

Simulation des systèmes de solides rigides polyarticulés

Examen 2020-2021

Le sujet est composé de deux parties distinctes. La première est un QCM qui compte pour 7 points dans la note finale et qui doit être répondu en 30 minutes à compter de son lancement. La seconde partie porte sur l'analyse du comportement dynamique d'un robot de pick-and-place type RPR.

La seconde partie fera l'objet d'un document réponse numérique à rendre avec l'ensemble des fichiers de simulation au format .zip.

Le barème est donné à titre indicatif et peut changer en fonction de la réussite générale de la promotion ^^

A. QCM (noté sur 13)

Le QCM, une fois lancé, doit être réalisé dans les 30 minutes qui suivent (il est automatiquement sauvé et envoyé à l'issue des 30 minutes). Un seul essai autorisé. Le QCM représente 7 points de la note de l'examen sur 20. **Attention, les mauvaises réponses peuvent entraîner des points négatifs.**

B. Robot RPR (noté sur 34)

L'objectif de cette partie est l'étude du comportement dynamique d'un robot. La partie représente 13 points de la note finale et doit faire l'objet d'un document réponse en plus des fichiers de simulation à compléter.

Le robot choisi pour cette étude est un robot de pick-and-place type RPR. Ce type d'architecture (rotoïde-prismatique-rotoïde) est peu courante mais peut présenter un intérêt dans des espaces encombrés du fait du bras télescopique central (axe 2 prismatique). La tâche de pick-and-place, comme son nom l'indique, est une tâche de prise dépose nécessitant de la précision, de la vitesse et de la répétabilité de la part du robot. Le paramétrage du robot a été réalisé selon les règles de Denavit-Hartenberg modifié :

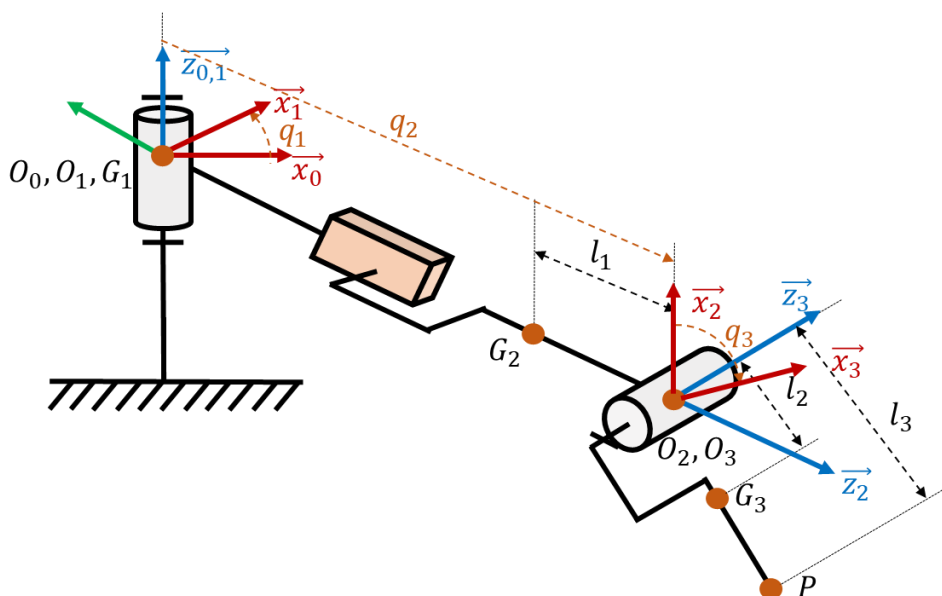


Figure 1 - Robot RPR et son paramétrage

Afin de minimiser les paramètres géométriques, l'origine du repère « monde » O_0 est confondue avec l'origine du repère du premier segment du robot O_1 . L'origine du repère associé au second segment O_2 est confondue avec l'origine du repère associé au troisième segment.

