

Université des Sciences et de la Technologie
Houari Boumédiène

Faculté de Génie Electrique - Département de Télécommunication

Licence de Télécommunications

Matière

Communications Numériques

TP #3

Codage en ligne

Année universitaire 2024/2025

1. Objectif du TP

L'objectif du TP est d'étudier les propriétés quelques techniques de codage pour la transmission d'une séquence binaire. L'une des propriétés les plus importantes dans le choix d'un code en ligne est la largeur de la bande de fréquence du filtre de mise en forme. Dans le cadre de ce TP, plusieurs codes en ligne sont testés pour évaluer leur efficacité spectrale. Les filtres de mise en forme à étudier sont :

- NRZ unipolaire et polaire;
- RZ unipolaire et polaire;
- Manchester unipolaire et polaire,
- NRZ M-aires unipolaire et polaire.

2. Préparation théorique

Afin de préparer le schéma de chaque codage dans l'application GNU Radio Companion (GRC), une étude théorique est utile pour mener à bien le TP.

Dans ce TP, nous utilisons la même séquence binaire pour évaluer chaque technique de codage pour une bonne comparaison. Dans ce cas, la séquence binaire composée de 24 bits est définie par :

$$d = 001000010011101110111010$$

L'évaluation de chaque technique de codage repose sur la mesure de sa largeur de bande spectrale du lobe principal où la puissance maximale est concentrée. La largeur de bande spectrale d'un code en ligne est mesurée entre deux bornes négative et positive $[-B, +B]$ où B désigne la largeur de la bande positive du lobe principal. La connaissance préalable de la largeur de bande spectrale est une étape importante pour s'assurer de la bonne transmission des symboles dans un canal.

Les expérimentations sont menées en considérant les données suivantes :

- Débit binaire : $D_b = 1200\text{bps}$
 - Fréquence d'échantillonnage : $F_e = 12000\text{Hz}$
1. Calculer la période binaire. En déduire alors la période d'horloge.
 2. Calculer la période symbole pour 4-aires et 8-aires. Que peut-on constater.
 3. Calculer la durée de transmission des symboles binaires.
 4. Quel est le nombre de symboles M-aires qu'on peut transmettre pour la même durée de transmission des symboles binaires.

5. Donner la liste des symboles binaires, 4-aires et 8-aires pour un codage unipolaire puis polaire.
6. Calculer la moyenne et la variance des symboles de la séquence binaire pour un codage unipolaire puis polaire. Que peut-on dire ?
7. Pour implémenter numériquement les filtres NRZ, RZ et Manchester, on utilise le filtre à réponse impulsionnelle finie (FIR : Finite Impulse Response) avec des coefficients d'amplitude 1. Pour cela, on doit préciser le nombre de coefficients qui dépend de la durée de la période binaire en utilisant l'équation :

$$N_b = T_b F_e$$

Comme N_b définit le nombre d'échantillons durant une période binaire, alors, souvent, on l'appelle sps pour samples per symbol.

Calculer alors le nombre de coefficients (sps) du filtre FIR.

8. Pour un codage M-aires, la durée d'un symbole est liée à la durée binaire et du nombre de symboles M-aires par l'équation :

$$T_s = \log_2(M) T_b$$

Déterminer le nombre de coefficients du filtre FIR pour 4-aires puis 8-aires.

9. En déduire alors les largeurs des bandes spectrales pour chacun des filtres NRZ, RZ, Manchester et NRZ 4-aires puis 8-aires. Que peut-on dire ?

3. Implémentation du codage NRZ

Nom du schéma 🐞 : Enregistrer le schéma avec le nom de fichier TP3_1.grc.

L'implémentation du codage NRZ nécessite les blocs suivants :

- Variable : pour définir les différentes variables à utiliser dans la simulation (débit binaire, fréquence d'échantillonnage, nombre de coefficients du filtre FIR, nombre de bits à visualiser, la valeur de π).
- Vector Source : pour définir la séquence binaire.
- Signal Source : pour générer l'horloge.
- Multiply Const et Add Const : pour définir l'amplitude des symboles.

- Interpolating FIR Filter : pour définir la durée d'un symbole en fixant le nombre de coefficients du filtre.
- QT GUI Time Sink : pour l'affichage temporel de la séquence symbole et du signal physique.
- QT GUI Frequency Sink : pour l'affichage fréquentiel du signal physique.
- Virtual Sink : pour afficher une source virtuelle.
- Virtual Source : pour générer une source virtuelle.
- Throttle : pour limiter la puissance de calcul du processeur.

