Rapport de TP

Communication Numérique

Chaine de simulation d'une chaine de transmission numerique sur canal gaussien a bande limitée

Réalisé par :

Younes Hallal & Abdessamad Maftah

Année universitaire 2024–2025
Grenoble INP Phelma

Table des matières

1 Introduction

Ce rapport s'inscrit dans le cadre du TP de Communication Numérique portant sur la simulation d'une chaîne de transmission numérique sur un canal gaussien à bande limitée. Nous utilisons ici une modulation 2-PAM et nous nous basons sur les outils MATLAB pour simuler les différentes étapes de transmission, de la génération des données binaires à l'analyse du signal modulé et de sa constellation. Les résultats obtenus sont commentés en les liant aux concepts théoriques abordés en cours, afin de vérifier la cohérence entre théorie et expérimentation.

Initialisation du code MATLAB :

```
%% Initialisation
clear all; close all; clc;

%% Parametres generaux
%N = 16;            % Nombre de bits (Question 1)
N = 2048;
T = 1e-6;           % Dur e symbole (1  s ) (Question 3)
F = 16;             % Facteur de surechantillonnage (Question 5)
alpha = 0.5;        % Roll-off pour SRRC (Question 6)
L = 8;              % Longueur du filtre (en symboles) (Question 6)
EbN0_dB = 100;       % Rapport signal/bruit (dB) (Question 11)
```

2 Question 1 ; Génération aléatoire des éléments binaires

Code MATLAB

```
%% === Question 1 === (Avec rand et round)
N = 2048;
bn = round(rand(1, N));          % G n ration des bits
moyenne_bn = mean(bn);
variance_bn = var(bn);

disp('=== Question 1 ===');
disp(['Moyenne empirique : ', num2str(moyenne_bn), ' (th orie : 0.5)']);
disp(['Variance empirique : ', num2str(variance_bn), ' (th orie : 0.25)']);
```

Commentaire

La suite $\{b_n\}$ contient des données binaires aléatoires et équiprobables. Le cours indique que ces bits peuvent être modélisés par une variable aléatoire suivant une loi de Bernoulli de paramètre $p = 0{,}5$, donc :

- **Espérance théorique** : $\mathbb{E}[b_n] = p = 0{,}5$
- **Variance théorique** : $\mathrm{Var}[b_n] = p(1-p) = 0{,}25$

Les valeurs empiriques obtenues sont :

- Moyenne empirique : 0.50732
- Variance empirique : 0.25007

Ces résultats sont cohérents avec la théorie, ce qui valide la méthode de génération utilisée.

3 Question 2 : Mapping 2-PAM

Code MATLAB

```
%% === Question 2 ===
ak = 2 * bn - 1;                    % Mapping 2-PAM

moyenne_ak = mean(ak);              %moyenne
variance_ak = var(ak);              %variance
puissance_ak = mean(ak.^2);         %puissance moyenne
disp('=== Question 2 ===');
disp(['Moyenne empirique de ak : ', num2str(moyenne_ak), ' (theorie : 0)']);
disp(['Variance empirique de ak : ', num2str(variance_ak), ' (theorie : 1)']);
disp(['Puissance moyenne de ak : ', num2str(puissance_ak), ' (theorie : 1)']);
```

Commentaire

Le mapping 2-PAM transforme les bits binaires selon la relation $a_k = 2b_k - 1$, ce qui donne une suite $\{a_k\}$ prenant des valeurs dans $\{-1, +1\}$.

Ces symboles sont supposés être centrés et équiprobables, donc leur distribution est symétrique autour de 0 :

- **Espérance théorique** : $\mathbb{E}[a_k] = 0{,}5 \times (+1) + 0{,}5 \times (-1) = 0$
- **Variance théorique** : $\mathrm{Var}[a_k] = \mathbb{E}[a_k^2] - \mathbb{E}[a_k]^2 = 1 - 0 = 1$
- **Puissance théorique** : $P(a_k) = \mathbb{E}[a_k^2] = 1$ car les seules valeurs possibles de a_k sont ± 1, donc $a_k^2 = 1$ pour tout k

Les résultats empiriques sont :

- Moyenne : 0.014648
- Variance : 1.0003
- Puissance moyenne : 1

Les valeurs mesurées sont en accord avec la théorie. Cela signifie que le mapping a bien été effectué et que les symboles sont conformes à l'hypothèse de modulation en 2-PAM centrée.

4 Question 3 : Représentation temporelle de a_k

Code MATLAB

```
%% === Question 3 ===
N = 16;
t_a = (0:N-1) * T;
figure;
plot(t_a, ak, '*');
xlabel('Temps (s)'); ylabel('Amplitude (V)');
title('Question 3 : Symboles 2-PAM dans le temps');
grid on;
```

Figure

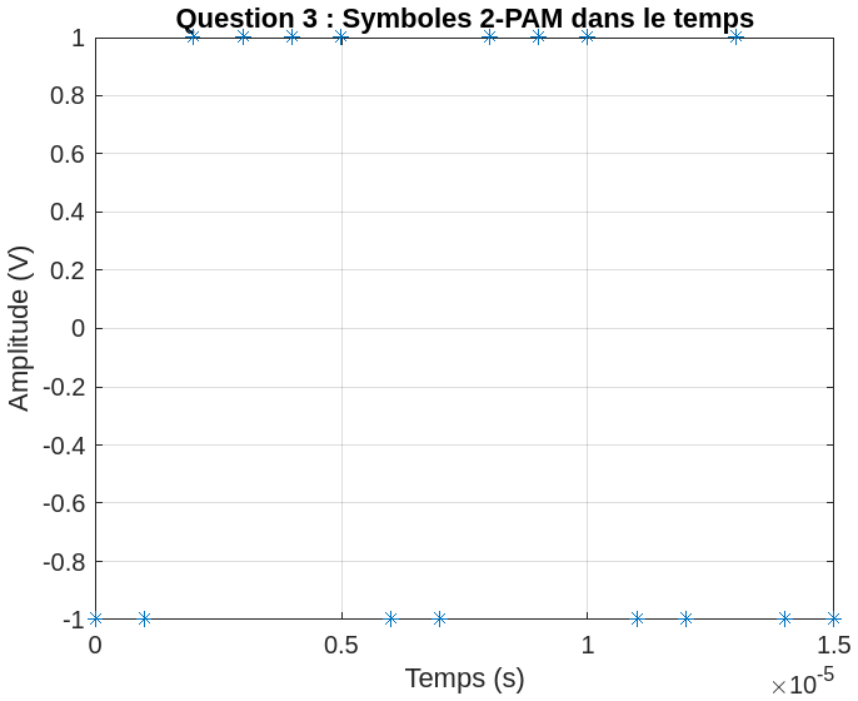

FIGURE 1 – Représentation temporelle d'une suite $\{a_k\}$ pour $N = 16$

Commentaire

On observe que les symboles a_k sont bien représentés comme des impulsions discrètes, placées à intervalles réguliers $T = 1\ \mu s$. Cela permet de visualiser la suite de symboles qui sera ensuite convertie en signal continu par mise en forme.

5 Question 4 : Constellation 2-PAM

Code MATLAB

```
    %% === Question 4 ===
figure;
plot(real(ak), imag(ak), 'x');
xlabel('Partie R elle'); ylabel('Partie Imaginaire');
title('Question 4 : Constellation 2-PAM');
axis([-1.5 1.5 -0.1 0.1]);
grid on;
```

À insérer par le binôme

Figure

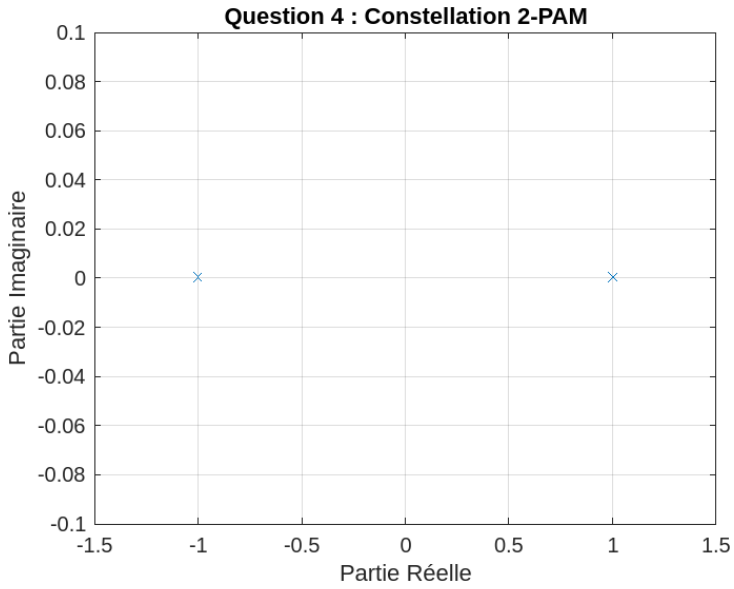

FIGURE 2 – Constellation des symboles 2-PAM dans le plan complexe

Commentaire

La constellation 2-PAM est constituée de deux points situés symétriquement sur l'axe réel, en $+1$ et -1, et avec une partie imaginaire nulle. Cela correspond à une modulation en bande de base réelle, ce qui est le cas de la 2-PAM. Cette disposition assure une bonne distance minimale entre symboles, favorable à la détection au récepteur en présence de bruit.

