TP5-23/24. Étude du filtrage numérique.

APPORTER UNE clé USB personnelle et (de préférence) son ORDINATEUR portable personnel

Matériel :

Boîte de composants radio	Plaquette LAB
Oscilloscope Agilent 4 voies d'entrée interfaçable	Générateur BF
Carte d'acquisition SYSAM	ordinateur doté de logiciels (Python, Latis Pro, Regressi)
Alimentation continue [-15,0,15]	Multimètres autonomes et fils

Pré-requis

Compétences expérimentales acquises	en électrocinétique (PCSI/MPSI)
Maîtrise des logiciels Latis-Pro et Python	

Compétences à viser :

Faire une analyse du problème :	
Proposer un protocole de mesure	Tracer un schéma du circuit Choisir des appareils de mesure
Réaliser le protocole expérimentalement :	
Câbler le montage	Régler l'oscilloscope
Identifier les grandeurs à mesurer	Choisir le calibre des multimètres
Faire une validation :	
Faire du traitement de données sous LatiPro ou Python	
Pour chaque grandeur entre grandeur mesurée	calculer l'écart relatif $\frac{\|G_m - G_{theo}\|}{G_m}$ et calculée via une formule semi-théorique
Pendre en compte les incertitudes avec confiance à 95%	Présenter chaque résultat $g = (G_m \pm u_g)$ (unité) vérifier que l'écart est inférieur à l'incertitude
Rédiger un compte-rendu écrit :	
Introduction : buts poursuivis et moyens alloués	
Compte rendu structuré en paragraphes	graphes légendés et gradués
Présence de tableaux de mesures	ou de graphes précis
protocoles explicités	schémas illustratifs
confrontation mesures/formules théoriques	Conclusion/ évaluation des protocoles

Le but de ce TP est de réaliser un filtrage numérique passe-bas ou passe-bande d'un signal T-périodique -avec Latispro ou sous Python- et d'évaluer la limitation introduite par l'échantillonnage.

I. Introduction

1. Qu'est-ce que le filtrage numérique d'un signal ?

C'est une fonction réalisé par un calculateur sur un signal préalablement numérisé par une CAN. Le schéma bloc de cette fonction est en figure 1. Le calculateur peut être un ensemble de circuits intégrés (puces), des processeurs programmables, ou un logiciel dans un ordinateur. C'est ce dernier cas que nous envisageons dans ce TP.

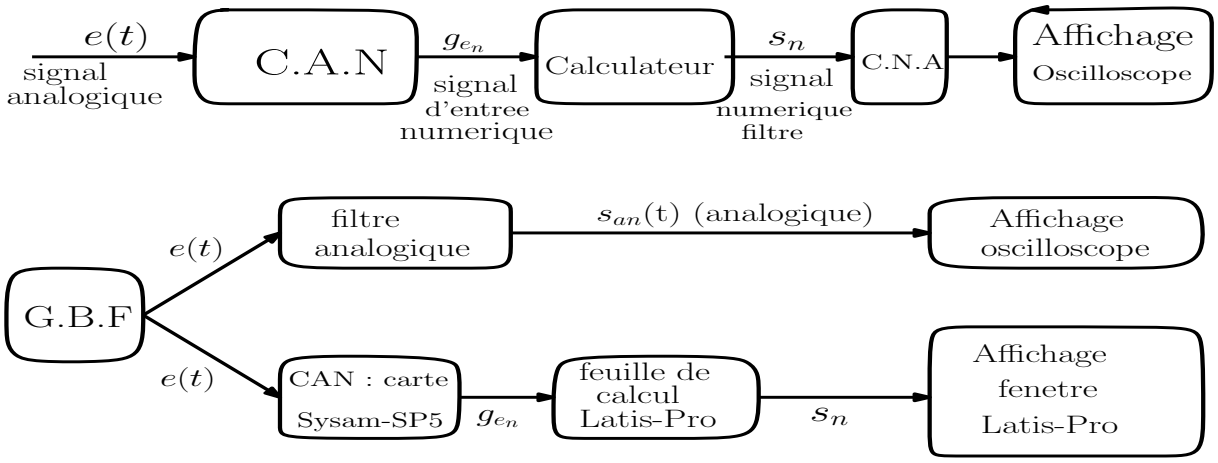

FIGURE 1 – Principe du filtrage numérique, cas où il est réalisé avec Latis Pro.

