

# TP de communications numériques

## module TS227 – année 2020/2021

D. Dallet, G. Ferré et R. Tajan

### 1 Objectifs et évaluation

L'objectif des TP de communications numériques est de simuler à l'aide du logiciel Matlab plusieurs chaînes de communications numériques (en bande de base et sur fréquence porteuse) pour une modulation QPSK afin d'évaluer notamment leurs performances en termes de probabilité d'erreur binaire.

Concernant le rapport, il ne doit pas excéder **8 pages**, vous devez fournir un document scientifique et technique, qui doit présenter votre travail, vos choix techniques et dans lequel tous les résultats obtenus doivent être interprétés et commentés. Les codes Matlab doivent également être transmis à votre enseignant. Ils doivent pouvoir être compris rapidement. Cela passe par l'utilisation de commentaires. Les commentaires doivent permettre de répondre au moins à la question : que fait la ligne de code ? Une attention particulière doit être portée à la lisibilité du programme. Les rapports doivent être au format **pdf** et les codes au format **zip** comme imposé par la plateforme Thor. Ces deux fichiers doivent être transmis à votre encadrant par l'intermédiaire de Thor.

### 2 (R)appel concernant Matlab

Nous détaillons ici quelques fonction qui pourraient vous servir pour ce TP. **Il n'est ni obligatoire ni nécessaire de les utiliser.**

- La fonction "doc" permet d'obtenir une documentation sur une fonction particulière (**cette fonction est maintenant votre meilleure amie**)
- Pour un vecteur X, "X(i)" représente le ième élément de X. Les indices Matlab commencent à 1.
- Pour une matrice X, "X(i,j)" représente l'élément à la ième ligne et jème colonne de X. Les indices Matlab commencent à 1.
- Pour trois réels A,B et C, "A:B:C" permet de créer le vecteur suivant

$$\left[ A, A + B, A + 2B, \dots, A + \left\lfloor \frac{C - A}{B} \right\rfloor B \right]$$

- Pour trois entiers A,B et C, et un vecteur X "X(A:B:C)" permet de créer le vecteur suivant

$$\left[ X(A), X(A + B), X(A + 2B), \dots, X\left(A + \left\lfloor \frac{C - A}{B} \right\rfloor B\right) \right]$$

- Pour deux nombres A et B, "A>B" est un booléen valant "true" si A>B et "false" sinon.

- Pour deux matrices de même taille  $X$  et  $Z$ , " $Y = X > Z$ " renvoie une matrice de la même taille que  $X$  où  $Y(i,j) = X(i,j) > Z(i,j)$ . Ceci fonctionne aussi pour les opérations  $<$ ,  $==$ , ou encore  $\sim$ .
- La structure conditionnelle de Matlab s'écrit de la façon suivante :
 

```
if condition
    % Faire quelque chose si la condition est vraie
elseif condition2
    % Faire quelque chose si la condition est fausse et la condition 2 est vraie
else
    % Faire autre chose sinon
end
```
- Une boucle for s'écrit sous Matlab :
 

```
for i = A:B:C
    % i parcourt le vecteur A:B:C
    % Faire quelque chose pour la valeur courante de i
end
```
- La fonction `randi([i_min, i_max], n, m)` vous permet de générer aléatoirement une matrice de taille  $m \times n$  d'entiers compris dans l'intervalle  $[i_{min}, i_{max}]$ . Les composants de cette matrices sont indépendants et uniformément distribués,
- La fonction `upsample` vous permet de sur-échantillonner un signal discret,
- La fonction `downsample` vous permet de sous-échantillonner un signal discret (**pour ce TP, privilégier la forme `X(A:B:C)`**),
- La fonction `randn(m,n)` vous permet de générer aléatoirement une matrice de taille  $m \times n$  d'échantillons gaussiens iid de moyennes nulles et de variances 1,
- La fonction `scatterplot` vous permet d'afficher un diagramme de constellation,
- Les fonctions `pskmod`, `pskdemod` vous permettent respectivement de générer et de démoduler des symboles M-PSK,
- Les fonctions `pammod` `pamdemod` vous permettent respectivement de générer et de démoduler des symboles M-PAM (M-ASK),
- Les fonctions `qammod` `qamdemod` vous permettent respectivement de générer et de démoduler des symboles M-QAM,
- La fonction `rcosfir` vous permet de générer la réponse impulsionnelle d'un filtre en racine de cosinus sur-élevé.

