CHATIRON Thibault

DENG Fangzhou

SRT3

Automne 2013

# TP n°3 :

# Multiplexage

*SY06*

## Sommaire

## Introduction :

L’objectif de ce TP est de montrer comment optimiser l’occupation fréquentielle d’un signal grâce à un multiplexage en phase. Contrairement à une modulation classique, le multiplexage permet d’émettre deux signaux sur une même bande de fréquence.

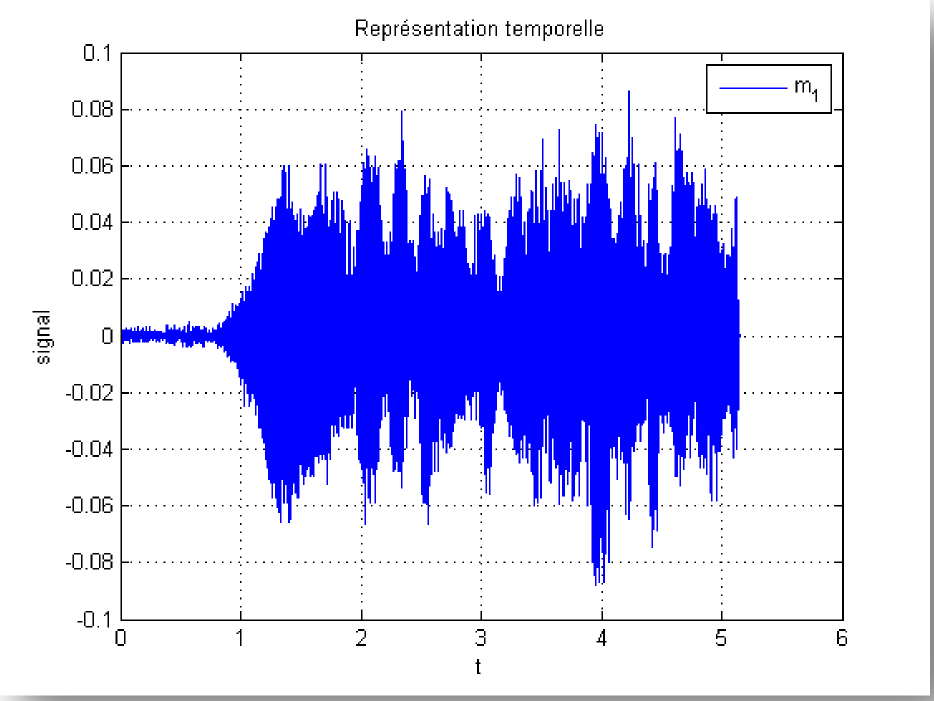
On va voir comment dissocier deux signaux possédant la même plage de fréquence, mais une phase différente. Ils seront ici en quadrature de phase pour plus de lisibilité.

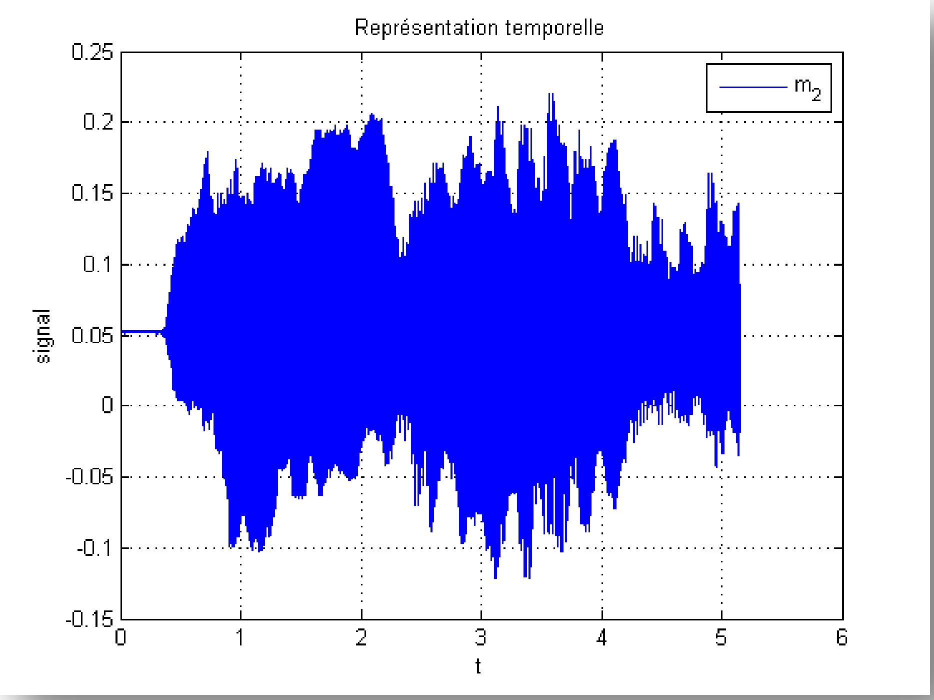
## Etude des exemples dans le domaine temporel

On commence le TP par déclarer les variables et charger nos deux exemples qui sont des sons, extrait de musique de Bach.
On commence simplement par représenter ces signaux dans le domaine temporel.

**Code Matlab**

```
clc
clear
close all

load('son1');
load('son2');

f0 = 10000;
Fe = fe;
m1 = vX1;
m2 = vX2;

N = length(vX1);
t = (0:1/Fe:(N-1)/Fe)';

figure(1);
plot(t,m1);
xlabel('t');
ylabel('signal');
grid on;
legend('m_1');
title('Représentation temporelle');

N = length(vX2);
t = (0:1/Fe:(N-1)/Fe)';

figure(2);
plot(t,m2);
xlabel('t');
ylabel('signal');
grid on;
legend('m_2');
title('Représentation temporelle');
```


Représentation temporelle
m1
signal
t
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
0
1
2
3
4
5
6

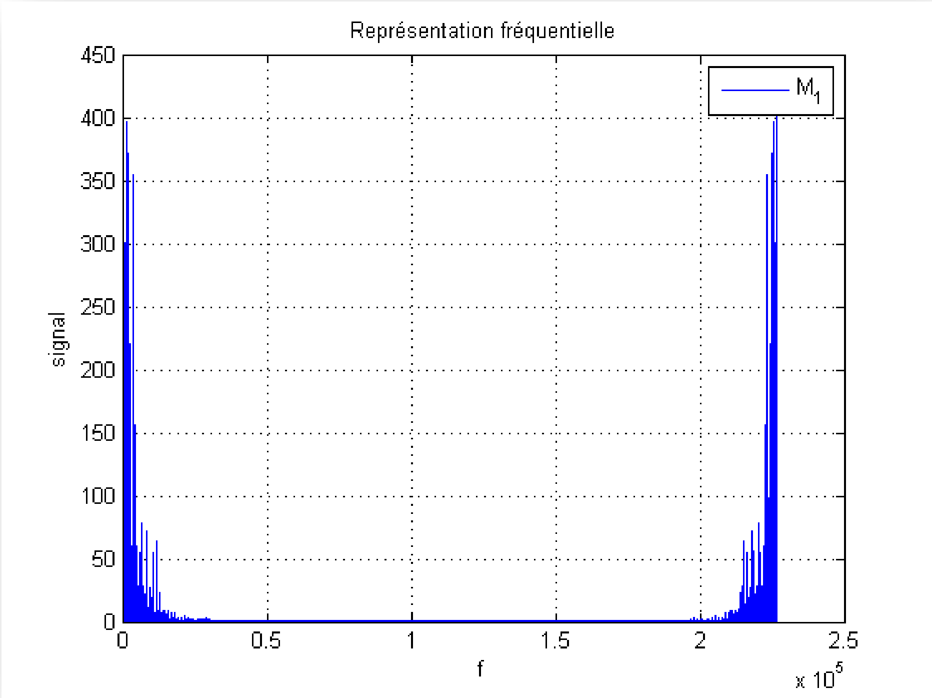

Représentation temporelle
m2
signal
t
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
0
1
2
3
4
5
6

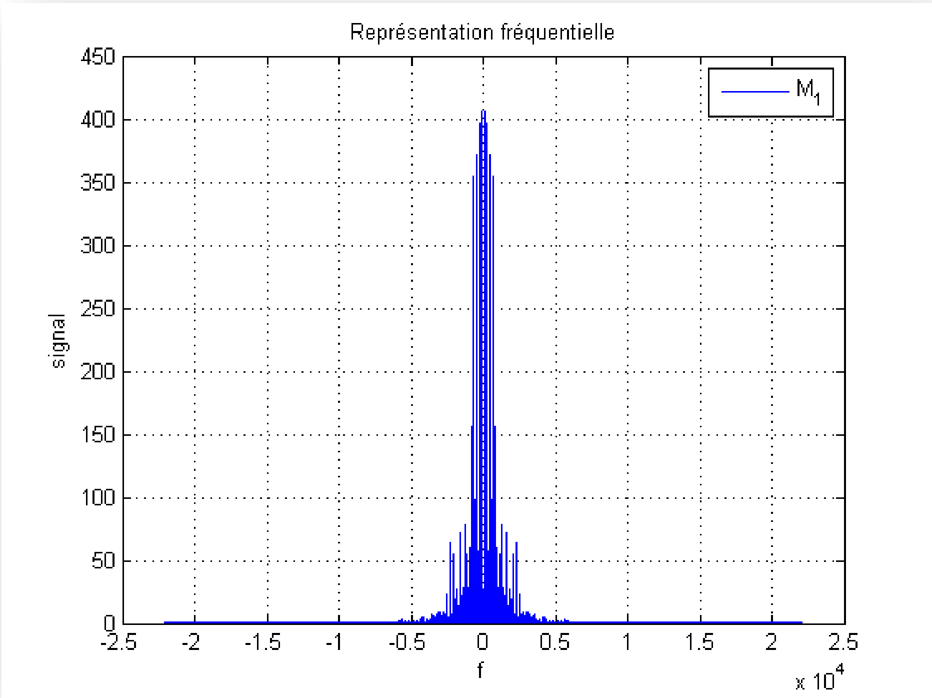
## Etude des exemples dans le domaine fréquentiel

