

S1916 - Analyse et Compression du Signal Audionumérique - Examen

Sébastien Boisgérault, Mines ParisTech

Jeudi 19 Mars 2015

Table des Matières

Modalités de l'Examen	1
Audio Haute Résolution	2
Données Binaires OpenWRT	2
Codage des Nombres Entiers	2
Codage de l'ADN	3
Nonlinéarité, Distorsion et Intermodulation	3
Interpolation Linéaire	4
To B. or not to B.	5
Correction de Tonalité	5
Prédiction Linéaire – Ordre et Précision	6
Analyse de la Voix	6
Convolution(s)	6
Prediction Linéaire - Covariance et Autocorrélation	7

Modalités de l'Examen

- **Durée:** 1h30.
- **Autorisés:** documents (papier, électronique) et calculette (ou équivalent),
- **Interdits:** interpréteur Python et accès Internet.

Audio Haute Résolution

Récemment, plusieurs acteurs économiques ont utilisé le terme “Audio Haute Résolution” pour désigner des données audio-numériques non compressées de type 96 kHz / 24 bit (fréquence d’échantillonnage / profondeur de bits), voire plus précises.

- Dans quelle proportion ce type de format occupe-t’il plus d’espace que le CD audio (CD-DA) ? Qu’un fichier MP3 avec un débit typique ?
- Un auditeur est-il a priori à même de percevoir les bénéfices d’une fréquence d’échantillonnage aussi élevée ? De l’utilisation de 24 bits pour coder chaque échantillon ? Argumenter. Quel peut être l’intérêt de ce type de formats ?

Sources:

- Sony: <http://discover.store.sony.com/High-Resolution-Audio>
- Pono: <https://ponomusic.force.com>

Données Binaires OpenWRT

La sequence binaire

11011110101011011100000011011110

est utilisée dans OpenWRT (une distribution Linux pour les systèmes embarqués) comme un marqueur pour signaler le début d’une zone destinée à un fichier journalisé de type jffs2.

- Convertir cette séquence en hexadécimal.

Source: [Wikipedia – Magic Number](#).

Codage des Nombres Entiers

Montrer qu’il existe un code binaire sans préfixe c associant à tout entier naturel $n \in \mathbb{N}$ une suite de bits de longueur $n + 1$.

Peut-on trouver un autre code binaire sans préfixe pour lequel la longueur de l’une de ces suites de bits est raccourcie (les autres conservant la même longueur) ?

Codage de l'ADN

L'ADN est une macro-molécule composée de quatre nucléotides différents notés A (Adénine), G (Guanine), C (Cytosine) et T (Thymine). L'ADN a une structure de double hélice; sur chaque brin, chaque nucléotide fait face à un nucléotide complémentaire (A à T, T à A, C à G, G à C). Ces règles d'appariement imposent que sur un génome complet, la fréquence d'apparition des nucléotides A et T d'une part, et C et G d'autre part, soient égales.

Chez les bactéries de type *Streptomyces*, on a

$$f_A + f_T = 0.28 \text{ et } f_G + f_C = 0.72.$$

- Quelle information apporte a priori la lecture d'un nucléotide dans une séquence ? Combien de bits seraient nécessaires pour décrire un nucléotide avec un code de longueur fixe ?
- Proposer un code binaire auto-délimitant pour la description de séquences de nucléotides qui soit plus compact qu'un code binaire de longueur fixe (on calculera la longueur moyenne du code binaire représentant un nucléotide).

Chez l'*Homo Sapiens*, on a

$$f_A + f_T = 0.58 \text{ et } f_G + f_C = 0.42.$$

- Donner une borne inférieure de la longueur moyenne du code auto-délimitant optimal associé à ces statistiques. Peut-on faire mieux qu'un code de longueur fixe ?
- Quelle hypothèse sous-tend depuis le début nos calculs de quantité d'information présente dans les fragments ADN ? Suggérer une étude statistique de l'ADN d'une espèce qui permette de concevoir des codes plus compacts.

