

Quantification et amplification

1. Introduction

L'objectif de ce TP est d'étudier l'influence de la quantification qui se fait lors de la conversion analogique-numérique. On étudiera le bruit de quantification et le rapport signal sur bruit associé. On verra aussi pourquoi la numérisation d'un signal de très faible amplitude nécessite une amplification de tension et comment réaliser cette amplification au moyen d'un amplificateur linéaire intégré (ALI).

Matériel :

- ▷ ALI TL081 (ou 741).
- ▷ Plaque d'essai.
- ▷ Fils avec embouts dénudés.
- ▷ Résistances 10 k Ω et 100 k Ω .
- ▷ Deux condensateurs céramiques de 100 nF.
- ▷ Carte Sysam SP5.
- ▷ Générateur de signaux.
- ▷ Oscilloscope.
- ▷ Alimentation -15/0/+15 V.
- ▷ Boîte AOIP de 10 Ω .

2. Quantification

2.a. Bruit de quantification

La quantification d'un signal analogique intervient au cours de la conversion analogique-numérique : la tension $u(t)$ à un instant $t = t_k$ est convertie en un nombre entier codé sur p bits. Le convertisseur analogique-numérique (CAN) est généralement précédé d'un amplificateur (dont on verra le fonctionnement plus loin). Supposons que le système amplificateur-CAN soit configuré pour numériser des tensions dont la valeur est dans l'intervalle $[-U_m, U_m]$. La largeur de la plage de tensions pouvant être numérisées est donc $U_0 = 2U_m$. On définit le *pas de quantification* par :

$$\delta = \frac{U_0}{2^p} \quad (1)$$

Les valeurs possibles des nombres entiers sont :

$$0, \delta, 2\delta, 3\delta, \dots, 2^{p-1}\delta$$

Le CAN donne le nombre entier n tel que $u_k = -U_m + n\delta$ soit la meilleure approximation de $u(t_k)$. Considérons l'écart entre cette approximation et la valeur exacte, nommée *erreur de quantification* :

$$\epsilon_k = u_k - u(t_k)$$


On a par définition :

$$\epsilon_k \in \left[-\frac{\delta}{2}, \frac{\delta}{2} \right]$$

Lorsqu'on réalise un grand nombre de conversions (échantillonnage temporel du signal), l'indice k varie et l'erreur de quantification présente les propriétés d'une variable aléatoire (si l'amplitude du signal est beaucoup plus grande que δ). La valeur fournie par le CAN à l'instant $t_k = kT_e$ s'écrit :

$$u_k = u(t_k) + \epsilon_k$$

L'effet de la quantification est donc équivalent à ajouter au signal une variable aléatoire. Le signal ϵ_k est le *bruit de quantification*. Plus p est grand, plus δ est petit donc plus le bruit de quantification est faible.


[1]  Calculer la valeur de δ pour la carte Sysam SP5 (convertisseur 12 bits) utilisée avec le calibre $[-10, 10]$ volts.


On se propose de faire une simulation de Monte-Carlo du bruit de quantification. On utilise pour cela le module `numpy.random`.

[2] Attribuer des valeurs à p et U_m et calculer U_0 et δ . Au moyen de la fonction `numpy.random.uniform`, générer un tableau de $N = 100\,000$ valeurs réelles de tension u réparties selon une loi de probabilité uniforme.

[3] Au moyen de la fonction `numpy.around`, calculer le tableau contenant le résultat de la quantification, c'est-à-dire les entiers n . Calculer le tableau contenant les valeurs de ϵ_k divisées par δ .


[4] Vérifier graphiquement que ϵ_k est bien dans l'intervalle $[-\frac{\delta}{2}, \frac{\delta}{2}]$.


[5]  Au moyen de la fonction `matplotlib.pyplot.hist`, tracer l'histogramme des valeurs de ϵ_k . Que peut-on en déduire sur la densité de probabilité de cette variable aléatoire ?

[6]  Démontrer que la variance du bruit de quantification est donnée par :

$$\sigma^2 = \frac{\delta^2}{12} \quad (2)$$

La valeur efficace du bruit de quantification est l'écart-type, c'est-à-dire σ .

[7]  Calculer l'écart-type des valeurs de ϵ_k et vérifier numériquement le résultat précédent.

