ECHANTILLONNAGE D'UN SIGNAL ANALOGIQUE

Extraits du programme :
Partie 2 - Formation expérimentale

Analyse spectrale.	Mettre en évidence le phénomène de repliement du spectre provoqué par l'échantillonnage avec un oscilloscope numérique ou une carte d'acquisition. Choisir les paramètres d'une acquisition numérique destinée à une analyse spectrale afin de respecter la condition de Nyquist-Shannon, tout en optimisant la résolution spectrale.

Partie 3 - Formation disciplinaire

Notions et contenus	Capacités exigibles
1.4. Électronique numérique	
Échantillonnage.	**Expliquer l'influence de la fréquence d'échantillonnage.**
Condition de Nyquist-Shannon.	Utiliser la condition de Nyquist-Shannon. **Mettre en évidence le phénomène de repliement de spectre au moyen d'un oscilloscope numérique ou d'un logiciel de calcul numérique.**
Analyse spectrale numérique.	Choisir les paramètres (durée, nombre d'échantillons, fréquence d'échantillonnage) d'une acquisition numérique afin de respecter la condition de Nyquist-Shannon. Capacité numérique : calculer, à l'aide d'un langage de programmation, la transformée de Fourier discrète d'un signal numérique.

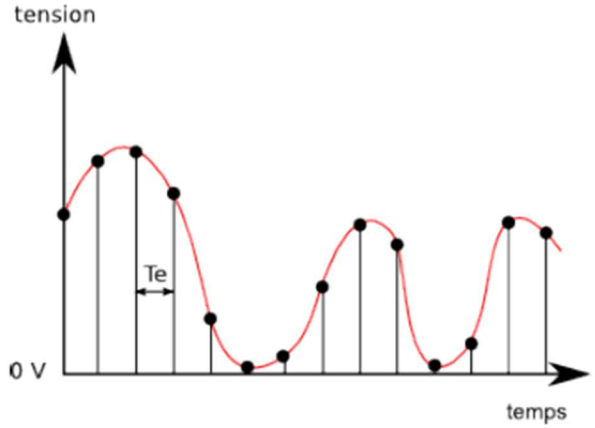
La plupart des signaux physiques à traiter ou à transmettre (signaux en sortie de capteurs, voie humaine ....) sont **analogiques:** ils évoluent de façon continue au cours du temps et prennent leurs valeurs dans un espace continu de valeurs.
Pour réaliser un traitement ou une transmission numérique, il faudra donc dans un premier temps convertir le signal analogique en signal **numérique**. Cette conversion s'effectue en général en deux étapes : une discrétisation en temps : l'échantillonnage et une discrétisation en valeur, la conversion analogique numérique CAN, qui fera l'objet d'un autre TP.

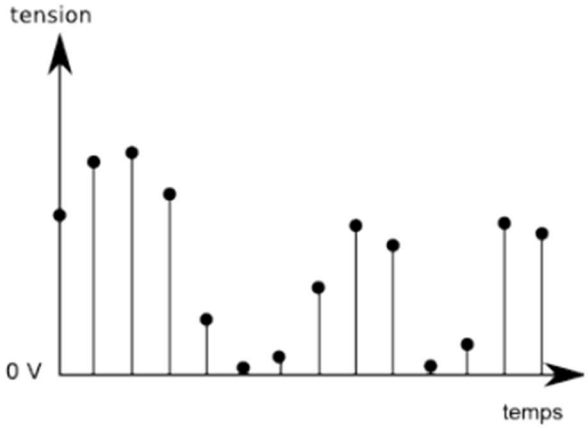
Matériel :
Centrale d'acquisition Sysam 5, oscilloscope, logiciels Latis-Pro et un éditeur python.

1- Préparation : échantillonnage d'un signal analogique :

a) Principe :

Échantillonner (ou discrétiser) un signal analogique consiste à prendre des valeurs de ce signal (échantillons) à des instants donnés discrets (instants d'échantillonnage) régulièrement espacés.
Les logiciels d'acquisition (comme Latis Pro) transforment **un signal analogique** en **un signal numérique** grâce à un convertisseur analogique-numérique (CAN).





Le CAN va **échantillonner** N points espacés d'une durée Te appelée **période d'échantillonnage**, et variable entre une valeur minimale (T_{emin} = 100 ns pour la carte Sysam 5), et une valeur maximale (environ 1000 s) utilisée pour des acquisitions de signaux variant lentement.

La fréquence d'échantillonnage est Fe = 1 / Te. La durée totale d'acquisition est donc : $\Delta t = N.\ T_e$

Il est intuitif que le spectre du signal numérique ne sera pas le même que celui du signal analogique réel, mais qu'il en sera d'autant plus proche que la densité des échantillons sera importante, donc que F_e sera grande.