### 3 Simulation d'une modulation QPSK en bande de base (main\_QPSK.m)

Dans cette partie vous allez vous intéresser au cas des communications numériques en bande de base émettant des symboles QPSK (Quadrature Phase Shift Keying). L'architecture bande de base à considérer est présentée sur la figure 1.

A travers la simulation Matlab, vous vérifierez entre autre une partie de vos résultats théoriques de TD et évalueriez les performances de la chaîne de communications numériques en terme de probabilité d'erreur binaire.

#### 3.1 Paramètres de simulation :

Les paramètres à considérer dans votre simulation sont les suivants :

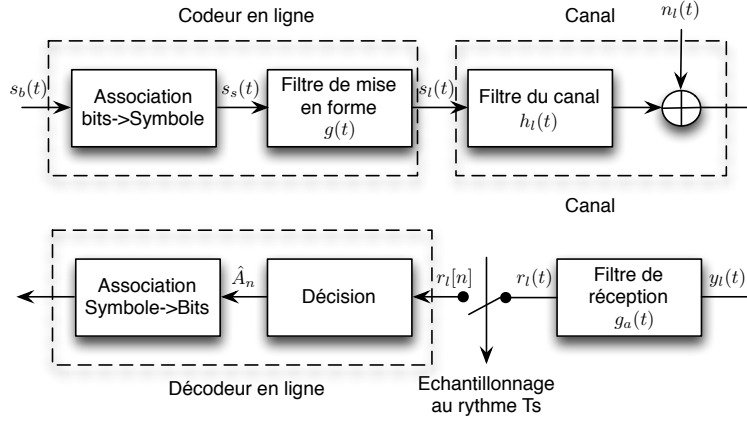


FIGURE 1 – Architecture Bande de base simplifiée

- Fréquence d'échantillonnage :  $f_e = \frac{1}{T_e} = 10\text{kHz}$ ,
- Le débit symbole  $D_s = 1\text{kSymboles/s} \Rightarrow T_s = 1\text{ms}$ ,
- Le filtre de mise en forme est :

$$g(t) = \begin{cases} 1 & 0 \leq t < T_s \\ 0 & \text{ailleurs} \end{cases}.$$

- La séquence de bits doit être générée aléatoirement de manière uniforme (i.e.  $\mathbb{P}(b_k = 0) = \mathbb{P}(b_k = 1) = \frac{1}{2}$ ),
- $F_{se} = \frac{T_s}{T_e}$  représente le facteur de sur-échantillonnage permettant d'adapter le rythme du signal présenté en entrée du filtre de mise en forme,
- Les symboles  $A_k$  sont iid et issus d'une modulation QPSK et  $A_k \in \{\pm \frac{1}{\sqrt{2}} \pm \frac{j}{\sqrt{2}}\}$ ,
- Le nombre de symboles à émettre par paquet est  $N_s = 5000$ ,
- Toutes les transformées de Fourier (TF) seront faites sur  $N_{fft} = 512$  points

## 3.2 Simulation du cas QPSK sans bruit

### 3.2.1 Écriture du script principal (main\_QPSK.m)

A partir des paramètres de simulations et en vous servant des fonctions Matlab décrites dans la section 2, coder la chaîne de communications numériques en utilisant le squelette de code suivant :

Listing 1 – Fichier **main\_QPSK.m**.

```

1 %% noms des binômes
2 clear; % Efface les variables de l'environnement de
   travail
3 close all; % Ferme les figures ouvertes
4 clc; % Efface la console
5
6 %% Initialisation des paramètres
7 fe = 1e4; % Fréquence d'échantillonnage

```

```

8 | M = 4; % Nombre de symboles dans la modulation
9 | n_b = log2(M); % Nombre de bits par symboles
10 | % ... autres paramètres
11 |
12 | %% Émetteur
13 | %% Récepteur
14 | %% Affichage des résultats

```

Avant de passer à la suite, **vous devez tester votre chaîne de communications numériques**. Pour cela, vérifier que les bits estimés en sortie du bloc "Association Symbole→bits" sont bien les mêmes que les bits émis, lorsque la variance du bruit  $n_l(t)$  vaut  $\sigma_{n_l}^2 = 0$ . Pour cela calculer le taux d'erreur binaire (TEB) et vérifiez qu'il vaut 0 en l'absence de bruit.

