TRAVAUX PRATIQUES
SCIENCES INDUSTRIELLES POUR L'INGENIEUR

Lycée Eiffel

Code COMAX	DC23 Modéliser la chaine de transmission de puissance d'un système	Série 3 Activité 1

Problématique	Comment modéliser un système pluri-technologique ?

Présentation

Le robot **Comax** est un robot collaboratif. Il fait partie de la famille des COBOT, robots dont la fonction est d'assister l'opérateur dans des opérations de déplacement d'objets de poids élevé.

Le **Comax** fait l'acquisition de l'intention de l'opérateur par un capteur d'effort. Cette information est traîtée au travers d'un algorithme complexe afin de piloter un moteur à courant continu pour assister l'opérateur dans l'effort développé.

Compétences

- 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 chaine 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	*doc. Constructeur*	**fltsi.fr rubrique Systèmes**
	Procédure	**Pilotage_comax.pdf**
	doc réponse	**COMAX_A3_DR1**

Mesures

Q1. Mettre en service le système à l'aide du logiciel en veillant à respecter les consignes de la procédure Pilotage_comax. Faire un essai en mode collaboratif avec les 3 masses additives en place.

Choisir le mode « asservissement de position » paramétrer le correcteur : Kp=100 ; Kd=0 ; Ki=0.

Q2. Procéder à un essai indiciel d'environ 100mm, et enregistrer les tracés de la position et du courant moteur.

Q3. Relever pour la position les résultats :

- 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,
- symétrie ou non du fonctionnement pour les 2 sens de rotation du bras.

Q4. Relever la valeur du pic de courant et du courant permanent lorsque la position est atteinte.

Q5. Augmenter la valeur du correcteur P afin d'obtenir un comportement pseudo périodique.

Q6. Effectuer le même essai et les mêmes relevés en enlevant 2 masses additives, comparer le temps de réponse et justifier son évolution.

Q7. Conclure sur le rôle du correcteur P et l'influence des paramètres dynamiques.

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 tracé l'évolution de la position et de calculer la vitesse à l'aide d'un programme Python.

Q8. A partir d'un essai indiciel, enregistrer les valeurs dans un fichier.

Q9. Ouvrir le fichier texte à l'aide de `Notepad++` (‘ouvrir avec’ à l'aide d'un clic droit).

Q10. Effacer toutes les lignes qui ne sont pas des valeurs au début et à la fin du fichier.

Q11. 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 `comax.py` **dans le même dossier.**

Q12. Ouvrir le fichier `comax.py` à l'aide de **Pyzo**.

Q13. Indiquer dans la fonction `os.chdir()`, le nom du chemin du dossier contenant le fichier de mesures.

Q14. Indiquer dans la fonction `open()`, le nom de votre fichier de mesures.

Q15. Tester le programme.

Q16. Sur le document-réponse 1, commenter chaque ligne des deux fonctions `valeurs` et `vitesse` en expliquant ce que le code effectue.

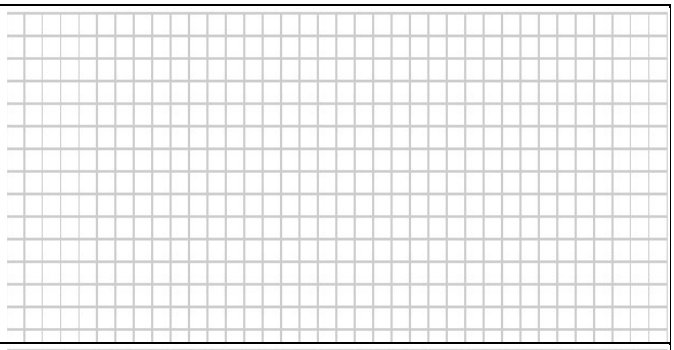
Q17. Compiler le programme et observer les courbes obtenues. Imprimer les pour le joindre au compte-rendu.

Q18. Modifier la fonction `valeurs` pour afficher le temps en ms plutôt qu'en secondes.

Q19. Imprimer les nouvelles courbes et les joindre au compte-rendu.

COMAX_A1_DR1

```
def valeurs(n):
    T=[]
    P=[]
    for i in range(n):
            donnees.readline(7)
            T.append(float(donnees.readline(3))*0.001)
            donnees.readline(14)
            P.append(float(donnees.readline(3)))
            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
```