

3<sup>ème</sup> année Sciences du Numérique  
Parcours Télécoms Sans Fil et Objets Connectés  
BE Détection Multi-Utilisateurs

**Note :** récupérer le fichier tp\_mud.zip sur <http://moodle-n7.inp-toulouse.fr/course/view.php?id=2467>

## 1 Performances théoriques du filtre adapté, du décorrélateur, et du détecteur MMSE

On considère une transmission mutli-utilisateur synchrone à  $K = 2$  utilisateurs. On note  $R$  la matrice de corrélations entre les signatures,  $\sigma^2$  la variance du bruit additif, et  $A_1$  et  $A_2$  les amplitudes des utilisateurs.

On donne la fonction `[teb_fa,teb_dec,teb_mmse]=teb_theorique(A,R,SNR)` ( $A$  est un vecteur  $2 \times 1$ , et  $R$  une matrice  $2 \times 2$ ,  $SNR$  est un vecteur de taille quelconque) renvoyant le TEB théorique en fonction du rapport  $SNR$  pour les détecteurs suivants :

- filtre adapté ;
  - décorrélateur ;
  - MMSE linéaire.
- Tester ce programme avec les paramètres suivants :
    - $A_1 = A_2 = 1$  ;
    - $R = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix}$  ;
    - 50 valeurs du  $SNR$  entre  $-5$  et  $15$  (commande `linspace`).

Commenter les différences/ressemblances éventuelles entre ces courbes.

- Faire un nouveau test avec  $R = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$ . Commenter les nouvelles courbes, et comparer avec les précédentes.
- Faire un autre test permettant de constater l'effet d'éblouissement (*near-far*) (donner le jeu de paramètres utilisé en expliquant le choix de ceux-ci). Commenter.

## 2 Détecteur MMSE adaptatif

On considère une transmission mutli-utilisateur synchrone à  $K = 2$  utilisateurs, dans un contexte où cette fois-ci le canal peut varier avec le temps, c'est-à-dire que les amplitudes  $A_1$  et  $A_2$  sont *a priori* fonction du temps. Dans le détecteur MMSE, on cherche pour chaque utilisateur  $k$  la fonction  $c_k$  qui minimise l'erreur quadratique  $E[(b_k - \langle c_k, y \rangle)^2]$ . En décomposant la fonction  $c_k$  sur les formes d'onde  $s_1$  et  $s_2$ , le problème revient à trouver les coefficients  $\underline{c}_k = (c_k^1, c_k^2)^T$  qui minimisent  $E[(b_k - \underline{c}_k^T \underline{y})^2]$ , où  $\underline{y}$  est le vecteur des sorties des filtres adaptés. La décision est alors :  $\hat{b}_k = \text{sign}(\underline{c}_k^T \underline{y})$ . Dans l'algorithme MMSE adaptatif, on construit pour chaque utilisateur  $k$  une suite de coefficients  $\underline{c}_k(n)$ . En mode d'apprentissage, les bits émis par l'émetteur sont connus par le récepteur, ce qui permet de commencer à faire converger les coefficients : ces coefficients sont mis à jour à chaque arrivée d'un bit  $b_k(n)$  par la formule

$$\underline{c}_k(n) = \underline{c}_k(n-1) - \mu (\underline{c}_k(n-1)^T \underline{y}(n) - b_k(n)) \underline{y}(n)$$

En mode poursuite, la mise à jour se fait avec la décision  $\hat{b}_k(n)$  prise sur le bit émis :

$$\underline{c}_k(n) = \underline{c}_k(n-1) - \mu \left( \underline{c}_k(n-1)^T \underline{y}(n) - \hat{b}_k(n) \right) \underline{y}(n)$$

On donne les fonctions Matlab suivantes :

- `[signatures, Rest] = generer_signatures(Nbchips, R)`, qui renvoie une matrice comportant des signatures (une signature par ligne) telle que la matrice de corrélation soit à peu près  $R$  (“à peu près”, car avec un nombre fini de chips, on ne peut pas obtenir n’importe quel coefficient de corrélation dans  $[-1, 1]$ ). `Rest` est la matrice de corrélation réelle ;
  - `[ycont, bits] = signal(signatures, A, sigma2)`, qui renvoie un vecteur contenant le signal multi-utilisateur “continu” correspondant à une période bit, ainsi que le vecteur des 2 bits générés (c’est-à-dire 1 bit pour chacun des 2 utilisateurs).
  - `y = filtre_adapte(ycont, signatures)`, qui renvoie le vecteur des sorties des filtres adaptés aux signatures correspondant au signal “continu” `ycont`.
  - Créer un programme `exo3.m` permettant de mettre en oeuvre le MMSE adaptatif en commençant par le mode apprentissage.
- Tester ce programme avec  $K = 2$ ,  $A_1 = A_2 = 1$ ,  $R = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$ ,  $N_c = 32$  chips,  $\sigma^2 = 0.05$ ,  $\mu = 0.05$ , et 5000 itérations d’apprentissage. Visualiser la convergence des coefficients et la courbe d’erreur (entre les coefficients mis à jour et les coefficients théoriques).
- Compléter le programme avec le mode poursuite, et le tester avec les mêmes paramètres et 5000 itérations de poursuite. Déterminer alors le TEB (sur la poursuite) et comparer aux performances théoriques.
  - Refaire une simulation avec les mêmes paramètres, mais en faisant varier les amplitudes sous la forme

$$\begin{aligned} A_1(i) &= 1 + 0.8 \cos(2\pi f_0 i) \\ A_2(i) &= 2(1 + 0.8 \cos(2\pi f_0 i)) \end{aligned}$$

avec  $f_0 = 10^{-3}$  ( $i$  correspond au numéro du bit). Visualiser l’évolution des coefficients et commenter.