

TP 5 : Génération de trajectoires en mode articulaire et cartésien

1. Objectifs

L'objectif de ce travail pratique est d'étudier les méthodes de génération de trajectoires pour un robot manipulateur. À l'issue de cette séance, l'étudiant doit être capable de :

- Comprendre la différence entre trajectoire articulaire et trajectoire cartésienne.
 - Implémenter des générateurs de trajectoire polynomiale (ordre 3, ordre 5) et trapézoïdale (ligne droite avec accélérations constantes).
 - Générer des trajectoires dans l'espace articulaire et dans l'espace opérationnel.
 - Analyser les contraintes et les limites de chaque approche (passage par des singularités, respect des contraintes cinématiques).
 - Visualiser et comparer les trajectoires obtenues.
-

2. Manipulations

2.1. Définition du robot

On reprend le robot plan RRR à 3 degrés de liberté défini dans les TP précédents.

Rappel des fonctions utiles :

- `robot.fkine(q)` : modèle géométrique direct
- `robot.ikine(T, q0)` : modèle géométrique inverse (itératif)
- `robot.jacobe(q)` : matrice Jacobienne
- `robot.plot(q)` : visualisation

2.2. Génération de trajectoires dans l'espace articulaire

L'espace articulaire consiste à planifier l'évolution des variables articulaires $q(t)$ entre une configuration initiale q_i et une configuration finale q_f .

2.2.1. Trajectoire polynomiale d'ordre 3 (cubique)

Un polynôme d'ordre 3 permet d'imposer les positions et vitesses initiales et finales :

$$q(t) = a_0 + a_1 * t + a_2 * t^2 + a_3 * t^3$$

1. Implémentation :

Écrivez une fonction $[q, qd, qdd, t] = \text{traj_cubique}(q_i, q_f, T, dt)$ qui génère une trajectoire cubique entre q_i et q_f en temps T (avec vitesses initiale et finale nulles).

2. Test :

- Générez une trajectoire cubique pour l'articulation 1 entre $q_i = 0$ et $q_f = \pi/2$ en $T = 2s$ avec $dt = 0.01s$.
- Tracez $q(t)$, $qd(t)$ et $qdd(t)$ sur trois graphiques différents. Commentez la continuité des profils.

2.2.2. Trajectoire polynomiale d'ordre 5 (quintique)

Un polynôme d'ordre 5 permet d'imposer en plus les accélérations initiales et finales (généralement nulles).

1. Implémentation :

Écrivez une fonction $[q, qd, qdd, t] = \text{traj_quintique}(q_i, q_f, T, dt)$ avec contraintes : $q(0)=q_i$, $q(T)=q_f$, $qd(0)=0$, $qd(T)=0$, $qdd(0)=0$, $qdd(T)=0$.

Les coefficients sont donnés par :

$$\begin{array}{lll} a_0 = q_i & a_2 = 0 & a_4 = -15 * (q_f - q_i) / T^4 \\ a_1 = 0 & a_3 = 10 * (q_f - q_i) / T^3 & a_5 = 6 * (q_f - q_i) / T^5 \end{array}$$

2. Comparaison :

- Générez la même trajectoire qu'avec la cubique.
- Superposez les courbes de vitesse et d'accélération des deux méthodes. Que constatez-vous ? Quelle trajectoire est la plus lisse ?

2.2.3. Trajectoire trapézoïdale (LSPB - Linear Segment with Parabolic Blends)

Ce type de trajectoire est caractérisé par une phase d'accélération constante, une phase à vitesse constante, et une phase de décélération constante.

1. Principe : Pour une vitesse maximale $v_{\max}$ et une accélération $a_{\max}$ données, on calcule :

- Le temps d'accélération : $t_{\text{acc}} = v_{\max} / a_{\max}$
- La distance parcourue pendant les phases d'accélération/décélération : $d_{\text{acc}} = 0.5 * a_{\max} * t_{\text{acc}}^2$
- Si $2 * d_{\text{acc}} < |q_f - q_i|$, il y a une phase à vitesse constante.
- Sinon, le profil est triangulaire (pas d'atteinte de la vitesse maximale).

2. Implémentation :

Écrivez une fonction $[q, qd, qdd, t] = \text{traj_trapeze}(q_i, q_f, v_{\max}, a_{\max}, dt)$.

3. Test :

- Testez avec $q_i = 0$, $q_f = \pi$, $v_{\max} = 1 \text{ rad/s}$, $a_{\max} = 2 \text{ rad/s}^2$.

- Tracez les profils. Identifiez les trois phases sur les graphiques.

2.2.4. Trajectoire multi-articulations

Pour un robot à plusieurs articulations, on génère une trajectoire pour chaque articulation indépendamment.

1. Configuration initiale et finale :

```
qi = [0, 0, 0];
qf = [pi/2, -pi/3, pi/4];
T = 3;
dt = 0.01;
```

2. Génération :

- Générez une trajectoire quintique pour chaque articulation (avec le même T).
- Concaténez les résultats dans des matrices Q, Qd, Qdd de dimensions (n_points x 3).
- Visualisez l'évolution des trois angles sur un même graphique.

3. Animation :

- Utilisez robot.plot(Q) pour animer le mouvement du robot.
- Observez le mouvement dans l'espace cartésien. La trajectoire de l'effecteur est-elle une ligne droite ?