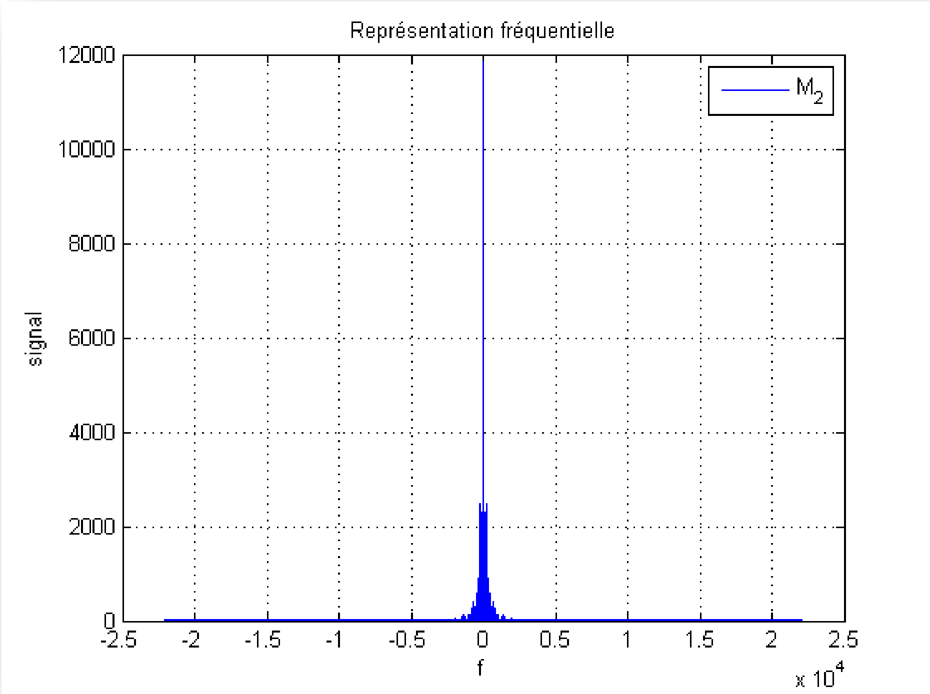
Après avoir étudié nos signaux sonores dans le domaine temporel, nous faisons de même dans le domaine fréquentiel.
Le premier graphique n'est pas totalement exploitable, c'est donc les deux graphiques d'après (obtenu avec les fonctions *linspace* et *fftshift*) qui nous permettra d'utiliser au mieux les éléments du TP et d'obtenir une meilleure démodulation à la suite.

**Code Matlab**

```
N = length(m1);
vecF = (-Fe/2:Fe/N:Fe/2);

M1 = fft(m1);

figure(3);
plot(abs(M1));
xlabel('f');
ylabel('signal');
grid on;
legend('M_1');
title('Représentation fréquentielle');

figure(4);
plot(linspace(-Fe/2,Fe/2,N),fftshift(abs(M1)));
xlabel('f');
ylabel('signal');
grid on;
legend('M_1');
title('Représentation fréquentielle');

M2 = fft(m2);

figure(5);
plot(linspace(-Fe/2,Fe/2,N),fftshift(abs(M2)));
xlabel('f');
ylabel('signal');
grid on;
legend('M_2');
title('Représentation fréquentielle');
```


Représentation fréquentielle
M1
signal
f
x 10^5
0
0.5
1
1.5
2
2.5
450
400
350
300
250
200
150
100
50


Représentation fréquentielle
M1
signal
f
x 10^4
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
450
400
350
300
250
200
150
100
50

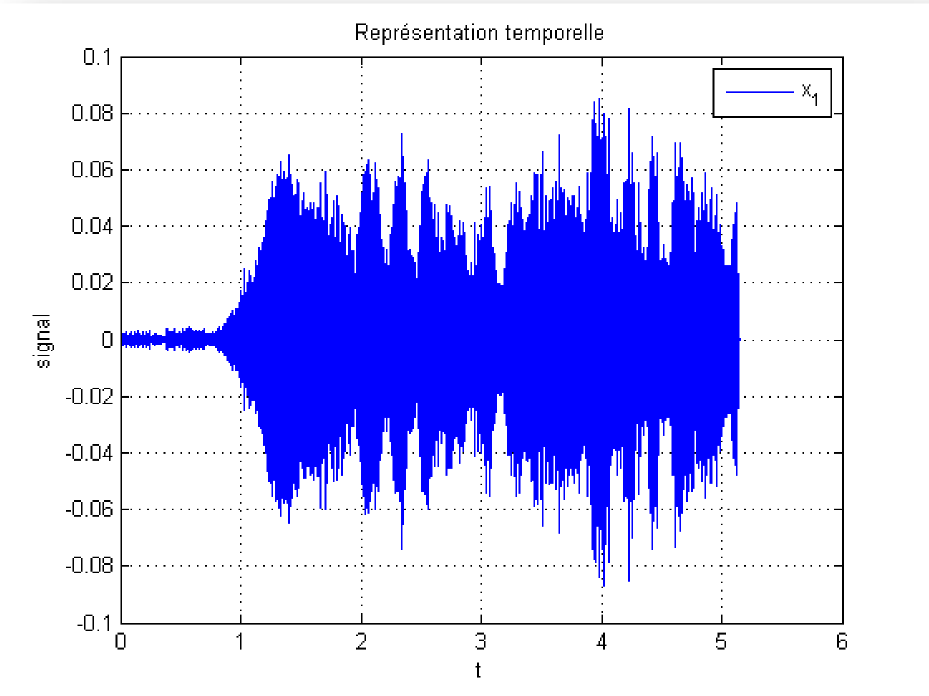

Représentation fréquentielle
M2
signal
f
x 10^4
12000
10000
8000
6000
4000
2000
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5

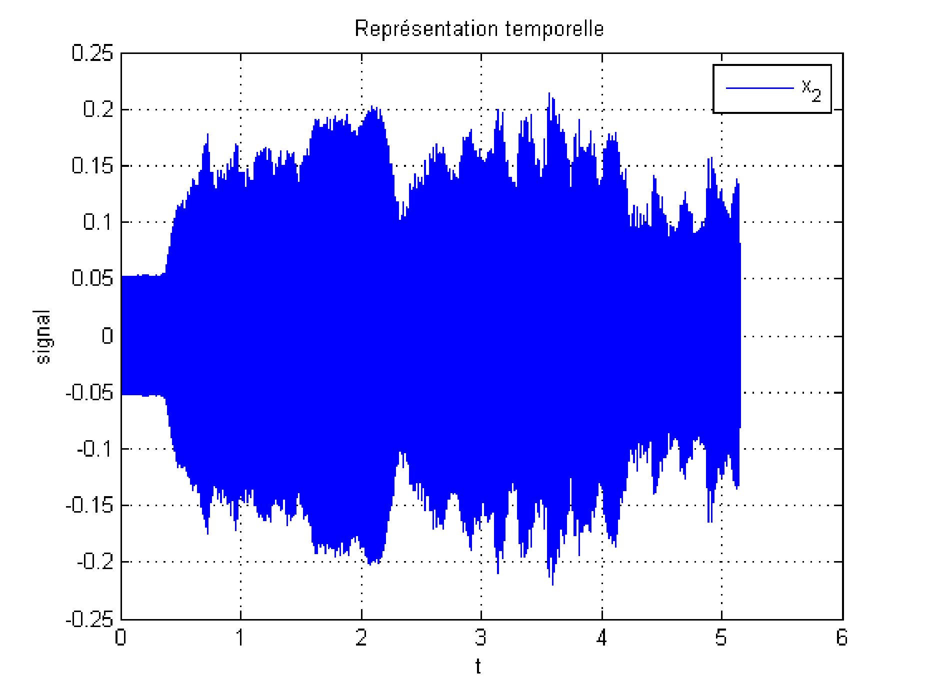
## Etude des signaux $x_1(t)$ et $x_2(t)$

Les signaux $x_1$ et $x_2$ représentent les signaux modulés, c'est-à-dire que l'on multiplie nos signaux initiaux $m_1$ et $m_2$ par respectivement cos(2*pi*f0*t) et sin(2*pi*f0*t).
On va à la suite tracer leur représentation temporelle.

**Code Matlab**

```
x1 = m1.*cos(2*pi*f0*t);

figure(6);
plot(t,x1);
xlabel('t');
ylabel('signal');
grid on;
legend('x_1');
title('Représentation temporelle');*

x2 = m2.*sin(2*pi*f0*t);

figure(7);
plot(t,x2);
xlabel('t');
ylabel('signal');
grid on;
legend('x_2');
title('Représentation temporelle');
```





