Licence 3 E3A

TP : ANALYSE SPECTRALE ET FILTRAGE

Table des matières

L'analyse spectrale est un outil qui permet de mettre en évidence des périodicités ou des pseudo-périodicités dans un phénomène observé, par l'intermédiaire de signaux mesurés. Plus généralement c'est l'outil d'étude du contenu fréquentiel d'un signal et il est fondé sur la transformée de Fourier. L'objectif de ce travail est de mettre en pratique une partie des outils présentés dans différents cours, en plus de vous familiariser avec un outil logiciel très puissant : `MatLab`.

1. Simulation de signaux

On considère une sinusoïde pure à la fréquence $f_0 = 440\ Hz$

$$x(t) = \alpha \sin 2\pi(f_0 t + \varphi)\ \forall t \in \mathbb{R}\ ,$$

où $\alpha \in \mathbb{R}$ est l'amplitude et $\varphi \in [0,1]$ est la phase réduite. Il s'agit du « la »du diapason et c'est aussi la fréquence que vous entendez lorsque vous décrochez votre téléphone.

On échantillonne ce signal à la cadence $F_e = 1/T_e = 10\ kHz$ ce qui donne un signal à temps discret

$$x_n = \alpha \sin 2\pi(\nu_0 n + \varphi)\ \forall n \in \mathbb{Z}\ ,$$

où ν_0 est la fréquence réduite. *Que vaut-elle en fonction de f_0 et F_e ?*

Sous `MatLab`, on commence par fabriquer un vecteur contenant les échantillons de ce signal, pour $n = 0, 1, \dots, N-1$. Pour cela, créez un fichier de commande `MatLab` contenant la séquence de commande suivante.

```
% Nettoyage
>clear all, close all

% Définition des constantes
```

```
>f0 = 440;
>Fe = 1e4;
>Phi = 0.27;
>Alpha = 1.2;
>N = 1024;
```

Exécutez alors le programme et assurez-vous que tout s'est passé comme attendu. La commande `whos` donne la liste des variables en mémoire. Pour obtenir la valeur d'une variable, tapez simplement son nom (puis la touche `enter`).

Remarque 1. *Pour toutes les commandes `MatLab` vous avez accès à une aide en ligne en tapant `help`. Par exemple, `help plot` donne le manuel d'utilisation de la commande de tracé `plot`. La fin de chaque manuel renvoie à d'autres commandes. La fin du manuel de la commande `plot` renvoie par exemple aux commandes `title`, `xlabel`, `ylabel` qui permettent d'inclure des titres et légendes aux figures.*

Par ailleurs, la commande `lookfor` permet de faire une recherche dans les manuels : `lookfor fft` renvoie la liste des comandes `MatLab` dont le manuel contient `fft`.

La création du signal lui même est simple :

```
% Création du signal
> Time = linspace(0,N/Fe,N)';
> Sig = Alpha * sin(2*pi*(f0*Time+Phi));
```

La première ligne fabrique un vecteur permettant une discrétisation de l'axe temporelle associée à la fréquence d'échantillonnage. Le vecteur `Time` contient N points espacés linéairement entre le temps $t = 0$ s et le temps $t = N/Fe$ s. La seconde est *vectorielle* : chaque élément du vecteur `Time` est multiplié par la valeur `f0`, la valeur de `Phi` est ajoutée à chaque composante obtenue, puis chaque composante est multipliée par 2π ; enfin, la fonction `sin` donne le sinus de chacune des composantes. Il s'agit là d'une caractéristique essentielle du langage `MatLab` : toutes les opérations sont *matricielles* ou *vectorielles*. L'ensemble du TP se réalise donc **sans** boucle « for ». Exécutez à nouveau le programme et assurez vous que tout s'est passé comme prévu. *Les vecteurs `TpsD` et `Sig` sont-ils des vecteurs lignes ou des vecteurs colonnes ?*

Le tracé est également très simple.

```
% Tracé du signal
> figure(1),clf
> plot(Time,Sig)
```

L'axe horizontal est alors gradué selon l'axe temporel en seconde. Prenez le temps d'apprendre le rôle de chacune des commandes rencontrées. Voyez les possibilités des commandes `axis`, `grid`, `title`, `xlabel`. Essayez également la séquence suivante :

```
% Tracé du signal
> figure(1),clf
> subplot(211),plot(Time,Sig)
> subplot(212),plot(Time(10:100),Sig(10:100))
```

Observez l'effet obtenu si vous multipliez la fréquence d'échantillonnage par 10 ou par 100. Si vous la divisez par 10 ou par 100.

