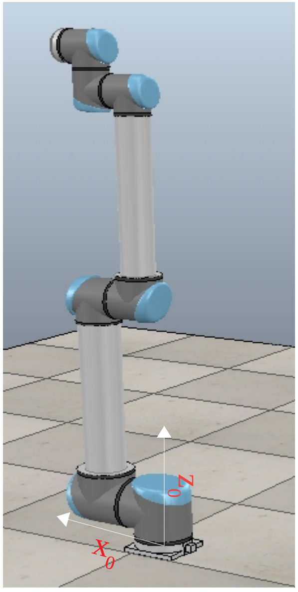
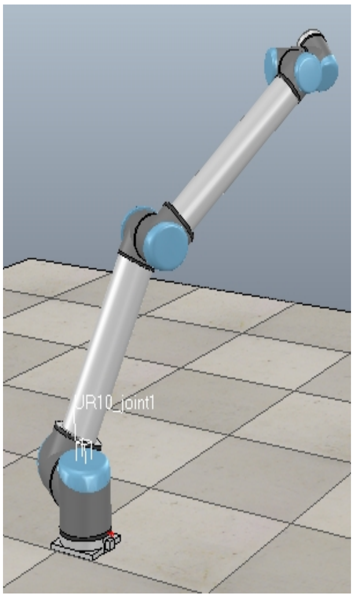
On considère un robot ur10 équipé d'une ventouse, tel que représenté ci-dessous, ainsi que son tableau de Denavit-Hartenberg (l'origine O_0 est à l'intersection du sol et de l'axe z_0 de l'articulation de base)

Robot au repos, robot en action (6 degrés de liberté)





- Tableau de Denavit-Hartenberg (d6 est choisi pour que O_6 soit l'extrémité de la ventouse)

	θ_i	a_i	d_i	α_i
0->1	$q_1=\theta_1^*$	0	d1=0.128m	90°
1->2	$q_2=\theta_2^*$	a2=0.612m	0	0
2->3	$q_3=\theta_3^*$	a3=0.5722m	0	0
3->4	$q_4=\theta_4^*$	0	d4=0.1639m	90°
4->5	$q_5=\theta_5^*$	0	d5=0.1157m	-90°
5->6	$q_6=\theta_6^*$	0	d6=0.1047m	0

1. Partie calcul symbolique

Ouvrir le Fichier *ur10SymbolicEtudiant.m*, et le compléter en programmant les réponses aux questions ci-après

1.1. Modèle Géométrique Direct (MGD)

Les matrices ${}^0T_1$ à ${}^5T_6$ du modèle géométrique direct sont déjà calculées dans la partie 1.1 du programme.

☐ Compléter cette partie en calculant les matrices ${}^0T_2, \cdots, {}^0T_6$ et leurs inverses

La configuration *Robot au repos* correspond aux valeurs angulaires $\{q_1=90°, q_2=90°, q_3=0°, q_4=90°, q_5=0°, q_6=0°\}$.

☐ Analyser les matrices ${}^0T_1, \cdots, {}^0T_6$ correspondantes, et en déduire une représentation graphique des axes x_i, z_i et origines o_i des repères 0 à 6 dans ce cas. Si vous êtes plus doués que le prof en dessin technique, ce qui ne devrait pas poser de problèmes, vous pouvez également représenter les constantes :d1 a2 a3 d4 d5 d6.

1.2. Modèle cinématique direct,(MCD) calcul des jacobiennes en position

☐ Déterminer les jacobiennes en position ${}^0J_{P1}, \cdots, {}^0J_{P6}$, de points *P1... P6* , respectivement liés aux repères

1 à 6. et générer les fonctions matlab *clcJ0P1* , *clcJ0P6* correspondantes.

1.3. Modèle cinématique direct,(MCD) calcul des jacobiennes en orientation

☐ Déterminer les jacobiennes en orientation $^{0}J_{u1},\cdots,{}^{0}J_{u6}$ des repères 1...6 par rapport au repère 0, de 2 façons différentes

1- Directement depuis $^{0,}T_{1}\cdots,{}^{0,}T_{6}$,en n'émettant aucune hypothèse sur la façon dont on a obtenu cette matrice depuis les degrés de liberté q_i (on générera également la fct Matlab correspondante *clcJ0ui*)

2- En employant le fait que l'on a employé le formalisme de Denavit-Hartenberg pour décrire les changements de repère.(on générera la fct Matlab correspondante *clcJ0ui_dh*)

2. partie numérique : Modèle cinématique direct (MCD)

2.1. Vérification numérique du MCD

☐ Ouvrir le fichier *ur10NumericEtudiant.m*, et vérifier numériquement la correspondance entre les 2 expressions des jacobiennes en orientation , pour la configuration proposée du robot

2.2. MCD en position(facile)

☐ Vérifier que pour de petites variations articulaires *dq*, les petites variations des coordonnées de l'extrémité du robot dans le repère 0 correspondent bien à ce qui est prédit par la matrice Jacobienne.

