

Master Mathématiques et Applications, 1-ère année, Aix-Marseille Université

Modélisation en Traitement du Signal, TP 2

Année 2015-16

1 Quantification

Le but de cette série d'exercices est d'analyser les performances d'un quantificateur Q lorsqu'il est appliqué à des réalisations d'une variable aléatoire de loi donnée. On rappelle qu'un quantificateur $Q : \mathbb{R} \rightarrow E_M = \{y_-, y_0, \dots, y_{M-3}, y_+\}$, à M niveaux de quantification est défini par la donnée de réels $\{x_0, \dots, x_{M-2}\}$ (les bornes des intervalles de quantification) et $\{y_-, y_0, \dots, y_{M-3}, y_+\}$ (les valeurs quantifiées), par

$$Q(x) = \begin{cases} y_- & \text{si } x \leq x_0 \\ y_k & \text{si } x \in [x_k, x_{k+1}[, \quad k = 0, \dots, M-3 \\ y_+ & \text{si } x > x_{M-2} \end{cases}$$

1.1 Implémentation d'un quantificateur uniforme

Exercice 1.1 *Etudier la fonction MATLAB/OCTAVE `quant0.m` fournie en ligne, qui effectue une quantification prenant en entrée un réel u , et deux vecteurs X et Y représentant respectivement les bornes des intervalles de quantification et les valeurs quantifiées, et retournant une valeur quantifiée $v = Q(u)$.*

Syntaxe : $v = \text{quant0}(u, X, Y)$.

On rappelle que pour un **quantificateur uniforme**, les x_k sont régulièrement espacés, ainsi que les y_k :

$$x_k = x_0 + k\Delta, \quad y_k = \frac{x_k + x_{k+1}}{2} = y_0 + k\Delta, \quad k = 0, \dots, M-3.$$

On peut choisir de prendre, pour simplifier $y_- = y_0 - \Delta$ et $y_+ = y_{M-3} + \Delta$

Exercice 1.2 *Ecrire une fonction MATLAB/OCTAVE `unifpart.m` prenant en entrée les bornes minimale et maximale $x_{\min} = x_0$ et $x_{\max} = x_{M-2}$ des intervalles de quantification, ainsi que le nombre M de niveaux de quantification, et retournant les vecteurs X et Y représentant respectivement les bornes des intervalles de quantification et les valeurs quantifiées.*

Syntaxe : $[X, Y] = \text{unifpart}(x_{\min}, x_{\max}, M)$

Exercice 1.3 *Ecrire une fonction MATLAB/OCTAVE `unifquant.m` prenant en entrée un vecteur U de N réels, les bornes $x_{\min} = x_0$ et $x_{\max} = x_{M-2}$ des intervalles de quantification, et le nombre M de niveaux de quantification, et retournant en sortie le vecteur V des valeurs quantifiées des éléments de U .*

Syntaxe : $V = \text{unifquant}(U, x_{\min}, x_{\max}, M)$

1.2 Etude du quantificateur uniforme

Pour étudier les performances de Q , on étudie sa précision lorsque l'on l'applique à des réalisations d'une variable aléatoire X de densité de probabilités donnée. A X sont associées la variable aléatoire discrète $Y = Q(X)$, à valeurs dans l'ensemble fini E_M , et l'erreur de quantification Z , c'est à dire la variable aléatoire $Z = X - Y = X - Q(X)$.

Le biais B , la distorsion D et le rapport *signal à bruit* de quantification SNR sont définis par

$$B = \mathbb{E}\{Z\}, \quad D = \mathbb{E}\{Z^2\}, \quad \text{SNR} = 10 \log_{10} \left(\frac{\mathbb{E}\{X^2\}}{\mathbb{E}\{Z^2\}} \right).$$

L'objectif de cette étude est de comparer les distorsions obtenues avec des quantificateurs sur un nombre variable de bits R (le nombre de valeurs quantifiées vaut donc $M = 2^R$), pour des variables aléatoires de lois diverses. On se limitera ici au quantificateur uniforme développé dans les exercices précédents.

