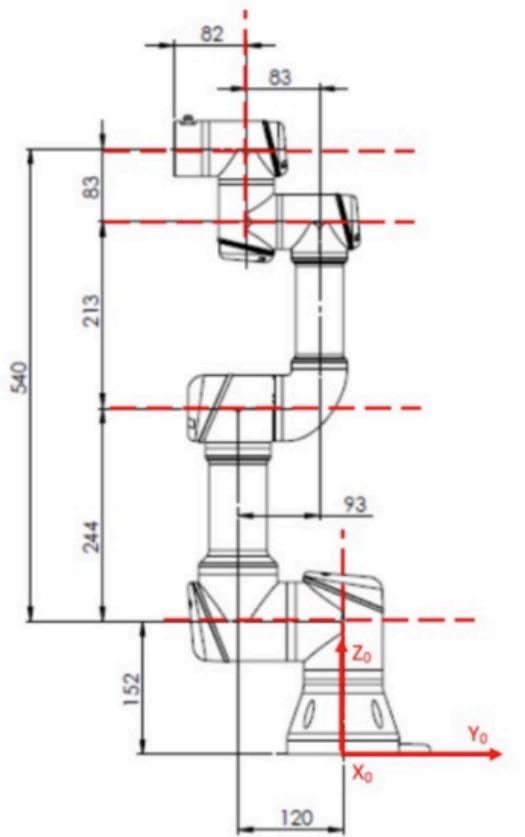


Rapport BE UR3

Github du projet : https://github.com/Alecbossard/robot_UR3.git

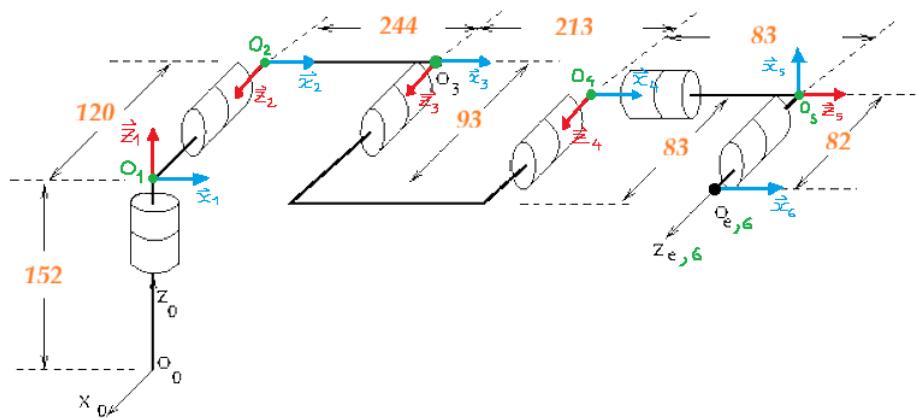
Modélisation UR3

Dimensions du robot :



Calcul du Modèle Géométrique Direct (MGD) du UR3

Placement des repères et paramètres de DHM



Afin d'établir le modèle cinématique, un repère O_i est associé à chaque articulation du manipulateur. Le placement des axes suit la convention de Denavit–Hartenberg modifiée (DHM) :

- L'axe z_i est colinéaire à l'axe de rotation de l'articulation i
- L'axe x_{i-1} est choisi selon la normale commune aux axes z_{i-1} et z_i
- L'axe y_i complète le trièdre direct afin de garantir l'orientation droite du repère.

Ce choix garantit une paramétrisation systématique et directement exploitable pour construire les transformations homogènes entre repères successifs.

Paramètres DH

À partir du placement des repères, les paramètres DHM suivants sont identifiés pour chaque articulation i : a_{i-1} (décalage selon x_{i-1}) α_{i-1} (rotation autour de x_{i-1}), r_i (décalage selon z_i) et θ_i (angle articulaire variable q_i). Les paramètres retenus sont :

	1	2	3	4	5	6
σ_i	0	0	0	0	0	0
a_{i-1}	0	0	a_2	a_3	0	0
α_{i-1}	0	$\pi/2$	0	0	$\pi/2$	$-\pi/2$
r_i	r_1	$-r_2$	0	r_4	r_5	r_6
Θ_i	q_1	q_2	q_3	q_4	q_5	q_6
q_{ifig}	$\pi/2$	0	0	$\pi/2$	0	$-\pi/2$