Source: [Cours de Biologie Computationelle, Equipe "Biocomputing and Structure", Laboratoire de Biochimie - UMR 7654](#)

Nonlinéarité, Distorsion et Intermodulation

Les dispositifs permettant la restitution des sons digitaux (amplificateurs, enceintes, etc.) ne sont pas strictement linéaires. Dans cet exercice, on modélise cet effet non-linéaire en général indésirable par une transformation en bout de chaîne, de la forme

$$S : x(t) \in [-1.0, +1.0] \mapsto y(t) = x(t) - \frac{x(t)^3}{3} \in [-1.0, +1.0].$$

On suppose que $\Delta f = 44.1$ kHz, $f = 7040$ Hz, et

$$x(t) = \cos 2\pi ft, \quad t \in \mathbb{Z}/\Delta f.$$

- Déterminer $y(t)$ (Indication: on pourra poser $z = \exp(i2\pi ft)$).
- Déterminer le rapport signal sur bruit (en dB) associé à la transformation S pour le signal $x(t)$.
- Est-ce que tous les composants fréquentiels du bruit induit par la transformation S sont perçus par l'auditeur ? Calculer si nécessaire le rapport signal sur bruit "effectif" qui tient compte de cette analyse.

On suppose désormais que $\Delta f = 192$ kHz, $f = 24$ kHz, $f' = 56$ kHz et

$$x(t) = 0.5 (\cos 2\pi ft + \cos 2\pi f't), \quad t \in \mathbb{Z}/\Delta f.$$

- Montrer que $y(t)$ possède une composante à 8 kHz dont on calculera le niveau sonore (en dB) (Indication: on pourra poser $z = \exp(i2\pi ft)$ et $w = \exp(i2\pi f't)$).
- Le signal $x(t)$ est-il audible ? Et le signal $y(t)$?

Source: "192 kHz considered harmful" sur xiph.org

Interpolation Linéaire

A partir d'un signal à temps discret $x(t)$ de période d'échantillonnage Δt , il est possible de construire un signal $y(t)$ de période d'échantillonnage $\Delta t/2$ par interpolation linéaire:

$$y(t) = \begin{cases} x(t) & \text{si } t = k\Delta t, k \in \mathbb{Z}, \\ x(t - \Delta t/2)/2 + x(t + \Delta t/2)/2 & \text{si } t = (2k + 1)\Delta t/2, k \in \mathbb{Z}. \end{cases}$$

1. Déterminer la représentation fréquentielle $y(f)$ du signal interpolé en fonction de celle $x(f)$ du signal original (remarque: sans la réponse à cette question, il est possible d'admettre les résultats de la question 2. et de poursuivre l'exercice).
2. Montrer que cette interpolation linéaire peut être réalisée au moyen d'une expansion d'ordre 2 (insertion d'un zéro entre deux données consécutives) suivie de l'application d'un filtre dont la réponse fréquentielle est

$$h(f) = \frac{1}{2} \left(1 + \cos \left(2\pi f \frac{\Delta t}{2} \right) \right)$$

puis d'une amplification d'un facteur 2.

3. Le filtre de réponse fréquentielle $h(f)$ se comporte-t'il plutôt comme un filtre passe-bas ou comme un filtre passe-haut ?
4. Quelle est la fonction de transfert $h(z)$ du filtre ? Sa réponse impulsionnelle $h(t)$? Est-ce un filtre FIR (à réponse impulsionnelle finie) ou un filtre AR (auto-régressif) ? Est-il strictement causal ? causal ?

To B. or not to B.

Monsieur B. habite à 200 m d'une école de commerce dont les fêtes sont quelque peu bruyantes. La réglementation française limite le niveau sonore des discothèques à 120 dB (en niveau de crête) ; on supposera que cette mesure est faite à 10 m des enceintes, et l'on admettra également que la réglementation est respectée.