Pour mener à bien la simulation du codage NRZ, on se propose de l'organiser en plusieurs schémas distincts. Chaque schéma réalise une tâche spécifique.

3.1. Schéma 1 : Génération de la séquence binaire

Pour la génération de la séquence binaire, le schéma comprend trois blocs connectés selon le cheminement suivant :

Vector Source → Throttle → Virtual Sink (Type : Byte)

Appeler le ID du bloc Virtual Sink `binary_sequence` pour l'utiliser par la suite dans le bloc Virtual Source.

Visualisation temporelle :

Virtual Source (`binary_sequence`) → Char To Float → QT GUI Time Sink

Variable 1  : Utiliser un bloc Variable pour nommer `bit_rate` correspondant au débit binaire fixé à 1200 bps.

Variable 2  : Utiliser un bloc Variable pour nommer `sample_rate` correspondant à la fréquence d'échantillonnage fixée à 12000 Hz.

Variable 3  : Utiliser un bloc Variable pour nommer `k_bits` égal à 24 où `k_bits` est le nombre de bits de la séquence binaire.

Réglage 1  : Pour la visualisation de la séquence binaire, fixer le paramètre Number of Points à `k_bits` et le paramètre Sample Rate à `bit_rate`.

3.2. Schéma 2 : Génération de la séquence symbole

Pour la génération de la séquence symbole, le schéma comprend cinq blocs connectés selon le cheminement suivant :

Vector Source (binary_sequence) → Char To Float → Multiply Const → Add Const → Virtual Sink

Appeler le ID du bloc Virtual Sink symbol_sequence pour l'utiliser par la suite dans le bloc Virtual Source.

Réglage 2 ⚙️ : Fixer les constantes des blocs Multiply Const et Add Const pour générer des symboles unipolaires entre 0 et 1 puis polaires entre -1 et +1.

Visualisation temporelle ⌚ :

Virtual Source (symbol_sequence) → QT GUI Time Sink

Réglage 3 ⚙️ : Voir Réglage 1 du schéma 1.

3.3. Schéma 3 : Génération du signal physique

Pour la génération du signal physique, le schéma comprend six blocs connectés selon le cheminement suivant :

Vector Source → Char To Float → Interpolating FIR Filter → Multiply Const → Add Const → Virtual Sink

Appeler le ID du bloc Virtual Sink nrz_signal pour l'utiliser par la suite dans le bloc Virtual Source.

Variable 4 ⚙️ : Utiliser un bloc Variable pour nommer sps défini par $\text{int}(\text{sample_rate}/\text{bit_rate})$.

Variable 5 ⚙️ : Utiliser un bloc Variable pour nommer le filtre NRZ h_e défini par $\text{sps} * [1]$

Variable 6 ⚙️ : Utiliser un bloc Variable pour nommer n_points défini par $k_bits * \text{sps}$ pour l'affichage.

Variable 7  : Utiliser un bloc Variable pour nommer pi afin de fixer le paramètre Initial Phase (Radians) à 3.1415926535897932384626433832795 du bloc Signal Source pour permettre à l'horloge de se déclencher à partir de $t = 0$.

Réglage 3  : Pour le bloc Interpolating FIR Filter, fixer les paramètres suivants :

- Interpolation : sps
- Taps : h_e

Visualisation temporelle  : Comme première étape, générer une source virtuelle pour l'horloge en l'appelant clock dans le bloc Signal Source :

Signal Source (Square) → Virtual Source

La deuxième étape consiste à connecter les blocs Virtual Source de l'horloge et Virtual Source du signal physique NRZ au même bloc QT GUI Time Sink pour visualiser simultanément le signal d'horloge et le signal physique NRZ selon le cheminement suivant :

Virtual Source (clock) → QT GUI Time Sink
Virtual Source (nrz_signal) → QT GUI Time Sink

Visualisation fréquentielle  : Connecter le bloc Virtual Source nrz_signal au bloc QT GUI Frequency Sink pour visualiser le spectre du signal physique NRZ selon le cheminement suivant :

Virtual Source (nrz_signal) → QT GUI Frequency Sink

Réglage 4  : Pour la visualisation correcte du signal NRZ dans le domaine fréquentiel, sélectionner le paramètre FFT Size à 1024 et Window Type Blackman-Harris.

Question 1 : Quelle est la différence spectrale entre le codage NRZ unipolaire et le codage NRZ polaire ?

4. Implémentation du codage RZ

Nom du schéma  : Enregistrer le schéma avec le nom de fichier TP3_2.grc.

