TP 1

Outils pour la simulation de chaînes de communications numériques

1 Objectif

Le but de ce TP est de se familiariser avec quelques notions essentielles de communications numériques par le biais de la simulation sous Matlab.

2 Introduction

Nous souhaitons transmettre sur un canal de type BBAG (Bruit Blanc Additif Gaussien) une suite de symboles $\{a_k\}$ d'information à la cadence $1/T_b$ comme le montre la Figure 1.

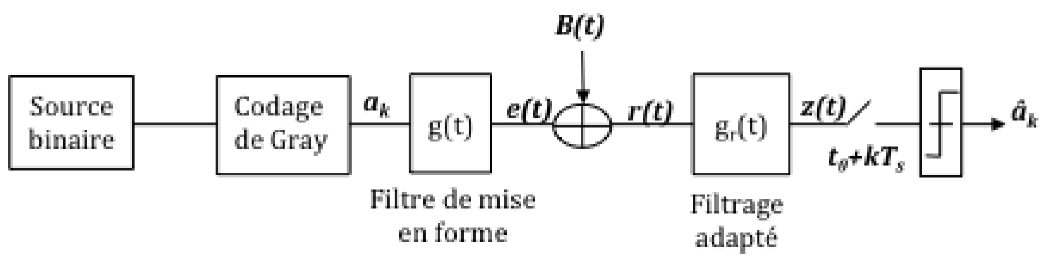

FIGURE 1 – Chaîne de transmission numérique utilisant une modulation MDA

Avant d'être émise sur le canal physique, cette suite est mise en forme par le biais d'un filtre linéaire analogique de mise en forme $g(t)$. Ainsi, le signal analogique émis $e(t)$ est le suivant :

$$e(t) = \sum_k a_k g\left(t - kT_b\right). \tag{1}$$

Le signal reçu, $r(t)$, après le passage du signal émis $e(t)$ dans le canal est donné par :

$$r(t) = e(t) + B(t), \tag{2}$$

où $B(t)$ est le BBAG.

Ce TP a pour objectif d'étudier chaque bloc constituant la chaîne de transmission numérique représentée dans la Figure 1.

3 Travail demandé

Partie I

1. Générer à l'aide de la fonction Matlab "randint" N échantillons d'une source binaire où N est le nombre de bits à émettre. Les N échantillons seront stockés dans un vecteur ligne.
2. Effectuer le codage de Gray de la source binaire en symboles a_k dans l'alphabet M-aire $\{\pm A, \pm 3A, \cdots, \pm(M-1)A\}$ où A est un réel strictement positif permettant de contrôler l'énergie moyenne transmise par bit. Dans la suite, nous nous limitons au cas binaire : $M = 2$.

   On doit convertir le signal en temps discret composé des symboles a_k en un signal analogique :

   $$a(t) = \sum_k a_k \delta(t - kT_b) \tag{3}$$

   où $\delta(t)$ est l'impulsion de Dirac et T_b correspond à la durée d'un bit.

   Ensuite, nous proposons d'échantillonner ce signal à la fréquence d'échantillonnage égale à F/T_b où F est le facteur de sur-échantillonnage que l'on fixera à 8. Echantillonner le signal $a(t)$ à $f_e = F/T_b$ revient à insérer $F - 1$ zéros entre deux symboles consécutifs.
3. Représenter le signal discret ainsi obtenu (utiliser la fonction "scatter" de Matlab).

Partie II

On désire simuler un filtre dit filtre de mise en forme en racine de cosinus surélevé, couramment utilisé dans les chaînes de communications numériques. Sa réponse implusionnelle a pour expression :

$$g(t) = \frac{1}{\sqrt{T_b}} \times \frac{\sin\left(\frac{\pi t}{T_b}(1-\alpha)\right) + \cos\left(\frac{\pi t}{T_b}(1+\alpha)\right)}{\frac{\pi t}{T_b}\left[1 - \left(\frac{4\alpha t}{T_b}\right)^2\right]} \tag{4}$$

où α est dit coefficient de retombé (roll-off) compris entre 0 et 1. On fixera $T_b = 1$. Il est à noter que $g(t)$ est de durée infinie et présente trois formes indéterminées en $t = 0$ et $t = \pm T_b/4\alpha$.

1. Calculer les limites correspondantes par développement limité ou en utilisant la fonction "limit" de Matlab. Etudier cette fonction pour l'utiliser ultérieurement.
2. Pour les valeurs de $\alpha = 0,1$, $0,25$ et $0,9$, stocker dans un vecteur ligne les échantillons de la troncature de $g(t)$ sur $[-KT_b, KT_b]$ où $K = 8$.
3. Tracer sur la même figure l'évolution des échantillons du filtre $g(t)$ en fonction du temps pour les différentes valeur de α (fonction "subplot" de Matlab).
4. Vérifier que l'énergie du filtre échantillonné $g(t)$ vaut F.

5. Générer une version échantillonnée du signal émis, $e(t)$, en effectuant la convolution discrète des versions échantillonnées de $a(t)$ et $g(t)$. On utilisera la commande Matlab "conv".

Partie III

1. Etudier la fonction "randn" de Matlab. Déduire comment on peut générer un bruit Gaussien de moyenne nulle et de variance σ^2 quelconque.
2. Utiliser cette fonction pour générer des échantillons d'un bruit B_k discret, Gaussien et de variance $N_0/2$. La valeur de N_0 sera déduite de la valeur en décibel de E_b/N_0 qu'on désire simuler.

Partie IV

On fixera $T_s = T_b = 1$.

1. Déterminer le filtre adapté.
2. Tracer le diagramme de l'œil pour $\alpha = 0,1$, $0,25$ et $0,9$. On rappelle que le diagramme de l'œil s'obtient en superposant toutes les traces de la sortie du filtre adapté en absence de bruit. Interpréter l'ouverture horizontale du diagramme de l'œil en fonction de la valeur du coefficient de retombée.
3. Etudier l'effet de l'interférence entre symboles à travers la fermeture verticale du diagramme de l'œil pour une troncature très sévère du filtre de mise en forme $g(t)$. On rappelle qu'en pratique le critère de Nyquist n'est vérifié qu'approximativement pour des grandes valeurs de K. Prendre par exemple $K = 2$ et $K = 4$ pour $\alpha = 0,25$.
4. Générer le signal filtré décimé d'un facteur F servant à la prise de décision. La phase d'échantillonnage t_0 doit correspondre à l'ouverture maximale du diagramme de l'œil.

   Le filtre $g(t)$ étant non causal, on a dû le retarder lors de l'implémentation sous Matlab. Ce qui fait que la sortie du filtre de mise en forme est retardée de la moitié de la taille du filtre $g(t)$. De même pour le filtre adapté. Ainsi la phase d'échantillonnage t_0 vaut la taille du filtre $g(t)$ en nombre d'échantillons. Autrement dit, la variable de décision permettant d'estimer le premier symbole émis se trouve à l'indice taille du filtre $g(t)$ en nombre d'échantillons. Pour récupérer le symbole suivant, il suffit d'avancer d'une période symbole, c'est-à-dire de F échantillons.
5. Vérifier qu'en absence de bruit, la variable de décision correspondante au symbole a_n vaut :

   $$z\left(t_0 + nT_s\right) = a_n \sum_k g\left(kT_e\right)^2 = a_n F \tag{5}$$
6. Générer des échantillons du signal reçu bruité : $z(t) = e * g_r(t) + b(t)$.