Séances 3 et 4
Réalisation expérimentale d'un robot sériel : génération de mouvements et commande.

1 Rappel du travail déjà effectué

A l'issue du TP1, vous savez normalement manipuler les modèles géométriques direct et inverse d'un modèle de robot sériel à 3 degrés de liberté, en utilisant le modèle géométrique direct et inverse.

A l'issue du TP2, vous savez normalement contrôler les actionneurs DYNAMIXEL constituant les articulations du robot sériel qui vous est fourni.

2 Exploitation expérimentale des modèles direct et inverse

Vous disposez maintenant de 2 séances pour coupler les 2 points évoqués précédemment, à savoir l'utilisation des modèles géométriques direct et inverse à un robot réel.

1. Réaliser un schéma, joint au compte rendu final, de la cinématique du robot réalisé. Nommer les angles de rotation, et identifier les paramètres géométriques nécessaires.
2. Utiliser la classe `Robot` et `Univers` pour déclarer le modèle de votre robot (comme lors du TP1). Afficher le bras ainsi déclaré à l'aide de la fonction `plot` et vérifier que le robot simulé représente bien le robot réel.
3. Utilisez la fonction `simulateRealRobot` pour introduire vos codes.
4. A l'aide de la fonction `MGD`, déterminer la pose (x, y, θ) de l'organe terminal pour 3 configurations articulaires q_1, q_2 et q_3 de votre choix. Appliquer ensuite ces configurations au robot réel à l'aide de la fonction `setPosition`, et vérifier que la pose atteinte corresponde bien à celle obtenue en simulation. On fera plus particulièrement attention :
   (a) à la conversion des configurations articulaires q_i en commandes motrices compréhensibles par les actionneurs (step $\leftrightarrow$ rad),
   (b) à la présence potentielle de butées mécaniques limitant le débattement angulaire des actionneurs (cf. TP2, fonction `new_pos = limit(pos,val_min,val_max)`),
   (c) à l'adaptation de ces commandes en fonction du modèle cinématique (réglage du $0°$ en particulier).
5. Commenter le mouvement du robot. Partant d'une position initiale quelconque, maîtrise-t'on sa trajectoire dans l'espace opérationnel ? Commenter.
6. Décider d'une pose (x, y, θ) cible dans l'espace opérationnel. A l'aide de la fonction `mod_geo_inv`, déterminer les configuration articulaires q permettant d'atteindre cette pose. Comme précédemment, on fera tout particulièrement attention à la présence de butées mécaniques limitant le débattement angulaire des actionneurs. Exploiter alors ces configurations articulaires pour en déduire les commandes motrices à appliquer au robot réel. Comparer la pose atteinte par son organe terminal avec la consigne. Commenter.
7. A l'aide d'un boucle pendant le mouvement, afficher à l'écran la pose en temps réel du robot et animer le modèle.

3 Génération de trajectoires

On souhaite maintenant faire en sorte que le robot soit capable de suivre une trajectoire, exprimée dans l'espace opérationnel en terme de positions et orientations que l'organe terminal du bras devra atteindre au cours du temps. En pratique, cette trajectoire sera obtenue par *interpolation* entre des configurations opérationnelles successives appartenant à la trajectoire désirée. La définition de la trajectoire peut par exemple permettre d'éviter un obstacle présent autour du robot, etc.

Dans ce projet, nous nous focaliserons sur une interpolation exprimée dans l'espace opérationnel uniquement : soit une cible opérationnelle (x, y, θ) et une pose initiale (x^0, y^0, θ^0) de l'organe terminal. On peut alors déterminer par *interpolation linéaire* un certain nombre de configurations intermédiaires partant de (x^0, y^0, θ^0) et menant en (x, y, θ), et pour chacune d'entre elles en déduire les ordres moteurs q_i à appliquer.

L'objectif est donc réaliser des trajectoires par interpolation linéaire entre deux configurations opérationnelles. Pour cela :

— en choisissant judicieusement une configuration de départ et d'arrivée qui peuvent être connectées en ligne droite en respectant les limitations en rotation des articulation, calculer n configurations intermédiaires dans l'espace opérationnel ;
— pour chaque configuration intermédiaire, retrouver à l'aide de la fonction `MGI`, les valeurs des articulations q_i qui y conduisent ;
— proposer une (voir plusieurs) critères qui permettent de choisir une des deux solutions du modèle inverse ;
— testez votre programme sur le modèle du robot pour le valider. Vérifier notamment le respect des butées des articulations ;
— implémentez sur le robot réel en gardant l'affichage simultané du mouvement à l'écran. Vous pouvez aussi placer un stylo à l'extrémité du robot pour dessiner sur une feuille ;
— Expérimenter avec des nombres de points intermédiaires n différents. Commenter.