6 Question 5 : Expansion temporelle du signal — Surechantillonnage

Code MATLAB

```
    %% === Question 5 ===
st = zeros(1, N * F);
st(1:F:length(ak)*F) = F * ak;
t_s = (0:length(st)-1) * T / F;
figure;
plot(t_s, st);
xlabel('Temps (s)'); ylabel('Amplitude (V)');
title('Question 5 : Signal surechantillonn  st');
grid on;
```

Figure

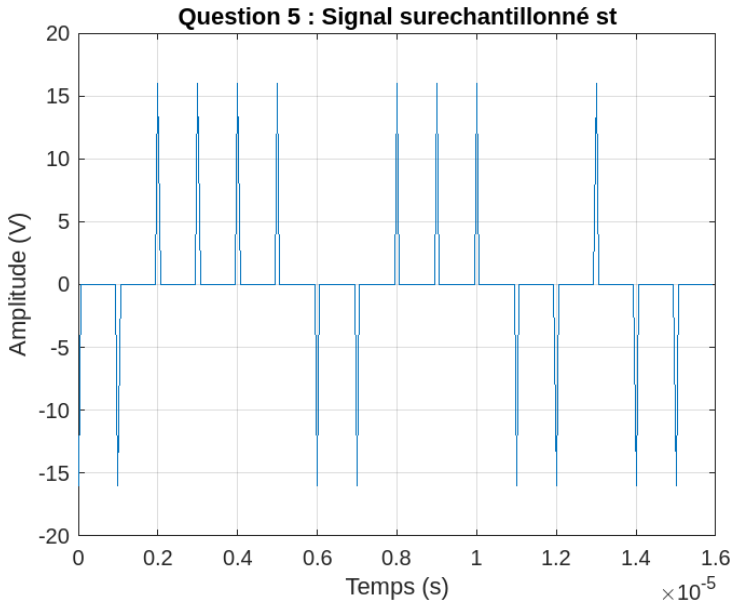

FIGURE 3 – Représentation du vecteur st

Commentaire

L'objectif ici est de simuler la conversion numérique–analogique via une expansion temporelle du signal : on insère $F-1$ zéros entre chaque symbole de la suite $\{a_k\}$, ici avec $F = 16$. On obtient ainsi le vecteur s_t, dont les valeurs non nulles apparaissent tous les F échantillons.

Chaque impulsion dans s_t est également multipliée par F. Ce facteur n'est pas arbitraire : il permet de garantir que, malgré l'insertion de zéros, la puissance du signal filtré ultérieurement reste comparable à celle de la suite initiale a_k. En effet, dans le domaine du traitement du signal, toute expansion temporelle dilue la puissance, et la compensation par F rétablit un équilibre énergétique.

En termes de puissance, le signal s_t a une puissance moyenne donnée par :

$$P(s_t) = F \cdot P(a_k)$$

Ce résultat s'explique simplement : une impulsion $F \cdot a_k$ au lieu de a_k a une énergie $F^2 \cdot a_k^2$, mais elle n'apparaît qu'une fois tous les F échantillons. Donc, la moyenne sur F échantillons est bien $F \cdot \mathbb{E}[a_k^2] = F \cdot P(a_k)$. Dans notre cas, avec $P(a_k) = 1$, on a donc $P(s_t) = 16$.

Enfin, la représentation graphique de s_t permet de vérifier :

— que l'espacement temporel est bien de $F \cdot T$ entre impulsions,

— que l'amplitude des impulsions est bien multipliée par F,

— et que la structure temporelle respecte le schéma attendu d'un vecteur suréchantillonné.

7 Question 6 : Étude des filtres de mise en forme

Code MATLAB

```
    %% === Question 6 ===
t_filtre = 0:T/F:L*T - T/F;
h_e = gen_filters3('srrc', t_filtre, T, F, L, alpha);


% R ponse impulsionnelle
figure;
```

```
subplot(2,1,1);
plot(t_filtre, h_e);
title('Question 6 : R ponse impulsionnelle du filtre SRRC');
xlabel('Temps (s)'); ylabel('Amplitude');

% R ponse fr quentielle
subplot(2,1,2);
spectrum2(h_e, T/F, 'lin');
title('R ponse fr quentielle');
```

Filtre NRZ (Non-Return to Zero)

Réponse impulsionnelle $h_t^{(e)}$ (NRZ)

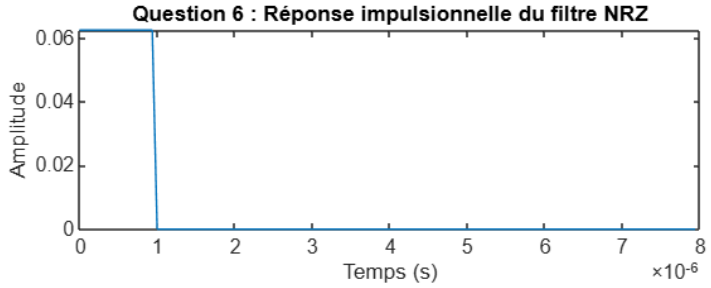

FIGURE 4 – Réponse impulsionnelle NRZ

Réponse fréquentielle $h_t^{(e)}$ (NRZ)

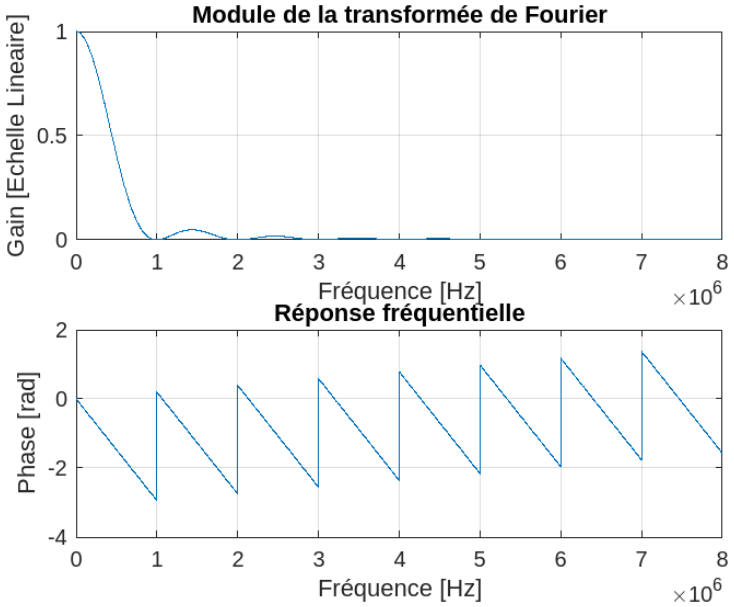

FIGURE 5 – Réponse fréquentielle NRZ

Commentaire

Le filtre NRZ a une réponse impulsionnelle constante sur une durée T. Son spectre est une fonction sinc (type sin(x)/x), avec une bande passante relativement étroite mais des lobes secondaires importants, ce qui peut engendrer de l'interférence inter-symboles (IES) si le canal ne le transmet pas parfaitement.

Filtre RZ (Return to Zero)

Réponse impulsionnelle $|H^{(e)}(f)|$ (RZ)

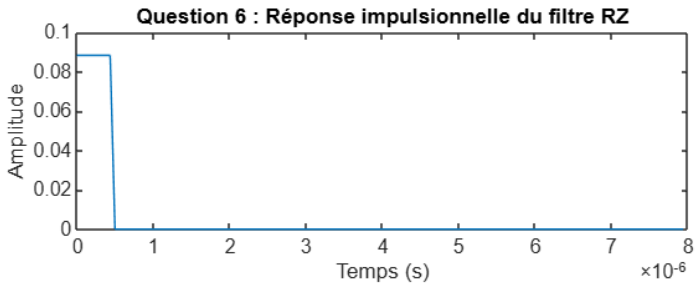

FIGURE 6 – Réponse impulsionnelle RZ

Réponse fréquentielle $|H^{(e)}(f)|$ (RZ)

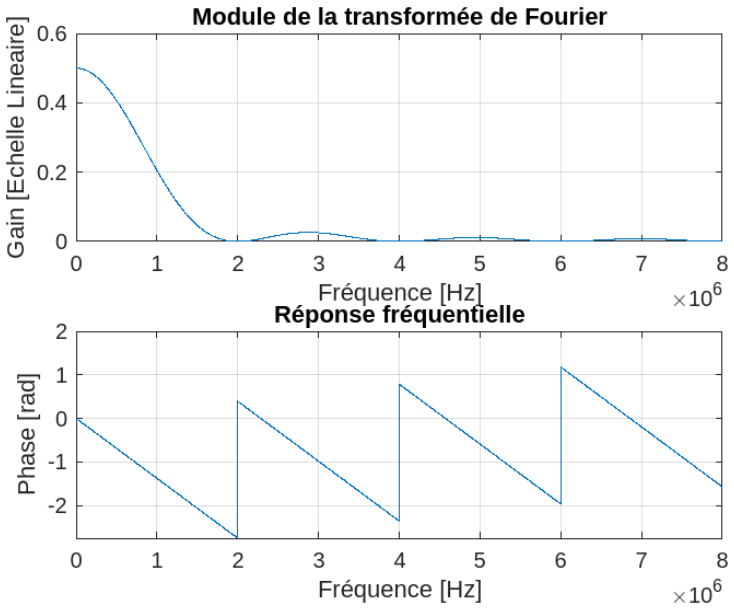

FIGURE 7 – Réponse fréquentielle RZ

Commentaire

Le filtre RZ présente une impulsion de largeur $T/2$ revenant à zéro sur la deuxième moitié du symbole. Cela donne une meilleure distinction temporelle des symboles mais au prix d'un spectre plus large, donc moins compact que le NRZ. La puissance du signal est également répartie sur une durée plus courte, ce qui peut affaiblir le signal transmis.

