

# TP Traitement du Signal — Protocole

Philippe Müllhaupt<sup>1,\*</sup>

<sup>1</sup>SGM-ENS, EPFL, CH-1015, Lausanne

\*philippe.muellhaupt@epfl.ch

## DESCRIPTION

Ce document décrit les étapes des expériences numériques à effectuer sur le dispositif d'acquisition et de traitement des signaux. Certaines étapes sont à effectuer dans MATLAB/SysQuake, d'autres directement dans l'environnement C++ ou C du système embarqué. Certaines étapes nécessitent une bonne compréhension de la partie théorique (cf. document sur la partie 'Théorie').

## Acquisition et analyse fréquentielle

1. Monter le capteur (accéléromètre ACH01) et assembler le tout de telle sorte à constituer
  - Un pendule simple
  - Une barre de flexion
2. Connecter les fils du capteur à la partie électronique. Examiner le circuit et déduire le rôle. Déduire le fonctionnement du circuit.
3. L'aspect de l'acquisition se décompose en deux modules. Le premier est le convertisseur analogique/digital (A/D) et son micro-contrôleur (Arduino et STM32 DISCOF7). Le second est l'ordinateur (PC/Mac). Les deux sont connectés par un câble USB.
  - Expliquer ce que signifie le Baud/rate et examiner les nombres et leur quantification.
  - Sachant que le convertisseur A/D opère sur 10 bits, déterminer la fréquence d'acquisition lorsque chaque transmission sur la ligne USB comporte 1 caractère par transmission.
  - Même question lorsque la transmission comporte 2, 3, ..., 10 caractères.
  - Examiner le dessous du STM-32 DISCOF7 et comparer avec l'Arduino.
4. Examiner les "scripts" au niveau
  - (a) L'Arduino: code en .c
  - (b) Le software "Processing" en langage Java.
  - (c) Le fichier texte contenant les données que l'on peut ensuite lire dans MATLAB.
  - (d) Le fichier `main.cpp` du programme embarqué pour le STM32 DISCOF7.

5. Effectuer plusieurs expériences en changeant les conditions initiales du mécanisme. Lancer l'acquisition dans l'Arduino. Puis visualiser le tout dans "Processing".
  - A l'aide d'un chronomètre déterminer le nombre de caractères transmis sur la ligne série USB par acquisition.
6. Utiliser le code "Processing" pour constituer un fichier "texte" qui contiendra les échantillons avec le bon Baud/rate. Ouvrir le fichier ainsi constitué dans MATLAB afin de visualiser le signal acquis. Ceux qui souhaiteraient utiliser SysQuake au lieu de MATLAB écrivent un petit script .m afin d'ouvrir le fichier et de constituer la variable contenant les échantillons.
7. Déterminer le nombre d'échantillons et décrire le signal à l'aide du concept de fenêtre.
  - Supprimer l'artefact du début d'acquisition et de la fin.
8. Déterminer la transformée discrète du signal théorique associée à un mouvement harmonique simple, c.-à-d. la transformée d'une onde sinusoïdale échantillonnée et fenêtrée ayant grosso-modo les mêmes paramètres (amplitude, fréquence, cadence d'échantillonnage, et durée (nombre d'échantillons)).
9. Calculer à l'aide de MATLAB/SysQuake la TFD (Transformée de Fourier Discrète).
  - Expliquer la raison de la grande valeur de la première composante du vecteur correspondant à la TFD.
  - Déterminer par un moyen simple la réduction, voire la suppression de celui-ci.
10. Détecter la fondamentale et trouver la pulsation et la fréquence de celle-ci.
11. Utiliser une fenêtre de type
  - (a) Hann
  - (b) Blackman
  - (c) Hamming
  - (d) etc.

et déterminer la TFD d'un signal sinusoïdal fenêtré. Utiliser uniquement MATLAB/SysQuake en générant des signaux "idéaux". Comparer les résultats avec et sans fenêtrage.

12. Effectuer les mêmes opérations qu'au point 11, mais cette fois-ci avec le signal acquis sur le mécanisme. Expliquer et quantifier la différence entre le cas idéal et le cas réel. (i.e. Porter votre attention sur l'énergie du bruit et sa répartition fréquentielle.)
13. Déterminer une fréquence de coupure d'un filtre sensé supprimer ou du moins atténuer la majeure partie du "bruit" que vous avez remarqué au point 12.

14. Constituer les quatre filtres analogiques suivants:

- (a) 1er ordre,
- (b) 2ème ordre,
- (c) Butterworth,
- (d) Chebychev,

ayant la fréquence de coupure établie au point 13. Dessiner la représentation fréquentielle des filtres a, b, c et d, obtenus en dessinant soigneusement l'amplitude et la phase des réponses fréquentielle analogiques.

- 15. Transformer les filtres analogiques  $G(s)$  du point 14 en des filtres discrets  $H(z)$ . (Indication: utiliser la commande 'c2d' de MATLAB pour les filtres de Butterworth et Chebychev, calculer explicitement la transformée  $H(z)$  pour le premier ordre.)
- 16. Coder les filtres discrets  $H(z)$  obtenus en effectuant un petit script '.m' en MATLAB (ou .lml en SysQuake) et comparer vos résultats avec ceux obtenus avec la commande 'filter'.
- 17. Dessiner les réponses indicielles et superposer ceux-ci pour les quatre filtres a,b,c, et d.
- 18. Déterminer les réponses fréquentielles discrètes du signal filtré par les filtres discrets a,b,c et d.
- 19. Modifier la fréquence de coupure du filtre en vous rapprochant de la fréquence fondamentale. Décrire ce qui se passe.
- 20. Implémenter les filtrages sur le système embarqué STM32 DISCOF7 et dessiner avec une autre couleur le signal filtré sur l'écran du système embarqué.

## Harmoniques supérieures

- 1. Augmenter l'angle du pendule (changement des conditions initiales) et essayer de détecter des harmoniques supérieures.
- 2. Effectuer la même expérience, mais de manière virtuelle sur un pendule simple que vous simuler. Utiliser la commande 'ode45' de MATLAB/SysQuake.
- 3. Augmenter l'angle du pendule virtuel pour favoriser l'apparition des harmoniques supérieures.
- 4. Comparer les mesures avec du bruit et effectuer les filtrages.
- 5. Comment garantir un bon filtrage tout en conservant l'apparition des harmoniques ?

## Repliement spectral

1. Simuler le pendule en continu. Ceci permettra d'obtenir le fichier `penduleSimple.m` dont il est question dans la partie théorique.
2. Echantillonner le signal obtenu avec des fréquences très basses et plus hautes.
3. Effectuer l'analyse fréquentielle.
4. Décrire le repliement fréquentiel (spectral).