t)+f2(t)+f3(t)+f4(t)+f5(t) for t in temps]
    return (temps,signaldesortie)
```



##

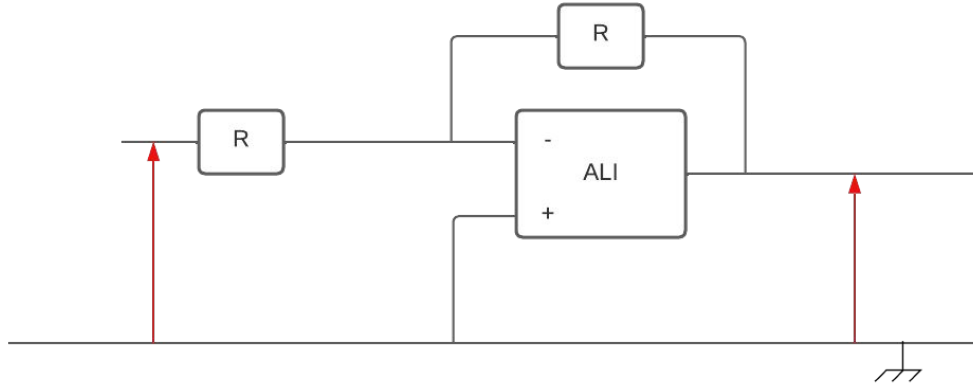
```
dephasage = [1 for i in range(len(listefrequencies))]  
  
for i in range (len(listefrequencies)):  
    valeuramplitude = listeamplitudes[i]  
  
    while valeuramplitude>listeamplitudes[i]/2:  
        contre = np.array(creationsignalcontraire(listefrequencies,listeamplitudes,dephasage)[1])  
        # somme des deux signaux  
        final = signal + contre  
        # transformation en une liste  
        finalliste = []  
        for k in range (len(temps)):  
            finalliste.append( final[k] )  
        # création du spectre en fréquence de la somme  
        spectrefinal = 2*np.absolute(np.fft.rfft(finalliste))/Ne  
        valeuramplitude = rechercheamplitude(listefrequencies[i],frequences,spectrefinal)  
        dephasage[i] += 1
```

(simulation chantier)

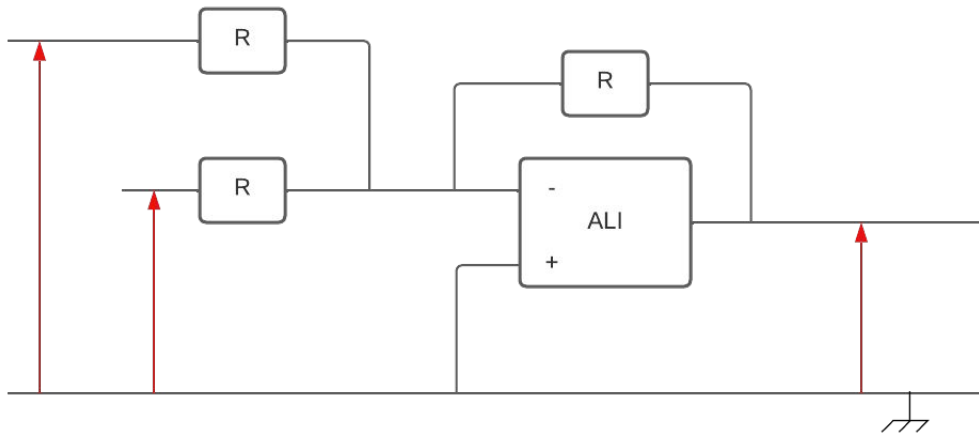


Méthode 2

montage inverseur ($R=10\text{k}\Omega$)



montage sommateur ($R=10\text{k}\Omega$)



Méthode 3

Review and Comparison of Variable Step-Size LMS Algorithms

Dariusz Bismor, Krzysztof Czyz and Zbigniew Ogonowski

Institute of Automatic Control, Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland

LMS

```
#nombre d'itérations
nb = 2000

y = np.zeros(nb)
e = np.zeros(nb)
s = np.zeros(nb)

for k in range (nb):
    #mise a jour des coeffs de X
    X[1:]=X[0:-1]
    X[0]=x(k)

    #signal contraire à émis à l'instant k
    y[k]=np.dot(X,W)
    s[k]=-y[k]

    #calcul de l'erreur
    e[k]=x(k)-y[k]

    #mise à jour de la matrice de l'équation pondérée
    W=W+mu*e[k]*X
```

NLMS

```
#nombre d'itérations
nb = 2000

y = np.zeros(nb)
e = np.zeros(nb)
s = np.zeros(nb)

mulim = 0.5
mu = mulim

for k in range (nb):
    #mise a jour des coeffs de X
    X[1:]=X[0:-1]
    X[0]=x(k)

