

Département Génie des Communications et Réseaux

# Compte Rendu

## TP4

---

### *Chaine de transmission numériques*

### ***COMMUNICATION NUMERIQUES***

---

Elaborés par :

Wissem Bagga

Génie des communications et Réseaux GCRA2 - Grp1

*Date: 21/10/2024*

Année Universitaire: 2023/2024

### I. Introduction

Ce TP a permis d'étudier et de comparer les performances des différents types de codage numérique (NRZ, RZ, et Manchester) lors de leur transmission sur une ligne affectée par un bruit gaussien additif (AWGN). Le processus a consisté à générer les signaux codés, les transmettre dans un environnement bruité, puis les filtrer pour estimer les symboles initiaux. Les résultats obtenus ont été analysés en termes de distorsion, de robustesse au bruit et d'erreurs de transmission.

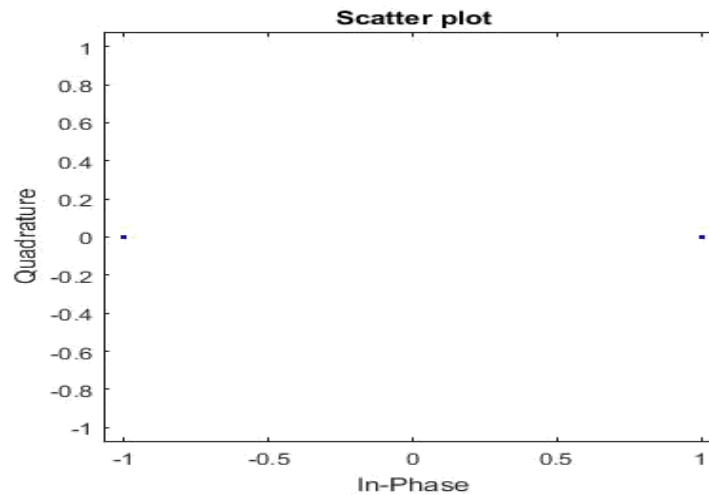
### II. Génération d'une suite binaire et Modulation BPSK

#### 1. Code Matlab

```
clear all ;
close all ;
N=10^6 ;
SNR=[0:10];
M=2; %Modulation
ncoef=3;%nombre des coefficients d'egaliseur ZF
%generation d'un signal aleatoire
x= randi([0 M-1],1,N) ;
figure(1)
plot(x)
title('signal aleatoire');
xlabel('temps');
ylabel('amplitude');
grid on ;
%Modulation bpsk
x2=pskmod(x,M);
scatterplot(x2);
title(' constellation BPSK');
```

---

## 2. Représentation graphique :



## III. Génération du canal h

Génération du canal  $h = [0.2, 0.9, 0.3]$

Génération du signal à la sortie du canal

Ajout de bruit blanc (SNR=5dB)

