compte rendu du :

TP4- Filtrage Analogique

Réalisé par : Hiba QOUIQA

Encadré par : Mr. Alae AMMOUR

filière : IA

2022-2023

Sommaire :

1. Buts du Tp :

- Appliquer un filtre réel pour supprimer les composantes indésirables d'un signal.
- Améliorer la qualité de filtrage en augmentant l'ordre du filtre.

2. Filtrage et diagrame de Bode :

1.Nous souhaitons appliquer un filtre passe-haut pour supprimer la composante à 50 Hz.

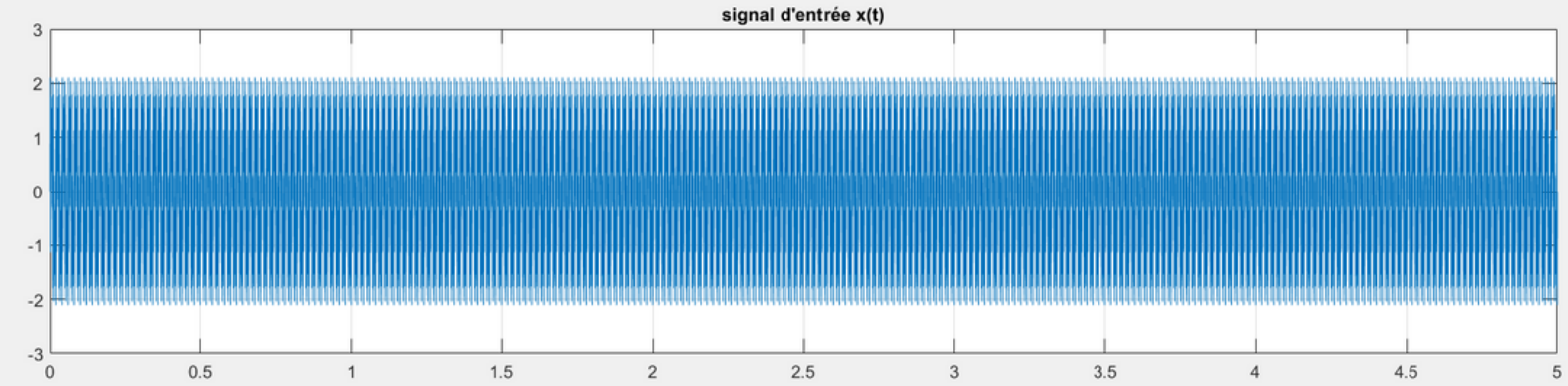
Soit notre signal d'entrée : **x(t) = sin(2.pi.f1.t) + sin(2.pi.f2.t) + sin(2.pi.f3.t)**
Avec f1 = 500 Hz, f2 = 400 Hz et f3 = 50 Hz

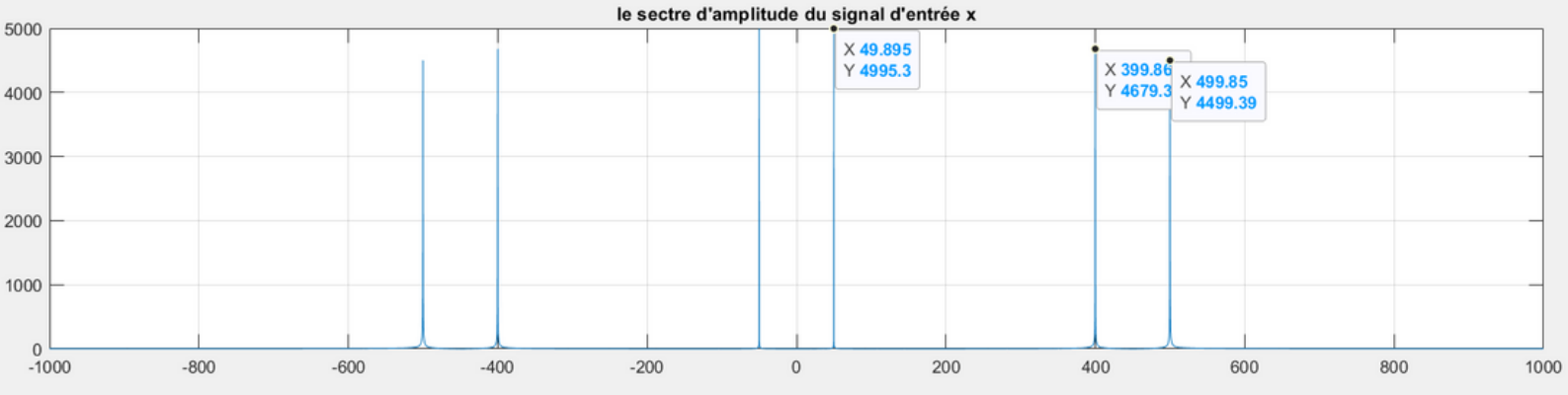
1. On défini le signal sur t[0 5] avec Te=0.0001s :

```
Te=0.0001;
fe=1/Te;
t=0:Te:5;


x=sin(2*pi*500*t)+sin(2*pi*400*t)+sin(2*pi*50*t);
```

