SRI 2A

Promotion : 2024-2025

RAPPORT
BE-GT 2024

Monge Rouchdi Ismaël
Belgnaoui Othman
Crampette Olivier
Danton Laloy Solal

GitHub du Projet : **(LIEN)**

Sommaire :

I. Modélisation du robot

I.1 Dimensions du robot

O0O1 = 550 ; O1O2 =150 ; O2O3 = 825 ; O3O4 =735

I.2 Placement des repères et paramètres de DHM

➔

➔ **I.2.a Placement des repères**

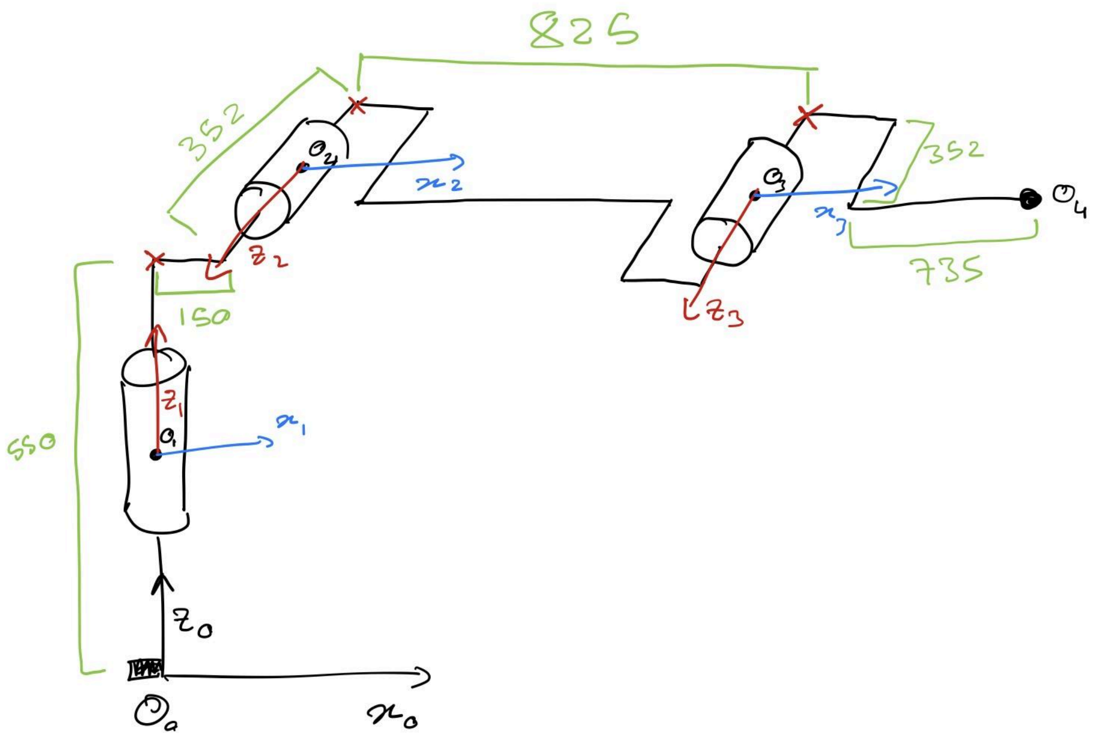


Figure - Placement des repères

→ **I.2.b Paramètres de DHM**

	1	2	3
σi	0	0	0
θi	q	q	q
ri	O0 O1	0	0
ai	0	O1 O2	O2 O3
αi	0	π/2	0
qi_fig	0	0	0

Figure - Paramètres de DHM

- **I.3 Equation du MGD**

$$T01 = \begin{bmatrix} \cos q1 & \text{-}\sin q1 & 0 & 0 \\ \sin q1 & \cos q1 & 0 & 0 \\ 0 & 0 & 1 & 550 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 550 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T12 = \begin{bmatrix} \cos q2 & \text{-}\sin q2 & 0 & 150 \\ 0 & 0 & \text{-}1 & 0 \\ \sin q2 & \cos q2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 150 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T23 = \begin{bmatrix} \cos q3 & \text{-}\sin q3 & 0 & 825 \\ \sin q3 & \cos q3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 825 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T34 = \begin{bmatrix} \cos q4 & 0 & 0 & 735 \\ 0 & \cos q4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 735 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure - Matrices Ti-1,i

Et en calculant T04 nous pouvons trouver les équations de la position de l'organe terminale:

$$x = (150 + 825 * \cos q2 + 735 * \cos(q2 + q3)) * \cos q1$$
$$y = (150 + 825 * \cos q2 + 735 * \cos(q2 + q3)) * \sin q1$$
$$z = 550 + 825 * \sin q2 + 735 * \sin(q2 + q3)$$

- **I.4 Résolution du MGI**

L'objectif de du Modèle Géométrique Inverse (MGI) étant d'obtenir des possibles configurations des angles des liaisons pour atteindre un point de l'organe terminale désiré nous résolvons l'égalité suivante ;

