

# L3 - Signaux et systèmes continus

## TP1 - Signaux, transformée de Fourier, filtrage

Rémi Flamary

Ce TP est une initiation au traitement du signal sous Matlab. Bien que le cours porte sur l'étude de signaux continus, vous travaillerez sur des signaux discrets. Le but de ce TP est de vous apprendre à manipuler des signaux et à les traiter en appliquant un filtrage

### 1 Représentation temporelle

1. Créer un vecteur  $t$  contenant le temps dans l'intervalle  $[-5, 5]$ . Le vecteur de temps est initialisé par un commande de type `[t=linspace(-5,5,N)]`. On prendra  $N = 1024$ , ce qui implique une fréquence d'échantillonnage de  $f_e = N/10$ .
2. Tracer les signaux suivants vus en TD :
  - (a)  $x_1(t) = \Pi_T(t)$  pour  $T = 1$
  - (b)  $x_2(t) = (-t - 3)\Gamma(t - 2)\Gamma(-t + 3)$
  - (c)  $x_3(t) = e^{-at}\Gamma(t)$
  - (d)  $x_4(t) = \cos(2\pi f_0 t)$  avec  $f_0 = 1$

On utilisera `[figure]` pour ouvrir des figures et `[plot]` pour tracer les signaux.

3. Visualiser les signaux avec différentes transformations (décalage, changement d'échelle).

### 2 Représentation spectrale

Pour les signaux tracés précédemment,.

1. Créer le vecteur des fréquences `[f=linspace(-fe/2,fe/2,N)]`.
2. Calculer la transformée de Fourier avec `[fftshift(fft(x))/N]`.
3. Visualiser le module de la transformée de Fourier en fonction de la fréquence.

### 3 Filtrage idéal analogique

1. On travaille maintenant sur le signal (d) de la partie 1.
2. Calculer la  $X(f)$  transformée de Fourier de  $x(t)$  et initialiser le vecteur de fréquence  $f$  comme on l'a vu précédemment.
3. Créer le vecteur  $H(f)$  correspondant à la transformée de Fourier du filtre passe bas idéal de fréquence de coupure  $f_c = 10$ .
4. Filtrer le signal  $X$  dans l'espace de Fourier par le filtre passe bas idéal  $H$  pour obtenir la transformée de Fourier  $Y(f)$ . Visualiser  $X(f), H(f), Y(f)$ .
5. Inverser la transformée de Fourier avec `[real(iff(fftshift(x))*length(x))]`.
6. Tracer le signal filtré  $y(t)$  pour différentes fréquences de coupure (0.3, 10). Vérifier que le filtre est idéal, qu'il laisse passer exactement ce qui est dans la bande passante.
7. Refaire les étapes précédentes sur un signal  $x(t)$  bruité par un bruit Gaussien (`[randn]`).

## 4 Signal réel

On veut maintenant d'ébruiter un signal audio qui a été bruité lors de l'acquisition par un bruit haute fréquence de forte amplitude. Ceci se fera dans un autre fichier de script.

1. Écouter le fichier **stairwayb.wav**. On devine difficilement le signal audio d'origine.
2. Charger en mémoire le signal réel à partir du fichier wave **stairwayb.wav** en utilisant la fonction `[wavread]`. récupérer également la fréquence d'acquisition  $F_s$  et la longueur du signal `[N=length(x)]`.
3. Calculer la transformée de Fourier du signal sonore et visualiser son module. D'après vous à partir de quelle fréquence le signal est fortement bruité ?
4. Appliquer un filtre analogique idéal passe bas pour différentes fréquences de coupure au signal puis effectuer sa transformée de Fourier inverse (vu dans la section précédente). Sauvegarder ensuite le signal filtré dans un fichier wav pour pouvoir écouter le résultat (`[wavwrite]`).
5. Pour quelle fréquence de coupure le signal est le mieux reconstruit ?

## 5 Applications audio

Dans cette section on vous demande d'appliquer quelques effets audio aux fichiers wave. Sauvegardez le résultat de chaque effet dans un fichier wav différent pour pouvoir comparer.

### 1. Echo

- (a) Écouter et charger en mémoire le fichier **inigo.wav**.
- (b) L'écho peut être obtenu en convoluant une réponse impulsionnelle  $h(n) = \delta(n) + 1/2\delta(n-k)$  avec la fonction `[filter(h,1,x)]`. Créer la réponse impulsionnelle  $h$ , et écouter le résultat pour un écho à 1 seconde ( $k = 16000$ ).
- (c) Tester d'autres types d'échos (multiples, différents délais).

### 2. Séparation de source

- (a) Écouter et charger en mémoire le fichier **batterie.wav**.
- (b) Filtrer le signal (passe-bas) pour ne conserver que la grosse caisse.
- (c) Soustraire le signal de la grosse caisse au signal original pour ne conserver que les autres instruments.