(Bonus) Dessinez des formes plus complexes. (Par exemple les lettres L, N, A... ou une étoile à 5 branches.)
(Bonus) Comment faire des trajectoires courbes ? Dessinez par exemple un cercle.

BONUS (grosse récompense !)

Pour éviter les saccades observées dans l'implémentation précédente, on désire utiliser la commande en vitesse à la place de l'interpolation opérationnelles. Pour simplifier la tâche dans la suite, fixer la troisième articulation à $\theta_3 = 0$ pour traiter le robot comme un 2R. Dans ce cas, les 2 degrés de libertés sont les coordonnées x et y, à commander à travers les moteurs q_1 et q_2.

— Écrire le jacobien J du robot 2R (*cf.* TD1-2). En déduire deux fonctions pour les modèles cinématiques directe et inverse, qui permettent de calculer les relations entre les vitesses articulaires $\dot{q}$ et $(\dot{x}, \dot{y})$.
— Choisir deux configurations de départ et d'arrivée (P_0, P_1) comme précédemment. Écrire un script qui permettra d'atteindre P_1 comme suit :

1. En utilisant le modèle géométrique inverse, calculer les valeurs aux articulation q_i qui correspondent à P_1. A partir de la configuration P_0, envoyez ces valeurs comme objectif à chaque articulation pour démarrer le mouvement vers P_1.
2. Écrire une boucle pour calculer pendant le mouvement, à chaque pas, le vecteur $\vec{v}$ qui pointe vers P_1 depuis la position courante.
3. A chaque pas, utiliser le modèle cinématique inverse J^{-1} pour obtenir les vitesses $\dot{q}_i$ à chaque articulation pour que le mouvement du robot soit colinéaire à $k.\vec{v}$, avec k un scalaire à choisir. Imposez ces vitesses $\dot{q}_i$ aux moteurs par la commande `setSpeed()`.

— Comparez le comportement avec la méthode d'interpolation. Tracez la trajectoire, ainsi que les vitesses $\dot{q}_i$, $(\dot{x}, \dot{y})$ au cours du mouvement. Discuter de l'influence du choix de k.

Instructions pour le Rapport

Vous devez rendre à l'issue des séances de TP de cette UE un rapport de projet comportant au minimum (liste non-exhaustive !) :

— l'ensemble des réponses aux questions clairement identifiées dans l'ensemble des sujets (texte en gras ou mis en évidence). *Identifier clairement les questions auxquelles vous répondez !*
  — pour les questions nécessitant l'écriture d'une fonction ou d'un script Matlab, vous devez fournir dans le rapport le code **commenté** correspondant à la fonction ou au script demandé, ni plus ni moins ;
  — pour les questions demandant l'affichage du robot simulé ou d'une figure, insérer dans votre rapport une capture d'écran de la figure obtenue.

  RAPPEL : une courbe sans titre, sans légende, sans unité, et sans commentaire ne présente évidemment aucun intérêt et ne sera pas considérée.
— pour la partie 3, en plus des réponses au problème posé, on attend également un tracé des trajectoires obtenues, ainsi qu'une comparaison entre les trajectoires désirées et celles mesurées sur le robot réel ;
— enfin, fournir une vidéo (de 1 minute 30s maximum) **commentée** montrant l'ensemble de votre réalisation. On pourra par exemple y retrouver une compilation des expériences, simulations et tout autre élément jugé pertinent pour votre évaluation.

L'ensemble devra être rendu sous la forme d'un rapport au format **AU FORMAT PDF**, d'un fichier `.zip` contenant vos codes commentés, et d'une vidéo (attention à la taille de celle-ci, elle doit être raisonnable ... pas besoin d'une vidéo UHD4K !). Le tout devra être déposé sur Moodle avant la date limite qui vous sera communiquée en

séance. Au delà, le site de dépôt sera automatiquement fermé et aucun rendu ne sera possible. Aucun délais supplémentaire ne sera accordé ; faute de dépôt dans les temps, une note de 0 sera donnée aux binômes n'ayant pas rendu leur rapport.

Toute forme de triche : plagiat, recopie partielle ou complète du travail d'un autre binôme, du travail remis les années passées, etc. seront sévèrement sanctionnés, et les étudiants concernés seront envoyés en conseil de discipline.