La ligne q_{ifig} correspond aux décalages permettant de relier la configuration de référence (figure) au repérage articulaire du modèle.

$$r1 = 0.152$$

$$r2 = 0.120$$

$$a2 = 0.244$$

$$a3 = 0.213$$

$$r4 = 0.010$$

$$r5 = 0.083$$

$$r6 = 0.082$$

Matrices de passage élémentaires

Pour chaque articulation, la transformation homogène élémentaire $T_{i-1} T_i$ est construite à partir des paramètres DHM. Elle traduit le passage du repère $i-1$ au repère i et dépend de la variable articulaire q_i , ainsi que des constantes géométriques.

$$T_{01} = \begin{bmatrix} -\sin(q_1) & -\cos(q_1) & 0 & 0 \\ \cos(q_1) & -\sin(q_1) & 0 & 0 \\ 0 & 0 & 1 & r_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{12} = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & 0 \\ 0 & 0 & -1 & r_2 \\ \sin(q_2) & \cos(q_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{23} = \begin{bmatrix} \cos(q_3) & -\sin(q_3) & 0 & a_2 \\ \sin(q_3) & \cos(q_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{34} = \begin{bmatrix} -\sin(q_4) & -\cos(q_4) & 0 & a_3 \\ \cos(q_4) & -\sin(q_4) & 0 & 0 \\ 0 & 0 & 1 & r_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{45} = \begin{bmatrix} \cos(q_5) & -\sin(q_5) & 0 & 0 \\ 0 & 0 & -1 & -r_5 \\ \sin(q_5) & \cos(q_5) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{56} = \begin{bmatrix} \sin(q_6) & \cos(q_6) & 0 & 0 \\ 0 & 0 & 1 & r_6 \\ \cos(q_6) & -\sin(q_6) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{0,1} = \begin{bmatrix} -0.09983342 & -0.99500417 & 0 & 0 \\ 0.99500417 & -0.09983342 & 0 & 0 \\ 0 & 0 & 1 & 0.152 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{0,2} = \begin{bmatrix} -0.0876120655 & -0.0478626895 & 0.995004165 & -0.119400500 \\ 0.873198304 & 0.477030408 & 0.0998334166 & -0.0119800100 \\ -0.479425539 & 0.877582562 & 6.12323400 \times 10^{-17} & 0.152000000 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{0,3} = \begin{bmatrix} -0.0953745058 & 0.0295027919 & 0.995004165 & -0.140777844 \\ 0.950563786 & -0.294043837 & 0.0998334166 & 0.201080376 \\ 0.295520207 & 0.955336489 & 6.12323400 \times 10^{-17} & 0.0350201686 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{0,4} = \begin{bmatrix} 0.00996671108 & 0.0993346654 & 0.995004165 & -0.151142572 \\ -0.0993346654 & -0.990033289 & 0.0998334166 & 0.404548797 \\ 0.995004165 & -0.0998334166 & 6.12323400 \times 10^{-17} & 0.0979659726 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{0,5} = \begin{bmatrix} 0.48577702 & 0.86842001 & -0.09933467 & -0.15938735 \\ -0.03931168 & 0.13523564 & 0.99003329 & 0.48672156 \\ 0.87319830 & -0.47703041 & 0.09983342 & 0.10625215 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{0,6} = \begin{bmatrix} -0.05034163 & 0.49326708 & 0.86842001 & -0.08817691 \\ 0.98116263 & -0.13795369 & 0.13523564 & 0.49781088 \\ 0.18650904 & 0.85886924 & -0.47703041 & 0.06713565 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Correspondance angles modèle et angles robot

Les angles articulaires calculés par le modèle géométrique direct (conventions DH) ne correspondent pas directement aux angles utilisés par le robot simulé sous PyBullet, en raison de différences de conventions (sens de rotation des axes et choix des zéros articulaires dans l'URDF). Une correction est donc appliquée articulation par articulation sous la forme :

$$q_{\text{sim}} = S q_{\text{mgd}} + o$$

avec inversion de signe sur les articulations 2 et 3, et offsets $-\pi/2$ sur l'articulation 4 et π sur l'articulation 5. L'inverse $q_{\text{mgd}} = S (q_{\text{sim}} - o)$ permet de repasser de la simulation au modèle.