[8]  Lorsque p augmente de 1 bit, de quel facteur l'écart-type σ est-il diminué ? Exprimer cette diminution en décibel.

2.b. Numérisation d'un signal sinusoïdal

On cherche à étudier expérimentalement le bruit de quantification lorsqu'on numérise un signal sinusoïdal. Voici le script Python pour effectuer la numérisation dans la plage de tensions $[-10\text{ V}, 10\text{ V}]$. La carte Sysam SP5 effectue une numérisation en $p = 12$ bits.

[analyseSignal.py](#)

```
import numpy as np
import matplotlib.pyplot as plt
from numpy.fft import fft, ifft
import pycanum.main as pycan
```

```

can = pycan.Sysam("SP5")
T=1.0
fe = 100000
te=1.0/fe
N = int(T/te)
Um = 10
can.config_entrees([0],[Um])
can.config_echantillon(te*10**6,N)
can.acquerir()
t=can.temps()[0]
u=can.entrees()[0]
can.fermer()

N = len(u)
te = t[1]-t[0]
fe=1/te
tfd = fft(u)
spectre = np.absolute(tfd)*2/N
freq = np.arange(N)*1/(N*te)
A = spectre.max()
SdB = 20*np.log10(spectre/A)

plt.figure()
plt.plot(freq,SdB)
plt.xlabel('f (Hz)')
plt.ylabel('S (dB)')
plt.grid()
plt.ylim(-120,0)
plt.show()

```

[9] Effectuer avec ce script la numérisation d'un signal sinusoïdal d'amplitude $A = 1,0$ V et de fréquence environ 1000 Hz.


[10] Ajuster l'échelle verticale du spectre afin que la totalité du bruit soit visible.

[11]  Outre le bruit de quantification, quelles sont les autres causes possibles du bruit ?

On définit le *rapport signal sur bruit* comme le rapport de la valeur efficace du signal par la valeur efficace du bruit. Pour un signal sinusoïdal d'amplitude A , ce rapport s'écrit :

$$r = \frac{A}{\sqrt{2}\sigma} \quad (3)$$

Le rapport signal sur bruit est généralement exprimé en décibel, soit $r_{dB} = 20 \log(r)$.

[12]  Dans le cas où le bruit est dominé par le bruit de quantification, exprimer r_{dB} en fonction de A/U_m et de p . Ajouter dans le script le calcul et l'affichage de ce rapport signal sur bruit.

Pour mesurer la valeur efficace du bruit, il faut faire un filtrage afin d'éliminer la composante sinusoïdale. Ce filtrage peut être fait en annulant des termes de la TFD de manière à faire disparaître les deux raies associées à cette composante. Ce type de filtrage, qui consiste à modifier directement la TFD, est un filtrage dans le domaine fréquentiel. L'application de la transformée de Fourier inverse à la TFD modifiée permet d'obtenir la représentation temporelle du bruit. Voici le code python à ajouter au script pour faire ce traitement :


```


i = np.argmax(spectre[0:N//2])
f = i*fe/N
print("f = %f Hz"%f)
di = 100 # demi-intervalle à modifier
tfd[i-di:i+di] = 1e-12
tfd[N-1-i-di:N-1-i+di] = 1e-12
spectre = np.absolute(tfd)*2/N
SdB = 20*np.log10(spectre/A)
bruit = ifft(tfd) # transformée de fourier inverse
sigma_bruit = bruit[N//4:N-N//4].std()
rdB = 20*np.log10(u.std()/sigma_bruit)
print('rdB exp =%f'%rdB)


plt.figure() # spectre modifié
plt.plot(freq,SdB)
plt.xlabel('f (Hz)')
plt.ylabel('S(dB)')
plt.grid()
plt.ylim(-120,0)

plt.figure() # bruit en fonction du temps
plt.plot(t,bruit)
plt.xlabel('t (s)')
plt.ylabel('bruit (V)')
plt.grid()


```


[13]  À quoi sert la ligne `i = np.argmax(spectre[0:N//2])` ?


[14]  Observer le spectre modifié pour une amplitude $A = 1,0 \text{ V}$. Augmenter la valeur de di jusqu'à éliminer complètement les deux raies du signal sinusoïdal.

[15]  Observer la représentation temporelle du bruit. Que constate-t-on ?

