

Département Génie des Communications et Réseaux

Compte Rendu

TP2

Chaines de transmission numériques

COMMUNICATION NUMERIQUES

Elaborés par :

Wissem Bagga

Génie des communications et Réseaux GCRA2 - Grp1

Date: 19/11/2024

Année Universitaire: 2023/2024

I. Objectifs

- Étudier la performance de la transmission des différents signaux numériques NRZ, RZ et Manchester.
- Analyser l'effet du bruit additif gaussien (AWGN) sur ces signaux.
- Générer les signaux, ajouter du bruit, et calculer le taux d'erreur binaire (BER) en fonction du rapport signal sur bruit (SNR).

II. Génération des Signaux NRZ, RZ et Manchester

- **Description**

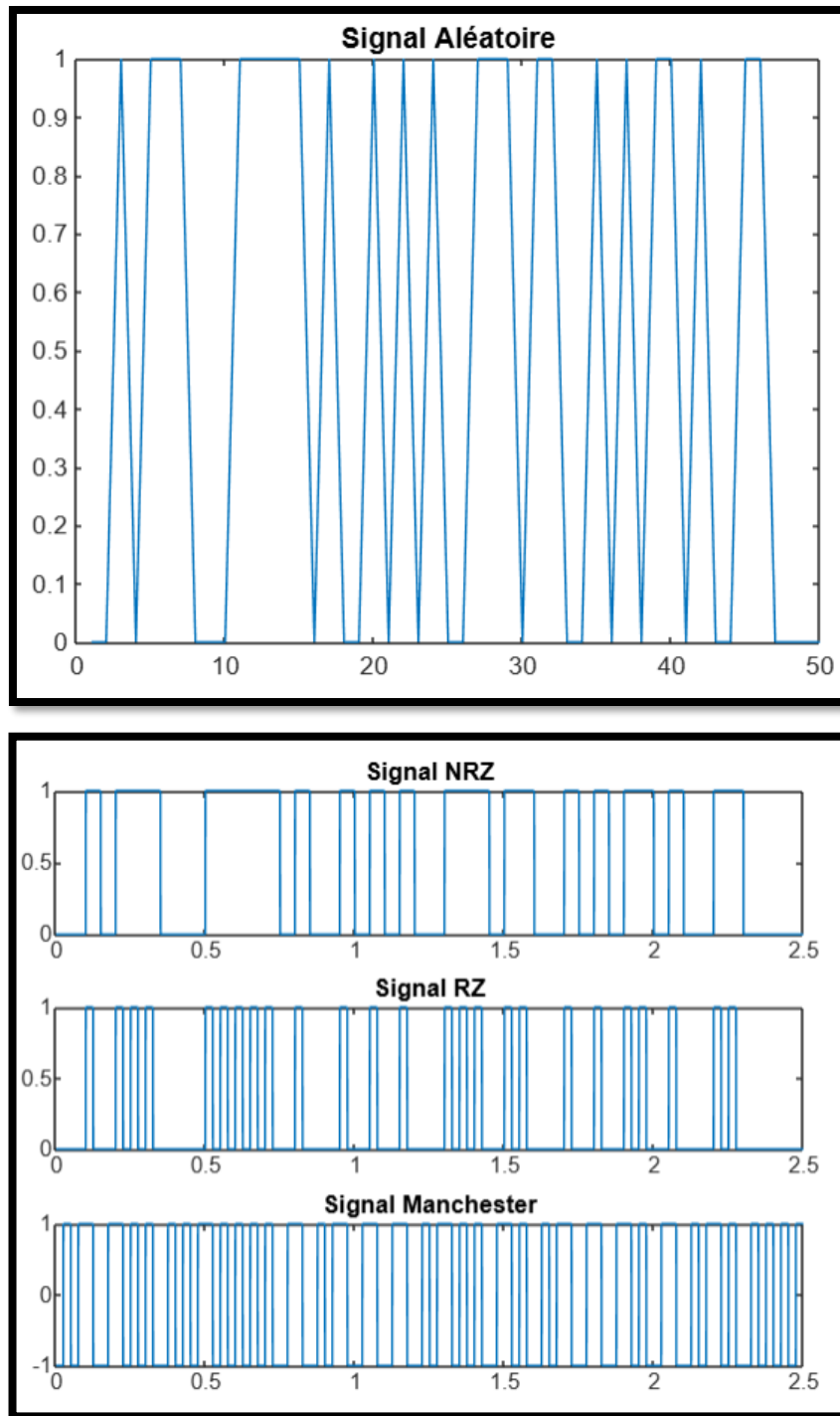
Nous avons généré trois types de signaux binaires : NRZ, RZ et Manchester. Chaque signal est converti en fonction de son schéma de codage respectif, et visualisé.