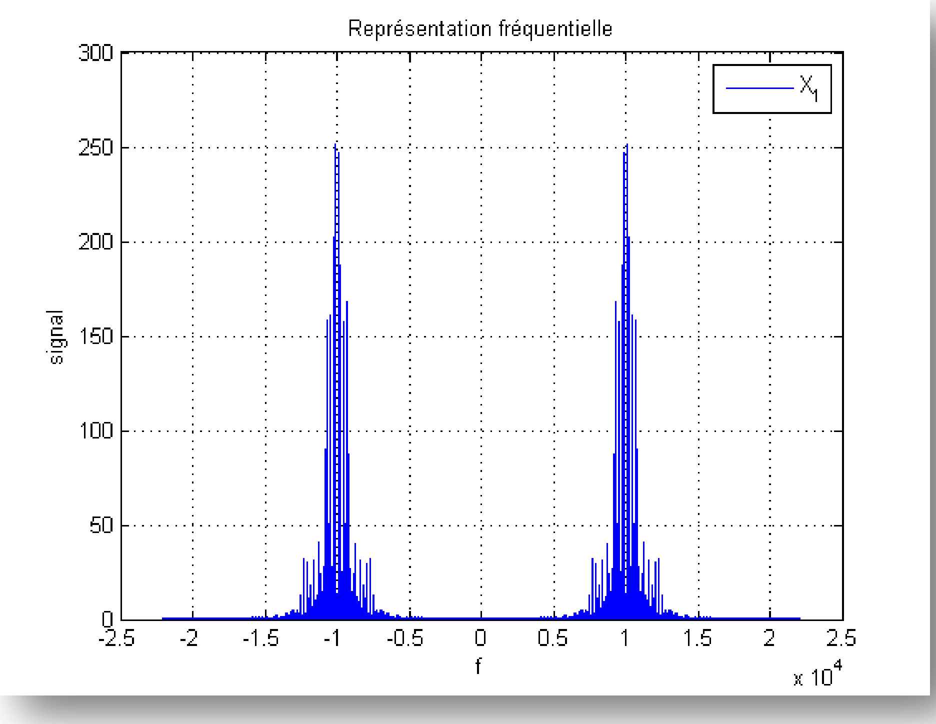

## Etude des signaux $X_1(f)$ et $X_2(f)$

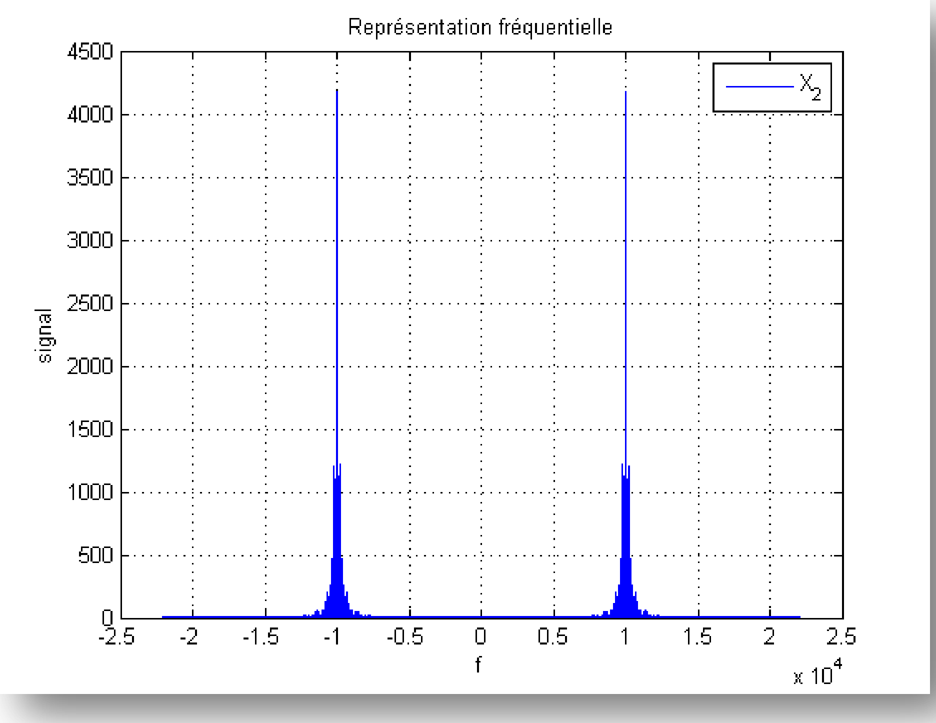
Afin d'observer notre signal d'un point de vu fréquentiel, on calcule la transformée de Fourier des deux signaux $x_1$ et $x_2$.

Les deux signaux modulés obtenus ont des pics en fréquence f0 et –f0 c'est-à-dire 10 kHz. Les deux signaux ont bien les mêmes fréquences d'utilisation une fois modulé.
Comme on module le premier signal par un cosinus à une fréquence f0 et le second signal par un sinus à une fréquence f0, alors les signaux $m_1$ et $m_2$ sont en quadrature de phase.

**Code Matlab**

```
X1 = fft(x1);
X2 = fft(x2);

figure(8);
plot(linspace(-Fe/2,Fe/2,N),fftshift(abs(X1)));
xlabel('f');
ylabel('signal');
grid on;
legend('X_1');
title('Représentation fréquentielle');

figure(9);
plot(linspace(-Fe/2,Fe/2,N),fftshift(abs(X2)));
xlabel('f');
ylabel('signal');
grid on;
legend('X_2');
title('Représentation fréquentielle');
```


Représentation fréquentielle
300
250
200
150
100
50
0
signal
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
f
x 10^4
X1

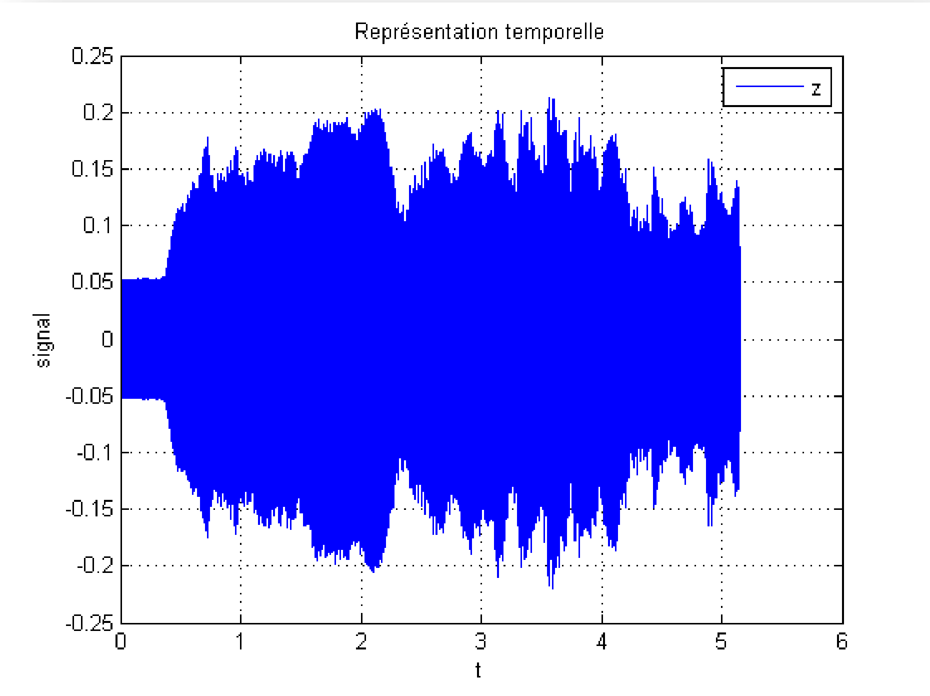

Représentation fréquentielle
4500
4000
3500
3000
2500
2000
1500
1000
500
0
signal
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
f
x 10^4
X2

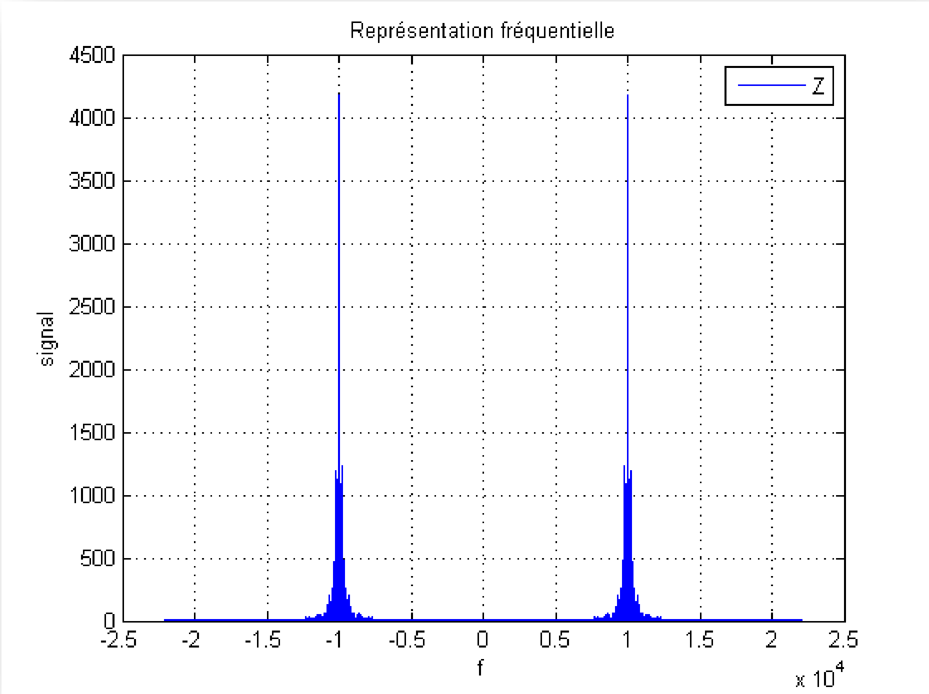
## Etude de la somme des deux signaux

Pour émettre les signaux, on réalisera simplement une addition des deux signaux que l'on représentera aussi bien dans le domaine temporel que fréquentiel.

**Code Matlab**

```
z = x1 + x2;

figure(10);
plot(t,z);
xlabel('t');
ylabel('signal');
grid on;
legend('z');
title('Représentation temporelle');

Z = fft(z);

figure(11);
plot(linspace(-Fe/2,Fe/2,N),fftshift(abs(Z)));
xlabel('f');
ylabel('signal');
grid on;
legend('Z');
title('Représentation fréquentielle');
```






## Démodulation

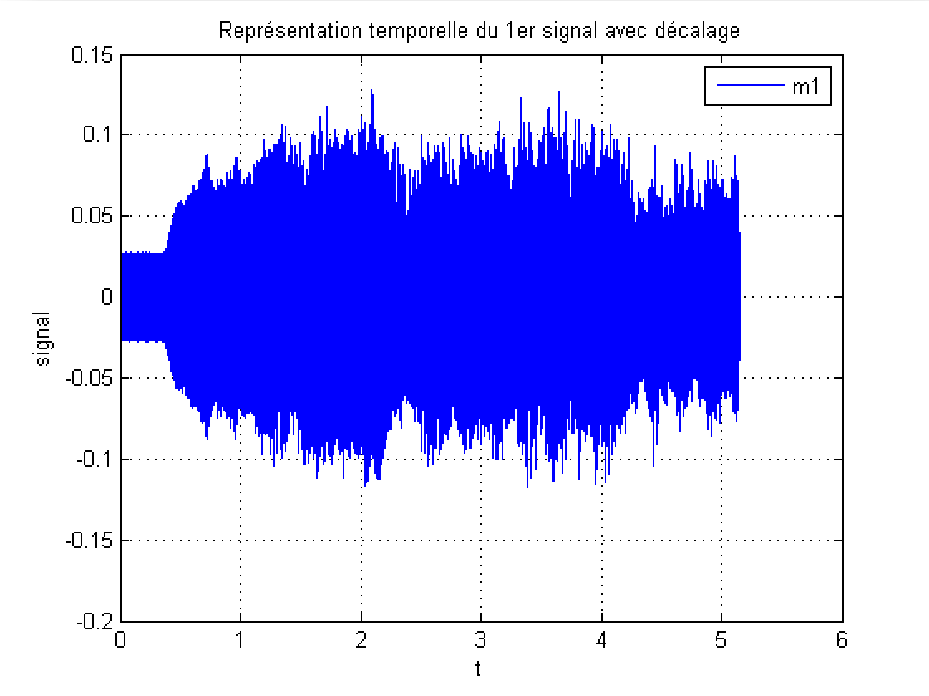
La démodulation consiste à séparer le signal reçu en deux signaux qui sont sur la même bande de fréquence en multipliant le signal reçu par les porteuses qui ont servies à les moduler précédemment, c'est-à-dire cos(2*pi*f0*t) et sin(2*pi*f0*t).