Filtre SRRC (Root Raised Cosine)

Réponse impulsionnelle $|H^{(e)}(f)|$ (SRRC)

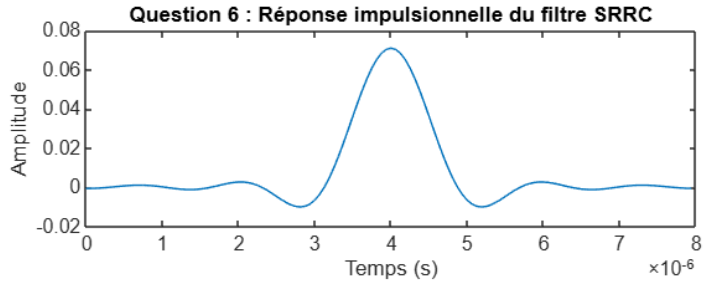

FIGURE 8 – Réponse impulsionnelle SRRC

Réponse fréquentielle $|H^{(e)}(f)|$ (SRRC)

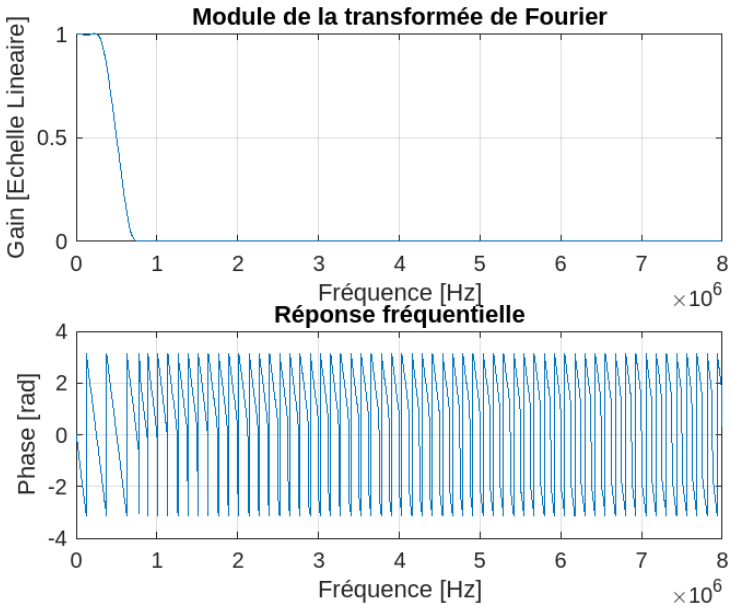

FIGURE 9 – Réponse fréquentielle SRRC

Commentaire

Le filtre SRRC est un filtre optimal pour les transmissions numériques car il permet une transmission sans IES (satisfait le critère de Nyquist) tout en limitant la bande passante. Le facteur de roll-off α ajuste l'équilibre entre largeur spectrale et durée temporelle de l'impulsion :

- α proche de 0 : spectre très étroit mais impulsion longue
- α proche de 1 : impulsion courte mais spectre plus large

Pourquoi observe-t-on une variation linéaire de la phase avec la fréquence ? Lorsqu'un filtre présente une phase linéaire, cela signifie que la phase $\phi(f)$ varie proportionnellement à la fréquence :

$$\phi(f) = -2\pi f t_0$$

Cette relation correspond à un simple retard temporel de t_0 appliqué au signal dans le domaine fréquentiel. Toutes les composantes fréquentielles du signal sont donc retardées de la même manière, ce qui est équivalent à un décalage global dans le temps sans distorsion. Le signal conserve sa forme et ne subit pas d'étalement ni d'interférence inter-symboles liée à la phase.

Dans le cas de NRZ par exemple , on sait que la réponse impulsionnelle est une porte décalée de t0 =T/2 . Ainsi ce retard apparait en forme d'exponentielle d'une fonction linéaire de f dans la phase .

Les filtres utilisés ici (NRZ, RZ, SRRC) sont tous conçus pour avoir une phase linéaire dans leur bande passante utile, assurant ainsi un traitement cohérent du signal dans le domaine temporel.

Conclusion Chaque filtre représente un compromis différent entre temps, bande passante et performance. Le SRRC, grâce à sa souplesse via le paramètre α, est souvent préféré dans les systèmes modernes.

8 Question 7 : Visualisation des signaux s_t et x_t

Code MATLAB

```
    % Filtrage par convolution (xt)
N = 16;
xt = conv(st, h_e);      % 'full' pour garder toute la convolution
L_xt = length(xt);              % Longueur du signal filtr

% Vecteurs temps
t_st = (0:length(st)-1) * (T/F);               % Temps pour st
t_xt = (0:L_xt-1) * (T/F);                     % Temps pour xt

% Affichage (subplot)
figure;
subplot(2,1,1);
plot(t_st, st);
grid;
axis([0 L_xt*T/F -1.5 1.5]);    % M me   chelle   de temps pour les deux signaux
title('Signal avant filtre de mise en forme (st)');
xlabel('Temps [s]');
ylabel('Amplitude [V]');

subplot(2,1,2);
plot(t_xt, xt, 'r');
grid;
axis([0 L_xt*T/F -1.5 1.5]);
title('Signal apr s filtre de mise en forme SRRC (xt)');
xlabel('Temps [s]');
ylabel('Amplitude [V]');
```

Figure

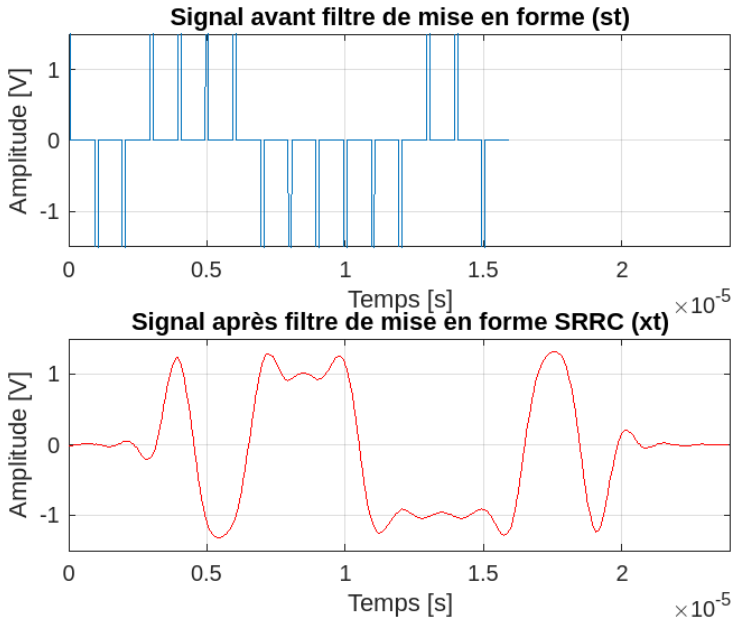

FIGURE 10 – Signaux st et xt

Commentaire

On compare deux signaux :

— s_t est un signal discret obtenu après expansion temporelle. Il contient des zéros entre les impulsions (surechantillonnage), ce qui donne une suite de pics espacés dans le temps.

— x_t est obtenu par filtrage de s_t avec le filtre $h_t^{(e)}$. Chaque impulsion devient alors une forme *lissée* (comme une porte, un RZ ou un SRRC selon le filtre choisi).

Ce qu'on observe :

— Le signal x_t est plus régulier et sans discontinuités contrairement à s_t.

— La longueur de x_t est plus grande que celle de s_t, car la convolution allonge le signal :

$$L_{x_t} = L_{s_t} + L_{h^{(e)}} - 1$$

Définition du vecteur temps

Le vecteur temps associé à x_t est défini comme :

```
t_xt = (0:L_xt-1) * (T/F);
```

Il permet d'afficher correctement le signal x_t sur l'axe temporel.

9 Question 8 : Mesure de la puissance de x_t

Code MATLAB

```
    %% === Question 8 ===
N = 2048;
P_xt = mean(xt.^2);
disp('=== Question 8 ===');
disp(['Puissance de xt : ', num2str(P_xt), ' (th orie      1)']);
```

Commentaire

On mesure ici la puissance moyenne du signal x_t obtenu à l'émission.
La puissance empirique est définie discrètement comme :

$$P(x_t) = \frac{1}{N} \sum_{n=0}^{N-1} x_t[n]^2$$

Théorie : D'après le cours, la puissance du signal $x(t)$ dans un système de modulation linéaire est donnée par :

$$P(x) = \mathbb{E}[a_k^2] \cdot T \cdot \|h^{(e)}\|^2$$

Dans notre cas discret avec un facteur de surechantillonnage F, la formule devient :

$$P(x_t) = F \cdot \|h^{(e)}\|^2 \cdot \mathbb{E}[a_k^2]$$

Or, on sait que :
- $\mathbb{E}[a_k^2] = 1$
- $\|h^{(e)}\|^2 = \frac{1}{F}$ (comme indiqué dans l'énoncé)

Donc la puissance théorique est :

$$P(x_t) = F \cdot \frac{1}{F} \cdot 1 = 1$$

Ce résultat montre que le système conserve bien la puissance initiale des symboles à travers les étapes d'expansion et de mise en forme.

Cohérence : On compare maintenant cette valeur à la puissance mesurée empiriquement :
- Puissance théorique : 1
- Puissance empirique : 0.99614

10 Question 9 : Visualisation de la densité spectrale de puissance (DSP) de x_t

Code MATLAB

```
N = 500;
    figure;
spectrum2(xt, T/F, 'log');
title('Question 9 : DSP du signal  mis  ( chelle  log)');
```

Figure

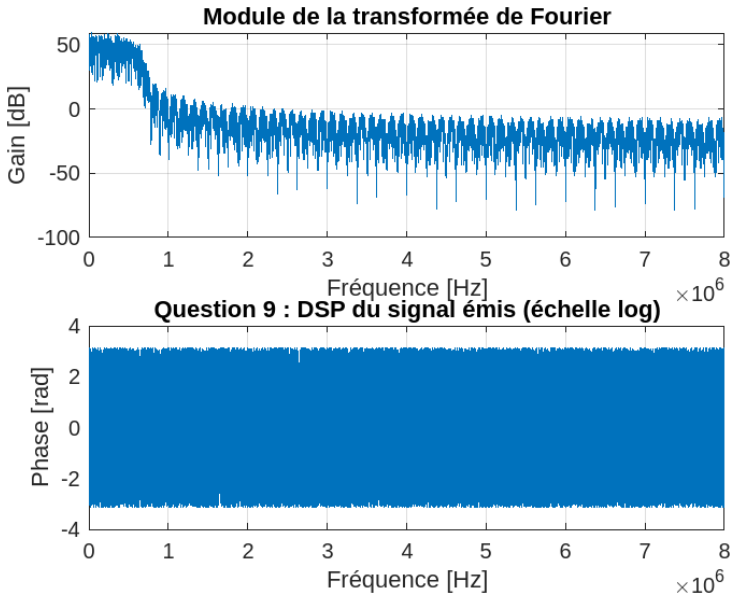

FIGURE 11 – DSP de xt (filtre SRRC)

Commentaire

On analyse ici la répartition de l'énergie du signal x_t en fonction des fréquences grâce à sa densité spectrale de puissance (DSP).

