

TP 3 : Modélisation cinématique directe et inverse

1. Objectifs

L'objectif de ce travail pratique est d'étendre la modélisation du robot plan à l'étude de sa cinématique (vitesses). À l'issue de cette séance, l'étudiant doit être capable de :

- Comprendre et calculer la **matrice Jacobienne** d'un robot manipulateur.
- Interpréter physiquement la Jacobienne (relation entre vitesses articulaires et vitesses opérationnelles).
- Mettre en œuvre le **modèle cinématique inverse** pour le suivi de trajectoire.
- Identifier et analyser les **singularités** cinématiques à partir de la Jacobienne.
- Aborder la notion de **redondance** et les espaces associés.

2. Manipulations

2.1. Rappel et préparation

On reprend le robot plan RRR à 3 degrés de liberté défini dans le TP2 :

- Longueurs : $L1 = 1$, $L2 = 0.8$, $L3 = 0.4$.
 - Variables articulaires : $q = [q1, q2, q3]$.
1. **Chargement du robot** : Rechargez le modèle du robot créé précédemment ou recréez-le si nécessaire.
 2. **Rappel du MGD** : Pour une configuration $q = [\pi/4, -\pi/6, \pi/3]$, calculez la position (x, y) de l'effecteur et son orientation ϕ .

2.2. Calcul de la matrice Jacobienne

La matrice Jacobienne $J(q)$ relie les vitesses articulaires $\dot{q}$ aux vitesses opérationnelles $\dot{X}$ (vitesses linéaires et angulaire de l'effecteur) :

$$\dot{X} = J(q) * \dot{q}$$

1. Jacobienne géométrique avec la toolbox :

- Utilisez la méthode `robot.jacobe(q)` pour calculer la Jacobienne (dans le repère de base) pour la configuration q définie ci-dessus. La toolbox retourne une matrice de dimension (6x3) car elle inclut les 3 vitesses linéaires et les 3 vitesses angulaires. Dans notre cas plan, seule une sous-matrice nous intéresse.
- Affichez `J_full` et identifiez les lignes correspondant à v_x , v_y et ω_z (vitesse angulaire autour de z).

2. Extraction de la Jacobienne opérationnelle : Extrayez la matrice Jacobienne réduite `J_op` de dimension (3x3) qui relie $\dot{q}$ à $[v_x; v_y; \omega_z]$.

3. Interprétation :

- Que vaut la troisième colonne de `J_op` ? Que signifie-t-elle physiquement ?
- Calculez le déterminant de `J_op`. Que représente-t-il ?

4. Validation par le calcul analytique :

- En utilisant les expressions du MGD du TP2, calculez analytiquement les dérivées partielles de x , y et ϕ par rapport à q_1 , q_2 , q_3 . Vous devez obtenir une matrice 3x3.
- Évaluez cette matrice pour la même configuration q et comparez avec `J_op` obtenue précédemment (les valeurs doivent être identiques).

2.3. Étude des singularités

Les singularités correspondent aux configurations où la Jacobienne perd son rang (déterminant nul). Dans ces configurations, le robot perd une ou plusieurs capacités de mouvement dans certaines directions de l'espace opérationnel.

1. Recherche des singularités :

- Définissez une plage de variation pour q_1 et q_2 (par exemple $q_1 = -\pi:0.1:\pi$, $q_2 = -\pi:0.1:\pi$). Fixez $q_3 = 0$ (ceci n'affecte pas le rang de la Jacobienne pour les singularités de bras).
- Pour chaque couple (q_1, q_2) , calculez la Jacobienne opérationnelle `J_op` et son déterminant.
- Tracez une carte 2D (en utilisant `surf` ou `imagesc`) de la valeur absolue du déterminant en fonction de q_1 et q_2 . Identifiez les zones où le déterminant s'annule.
- À l'aide de `robot.plot`, visualisez les configurations correspondant à ces zones. Qu'ont-elles en commun ?

2. Analyse des types de singularités :

- **Singularité de bord d'espace de travail** : Pour $q_2 = 0$ (bras déplié), que vaut le déterminant ? Placez le robot dans cette configuration et essayez de déplacer l'effecteur radialement (vers l'extérieur) puis orthogonalement. Que constatez-vous ?
- **Singularité intérieure** : Pour $q_2 = \pi$ (bras replié), même analyse. Quelle direction de mouvement devient impossible ?

3. Manipulabilité :

- Calculez et tracez l'indice de manipulabilité de Yoshikawa $w = \sqrt{\det(J_{op} * J_{op}^T)}$ (qui se réduit à $\text{abs}(\det(J_{op}))$ pour une matrice carrée). Commentez la forme de cette carte.