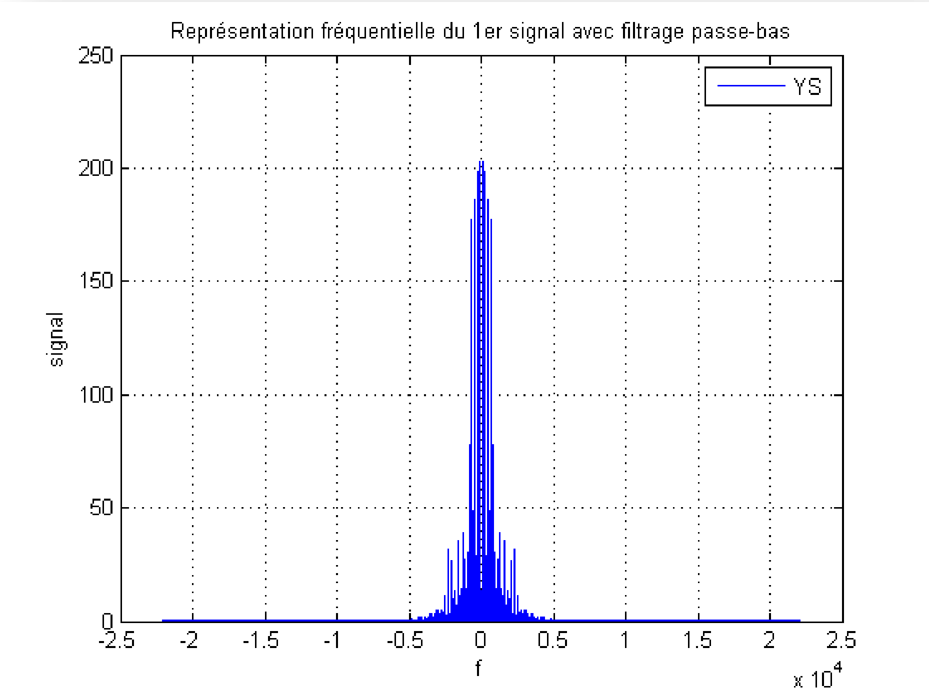
On applique un filtre passe-bas pour supprimer le spectre parasite généré par la démodulation.

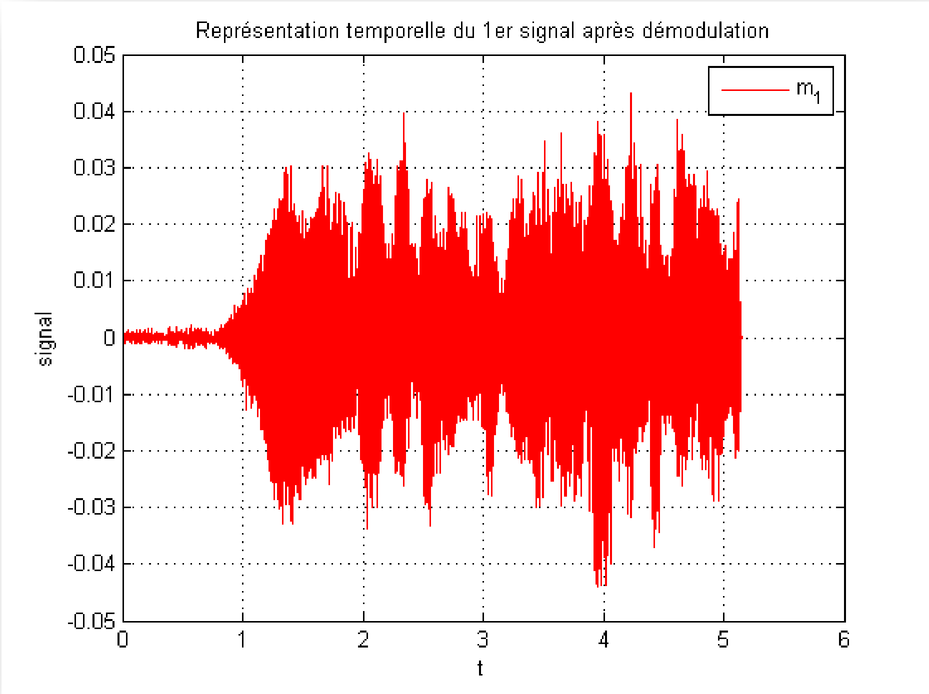
**Code Matlab**

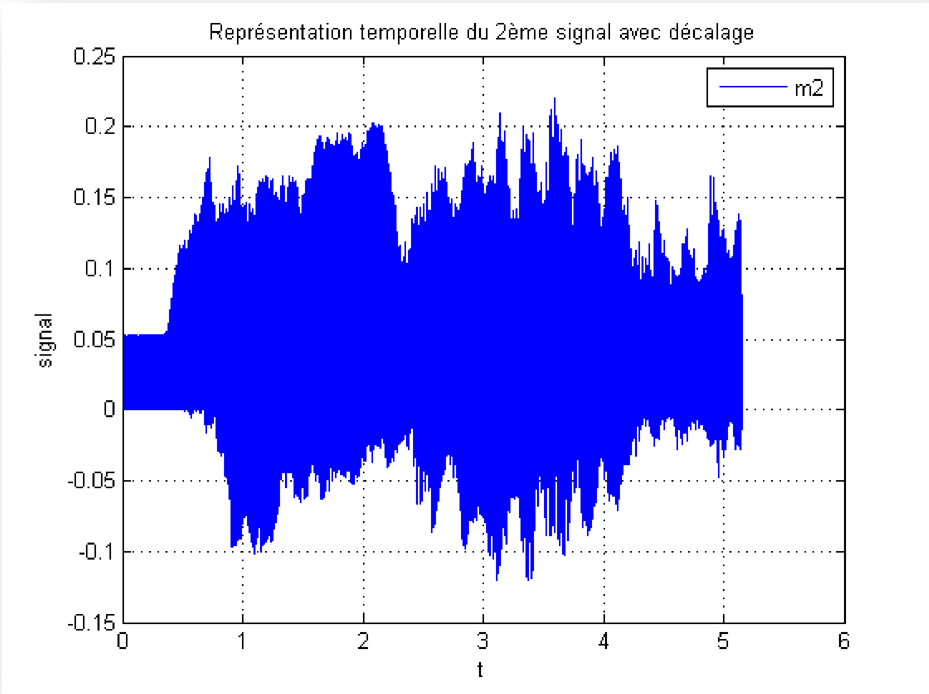
```
%1er signal

m11 = z.*cos(2*pi*f0*t);

figure(12);
plot(t,m11);
xlabel('t');
ylabel('signal');
grid on;
legend('m1');
title('Représentation temporelle du 1er signal avec décalage');

[vB,vA] = butter(8,5000/(Fe/2)); %filtre passe-bas
ys1 = filter(vB,vA,m11);
YS = fft(ys1);

figure(13);
```

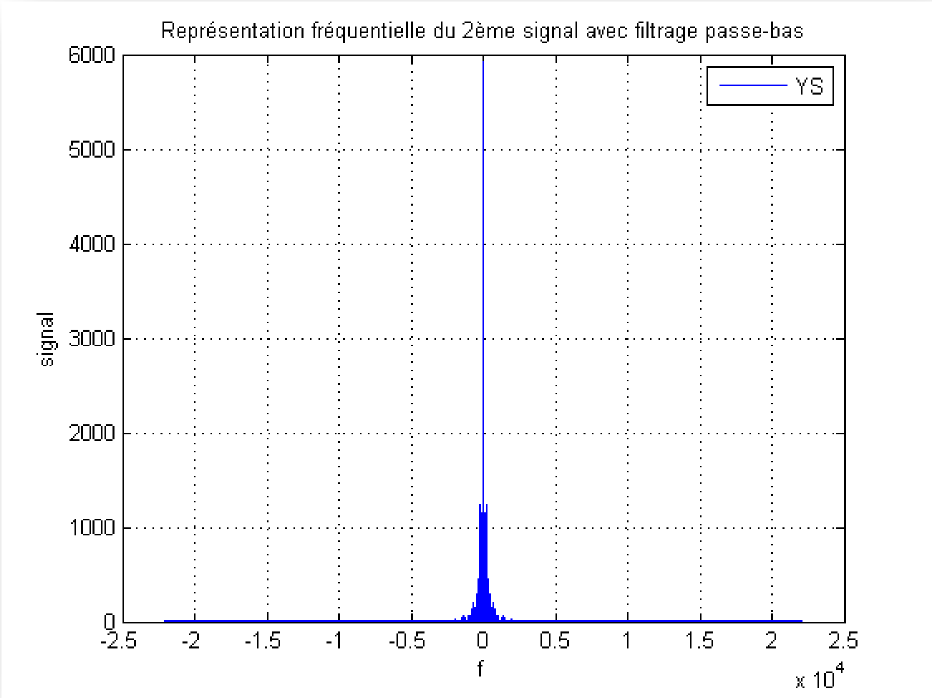
```
plot(linspace(-Fe/2,Fe/2,N),fftshift(abs(YS)));
xlabel('f');
ylabel('signal');
grid on;
legend('YS');
title('Représentation fréquentielle du 1er signal avec filtrage passe-
bas');

figure(14);
plot(t,ys1,'r');
xlabel('t');
ylabel('signal');
grid on;
legend('m_1');
title('Représentation temporelle du 1er signal après démodulation');

%2eme signal

m22 = z.*sin(2*pi*f0*t);

figure(15);
plot(t,m22);
xlabel('t');
ylabel('signal');
grid on;
legend('m2');
title('Représentation temporelle du 2ème signal avec décalage');

[vB,vA] = butter(8,5000/(Fe/2)); %filtre passe-bas
ys2 = filter(vB,vA,m22);
YS = fft(ys2);

figure(16);
plot(linspace(-Fe/2,Fe/2,N),fftshift(abs(YS)));
xlabel('f');
ylabel('signal');
grid on;
legend('YS');
title('Représentation fréquentielle du 2ème signal avec filtrage passe-
bas');

figure(17);
plot(t,ys2,'r');
xlabel('t');
ylabel('signal');
grid on;
legend('m_2');
title('Représentation temporelle du 2ème signal après démodulation');
```

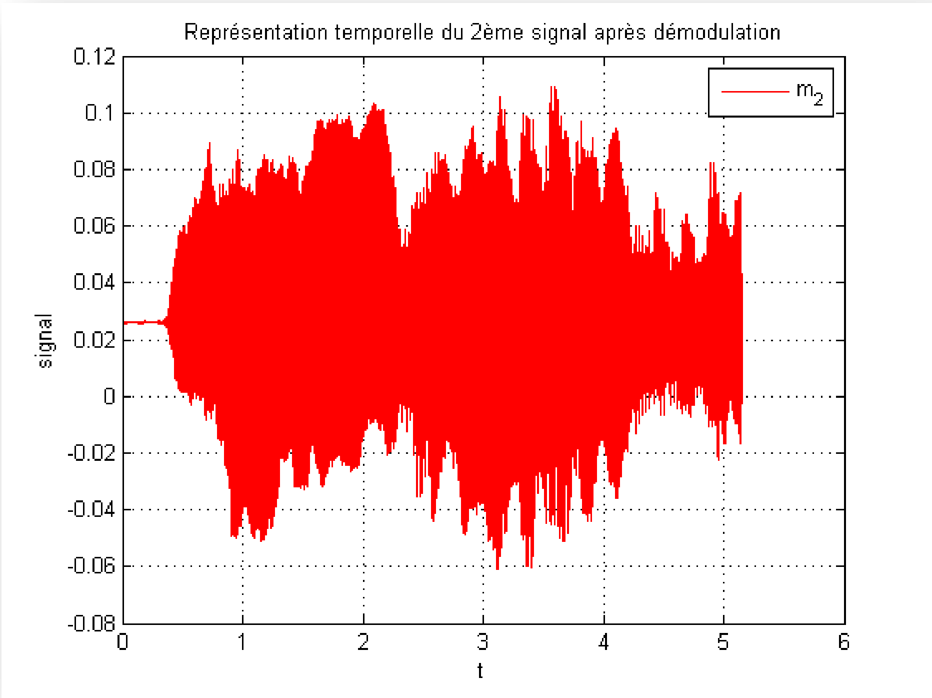

Représentation temporelle du 1er signal avec décalage
m1
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
signal
0
1
2
3
4
5
6
t