Le signal x_t est obtenu par mise en forme d'une suite aléatoire de symboles a_k, via le filtre $h_t^{(e)}$. D'après le cours, la DSP du signal x_t est liée à celle des symboles modulés et au filtre de mise en forme :

$$S_{x_t}(f) = P(a_k) \cdot |H^{(e)}(f)|^2$$

Puisque $P(a_k) = 1$, on retrouve :

$$S_{x_t}(f) = |H^{(e)}(f)|^2$$

Ainsi, le spectre du signal x_t correspond au module au carré de la réponse en fréquence du filtre. On retrouve alors les caractéristiques spectrales du filtre utilisé :
— Pour un NRZ : forme en sinc^2 avec des lobes secondaires marqués
— Pour un RZ : bande plus étendue
— Pour un SRRC : bande plus compacte si α faible, ou plus large si α grand

La figure obtenue permet donc de vérifier que le signal x_t occupe bien la bande de fréquence prévue, et que la mise en forme a bien un effet de filtrage en fréquence.

11 Question 10 : Effet des symboles non centrés

Code MATLAB

```
    %% === Question 10 ===
N = 16;
% G n ration des symboles non centr s
ak_non_centre = 2 * bn;  % 0     0, 1     2

% Construction de st_non_centre
st_non_centre = zeros(1, N * F);
```

```
st_non_centre(1:F:end) = F * ak_non_centre;

% Filtrage
xt_non_centre = conv(st_non_centre, h_e);

% Affichage comparatif
figure;
subplot(2,1,1);
plot(t_xt, xt);  % signal centr
title('Signal x_t avec symboles centr s');
xlabel('Temps');
ylabel('Amplitude');
grid on;

subplot(2,1,2);
plot(t_xt, xt_non_centre);
title('Signal x_t avec symboles non centr s');
xlabel('Temps');
ylabel('Amplitude');
grid on;
```

Figures

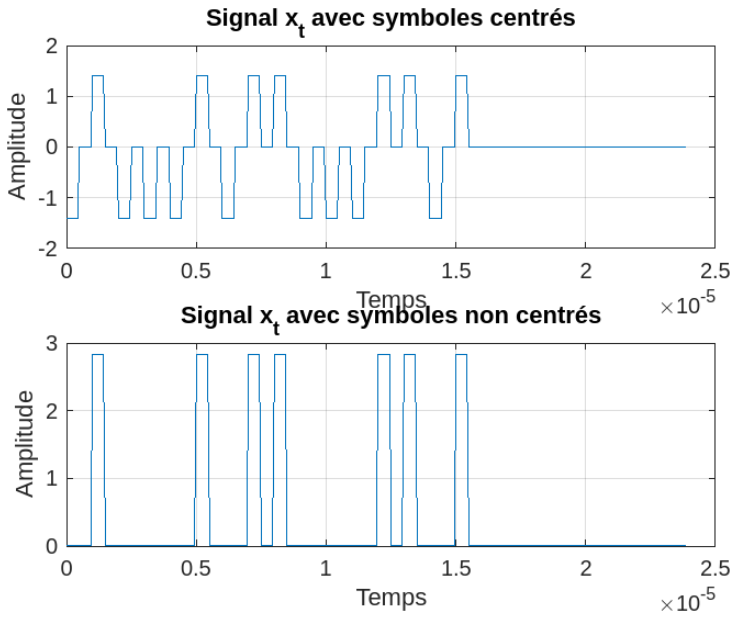

FIGURE 12 – Signal avec symboles centrés vs non centrés

Commentaire

Lorsqu'on utilise des symboles non centrés, comme par exemple $\{0, 1\}$ au lieu de $\{-1, +1\}$, le signal transmis a une composante continue non nulle. Cela se traduit par un décalage vertical du signal dans le temps.

Ce phénomène est bien visible lorsqu'on utilise un filtre RZ, car il fait revenir le signal à zéro entre chaque impulsion. Avec des symboles centrés, la ligne moyenne du signal est nulle. Avec des symboles non centrés, on observe que le signal est globalement décalé vers le haut.

Ce décalage ne transporte pas d'information utile, et peut poser des problèmes pratiques (ex. saturation de certains composants). Il est donc préférable d'utiliser des symboles centrés.

12 Question 11 : Justification théorique du bruit ajouté

Dans cette question, on souhaite ajouter un bruit blanc gaussien de variance σ_n^2 à un signal x_t, en fonction du rapport $\frac{E_b}{N_0}$.

Lien entre $P(x_t)$ et E_b

La puissance moyenne du signal filtré x_t, notée $P(x_t)$, est directement reliée à l'énergie par bit E_b par la relation fondamentale suivante :

$$E_b = P(x_t) \cdot T$$

Dans le cas particulier de notre TP, les symboles émis sont centrés (moyenne nulle), de variance unité et le filtre d'émission est normalisé (critère précisé à la question 8), ce qui implique une puissance moyenne normalisée :

$$P(x_t) = 1$$

Ainsi, dans ce contexte précis, nous obtenons directement :

$$E_b = T$$

Justification de la formule pour σ_n^2

La densité spectrale de puissance bilatérale du bruit blanc gaussien étant $\frac{N_0}{2}$, et le signal étant suréchantillonné par un facteur F, la fréquence d'échantillonnage est donc $\frac{F}{T}$.

La variance du bruit échantillonné (σ_n^2) est alors la puissance totale du bruit dans la bande d'échantillonnage :

$$\sigma_n^2 = \frac{N_0}{2} \cdot \frac{F}{T}$$

En utilisant la relation $E_b = T$ précédemment établie, on obtient :

$$\frac{E_b}{N_0} = \frac{T}{N_0} \quad \Longrightarrow \quad N_0 = \frac{T}{E_b/N_0}$$

Ainsi, la variance du bruit s'exprime en fonction du rapport E_b/N_0 comme suit :

$$\sigma_n^2 = \frac{F}{2 \cdot (E_b/N_0)}$$

Ce résultat est cohérent avec la simulation MATLAB, où la formule finale utilisée est :

$$\sigma_n = \sqrt{\frac{F}{2 \cdot (E_b/N_0)}}$$

La Figure 13 ci-dessous présente l'énoncé original de cette question afin de faciliter sa lecture directe :

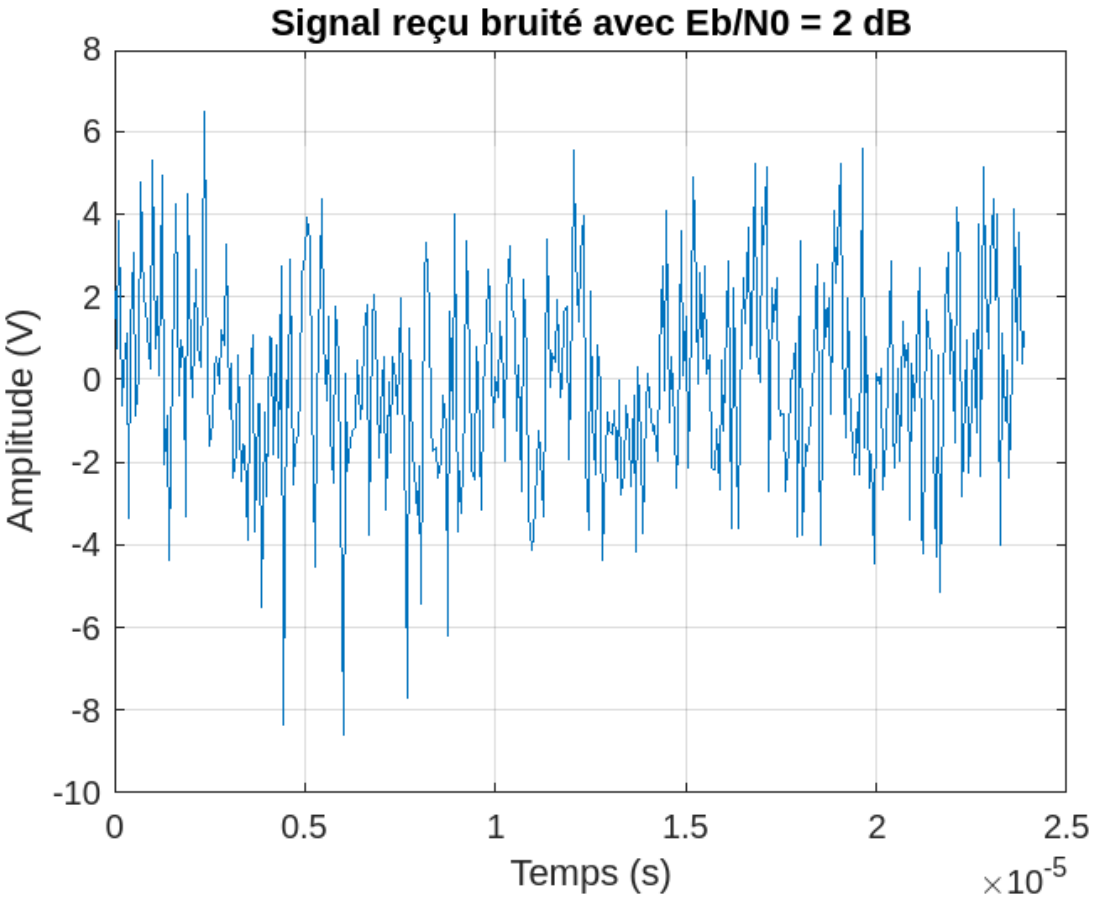

FIGURE 13 – Énoncé original de la question 11

Implémentation MATLAB correspondante modifiée

Voici le code MATLAB modifié pour préciser les axes des abscisses et ordonnées :

```
EbN0_dB = 2;
EbN0_lin = 10^(EbN0_dB / 10);
sigma_n = sqrt(F / (2 * EbN0_lin));
nt = sigma_n * randn(1, length(xt));
rt = xt + nt;

figure;
plot(t_xt, rt);
xlabel('Temps (s)');
ylabel('Amplitude (V)');
title('Signal re u bruit  avec Eb/N0 = 2 dB');
grid on;
```

Cette démonstration théorique justifie clairement l'expression utilisée dans le script MATLAB pour ajouter le bruit gaussien simulant un canal BBAG.

