Langage Python A3 TD6

Modélisation géométrique du mouvement de la main mobile d'un robot

On propose de construire le modèle géométrique direct qui donne la position cartésienne de la main. Il faut donc le modéliser mathématiquement, afin de trouver la relation qui lie l'espace des coordonnées cartésiennes de la main (coordonnées locaux) et l'espace des coordonnées articulaires du robot(coordonnées absolues). Pour cela, il faut décrire à l'aide de matrices homogènes, l'effet de chaque degré de liberté sur la main.

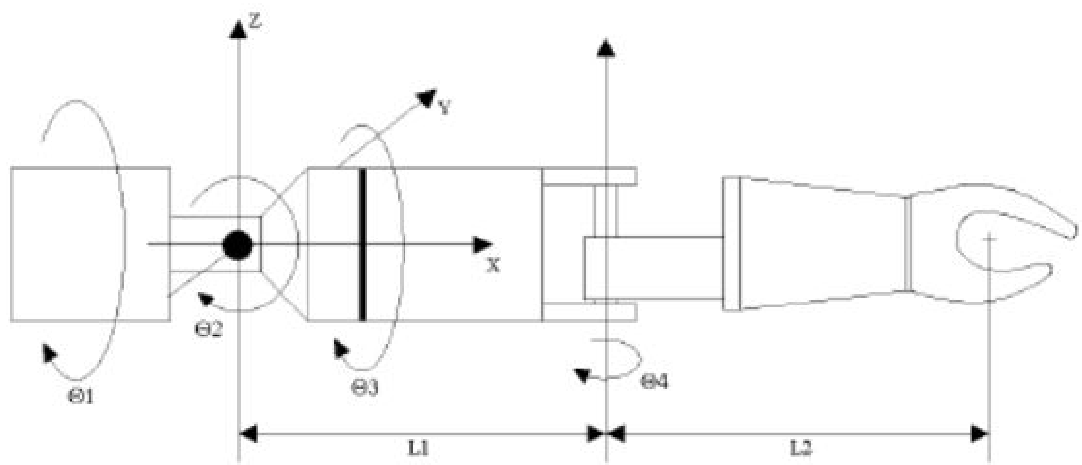


Fig.1: schéma de la modélisation géométrique de la main mobile d'un robot

Le modèle de la figure 1 montre que le modèle géométrique est de 4 degrés de liberté. La position de référence choisie est celle du bras tendu à l'horizontale. Pour faciliter les calculs, il faut commencer par le dernier degré de liberté et revenir vers le premier. En positionnant l'origine des coordonnées cartésiennes à l'épaule et l'axe x dans la direction du bras, cela donne :

1 : Une rotation de $\theta4$ autour de l'axe z décalée de L1 le long de l'axe x.
2 : Une rotation de $\theta3$ autour de l'axe x.
3 : Une rotation de $\theta2$ autour de l'axe y.
4 : Une rotation de $\theta1$ autour de l'axe x.

chaque transformation géométrique se traduit par une matrice (voir annexe au dessous). Une simple multiplication (matrice de transformation x vecteur de coordonnées (X, Y, Z)) servira pour obtenir les coordonnées (X', Y', Z') transformées d'un point :

$$\begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix} = (Matrice\ 4\text{D}\ de\ transformation) \times \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Formule 1

Création du langage script pour modéliser le problème du bras mobile du robot

1. Créer une classe **Vect4D** qui permet : (déjà fait ne perdez pas trop de temps ici)

- Initialiser un vecteur 4D ayant comme coordonnées (x,y,z,t)
- Afficher les coordonnées d'un vecteur
- Calculer le module d'un vecteur
- surcharger les opérations:
  - addition,
  - soustraction,
  - l'opérateur ==
  - multiplication scalaire entre 2 vecteurs
  - multiplication par un scalaire
  - set item et get item

2. Créer en utilisant la classe Vect4D et l'annexe ci-joint, une classe **Mat4D** qui permet:

- Initialiser une matrice 4D à travers 4 vecteurs lignes (v1,v2,v3,v4) de dimension 4 chacun
- Afficher la matrice
- Créer les **fonction**s **globales** suivantes (ne sont pas membres de la classe Mat4D)
  - La fonction Id4D() qui retourne la matrice identité
  - Les 3 fonctions SymX(), SymY(), et SymZ(), qui retournent respectivement les matrices de symétries par rapport aux axes x, y et z.

- Les 3 fonctions TransX(a), TransY(a), TransZ(a), qui retournent respectivement les 3 matrices de translation d'un vecteur de longueur a donné par rapport aux axes x, y et z.

- Les 3 fonctions RotX(θ), RotY(θ) et RotZ(θ) qui retournent respectivement les matrices de rotation d'un angle θ donnée par rapport aux axes x, y et z.

