

TP4 de communications numériques

1 Objectifs et évaluation

Dans ce TP nous allons étudier le multiplexage fréquentiel de plusieurs émetteurs. L'émetteur 1 sera notre émetteur de référence. Les autres émetteurs seront considérés comme des interférences.

Dans cette séance, vous écrirez un script interactif de Matlab (notebook) permettant

1. de mettre en œuvre la chaîne de communications numériques sur fréquence porteuse
2. d'étudier l'impact des autres émetteurs sur la réception de l'émetteur de référence taux d'erreur binaire

Vos codes et votre script interactif sont à rendre en fin de séance sur l'interface Thor <https://thor.enseirb-matmeca.fr/ruby/>.

Pour ce troisième TP, les fonctions du premier et du deuxième TP vous sont fournies en .p. Vous pouvez faire appel à ces fonctions si jamais vous n'avez pas réussi à développer toutes les fonctions des premiers TP.

2 Simulation sur fréquence porteuse

2.1 Implémentation de la chaîne de transmission

Nous allons étudier la mise sur fréquence porteuse d'un signal numérique. Pour cela, nous allons moduler un signal en bande de base $s_l(t)$ sur une porteuse de fréquence F_p . Cette modulation est réalisée par la relation suivante :

$$s(t) = \sqrt{2} \operatorname{Re} (s_l(t) e^{j2\pi F_p t}) \quad (1)$$

où $s_l(t)$ est le signal en bande de base et F_p la fréquence porteuse.

Dans votre code : écrire la fonction `tx` qui effectue les opérations suivantes :

1. génération de N_s bits aléatoires,
2. mapping des bits sur les symboles de la constellation,
3. calcul de s_l , signal en bande de base,
4. modulation de s_l sur la fréquence porteuse F_p .

Cette fonction aura le prototype suivant

Listing 1 – Prototype de la fonction `tx`

```
1 %% tx.m
2 % Fonction qui génère le signal émis
3 %
4 %% Inputs :
5 %   Ns           : (int) nombre de symboles
6 %   constellation : (array) tableau de la constellation
7 %   Fse          : (int) facteur de sur-échantillonnage
8 %   fp           : (int) fréquence porteuse (réduite)
9 %   g            : (array) filtre de mise en forme
10 %
11 %% Outputs :
12 %   sp          : (array) signal émis
13 %   sl          : (array) signal émis en bande de base
14 %   s           : (array) symbole émis
15 %   u           : (array) bits émis
16
17 function [sp, sl, s, u] = tx(Ns, constellation, Fse, fp, g)
```

Dans votre notebook : synthétiser un signal numérique sur fréquence porteuse et afficher sa DSP. Ce signal aura les caractéristiques suivantes :

- $N_s = 10000$ symboles,
- $F_e = 20$ MHz,
- $T_s = 1\mu s$
- $F_p = 1'$ MHz,
- constellation : $M = 4$ (QPSK)
- filtre de mise en forme : `rcosdesign(0.5,20,Fse,"sqrt")`

Pour le calcul du spectre, vous utiliserez un périodogramme moyenné sans fenêtrage ni chevauchement. Vous comparerez le périodogramme à DSP théorique du signal modulé $s(t)$.

2.2 Implémentation de la chaîne de réception

Maintenant que nous avons un signal modulé sur fréquence porteuse, nous allons étudier la chaîne de réception. Théoriquement, la chaîne de réception commence par :

$$\tilde{y}(t) = \sqrt{2}y(t)e^{-j2\pi F_p t} \quad (2)$$

où $y(t)$ est le signal reçu et $\tilde{y}(t)$ un signal intermédiaire. L'enveloppe complexe de $y(t)$ est obtenue en filtrant $\tilde{y}(t)$ par un **filtre passe bas**. Dans notre cas, **le filtre adapté fera office de filtre passe-bas**.

Dans votre code : écrire la fonction `rx` qui effectue les opérations suivantes :

1. Calcul du signal intermédiaire $\tilde{y}(t)$,
2. filtrage du signal intermédiaire par le filtre adapté $g_a(t)$,
3. échantillonnage au temps symbole,
4. décision et démapping des symboles sur les bits.

