TP 9 - Filtre intégrateur numérique

1. Introduction

L'intégration d'un signal est une opération courante en traitement du signal. Par exemple, le signal délivré par un capteur de vitesse doit être intégré pour donner la position.
Pour un signal analogique, l'intégration est définie par :

$$y(t) = \frac{1}{\tau}\int_0^t x(t')\,dt' \tag{1}$$

où τ est une constante (éventuellement négative). Si $x(t)$ et $y(t)$ sont de mêmes dimensions, τ est un temps. Sa fonction de transfert en régime sinusoïdal permanent est :

$$H(\omega) = \frac{1}{j\tau\omega} \tag{2}$$

En régime sinusoïdal, un intégrateur est donc caractérisé par un déphasage de $\pm\frac{\pi}{2}$ et par un gain en décibel décroissant à -20 dB par décade.
Considérons un signal d'entrée périodique dont la série de Fourier s'écrit :

$$x(t) = \frac{C_0}{2} + \sum_{n=1}^{\infty} C_n \cos\left(2\pi n f\, t + \phi_n\right) \tag{3}$$

La composante constante $C_0/2$ (appelée *composante continue*) donne en sortie :

$$y(t) = \frac{1}{\tau}\frac{C_0}{2}t \tag{4}$$

Si le signal comporte une composante continue, même très petite, il se produit une dérive de la tension de sortie. Le plus souvent, la composante continue provient d'un décalage très faible introduit par l'électronique ; la dérive est alors un phénomène indésirable qu'il convient d'éliminer.
L'objectif de ce TP est la mise en œuvre d'un filtre intégrateur numérique.

2. Intégrateur numérique parfait

2.a. Définition du filtre

Soit x_n le signal numérique obtenu par échantillonnage de $x(t)$ à la fréquence d'échantillonnage T_e, et y_n celui obtenu par échantillonnage de $y(t)$.

[1] Écrire $x(t)$ en fonction de la dérivée de $y(t)$. Après avoir exprimé la dérivée sous la forme d'une différence finie, en déduire la relation suivante :

$$y_n = y_{n-1} + \frac{T_e}{\tau}x_n \tag{5}$$

La relation précédente est de la forme suivante :

$$a_0 y_n = -a_1 y_{n-1} + b_0 x_n + b_1 x_{n-1} \tag{6}$$

Dans un filtrage par convolution, la sortie se calcule uniquement avec les valeurs de l'entrée. Dans le cas présent, le calcul de la sortie se fait avec les valeurs de l'entrée à l'instant n et à

l'instant $n-1$ mais aussi avec la valeur de la sortie à l'instant $n-1$. Lorsque la sortie à l'instant n dépend explicitement des valeurs de la sortie des instants antérieurs, le filtre numérique est qualifié de *filtre récursif*. Pour étudier la réponse fréquentielle d'un filtre récursif, il faut considérer sa fonction de transfert en Z. Dans le cas présent, elle s'écrit :

$$H(z) = \frac{b_0 + b_1 z^{-1}}{a_0 + a_1 z^{-1}} \tag{7}$$

avec :

$$z = e^{i2\pi \frac{f}{fe}} = e^{i2\pi u} \tag{8}$$

Le rapport u de la fréquence sur la fréquence d'échantillonnage varie entre 0 et $1/2$.
La réponse fréquentielle d'un filtre, avec un gain en décibel et une échelle de fréquence logarithmique, peut être obtenue de la manière suivante :

```
import scipy.signal
import numpy
import matplotlib.pyplot as plt

w,h=scipy.signal.freqz(b,a)
plt.figure()
plt.subplot(211)
plt.plot(w/(2*numpy.pi),20*numpy.log10(numpy.absolute(h)))
plt.xscale('log')
plt.xlabel("f/fe")
plt.ylabel("GdB")
plt.grid()
plt.subplot(212)
plt.plot(w/(2*numpy.pi),numpy.angle(h))
plt.xscale('log')
plt.xlabel("f/fe")
plt.ylabel("phase")
plt.grid()
plt.show()
```

[2] Tracer la réponse fréquentielle du filtre précédent dans le cas $T_e/\tau = 1$. Est-ce bien un filtre intégrateur ?
La relation de récurrence précédente n'étant pas satisfaisante, on la modifie en remplaçant x_n par :

$$\frac{x_n + x_{n-1}}{2} \tag{9}$$

[3] Écrire la relation de récurrence et en déduire les coefficients de ce filtre. Tracer sa réponse fréquentielle et conclure.

