

TD : Bras artificiel : Correcteur numérique

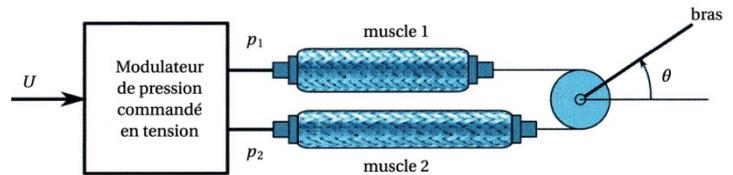
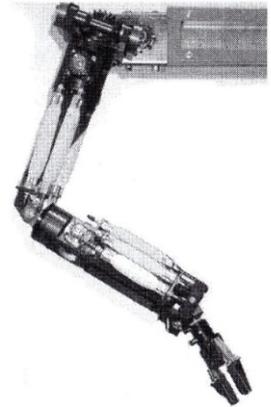
1- Présentation du système

Mise en situation

Cette étude concerne un manipulateur développé en laboratoire. Il se caractérise par une structure anthropomorphique à 7 degrés de liberté activés par des paires de muscles montés en opposition.

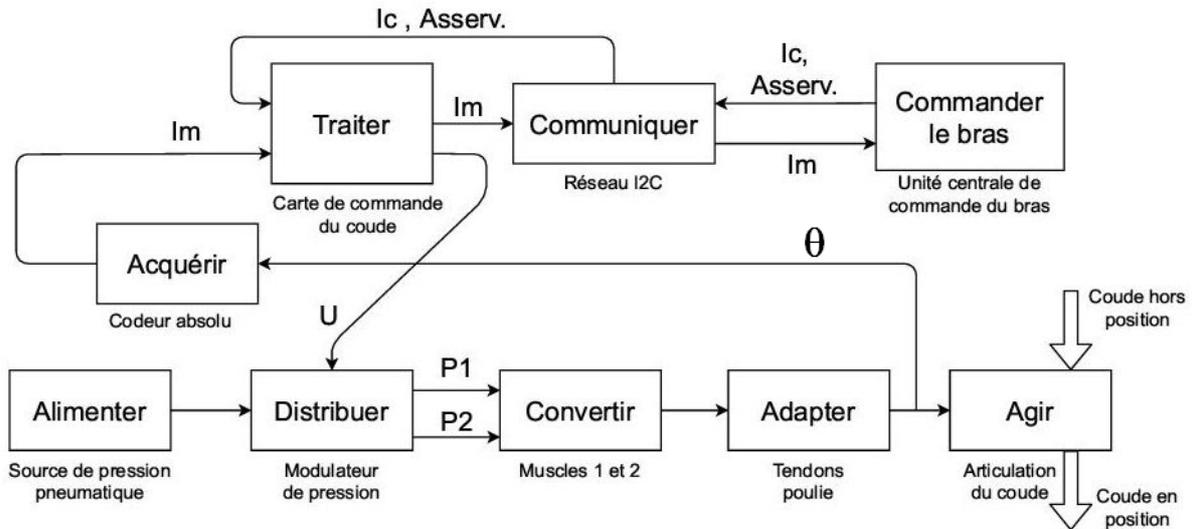
Un muscle est constitué d'une vessie en caoutchouc emprisonnée dans une tresse de fils. L'angle d'inclinaison de cette tresse permet de convertir le gonflement de la vessie, sous l'effet de la pression, en effort de traction.

La modulation de pression, réalisée à partir d'une tension de commande $u(t)$, permet alors de faire varier l'effort de traction. En associant deux muscles en opposition, on peut ainsi activer une articulation à l'aide d'un fil (tendon) relié aux deux extrémités des muscles et roulant sans glisser sur une poulie.



Structure de l'articulation du coude

La chaîne fonctionnelle du coude est décrite par le schéma ci-dessous :



On s'intéresse ici à la carte de commande du coude qui assure la commande de l'articulation du coude. Il s'agit d'un microprocesseur dont le code est écrit en Python. La carte de commande est en communication avec l'unité centrale de commande du bras par un réseau « I2C ».

- ☞ La commande « Acqu_Ic_Ass() » permet d'interroger l'unité centrale qui répond en renvoyant la consigne et si le coude doit être asservi en position. Cette fonction renvoie donc un entier (image de la position de consigne du coude) et un booléen qui indique si le système doit être asservi.
- ☞ La commande « Informer(Im) » permet d'envoyer sur le réseau « I2C » une trame à l'unité centrale qui indique la position réelle mesurée par le codeur absolu.
- ☞ La commande « Acqu_Position() » permet d'interroger le codeur absolu qui renvoie l'image de la position réelle du coude : Im. Cette fonction retourne donc l'entier Im image de θ .
- ☞ La commande « Envoyer_commande(U) » permet d'attribuer à la sortie de la carte de commande reliée au modulateur de pression, une tension U qui assure la commande du modulateur de pression.

