

Modulation et démodulation  
en présence de bruit.

# TP RE02

Compte Rendu

BLAMPEY Florian – CHÂTIRON Thibault

---

## Contenu

Introduction .....	2
Exercice 1 .....	3
I. Le signal modulé .....	3
a. Représentation temporelle.....	3
b. Représentation fréquentielle.....	4
II. Le bruit .....	5
III. Le signal en présence de bruit .....	6
a. Représentation temporelle.....	6
b. Représentation fréquentielle.....	7
Exercice 2 .....	8
I. Les signaux démodulés .....	8
II. Estimation de l'amplitude et de la phase .....	9
III. Représentation sur Matlab .....	10
a. Représentation temporelle des signaux modulés $X(t)$ et $Y(t)$ .....	10
b. Représentation temporelle de l'amplitude .....	11
c. Représentation de l'erreur quadratique moyenne.....	12
Conclusion.....	14
Annexes.....	15
Exercice 1 .....	15
Exercice 2 .....	16

## Introduction

Durant ce TP, nous allons nous intéresser à la modulation et à la démodulation d'amplitude de signal en présence de bruit.

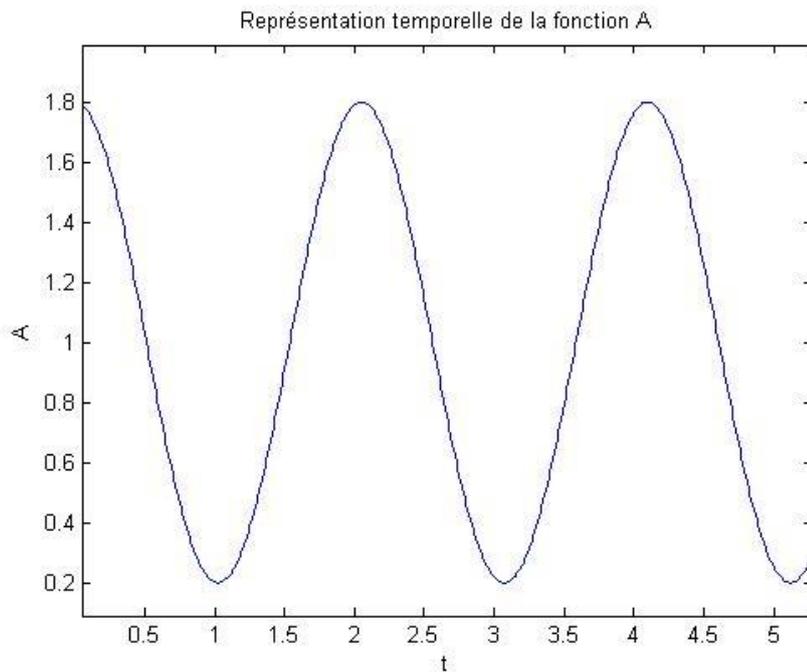
### Qu'est-ce que la modulation d'amplitude ?

Il s'agit de multiplier un signal par un cosinus appelé porteuse.

### Qu'est-ce que la démodulation d'amplitude ?

Il s'agit de multiplier le signal échantillonné reçu à l'entrée d'un récepteur par un  $\cos(w_0t)$  et un  $\sin(w_0t)$ .

Voici le signal que nous souhaitons émettre :



*Formules nécessaires :*

$$Z(t) = A(t) \cos(w_0t + \text{phi}(t)) + N(t)$$

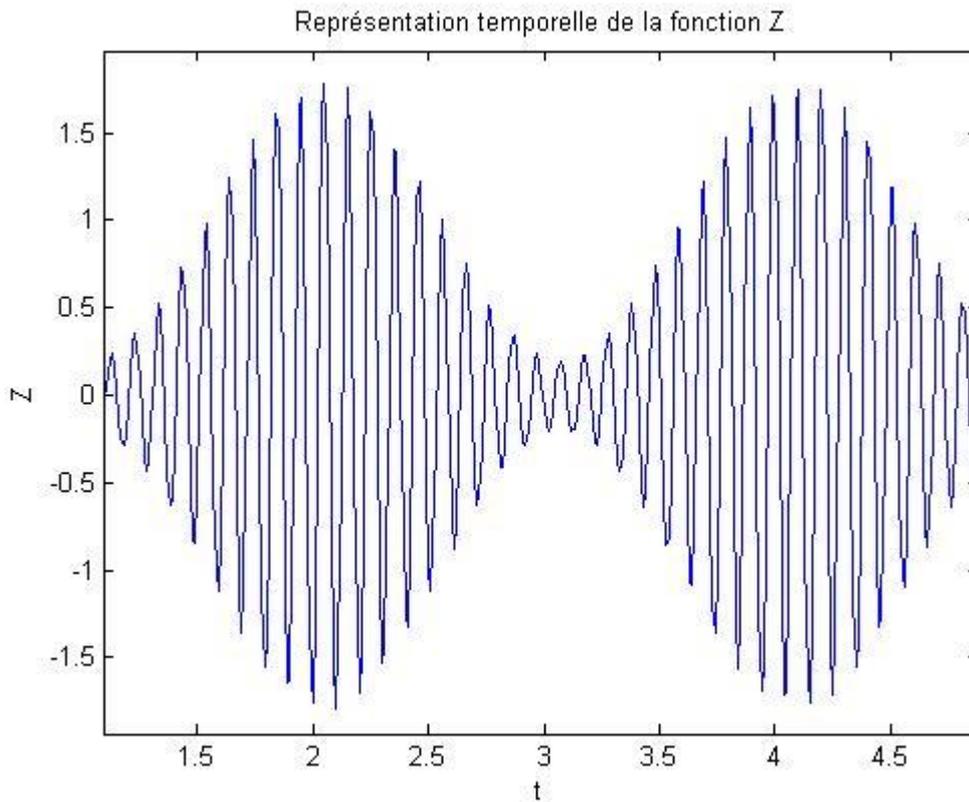
$$N(t) = N'(t) \cos(w_0t) + N''(t) \sin(w_0t)$$