Notre MGD vers la simulation :

Signes : $S = \text{diag}(1, -1, -1, 1, 1, 1)$

$$\text{Offsets : } o = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{\pi}{2} \\ \pi \\ 0 \end{bmatrix}$$

Simulation vers MGD :

$$S = \text{diag}(1, -1, -1, 1, 1, 1)$$

$$o = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{\pi}{2} \\ \pi \\ 0 \end{bmatrix}$$

Comparaison de votre MGD et du MGD calculé par PyBullet

Configuration de base : $q = [0, 0, 0, 0, 0, 0]$

Théorique (Notre MGD):

$$T_{06} = \begin{bmatrix} 0 & 0 & 1 & -0.03 \\ 1 & 0 & 0 & 0.54 \\ 0 & 1 & 0 & 0.15 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

PyBullet :

$$T_0^6 = \begin{bmatrix} 0 & 1 & 0 & 0.457 \\ 1 & 0 & 0 & 0.192 \\ 0 & 0 & -1 & 0.069 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Les matrices de MGD issues du modèle DH et de PyBullet ne coïncident pas directement car elles sont exprimées dans des repères différents (base et repère effecteur). Après conversion des angles (signes + offsets), les poses deviennent cohérentes.

Calcul du Modèle Différentiel Direct (MDD) avec explication de vos choix.

Après avoir calculé avec les formules du cours et après vérification grâce à un code python, on trouve la jacobienne géométrique suivante :

$$J = \begin{bmatrix} -4.97810882 \times 10^{-1} & -8.47229774 \times 10^{-3} & 3.20619851 \times 10^{-3} & -3.07789617 \times 10^{-3} & -3.98337156 \times 10^{-2} & 0 \\ -8.81769084 \times 10^{-2} & 8.44403790 \times 10^{-2} & -3.19550405 \times 10^{-2} & 3.06762967 \times 10^{-2} & 3.22355781 \times 10^{-3} & 0 \\ 0 & 5.04126904 \times 10^{-1} & 2.89996758 \times 10^{-1} & 8.65100863 \times 10^{-2} & -7.16022610 \times 10^{-2} & 0 \\ 0 & 9.95004165 \times 10^{-1} & 9.95004165 \times 10^{-1} & 9.95004165 \times 10^{-1} & -9.93346654 \times 10^{-2} & 8.68420009 \times 10^{-1} \\ 0 & 9.98334166 \times 10^{-2} & 9.98334166 \times 10^{-2} & 9.98334166 \times 10^{-2} & 9.90033289 \times 10^{-1} & 1.35235641 \times 10^{-1} \\ 1 & 6.12323400 \times 10^{-17} & 6.12323400 \times 10^{-17} & 6.12323400 \times 10^{-17} & 9.98334166 \times 10^{-2} & -4.77030408 \times 10^{-1} \end{bmatrix}$$

Le Modèle Différentiel Direct relie les vitesses articulaires \dot{q} à la vitesse de l'outil dans l'espace opérationnel :

$$\dot{X} = [\dot{p} \quad \omega]^T = J(q), \dot{q}$$

où p est la position de l'effecteur, ω sa vitesse angulaire, et $J(q)$ la jacobienne du robot.

Nous avons choisi d'utiliser la jacobienne géométrique car elle s'obtient directement à partir des repères du MGD et fournit directement \dot{p} et ω .

À partir des transformations homogènes ${}^0T_i(q)$, on récupère pour chaque articulation i l'axe z_{i-1} (exprimé dans la base) et l'origine ${}^0o_{i-1}$. Pour une articulation rotative, la i -ème colonne de la jacobienne géométrique s'écrit :

$$J_v^{(i)} = z_{i-1} \times (o_6 - o_{i-1})$$

$$J_\omega^{(i)} = z_{i-1}$$

On obtient alors :

$$J(q) = [J_v \quad J_\omega]$$

Dans notre utilisation, seule la partie linéaire J_v est nécessaire pour calculer la vitesse cartésienne de l'outil :

$$\dot{p}(t) = J_v(q(t)), \dot{q}(t)$$

et comparer la vitesse obtenue en simulation à la consigne $\dot{p}_{ref}(t)$.

2. Génération de mouvement pour le robot UR3

2.1. Objectif et notations

On veut générer une trajectoire circulaire dans le plan (X, Z) , de centre $O = (C_x, C_y, C_z)$, de rayon R , parcourue à une vitesse tangentielle de consigne V .

