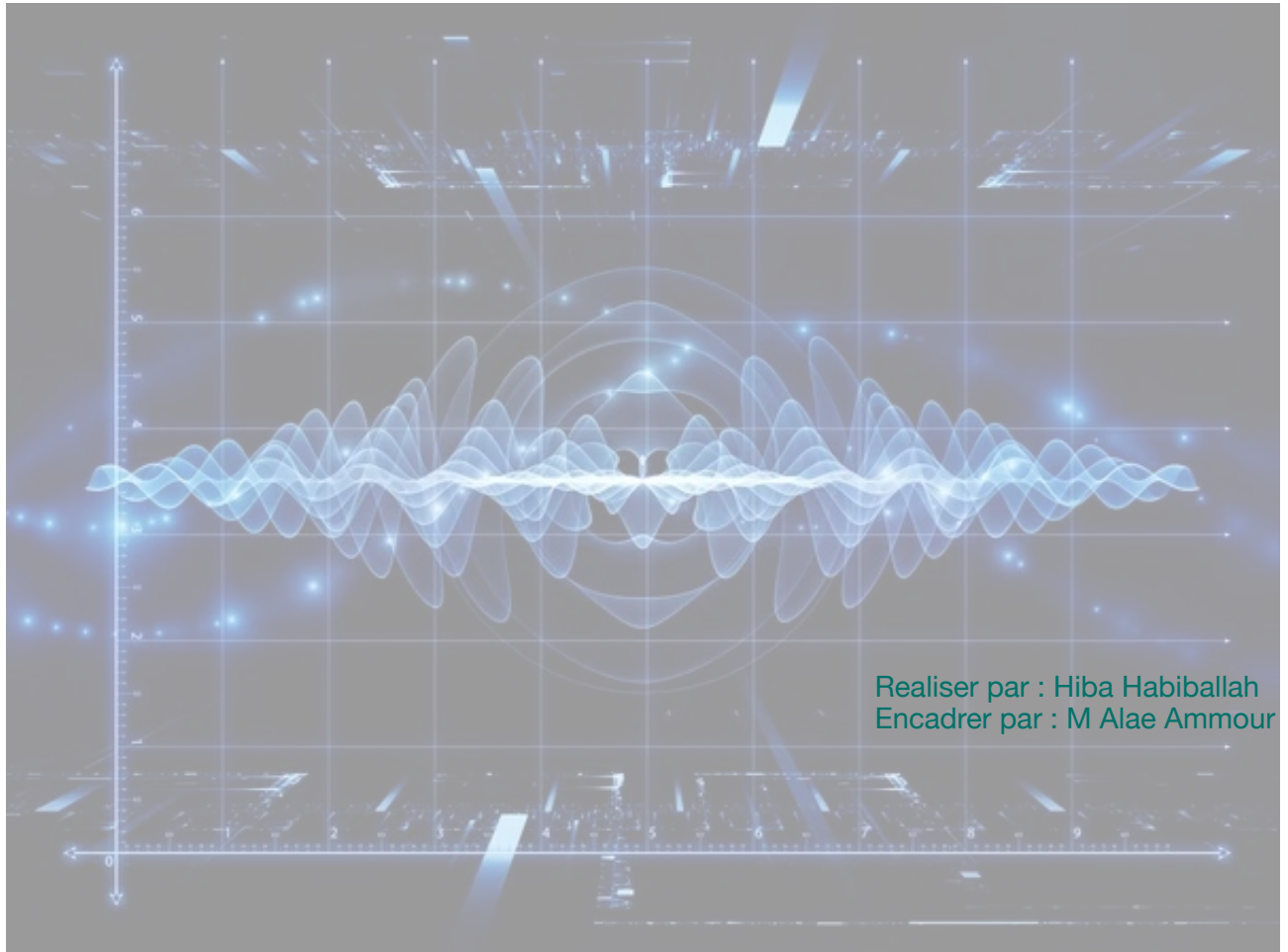


Rapport TP4

Filtrage et diagramme de Bode



Objectifs :

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

1) Définir le signal $x(t)$ sur $t = [0 \ 5]$ avec $T_e = 0,0001$ s.

