

Génération de trajectoires pour un robot manipulateur RRPR

I Objectif

Le but de ce bureau d'études est de développer une primitive de mouvement afin de faire réaliser à l'organe terminal d'un robot manipulateur une trajectoire rectiligne entre deux points de l'espace 3D à orientation imposée.

La trajectoire dans l'espace opérationnel se fait à vitesse imposée et les modèles d'un robot manipulateur permettent de calculer les commandes en position, vitesse et accélération à envoyer au robot pour générer le mouvement désiré.

La tâche est donc définie dans l'espace opérationnel et il est demandé de mettre en évidence le lien entre l'espace opérationnel et l'espace généralisé.

Travail à réaliser :

- calculer les lois de mouvement en temps,
- calculer les trajectoires $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$, $\ddot{\mathbf{q}}(t)$ (qui seront les consignes à envoyer à la commande).

La programmation des différentes fonctions sera effectuée en python.

Evaluation : chaque groupe doit rendre un rapport ainsi que le code à la fin du projet. Une évaluation partielle est faite à chaque séance et complétée ensuite par des tests après la livraison finale.

Temps estimé : 10 h en présence des encadrants (3 séances) et 8h-12h hors encadrement (loi de mouvement : 4h-6h, modélisation : 4h, intégration finale : 4h, rapport : 4h-6h).

Il est fortement conseillé de se répartir le travail au sein du groupe afin d'optimiser le temps de travail et de ne pas déborder du volume horaire indiqué.

II Cahier des charges

Le cahier des charges impose de développer une primitive de mouvement qui fait effectuer à l'O.T. une trajectoire rectiligne, à orientation imposée, entre deux points dans l'espace 3D.

Le profil de vitesse imposé dans l'espace opérationnel est un profil triangulaire comme indiqué sur la figure 1 avec $\dot{s}(t)$ vitesse de l'abscisse curviligne de l'O.T. et V la vitesse maximale imposée par l'utilisateur.

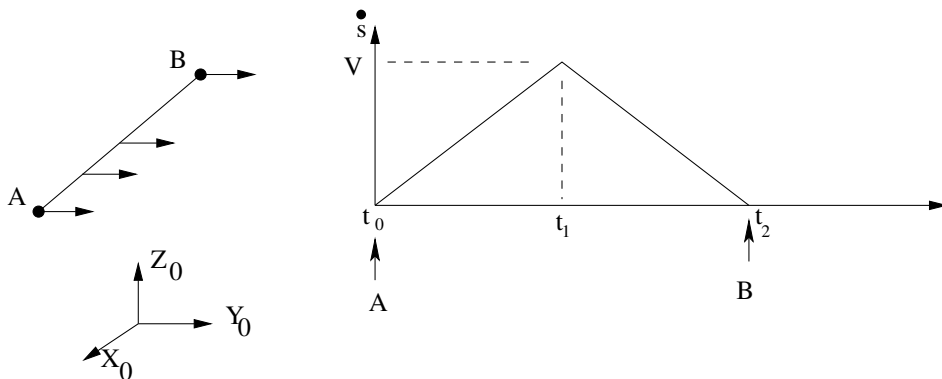


FIGURE 1 – Segment de droite (avec orientation de l'outil) et profil de vitesse associé

On note $\text{traj}(A, B, V)$ cette primitive qui doit retourner les vecteurs $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$ et $\ddot{\mathbf{q}}(t)$.

