Chapitre 2

Quantification scalaire des signaux discrets

Sommaire du chapitre

2.1 Introduction

Après avoir échantillonné un signal analogique, on dispose d'un signal *discret*, c'est-à-dire une suite de nombres réels. À ce stade, la conversion analogique-numérique (souvent désignée par *CAN*) n'est pas encore achevée. Il nous faut encore transformer une suite de nombres réels en une suite de 0 et de 1. C'est l'objet de ce chapitre.

On considère donc une *source* [1] émettant un signal discret en temps. La transformation des nombres émis en suite de nombres binaires consiste en fait en deux opérations : le *codage* et le *décodage*.

Coder, ou *encoder* un signal, c'est appliquer à chacun des termes de la suite qu'il constitue une fonction de la forme :

$$Q : \begin{array}{ccc} [-M, M] & \to & \{I_n\}_{0 \leq n \leq N} \\ s(t) & \mapsto & I_k \end{array} \tag{2.1}$$

où :

- l'ensemble $\{I_n\}_{0 \leq n \leq N}$ est une famille d'intervalles disjoints formant une partition de l'intervalle $[-M, M]$ (N est un entier naturel),
- $s(t)$ est un réel, représentant la valeur du signal s à l'instant t, supposé appartenir à l'intervalle $[-M, M]$ (M est un réel, qui peut être inconnu).

L'encodage est donc une opération irréversible car faisant intervenir une fonction non-injective, qui entraîne une perte d'information.

Un exemple courant est la quantification sur 3 bits. Dans ce cas, l'intervalle $[-M, M]$ est partitionné en $N = 2^3 = 8$ intervalles.

Décoder un signal, c'est réaliser l'opération inverse, c'est-à-dire appliquer à chacun des termes d'une suite d'intervalles une fonction de la forme :

$$I_n \to y_n,$$

où y_n est un réel représentant l'intervalle I_n, par exemple la valeur de son milieu.

Remarque 2.1 :

Les valeurs y_n sont ensuite elles-même codées par des entiers binaires.

La figure 2.1 représente un signal avant et après quantification.

1. Ce terme sera très largement employé dans la suite et désigne un dispositif émettant des signaux à partir de maintenant supposés discrets en temps.

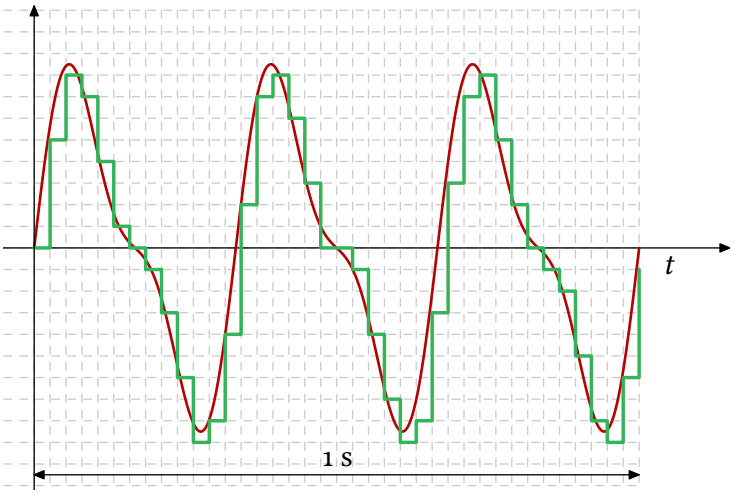

Figure 2.1 : Exemple de quantification d'un signal.

2.2 Formulation mathématique

Dans cette section, on formalise mathématiquement le problème de la quantification. Nous allons voir que le problème de quantification fait intervenir beaucoup de paramètre, et nous allons formuler plusieurs problèmes d'optimisation qui permettent de déterminer la quantification. Comme il est courant dans ce genre de modélisation, les problèmes qui en découlent sont souvent trop difficile à attaquer, et il faut passer par des approximations (fixer *a priori* des paramètres, simplifier le modèle, utiliser des méthodes numériques pour résoudre, etc.).

2.2.1 Cadre

On considère donc une source émettant un signal *aléatoire* S de densité de probabilité $f_S \in L^1(\mathbf{R})$. Pour construire une quantification, il faut donc définir une famille d'intervalles (*i.e.* la famille $\{I_n\}_{0\leq n\leq N}$ de l'introduction), donc de frontières, et de valeurs d'assignation (*i.e.* les valeurs y_n de l'introduction).