L'OMS préconise un niveau sonore maximal de 30 dB dans la chambre à coucher. Monsieur B. dispose heureusement d'un double vitrage qui atténue les sons de 30 dB. Est-ce que vous lui conseillez d'utiliser en plus des bouchons d'oreille anti-bruit (atténuation de 30 dB) pour dormir ? De déménager ?

(Indication: on suppose que la propagation du son est isotrope et que l'énergie acoustique se conserve. L'intensité sonore – en W/m^2 – décroît alors comme le carré de la distance).

Sources:

- [Le Parisien – Discothèques, l'étude alarmiste](#)
- [Bruitparif – Bruit et troubles du sommeil](#)

Correction de Tonalité

En explorant le code source d'un logiciel de traitement audio, vous tombez sur:

```
def tune(f):
    "f: frequency in Hz"
    n = round_(12.0 * log2(f / 27.5))
    return 27.5 * 2 ** (n / 12.0)
```

- Montrer que cette fonction implémente un quantificateur.
- Quelles sont les valeurs représentées exactement par le quantificateur ? Dans le contexte musical, à quoi ce quantificateur peut-il être utile ?
- Déterminer (une approximation de) l'écart quadratique moyen entre la fréquence d'entrée et de sortie de cette fonction lorsque la fréquence d'entrée est une valeur aléatoire de probabilité uniforme entre 2000 et 4000 Hz.

Prédiction Linéaire – Ordre et Précision

Lorsque l'on effectue une prédiction linéaire d'un signal, diminue-t'on nécessairement de façon significative l'énergie du résidu en augmentant l'ordre de la prédiction ?

Analyse de la Voix

Proposer une méthode permettant de distinguer deux signaux de voix parlée – une voix de femme et d'homme – en réutilisant les composants déjà implémentés d'un codeur de voix LPC classique.

Convolution(s)

Une librairie open-source entreprend d'optimiser le code de NumPy qui calcule la convolution de deux tableaux. Selon les dires des développeurs:

The NumPy code `c = convolve(a,b)` relies on the formula:

$$c[n] = \sum_m (a[m] * b[n-m])$$

We implement a faster version of `convolve` named `fast_convolve`.
For the sake of simplicity, we assume `len(a) == len(b)`.

En fait, la fonction `fast_convolve` semble avoir connu trois versions différentes:

```
def fast_convolve(a, b):
    assert len(a) == len(b)
    return ifft(fft(a) * fft(b))

def fast_convolve(a, b):
    assert len(a) == len(b)
    pad = zeros(len(a) - 1)
    fft_a = fft(r_[a, pad])
    fft_b = fft(r_[b, pad])
    return ifft(fft_a * fft_b)

def fast_convolve(a, b):
    assert len(a) == len(b)
    N = len(a) + len(b) - 1
    M = 2 ** ceil(log2(N))
    pad = zeros(M - len(a))
    fft_a = fft(r_[a, pad])
    fft_b = fft(r_[b, pad])
    return ifft(fft_a * fft_b)[:N]
```

Il n'y a malheureusement aucun commentaire ou documentation qui explique la raison d'être de ces différentes versions. Pouvez-vous expliquer l'évolution de ce code source en déterminant les avantages, limitations et problèmes liés aux différentes versions ?

Prediction Linéaire - Covariance et Autocorrélation

On considère la sequence $x_i, i \in \mathbb{Z}$, définie par

$$x_i = \begin{cases} 0 & \text{si } i < 0, \\ 1 & \text{si } i \in \{0, 1\} \\ 0 & \text{si } i > 1. \end{cases}$$

- Déterminer les coefficients $a_i, i \in \mathbb{N}^*$, de la prédiction linéaire à l'ordre $m = 1$ des x_i par les méthodes de covariance, puis d'autocorrélation.
- Déterminer les coefficients $a_i, i \in \mathbb{N}^*$, de la prédiction linéaire à l'ordre $m = 2$ par la méthode d'autocorrélation. Calculer la prédiction $\hat{x}_i, i \in \mathbb{Z}$, et le résidu $e_i, i \in \mathbb{Z}$, associés au signal x_i par cette méthode.