Cette fonction aura le prototype suivant

Listing 2 – Prototype de la fonction `rx`

```
1 %% rx.m
2 % Fonction démodule le signal reçu
3 %
4 %% Inputs :
5 %   y           : (array) signal reçu
6 %   Ns          : (int) nombre de symboles
7 %   constellation : (array) tableau de la constellation
8 %   Fse         : (int) facteur de sur-échantillonnage
9 %   fp          : (int) fréquence porteuse (réduite)
10 %   g           : (array) filtre de mise en forme
11 %
12 %% Outputs :
13 %   rn : (array) symboles après filtrage adapté et
14 %       décimation
15 %   uh : (array) bits reçus
16 function [rn, uh] = rx(y, Ns, constellation, Fse, fp, g)
```

Dans votre notebook : appeler votre chaîne de réception pour démoduler le signal généré dans la partie précédente et afficher la constellation de r_n (signal après filtrage adapté et décimation). Commenter la constellation obtenue et la comparer à la constellation utilisée à l'émission.

2.3 Calcul du taux d'erreur binaire

Nous allons maintenant calculer le taux d'erreur binaire (TEB) de la chaîne de réception.

Dans votre code : écrire la fonction `compute_TEB` qui effectue les opérations suivantes :

1. appel de la fonction `tx` pour générer un signal numérique sur fréquence porteuse,
2. appel de la fonction `rx` pour démoduler le signal reçu,
3. calcul du TEB.
4. calcul de la probabilité d'erreur théorique P_b .

Cette fonction aura le prototype suivant

Listing 3 – Prototype de la fonction compute_TEB_FP

```

1 %% compute_TEB_FP.m
2 % Fonction qui calcule le Taux d'Erreur Binaire (TEB) pour
  une
3 % constellation donnée et un filtre de mise en forme donné
4 %
5 %% Inputs :
6 % eb_n0_dB      : (array) ensemble des valeurs de Eb/N0 à
  tester en dB
7 % Ns           : (int) Nombre de symboles à envoyer dans
  un paquet
8 % constellation : (array) constellation des symboles
9 % Fse          : (int) facteur de sur-échantillonnage
10 % fp          : (float) fréquence porteuse (réduite)
11 % g           : (array) filtre de mise en forme
12 % max_bit_err  : (int) Nombre d'erreurs binaires à
  attendre avant de passer au point de Eb/N0 suivant
13 %
14 %% Output :
15 % TEB : (array) Une valeur de Taux d'Erreur Binaire (TEB)
  par point de Eb/N0
16 % Pb  : (array) Une valeur de probabilité d'erreur par
  point de Eb/N0
17 function [TEB, Pb] = compute_TEB_FP(eb_n0_dB, constellation
  , g, Fse, Ns, max_bit_err)

```

Dans votre notebook : tracer le TEB en fonction de $\frac{E_b}{N_0}$ pour la modulation QPSK.

3 Multiplexage fréquentiel

Nous allons maintenant étudier l'impact de la présence d'autres émetteurs sur le TEB de l'émetteur de référence. Pour cela, nous allons considérer un signal de référence $s(t)$ et 1 un signal parasite $s'(t)$. Pour le canal, le signal reçu sera modélisé par la relation suivante :

$$y(t) = s(t) + s'(t) \quad (3)$$

Nous allons considérer que $s'(t)$ est un signal modulé sur la porteuse décalée de Δ_F par rapport à la porteuse de référence

$$F'_p = F_p + \Delta_F. \quad (4)$$

Dans votre notebook :

1. tracer et commenter les courbes suivantes :
 - sur un même graphique, pour $\Delta_F = 5$ MHz :
 - la DSP du signal de référence $s(t)$,
 - la DSP du signal interférent $s'(t)$,
 - la DSP du signal reçu $y(t)$.
 - La constellation des échantillons reçus r_n .

2. En illustrant votre démarche par des graphiques, trouver la valeur minimale de Δ_F qui garantit que l'interférence entre les deux émetteurs est négligeable.
3. Tracer la puissance de l'interférence sur le signal r_n de référence en fonction de Δ_F .
4. Tracer le TEB de l'émetteur de référence en fonction de Δ_F pour $Eb/N_0 = 10$ dB.
5. En illustrant votre démarche par des graphiques, trouver le nombre maximal d'émetteurs qui peuvent communiquer simultanément sans interférer entre eux. Attention, la fréquence 0 Hz ne peut être utilisée par aucun émetteur.
6. Reprendre les 3 derniers points pour un filtre de mise en forme porte (`g = ones(1,Fse)`).

4 Contacts

- Alexandre Valade - alexandre.valade@bordeaux-inp.fr
- Romain Tajan - romain.tajan@bordeaux-inp.fr