## Exercice 1

### I. Le signal modulé

#### a. Représentation temporelle

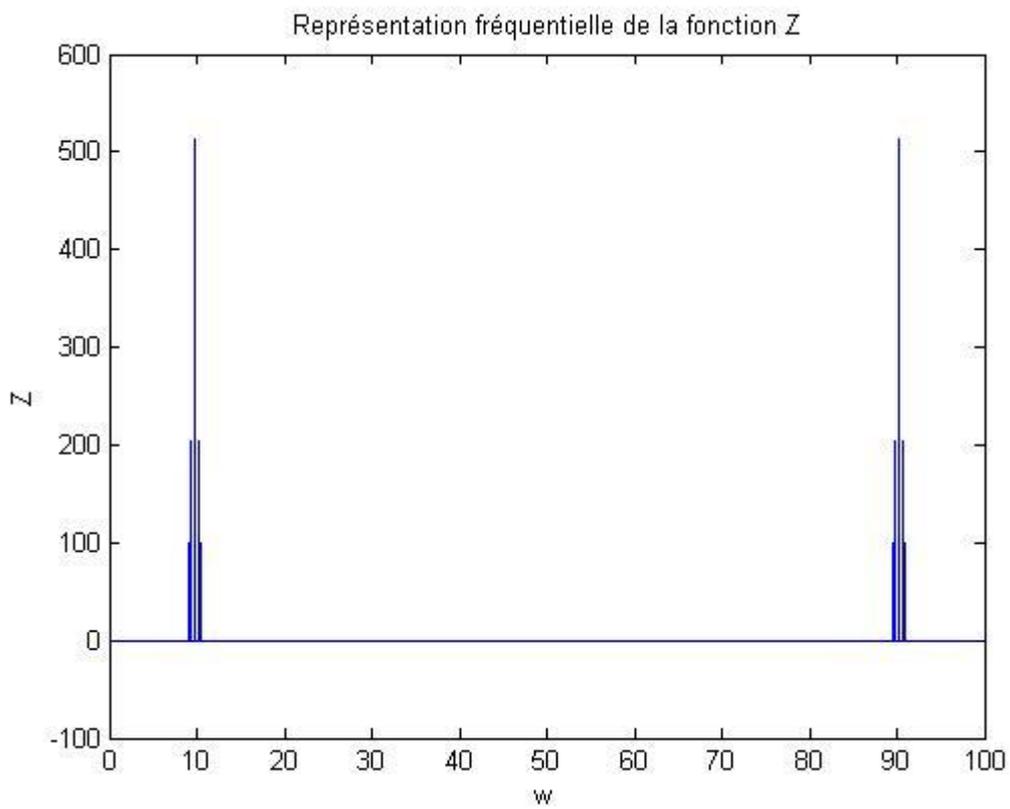
Dans un premier temps, nous générons le signal échantillonné sans bruit et modulé en amplitude, dont voici la représentation temporelle :



#### Code Matlab

```
w1=2*pi*f1;  
w0=2*pi*f0;  
A=ones(1,N)+m*cos(w1*t);  
Z=A.*cos(w0*t+phi);  
  
figure(1);  
plot(t,Z);  
title('Représentation temporelle de la fonction Z');  
ylabel('Z');  
xlabel('t');
```

## b. Représentation fréquentielle



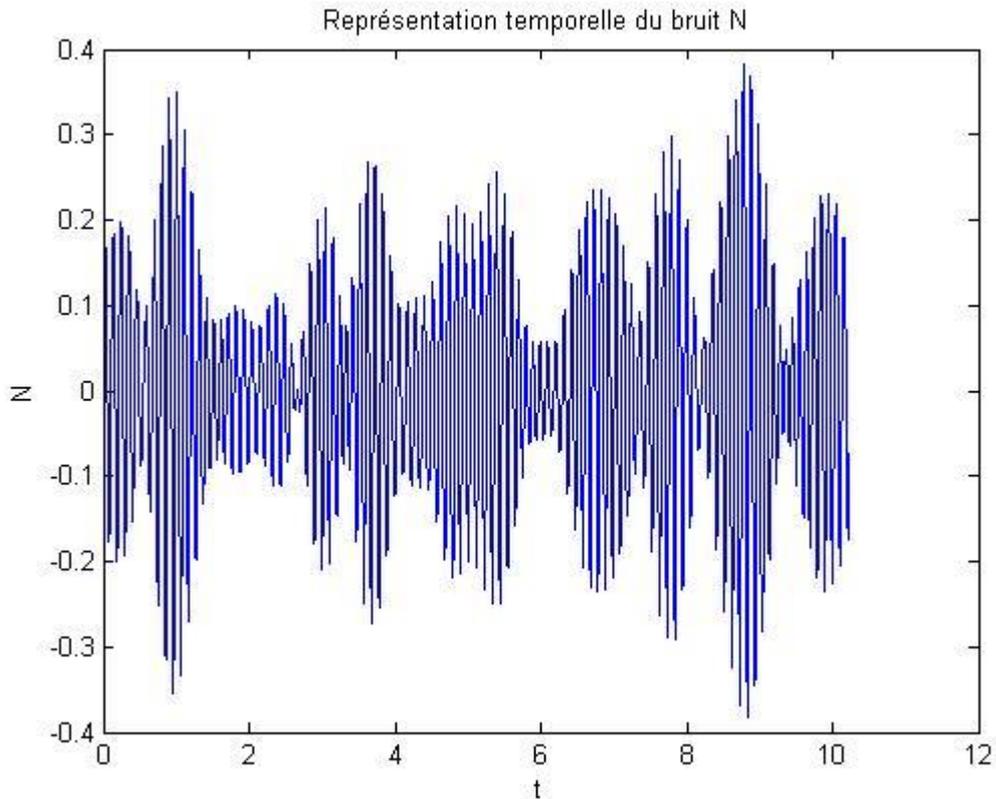
*Observation* : Deux raies situées aux pulsations 10 et 90 radians par seconde.

