TP 1 de communications numériques

1 Objectifs et évaluation

Le but de ce TP est de développer, à l'aide du logiciel MATLAB, un modulateur et un démodulateur numérique pour des modulations M-PSK.

Dans cette séance, vous écrirez un script interactif de Matlab (notebook) ainsi que des fonctions Matlab permettant :

1. de créer une table de correspondance décrivant l'association bits/symboles,
2. de moduler un signal binaire avec une modulation M-PSK pour lequel l'utilisateur pourra choisir
   — M l'ordre de la modulation,
   — le mapping (naturel ou Gray),
3. de démoduler un signal M-PSK dans du bruit

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

2 Développement du modulateur

2.1 Association bits/symboles

L'objectif de cette partie est de créer la fonction `cstl_psk` qui construit une table de correspondance représentant l'association bit/symboles pour un nombre de symbole M donné.

Cette fonction aura le prototype décrit dans le listing 1 ci-dessous.

Listing 1 – Fonction **cstl_psk.m**.

```
1 function constellation = cstl_psk(M, is_gray)
2 % Fonction qui génère un tableau ordonné de la
     constellation
3 % M       : (int) ordre de la modulation
4 % is_gray : (bool) type du mapping
5 %           true => "Gray", false => "Naturel"
```

La fonction `cstl_psk` construit une "constellation" comme une table de correspondance et prends en entrée trois paramètres :

— `M` : l'ordre de la modulation,

— `is_gray` : un booléen qui indique si l'étiquetage est de Gray ou non.

Le vecteur retourné est nommé `constellation`, il possède une taille M et il contient tous les symboles de la modulation numérique et est construit de telle sorte que sa i^e case contienne le symbole associé à la représentation binaire de $i-1$ ($i-1$ et non i à cause des indices de Matlab).

Exemple : Imaginons que pour $M = 4$, `cstl_psk` retourne le tableau `constellation = [1, j, -1 -j]`. Cette table correspondra au mapping :

- $[0,0] \mapsto 1$
- $[0,1] \mapsto j$
- $[1,0] \mapsto -1$
- $[1,1] \mapsto -j$

Pour comprendre le principe de fonctionnement de cette table, pour trouver le symbole $[1,0]$ (2 en décimal) il suffit de lire `constellation(1 + 2) = -1` (les indices de Matlab commençant à 1).

Afin de vérifier vos résultats, la fonction `plot_cstl` vous est fournie. Elle vous permettra d'afficher un tableau de constellation ainsi que l'association bits/symboles correspondant à l'ordre des symboles dans le tableau.

2.1.1 Mapping Naturel

Compléter la fonction `cstl_psk` afin qu'elle renvoie la constellation d'une M-PSK avec un mapping naturel. On rappelle que les symboles M-PSK sont à considérer dans l'ensemble

$$\mathcal{M} = \left\{ e^{j2\pi \frac{k}{M}} | k \in [0, M-1] \right\}$$

Dans votre notebook : Utilisez la fonction `plot_cstl` pour illustrer le bon fonctionnement de votre mapping pour $M = 4$, $M = 8$ et $M = 16$.

2.1.2 Réalisation du modulateur

Écrire la fonction `mod_psk` qui permettra de créer un vecteur de symboles à partir d'un vecteur binaire et d'un tableau de constellation tel que créé dans la partie précédente.

Listing 2 – Fonction **mod_psk.m**.

```
1 function s = mod_psk(b, constellation)
2 % b             : (vecteur) vecteur contenant les bits à
     moduler
3 % constellation : (vecteur) vecteur la constellation
4 % s             : (vecteur) vecteur de symboles modulés
```

2.1.3 Mapping Gray

On souhaite maintenant que le mapping de Gray soit utilisé. Le mapping de Gray devra être construit en utilisant la construction miroir présentée dans la Figure 1.

