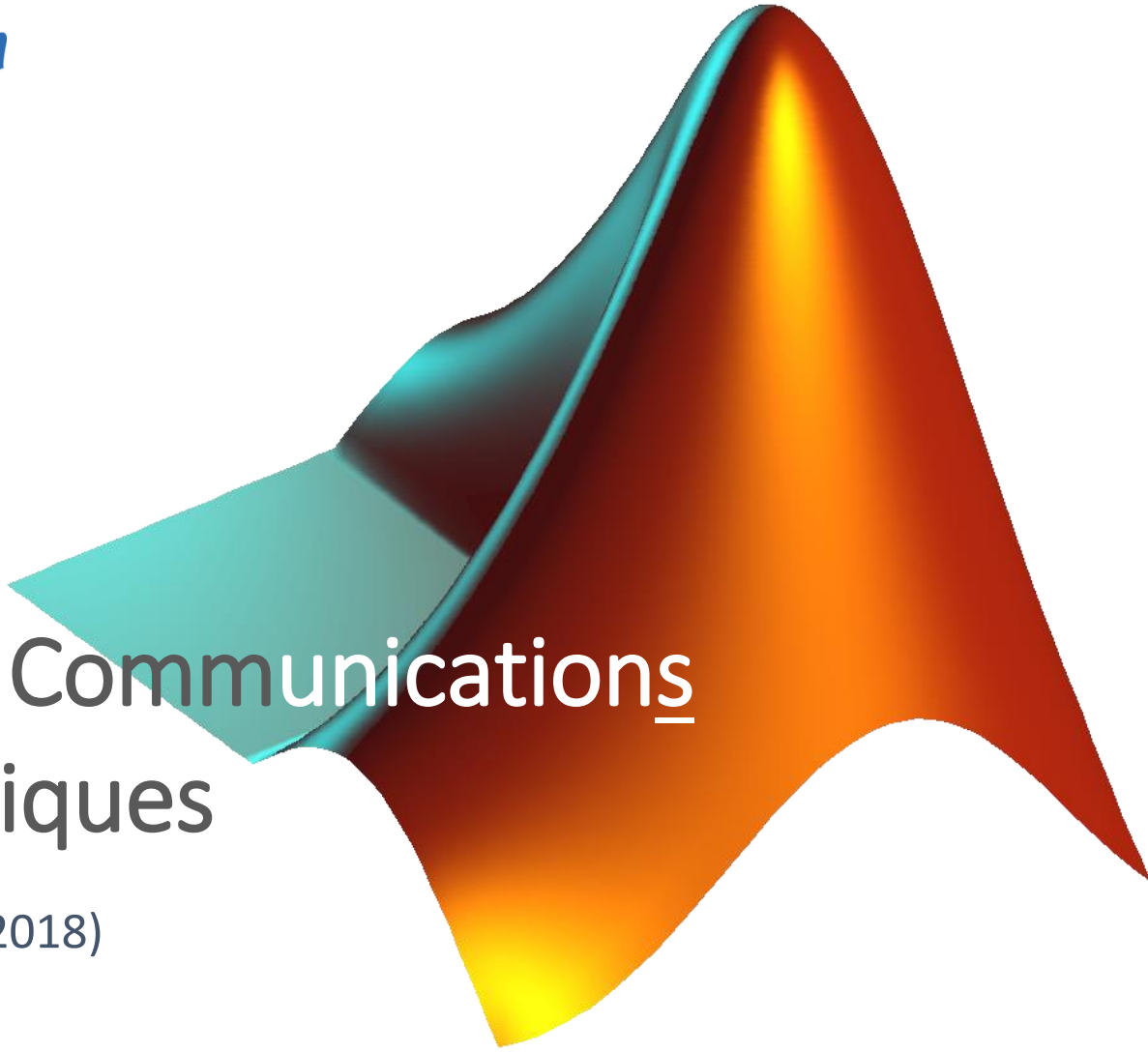


Atelier Communications Numériques

TP5 (16/04/2018)

Oueslati Mohamed melek

Master SmartCom

Année universitaire 2017-2018

Détails du TP

TP :

TP numéro 5 de la date : 16-04-2018

Objectifs du TP :

On va donner les constellations d'un signal avons et après le bruit on compare avec les résultats théoriques.
Puis on va appliquer quelque modification sur le code pour qu'il soit fonctionnelle M-qam

Logiciel utilisé :