### Code Matlab

```
figure(2);  
plot(indice*1/Te/N,fft(Z));  
title('Représentation fréquentielle de la fonction Z');  
ylabel('Z');  
xlabel('w');
```

## II. Le bruit

Ensuite, nous chercherons à générer le bruit grâce à deux bruits Gaussiens indépendants qu'on filtrera et modulera.



### Code Matlab

```
v=1;
N1=sqrt(v)*randn(1,N);
N2=sqrt(v)*randn(1,N);
N1f=fft(N1);
N2f=fft(N2);
filtre=zeros(1,N);
filtre(1,1:B*N*Te)=ones(1,B*N*Te);
filtre(1,N-B*N*Te+1:N)=ones(1,B*N*Te);
N1t=ifft(N1f.*filtre);
N2t=ifft(N2f.*filtre);

figure;
plot(1:1024,filtre);

Nt=N1t.*cos(w0*t)+N2t.*sin(w0*t);

figure(3);
plot(t,Nt);
title('Représentation temporelle du bruit N ');
ylabel('N');
xlabel('t');
```

$Filtre = zeros(1, N)$

$Filtre(1, 1: B * N * Te) = ones(1, B * N * Te)$

$Filtre(1, N - B * N * Te + 1 : N) = ones(1, B * N * Te)$

La première opération permet de construire un tableau rempli de 0

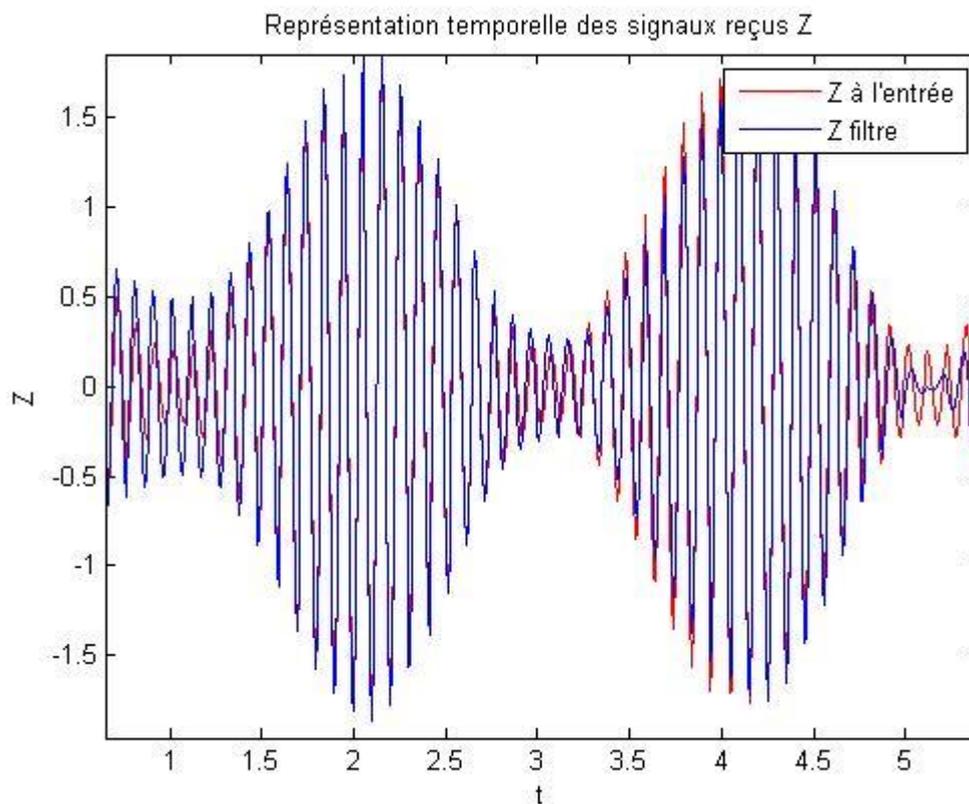
La seconde étape permet de remplir une première partie du tableau avec des 1 de 1 à 10 ( $B * N * Te$ )

La dernière étape permet de remplir une autre partie du tableau avec des 1 de 1015 ( $N - B * N * Te$ ) à 1024 ( $N$ ).