2. On trace le signal x(t) et sa transformé de Fourrier.

```
%%question2

Te=0.0005;
fe=1/Te;
t=0:Te:5;


x=sin(2*pi*500*t)+sin(2*pi*400*t)+sin(2*pi*50*t);

N=length(t);%nombre d'echantillons egal à la longeur du t

f = (-N/2:N/2-1)*(fe/N); % le pas de discrétisation est fe/N

y=fft(x);


subplot(211)
plot(t,x)
title("signal d'entrée x(t)")
grid on
subplot(212)
plot(f,fftshift(abs(y)));
title("le sectre d'amplitude du signal d'entrée x")
grid on
```

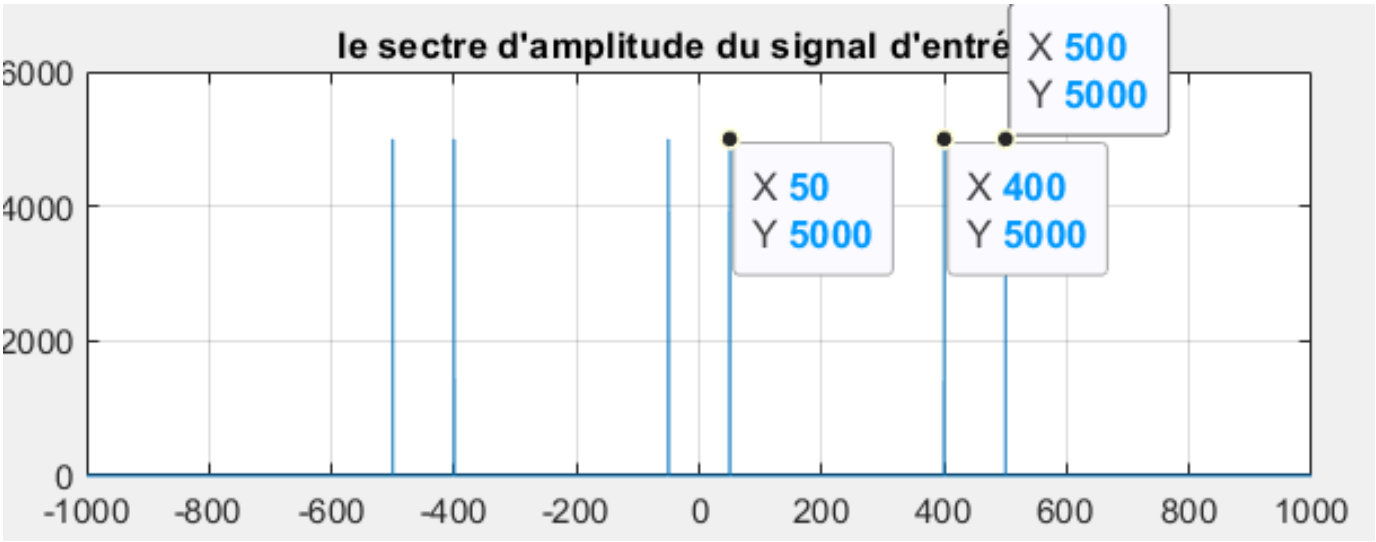




- Remarque:

- Après le traçage du spectre d'amplitude du signal x on remarque que les piques des fréquences sont sur des fréquences qui sont presque égales aux fréquences 50Hz , 400Hz et 500Hz qui constituent le signal x .Cette différence de valeur est due à la fuite spectrale , pour la fixer on peut tout simplement changer t à t[0 5-Te]

```
t=0:Te:5-Te;
```



