

RAPPORT TPs TÉLÉCOMMUNICATIONS

Adrien ANDRÉ

Table des matières

1	Etude de chaînes de transmission en bande de base sur canal AWGN	2
1.1	Mapping 1	2
1.1.1	Modulation	2
1.1.2	Démodulation sans bruit	2
1.1.3	Démodulation avec bruit	3
1.2	Mapping 2	4
1.2.1	Modulation	4
1.2.2	Démodulation sans bruit	5
1.2.3	Démodulation avec bruit	5
1.3	Mapping 3	6
1.3.1	Modulation	6
1.3.2	Démodulation sans bruit	6
1.3.3	Démodulation avec bruit	7
1.4	Mapping 4	7
1.4.1	Modulation	7
1.4.2	Démodulation sans bruit	7
1.4.3	Démodulation avec bruit	8
1.5	Mapping 5	8
1.5.1	Modulation	8
1.5.2	Démodulation sans bruit	9
1.5.3	Démodulation avec bruit	9
1.6	Comparaison entre les 4 chaînes	11
1.6.1	Efficacité spectrale	11
1.6.2	Efficacité en puissance	12
2	Étude de chaînes de transmission sur fréquence porteuse, sur canal AWGN	13
2.1	Implantation d'une chaîne de transmission QPSK	13
2.1.1	Implantation de la chaîne sur fréquence porteuse	13
2.1.2	Implantation de la chaîne passe-bas équivalente	15
2.2	Comparaison des modulations BPSK, QPSK, 8-PSK et 16-QAM	17
2.2.1	Constellations	17
2.2.2	TEB	20
2.2.3	Efficacité spectrale	21
2.2.4	Efficacité en puissance	22

1 Etude de chaines de transmission en bande de base sur canal AWGN

Pour les 5 mappings, on choisi les paramètres suivants :

```
n=10000;  
Ts=4;  
Ns = 4;  
Te= Ts/Ns;
```

1.1 Mapping 1

Mapping binaire à moyenne nulle (symboles $a_k \in -1, 1$), mise en forme rectangulaire de durée T_S (forme d'onde du GPS), filtre de réception de réponse impulsionnelle rectangulaire de durée T_S .

1.1.1 Modulation

On commence par générer les symboles a_k , puis on on suréchantillonne à l'aide d'une suite de Diracs :

```
%Generation des symboles (-1,1)  
SymbolesMapping = 2*randi([0, 1], 1, n) - ones(1,n);  
  
%Generation de la suite de Diracs ponderees  
Suite_diracs = kron(SymbolesMapping, [1 zeros(1, Ns - 1)]);
```

Ensuite, il s'agit du filtrage de mise en forme, ici on utilise le filtre h rectangulaire de durée T_S .

```
%Reponse impulsionnelle du filtre de mise en forme  
h= ones(1, Ts);  
  
%Filtrage de mise en forme  
SignalFiltre = filter(h, 1, Suite_diracs);
```

1.1.2 Démodulation sans bruit

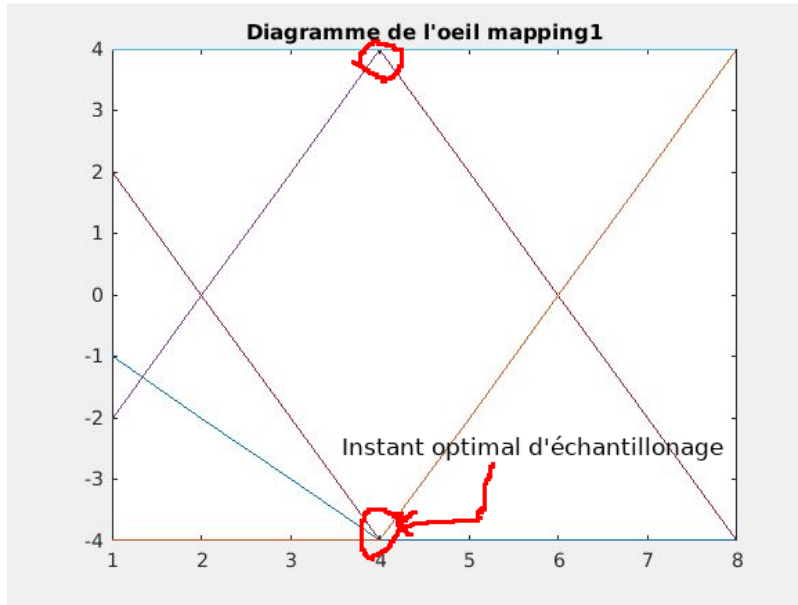
Ensuite on continue la chaine en la réalisant tout d'abord sans bruit. Pour la démodulation, on commence par effectuer le filtrage de réception, avec encore une fois un filtre h_r rectangulaire de durée T_S

```
%Reponse impulsionnelle  
hr = ones(1,Ts);  
%Filtrage  
z_sans_bruit= filter(hr,1,SignalFiltre);
```

Il s'agit ensuite d'échantillonner, pour échantillonner à un instant optimal, on utilise un diagramme de l'oeil qui va nous permettre de déterminer l'instant optimal d'échantillonnage :

```
%Echantillonnage  
diagramme_oeil_sans_bruit = reshape(z_sans_bruit, 2*Ts, []);  
t0= Ts;  
ze_sans_bruit = z_sans_bruit(t0:Ts:end);
```

Le diagramme de l'oeil obtenu est le suivant :



On obtient un temps optimal d'échantillonnage de 4s, ce qui correspond en fait à T_S . On échantillonne donc à partie de T_S (voir ci-dessus). Enfin, on fait le mapping inverse. Pour cela, on met en place un décideur et un détecteur. Celui-ci va regarder le signe du signal échantillonné et ainsi lui attribuer une valeur de bits (-1 ou 1).

```
%Decisions - detecteur
seuil = 0;
indices_positifs = (ze_sans_bruit >= seuil);
indices_negatifs = (ze_sans_bruit < seuil);
signal_detecte_sans_bruit = zeros(size(ze_sans_bruit));
signal_detecte_sans_bruit(indices_positifs) = 1;
signal_detecte_sans_bruit(indices_negatifs) = -1;
```

NB : On aurait aussi pu aussi utiliser la fonction `sign` de matlab.
On peut alors calculer le TEB sans bruit.

```
%Demapping
bit_transmis = length(SymbolesMapping);
bit_errone_sans_bruit = sum(ne(signal_detecte_sans_bruit, SymbolesMapping));
TEB_sans_bruit = bit_errone_sans_bruit/bit_transmis;
fprintf("le TEB du signal sans bruit est: TEB=%0.3e\n", TEB_sans_bruit)
```

On obtient alors bien un TEB nul :

```
>> mapping1
le TEB du signal sans bruit est: TEB=0.000e+00
```

1.1.3 Démodulation avec bruit

On va ajouter un bruit de la forme :

$$\sigma_n^2 = \frac{P_r N_s}{2 \log_2(M) \frac{E_b}{N_0}}$$

avec :

$$\frac{E_b}{N_0} = 10^{\frac{SNR_{dB}}{10}}$$

. Pour cela, on calcule P_r et on prend 7 valeurs de SNR_{dB} allant de 0 à 6dB.

```
Pr = mean(abs(SignalFiltre).^2);
SNR_dB = linspace(0,6,7);
N = length(SignalFiltre);
```

On peut alors faire une boucle `for` pour réaliser les mêmes étapes que précédemment en tenant compte du bruit pour les 7 valeurs de SNR_{dB}

```

for i=1:length(SNR_dB)
    %Ajout du bruit
    Pb = Pr*Ns/(2*log2(M)*(10^(SNR_dB(i)/10)));
    bruit_gauss = sqrt(Pb) * randn(1,N);
    canal = SignalFiltre + bruit_gauss;
    %Filtrage
    z_avec_bruit = filter(hr,1,canal);
    %Echantillonnage
    t0 = Ts;
    ze_avec_bruit = z_avec_bruit(t0:Ts:length(z_avec_bruit));
    %Decisions - detecteur
    seuil = 0;
    signal_detecte_avec_bruit = zeros(size(ze_avec_bruit));
    indices_positifs = (ze_avec_bruit >= seuil);
    indices_negatifs = (ze_avec_bruit < seuil);
    signal_detecte_avec_bruit(indices_positifs) = 1;
    signal_detecte_avec_bruit(indices_negatifs) = -1;
    %Demapping
    bit_errone_avec_bruit = sum(ne(signal_detecte_avec_bruit, SymbolesMapping));
    TEB_bruit(i) = bit_errone_avec_bruit/bit_transmis;
    %fprintf("le TEB est: TEB=%0.4e\n", TEB_bruit(i));
end;

```

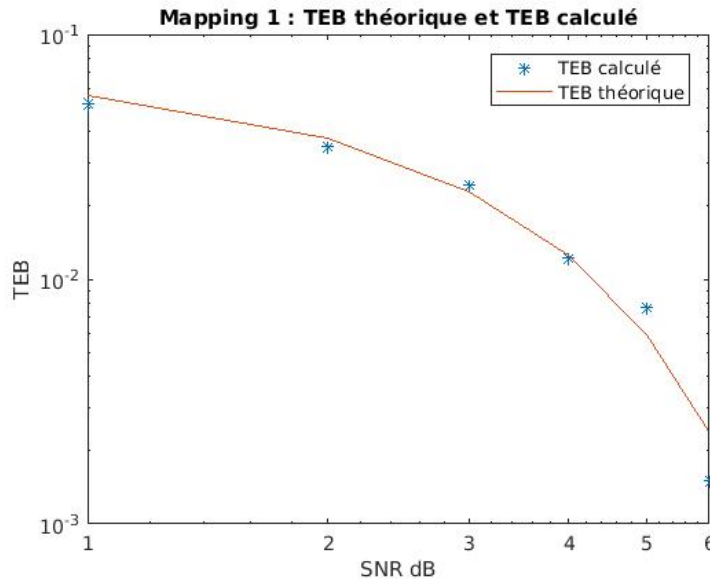
On peut également calculer le TEB théorique à l'aide de la formule donnée dans les planches de cours :

```

|| TEB_theorique = qfunc(sqrt(2*10.^(SNR_dB/10)));

```

Cela nous permet de tracer un graph superposant les tracés des TEB théoriques et calculés sur une échelle logarithmique :



On remarque alors que le TEB théorique est une très bonne approximation du TEB calculé.