2.3. Génération de trajectoires dans l'espace cartésien

L'espace cartésien consiste à planifier une trajectoire pour l'organe effecteur directement dans l'espace opérationnel (x, y, ϕ), puis à calculer les configurations articulaires correspondantes par MGI.

2.3.1. Trajectoire rectiligne

1. Génération d'une trajectoire cartésienne :

- Définissez une pose initiale T_i et une pose finale T_f (matrices de transformation homogène).
- Utilisez la fonction trinterp pour interpoler linéairement entre ces deux poses :
- T_{traj} est un tableau de matrices de transformation (4x4xn_points).

2. Extraction de la trajectoire :

- Pour chaque point, extrayez la position (x, y) et l'orientation ϕ (utilisez transl(T) et tr2rpy(T)).
- Tracez la trajectoire cartésienne dans le plan xy. Vérifiez qu'il s'agit bien d'une ligne droite.

2.3.2. Passage dans l'espace articulaire

1. MGI pour chaque point :

- Pour chaque pose $T_{\text{traj}}(:, :, k)$, calculez la configuration articulaire correspondante en utilisant robot.ikine().

- Attention : il faut fournir une configuration initiale q_0 pour l'algorithme itératif. Pour la continuité, utilisez le résultat du pas précédent comme initialisation.

2. Analyse :

- Tracez l'évolution des articulations $q_{\text{cart}}(:,1)$, $q_{\text{cart}}(:,2)$, $q_{\text{cart}}(:,3)$ en fonction du temps.
- Comparez ces profils avec ceux obtenus en trajectoire articulaire (mêmes points de départ et d'arrivée).
- Animez le mouvement avec `robot.plot(q_cart)`. Observez la trajectoire de l'effecteur. Est-elle toujours rectiligne ?

2.3.3. Problème des singularités

1. Choix d'une trajectoire problématique :

- Choisissez une trajectoire cartésienne qui passe près d'une singularité du robot (par exemple, q_2 proche de 0 ou π).

2. Génération et observation :

- Générez la trajectoire cartésienne entre ces deux poses.
- Calculez les configurations articulaires correspondantes.
- Que se passe-t-il au voisinage de la singularité ? Observez les vitesses articulaires (différence finie entre points successifs).
- Calculez le conditionnement de la Jacobienne le long de la trajectoire (`cond(robot.jacobe(q))`). Tracez son évolution.

2.4. Comparaison et analyse

2.4.1. Critères de comparaison

Pour les deux types de trajectoire (articulaire et cartésienne) entre les mêmes poses initiale et finale, calculez et comparez :

1. **Temps de calcul** : Utilisez tic et toc pour mesurer le temps de génération.
2. **Respect de la trajectoire cartésienne** : Pour la trajectoire articulaire, recalculez la position cartésienne par MGD et mesurez l'écart par rapport à la ligne droite idéale.
3. **Sollicitation des actionneurs** : Comparez les vitesses et accélérations maximales sur chaque articulation.
4. **Passage par les singularités** : Lequel des deux types de trajectoire est le plus robuste face aux singularités ?

2.4.2. Synthèse

Remplissez un tableau comparatif :

Critère	Trajectoire articulaire	Trajectoire cartésienne
Simplicité d'implémentation		
Contrôle de la trajectoire de l'effecteur		
Gestion des singularités		

Respect des limites articulaires		
Charge de calcul		

2.5. Application : Suivi de contour (optionnel)

1. **Trajectoire complexe** : Générez une trajectoire cartésienne suivant une forme géométrique simple (cercle, carré, Lemniscate de Bernoulli).

Ecrire un programme permettant de former un cercle de rayon $r = 0.5$ centré en $(1, 0.8)$.

2. **Génération de la trajectoire articulaire** :

- Pour chaque point (x, y, ϕ) , construisez la matrice de transformation T et calculez q par MGI.
- Vérifiez la continuité de la solution (peut nécessiter de "dérouler" les angles pour éviter les sauts de 2π).

3. **Simulation** :

- Simulez le suivi de cette trajectoire en boucle ouverte (en appliquant les consignes articulaires calculées).
- Visualisez le résultat. L'effecteur suit-il exactement le cercle ? Pourquoi ?

4. Compte-rendu

Le compte-rendu devra contenir :

1. **Introduction** : Rappel des objectifs et du robot étudié.
2. **Méthodologie** :
 - Description des algorithmes de génération de trajectoire implémentés.
 - Explication du passage de l'espace cartésien à l'espace articulaire.
3. **Résultats expérimentaux** :
 - Graphiques comparatifs des différents profils (cubique vs quintique, trapézoïdal).
 - Visualisation des trajectoires articulaires et cartésiennes.
 - Analyse du comportement près des singularités.
 - Tableau comparatif des deux approches.
4. **Discussion** :
 - Avantages et inconvénients de chaque méthode.
 - Problèmes rencontrés (singularités, sauts d'angles, temps de calcul).
 - Critères de choix selon l'application.
5. **Conclusion** : Synthèse des compétences acquises.
6. **Annexes** : Codes MATLAB commentés.

Les fichiers source Matlab (.m) devront être joints au compte-rendu.