2. Filtre passe-bas :

Rappeler la forme canonique de la fonction de transfert d'un filtre passe-bas d'ordre 1, de pulsation de coupure ω_c et de transfert statique 1. On note $e(t)$ et $S(t)$ les signaux d'entrée et de sortie du filtre.

On choisit une fréquence de coupure de 100 Hz. Quelles valeurs de R et C conviennent pour réaliser le filtrage analogique ?

Écrire l'équation différentielle vérifiée par $S(t)$ en faisant intervenir une constante de temps τ.

Réponse :

$$\tau \frac{dS}{dt} + S(t) = H_0\, e(t) \quad \text{avec} \quad H_0 = 1 \quad \text{et} \quad \tau = \frac{1}{\omega_c}.$$

II. Filtrage analogique ou filtrage numérique avec Latis-Pro :

On passe au numérique en notant :

— T_e la période d'échantillonnage des signaux,

— g_{e_n} et S_n les valeurs respectives du e(t) et de $S(t)$ aux dates $t_n = n\,T_e$.

1. Calculs de la méthode d'Euler :

- Si la période d'échantillonnage est suffisamment petite, donner une approximation à l'ordre 1 de la dérivée $\left(\frac{\mathrm{dS}}{\mathrm{dt}}\right)_n$ à la date t_n [ou t_{n-1}] :

— 1^{ere} méthode : en faisant intervenir S_n, S_{n+1} et T_e,

— 2^{de} méthode : en faisant intervenir S_n, S_{n-1} et T_e.

Réponse :

$$\left(\frac{\mathrm{d}S}{\mathrm{d}t}\right)_n = \frac{S_{n+1} - S_n}{T_0}$$

- En déduire l'équation aux différences, pour un filtre d'ordre 1, donnant :

— S_{n+1} en fonction de g_{e_n}, τ, T_e et S_n,

— S_n en fonction de $g_{e_{(n-1)}}$, τ, T_e et S_{n-1}.

Dans chacun des deux cas, comment s'appelle le type de méthode d'Euler de résolution d'une équation différentielle d'ordre 1 ?

Réponse :

$$\tau \frac{S_{n+1} - S_n}{T_e} + S_n = g_{e_n} \quad \Leftrightarrow \quad S_{n+1} = (g_{e_n} - S_n) \times \frac{T_e}{\tau} + S_n \quad : \quad methode \ \ d'Euler$$

2. Mise en œuvre des deux types de filtrage :

- Câbler un filtre analogique (R,C) sur plaquette Lab, de tension d'entrée e(t) et de tension de sortie $s_{ana}(t)$.
- On génère au GBF un signal sinusoïdal e(t) d'entrée de fréquence f=100 Hz, d'amplitude crête à crête 10V.
- Faire une acquisition avec la carte Sysam-SP5 sur l'entrée EA0 avec la fréquence d'échantillonnage f_e=5 kHz.
- Feuille de calcul (2^{de} méthode d'Euler) :

Dans Latis-Pro :	
Traitement	Feuille de calcul
Code à taper :	Te=2e-4 fe=1/Te fc=100 tau=1/(2*3.14*fc) S=Table(0) S=S[n-1]+Te/tau*(EA0[n-1]-S[n-1])

- Activer la sortie S de la carte Sysam dans Latis Pro (voir fig. 2), pour qu'on y recueille un signal analogique S(t) issu de la CNA de la carte Sysam :

en allant dans le menu Paramètres/Émission	en cochant Mode GBF
en cochant Sortie Active	en choisissant S dans le menu déroulant

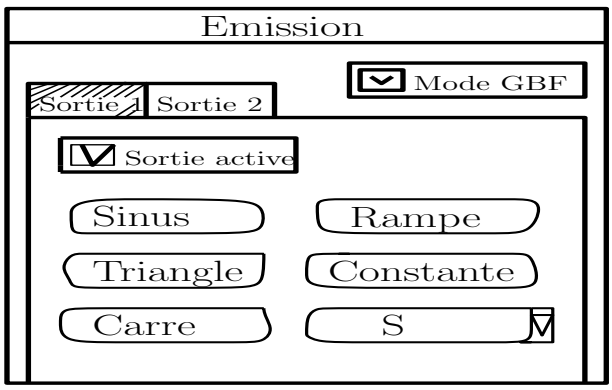

FIGURE 2 – Activer la sortie S de la Sysam.