1.2 Mapping 2

Mapping binaire à moyenne non nulle (symboles $a_k \in \{0, 1\}$), mise en forme rectangulaire de durée T_S , filtre de réception de réponse impulsionnelle rectangulaire de durée T_S .

1.2.1 Modulation

Pour ce deuxième mapping, la seule chose changeant par rapport au premier au niveau de la modulation est le mapping. Pour le réaliser on génère une suite aléatoire de n bits correspondant aux symboles a_k :

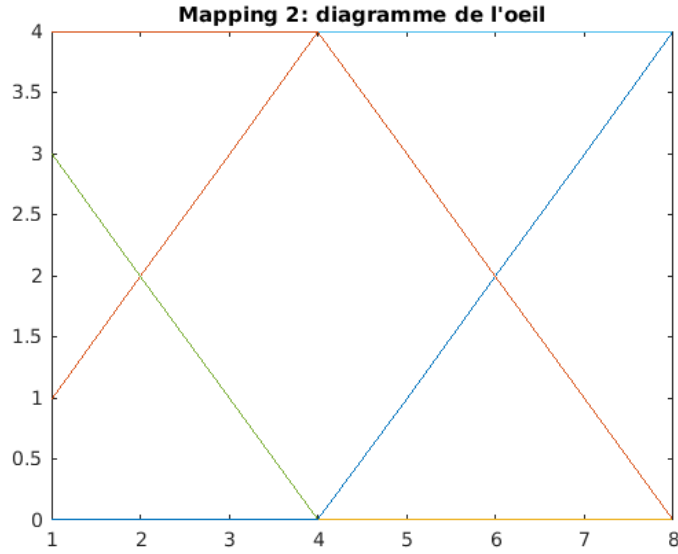
```

|| SymbolesMapping = randi([0, 1], 1, n);

```

1.2.2 Démodulation sans bruit

On obtient ici le diagramme de l'oeil suivant :



Ce diagramme nous permet de déterminer l'instant d'échantillonnage qui est : $t_0 = T_S$. Le détecteur n'est plus le même que dans le premier mapping. En effet, le seuil se situe désormais à $N_s/2$:

```
seuil = Ns/2;
indices_positifs = (ze_sans_bruit >= seuil);
indices_negatifs = (ze_sans_bruit < seuil);
signal_detecte_sans_bruit = zeros(size(ze_sans_bruit));
signal_detecte_sans_bruit(indices_positifs) = 1;
signal_detecte_sans_bruit(indices_negatifs) = 0;
```

On fait de même que dans le premier mapping pour calculer le TEB sans bruit et on obtient un TEB de 0 :

```
>> mapping2
le TEB du signal sans bruit est: TEB=0.000e+00
```

1.2.3 Démodulation avec bruit

Le TEB théorique n'est pas donné dans les planches de cours. Il faut le calculer.

On a :

$$TEB_{theorique} = Q\left(\frac{D_{min}}{2\sigma}\right)$$

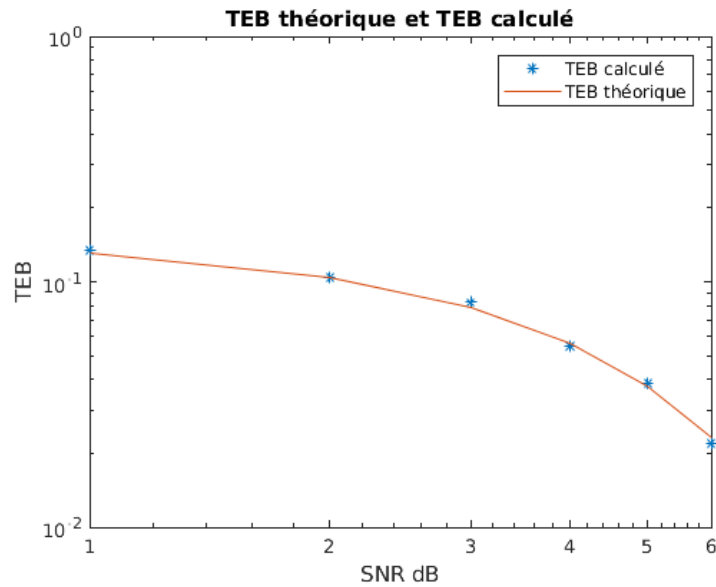
Or,

$$\begin{aligned}
D_{min} &= \frac{T_S}{2} \\
\sigma^2 &= \int_R S_{n*hr}(f) df \\
\sigma^2 &= \int_R S_n(f) |H_r(f)|^2 df \\
\sigma^2 &= \frac{N_0}{2} \int_R |H_r(f)|^2 df \\
\sigma^2 &= \frac{N_0}{2} \int_{T_S} |h_r(t)|^2 dt \\
\sigma^2 &= \frac{N_0 T_S}{2}
\end{aligned}
\quad
\begin{aligned}
E_b &= P_{signal} \times T_b \\
E_b &= \int_R S_x(f) df \times T_S \\
E_b &= \frac{1}{4} \int |h(t)|^2 dt + \frac{T_S}{2} \\
E_b &= \frac{T_S}{2}
\end{aligned}$$

Ainsi,

$$TEB_{theorique} = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

On obtient alors :



1.3 Mapping 3

Mapping binaire à moyenne nulle (symboles $a_k \in \{-1, 1\}$), mise en forme rectangulaire de durée T_S , filtre de réception de réponse impulsionnelle rectangulaire de durée $T_S/2$.

1.3.1 Modulation

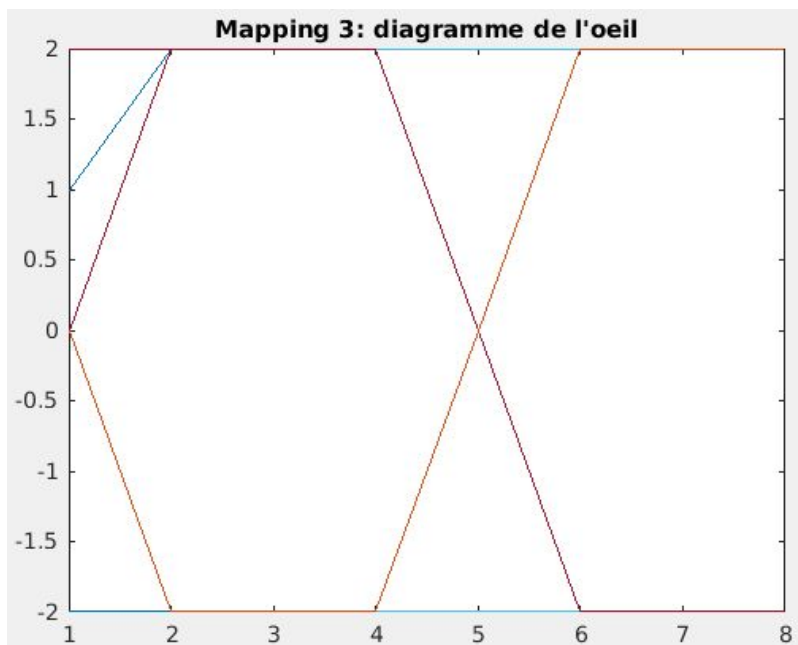
cf 1.1.1

1.3.2 Démodulation sans bruit

Par rapport au mapping 1, on change la durée du filtre de réception :

```
|| hr = ones(1, Ts/2);
```

On trace le diagramme de l'oeil et on obtient $t_0 = \frac{T_S}{2}$



On fait de même que dans le premier mapping pour calculer le TEB sans bruit et on obtient un TEB de 0 :

```
>> mapping3
le TEB du signal sans bruit est: TEB=0.000e+00
```

1.3.3 Démodulation avec bruit

Le TEB théorique n'est pas donné dans les planches de cours. Il faut le calculer.