On paramètre le mouvement par l'abscisse curviligne s (distance le long du cercle) :

- $s(t)$: distance parcourue (m)
- $\dot{s}(t)$: vitesse le long de l'arc (m/s)
- $\ddot{s}(t)$: accélération le long de l'arc (m/s^2)

Le mouvement est discrétisé avec un pas $dt = 0,005s$.

2.2. Profil temporel (3 phases) et temps de commutation

Le tour complet vaut $2\pi R$. Le code utilise 3 phases (accélération, vitesse constante, décélération) sur des portions : quart de tour, quart de tour, demi-tour.

Phase 1 : accélération (quart de tour)

$$\text{Distance : } d_{\text{acc}} = \frac{\pi R}{2}$$

Accélération constante choisie :

$$a = \frac{V^2}{2d_{\text{acc}}} = \frac{V^2}{\pi R}$$

Temps de commutation :

$$t_1 = \frac{V}{a} = \frac{\pi R}{V}$$

Phase 2 : vitesse constante (quart de tour)

$$\text{Distance : } d_{\text{cst}} = \frac{\pi R}{2}$$

Durée :

$$\Delta t_{\text{cst}} = \frac{d_{\text{cst}}}{V} = \frac{\pi R}{2V}$$

Deuxième commutation :

$$t_2 = t_1 + \Delta t_{\text{cst}} = \frac{3\pi R}{2V}$$

Phase 3 : décélération (demi-tour)

Distance : $d_{\text{dec}} = \pi R$

Décélération (valeur positive) :

$$d = \frac{V^2}{2d_{\text{dec}}} = \frac{V^2}{2\pi R}$$

Durée :

$$\Delta t_{\text{dec}} = \frac{V}{d} = \frac{2\pi R}{V}$$

Temps final :

$$t_f = t_2 + \Delta t_{\text{dec}}$$

Vérification : $d_{\text{acc}} + d_{\text{cst}} + d_{\text{dec}} = 2\pi R$ (un tour complet).

2.3. Lois $s(t)$, $\dot{s}(t)$, $\ddot{s}(t)$ (par morceaux)

Phase 1 : $0 \leq t \leq t_1$

$$\ddot{s}(t) = a$$

$$\dot{s}(t) = at$$

$$s(t) = \frac{1}{2}at^2$$

Phase 2 : $t_1 < t \leq t_2$

$$\ddot{s}(t) = 0$$

$$\dot{s}(t) = V$$

$$s(t) = d_{\text{acc}} + V(t - t_1)$$

Phase 3 : $t_2 < t \leq t_f$ (avec $\tau = t - t_2$)

$$\ddot{s}(t) = -d$$

$$\dot{s}(t) = V - d\tau$$

$$s(t) = d_{\text{acc}} + d_{\text{cst}} + V\tau - \frac{1}{2}d\tau^2$$

Le code applique aussi une saturation numérique en fin de trajectoire : $\dot{s}(t) \geq 0$ et $s(t) \leq 2\pi R$ (évite les dépassemens dus aux arrondis).

2.4. Trajectoire opérationnelle $X(s)$ puis $X(t)$

On définit l'angle paramétrique :

$$u(t) = \frac{s(t)}{R}$$

La trajectoire circulaire (avec y constant) est :

$$x(t) = C_x - R \sin(u(t))$$

$$y(t) = C_y$$

$$z(t) = C_z + R \cos(u(t))$$

2.5. Vitesses et accélérations opérationnelles

$$\text{En utilisant } \dot{u}(t) = \frac{\dot{s}(t)}{R} :$$

Vitesses :

$$\dot{x}(t) = -\dot{s}(t) \cos(u(t))$$

$$\dot{y}(t) = 0$$

$$\dot{z}(t) = -\dot{s}(t) \sin(u(t))$$

Accélérations (termes tangentiels + centripètes) :

$$\ddot{x}(t) = -\ddot{s}(t) \cos(u(t)) + \frac{\dot{s}(t)^2}{R} \sin(u(t))$$

$$\ddot{y}(t) = 0$$

$$\ddot{z}(t) = -\ddot{s}(t) \sin(u(t)) - \frac{\dot{s}(t)^2}{R} \cos(u(t))$$

Contrôle de cohérence (utile pour l'analyse) :

$$|\dot{X}(t)| = \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2} = |\dot{s}(t)|$$

Donc la norme de la vitesse outil doit suivre exactement le profil $\dot{s}(t)$.