### III. Le signal en présence de bruit

#### a. Représentation temporelle

La suite logique du TP est d'ajouter au signal le bruit représenté précédemment pour obtenir le signal échantillonné reçu à l'entrée d'un récepteur. La figure ci-dessous représente les signaux avec et sans filtre.



#### Code Matlab

```
Zt=Z+Nt;
```

```
figure(4);
```

```
plot(t,Z,t,Zt);
```

```
title('Représentation temporelle des signaux reçus Z ');
```

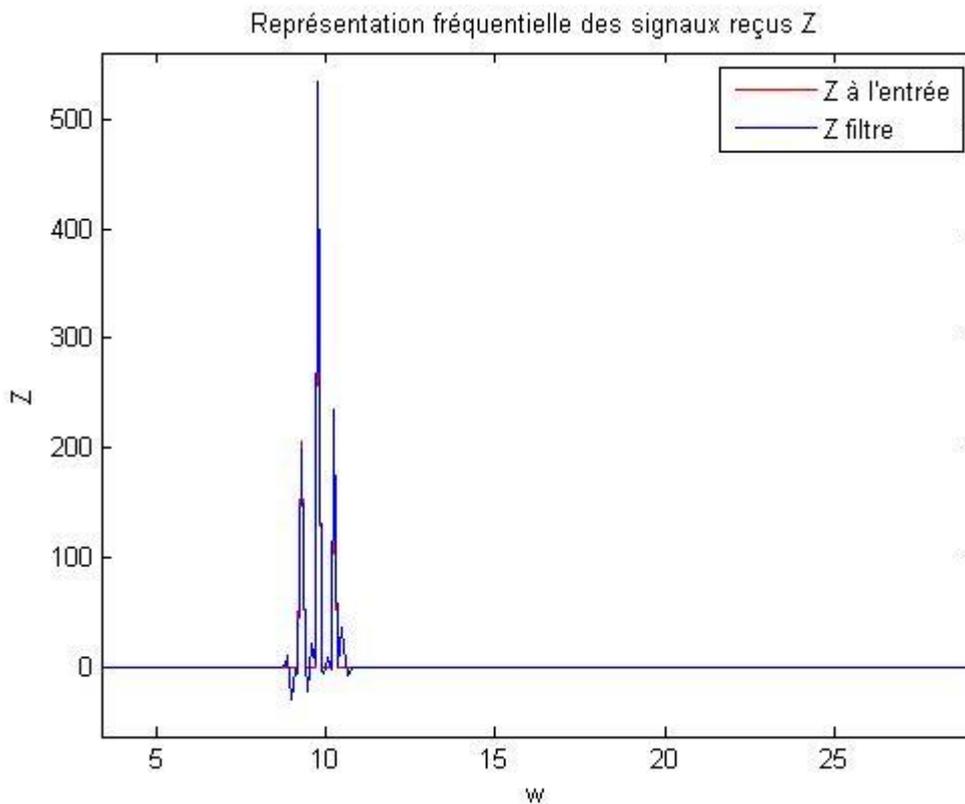
```
ylabel('Z');
```

```
xlabel('t');
```

```
legend('Z à l'entrée','Z filtre');
```

## b. Représentation fréquentielle

Grâce à la représentation spectrale du signal reçu  $Z$ , nous pouvons observer deux raies placées aux mêmes endroits que précédemment. Nous avons choisi de n'en afficher qu'une seule. On distingue deux courbes différentes : la courbe bleue avec le filtre permet de voir que le bruit est absorbé comme le témoigne la courbe rouge qui est la représentation brute du signal sans filtrage.



### Code Matlab

```
figure(5);  
plot(indice*1/Te/N,fft(Z),'-r',indice*1/Te/N,fft(Zt));  
title('Représentation fréquentielle des signaux reçus Z ');  
ylabel('Z');  
xlabel('w');  
legend('Z à l'entrée','Z filtre');
```

## Exercice 2

### I. Les signaux démodulés

Pour obtenir les signaux démodulés, on multiplie le signal échantillonné reçu  $Z$  par  $\cos(w_0 t)$  et  $\sin(w_0 t)$ , suivi d'un filtrage.

$$\begin{aligned}Z(t) &= A * \cos(w_0 * t + \text{phi}(t)) + N(t) \\N(t) &= N'(t) * \cos(w_0 * t) + N''(t) * \sin(w_0 * t)\end{aligned}$$

Commençons par multiplier  $Z(t)$  par  $\cos(w_0 t)$  :

$$\begin{aligned}Z(t) * \cos(w_0 * t) &= (A * \cos(w_0 * t + \text{phi}(t)) + N(t)) * \cos(w_0 * t) \\&= A * \cos(w_0 * t + \text{phi}(t)) * \cos(w_0 * t) + N'(t) * \cos^2(w_0 * t) + N''(t) * \sin(w_0 * t) * \cos(w_0 * t) \\&= A[\cos(w_0 * t) * \cos(\text{phi}(t)) - \sin(w_0 * t) * \sin(\text{phi}(t))] * \cos(w_0 * t) + N'(t) * \cos^2(w_0 * t) \\&\quad + N''(t) * \sin(w_0 * t) * \cos(w_0 * t) \\&= A \left[ \cos^2(w_0 * t) * \cos(\text{phi}(t)) - \frac{\sin(2 * w_0 * t)}{2} * \sin(\text{phi}(t)) + N'(t) * \cos^2(w_0 * t) + \frac{N''(t)}{2} \right. \\&\quad \left. * \sin(2 * w_0 * t) \right) \\&= A \left[ \frac{1 + \cos(2 * w_0 * t)}{2} * \cos(\text{phi}(t)) - \frac{\sin(2 * w_0 * t)}{2} * \sin(\text{phi}(t)) \right] + N'(t) * \frac{1 + \cos(2 * w_0 * t)}{2} \\&\quad + \frac{N''(t)}{2} * \sin(2 * w_0 * t)\end{aligned}$$