13 Question 12 : Filtrage adapté

Définition du filtre adapté

Le filtre adapté, noté généralement $h(t)$, est un filtre dont la réponse impulsionnelle est égale à la version conjuguée et renversée temporellement du signal utile à détecter. Formellement, pour un signal utile $s(t)$ émis sur l'intervalle $[0, T]$, le filtre adapté est défini

par :

$$h(t) = s^*(T - t)$$

Justification de l'utilisation

L'utilisation du filtre adapté dans une chaîne de communication numérique est fondamentale, car il permet de maximiser le rapport signal sur bruit (SNR) à l'instant d'échantillonnage, garantissant ainsi une détection optimale du signal transmis en présence de bruit gaussien.

Ce critère garantit toujours une détection optimale selon le critère de maximisation du rapport signal à bruit, c'est-à-dire :

$$\text{maximisation du rapport } \frac{\text{Puissance signal utile}}{\text{Puissance du bruit}}$$

C'est exactement ce que prescrit la théorie de la détection optimale pour un canal à bruit blanc additif gaussien (BBAG).

14 Question 13 : Réponse impulsionnelle du filtre équivalent émission-réception

L'objectif de cette question est d'observer la réponse impulsionnelle globale p_t obtenue par la convolution du filtre d'émission $h_t^{(e)}$ avec sa version inversée temporellement (filtre adapté causal $h_t^{(r)}$) :

$$p_t = h_t^{(e)} * \texttt{fliplr}(h_t^{(e)})$$

Le code MATLAB suivant permet d'afficher simultanément $h_t^{(e)}$ et p_t :

```
p_t= conv(h_e,fliplr(h_e));
t2 = (0:length(h_e)-1) * T / F;
t3 = (0:length(p_t)-1) * T / F;

% affichage de h_e
figure;
subplot(2,1,1);
plot(t2, h_e);
title('Question 13 : R ponse impulsionnelle du filtre SRRC');
xlabel('Temps (s)'); ylabel('Amplitude');

%affiche de p_t
subplot(2,1,2);
plot(t3, p_t);
title('R ponse impulsionelle du filtre h_t convolu  a h_e');
xlabel('Temps (s)'); ylabel('Amplitude');
```

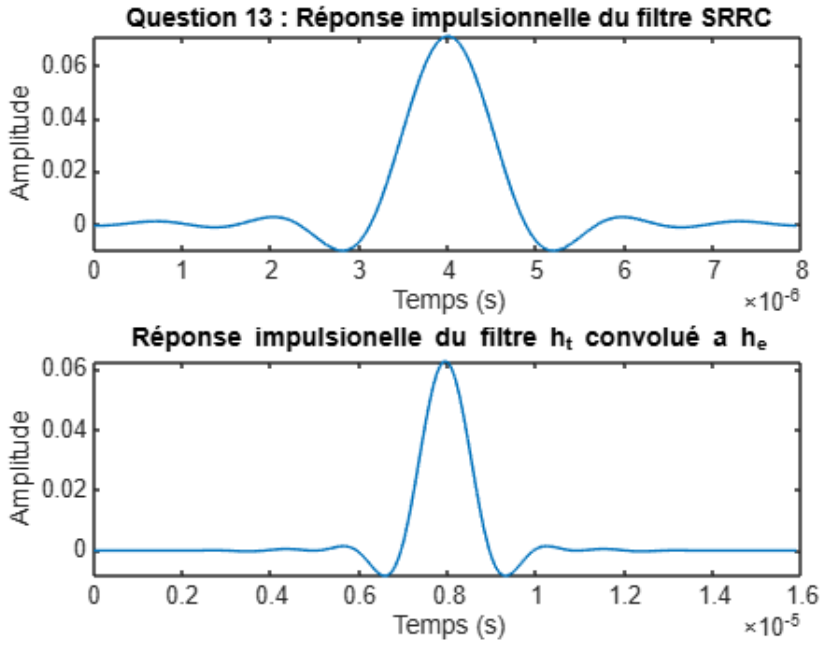

FIGURE 14 – Réponse impulsionnelle du filtre SRRC (haut) et du filtre équivalent p_t (bas)

Analyse des résultats

La réponse p_t montre un pic central clair, entouré de zéros aux instants multiples de T. Cette forme correspond exactement au **critère de Nyquist** pour l'annulation des interférences inter-symboles (ISI) :

$$p_t(nT) = \delta[n] \quad \text{(Nyquist pour ISI nulle)}$$

Ce comportement confirme que le filtre SRRC, utilisé à l'émission et à la réception, forme un système optimal pour la transmission sans ISI dans un canal BBAG.

15 Questions 14 à 15 : Diagramme de l'œil et critère de Nyquist

Vérification du critère de Nyquist

La réponse impulsionnelle globale p_t, obtenue par la convolution du filtre de mise en forme $h_t^{(e)}$ et du filtre adapté $h_t^{(r)}$, permet de vérifier visuellement le critère de Nyquist.

Selon ce critère, tous les échantillons de p_t prélevés aux instants nT (avec $n \in \mathbb{Z}$) doivent être nuls, sauf pour $n = 0$, où l'échantillon doit atteindre un maximum. Cela garantit l'absence d'interférences inter-symboles (ISI).

```
indices_nyquist = 1:F:length(p_t);
figure;
stem(indices_nyquist, p_t(indices_nyquist));
title(    chantillons     de pt aux instants multiples de  T  );
xlabel(    Index    ); ylabel(    Amplitude    );
grid on;
```

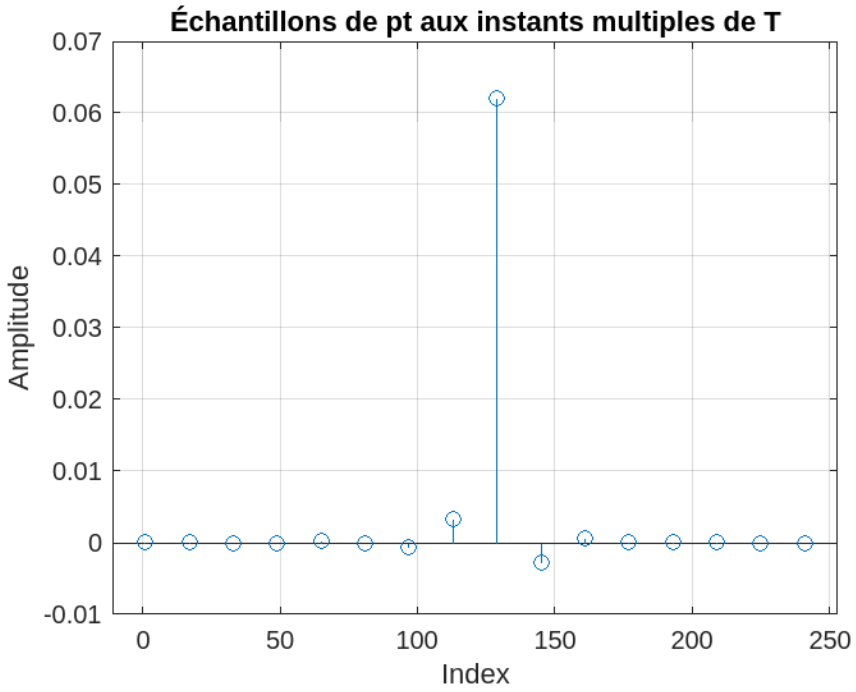

FIGURE 15 – Échantillons de p_t aux instants multiples de T : vérification du critère de Nyquist

Observation : on observe que seul l'échantillon central atteint un maximum significatif, tandis que tous les autres sont quasiment nuls. Cela confirme que le filtre équivalent satisfait bien le critère de Nyquist, condition indispensable pour éviter l'ISI en réception.

Cette vérification graphique est fondamentale, car elle assure que les symboles transmis seront échantillonnés sans interférence mutuelle.