Le tableau des paramètres de Denavit-Hartenberg a été rempli de la manière suivante :

j	α_j	d_j	r_j	θ_j
1	0	0	0	q_1
2	$\pi/2$	0	q_2	$\pi/2$
3	$\pi/2$	0	0	q_3

Table 1 – Paramètres de Denavit-Hartenberg

Ces paramètres ont permis d'établir le modèle géométrique direct du robot en exploitant les matrices de transformation homogènes associées à chacun des repères de segments :

$${}^{i-1}T_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & d_i \\ \cos(\alpha_i)\sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i) & -r_i\sin(\alpha_i) \\ \sin(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\cos(\theta_i) & \cos(\alpha_i) & r_i\cos(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La pince du robot est positionnée sur le troisième segment au point P , défini tel que $\overrightarrow{O_3P} = -l_3\vec{y}_3$.

Les propriétés de masse et d'inertie ont été définies de la manière suivante :

Solide	Masse	Inertie au centre de masse et dans le repère associé au segment	Centre de masse
1	m_1	$I(S_1, G_1) = I_1 = \begin{bmatrix} I_{xx1} & 0 & 0 \\ 0 & I_{yy1} & 0 \\ 0 & 0 & I_{zz1} \end{bmatrix}_{(\vec{x}_1, \vec{y}_1, \vec{z}_1)}$	$\overrightarrow{O_1G_1} = \vec{0}$
2	m_2	$I(S_2, G_2) = I_2 = \begin{bmatrix} I_{xx2} & 0 & 0 \\ 0 & I_{yy2} & 0 \\ 0 & 0 & I_{zz2} \end{bmatrix}_{(\vec{x}_2, \vec{y}_2, \vec{z}_2)}$	$\overrightarrow{O_2G_2} = -l_1\vec{z}_2$
3	m_3	$I(S_3, G_3) = I_3 = \begin{bmatrix} I_{xx3} & 0 & 0 \\ 0 & I_{yy3} & 0 \\ 0 & 0 & I_{zz3} \end{bmatrix}_{(\vec{x}_3, \vec{y}_3, \vec{z}_3)}$	$\overrightarrow{O_3G_3} = -l_2\vec{y}_3$

Table 1 – Paramètres inertiels du robot

B.1. Obtention du modèle dynamique du robot (19 points)

Etapas préliminaires :

- dézippez le fichier dans lequel se trouve ce sujet ainsi que les fichiers de simulation Matlab
- Sauvez les fichiers dans vos documents
- Ouvrez Matlab
- Ajoutez l'ensemble des dossiers de travail de l'examen au path de matlab
- Ouvrez le fichier nommé « **analytique_RPR.mlx** »

Question 1 (2 pts) : Commentez le fichier que vous venez d'ouvrir. A quoi servent les lignes de code 1 à 35 ?

Le fichier analytique_RPR.mlx est un live script (script matlab mêlant codes et commentaires) qui contient un squelette de calcul formel pour l'obtention du modèle dynamique analytique du robot. En particulier, les premières lignes précisent les paramètres géométriques, de mouvement et inertiels du robot, et établissent le modèle géométrique direct du robot à l'aide de matrices de transformation homogène.

A partir de maintenant, les questions 2 à 10 sont à répondre directement dans le fichier.

Question 2 (2 pts) : Ecrire le code permettant d'obtenir les positions des centres de masse des solides 1, 2 et 3 dans le repère 0.

Question 3 (2 pts) : Ecrire le code permettant d'obtenir les vitesses des centres de masse des solides 1, 2 et 3 dans le repère 0.

Question 4 (3 pts) : Ecrire le code permettant de calculer l'énergie cinétique des solides 1, 2 et 3, puis l'énergie cinétique totale du robot.



Attention à bien projeter les vitesses angulaires de chacun des solides dans leurs repères locaux (dans lesquels sont exprimées les matrices d'inerties).

