TP de communications numériques
module TS227 – année 2019/2020

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) afin d'évaluer notamment leurs performances en termes de probabilité d'erreur binaire.

Les TP se font en **binôme ou monôme** et l'évaluation porte sur une note de rapport et une note de travail continu (en séances).

Concernant le rapport, il ne doit pas excéder **15 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

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

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

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

— 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 `~=`.

— 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 2-PAM en bande de base (main_2PAM.m)

Dans cette partie vous allez vous intéresser au cas des communications numériques en bande de base émettant des symboles 2-PAM. 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 évaluerez 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 :

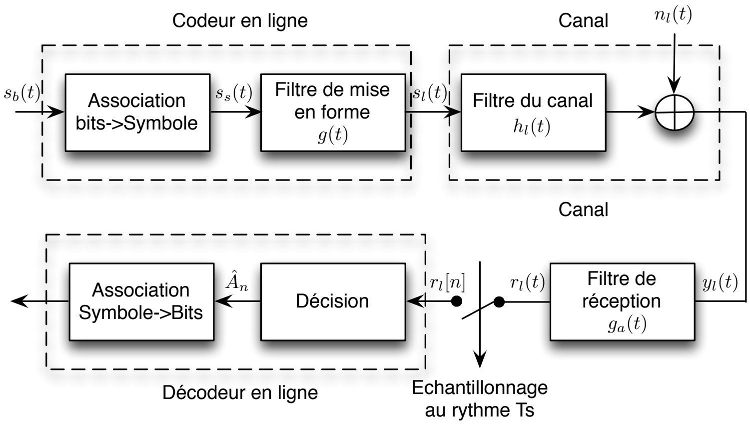

FIGURE 1 – Architecture Bande de base simplifiée

— 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 :

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

— 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 $A_k \in \{\pm 1\}$,
— 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, et vous représenterez les modules des différents spectres entre $[\frac{-f_e}{2}, \frac{f_e}{2} - \frac{f_e}{N_{fft}}]$.

Les hypothèses sur le canal de propagation et le bruit sont les mêmes que celles de l'étude théorique faite en TD et le mapping est le suivant : si $b_k = 0 \Rightarrow A_k = -1$ et si $b_k = 1 \Rightarrow A_k = +1$.

3.2 Simulation du cas 2-PAM sans bruit

3.2.1 Écriture du script principal (main_2PAM.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_2PAM.m**.

```
%% noms des binômes
clear;      % Efface les variables de l'environnement de
   travail
close all;  % Ferme les figures ouvertes
clc;        % Efface la console
```

```

%% Initialisation des paramètres
fe = 1e4; % Fréquence d'échantillonnage
% ... autres paramètres

%% Émetteur
%% Récepteur

%% 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 $\sigma^2_{n_l} = 0$. 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 des signaux $s_sl(t)$, $s_l(t)$ et $r_l(t)$ pour $t \in [0,\ 50T_s - T_e]$. 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. Diagramme de l'oeil de $r_l(t)$ sur les 1000 premiers symboles, en superposant des signaux de durée $3T_s$,
3. DSP [1] de $s_s(t)$ et de $s_l(t)$ à comparer avec les DSP théoriques.

3.3 Simulation du cas 2-PAM en présence de bruit

3.3.1 Modification du script principal (main_2PAM.m)

Modifier votre script `main_2PAM.m` afin de prendre en compte le bruit. Nous voulons réaliser une simulation pour différentes variances de bruit.

Listing 2 – Fichier **main_2PAM.m** modifié pour ajouter du bruit.

```
%% noms des binômes
clear;     % Efface les variables de l'environnement de
  travail
close all; % Ferme les figures ouvertes
clc;       % Efface la console

%% Initialisation des paramètres
fe = 1e4; % Fréquence d'échantillonnage
% ... autres paramètres
eb_n0_dB = 0:0.5:10; % Liste des Eb/N0 en dB
eb_n0    = 10.^(eb_n0_dB/10); % Liste des Eb/N0
TEB = zeros(size(eb_n0)); % Tableau des TEB (résultats)
```

1. Le calcul des DSP expérimentales se fera en utilisant la méthode du périodogramme de Welch décrite en cours (sans fenêtrage ni chevauchement), pour cela vous découperez le signal $s_l(t)$ en signaux de N échantillons.

```
Pb   = qfunc(sqrt(2*eb_n0)); % Tableau des probabilités d'
    erreurs théoriques

for i = 1:length(eb_n0)
    %% Émetteur
    %% Canal
    %% Récepteur
end

%% 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[2] 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 : $P_b = \frac{1}{2}erfc\left(\sqrt{\frac{E_b}{N_0}}\right)$. Que constatez-vous ?
2. Refaire la même manipulation que précédemment mais en introduisant une erreur de synchronisation temporelle en sortie de filtre adapté au moment de l'échantillonnage de $r_l(t)$ au rythme T_s. Cette erreur sera prise égale à 10% de T_s soit un nombre de $10\% F_{se}$ échantillons. Que constatez vous ? Quel phénomène introduit cette erreur de synchronisation ?
3. Quelle est en dB la perte de sensibilité du récepteur si la qualité de service de la transmission est telle que le TEB$<10^{-3}$?
4. Reprendre les questions précédentes (i.e. des parties 2.2.2 et 2.3.1) 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 $T_g = 4T_s$.
5. Dresser un bilan comparatif de l'utilisation des deux filtres de mise en forme (DSP, efficacité spectrale, P_b, sensibilité à la désynchronisation temporelle, etc.).