2.3. MCD en orientation(pas facile)

Pourquoi est-ce plus difficile qu'avec les positions ?

Parce que, contrairement à ce que l'on pourrait croire : *la rotation de vecteur angle-axe* $^{0}\delta u$ <u>*ne permet pas*</u> *de passer de l'orientation* $^{0}u_{6init}$ *à l'orientation* $^{0}u_{6final}={}^{0}u_{6init}+{}^{0}\delta u$ …

> Détermination de $^{0}u_{6final}=f\left({}^{0}u_{6init},{}^{0}\delta u\right)\neq{}^{0}u_{6init}+{}^{0}\delta u$.
>
> Soient
>
> $^{0}\delta u$: le vecteur angle axe correspondant à la variation δq des positions articulaires.
>
> ^{0}R : la matrice de rotation correspondante (<u>exprimée dans le repère 0</u>, tout comme $^{0}\delta u$)
>
> $^{0}u_{6final},{}^{0}u_{6init}$:les orientations du repère 6 par rapport au repère 0, après et avant avoir appliqué la rotation ^{0}R
>
> on a alors : $^{0}R_{6final}={}^{0}R\,.{}^{0}R_{6init}$ (^{0}R est appliquée<u> à gauche,</u> car elle est exprimée dans le repère 0)
>
> le vecteur angle axe $^{0}u_{6final}$ peut alors être déduit de l'expression de $^{0}R_{6final}$
>
> { pour les formules permettant de passer de *R* à *u* et réciproquement : fonctions *get_u*, *get_rot_u* , voir Tr Homogene 4.2 slide 38 }
>
> *(Une méthode alternative correspond à utiliser les quaternions d'orientation ...)*

☐ Vérifier <u>en appliquant cette méthode</u> que pour de petites variations articulaires *dq*, les petites variations des l'orientation de l'extrémité du robot dans le repère 0 correspondent bien à ce qui est prédit par la matrice Jacobienne.

3. Partie numérique :Modèle cinématique inverse (MCI)

Le principe de génération de trajectoires articulaires *q*, depuis les jacobiennes *J*(*q*) et les trajectoires opérationnelles, consiste à effectuer une série de petits déplacements pour lesquels on emploie les approximations du premier ordre suivantes :

$$\underbrace{{}^0O_6(q+\delta q)}_{\text{point visé}}-\underbrace{{}^0O_{06}(q)}_{\text{point actuel}}\approx\underbrace{J_{0O6}(q)}_{\text{jacobienne connue}}\ .\ \underbrace{\delta_q}_{\text{à déterminer}}$$

$${}^0\delta u_6(\delta q)\approx\underbrace{J_{0U6}(q)}_{\text{jacobienne connue}}\ .\ \underbrace{\delta_q}_{\text{à déterminer}}$$

l’objectif est de déterminer une succession de petits déplacement articulaires δq_k permettant de parcourir le chemin entre les points de départ et d’arrivée : $P_{dep}=\begin{bmatrix}{}^0O_{6dep}\\{}^0U_{6dep}\end{bmatrix}$, $P_{arr}=\begin{bmatrix}{}^0O_{6arr}\\{}^0U_{6arr}\end{bmatrix}$

3.1. petits déplacements en position

□ En employant l’approximation du premier ordre correspondante, déterminer δ_q pour que l’extrémité du robot se déplace de +0,01m colinéairement à $\vec{x}_0$, partant de sa configuration initiale

□ Calculer la norme de l’erreur entre les coordonnées obtenues et désirées

3.2. petits déplacements en orientation

□ En employant l’approximation du premier ordre correspondante, déterminer δ_q pour que le repère lié à l’extrémité du robot tourne de +0,01rad , autour de $\vec{x}_0$, partant de sa configuration initiale

□ Calculer la norme de l’erreur angulaire entre les orientations obtenue et désirée

3.3. petits déplacements simultanés

□ Déterminer δ_q pour que simultanément, partant de la configuration initiale, l’extrémité du robot se déplace de +0,01m colinéairement à $\vec{x}_0$ et tourne de +0,01rad autour de $\vec{x}_0$

□ (optionnel, si on a le temps après la partie 4) proposer une méthode itérative telle que finalement on obtienne une erreur inférieure à 1 micro mètre sur la position finale, et à 1 micro radian sur l’orientation finale.

