

Mini Projet - LEGO EV3

Robot SCARA

version 2.0

Richard MOREAU

23 mars 2023

1 Objectif

Un robot SCARA (Selective Compliance Articulated Robot Arm) est un bras robotique en série qui peut être comparé au bras humain. Dans ce projet, le robot est composé de deux solides montés en série et reliés par des liaisons pivots parallèles. C'est un robot couramment utilisé dans l'industrie. Une vidéo de démonstration du robot Staübli TP80 est téléchargeable ici.

L'objectif de ce mini projet est de modéliser, concevoir et commander ce robot afin de réaliser des suivis de trajectoires.

2 Asservissement en position

Dans l'exercice 1.4, il a été demandé de faire tourner un moteur afin de réaliser 3 tours. Voici une proposition de solution :

```
from ev3dev2.motor import Motor, OUTPUT_A

# Création des variables
mon_moteur = Motor(OUTPUT_A)
mon_moteur.position=0 # Pour bien initialiser l'encodeur
puissance=30

# Boucle pour gérer le moteur
while mon_moteur.position<1080:
    mon_moteur.on(puissance)

# Fin du programme
mon_moteur.on(0)
```

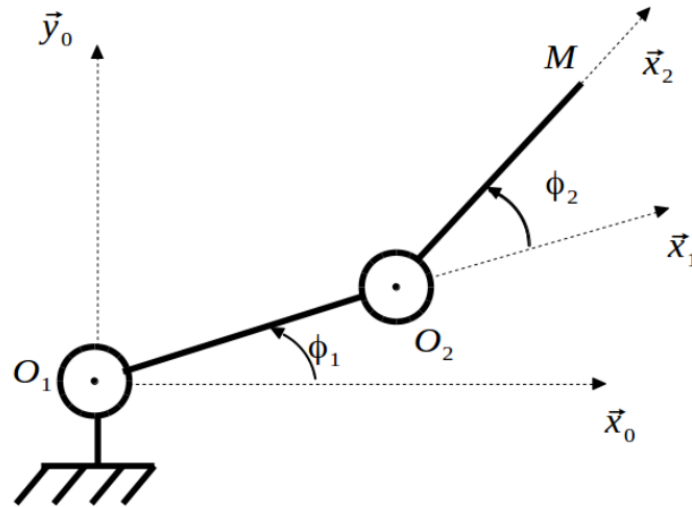
Or avec l'inertie du moteur, le moteur effectue un peu plus de 3 tours. Maintenant que les notions d'asservissement en boucle fermée sont assimilées, écrivez un programme avec une commande proportionnel permettant d'asservir le moteur en position et de le maintenir dans cette position. Testez différentes valeurs de gain et essayez de déplacer le moteur ainsi asservi à la main.

3 Suivi de trajectoire

On souhaite maintenant commander le moteur de manière à suivre une trajectoire donnée. Pour cela, créer un tableau décrivant une trajectoire angulaire que le moteur devra suivre point à point. Par exemple, votre moteur devra se déplacer de 0 ° à 90 ° avec une vitesse de 20 deg/s puis de 90 ° à 45 ° à une vitesse de 10 deg/s.

4 Modélisation du robot

Le robot étudié est un robot 2 axes. Il peut être représenté par le schéma cinématique suivant :



4.1 Paramétrage

les paramètres cinématiques sont ϕ_1 et ϕ_2 avec

— $\phi_1 = (\vec{x}_0, \vec{x}_1)$

— $\phi_2 = (\vec{x}_1, \vec{x}_2)$

les paramètres géométriques sont L_1 et L_2 avec

— $L_1 = \overrightarrow{O_1O_2} \cdot \vec{x}_1$

— $L_2 = \overrightarrow{O_2M} \cdot \vec{x}_2$

4.2 Position initiale et amplitudes

La position initiale est le bras tendu c'est à dire : $\phi_1 = 0$ et $\phi_2 = 0$.

On limitera ses amplitudes de mouvement selon ces bornes :

	minimum [deg]	maximum [deg]
ϕ_1	-100 °	100 °
ϕ_2	-110 °	110 °

4.3 Modèle Géométrique Direct (MGD)

Prenez un crayon et une feuille de papier et écrire les équations permettant d'exprimer les positions cartésiennes du point M (notées x_M et y_M) à partir des positions articulaires (ϕ_1 et ϕ_2).

4.4 Modèle Géométrique Inverse (MGI)

Avec le même crayon, écrivez maintenant les équations qui permettent d'exprimer les positions articulaires (ϕ_1 et ϕ_2) à partir des positions cartésiennes du point M .

