



# TELECOMMUNICATION

Etude des chaines de transmission sur fréquence porteuse

Première Année, Département SN



Issam HABIBI  
Younes SAOUDI

2019-2020

# Contents

<b>1</b>	<b>Utilisation de la chaîne passe-bas équivalente pour le calcul et l'estimation du taux d'erreur binaire</b>	<b>2</b>
1.1	Etude Théorique . . . . .	2
1.2	Première chaîne à étudier : "Chaîne sur Fréquence Porteuse" Implantation sous Matlab . . .	4
1.3	Deuxième chaîne à étudier : "Chaîne Passe-Bas Equivalente" Implantation sous Matlab . . .	7
<b>2</b>	<b>Comparaison de modulations sur fréquence porteuse</b>	<b>12</b>
2.1	Etude Théorique . . . . .	12
2.2	Troisième chaîne à étudier : "Chaîne de Transmission 4-ASK" Implantation sous Matlab . . .	16
2.3	Quatrième chaîne à étudier : "Chaîne de Transmission QPSK" Implantation sous Matlab . . .	18
2.4	Cinquième chaîne à étudier : "Chaîne de Transmission 8-PSK" Implantation sous Matlab . .	21
2.5	Sixième chaîne à étudier : "Chaîne de Transmission 16-QAM" Implantation sous Matlab . . .	24
<b>3</b>	<b>Code Source</b>	<b>28</b>
3.1	Chaîne Fréquence sur Porteuse . . . . .	28
3.2	Chaîne Passe-Bas Equivalente . . . . .	31
3.3	Chaîne 4-ASK . . . . .	34
3.4	Chaîne Q-PSK . . . . .	36
3.5	Chaîne 8-PSK . . . . .	39
3.6	Chaîne 16-QAM . . . . .	41

# Utilisation de la chaîne passe-bas équivalente pour le calcul et l'estimation du taux d'erreur binaire

## 1.1 Etude Théorique

**1.** Calculer le symbole  $E_s$  à l'entrée du récepteur. Attention  $E_s$  représente la véritable énergie reçue, c'est à dire qu'elle doit être calculée à partir de la véritable puissance du signal reçu, pas à partir de celle de l'enveloppe complexe associé.

On rappelle la formule suivante:

$$S_x(f) = \frac{\sigma_a^2}{T_s} |H(f)|^2 + 2 \frac{\sigma_a^2}{T_s} |H(f)|^2 \sum_{n=1}^{\infty} \text{Re}[R_a(k) \exp(j2\pi f k T_s)] + \frac{|m_a|^2}{T_s^2} \sum_k |H(\frac{k}{T_s})|^2 \delta(f - \frac{k}{T_s})$$

Sachant que  $\sigma^2 = 1$  et  $m_a = 0$  on a :  $S_{xe}(f) = \frac{\sigma_d^2}{T_s} |H(f)|^2$

Donc  $S_{xe}(f) = \frac{1}{T_s} G(f)$

Or, le signal est de la forme :  $x_e(t) = I(t) + jQ(t)$

Donc la puissance est :  $P = \int_{\mathbf{R}} S_x(f) df = \frac{1}{4} \int_{\mathbf{R}} S_{xe}(f - f_p) + S_{xe}(f + f_p) df = \frac{1}{2} \int_{\mathbf{R}} S_{xe}(f) df$

D'où  $P = \frac{1}{2} \int_{\mathbf{R}} \frac{1}{T_s} |G(f)| df = \frac{1}{2T_s} \int_{\frac{1+\alpha}{2T_s}}^{\frac{\alpha-1}{2T_s}} |G(f)| df$

Finalement,  $P = \frac{1}{2T_s}$

donc  $E_s = PT_s = \frac{1}{2}$

**2.** Calculer la puissance du bruit sur chaque voie (I et Q) en sortie du filtre de réception.

Le bruit complexe associé équivalent au bruit introduit par le canal de propagation et filtré sur la bande du signal modulé prend la forme :  $n_e(t) = n_I(t) + jn_Q(t)$

Où  $S_{ne}(f) = 4S_n(f + f_p)U(f + f_p) = N_0$

Il a la même puissance sur chaque voie :  $\sigma_n^2 = \sigma_n^2 = N_0 F_e$

**3.** les deux voies I, Q étant indépendantes, donner le taux d'erreur symbole de la modulation QPSK en fonction de ceux des voies I et Q ( $TES_I$  et  $TES_Q$ )

ici on a  $M = 4$  donc  $QPSK = 4 - PSK$

donc  $TES = 2Q(\sqrt{\frac{2E_s}{N_0}} \sin(\frac{\pi}{4})) = 2Q(\sqrt{\frac{2E_s}{N_0}} \frac{\sqrt{2}}{2}) = 2Q(\sqrt{\frac{E_s}{N_0}})$

Par indépendance des deux voies  $\boxed{TES_I = TES_Q = Q(\sqrt{\frac{E_s}{N_0}})}$

**4.** En supposant les termes du deuxième ordre négligeables ( $TES_I * TES_Q = 0$ ), donner le taux d'erreur symbole de la modulation QPSK en fonction de  $TES_I$  uniquement.

Selon les hypothèses :  $\boxed{TES = 2TES_I}$

**5.** Déterminer  $TES_I$  en fonction de  $\frac{E_s}{N_0}, E_s$  correspondant à la véritable énergie reçue. On supposera que les instants d'échantillonnage et l'organe de décision sont optimaux.

Selon les hypothèses:  $\boxed{TES_I = \frac{TES}{2} = Q(\sqrt{\frac{E_s}{N_0}})}$

**6.** En déduire le taux d'erreur binaire de la chaîne de transmission QPSK en fonction de  $\frac{E_b}{N_0}$

On sait que  $TEB = \frac{TES}{\log_2(M)} = \frac{TES}{2} = Q(\sqrt{\frac{E_s}{N_0}})$

D'autre part,  $E_s = \log_2(M)E_b = 2E_b$

Il vient donc :  $\boxed{TEB = Q(\sqrt{\frac{2E_b}{N_0}})}$

## 1.2 Première chaine à étudier : "Chaine sur Fréquence Porteuse" Implantation sous Matlab

1. Génération de l'information binaire à transmettre avec un MAPPING de Grey

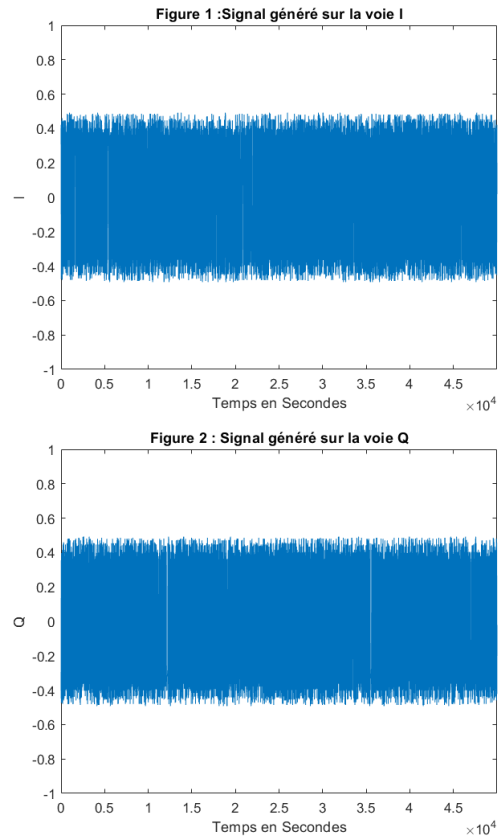
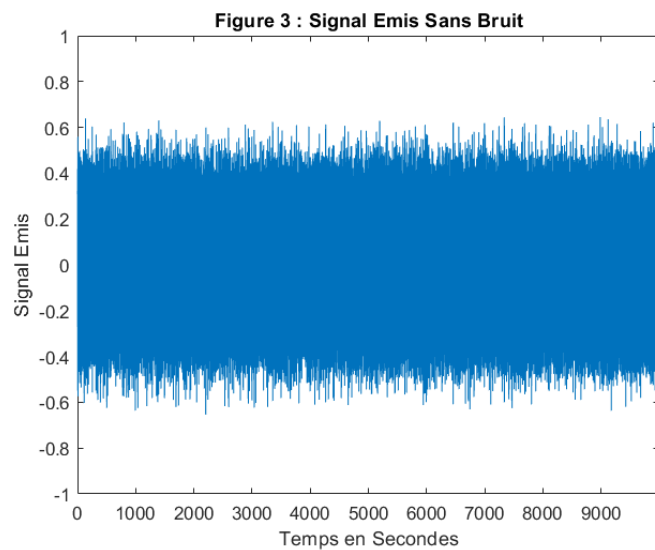
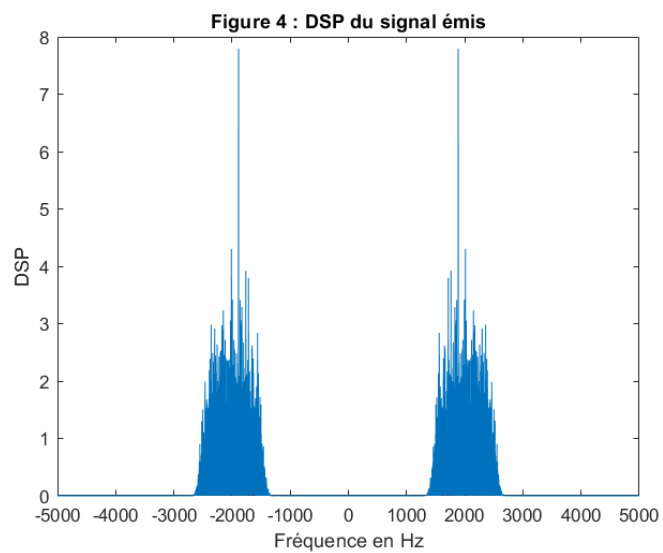


Figure 1.1: Signal tracé dans la voie I et en quadrature dans la voie Q

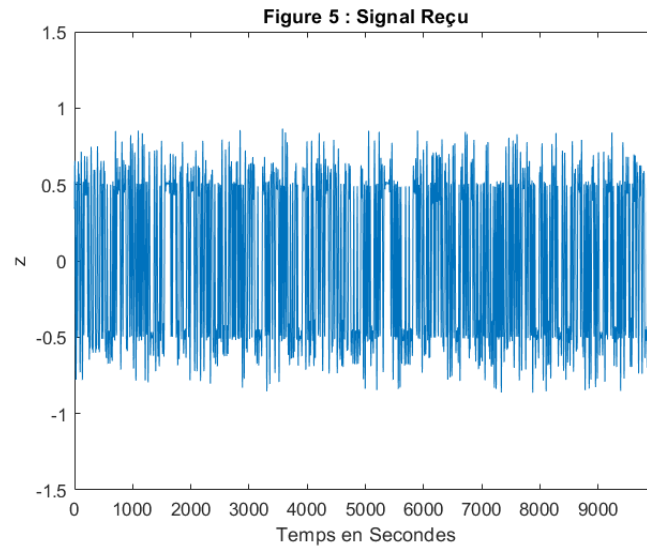
2. Emission du signal



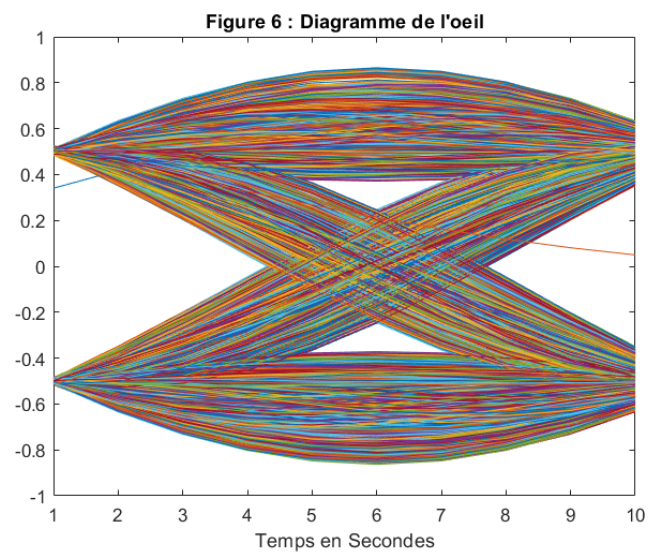
3. La densité spectrale du signal, générée par périodogramme