Filtre passe-bas : On annule les  $\cos(2 * w_0 * t)$  et  $\sin(2 * w_0 * t)$

D'où,

$$Z(t) * \cos(w_0 * t) = A * \frac{\cos(\text{phi}(t))}{2} + \frac{N'(t)}{2}$$

Ensuite, nous multiplions  $Z(t)$  par  $\sin(w_0 * t)$  :

$$\begin{aligned}Z(t) * \sin(w_0 * t) &= (A * \cos(w_0 * t + \text{phi}(t)) + N(t)) * \sin(w_0 * t) \\&= A * \cos(w_0 * t + \text{phi}(t)) * \sin(w_0 * t) + N'(t) * \cos(w_0 * t) * \sin(w_0 * t) + N''(t) * \sin^2(w_0 * t) \\&= A * \frac{\sin(2 * w_0 * t + \text{phi}(t)) - \sin(\text{phi}(t))}{2} + N'(t) * \frac{\sin(2 * w_0 * t)}{2} + N''(t) * \frac{1 - \cos(2 * w_0 * t)}{2}\end{aligned}$$

Filtre passe-bas :

$$Z(t) * \sin(w_0 * t) = -A * \frac{\sin(\text{phi}(t))}{2} + \frac{N''(t)}{2}$$

D'où,

$$\begin{aligned}X(t) &= A * \cos(\text{phi}(t)) + N'(t) \\Y(t) &= A * \sin(\text{phi}(t)) + N''(t)\end{aligned}$$

## II. Estimation de l'amplitude et de la phase

Pour estimer l'amplitude et la phase, on cherche à maximiser la fonction  $f(Xt, Yt)$ . Ainsi, on calcule les dérivées partielles de la fonction  $f(Xt, Yt)$

$$f(Xt, Yt) = \frac{1}{2\pi\sigma^2} e^{-1/2((Xt - At \cdot \cos(\text{phi}(t)))^2 + (Yt - At \cdot \sin(\text{phi}(t)))^2)}$$

$\frac{df}{dX} = 0$ , car on cherche à maximiser la fonction

La dérivée partielle s'annule donc pour l'élément situé dans l'exponentiel, d'où :

$$\frac{df}{dX} = -\frac{1}{2} * (2Xt - 2 * A * \cos(\text{phi}(t)))$$

$$\Leftrightarrow X = A * \cos(\text{phi}(t))$$

$$\frac{df}{dY} = -\frac{1}{2} * (2Yt - 2 * A * \sin(\text{phi}(t)))$$

$$\Leftrightarrow Y = A * \sin(\text{phi}(t))$$

$$\begin{cases} A = Xt / \cos(\text{phi}(t)) \\ A = Yt / \sin(\text{phi}(t)) \end{cases}$$

$$\begin{cases} A^2 = Xt^2 / \cos^2(\text{phi}(t)) \\ A^2 = Yt^2 / \sin^2(\text{phi}(t)) \end{cases}$$

$$\cos^2(\text{phi}(t)) + \sin^2(\text{phi}(t)) = 1$$

D'où :

$$X^2 + Y^2 = A^2$$

$$\Leftrightarrow A = \sqrt{X^2 + Y^2}$$

$$\begin{cases} A = Xt / \cos(\text{phi}(t)) \\ A = Yt / \sin(\text{phi}(t)) \end{cases}$$