- Observer dans une même fenêtre Latis pro, le signal d'entrée EA0 et le signal S numériquement filtré. Mesurer les caractéristiques (onglet Outils/Mesures automatiques). Vérifier la cohérence avec les valeurs attendues.
- Recommencer en changeant de fréquence (f=50 Hz, f=500 Hz...).
- Recommencer avec un signal en créneaux pour f=500 Hz et f=50 Hz.
- Si on a le temps, recommencer avec la première méthode d'Euler.

3. Observations à l'oscilloscope :

- Observer à l'oscilloscope, le signal d'entrée e(t) (mettre la voie CH1 sur EA0) et le signal S(t) numériquement filtré puis converti en signal analogique par Latis-Pro et la carte Sysam (mettre CH2 sur la sortie SA1 du boîtier de la carte Sysam d'acquisition).

— Observer les deux voies de l'oscilloscope et comparer avec ce qu'on a observé dans la fenêtre Latispro. Remarquer qu'il n'y a aucune relation de phase entre les deux signaux à l'oscilloscope. Ceci est dû au fait que le traitement numérique prend un temps inconnu et variable.

4. **Influence de f_e :**

   En gardant f_e = 5 kHz, prendre successivement pour fréquence f du signal e(t) les valeurs 50 Hz, 1 kHz, 5 kHz et 6 kHz. Commenter les observations (critère de Shannon ?, repliement de spectre ? éventuelle fréquence repliée ?).

5. **Comparaison des filtres :**
   — Alimenter le filtre passe-bas analogique de f_c=100 Hz câblé sur plaquette LAB avec le signal d'entrée e(t) envisagé dans le filtrage numérique.
   — Observer à l'oscilloscope les signaux de sortie respectifs du filtre numérique (SA1) et du filtre analogique.
   — Conclure quant aux qualités et aux défauts du filtrage numérique par rapport au filtrage analogique.

III. Filtrage numérique sous Python :

Les buts de cette partie est :

- de capter sur la face avant de l'oscilloscope Agilent via une clé USB, et d'enregistrer dans l'ordinateur, un fichier .csv contenant les données [t, e(t)] du signal e(t) d'entrée, généré au GBF ;

- d'écrire un code Python lisant ce fichier, et calculant le signal de sortie qu'en donne un filtre (F) prédéfini ; l'algorithme de ce code peut utiliser la méthode d'Euler ou la fonction odeint ;

- d'écrire un code Python lisant ce fichier, calculant la transformée de Fourier rapide (FFT) et traçant le spectre d'amplitude du signal e(t) ; ce code peut utiliser les fonctions FFT ou RFFT de numpy ;

Notons que Python propose de nombreux modules tout pré-codés, réalisant des opérations de filtrage numérique de signaux, très utiles pour les TIPE.

1. **COMPETENCE 1 : enregistrer sur clé USB un signal e(t) numérisé et transformé en fichier .csv par l'oscilloscope ; extraire des données du fichier .csv et les mettre dans des listes ou tableaux PYTHON :**

Enregistrement du signal e(t) sur clé USB :

— Connecter la sortie du générateur BF arbitraire délivrant le signal V_e à une des voies d'entrée de l'oscilloscope Agilent.
— Connecter la clé USB sur la face avant de l'oscilloscope.

"Utility"	"Explor. fichiers"	"Ap. pr aller USB"	⇒ lecture des fichiers de la clé USB
"Save Recall" (face avant oscillo)	"Enregistrer"	"Format PNG" (menu en bas)	choisir le format "Données CSV" en tournant le bouton "Push to select"
"Appuyer pour enregistrer"			

Le message " Fichier enregistré avec succès" apparaît fugitivement.

Enlever la clé de la face avant de l'oscillo pour la brancher sur la face avant de l'ordinateur.
On propose trois méthodes pour transférer et traiter les données du fichier .csv :

i. **Première METHODE : conversion du fichier CSV de données-*string* en deux listes Python de données-*float* :**

— Ouvrir l'explorateur de fichiers, (3^{eme} icône en bas de l'écran de l'ordi. en partant de la gauche), et sélectionner (en cliquant) " disque amovible".

Si on ne reconnaît pas le fichier de données qu'on vient de sauvegarder sur la clé usb, trier les fichiers de cette clé par dates. Usuellement, ce fichier s'appelle "scope_0" et son type est CSV.