Représentation fréquentielle du 1er signal avec filtrage passe-bas
YS
250
200
150
100
50
0
signal
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
f
x 10^4


Représentation temporelle du 1er signal après démodulation
$m_1$
signal
t
0.05
0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
-0.05
0 1 2 3 4 5 6


Représentation temporelle du 2ème signal avec décalage
m2
signal
t
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
0 1 2 3 4 5 6


Représentation fréquentielle du 2ème signal avec filtrage passe-bas
YS
signal
6000
5000
4000
3000
2000
1000
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
f
x 10^4

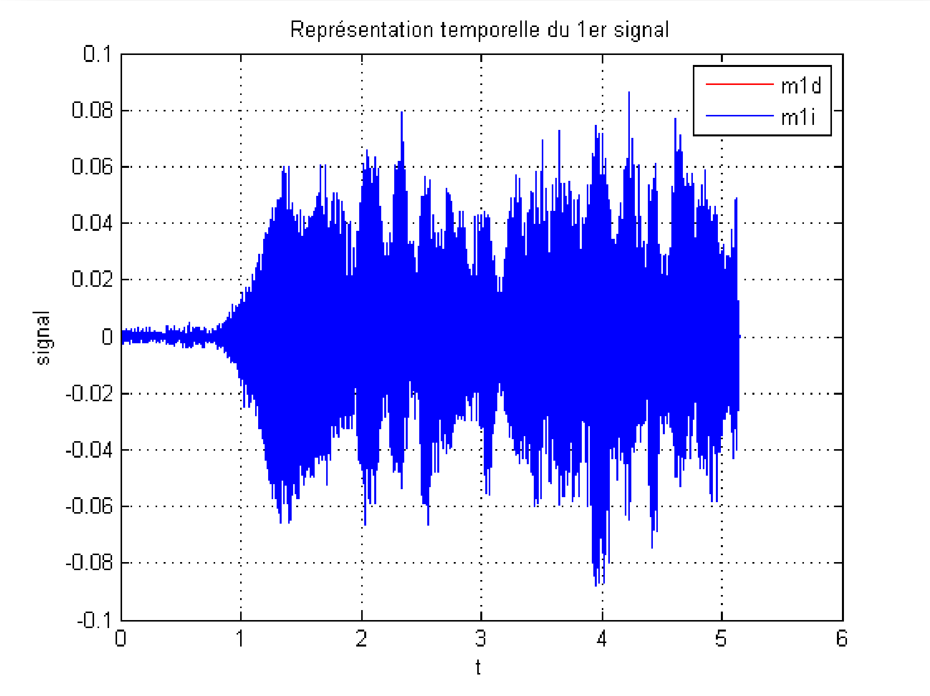

Représentation temporelle du 2ème signal après démodulation
m2
signal
0.12
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
0
1
2
3
4
5
6
t

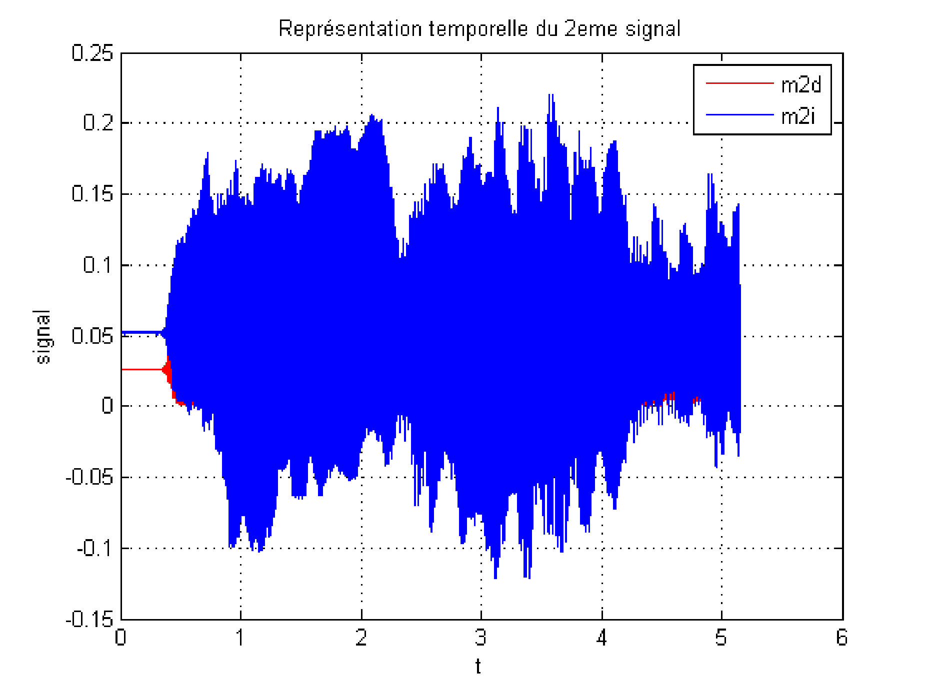
## Comparaison des signaux avant et après démodulation

On ne voit pas vraiment la différence entre les deux signaux mais si l'on fait un zoom on voit un retard et même une atténuation du signal.

**Code Matlab**

```
figure(18);
plot(t,ys1,'r', t,m1);
xlabel('t');
ylabel('signal');
grid on;
legend('m1d','m1i');
title('Représentation temporelle du 1er signal');

figure(19);
plot(t,ys2,'r', t,m2);
xlabel('t');
ylabel('signal');
grid on;
legend('m2d','m2i');
title('Représentation temporelle du 2eme signal');
```






## Densité interspectrale

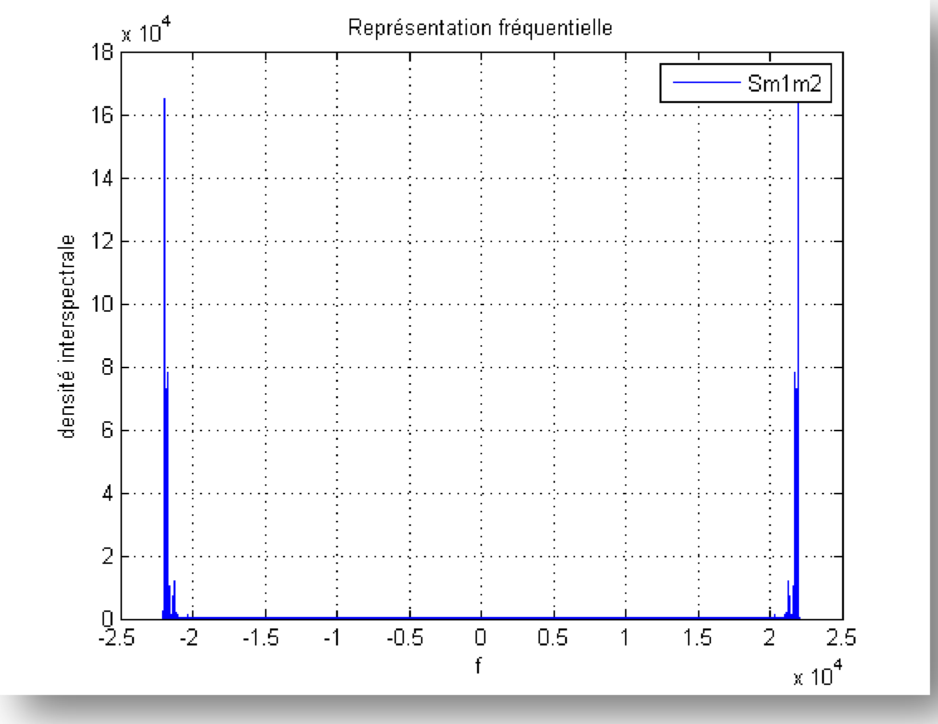
Afin de vérifier l'occupation fréquentielle des signaux, on calcule la densité spectrale d'énergie mutuelle des signaux m1 et m2. Pour cela, on applique la définition de densité spectrale mutuelle : `Sm1m2 = M1.*conj(M2)` avec conj(M2), le conjugué de M2.

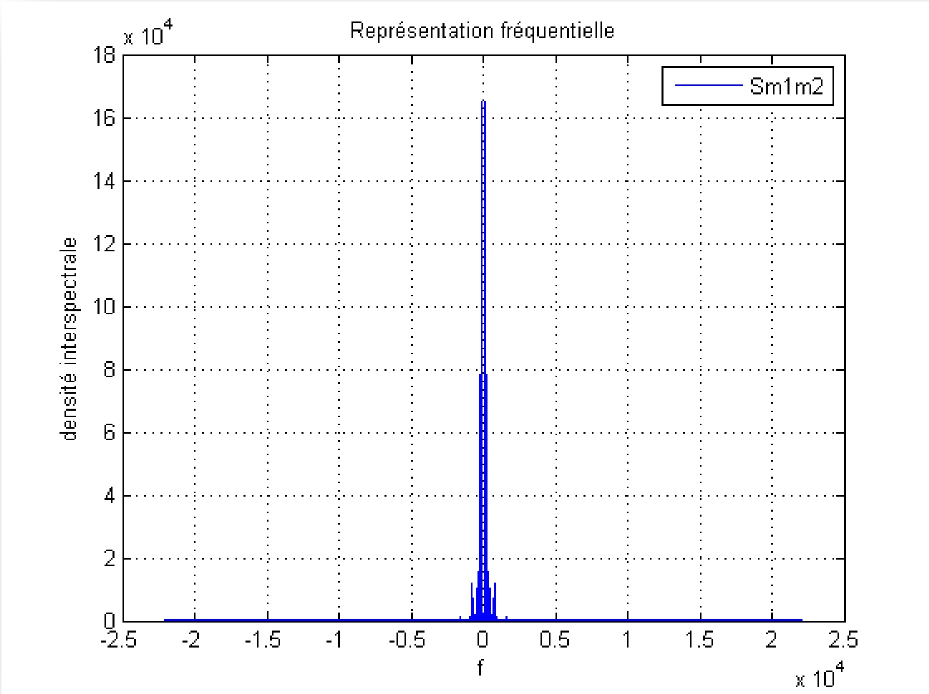
On constate bien que les deux signaux modulés occupent une bande de fréquence à 20kHz.