On a :

$$TEB_{theorique} = Q\left(\frac{D_{min}}{2\sigma}\right)$$

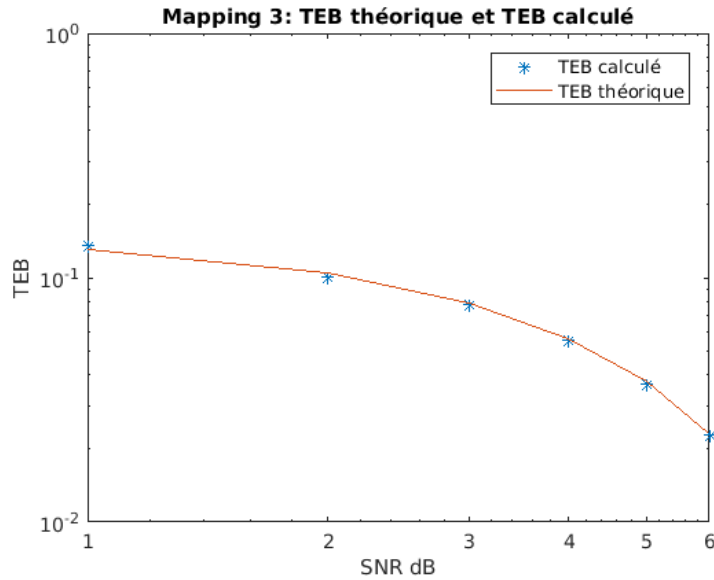
Or,

$$\begin{aligned} \sigma^2 &= \int_R S_{n*hr}(f) df & E_b &= P_{signal} \times T_b \\ \sigma^2 &= \int_R S_n(f) |H_r(f)|^2 df & E_b &= \int_R S_x(f) df \times T_S \\ D_{min} &= T_S & E_b &= \frac{\sigma^2}{T_S} \int_R |H(f)|^2 \times T_S df \\ \sigma^2 &= \frac{N_0}{2} \int_R |H_r(f)|^2 df & E_b &= \frac{\sigma^2}{T_S} \int_{T_S} |h(t)|^2 dt \\ \sigma^2 &= \frac{N_0}{2} \int_{T_S} |h_r(t)|^2 dt & E_b &= T_S \\ \sigma^2 &= \frac{N_0 T_S}{2} & & \end{aligned}$$

Ainsi,

$$TEB_{theorique} = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

On obtient alors :



1.4 Mapping 4

Mapping binaire à moyenne nulle (symboles $a_k \in \{-1, 1\}$), mise en forme de type front de durée T_S (forme d'onde Ethernet), filtre de réception adapté à la forme d'onde reçue.

1.4.1 Modulation

On implémente le filtre de mise en forme de type front de durée T_S :

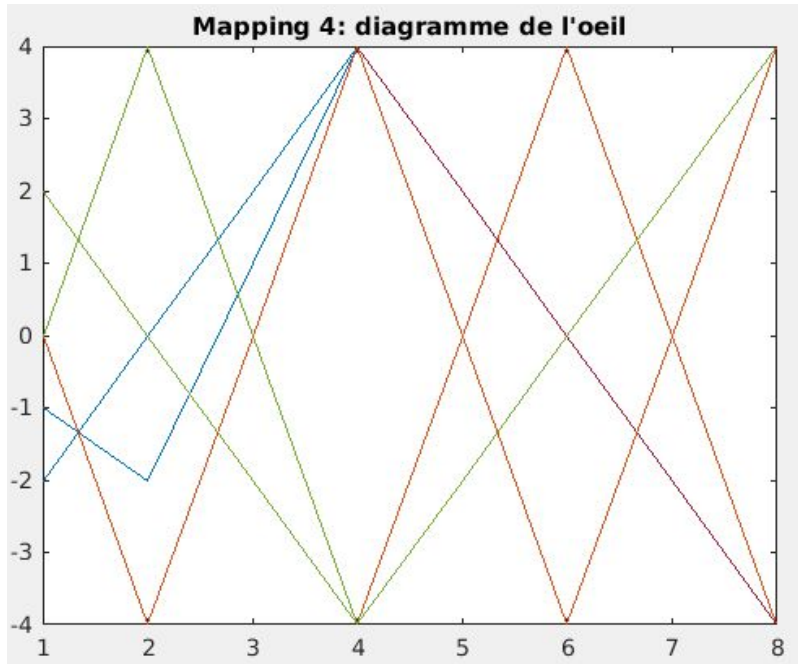
```
|| h = [ones(1, Ts/2) -ones(1, Ts/2)];
```

1.4.2 Démodulation sans bruit

On implémente le filtre de réception :

```
|| hr = -h
```

On trace alors le diagramme de l'oeil et on obtient $t_0 = T_S$

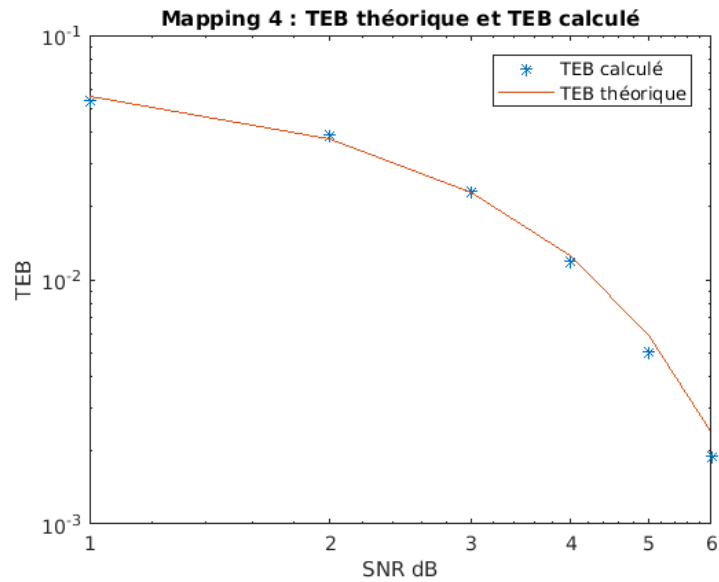


On fait de même que dans le premier mapping pour calculer le TEB sans bruit et on obtient un TEB de 0 :

```
>> mapping4
le TEB du signal sans bruit est: TEB=0.000e+00
```

1.4.3 Démodulation avec bruit

On utilise le TEB théorique donné dans les planches de cours et on obtient le graphique suivant :



1.5 Mapping 5

Mapping binaire à moyenne nulle (symboles $a_k \in \{-1, 1\}$), mise en forme en racine de cosinus surélevé de roll-off 0.35 (forme d'onde du DVB-S), filtre de réception adapté à la forme d'onde reçue.

1.5.1 Modulation

On met en place le filtre de mise en forme en racine de cosinus surélevé de roll-off 0.35. On prend bien soin de prendre en compte le retard induit par le filtre :


```

%Reponse impulsionnelle du filtre de mise en forme
alpha = 0.35;
h = rcosdesign(alpha,10,Ts);
%Filtrage de mise en forme
retard = (length(h)-1)/2;
SignalFiltre = filter(h, 1, [Suite_diracs zeros(1,retard)]);
SignalFiltre = SignalFiltre(1+retard:end);

```

1.5.2 Démodulation sans bruit

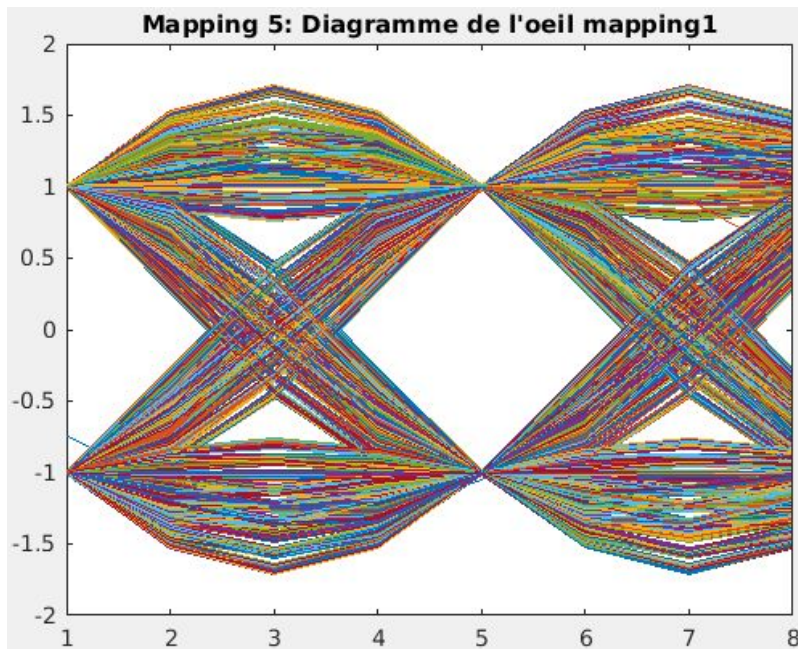
On implémente le filtre de réception, et on prend encore en compte le retard qu'il induit :

```

%Reponse impulsionnelle
hr = fliplr(h);
%Filtrage
z_sans_bruit = filter(hr,1,[SignalFiltre zeros(1,retard)]);
z_sans_bruit = z_sans_bruit(1+retard:end);

