Rapport TP4 :
Commande d’un robot mobile à roues.

Table des matières

Modèle Géométrique

Question 1 :

Modèle géométrique direct :

$$x = x_1 + x_2 \,; y = y_1 + y_2$$

$$x_1 = l_1 \cos(\theta_1) \;;\; y_1 = l_1 \sin(\theta_1)$$

$$x_2 = l_2 \cos(\theta_1 + \theta_2) \;;\; y_2 = l_2 \sin(\theta_1 + \theta_2)$$

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$

$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

Question 2 :

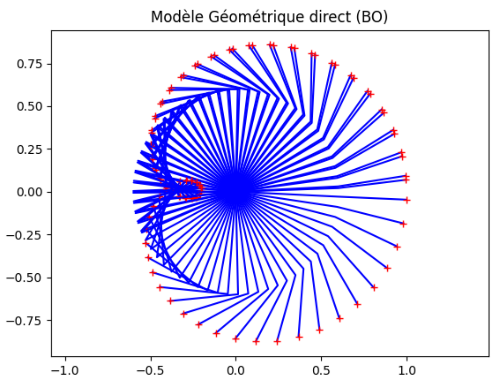
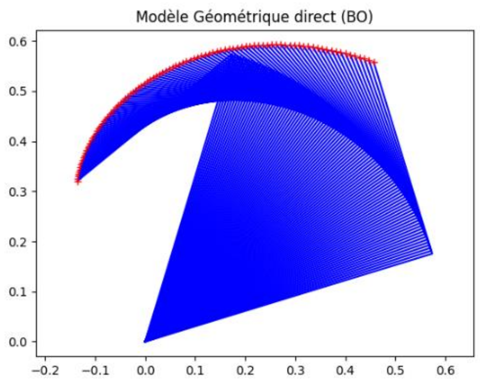
Il ne s’agit pas d’un modèle linéaire : nous avons un cosinus et un sinus dans les équations.

Question 3 :

La fonction MGD réalise le simulateur demandé :

```
x1 = l1*cos(theta1)
y1 = l1*sin(theta1)
x2 = l2*cos(theta1 + theta2)
y2 = l2*sin(theta1 + theta2)
x = x1 + x2; y = y1 + y2
x_bras = [0,x1,x]; y_bras = [0,y1,y]
x_bras = [0,x1,x]; y_bras = [0,y1,y]
```

Question 4 :

Vitesse = 10 ; angle=2° & 5°	Vitesse = 1 ; angle=17° & 90°
	

On observe qu'avec cette commande, chaque bras du robot tourne suis son axe, sans s'arrêter. Modifier la vitesse de rotation ou les coordonnées d'origines ne modifient en rien ce comportement.

Modèle cinématique direct

Question 5 :

Matrice Jacobienne du robot

$$J(q) = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} \end{bmatrix}$$

$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$

$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$

$\dot{x} = -l_1\dot{\theta}_1 \sin(\theta_1) - l_2(\dot{\theta}_1 + \dot{\theta}_2)\sin(\theta_1 + \theta_2)$

$\dot{y} = l_1\dot{\theta}_1 \cos(\theta_1) + l_2(\dot{\theta}_1 + \dot{\theta}_2)cos(\theta_1 + \theta_2)$

Donc :

$$J(q) = \begin{bmatrix} -l_1\dot{\theta}_1 \sin(\theta_1) - l_2(\dot{\theta}_2 + \dot{\theta}_2)\sin(\theta_1 + \theta_2) & -\theta_1 l_2 \sin(\theta_1 + \theta_2) \\ l_1\dot{\theta}_1 \cos(\theta_1) + l_2(\dot{\theta}_2 + \dot{\theta}_2)cos(\theta_1 + \theta_2) & l_2\dot{\theta}_2 cos(\theta_1 + \theta_2) \end{bmatrix}$$

Commande par modèle cinématique inverse

Question 6 :

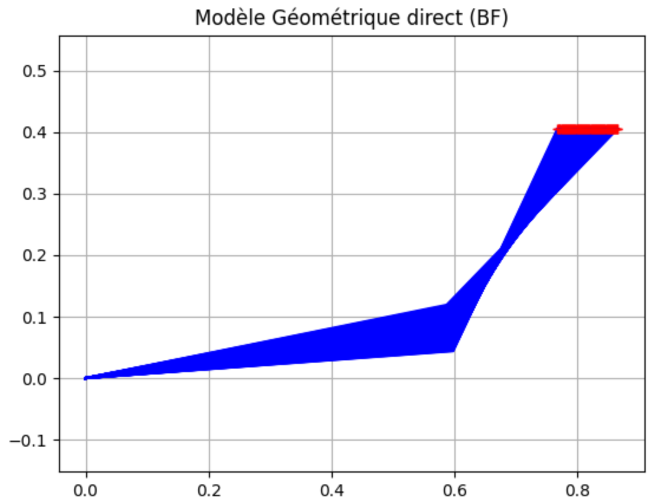
```python
for i in range(1,100):
    theta1 = q[0]
    theta2 = q[1]
    J = np.array([[-l1*sin(theta1)-l2*sin(theta1+theta2),-l2*sin(theta1+theta2)],
                  [l1*cos(theta1)+l2*cos(theta1+theta2),l2*cos(theta1+theta2)]])
    # a, b, c, d = J.flatten()
    # determinant = a * d - b * c
    # J1 = np.array([[d, -b], [-c, a]]) / (a * d - b * c)
    J1 = np.linalg.inv(J.T@J+1e-5*np.identity(2))@J.T

    q = (q + np.dot(J1,vitesse)*dT).reshape((2,1))

    MGD(theta1,theta2,i)
plt.title("Modèle Géométrique direct (BF)")
plt.show()
```

