

Génération et affichage de signaux

2.1 Objectif du TP

Le traitement du signal consiste à : étudier et analyser le signal, en extraire les informations pertinentes (ex : signaux radars) ; modifier le signal (ex : enlever les parasites d'un son ou éclaircir une image) et reproduire des signaux nouveaux (ex : voix artificielles). La plupart de ces signaux sont analogiques par nature, c'est-à-dire qu'ils sont fonction d'une variable continue, le temps, et qu'eux-mêmes varient de manière continue. Ces signaux peuvent être aussi traité et analysé sous forme numérique, à l'aide de convertisseurs analogiques-numériques (échantillonnage, quantification et numérisation des signaux analogiques).

L'objectif de ce TP est d'apprendre à **générer, visualiser et analyser** sous MatLab quelques signaux analogiques et échantillonnées. Ainsi que, de mettre en application les connaissances acquises sur l'**échantillonnage**, notamment le phénomène de recouvrement « *Aliasing* ».

2.2 Rappels

2.2.1 Définition

Un *signal* est la représentation physique d'une information à transmettre. C'est une expression d'un phénomène qui peut être mesurable par un appareil de mesure. Par exemple : les ondes acoustiques (courant délivré par un microphone (parole, musique, ...), les tensions aux bornes d'un condensateur en charge, les signaux géophysiques (vibrations sismiques), ... etc.

Le traitement du signal est un ensemble de techniques permettant de créer, d'analyser, de transformer les signaux en vue de leur exploitation. Ainsi que, d'extraction du maximum d'information utile d'un signal perturbé par le bruit.

Les principales fonctions du traitement de signal sont :

- **Synthèse** : création de signaux par combinaison de signaux élémentaires.
- **Modulation** : adaptation du signal au canal de transmission.
- **Détection** : isoler les composantes utiles d'un signal complexe, extraction du signal d'un bruit de fond.
- **Identification** : classement du signal (reconnaissance de la parole,...etc.).
- **Filtrage** : élimination de certaines composantes. Par exemple : annulation d'écho, détection de bruit sur une image,... etc.
- **Codage/compression** (image Jpeg, mp3, mpeg4, etc.)

2.2.2 Classification des signaux

Classification dimensionnelle

On peut classer les signaux suivant leurs dimensions :

- Signaux 1D : $x(t)$ est un signal qui évolue en fonction d'un seul paramètre, pas forcément le temps. Par exemple : signal reçu par un microphone monophonique.
- Signaux 2D : $I(x, y)$. Par exemple : image, signal reçu par un microphone stéréophonique.
- Signaux 3D : $I(x, y, z)$. Par exemple : séquence d'images 2D.
- Signaux 4D ou 3D + t : Par exemple : séquence d'images 3D.
- Signaux N-D : des signaux sont typiquement reçus par un réseau de capteurs. Par exemple : réseaux d'antennes, ou de microphones.

Classification phénoménologique

On peut aussi classer les signaux suivant leurs caractères. Il peut être à caractère *certain* (ou déterministe) ou *aléatoire* (ou stochastique).

Signal certain : est un signal dont l'évolution en fonction du temps t est décrit par un modèle mathématique.

Signal réel : est un signal qui représentant une grandeur physique. Son modèle mathématique est une fonction réelle. Par exemple : la tension aux bornes d'un condensateur.

Signal aléatoire : est un signal dont l'évolution temporelle est imprévisible et dont on ne peut pas prédire la valeur à un temps t . La description de ces signaux est basée sur les propriétés statistiques des signaux (moyenne, variance, loi de probabilité, ...)

Classification énergétique

Dans cette classification, nous distinguons les signaux qui satisfaisant la condition d'énergie finie à ceux qui présentant une puissance moyenne finie et une énergie infinie.

On appellera énergie d'un signal $x(t)$ la quantité :

$$E = \int_{-\infty}^{+\infty} |x(t)|^2 dt$$

et puissance moyenne de $x(t)$ la quantité :

$$P = \lim_{T_o \rightarrow \infty} \int_{-T_o/2}^{+T_o/2} |x(t)|^2 dt$$

Les *signaux à énergie finie* ont une puissance moyenne nulle. Généralement, ces signaux représentant une grandeur physique.

Les *signaux à énergie infinie* ont une puissance moyenne non nulle. Comme le cas des signaux périodiques

Classification morphologique

Dans cette classification, nous distinguons quatre types de signaux suivant la variable t (continu ou discret) :

- *Signal analogique* : le signal à amplitude et temps continus.
- *Signal quantifié* : le signal à amplitude discret et temps continu.
- *Signal échantillonné* : le signal à amplitude continue et temps discret.
- *Signal numérique* : le signal à amplitude discret et temps discret.