- **Code**

```
clc;
clear all;
close all;

% Paramètres
D = 0.05;
Ns = 50;
t = linspace(0, Ns * D, Ns * 100);
% Génération des symboles binaires aléatoires
symbols = randi([0, 1], 1, Ns);
figure;
plot(symbols);
title('Signal Aléatoire');
% Génération du signal NRZ
signal_NRZ = repelem(symbols, 100);
% Génération du signal RZ
signal_RZ = repelem(symbols, 100);
for i = 1:Ns
    if symbols(i) == 1
        signal_RZ((i - 1) * 100 + 51:i * 100) = 0;
    end
end
% Génération du signal Manchester
signal_Manchester = zeros(1, length(t));
for i = 1:Ns
    if symbols(i) == 0
        signal_Manchester((i - 1) * 100 + 1:(i - 1) * 100 + 50) = -1;
        signal_Manchester((i - 1) * 100 + 51:i * 100) = 1;
    else
        signal_Manchester((i - 1) * 100 + 1:(i - 1) * 100 + 50) = 1;
        signal_Manchester((i - 1) * 100 + 51:i * 100) = -1;
    end
end
% Visualisation
figure;
subplot(3, 1, 1);
plot(t, signal_NRZ);
title('Signal NRZ');
subplot(3, 1, 2);
plot(t, signal_RZ);
title('Signal RZ');
subplot(3, 1, 3);
plot(t, signal_Manchester);
title('Signal Manchester');
```

- **Représentation**



- **Interprétation**

Les trois schémas de codage présentent des différences claires dans la représentation des niveaux logiques. Le signal NRZ conserve une valeur constante sur toute la durée du symbole, tandis que le signal RZ revient à zéro à la moitié de chaque période. Le signal Manchester utilise des transitions au milieu de chaque période pour assurer une meilleure synchronisation.

III. Ajout du Bruit AWGN

- **Description**

Nous avons ajouté du bruit AWGN aux trois types de signaux afin de simuler un canal de communication réaliste avec des perturbations. Le bruit est ajouté pour différents niveaux de SNR afin d'évaluer l'impact du bruit sur la qualité de la transmission.