On constate que la représentation temporelle du signal filtré n'est pas correcte au début et à la fin de l'intervalle de temps. On s'intéresse à la partie du signal qui reproduit correctement le bruit.

[16]  Expliquer la ligne `sigma_bruit = bruit[N//4:N-N//4].std()`.

[17]  Faire des mesures du rapport signal sur bruit (en décibel) pour une sinusoïde de fréquence d'environ 1000 Hz et pour les amplitudes $A = 1,0 \text{ V}$, $A = 100 \text{ mV}$ et $A = 10 \text{ mV}$. Relever aussi le rapport signal sur bruit théorique pour le bruit de quantification. Reporter ces valeurs dans un tableau. Refaire ces mesures pour une fréquence d'environ 10 kHz.

[18]  Conclusion : le bruit observé sur le signal numérisé est-il du bruit de quantification ?

Le filtrage du signal pour extraire le bruit peut aussi être fait dans le domaine temporel, au moyen d'un filtre de convolution. Cette méthode est présentée en annexe.

3. Amplification

3.a. Amplificateur linéaire intégré

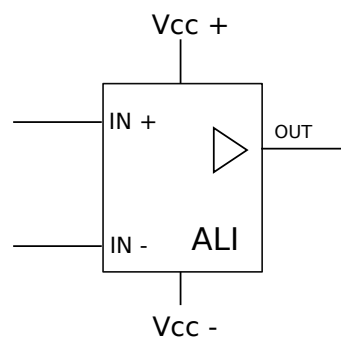
Lorsque le rapport A/U_m est trop faible, le rapport signal sur bruit est trop faible. Pour l'augmenter, il faut augmenter l'amplitude du signal avant sa numérisation, au moyen d'un *amplificateur de tension*.

Pour réaliser un amplificateur de tension, on utilise un *amplificateur linéaire intégré* (ALI). Il s'agit d'un circuit intégré qui fonctionne avec une alimentation. Il comporte 5 bornes :

- ▷ Vcc+ : borne d'alimentation positive.
- ▷ Vcc- : borne d'alimentation négative.
- ▷ IN + : entrée non inverseuse.
- ▷ IN - : entrée inverseuse.
- ▷ OUT : sortie.

Le circuit interne de l'ALI TL081 est donné en annexe. Il comporte une quinzaine de transistors et quelques résistances.

Voici le schéma d'un ALI :



La tension de sortie est comprise entre Vcc- et Vcc+. En conséquence, si l'on veut que la tension de sortie puisse prendre des valeurs négatives par rapport à la masse, la tension d'alimentation Vcc- doit être négative par rapport à la masse. On doit donc utiliser une alimentation double, par exemple une alimentation -15/0/+15 V. La borne -15 V de l'alimentation est connectée à la borne Vcc- de l'ALI, la borne +15 V à la borne Vcc+ et la borne 0 à la masse du circuit.

L'ALI est un amplificateur différentiel : la tension de sortie (par rapport à la masse) est égale à la différence des tensions entre IN+ et IN- multipliée par un gain. Le gain de l'ALI est très élevé (de l'ordre de 100 000) car l'ALI est conçu pour être utilisé avec une rétroaction. Par ailleurs, l'ALI se comporte comme un filtre passe-bas, c'est-à-dire que son gain diminue lorsque la fréquence augmente. Pour définir le gain de L'ALI, on introduit les tensions complexes en régime sinusoïdal : $\underline{V_s}$ la tension de sortie, $\underline{V_-}$ la tension de l'entrée inverseuse et $\underline{V_+}$ celle de l'entrée non-inverseuse. Ces trois tensions sont des différences de potentiel par rapport à la masse. Il faut bien sûr que la masse soit à un potentiel déterminé par rapport aux tensions d'alimentation. L'ALI est caractérisé par la relation linéaire suivante :