```

On trace alors le diagramme de l'oeil et on obtient $t0 = 1$



On fait de même que dans le premier mapping pour calculer le TEB sans bruit et on obtient un TEB de 0 :

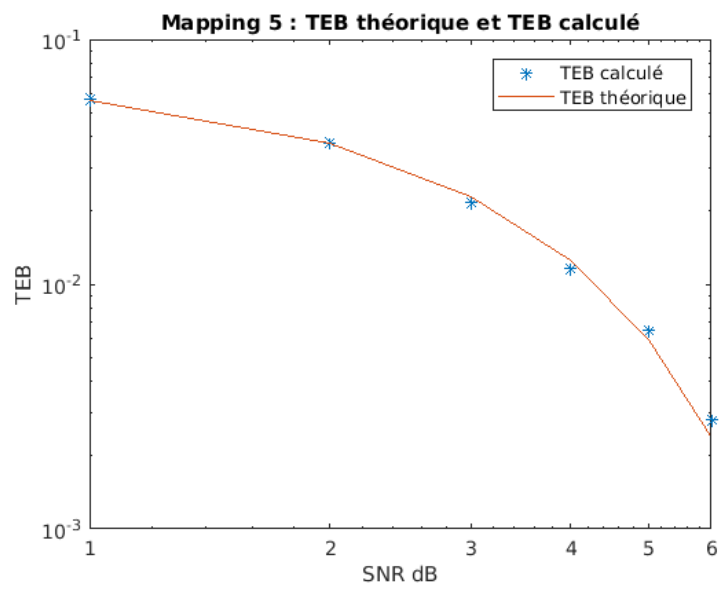
```

>> mapping5
le TEB du signal sans bruit est: TEB=0.000e+00

```

1.5.3 Démodulation avec bruit

On utilise le TEB théorique donné dans les planches de cours et on obtient le graphique suivant :



1.6 Comparaison entre les 4 chaînes

1.6.1 Efficacité spectrale

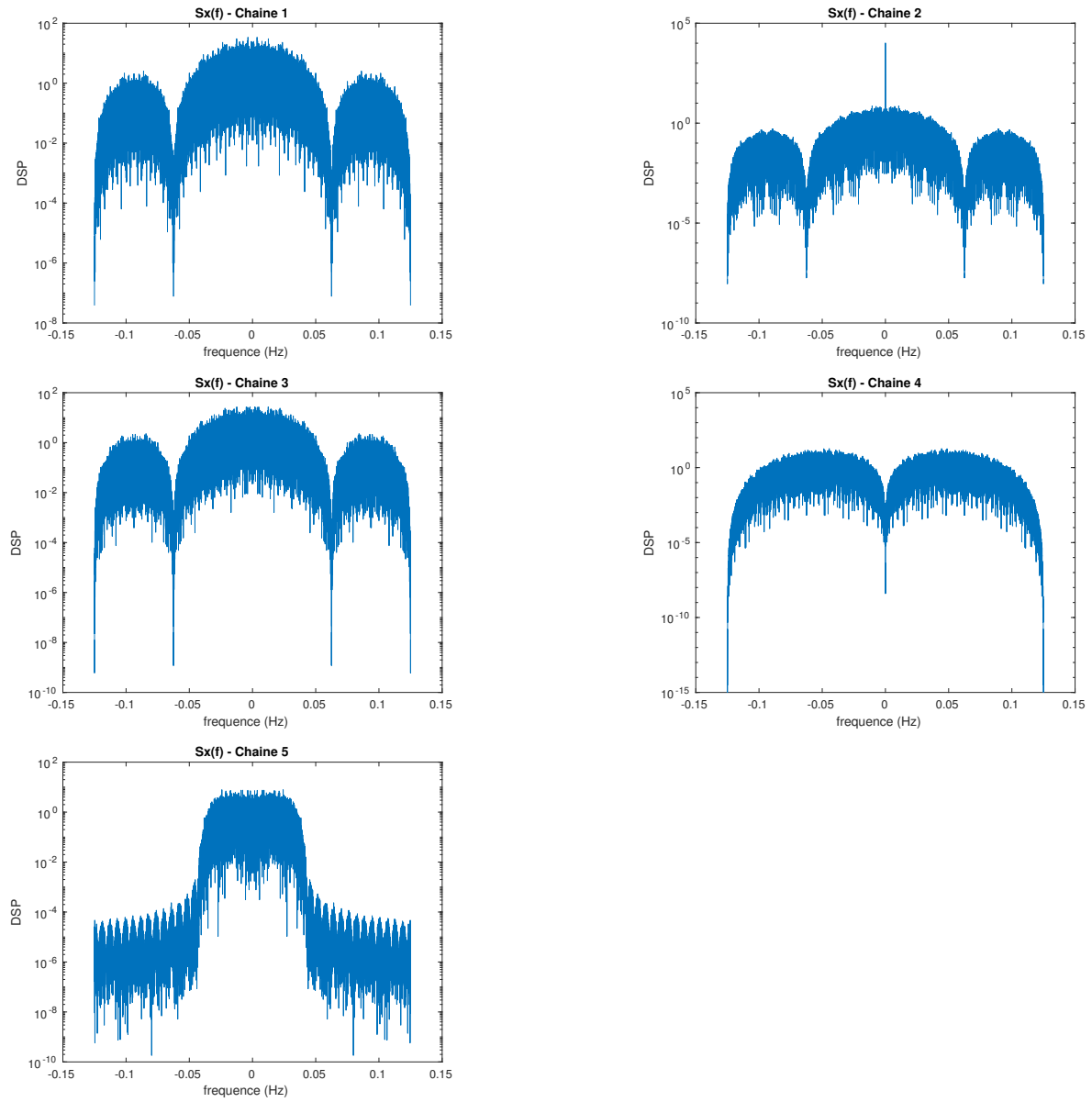


FIGURE 1 – Densité spectrale de puissance des différents mappings

La chaîne 5 a la bande passante la plus faible, c'est donc la chaîne la plus efficace spectralement. La chaîne 4 possède la plus grande bande, c'est donc la moins efficace.

1.6.2 Efficacité en puissance

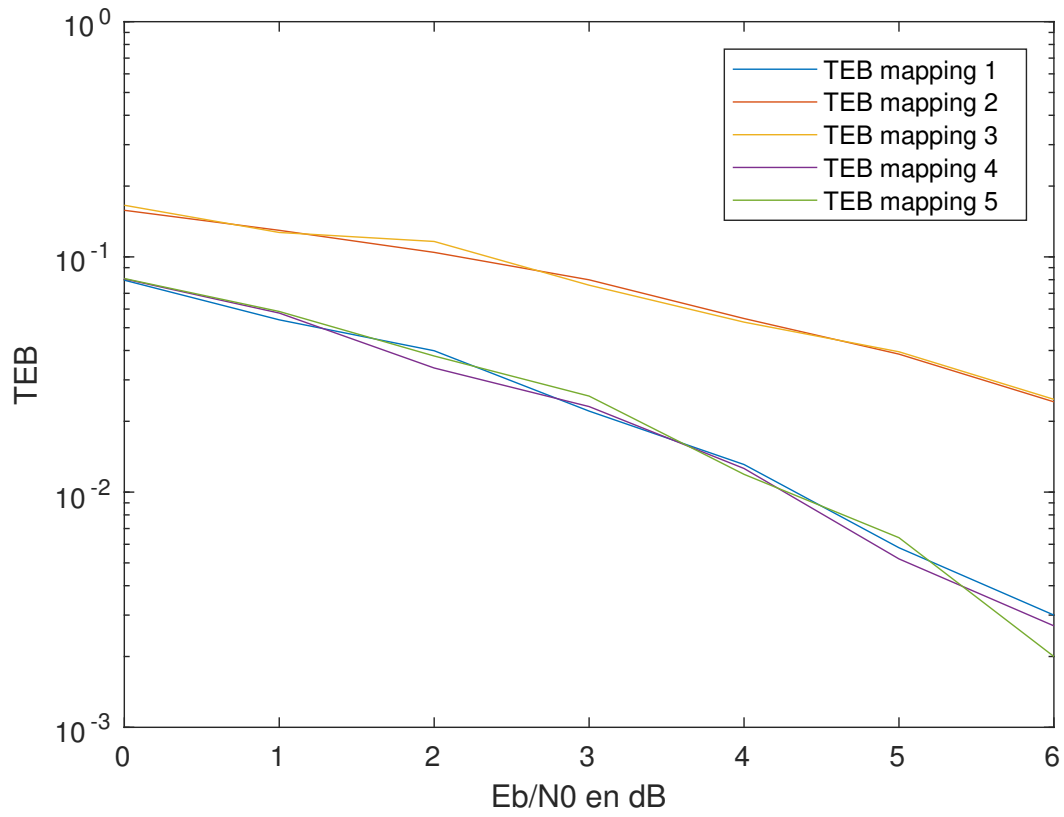


FIGURE 2 – TEB calculés des différents mappings

Pour une valeur de TEB fixée, on a $(\frac{E_b}{N_0})_{1,4,5} < (\frac{E_b}{N_0})_{2,3}$. Donc les mappings 1,4,5 sont les plus efficaces en puissance.