- **Code**

```
% Filtre adapté
h = ones(1,100);
% Ajout de bruit et transmission pour NRZ
normalized_signal_NRZ = (signal_NRZ - min(signal_NRZ))/(max(signal_NRZ)-min(signal_NRZ));
noisy_NRZ = imnoise(normalized_signal_NRZ, 'gaussian', 0, 0.01);
noisy_NRZ = noisy_NRZ * (max(signal_NRZ) - min(signal_NRZ)) + min(signal_NRZ);
received_NRZ = conv(noisy_NRZ, h, 'same');
estimated_NRZ = received_NRZ(1:100:end) > 0;

% Ajout de bruit et transmission pour RZ
normalized_signal_RZ = (signal_RZ - min(signal_RZ)) / (max(signal_RZ) - min(signal_RZ)); % Normalisation
noisy_RZ = imnoise(normalized_signal_RZ, 'gaussian', 0, 0.01);
noisy_RZ = noisy_RZ * (max(signal_RZ) - min(signal_RZ)) + min(signal_RZ); % Revenir à l'échelle originale
received_RZ = conv(noisy_RZ, h, 'same');
estimated_RZ = received_RZ(1:100:end) > 0;

% Ajout de bruit et transmission pour Manchester
normalized_signal_Manchester = (signal_Manchester - min(signal_Manchester)) / (max(signal_Manchester) - min(signal_Manchester));
noisy_Manchester = imnoise(normalized_signal_Manchester, 'gaussian', 0, 0.01);
noisy_Manchester = noisy_Manchester * (max(signal_Manchester) - min(signal_Manchester)) + min(signal_Manchester);
estimated_Manchester = received_Manchester(1:100:end) > 0;

% Calcul des erreurs
errors_NRZ = sum(estimated_NRZ ~= symbols);
errors_RZ = sum(estimated_RZ ~= symbols);
errors_Manchester = sum(estimated_Manchester ~= symbols);

% Visualisation pour NRZ
figure;
subplot(2, 1, 1);
plot(t, noisy_NRZ);
title('Signal NRZ bruité');
xlabel('Temps');
ylabel('Amplitude');

subplot(2, 1, 2);
plot(received_NRZ);
title('Signal NRZ reçu après filtrage');
xlabel('Temps');
ylabel('Amplitude');

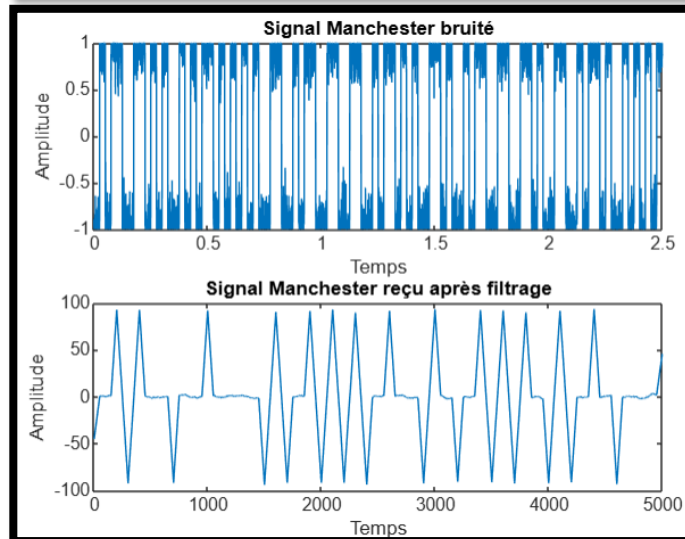
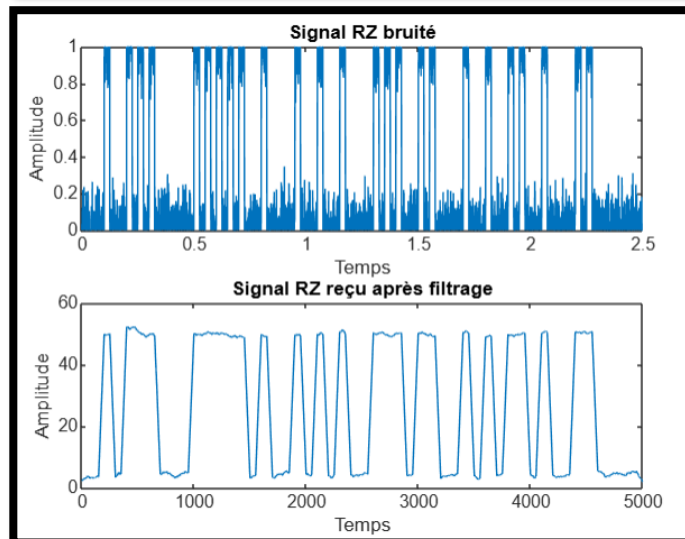
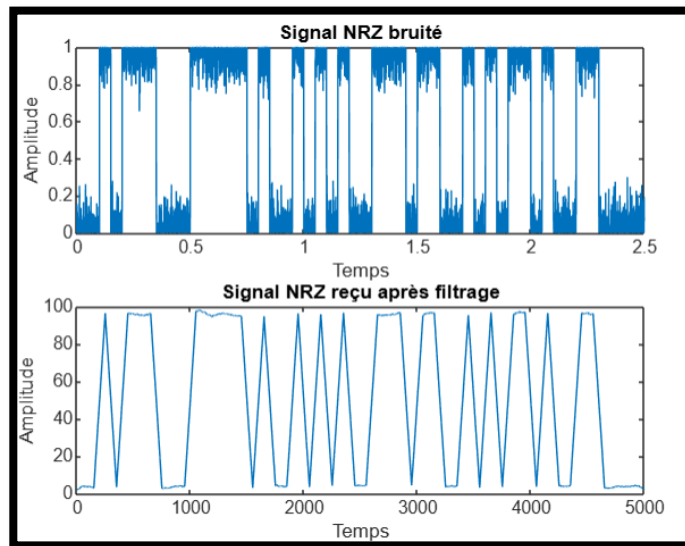
% Visualisation pour RZ
figure;
subplot(2, 1, 1);
plot(t, noisy_RZ);
title('Signal RZ bruité');
xlabel('Temps');
ylabel('Amplitude');

subplot(2, 1, 2);
plot(received_RZ);
title('Signal RZ reçu après filtrage');
xlabel('Temps');
ylabel('Amplitude');

% Visualisation pour Manchester
figure;
subplot(2, 1, 1);
plot(t, noisy_Manchester);
title('Signal Manchester bruité');
xlabel('Temps');
ylabel('Amplitude');

subplot(2, 1, 2);
plot(received_Manchester);
title('Signal Manchester reçu après filtrage');
xlabel('Temps');
ylabel('Amplitude');
```

