

|                     |  |                               |
|---------------------|--|-------------------------------|
| <b>Code<br/>NAO</b> | <b>DC23Modéliser la chaine de transmission de puissance d'un système</b> | <b>Série 3<br/>Activité 1</b> |
|---------------------|--|-------------------------------|

|                      |   |
|----------------------|---|
| <b>Problématique</b> | <b>Comment modéliser un système pluri-technologique ?</b> |
|----------------------|---|

|                |   |
|----------------|---|
| <b>Système</b> | <p><b>Robot Humanoïde NAO</b></p> <p>NAO est un robot humanoïde de 58cm conçu par Aldebaran Robotics, une entreprise parisienne à la pointe de la robotique mobile. Déjà produit à près d'un millier d'exemplaires, NAO fait figure de référence dans le monde de la robotique mobile. Il est notamment utilisé pour la coupe du monde de robotique. NAO est au coeur de nombreuses recherches préfigurant les applications de la robotique mobile: jeux multimédias, aide à l'apprentissage, assistance aux personnes handicapées, interventions en milieu extrême, surveillance de lieux...</p> |
|----------------|---|



|                    |   |
|--------------------|---|
| <b>Compétences</b> | <ul style="list-style-type: none"> <li>Proposer un modèle de connaissance d'un système pluri-technologique</li> <li>Proposer un modèle de comportement d'un système pluri-technologique</li> <li>Analyser les performances d'un SLCI</li> <li>Utiliser une simulation numérique pour prévoir les performances d'un SLCI</li> <li>Proposer, justifier et mettre en œuvre un protocole expérimental</li> <li>Exploiter et interpréter les résultats d'un calcul ou d'une simulation</li> <li>Quantifier des écarts entre des valeurs mesurées et des valeurs obtenues par simulation</li> </ul> |
|--------------------|---|

|                   |  |
|-------------------|--|
| <b>Activité 1</b> | <b>Analyser les performances de l'asservissement</b> |
|-------------------|--|

|                   |   |
|-------------------|---|
| <b>Activité 2</b> | <b>Modéliser la chaine de transmission de puissance</b> |
|-------------------|---|

|                   |   |
|-------------------|---|
| <b>Activité 3</b> | <b>Réaliser la simulation numérique de l'asservissement</b> |
|-------------------|---|

*Chef de projet*

# Activité 1

Objectif : Analyser les performances de l'asservissement

|           |   |  |
|-----------|---|--|
| Documents | Document constructeur<br><i>doc réponse</i> | FLTSI.fr rubrique Systèmes<br>NAO_A3_DR1 |
|-----------|---|--|

**AVERTISSEMENT IMPORTANT** : Ne jamais agir manuellement sur la cheville alors que le système de positionnement est activé, ceci entraîne une dégradation mécanique de l'entraînement extrêmement fragile

Identifier l'axe de tangage directement sur la cheville, puis mettre en service le système.

## Commande en boucle ouverte

- Q1 Rappeler le principe d'un essai boucle ouverte, et préciser à quel endroit du système on ouvre la boucle.  
Q2 Accéder au réglage du correcteur PID de l'angle de tangage et régler la valeur du correcteur PID tel que  $K_p = 100$ ,  $K_i$  et  $K_d = 0$  et lancer un essai avec un échelon de  $10^\circ$  pour le moteur de tangage en boucle ouverte (BO). Conclure sur les phénomènes observés.

## Commande en boucle fermée

Mettre en œuvre le moteur de tangage pour un réglage en asservissement réducteur (on contrôle l'angle de l'axe du réducteur) avec  $K=300$ .

- Q3 Faire un essai avec un échelon de  $10^\circ$  pour le moteur de tangage et imprimer ce relevé. Enregistrer les tracés de la réponse en position angulaire du tangage et du courant moteur.  
Exploiter sur le relevé les valeurs importantes :
- valeur finale et valeur éventuelle du premier dépassement  $D1(\%)$ ,
  - Erreur absolue et relative,
  - temps de réponse à 5%,
  - pseudo période  $\omega_r$  éventuelle des oscillations,
  - symétrie ou non du fonctionnement pour les 2 sens de rotation du bras.
- Q4 Renouveler cet essai pour  $K=1000$ . Enregistrer les tracés de la réponse en position angulaire du tangage et du courant moteur.  
Q5 Comparer les résultats avec l'essai précédent  
Q6 Conclure sur l'influence du réglage du correcteur K.

Afin d'affiner les mesures des valeurs importantes précédentes, il est nécessaire de passer par un traitement numérique des données. En enregistrant un essai indiciel dans un fichier texte, il est possible de tracer l'évolution de la position et de calculer la vitesse à l'aide d'un programme Python.

- Q1. A partir d'un essai indiciel, enregistrer les valeurs dans un fichier.  
Q2. Ouvrir le fichier texte à l'aide de Notepad++ ('ouvrir avec' à l'aide d'un clic droit).  
Q3. Effacer toutes les lignes qui ne sont pas des valeurs au début et à la fin du fichier.  
Q4. Remplacer toutes les virgules , par des points . si nécessaire, sauvegarder le fichier sous le nom de `mesures.txt`.

Maintenant que le fichier de mesures est propre, il faut mettre en place le traitement du fichier par python. Pour cela, mettre le fichier `mesures.txt` nettoyé et le code fourni `nao.py` **dans le même dossier**.

- Q5. Ouvrir le fichier `nao.py` à l'aide de **Pyzo ou Spyder**.  
Q6. Indiquer dans la fonction `os.chdir()`, le nom du chemin du dossier contenant le fichier de mesures.  
Q7. Indiquer dans la fonction `open()`, le nom de votre fichier de mesures.  
Q8. Tester le programme.  
Q9. Sur le document-réponse 1, commenter chaque ligne des deux fonctions `valeurs` et `vitesse` en expliquant ce que le code effectue.  
Q10. Compiler le programme et observer les courbes obtenues. Imprimer les pour le joindre au compte-rendu.  
Q11. Modifier la fonction `valeurs` pour afficher le temps en ms plutôt qu'en secondes.  
Q12. Imprimer les nouvelles courbes et les joindre au compte-rendu.

```
def valeurs(n):
    T=[]
    P=[]
    for i in range(n):
        if i<10:
            donnees.readline(6)
            T.append(float(donnees.readline(8)))
            donnees.readline(5)
            P.append(float(donnees.readline(8)))
            donnees.readline()
        elif i>9 and i<100:
            donnees.readline(7)
            T.append(float(donnees.readline(8)))
            donnees.readline(4)
            P.append(float(donnees.readline(9)))
            donnees.readline()
        else:
            donnees.readline(8)
            T.append(float(donnees.readline(8)))
            donnees.readline(4)
            P.append(float(donnees.readline(9)))
            donnees.readline()
    return [T,P]
```

```
def vitesse(listet,listep):
    V=[0]
    for i in range(len(listet)-1):
        vi=(listep[i+1]-listep[i])/((listet[i+1]-listet[i]))
        V.append(vi)
    return V
```