[4] Comment la réponse fréquentielle est-elle modifiée lorsqu'on change le rapport T_e/τ ?
On remarque que la relation obtenue correspond à la méthode des trapèzes pour le calcul numérique d'une intégrale.

2.b. Réalisation du filtrage

Pour réaliser le filtrage d'un signal dont les échantillons sont dans un tableau, on remarque que le calcul de la sortie commence à y_1 et qu'il faut choisir une valeur pour y_0. On prendra $y_0 = 0$.

[5] Écrire une fonction `filtrage_recursif(x,a,b)` qui renvoie le tableau `y` des échantillons y_n. Les paramètres `a` et `b` sont deux listes de deux éléments qui contiennent les coefficients du filtre. On fera une implémentation itérative du filtrage (sans récursivité) car c'est la seule possible dans un traitement de signal en temps réel.

[6] Tester cette fonction sur une impulsion unité. Quelle est la réponse impulsionnelle ? Est-elle finie ?

Les filtres récursifs ont une réponse impulsionnelle infinie, c'est-à-dire contenant un nombre infini d'échantillons non nuls. Il est théoriquement impossible de réaliser une intégration numérique au moyen d'un filtre de convolution, car il faudrait un nombre infini de coefficients. En pratique, on peut tout de même y parvenir car le signal numérique commence à $n = 0$, à condition de définir une convolution avec un nombre de coefficients qui augmente au cours du temps. En effet, le filtrage défini par la somme suivante réalise bien une intégration (méthode des trapèzes) :

$$\begin{aligned} y_n =& \frac{x_n + x_{n-1}}{2} + \frac{x_{n-1} + x_{n-2}}{2} + \ldots + \frac{x_2 + x_1}{2} + \frac{x_1 + x_0}{2} \\ =& \frac{x_n}{2} + x_{n-1} + x_{n-2} + \ldots + x_1 + \frac{x_0}{2} \end{aligned}$$

Cependant, l'utilisation de ce filtrage dans un traitement en temps réel serait extrêmement inefficace puisque le nombre de termes de la somme augmente avec n. Un traitement efficace consiste à calculer la somme partielle au fur et à mesure de l'arrivée des échantillons x_n et c'est exactement ce que fait le filtre récursif (à condition qu'il ne soit pas implémenté en utilisant la récursivité).

Dans le cas du filtre intégrateur, la réponse impulsionnelle ne tend pas vers zéro, ce qui montre l'instabilité du filtre intégrateur. D'une manière générale, un filtre récursif peut être instable alors qu'un filtre de convolution (à nombre fixé de coefficients) est toujours stable. L'avantage des filtres récursifs est leur efficacité : le même effet qu'un filtre de convolution peut être obtenu avec beaucoup moins de calculs, ce qui est très important dans un traitement du signal en temps réel.

2.c. Test de l'intégrateur

Un bon moyen de tester un intégrateur est de lui fournir un signal créneau. Nous allons donc numériser un signal créneau délivré par le générateur de signaux avec la carte d'acquisition et le script suivant :

numerisation.py

```
import pycanum.main as pycan
import numpy
import matplotlib.pyplot as plt
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=0.1
te=1.0/fe
N = int(T/te)
can = pycan.Sysam("SP5")
can.config_entrees([0],[10.0])
can.config_echantillon(te*10**6,N)
can.acquerir()
t0=can.temps()[0]
u0=can.entrees()[0]
can.fermer()
```

[7] Numériser un signal créneau dont la fréquence est environ 100 fois moins que la fréquence d'échantillonnage. Tracer le spectre du signal échantillonné.

[8] D'après ce spectre, la fréquence d'échantillonnage est-elle assez grande ?

[9] Effectuer l'intégration et tracer les deux signaux sur la même figure (utiliser `subplot`).

[10] Quel est l'effet d'un décalage (offset) de tension en entrée ? Expliquer.

[11] Tracer le spectre du signal échantillonné et celui du signal après filtrage (utiliser `subplot`). Commenter le spectre de ce dernier.

