

TP7 : Synthèse numérique d'un signal périodique

1. Introduction

La synthèse numérique d'un signal consiste à générer les échantillons d'un signal par le calcul, avant d'en faire la conversion numérique-analogique. Les générateurs numériques de signaux (par ex. le SIGLENT SDG1025) fonctionnent de cette manière.

L'objectif de ces TP est d'apprendre à générer les échantillons du signal, à ajuster les différents paramètres pour régler sa fréquence, puis à en faire la conversion numérique-analogique avec la carte SysamSP5.

Matériel

- ▷ Oscilloscope.
- ▷ Carte Sysam SP5.
- ▷ Plaque de montage à fiches bananes.
- ▷ Résistance de 3,3 k Ω (ou valeur voisine) et condensateur de 10 nF.

2. Conversion numérique-analogique

2.a. Principe

La carte Sysam SP5 comporte deux convertisseurs numérique-analogique 12 bits (CNA). Ces deux convertisseurs délivrent une tension comprise entre -10 V et 10 V, respectivement sur les sorties SA1 et SA2. Le CNA reçoit des nombres entiers codés en 12 bits, compris entre 0 et 4095, la valeur 0 correspondant à la tension -10 V, la valeur 4095 à la tension 10 V. En conséquence, la tension en sortie ne peut prendre que 4096 valeurs.

Il est possible de programmer la carte afin qu'elle lise les échantillons dans une table (stockée dans sa mémoire interne) et fasse la conversion à une période d'échantillonnage donnée. La période d'échantillonnage est nécessairement multiple de la période de l'horloge τ . Celle-ci vaut $\tau = 0,200 \mu\text{s}$ pour la Sysam SP5, soit une fréquence d'échantillonnage maximale de 5 MHz. La période d'échantillonnage s'écrit donc : $T_e = q\tau$ où q est un nombre entier supérieur ou égal à 1.

La table contenant les échantillons (table d'entiers 12 bits) a une longueur maximale $N_{max} = 2^{17} - 1 = 131071$.

2.b. Programmation de la carte SysamSP5

On commence par importer le module et ouvrir l'interface :

```
import pycanum.main as pycan
sys = pycan.Sysam("SP5")
sys.ouvrir()
```

Supposons que `te` soit la période d'échantillonnage et que `echantillons` soit un tableau numpy contenant les échantillons (sous la forme de tensions en volt). La configuration de la sortie 1 (SA1) se fait de la manière suivante :

```
sortie = 1
sys.config_sortie(sortie, te, echantillons, -1)
```

La période d'échantillonnage t_e est donnée en microsecondes. Sa valeur effective est la période multiple de τ la plus proche de la période demandée. Le dernier argument indique combien de fois la conversion du tableau complet doit être répétée. Lorsque la valeur est -1, la répétition se fait sans fin, ce qui permet de générer un signal périodique.

Pour déclencher la conversion, il faut exécuter :

```
sys.declencher_sorties(1, 0)
```

Ici, on ne déclenche que la première sortie. Cette fonction retourne immédiatement. Il faut donc prévoir soit une attente avec `time.sleep`, soit une autre fonction bloquante, par exemple la fonction `matplotlib.pyplot.show`.

Voici ce qu'il faut faire pour bloquer l'exécution avec une fenêtre. Lorsque l'utilisateur ferme la fenêtre, on peut stopper la sortie et fermer l'interface.

```
plt.figure()
plt.show()
sys.stopper_sorties(1, 0)
sys.fermer()
```

3. Méthodes de synthèse

La mise au point des deux méthodes de synthèse (et les réponses aux différentes questions) se fera dans un notebook à créer sur le Jupyter Hub. La réalisation expérimentale avec la conversion numérique-analogique se fera évidemment dans un script exécuté localement.

3.a. Échantillonnage d'une période


Soit T la période du signal périodique à synthétiser. Pour définir la forme du signal indépendamment de sa fréquence, on utilise sa phase θ . Par exemple, un signal sinusoïdal d'amplitude 5 V sera défini par la fonction Python suivante :

```
def signal(theta):
    return 5.0*numpy.cos(theta)
```

D'après l'expression du signal comme somme d'une série de Fourier, un signal périodique est bien une fonction de la phase puisque l'harmonique de rang n est une fonction sinusoïdale de $n\theta$.


La première méthode consiste à échantillonner exactement une période, c'est-à-dire l'intervalle de phase $[0, 2\pi]$. Soit N le nombre d'échantillons sur une période. Pour obtenir une bonne reconstruction du signal, il faut être en sur-échantillonnage. Par exemple pour un signal sinusoïdal, on prendra si possible $N \approx 100$ ou plus. La table stockée dans la mémoire interne

de la carte Sysam comportera N échantillons. La conversion cyclique des valeurs contenues dans cette table permettra de générer le signal périodique voulu. Le choix de la fréquence d'échantillonnage (c.a.d. du nombre entier q) permettra d'obtenir une fréquence proche de la fréquence demandée, éventuellement en ajustant aussi la valeur de N .