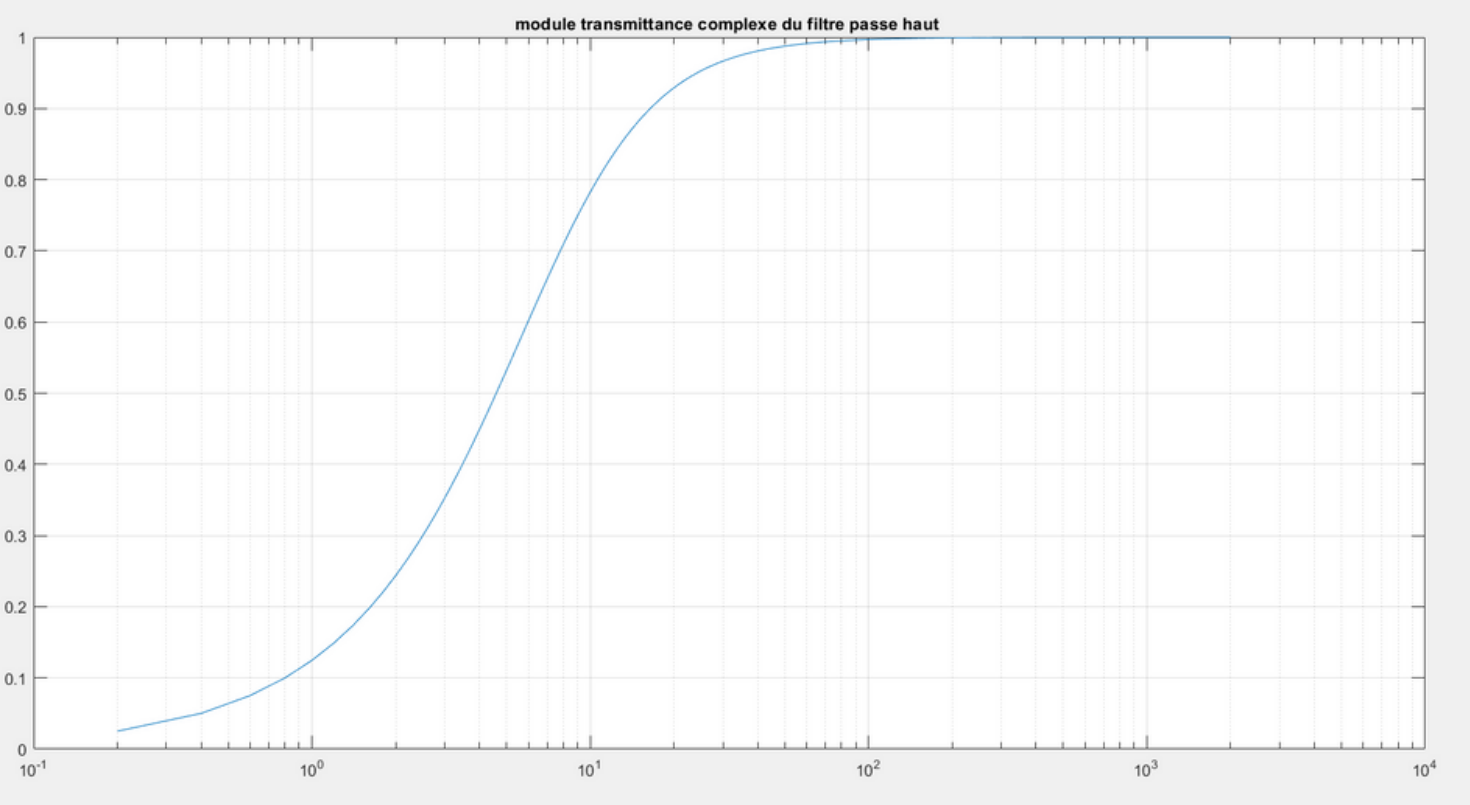
La fonction H(f) (transmittance complexe) du filtre passe haut de premier ordre est donnée par :

H(f) = (K.j.w/wc) / (1 + j. w/wc)

Avec K le gain du signal, w la pulsation et wc la pulsation de coupure.