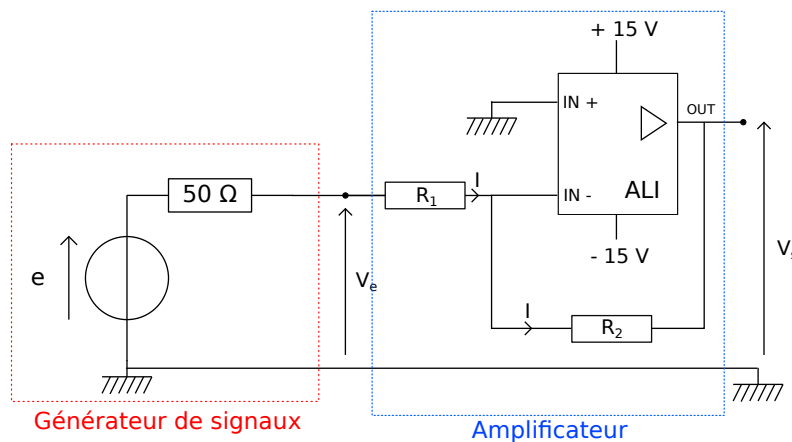
$$\underline{V_s} = \frac{\mu}{1 + j \frac{f}{f_c}} (\underline{V_+} - \underline{V_-}) \quad (4)$$

μ est le gain à fréquence nulle, dont la valeur est de l'ordre de 10^5 . La fréquence de coupure f_c est de l'ordre de 10 Hz. Cette relation est valable lorsque l'ALI fonctionne en régime linéaire.

Il peut arriver que l'ALI soit en régime non linéaire : dans ce cas, la tension de sortie est égale soit à V_{cc+} (saturation positive), soit à V_{cc-} (saturation négative). En réalité, la tension de saturation est souvent légèrement inférieure en valeur absolue à la tension d'alimentation (d'environ 0,7 V). Dans le cas de l'amplification, la saturation devra être évitée et l'ALI devra fonctionner en régime linéaire.

3.b. Amplificateur de tension

L'ALI ne peut être utilisé en boucle ouverte car son gain est beaucoup trop grand et sa fréquence de coupure beaucoup trop faible. On doit lui ajouter une *rétroaction*. Voici le circuit amplificateur que nous allons utiliser :



Ce circuit comporte une rétroaction de la sortie de l'ALI vers l'entrée IN-. Cette rétroaction est négative, ce qui est une condition nécessaire pour que l'ALI fonctionne bien en régime linéaire et que la relation (4) soit vérifiée.

Les entrées IN- et IN+ de l'ALI fonctionnent avec des courants extrêmement faibles (micro ou nano ampères), en général négligeables devant les autres courants du circuit. Le courant qui traverse la résistance R_1 (d'intensité I) traverse donc aussi la résistance R_2 . On a les relations suivantes :

$$\begin{aligned} \underline{V_e} - \underline{V_-} &= R_1 I \\ \underline{V_+} &= 0 \\ \underline{V_s} - \underline{V_-} &= -R_2 I \end{aligned}$$

Au moyen de ces trois relations et de la relation (4), on obtient :

$$\frac{\underline{V_s}}{\underline{V_e}} = \frac{-\frac{R_2}{R_1}}{1 + \left(1 + \frac{R_2}{R_1}\right) \frac{1+j\frac{f}{f_c}}{\mu}} \quad (5)$$

Le rapport R_2/R_1 est inférieur à 1000 et $\mu \approx 10^5$. Lorsque $f = 0$ on a donc :


$$\frac{\underline{V_s}}{\underline{V_e}} \approx -\frac{R_2}{R_1} \quad (6)$$

Si $R_2 > R_1$, ce circuit réalise bien une amplification de la tension. Cet amplificateur est qualifié d'inverseur car il introduit, en plus de l'amplification, un changement de signe du signal. La fonction de transfert est celle d'un filtre passe-bas du premier ordre (ce qui prouve la stabilité du système), dont la fréquence de coupure est :