Ajoutez une seconde raie assez proche de la première, par exemple à $f_1 = 460\ Hz$ et d'amplitude voisine. Ce signal permettra, dans la suite, d'évaluer le pouvoir de résolution de l'analyse de Fourier c'est-à-dire la capacité à séparer, à distinguer deux fréquences proches. Ajoutez également une composante à $2,5\ kHz$ d'amplitude moitié.

2. Transformée de Fourier

2.1. **Rappels.** *Quelle est la transformée de Fourier à temps continu du signal $x(t)$? Quelle est la transformée de Fourier à temps discret du signal $\{x_n\}_{n\in\mathbb{Z}}$? Quel lien y-a-t-il entre elles ? Pourquoi est-on obligé de définir une transformée de Fourier Finie ?*

2.2. **Manipulation de base.** En utilisant la fonction `fft` de `MatLab`, calculez la `fft` du vecteur `Sig` : `fft(Sig)`. Pour tracer le résultat, on veut centrer la courbe sur la fréquence nulle. Il faut pour cela utiliser la fonction `fftshift`. Consultez le manuel : `help fftshift`. Attention ! Cette fonction ne calcule pas de `fft`, elle modifie seulement l'ordre des échantillons d'un vecteur : essayez `x=(1:6)` suivi de `fftshift(x)`. Tracez alors le résultat de la fft en partie réelle, partie imaginaire, module au carré (le spectre) et phase (la fonction `real` donne la partie réelle d'un nombre complexe et le manuel de cette fonction renvoie aux fonctions permettant de déterminer la partie imaginaire, le module et la phase). Tracez également le $\log_{10}$ du module. Graduez correctement l'axe des fréquences, à la fois en fréquence réduite et en fréquence réelle. Pour cela, vous pouvez créer un vecteur contenant l'axe des fréquences en utilisant la commande `linspace`.

Vous en arrivez à une partie essentielle de ce TP et on se concentre sur l'analyse du spectre.

— Vous devez retrouver des « pics » aux fréquences $440\ Hz$, $460\ Hz$ et $2,5\ kHz$. Sont-ils positionnés précisément ? Évaluez cette précision en fonction de M. Expliquez la présence des trois autres pics.
— Vous devez constater également l'apparition de « *rebonds* » au pieds des pics. On parle de « *ringing* » dans la litérature anglosaxone. Expliquez leur origine.
— On se concentre maintenant sur les composantes à 440 et 460 Hz. Observez-les lorsque $N = 1024$, $N = 512$ et $N = 256$. Commentez.
— Que se passe-t-il si on change F_e en $F_e = 6\ kHz$, en $F_e = 4\ kHz$, en $F_e = 0,5\ kHz$? Commentez.

2.3. **Filtrage.** On veut réaliser maintenant une opération de filtrage passe-pas du signal, pour éliminer la composante à $2,5\ kHz$ et ne conserver que les composantes à $440\ Hz$ et $460\ Hz$. On éliminera pour cela les fréquences au delà de $2\ kHz$. Réalisez très simplement cette opération dans le domaine de Fourier : vous avez ainsi réalisé un filtrage passe-bas « idéal ».

Cette opération de filtrage idéal dans le domaine de Fourier ne peut pas être utilisée pour réaliser un filtrage en ligne. Expliquez pourquoi. Dans le cas où on est contraint de réaliser ce filtrage en ligne, une possibilité repose sur le filtrage convolutif. On pourra utiliser, par exemple, le filtre de réponse impulsionnelle h définie par

$$h(n) = \begin{cases} 1/P & \text{si } n \in [0; P-1] \\ 0 & \text{sinon} \end{cases}$$

S'agit-il bien d'un filtre passe-bas ? Quel est son transfert en fréquence ? Utilisez la fonction `filter` pour filtrer le signal d'origine et éliminer les composantes au delà de $2\ kHz$. Réglez empiriquement la valueur de P. Comment prévoir à l'avance une « bonne valeur » pour P. Comparez les résultats obtenus ici à ceux obtenus par filtrage idéal.