[1]  Exprimer la fréquence f du signal en fonction de N , q et τ . Exprimer la phase θ_n de l'échantillon d'indice n de la table (pour $n = 0, 1, \dots, N - 1$). Attention : le dernier échantillon ($n = N - 1$) ne doit pas se répéter lors de la lecture cyclique de la table.

[2] Écrire le code Python qui effectue, pour une valeur de N donnée, l'échantillonnage d'une sinusoïde et stocke les valeurs dans un tableau `numpy.ndarray`. Tracer les échantillons avec `matplotlib.pyplot.plot`.


Si N est fixé, il est possible de faire varier la fréquence en changeant q .


[3]  Pour $N = 100$, quelles sont les 5 plus grandes fréquences que l'on peut réaliser ? Est-il possible de générer un signal de fréquence 40000 Hz si $N = 100$? Conclure sur cette méthode.

Pour une fréquence f_d demandée, on cherche les valeurs optimales de N et de q permettant d'obtenir une fréquence effective f aussi proche que possible de la fréquence f_d . Dans un premier temps, on choisit une valeur de N notée N_d (valeur demandée).


[4]  Exprimer q en fonction de f_d , τ et N_d .


On choisit cette valeur mais si celle-ci est nulle, on prend $q = 1$. Une fois que q est déterminé, on calcule la valeur de N pour que la fréquence effective soit au plus proche de la fréquence f_d .

[5]  Exprimer cette valeur de N en fonction de q , τ et f_d .

[6]  Exprimer la valeur maximale de N en fonction de f_d et τ (si N dépasse cette valeur alors q est nul).


Le plus grand nombre entier inférieur ou égal à un nombre réel (à virgule flottante) peut être obtenu avec la fonction `np.floor`.


[7]  Écrire une fonction `f_echant(f_d, N_d)` qui renvoie `q, f_e, N, f` (valeur de q , fréquence d'échantillonnage, nombre d'échantillons par période, fréquence effective).

[8]  Pour $f_d = 40000$ Hz et $N_d = 100$, quelle est la fréquence effective ? Même question pour $f_d = 40100$ Hz. Est-il possible d'améliorer la précision en augmentant N_d ?


[9] Dans un script à exécuter sur le PC, programmer le convertisseur numérique-analogique afin d'obtenir un signal sinusoïdal d'une fréquence f_d demandée et d'un nombre $N_d = 100$ demandé. Pour calculer la période d'échantillonnage en microsecondes, on écrira `T_e=q*0.2`.


[10] Observer la sortie SA1 à l'oscilloscope. Mesurer la fréquence pour $f_d = 40100$ Hz et pour $f_d = 105$ Hz. Vérifier que les valeurs mesurées sont en accord avec les valeurs effectives calculées.

[11]  Avec la fonction FFT de l'oscilloscope, faire une analyse spectrale du signal. Peut-on voir sur ce spectre une distorsion par rapport à une sinusoïde parfaite ?

[12]  La valeur de q étant déterminée, exprimer la variation de fréquence effective Δf que l'on obtient en faisant varier N d'une unité. Montrer que si N est grand :

$$\Delta f \approx \frac{1}{N^2 q \tau} \quad (1)$$


[13]  En déduire la précision relative $\frac{\Delta f}{f}$ en fonction de N . Compte tenu de la valeur maximale de N , en déduire la précision relative minimale en fonction de f et τ .


[14]  Comparer la précision relative pour $f = 10$ kHz et pour $f = 100$ Hz. Conclure.


3.b. Échantillonnage de plusieurs périodes


Lorsque la mémoire interne du convertisseur numérique-analogique est assez grande (c'est le cas de la SysamSP5), on peut échantillonner plusieurs périodes. De cette manière, le nombre d'échantillons par période n'est plus nécessairement entier. Le nombre d'échantillons stockés dans la table est N . Soit P le nombre de périodes à échantillonner (nombre entier). Le nombre d'échantillons par période (nombre réel) est $N_p = N/P$. La période d'échantillonnage est $T_e = q\tau$.


La valeur de N est fixée à $N = 100000$. L'ajustement de la fréquence du signal se fait en modifiant les nombres entiers P et q . Initialement, on demande une valeur de P notée P_d .


[15]  Exprimer la fréquence f du signal en fonction de P , N , q et τ .


[16]  Si l'on souhaite que N_p soit proche de 100, quelle valeur faut-il donner à P_d (valeur de P demandée) ?

[17]  Exprimer q en fonction de P_d , N , f_d et τ . Si la valeur de q est nulle, on prend $q = 1$.


