Chambre de Commerce et d'Industrie de Paris-IdF ——— ESIEE	Unité : Traitement du signal TP n°1 - RI, Fourier, filtrage	Classe ISBS1

Remis par M. J.-F. BERCHER

ÉNONCÉ

TP SOUS PYTHON

Le TP sera réalisé sous Python. Sous Python, on dispose d'un certain nombre de bibliothèques scientifiques qui nous seront utiles, notamment *numpy*, *scipy* et *matplotlib*. Le module pylab permet de charger des commandes simplifiées analogues à celles que l'on trouve dans Matlab.

Spyder

Pour ces TP, vous utiliserez l'éditeur Spyder, qui précharge les modules scientifques et le mode Pylab. Dans *Outils>Préférences>Console IPython>Graphiques*, vous vérifierez que le mode graphique (Matplotlib) est activé, que le chargement automatique de Numpy et Pylab est coché, et enfin que la sortie graphique est positionnée sur « automatique ». Dans *Démarrage*, vous cocherez « ouvrir une console IPython au démarrage ».

Pour ceux qui voudraient installer un environnement équivalent sur un ordinateur personnel, il vous suffit

- sous Windows, de télécharger et installer une distribution Pythonxy ou WinPython
- sous Linux debian (ou ubuntu) : `sudo apt-get install python-numpy python-scipy python-matplotlib ipython ipython-notebook python-pandas python-sympy spyder python-nose python-pygments python-zmq python-setuptools python-sphinx`

Dans Spyder, vous pouvez entrer vos commandes soit directement dans la console, soit créer un script (enregistrez le nom sous un nom différent de temp) et l'exécuter par la touche de raccourci F5, ou n'exécuter que les lignes sélectionnées par F9.

Aide

Vous disposez d'une aide en ligne via l'insecteur d'objet, qui est également automatiquement activé lorsque vous entrez des commandes dans l'éditeur ou la console ; si vous entrez `plot(` par exemple, l'aide sur la fonction sera affichée dès que vous aurez entré la parenthèse ouvrante. Vous disposez également d'une aide en ligne par `help(nom_de_la fonction)`.

Imports

Vous aurez besoin à la fois d'une fonction de filtrage, `lfilter`, et des fonctions de transformée de Fourier (`fft` et `ifft`). Les imports se font par :

```
# On a besoin de la fonction de filtrage
import scipy from scipy.signal import lfilter
# On aura besoin des fonctions fft
from numpy.fft import fft, ifft
```

En outre, on fournit un fichier `defs_tp_RI.py`, qui contient quelques définitions, que vous téléchargerez et exécuterez une fois.

Tracer des graphiques

La commande `figure()` ouvre une nouvelle figure graphique. Si nécessaire `clf()` efface la figure courante. la commande `plot(x,y,label='texte', color='nom_de_couleur')` représente y en fonction de x. La commande `legend()` ajoute une légende construite avec les différents labels ; les paramètres labels, color, etc sont tous optionnels. La commande `stem`, avec les mêmes paramètres que plot, permet d'afficher une courbe en batons (points non reliés). Vous pourrez par exemple tester les lignes suivantes :

```
# Représentation
N=100
stem(range(N),dirac(N))
title("Dirac $\delta(n)$")
xlabel("n")
ylim([0, 1.2])     # Pour mieux voir l'impulsion xlim([-5, 10])
```

Pour les représentations en fréquence, prendre garde au fait que
(i) les TF sont complexes (prendre le module – `abs` le cas échéant), et
(ii) le résultat de la TF est rendu sur une période allant de 0 à 1 ;
il est possible de recentrer le résultat, et voir les fréquences négatives, en utilisant un décalage, fonction `fftshift`.
Pour les représentations en fréquence, prendre garde au fait que (i) les TF sont complexes (prendre le module – `abs` le cas échéant), et (ii) le résultat de la TF est rendu sur une période allant de 0 à 1 ; il est possible de recentrer le résultat, et voir les fréquences négatives, en utilisant un décalage, fonction `fftshift`.

I. Réponse impulsionnelle et fonctions de transfert pour des signaux discrets

Dans cet exercice, on travaillera avec des signaux échantillonnés, et si nécessaire, on consièrera que la fréquence d'échantillonnage vaut $Fe = 32$.
On considère la relation de filtrage décrite par l'équation aux différences suivante :

$$y(n) = ay(n-1) + x(n),$$

où $x(n)$ est l'entrée du filtre et $y(n)$ sa sortie.

A. Étude temporelle

1) Calculez la réponse impulsionnelle (RI), sur le papier, en fonction de a, en supposant le système causal, et les conditions initiales éventuelles nulles.
2) Sous Python, consultez l'aide de la fonction `lfilter`, par `help(lfilter)` et tâchez d'en comprendre le fonctionnement. Proposez à l'enseignant une méthode pour calculer numériquement la RI du filtre, puis contrôlez graphiquement l'allure de la RI, avec $a = 0.8$. On rappelle que la fonction `dirac` permet de générer une impulsion de Dirac à temps discret.
3) Calculez et visualisez la réponse impulsionnelle pour $a = -0.8$, $a = 0.99$ et pour $a = 1.01$. Il pourra être utile de définir une fonction qui rend directement la réponse impulsionnelle. Conclusions.

B. Étude fréquentielle

1) Donnez l'expression de la fonction de transfert en z correspondant à cette équation aux différences.
2) Donnez l'expression de la fonction de transfert $H(f)$, puis de $|H(f)|$ pour a quelconque. Précisez les amplitudes théoriques en $f = 0$ et $f = 1/2$. Sous Python, calculez la FT du filtre en prenant la TF (fonction `fft`) de la RI, pour $a = 0.8$ et $a = -0.8$, et visualisez les résultats. Conclusions.

C. Filtrage

1) Créez une sinusoïde x, à la fréquence $f_0 = 3$, échantillonnée à $Fe = 32$, sur 128 points :

```
# Création d'une petite sinusoïde
N, fo, Fe = 128, 3, 32
t=arange(N)/Fe
x=sin(2*pi*fo*t)
```

2) Filtrez cette sinusoïde par le filtre précédent
   — en utilisant la fonction `flilter`, `y1=lfilter([1],[1, -0.8],x)`
   — en utilisant une convolution, `y2=lfilter(h,[1],x)` Expliquez pourquoi ce dernier calcul correspond effectivement à une convolution.

   Comparez graphiquement ces deux résultats affichez les deux courves, voire la différence des signaux)

3) Calculez la TF X du signal x et la TF H de la réponse impulsionnelle h. Visualisez ces deux résultats. Calculez la TF inverse du produit $X(f)H(f)$: `y3=real(ifft(X*H));`. Comparez $y3$ et $y1$. Conclusions.

4) Mesurez la valeur du gain et du déphasage entre x et $y1$ (fonctions `plot(t,x,t,y1)` et `zoom`, ou xlim). Mesurez la valeur du gain et du déphasage, à la fréquence fo, sur la fonction de transfert H. Conclusions.

5) Reprenez ces questions en utilisant un train d'impulsions rectangulaires plutôt qu'une sinusoïde (utiliser `rectpulse` à la place de `sin`). Utilisez une la fréquence fo puis une fréquence $f_1 = 0.5$. Comparez l'entrée et la sortie du système et concluez sur l'invariance par filtrage de la classe des fonctions harmoniques.