

CI 2 : ALGORITHMIQUE ET PROGRAMMATION

CHAPITRE 5 – MANIPULATION DES FICHIERS

TP 6 : ANALYSE ET TRAITEMENT D'UN FICHIER DE MESURE

Objectifs

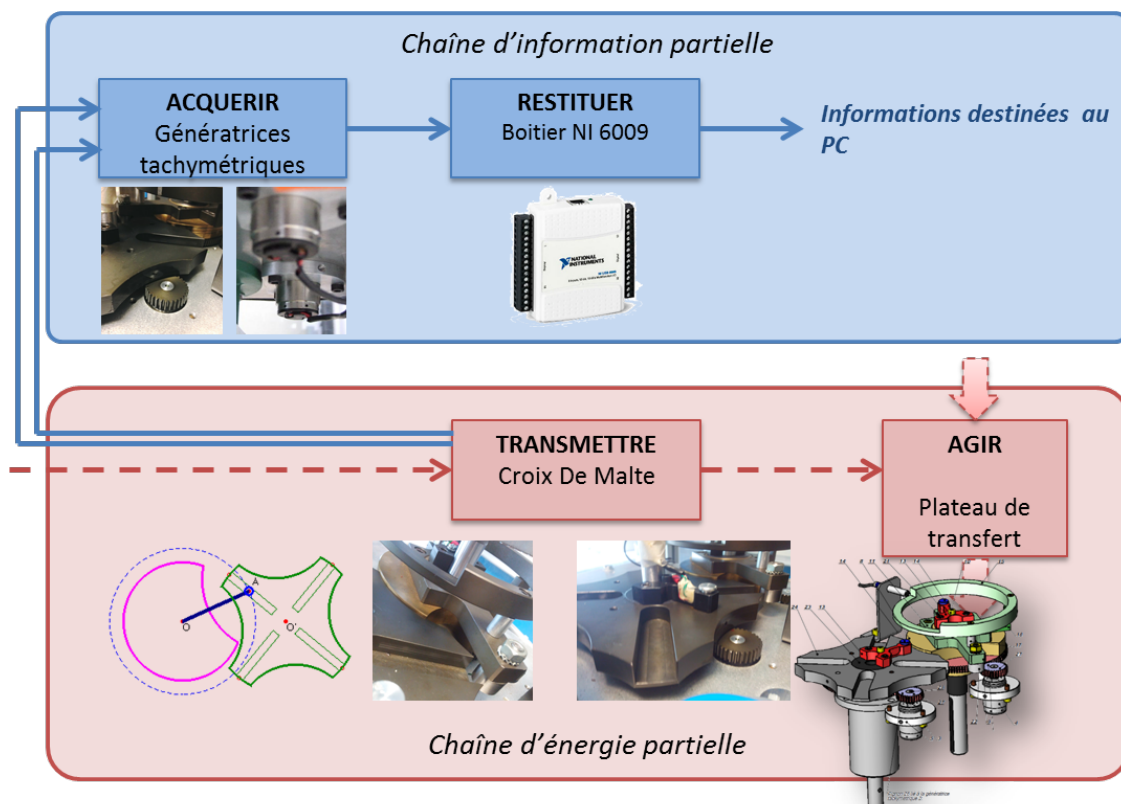
Le but de ce TP est de traiter des données de mesure afin qu'elles soient exploitables par l'utilisateur.

Mise en situation

Objectifs

Analyser un fichier de mesure.

En utilisant la capsuleuse de bocaux du laboratoire de SII, on a réalisé une mesure de vitesse du maneton et de la croix de Malte. On donne ci-dessous un extrait de la chaîne fonctionnelle de la capsuleuse. Un fichier de mesure est enregistrable sur le PC grâce à une application réalisée avec le logiciel Labview.



Ouvrir le fichier *MesureCapsuleuse.lvm* avec un éditeur de texte (Notepad ++ de préférence). La seule opération réalisée sur les données brutes est une conversion du signal électrique (en Volts) en une fréquence de rotation (en tours par minute).