Donc

$$\frac{Xt}{\cos(\text{phi}(t))} = \frac{Yt}{\sin(\text{phi}(t))}$$

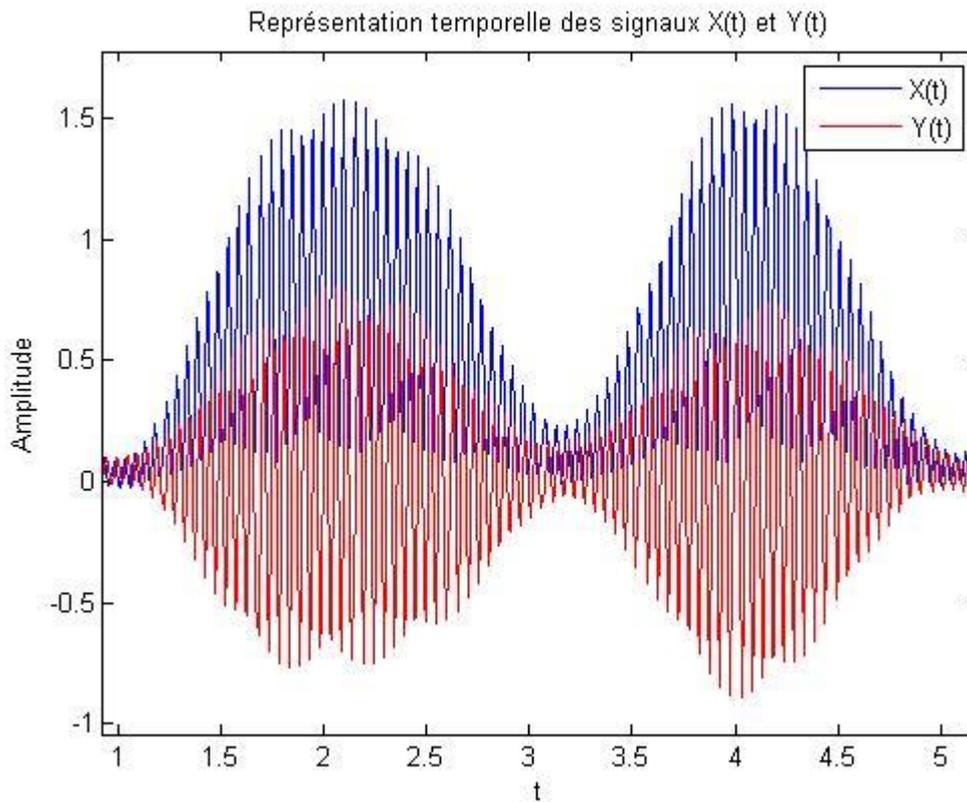
$$\Leftrightarrow \tan(\text{phi}(t)) = \frac{Yt}{Xt}$$

D'où

$$\text{phi}(t) = \arctan\left(\frac{Yt}{Xt}\right)$$

### III. Représentation sur Matlab

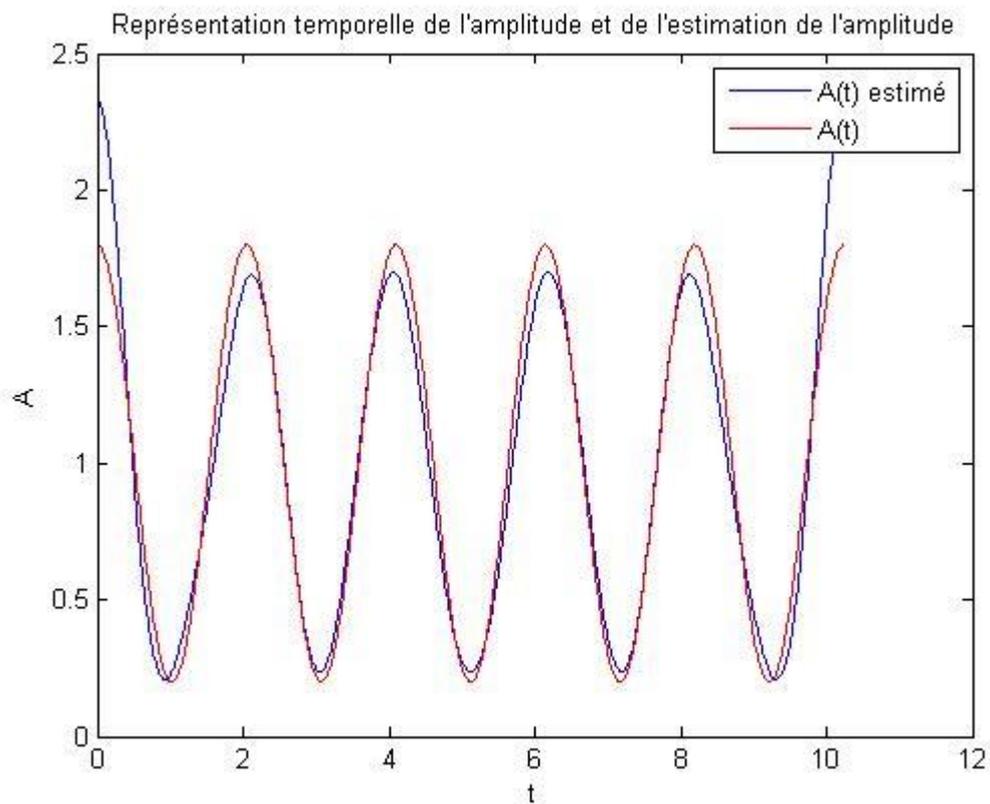
#### a. Représentation temporelle des signaux modulés $X(t)$ et $Y(t)$



#### Code Matlab

```
Xt=Zt.*cos(w0*t);  
Yt=Zt.*sin(w0*t);  
Xf=abs(fft(Xt)).*filtre;  
Yf=abs(fft(Yt)).*filtre;  
Xft=2*real(ifft(Xf));  
Yft=2*real(ifft(Yf));  
figure;  
plot(t,Xt,t,Yt,'-r');  
title('Représentation temporelle des signaux X(t) et Y(t)');  
ylabel('Amplitude');  
xlabel('t');  
legend('X(t)','Y(t)');
```

## b. Représentation temporelle de l'amplitude



On peut voir que le signal d'amplitude estimé s'approche du signal d'amplitude d'origine.