Code :

```
f1 = 500;
```

```
f2 = 400;
```

```
f3 = 50;
```

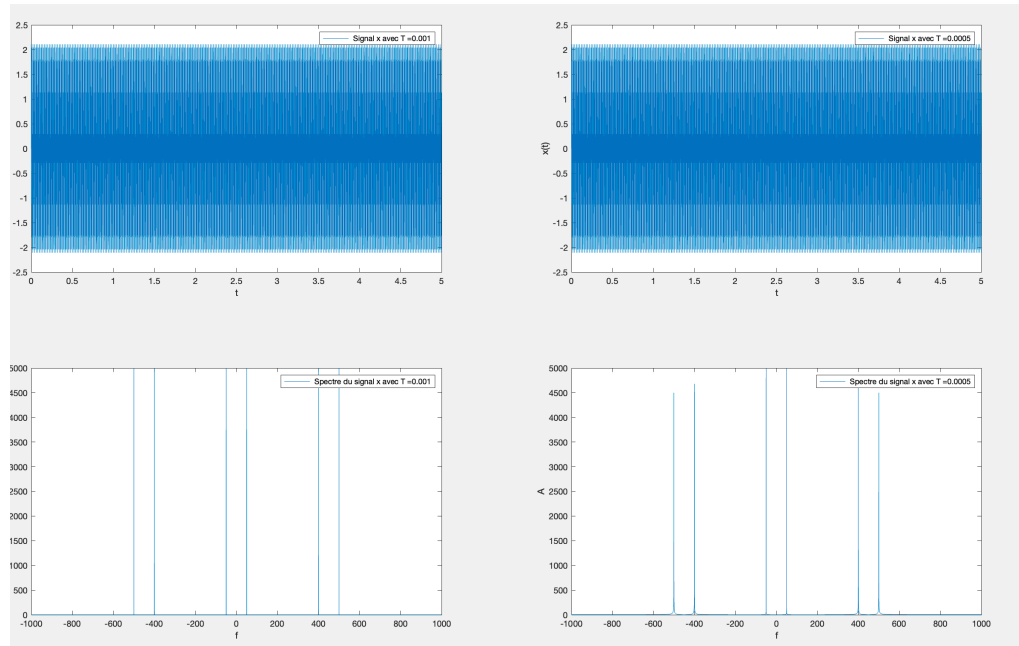
```
Te = 0.0001;
```

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

```
x = sin(2*pi*f1*t)+sin(2*pi*f2*t)+sin(2*pi*f3*t);
```

2) Tracer le signal $x(t)$ et sa transformé de Fourier

on remarque que le premier signal n'est pas précis



Code :

```
subplot(2,2,1)
    plot(t,x);
    legend(" Signal x avec T =0.001");
    xlabel("t");
    ylabel("x(t)");
subplot(2,2,3)
    y = fft(x);
    plot(fshift,fftshift(abs(y)));
    legend(" Spectre du signal x avec T =0.001");
    xlabel("f");
    ylabel("A");
subplot(2,2,2)
    Te2 = 0.0005 ;
    fe2 = 1/Te2;
    t2 = [0:Te2:5];
    x2 = sin(2*pi*f1*t2)+sin(2*pi*f2*t2)+sin(2*pi*f3*t2);
    N2 = length(t2);
    fshift2 = (-N2/2:(N2/2)-1)*(fe2/N2);
    plot(t2,x2);
```