Question 1 Détailler la structure du fichier. Quelle est sa taille (en octets) ? Justifier la taille de ce fichier.

Question 2 A partir de quelle ligne y-a-t il des informations utiles ? Combien y en a-t-il ? Comment ces lignes sont-elles formatées ?

Question 3 Donner (en Hertz) la fréquence d'échantillonnage du signal mesuré.

Affichage des mesures

Objectifs

L'objectif est d'afficher les mesures contenues dans le fichier `MesureCapsuleuse.lvm`.

Question 4 Réaliser une fonction en Python permettant de lire un fichier et de stocker les informations utiles dans une liste de chaîne de caractères. Les spécifications de la fonction sont les suivantes :

```
def lireFichierMesure ( file ):
    """
    Permet de lire un fichier de mesure et de stocker les lignes utiles dans une liste .
    Entrées :
        * file : str , nom du fichier
    Sortie :
        * tab : list , liste des lignes utiles
        (Exemple de ligne utile : '0,009800\t16,152722\t0,426743\n')
    """
```

Question 5 Modifier la fonction `lireFichierMesure()` afin d'obtenir une liste composée de sous-listes de la forme `[..., [0.009800, 16.152722, 0.426743], ...]`.

Question 6 Modifier la fonction `lireFichierMesure()` afin qu'elle retourne trois listes correspondant à chacune des colonnes du fichier de mesure.

Question 7 Afficher les courbes correspondant à la colonne seconde et à la troisième colonne en fonction de la première colonne. Qu'observez-vous ?

Question 8 En déduire quelle est la courbe associée au maneton et quelle est la courbe associée à la croix de Malte. Indiquer le nom des courbes sur la légende.

Filtrage des mesure

Comme constaté précédemment, les mesures effectuées sont fortement bruitées. On va donc chercher à les filtrer.

Objectifs

Filtrer les mesures dans le but de réduire le bruit.

Filtrage par calcul de points moyens

Question 9 Réaliser une fonction permettant de :

- réaliser un filtrage en réalisant une moyenne sur n échantillons ;
- tracer les courbes.

Filtrage par utilisation d'une moyenne glissante

On souhaite maintenant réaliser un filtrage en réalisant une moyenne glissante sur n échantillons.

Pour cela, on dispose d'une liste de valeurs de temps notée t et d'une liste de valeurs mesurées notées m . Pour calculer la moyenne glissante sur n échantillons, on va calculer au temps $t[i]$, la moyenne des mesures de l'échantillon $m[i]$ à l'échantillon $m[i + n]$.

Question 10 Réaliser une fonction permettant de :

- réaliser un filtrage en réalisant une moyenne glissante sur n échantillons ;
- tracer les courbes.

Analyse de la courbe

Question 11 À l'aide de la courbe filtrée, vérifier que la croix de Malte réalise bien un quart de tour par période.

Annexes – Documentation de fonctions Python

Lire les lignes d'un fichier



```
def lireFichierMesure ( file ):
    # Ouverture du fichier en lecture seule
    fid = open(file, 'r')
    # Lecture de chacune des lignes
    for ligne in fid:
        print ( ligne )
    # Fermeture du fichier
    fid . close ()
```

Supprimer ou remplacer un caractère dans une chaîne de caractères



```
chaîne = 'Chuckxstartsxthxznwxyzarxbyxroundhouszxkickingxthxoldxonz.'
chaîne . replace ('x', ' ')
chaîne . replace ('z', 'e')
```

Afficher des courbes



```
import numpy as np
import matplotlib.pyplot as plt

# t correspond à une liste de valeurs d'abscisses
# c1 et c2 correspondent à une liste de valeurs d'ordonnées
# t, c1 et c2 doivent avoir la même taille
plt.plot(t, c1, label="Légende courbe 1")
plt.plot(t, c2, label="Légende courbe 2")
plt.legend(loc='lower center')
plt.xlabel("Abscisse")
plt.ylabel("Ordonnée")
plt.show()
```