Code du microprocesseur

Le code (incomplet) du microprocesseur est donné ci-dessous. On ne donne que la fonction principale qui est exécuté en boucle lorsque la carte de commande est active.

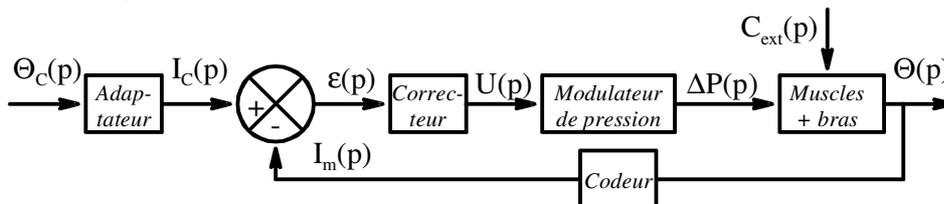
```
def asservissement():
    # Toutes les variables sont globales et sont initialisées à 0
    global Date,Datep,dt,Ic,Im,E,Ep,U1,U1p,U2,U2p,U,Up,M,Mp,Cext,Cons,Te,Tep,Tepp
    Cons=0.0      # Consigne de l'asservissement (Cons)
    Ic,Im=0,0     # Images de la consigne (Ic) et de la réponse (Im)
    E,Ep=0.0,0.0 # Ecart entre les images de la consigne et de la réponse
    U1,U1p=0.0,0.0 # Sortie du 1ier correcteur
    U2,U2p=0.0,0.0 # Sortie du 2nd correcteur
    U,Up=0.0,0.0  # Sortie du 3ième correcteur
    Date,Datep=0.0,0.0 # Dates actuelle et précédentes
    Asservissement = True # Booléen qui indique si l'articulation doit être asservie en position

    While Asservissement==True: # Boucle d'asservissement
        Datep=Date # Memorisation de la date précédente
        Date=t.time() # Acquisition de la date de l'horloge de la carte de commande
        dt=Date-Datep # Calcul de la durée d'échantillonnage
        Ic,Asservissement=Acqu_Ic_Ass() # Acquisition des consignes de l'unité centrale
        Im=Acqu_Position() # Pour mesurer l'image de la réponse à la Date actuelle
        Ep=E # Memorisation de l'écart précédent
        E=Ic-Im # Calcul de l'écart actuel
        Calcul_Commande() # Fonction à définir suivant le correcteur
        Envoyer_commande(U) # Envoie de la commande de tension au modulateur de pression
        t.sleep(0.005) # Temporisation de 5 ms

    Envoyer_commande(0) # Pour annuler la commande
```

Modélisation de l'asservissement

Une première étude a permis d'obtenir la structure de l'asservissement ci-dessous :



Une seconde étude a permis de déterminer la fonction de transfert non corrigée du système. Elle est la suivante : $H_{BONC}(p) = \frac{114,6}{(1 + 0,16.p + 0,1.p^2).(1 + 0,06.p)}$