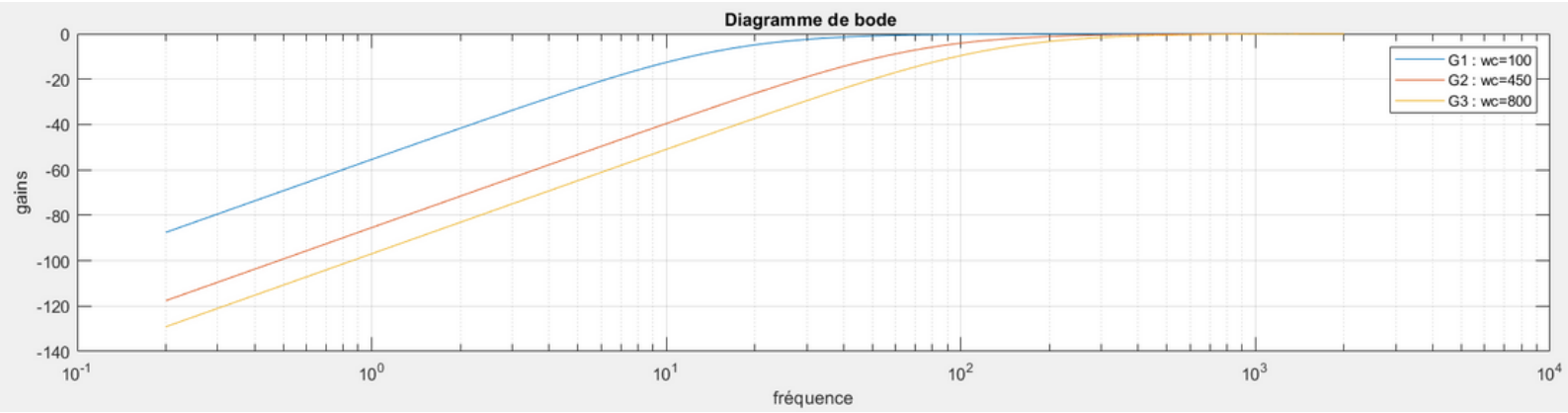
1. Tracer le module de la fonction H(f) avec K=1 et wc = 50 rad/s

```
%%question 1
f2=(0:N-1)*fe/N;
k=1;
w=2*pi*f2;
wc=50;
H=(k*1i*w/wc)./(1+1i*w/wc);
semilogx(f2,abs(H));
grid on
title("module transmittance complexe du filtre passe haut ")
```



2. On trace 20.log(|H(f)|) pour différentes pulsations de coupure wc,:

```
wc1= 100;
wc2 =450;
wc3 =800;

H1=(k*1i*w/wc1)./(1+1i*w/wc1);
H2=(k*1i*w/wc2)./(1+1i*w/wc2);
H3=(k*1i*w/wc3)./(1+1i*w/wc3);

%le gain pour differente fréquence de coupure

G = 20*log(abs(H));
G1 = 20*log(abs(H1));
G2 = 20*log(abs(H2));
G3 = 20*log(abs(H3));

a1= angle(H1);
a2= angle(H2);
a3= angle(H3);

subplot(211)
semilogx(f2,G1,f2,G2,f2,G3);
grid on
title("Diagramme de bode ")
ylabel("gains")
xlabel("fréquence")
```



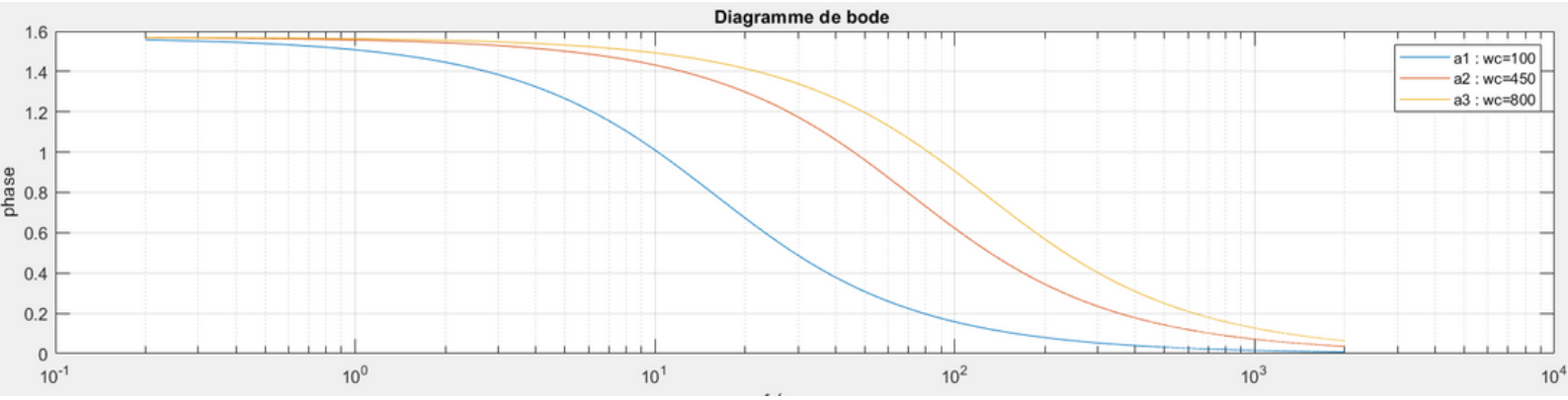
- Remarque:

