

# Traitement du signal multimédia : fréquences et linéarisation

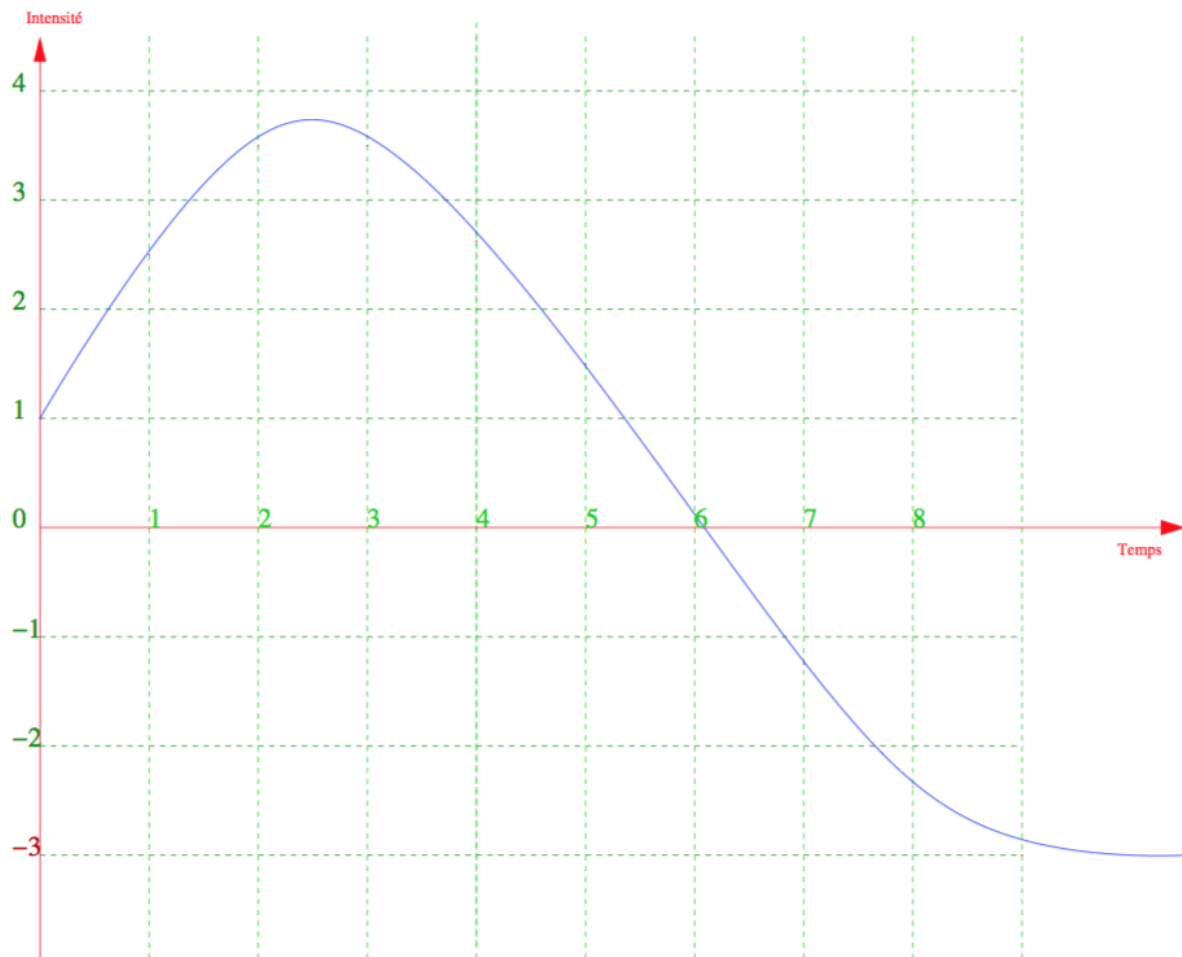
OIM - TD1

2014

## 1 Codage d'un signal

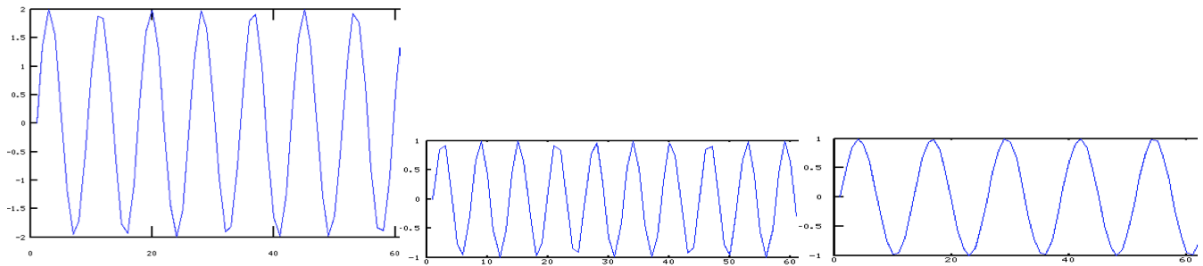
### 1.1 Transformation analogique-numérique

Codez en binaire le signal analogique représenté ci-dessous sur 3 bits en utilisant le quadrillage pour déterminer les valeurs à conserver.