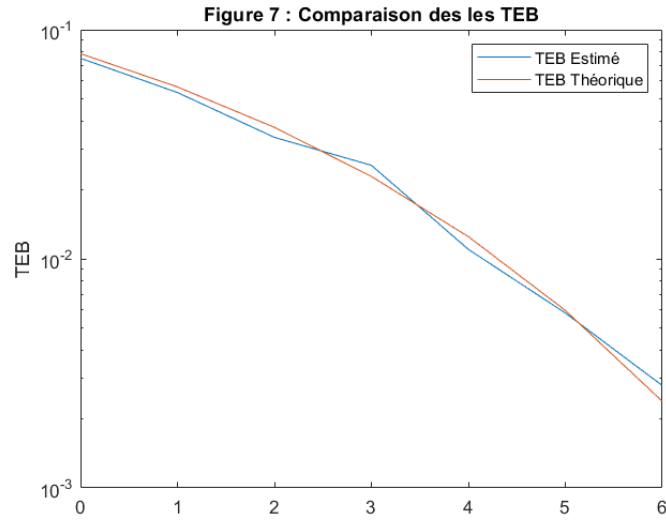
4. Signal Reçu à la sortie du filtre de réception



**5.** Diagramme de l'oeil en sortie du filtre de réception



**6.** Comparaison des TEB théorique et expérimental



### 1.3 Deuxième chaîne à étudier : "Chaine Passe-Bas Equivalente" Implantation sous Matlab

1. Génération de l'information binaire à transmettre avec un MAPPING tel que les symboles sont  $d_k = \pm 1 \pm j$



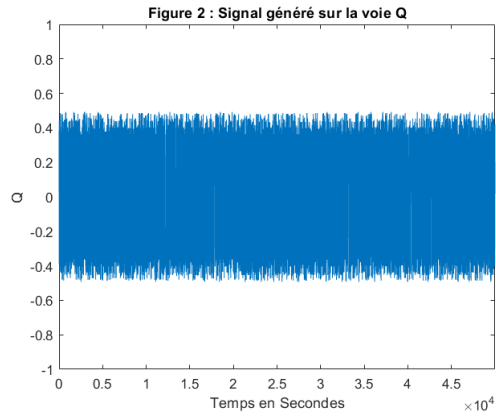
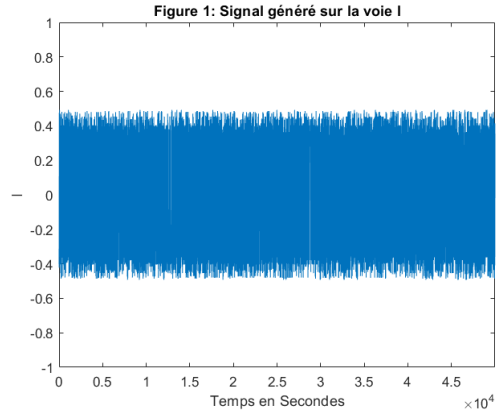
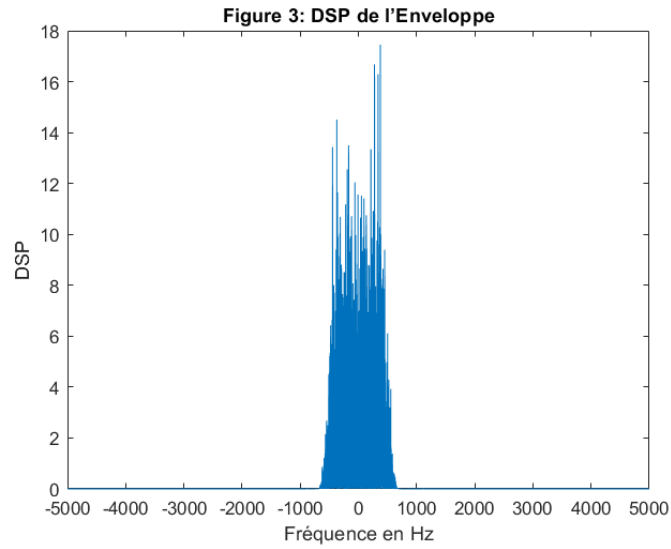
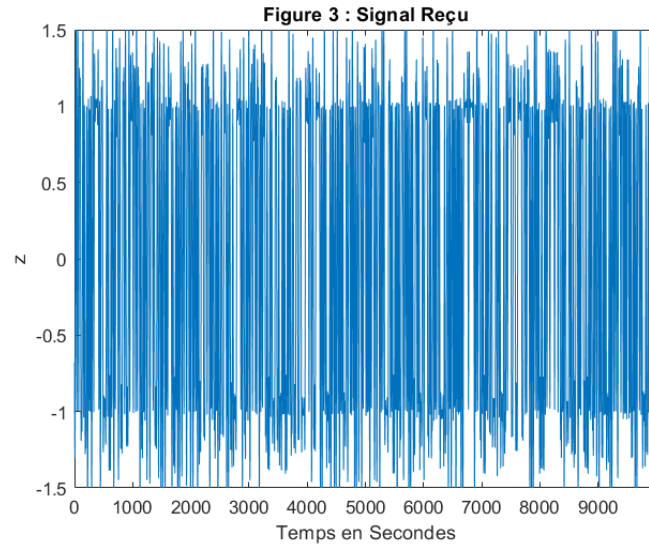


Figure 1.2: Signal tracé dans la voie I et en quadrature dans la voie Q

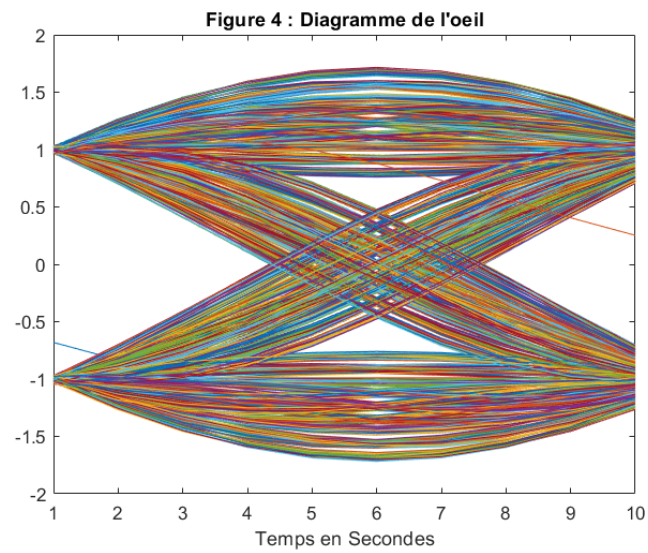
2. La Densité Spectrale de Puissance par périodogramme de l'enveloppe complexe  $x_e$



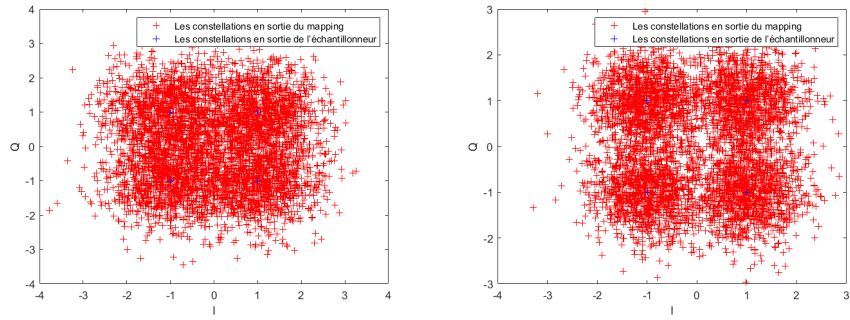
3. Signal Reçu à la sortie du filtre de réception



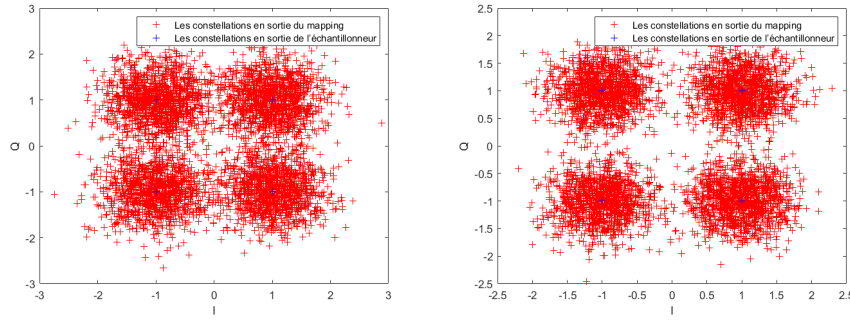
3. Le diagramme de l'oeil à la sortie du filtre de réception.



5. Les constellations en sortie du Mapping et de l'échantillonneur pour une valeur donnée de  $\frac{E_b}{N_0}$  :



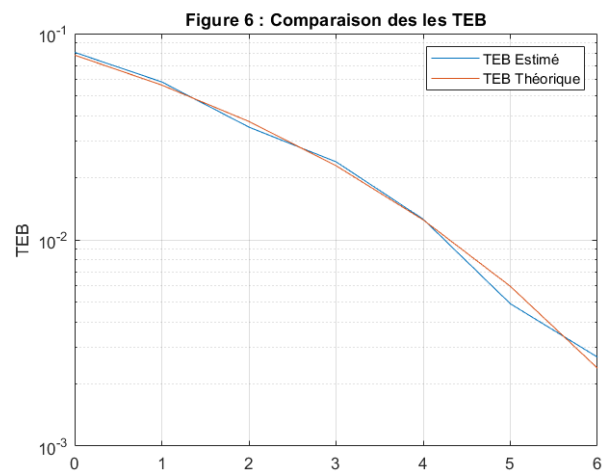
(a) Caption



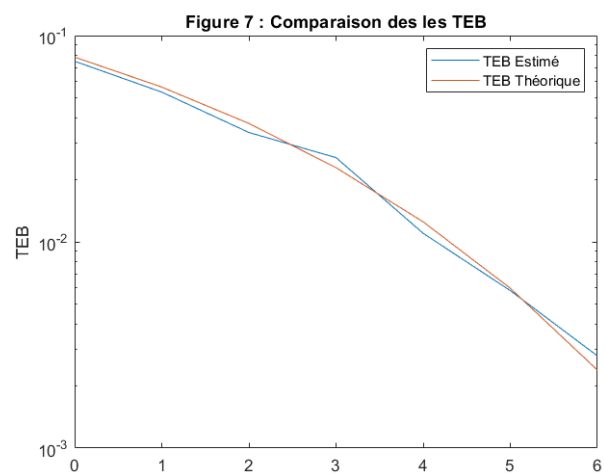
(b) Caption

Figure 1.3: Caption

**6.** Comparaison des TEB théorique et expérimental et comparaison du TEB de la chaîne passe-bas équivalente et celui de la chaîne fréquence sur porteuse