### 3.2.2 Affichage des résultats

Une fois l'étape précédente validée, tracer et interpréter les figures suivantes sachant que *chaque figure doit être interprétée et doit comporter un titre, des labels, des unités et une légende*.

1. Allure temporelle de la partie réelle des signaux  $s_l(t)$  et  $r_l(t)$  pour  $t \in [0, 10T_s - T_e]$ . En vous servant de ces tracés, illustrer le retard introduit par la causalité des filtres sur la prise de décision du premier symbole reçu.
2. Périodogramme de Welch de  $s_l(t)$  à comparer avec la DSP théorique  $\Gamma_{s_l}(f)$ . Le calcul des DSP expérimentales se fera ici en utilisant la méthode du périodogramme de Welch en utilisant une fenêtre rectangulaire de  $N_{FFT}$  échantillons et 0 échantillon en commun entre deux fenêtres successives. Vous pouvez utiliser la méthode `pwelch` de Matlab pour cela à condition de bien lire sa documentation.

## 3.3 Simulation d'une QPSK en bande de base en présence de bruit

### 3.3.1 Nouveau script principal (main\_QPSK\_TEB.m)

Créez un nouveau script principal pour réaliser des simulations prenant en compte le bruit. Nous voulons réaliser une simulation pour différentes variances de bruit.

Listing 2 – Fichier `main_QPSK_TEB.m`.

```

1 | %% noms des binômes
2 | clear; % Efface les variables de l'environnement de
   | travail
3 | close all; % Ferme les figures ouvertes
4 | clc; % Efface la console
5 |
6 | %% Initialisation des paramètres
7 | fe = 1e4; % Fréquence d'échantillonnage
8 | M = 4; % Nombre de symboles dans la modulation
9 | n_b = log2(M); % Nombre de bits par symboles
10 | % ... autres paramètres
11 |
12 | Eg = % Energie du filtre de mise en forme
13 | sigA2 = % Variance théorique des symboles
14 |

```

```

15 eb_n0_dB = 0:0.5:10; % Liste des Eb/N0 en dB
16 eb_n0     = 10.^(eb_n0_dB/10); % Liste des Eb/N0
17 sigma2    = sigA2 * Eg ./ (n_b * eb_n0); % Variance du bruit
    complexe en bande de base
18
19 TEB = zeros(size(eb_n0)); % Tableau des TEB (résultats)
20 Pb  = qfunc(sqrt(2*eb_n0)); % Tableau des probabilités d'
    erreurs théoriques
21
22 for i = 1:length(eb_n0)
23     error_cnt = 0;
24     bit_cnt = 0;
25     while error_cnt < 100
26
27         %% Émetteur
28         %% Canal
29         nl = sqrt(sigma2(i)/2) * (randn(size(s1)) + 1j*
            randn(size(s1))); % Génération du bruit blanc
            gaussien complexe
30         %% Récepteur
31
32         error_cnt = error_cnt + ... % incrémenter le
            compteur d'erreurs
33         bit_cnt   = bit_cnt + ... % incrémenter le
            compteur de bits envoyés
34     end
35     TEB(i) = error_cnt/bit_cnt;
36 end
37
38 %% Affichage des résultats

```

### 3.3.2 Affichage des résultats

Tracer et interpréter les figures suivantes sachant que *chaque figure doit être interprétée et doit comporter un titre, des labels, des unités et une légende.*

1. L'évolution du TEB en fonction de rapport  $\frac{E_b}{N_0}$  en dB, lorsque ce dernier varie de 0dB à 10dB par pas de 0.5dB. Superposer cette courbe avec celle de la probabilité d'erreur binaire théorique dans le cas 2-PAM :  $P_b = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{E_b}{N_0}} \right)$ . Que constatez-vous? On considère le point  $\text{TEB} = f\left(\frac{E_b}{N_0}\right)$  comme probable si il est obtenu avec un minimum de 100 erreurs binaires de transmission, ceci justifie l'emploi de la boucle **while** interne.
2. Reprendre les questions précédentes (i.e. des parties 3.2.2 et 3.3.1) en prenant cette fois-ci le filtre de mise en forme en racine de cosinus sur-élevé de roll-off  $\alpha = 0.5$  et de temps de propagation de groupe  $T_g = 4T_s$ .

