Filtre numérique réducteur de bruit

1. Introduction

Dans les TP Filtrage numérique, nous avons vu comment calculer la réponse impulsionnelle d'un filtre numérique passe-bas, et comment effectuer le filtrage par convolution sur un signal numérique.

L'objectif de ces TP est de mettre en œuvre un filtre passe-bas réducteur de bruit, et de le tester sur un signal périodique comportant plusieurs harmoniques. On verra aussi une application de ce filtre à la dérivation numérique.

2. Analyse d'un signal bruité

On se fixe comme objectif de numériser un signal dont les fréquences utiles sont dans la bande [0,2000] Hz. Celui-ci est délivré sur la sortie audio par le programme pure-data syntheseHarmonique.pd.

On utilisera pour cela la carte d'acquisition SysamSP5 pilotée par le script python suivant, qui contient aussi une fonction de calcul du spectre :

numerisation.py

```
import pycanum.main as pycan
import numpy
from matplotlib.pyplot import *
import scipy.signal

def spectre(t,u):
    N=len(u)
    te=t[1]-t[0]
    zeros=numpy.zeros(6*N)
    U = numpy.concatenate((u*scipy.signal.blackman(N),zeros))
    NN=len(U)
    spectre = numpy.absolute(numpy.fft.fft(U))*2.0/N/0.42
    freq = numpy.arange(NN)*1.0/(NN*te)
    return (freq,spectre)

fe=40000.0
T=1.0
te=1.0/fe
N = int(T/te)
can = pycan.Sysam("SP5")
can.config_entrees([0],[5.0])
can.config_echantillon(te*10**6,N)
can.acquerir()
t0=can.temps()[0]
u0=can.entrees()[0]
can.fermer()
```

La fréquence d'échantillonnage est ici $f_e = 40\,\text{kHz}$. La durée totale de l'analyse est $T = 1\,\text{s}$.

Générer avec pure-data un signal périodique dont toutes les harmoniques sont dans la bande [0,2000] Hz, avec un fort bruit ajouté. Compléter le script pour tracer la représentation temporelle du signal et son spectre.
Faire le branchement pour délivrer ce signal simultanément sur la voie 1 de l'oscilloscope et sur l'entrée EA0 de la carte SysamSP5.
Quel est le spectre du bruit ?
Justifier le choix de la fréquence d'échantillonnage de 40 kHz, alors que le signal utile ne dépasse pas 2 kHz.

3. Conception du filtre numérique passe-bas

L'objectif du filtre numérique passe-bas est de réduire les composantes spectrales du bruit situées dans la bande [2,20] kHz.

Nous allons utiliser un filtre à réponse impulsionnelle finie, analogue à celui étudié dans le TP Filtrage numérique. Les caractéristiques de ce filtre sont les suivantes :

- ▷ Fréquence d'échantillonnage $f_e = 40\,\text{kHz}$.
- ▷ Fréquence de coupure $f_c = 2\,\text{kHz}$.
- ▷ Taille de la réponse impulsionnelle $2P+1$ avec $P = 50$.
- ▷ Fenêtrage de Hamming pour enlever les ondulations du gain dans la bande passante.

La suite des coefficients du filtre, c'est-à-dire sa réponse impulsionnelle, est calculée de la manière suivante :

```
fc = 2000.0
P=50
b = scipy.signal.firwin(numtaps=2*P+1,cutoff=[fc/fe],window='hamming',nyq=0.5)
```

La réponse fréquentielle est obtenue avec la fonction `scipy.signal.freqz` :

```
w,h=scipy.signal.freqz(b)

import numpy
figure()
subplot(211)
plot(w/(2*numpy.pi),20*numpy.log10(numpy.absolute(h)))
xlabel("f/fe")
ylabel("GdB")
grid()
subplot(212)
plot(w/(2*numpy.pi),numpy.unwrap(numpy.angle(h)))
xlabel("f/fe")
ylabel("phase")
```

```
grid()
```

Calculer ce filtre. Tracer sa réponse impulsionnelle avec la fonction `stem`. Tracer sa réponse fréquentielle et vérifier qu'elle correspond bien aux spécifications recherchées.

4. Filtrage d'un signal bruité

Pour effectuer le filtrage sur le tableau des échantillons du signal, on peut utiliser la fonction `filtrage_convolution` définie dans le TP Filtrage numérique :

```
def filtrage_convolution(x,b):
    N = len(b)
    ne = len(x)
    y = numpy.zeros(ne)
    for n in range(N-1,ne):
        accum = 0.0
        for k in range(N):
            accum += b[k]*x[n-k]
        y[n] = accum
    return y
```

On peut aussi utiliser la fonction `scipy.signal.lfiltic` de la manière suivante :

```
zi = scipy.signal.lfiltic(b,[1],y=[0],x=[0]) # condition initiale
[y0,zf] = scipy.signal.lfilter(b,[1],x0,zi=zi)
```

`x0` est le signal à filtrer et `y0` le signal filtré. L'avantage de cette fonction est de permettre un filtrage de blocs d'échantillons consécutifs. La variable `zf` contient l'état du filtre à la fin du filtrage, qui peut être utilisé pour filtrer le bloc d'échantillons suivant.

Numériser un signal périodique bruité et réaliser son filtrage.
Tracer sur la même figure le signal brut et le signal filtré.
Comparer le spectre du signal brut et celui du signal filtré.

Tester le filtre avec une fréquence d'échantillonnage plus basse, par exemple 4 kHz. Commenter le résultat.

5. Filtre dérivateur

On souhaite effectuer une dérivation du signal numérique. La dérivation numérique est définie par la relation :

$$y_n = \frac{x_n - x_{n-1}}{T_e} \tag{1}$$

où T_e est la période d'échantillonnage.

On utilise donc un filtre de convolution dont la réponse impulsionnelle est :

$$b = (1/T_e, -1/T_e) \tag{2}$$

Compléter le script pour effectuer la dérivation du signal issu de l'acquisition, et du signal après traitement par le filtre anti-bruit. Tracer les deux signaux sur la même figure.
Tracer la réponse fréquentielle du filtre dérivateur.
Interpréter les observations et conclure sur l'utilité du filtre anti-bruit pour le calcul de la dérivée d'un signal.

6. Restitution analogique du signal filtré

Le signal filtré peut être converti en signal analogique avec le convertisseur N/A de la carte SysamSP5. La technique a été vue dans les TP Synthèse numérique d'un signal périodique.

Programmer la sortie de la carte SysamSP5 pour qu'elle délivre les échantillons du signal filtré à la bonne fréquence.
Déclencher la sortie en mode permanent et observer le résultat sur la voie 2 de l'oscilloscope.

On obtient finalement le signal analogique bruité sur la voie 1 et le signal filtré sur la voie 2 de l'oscilloscope. Bien sûr, il ne s'agit pas d'un filtrage en temps réel et donc ces deux signaux ne sont pas synchrones.

Observer en détail le signal filtré sur l'oscilloscope. En quoi ce signal diffère-t-il de la courbe tracée sur python ?
Expliquer comment il faudrait effectuer le lissage avec un filtre analogique.