- **Représentation**



- **Interprétation**

L'ajout du bruit AWGN **perturbe visiblement les signaux**. Pour des valeurs de SNR plus faibles, le signal est de plus en plus dégradé, ce qui **augmente les erreurs** lors de la détection des symboles.

IV. Calcul du Taux d'Erreur Binaire (BER)

- **Description**

Le taux d'erreur binaire (BER) est calculé pour chaque type de signal en fonction du SNR. Cela nous permet d'évaluer la robustesse de chaque schéma de codage face aux perturbations. Ainsi, nous comparons les trois signaux NRZ, RZ et Manchester en termes de performance de transmission en présence de bruit.

- **Code**

```
% Initialisation
SNR_range = -10:1:10;
BER_NRZ = zeros(size(SNR_range));
BER_RZ = zeros(size(SNR_range));
BER_Manchester = zeros(size(SNR_range));
for k = 1:length(SNR_range)
    noise_variance = 10^(-SNR_range(k) / 10);

    % Ajout de bruit pour chaque signal avec imnoise
    noisy_NRZ = imnoise((signal_NRZ - min(signal_NRZ)) / (max(signal_NRZ) - min(signal_NRZ)), 'gaussian', 0, noise_variance);
    noisy_NRZ = noisy_NRZ * (max(signal_NRZ) - min(signal_NRZ)) + min(signal_NRZ);

    noisy_RZ = imnoise((signal_RZ - min(signal_RZ)) / (max(signal_RZ) - min(signal_RZ)), 'gaussian', 0, noise_variance);
    noisy_RZ = noisy_RZ * (max(signal_RZ) - min(signal_RZ)) + min(signal_RZ);

    noisy_Manchester = imnoise((signal_Manchester - min(signal_Manchester)) / (max(signal_Manchester) - min(signal_Manchester)), 'gaussian', 0, noise_variance);
    noisy_Manchester = noisy_Manchester * (max(signal_Manchester) - min(signal_Manchester)) + min(signal_Manchester);

    % Filtrage et estimation
    filtered_NRZ = conv(noisy_NRZ, h, 'same');
    estimated_NRZ = filtered_NRZ(1:100:end) > 0;

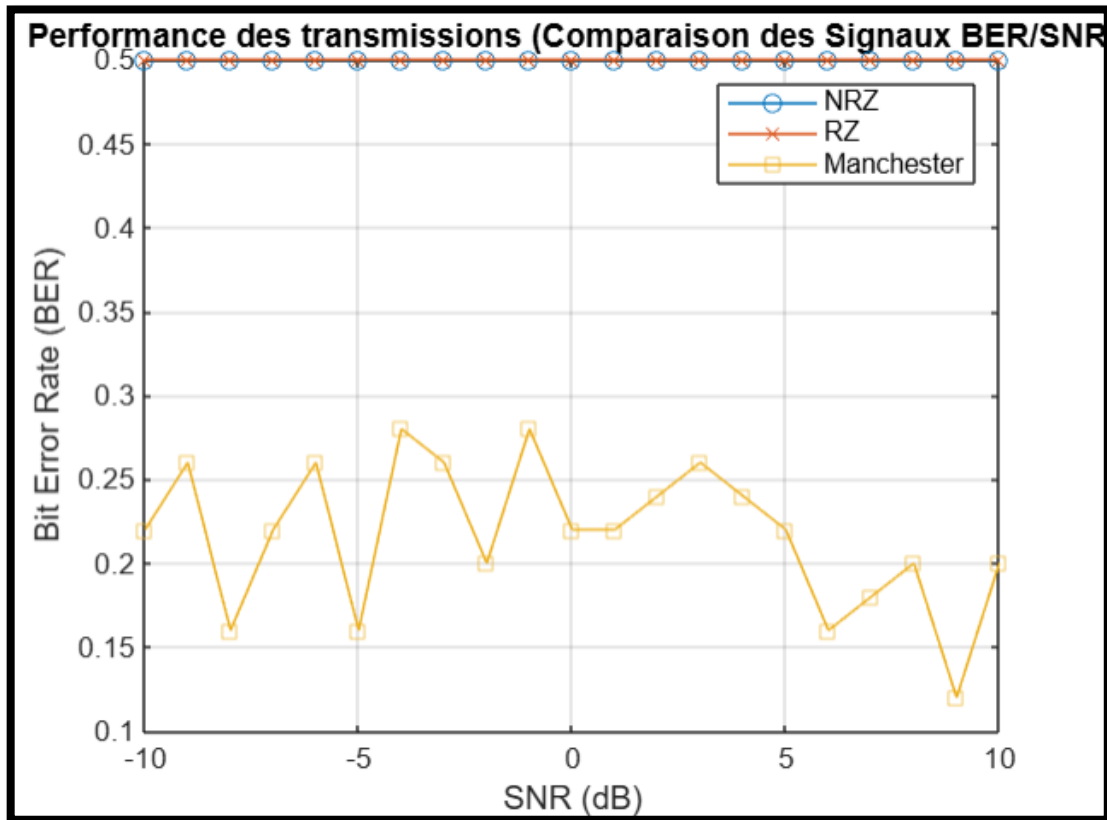
    filtered_RZ = conv(noisy_RZ, h, 'same');
    estimated_RZ = filtered_RZ(1:100:end) > 0;