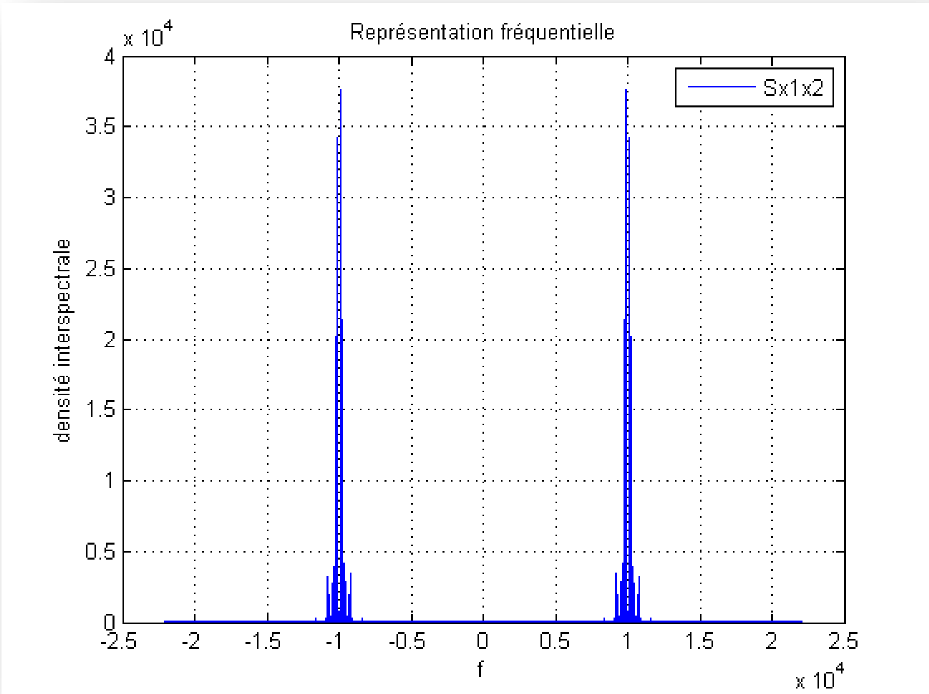
**Code Matlab**

```
Sm1m2 = M1.*conj(M2);

figure(20);
plot(linspace(-Fe/2,Fe/2,N),abs(Sm1m2));
xlabel('f');
ylabel('densité interspectrale');
grid on;
legend('Sm1m2');
title('Représentation fréquentielle');

figure(21);
plot(linspace(-Fe/2,Fe/2,N),fftshift(abs(Sm1m2)));
xlabel('f');
ylabel('densité interspectrale');
grid on;
legend('Sm1m2');
title('Représentation fréquentielle');

Sx1x2 = X1.*conj(X2);
```

```
figure(22);
plot(linspace(-Fe/2,Fe/2,N),fftshift(abs(Sx1x2)));
xlabel('f');
ylabel('densité interspectrale');
grid on;
legend('Sx1x2');
title('Représentation fréquentielle');
```




Représentation fréquentielle
x 10^4
18
16
14
12
10
8
6
4
2
0
densité interspectrale
Sm1m2
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
f
x 10^4


Représentation fréquentielle
x 10^4
4
3.5
3
2.5
2
1.5
1
0.5
0
densité interspectrale
Sx1x2
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
f
x 10^4

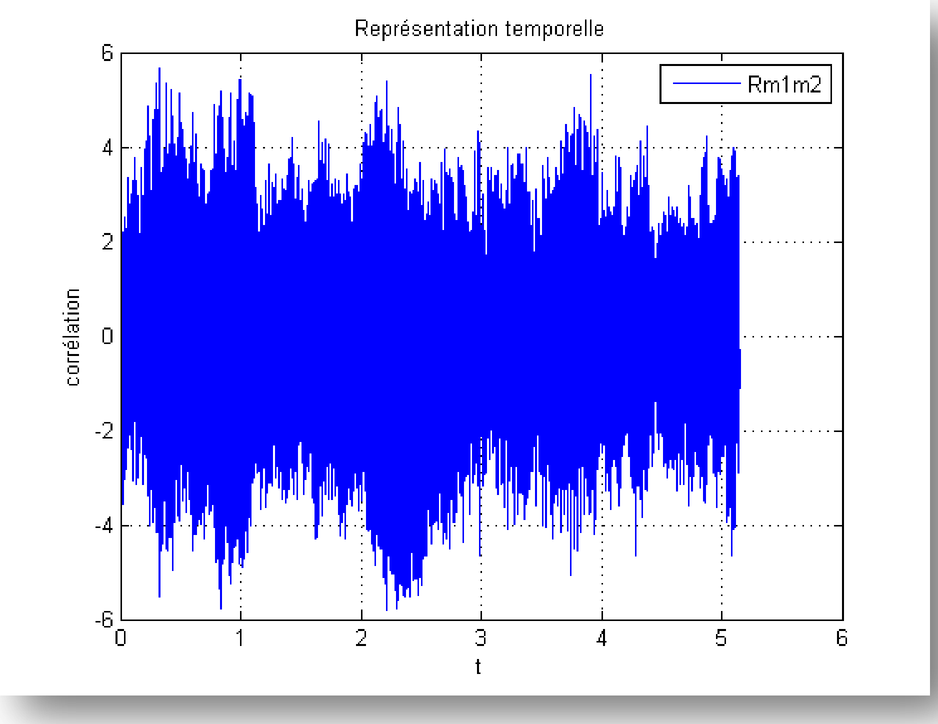
## Calcul de corrélation

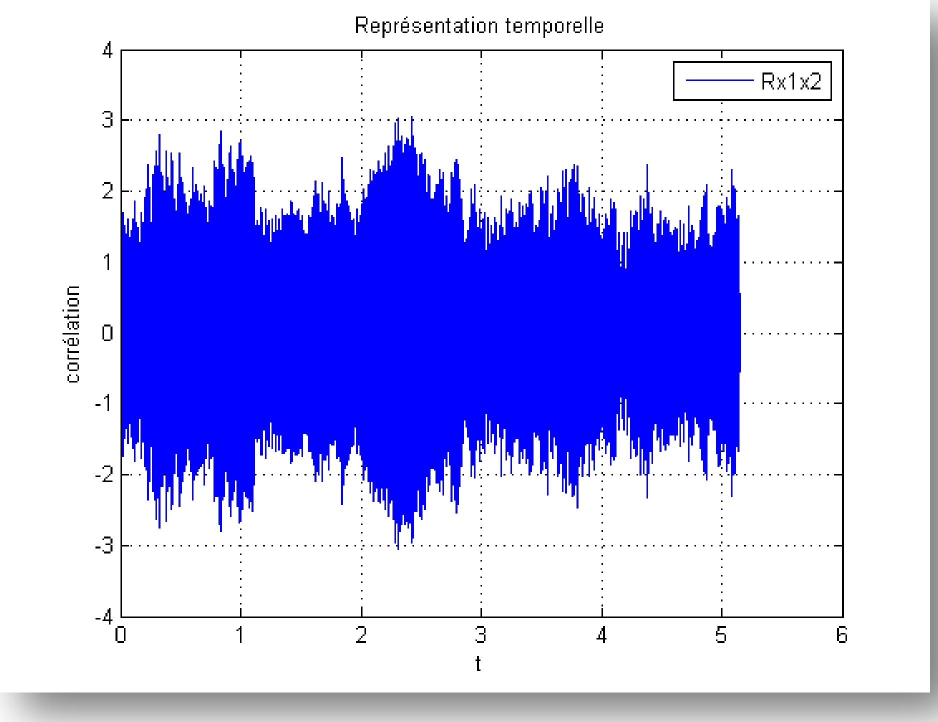
Pour vérifier la ressemblance entre les deux signaux, on utilise la fonction d'intercorrélation. Pour cela, on a utilisé deux manières. La première par définition, et donc que `Rm1m2` était égal à la Transformée de Fourier inverse de `Sm1m2`. La deuxième est obtenue avec la fonction *xcorr*.

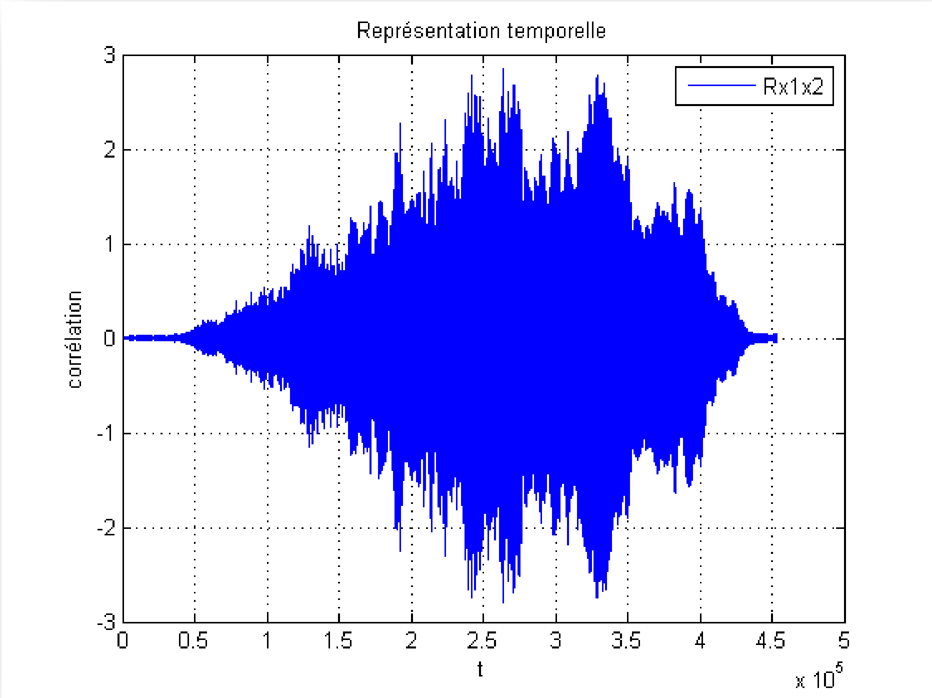
Grace à la représentation de l'intercorrélation des signaux modulés, on peut constater que les signaux sont similaires et occupent la même bande de fréquence.