### 3.3.3 Rapport

Dresser un bilan comparatif de l'utilisation des deux filtres de mise en forme. Ce bilan inclura la comparaison des points suivants :

- **latence de réception** du premier symbole : illustration avec les figures 3.2.2-1 des deux filtres de mise en forme
- **bande passante** (obtenue en mesurant la largeur du lobe principal) : illustration avec les figures 3.2.2-2 des deux filtres de mise en forme
- **efficacité spectrale** : obtenue par la formule suivante  $\eta = \frac{D_b}{B}$  où  $D_b$  est le débit binaire et  $B$  la bande passante.
- **Étude de la probabilité d'erreur** : illustration avec une figure contenant les courbes des taux d'erreurs binaires fonction de  $\frac{E_b}{N_0}$  ainsi que la courbe de probabilité d'erreur binaire théorique.

Conclure cette partie en trouvant deux limitations favorisant l'un ou l'autre des filtres de mise en forme.

## 4 Communications numériques sur fréquence porteuse - main\_QPSK\_fp.m

Dans cette partie vous allez vous intéresser au cas des communications numériques sur fréquence porteuse émettant des symboles QPSK ( $M = 4$ ), dans le cadre d'un canal à bande passante infinie. Vous vérifierez la cohérence des résultats théoriques obtenus en cours avec la simulation et évalueriez les performances de la chaîne de communications numériques en terme de probabilité d'erreur binaire.

### Paramètres de simulation :

- Fréquence d'échantillonnage :  $f_e = \frac{1}{T_e} = 10\text{kHz}$ ,
- Le débit symbole  $D_s = 1\text{kSymboles/s} \Rightarrow T_s = 1\text{ms}$ ,
- Le nombre de symboles de la constellation est  $M = 4$ ,
- La fréquence de la porteuse  $f_0 = 2,5\text{kHz}$ ,
- Le filtre de mise en forme  $g(t)$  est un filtre en racine de cosinus sur-élevé de roll-off  $\alpha = 0.5$  et de temps de propagation de groupe  $T_g = 4T_s$ ,
- La séquence de bits doit être générée aléatoirement de manière uniforme (i.e.  $\mathbb{P}(b_k = 0) = \mathbb{P}(b_k = 1) = \frac{1}{2}$ ),
- $F_{se} = \frac{T_s}{T_e}$  représente le facteur de sur-échantillonnage permettant d'adapter le rythme du signal présenté en entrée du filtre de mise en forme,
- Le nombre de symboles à émettre par paquet est  $N_s = 5000$ ,
- le canal de propagation  $h(t)$  est supposé à bande passante infinie,
- le bruit  $n(t)$  est réel, blanc, gaussien de variance  $\sigma_n^2 = \frac{N_0}{4}$ ,
- Toutes les transformées de Fourier (TF) seront faites sur  $N = 512$  points, et vous représenterez les modules des différents spectres entre  $[-\frac{f_e}{2}, \frac{f_e}{2} - \frac{f_e}{N}]$ .

### 4.1 Implémentation sous Matlab

1. Générer une porteuse complexe à  $f_0 = 2500\text{Hz}$ , multiplier ce signal à l'enveloppe complexe  $s_l(t)$  puis émettre  $s(t)$  sur un canal de propagation  $h(t)$  à bande passante infinie. Afin de reconstruire l'enveloppe complexe au récepteur, mettre en oeuvre avant le filtrage adapté la méthode par projections orthogonales (cf. Fig. 3). Vous vérifierez le bon fonctionnement de votre récepteur en traçant les constellations de  $s_s(t)$  et de  $r_l[n]$ .

