

TP 2 : Modélisation géométrique et inverse d'un robot Plan (3DDL)

1. Objectifs

L'objectif de ce travail pratique est de mettre en œuvre les concepts de modélisation géométrique pour un robot manipulateur sériel simple. À l'issue de cette séance, l'étudiant doit être capable de :

- Définir un modèle de robot à l'aide des paramètres de Denavit-Hartenberg (DH).
- Implémenter et valider le **modèle géométrique direct (MGD)** d'un robot plan.
- Implémenter et valider le **modèle géométrique inverse (MGI)** par une approche géométrique.
- Visualiser la configuration du robot et son espace de travail.
- Analyser les problèmes liés à la cinématique inverse (multiplicité des solutions, singularités).

2. Prérequis

- Logiciel : MATLAB avec la Robotics Toolbox (Peter Corke).
- Connaissances : TP1 (Transformations géométriques), paramètres de Denavit-Hartenberg, notions de cinématique directe et inverse.

3. Manipulation

3.1. Définition du robot

On considère un robot manipulateur plan à 3 articulations rotoïdes (RRR) évoluant dans le plan (x, y). Ses paramètres géométriques sont les suivants :

- Longueurs des segments : $L_1 = 1$, $L_2 = 0.8$, $L_3 = 0.4$.
- Les articulations sont toutes rotoïdes d'axe z (rotation dans le plan xy).

1. **Paramètres de Denavit-Hartenberg** : Complétez le tableau des paramètres de DH (convention standard) pour ce robot. Le repère outil est situé à l'extrémité du dernier segment.

Liens	θ_i	d_i	a_i	α_i
1				
2				
3				

2. **Création du robot dans la toolbox** : Utilisez la fonction Link de la Robotics Toolbox pour créer chaque articulation. Le constructeur est de la forme : `Link([theta, d, a, alpha])`, où θ est l'offset angulaire (ici, les variables articulaires sont q_1, q_2, q_3).
3. **Exploration** : Affichez les informations du robot avec la commande `robot.plot()` (pour une configuration nulle) et `robot.teach()`.

3.2. Modèle Géométrique Direct (MGD)

Le MGD permet de calculer la position (x, y) et l'orientation (ϕ) de l'organe effecteur en fonction des variables articulaires $q = [q_1, q_2, q_3]$.

1. **Calcul pour une configuration** : Pour une configuration articulaire $q = [\pi/4, -\pi/4, \pi/6]$, calculez la transformation homogène $T = \text{robot.fkine}(q)$. Extraire la position $(T.t)$ et l'orientation $(T.R$ ou $\text{tr2rpy}(T))$. Vérifiez la cohérence des résultats.
2. **Forme explicite** : En utilisant les lois de la trigonométrie, donnez l'expression littérale de (x, y) en fonction de q_1, q_2, q_3 et des longueurs. Comparez avec les résultats numériques obtenus avec `fkine`.
3. **Espace de travail** :
 - Générez aléatoirement un grand nombre de configurations articulaires (ex: 5000) dans les limites $[-\pi, \pi]$ pour chaque articulation.
 - Pour chaque configuration, calculez la position (x, y) de l'effecteur à l'aide du MGD.
 - Tracez ces positions sur un graphique 2D (`plot(x, y, 'r')`). Que représente ce nuage de points ?
 - Identifiez les limites de l'espace de travail atteignable.

3.3. Modèle Géométrique Inverse (MGI) - Approche Géométrique

Le MGI consiste à trouver les articulations q permettant d'atteindre une pose désirée (x, y, ϕ) . Pour ce robot plan, on peut le résoudre géométriquement en décomposant le problème.

1. Décomposition du problème :

- Soit (x, y) la position de l'effecteur et ϕ son orientation dans le plan (par rapport à l'axe x).
- La position du poignet (articulation 3) est notée (x_w, y_w) . Exprimez (x_w, y_w) en fonction de (x, y) , ϕ et L_3 .
- Le problème se ramène alors à un robot plan à 2 degrés de liberté (les segments 1 et 2) devant placer son effecteur (le poignet) en (x_w, y_w) .

2. Résolution pour q_1 et q_2 :

- Calculez la distance $r = \sqrt{x_w^2 + y_w^2}$ entre la base et le poignet.
- En utilisant le théorème d'Al-Kashi dans le triangle formé par les segments L_1 et L_2 , déterminez l'expression de q_2 . Combien de solutions existe-t-il pour q_2 ? À quelle condition ces solutions existent-elles ?
- Déterminez q_1 en fonction de l'angle β (angle du vecteur (x_w, y_w)) et de l'angle interne ψ (calculé à partir de L_1, L_2, r).

3. Résolution pour q_3 :

- Une fois q_1 et q_2 connus, déduisez q_3 sachant que l'orientation totale est $\phi = q_1 + q_2 + q_3$.

4. Implémentation :

- Écrivez une fonction `q = mgi_robot(robot, T_désirée, config)` où `config` est un paramètre permettant de choisir entre les différentes solutions possibles (ex: 'elbow_up' ou 'elbow_down' pour q_2). La fonction devra retourner un vecteur $q = [q_1, q_2, q_3]$.
- Gérer les cas où la pose désirée est hors d'atteinte.

3.4. Validation et Test

1. **Test sur une pose :** Choisissez une configuration articulaire q_{ini} (ex: $[0.5, 0.8, -0.3]$). Calculez la pose correspondante T_{ini} par MGD. Utilisez votre fonction MGI pour retrouver une configuration $q_{retrouve}$ à partir de T_{ini} . Vérifiez que `robot.fkine(q_retrouve)` redonne bien T_{ini} (à une tolérance près). Testez les deux configurations (coude haut/coude bas).
2. **Suivi de trajectoire :**
 - Définissez une trajectoire rectiligne dans l'espace opérationnel. Par exemple, l'effecteur doit se déplacer en ligne droite du point A $(0.7, 0.5)$ au point B $(1.5, 1.2)$, avec une orientation constante $\phi = 0$.
 - Générez une série de poses intermédiaires le long de cette droite (utilisez `trinterp` ou une interpolation linéaire).
 - Pour chaque pose intermédiaire, calculez la configuration articulaire correspondante à l'aide de votre MGI (en gardant une solution cohérente, par exemple en évitant les changements de configuration).
 - Visualisez le mouvement du robot le long de cette trajectoire avec `robot.plot(q_trajectoire)`.

3.5. Analyse des singularités

1. **Cas limites :** Testez votre MGI pour des positions à la limite de l'espace de travail (ex: $x_w = L1+L2, y_w = 0$). Que se passe-t-il pour q_2 ?
2. **Configuration singulière :** Placez le robot dans une configuration où le bras est complètement déplié ($q_2 = 0$). Animez l'effecteur pour qu'il décrive un petit cercle autour de cette position. Observez le comportement des articulations (notamment q_1). Est-il possible de passer en douceur de la solution "coude haut" à "coude bas" ?

4. Compte rendu

Le compte-rendu devra contenir :

- Le tableau des paramètres DH.
- Les codes MATLAB (script principal et fonctions) clairement commentés.
- Les figures illustrant l'espace de travail, la trajectoire suivie et les configurations du robot.
- Une analyse des résultats obtenus pour le MGD et le MGI, en insistant sur les problèmes de solutions multiples, de conditions d'existence et de singularités.
- Une interprétation physique des phénomènes observés lors de l'étude des singularités.

Le fichier source Matlab (`.m`) devra être joint au compte-rendu.