III Modélisation du robot

III.1 Présentation du robot

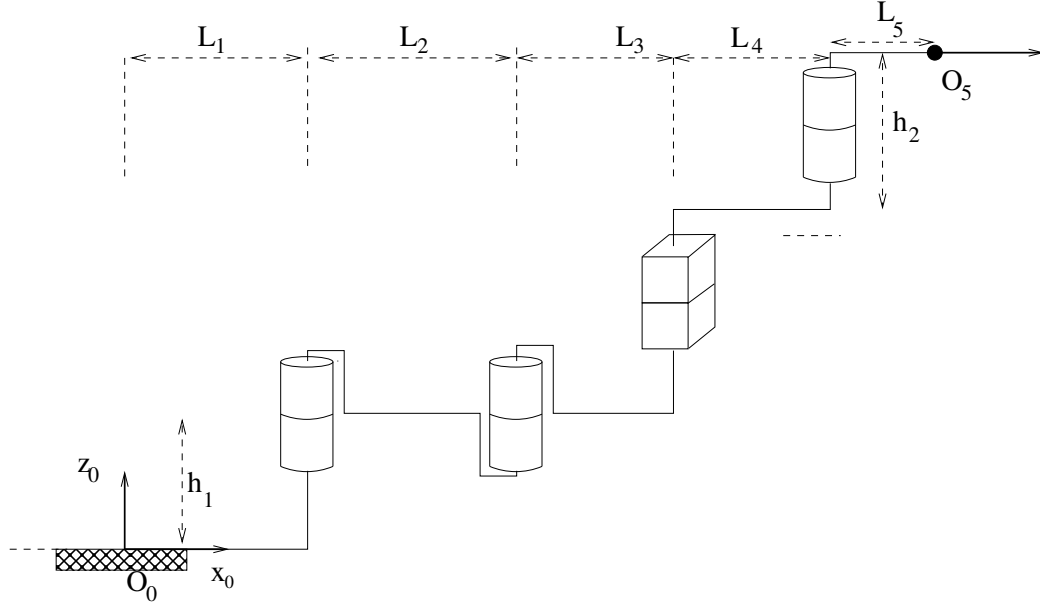


FIGURE 2 – Configuration figure du robot RRPR

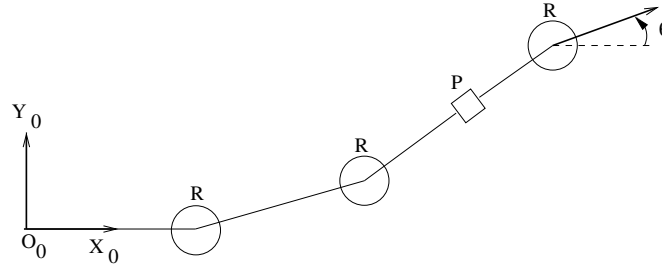


FIGURE 3 – Vue de dessus du robot pour une configuration quelconque

On considère un robot série *RRPR* possédant quatre liaisons lui permettant de déplacer son organe terminal dans l'espace 3D (voir figure 2).

La configuration du robot est définie dans l'espace généralisé par $\mathbf{q} = (q_1, q_2, q_3, q_4)^t$.

On ne considère pas de butée mécanique sur ce robot dans un premier temps, i.e., $\mathbf{q} \in \mathbb{R}^4$.

Les moteurs possèdent tous la même vitesse et accélération maximale (Kv, Ka).

La situation de l'O.T. est définie par la position du point O_5 et par l'orientation θ . O_5 est le point de référence de l'organe terminal par rapport au repère de base $R_0, (O_0, x_0, y_0, z_0)$, et θ et l'angle de l'orientation de l'O.T. par rapport à x_0 (voir figure 3).

Les distances L_i ($i = 1...5$) et h_j ($j = 1, 2$) sont supposées connues (constantes qui seront données en séances).

III.2 Modélisation

Dans le but de simuler une trajectoire dans l'espace généralisé à partir d'une trajectoire dans l'espace opérationnel, il est nécessaire de calculer les différents modèles inverses. On rappelle que :

Modèles géométriques	Modèles cinématiques	Modèles d'accélération
$\mathbf{X}(t) = F(\mathbf{q}(t))$	$\dot{\mathbf{X}} = J \dot{\mathbf{q}}$	$\ddot{\mathbf{X}} = J \ddot{\mathbf{q}} + \dot{J} \dot{\mathbf{q}}$
$\mathbf{q}(t) = F^{-1}(\mathbf{X}(t))$	$\dot{\mathbf{q}} = J^{-1} \dot{\mathbf{X}}$	$\ddot{\mathbf{q}} = J^{-1}(\ddot{\mathbf{X}} - \dot{J} \dot{\mathbf{q}})$

- Placer les repères.
- Déterminer les paramètres de DHM associés au placement des repères (figure 2).
- Calculer les modèles géométriques analytiques (MGD et MGI).
- Calculer les modèles cinématiques direct analytique.

IV Génération de mouvement

Rappel : Génération de mouvement

De manière générale, la génération de mouvement dans l'espace opérationnel s'effectue en deux étapes : tout d'abord, on définit la trajectoire géométrique désirée pour l'organe terminal du robot, $\mathbf{X}(s)$, avec s l'abscisse curviligne le long de la trajectoire. Puis on établit une loi d'évolution temporelle, $s(t)$, sur cette trajectoire, cette dernière permettant de caractériser le mouvement. On obtient $\mathbf{X}(s(t)) = \mathbf{X}(t)$.