    filtered_Manchester = conv(noisy_Manchester, h, 'same');
    estimated_Manchester = filtered_Manchester(1:100:end) > 0;

    % Calcul des BER
    BER_NRZ(k) = sum(estimated_NRZ ~= symbols) / Ns;
    BER_RZ(k) = sum(estimated_RZ ~= symbols) / Ns;
    BER_Manchester(k) = sum(estimated_Manchester ~= symbols) / Ns;
end

% Visualisation des BER
figure;
plot(SNR_range, BER_NRZ, '-o', 'DisplayName', 'NRZ');
hold on;
plot(SNR_range, BER_RZ, '-x', 'DisplayName', 'RZ');
plot(SNR_range, BER_Manchester, '-s', 'DisplayName', 'Manchester');
xlabel('SNR (dB)');
ylabel('Bit Error Rate (BER)');
legend;
title('Performance des transmissions (Comparaison des Signaux BER/SNR)');
grid on;
```

- **Représentation**



- **Interprétation**

- Le BER des signaux NRZ, RZ, et Manchester est comparé. Le signal Manchester présente un BER plus faible pour des faibles valeurs de SNR, montrant une meilleure robustesse au bruit grâce à sa fréquence de transition constante, ce qui facilite la synchronisation et réduit les erreurs.
- Le signal Manchester se distingue par sa meilleure performance en termes de BER, en particulier pour des valeurs de SNR faibles. Cela est dû à sa capacité à **assurer une synchronisation correcte** même en présence de bruit, contrairement aux signaux NRZ et RZ qui sont plus sensibles à des longues séquences de bits identiques.

V. Conclusion

- **Le signal Manchester s'est montré plus robuste** que les autres grâce à sa **fréquence de changement constante** qui aide à la **synchronisation**. Ces résultats illustrent l'importance du choix du schéma de modulation pour garantir la fiabilité des systèmes de communication numérique.
- Les simulations ont été très utiles pour **visualiser** l'impact du bruit **sur les différents signaux** et pour **comprendre** comment le **SNR** affecte la **qualité** de la transmission.