2.4. Modèle cinématique inverse (MCI)

Le MCI consiste à calculer les vitesses articulaires $\dot{q}$ nécessaires pour obtenir une vitesse opérationnelle désirée $\dot{X}_{des}$. On utilise pour cela l'inverse de la Jacobienne (hors singularités) : $\dot{q} = J_{op}^{-1} * \dot{X}_{des}$.

1. **Pseudo-inverse** : En vue des singularités, on utilisera plutôt la **pseudo-inverse** $J^+ = J_{op}^T * \text{inv}(J_{op} * J_{op}^T)$ (moindres carrés). La toolbox fournit directement `pinv(J_op)`.
2. **Suivi de trajectoire par commande en vitesse** :
 - Définissez une trajectoire circulaire dans l'espace opérationnel, centrée en (0.8, 0.8) de rayon 0.3, à parcourir en 10 secondes. L'orientation peut être maintenue constante $\phi = 0$.
 - Générez les points de consigne ($x_d(t)$, $y_d(t)$, $\phi_d(t)$) et leurs dérivées ($\dot{x}_d(t)$, $\dot{y}_d(t)$, $\dot{\phi}_d(t)$) (pour un cercle, ces dérivées s'obtiennent analytiquement).
 - Implémentez une boucle d'intégration temporelle (schéma d'Euler) pour calculer l'évolution des articulations
 - Visualisez la trajectoire du robot avec `robot.plot(q_trajectory)`. Tracez également l'évolution des erreurs de position en fonction du temps.
3. **Analyse des résultats** :
 - Que se passe-t-il lorsque la trajectoire approche une configuration singulière ?
 - Modifiez la trajectoire pour passer volontairement près d'une singularité (ex: bras déplié). Observez l'explosion des vitesses articulaires. Proposez une solution pour limiter ce phénomène (amortissement, `pinv(J_op, epsilon)`).

2.5. Extension au cas redondant (optionnel)

Notre robot a 3 articulations pour 3 tâches opérationnelles (x , y , ϕ) : il est non redondant. Pour explorer la redondance, nous allons considérer que seule la position (x , y) nous intéresse, l'orientation devenant libre. La Jacobienne devient alors de dimension (2x3).

1. **Nouvelle Jacobienne** : $J_{red} = J_{op}(1:2, :)$ (les lignes de $\dot{x}$ et $\dot{y}$ seulement).
2. **Espace nul** : Calculez et visualisez l'espace nul de J_{red} pour différentes configurations. Un vecteur $\dot{q}_{nul}$ dans l'espace nul produit un mouvement des articulations sans déplacer l'effecteur. C'est ce qu'on appelle les **mouvements autopalmes**.
3. **Commande avec optimisation** : Utilisez la forme générale de la commande pour un robot redondant :

$$\dot{q} = J_{red}^+ * \dot{X}_d + (I - J_{red}^+ * J_{red}) * \dot{q}_0$$

où $\dot{q}_0$ est un vecteur de vitesses articulaires correspondant à une tâche secondaire (ex: éviter les butées, éviter les singularités, garder l'effecteur éloigné d'un obstacle).

4. Simulation :

- Faites suivre à l'effecteur une trajectoire rectiligne simple.
- Sans tâche secondaire ($\dot{q}_0 = 0$), observez l'évolution de q_3 (l'orientation libre).
- Ajoutez une tâche secondaire qui cherche à ramener q_3 vers zéro (ou à éviter une butée). Implémentez cela en choisissant $\dot{q}_0 = -K * (q_3 - q_{3_desire})$ par exemple. Observez l'effet sur le mouvement.

4. Compte-rendu

Le compte-rendu devra contenir :

1. **Introduction** : Rappel du robot étudié et des objectifs.
2. **Méthodologie** : Explication des différentes approches (analytique, toolbox) pour le calcul de la Jacobienne.
3. **Résultats expérimentaux** :
 - La matrice Jacobienne obtenue et sa validation.
 - Les cartes du déterminant et de la manipulabilité avec analyse des zones singulières.
 - Les graphiques d'évolution des erreurs de position lors du suivi de trajectoire.
 - Les observations sur le comportement aux abords des singularités.
 - (Optionnel) Les résultats sur la redondance et les mouvements autopalmes.
4. **Discussion et analyse** : Interprétation physique des singularités, limites de l'approche par inversion de Jacobienne, solutions pour les singularités, intérêt de la redondance.
5. **Conclusion** : Bilan des compétences acquises.
6. **Annexes** : Codes MATLAB commentés.

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