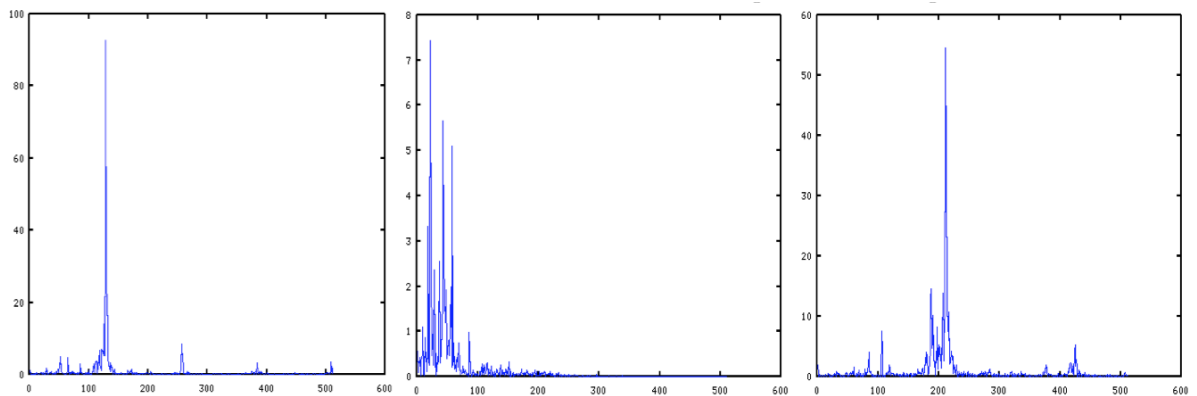


## 2 Analyse de signal

1. **Ordonnancement des signaux** : les figures ci-dessous représentent l'amplitude du signal au cours du temps. Classez les signaux du plus grave au plus aigü.



2. **Ordonnement des spectres** : classez les spectres suivants du plus grave au plus aigü.



3. **Ordonnement de fonctions.**

- Soient les deux sinusoides  $f_1(x) = \cos(2\pi x)$  et  $f_2(x) = \cos(4\pi x)$ . Laquelle à la plus grande fréquence ?
- Quel est le signal avec la plus basse fréquence possible ?
- Quel est le signal avec la plus haute fréquence possible ?

## 3 Traitement d'images

### 3.1 Codage d'une image

Soit une image en niveau de gris de taille  $128 \times 128$  pixels, codés sur 8 bits.

- Quelle est la taille de cette image en kilo-octets.
- Si on code les pixels sur 7 bits ou sur 2 bits, quelles sont les tailles en kilo-octets correspondantes.
- Pour une image codée sur 2 bits par pixel, que doit-on constater visuellement ?

### 3.2 Analyse d'une image

Soit l'image suivante, de résolution  $5 \times 5$  pixels, codée sur 8 bits par pixel.

1	2	1	2	1
0	120	0	120	0
0	120	120	120	0
0	120	0	120	0
1	2	1	2	1

- Binariser l'image en seuillant la valeur des pixels à 128. Que remarquez-vous ?
- Tracez l'histogramme des niveaux de gris de cette image.
- Binariser l'image en tenant compte de la distribution des valeurs des pixels sur l'histogramme

4. Tracez une courbe montrant l'évolution de la valeur des pixels sur la première ligne. Est-ce une zone de haute fréquences ou de basse fréquences.
5. Tracez une courbe montrant l'évolution de la valeur des pixels sur la troisième ligne. où sont les zones de haute fréquences et de basse fréquences ?
6. En déduire une façon de détecter le **H** dans l'image.

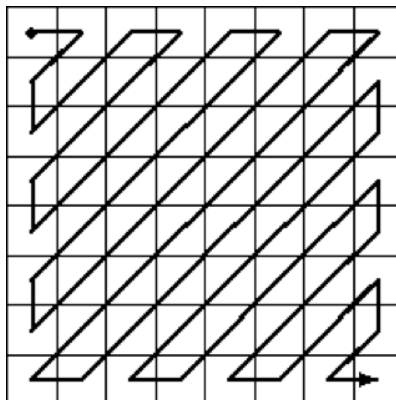
### 3.3 Linéarisation d'une image

Nous considérons qu'une image 2D est représentée par le type de données C suivant :

Code 1 – Définition du type C pour une image 2D

```
typedef struct image_s {
    int width;
    int height;
    unsigned char **pixels; /* 2D array of gray level */
} image_t;
```

On souhaite transformer cette image en un vecteur construit par linéarisation de blocs de pixels de taille  $8 \times 8$ . Chaque bloc sera linéarisé en ordonnant ses pixels selon le schéma suivant :



1. Définir un itérateur séquentiel pour un tableau 2D sous forme d'une table état/transition permettant de reproduire ce parcours.
2. Définir la fonction *C* bloc2vector permettant de transformer un bloc de pixel en un vecteur.
3. Définir la fonction *C* picture2blocvector permettant de transformer une image en un vecteur linéarisé par bloc.