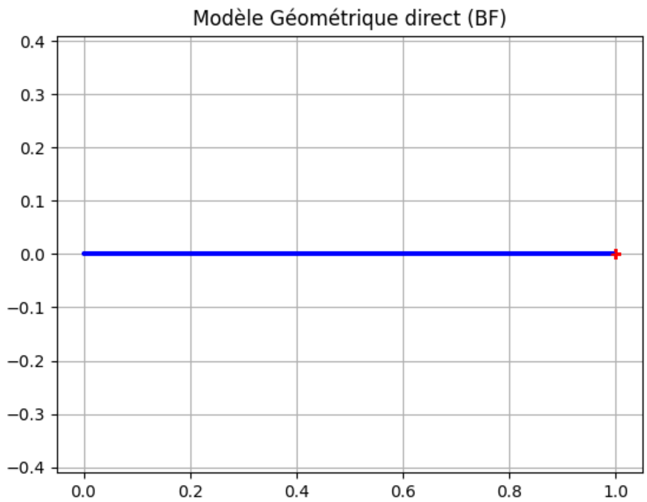
La matrice q conserve les angles. On stock les angles au moment n-1 dans des variables puis on calcul la matrice Jacobienne du robot, on l'inverse et on trace. Entre temps, on calcul et on stock les angles au moment n dans q.

Question 7 :



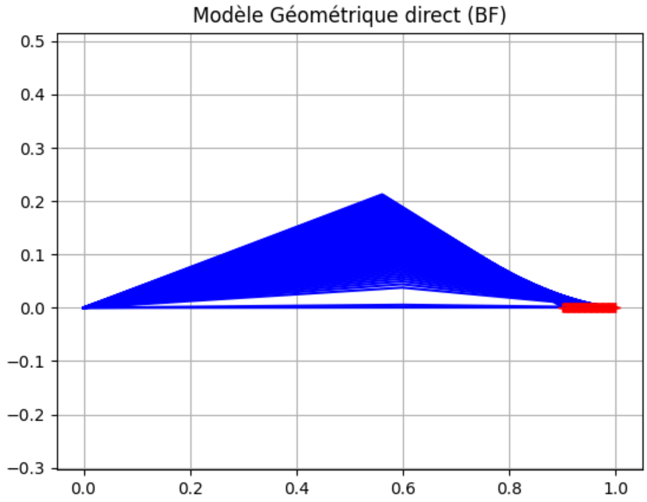
Le robot se déplace uniquement à l'horizontal, vers la gauche. Le programme réalise bien ce qu'on lui a demandé.

Question 8 :



Avec des angles initiaux nuls, les bras du robot ne bougent pas.

Question 9 :



Avec des angles initiaux différents de zéro (mais proche), le robot effectue bien la commande qu'on lui donne (déplacement vers la gauche).

Question 10 :

Question 11 :

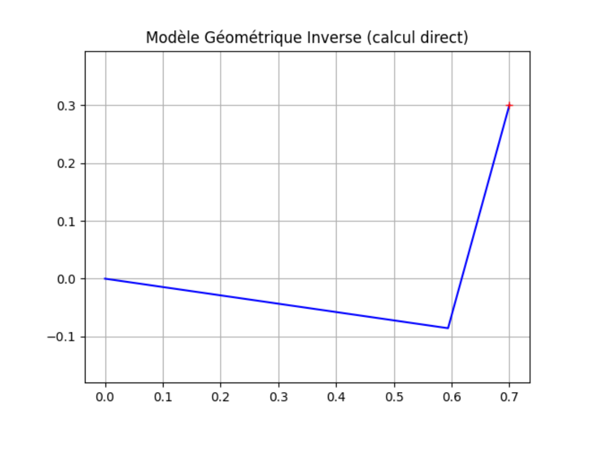
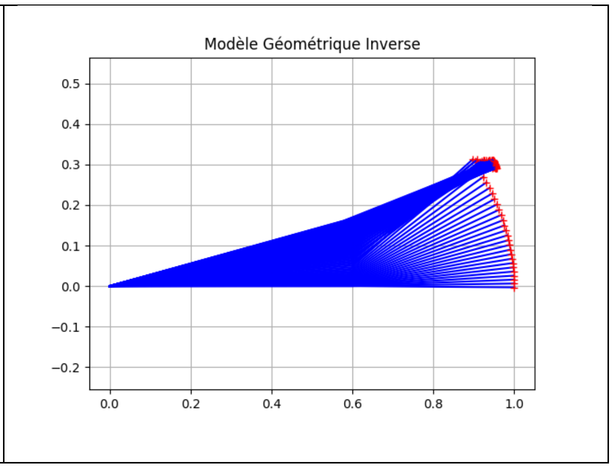
On souhaite maintenant faire une commande inverse, c'est-à-dire que l'on commande les angles en fonction des coordonnées inverse.

$$\theta_2 = \pi - arccos\left(\frac{{l_2}^2 + {l_1}^2 - l^2}{2 \times l_2 \times l_1}\right)$$

Cependant, après test, cette équation ne fonctionnait pas à cause du « $\pi -$ ». Je l'ai donc intégré dans l'équation :

$$\theta_2 = arccos\left(\frac{-{l_2}^2 - {l_1}^2 + l^2}{2 \times l_2 \times l_1}\right)$$

$$\theta_1 = \arctan\left(\frac{y}{x}\right) - arccos\left(\frac{{l_2}^2 + l^2 - {l_1}^2}{2 \times l_2 \times l}\right)$$


Modèle Géométrique Inverse (calcul direct)
0.3
0.2
0.1
0.0
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

Modèle Géométrique Inverse
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
0.0
0.2
0.4
0.6
0.8
1.0

Question 12 :