Quantification uniforme

DESS "Génie Informatique et Statistique"
Option "Codage et Compression des Signaux"

On s'intéresse au problème de la quantification d'une variable aléatoire de distribution donnée: étant donnée une variable aléatoire X de loi caractérisée par la densité ρ_X, on se donne le quantificateur Q défini par

$$Q : \mathbb{R} \to E_M = \{y_-, y_0, \dots y_{M-1}, y_+\} ,$$

où

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

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

$$Z = X - Y = X - Q(X) .$$

La distorsion et le rapport *signal à bruit* de quantification sont donnés par

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

L'objectif de ce TP est de comparer les distorsions obtenues avec des quantificateurs sur un nombre variable de bits R, pour des variables aléatoires de lois diverses.

On considère un quantificateur uniforme, c'est à dire défini par des intervalles $[x_k, x_{k+1}[$ de longueur constante et $y_k = (x_k + x_{k+1})/2$. On négligera le bruit de saturation (voir plus bas), et on se contentera d'un quantificateur sur $M = 2^R$ niveaux (en ne considérant donc pas y_- et y_+). On se limite donc à

$$Q(x) = y_k \text{ si } x \in [x_k, x_{k+1}[\, , k = 0, \dots M-1 .$$

On simulera un grand nombre de réalisations de variables `i.i.d.` X avec diverses lois, et on tracera dans chaque cas la courbe "débit-distorsion" correspondante, par exemple SNR_Q en fonction de R. Il s'agit essentiellement d'estimer les variances de X et Z.

On considèrera en particulier les lois suivantes

- Loi uniforme (fonction matlab `rand`)
- Loi normale (fonction matlab `randn`)
- ...

et on comparera le résultat obtenu aux prédictions théoriques.

Remarque 1: pour s'affranchir du bruit de saturation, on peut ajuster les bornes x_0 et x_M en fonction des valeurs simulées minimale et maximale.

Remarque 2: 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`.