Travaux pratiques pour modéliser un système de communication

P. Favrat

11 mars 2024

1 Objectif

L'objectif de ce travail pratique est d'illustrer une chaine de communication sans-fils avec calcul du BER (Bit Error Rate). Le travail s'effectuera sous forme d'un notebook Python. Les librairies Python nécessaires sont : numpy, matplotlib et commpy.
Pour installer la librairie commpy : pip install scikit-commpy

2 TP Modulateur

On commencera par réaliser un modulateur simple à 2 ou 4 niveaux suivi d'un filtre de pulse shaping.

1. Générer une suite de bits aléatoires d'une longueur de N bits.
   Python : numpy.random.randint(2, size=N*np.log2(M)) (M est le nombre d'état).
2. Programmer une fonction qui converti des bits d'information en 2 ou 4 niveaux avec la signalisation NRZ (Non Return to Zero).
3. Programmer ou utiliser une fonction qui upsample un flux de donnée par un facteur sps (samples per symbol). Le plus simple est de rajouter des zéros entre les bits d'information.
4. Créer un filtre FIR (Finite Impulse Response) de type root raised cosine (RRC).
   Python : utiliser la fonction rrcosfilter de la librairie commpy.filters .
5. Afficher la réponse impulsionnelle du filtre RRC (Rappel les coefficients d'un filtre FIR sont les points de la réponse impulsionnelle).
6. Convoluer (filtrer) la suite de bits avec le filtre RRC et afficher le diagramme en oeil. Python : fonction np.convolve

3 TP Mesure du BER

Pour compléter le systéme nous devons ajouter un canal de type AWGN (Additive White Gaussian Noise) et simuler un récepteur. Finalement on pourra mesurer le BER (Bit Error Rate) en variant le rapport signal sur bruit.

1. Générer un filtre FIR qui réalise la fonction RRC (Root Raised Cosine) comme sous §2.4 mais pour le récepteur. Python : utiliser la fonction rrcosfilter de la librairie commpy.filters .
2. Afficher la réponse impulsionnelle du filtre RRC (Rappel les coefficients d'un filtre FIR sont les points de la réponse impulsionnelle) voir §2.5 .
3. Ajouter un modèle de canal AWGN entre les filtres RRC (comme sur la fig. 1). Python : fonction awgn de commpy.channels .

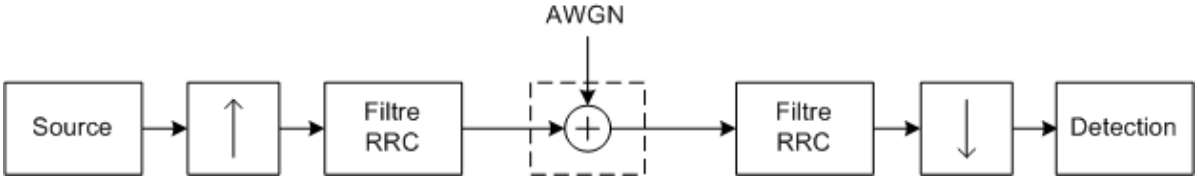

FIGURE 1 – Set-up de test avec filtres de pulse shaping

4. Placer le récepteur à la suite et calculer le BER. Le récepteur est simplement le filtre RRC suivi d'un échantillonnage au bon moment avec un seuil.
5. Utiliser une boucle pour faire varier le rapport signal sur bruit élémentaire E_b/N_0 de 0 à 10dB par pas de 0.1dB et plotter le diagramme BER vs. E_b/N_0 pour M=2 et M=4 selon le format de la figure 2.

4 TP Calcul du BER théorique

En connaissant la statistique du bruit, il est possible d'afficher directement la courbe du BER avec la formule suivante pour les modulations M-aire ASK (Proakis 1983) :

$$BER_{ASK} = \frac{1}{k}(\frac{M-1}{M}) \cdot erfc\sqrt{\frac{3}{(M^2-1)} \cdot k\frac{E_b}{N_0}} \quad (1)$$

La fonction erfc (error function) calcule la surface des queues d'une Gaussienne à partir d'un seuil qui représente l'écart maximum autorisé de l'amplitude pour un symbole. Les erreurs se trouvent donc au-delà de ce seuil.

1. Afficher les courbes pour M=2 et M=4, (k = log2(M)) selon le format de la figure 2.
2. Superposer les courbes de BER théoriques au courbes de BER mesurées au point 3.5. Que constatez-vous ?

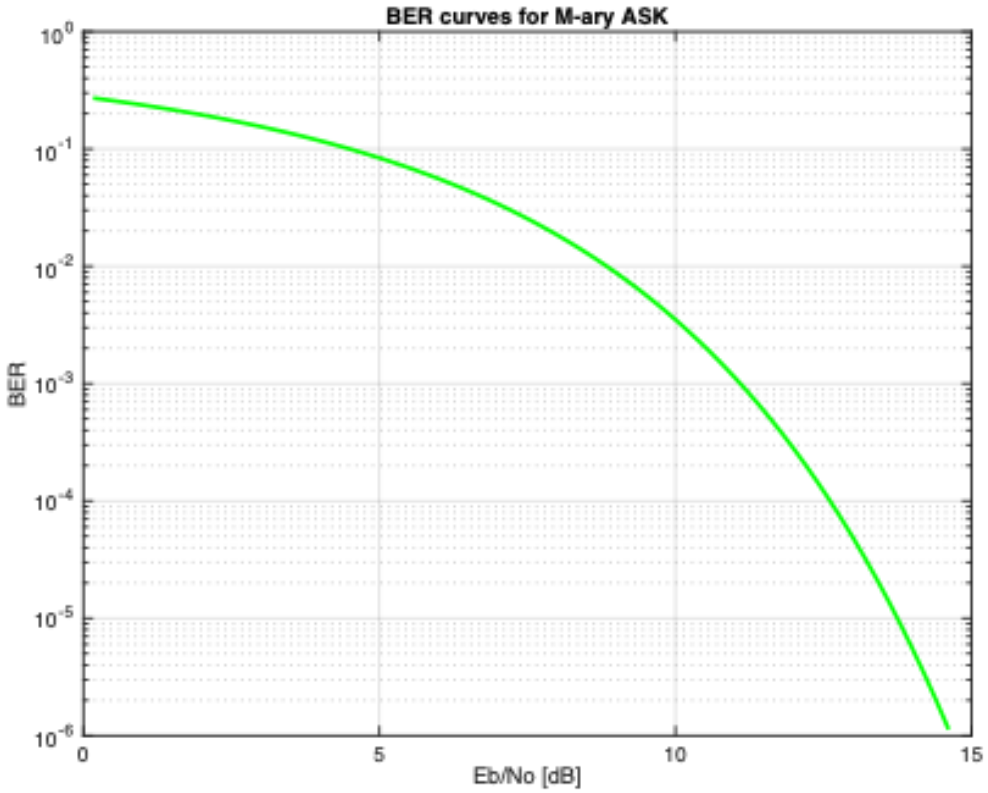

FIGURE 2 – Exemple de graphique pour le BER