BE n°4 – Probabilités/Statistique avec MATLAB
Détection – Tests statistiques

1 Transmission de texte par modulation BPSK

En transmission BPSK ("Binary Phase Shift Keying") antipodale, on émet des bits "1" ou "0" (codés respectivement pas des niveaux "+1" et "−1") en les "modulant" par une sinusoïde à la fréquence f_0. Plus précisément, pour transmettre le bit $b_k = 1$, on émet un signal sinusoïdal $s(t)$ pendant un intervalle $[kT; (k+1)T]$, et pour le bit $b_k = 1 = -1$, on émet le signal $-s(t)$, avec

$$s(t) = \sin(2\pi f_0 t).$$

Remarquons que

$$-s(t) = -\sin(2\pi f_0 t) = \sin(2\pi f_0 t + \pi).$$

1.1 Détection d'une suite de bits sur un canal gaussien

Le signal émis passe dans un canal dans lequel il est multiplié par une amplitude constante positive A, et perturbé par un bruit additif gaussien $w(t)$ de moyenne 0 et de variance σ^2. On note $y(t)$ le signal reçu. Pour chaque bit b_k émis, on a donc le problème de détection suivant :

$$\begin{array}{ll} \mathrm{H}_1(b_k = 1): & y(t) = y_1(t) = As(t) + w(t),\ t \in [kT, (k+1)T] \\ \mathrm{H}_{-1}(b_k = -1): & y(t) = y_{-1}(t) = -As(t) + w(t),\ t \in [kT, (k+1)T]. \end{array}$$

On envisage ici une approche bayésienne. Plus précisément, on étudie la détection par Maximum A Posteriori (MAP), c'est-à-dire que le bit $\hat{b}_k$ détecté est celui qui maximise la probabilité a posteriori

$$\hat{b}_k = \underset{\epsilon \in \{1,-1\}}{\operatorname{argmax}} \mathrm{P}\left[b_k = \epsilon | y(t)\right] \tag{1}$$

Pour des bits équiprobables a priori, (i.e. $\mathrm{P}[b_k = -1] = \mathrm{P}[b_k = 1]$), on montre que l'estimateur MAP $\hat{b}_k$ est donné par

$$\hat{b}_k = \text{signe}\left[\int_{kT}^{(k+1)T} y(t)s(t)dt\right]. \tag{2}$$

Le taux d'erreur de bits, c'est-à-dire la probabilité d'erreur $\mathrm{P}\left[\hat{b}_k \neq b_k\right]$, est alors donné par

$$P_e = 1 - \Phi\left(\frac{A\,\|s\|}{\sigma}\right).$$

où $\Phi(\cdot)$ est la fonction de répartition de la loi normale centrée réduite :

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt. \tag{3}$$

Pour simuler ce problème sur Matlab, il faut travailler avec des données discrètes. On remplace donc le signal $\sin(2\pi f_0 t)$ sur $[0;T]$ par le vecteur $s = [\sin(2\pi \tilde{f}_0 n)]$, $n = 1\ldots,N$, et le signal $w(t)$ par le vecteur gaussien $w = [w(1),\ldots,w(N)]^T$ de loi $\mathcal{N}(0,\sigma^2 I_N)$. On travaille donc avec le signal numérique

$$y(n) = \pm A\sin(2\pi\tilde{f}_0 n) + w(n), \quad n = 1,\ldots,N.$$

La décision (2) est alors remplacée par

$$\hat{b}_k = \text{signe}\left[\sum_{n=1}^{N} y(n)s(n)\right]. \tag{4}$$

On définit d'autre part le rapport signal-sur-bruit par

$$\text{RSB} = 10\log_{10}\left(\frac{\frac{1}{N}\sum_{n=1}^{N} s_n^2}{\sigma^2}\right)$$

qui mesure la puissance du signal $\frac{1}{N}\sum_{n=1}^{N} s_n^2$ comparée à celle du bruit σ^2.