On démontre que lorsqu'un signal sinusoïdal de fréquence f_0 est échantillonné avec une fréquence d'échantillonnage F_e, le signal échantillonné possède plusieurs fréquences dont les plus basses sont f_0 et $F_e - f_0$.

Q1. On numérise un signal de fréquence f_0 = 100 Hz à Fe = 1000 Hz.
Visualiser le signal analogique et le signal échantillonné sur
https://physique.ostralo.net/echantillonnage/
Comparer la période du signal analogique et celle du signal échantillonné. L'échantillonnage est-t-il correct ?
Représenter le spectre du signal échantillonné dans l'intervalle [0, $F_E/2$].

b) Repliement du spectre :

Q2. Mêmes questions si l'on numérise un signal de fréquence f_0 = 900 Hz à F_e = 1000 Hz.
Peut-on récupérer le signal analogique ? Dans ce dernier cas, on parle de **repliement du spectre.**

c) Critère de Shannon :

Q3. Dans le cas général, si l'on souhaite observer un spectre correct dans l'intervalle [0, $F_E/2$], quelle inégalité doivent vérifier F_e et f_0 ?

Théorème de Shannon :
On ne peut numériser correctement un signal analogique que si celui-ci a un spectre de fréquence maximale f_{max} inférieur à $F_e/2$.

2- Réalisation d'une première analyse spectrale :

On utilise la centrale d'acquisition Sysam – SP5 et le logiciel LatisPro.

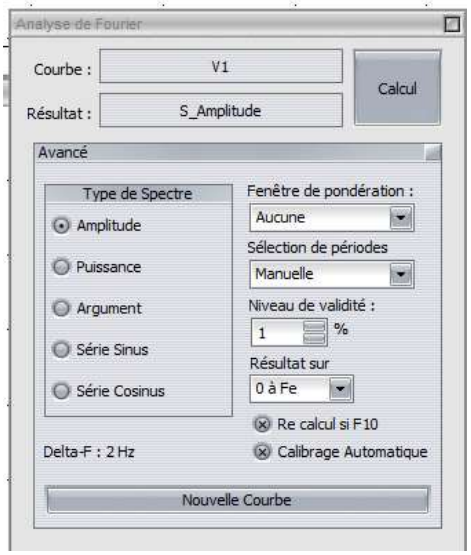
Dans la fenêtre « Acquisition » lire (en passant la souris) la période d'échantillonnage minimale T_{emin} de la carte d'acquisition. En déduire la fréquence limite de Shannon de traitement des signaux analogiques.

Générer à l'aide du générateur de la centrale d'acquisition un signal sinusoïdal de fréquence f_o = 900 Hz.
Choisir N = 512 points et une fréquence d'échantillonnage vérifiant le critère de Shannon, noter sa valeur.

Réaliser l'acquisition sur la voie EA0 par exemple de ce signal : on choisira le style de trait « trait avec ronds » pour voir le signal et les points d'acquisition.
Former le spectre du signal acquis sans modifier les réglages par défaut du logiciel : touche F6.
Recopier le spectre obtenu. Est-il correct ?

3- Mise en évidence du phénomène de repliement du spectre :

Echantillonner le signal sinusoïdal de fréquence f_o = 900 Hz à F_e = 1000 Hz avec toujours N = 512 points.
Former le spectre et expliquer son allure.
Remarque : on peut observer le **repliement** en dépliant le menu Avancé dans la fenêtre Traitement/Analyse de Fourier et en demandant le résultat sur 0 à Fe.
Observer le phénomène de repliement du spectre à l'aide d'un signal carré. Interpréter le spectre obtenu.



4- Optimisation de la résolution spectrale :

Réaliser l'acquisition et l'analyse spectrale du signal à f_0 = 900 Hz en respectant le critère de Shannon.
Reprendre la fenêtre Traitement/Analyse de Fourier, mais cette fois déplier le menu Avancé, et choisir la sélection de périodes **manuelle** : deux curseurs apparaissent sur le signal ; on les déplace à la souris.
Sélectionner un nombre **non-entier** de périodes et lancer le calcul (Enter).
Recopier ou enregistrer le spectre obtenu. Est-il correct ?
L'explication est la suivante : lorsque vous ne faites aucune sélection de durée, le logiciel suppose que la période du signal est la durée entre deux passages par une valeur donnée avec une pente de même signe ; pour des signaux simples, c'est vrai, l'analyse marche.
Mais si le signal est complexe, ou que vous sélectionnez comme ici une durée d'analyse T_a, le logiciel choisit pour fréquence fondamentale $1/T_a$ (en général c'est faux).
La fréquence associée $\delta f = 1 / T_a$ est la résolution de l'analyseur de spectre ; pour le logiciel d'acquisition, le spectre du signal est donc composé des fréquences 0, δf, 2 δf, etc... : on ne pourra mesurer une fréquence dans le spectre qu'à δf près.
Remarque : δf est indiquée dans la fenêtre de calcul du spectre.
Afin d'éviter ce problème, on peut :

- Soit sélectionner manuellement un nombre de périodes entier du signal, s'il est « simple » ;
- Soit s'arranger pour que δf soit le plus petit possible, càd augmenter T_a

Conclure sur les deux conditions permettant de faire une bonne analyse spectrale.