### 1. Code

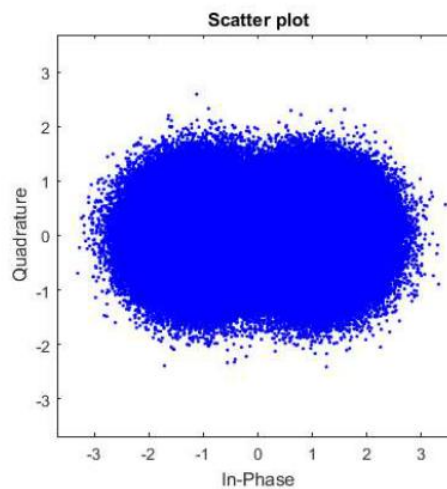
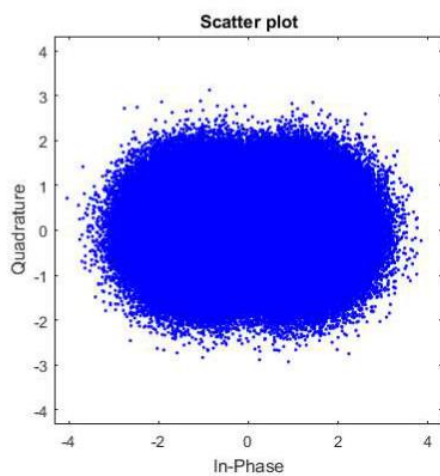
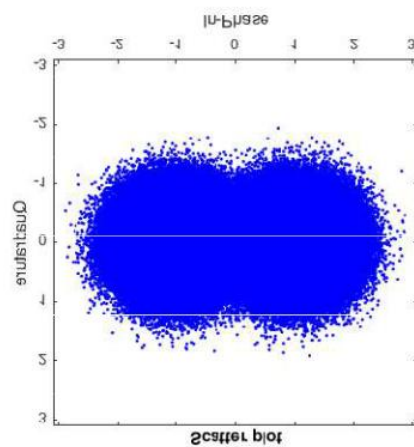
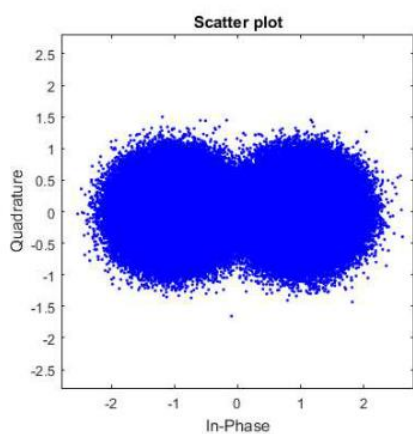
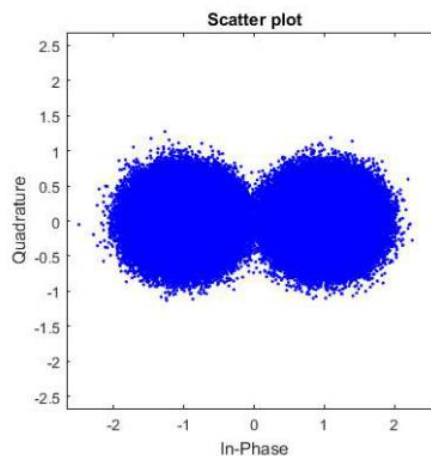
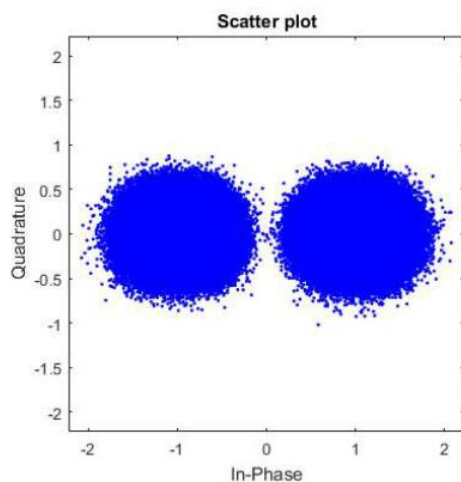
```
close all;
clear all;
clc;
N=10^6;
M=2;
x=randint(1,N,M);
y=pskmod(x,M);
scatterplot(y);
snr=[0 : 10];
h=[0.2 0.9 0.3];
w=conv(y,h);
for j =1 : length(snr)
n=1/sqrt(2)*(randn(1,N+length(h)-1)+1i*randn(1,N+length(h)-1));
y1=w + 10^(-snr(j)/10)*n;

T=toeplitz([h(2 : end) zeros(1,2+1-3+1)], ...
[h(2:-1:1) zeros(1,2+1-3+1)]);
```

---

```
d=zeros(1,3);
d(2)=1;
c = [inv(T)*d.'].';
yfilt=conv(y1,c);
yfilt1=yfilt(3 : end);
ysamp=yfilt1(1:1:N);
scatterplot(ysamp);
xestime=pskdemod(ysamp,M);
err(1,j)=size(find([x-xestime]),2);
end
simBER=err/N;
TheoryBER=(1/2)*erfc(sqrt(10.^(snr/10)));
semilogy(snr,simBER,'bs-');
hold on
semilogy(snr,TheoryBER,'rx-');
axis([0 10 10^-3 0.5]);
grid on
legend('simule(3coef)','theorique');
xlabel('snr');
```

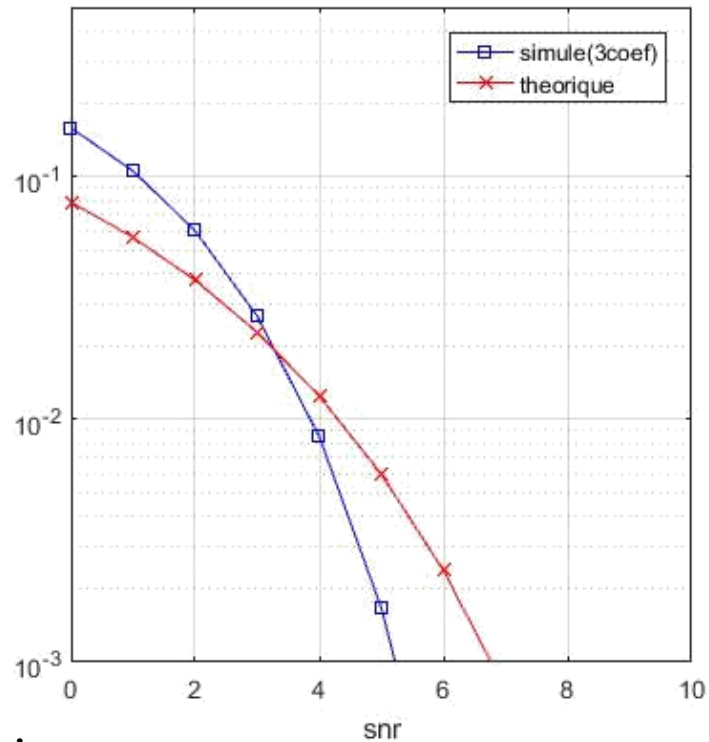
## 2. Representation



---

## IV. Analyse des performances en fonction du SNR

### 1. Code Matlab



## V. Conclusion

Pour un SNR faible (BER élevé), on utilise un égaliseur or pour le SNR fort (BER faible), on utilise theorie comme il est comparé dans la gure 12.