— Cliquer sur le fichier "scope_0". L'ordinateur l'ouvre sous forme d'un tableau Excel. Il possède 2000 lignes et chaque ligne comporte une unique cellule remplie. Cette cellule contient deux données : une date t (en seconde) et une tension $V_e(t)$ (en Volt), séparées par une virgule, le tout encadré par des apostrophes.

— Cliquer sur la première icône du menu situé en bas de l'écran de l'ordi. (globe terrestre), puis sélectionner " tous les programmes", puis sélectionnner "Pyzo". Dans la fenêtre "bienvenue dans IEP" choisir "OK".

— Si on s'est loggué au réseau Joffre avec son identifiant et son mdp, on est par exemple dans :

$$C : \backslash USERS \backslash BERGER1.$$

Pour traiter dans Pyzo le fichier de données "scope_0", il faut se placer dans le dossier de la clé USB.

— Cliquer sur l'icône de forme carrée munie d'une flèche courbe vers le haut. Les différents dossiers de l'ordi. apparaissent. C :\ est le disque dur. Chercher celui de la clé qui es usuellement F :\. Cliquer sur l'étoile puis sur la petite flèche de l'icône carrée qui les contient.

— Sélectionner "Go to this directory", puis vérifier dans le *shell* de Pyzo (fenêtre du haut de l'écran), que s'affiche la ligne :

$$cd\ f : \backslash$$

On est passé dans le dossier de la clé USB !

— Avec Pyzo, coder une fonction **Lecture_fichier** qui lise le fichier "scope_0" de type CSV, et le transforme en une liste python nommée **data**, dont chaque élément soit une ligne de texte du fichier CSV de départ.

— Coder une fonction **PostTraitement** qui transforme la liste **data** de lignes de texte, en deux listes Python nommées **temps** et **tension**, dont les éléments soient des données de type "float".

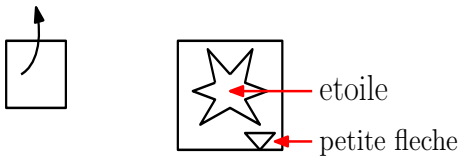

FIGURE 3 – Icônes Pyzo.

```
import matplotlib.pyplot as plt
plt.close('all')

def Lecture_fichier(fichier_texte) :
    fichier=open(fichier_texte,'r')
    data=[]
    for chaque_ligne in fichier :
        data.append(chaque_ligne)
    fichier.close()
return data
```

```
def PostTraitement(fichier_texte) :
    data=Lecture_fichier(fichier_texte)
    temps=[]
    tension=[]
    N=Len(data)
    for i in range(2,N) :
        (t,V)=data[i].split('')
        temps.append(float(t))
        tension.append(float(V))
return temps, tension
time,voltage=PostTraitement('scope_0.csv')
plt.figure()
plt.plot(time,voltage)
plt.show()
```

— Dans le *shell* de Pizo, taper "time" puis " voltage" (ou bien utiliser le raccourci-clavier "Control+E"), pour tester le remplissage de ces deux listes.

— Observer la figure tracée.

Les éléments génériques voltage[n] et time[n] des deux listes de données-float que nous avons créées, vérifient $n \in [0, 1998]$. En effet, les deux premières lignes du fichier scope_0.csv étant occupées par des explications-textes, et non par des données physiques du signal, nous ne les avons pas prises dans les listes time et voltage.

ii. **SECONDE METHODE :**

Voir le code Python conversion-csv.py déposé dans la Dropbox

Psistarall>Physique Chimie>codes_python>TP5_Filtrage_FFT

Ce code utilise une fonction (toute faite) de numpy pour réaliser l'extraction des données du fichier .csv.

Tester ce code sur le fichier .csv que vous avez transféré dans l'ordinateur.

Écrire un code Python simplifié à votre idée, inspiré de conversion-csv.py, que vous conserverez et utiliserez le cas échéant en TIPE.

iii. 3^{eme} **METHODE :**

Voir le code Python TP5-FFT.py déposé dans la Dropbox

Psistarall>Physique Chimie>codes_python>TP5_filtrage_FFT

Ce code manipule des chaînes de caractères (comme vous l'avez appris en ITC).

Exécuter le script Python par étapes, entre les #, pour le comprendre.