4. Partie numérique : modèle de force statique

Le Modèle de force statique permet d’établir le lien entre des forces ou moments exercées par le robot sur l’environnement, ou vice-versa. Le principe est relativement simple si l’on se reporte à la notion de travail.

Pour une force F_i appliquée en un point P_k , fixe dans le repère , le travail s’écrit de 2 façons :

$$\underbrace{{}^0F_i^T.d\,{}^0P_k}_{\text{travail élémentaire Force.déplacement}}=\underbrace{Fq^T.dq}_{\text{travail élémentaire = Force articlaire .déplacement articulaire}}$$

Pour un moment Γ_i appliqué à un solide lié au repère R_k , de vecteur angle axe de coordonnées ${}^0U_k$:

$$\underbrace{{}^0\Gamma_i^T.d\,{}^0U_{Rk}}_{\text{travail élémentaire Force.déplacement}}=\underbrace{Fq^T.dq}_{\text{travail élémentaire = Force articlaire .déplacement articulaire}}$$

de plus, comme on a, pour tout q:

$$d\,{}^0P_k=\underbrace{{}^0J_{Pk}}_{\text{jacobienne en position}}.dq\ \text{, et}\ d\,{}^0U_k=\underbrace{{}^0J_{Uk}}_{\text{jacobienne en orientation}}.dq$$

On obtient donc finalement le lien entre les forces (moments) articulaires, et forces/ moments opérationnels

$$\underbrace{Fq^T}_{\text{Force articulaire due à } F_i}\cancel{dq}={}^0F_i^T.\underbrace{{}^0J_{Pk}}_{\text{jacobienne en position au point P\_k}}.\cancel{dq}\Leftrightarrow Fq=\left[{}^0J_{Pk}\right]^T.{}^0F_i$$

$$\underbrace{Fq^T}_{\text{Force articulaire due à } \Gamma_i} \cancel{dq} = {}^0\Gamma_i^T . \underbrace{{}^0J_{uk}}_{\text{jacobienne en orientation du repère U\_k}} . \cancel{dq} \Leftrightarrow Fq = \left[{}^0J_{uk}\right]^T . {}^0\Gamma_i$$

4.1. forces sur un des bras de robots

Choisir quelques configurations permettant de mettre en évidence les concepts suivants

□ Couples articulaires minimum et maximum du robot pour compenser la force de gravité (masse de 10Kg à l'extrémité)

□ Seules les actions (moments et forces, pas forcément appliqués sur l'extrémité) compatibles avec les déplacements du robot produisent des couples articulaires.

5. Travail personnel optionnel

Ouvrir avec V-REP le fichier *ur10Etudiant.tt* et analyser rapidement le script *UR10.*

L'objectif est de calculer les petits déplacement que le robot ur10 attrape la boîte rouge et la pose dans la caisse verte, avant de revenir à sa position de repos.

Le fichier matlab *ur10TrajectoireCinematiqueEtudiant.m* génére automatiquement le texte à intégrer dans le script *UR10, correspondant à la trajectoire générée.*

éditer et exécuter ce fichier avec Matlab (noter que par défaut il n'y a pas de déplacement)

□ *A*dapter le tableau *pts* des points à atteindre (ligne 35…), en relevant les coordonnées et dimensions de la boîte et de la caisse sous V-REP

□ Dans la boucle de suivi de trajectoire, proposer un algorithme de calcul des déplacements articulaires *dq*, pour chacun des points à atteindre.

□ intégrer la trajectoire générée sous V-REP (dans le script *UR10* du fichier), jusqu'à ce que vous la jugiez satisfaisante.

Remarque : Il y a une quantité importante de ‘’petits détails’’ entre la version naïve de suivi de trajectoire en 3.3 et la version finale fonctionnelle: (gestion des points singuliers, limitation de la norme des déplacements élémentaires, points non faisables à cause de collisions…), que vous êtes invités à gérer seuls...