Classification spectrale

Les signaux sont aussi classer suivant leur répartition énergétique en fonction de la fréquence :

- Signaux de basses fréquences.
- Signaux de hautes fréquences.
- Signaux à bande étroite.
- Signaux à large bande.

2.2.3 Autocorrélation et intercorrélation

Fonction d'autocorrélation

L'autocorrélation réalise une comparaison entre un signal $x(t)$ et ses copies retardées :

Pour les signaux à énergie finie : $C_{xx}(\tau) = \int_{-\infty}^{\infty} x(t) \cdot x^*(t - \tau) dt$

$\Rightarrow$ Homogène à une énergie, $C_{xx}(0)$ est l'énergie du signal.

Pour les signaux à énergie infinie : $C_{xx}(\tau) = \lim_{T \rightarrow \infty} \int_{-T/2}^{+T/2} x(t) \cdot x^*(t - \tau) dt$

$\Rightarrow$ Homogène à une puissance, $C_{xx}(0)$ est la puissance moyenne du signal

Propriétés

1. $C_{xx}(\tau) \leq C_{xx}(0)$: Maximum en 0.
2. Si $x(t)$ est périodique alors $C_{xx}(t)$ est périodique de même période.
3. Pour des signaux réels $C_{xx}(t)$ est paire.

Fonction d'intercorrélation

L'intercorrélation, noté $C_{xy}(\tau)$, compare deux signaux $x(t)$ et $y(t)$ retardée. Cette fonction permet donc de mesurer la similitude entre ces deux signaux.

Pour les signaux à énergie finie : $C_{xy}(\tau) = \int_{-\infty}^{\infty} x(t) \cdot y^*(t - \tau) dt$

Pour les signaux à énergie infinie : $C_{xy}(\tau) = \lim_{T \rightarrow \infty} \int_{-T/2}^{+T/2} x(t) \cdot y^*(t - \tau) dt$

2.2.4 Quelques signaux élémentaires

Impulsion de Dirac

L'impulsion de Dirac $\delta(t)$, aussi appelée impulsion unité ou distribution delta, est définie par le produit scalaire :

$$x(t_o) = \int_{-\infty}^{\infty} x(t) \delta(t - t_o) dt$$

en particulier, en posant $x(t) = 1$, on obtient : $\int_{-\infty}^{\infty} \delta(t) dt = 1$

Signal signe

$$\text{sgn}(t) = \begin{cases} -1 & \text{Si } t < 0 \\ +1 & \text{Si } t > 0 \end{cases}$$

Signal saut unité (ou Echelon)

$$\Gamma(t) = 1/2 + 1/2 \text{sgn}(t) = \begin{cases} 0 & \text{Si } t < 0 \\ 1 & \text{Si } t > 0 \end{cases}$$

Signal rampe

$$r(t) = \int_{-\infty}^t \Gamma(\tau) d\tau = t\Gamma(t) \Rightarrow \Gamma(t) = \partial r(t)/\partial t \text{ pour } : t \neq 0$$

Signal porte ou rectangle

$$\text{rect}(t) = \Gamma(t + 1/2) - \Gamma(t - 1/2) = \begin{cases} 1 & \text{Si } |t| < 1/2 \\ 0 & \text{Si } |t| > 1/2 \end{cases}$$

Signal triangulaire

$$\text{tri}(t) = \begin{cases} 1 - |t| & \text{Si } |t| \leq 1 \\ 0 & \text{Si } |t| \geq 1 \end{cases}$$

Exponentielle décroissante

$$y(t) = \Gamma(t) \cdot e^{-at} \text{ Si } a > 0$$

2.2.5 Théorème de l'échantillonnage

L'échantillonnage est une étape importante dans l'analyse du signal. Cette opération consiste à convertir un signal continu en un signal discret (c'est-à-dire prélever un échantillon du signal à temps continu tous les instants).

Pour un signal continu $x(t)$, qu'on échantillonne à toutes les T_s secondes, on obtient un signal discret $x(n)$:

$$x(n) = x(t = nT_s) \quad n = 0, 1, 2, \dots, N$$

La conversion du signal à temps continu au temps discret entraîne en général à une perte importante d'informations et à des erreurs lors de l'analyse. Pour l'éviter, le *théorème* suivant doit être respectée :

Théorème de Shannon (ou de Nyquist) : Pour un signal analogique de fréquence maximale f_{max} , la fréquence d'échantillonnage f_s doit être plus grande que $2f_{max}$. $f_s \geq 2f_{max}$.

Remarque : Le spectre d'un signal échantillonné est la somme d'une répétition périodique du spectre du signal analogique $X(f)$ en tous les multiples de la fréquence d'échantillonnage f_s .

