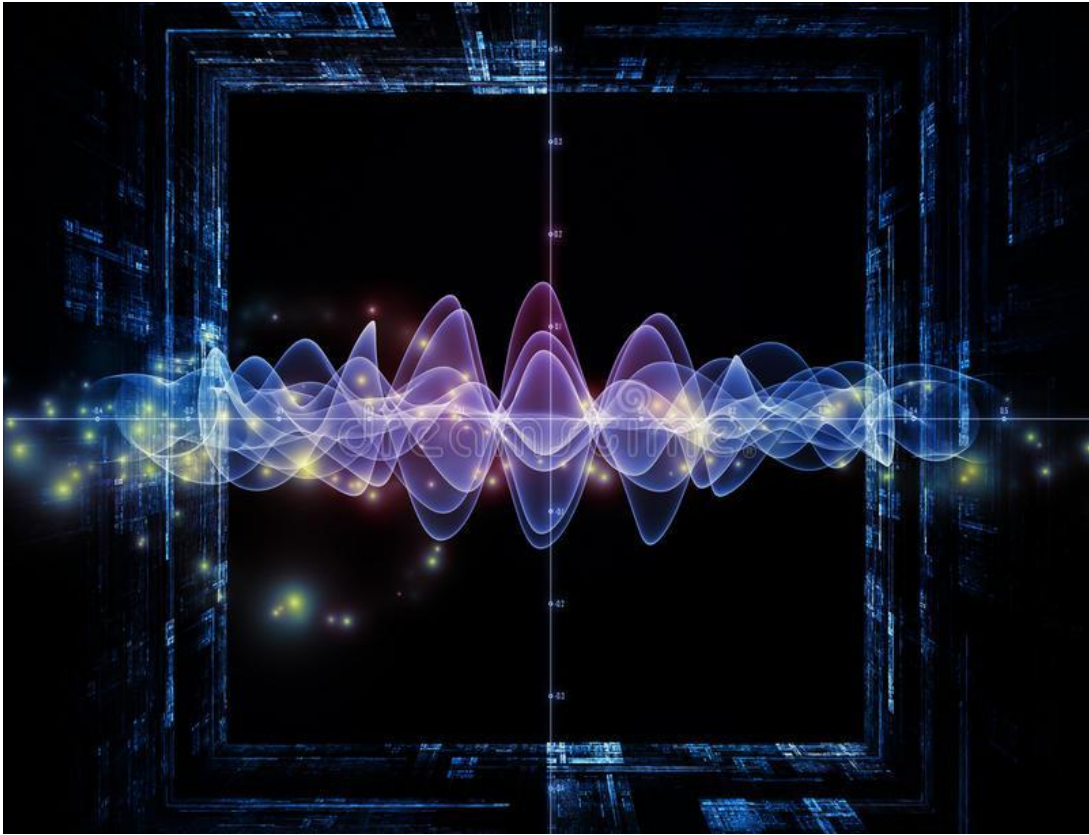


TP4 : Filtrage Analogique :



Réaliser par : Youssef laamari

Module : Traitement de signal

Prof : AMMOUR Alae

1-

```
fs = 1/0.0001; % Fréquence d'échantillonnage
t = 0:0.0001:5; % Intervalle de temps
f1 = 500; % Fréquence 1
f2 = 400; % Fréquence 2
f3 = 50; % Fréquence 3
x = sin(2*pi*f1*t) + sin(2*pi*f2*t) +
sin(2*pi*f3*t); % Signal d'entrée
```

2-

Code :

```
clear all
close all
clc
```

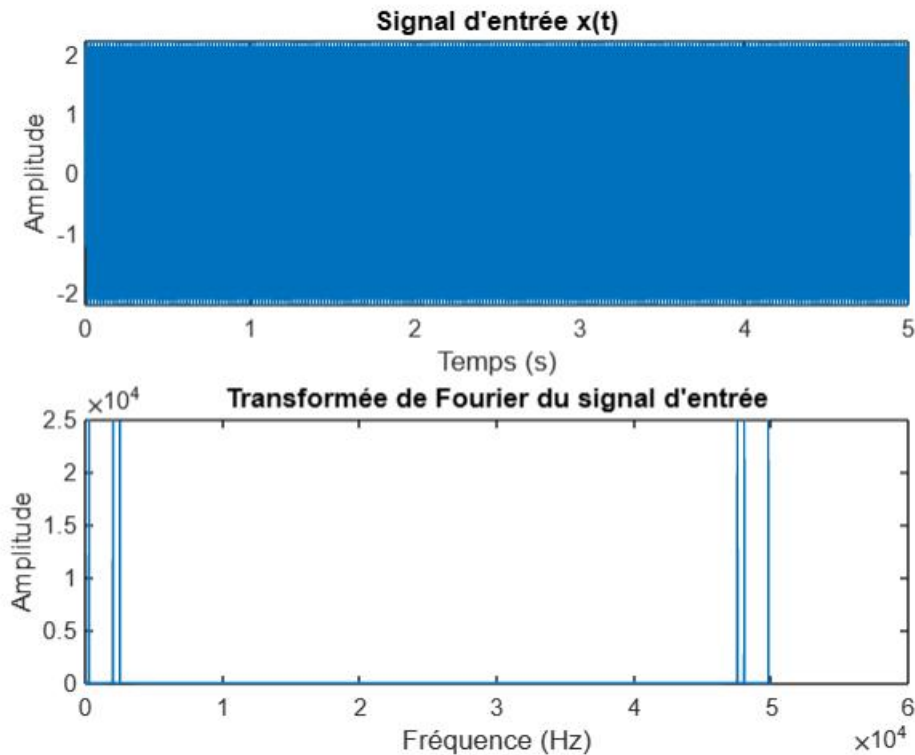
```
fs = 1/0.0001; % Fréquence d'échantillonnage
t = 0:0.0001:5; % Intervalle de temps
f1 = 500; % Fréquence 1
f2 = 400; % Fréquence 2
f3 = 50; % Fréquence 3
x = sin(2*pi*f1*t) + sin(2*pi*f2*t) +
sin(2*pi*f3*t); % Signal d'entrée
```

```
% Tracer le signal d'entrée x(t)
figure
subplot(2,1,1)
plot(t,x)
xlabel('Temps (s)')
ylabel('Amplitude')
title('Signal d'entrée x(t)')
```

```
% Calculer la transformée de Fourier
xf = fft(x);
```

```
% Tracer la transformée de Fourier
subplot(2,1,2)
plot(abs(xf))
xlabel('Fréquence (Hz)')
```

```
ylabel('Amplitude')
title('Transformée de Fourier du signal d\'entrée')
```