(a) Passe-bas équivalente



(b) Fréquence sur porteuse

# Comparaison de modulations sur fréquence porteuse

## 2.1 Etude Théorique

1. Tracer les constellations des quatres modulations considérées.

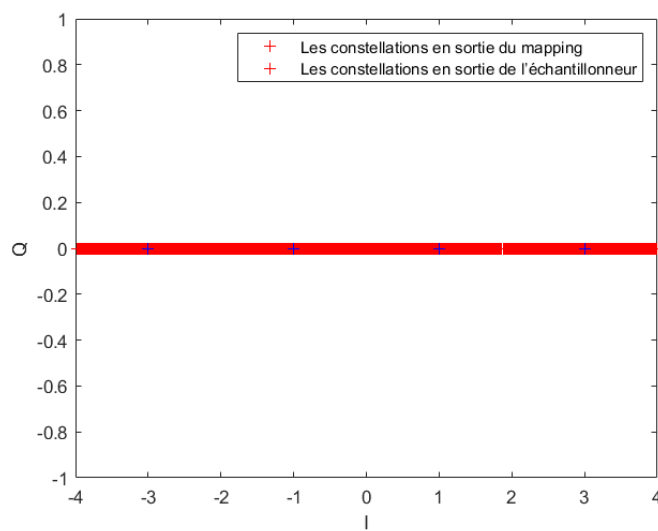


Figure 2.1: 4-ASK

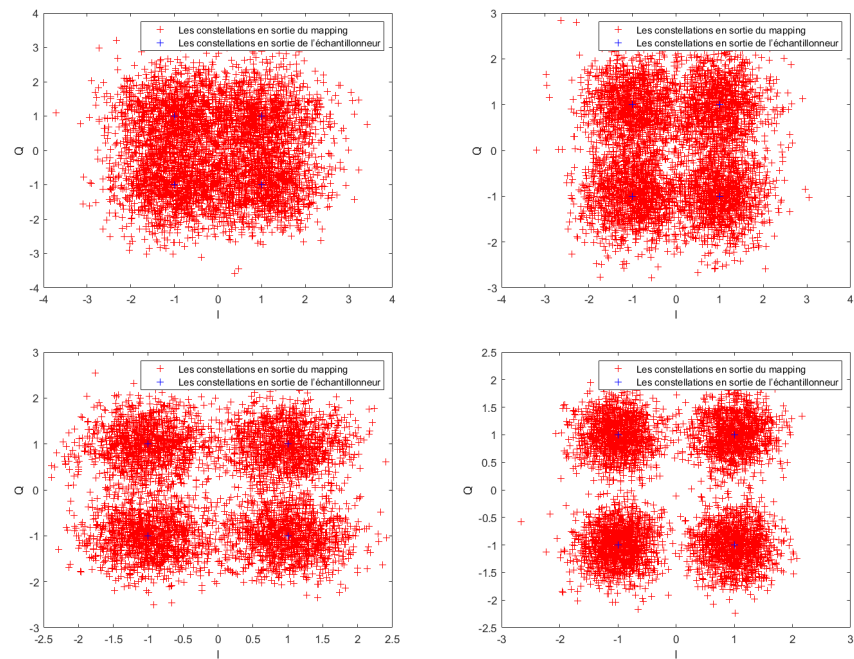


Figure 2.2: QPSK

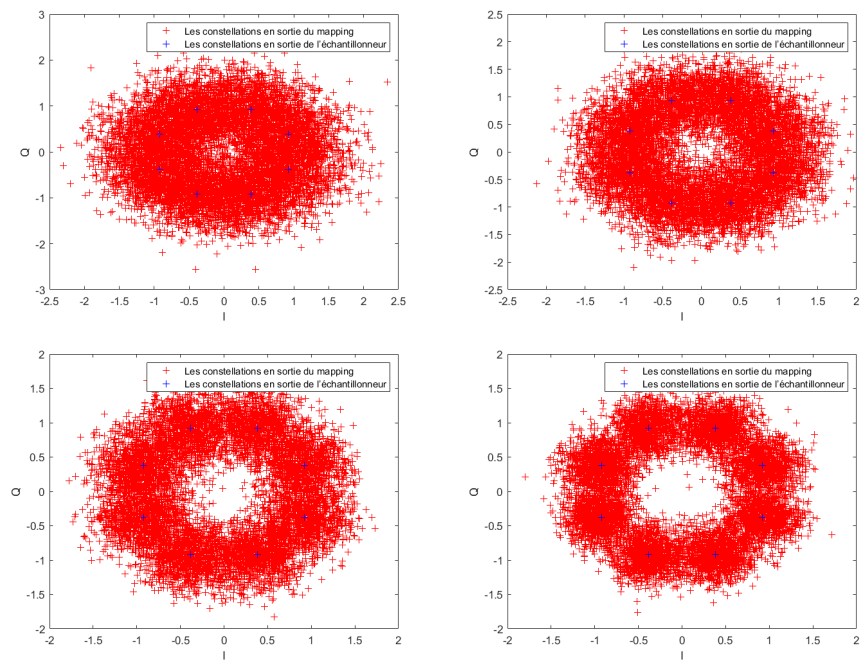


Figure 2.3: 8-PSK

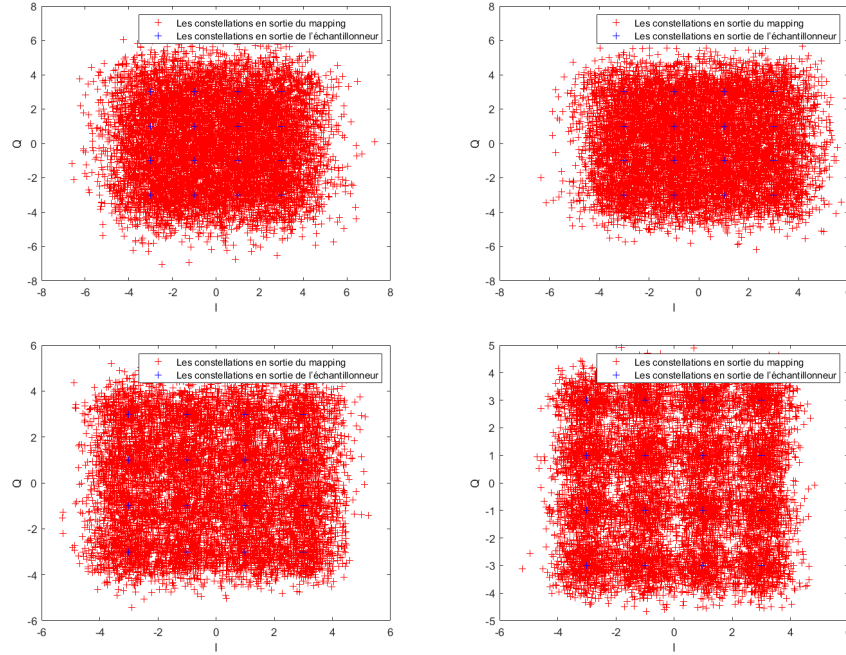


Figure 2.4: 16-QAM

**2.** Déterminer le débit symbole  $R_s$  dans les quatres cas.

La formule générale du débit symbole est :  $R_s = \frac{R_b}{\log_2(M)}$

Pour la chaine 4-ASK :  $R_s = \frac{48000}{2} = 24000$

Pour la chaine QPSK :  $R_s = \frac{48000}{2} = 24000$

Pour la chaine 8-PSK :  $R_s = \frac{48000}{3} = 16000$

Pour la chaine 16-QAM :  $R_s = \frac{48000}{4} = 12000$

**3.** Calculer les efficacités spectrales des quatres transmissions proposées. Quelle est la transmission la plus efficace spectralement ? Qu'est cela veut dire ?

On note  $e$  l'efficacité spectrale.  $e = \frac{R_b}{B} = \frac{R_b}{kR_s} = \frac{\log_2(M)}{k}$

Or dans le cas du cosinus surlevée  $B = (1 + \alpha)R_s$  et  $B = \frac{\log_2(M)}{k}$  donc  $k = 1 + \alpha$

Pour la chaine 4-ASK :  $e = \frac{2}{1.5} = 1.3bit$

Pour la chaine QPSK :  $e = \frac{2}{1.5} = 1.3bit$

Pour la chaine 8-PSK :  $e = \frac{3}{1.5} = 2bit$

Pour la chaine 16-QAM :  $e = \frac{4}{1.5} = 2.6bit$

La transmission 16-QAM est la plus efficace spectralement, ceci était prévisible car elle possède le nombre

de symboles le plus élevé.

**4-a.** Déduire les valeurs de  $\frac{E_b}{N_0}$  pour satisfaire à la spécification du TEB. Quel est le système le plus efficace en terme de puissance ? Justifier votre réponse

Sachant que  $TEB = 10^{-2}$  on obtient :

Pour la chaine 4-ASK :  $\frac{E_b}{N_0} = 8.3Db$

Pour la chaine QPSK :  $\frac{E_b}{N_0} = 4.4Db$

Pour la chaine 8-PSK :  $\frac{E_b}{N_0} = 7.5Db$

Pour la chaine 16-QAM :  $\frac{E_b}{N_0} = 8.3Db$

Donc QPSK est la chaine la plus efficace en terme de puissance car elle possède le  $\frac{E_b}{N_0}$  le plus faible.

**4-b.** La chaine de transmission utilisant la modulation 4-ASK et la chaine de transmission utilisant la modulation 16-QAM présentent le même taux d'erreur binaire. Qu'est ce qui pourrait justifier le choix de l'une ou l'autre?

Puisque les deux chaines ont le même efficacité en terme de puissance, le facteur décisif serait l'efficacité spectrale la plus forte qui est en faveur de la chaine utilisant la modulation 16-QAM.

**5.** Si on souhaitait réaliser la transmission à travers un canal de propagation supposé à bruit additif blanc Gaussien (AWGN) de bande passante  $20kHz$ , serait il possible de réaliser chaque transmission proposée en trouvant, au niveau du récepteur, un instant optimal d'échantillonnage sans interférence entre symboles ? Expliquez votre réponse

Ceci est possible , il suffit de trouver un instant qui vérifie le critère de Nyquist en domaine fréquentiel suivant:

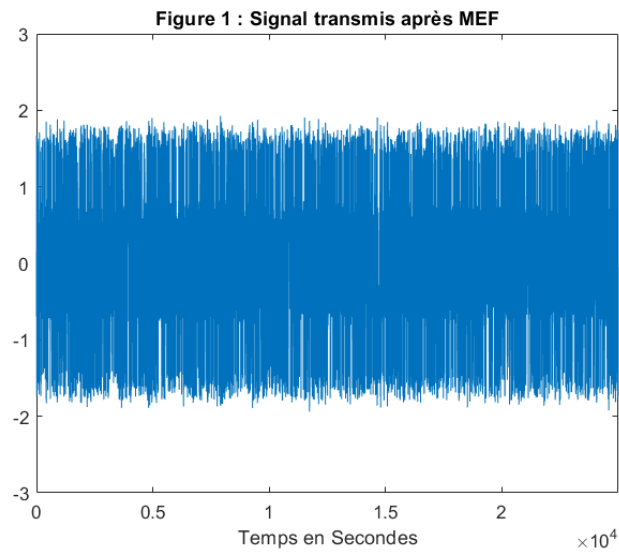
$$\sum_k G_0^t(f - \frac{k}{T_s}) = cste$$

ce qui correspond à  $(1 + \alpha)R_s \leq 20kHz$

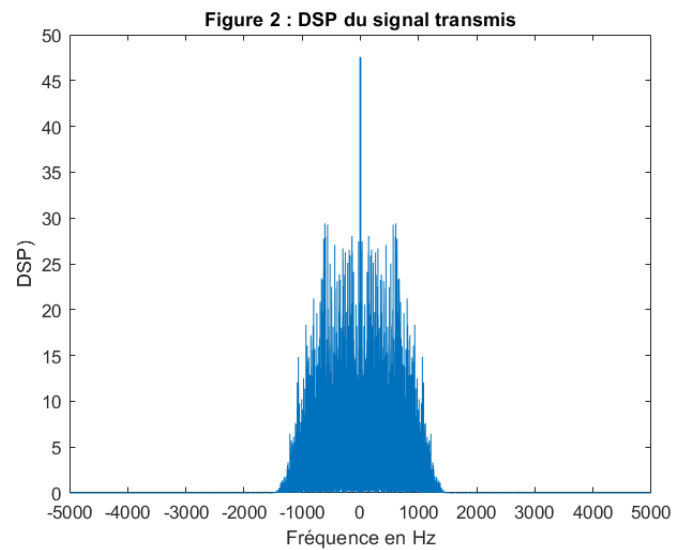


## 2.2 Troisième chaine à étudier : "Chaîne de Transmission 4-ASK" Implantation sous Matlab

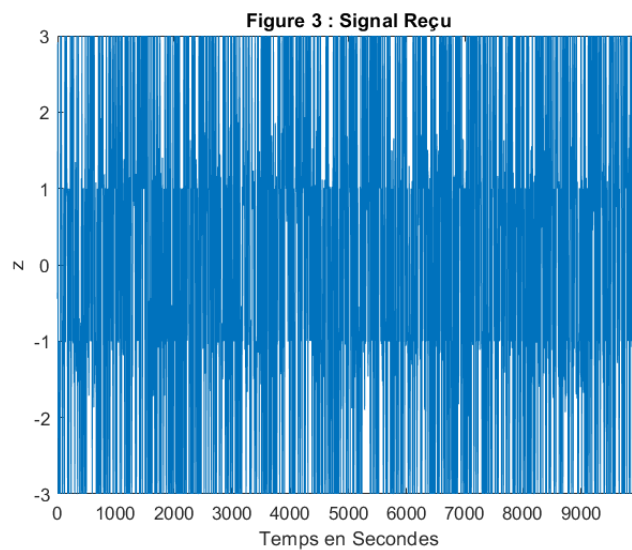
1. Génération de l'information binaire à transmettre avec un MAPPING de Grey