L'implémentation du codage RZ peut se faire en exploitant le schéma 3 du codage NRZ avec la modification suivante :

Vector Source → Char To Float → Interpolating FIR Filter → Multiply Const → Add Const → Multiply (Horloge × Signal NRZ) → Virtual Sink

Réglage 1 ↗ : Utiliser le bloc Multiply composé de deux entrées pour faire la multiplication du signal NRZ avec l'horloge pour générer le signal RZ.

Question 2 : Quelle est la différence spectrale entre le codage RZ unipolaire et le codage RZ polaire ?

Question 3 : Comparer les spectres des signaux NRZ et RZ. Que peut-on dire ?

5. Implémentation du codage Manchester

Nom du schéma ✂ : Enregistrer le schéma avec le nom de fichier TP3_3.grc.

L'implémentation du codage Manchester peut se faire en exploitant le schéma 3 du codage NRZ avec la modification suivante :

Vector Source → Char To Float → Interpolating FIR Filter → Float To Char → XOR (NRZ ⊕ Horloge) → Char To Float → Multiply Const → Add Const → Virtual Sink

Réglage ↗ : Utiliser le bloc Xor composé de deux entrées pour faire l'opération XOR entre le signal NRZ et l'horloge pour générer le signal Manchester.

Question 4 : Expliquer comment peut-on générer le signal Manchester à partir de la fonction XOR du signal NRZ et de l'horloge.

Question 5 : Quelle est la différence spectrale entre le codage Manchester unipolaire et le codage Manchester polaire ?

Question 6 : Comparer les spectres des signaux NRZ, RZ et Manchester. Que peut-on dire ?

6. Implémentation du codage NRZ M-aires

Nom du schéma  : Enregistrer le schéma avec le nom de fichier TP3_4.grc.

L'implémentation du codage NRZ M-aire peut se faire en exploitant les schémas 2 et 3 pour la génération de la séquence symbole et du signal NRZ M-aires. L'idée est d'ajouter un bloc pour grouper m bits. Pour cela, on exploite le bloc Pack K Bits pour fixer le nombre de bits à grouper.

Les schémas 2 et 3 sont alors modifiés pour la séquence symbole :

Vector Source (binary_sequence) → Pack K Bits → Char To Float → Multiply Const → Add Const → Virtual Sink

et la génération du signal NRZ M-aires :

Vector Source → Pack K Bits → Char To Float → Interpolating FIR Filter → Multiply Const → Add Const → Virtual Sink

En exploitant les variables du schéma NRZ, ajouter les variables suivantes :

Variable 1  : Utiliser un bloc Variable pour nommer la variable m_bits pour fixer le nombre de bits à grouper.

Variable 2  : Utiliser un bloc Variable pour nommer la variable $k_symbols$ pour fixer le nombre de symboles composant la séquence binaire en utilisant l'instruction (k_bits/m_bits) .

Variable 3  : Utiliser un bloc Variable pour nommer la variable $symbol_rate$ pour fixer le débit symbole en utilisant l'instruction $(sample_rate/m_bits)$.

Variable 4  : Utiliser un bloc Variable pour nommer la variable $h_samples$ pour fixer le nombre d'échantillons du filtre NRZ M-aires en utilisant l'instruction $(m_bits*sps)$.

Variable 5  : Utiliser un bloc Variable pour nommer la variable h_e pour fixer le nombre d'échantillons du filtre NRZ en utilisant l'instruction $h_samples * [1]$.

Réglage 1 ↗ : Pour le bloc Interpolating FIR Filter, fixer les paramètres suivants :

- Interpolation : h_{samples}
- Taps : h_e

où h_e est le filtre émetteur défini dans le bloc Variable h_e .

Réglage 2 ↗ : Pour la visualisation temporelle de la séquence symbole, le paramètre Number of Points est fixé à k_{symbols} et le paramètre Sample Rate est fixé à symbol_rate .

Réglage 3 ↗ : Pour la visualisation temporelle du signal NRZ M-aire, le paramètre Number of Points est fixé à $k_{\text{symbols}} * h_{\text{samples}}$ et le paramètre Sample Rate est fixé $\text{symbol_rate} * h_{\text{samples}}$.

Question 7 : Quel est le nombre de symboles affichés pour la séquence $m=1$ ($M=2$), $m=2$ ($M=4$) et $m=3$ ($M=8$) ? Que peut-on constater ?

Question 8 : Comparer les spectres des signaux NRZ pour $m=1$ ($M=2$), $m=2$ ($M=4$) et $m=3$ ($M=8$). Que peut-on dire ?