Quelques éléments sont à taper dans le shell de Pizo, pour comprendre les opérations effectuées successivement par le script.

iv. **Juxtaposer les graphes du signal e(t) obtenus à l'oscilloscope et sous Python** :

- Tracer le graphe de e(t) sous Python à partir des données numériques du fichier .csv.

- Changer de forme d'onde pour e(t) (de sinus, à triangle, à créneau). Comparer le graphe d'oscillo et le graphe tracé sous Python.

2. COMPETENCE 2 : Effectuer un filtrage numérique de la tension e(t) issue d'une acquisition à l'oscilloscope :

i. **Calculer sous Python la sortie que donne un filtre de fonction de transfert connue, au signal $e(t)$:**

Soit le filtre passe-bas d'ordre 1 de fonction de transfert suivante, où $(H_0, \omega_0) \in (\mathbb{R}_+^*)^2)$:

$$\underline{H}(\mathrm{j}\omega) = \frac{H_0}{1 + \mathrm{j}\frac{\omega}{\omega_0}}.$$

correspondant à l'équation différentielle :

$$V_s + \frac{1}{\omega_0}\frac{\mathrm{d}V_s}{\mathrm{d}t} = H_0\, V_e(t)$$

À partir de la première méthode d'acquisition des données du signal e(t), on va coder une fonction Filtrage, incluant une boucle for, pour y parvenir numériquement.

La tension d'entrée est dans la liste ordonnee[n], variable locale correspondant à la variable globale voltage[n].

La tension de sortie est dans la liste V_filtree[n], variable locale correspondant à la variable globale tension_filtree[n].

La date est dans la liste abscisse[n], variable locale correspondant à la variable globale time[n].

La ligne de calcul de l'algorithme d'Euler explicite est :

V_filtree[i+1] = $H_0 * \omega_0$*ordonnee[i]+(1-ω_0 (abscisse[i+1]-abscisse[i]))V_filtree[i]

La condition de stabilité de l'algorithme est : (1-ω_0 (abscisse[i+1]-abscisse[i])) > 0.

```
def Filtrage(abscisse,ordonnee,H0,W0)
    V_filtree=[ordonnee[0]]
    for i in range(Len(abscisse)-1) :
                            valeur=H0*W0*(abscisse[i+1]-abscisse[i])*ordonnee[i]+(1-W0*(abscisse[i+1]-
abscisse[i])*V_filtree[i]
return V_filtree

tension_filtree=Filtrage(time,voltage,1,10**(-3))
plt.plot(time,voltage)
plt.plot(time,tension_filtree)
plt.show()
```

Commenter le filtrage numérique.

3. COMPETENCE3 : Calculer sous Python la FFT du signal e(t) issu d'une acquisition de tension à l'oscillo :

i. Charger le code Python nommé TP5_FFT.py déposé dans la Dropbox

Psistarall>Physique Chimie>codes_python>TP5_Filtrage_FFT

Ce code lit les données du fichier .csv (voir les commentaires). Le faire tourner en allant chercher le fichier .csv que vous avez enregistré sur votre clé USB.

ii. Écrire un code Python à votre idée, clair, que vous conserverez et utiliserez éventuellement en TIPE.

iii. Comparer le spectre de FFT et le spectre «théorique» de e(t) :

- Le spectre d'amplitude FFT de e(t) est tracé sur l'écran de l'ordinateur.

- Le spectre «théorique» est tracé à la main, ou grâce aux codes Python du dossier decomposition&synthese_Fourier situé dans la Dropbox.

Psistarall>Physique-Chimie>codes_python

iv. Évaluer les pourcentages et les sources de divergence entre spectres «théorique» et de FFT ?

v. Évaluer le taux de distorsion harmonique de e(t).

ANNEXE :

Série de Fourier du signal-créneaux de moyenne nulle, amplitude E, pulsation ω :
$u(t) = \frac{4E}{\pi} \sum_{k=0}^{\infty} \frac{\sin([2k+1]\omega t)}{2k+1}$
Série de Fourier du signal-triangle de moyenne nulle d'amplitude E pulsation ω :
$u(t) = \frac{8E}{\pi^2} \sum_{k=0}^{\infty} \frac{\cos([2k+1]\omega t)}{(2k+1)^2}$
(Radiospare) incertitude relative sur composants : capacité : $\frac{\Delta C}{C} = 10\%$ résistance : voir dernier anneau (or ou argent) du code de couleur.