

# OptoElec & ONIP-1 / TD Systèmes et Signaux

---

## SÉANCE 1 / SYSTÈMES ASSERVIS

---

### Exercice 1 / Modélisation numérique d'un système / ALI

Soit un système  $A(j\omega)$  de type passe-bas, d'ordre 1, de gain statique  $A_0$  et de pulsation de coupure  $\omega_0$ .

1. Donner la fonction de transfert de ce système.
2. A l'aide de la bibliothèque **control** sous Python, définir la fonction de transfert de ce système à l'aide de la fonction *tf*.
3. Tracer la **réponse en fréquence** de ce système à l'aide de la fonction *bode\_plot*.
4. Tracer la **réponse indicielle** à l'aide de la fonction *step\_response*, puis **impulsionnelle** à l'aide de la fonction *impulse\_response*.
5. Calculer et afficher la **transformée de Fourier** de la réponse impulsionnelle. Que pouvez-vous en conclure ?

Applications Numériques :  $A_0 = 10^5$ ,  $f_0 = 30$  Hz.

---

### Exercice 2 / Rebouclage d'un système - Système asservi / ALI en régime "linéaire"

Il est possible de reboucler un système à l'aide d'un autre système. On parle alors d'un système asservi.

On prendra ici le système  $A$  pour la boucle d'action et un système  $B(j\omega) = 1/K$ , où  $K$  est une constante, comme système de contre-réaction.

1. Tracer le **schéma bloc** de ce système puis donner la **fonction de transfert** de ce système.
2. Que vaut le **produit** du gain statique par la fréquence de coupure de ce système ? Comparer cette valeur à celui du système  $A$ . Que pouvez-vous en conclure ?
3. Définir la **fonction de transfert** du système  $B$  à l'aide de la fonction *tf*.
4. Définir un système  $C$  qui est le système complet avec la rétro-action, à l'aide de la fonction *feedback*.
5. Tracer la **réponse en fréquence** des systèmes  $A$  et  $C$  à l'aide de la fonction *bode\_plot* sur un même graphique.
6. Tracer la **réponse indicielle** des systèmes  $A$  et  $C$  à l'aide de la fonction *step\_response* sur un même graphique.

Applications Numériques :  $A_0 = 10^5$ ,  $f_0 = 30$  Hz et  $K = 1$  puis  $K = 10$ .

---

### Exercice 3 / Système du second ordre

On se propose de simuler un système du second ordre dont la fonction de transfert peut être mise sous la forme suivante :

$$H(j\omega) = \frac{A_0 \cdot \left(\frac{j\omega}{\omega_0}\right)^2}{1 + 2 \cdot m \cdot \frac{j\omega}{\omega_0} + \left(\frac{j\omega}{\omega_0}\right)^2}$$

1. Définir ce système.
2. Tracer la réponse en fréquence de ce système pour  $m = [0.1, 0.5, 0.7, 1.0, 2]$  sur un même graphique.
3. Tracer la réponse indicielle de ce système pour  $m = [0.1, 0.5, 0.7, 1.0, 2]$  sur un même graphique.

Applications Numériques :  $A_0 = 10$ ,  $f_0 = 1000$  Hz.

---

## BIBLIOTHÈQUE CONTROL

---

Plus d'aide sur la bibliothèque **control** : <https://python-control.readthedocs.io/en/0.10.1/>

Pour importer la bibliothèque **control** :

```
1 import control as ct
```

---

### Définir un système

La **définition d'un système** par l'intermédiaire d'une fonction de transfert se fait à l'aide de la fonction `tf` :

```
1 # System with a transfert function : (3s + 4) / (6s^2 + 5s + 4).
2 num = [3, 4]
3 den = [6, 5, 4]
4 sys1 = ct.tf(num, den)
```

---

### Tracer la réponse en fréquence d'un système

Il est possible de tracer la réponse en fréquence d'un système à l'aide de la fonction `bode_plot` :

```
1 ct.bode_plot(sys1)
2 plt.show()
```

*Attention : cette fonction se base sur la bibliothèque Pyplot de Matplotlib. Il est indispensable de l'importer et de faire appel à la fonction `show()` pour visualiser les graphiques.*

Il est également possible de passer une liste de système en argument de la fonction `bode_plot` afin de comparer plusieurs systèmes entre eux.

---

### Tracer la réponse indicielle d'un système

Il est possible de tracer la réponse à un échelon d'un système à l'aide de la fonction `step_response` :

```
1 time = np.arange(0, 0.1, 0.0001)
2 T, yout = ct.step_response(sys1, time)
```

Cette fonction renvoie deux vecteurs : un vecteur temps ( $T$ ) et le signal de sortie de la réponse à l'échelon ( $yout$ ). Le vecteur `time` n'est pas indispensable. S'il n'est pas fourni, il est automatiquement calculé par la fonction `step_response`.

Pour pouvoir afficher le graphique associé, il est indispensable d'utiliser une bibliothèque graphique de type Pyplot de Matplotlib.

---

### Tracer la réponse impulsionnelle d'un système

Il est possible de tracer la réponse à une impulsion d'un système à l'aide de la fonction `impulse_response` :

```
1 time = np.arange(0, 0.1, 0.0001)
2 T, yout = ct.impulse_response(sys1, time)
```

Cette fonction renvoie le même type de données que `step_response`

---

### Reboucler un système

Il est possible de reboucler un système par un autre système à l'aide de la fonction `feedback` :

```
1 sys = ct.feedback(sys1, sys2)
```