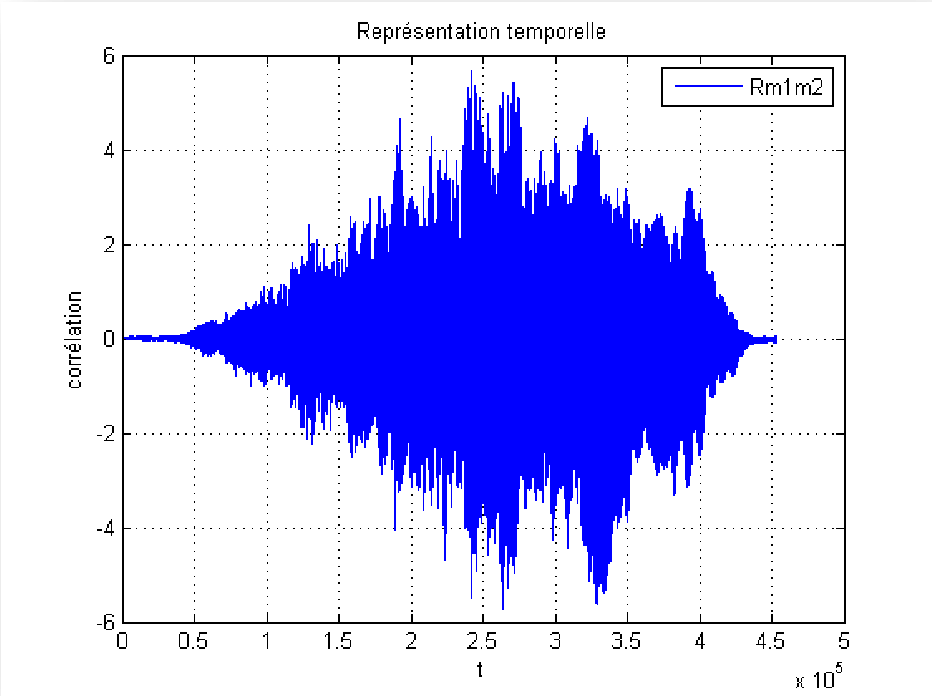
**Code Matlab**

```
Rm1m2 = ifft(Sm1m2);

figure(23);
plot(t,Rm1m2);
xlabel('t');
ylabel('corrélation');
grid on;
legend('Rm1m2');
title('Représentation temporelle');

Rx1x2 = ifft(Sx1x2);

figure(24);
plot(t,Rx1x2);
xlabel('t');
ylabel('corrélation');
grid on;
legend('Rx1x2');
title('Représentation temporelle');

Rx1x2 = xcorr(x1,x2);

figure(25);
plot(Rx1x2);
xlabel('t');
ylabel('corrélation');
grid on;
legend('Rx1x2');
title('Représentation temporelle');

Rm1m2 = xcorr(m1,m2);

figure(26);
plot(Rm1m2);
xlabel('t');
ylabel('corrélation');
grid on;
legend('Rm1m2');
title('Représentation temporelle');
```


Représentation temporelle
Rm1m2
corrélation
t
6
4
2
0
-2
-4
-6
0
1
2
3
4
5
6


Représentation temporelle
Rx1x2
corrélation
t
4
3
2
1
0
-1
-2
-3
-4
0
1
2
3
4
5
6


Représentation temporelle
Rx1x2
corrélation
t
x 10^5
3
2
1
0
-1
-2
-3
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


Représentation temporelle
Rm1m2
corrélation
t
x 10^5
6
4
2
0
-2
-4
-6
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

## Energie d'interaction

Ici, on calcule l'énergie d'interaction et on sait que l'on peut la calculer à partir de la fonction d'intercorrélation pour tau=0, on a E=0.

Code Matlab

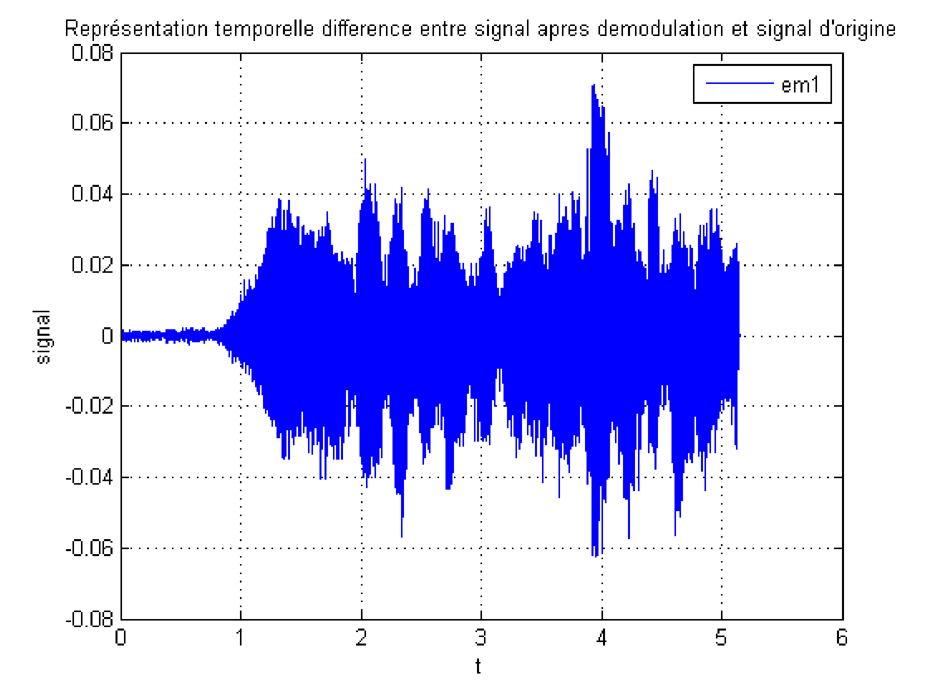
```
Ex1x2 = x1'*conj(x2);
Em1m2 = m1'*conj(m2);

% tau= 0 E=0
```

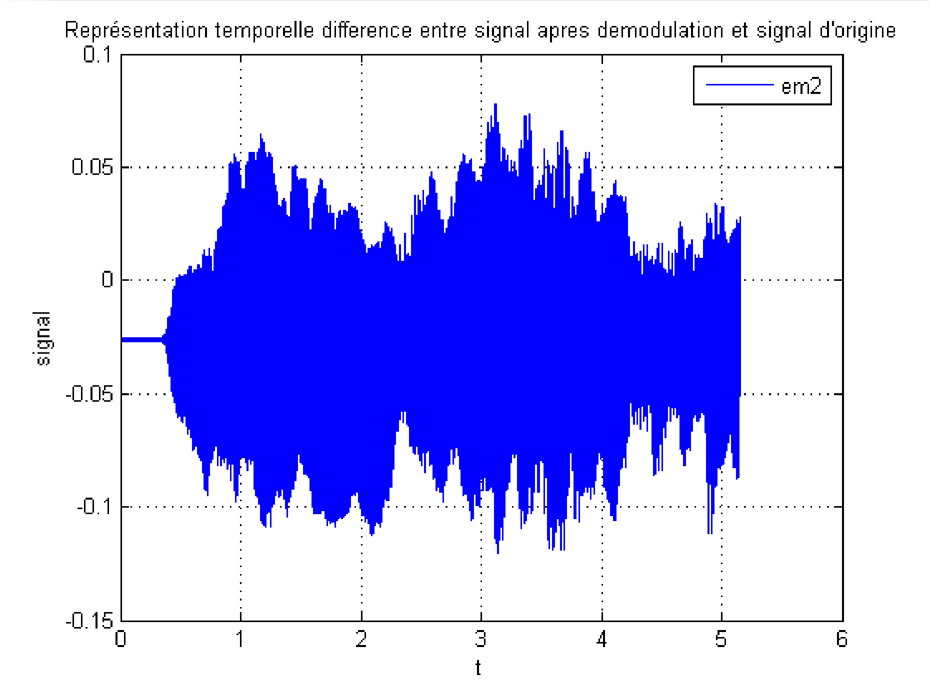
## Différence entre le signal démodulé et le signal d'origine

On peut voir si l'on représente la différence de ces deux signaux dans le domaine temporel qu'il y a une différence.
Pourtant, nous pensions obtenir une courbe quasi nulle car à l'écoute, le son était, à l'oreille en tout cas, en tout point identique.

Code Matlab

```
%em1 = sum(abs(ys1 - m1));
%em2 = sum(abs(ys2 - m2));

em1 = ys1 - m1;
em2 = ys2 - m2;

figure(27);
plot(t,em1);
xlabel('t');
ylabel('signal');
grid on;
legend('em1');
title('Représentation temporelle difference entre signal apres
demodulation et signal d'origine') ;

figure(28);
plot(t,em2);
xlabel('t');
ylabel('signal');
grid on;
legend('em2');
title('Représentation temporelle difference entre signal apres
demodulation et signal d'origine') ;
```


Représentation temporelle difference entre signal apres demodulation et signal d'origine
em1
signal
t
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
0
1
2
3
4
5
6


Représentation temporelle difference entre signal apres demodulation et signal d'origine
em2
signal
t
0.1
0.05
0
-0.05
-0.1
-0.15
0
1
2
3
4
5
6

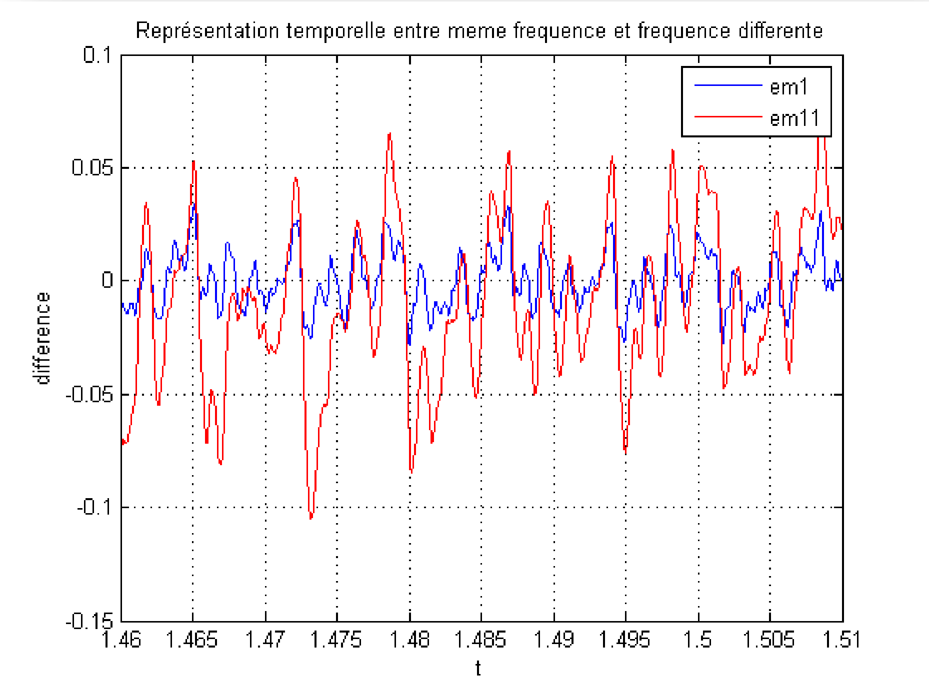
## Différence dans le cas où une fréquence différente est appliquée pour la démodulation