On souhaite alors émettre une suite de bits de longueur N_{bits}, détecter chacun d'entre eux à l'aide de la méthode donnée ci-dessus, puis estimer le Taux d'Erreur de Bits (TEB), c'est-à-dire le pourcentage de bits mal détectés, en fonction de RSB. On va donc générer une suite de bits équiprobables, moduler chacun d'entre eux par la sinusoïde appropriée, et effectuer sur chacune d'elle la détection (4).

<u>Travail à réaliser :</u>

1. Générer un vecteur `bits` de $N_{\text{bits}} = 100000$ bits équiprobables.
2. Créer un vecteur `bits_modules` de bits modulés, c'est-à-dire multiplier chaque bit généré par une sinusoïde de fréquence $\tilde{f}_0 = 0.1$ sur $N = 20$ échantillons, et stocker chacune de ces sinusoïdes dans le vecteur `bits_modules`. On obtient alors un vecteur de $N \times Nbits$ points (utiliser par exemple la fonction `kron`, ou à défaut faire une boucle).
3. Créer un bruit gaussien (variable `bruit`) (prendre RSB$= 10$). Ajouter ce bruit aux bits modulés pour obtenir le signal émis (variable `signal_emis`). Appliquer à chaque segment de N points de `signal_emis` la prise de décision (4) pour déterminer les bits détectés (on pourra utiliser `reshape`). Stocker dans un vecteur `bits_detectes`.
4. Estimer alors le TEB en comparant `bits` à `bits_detectes`.
5. Reprendre les questions 1 à 4 pour 20 valeurs de RSB entre -10 et 0. Afficher alors la courbe du TEB en fonction de RSB en échelle logarithmique.

1.2 Application à la transmission d'un texte

Remarque : pour les besoins de cette partie, récupérer les fichiers `c_source.m`, `d_source.m`, et `dico.mat` situé à l'adresse

http://dobigeon.perso.enseeiht.fr/teaching/signal/Matlab_be4_data.zip

On souhaite maintenant transmettre un texte à l'aide du principe d'émission/détection de bits donné ci-dessus. Pour cela, on dispose des fonctions `c_source` et `d_source` qui permettent respectivement de générer une suite de bits à partir d'un texte, et de reconstituer un texte à partir d'une suite de bits. Le codage utilisé est tel que les caractères les plus courants sont codés avec le moins de bits possibles, et vice versa, et ce pour avoir une suite de bits la plus courte possible (le codage utilisé est appelé codage de Huffman).

Reprendre la question précédente en prenant un RSB fixé (par exemple $RSB = 0$). Créer un texte (une cinquantaine de caractères environ), et le transformer en une suite de bits à l'aide de la fonction `c_source` (cette opération remplace donc la génération de bits du paragraphe précédent). Effectuer la détection comme précédemment. Puis reconstituer le texte à partir de la suite de bits obtenue à l'aide de la fonction `d_source`. (**Attention** : la fonction `c_source` renvoie des bits 0/1 qu'il faudra transformer en niveaux ± 1. De même, après la détection, il faut convertir les niveaux ± 1 en bits 0/1 avant de faire le décodage). Faire plusieurs expériences avec différentes valeurs de RSB.

2 Détection d'un signal dans du bruit

Un signal $s = (s_1, \ldots, s_N)^T$ est transmis sur un canal de transmission. Ce signal est modulé en amplitude par un paramètre θ *constant sur toute la durée du signal*. Ce signal modulé est alors perturbé par un bruit additif $b = (b_1, \ldots, b_N)^T$ gaussien $\mathcal{N}(0, \Sigma)$, de telle sorte que le signal reçu $y = (y_1, \ldots, y_N)^T$ s'écrive sous la forme :

$$y_n = \theta s_n + b_n, \quad 1 \leq n \leq N.$$

On suppose désormais que le paramètre θ est une variable aléatoire. On fait alors l'hypothèse que θ est une variable binaire valant 1 ou -1 avec les mêmes probabilités ($\mathrm{P}[\theta = 1] = \mathrm{P}[\theta = -1]$).