Filtrage du signal reçu et affichage du diagramme de l'œil

Le signal reçu bruité r_t est filtré par le filtre adapté (inversé du SRRC) :

```
y_t = conv(rt, fliplr(h_e));
taille_visu = 2;
begin_offset = length(h_e);
end_offset = 2 * length(h_e);
eyepattern(y_t, T, F, taille_visu, begin_offset, end_offset);
```

Diagramme de l'œil pour différentes valeurs de E_b/N_0

Le diagramme de l'œil est une représentation temporelle superposée de trames successives. Une ouverture nette au centre indique un échantillonnage fiable. Ci-dessous, les résultats pour trois valeurs différentes de E_b/N_0 :

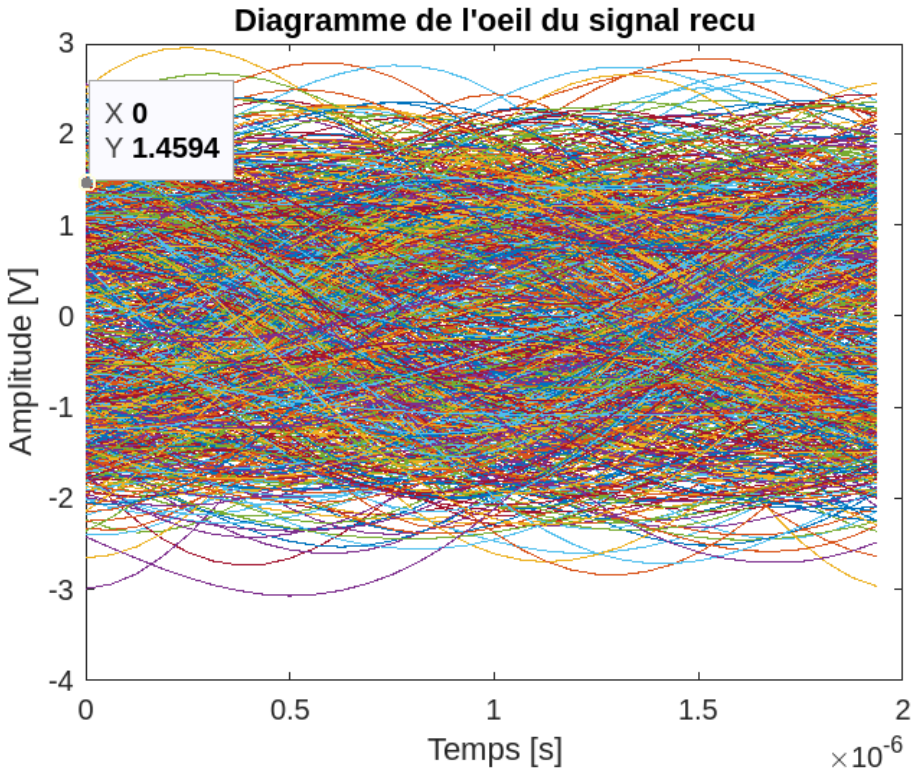

FIGURE 16 – Diagramme de l'œil pour $E_b/N_0 = 2$ dB : ouverture réduite, décision difficile.

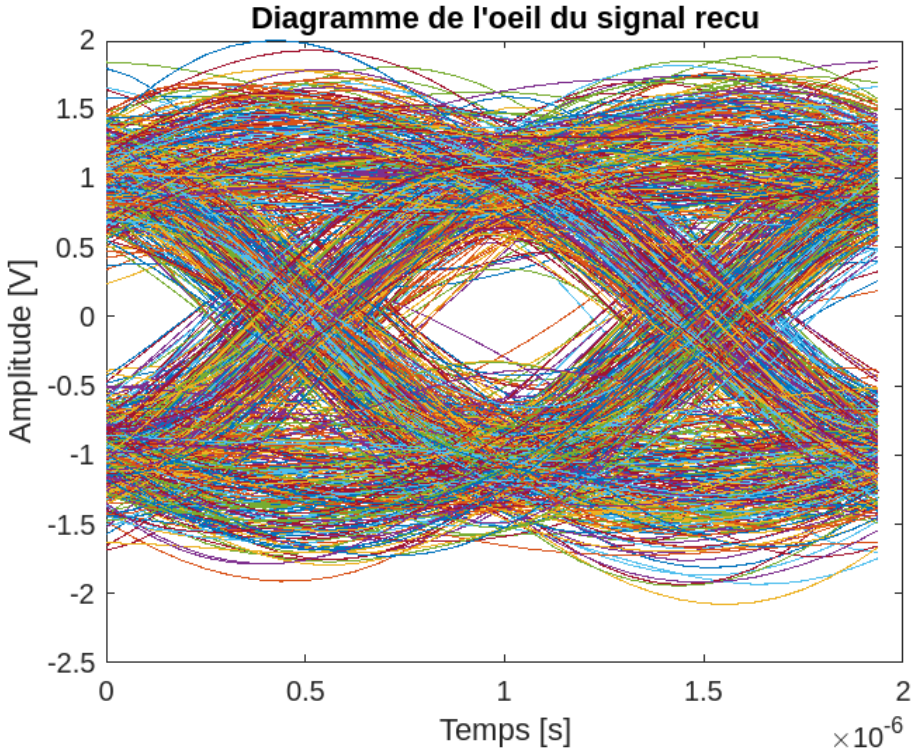

FIGURE 17 – Diagramme de l'œil pour $E_b/N_0 = 10$ dB : ouverture correcte.

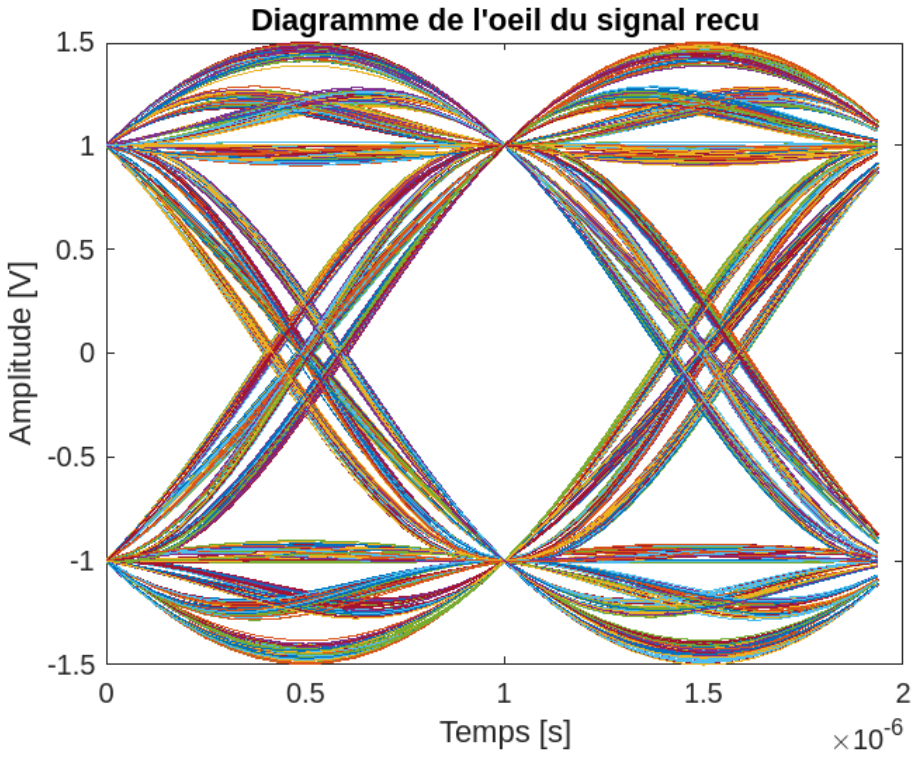

FIGURE 18 – Diagramme de l'œil pour $E_b/N_0 = 100$ dB : ouverture parfaite, ISI nul.

Interprétation et anomalies

On observe que plus le rapport E_b/N_0 est élevé, plus l'œil est ouvert, ce qui reflète une meilleure décision symbolique. Pour les faibles valeurs de E_b/N_0 (comme 2 dB), l'œil se referme et l'incertitude autour des niveaux de décision augmente fortement, traduisant un risque élevé d'erreurs de décision.

Ce comportement est conforme à la théorie vue en cours : un canal BBAG affecte directement la qualité de la décision par son bruit, et la quantité E_b/N_0 est le paramètre clé pour dimensionner la fiabilité d'une chaîne de transmission numérique.

16 Question 16 : Instants d'échantillonnage optimaux

Dans une chaîne de transmission avec un filtre de mise en forme et un filtre adapté, chacun de longueur $L \cdot T$, la réponse impulsionnelle globale p_t est centrée sur $t_0 = L \cdot T$.

En simulation, avec un suréchantillonnage par un facteur F, cet instant devient un index :

$$t_0 = L \cdot F$$

C'est à cet instant qu'il faut commencer à échantillonner le signal y_t tous les F échantillons, afin de récupérer les symboles alignés sur les pics de p_t, là où l'influence des autres symboles est nulle (critère de Nyquist satisfait). Ce choix garantit une reconstruction optimale des symboles transmis, sans interférence inter-symboles (ISI).

17 Question 17 : Effet d'un mauvais choix de synchronisation

Un mauvais choix de t_0, c'est-à-dire un décalage dans le point de départ de l'échantillonnage, implique que les symboles sont récupérés dans une zone où les contributions

des symboles voisins ne sont pas nulles. Cela génère des interférences inter-symboles (ISI) et perturbe la détection.

Conséquences :

— Fermeture ou brouillage du diagramme de l'œil ;
— Disparition des zones de décision stables ;
— Taux d'erreur binaire (TEB) plus élevé ;
— Constellation plus dispersée.

Cela illustre l'importance de la **synchronisation symbolique** : sans un bon alignement temporel, même une chaîne bien conçue et un bon rapport E_b/N_0 peuvent conduire à des performances dégradées. C'est une condition essentielle en réception numérique.

18 Question 18 : Décimation du signal reçu et visualisation des instants d'échantillonnage

L'objectif de cette question est de visualiser le signal y_t issu du filtrage adapté, puis de repérer les instants d'échantillonnage optimaux à partir de l'instant $t_0 = L \cdot F$, correspondant au pic de la réponse impulsionnelle globale.