3. Filtre passe-bas du premier ordre

3.a. Filtre analogique

Pour stabiliser le filtre intégrateur afin de ne pas avoir de dérive en sortie, il faut ramener le gain à fréquence nulle à une valeur finie. Un moyen simple de le faire est d'utiliser un filtre passe-bas du premier ordre.
Considérons un filtre passe-bas dont la fonction de transfert est :

$$H(\omega) = \frac{g}{1 + j\tau\omega} \tag{10}$$

Ce filtre est pratiquement un intégrateur lorsque $\omega \gg 1/\tau$.
L'équation différentielle associée à ce filtre est :

$$y(t) + \tau\frac{dy(t)}{dt} = gx(t) \tag{11}$$

[12] Retrouver la fonction de transfert à partir de cette équation différentielle.

3.b. Filtre numérique

La discrétisation de l'équation différentielle précédente conduit à la relation suivante :

$$y_n = \frac{1}{1 + \frac{T_e}{\tau}} y_{n-1} + \frac{g}{2}\frac{\frac{T_e}{\tau}}{1 + \frac{T_e}{\tau}}\left(x_n + x_{n-1}\right) \tag{12}$$

[13] Montrer qu'on obtient bien cette relation.

[14] À l'aide de sa fonction de transfert en Z, montrer que le gain à fréquence nulle est g.

On obtient un filtre passe-bas dont la fréquence de coupure peut être rendue très basse en choisissant τ très grand devant T_e. Le gain g sera choisi en fonction de l'amplitude souhaitée pour le signal de sortie. Il n'a pas d'effet sur la forme de la réponse fréquentielle.

[15] Tracer la réponse fréquentielle de ce filtre en choisissant le rapport T_e/τ. Ajuster T_e/τ pour que le filtre soit approximativement intégrateur à partir du centième de la fréquence d'échantillonnage.

[16] Tester le filtre sur un signal créneau. Faire varier la fréquence du signal.

[17] Quelle est la conséquence d'un décalage de tension en entrée ?

[18] Quel est le temps de réponse de ce filtre ? Comparer à τ.

[19] Calculer et tracer la réponse impulsionnelle de ce filtre.

Dans certains cas, on est amené à calculer l'intégrale suivante :

$$y(t) = \int_0^t x(t')\,dt' \tag{13}$$

Dans ce cas, les signaux $x(t)$ et $y(t)$ n'ont pas les mêmes dimensions. On définit alors l'intégrateur en posant $g = \tau$, ce qui donne :

$$y_n = r y_{n-1} + \frac{T_e}{2}(x_n + x_{n-1}) \tag{14}$$

où r est un paramètre proche de 1 mais inférieur à 1, qui permet de contrôler la fréquence de coupure.

4. Filtre intégrateur avec coupure de la fréquence nulle

Le filtre précédent peut être amélioré en annulant le gain à fréquence nulle. Il s'agit de faire un filtre passe-bande avec une fréquence de bande passante très faible. Un tel filtre peut être défini par la fonction de transfert en Z suivante :

$$H(z) = \frac{T_e}{\tau}\frac{g}{2}\frac{1 - z^{-2}}{1 - 2rz^{-1} + r^2 z^{-2}} \tag{15}$$

où r est un paramètre proche de 1 mais inférieur à 1.

[20] Vérifier que le gain à fréquence nulle est nul.

La relation de récurrence est :

$$y_n = 2r y_{n-1} - r^2 y_{n-2} + \frac{T_e}{\tau}\frac{g}{2}(x_n - x_{n-2}) \tag{16}$$

qui est un cas particulier de la forme générale suivante :

$$a_0 y_n = -a_1 y_{n-1} - a_2 y_{n-2} + b_0 x_n + b_1 x_{n-1} + b_2 x_{n-2} \tag{17}$$

[21] Tracer la réponse fréquentielle de ce filtre et ajuster le paramètre r pour que l'intégration se fasse à partir du centième de la fréquence d'échantillonnage.

[22] Écrire une fonction `filtrage_recursif(x,a,b)` pour réaliser ce filtrage. La condition initiale est $y_0 = y_1 = 0$.

[23] Appliquer ce filtre à un signal créneau échantillonné. Vérifier que le décalage en entrée est annulé en sortie.

[24] Quelle est l'influence du paramètre r sur le temps de réponse du filtre ?

Pour finir, on fera fonctionner le filtre intégrateur en visualisant le résultat en temps réel, avec le script integrateurPermanent.py. La durée totale de l'acquisition est de une minute.