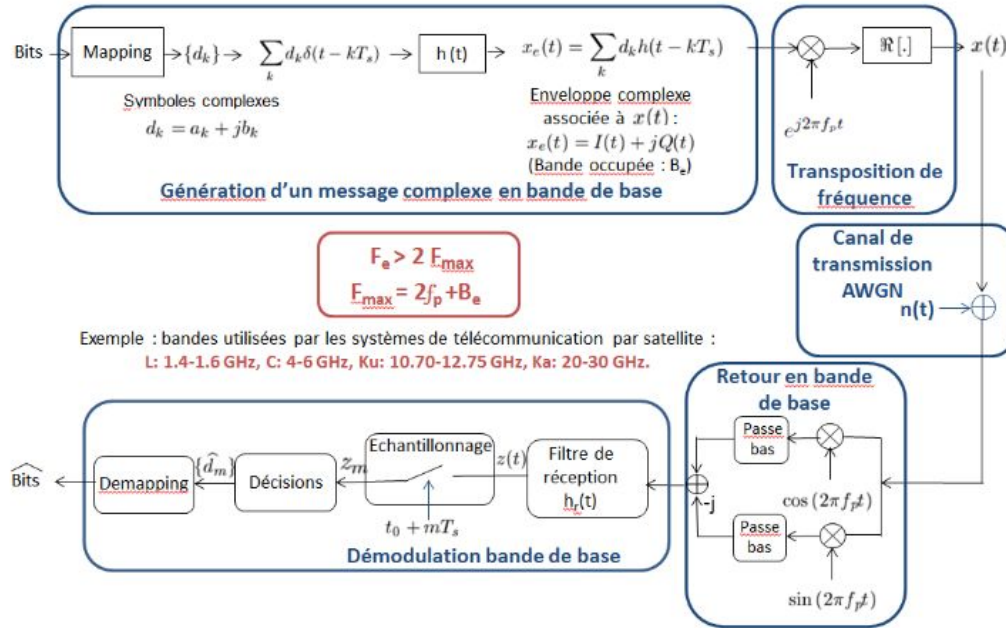
2 Étude de chaînes de transmission sur fréquence porteuse, sur canal AWGN

L'objectif de cette partie est d'implémenter la chaîne de transmission QPSK. D'abord sur fréquence porteuse puis avec un passe bas équivalent. On utilise les mêmes méthodes que pour le mapping 5 de la partie précédente. Il faut cependant générer des signaux en phase, en quadrature et sur fréquence porteuse :

2.1 Implantation d'une chaîne de transmission QPSK

2.1.1 Implantation de la chaîne sur fréquence porteuse

On réalise la chaîne de transmission suivante :



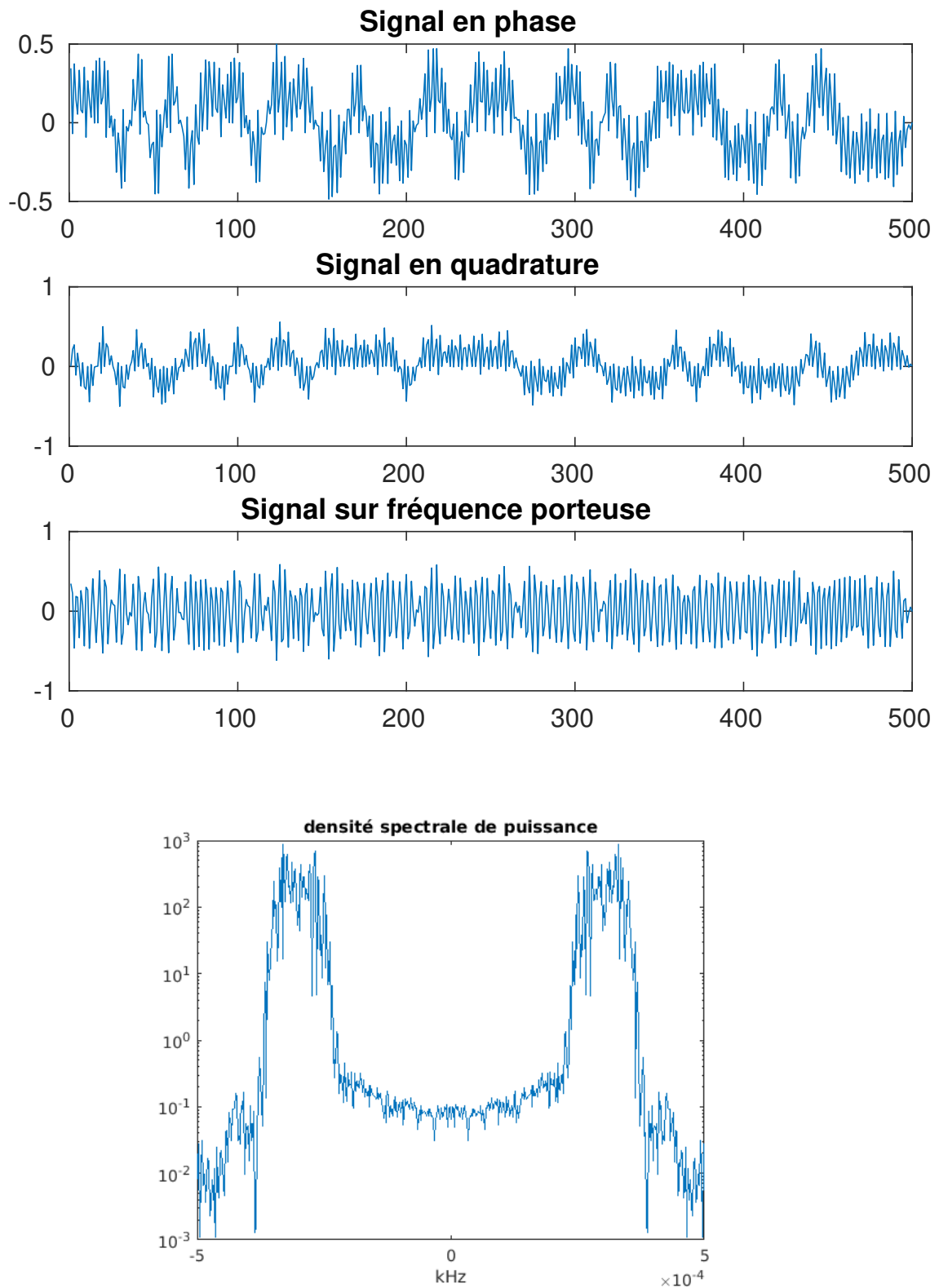
On doit donc ici générer des symboles complexes :

```
% Generation de 1 suite de bits 0 1 equiprobable dans un vecteur ligne de n elements
bits = randi([0,1],1,n);
% Separation en 2 de la suite de bits
bitsak = bits(1:2:end);
bitsbk = bits(2:2:end);
% Mapping
ak = bitsak*2-1; % on remplace les 0 par -1
bk = bitsbk*2-1;
dk = ak+1i*bk;
```

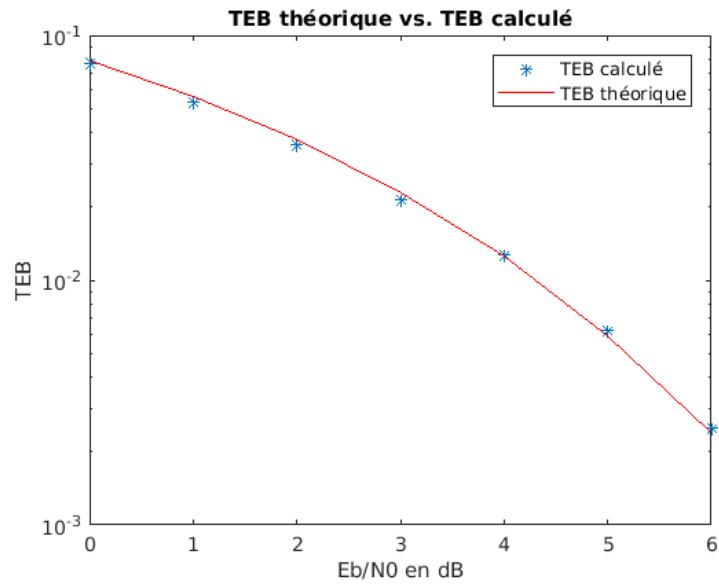
On réalise le filtrage de mise en forme de la même manière que pour le mapping 5 et on effectue une transposition de fréquence, ainsi qu'un retour en bande de base :

```
% Transposition de frequence
x_h_exp = x_h.*exp(1i*2*pi*fp/Fe*temps);
x = real(x_h_exp);
% Retour en bande de base
x_cos = x.*cos(2*pi*fp/Fe*temps);
x_sin = x.*sin(2*pi*fp/Fe*temps);
```

On peut alors tracer les signaux obtenus en quadrature et en phase, ainsi que la densité spectrale de puissance :