Code MATLAB utilisé

```
% Calcul de l'instant optimal
t0 = L * F;

% Visualisation de y_t sur 500   chantillons
fenetre_plot = 500;
figure;
plot(y_t(1:fenetre_plot), 'b'); hold on;
stem(indices_ech(indices_ech <= fenetre_plot), ...
     y_t(indices_ech(indices_ech <= fenetre_plot)), 'r', 'filled');
title('Visualisation de y_t et instants    d   chantillonnage   ');
xlabel('Index'); ylabel('Amplitude');
legend('Signal y_t', 'Instants d''  chantillonnage   ');
grid on;

% Extraction des symboles re us
indices_ech = t0:F:(N-1)*F + t0;
y_k = y_t(indices_ech);

% Affichage console
disp('=== Question 18 ===');
disp(['Taille de y_k : ', num2str(length(y_k))]);
disp(['Taille attendue (N) : ', num2str(N)]);
if length(y_k) == N
    disp('>> La taille de y_k est correcte et correspond    celle de ak.');
else
    disp('>>        Taille incoh rente entre y_k et ak.');
end
```

Résultat de l'exécution

```
=== Question 18 ===
Taille de y_k : 2048
Taille attendue (N) : 2048
>> La taille de y_k est correcte et correspond    celle de ak.
```

Interprétation

Le graphique ci-dessous montre clairement les instants d'échantillonnage (en rouge) superposés au signal y_t (en bleu). On observe que ces instants coïncident bien avec les pics du signal filtré, ce qui confirme que le choix de $t_0 = L \cdot F$ permet un échantillonnage optimal, tel que prescrit par le critère de Nyquist.

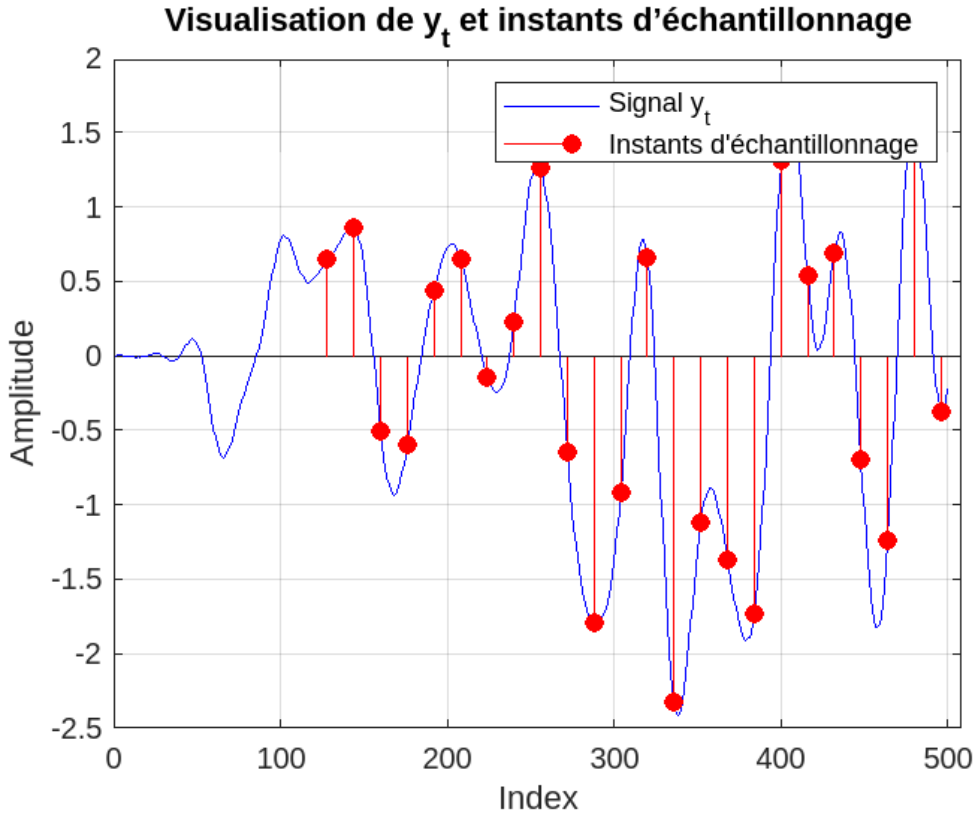

FIGURE 19 – Visualisation de y_t et des instants d'échantillonnage (Question 18)

Ce résultat est conforme au modèle théorique vu en cours, où l'échantillonnage à l'instant du maximum de la réponse impulsionnelle globale assure l'absence d'interférences inter-symboles (ISI).

19 Question 19 : Constellation des symboles reçus et seuil de décision

Affichage de la constellation

On représente les symboles reçus y_k dans le plan complexe afin d'observer leur répartition après transmission et filtrage :

```
figure;
plot(real(y_k), imag(y_k), 'x');  % Imaginaire nul ici
title('Constellation des symboles re us y_k');
xlabel('Partie r elle'); ylabel('Partie imaginaire');
grid on;
```

Seuil de décision optimal

Dans le cas d'une modulation 2-PAM avec symboles $a_k \in \{-1, +1\}$, le seuil optimal de décision est :

$$\text{Seuil optimal} = 0$$

Cette valeur minimise le taux d'erreur de bit (TEB) en milieu gaussien, en supposant :
— un canal BBAG (bruit blanc additif gaussien),
— des symboles équiprobables,
— une détection symétrique.

Visualisation pour différentes valeurs de E_b/N_0

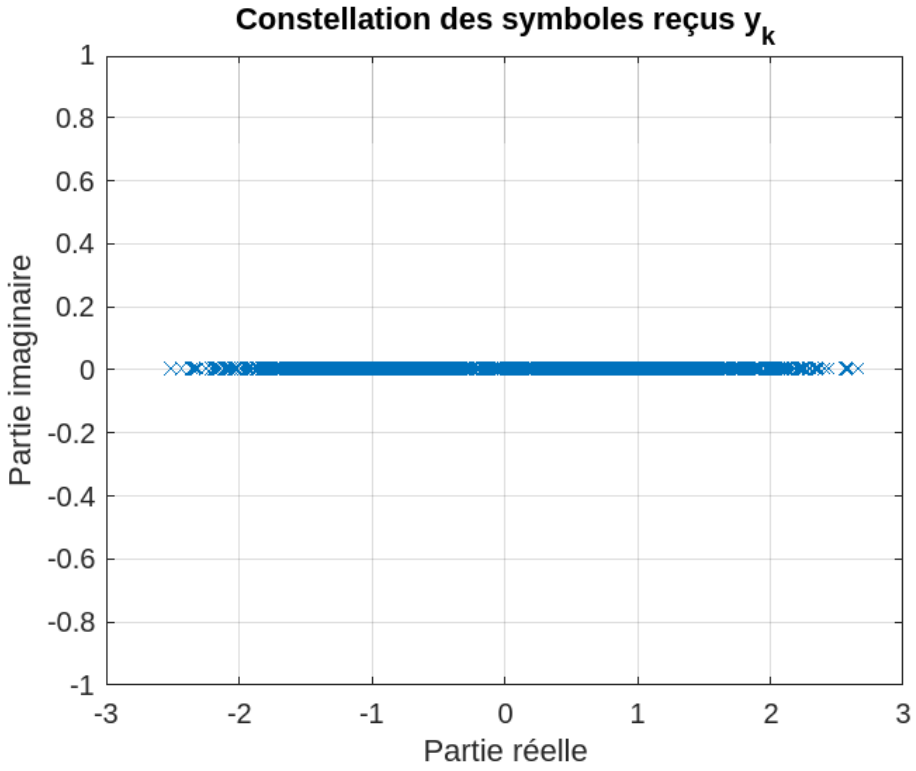

FIGURE 20 – Constellation pour $E_b/N_0 = 2$ dB : nuage dispersé, seuil difficile à appliquer.

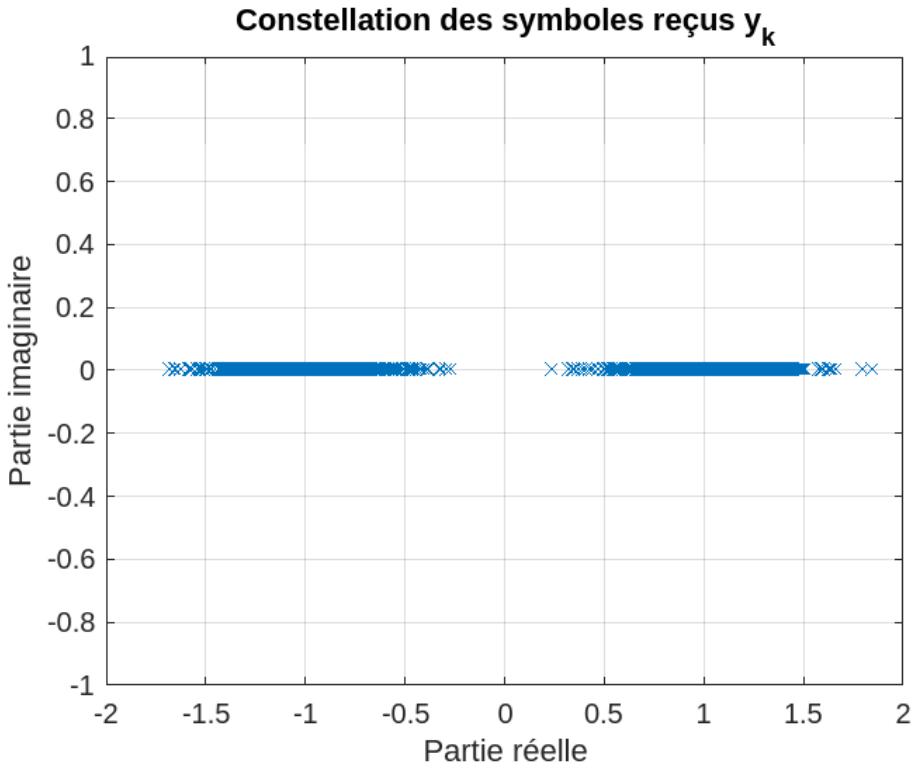

FIGURE 21 – Constellation pour $E_b/N_0 = 10$ dB : séparation plus nette entre les deux classes.