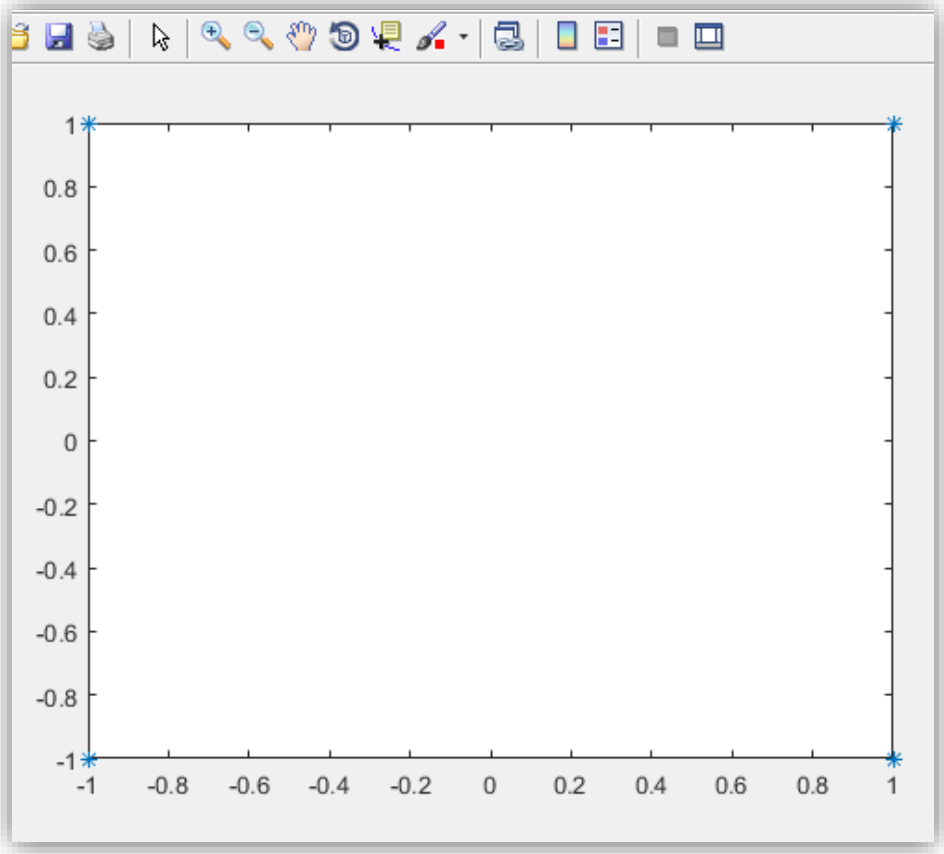
Matlab version 2014

Question 1

Il est demandé de donner la constellation du signal modulé TX

```
hMod = modem.qammod(M);
hMod.InputType = 'Bit';
hMod.SymbolOrder = 'Gray';
hDemod = modem.qamdemod(hMod);
x = randi([0 1],n,1);
tx = modulate(hMod,x);
plot(tx,'*');
```

La génération du signal modulé TX



	1
1	-1.0000 + 1.0000i
2	1.0000 + 1.0000i
3	1.0000 - 1.0000i
4	-1.0000 + 1.0000i
5	-1.0000 + 1.0000i
6	1.0000 + 1.0000i
7	1.0000 - 1.0000i
8	1.0000 + 1.0000i
9	-1.0000 - 1.0000i
10	-1.0000 - 1.0000i
11	-1.0000 - 1.0000i
12	1.0000 + 1.0000i
13	-1.0000 - 1.0000i
14	-1.0000 + 1.0000i
15	-1.0000 + 1.0000i
16	1.0000 + 1.0000i
17	-1.0000 - 1.0000i
18	1.0000 + 1.0000i
19	1.0000 + 1.0000i
20	1.0000 - 1.0000i
21	1.0000 + 1.0000i