5- Analyse spectrale avec Python :

a) Série de Fourier d'un signal périodique :

D'après le théorème de Fourier, un signal u(t) réel ou complexe périodique et ne présentant par période T qu'un nombre fini de discontinuités, est égal à la somme d'une série trigonométrique qu'on appelle sa série de Fourier.

Le signal u(t) s'écrit alors :

$$u(t) = A_0 + \sum_{n=1}^{+\infty} [A_n \cos(n\omega t) + B_n \sin(n\omega t)]$$

L'amplitude de l'harmonique de rang n est alors $\sqrt{A_n^2 + B_n^2}$.

Pour les calculs, il est commode d'introduire la série de Fourier sous forme complexe, en généralisant à des fréquences négatives :

$$u(t) = \sum_{n=-\infty}^{+\infty} \underline{C_n} \exp(in\omega t)$$

On montre qu'on a :

$$C_0 = A_0 \; ; \; \underline{C_n} = \frac{1}{2}(A_n - \mathrm{j}B_n), soit \; |\underline{C_n}| = \frac{1}{2}\sqrt{A_n^2 + B_n^2} : remarquer \; le \; facteur \frac{1}{2}$$

On admet que les coefficients $\underline{C_n}$ se calculent par :

$$\underline{C_n} = \frac{1}{T}\int_t^{t+T} u(t)\exp(-in\omega t)dt \quad (1)$$

b) Transformée de Fourier discrète (paragraphe facultatif) :

La transformée de Fourier discrète (TFD) est la transformation qui permet de calculer le spectre d'un signal discret, obtenu par échantillonnage d'un signal continu.
La période du signal est en général inconnue : ce sera (comme avec LatisPro par défaut) la durée totale T du signal, la pulsation fondamentale sera alors $\omega = 2\pi/T$.
Pour un échantillonnage de N points sur une durée totale T d'un signal u(t), on obtient N valeurs de u(t) aux instants t_k = k.T/N (0 <= k <= N-1) :

$$u_k = u(t_k)$$

On peut alors écrire :

$$u_k = \sum_{n=-\infty}^{+\infty} \underline{C_n} \exp(in\omega t_k)$$

Une valeur approchée des Cn peut être obtenue d'après (1) par la méthode des rectangles :

$$\underline{C_n} = \frac{1}{T}\sum_{k=0}^{N-1} \frac{T}{N}.u_k.\exp\left(-\frac{in2\pi k}{N}\right) = \frac{1}{N}\sum_{k=0}^{N-1} u_k.\exp(-\frac{in2\pi k}{N}) \quad (2)$$

Remarquer qu'ici T n'est plus la période du signal mais la durée totale d'acquisition.

c) Calcul avec Python : rfft et rfftfreq.

Pour obtenir la TFD d'un signal échantillonné réel, on utilise la fonction rfft du module numpy.fft : rfft(x) s'applique à un signal réel échantillonné sur N points et renvoie une liste des Cn de N/2 valeurs, jusqu'à Fe/2, mais sans le facteur 1/N de l'expression (2). On doit donc multiplier l'argument par 1/N.
On prend le module et on multiplie par 2 et pour avoir les amplitudes $\sqrt{A_n^2 + B_n^2}$.

Pour tracer le spectre, on doit avoir également une liste des fréquences, ce qui s'obtient avec la fonction rfftfreq du module numpy.fft : rfftfreq(N, dt) avec N = nombre d'échantillons, dt = durée entre deux échantillons, renvoie une liste contenant N/2 valeurs positives entre 0 et Fe/2.

d) Manipulations :

Réaliser l'acquisition d'un signal audio simple (« Laaaaaaa» !) à l'aide d'un microphone et de la carte son.
Noter les paramètres d'acquisition.
Réaliser l'analyse spectrale à l'aide de LatisPro.
Exporter le fichier son en format csv, dans votre répertoire personnel.
Ouvrir un éditeur python et charger le fichier TP_FFT_Eleves.py. Sauvegarder sur votre répertoire personnel.
Lire attentivement et compléter le script afin de réaliser le spectre du signal audio.
Comparer avec le résultat donné par LatisPro.
Que représentent les grandeurs Mesure1 et Mesure2 ?
On suppose que les paramètres d'acquisition ne sont pas connus : comment les récupérer à partir des données du fichier csv ?