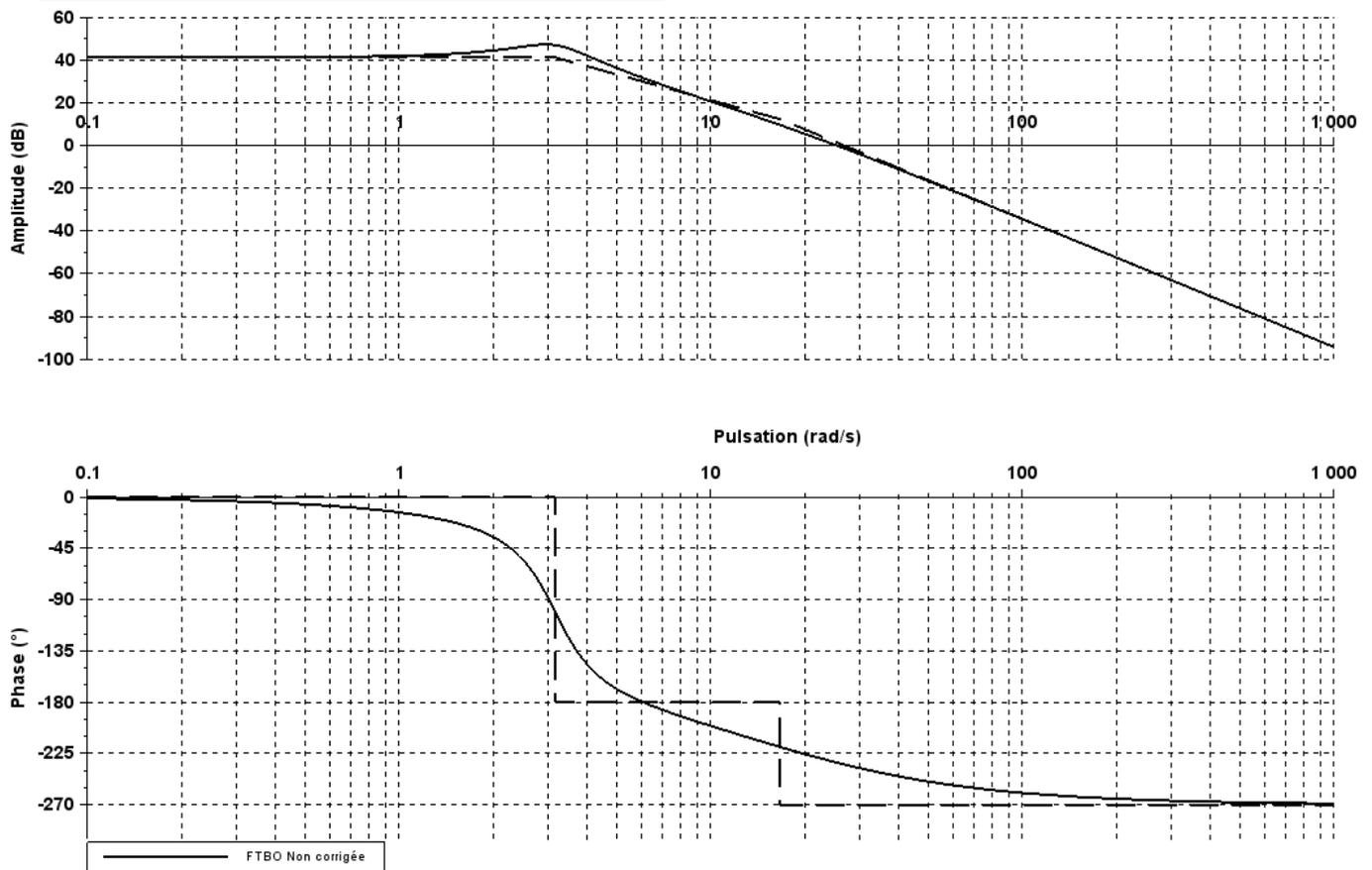
A partir du diagramme de Bode de la FTBO non corrigée (voir page suivante) et du cahier des charges, une 3^{ième} étude montre qu'il faut un correcteur à trois étages : $C(p) = K \cdot C_1(p) \cdot C_2(p) \cdot C_3(p)$:

$$\begin{matrix} \varepsilon(p) & \rightarrow & \boxed{C_1(p)} & \xrightarrow{U_1(p)} & \boxed{C_2(p)} & \xrightarrow{U_2(p)} & \boxed{C_3(p)} & \xrightarrow{U(p)} \end{matrix}$$

Avec : $C_1(p) = C_2(p) = \frac{1 + 0,26.p}{1 + 0,06.p}$

et : $C_3(p) = \frac{0,034.(1 + 0,343.p)}{p}$

Dans le code Python du microprocesseur les valeurs aux dates t et t-dt des fonctions temporelles ε(t) & ε(t-dt), u₁(t) & u₁(t-dt), u₂(t) & u₂(t-dt), et u(t) & u(t-dt) sont respectivement E & Ep , U1 & U1p , U2 & U2p , U & Up.

Diagramme de Bode de la FTBO non corrigée**Objectif du problème**

L'objectif du problème est de coder le correcteur. C'est-à-dire écrire en langage Python la fonction « Calcul_Commande() » qui permet d'obtenir la tension $u(t)$ à la sortie du correcteur (la variable « U »).

Travail demandé

1.1- Ecrire les relations numérique (avec la variable de Laplace) liant les fonctions symboliques, du correcteur : $\varepsilon(p)$, $U_1(p)$, $U_2(p)$ et $U(p)$.

1.2- Passer ces relations dans le domaine temporel.

1.3- Discrétiser ces relations : $\frac{d f(t)}{dt} = \frac{f(t) - f(t-dt)}{dt}$ et en déduire les expressions numériques de $u_1(t)$, $u_2(t)$ et $u(t)$ en fonction de $\varepsilon(t)$, $\varepsilon(t-dt)$, $u_1(t)$, $u_1(t-dt)$, $u_2(t)$, $u_2(t-dt)$ et $u(t-dt)$.

1.4- Le début du code Python de la fonction « Calcul_Commande() » est donné ci-dessous :

```
def Calcul_Commande():
    global dt, E, Ep, U1, U1p, U2, U2p, U
```

La commande de la première ligne permet d'indiquer que les variables dt , E , Ep , U_1 , U_{1p} , U_2 , U_{2p} et U sont des variables globales (et non pas des variables internes à la fonction).

Cette fonction ne renverra donc aucune valeur mais calculera les variables définies dans la fonction « Asservissement() » qui sont les mêmes que dans la fonction « Calcul_Commande() ».