

Correction Exercice 6.a : dérivation position tête imprimante et bras maxpid

1. Correction dérivation position tête Imprimante

```
# dérivée numérique
def derive(x,y):
    dy=[0.0] # initialisation
    for i in range(len(x)-1):
        dy.append((y[i+1]-y[i])/(x[i+1]-x[i]))
    return(dy)

import numpy as np; import matplotlib.pyplot as plt

# importation des mesures
mesures = np.loadtxt('BO200imprimante.csv', delimiter=';', skiprows=1, dtype=str)
mesures = np.char.replace(mesures, ',', '.');
mesures = mesures.astype(float)
temps = mesures[:,0]; position = mesures[:,1]; vitesse = mesures[:,2]

vit_calc=derive(temps,position)

# Affichage
plt.plot(temps, vitesse, label=" vitesse mesurée (mm/s) ")
plt.plot(temps, position, label=" positon mesurée (mm) ")
plt.plot(temps, vit_calc, label=" vitesse calculée (mm/s) ")
plt.xlabel("Temps (s)")
plt.title("Vitesse tête Imprimante") ; plt.legend() ; plt.grid(True) ; plt.show()
```

On se rend compte que la vitesse n'est pas mesurée sur la machine, mais calculée : la courbe calculée se superpose parfaitement à celle issue du fichier de mesure.

Il y a toujours un problème avec le fait que l'on doit fixer la première valeur (vitesse=[0]). Il serait aussi possible de remplacer cette première valeur par une valeur calculée à partir des dérivées indice 1 et 2.

2. Correction position bras MaxPID

```
# dérivée numérique (identique à la question précédente)
def derive(x,y):
# initialisation en calculant dy[0] à partir de la pente entre y[1] et y[2]
    dy=[y[1]-(y[2]-y[1])*(x[1]-x[0])/(x[2]-x[1])]
    dy=[0.0] # initialisation
    for i in range(len(x)-1):
        dy.append((y[i+1]-y[i])/(x[i+1]-x[i]))
    return(dy)

import numpy as np; import matplotlib.pyplot as plt

# importation des mesures
mesures = np.loadtxt('BFtrapeze90_maxpid.csv', delimiter=';', skiprows=2, dtype=str,
encoding='Latin-1')

mesures = np.char.replace(mesures, ',', '.');
mesures = mesures.astype(float)
temps = 0.001*mesures[:,1];
pos_bras = np.pi/180*mesures[:,3];
vit_bras=mesures[:,4];
Vit_moteur = mesures[:,5]

vit_calc=derive(temps,position)

# Affichage
plt.plot(temps, pos_bras, label=" position bras mesurée (rd) ")
plt.plot(temps, vit_bras, label=" vitesse bras mesurée (rd/s) ")
plt.plot(temps, vit_calc, label=" Vitesse calculée ")
plt.xlabel("Temps (s)")
plt.title("Vitesse calculée du bras MaxPID ") ; plt.legend() ; plt.grid(True) ;
plt.show()
```

La vitesse obtenue sur le MaxPID n'est pas mesurée mais elle aussi calculée sur la machine. En revanche, elle est probablement filtrée ou moyennée car la courbe de vitesse calculée est assez clairement "bruitée".

Pour information Le module Scipy propose aussi des fonctions de calcul intégral, entre autre la méthode des trapèzes.

Elle s'appelle de la façon suivante :

```
# appel de méthode trapèze de scipy
pos_cal_trasc=integrate.cumtrapz(vitesse,temps)
```