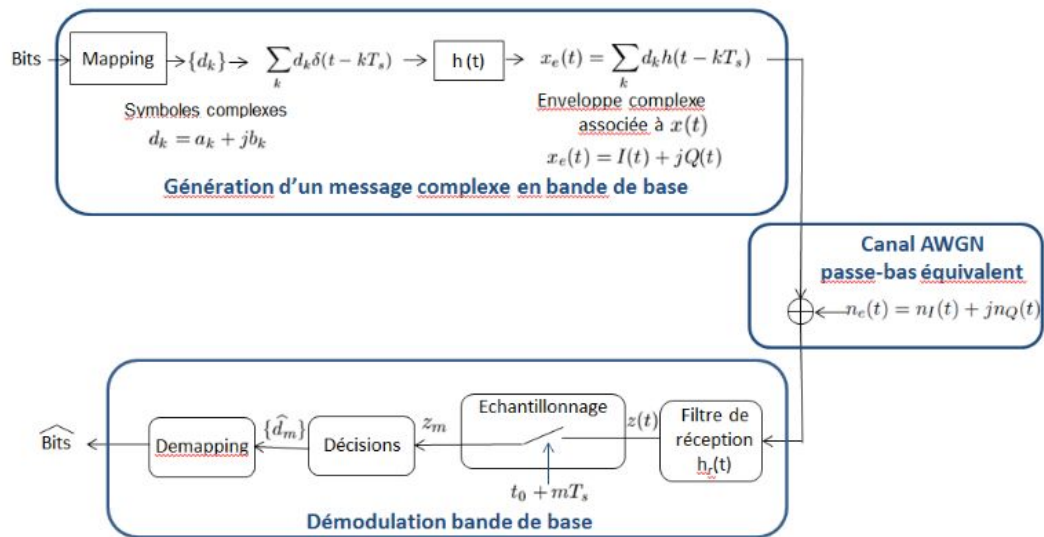


De même que précédemment on utilise les planches de cours pour déterminer le TEB théorique et le superposer au TEB calculé :

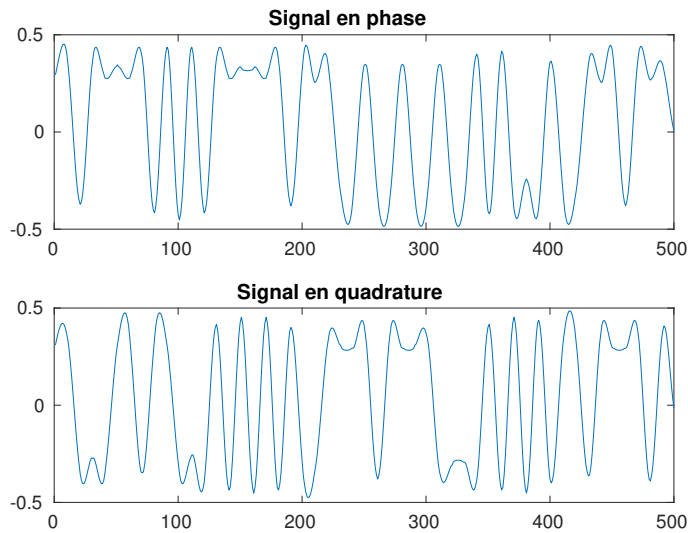


2.1.2 Implantation de la chaîne passe-bas équivalente

On réalise la chaîne de transmission suivante :



On fait le même travail que précédemment sans transposition de fréquence (et donc sans retour en bande de base). On peut tracer les signaux en phase et en quadrature :

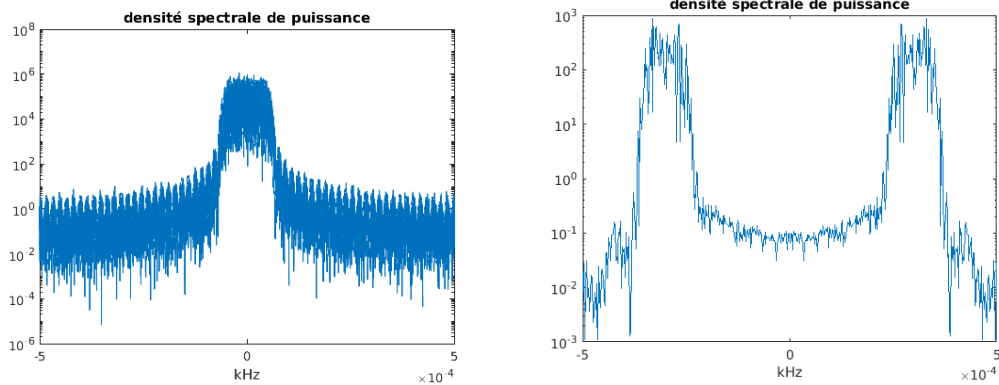


Le TEB sans bruit est bien nul.

Le bruit ajouté est un bruit complexe :

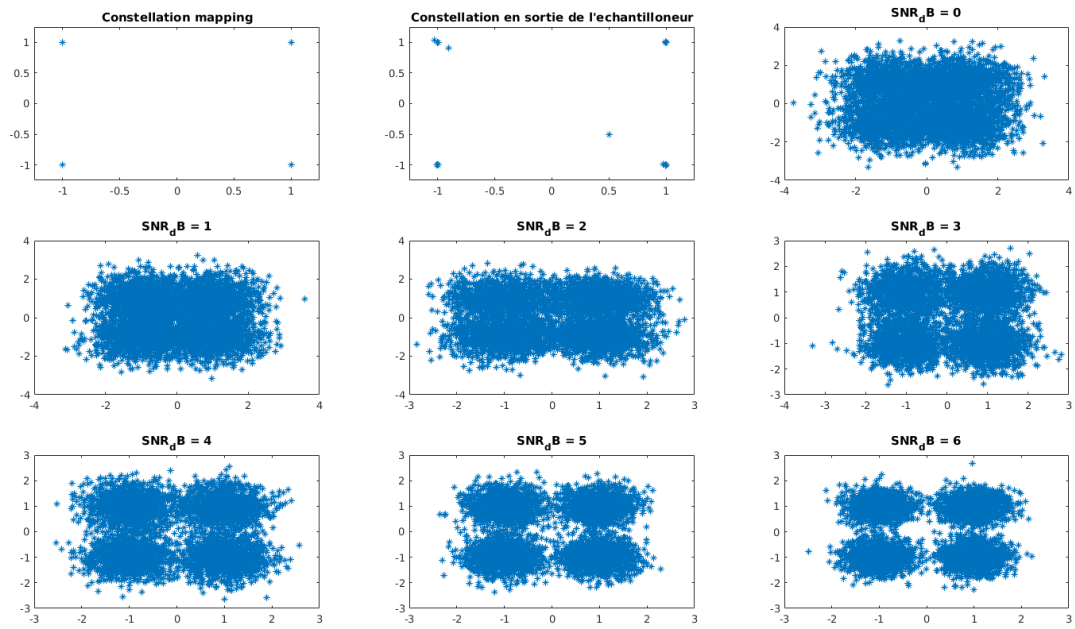
```
% Ajout du bruit
Pb = Pr*Ns/(2*log2(M)*Eb_NO(i));
nI = sqrt(Pb)*randn(1,n*Ns/2);
nQ = nI;
bruit = nI + 1i*nQ;
x_bruit = x_h + bruit;
```

On peut tracer la densité spectrale de puissance de l'enveloppe complexe $x_e(t)$ associée au signal modulé sur fréquence porteuse $x(t)$ (à gauche) que l'on peut comparer à la densité spectrale de puissance de la chaîne précédent (à droite) :

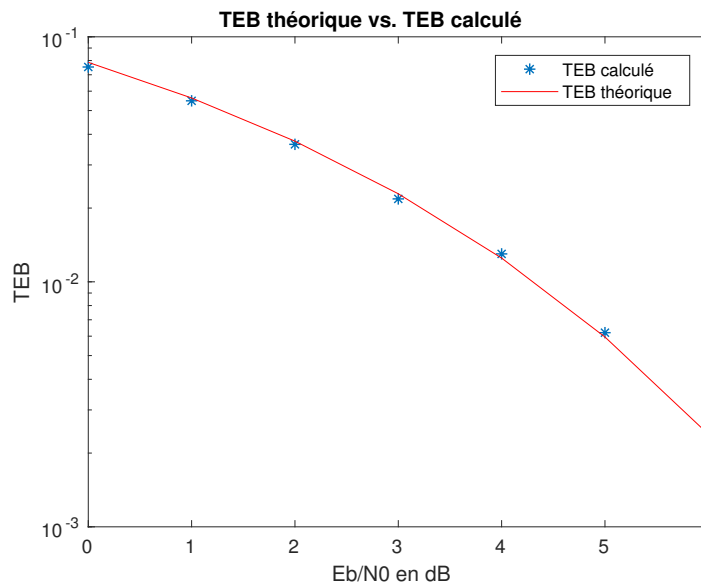


Au niveau de la puissance, on obtient une puissance de 0.1 pour la chaîne sur fréquence porteuse contre 0.2 pour la chaîne équivalente passe-bas, soit le double.