- surcharger les opérations:
    - addition,
    - soustraction,
    - set item et get item
    - l'opérateur ==
    - multiplication par un scalaire
    - multiplication M*N qui effectue les multiplications matrice*matrice et matrice*vecteur selon le cas où N est une instance de Mat4D ou Vect4D

3. Créer un **Widget** (fenêtre) de la library Tkinter de Python (voir l'exemple dans l'annexe ci dessous) qui prend en entrée les valeurs de θ1, θ2, θ3 , θ4 et L puis les coordonnées locaux X, Y et Z d'un point du bras afin d'obtenir les coordonnées absolues X', Y' et Z' du bras après le mouvement effectué. Les coordonnées X', Y' et Z' sont obtenues par la formule 1 défini ci-d essus où la matrice de transformation est M1*M2*M3*M4*L avec:
    - M1 est la matrice RotX(θ1)
    - M2 est la matrice RotY(θ2)
    - M3 est la matrice RotX(θ3)
    - M4 est la matrice RotZ(θ4)
    - L est la matrice TranX(L)

4. Proposer une approche pour enregistrer/charger données (à vous de choisir pickle ou json)

Annexe

La matrice d'identité est définie par:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Les matrices de symétrie par rapport aux axes x, y et z sont donnée respectivement par:

$$\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} , \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

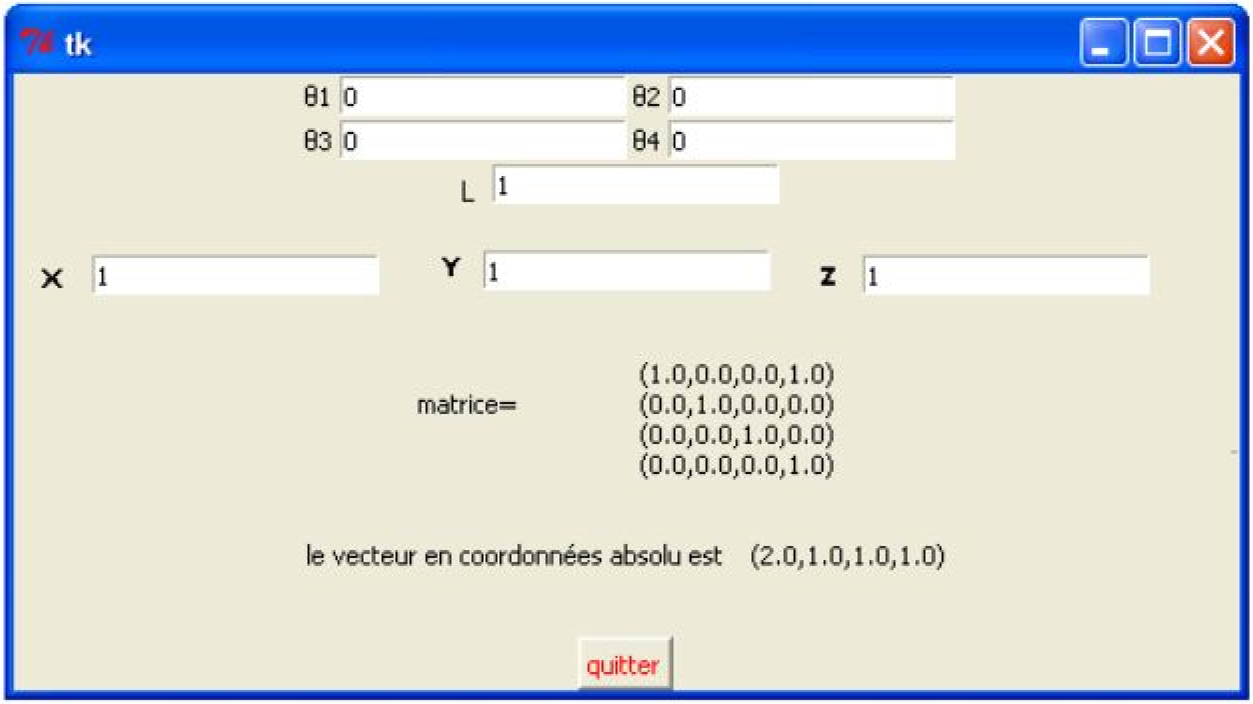
La matrice de translation par rapport à un vecteur V(X,Y,Z) est donnée par

$$\begin{pmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Les matrices de rotation d'un angle θ (en degré) tout autour des axes x, y et z sont donné respectivement par:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} , \quad \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Exemple d'un Widget (simpliste, à vous de faire mieux) affichant comme résultat la matrice de transformation et les coordonnées dans le repère absolue d'un point de la main du robot:


tk
θ1 0
θ2 0
θ3 0
θ4 0
L 1
X 1
Y 1
Z 1
matrice=
(1.0,0.0,0.0,1.0)
(0.0,1.0,0.0,0.0)
(0.0,0.0,1.0,0.0)
(0.0,0.0,0.0,1.0)
le vecteur en coordonnées absolu est (2.0,1.0,1.0,1.0)
quitter