**Remarque :** dans le cadre de la simulation vous n'avez pas besoin de mettre en oeuvre les filtres passe-bas après les mélangeurs. En effet, le filtrage de la composante à  $2f_0$  après les mélangeurs en réception est réalisé par le filtre adapté qui est un filtre passe bas.

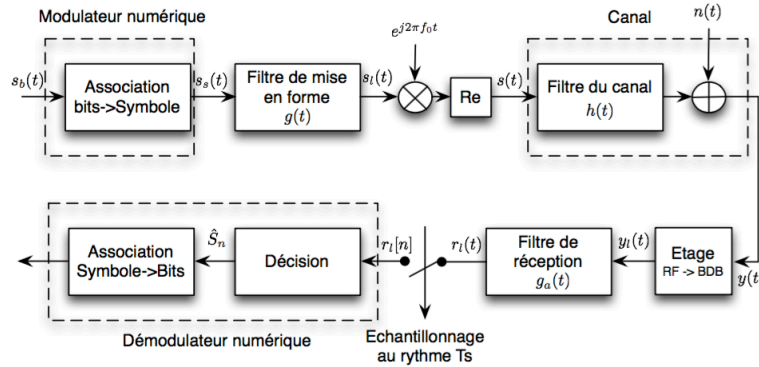


FIGURE 2 – Chaîne de communications numériques sur fréquence porteuse

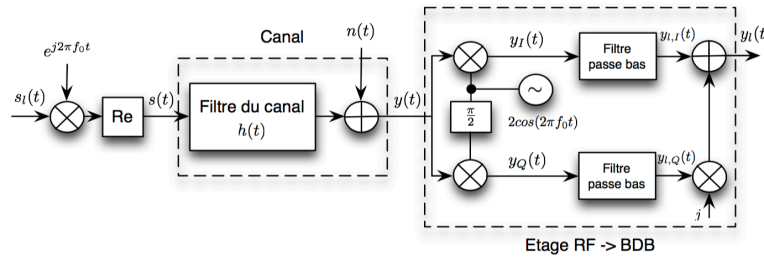


FIGURE 3 – Architecture de réception par projections orthogonales

2. Comparer la DSP théorique de  $s(t)$  avec la DSP expérimentale (obtenue par la méthode de Welch).
3. **Après avoir introduit le bruit**, tracer l'évolution  $TEB=f(\frac{E_b}{N_0})$  et comparer avec le résultat théorique de la probabilité d'erreur binaire  $P_b$  d'une QPSK.
4. Déphaser l'oscillateur local en réception de  $\frac{\pi}{6}$ , tracer la constellation des symboles estimés après filtrage adapté pour un  $\frac{E_b}{N_0}=10\text{dB}$ .
5. Introduire une erreur sur la fréquence porteuse de réception de 10Hz, tracer la constellation des symboles estimés après filtrage adapté et échantillonnage au rythme symbole pour un  $\frac{E_b}{N_0}=10\text{dB}$ .
6. Introduire une erreur sur la fréquence porteuse de réception de 1000Hz, tracer la constellation des symboles estimés après filtrage adapté et échantillonnage au rythme symbole pour un  $\frac{E_b}{N_0}=10\text{dB}$ .

## 4.2 Rapport

Dans le rapport présenter les points suivants :

- Illustrer le principe de mise sur fréquence porteuse en répondant à la question 4.1-2. Sur cette figure, faire apparaître la **fréquence porteuse**, la **bande passante**.

- **Étude de la probabilité d'erreur** : illustration avec une figure contenant les courbes des taux d'erreurs binaires fonction de  $\frac{E_b}{N_0}$  ainsi que la courbe de probabilité d'erreur binaire théorique en bande de base.
- **Étude d'un récepteur présentant des défauts** : présenter l'impact sur la constellation en réception d'un déphasage et d'une erreur de fréquence porteuse (questions 4.1-4 à 4.1-6). Enfin, proposer une démonstration mathématique de ces trois résultats.

## 5 Contacts

- Dominique Dallet - dominique.dallet@ims-bordeaux.fr
- Guillaume Ferré - guillaume.ferre@ims-bordeaux.fr
- Romain Tajan - romain.tajan@ims-bordeaux.fr