Le récepteur, en possession du signal $y = (y_1, \ldots, y_N)^T$, souhaite savoir si ce signal ne contient que du bruit, ou s'il contient le signal modulé perturbé par le bruit additif. Ce problème de détection de signal dans du bruit additif s'écrit donc sous la forme du test d'hypothèses suivant :

$$\begin{array}{lll} H_0 : & y_n = b_n, & 1 \leq n \leq N \\ H_1 : & y_n = \theta s_n + b_n, & 1 \leq n \leq N \end{array}$$

La statistique de test donnée par le lemme de Neyman-Pearson s'écrit :

$$T(y_1, \ldots, y_n) = \left| s^T \Sigma^{-1} y \right|.$$

Pour une probabilité de fausse alarme α, la région critique (zone de rejet de H_0) est donnée par :

$$\begin{aligned} R_\alpha = & \left\{ (y_1, \ldots, y_n) \in \mathbb{R}^N \middle| \left| s^T \Sigma^- 1 y \right| > \lambda_\alpha \right\} \\ = & \left\{ (y_1, \ldots, y_n) \in \mathbb{R}^N | s^T \Sigma^{-1} y \in] - \infty, -\lambda_\alpha] \cup] \lambda_\alpha, +\infty[\right\} \end{aligned} \tag{5}$$

Dans ce cas, le seuil de décision s'écrit

$$\lambda_\alpha = -\sigma_s \Phi^{-1}\left(\frac{\alpha}{2}\right)$$

où $\sigma_s^2 = s^T \Sigma^{-1} s$. La probabilité de non-détection (ou risque de 2ième espèce) s'écrit :

$$\beta = \Phi\left(-\Phi\left(\frac{\alpha}{2}\right) + \sigma_s\right) - \Phi\left(\Phi\left(\frac{\alpha}{2}\right) + \sigma_s\right)$$

On souhaite tracer les courbes théoriques de la puissance du test $\pi = 1 - \beta$ en fonction de la probabilité de fausse alarme α, puis retrouver ces courbes par simulations.

Travail à réaliser :

1. En utilisant les fonctions `norminv` et `normcdf`, écrire une fonction

   `p = pi_theorique(signal,Sigma,N)`

   qui renvoie la puissance théorique π du test pour $\alpha \in \{0.01, 0.02, \ldots, 0.98, 0.99\}$ en fonction de `signal`, `Sigma` et `N`. Afficher le vecteur obtenu avec $N = 20$, $s_n = \sin(2\pi \tilde{f}_0 n)$, $\tilde{f}_0 = 0.1$, $1 \leq n \leq N$, $\Sigma = I_N$. Superposer alors les courbes obtenues pour $\Sigma = 2I_N$ et $\Sigma = 3I_N$. Commenter (à l'aide des expressions de λ_α, σ_s^2 et β).
2. On cherche maintenant à retrouver ces résultats par simulations. Ecrire une fonction `Y = generer(signal, Sigma,N,K)` qui renvoie une matrice `Y` de taille $N \times K$, dont chaque colonne contient une réalisation du signal $y = (y1, \ldots, y_N)$ associée à l'hypothèse H_1.
   Remarque : le paramètre θ doit changer aléatoirement à chacune des K réalisations (mais en restant constant pour les N points d'une réalisation).
3. Ecrire une fonction

   `p = pi_estimee(signal,Sigma,N,K)`

   qui renvoie la puissance estimée $\hat{\pi}$ du test pour $\alpha \in \{0.01, 0.02, \ldots, 0.99\}$, en fonction de `signal`, `Sigma`, `N`, et du nombre de simulations `K` (cette fonction utilisera bien sûr la fonction `generer`). Superposer à la courbe théorique obtenue à la question 1 les résultats obtenus avec $N = 20$, $s_n = \sin(2\pi f_0 n)$, $f_0 = 0.1$, $1 \leq n \leq N$, $\Sigma = I_N$ et $K = 500$. Commenter.