La représentation graphique de la constellation du signal modulé TX

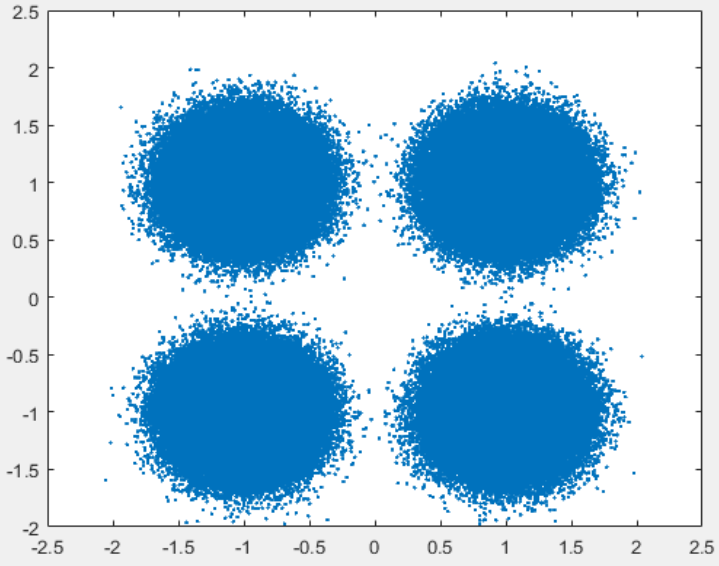
Question 2

Il est demandé de donner la constellation du signal modulé RX pour EbNo=2dB

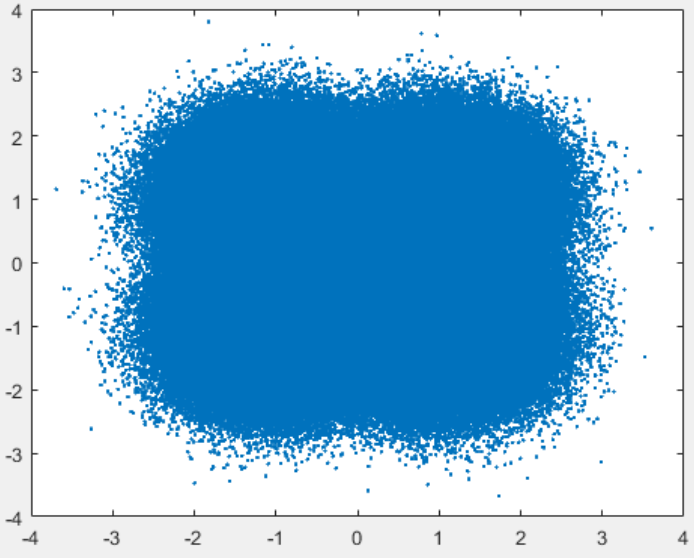
```
EbNo = 0:2; % en dB
SNR = EbNo + 10*log10(k);
rx = zeros(nSyms,length(SNR));
bit_error_rate = zeros(length(SNR),1);
for i=1:length(SNR)
rx(:,i) = awgn(tx,SNR(i),'measured');
end
plot(rx,'.');
```

La génération du signal modulé RX (bruité)

	1	2	3
1	-1.9084 + 0.7652i	-0.2577 + 1.0302i	-0.9327 + 0.7162
2	-0.5331 + 1.6433i	-1.5083 + 1.6751i	-0.6858 + 0.1688
3	1.7684 + 0.8110i	1.6601 + 1.6333i	1.3961 - 0.2549
4	-1.9996 - 0.7800i	-0.3874 - 0.4297i	-0.2411 - 1.6148
5	-0.3450 - 1.3623i	-0.8922 - 0.6710i	-0.8881 - 1.0681
6	-0.5768 + 2.0563i	-1.1135 - 0.1635i	-1.6301 - 0.0144
7	-2.2570 + 0.6733i	-1.4111 + 1.5845i	-1.7015 + 1.0076
8	-1.1861 + 1.3119i	-0.6434 + 2.0965i	-0.5452 - 0.2997
9	0.5339 - 1.6885i	0.6771 - 1.7157i	0.6481 - 1.3021
10	-0.9370 + 0.5207i	-1.3150 + 1.6123i	-1.4956 + 1.0826
11	1.1850 + 1.7437i	1.0852 + 2.0283i	1.1536 - 0.0292
12	-0.7923 - 0.0684i	-0.1283 - 0.8310i	-1.2358 - 0.3773
13	-0.0058 - 1.3597i	-0.4059 - 1.0554i	-0.7693 - 1.7545
14	-0.8986 + 0.3822i	-1.2112 + 1.0695i	-1.2741 + 1.6371
15	0.1234 + 1.2085i	0.3914 + 0.1985i	1.6160 + 0.8428