Tache 2 :

1)

Code :

```
K = 1;
wc = 50;
w = logspace(-2,2,1000);
H = K*w./(w+wc);
```

```
figure(2)
semilogx(w,abs(H),'LineWidth',2)
xlabel('Fréquence (rad/s)')
ylabel('Module')
title('Diagramme de Bode du filtre passe-haut')
grid
```

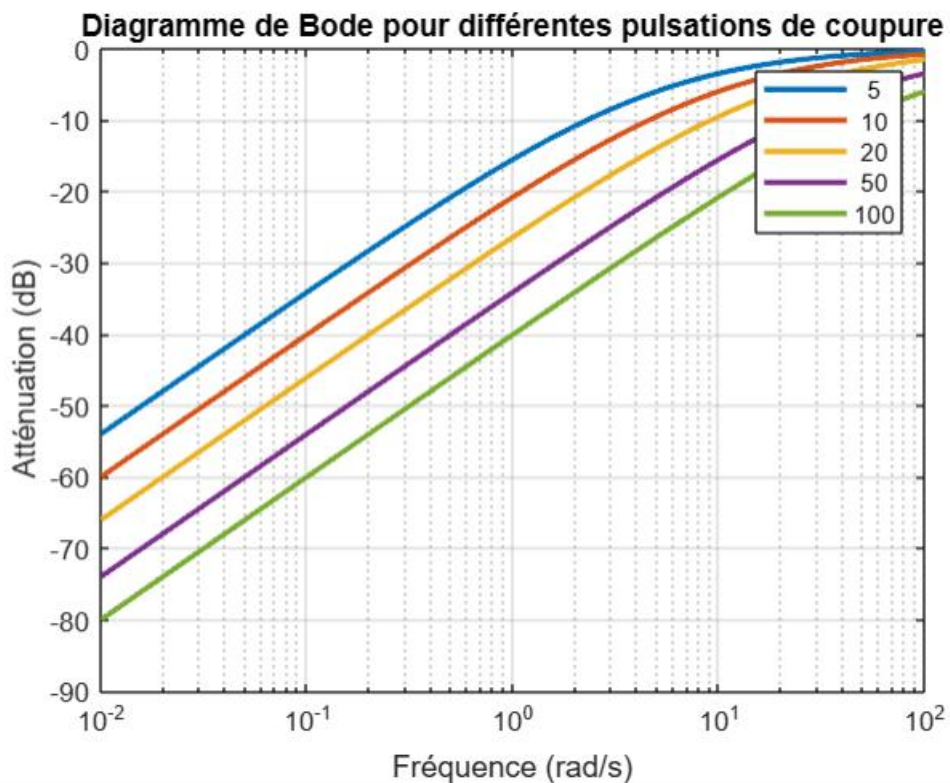
2)

Code :

```

figure(3)
wc = [5,10,20,50,100];
for i = 1:length(wc)
H = K*w./(w+wc(i));
semilogx(w, 20*log10(abs(H)),'LineWidth',2)
hold on
end
xlabel('Fréquence (rad/s)')
ylabel('Atténuation (dB)')
title('Diagramme de Bode pour différentes pulsations de coupure')
legend(num2str(wc'))
grid

```



Dé-bruitage d'un signal sonor :

1)

```
% Charger le signal audio
[x, fs] = audioread('test.wav');
```

2)

```
% Définir la fréquence de coupure du filtre passe-bas
fc = 10000; % Hz
```

```
% Appliquer le filtre passe-bas
[b, a] = butter(2, fc/(fs/2), 'low');
y = filter(b, a, x);
```

```
% Enregistrer le signal filtré
audiowrite('signal_filtre.wav', y, fs);
```

3)

Le paramètre K du filtre est le gain du filtre. Il permet de régler l'amplitude du signal filtré par rapport au signal d'entrée. Plus K est élevé, plus le signal filtré sera amplifié.

4)

```
[b, a] = butter(4, fc/(fs/2), 'low');
y = filter(b, a, x);
```