4 Communications numériques sur fréquence porteuse - main_QPSK.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 évaluerez 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 $f_0 = 2,5\text{kHz}$,

2. On considère le point TEB=f($\frac{E_b}{N_0}$) comme probable si il est obtenu avec un minimum de 100 erreurs binaires de transmission.

— 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 supposé 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 $[\frac{-f_e}{2}, \frac{f_e}{2} - \frac{f_e}{N}]$.

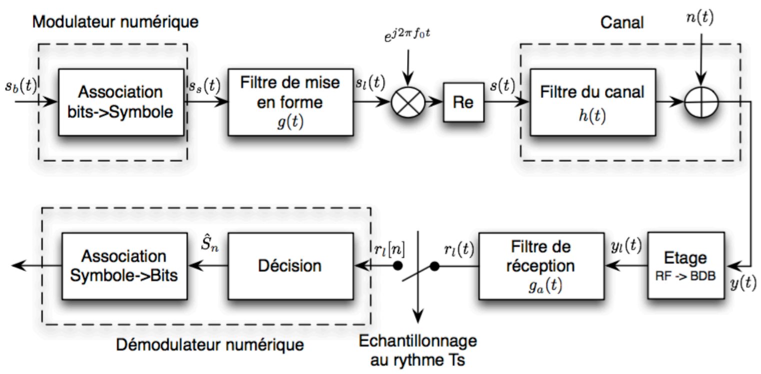

FIGURE 2 – Chaîne de communications numériques sur fréquence porteuse

Les simulations nécessaires pour répondre aux questions 1 à 4 doivent être effectuées sans bruit.

1. Pour cette question, **rester en bande de base**. A partir notamment des fonctions "bi2de.m", "pskmod.m", "pskdemod.m" et de "de2bi.m", implémenter un modulateur numérique QPSK et son démodulateur en prenant une phase initiale de $\frac{\pi}{4}$.
$\Rightarrow$ Remarque : l'utilisation des fonctions précédentes rend le programme adaptable aux modulations M-ASK (M-PAM) et M-APK (M-QAM), puisqu'il suffit de remplacer pskmod() par qammod()/pammod() et pskdemod() par qamdemod()/pamdemod(), etc.
Indications : En sortie du filtre adapté, veillez à tenir compte du retard introduit par la causalité des filtres (mise en forme et adapté) afin de récupérer au bon instant le premier échantillon de symbole détecté.
2. Tracer le diagramme de l'oeil de $s_l(t)$ ainsi que celui de $r_l(t)$ (toujours sans bruit), que constatez-vous ? Interprétez.
3. Générer une porteuse complexe à $f_0 = 2500$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). Tracer les constellations de $s_s(t)$ et de $r_l[n]$, que constatez-vous ? Interprétez.
4. Comparer la DSP théorique de $s(t)$ avec la DSP expérimentale (obtenue par la méthode de Welch).

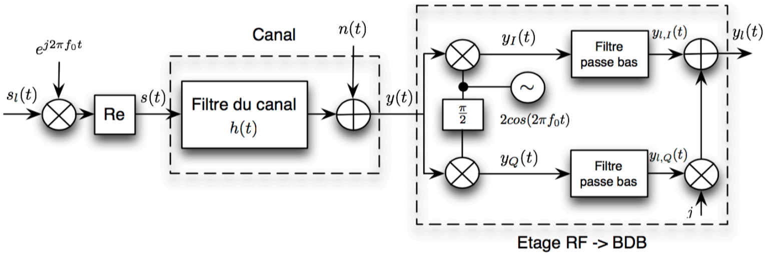

FIGURE 3 – Architecture de réception par projections orthogonales

Remarque : dans le cadre de la simulation vous n'avez pas besoin de mettre en oeuvre les filtres passe-bas après les mélangeurs [3].

5. Adapter le calcul de E_b afin d'évaluer correctement la variance du bruit en entrée du récepteur.
6. **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.
7. Déphaser l'oscillateur local en réception de $\frac{\pi}{6}$, tracer la constellation des symboles estimés après filtrage adapté pour un $\frac{E_b}{N_0}$=10dB, que constatez-vous ? quelle méthode proposez-vous pour corriger cette distorsion ?

5 Contacts

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

3. En effet, le filtrage de la composante à $2f_0$ après les mélangeurs en réception est obtenu par le filtre adapté.