$$f'_c = \frac{\mu f_c}{1 + \frac{R_2}{R_1}} \quad (7)$$

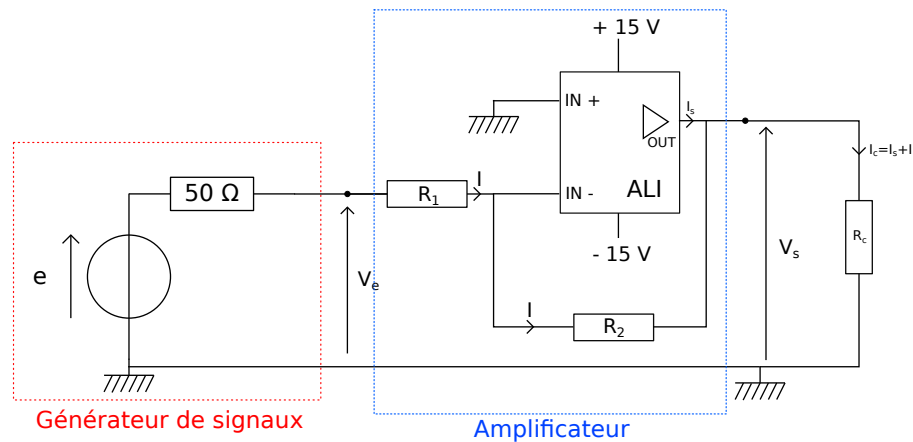
Le gain dans la bande passante est :

$$G_0 = \frac{R_2}{R_1} \quad (8)$$

[19]  Si le gain G_0 est égal à 10, quelle est la fréquence de coupure ? Même question pour un gain de 100. Quelle relation simple existe-t-il entre le gain et la largeur de la bande passante ?

Remarque : le gain dans la bande passante peut être obtenu rapidement à l'aide du modèle d'ALI idéal. Dans ce modèle, le gain μ est considéré comme infini donc, d'après la relation (4), le fonctionnement de l'ALI en régime linéaire suppose que $V_+ \approx V_-$. L'utilisation de cette relation et des trois relations écrites plus haut permet de retrouver rapidement la relation (6). Cependant, ce modèle d'ALI idéal ne permet pas d'étudier la stabilité de l'amplificateur ni de trouver sa bande passante. La stabilité est nécessaire pour que l'ALI fonctionne effectivement en régime linéaire (et non en saturation). D'une manière générale, la présence d'une rétroaction négative et l'absence de rétroaction positive (comme dans le circuit étudié ici) garantit la stabilité du système.

Pour un ALI de faible puissance, comme celui que nous allons utiliser, le courant de sortie ne peut dépasser environ 20 mA. Pour comparaison, le courant de sortie d'un générateur de signaux peut aller jusqu'à environ 200 mA sur les générateurs de laboratoire courants. En revanche, le montage amplificateur présenté ci-dessus présente une impédance de sortie extrêmement faible (très inférieure à 1Ω), à comparer à la résistance de sortie de 50Ω du générateur.



Soit I_s l'intensité du courant qui sort de l'ALI. L'intensité du courant dans la résistance de charge est $I_c = I_s + I$. Si la résistance de charge est inférieure à 1000Ω , elle est au moins 100 fois plus petite que R_2 , ce qui fait que I est négligeable devant I_c . On a donc $I_c \approx I_s$. L'intensité I_s ne peut dépasser environ 20 mA.

[27] Si la tension d'entrée V_e est sinusoïdale et d'amplitude $A = 10 \text{ mV}$, quelle valeur R_c doit-elle avoir au minimum pour que I_s reste toujours inférieure à 20 mA ?

[28] Placer en sortie une résistance variable (boîte AOIP de 10Ω). Vérifier le résultat précédent.

[29] Faire varier R_c tout en maintenant le courant de sortie en dessous de sa limite. Que constate-t-on sur la tension V_s ?

[30] Refaire la même expérience en plaçant R_c directement en sortie du générateur, avec une tension sinusoïdale d'amplitude 100 mV. Que peut-on en conclure sur la résistance de sortie du montage amplificateur ?