Pour fixer les idées conservons les notations de l'introduction et désignons par N le nombre d'intervalles constituant la partition et donc également le nombre de valeurs d'assignations. Le nombre de valeurs frontières, aussi appelées *valeurs de décision*, que l'on note b_i est alors $N+1$. On pose *a priori* $b_0 = -\infty$ et $b_N = +\infty$ et on signalera dans la suite les cas où l'on adopte une autre convention. La fonction de quantification s'écrit :

$$Q(x) = y_n,$$

pour $x \in]b_{n-1}, b_n]$.

2.2.2 Erreur de distorsion de quantification

Définition 2.1 — Erreur de distorsion de quantification

On définit l'erreur quadratique moyenne de quantification par :

$$\begin{aligned}\sigma_q^2(B,Y) &= \int_{-\infty}^{+\infty} (x - Q(x))^2 f_S(x)\mathrm{d}x \\ &= \sum_{n=1}^{N} \int_{b_{n-1}}^{b_n} (x - y_n)^2 f_S(x)\mathrm{d}x,\end{aligned} \tag{2.2}$$

où l'on a noté $B = \{b_n\}_{0\leq n\leq N}$ et $Y = \{y_n\}_{1\leq n\leq N}$. La quantité σ_q est aussi appelée *erreur de distorsion de quantification.*

Une formalisation possible de l'erreur de quantification consiste à considérer son effet comme celui d'un bruit externe affectant le signal S.

Le problème est alors le suivant : étant donné f_S et N le nombre d'intervalles de quantification, trouver les valeurs b_n et y_n solution du problème

$$\min_{B,Y} \sigma_q^2(B,Y).$$

En conclusion, l'erreur de distorsion de quantification dépend à la fois de la partition choisie et du choix du représentant.

2.2.3 Taux de quantification

Définition 2.2 — Taux de quantification

On considère un codage des y_n en entiers binaires de *taille variable* ℓ_n, la taille moyenne des symboles après codage, appelé taux de quantification, est :

$$R = \sum_{n=1}^{N} \ell_n p(y_n), \tag{2.3}$$

où l'on a noté $p(y_n)$ la probabilité d'apparition après codage du symbole correspondant à y_n.

Attention, la valeur R dépend des frontières car :

$$p(y_n) = \int_{b_{n-1}}^{b_n} f_S(x)\mathrm{d}x,$$

et donc :

$$R = \sum_{n=1}^{N} \ell_n \int_{b_{n-1}}^{b_n} f_S(x)\mathrm{d}x.$$

On considère donc selon les cas l'une ou l'autre des reformulations du problème suivantes.

Problème 1

Si l'on se donne une contrainte de distorsion de quantification

$$\sigma_q \leq \sigma^\star, \tag{2.4}$$

où $\sigma^\star$ est un réel fixé, trouver le nombre N, les frontières B et les représentants Y minimisant le taux R, défini par (2.3) et satisfaisant (2.4).

Problème 2

Si l'on se donne une contrainte sur le taux

$$R \leq R^\star, \tag{2.5}$$

où $R^\star$ est un réel fixé, trouver le nombre N, les frontières B et les représentants Y minimisant la distorsion quantification et satisfaisant (2.5).

Remarque 2.2 :

L'une où l'autre de ces formulations constitue un problème plus général que celui que nous allons considérer dans ce chapitre. On se restreint en effet dans la suite à de codage de tailles fixes, si bien que R est constant par rapport à B et Y. Le problème du codage en taille variable sera quant à lui traité au chapitre 3.

2.3 Quantification uniforme

La solution de quantification la plus simple est la quantification uniforme. Dans ce cadre, tout les intervalles ont la même longueur, que l'on note dans la suite Δ. On parle dans ce cas de *quantification uniforme*.

On suppose que l'on adopte une stratégie de codage où N le nombre d'intervalles est pair, c'est-à-dire que $0 \in B$.

