TP 5

Identification d'un SLCI du second ordre peu amorti

Instructions préalables : Copier l'ensemble du dossier ***B410CPGE/PTSI2/TP5*** sur votre compte. Il contient 3 fichiers : le sujet de TP, le fichier python ***TP5_python.py*** à compléter et un fichier issu d'un essai expérimental ***MaxPID_essai.csv*** .

Objectifs:

1) Exploiter les notions vues en informatique pour trouver les caractéristiques d'un Système Linéaire Continu Invariant (SLCI), en exploitant sa réponse indicielle (*i.e.* réponse à une entrée de type échelon).
2) Apprendre à tracer des courbes en python.

1 *Questions préliminaires*

1) a) Ecrire la fonction `moyenne_fin(L,n)`, qui prend en argument une liste `L` de nombres et un entier `n`, et qui renvoie la moyenne des n derniers éléments de `L`.

```
# voici un exemple d 'appel :
>>> moyenne_fin([2,1,6,8,3,1],3)
4.0
```

b) Ecrire la fonction `max_liste_i(L)`, qui prend en argument une liste `L` de nombres, et qui renvoie, le maximum de `L`, et l'indice correspondant dans un n-uplet. On suppose que ce maximum n'apparaît qu'une seul fois dans la liste.

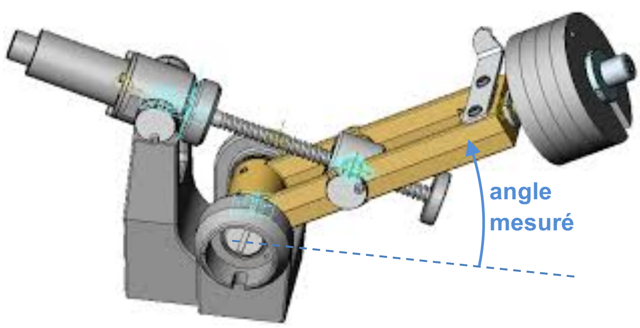
```
# voici un exemple d 'appel :
>>> max_liste_i([2,1,6,8,3,1])
(8,3)
```

c) Compléter la fonction `diff2_f(Lx,Ly,f)`, qui prend en argument deux listes de nombres, `Lx` listant des abscisses x_i, et `Ly` les ordonnées y_i correspondantes, la fonction f qui ne prend qu'un argument, et qui renvoie la moyenne de la différence au carré entre $f(x_i)$ et y_i soit :

$$\frac{1}{n}\sum_0^{n-1}(f(x_i)-y_i)^2 \quad \text{avec } n \text{ est la taille de Lx}$$

```
# voici un exemple d 'appel :
>>> diff2_f(Lxtest,Lytest,ftest)
0.265
```

Cette grandeur nous servira d'indicateur d'erreur, entre la fonction f et la liste des points représentés par les coordonnées x_i et y_i, contenus dans `Lx` et `Ly`.



Structure du bras *MaxPID.*

Nous allons maintenant exploiter des données issues d'un essai effectué sur l'un des systèmes du laboratoire de SI : le bras *MaxPID*. Le résultat de l'essai est stocké dans le fichier ***MaxPID_essai.csv*** du dossier de TP. Il correspond à l'évolution de l'angle mesuré en fonction du temps, pour une consigne de type échelon, d'amplitude 10°. Ce fichier est constitué de deux colonnes, l'une représentant les instants des mesures, et l'autre la mesure correspondante de position angulaire, en degrés. L'objectif de ce deuxième exercice est d'extraire les données de ce fichier puis de tracer le résultat de cet essai.

2 *Récupération et tracé des données expérimentales*

2) a) Décommenter, analyser puis compléter (les lignes `#à compléter`) les instructions permettant de lire les informations du fichier ***MaxPID_essai.csv.*** Ces instructions doivent permettre de stocker les instants de mesure dans la liste `Ltemps`, et les valeurs des angles correspondants dans la liste `Langle`.