Constellation pour EbN0=10db



Constellation pour EbN0=2db

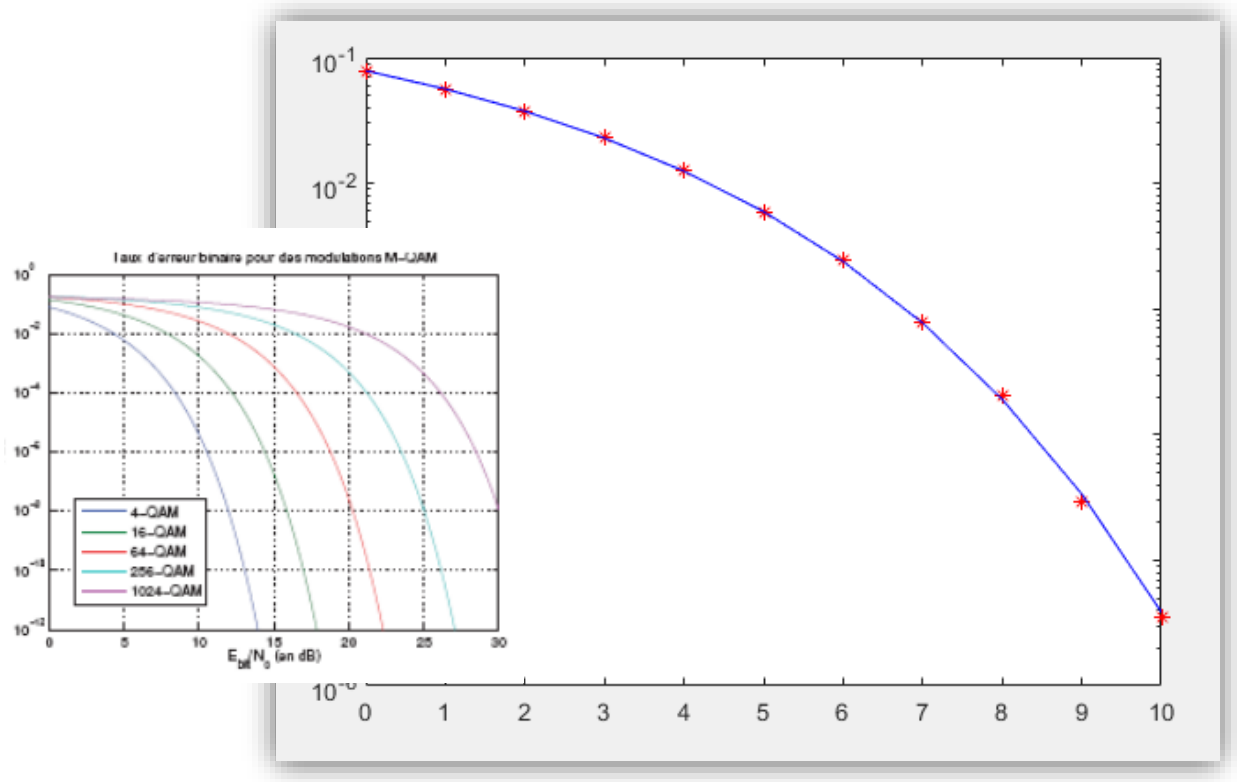
⇨ Tout on diminue le EbN0 le bruit augmente

Question 3

Il est demandé de comparer les performances de cette transmission avec les résultats théoriques.

```
BERth=(2*(sqrt(M)-1)/(sqrt(M)*log2(sqrt(M))))*qfunc(sqrt(2*(10.^(EbNo/10))*((3*log2(sqrt(M)))/(M-1))));
semilogy(EbNo,bit_error_rate,'r*');
hold on;
semilogy(EbNo,BERth,'b-');
```

Code pour la représentation graphique des performances



Comparaison des performances sur une même figure

Les résultats pratiques et théoriques sont similaires : Les deux courbes sont confondues, ont la même allure et la même origine pour M=4.

Question 4

Généraliser ce code pour les modulations M-QAM (Sous forme d'une fonction de Paramètre d'entrée M).

```
function M_QAM(M)
k = log2(M);
nSyms = 10^6;
n = nSyms*k;
hMod = modem.qammod(M);
hMod.InputType = 'Bit';
hMod.SymbolOrder = 'Gray';
hDemod = modem.qamdemod(hMod);
x = randi([0 1],n,1);
tx = modulate(hMod,x);
%plot(tx,'*');

EbNo = 0:2; % en dB
SNR = EbNo + 10*log10(k);
rx = zeros(nSyms,length(SNR));
bit_error_rate = zeros(length(SNR),1);

for i=1:length(SNR)
rx(:,i) = awgn(tx,SNR(i),'measured');
end
plot(rx,'.');

rx_demod = demodulate(hDemod,rx);
for i=1:length(SNR)
[~,bit_error_rate(i)] = biterr(x,rx_demod(:,i));
end
```