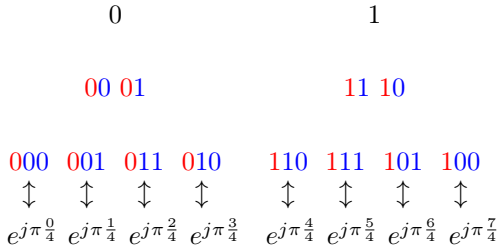

FIGURE 1 – Construction miroir du mapping de Gray pour une 8-PSK

Dans la Figure 1 vous remarquerez que

1. les étiquettes de gauche en bleu sont identiques aux étiquettes de l'itération précédente,
2. les étiquettes de gauche commencent par un 0 (rouge),
3. les étiquettes de droite en bleu sont "symétriques" par rapport aux étiquettes de l'itération précédente,
4. les étiquettes de droite commencent par un 1.

Une fois la liste de M étiquette obtenue, créer la liste de constellation de sorte que la $k^{\text{éme}}$ étiquette de la liste corresponde au symbole $e^{j2\pi\frac{(k-1)}{M}}$. (cf. dernière ligne de la Figure 1).

Astuce : Il n'est pas utile d'implémenter le mapping miroir avec des vecteurs binaires, il est plus simple et plus efficace de l'implémenter en travaillant directement sur les entiers correspondants.

Compléter la fonction `cstl_psk` afin qu'elle renvoie la constellation d'une M-PSK avec un mapping de Gray si `is_gray == true`.

Dans votre notebook : Illustrer le bon fonctionnement de votre mapping de Gray pour $M = 4$, $M = 8$ et $M = 16$. Vous expliquerez, à l'aide du graphique pour $M = 16$ l'avantage de ce mapping.

3 Développement du démodulateur

L'objectif de cette partie est de développer une fonction Matlab qui démodule un signal `y` (éventuellement bruitée) pour une `constellation` donnée en argument.

Listing 3 – Fonction **demod_psk.m**.

```
function bh = demod_psk(y, constellation)
% y             : (vecteur) vecteur du signal reçu
% constellation : (vecteur) vecteur de la constellation
% bh            : (vecteur) vecteur des bits démodulés
```

Cette fonction s'appuiera sur la méthode des plus proches voisins (le symbole décidé sera le symbole de la constellation le plus proche de y_i). Cette fonction renvoie le vecteur binaire correspondant aux symboles décidés.

Dans votre notebook :

1. Générer un vecteur binaire `b` contenant $N_b = 4000$ bits,
2. Moduler le vecteur `b` avec une modulation 16-PSK vérifiant le mapping de Gray, on appellera `s` le vecteur de symboles,
3. Démoduler le vecteur `s` et comparer les bits démodulés aux bits initiaux en calculant le Taux d'Erreurs Binaires (TEB).
4. Commenter votre résultat.

4 Comparaison des mappings en présence de bruit

On souhaite maintenant analyser le comportement des mappings naturel et Gray d'une 16-PSK en présence de bruit.

Dans votre notebook écrire un programme qui effectue les tâches suivantes :

1. Générer un vecteur binaire `b` contenant $N_b = 4000$ bits,
2. Moduler le vecteur `b` avec une modulation **16-PSK avec un mapping Naturel**, on appellera `s1` le vecteur de symboles obtenu,
3. Moduler le vecteur `b` avec une modulation **16-PSK avec un mapping de Gray**, on appellera `s2` le vecteur de symboles obtenu,
4. Ajouter un **même** bruit gaussien complexe de moyenne 0 et de variance $\sigma^2 = 0.02$ à `s1` et `s2`, les vecteurs obtenus sont appelés respectivement `y1` et `y2`,
5. Démoduler `y1` et `y2` et calculer le Taux d'Erreurs Binaires (TEB) dans les deux cas.
6. Afficher `y1` et `y2` comme des nuages de points sur deux figures indépendantes.
7. **À l'aune des figures obtenues dans le point 6, commenter le résultat obtenu dans le point 5**.

5 Contacts

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