- Après le traçage du diagramme de Bode du gain pour des pulsations de coupure différentes ,on remarque :

- pour Les signaux qui auront une pulsation supérieure à wc (hautes fréquences) , sont dans une zone où le gain est proche de 0, ce qui signifie que le module du signal de sortie est à peu près identique au module du signal d'entrée . Les signaux ont donc quasiment la même amplitude. Le signal d'entrée sera conservé en sortie.
- Par contre, les signaux ayant une pulsation inférieure à wc (basses fréquences) vont être atténués (d'autant plus fortement que l'on s'éloigne de wc). En effet, un gain négatif signifie un rapport des modules inférieur à 1. L'amplitude du signal de sortie est donc inférieure à celui d'entrée.
- Plus que wc augmente plus que l'amplitude du signal de sortie sera inférieure à celui d'entrée.

On trace le diagramme de Bode de phase :

```
a1= angle(H1);
a2= angle(H2);
a3= angle(H3);

semilogx(f2,a1,f2,a2,f2,a3);
grid on
title("Diagramme de bode ")
ylabel("phase")
xlabel("fréquence")
legend("a1 : wc=100","a2 : wc=450","a3 : wc=800")
```



- Remarque:

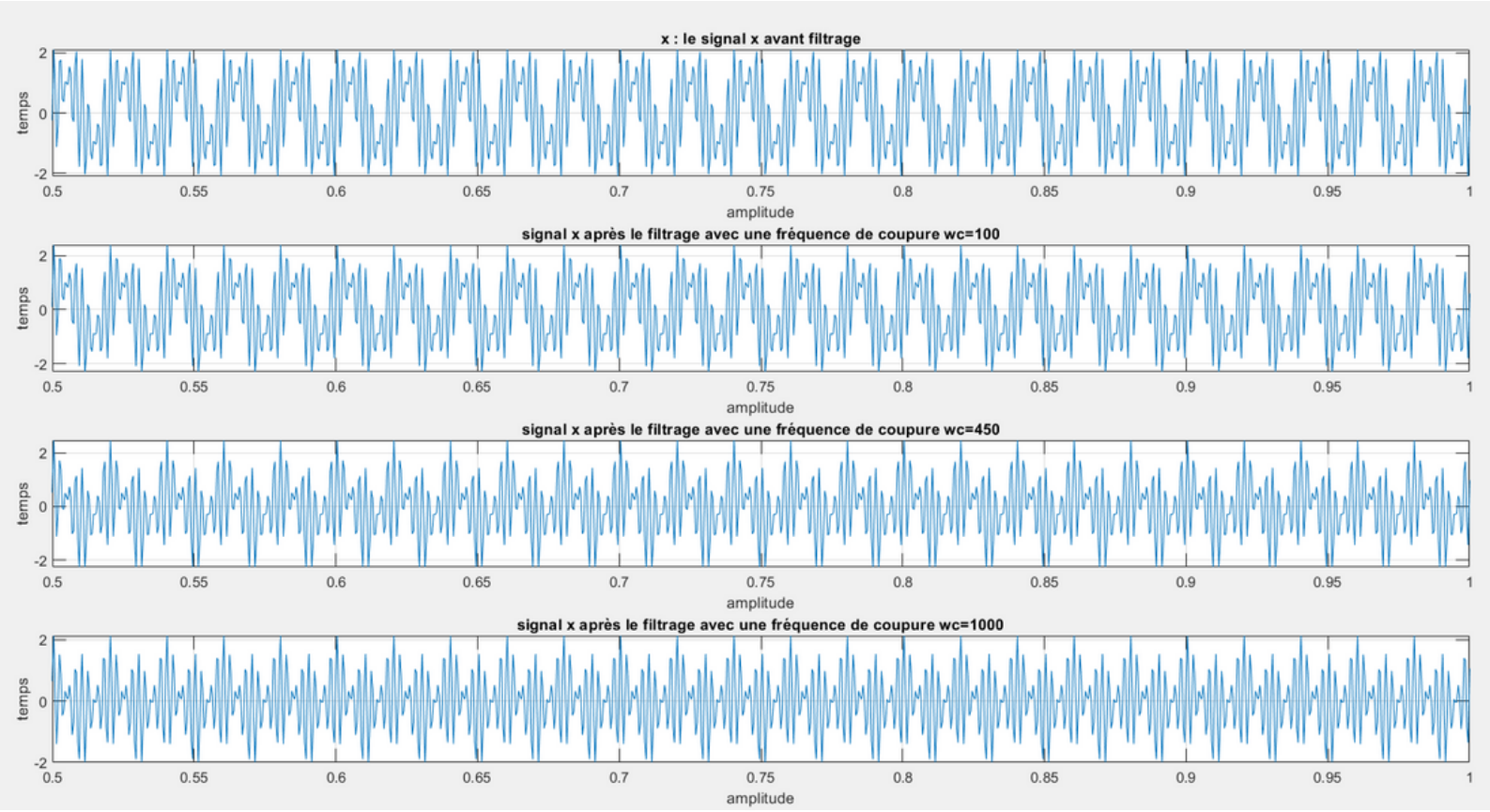
- De meme pour la phase ,on remarque :