3. Primitive de mouvement

3.1. Principe de la fonction $traj(O, R, V)$

La primitive $traj(O, R, V)$ enchaîne les modules pour produire une trajectoire articulaire cohérente avec la consigne circulaire :

Étape A — Loi de mouvement (V.1)

- Calcul de $time$, $s(t)$, $\dot{s}(t)$, $\ddot{s}(t)$ et des temps (t_1, t_2, t_f) à partir du profil 3 phases.

Étape B — Trajectoire opérationnelle (V.2)

- Construction de $X_{ref}(t)$, $\dot{X}_{ref}(t)$, $\ddot{X}_{ref}(t)$ via les équations du cercle et les dérivées.

Étape C — Passage en espace articulaire (V.4)

Pour chaque instant t_k :

1. Position articulaire (MGI numérique)

On cherche $q_k \in \mathbb{R}^6$ tel que le MGD donne la position :

$$f(q_k) \approx X_{ref}(t_k)$$

2. En pratique, on résout numériquement en initialisant avec la solution précédente (continuité) :

$$q_k = \text{MGI}(X_{ref}(t_k), q_{k-1})$$

3. Vitesse articulaire (Jacobienne)

On calcule la Jacobienne géométrique $J(q_k)$, et on ne conserve que la partie linéaire

$$J_v(q_k) \in \mathbb{R}^{3 \times 6}$$
 (pilotage de x, y, z uniquement).

On obtient la vitesse articulaire par pseudo-inverse :

$$\dot{q}_k = J_v(q_k)^+ \dot{X}_{ref}(t_k)$$

4. Accélération articulaire

Elle est obtenue par dérivation numérique de $\dot{q}(t)$ sur la grille temporelle :

$$\ddot{q}_k \approx \frac{\dot{q}_{k+1} - \dot{q}_{k-1}}{2, dt}$$

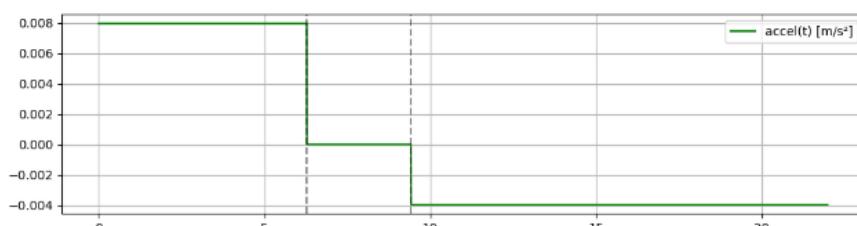
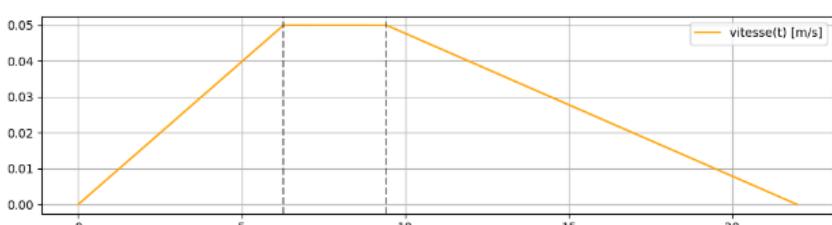
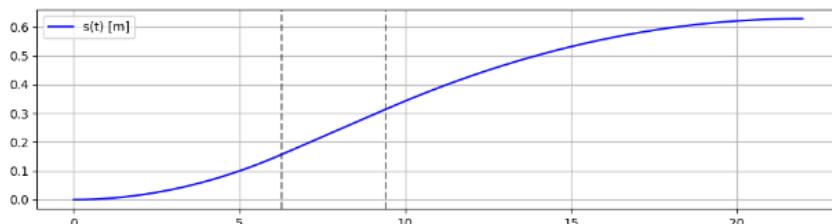
(selon l'opérateur de gradient utilisé).