4. Annexe : extraction du bruit par convolution

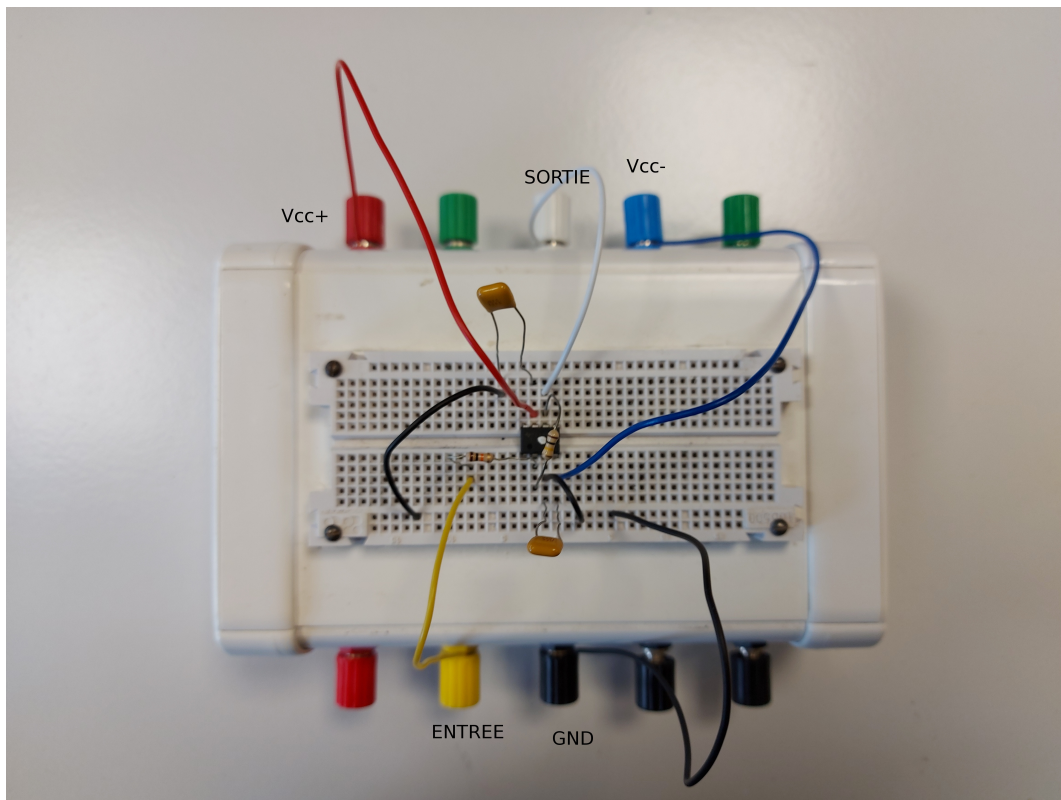
Voici le code qui permet de filtrer un signal sinusoïdal avec un filtre de convolution coupe-bande permettant d'éliminer la partie sinusoïdale du signal, afin d'extraire le bruit.

```
P=500 # 2P+1 = ordre du filtre
c1=f*0.94/fe # fréquence de coupure basse
c2=f*1.06/fe # fréquence de coupure haute
b = scipy.signal.firwin(numtaps=2*P+1,cutoff=[c1,c2],pass_zero=True>window='hann',nyq=0.5)
y = scipy.signal.convolve(u,b,mode='same') # filtrage par convolution
sigma_bruit = y[N//4:N-N//4].std()
rdB = 20*np.log10(u.std()/sigma_bruit)
print('rdB exp =%f'%rdB)

plt.figure()
plt.plot(t,y)
plt.xlabel('t (s)')
plt.ylabel('bruit (V)')
plt.grid()
```

Le filtrage par convolution (domaine temporel) donne les mêmes résultats que le filtrage direct de la TFD (domaine fréquentiel). Les effets de bords (au début et à la fin de l'intervalle de temps) sont similaires. Le filtrage direct de la TFD est cependant plus facile à réaliser car on voit immédiatement l'effet sur le spectre. Si le signal à filtrer est périodique, avec plusieurs harmoniques à éliminer, le filtrage direct de la TFD est de loin la méthode la plus simple car il faudrait de toute façon utiliser la FFT pour calculer la réponse impulsionnelle du filtre de convolution (à partir de la réponse fréquentielle du filtre).

5. Annexe : câblage du circuit sur plaque d'essai



Les bornes du montage doivent être reliées aux fiches bananes femelles par des fils, en respectant les couleurs (couleur du fil = couleur de la fiche) :

- ▷ Fil rouge : Vcc + et borne +15 V de l'alimentation.
- ▷ Fil bleu : Vcc- et borne -15 V de l'alimentation.
- ▷ Fil noir : GND du montage et borne 0 de l'alimentation.
- ▷ Fil jaune : entrée de l'amplificateur.
- ▷ Fil blanc : sortie de l'amplificateur.

On branchera aussi un condensateur entre Vcc+ et GND et un condensateur entre Vcc- et GND. Il s'agit de condensateurs de filtrage, qui ont pour fonction d'éliminer d'éventuelles perturbations des tensions d'alimentation aux bornes de l'ALI.