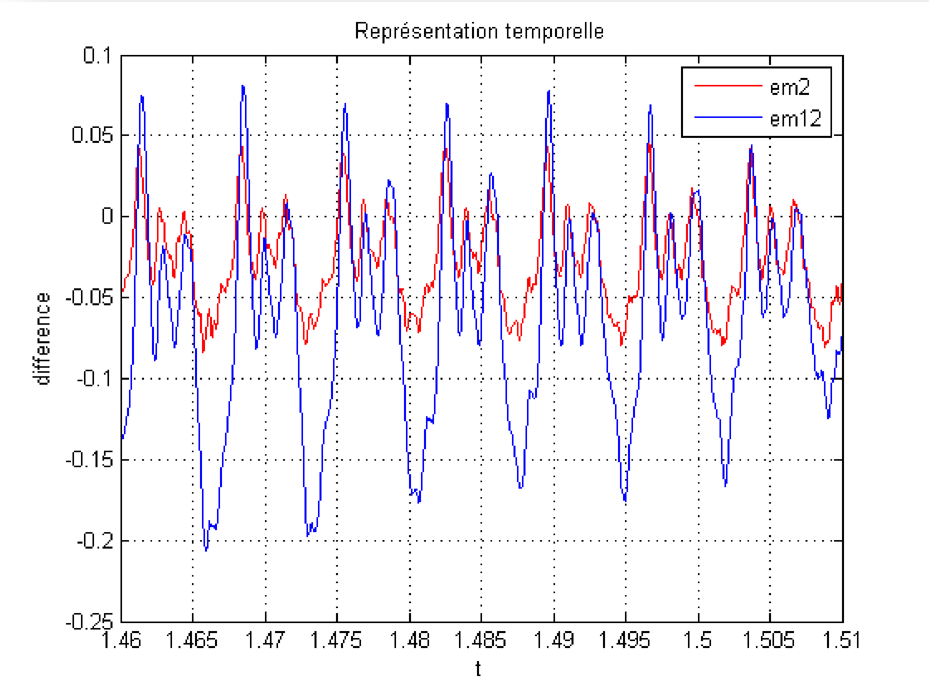
On a ici, décidé de changer la valeur de f0. Nous l'avons donc seulement augmenté de 5Hz. Nous avons tracé la représentation temporelle des différences des deux signaux avant et après le changement de fréquence.

Nous voyons qu'après changement, les amplitudes sont plus grandes. Et c'est acoustiquement que nous voyons vraiment la différence.

**Code Matlab**

```
f0 = 10005;

m11 = z.*cos(2*pi*f0*t);

[vB,vA] = butter(8,5000/(Fe/2)); %filtre passe-bas
ys1 = filter(vB,vA,m11);

m12 = z.*sin(2*pi*f0*t);

[vB,vA] = butter(8,5000/(Fe/2)); %filtre passe-bas
ys2 = filter(vB,vA,m11);

em11 = ys1 - m1;
em12 = ys2 - m2;

figure(29);
plot(t,em1, t,em11, 'r');
set(gca,'xlim',[1.46 1.51]);
xlabel('t');
ylabel('difference');
grid on;
legend('em1','em11');
title('Représentation temporelle entre meme frequence et frequence
differente');

figure(30);
plot(t,em2, 'r', t,em12);
set(gca,'xlim',[1.46 1.51]);
xlabel('t');
ylabel('difference');
grid on;
legend('em2','em12');
title('Représentation temporelle');
```

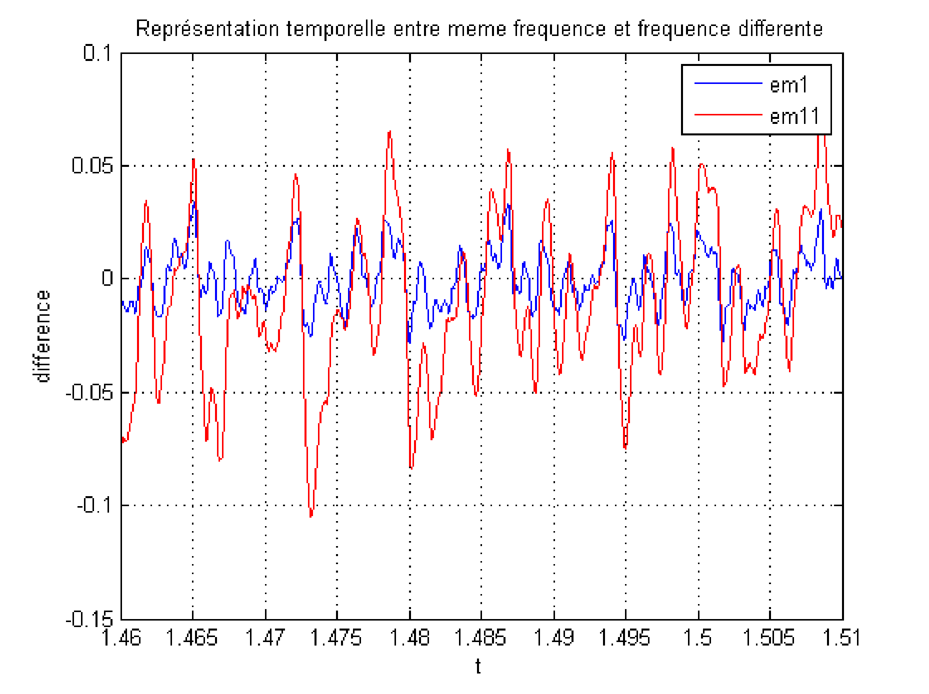

Représentation temporelle entre meme frequence et frequence differente
em1
em11
difference
t
0.1
0.05
0
-0.05
-0.1
-0.15
1.46
1.465
1.47
1.475
1.48
1.485
1.49
1.495
1.5
1.505
1.51

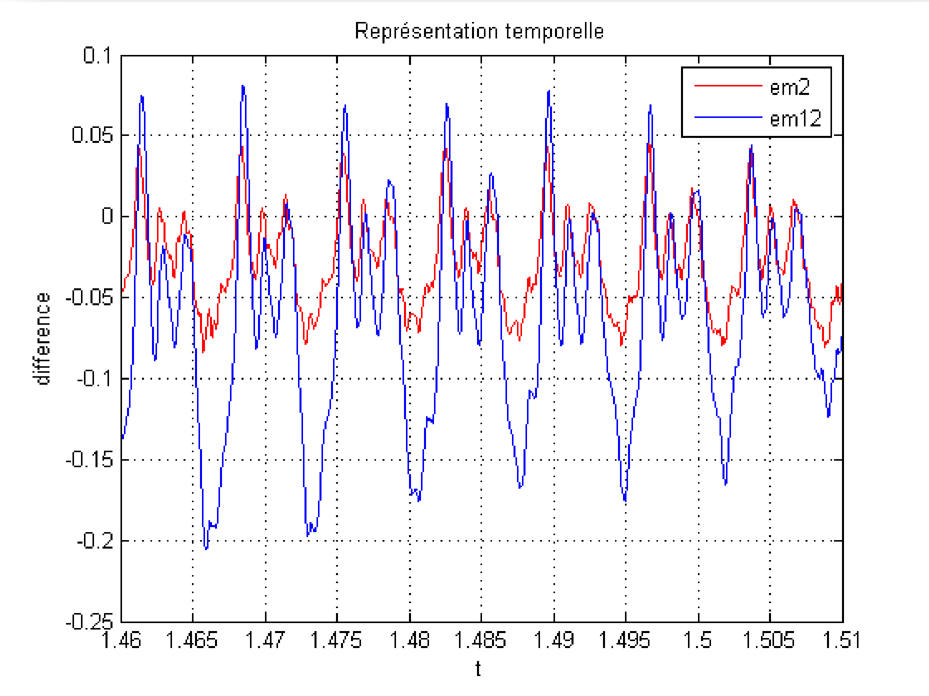

Représentation temporelle
em2
em12
difference
t
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
1.46
1.465
1.47
1.475
1.48
1.485
1.49
1.495
1.5
1.505
1.51

Ici, on a voulu ajouter en plus un déphasage de pi/500. On voit toujours une différence d'amplitude.

**Code Matlab**

```
psi = pi/500;

m11 = z.*cos(2*pi*f0*t+psi);

[vB,vA] = butter(8,5000/(Fe/2)); %filtre passe-bas
ys1 = filter(vB,vA,m11);
m12 = z.*sin(2*pi*f0*t+psi);

[vB,vA] = butter(8,5000/(Fe/2)); %filtre passe-bas
ys2 = filter(vB,vA,m11);

em11 = ys1 - m1;
em12 = ys2 - m2;

figure(31);
plot(t,em1, t,em11, 'r');
set(gca,'xlim',[1.46 1.51]);
xlabel('t');
ylabel('difference');
grid on;
legend('em1','em11');
title('Représentation temporelle entre meme frequence et frequence
differente');

figure(32);
plot(t,em2, 'r', t,em12);
set(gca,'xlim',[1.46 1.51]);
xlabel('t');
ylabel('difference');
grid on;
legend('em2','em12');
title('Représentation temporelle');
```


Représentation temporelle entre meme frequence et frequence differente
em1
em11
difference
t
0.1
0.05
0
-0.05
-0.1
-0.15
1.46 1.465 1.47 1.475 1.48 1.485 1.49 1.495 1.5 1.505 1.51


Représentation temporelle
em2
em12
difference
t
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
1.46 1.465 1.47 1.475 1.48 1.485 1.49 1.495 1.5 1.505 1.51

## Conclusion

Dans ce TP, nous avons démontré qu'il était tout à fait possible de transmettre de manière simultanée deux signaux occupant la même bande de fréquence. Ce procédé offre l'avantage d'optimiser la bande passante.

Pour séparer les signaux, il convient de multiplier la somme des signaux modulés (signal reçu) par les porteuses des signaux d'origine, puis de les filtrer (passe bas).

L'amplitude des signaux démodulés est deux fois inférieure à celle des signaux émis, cela est dû aux propriétés des porteuses.

Cependant, pour pouvoir utiliser le multiplexage en phase, il faut connaitre la porteuse et être cadencer avec elle. Sinon, le signal reçu est bruité à cause du déphasage ou une variation de la fréquence. On a donc acoustiquement, un son qui peut sembler plus lent.