3.2. Remarques importantes (validité / limites)

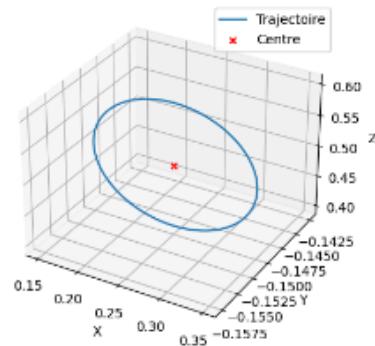
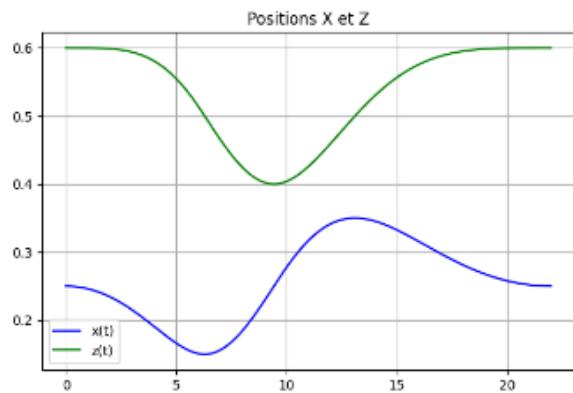
- Comme on ne constraint pas l'orientation, on n'impose que $X = (x, y, z)$: d'où l'utilisation de J_v (3 lignes).
- La pseudo-inverse J_v^+ donne une solution au sens des moindres carrés, mais peut générer de grandes vitesses près des singularités.
- Le MGI numérique dépend fortement de l'initialisation et de l'atteignabilité de $X_{\text{ref}}(t)$ (si le point est hors espace de travail, la convergence peut échouer).
- La dérivation numérique peut amplifier le bruit numérique (surtout sur \ddot{q}), d'où l'intérêt de garder un dt stable et suffisamment fin.

Résultats

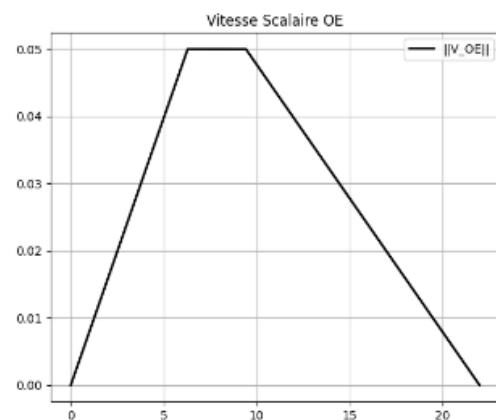
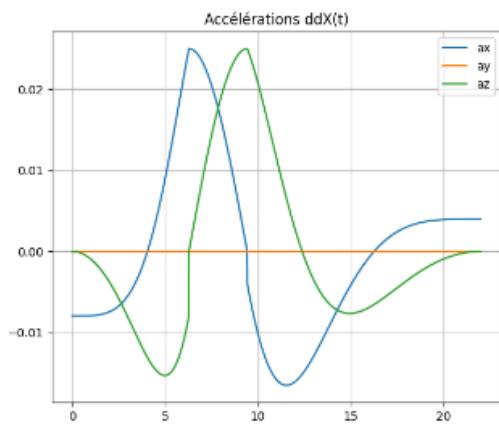
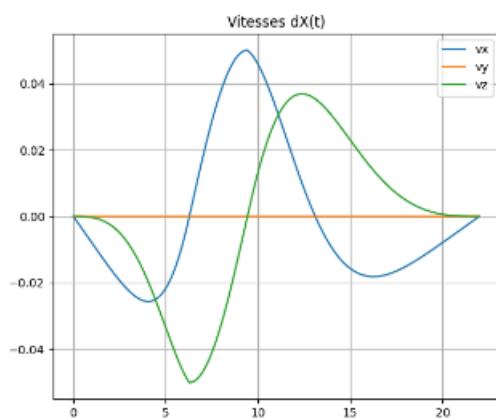
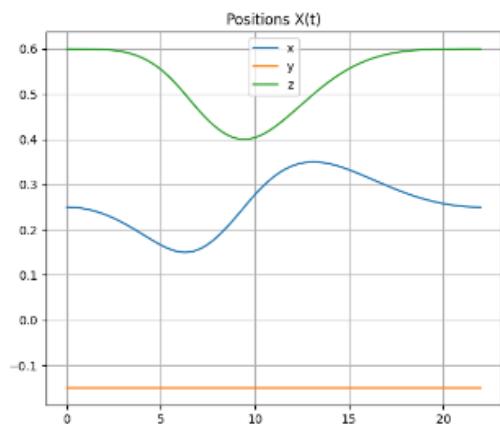
Loi de mouvement temporelle :