Pour tracer une courbe, nous allons utiliser la bibliothèque `matplotlib.pyplot` renommée `plt` et déjà importée grâce à la commande (voir premières lignes du fichier python) :

```
import matplotlib.pyplot as plt
```

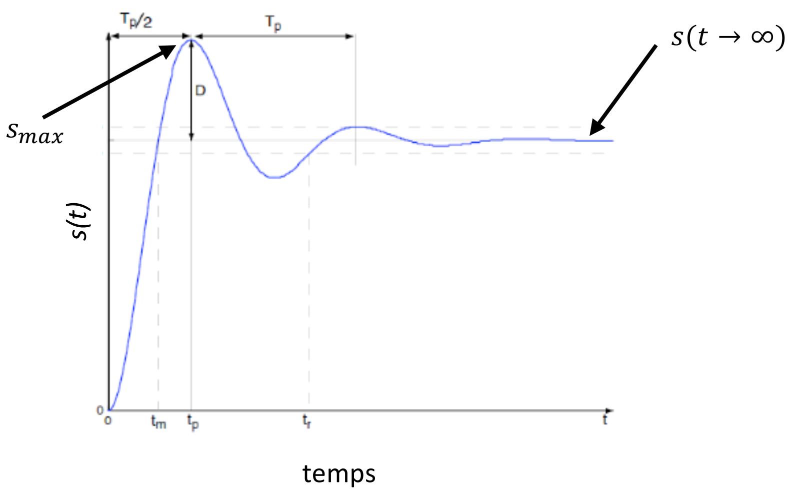
b) Décommenter les lignes proposées et exécuter. Vous devriez obtenir la courbe représentant l'évolution de l'angle du bras *MaxPID* en fonction du temps. Expliquer le rôle de chaque ligne de cette partie.

La réponse tracée montre clairement la réponse d'un système assimilable à un second ordre (tangente nulle à l'origine, àl'échantillonage près) peu amortie (présence d'un dépassement). On rappelle la fonction de transfert correspondante dans le domaine de Laplace :

$$H(p) = \frac{K}{1 + \frac{2z}{\omega_0}p + \frac{p^2}{\omega_0^2}}$$

Identifier ce système revient donc à déterminer son gain statique K, son amortissement z et sa pulsation propre non amortie ω_0.

On rappelle le résultat de cours de SI :



Réponse indicielle d'un système du second ordre peu amorti

$s(t)$ représente la sortie, D le premier dépassement et T_p la pseudo-période. On sait que :

- $K = \frac{s(t\to\infty)}{E_0}$
- $D = KE_0 e^{\frac{-\pi z}{\sqrt{1-z^2}}}$ d'où $z = \sqrt{-\frac{ln\frac{D}{KE_0}}{ln^2\frac{D}{KE_0}+\pi^2}}$
- $\omega_0 = \frac{2\pi}{T_p\sqrt{1-z^2}}$

avec E_0 l'amplitude de l'entrée. Dans le cas du résultat d'essai donné, $E_0 = 10°$.
Les formules en rouge sont celles qu'il faudra exploiter dans la partie suivante.

3 Identification du système

3) a) En utilisant la fonction `moyenne_fin(L,n)`, estimer la valeur finale de la réponse du bras *MaxPID*, en faisant une moyenne sur les 10 dernières valeurs. En effet, compte tenu du bruit de mesure, il est en général plus rigoureux de prendre une moyenne des dernières valeurs. En déduire la valeur de *K.*

b) En utilisant la fonction `max_liste_i(L)`, estimer la valeur de D. En déduire la valeur de z.

c) En utilisant la fonction `max_liste_i(L)`, estimer la valeur de T_p. En déduire la valeur de ω_0.

4 Comparaison essai/modèle

On se propose dans cette dernière partie de comparer le résultat expérimental avec le modèle identifié. On donne la fonction `ftheo(t)` qui prend en argument l'instant *t,* et qui renvoie la valeur théorique de réponse du système identifié précédemment.

4) a) En utilisant la fonction `diff2_f(Lx,Ly,f)` estimer l'erreur faite entre modèle et essai. Modifier légèrement la valeur de `K` dans `ftheo(t)` et vérifier que l'erreur augmente.

b) Décommenter puis compléter les dernières lignes de code qui permettent de :

- construire la liste des valeurs théoriques du modèle identifié pour les mêmes instants que l'essai :
  `Langletheo=[ftheo(`t_0`),ftheo(`t_1`),ftheo(`t_2`),…]`
- Tracer sur le même graphe les courbes théoriques et expérimentales (ne pas modifier ce qui est donné).

5 Travail supplémentaire

Proposer une démarche similaire à la partie 3, mais pour identifier un système du premier ordre. Votre enseignant peut vous fournir un résultat d'essai pour tester votre proposition.