5 Programmation du MGD

Faites un programme qui permet de passer des coordonnées articulaires aux coordonnées cartésiennes. Testez le sur des valeurs simples. Par exemple :

ϕ_1	ϕ_2	\Rightarrow	x_M	y_M
0	0		$L_1 + L_2$	0
$\pi/2$	0		0	$L_1 + L_2$
0	$\pi/2$		L_1	L_2
$\pi/2$	$-\pi/2$		L_2	L_1
$\pi/2$	$\pi/2$		$-L_2$	L_1

6 Programmation du MGI

Compléter votre programme avec le MGI pour passer des coordonnées cartésiennes aux coordonnées articulaires.

Testez le avec les valeurs précédentes et avec d'autres valeurs issues de votre MGD.

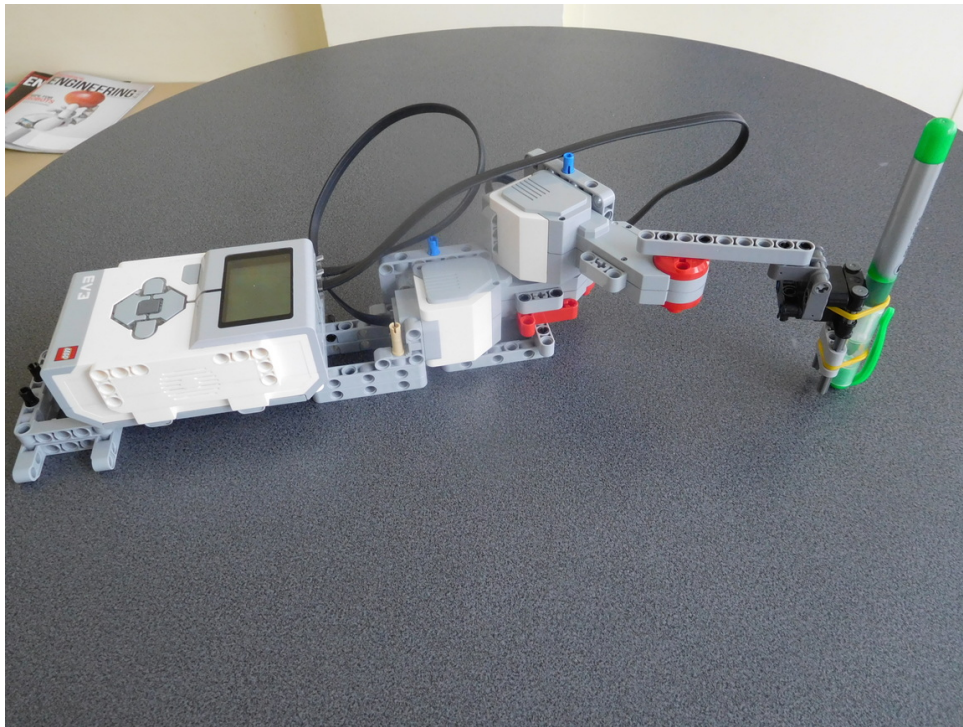
Écrivez un programme en Python qui permet de calculer les coordonnées cartésiennes à partir des coordonnées articulaires et inversement.

Remarques :

- Il faut utiliser la méthode `math.atan2(y,x)` afin d'avoir un résultat entre 0 et $2 * \pi$.
- Dans les fonctions trigonométriques, les angles doivent être exprimés en radians. N'oubliez pas que les codeurs des moteurs renvoient une position angulaire en degré. Vous pouvez utiliser `math.radians(angDeg)` ou `math.degrees(angRad)` pour passer d'une représentation à l'autre.
- Attention il peut exister plusieurs solutions pour une position de M donnée.

7 Conception du robot

Réaliser un robot de type SCARA à 2 axes en Lego.



8 Premiers essais

L'objectif de ces premiers essais est de valider la commande et le MGI.

Commander votre robot de manière qu'il se déplace d'un point à un autre. Par exemple, à partir de sa position initiale c'est à dire $(x_M, y_M) = (L_1 + L_2, 0)$ en coordonnées cartésiennes ce qui correspond à $(\phi_1, \phi_2) = (0, 0)$ en coordonnées articulaires, le robot doit se rendre au point $(x_M, y_M) = (L_1, L_2)$ en

coordonnées cartésiennes. Il est à noter que pour cette position il existe deux solutions en coordonnées articulaires $(\phi_1, \phi_2) = \left(0, \frac{\pi}{2}\right)$ ou $(\phi_1, \phi_2) = \left(\frac{\pi}{2}, -\frac{\pi}{2}\right)$.

9 Suivi de trajectoire

Maintenant que notre robot est capable de se déplacer en n'importe quel point de son espace de travail. Nous allons faire un suivi de trajectoire. Par exemple, nous pouvons définir une droite que l'organe terminal (le point M) doit suivre ou un cercle.

9.1 Suivi d'une droite

Déplacement du point M selon 1 droite de (x_1, y_1) à (x_2, y_2) .

9.2 Suivi d'un cercle

Déplacement du point M selon 1 cercle de centre $C (x_C, y_C)$ et de rayon R .