[18]  Lorsque la valeur de q est déterminée, quelle est la valeur optimale de P , en fonction de f_d , N , q et τ ?

[19]  Écrire une fonction `fechant_bis(fd, Pd, N)` qui renvoie q , f_e , P , f .

[20]  Tester la fonction avec $f_d = 40100$ Hz et $P_d = 1000$ puis avec $f_d = 105$ Hz et $P_d = 1000$.

[21]  La valeur de q étant déterminée, exprimer la variation de fréquence effective Δf que l'on obtient en faisant varier P d'une unité. Montrer que :

$$\frac{\Delta f}{f} = \frac{1}{P} \quad (2)$$

[22]  Comparer la précision relative à celle de la première méthode. Dans quel domaine de fréquence cette méthode est-elle plus précise ?

[23] Générer le tableau contenant les N échantillons répartis sur exactement P périodes.


[24] Dans un script à exécuter sur le PC, programmer le convertisseur numérique-analogique afin d'obtenir un signal sinusoïdal d'une fréquence f_d demandée et d'un nombre $P_d = 100$ demandé. Analyser le signal généré pour $f_d = 40100$ Hz.


3.c. Génération d'un signal avec des harmoniques

[25] Après avoir choisi la meilleure méthode, générer un signal périodique de fréquence 440 Hz comportant des harmoniques de rang 1, 3 et 5, avec environ 100 échantillons par période.

[26] Observer le signal avec l'oscilloscope. Réaliser l'analyse spectrale au moyen de la fonction FFT. Cette analyse confirme-t-elle les amplitudes des harmoniques programmées ? Y-a-t-il d'autres harmoniques visibles sur le spectre ?

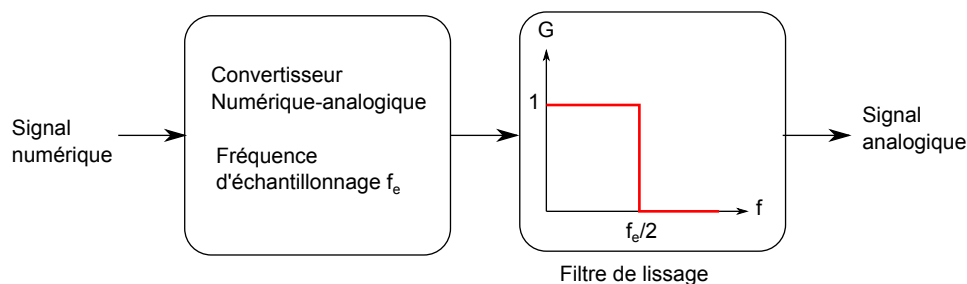
4. Filtre de lissage

[27]  En utilisant la première méthode (échantillonnage d'une seule période), générer un signal sinusoïdal de fréquence demandée $f_d = 100$ Hz. Observer en détail la forme du signal. Que remarque-t-on ?

[28]  Au moyen de la fonction FFT de l'oscilloscope, faire une analyse spectrale du signal. La fréquence d'échantillonnage est-elle visible sur ce spectre ?


Le signal obtenu en sortie du convertisseur numérique-analogique n'est pas tout à fait le signal analogique souhaité, car la tension reste constante pendant une durée égale à la période d'échantillonnage, ce qui donne à cette tension une allure en « marche d'escalier ».


Lorsque la fréquence d'échantillonnage est connue, on peut remédier à ce problème en plaçant en sortie du convertisseur N/A un filtre de lissage. La figure suivante montre le principe du lissage :





Le filtre de lissage doit avoir en principe une fréquence de coupure égale à la moitié de la fréquence d'échantillonnage. Idéalement, il doit avoir un gain constant dans la bande passante (et un déphasage linéaire) et un gain nul dans la bande atténuée. En pratique, on doit bien sûr se contenter de la réponse fréquentielle réelle du filtre analogique utilisé.

La carte Sysam SP5 comporte bien un filtre de lissage pour sa sortie SA1 mais il est conçu pour la fréquence d'échantillonnage la plus grande, soit 5 MHz. Pour des fréquences d'échantillonnage de l'ordre de 10 kHz ou moins, ce filtre est inefficace. Il faut donc dans ce cas ajouter un filtre passe-bas en sortie.

[29]  Faire le schéma d'un filtre de lissage du premier ordre (filtre RC). Pour $C = 10$ nF, quelle valeur de R faut-il approximativement ?

[30]  Mettre en place le filtre de lissage. Observer simultanément le signal en sortie du convertisseur et en sortie du filtre. Le lissage fonctionne-t-il ?

[31]  Augmenter la fréquence. Quel est l'inconvénient du filtre de lissage ?

[32]  Dans le cas où on veut générer un signal périodique avec des harmoniques, le filtre de lissage altère-t-il la forme du signal ?