2.3 Manipulations

2.3.1 Génération et visualisation de signaux

1. Quel est le type de signal générer par les scripts MatLab suivant :

Script 1

```
t=-10:1:20;
Imp=[zeros(1,10),ones(1,1),zeros(1,20)];
stem(t,Imp);
xlabel('Temps indexé en n');
ylabel('Amplitude');
```

Script 2

```
t = -1:1e-5:1 ;
x1 = rectpuls(t, 0.05);
plot(t,x1) ;
axis([-0.1 0.1 -0.2 1.2]);
xlabel('Temps(sec)');
ylabel('Amplitude');
```

Script 3

```
t = -1:1e-5:1 ;
x2 = tripuls(t,0.04);
plot(t,x2);
axis([-0.1 0.1 -0.2 1.2]);
xlabel('Temps (sec)');
ylabel('Amplitude');
```

Script 4

```
fs = 10000 ;
t = 0:1/fs:1.5;
y1 = sawtooth(2*pi*50*t);
plot(t,y1);
axis([0 0.1 -1.2 1.2]);
xlabel('Temps (sec)');
ylabel('Amplitude');
```

2. Ecrire des fonctions MatLab permettent de générer des signaux ci-dessous décalée de t_0 , d'amplitude A et de longueur arbitraire N .

- (a) Une impulsion unité $Y = \text{imp}(t) = A \cdot \delta(t - t_0)$
- (b) Une fenêtre Rectangulaire $Y = A \cdot \text{Rec}(t - t_0)$
- (c) Une fenêtre Triangulaire $Y = A \cdot \text{Tri}(t - t_0)$

3. Ecrire un script **MatLab**, faisant appel aux fonctions précédentes pour générer et représenter graphiquement chacune des séquences $x_i(n)$ suivantes :

$$\begin{aligned} x_1(n) &= 1.2 * \delta(n - 3) & -20 \leq n \leq 20 \\ x_2(n) &= 1.8 * \text{Rec}(t - 20) & N = 50 \\ x_3(n) &= 3.2 * \text{Tri}(t - 35) & N = 25 \end{aligned}$$

4. Ecrire une fonction **MatLab** qui permet de générer une sinusoïde de longueur finie incluse dans l'intervalle $[t_1, t_2]$ avec les paramètres suivant : A : Amplitude, f : fréquence et Φ phase initiale.
5. Ecrire un script **MatLab**, faisant appel à la fonction précédente pour générer et représenter graphiquement chacune des signaux suivants :

$$\begin{aligned} x_1(t) &= 4 \cos(4\pi t/4) & t \leq 2s \\ x_2(t) &= 3 \cos(4t) + \sin(\pi t) & t \leq 20s \end{aligned}$$

6. Ecrire un script **MatLab** qui permet de générer le signal complexe suivant : $z(n) = e^{(jn/2)}$ pour : $0 < n < 50$.
7. Représenter graphiquement le signal $z(n)$ avec les commandes : `plot(z)`, `plot(n,z)` et `stem(z)`. Qu'observez-vous ?

2.3.2 Échantillonnage des signaux analogiques

1. Ecrire un script **MatLab** qui permet de générer et représenter un signal sinusoïdal $x(t) = A \sin(2\pi f t)$ d'amplitude $A = 1$, des fréquences maximums $f = 5 \text{ Hz}$, 30 Hz et 80 Hz et échantillonnés à 100 Hz . Le vecteur de temps contient les instants des échantillons avec une période d'échantillonnage T_e , $t = (0 : N) * T_e$, (N : nombre total d'échantillons).
2. Représenter le module de la transformée de Fourier discret TFD du signal $x(t)$.
3. Le théorème d'échantillonnage est-il respecté ? Que concluez-vous ?
4. Ecrire un script **MatLab** qui permet de générer et représenter un signal sinusoïdal d'amplitude $A = 1$, de fréquence maximum $f = 1000 \text{ Hz}$ et échantillonnés respectivement à 20000 Hz , 5000 Hz et 1500 Hz .
5. Le théorème d'échantillonnage est-il respecté ? Que concluez-vous ?
6. Ecrire un script **MatLab** qui permet de générer et représenter graphiquement un signal :
 - Carré de durée 1.5 secondes , de fréquence $f = 180 \text{ Hz}$ et échantillonné à une fréquence de 10 kHz .

- Dent de scie de durée 1.5 *secondes*, de fréquence $f = 180 \text{ Hz}$ et échantillonné à une fréquence de 10 *kHz*.
- Représenter le module de la transformée de Fourier discret *TFD* de chaque signal. Qu'observez-vous ?