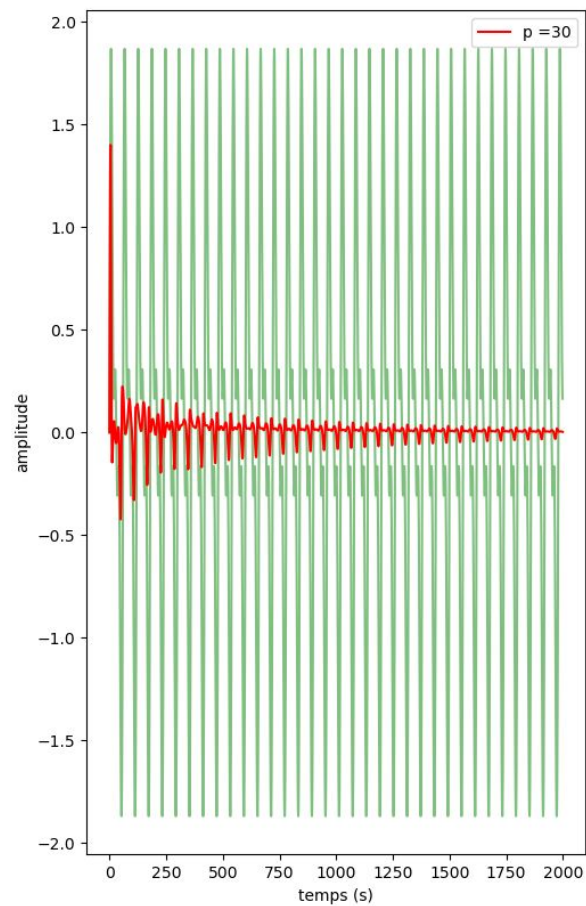
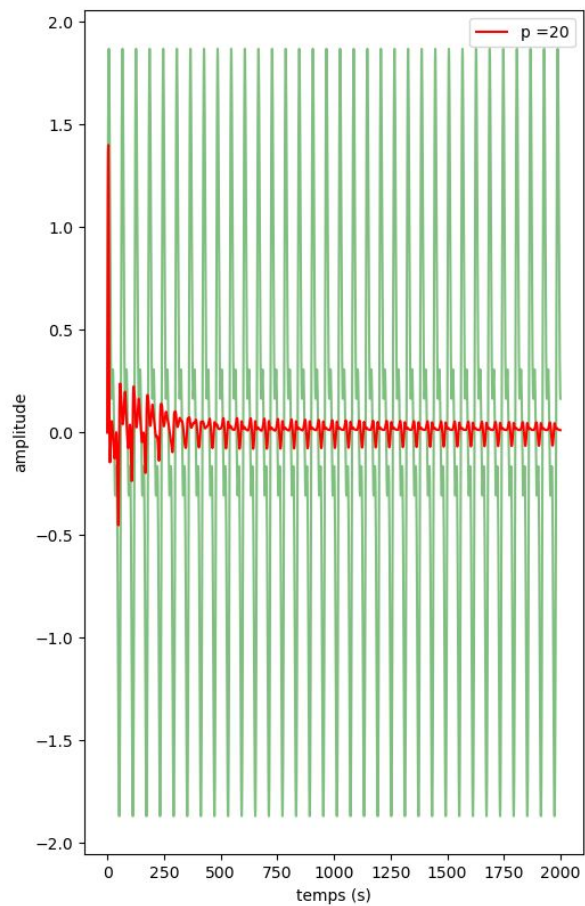
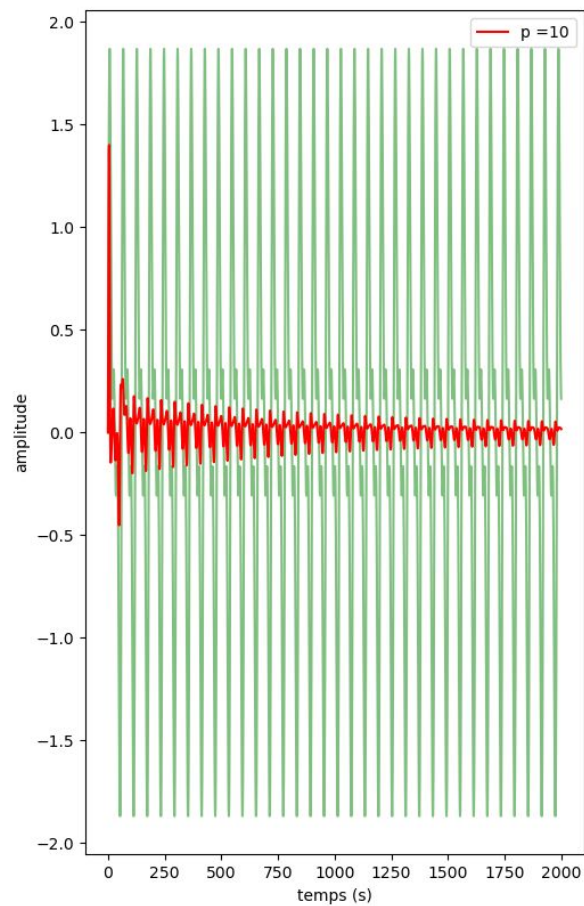
    #signal contraire à émis à l'instant k
    y[k]=np.dot(X,W)
    s[k]=-y[k]