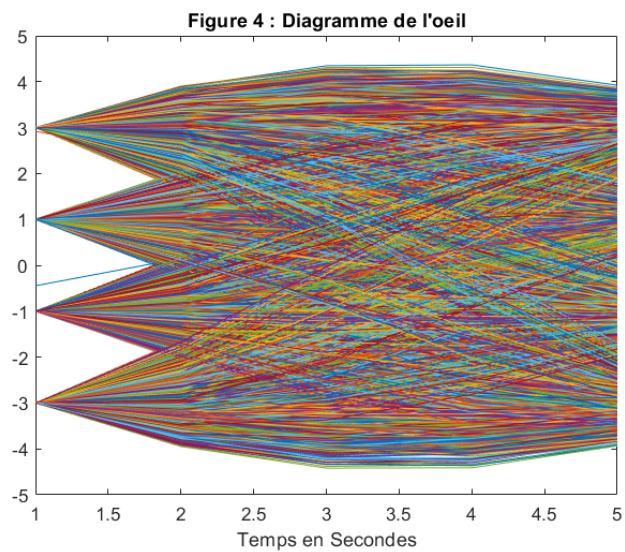
2. Densité spectrale de puissance par périodogramme



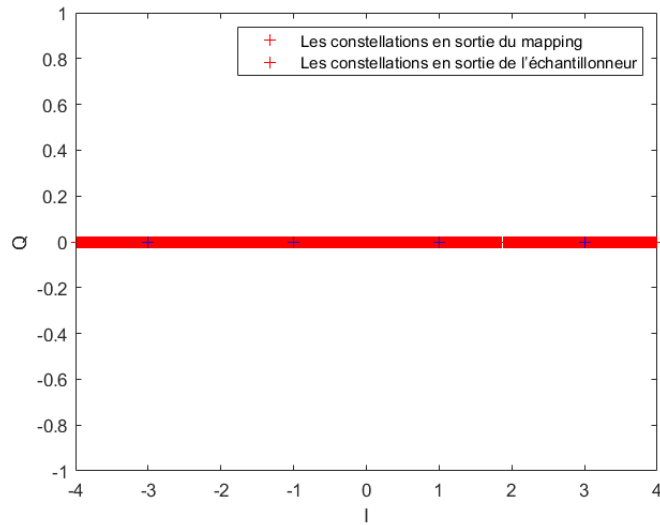
3. Signal reçu à la sortie du filtre de réception



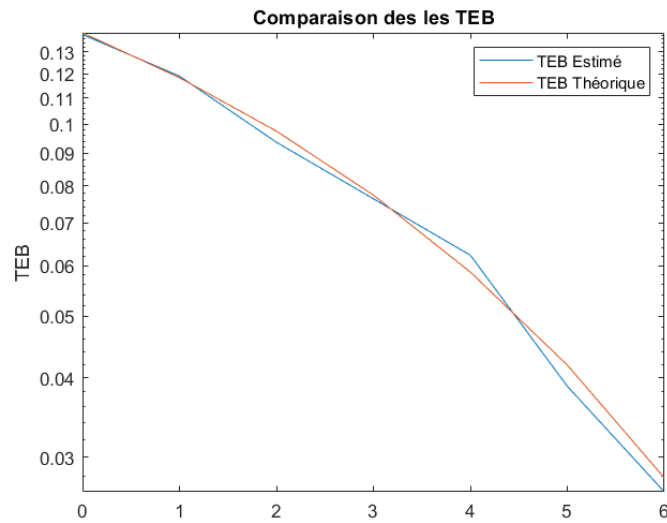
4. Diagramme de l'oeil du signal reçu à la sortie du filtre de réception



4. Les constellation en sortie du Mapping et de l'échantillonneur.



## 5. Comparaison des TEB théorique et expérimental



## 2.3 Quatrième chaine à étudier : "Chaîne de Transmission QPSK" Implantation sous Matlab

1. Génération de l'information binaire à transmettre avec un MAPPING tel que les symboles sont  $d_k = \pm 1 \pm j$

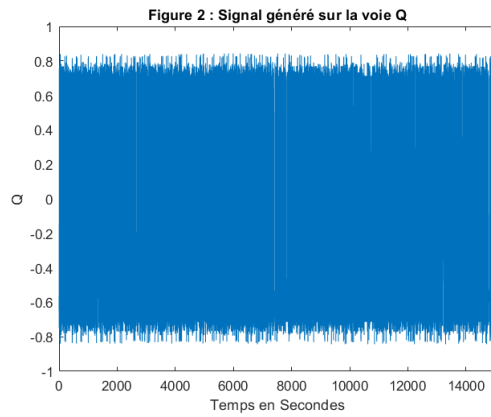
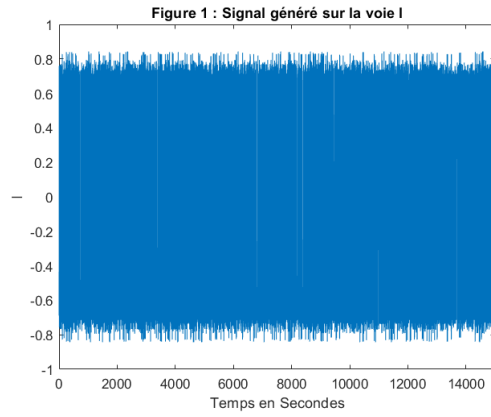
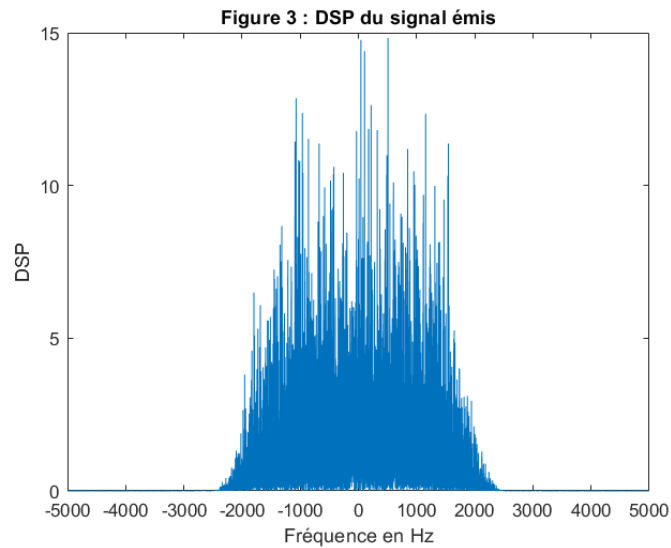
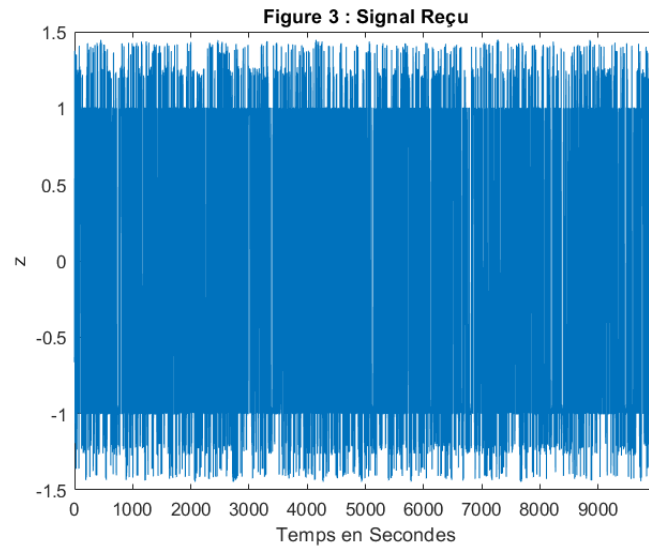


Figure 2.5: Signal tracé dans la voie I et en quadrature dans la voie Q

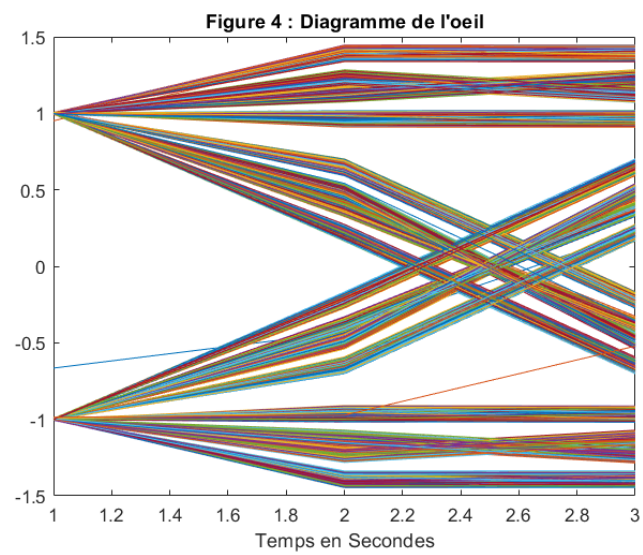
### 3. Densité spectrale de puissance par périodogramme



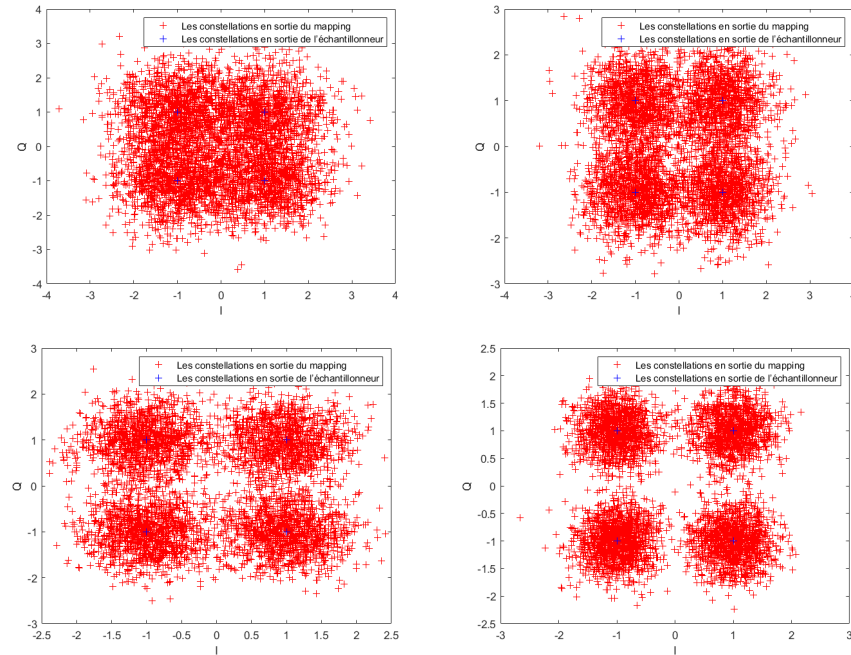
### 4. Signal reçu à la sortie du filtre de réception



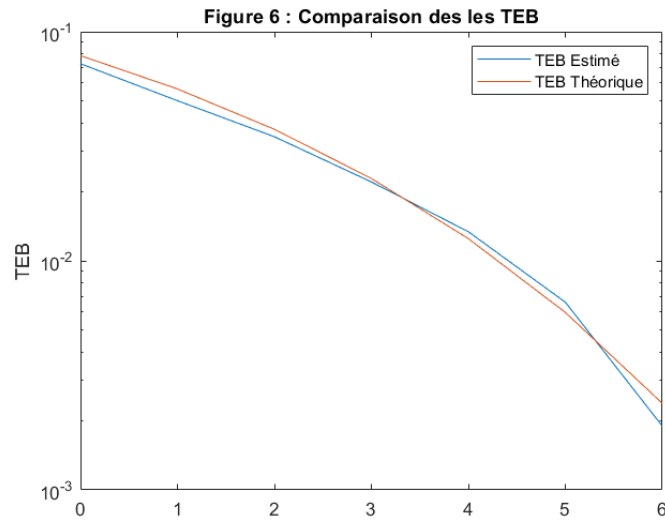
4. Le diagramme de l'oeil du signal à la sortie du filtre de réception.