- pour Les signaux qui auront une pulsation supérieure à wc (hautes fréquences) , sont dans une zone où la phase est proche de 0, ce qui signifie que la phase du signal de sortie est à peu près identique à la phase du signal d'entrée . Les signaux ont donc quasiment la même phase. Le signal d'entrée sera conservé en sortie.
- Par contre, les signaux ayant une pulsation inférieure à wc (basses fréquences) vont être déphasés (d'autant plus fortement que l'on s'éloigne de wc). Le signal de sortie est donc déphasé à celui d'entrée.
- Plus que wc augmente plus que le signal de sortie sera déphasé de celui d'entrée.

3. Dans cette étape on applique le filtrage :

```
%% question3

%concption du filtre
x_filtre1 = H1.*y; %on multiple le filtre par le signal
x_filtre2 = H2.*y;
x_filtre3 = H3.*y;

x1 = ifft(x_filtre1,"symmetric");% on applique la transformée de fourier inverse
% pour visualiser le signal filtré dans le domaine temporel
x2 = ifft(x_filtre2,"symmetric");
x3 = ifft(x_filtre3,"symmetric");
```

```
subplot(411)
plot(t,x);
grid on
ylabel("temps")
xlabel("amplitude")
xlim([0.5 0.52]);
title("x : le signal x avant filtrage")
```



- Remarque:

On remarque que plus la fréquence de coupure augmente plus que le bruit s'attenue .

4. wc=1000Hz est la fréquence de coupure parfaite car d'après le graph on remarque que cette fréquence nous a permis d'enlever le bruit en gardant l'information.

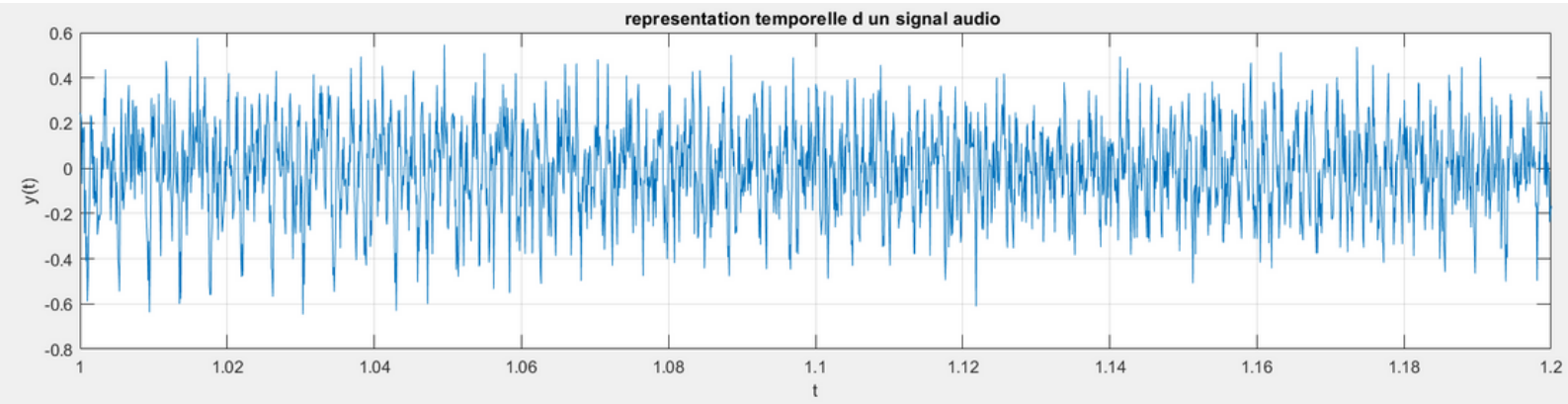
3.Dé-bruitage d'un signal sonore:

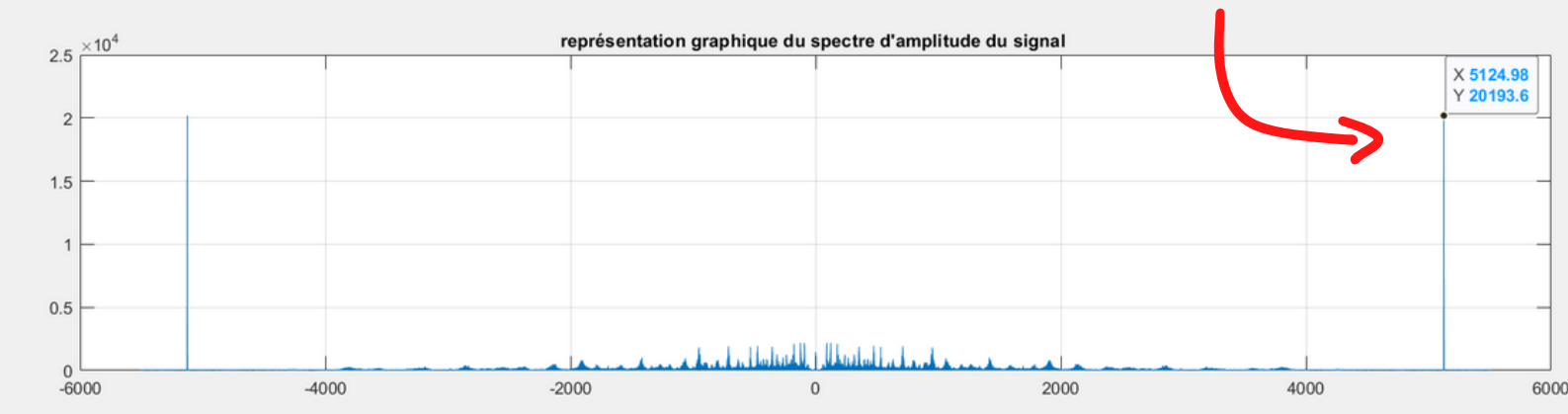
Dans son petit studio du CROUS, un mauvais futur ingénieur a enregistré une musique en « .wav » avec un très vieux micro. Le résultat est peu concluant, un bruit strident s'est ajouté à sa musique. Heureusement son voisin, expert en traitement du signal est là pour le secourir :

1. Pour supprimer le bruit on prpose utiliser un filtre-passe bas reel

2. D'abord pour filtrer le bruit on doit télécharger l'audio on l'applique la transformée de fourier pour visualiser le spectre des fréquences

```
[y,Fs]=audioread("test.wav");

Ts= 1/Fs;
N=length(y); % le nbr d'echantillons égal à la taille du vecteur y
t=0:Ts:(N-1)*Ts;

subplot(211)
plot(t,y)
title('representation temporelle d un signal audio ')
xlabel('t')
ylabel('y(t)')
xlim([1 1.2]);
grid on

fshift = (-N/2:N/2-1)*(Fs/N); % le pas de discrétisation est fe/N
x=fft(y);

subplot(212)
plot(fshift,fftshift(abs(x)));
grid on
title(" représentation graphique du spectre d'amplitude du signal  ")
```