Question 5 (2 pts) : Ecrire le code permettant de calculer l'énergie potentielle des solides 1, 2, et 3, puis l'énergie potentielle totale du robot.



La pesanteur est dirigée selon $-\vec{z}_0$. L'énergie potentielle de pesanteur est considérée nulle dans le plan $(O_0, \vec{x}_0, \vec{y}_0)$.

Question 6 (1 pt) : Ecrire le code permettant d'obtenir la matrice de masse du robot.

Question 7 (1 pt) : Ecrire le code permettant d'obtenir la matrice des effets coriolis/centrifuges du robot.



Testez régulièrement votre code afin de vérifier qu'il compile.

Question 8 (1 pt) : Ecrire le code permettant d'obtenir le vecteur des actions de pesanteur.

Question 9 (2 pts) : Ecrire le code permettant d'obtenir le modèle dynamique inverse du robot.

Question 10 (2 pts) : Ecrire le code permettant d'obtenir le modèle dynamique direct du robot.



Ne réutilisez pas les efforts d'actionnement obtenus à la question 9 dans la question 10...

Question 11 (1 pt) : Quelle est la particularité de l'effort d'actionnement τ_2 , vis-à-vis de τ_1 et τ_3 ?

L'effort τ_2 est une force en N, contrairement à τ_1 et τ_3 qui sont des couples en N.m. Ces propriétés proviennent du paramétrage des liaisons (l'axe 2 du robot étant prismatique).

B.2. Mise en œuvre de la dynamique inverse et caractérisation de l'axe 2 (15 points)

Etapas préliminaires :

- Ouvrez le fichier nommé « **main_RPR.mlx** »
- Exécutez le fichier

Dans cette étape, on cherche à étudier une tâche de pick-and-place représentative des mouvements que devra effectuer le robot lors de sa mise en service. On va chercher en particulier à quantifier les besoins en efforts sur l'axe prismatique (axe 2).

Question 12 (3 pts) : Commentez le script que vous avez devant les yeux. Expliquez les étapes nécessaires à la définition de la tâche de pick-and-place, notamment comment la loi de mouvement articulaire à appliquer sur chacun des axes est obtenue.

Ce script est un live script permettant l'application des méthodes de dynamique inverse et directe sur le robot. On voit que le script permet d'abord de tracer le volume de travail du robot (parcours des amplitudes articulaires du robot). Ensuite, la trajectoire de pick and place est générée. Ici, deux points du volume de travail ont été choisis. Les configurations articulaires du robot correspondantes ont été obtenues par cinématique inverse (approche itérative à l'aide de la jacobienne du robot). Une fois obtenues les configurations articulaires, une génération de mouvement est réalisée : il s'agit de générer des trapèzes de vitesses sur l'ensemble des axes permettant de parcourir la distance articulaire entre les deux configurations. On remarque que les trapèzes de vitesse sont synchronisés sur l'axe le plus lent (phases d'accélération, vitesse constante, décélération synchronisées).

Question 13 (2 pts) : Créez une fonction de dynamique inverse prenant en entrée les positions, vitesses et accélérations articulaires du robot à un instant donné ainsi que ses caractéristiques géométriques et inertielles, et donnant en sortie les efforts d'actionnement appliqués sur les axes du robot.



L'entête de la fonction devrait ressembler à

function [Tau]=ID_RPR(t,q,dq,ddq,specs,g)

Question 14 (2 pts) : Mettre en œuvre la fonction créée à la question 13 sur la trajectoire définie dans le script principal. Tracer les efforts d'actionnement, en particulier τ_2 .

Afin de se rapprocher des conditions réelles de fonctionnement du robot, on décide de simuler la l'objet transporté pendant la tâche de pick-and-place. On considère que l'objet transporté peut être considéré comme une masse ponctuelle m_4 positionnée au point P . Cette masse est définie à 1kg dans les données numériques associées au modèle.

Question 15 (2 pts) : Expliciter la démarche permettant d'obtenir le modèle dynamique complété par cette masse ponctuelle.