2.3.1 Quantification uniforme des sources uniformes

Supposons que pour tout t, $s(t) \in [-S_{\max}, S_{\max}]$ avec une probabilité uniforme. Dans ce cas, on fixe cette fois-ci $b_0 = -S_{\max}$ et $b_N = S_{\max}$. Alors $\Delta = \frac{2S_{\max}}{N}$ et

$$\begin{aligned}\sigma_q^2(B,Y) &= 2\sum_{n=1}^{N/2}\int_{(n-1)\Delta}^{n\Delta}\left(x - \frac{2n-1}{2}\Delta\right)^2 \frac{1}{2S_{\max}}\mathrm{d}x \\ &= \frac{\Delta^2}{12}.\end{aligned}$$

Afin d'évaluer l'effet de l'augmentation du nombre d'intervalle sur la qualité du codage, on considère alors le rapport signal/bruit défini par :

$$SNR = 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_q^2}\right),$$

où σ_x désigne l'écart type du signal S.

Puisque S est une variable aléatoire de loi uniforme à valeurs dans $[-S_{\max}, S_{\max}]$, on a alors :

$$\sigma_x^2 = \sqrt{\mathbb{E}[X^2] - \underbrace{\mathbb{E}[X]^2}_{=0}} = \frac{1}{2S_{\max}}\int_{-S_{\max}}^{S_{\max}} x^2\mathrm{d}x = \frac{S_{\max}^2}{3}.$$

Par conséquent, le rapport signal/bruit vaut :

$$\begin{aligned}SNR &= 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_q^2}\right) \\ &= 10\log_{10}\left(\frac{(2S_{\max})^2}{12} \times \frac{12}{\left(\frac{2S_{\max}}{N}\right)^2}\right) \\ &= 10\log_{10}(N^2) \\ &= 20\log_{10}(2^k) \\ &= 6.02k \text{ dB},\end{aligned} \tag{2.6}$$

où k représente le nombre de bits utilisés pour le codage binaire des représentants y_n.

Définition 2.3 — Décibel

Le décibel (dB) est une unité de grandeur sans dimension définie comme dix fois le logarithme décimal du rapport entre deux puissances, utilisé dans les télécommunications, l'électronique et l'acoustique.

La conclusion du calcul précédent est que l'ajout d'un bit de quantification augmente le rapport signal/bruit d'approximativement 6 dB. C'est un résultat très standard et qui est utilisé comme indication du gain maximal possible lorsqu'on augmente le taux de quantification. Ceci n'est qu'une indication car les hypothèses sur le signal sont très fortes et rarement vérifiées en pratique.

2.3.2 Quantification uniforme des sources non-uniformes

Pour introduire ce qui va suivre, considérons l'exemple d'une source émettant dans $[-100, 100]$ dont 95% des valeurs sont dans $[-1, 1]$. Supposons que l'on ait adopté la stratégie de quantification uniforme de la section précédente et que le codage soit effectué sur $k = 3$ (codage sur 3 bits), ce qui revient à considérer 8 intervalles de longueur 25. On a donc, dans 95% des cas une erreur minimum de 11,5. Cet exemple est représentatif des sources gaussiennes et montre le défaut de la quantification uniforme telle que présentée dans la section précédente.

Si l'on souhaite rester dans le cadre de la quantification uniforme, une solution consiste à utiliser la fonction de répartition pour optimiser la taille des intervalles considérés. Concrètement cela revient à considérer σ_q^2 comme une fonction de Δ et à résoudre :

$$\min_{\Delta} \sigma_q^2(\Delta).$$

Explicitons le début du calcul menant à la résolution de ce problème. On a :

$$\sigma_q^2(\Delta) = 2 \sum_{n=1}^{N/2-1} \int_{(n-1)\Delta}^{n\Delta} \left(x - \frac{2n-1}{2}\Delta\right)^2 f_S(x)\mathrm{d}x + 2\int_{(N/2-1)\Delta}^{+\infty} \left(x - \frac{N-1}{2}\Delta\right)^2 f_S(x)\mathrm{d}x.$$

Exercice 2.1 :

Montrer que :

$$\frac{\partial \sigma_q^2(\Delta)}{\partial \Delta} = -\sum_{n=1}^{N/2-1} (2n-1) \int_{(n-1)\Delta}^{n\Delta} \left(x - \frac{2n-1}{2}\Delta\right) f_S(x)\mathrm{d}x - (N-1)\int_{(N/2-1)\Delta}^{+\infty} \left(x - \frac{N-1}{2}\Delta\right) f_S(x)\mathrm{d}x.$$

Pour trouver un extremum de σ_q, il faut donc annuler cette dernière valeur. L'équation en découlant n'est pas résoluble algébriquement dans le cas général.

