

Ouverture et utilisation d'un fichier CSV issu d'un oscilloscope Keysight

Fiche étudiant

Christophe Caire

20 juin 2019

I Détermination des caractéristiques d'un signal périodique

I.1 Cadre de travail

Objectifs :

- Récupérer le signal issu d'un fichier de sauvegarde d'oscilloscope
- Identifier la période du signal mesuré
- En déduire les grandeurs statistiques du signal

Nous commençons pas les classiques imports de bibliothèques.

OS et **tkinter** permettent la gestion fenêtrée des fichiers à analyser, vous pouvez vous en dispenser si vous préférez gérer cela "à la main".

```
In [4]: import numpy as np
import os

# Choisir un fichier à ouvrir grace à tk dialog
import tkinter as tk
import tkinter.filedialog as fd

# Paramètres esthétiques pour les figures
import matplotlib
matplotlib.rcParams['figure.figsize'] = (13,7)
matplotlib.rcParams['savefig.dpi'] = 100
import matplotlib.pyplot as plt

%matplotlib inline
# %matplotlib auto
```

numpy est utilisé comme un complément naturel de python.

La librairie **npuplotpyplot** permet l'affichage simple des graphiques.

La librairie **os** permet la gestion des noms de fichiers quel que soit l'os de déploiement.

Moins commune, la librairie **tkinter** vous permettra d'user d'un mode de dialogue plus interactif que l'usuel "entrez le nom de votre fichier :'" qui est toujours source de complexité chez les utilisateurs qui ne maîtrisent pas la structure de gestion de ces fichiers.

Le dernier appel en matplotlib ne sert qu'à faire joli!

I.2 Fichiers de données

Nous supposons que vos données sont des trames temporelles issues d'un oscilloscope keysight sous la forme de fichiers CSV. Choisissez le fichier de votre choix à importer lors de l'exécution du code ci-dessous.

Les données seront chargées dans le tableau à deux dimensions **mydata**.

```
In [2]: root = tk.Tk()
        root.filename = fd.askopenfilename(title = "Select file",
        filetypes = (("csv files", "*.csv"), ("all files", "*.*")))
        f=root.filename
        root.destroy()

        # Import d'un fichier
        # les deux premières lignes sont ignorées - propre au Keysight
        mydata=np.genfromtxt(f, skip_header=2, delimiter=',', usecols = (0,1))
```

I.3 Affichage de données

L'affichage de données sous format confortable est un peu "délicat".

Ne faites donc pas trop attention au code ci-après c'est de la *gestion graphique*.

```
In [5]: # transfert des données sur tableau
        t,x=mydata[:,0],mydata[:,1]

        # dictionnaire de dimensionnement
        tempdic={i+1:j for i,j in enumerate(['s','ms','μs','ns'])}
        voltdic={i+1:j for i,j in enumerate(['V','mV'])}

        rt=2
        rx=1
        t=10**(3*(rt-1))*t
        x=10**(3*(rx-1))*x

        # Gestion de l'affichage
        fig=plt.figure()
        ax = fig.add_subplot(1, 1, 1)

        ax.plot(t,x, lw=2)
        ax.set_xlabel("t (" + tempdic[rt] + ")")
        ax.set_ylabel(voltdic[rx])

        # Configuration des axes
        ax.axhline(0, lw=1, color='k')
        ax.axvline(0, lw=1, color='k')
        ax.autoscale(axis='x', tight=True)

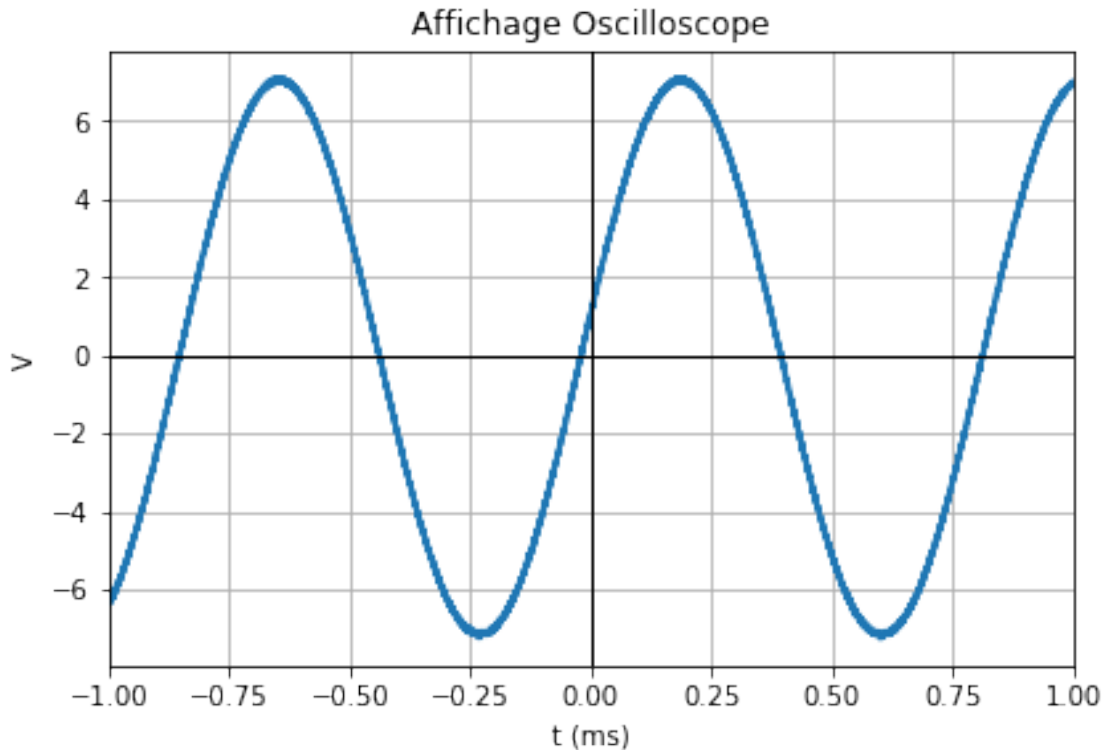
        ax.grid(True)
        fig.tight_layout()
```

```
plt.title("Affichage Oscilloscope")
fig.show()

# sauvegarde image
fig.savefig("oscilloscope00.png")

print("nombre de points : ",x.size)
```

nombre de points : 50000



I.4 Traitement des données

A vous d'imaginer un moyen de mesurer la période de ce signal. Vos enregistrements n'apparaîtront peut être pas périodique en raison des défauts de vos mesures (vibration du portable, éclairage périphérique). Il faudra essayer d'imposer le caractère périodique sur la base d'un critère.

Nous attendons vos suggestions.

La ligne suivante met les données de data dans les tableaux t,x (temps, valeur).

```
In [6]: # transfert des données sur tableau
t,x=mydata[:,0],mydata[:,1]
```