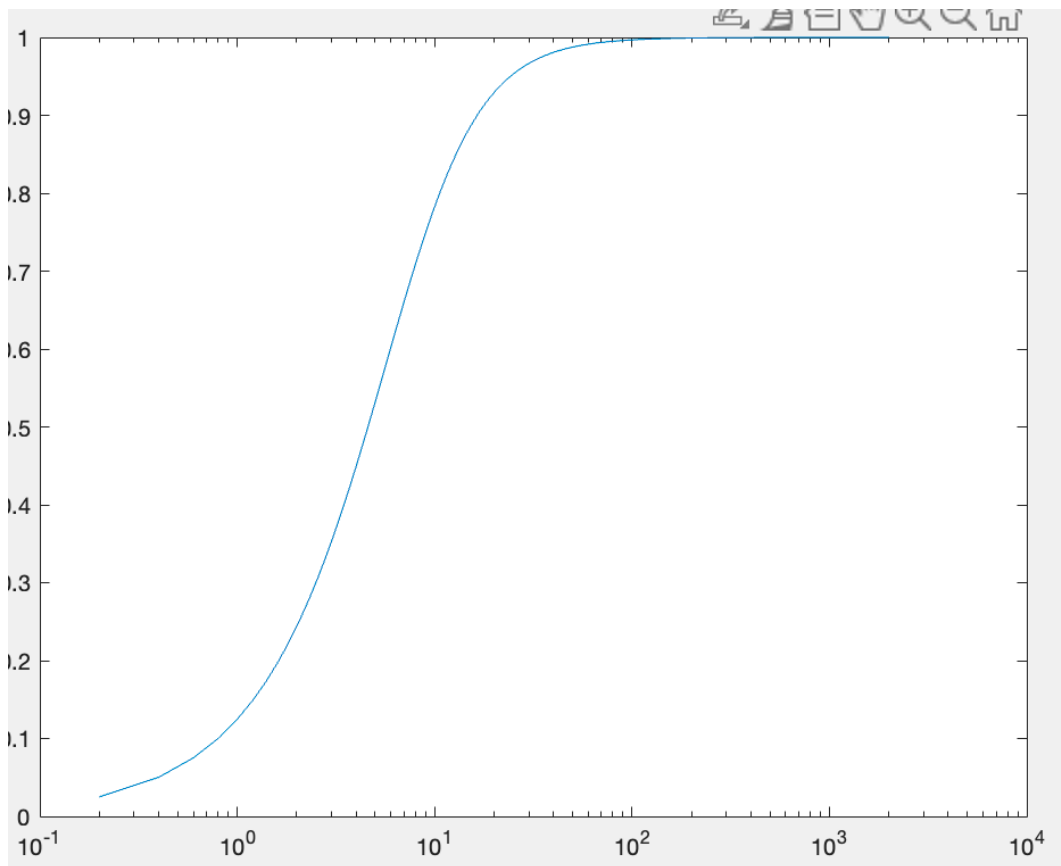
```

legend(" Signal x avec T =0.0005");
xlabel("t");
ylabel("x(t)");
subplot(2,2,4)
y2 = fft(x2);
plot(fshift2,fftshift(abs(y2)));
legend(" Spectre du signal x avec T =0.0005");
xlabel("f");
ylabel("A");

```

$$H(f) = (K \cdot j \cdot \omega / \omega_c) / (1 + j \cdot \omega / \omega_c)$$

1) Tracer le module de la fonction H(f) avec K=1 et $\omega_c = 50$ rad/s



Code :

```

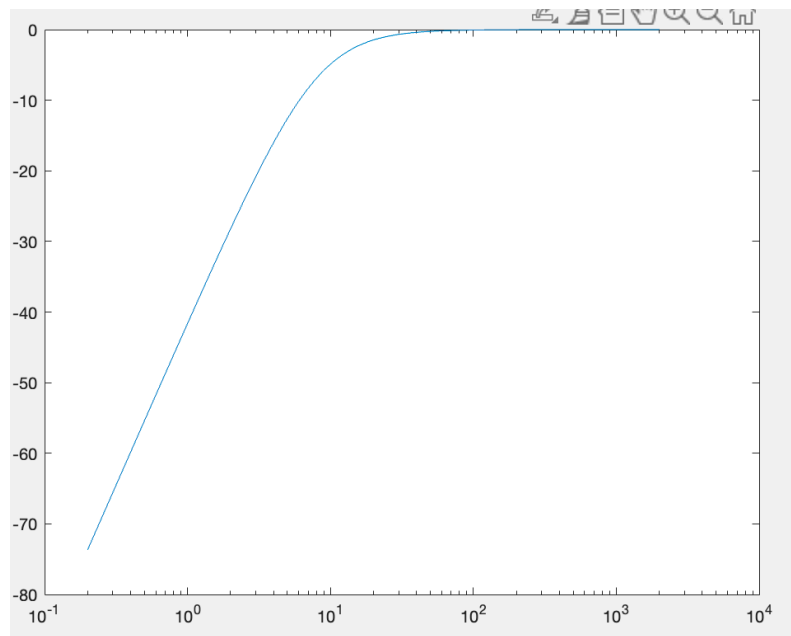
% frequence wc=50
H = (K*1i*w/wc)./(1+1i*w/wc);
semilogx(f,abs(H));