5. Constellation à la sortie du mapping et de l'échantillonneur



## 6. Comparaison des TEB théorique et expérimental



## 2.4 Cinquième chaine à étudier : "Chaîne de Transmission 8-PSK" Implantation sous Matlab

1. Génération de l'information binaire à transmettre avec un MAPPING qui utilise `pskmod`

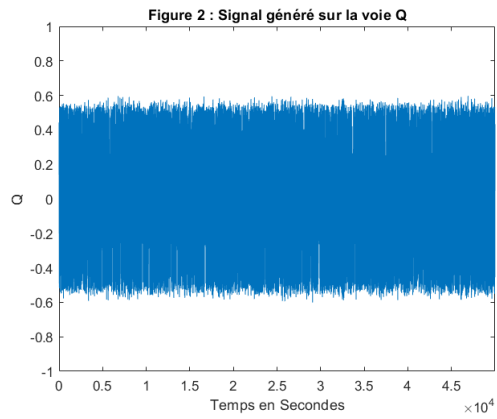
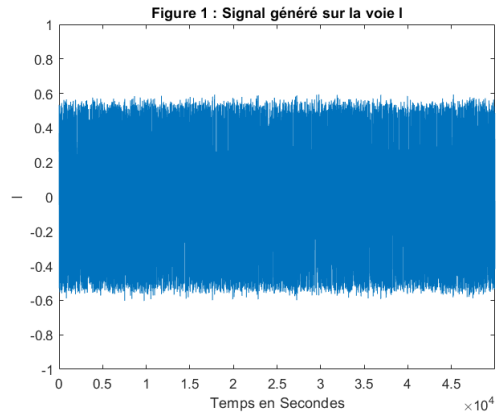
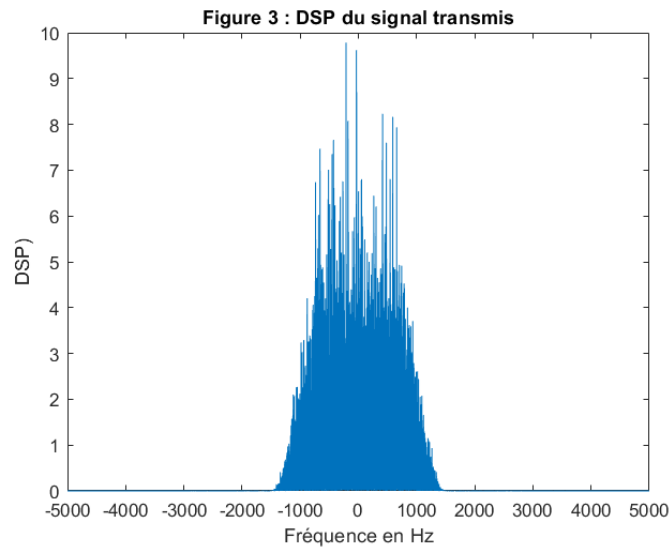
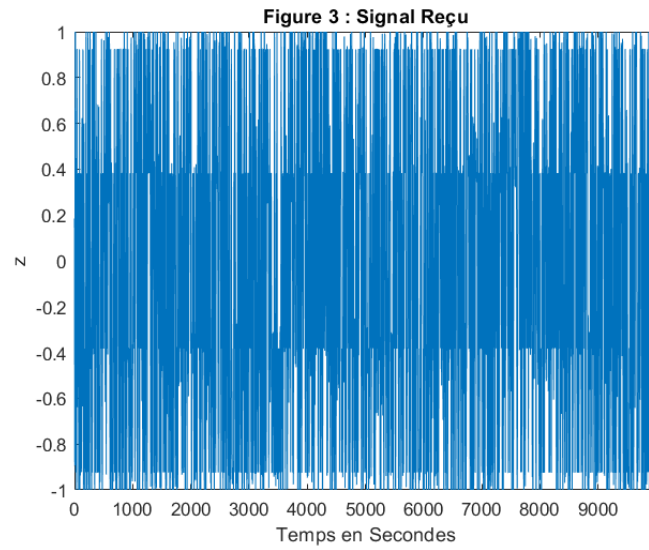


Figure 2.7: Signal tracé dans la voie I et en quadrature dans la voie Q

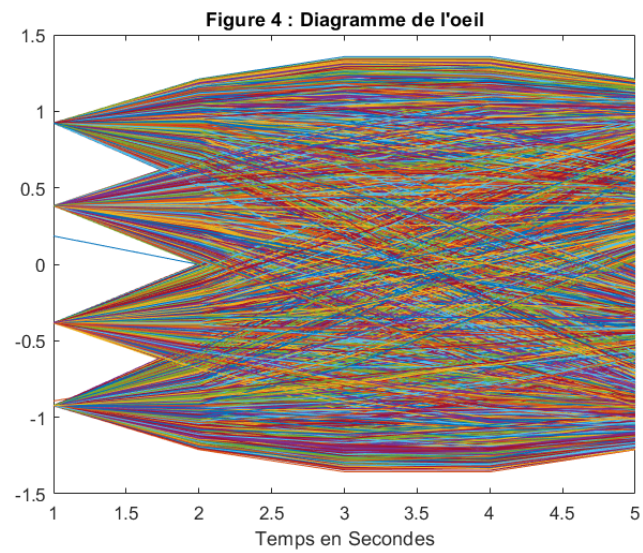
### 3. Densité spectrale de puissance par périodogramme



### 4. Signal reçu à la sortie du filtre de réception

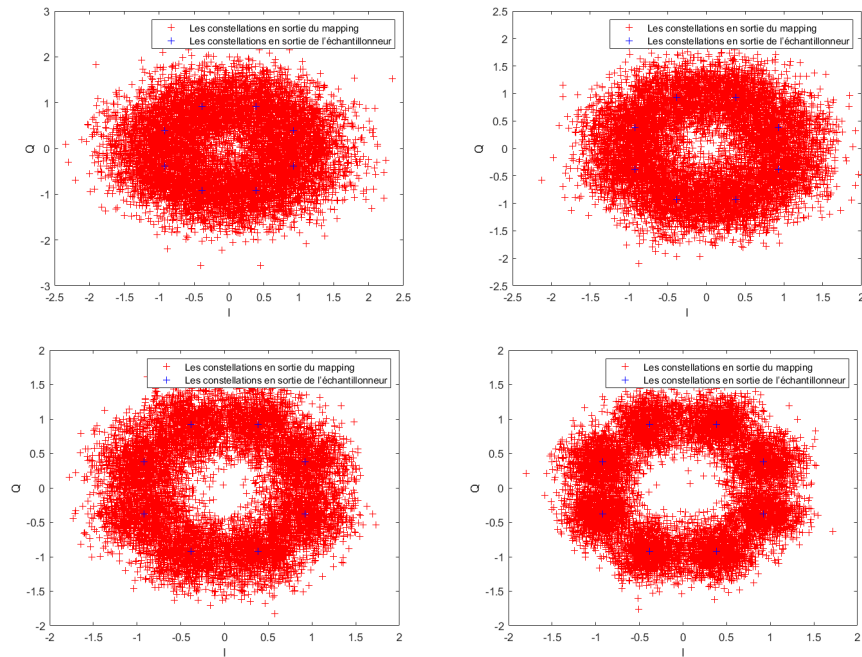


4. Diagramme de l'oeil du signal à la sortie du filtre de réception

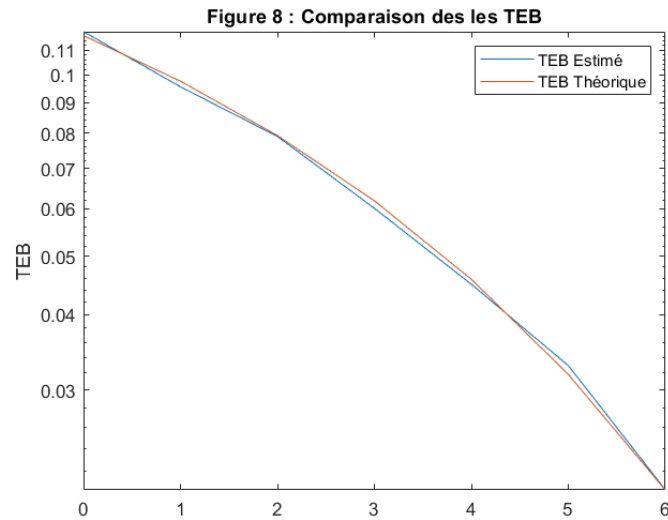


5. Constellations à la sortie du Mapping et de l'échantillonneur





## 6. Comparaison des TEB théorique et expérimental



## 2.5 Sixième chaine à étudier : "Chaîne de Transmission 16-QAM" Implantation sous Matlab

1. Génération de l'information binaire à transmettre avec un MAPPING qui utilise `qammod`

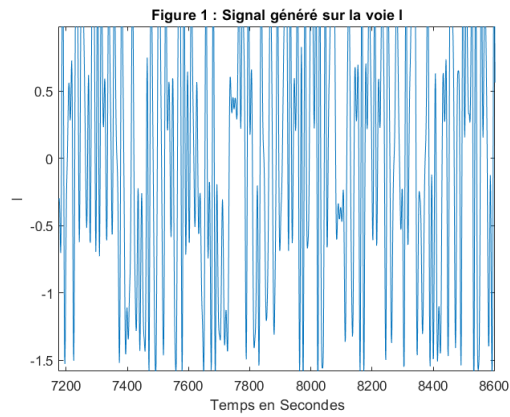
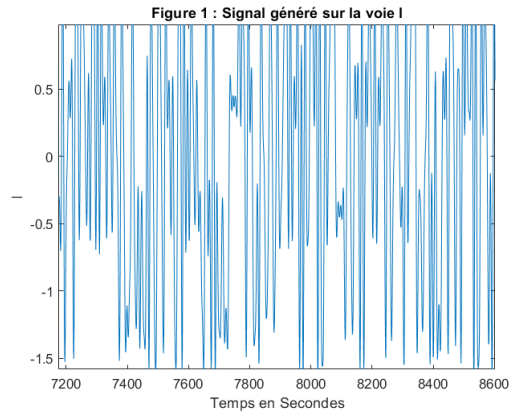
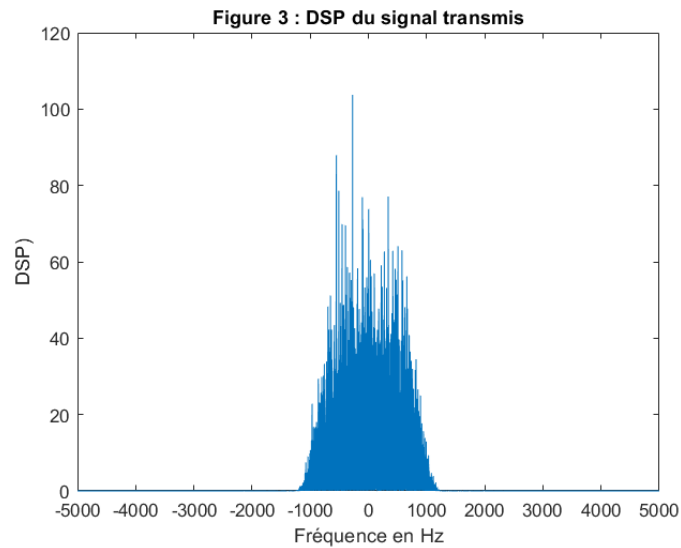
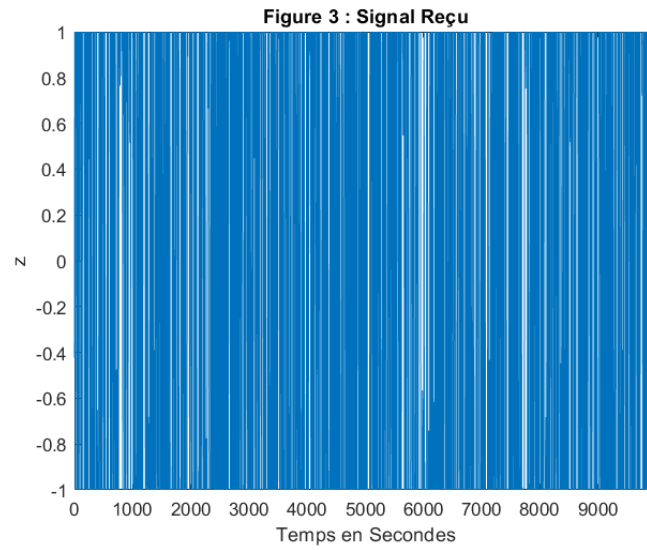


