

Python pour l'automatique

Démarrage

Lancer le poste et démarrer Spyder et importer les bibliothèques nécessaire à ce TP.

```
import matplotlib.pyplot as plt
import numpy as np
from scipy import signal
```

Si spyder n'est pas présent lancez Basthon sur votre navigateur ou au lien suivant :
<https://console.basthon.fr/>

Création d'une fonction de transfert

Nous allons d'abord étudier un système du second ordre :

$$H(s) = \frac{10}{p^2 + 2p + 10}$$

Ecriture de la fonction de transfert :

```
num = 10
den = [1 2 10]
ft = signal.lti(num,den)
```

Q1. Que représentent num et den ?

Etude de la réponse à un échelon :

Q2. Construire une liste t allant de 0 à 10 avec 100 échantillons équiréparties. Vous pourrez vous aider de la fonction np.linspace() .

```
T, S = signal.step(ft,T=t)
```

T stocke alors les instants des échantillons et S la valeur prise par le signal de sortie en réponse à un échelon unitaire à ces instants. Nous pouvons alors tracé le résultat :

```
plt.plot(T,S)
plt.show()
```

Q3. Vérifier que le tracé est cohérent avec la fonction de transfert (stabilité, précision, allure)

Nous pouvons également ajouter des commentaires sur la courbe pour améliorer sa lisibilité :

```
plt.title('Réponse à un échelon')
plt.xlabel('temps (s)')
plt.ylabel('Sortie')
```

Penser à ajouter ces éléments avant de faire appel à `plt.show()` sans quoi ils ne seront pas visibles.

Q4. Ajouter sur le tracé un aperçu du tube à 5% en rouge avec une ligne en pointillé.

Q5. Vérifier la valeur du temps de réponse à 5%, du premier dépassement et la pseudo période.

Etude de la réponse à une rampe

Pour une réponse à une rampe ou à une entrée quelconque il faut utiliser la fonction `signal.lti()`.

Il faut d'abord définir la rampe, ici de pente 3 :

$$R = 3 \cdot t$$

Et ensuite calculer la réponse au système sur une échelle de temps t :

$$T, S, _ = \text{signal.lsim}(ft, R, t)$$

Q6. Tracer la réponse du système à une rampe en ajoutant un titre et les noms des axes.

Q7. Tracer également le signal d'entrée U et mesurer l'erreur en poursuite. Est-elle conforme à la valeur théorique que vous savez calculer ?

Etude de la réponse à une entrée sinusoïdale

Sur la base de la démarche précédente :

Q8. Tracer la réponse du système à une entrée sinusoïdale d'amplitude 1 et de pulsation 2rad/s

Q9. Mesurer le gain et la phase de la réponse au système

Q10. Réitérer l'expérience pour les pulsations suivantes : 1 rad/s, 5 rad/s, 10rad/s, 20rad/s.

Etude fréquentielle

Pour étudier plus en détail l'impact de la pulsation sur le comportement de la fonction de transfert nous allons tracer et observer le diagramme de Bode de celui-ci.

Nous avons tout d'abord besoin de la liste des valeurs de w . Cette dernière n'évoluant pas linéairement (tracé semilog) nous utiliserons la fonction `np.logspace()`

```
| omega = np.logspace(-1,1,100)
```

⚠ Les arguments ici ne sont pas les valeurs de début et de fin de w , mais les puissances de 10 de début et de fin ($\omega[0]=10^{-1}$ et $\omega[-1]=10^1$) ⚠

Pour le calcul du gain et de la phase on utilise ensuite la fonction `signal.bode()`

```
| w, gain, phase = signal.bode(ft,w=omega)
```

Nous utiliserons par la suite 2 subplot (deux figures tracé l'une à côté de l'autre). Pour cela nous utiliserons les deux commandes suivantes :

```
| plt.subplot(211)
```

```
| # insérer ici le traitement à réaliser sur le premier tracé (le gain)
```

```
| plt.subplot(212)
```

```
| # insérer ici le traitement à réaliser sur le premier tracé (la phase)
```

Q11. Réaliser le tracer gain et phase et ajouter les notations nécessaire en utilisant la fonction `plt.semilogx()` pour réaliser le tracé. La fonction fonctionne de la même façon que `plt.plot` mais permet de changer l'échelle en x pour une échelle logarithmique. Vous pourrez ajouter un argument `lw=2` pour tracer le quadrillage derrière.

Q12. Comparer le tracer fait aux valeurs trouvées à la partie précédente

Activité

En reprenant les fonctions de transfert obtenu lors des TP précédents tracer la réponse de votre système à un échelon puis son diagramme de Bode.

Observer l'impact du correcteur sur le tracé.