    #calcul de l'erreur
    e[k]=x(k)-y[k]

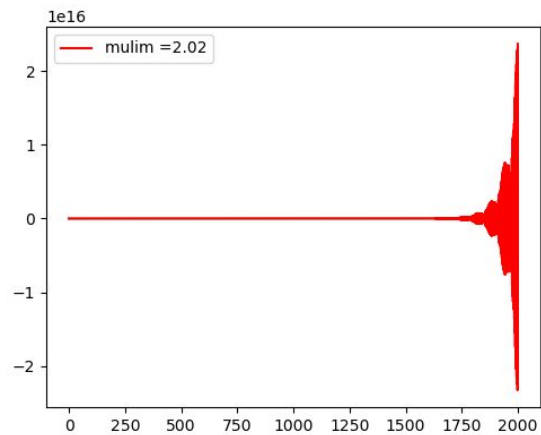
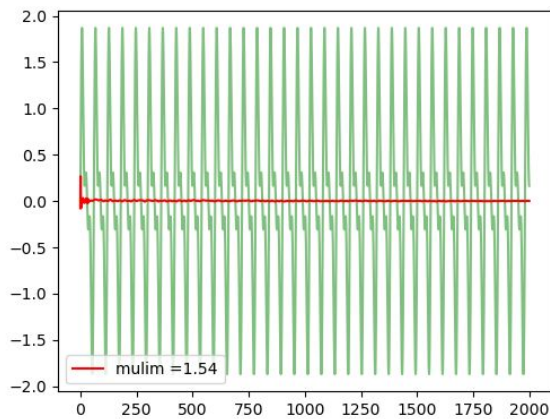
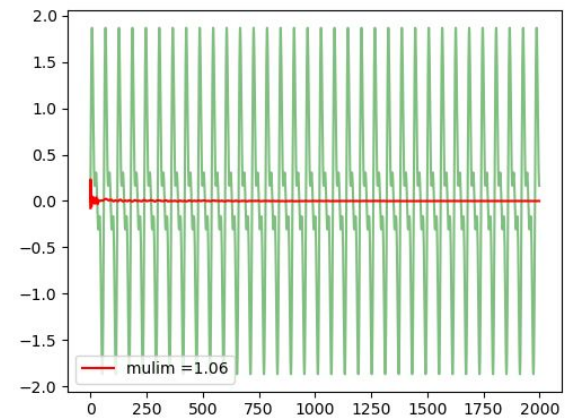
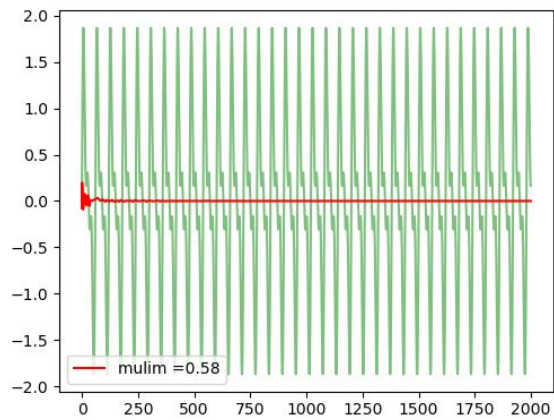
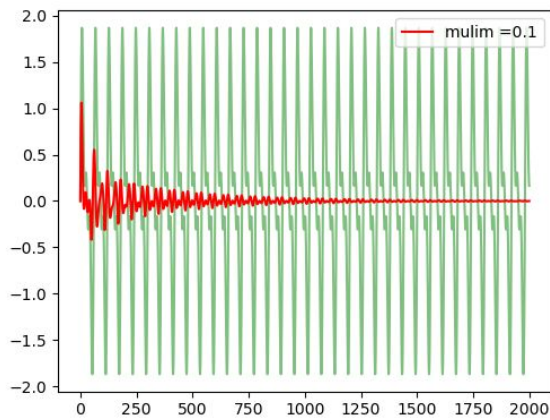
    #mise à jour du coefficient mu
    if np.dot(X,X)!=0:
        mu = mulim / np.dot(X,X)

    #mise à jour de la matrice de l'équation pondérée
    W=W+mu*e[k]*X
```

Variation de l'ordre p du filtre LMS



Variation de la constante μ_{lim}



Conclusion : limite pour les cas simples à $\mu_{\text{lim}} = 2$.