Figure 2.9: Signal tracé dans la voie I et en quadrature dans la voie Q

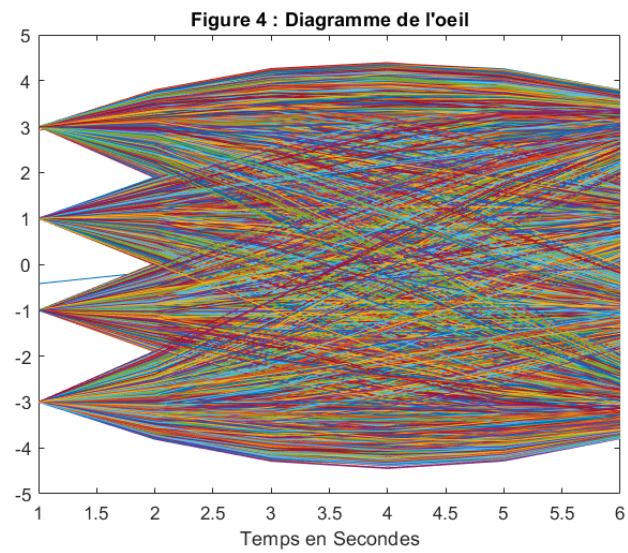
### 3. Densité spectrale de puissance par périodogramme



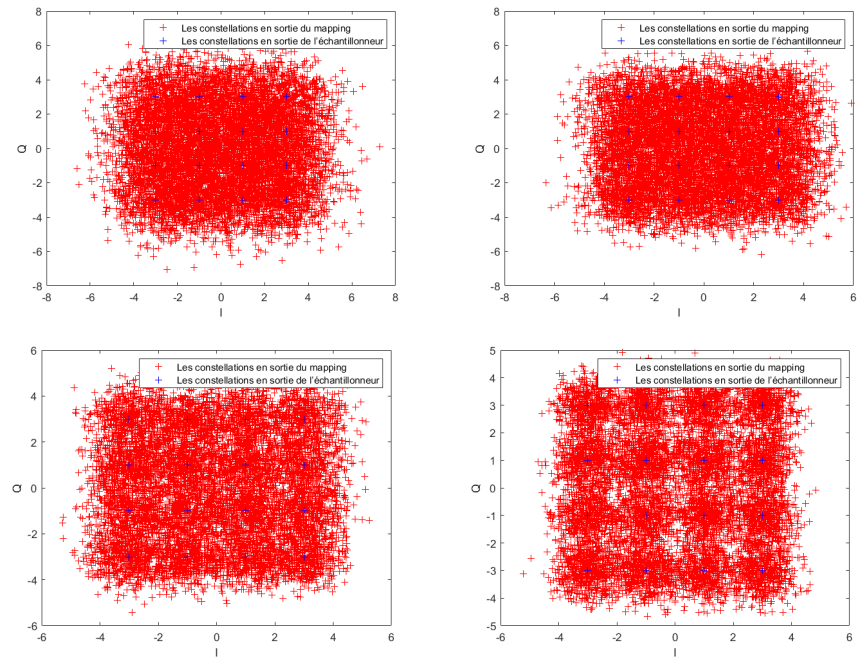
### 4. Signal reçu à la sortie du filtre de réception



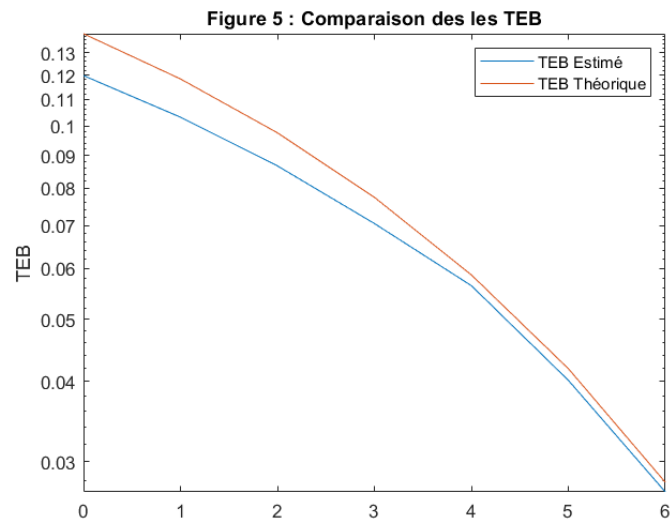
4. Diagramme de l'oeil du signal à la sortie du filtre de réception



5. Constellations à la sortie du Mapping et de l'échantillonneur



## 6. Comparaison des TEB théorique et expérimental



## Code Source

### 3.1 Chaîne Fréquence sur Porteuse

```
1 % ENSEEIHT - 1SN - Télécommunications
2 % Etudes de chaines de transmission sur Fréquence Porteuse
3 % Auteurs : Younes SAOUDI & Issam HABIBI
4
5 clear;
6 close all;
7 clc;
8
9 % Les données
10 bits_nbr = 10000;
11 Fe = 10000;
12 Rs = 1000;
13 Ns = Fe / Rs;
14 Te = 1 / Fe;
15 N = 50;
16
17 fp = 2000;
18 fc = 1500;
19 alpha = 0.35;
20 span = 8;
21
22 bits = randi([0, 1], 1, bits_nbr); %Génération de l'information binaire
23
24
25 symboles_ak = 2 * bits(1 : 2 : end) - 1;
26 symboles_bk = 2 * bits(2 : 2 : end) - 1;
27 symboles_dk = symboles_ak + 1j * symboles_bk; %Mapping +/-1 +/-j
28 Diracs1 = kron(symboles_ak, [1 zeros(1, Ns - 1)]); %Génération de la suite de Diracs
    pondérés par les symboles
29 Diracs2 = kron(symboles_bk, [1 zeros(1, Ns - 1)]);
30 h = rcosdesign(alpha, 8, Ns, 'sqrt'); %Réponse impulsionnelle du filtre de mise en forme
31 retard = (span * Ns) / 2;
32
33 I = filter(h, 1, [Diracs1 zeros(1, retard)]);
34 Q = filter(h, 1, [Diracs2 zeros(1, retard)]);
35 I = I(retard + 1 : end);
36 Q = Q(retard + 1 : end);
37
38
39 T = [0 : length(I) - 1] * Te;
40 x = I .* cos(2 * pi * fp * T) - Q .* sin(2 * pi * fp * T);
41
42 % Affichage du signal générée sur la voie I
43 figure ;
44 plot(I);
45 axis([0 length(I) - 1 -1 1]);
46 title('Figure 1 :Signal généré sur la voie I');
```

```

47 xlabel('Temps en Secondes');
48 ylabel('I');
49
50 % Affichage du signal généré sur la voie Q
51 figure ;
52 plot(Q);
53 axis([0 length(Q) - 1 -1 1]);
54 title('Figure 2 : Signal généré sur la voie Q');
55 xlabel('Temps en Secondes');
56 ylabel('Q');
57
58 % Affichage du signal
59 figure ;
60 plot(x);
61 axis([0 bits_nbr - 1 -1 1]);
62 title('Figure 3 : Signal Emis Sans Bruit');
63 xlabel('Temps en Secondes');
64 ylabel('Signal Emis');
65
66
67 % Calcul de la DSP
68 DSP_x = (1 / length(x)) * abs(fft(x, 2 ^ nextpow2(length(x)))) .^ 2;
69 figure;
70 plot(linspace(-Fe / 2, Fe / 2, length(DSP_x)), fftshift(DSP_x));
71 title('Figure 4 : DSP du signal émis');
72 xlabel('Fréquence en Hz');
73 ylabel('DSP');
74
75
76 % Retour en bande de base
77 signal_voieI = x .* cos(2 * pi * fp * T);
78 signal_voieQ = -j * x .* sin(2 * pi * fp * T);
79 filtre_passe_bas = 2 * (fc / Fe) * sinc(2 * (fc / Fe) * [-N : N]);
80 signal_conv = conv(signal_voieI + signal_voieQ, filtre_passe_bas, 'same');
81
82 % Filtrage de réception
83 h_r = h;
84 z = filter(h_r, 1, [signal_conv zeros(1,retard)]);
85 z = z(retard + 1 : end);
86 figure ;
87 plot(real(z));
88 axis([0 bits_nbr - 1 -1.5 1.5]);
89 title('Figure 5 : Signal Regu');
90 xlabel('Temps en Secondes');
91 ylabel('z');
92
93 % Diagramme de l'oeil
94 oeil = reshape(real(z), Ns, length(real(z)) / Ns);
95 figure;
96 plot(oeil);
97 title("Figure 6 : Diagramme de l'oeil");
98 xlabel('Temps en Secondes');
99
100 % Ajout du bruit
101 TES = zeros(1,7);
102 TEB = zeros(1,7);
103
104 for i = 0 : 6
105     N = randn(1, length(x));
106     Puissance_signal = mean(abs(x) .^ 2);
107     Puissance_bruit = Puissance_signal * Ns / (2 * log2(4) * 10 .^ (i / 10));
108     bruit_gaussien = sqrt(Puissance_bruit)*N;
109     y = x + bruit_gaussien;
110     signal_phaseI = y .* cos(2 * pi * fp * T);
111     signal_phaseQ = -1i * y .* sin(2 * pi * fp * T);

```

```

112 y = signal_phaseI + signal_phaseQ;
113
114 % Filtrage de réception
115 z = filter(h_r, 1, [y zeros(1,retard)]);
116 z = z(retard + 1 : end);
117 z_echantillonne = z(1 : Ns : end);
118
119 for j = 1 : length(z_echantillonne)
120     if (real(z_echantillonne(j)) <= 0 && imag(z_echantillonne(j)) <= 0)
121         symboles_decides(j) = -1 - 1i;
122
123     elseif (real(z_echantillonne(j)) >= 0 && imag(z_echantillonne(j)) >= 0)
124         symboles_decides(j) = 1 + 1i;
125
126     elseif (real(z_echantillonne(j)) <= 0 && imag(z_echantillonne(j)) >= 0)
127         symboles_decides(j) = -1 + 1i;
128
129     elseif (real(z_echantillonne(j)) >= 0 && imag(z_echantillonne(j)) <= 0)
130         symboles_decides(j) = 1 - 1i;
131     end
132 end
133
134 % TES et TEB
135 TES(i + 1) = length(find(symboles_decides ~= symboles_dk)) / (length(symboles_dk));
136 TEB(i + 1) = TES(i + 1) / 2;
137 end
138
139 %TEB Théorique et Estimé
140 figure;
141 semilogy([0 : 6], TEB);
142 hold on
143 semilogy([0 : 6], (4 * (1 - (1 / sqrt(4)))) * qfunc(sqrt(3 * log2(4)* 10 .^ ([0 : 6] / 10) /
144     (3)))) / log2(4));
144 title('Figure 7 : Comparaison des les TEB');
145 legend('TEB Estimé','TEB Théorique')
146 ylabel('TEB');