On la résout donc approximativement par des méthodes numériques. Il existe aussi des tableaux indiquant les solutions dans les cas de densités de probabilités classiques.

Remarque 2.3 :

Dans ce cas, on peut en fait distinguer deux types d'erreurs de quantification :

1. L'*erreur de surcharge*, qui correspond aux erreurs se produisant dans les deux intervalles extrêmes. On parle également de *bruit de saturation*.
2. L'*erreur granulaire*, qui correspond à l'erreur de quantification dans les autres intervalles. On parle alors de *bruit granulaire*.

Il arrive que l'on ne dispose que d'information partielles sur la source et sa densité de probabilité f_S. Cette carence conduit à des erreurs de modélisation. La fonction f_S^{approx} considérée ne coïncide pas avec la vraie fonction f_S. Il faut donc en fait prévoir une série de tests sur la source pour estimer ses paramètres.

2.4 Quantification adaptative

Une solution simple aux problèmes évoqués dans la section précédente consiste à fonder la stratégie de codage sur des intervalles de longueurs variables. On parle dans ce cas de *quantification adaptative*. Le but est bien sûr d'adapter les paramètres de quantification à la source considérée.

2.4.1 Approches *online* et *offline*

Deux approches sont généralement considérées, l'approche *online* conduisant à une adaptation de la quantification *rétrograde*, c'est-à-dire effectuée en même temps que la quantification elle-même et une adaptation *offline*, où les paramètres de la quantification sont calculés de manière *directe*, c'est-à-dire en fixant *a priori* les paramètres de la quantification.

2.4.2 Quantification adaptative directe – *offline*

Dans cette approche le signal émis est divisé en blocs temporels et chaque bloc est analysé avant la quantification. Les paramètres de la quantification sont calculés en fonction de l'analyse. On a donc, dans ce cas, besoin d'adjoindre à chaque bloc codé un bloc d'information supplémentaire permettant d'indiquer au décodeur les paramètres de quantification qui ont finalement été retenus. Deux problèmes apparaissent dans cette approche.

1. Un problème de synchronisation, car deux types de données sont transmis : le code du signal et les blocs d'informations.
2. Un problème de choix de la taille des bloc de codage considérés. Il faut alors trouver un compromis entre des bloc grands, qui seront alors plus grossièrement décrits par les paramètres de quantifications retenus, et des blocs petits, qui conduiront à de nombreux blocs d'information.

Donnons maintenant un exemple de procédure d'estimation des paramètres d'un bloc à quantifier. Étant donné un bloc de taille M, un estime sa variance [2] au voisinage du temps n par

$$\hat{\sigma}_x^2 = \frac{1}{M} \sum_{i=0}^{M-1} x_{i+n}^2,$$

où on a supposé que notre signal d'entrée avait une valeur moyenne nulle.

Pour quantifier le bloc considéré, une stratégie possible est alors de considérer une densité de probabilité f_S pour le bloc considéré du signal qui par exemple peut être une gaussienne avec pour variance la variance empirique précédemment calculée et utiliser la stratégie uniforme indiquée à la section 2.3.2.

Évidemment d'autres raffinements sont possible mais dépasse le cadre de cette introduction à la quantification.

Remarque 2.4 :

Attention, il faudra aussi quantifier les informations des blocs d'information, car tout doit être *in fine* sous forme binaire !

2.4.3 Quantification adaptative rétrograde – *online*

Dans cette second approche, l'adaptation se fait à la sortie du quantificateur. Elle ne nécessite pas de bloc d'information supplémentaire. Ici, seulement les valeurs quantifiés passées sont accessible pour adapter la quantification.

La méthode consiste à fixer le modèle, par exemple gaussien, et à adapter au cours du temps la valeur de Δ en observant les histogrammes issus de la quantification. Au bout d'un temps long, on aura de *bonnes* informations sur le signal, mais comment adapter la quantification avant ?

Évidemment, l'idée est de n'avoir ni un paramètre Δ trop petit, ni trop grand.

Nous allons présenter ici une idée de quantification adaptative qui n'utilise que la dernière valeur du signal passé. Il s'agit de la quantification de Jayant. L'idée derrière cet algorithme est très simple : pour un nombre fixé d'intervalles

2. Il s'agit en fait de ce qu'on appelle en probabilité la variance empirique.

N, on considère des multiplicateurs M_k pour chaque intervalle. On numérotera de 0 à $N/2-1$ les intervalles positifs, avec la relation

$$M_k = M_{k+N/2},$$

pour tout k entre 0 et $N/2-1$. On aura donc deux intervalles pour $k = N/2-1$ et $k = N-1$ qui seront de la forme $[b_k, +\infty[$ et $]-\infty, -b_k]$ et qui sont des intervalles en dehors de la zone de quantification.

