TP noté de communications numériques

R. Tajan (romain.tajan@bordeaux-inp.fr)

20 mai 2026

1 Objectifs et évaluation

Dans ce TP nous allons étudier le multiplexage fréquentiel de plusieurs émetteurs.
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 le taux d'erreur symbole à la réception de l'émetteur de référence

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

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 :

$$s(t) = \sqrt{2} Re\left(s_l(t) e^{j2\pi F_p t}\right) \tag{1}$$

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

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

1. génération de N symboles aléatoires,
2. calcul de s_l, signal en bande de base,
3. modulation de s_l sur la fréquence porteuse F_p en utilisant l'équation 1.

Cette fonction aura le prototype donné dans le Listing 1.

Listing 1 – Prototype de la fonction tx

```
%% tx.m
% Fonction qui génère le signal émis
%
%% Inputs :
%   Ns   : (int) nombre de symboles
%   cstl : (array) tableau de la constellation
%   F    : (int) facteur de sur-échantillonnage
%   Fp   : (int) fréquence porteuse (réduite = divisée par
   Fe)
%   h    : (array) filtre de mise en forme
%
%% Outputs :
%   bits    : (array) bits émis (entiers entre 0 et M-1)
%   S       : (array) Symboles émis
%   sl      : (array) signal émis en bande de base
%   s       : (array) symbole émis sur fréquence porteuse

function [bits, S, sl, s] = tx(Ns, cstl, F, Fp, h)
```

Vous utiliserez les paramètres suivants :

- $N = 10000$ symboles,
- $Fe = 20$ MHz,
- $F = 20$ (facteur de sur-échantillonnage),
- $F_p = 4$ MHz,
- Constellation : 4-PAM,
- filtre de mise en forme en racine de cosinus surélevé re roll-off 0.5 obtenu par la commande : `h = rcosdesign(0.5,30,F,"sqrt");`
- $P = 512$ taille de la fenêtre pour le périodogramme,
- $Nfft = 2048$ nombre de points pour le périodogramme.

Dans votre notebook - Synthétiser un signal numérique sur fréquence porteuse et afficher sa DSP. Pour afficher la DSP, vous pouvez utiliser un outil appelé périodogramme de Welch. Sous Matlab, vous utiliserez la fonction `pwelch` :

```
[Gamma_s, f] = pwelch(s, ones(1, P), 0, Nfft, Fe, 'centered')
```

L'affichage de la DSP se fera en échelle logarithmique (dB) sur les ordonnées.

Dans votre notebook - Commenter la DSP obtenue et identifier les différentes composantes du signal (bande passante, fréquence porteuse). En particulier, vous mesurerez la bande passante du signal modulé sur fréquence porteuse et la comparerez au débit symbole.

2.2 Implémentation de la chaîne de réception sans bruit

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

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

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 $h_a(t)$,
3. échantillonnage au temps symbole,
4. décision.

Cette fonction aura le prototype donné dans le Listing 2.

Listing 2 – Prototype de la fonction rx

```
%% rx.m
% Fonction qui démodule le signal reçu
%
%% Inputs :
%   y    : (array) signal reçu
%   N    : (int) nombre de symboles a récupérer
%   ctl  : (array) tableau de la constellation
%   F    : (int) facteur de sur-échantillonnage
%   Fp   : (int) fréquence porteuse (réduite)
%   h    : (array) filtre de mise en forme
%
%% Outputs :
%   rn          : (array) symboles apès filtrage adapté et
   sous-échantillonnage
%   bits_recus : (array) entiers représentant lesbits reçus

function [rn, bits_recus] = rx(y, N, ctl, F, Fp, h)
```

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.

Dans votre notebook : Calculer la variance du bruit présent sur r_n et la comparer à la variance du bruit ajouté dans le canal.

Dans votre notebook : Calculer le taux d'erreur symbole (TES) de votre chaîne de transmission sur fréquence porteuse. Commenter le TES obtenu.

2.3 Calcul du taux d'erreur symbole en présence de bruit

Nous allons maintenant calculer le TES en présence de bruit. Pour cela, ajouter un bruit blanc gaussien à votre signal modulé sur fréquence porteuse $s(t)$. Le signal reçu sera alors modélisé par la relation suivante :

$$y(t) = s(t) + z(t) \tag{3}$$

où $z(t)$ est un signal de bruit blanc gaussien centré de variance $\sigma^2 = 0.1$.

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.

Dans votre notebook : Vous calculerez également le taux d'erreur symbole (TES) de votre chaîne de transmission sur fréquence porteuse. Commenter le TES obtenu en le comparant avec la formule théorique (sur fréquence porteuse) suivante :

$$P_e = 2\frac{M-1}{M} Q\left(\frac{g_0}{\sigma}\right) \tag{4}$$

où $Q(x)$ est calculée à l'aide de la fonction `qfunc` de Matlab.

3 Multiplexage fréquentiel

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

$$y(t) = s(t) + s'(t) + z(t) \tag{5}$$

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

$$F'_p = F_p + \Delta_F. \tag{6}$$

Dans votre notebook - Commenter la DSP de $y(t)$ obtenue et identifier les différentes composantes suivantes :

— la DSP du signal de référence $s(t)$,
— la DSP du signal interférent $s'(t)$,
— la DSP du bruit $z(t)$.

Dans votre notebook - Commenter la constellation des échantillons reçus r_n pour le signal de référence.

Dans votre notebook - Calculer le TES de votre chaîne et le commenter (toujours au regard de la formule théorique).

Dans votre notebook - Tracer la courbe du TES en fonction de Δ_F, pour Δ_F variant de 0 à 5 MHz par pas de 5 kHz. Commenter la courbe obtenue et identifier des zones où l'interférence est forte, l'interférence est faible, l'interférence est nulle.

4 Bonus

Généraliser votre code pour simuler la présence de U émetteurs parasites. Puis trouvez le plus grand nombre d'émetteurs parasites U que votre chaîne de réception peut supporter tel que chaque émetteur puisse être détecté sans dégradation du TES. Vous devez en particulier garantir que le TES de chaque émetteur est dans l'intervalle $P_e \pm 10\%$.

Vous illustrerez votre raisonnement dans le notebook.

La personne qui aura trouvé le plus grand nombre d'émetteurs parasites U avec un raisonnement solide gagnera un point bonus au partiel ! En cas d'égalité le point de bonus sera partagé entre les gagnants.

5 Contacts

— Romain Tajan - romain.tajan@bordeaux-inp.fr