Les constellations obtenues sont les suivantes :



Le TEB théorique est donné dans les planches de cours. En le superposant aux différents TEB calculés on obtient le tracé suivant :



2.2 Comparaison des modulations BPSK, QPSK, 8-PSK et 16-QAM

2.2.1 Constellations

Les constellations ont été tracées avec $n = 1200$ bits

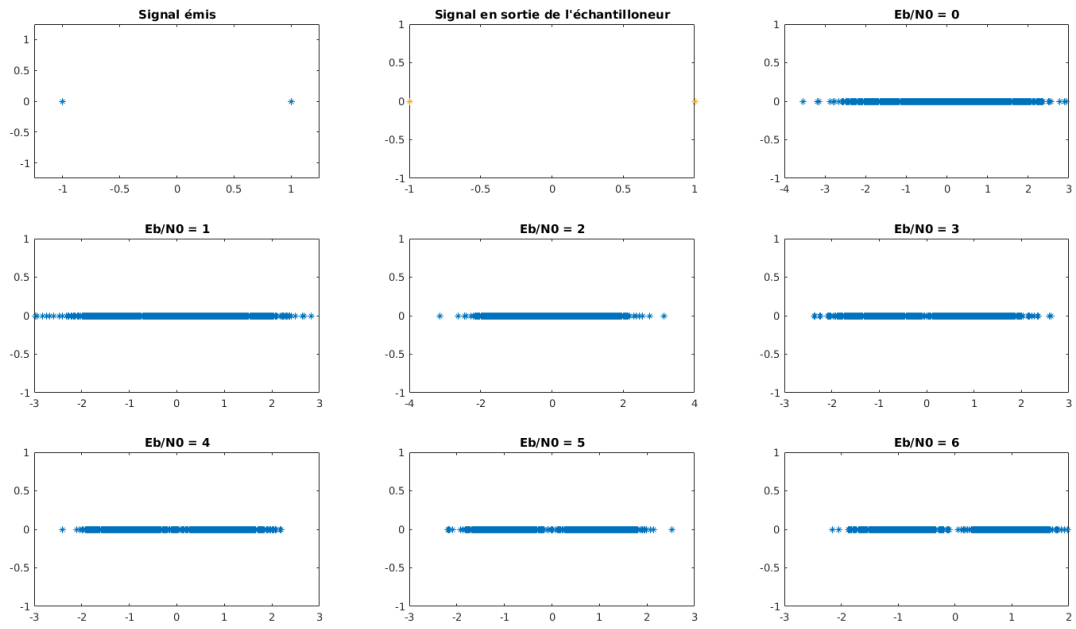


FIGURE 3 – Constellation BPSK

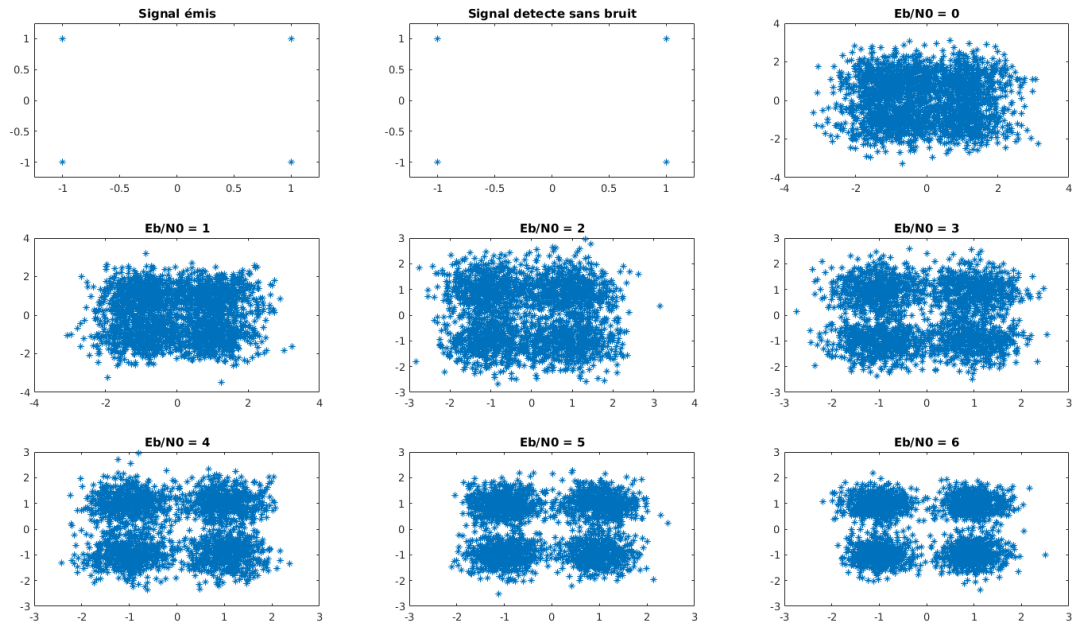


FIGURE 4 – Constellation QPSK

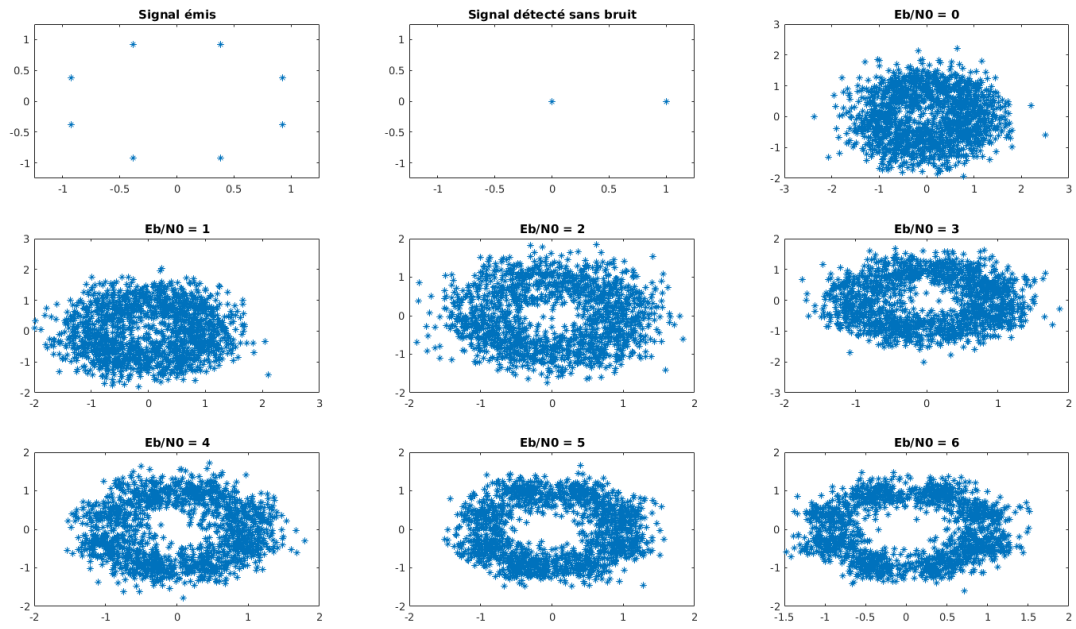


FIGURE 5 – Constellation 8-PSK

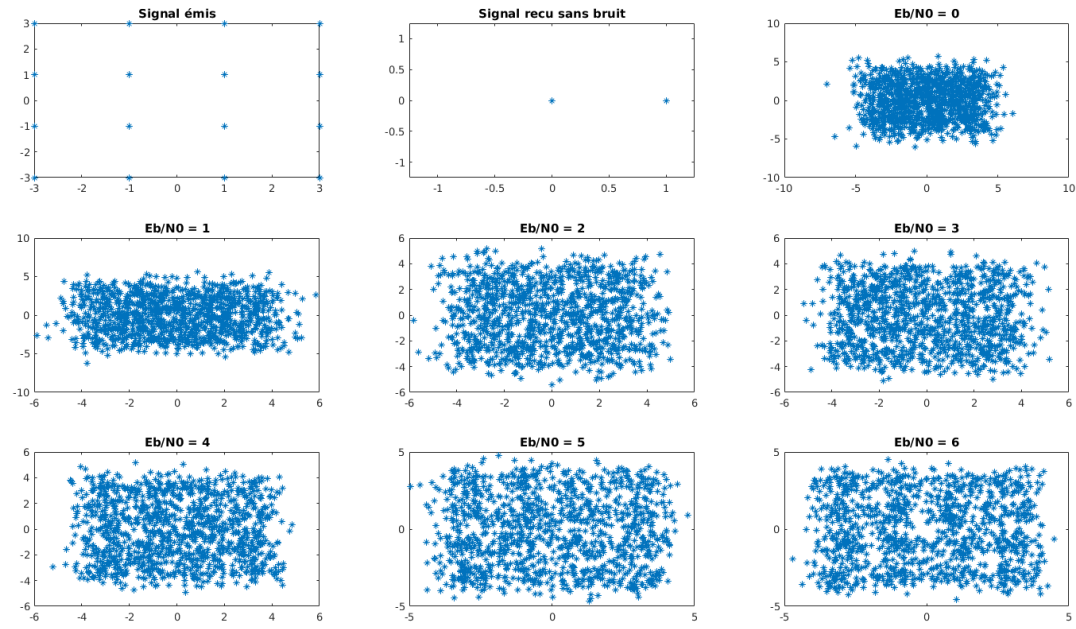
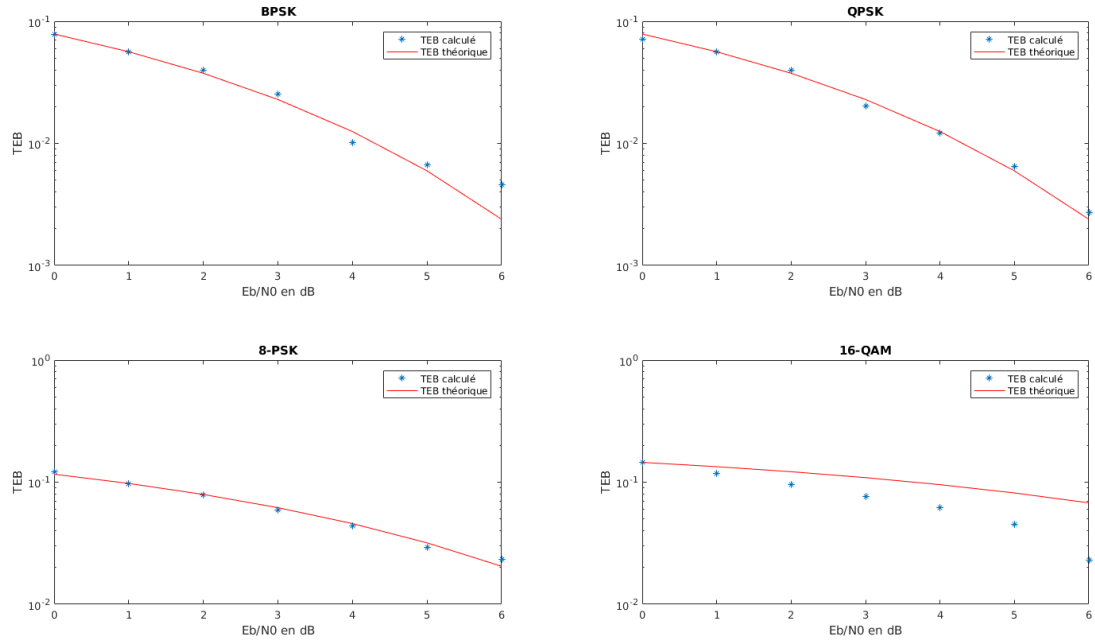