Partant de là, si la valeur à quantifier est dans les intervalles en dehors de la zone de quantification alors, le paramètre Δ doit être augmenter (pour tenter de faire rentrer la valeur dans la zone à l'intérieur de la quantification). Au contraire, si la valeur à quantifier est à l'intérieur de la zone de quantification, alors on peut essayer de diminuer la valeur de Δ.

On aura alors que pour les intervalles *internes*, $M_k \leq 1$ et pour les deux intervalles *externes* $M_k > 1$. L'algorithme est alors :

- la n-ième valeur est dans l'intervalle k,
- le paramètre de quantification est mise à jour par $\Delta_n = M_k * \Delta_{n-1}$,
- et les intervalles $(I_k^n)_{k\in[0,N-1]}$ sont recalculés avec le nouveau Δ_n.

Exercice 2.2 :

Appliquer l'algorithme pour :

- $M_0 = M_4 = 0.8$, $M_1 = M_5 = 0.9$, $M_2 = M_6 = 1$ et $M_3 = M_7 = 1.2$;
- la valeur initiale du paramètre de quantification $\Delta_0 = 0.5$;
- la séquence à quantifier $\{0.1, -0.2, 0.2, 0.1, -0.3, 0.1, 0.2, 0.5, 0.9, 1.5\}$.

Le choix des multiplicateurs est un choix crucial pour cette algorithme. Il est assez libre, mais plusieurs méthodes ont été proposées pour le faire. Nous renvoyons à [8] pour une description d'une d'entre elles.

2.5 Quantification non-uniforme

La dernière solution envisagée habituellement consiste à postuler une densité de probabilité, à considérer ensuite σ_q comme une fonction des $2N+1$ variables réelles contenues dans les variables Y et B et finalement à optimiser σ_q globalement. La taille des intervalles de quantification $\Delta_i = b_{i+1} - b_i$ n'est alors plus constant.

Nous présentons dans cette section un algorithme connu sous le nom d'algorithme de Lloyd ou de Lloyd-Max. Cette méthode a initialement été développée pour les problèmes de quantification (en 1957) mais a désormais des domaines d'application très variés notamment pour calculer le diagramme de Voronoï de k points, pour la résolution d'EDP, ou même désormais en *machine-learning*.

Un calcul de différentielle partielle de (2.2) conduit à l'algorithme suivant.

Algorithm 1 Algorithme de Lloyd-Max

Require : $b_1^0, \dots, b_{N-1}^0$, $y_1^0, \dots, y_N^0$ et f_S

1 : Pour tout i, $b_0^i = -\infty$, $b_N^i = +\infty$

2 : $i = 0$

3 : **repeat**

4 : Calculer :

$$y_n^{i+1} = \frac{\int_{b_{n-1}^i}^{b_n^i} x f_S(x) \mathrm{d}x}{\int_{b_{n-1}^i}^{b_n^i} f_S(x) \mathrm{d}x}, \quad \forall n \in \{1, \dots, N\}$$

$$b_n^{i+1} = \frac{y_{n+1}^{i+1} + y_n^{i+1}}{2}, \quad \forall n \in \{1, \dots, N-1\}$$

5 : **until** Convergence

avec $b_0 = -\infty$ et $b_N = +\infty$. Celles-ci permettent de définir un algorithme de point fixe. En effet, le calcul des y_n dépend des b_n, et le calcul des b_n dépend des y_n. On peut montrer rigoureusement que l'algorithme converge.

La méthode est donc d'itérer le calcul des jeux de variables (y_n) et (b_n) jusqu'à convergence.

De même que ce qu'à la section 2.3.2, ce système n'est, dans le cas général, par résoluble algébriquement. Il faut donc mettre en oeuvre une méthode numérique pour le résoudre approximativement.

Une preuve de convergence de l'algorithme de Lloyd peut se lire ici [11], mais a été largement étendue depuis.

2.6 Note bibliographique

Pour plus de détails sur les différentes techniques de quantification des signaux, on consultera le chapitre 9 de la référence [8].