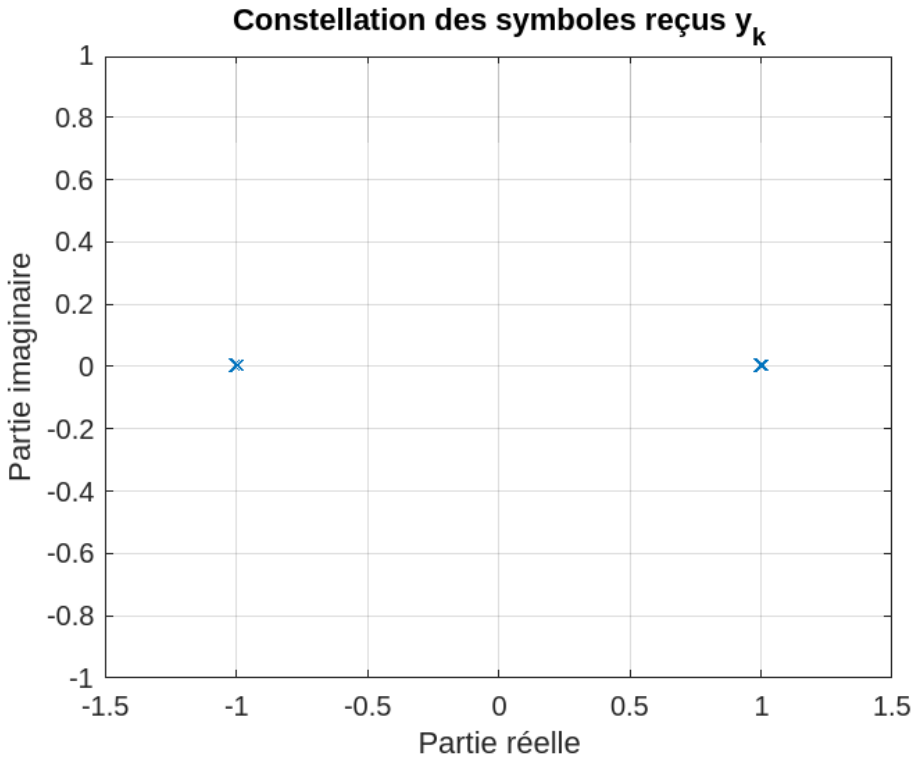

FIGURE 22 – Constellation pour $E_b/N_0 = 100$ dB : deux nuages bien séparés à -1 et $+1$.

Interprétation

On observe que plus E_b/N_0 est élevé, plus les symboles reçus sont concentrés autour de leurs valeurs théoriques. À 2 dB, les symboles sont très dispersés, rendant la décision

binaire plus sujette à erreur. À 100 dB, les clusters sont parfaitement séparés, ce qui valide l'efficacité du seuil théorique à zéro dans des conditions optimales.

Calcul du taux d'erreur binaire (TEB)

Le TEB permet d'évaluer la fiabilité de la chaîne de transmission. Il est défini comme :

$$TEB = \frac{\text{Nombre de bits erronés}}{\text{Nombre total de bits transmis}}$$

1. Code de base (TP)

Conformément aux consignes du TP, le calcul du TEB se fait par comparaison entre les bits transmis b_n et les bits estimés b_{decide} :

```
b_decide = y_k > 0;
erreurs = sum(xor(bn(1:length(bn)), b_decide));
teb = erreurs / N;
```

2. Résultats expérimentaux obtenus

```
--- Eb/N0 = 2 dB ---
TEB mesuré : 0.03906

--- Eb/N0 = 10 dB ---
TEB mesuré : 0.00000

--- Eb/N0 = 100 dB ---
TEB mesuré : 0.00000
```

Ces résultats montrent que :
— à faible E_b/N_0 (2 dB), le bruit engendre des erreurs notables (TEB 3.9%),
— à partir de 10 dB, le système devient fiable (aucune erreur observée sur 2048 bits),
— à 100 dB, les symboles sont parfaitement distinguables (cas idéal).

Le comportement du TEB est donc cohérent avec la théorie, où le TEB pour une modulation 2-PAM sur canal BBAG est donné par :

$$TEB = Q\left(\sqrt{2 \cdot \frac{E_b}{N_0}}\right)$$

où Q est la fonction de répartition complémentaire de la loi normale.

20 Question 20 : Mesures de performances et comparaison théorie/pratique

Objectif

Cette question vise à évaluer les performances de la chaîne de transmission en termes de taux d'erreur binaire (TEB), en fonction du rapport E_b/N_0, pour une modulation 2-PAM sur canal BBAG. Pour cela, on compare les courbes :
— du TEB **théorique** donné par la formule :

$$TEB_{\text{théorique}} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

— et du TEB **mesuré** expérimentalement via simulation MATLAB.

Code MATLAB utilisé

```
%% === Question 20 ===

% === Param tres  g  n  raux  ===
N = 100000;                  % Nombre de bits
EbN0_dB = 0:1:12;            % Plage de Eb/N0 en dB
F = 16; T = 1e-6; alpha = 0.5; L = 8;

% G n ration des bits + mapping
bn = round(rand(1, N));
ak = 2 * bn - 1;

% Sur chantillonnage
st = zeros(1, N * F);
st(1:F:end) = F * ak;

% Filtre SRRC
t_filtre = 0:T/F:L*T - T/F;
h_e = gen_filters3('srrc', t_filtre, T, F, L, alpha);

% Filtrage  mission
xt = conv(st, h_e);
P_xt = mean(xt.^2);

% R sultats
TEB_mesure = zeros(size(EbN0_dB));
TEB_theorique = zeros(size(EbN0_dB));

for i = 1:length(EbN0_dB)
    EbN0_lin = 10^(EbN0_dB(i)/10);
    sigma_n = sqrt(F / (2 * EbN0_lin));
    nt = sigma_n * randn(1, length(xt));
    rt = xt + nt;
    y_t = conv(rt, fliplr(h_e));
    t0 = L * F;
    y_k = y_t(t0:F:(N-1)*F + t0);
    b_decide = y_k > 0;
    erreurs = sum(xor(bn(1:N), b_decide));
    TEB_mesure(i) = erreurs / N;
    TEB_theorique(i) = qfunc(sqrt(2 * EbN0_lin));
end

% Trac  des courbes
figure;
semilogy(EbN0_dB, TEB_theorique, '-o', 'DisplayName', 'TEB th orique');
hold on;
semilogy(EbN0_dB, TEB_mesure, '--x', 'DisplayName', 'TEB mesur ');
grid on;
xlabel('E_b/N_0 (dB)');
ylabel('Taux d''erreur binaire (TEB)');
legend;
title('Courbes de performances TEB : th orie vs mesure');
```

Courbe obtenue

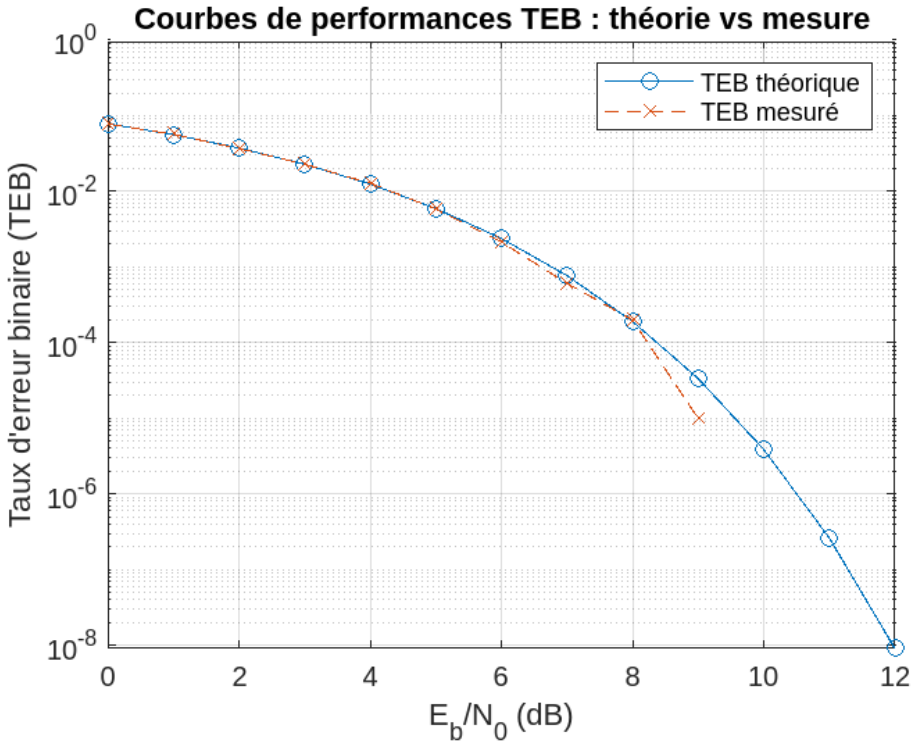

FIGURE 23 – Comparaison des courbes TEB mesuré vs théorique pour une modulation 2-PAM

Analyse et commentaire

On observe que la courbe du TEB mesuré est très proche de la courbe théorique sur l'ensemble de la plage testée ($E_b/N_0 \in [0, 12]$ dB). Cela confirme que :

— Le système simulé respecte bien le modèle BBAG supposé dans le calcul théorique.
— Le seuil de décision à 0 est optimal dans le cas d'une modulation 2-PAM symétrique.
— La conception du filtre de mise en forme et du filtre adapté (SRRC) garantit l'absence d'ISI.

Des écarts légers peuvent apparaître à très faible TEB (valeurs proches de 10^{-6}), notamment si le nombre de bits transmis N est insuffisant. Dans ce cas, il peut être nécessaire d'augmenter la taille du vecteur b_n pour obtenir une estimation plus fiable.

Ces résultats sont parfaitement cohérents avec la théorie présentée dans le cours, et valident la chaîne de simulation conçue.

Conclusion de cette partie

Le tracé des courbes TEB permet de valider empiriquement le bon fonctionnement d'une chaîne de communication numérique simulée. L'accord entre mesure et théorie constitue un critère fort de vérification de la conformité au modèle BBAG idéal.