Devoir surveillé d'informatique 3

CI 2 : Algorithmique et programmation

Tracé de l'abaque du temps de réponse réduit

Nom : ....

Objectifs

L'objectif de ces travaux est de construire le programme permettant de tracer l'abaque du temps de réponse réduit utilisé en asservissement pour connaître le temps de réponse à 5% des systèmes d'ordre 2.

Mise en situation

L'équation différentielle d'un système du second ordre peut se mettre sous la forme :

$$s(t)+\frac{2\xi}{\omega_0}\cdot\frac{\mathrm{d}s(t)}{\mathrm{d}t}+\frac{1}{\omega_0^2}\cdot\frac{\mathrm{d}^2 s(t)}{\mathrm{d}t^2}=K\cdot e(t)$$

en notant :
- K : le gain statique ;
- ξ : le coefficient d'amortissement ;
- $e(t)$ et $s(t)$: l'entrée et la sortie du système.

On suppose que toutes les conditions initiales sont nulles. Pour une entrée unitaire de type échelon unitaire $e(t)=u(t)$, $K=1$ et $t\geq 0$ on montre que :
- si $\xi<1$, le régime est pseudo périodique et :

$$s(t)=1-\frac{e^{-\xi\omega_0 t}}{\sqrt{1-\xi^2}}\sin\left(\omega_0 t\sqrt{1-\xi^2}+\arcsin\sqrt{1-\xi^2}\right)$$

- si $\xi=1$, le régime est critique et :

$$s(t)=1-(1+\omega_0 t)e^{-\omega_0 t}$$

- si $\xi>1$, le régime est apériodique et :

$$s(t)=1+\frac{e^{-\omega_0 t\left(\xi+\sqrt{\xi^2-1}\right)}}{2\left(\xi\sqrt{\xi^2-1}+\xi^2-1\right)}-\frac{e^{-\omega_0 t\left(\xi-\sqrt{\xi^2-1}\right)}}{2\left(\xi\sqrt{\xi^2-1}-\xi^2+1\right)}$$

Dans l'ensemble de ce sujet, on considèrera que s est une fonction du temps réduit $t\cdot\omega_0$.

Tracé de la réponse indicielle

On dispose des fonctions Python **f_pseudo** et **f_aperiodique** permettant d'évaluer la fonction pour s pour un couple $(t\omega_0,\xi)$.

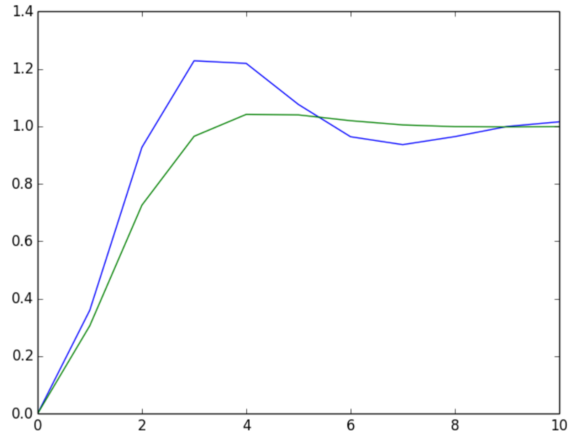
Question 1 *Donner, en Python, le contenu de la fonction f_critique permettant de définir la fonction $(t\omega_0)\rightarrow s(t\omega_0)$ dans le cas où $\xi=1$.*

Question 2 *Donner, en Python, le contenu de la fonction f_s permettant de définir la fonction $(t\omega_0,\xi)\rightarrow s(t\omega_0\xi)$ dans le cas où $\xi\in\mathbb{R}_+^*$. On donne ci-dessous les spécifications de la fonction.*

```python
def f_s(tom0,z):
    """
    Fonction permettant de calculer la réponse indicielle d'un système du second ordre.
    Entrées :
        * tom0, flt : temps de réponse réduit
        * z, flt : coefficient d'amortissement
    Sortie :
        * s(tom0,z)
    """
```

La fonction **trace_s** donnée ci-dessous permet de tracer $s(t\,\omega_0, \xi)$ pour $t\,\omega_0 \in [0,10]$ par pas de 1 et pour une valeur de ξ déterminée. Les deux appels successifs de la fonction **trace_s** permettent de réaliser le tracer les 2 courbes ci-dessous.

```python
# Définition de la fonction trace
def trace_s(z):
    x = []
    y = []
    for i in range(11):
        t = i
        x.append(t)
        y.append(f_s(t,z))
    plot(x,y)
# Appels de la fonction trace
trace_s(0.4)
trace_s(0.7)
```



Question 3 *Expliquer l'objectif des lignes 2 à 9.*

On observe que la courbe tracée n'est pas lissée. Pour avoir un meilleur rendu, il est nécessaire d'évaluer la fonction en davantage de points.

Question 4 *Modifier les lignes 5 et 6 pour que la courbe tracée soit réalisée en 1000 points sur un intervalle de $t\,\omega_0$ variant de 0 à 10.*

Tracé de l'abaque

On note t_r le temps de réponse à 5%. L'abaque du temps de réponse permet de tracer le produit $t_r\omega_0$ en fonction du coefficient d'amortissement ξ.

Question 5 *Dans les conditions de la fonction s définie dans la partie précédente, quelle est la valeur finale prise par $s(t)$?*

Question 6 *Écrire en Python la fonction **is_in_strip** ayant les spécifications suivantes :*

```python
def is_in_strip(x):
    """
    Fonction permettant de savoir si une valeur est dans la bande des + ou - 5% de la valeur finale.
    Entrée :
        x, flt : réel
    Sortie :
        True si la valeur est dans la bande à + ou - 5%
        False si la valeur n'est pas dans la bande à + ou - 5%
    """
```

On donne la fonction suivante permettant de connaître le temps de réponse réduit à partir duquel la réponse indicielle d'un système est dans la bande à plus ou moins 5%.

```python
def f(z):
    tom0 = 500
    pas_tom0 = 0.05
    x = f_s(tom0,z)
    while is_in_strip(x) :
        x = f_s(tom0,z)
        tom0 = tom0 - pas_tom0
    return tom0
```