

Code CONTROL'X	DC23 Modéliser la chaîne de transmission de puissance d'un système	Série 3 Activité 1
--------------------------	---	-------------------------------------

Problématique	Comment modéliser un système pluri-technologique ?
----------------------	---

Système	<p>CONTROL'X</p> <p>Le système industriel duquel est extrait Control'X est un robot portique 3 axes Lexium Max R du constructeur Schneider Electric.</p> <p>Ce robot portique permet d'apporter une solution fiable pour la manipulation de charges sur de longues distances : selon le modèle, des charges jusqu'à 50 kg peuvent être déplacées jusqu'à 5500 mm en X, 1500 mm en Y et 1200 mm en Z..</p> <p>Ces robots portiques, commercialisés préassemblés, offrent différentes options de configuration pour chaque axe dont la longueur, le choix entre différentes tailles et types de profilés, le choix entre différents types de guidages ...</p>
----------------	--



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

Activité 1	Analyser les performances de l'asservissement
-------------------	--

Activité 2	Modéliser la chaîne de transmission de puissance
-------------------	---

Activité 3	Réaliser la simulation numérique de l'asservissement
-------------------	---

Chef de projet

Activité 1

Objectif : Analyser les performances de l'asservissement

Documents	Document constructeur <i>doc réponse</i>	FLTSI.fr rubrique Systèmes CONTROLX_A3_DR1
-----------	---	--

Initialisation : Effectuer une réinitialisation en utilisant la fonction "Tout réinitialiser" du menu Initialisation.

Contexte

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 Commander à partir du schéma structurel un déplacement pendant 1s avec une tension de commande de 0,5V puis 1V puis 2V puis 4V.
- Q3 Commenter les déplacements obtenus en fonction de la tension de commande en expliquant les phénomènes observés.

Commande en boucle fermée

Régler le correcteur avec $K=0,07$ $T_i=0$ $T_d=0$ dans le menu Correcteur.

- Q4 Lancer le mode Analyse Temporelle et effectuer un essai de consigne de déplacement de 200 mm pendant environ 1s.
- Q5 Imprimer la courbe obtenue, et en déduire le type de la réponse ainsi que les caractéristiques associées.
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 ω_r éventuelle des oscillations,
- Q6 Renouveler l'essai pour un déplacement dans l'autre sens de 100 mm, déterminer si la réponse est identique et expliquer ce résultat.
- Q7 Modifier la valeur du correcteur proportionnel ($K=2$) et refaire un essai de déplacement de 200 mm à partir de l'initialisation. Imprimer la courbe et déterminer les nouvelles caractéristiques de la réponse.
- Q8 Renouveler l'essai si $K=4$. Que constatez-vous ? Conclure sur l'influence du correcteur proportionnel.

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 `controlx.py` **dans le même dossier.**

- Q5. Ouvrir le fichier `controlx.py` à l'aide de **Pyzo**.
- 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.

CONTROLX_A1_DR1

```
def tempsposition(A):  
    T=[]  
    F=[]  
    for k in range(len(A)):  
        n=[]  
        for i in range(len(A[k])):  
            if A[k][i]=='\t':  
                n.append(i)  
        T.append(float(A[k][0:n[0]]))  
        F.append(float(A[k][n[2]+1:len(A[k])-1]))  
    return [T,F]
```

```
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
```