Exercice 1.4 *Ecrire une fonction MATLAB/OCTAVE testunifquant.m prenant en entrée un vecteur U de réels, les nombres x_min et x_max, et un vecteur de bits de valeurs du débit R, calculant le vecteur V des valeurs quantifiées comme plus haut, et retournant en sortie une estimation du biais B, de la distorsion D et du rapport signal à bruit SNR. Pour évaluer ces quantités, on remplacera l'espérance par la moyenne. La fonction devra également tracer le SNR en fonction de débit.*

Syntaxe : `[B,D,SNR] = testunifquant(U,x_min,x_max,debit)`

On considèrera en particulier les lois suivantes : Loi uniforme (fonction MATLAB/OCTAVE rand), Loi normale (fonction MATLAB/OCTAVE randn), Loi Laplacienne (fonction MATLAB/OCTAVE randl, sur le site du cours). Dans tous les cas il conviendra de choisir les valeurs de x_min et x_max. Parmi les choix possibles, on peut prendre pour x_max la plus grande valeur du vecteur d'entrée U, et x_min = - x_max.

Remarque 1.1 *Comme d'habitude, il est préférable d'éviter de programmer des boucles. La quantification uniforme peut se faire facilement en utilisant des fonctions de type floor, round ou ceil.*

2 Modulation ASK

La modulation ASK, ou modulation par déplacement d'amplitude, associe à chaque bit b un symbole $s(b) \in \{-A, A\}$ puis une forme d'onde $x(t) = s(b)1_{[0,T]}(t)$. Etant donnée une suite binaire $B = b_0b_1 \dots b_{K-1}$, le modulateur ASK lui associe la forme d'onde

$$x(t) = \sum_{k=0}^{K-1} s(b_k)\varphi_k(t), \quad t \in [0,KT], \quad \text{où } \varphi_k(t) = \frac{1}{\sqrt{T}} 1_{[0,T]}(t - kT)$$

Les valeurs de t seront échantillonnées à raison de f_s valeurs par seconde (donc $L = f_s * T$ valeurs dans $[0, T]$); on pourra utiliser l'instruction linspace).

Exercice 2.1 *Écrire une fonction ask_modul.m, prenant comme variables d'entrée une suite binaire B, l'amplitude A, la fréquence d'échantillonnage et la longueur T, et retournant la forme d'onde x.*

Ecrire une fonction ask_demodul.m prenant comme variables d'entrée une forme d'onde x, l'amplitude A, la fréquence d'échantillonnage et la longueur T et retournant une suite binaire démodulée.

Ecrire une fonction ask_power.m calculant la puissance moyenne du modulateur.

Syntaxe : `x = ask_modul(B,A,fs,T); B = ask_demodul(x,A,fs,T); P=ask_power(A,fs,T);`

Exercice 2.2 *Modifier la fonction ask_demodul.m pour prendre en compte des probabilités différentes sur les bits 0 et 1 (voir TD 2).*

Exercice 2.3 1. *Ecrire un script effectuant les opérations suivantes : génération d'un mot binaire aléatoire de longueur K donnée, modulation ASK fournissant une forme d'onde x(t), ajout à x(t) d'un bruit aléatoire gaussien centré de variance égale à 1 et démodulation ASK. Le script calculera aussi l'erreur commise par ce MoDem, mesurée par la proportion de bits faux en sortie.*

2. *Etudier l'erreur en fonction de l'amplitude A. Tracer l'erreur en fonction de la puissance du modulateur.*

3 Devoir et compte-rendu

Mêmes instructions que le TP1. Prendre soin de bien présenter et commenter les résultats obtenus, et décrire les fonctions utilisées. Les devoirs sont à rendre sous forme d'une archive contenant tous les fichiers demandés, à envoyer par mail à `bruno.torresani@univ-amu.fr`