```

## 3.2 Chaîne Passe-Bas Equivalente

```
1 % ENSEEIHT - ISN - Télécommunications
2 % Etudes de chaînes de transmission sur Fréquence Porteuse
3 % Auteurs : Younes SAOUDI & Issam HABIBI
4
5 clear;
6 close all;
7 clc;
8
9
10 % Les données
11 bits_nbr = 10000;
12 Fe = 10000;
13 Rs = 1000;
14 Ns = Fe / Rs;
15 Te = 1 / Fe;
16 N = 50;
17
18 fp = 2000;
19 fc = 1500;
20 alpha = 0.35;
21 span = 8;
22
23
24 bits = randi([0, 1], 1, bits_nbr); %Génération de l'information binaire
25
26
27 symboles_ak = 2 * bits(1 : 2 : end) - 1;
28 symboles_bk = 2 * bits(2 : 2 : end) - 1;
29 symboles_dk = symboles_ak + 1j * symboles_bk; %Mapping +/-1 +/-j
30 Diracs1 = kron(symboles_ak, [1 zeros(1, Ns - 1)]); %Génération de la suite de Diracs
    pondérés par les symboles
31 Diracs2 = kron(symboles_bk, [1 zeros(1, Ns - 1)]);
32 h = rcosdesign(alpha, 8, Ns, 'sqrt'); %Réponse impulsionnelle du filtre de mise en forme
33 retard = (span * Ns) / 2;
34
35
36 I = filter(h, 1, [Diracs1 zeros(1, retard)]);
37 Q = filter(h, 1, [Diracs2 zeros(1, retard)]);
38 I = I(retard + 1 : end);
39 Q = Q(retard + 1 : end);
40
41 T = [0 : length(I) - 1] * Te;
42 x_enveloppe = I + 1i * Q;
43
44 % Affichage du signal générée sur la voie I
45 figure ;
46 plot(I);
47 axis([0 length(I) - 1 -1 1]);
48 title('Figure 1: Signal généré sur la voie I');
49 xlabel('Temps en Secondes');
50 ylabel('I');
51
52 % Affichage du signal généré sur la voie Q
53 figure ;
54 plot(Q);
55 axis([0 length(Q) - 1 -1 1]);
56 title('Figure 2 : Signal généré sur la voie Q');
57 xlabel('Temps en Secondes');
58 ylabel('Q');
59
60 % Calcul de la DSP de x_enveloppe
61 DSP_x_enveloppe = (1 / length(x_enveloppe)) * abs(fft(x_enveloppe, 2 ^ nextpow2(length(
```



```

        x_enveloppe))))).^2;
62 figure;
63 plot(linspace(-Fe / 2, Fe / 2, length(DSP_x_enveloppe)), fftshift(DSP_x_enveloppe));
64 title('Figure 3: DSP de l'Enveloppe');
65 xlabel('Fréquence en Hz');
66 ylabel('DSP');
67
68
69 % Filtrage de réception
70 h_r = h;
71 z = filter(h_r, 1, [x_enveloppe zeros(1, retard)]);
72 z = z(retard + 1 : end);
73 figure;
74 plot(real(z));
75 axis([0 bits_nbr - 1 -1.5 1.5]);
76 title('Figure 3 : Signal Reçu');
77 xlabel('Temps en Secondes');
78 ylabel('z');
79
80 % Diagramme de l'oeil
81 oeil = reshape(real(z), Ns, length(real(z)) / Ns);
82 figure;
83 plot(oeil);
84 title("Figure 4 : Diagramme de l'oeil");
85 xlabel('Temps en Secondes');
86
87
88 %Ajout du bruit
89 TES = zeros(1,7);
90 TEB = zeros(1,7);
91
92 for i = 0 : 6
93     Puissance_signal = mean(abs(x_enveloppe).^2);
94     Puissance_bruit = Puissance_signal * Ns / (2 * log2(4) * 10 .^(i / 10));
95     bruit_gaussien = (sqrt(Puissance_bruit) * randn(1, length(x_enveloppe))) + 1i * (sqrt(
96     Puissance_bruit) * randn(1, length(x_enveloppe)));
97     y = x_enveloppe + bruit_gaussien;
98
99     z = filter(h_r, 1, [y zeros(1,retard)]);% Filtrage de réception
100    z = z(retard + 1 : end);
101    z_echantillonne = z(1 : Ns : end);    % Echantillonnage du signal
102
103    figure;
104    plot(real(z_echantillonne), imag(z_echantillonne), 'r+');
105    hold on;
106    plot(symboles_ak, symboles_bk, 'b+');
107    legend('Les constellations en sortie du mapping','Les constellations en sortie de
108    l'échantillonneur')
109    xlabel('I');
110    ylabel('Q');
111
112    for j = 1 : length(z_echantillonne)
113        if (real(z_echantillonne(j)) <= 0 && imag(z_echantillonne(j)) <= 0)
114            symboles_decides(j) = -1 - 1i;
115        elseif (real(z_echantillonne(j)) >= 0 && imag(z_echantillonne(j)) >= 0)
116            symboles_decides(j) = 1 + 1i;
117        elseif (real(z_echantillonne(j)) <= 0 && imag(z_echantillonne(j)) >= 0)
118            symboles_decides(j) = -1 + 1i;
119        elseif (real(z_echantillonne(j)) >= 0 && imag(z_echantillonne(j)) <= 0)
120            symboles_decides(j) = 1 - 1i;
121        end
122    end
123
124    % TES et TEB
125    TES(i + 1) = length(find(symboles_decides ~= symboles_dk)) / (length(symboles_dk));

```

```

124     TEB(i + 1) = TES(i + 1) / 2;
125 end
126
127 %TEB Théorique et Estimé
128 figure;
129 semilogy([0 : 6], TEB);
130 hold on
131 semilogy([0 : 6], (4 * (1 - (1 / sqrt(4)))) * qfunc(sqrt(3 * log2(4)* 10 .^ ([0 : 6] / 10) /
    (3)))) / log2(4));
132 grid
133 title('Figure 6 : Comparaison des les TEB');
134 legend('TEB Estimé', 'TEB Théorique')
135 ylabel('TEB');

```

### 3.3 Chaîne 4-ASK

```
1 % ENSEEIHT - ISN - Télécommunications
2 % Etudes de chaînes de transmission sur Fréquence Porteuse
3 % Auteurs : Younes SAOUDI & Issam HABIBI
4
5 clear;
6 close all;
7 clc;
8
9 % Les données
10 bits_nbr = 10000;
11 Fe = 10000;
12 Rs = 2000;
13 Ns = floor(Fe / Rs);
14 Te = 1 / Fe;
15 N = 50;
16
17 fp = 2000;
18 fc = 1500;
19 alpha = 0.5;
20 span = 8;
21
22
23 bits = randi([0, 1], 1, bits_nbr); % Génération de linformation binaire
24
25
26 Symboles = (2 * bi2de(reshape(bits, 2, length(bits)/2).') - 3).'; % Mapping de Gray
27 Diracs = kron(Symboles, [1 zeros(1, Ns - 1)]); % Génération de la suite de Diracs pondérés par
    les symboles
28 h = rcosdesign(alpha, 8, Ns, 'sqrt'); % Réponse impulsionnelle du filtre de mise en forme
29 retard = (span * Ns) / 2;
30
31 x = filter(h, 1, [Diracs zeros(1, retard)]); % Filtrage de MEF
32 x = x(retard + 1 : end);
33
34 % Affichage du signal transmis après MEF
35 figure;
36 plot(x);
37 axis([0 length(x) - 1 -3 3]);
38 title('Figure 1 : Signal transmis après MEF');
39 xlabel('Temps en Secondes');
40
41 % Calcul de la DSP
42 DSP_x = (1 / length(x)) * abs(fft(x, 2 ^ nextpow2(length(x)))) .^ 2;
43 figure;
44 plot(linspace(-Fe / 2, Fe / 2, length(DSP_x)), fftshift(DSP_x));
45 title('Figure 2 : DSP du signal transmis');
46 xlabel('Fréquence en Hz');
47 ylabel('DSP');
48
49 % Filtrage de réception
50 h_r = h;
51 z = filter(h_r, 1, [x zeros(1, retard)]);
52 z = z(retard + 1 : end);
53 figure;
54 plot(z);
55 axis([0 bits_nbr - 1 -3 3]);
56 title('Figure 3 : Signal Reçu');
57 xlabel('Temps en Secondes');
58 ylabel('z');
59
60 % Diagramme de l'oeil
61 oeil = reshape(z, Ns, length(z) / Ns);
```

```

62 figure;
63 plot(oeil);
64 title("Figure 4 : Diagramme de l'oeil");
65 xlabel('Temps en Secondes');
66
67
68 %Ajout du bruit
69 TES = zeros(1,7);
70 TEB = zeros(1,7);
71
72 for i = 0 : 6
73     Puissance_signal = mean(abs(x) .^ 2);
74     Puissance_bruit = Puissance_signal * Ns / (2 * log2(4) * 10 .^ (i / 10));
75     bruit_gaussien = (sqrt(Puissance_bruit) * randn(1, length(x))) + 1i * (sqrt(
    Puissance_bruit) * randn(1, length(x)));
76     y = x + bruit_gaussien;
77
78
79     z = filter(h_r, 1, [y zeros(1,retard)]); % Filtrage de réception
80     z = z(retard + 1 : end);
81     z_echantillonne = z(1 : Ns : end); % Echantillonnage du signal
82
83     symboles_decides = zeros(1, length(z_echantillonne));
84
85     for j = 1 : length(z_echantillonne)
86         if (real(z_echantillonne(j)) > 2)
87             symboles_decides(j) = 3;
88         elseif (real(z_echantillonne(j)) >= 0)
89             symboles_decides(j) = 1;
90         elseif (real(z_echantillonne(j)) < -2)
91             symboles_decides(j) = -3;
92         else
93             symboles_decides(j) = -1;
94         end
95     end
96
97     % TES et TEB
98     TES(i + 1) = length(find(Symboles ~= symboles_decides)) / length(Symboles);
99     TEB(i + 1) = TES(i + 1) / 2;
100 end
101
102 figure;
103 plot(real(z_echantillonne), 0, 'r+');
104 xlim([-4 4])
105 hold on;
106 plot(Symboles, 0, 'b+');
107 legend('Les constellations en sortie du mapping','Les constellations en sortie de
    l'échantillonneur')
108 xlabel('I');
109 ylabel('Q');
110
111
112 % TEB Théorique et Estimé
113 figure;
114 semilogy([0 : 6], TEB);
115 hold on
116 semilogy([0 : 6], (3 / 4) * qfunc(sqrt((4 / 5) * 10 .^ ([0 : 6] / 10))));
117 title('Comparaison des les TEB');
118 legend('TEB Estimé','TEB Théorique')
119 ylabel('TEB');

```

### 3.4 Chaîne Q-PSK

```
1 % ENSEEIHT - 1SN - Télécommunications
2 % Etudes de chaines de transmission sur Fréquence Porteuse
3 % Auteurs : Younes SAOUDI & Issam HABIBI
4
5 clear;
6 close all;
7 clc;
8
9 % Les données
10 bits_nbr = 10000;
11 Fe = 10000;
12 Rs = 3000;
13 Ns = floor(Fe / Rs);
14 Te = 1 / Fe;
15 N = 50;
16
17 fp = 2000;
18 fc = 1500;
19 alpha = 0.5;
20 span = 8;
21
22
23 bits = randi([0, 1], 1, bits_nbr); %Génération de l'information binaire
24
25
26 symboles_ak = 2 * bits(1 : 2 : end) - 1;
27 symboles_bk = 2 * bits(2 : 2 : end) - 1;
28 symboles_dk = symboles_ak + 1j * symboles_bk; %Mapping +/-1 +/-j
29 Diracs1 = kron(symboles_ak, [1 zeros(1, Ns - 1)]); %Génération de la suite de Diracs
   pondérés par les symboles
30 Diracs2 = kron(symboles_bk, [1 zeros(1, Ns - 1)]);
31 h = rcosdesign(alpha, 8, Ns, 'sqrt'); %Réponse impulsionnelle du filtre de mise en forme
32 retard = (span * Ns) / 2;
33
34 I = filter(h, 1, [Diracs1 zeros(1, retard)]);
35 Q = filter(h, 1, [Diracs2 zeros(1, retard)]);
36 I = I(retard + 1 : end);
37 Q = Q(retard + 1 : end);
38
39 x = I + 1i * Q;
40
41 % Affichage du signal générée sur la voie I
42 figure ;
43 plot(I);
44 axis([0 length(I) - 1 -1 1]);
45 title('Figure 1 : Signal généré sur la voie I');
46 xlabel('Temps en Secondes');
47 ylabel('I');
48
49 % Affichage du signal généré sur la voie Q
50 figure ;
51 plot(Q);
52 axis([0 length(Q) - 1 -1 1]);
53 title('Figure 2 : Signal généré sur la voie Q');
54 xlabel('Temps en Secondes');
55 ylabel('Q');
56
57 % Calcul de la DSP
58 DSP_x = (1 / length(x)) * abs(fft(x, 2 ^ nextpow2(length(x)))) .^ 2;
59 figure;
60 plot(linspace(-Fe / 2, Fe / 2, length(DSP_x)), fftshift(DSP_x));
61 title('Figure 3 : DSP du signal émis');
```

```

62 xlabel('Fréquence en Hz');
63 ylabel('DSP');
64
65 % Filtrage de réception
66 h_r = h;
67 z = filter(h_r, 1, [x zeros(1,retard)]);
68 z = z(retard + 1 : end);
69 figure ;
70 plot(real(z));
71 axis([0 bits_nbr - 1 -1.5 1.5]);
72 title('Figure 3 : Signal Reçu');
73 xlabel('Temps en Secondes');
74 ylabel('z');
75
76 % Diagramme de l'oeil
77 oeil = reshape(real(z), Ns, length(real(z)) / Ns);
78 figure;
79 plot(oeil);
80 title("Figure 4 : Diagramme de l'oeil");
81 xlabel('Temps en Secondes');
82
83
84 %Ajout du bruit
85 TES = zeros(1,7);
86 TEB = zeros(1,7);
87
88 for i = 0 : 6
89     Puissance_signal = mean(abs(x) .^ 2);
90     Puissance_bruit = Puissance_signal * Ns / (2 * log2(4) * 10 .^ (i / 10));
91     bruit_gaussien = (sqrt(Puissance_bruit) * randn(1, length(x))) + 1i * (sqrt(
92     Puissance_bruit) * randn(1, length(x)));
93     y = x + bruit_gaussien;
94
95     % Filtrage de réception
96     z = filter(h_r, 1, [y zeros(1,retard)]);
97     z = z(retard + 1 : end);
98     z_echantillonne = z(1 : Ns : end);
99
100    % Les constellations en sortie du mapping et de léchantillonneur
101    figure;
102    plot(real(z_echantillonne), imag(z_echantillonne), 'r+');
103    hold on;
104    plot(symboles_ak, symboles_bk, 'b+');
105    legend('Les constellations en sortie du mapping','Les constellations en sortie de
106    léchantillonneur')
107    xlabel('I');
108    ylabel('Q');
109
110    for j = 1 : length(z_echantillonne)
111        if (real(z_echantillonne(j)) <= 0 && imag(z_echantillonne(j)) <= 0)
112            symboles_decides(j) = -1 - 1i;
113
114        elseif (real(z_echantillonne(j)) >= 0 && imag(z_echantillonne(j)) >= 0)
115            symboles_decides(j) = 1 + 1i;
116
117        elseif (real(z_echantillonne(j)) <= 0 && imag(z_echantillonne(j)) >= 0)
118            symboles_decides(j) = -1 + 1i;
119
120        elseif (real(z_echantillonne(j)) >= 0 && imag(z_echantillonne(j)) <= 0)
121            symboles_decides(j) = 1 - 1i;
122        end
123    end
124
125    % TES et TEB
126    TES(i + 1) = length(find(symboles_decides ~= symboles_dk)) / (length(symboles_dk));