```

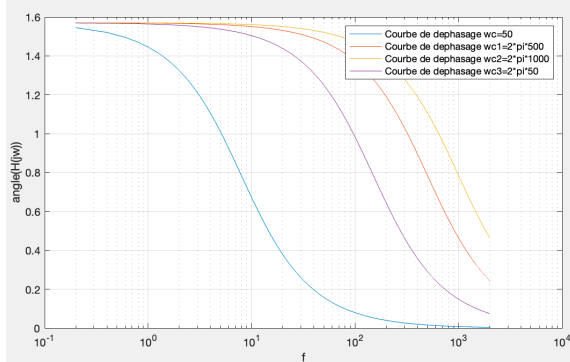
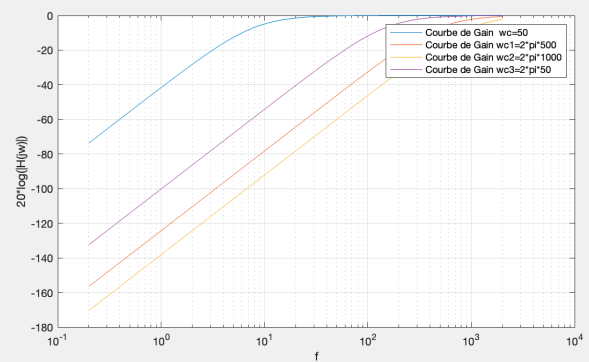
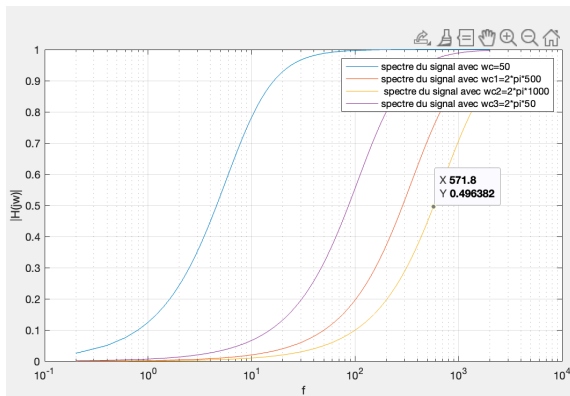
2) Tracer $20 \cdot \log(|H(f)|)$ pour différentes pulsations de coupure ω_c , qu'observez-vous? (Afficher avec semilogx)

Code :

```
G = 20*log(abs(H));
semilogx(f,G);
```



3) Choisissez différentes fréquences de coupure et appliquez ce filtrage dans l'espace des fréquences.



Code :

```
subplot(2,2,1);

semilogx(f,abs(H),f,abs(H1),f,abs(H2),f,abs(H3));
legend("spectre du signal avec wc=50","spectre du signal avec wc1=2*pi*500","
spectre du signal avec wc2=2*pi*1000","spectre du signal avec wc3=2*pi*50");
grid on
xlabel("f");
ylabel("|H(jw)|");
subplot(2,2,2);
semilogx(f,G,f,G1,f,G2,f,G3);
legend("Courbe de Gain wc=50","Courbe de Gain wc1=2*pi*500","Courbe de Gain
wc2=2*pi*1000","Courbe de Gain wc3=2*pi*50");
grid on
xlabel("f");
ylabel("20*log(|H(jw)|)");

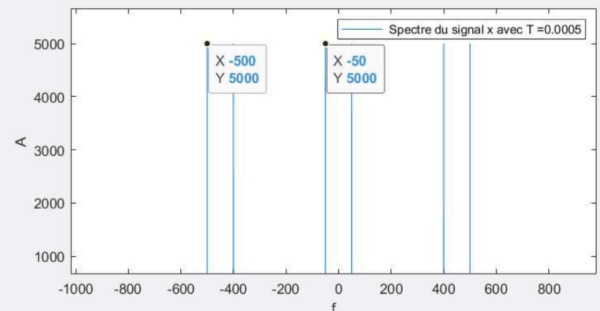
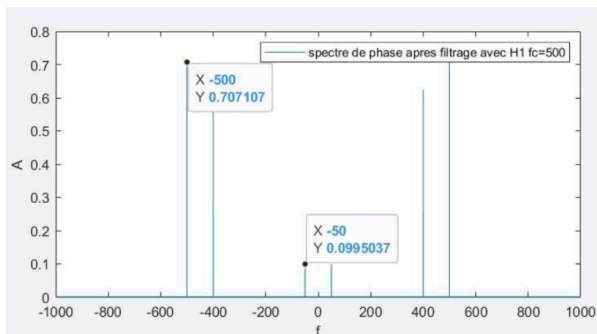
subplot(2,2,3);

semilogx(f,Ang,f,Ang1,f,Ang2,f,Ang3);
grid on
legend("Courbe de dephasage wc=50","Courbe de dephasage
wc1=2*pi*500","Courbe de dephasage wc2=2*pi*1000","Courbe de dephasage
wc3=2*pi*50");
xlabel("f");
ylabel("« angle(H(jw)) »);
```

4) Choisissez wc qui vous semble optimal. Le filtre est-il bien choisi ? Pourquoi?