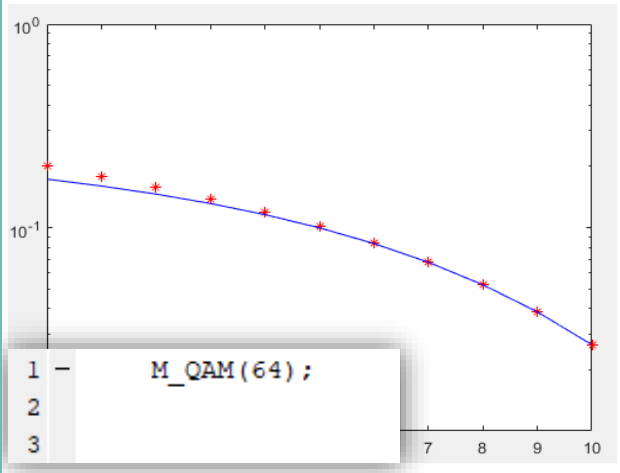
Fonction M_QAM

```
1 -     M_QAM(64);
2
3
```

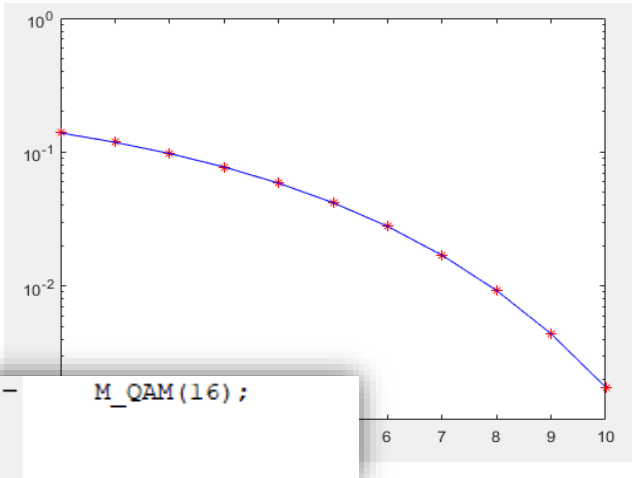
L'appel de la fonction M_QAM

Question 5

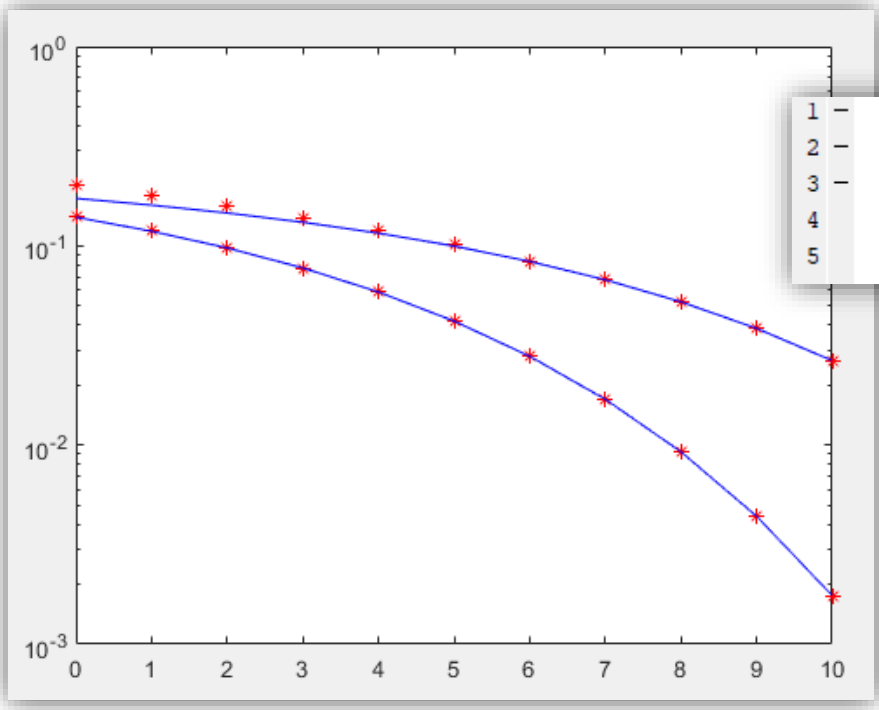
Valider le code en donnant su la même figure les performances simulées et Théoriques pour M=16 et 64.



```
1 -     M_QAM(64);
2
3
```



```
1 -     M_QAM(16);
2
3
```



```
1 -     M_QAM(16);
2 -     hold on ;
3 -     M_QAM(64);
4
5
```

=> Le taux d'erreur dépond de M, lorsque M augmente le taux augmente et inversement