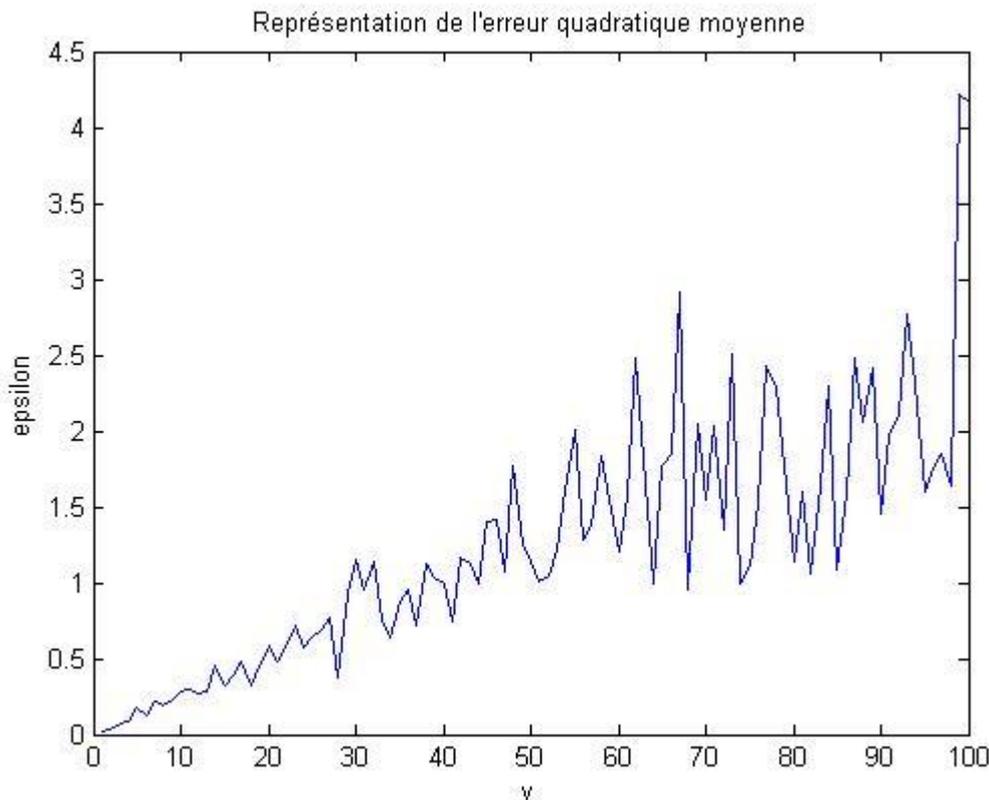
Avec ce niveau de bruit, l'erreur quadratique moyenne est faible.

Etudions maintenant ce qui se passe avec différents niveaux de bruit.

### Code Matlab

```
At=sqrt((Xft.*Xft+Yft.*Yft));  
figure;  
plot(t,At,t,A,'-r');  
title('Représentation temporelle de l'amplitude et de l'estimation de l'amplitude');  
ylabel('A');  
xlabel('t');  
legend('A(t) estimé','A(t)');
```

### c. Représentation de l'erreur quadratique moyenne



Nous pouvons voir que notre filtre n'est pas parfait, car plus l'on augmente la variance du bruit, l'erreur quadratique moyenne augmente.

#### Code Matlab

```
eqmv=zeros(1,100);
for v=1:1:100
    N1=sqrt(v)*randn(1,N);
    N2=sqrt(v)*randn(1,N);
    N1f=fft(N1);
    N2f=fft(N2);
    filtre=zeros(1,N);
    filtre(1,1:B*N*Te)=ones(1,B*N*Te);
    filtre(1,N-B*N*Te+1:N)=ones(1,B*N*Te);
    N1t=ifft(N1f.*filtre);
    N2t=ifft(N2f.*filtre);
    Nt=N1t.*cos(w0*t)+N2t.*sin(w0*t);
    Zt=Z+Nt;
    Xt=Zt.*cos(w0*t);
    Yt=Zt.*sin(w0*t);
    Xf=abs(fft(Xt)).*filtre;
    Yf=abs(fft(Yt)).*filtre;
    Xft=2*real(ifft(Xf));
    Yft=2*real(ifft(Yf));
    %figure;
    %plot(t,Xt,t,Yt,'-r');

    At=sqrt((Xft.*Xft+Yft.*Yft));
    eqm=1/N*sum((A-At).^2);
    eqmv(v)=eqm;
end
```

```
end
figure;
plot(1:100,eqmv);
title('Représentation de l''erreur quadratique moyenne');
ylabel('epsilon');
xlabel('v');
```

## Conclusion

Ce TP est basée sur l'étude de la modulation et la démodulation d'un signal. Cette étude nous a permis d'étudier les signaux : signal échantillonné reçu au niveau d'un récepteur ainsi que le bruit.

Nous avons remarqué que le bruit, selon des niveaux de variations différentes, a des conséquences sur la réception du signal. En effet, plus le bruit augmente, moins bon est le signal à la réception. Nous avons donc utilisé des filtres.

Ainsi, subsiste une différence entre l'amplitude reçue et l'amplitude estimée évoluant avec le niveau de bruit. Un niveau de bruit entraîne une variation du signal. C'est pourquoi, nous avons voulu représenter l'erreur quadratique moyenne. Plus la valence augmente, plus l'erreur est grande. Le filtre que nous avons donc utilisé n'est pas parfait.