Trajectoire opérationnelle X(s) de type cercle :

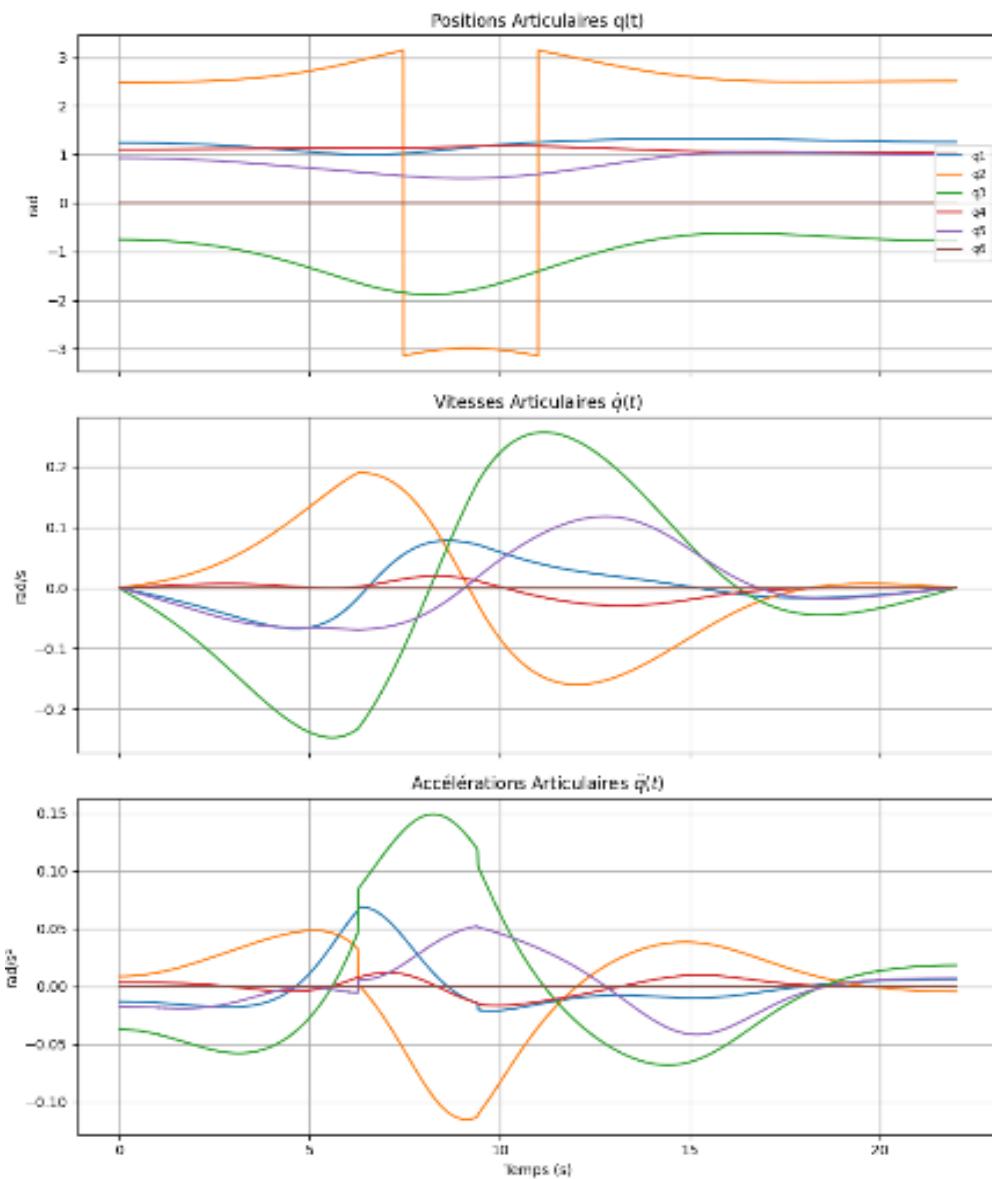


Génération de mouvement dans l'espace de la tâche X(t) :



Vitesse max atteinte par OE : 0.0500 m/s (Cible : 0.05 m/s)

Génération de mouvement dans l'espace articulaire (généralisé) :



Trajectoire générée : 4399 points → Affichage des résultats articulaires