On a appliqué sur le signal la transmittance complexe de fréquence 500 on peut remarqué une atténuation sur tout le signal mais elle est plus importante dans les basses fréquence.

5) Observez le signal $y(t)$ obtenu

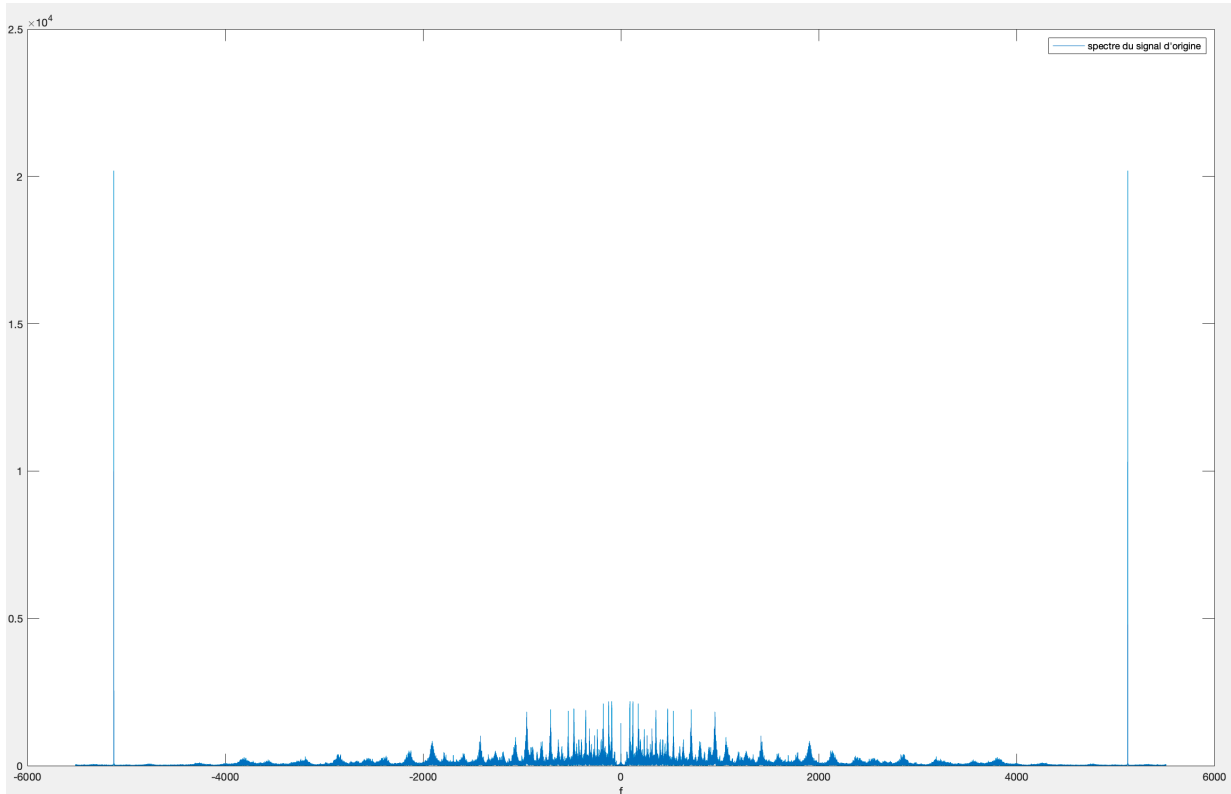


Dé-bruitage d'un signal sonore

1) Proposer une méthode pour supprimer ce bruit sur le signal.

filtrage

2) Mettez-la en oeuvre :



Code :

```
fc = 5000;  
K = 1;  
  
H = K./(1+1i*(f/fc).^100);  
Hpass=[H(1:floor(N/2)),flip(H(1:floor(N/2)))];  
  
y_filtre = spectre_music(1:end-1).*Hpass;  
sig_filtred= ifft(y_filtre,"symmetric");  
plot(fshift(1:end-1),fftshift(abs(fft(sig_filtred))))  
legend("spectre du signal après filtrage");  
xlabel("f");  
ylabel("A");
```

3) Quelles remarques pouvez-vous faire notamment sur la sonorité du signal final.

le filtre analogique ne fait que diminué le bruit et comme on a vu précédemment lors de l'atténuation du bruit dans l'autre signal on arrive pas à filtrer le signal avec une transmittance complexe d'ordre 1 sans affecter le signal. Donc un filtre pass bas de premier ordre ne sera pas efficace pour éliminer le bruit

4) Améliorer la qualité de filtrage en augmentant l'ordre du filtre.