FIGURE 6 – Constellation 16-QAM

2.2.2 TEB

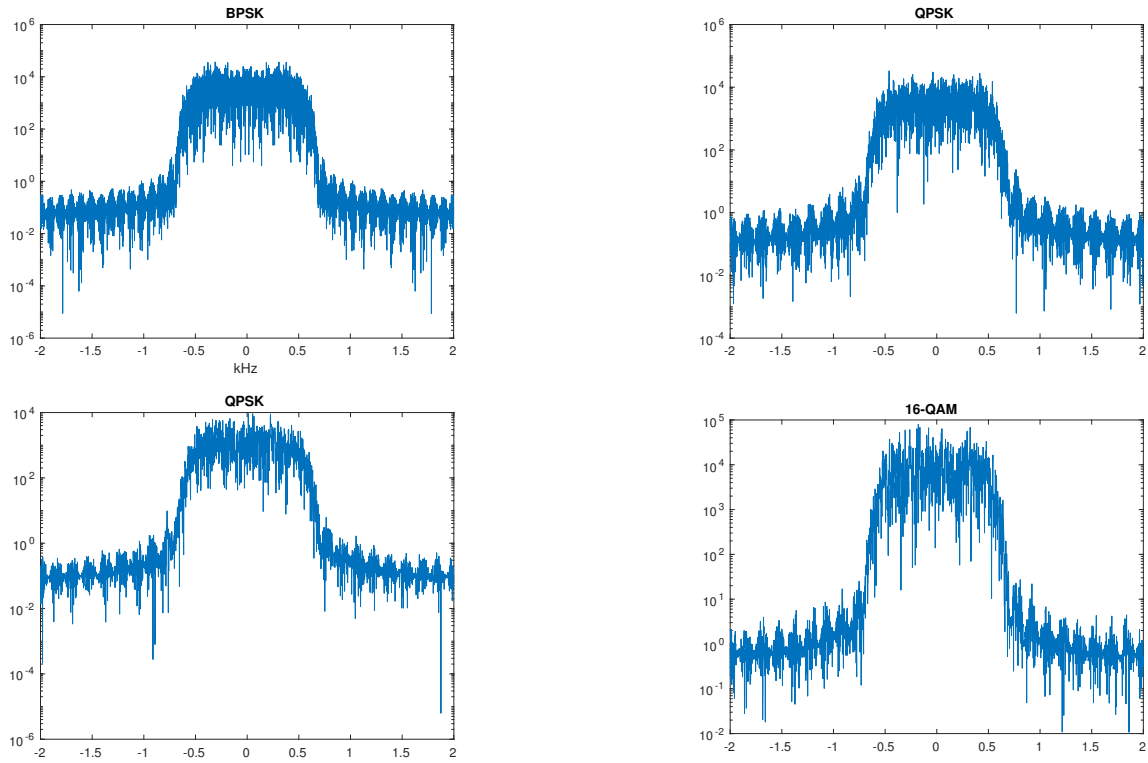


Remarque : Les TEB calculés correspondent globalement bien aux TEB théoriques (hormis peut être pour le 16 QAM).

Détail des calculs des TEBs :

- $TEB_{BPSK} = Q(\sqrt{2\frac{E_b}{N_0}})$
- $TEB_{QPSK} = Q(\sqrt{2\frac{E_b}{N_0}})$
- $TEB_{8-PSK} = \frac{2}{\log 2(M)} \times Q(\sqrt{2\frac{E_b}{N_0} \log 2(M) \sin(\frac{\pi}{M})})$
- $TEB_{16-QAM} = \frac{4}{\log 2(M)} \times (1 - \frac{1}{\sqrt{(M)}}) Q(\sqrt{\frac{3\log 2(M)E_b}{(M-1)N_0}})$

2.2.3 Efficacité spectrale

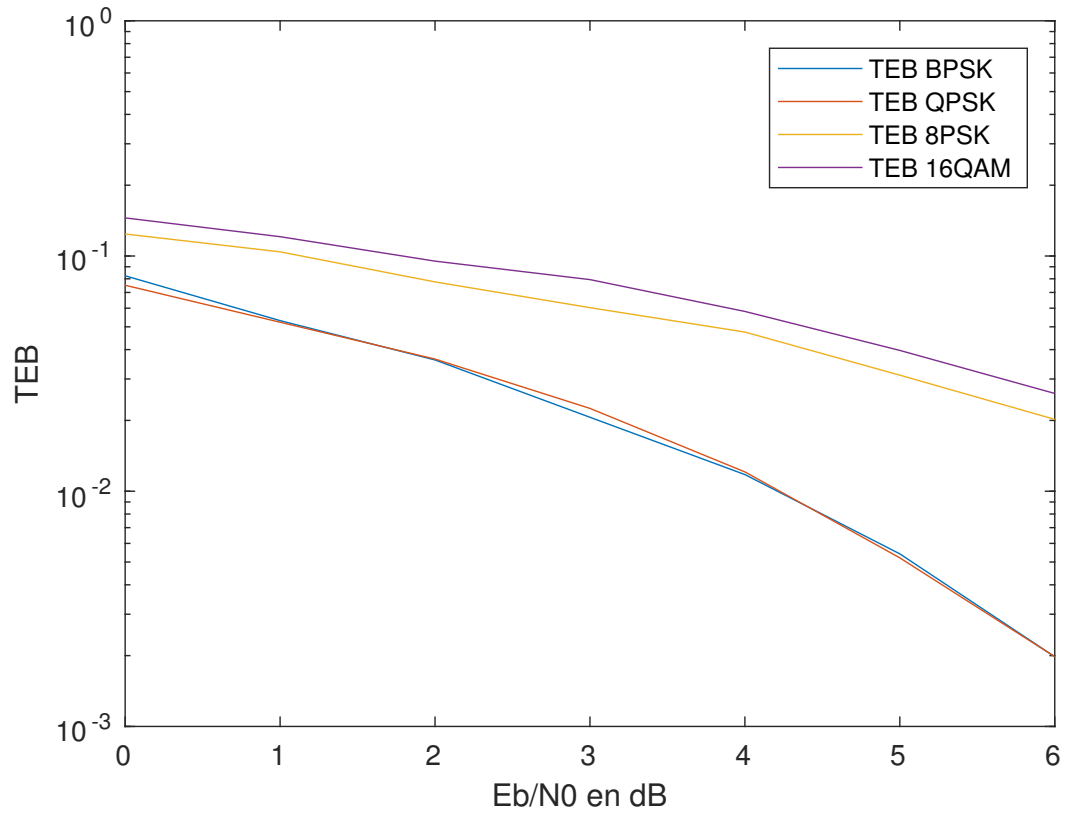


On re

FIGURE 7 – Densité spectrale de puissance des différents mappings

Les modulations semblent assez équivalente spectralement, même si la 16-QAM est peut être légèrement plus efficace.

2.2.4 Efficacité en puissance



remarque que pour une même valeur de TEB donnée, le SNR_{dB} le plus faible correspond aux modulations BPSK et QPSK, ce sont donc les deux modulations les plus efficaces spectralement.

On