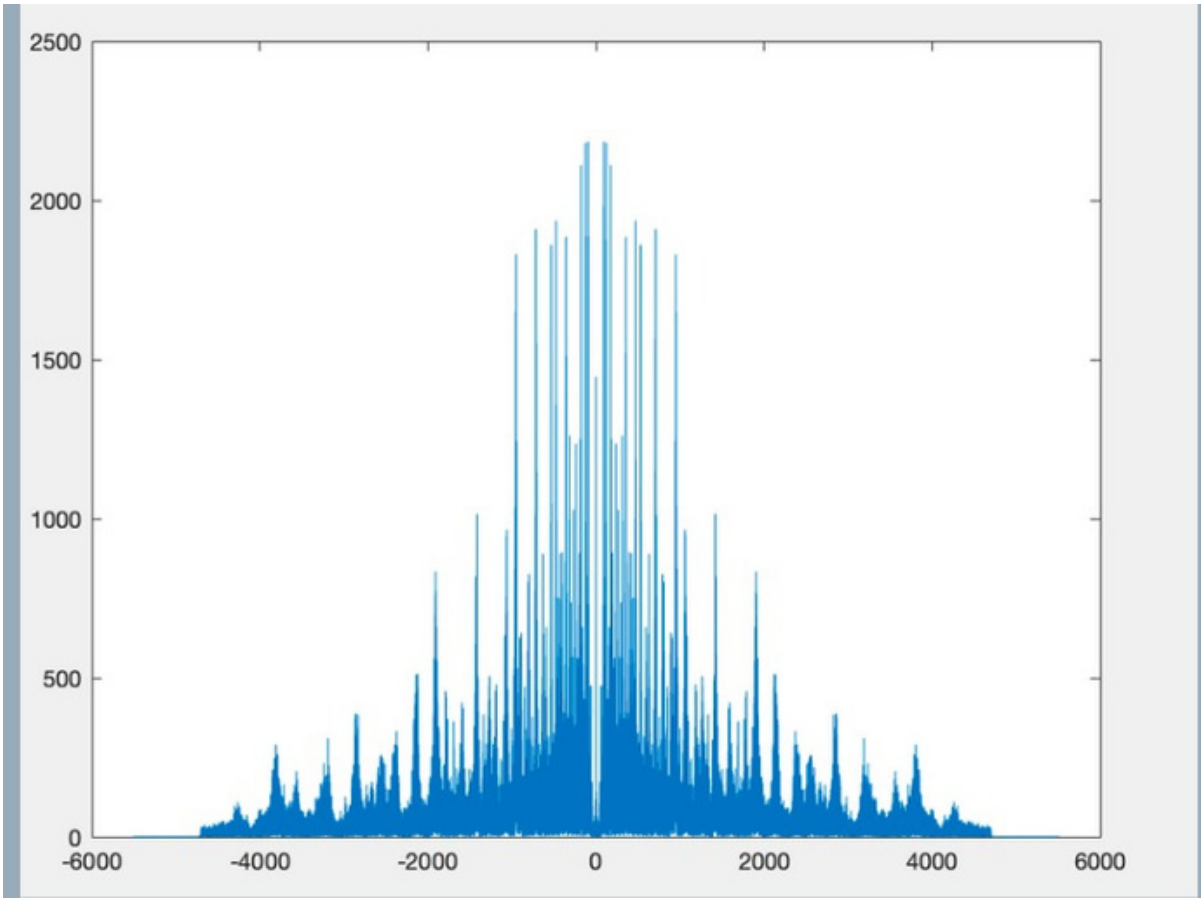
- Remarque:

On remarque que le bruit causé est du à une fréquence de presque 5124 Hz ,pour l'enlever on propose un filtre pass_bas avec une fc=4700Hz car pour notre filtre reel n'applique pas un filtrage parfait ,c-àd dans la bande passante notre filtre tend vers 1 mais n'est pas égal à un ce qui permet de passer qlqs fréquences indésirable.

La fonction H(f) (transmittance complexe) du filtre passe_bas de premier ordre est donnée par : **H(f) = K / (1 + j. (w/wc)^1000))**

-avec : k=1 ,fc=4700Hz

4. wc=1000Hz est la fréquence de coupure parfaite car d'après le graph on remarque que cette fréquence nous a permis d'enlever le bruit en gardant l'information.



Concusion :

En conclusion, dans ce TP de filtrage analogique, on a exploré différentes méthodes pour filtrer des signaux, y compris la création de filtres passe-haut et la suppression de bruits dans les signaux sonores. On a tracé des diagrammes de Bode pour les filtres et on a choisi une fréquence de coupure optimale. On a également mis en œuvre ces filtres pour en déterminer l'effet sur les signaux. Finalement, on a amélioré la qualité de filtrage en augmentant l'ordre du filtre. En général, on a compris l'importance du filtrage pour sélectionner les composantes souhaitées d'un signal et en supprimant les composantes indésirables.