IV.1 Loi de mouvement avec vitesse imposée

Déterminer les lois d'évolution $s(t)$, $\dot{s}(t)$, $\ddot{s}(t)$ de telle sorte que la trajectoire soit parcourue en respectant les contraintes

Programmation : sachant que vous devez envoyer les consignes au robot avec une période T_e (de l'ordre de 1 à 10 ms) vous devez calculer vos lois à chaque instants d'échantillonnage.

- Programmer le calcul de la loi de mouvement $s(t)$, $\dot{s}(t)$, $\ddot{s}(t)$.
- Afficher les courbes correspondantes.

IV.2 Trajectoire opérationnelle $\mathbf{X}(s)$ de type segments de droite

La trajectoire projetée est rectiligne entre les points (A, B) .

(x_i, y_i, z_i) représentent respectivement les coordonnées des points i dans le repère (O_0, x_0, y_0, z_0) .

- Calculer les coordonnées $(x(s), y(s), z(s))$ des points du segment de droite en fonction de s .
- Calculer les vitesses et les accélérations opérationnelles $(\dot{x}(s), \dot{y}(s), \dot{z}(s))$, $(\ddot{x}(s), \ddot{y}(s), \ddot{z}(s))$.
- Programmer le calcul de la trajectoire opérationnelle.
- Afficher la trajectoire opérationnelle.

IV.3 Génération de mouvement dans l'espace de la tâche $\mathbf{X}(t)$

- Connaissant $s(t)$, $\dot{s}(t)$, $\ddot{s}(t)$ et $x(s)$, $y(s)$, $z(s)$, $\dot{x}(s)$, $\dot{y}(s)$, $\dot{z}(s)$, $\ddot{x}(s)$, $\ddot{y}(s)$, $\ddot{z}(s)$ afficher $\mathbf{X}(t)$, $\dot{\mathbf{X}}(t)$, $\ddot{\mathbf{X}}(t)$.
- Calculer et afficher la vitesse du point O_5 .

IV.4 Génération de mouvement dans l'espace articulaire (généralisé)

- A l'aide des modèles inverses, calculer $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$, $\ddot{\mathbf{q}}(t)$ (entrées de la commande)
- Programmer votre fonction $traj(A, B, \theta, V)$ qui prend en entrée les coordonnées des points, l'orientation de l'outil et sa vitesse maximale et qui retourne $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$, $\ddot{\mathbf{q}}(t)$.
- Tester votre programme dans différents cas pour s'assurer de la robustesse de votre code.

IV.5 Méthodologie de travail

Vous allez calculer un mouvement de manière discrète, à chaque échantillon de temps, qui implique de mémoriser toutes vos variables importantes à chaque instant d'échantillonnage ($q_1(j), x(j), \dots$).

Commencer par calculer le temps final du mouvement pour en déduire le nombre de valeurs discrètes de vos variables entre $t = 0$ et $t = t_f$ ainsi que pour chaque intervalle de commutation.

Programmation

A l'aide des modèles inverses, programmer les différentes fonctions suivantes :

- la fonction $mg_i(\mathbf{\dot{X}}(t))$ qui retourne les valeurs de $\mathbf{q}(t)$ pour les deux (ou plus ?) solutions ($\mathbf{q}^+, \mathbf{q}^- \dots$),
- la fonction $mdi(\mathbf{\dot{X}}(t), \mathbf{q}(t))$ qui retourne les valeurs de $\mathbf{\dot{q}}(t)$,
- la fonction $mai(\mathbf{\ddot{X}}(t), \mathbf{q}(t), \mathbf{\dot{q}}(t))$ qui retourne les valeurs de $\mathbf{\ddot{q}}(t)$

Ces fonctions retournent les valeurs pour chaque instant d'échantillonnage le long de la trajectoire.

Visualisation

- Il est nécessaire d'afficher les résultats : $(s(t), \dot{s}(t), \ddot{s}(t))$, $(x(t), \dot{x}(t), \ddot{x}(t))$, ..., $(q_i, \dot{q}_i, \ddot{q}_i)$, ...

La fonction `affiche3courbes(numfig, nom, f, fd, fdd, t, tc)` (voir fichier `afficheCourbesTP.py`) affiche les trois courbes f, fd, fdd avec t contenant les $(n + 1)$ instants de calcul des courbes. La variable tc permet d'afficher les instants de commutation.

ATTENTION : Nous ne calculerons pas les modèles d'accélération $\mathbf{\ddot{X}}(t)$ et de $\mathbf{\ddot{q}}(t)$. Ce calcul serait nécessaire pour faire de la commande *Feedforward*, voir cours de commande.