Il suffit d'ajouter une masse ponctuelle au système, située en P, et considérer que la vitesse de ce point P est bien celle du point P appartenant au solide 3. On ne peut pas déclarer cette masse comme étant extérieure au système : sa présence va impacter l'inertie du robot, et donc les quantités dynamiques. Il faut donc recalculer les forces de pesanteur et l'énergie cinétique en tenant compte de cette masse.

Question 16 (2 pts) : Mettre en œuvre cette démarche et obtenir le nouvel effort d'actionnement τ_2 .

Question 17 (2 pts) : Réaliser le bilan des informations connues pour le choix de l'actionneur de l'axe 2.

On voit que la force max à fournir est de l'ordre de 65 N. on a également dans les caractéristiques une vitesse max de 1m/s (voir specs du robot) et une accélération max de 10m/s. A partir de ce moment, nous avons donc tous les éléments dimensionnants à l'exception des problématiques d'encombrement du système. Néanmoins les caractéristiques géométriques du robot donnent également des indications sur ce sujet. Au vu des accélérations et vitesses demandées, il est probable qu'un axe mécanique linéaire puisse être utilisé (liaison hélicoïdale par exemple).

Question 18 (2 pts): Quelles sont les limites de la démarche réalisée dans cette partie ? Quelles sont notamment les limites du modèle utilisé ?

La démarche est limitée par les imprécisions du modèle. On utilise des liaisons simplifiées (on ignore de nombreux constituants des liaisons), parfaites (pas de frottement sec ou visqueux). De surcroît, on ne modélise pas les rendements des actionneurs ainsi que leur dynamique propre.

B.3. Mise en œuvre de la dynamique directe et asservissement (BONUS)

Ces questions sont hors barème, néanmoins leur traitement vous permettra d'augmenter votre note en cas de réussite.

Question 19 : Créez une fonction de dynamique directe prenant en entrée les efforts d'actionnement et l'état courant du robot ainsi que ses caractéristiques géométriques et inertielles, et donnant en sortie l'accélération articulaire du robot à cet instant.

La méthode doit être créée en changeant de variable : on prend un état $[q, \dot{q}]$ et on estime sa dérivée à l'aide des équations du mouvement. C'est ce que l'on va donner à manger à un schéma temporel.

Une méthode de Runge-Kutta est fournie avec le code de l'examen.

Question 20 : Expliquer en quelques mots la nécessité de cette méthode pour simuler le comportement dynamique du robot en dynamique directe.

Le schéma d'intégration temporelle permet d'intégrer un système d'équations du premier ordre. En l'occurrence, la fonction obtenue à la question précédente est intégrable. La méthode choisie est une méthode de Runge-Kutta classique, à pas intermédiaire. Elle est plus précise qu'une méthode d'Euler, néanmoins le pas entre chaque intégration est fixe ce qui peut limiter son usage (obligation d'utiliser un pas global « petit » pour des systèmes ayant des dynamiques élevées).

Question 21 : Faire appel à cette fonction pour simuler le comportement du robot en dynamique directe, en appliquant les efforts obtenus lors de l'étape de dynamique inverse. Expliquer les différents ajustements nécessaires à cette simulation.

Afin de s'assurer de la bonne intégration, il est nécessaire de créer une variable « état » appelée ici `qd`, contenant du coup à la fois la position courante mais aussi la vitesse. L'intégration se fait pas à pas. Pour avoir une bonne estimation des couples appliqués, on rentre les couples calculés précédemment entièrement afin de pouvoir interpoler leur valeur sur les pas intermédiaires de Runge-Kutta (fonction `interp1`).

Question 22 : Réaliser un asservissement proportionnel sur l'ensemble des axes du robot et simuler le système en dynamique directe pour lui faire suivre la trajectoire de pick-and-place.

Dans la correction, l'asservissement proposé est un proportionnel dérivé. On peut tester l'influence des paramètres de correction en changeant les gains.