```

```

125     TEB(i + 1) = TES(i + 1) / 2;
126 end
127
128 %TEB Théorique et Estimé
129 figure;
130 semilogy([0 : 6], TEB);
131 hold on
132 semilogy([0 : 6], (4 * (1 - (1 / sqrt(4)))) * qfunc(sqrt(3 * log2(4)* 10 .^ ([0 : 6] / 10) /
    (3)))) / log2(4));
133 title('Figure 6 : Comparaison des les TEB');
134 legend('TEB Estimé','TEB Théorique')
135 ylabel('TEB');

```

### 3.5 Chaîne 8-PSK

```
1 % ENSEEIHT - 1SN - Télécommunications
2 % Etudes de chaînes de transmission sur Fréquence Porteuse
3 % Auteurs : Younes SAOUDI & Issam HABIBI
4
5 clear;
6 close all;
7 clc;
8
9 % Les données
10 bits_nbr = 10000;
11 Fe = 10000;
12 Rs = 2000;
13 Ns = floor(Fe / Rs);
14 Te = 1 / Fe;
15 M = 8;
16 fp = 2000;
17 fc = 1500;
18 alpha = 0.5;
19 span = 8;
20
21 bits = randi([0, M - 1], 1, bits_nbr); %Génération de l'information binaire
22
23 symboles_dk = pskmod(bits, M, pi / M); % Mapping
24 symboles_ak = real(symboles_dk);
25 symboles_bk = imag(symboles_dk);
26 Diracs1 = kron(symboles_ak, [1 zeros(1, Ns - 1)]); %Génération de la suite de Diracs
    pondérés par les symboles
27 Diracs2 = kron(symboles_bk, [1 zeros(1, Ns - 1)]);
28 h = rcosdesign(alpha, 8, Ns, 'sqrt'); %Réponse impulsionnelle du filtre de mise en forme
29 retard = (span * Ns) / 2;
30
31 I = filter(h, 1, [Diracs1 zeros(1, retard)]);
32 Q = filter(h, 1, [Diracs2 zeros(1, retard)]);
33 I = I(retard + 1 : end);
34 Q = Q(retard + 1 : end);
35
36 x = I + 1i * Q;
37
38 % Affichage du signal générée sur la voie I
39 figure ;
40 plot(I);
41 axis([0 length(I) - 1 -1 1]);
42 title('Figure 1 : Signal généré sur la voie I');
43 xlabel('Temps en Secondes');
44 ylabel('I');
45
46 % Affichage du signal généré sur la voie Q
47 figure ;
48 plot(Q);
49 axis([0 length(Q) - 1 -1 1]);
50 title('Figure 2 : Signal généré sur la voie Q');
51 xlabel('Temps en Secondes');
52 ylabel('Q');
53
54
55 % Calcul de la DSP
56 DSP_x = (1 / length(x)) * abs(fft(x, 2 ^ nextpow2(length(x)))) .^ 2;
57 figure;
58 plot(linspace(-Fe / 2, Fe / 2, length(DSP_x)), fftshift(DSP_x));
59 title('Figure 3 : DSP du signal transmis');
60 xlabel('Fréquence en Hz');
61 ylabel('DSP');
```



```

62
63 %Filtrage de réception
64 h_r = h;
65 z = filter(h_r, 1, [x zeros(1,retard)]);
66 z = z(retard + 1 : end);
67 figure ;
68 plot(real(z));
69 axis([0 bits_nbr - 1 -1 1]);
70 title('Figure 3 : Signal Reçu');
71 xlabel('Temps en Secondes');
72 ylabel('z');
73
74 % Diagramme de l'oeil
75 oeil = reshape(real(z), Ns, length(real(z)) / Ns);
76 figure;
77 plot(oeil);
78 title("Figure 4 : Diagramme de l'oeil");
79 xlabel('Temps en Secondes');
80
81 %Ajout du bruit
82 TES = zeros(1,7);
83 TEB = zeros(1,7);
84
85 for i = 0 : 6
86     Puissance_signal = mean(abs(x) .^ 2);
87     Puissance_bruit = Puissance_signal * Ns / (2 * log2(M) * 10 .^ (i / 10));
88     bruit_gaussien = (sqrt(Puissance_bruit) * randn(1, length(x))) + 1i * (sqrt(
89     Puissance_bruit) * randn(1, length(x)));
90     y = x + bruit_gaussien;
91
92     z = filter(h_r, 1, [y zeros(1,retard)]); % Filtrage de réception
93     z = z(retard + 1 : end);
94     z_echantillonne = z(1 : Ns : end); % Echantillonnage du signal
95
96
97     figure();
98     plot(real(z_echantillonne), imag(z_echantillonne), 'r+');
99     hold on;
100    plot(symboles_ak, symboles_bk, 'b+');
101    legend('Les constellations en sortie du mapping','Les constellations en sortie de
102    l'échantillonneur')
103    xlabel('I');
104    ylabel('Q');
105
106    bits_decides = pskdemod(z_echantillonne, M, pi / M);
107
108    %TES et TEB
109    TES(i + 1) = length(find(bits_decides ~= bits)) / length(bits);
110    TEB(i + 1) = TES(i + 1) / log2(M);
111 end
112
113 % TEB Théorique et Estimé
114 figure;
115 semilogy([0 : 6], TEB);
116 hold on
117 semilogy([0 : 6], (2 / log2(M)) * qfunc(sqrt(2 * log2(M) * 10 .^ ([0 : 6] / 10)) * sin(pi /
118 M)));
119 title('Figure 8 : Comparaison des les TEB');
120 legend('TEB Estimé','TEB Théorique')
121 ylabel('TEB');

```

## 3.6 Chaîne 16-QAM

```
1 % ENSEEIHT - ISN - Télécommunications
2 % Etudes de chaînes de transmission sur Fréquence Porteuse
3 % Auteurs : Younes SAOUDI & Issam HABIBI
4
5 clear;
6 close all;
7 clc;
8
9
10 % Les données
11 bits_nbr = 10000;
12 Fe = 10000;
13 Rb = 48000;
14 M = 16;
15 Rs = Rb / (log2(M) * 8);
16 Ns = floor(Fe / Rs);
17 Te = 1 / Fe;
18 fp = 2000;
19 fc = 1500;
20 alpha = 0.5;
21 span = 8;
22
23
24
25 bits = randi([0, M - 1], 1, bits_nbr); %Génération de l'information binaire
26
27 symboles_dk = qammod(bits, M); % Mapping
28 symboles_ak = real(symboles_dk);
29 symboles_bk = imag(symboles_dk);
30 Diracs1 = kron(symboles_ak, [1 zeros(1, Ns - 1)]); %Génération de la suite de Diracs
    pondérés par les symboles
31 Diracs2 = kron(symboles_bk, [1 zeros(1, Ns - 1)]);
32 h = rcosdesign(alpha, 8, Ns, 'sqrt'); %Réponse impulsionnelle du filtre de mise en forme
33 retard = (span * Ns) / 2;
34
35 I = filter(h, 1, [Diracs1 zeros(1, retard)]);
36 Q = filter(h, 1, [Diracs2 zeros(1, retard)]);
37 I = I(retard + 1 : end);
38 Q = Q(retard + 1 : end);
39
40 x = I + 1i * Q;
41
42
43 % Affichage du signal générée sur la voie I
44 figure ;
45 plot(I);
46 axis([0 length(I) - 1 -1 1]);
47 title('Figure 1 : Signal généré sur la voie I');
48 xlabel('Temps en Secondes');
49 ylabel('I');
50
51 % Affichage du signal généré sur la voie Q
52 figure ;
53 plot(Q);
54 axis([0 length(Q) - 1 -1 1]);
55 title('Figure 2 : Signal généré sur la voie Q');
56 xlabel('Temps en Secondes');
57 ylabel('Q');
58
59
60 % Calcul de la DSP
61 DSP_x = (1 / length(x)) * abs(fft(x, 2 ^ nextpow2(length(x)))) .^ 2;
```

```

62 figure;
63 plot(linspace(-Fe / 2, Fe / 2, length(DSP_x)), fftshift(DSP_x));
64 title('Figure 3 : DSP du signal transmis');
65 xlabel('Fréquence en Hz');
66 ylabel('DSP');
67
68 % Filtrage de réception
69 h_r = h;
70 z = filter(h_r, 1, [x zeros(1,retard)]);
71 z = z(retard + 1 : end);
72 figure ;
73 plot(real(z));
74 axis([0 bits_nbr - 1 -1 1]);
75 title('Figure 3 : Signal Reçu');
76 xlabel('Temps en Secondes');
77 ylabel('z');
78
79 % Diagramme de l'oeil
80 oeil = reshape(real(z), Ns, length(real(z)) / Ns);
81 figure;
82 plot(oeil);
83 title("Figure 4 : Diagramme de l'oeil");
84 xlabel('Temps en Secondes');
85
86 %Ajout du bruit
87 TES = zeros(1,7);
88 TEB = zeros(1,7);
89
90 for i = 0 : 6
91     Puissance_signal = mean(abs(x) .^ 2);
92     Puissance_bruit = Puissance_signal * Ns / (2 * log2(M) * 10 .^ (i / 10));
93     bruit_gaussien = (sqrt(Puissance_bruit) * randn(1, length(x))) + 1i * (sqrt(
94     Puissance_bruit) * randn(1, length(x)));
95     y = x + bruit_gaussien;
96
97     % Filtrage de réception
98     z = filter(h_r, 1, [y zeros(1,retard)]);
99     z = z(retard + 1 : end);
100     z_echantillonne = z(1 : Ns : end);
101
102     % Les constellations en sortie du mapping et de l'échantillonneur
103     figure;
104     plot(real(z_echantillonne), imag(z_echantillonne), 'r');
105     hold on;
106     plot(symboles_ak, symboles_bk, 'b+');
107     legend('Les constellations en sortie du mapping','Les constellations en sortie de
108     l'échantillonneur')
109     xlabel('I');
110     ylabel('Q');
111
112     bits_decides = qamdemod(z_echantillonne, M);
113
114     % TES et TEB
115     TES(i + 1) = length(find(bits_decides ~= bits)) / (length(bits));
116     TEB(i + 1) = TES(i + 1) / log2(M);
117 end
118 %TEB Théorique et Estimé
119 figure;
120 semilogy([0 : 6], TEB);
121 hold on
122 semilogy([0 : 6], (1 - (1 / sqrt(M))) * qfunc(sqrt((4 / 5) * 10 .^ ([0 : 6] / 10))));
123 title('Figure 5 : Comparaison des les TEB');
124 legend('TEB Estimé','TEB Théorique')

```

```
125 ylabel('TEB');
```