## Annexes

### Exercice 1

#### Code Matlab

```
clear all
close all
clc
%% Exercice 1
%% Les données
N=1024;
Te=0.01;
B=10/(N*Te);
f1=B/2;
f0=10*B;
m=0.8;
phi=0;
indice=(0:1:N-1);
t=Te*indice;

%% Ex 1 Modulation d'amplitude
%% 1er point
w1=2*pi*f1;
w0=2*pi*f0;
A=ones(1,N)+m*cos(w1*t);
Z=A.*cos(w0*t+phi);

figure(1);
plot(t,Z);
title('Représentation temporelle de la fonction Z');
ylabel('Z');
xlabel('t');

figure;
plot(t,A);
title('Représentation temporelle de la fonction A');
ylabel('A');
xlabel('t');

figure(2);
plot(indice*1/Te/N,fft(Z));
title('Représentation fréquentielle de la fonction Z');
ylabel('Z');
xlabel('w');

%% 2ème point
v=1;
N1=sqrt(v)*randn(1,N);
N2=sqrt(v)*randn(1,N);
N1f=fft(N1);
```

```

N2f=fft(N2);
filtre=zeros(1,N);
filtre(1,1:B*N*Te)=ones(1,B*N*Te);
filtre(1,N-B*N*Te+1:N)=ones(1,B*N*Te);
N1t=ifft(N1f.*filtre);
N2t=ifft(N2f.*filtre);

figure;
plot(1:1024,filtre);

Nt=N1t.*cos(w0*t)+N2t.*sin(w0*t);

figure(3);
plot(t,Nt);
title('Représentation temporelle du bruit N ');
ylabel('N');
xlabel('t');

%% 3ème point

Zt=Z+Nt;

figure(4);
plot(t,Z,t,Zt);
title('Représentation temporelle des signaux reçus Z ');
ylabel('Z');
xlabel('t');
legend('Z à l\'entrée','Z filtre');

figure(5);
plot(indice*1/Te/N,fft(Z),'-r',indice*1/Te/N,fft(Zt));
title('Représentation fréquentielle des signaux reçus Z ');
ylabel('Z');
xlabel('w');
legend('Z à l\'entrée','Z filtre');

```

## Exercice 2

### Code Matlab

```

clear all
close all
clc
%% Exercice 1
%% Les données
N=1024;
Te=0.01;
B=10/(N*Te);
f1=B/2;
f0=10*B;
m=0.8;
phi=0;
indice=(0:1:N-1);

```

```

t=Te*indice;

%% Ex 1 Modulation d'amplitude
%% 1er point
w1=2*pi*f1;
w0=2*pi*f0;
A=ones(1,N)+m*cos(w1*t);
Z=A.*cos(w0*t+phi);

%% 2ème point

v=1;
N1=sqrt(v)*randn(1,N);
N2=sqrt(v)*randn(1,N);
N1f=fft(N1);
N2f=fft(N2);
filtre=zeros(1,N);
filtre(1,1:B*N*Te)=ones(1,B*N*Te);
filtre(1,N-B*N*Te+1:N)=ones(1,B*N*Te);
N1t=ifft(N1f.*filtre);
N2t=ifft(N2f.*filtre);

Nt=N1t.*cos(w0*t)+N2t.*sin(w0*t);

%% 3ème point

Zt=Z+Nt;

%% Exercice 2
Xt=Zt.*cos(w0*t);
Yt=Zt.*sin(w0*t);
Xf=abs(fft(Xt)).*filtre;
Yf=abs(fft(Yt)).*filtre;
Xft=2*real(ifft(Xf));
Yft=2*real(ifft(Yf));
figure;
plot(t,Xt,t,Yt,'-r');
title('Représentation temporelle des signaux X(t) et Y(t)');
ylabel('Amplitude');
xlabel('t');
legend('X(t)', 'Y(t)');

At=sqrt((Xft.*Xft+Yft.*Yft));
figure;
plot(t,At,t,A,'-r');
title('Représentation temporelle de l''amplitude et de l''estimation de l''amplitude');
ylabel('A');
xlabel('t');
legend('A(t) estimé', 'A(t)');

eqmv=zeros(1,100);

```

```

for v=1:1:100
    N1=sqrt(v)*randn(1,N);
    N2=sqrt(v)*randn(1,N);
    N1f=fft(N1);
    N2f=fft(N2);
    filtre=zeros(1,N);
    filtre(1,1:B*N*Te)=ones(1,B*N*Te);
    filtre(1,N-B*N*Te+1:N)=ones(1,B*N*Te);
    N1t=ifft(N1f.*filtre);
    N2t=ifft(N2f.*filtre);

    Nt=N1t.*cos(w0*t)+N2t.*sin(w0*t);

%% 3ème point

    Zt=Z+Nt;

%% Exercice 2
    Xt=Zt.*cos(w0*t);
    Yt=Zt.*sin(w0*t);
    Xf=abs(fft(Xt)).*filtre;
    Yf=abs(fft(Yt)).*filtre;
    Xft=2*real(ifft(Xf));
    Yft=2*real(ifft(Yf));
    %figure;
    %plot(t,Xt,t,Yt,'-r');

    At=sqrt((Xft.*Xft+Yft.*Yft));
    %figure;
    %plot(t,At,t,A,'-r');

    eqm=1/N*sum((A-At).^2);
    eqmv(v)=eqm;
end
figure;
plot(1:100,eqmv);
title('Représentation de l''erreur quadratique moyenne');
ylabel('epsilon');
xlabel('v');

```