T10*T0T=T12*T23*T34

Avec T0T étant la matrice de translation représentant la position et configuration de l'organe terminale dans la base 0, où dans notre cas on sera uniquement intéressés par les 3 premières lignes de la dernière colonne de cette matrice car elles représentent les coordonnées x, y et z respectivement de l'organe terminale. On trouve donc ceci:

$$\begin{bmatrix} X & X & X & x*c1+y*s1 \\ X & X & X & y*c1-x*s1 \\ X & X & X & z-550 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c4(c2*c3-s2*s3) & c4(-c2*s3-s2*c3) & 0 & 735c2*c3+825c2-735s2*s3+150 \\ 0 & 0 & -1 & 0 \\ c4(c2*s3+s2*c3) & c4(c2*c3-s2*s3) & 1 & 735c2*s3+825s2+735s2*c3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Nous donnant les égalitées suivantes:

$x * c1 + y * s1 = 735c2 * c3 + 825c2 - 735s2 * s3 + 150$

$x * c1 - y * s1 = 0$

$z - 550 = 735c2 * s3 + 825s2 - 735s2 * c3$

Avec q1 facilement à déduire comme $q1 = Atan2(y, x)$ ou $q1 = Atan2(y, x) - \pi$

Et sachant que

$735c2 * c3 - 735s2 * s3 = 735 * cos(q2 + q3)$

et $735c2 * s3 - 735s2 * c3 = 735 * sin(q2 + q3)$

On retrouve finalement les égalités suivantes:

$x * c1 + y * s1 = 735 * cos(q2 + q3) + 825c2 + 150$

$z - 550 = 735 * sin(q2 + q3) + 825s2$

Et finalement utilisant les formules du cours suivantes:

$X.\cos(q_i) + Y.\cos(q_i + q_j) = Z_1$ et
$X.\sin(q_i) + Y.\sin(q_i + q_j) = Z_2$

Equations au carré et addition $\Rightarrow$ élimine q_i.

On déduit $\cos(q_j) = c_j = \frac{Z_1^2 + Z_2^2 - X^2 - Y^2}{2.X.Y}$ $\quad q_j = Atan2(\pm\sqrt{1-(c_j)^2}, c_j)$

On pose $B_1 = X + Y.c_j$ et $B_2 = Y.s_j$ et on a :

$\sin(q_i) = s_i = \frac{B_1.Z_2 - B_2.Z_1}{B_1^2 + B_2^2}$ et $\cos(q_i) = c_i = \frac{B_1.Z_1 + B_2.Z_2}{B_1^2 + B_2^2}$.

$q_i = Atan2(s_i, c_i)$ (liaison rotoïde)

On trouve $c3 = \frac{(c1*x+s1*y-150)^2+(z-550)^2-825^2-735^2}{2*825*735}$

$$q3 = Atan2(+\sqrt{1-(c3)^2}, c3)$$

ou

$$q3 = Atan2(-\sqrt{1-(c3)^2}, c3)$$

Et une foi qu'on connaît q3 on peut déduire q2 comme:

$$s2 = \frac{(x+y*c3)*(z-550)-y*s3*(c1*x+s1*y-150)}{(x+y*c3)^2+(y*s3)^2}$$

$$c2 = \frac{(x+y*c3)*(c1*x+s1*y-150)+y*s3*(z-550)}{(x+y*c3)^2+(y*s3)^2}$$

$$q2 = Atan2(s2, c2)$$

Voici les 4 possibles configurations:

q1	q2	q3
$Atan2(y, x)$	$Atan2(s2, c2)$	$Atan2(+\sqrt{1-(c3)^2}, c3)$
$Atan2(y, x)$	$Atan2(s2, c2)$	$Atan2(-\sqrt{1-(c3)^2}, c3)$
$Atan2(y, x) - \pi$	$Atan2(s2, c2)$	$Atan2(+\sqrt{1-(c3)^2}, c3)$
$Atan2(y, x) - \pi$	$Atan2(s2, c2)$	$Atan2(-\sqrt{1-(c3)^2}, c3)$

- ## I.5 Equation du MDD et calcul de la jacobienne

➔ I.5.a Equation du MDD

L'objectif de du Modèle Dynamique Directe (MDD) étant d'obtenir vitesses linéaires et angulaires de l'organe terminale pour atteindre des vitesses articulaires désirées des différentes liaisons

$dX = J(q) * dq$

➔ I.5.b Calcul de la jacobienne géométrique

$J(q) = \begin{pmatrix} \frac{Jp_i(q)}{Jo_i(q)} \end{pmatrix}$ avec $Jp_i(q) = z_{i-1} \wedge (X_T - X_i)$pour une liaison rotoïde, si prismatique on le laisse comme $Jp_i(q) = z_{i-1}$ et pour n'importe la liaison son vecteur $Jo_i(q) = z_{i-1}$, dans notre cas on avait 3 liaisons rotoïdes donc notre matrice J(q) avait la forme suivante:

$$J(q) = \begin{pmatrix} Jp_1 & Jp_2 & Jp_3 \\ Jo_1 & Jo_2 & JO_3 \end{pmatrix} = \begin{pmatrix} z_0 \wedge (X_T - X1) & z_1 \wedge (X_T - X2) & z_2 \wedge (X_T - X3) \\ z_0 & z_1 & z_2 \end{pmatrix}$$

avec X_T les coordonnées de l'organe terminale, X_1 les coordonnées de la première liaison, X_2 les coordonnées de la deuxième liaison et X_3 les coordonnées de la troisième liaison, toutes dans la base 0. Pour les matrices de translation dans sa configuration initiale (q1=0, q2=0 et q3=0) qu'on a montré au début on trouve:

$$z_0 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, z_1 = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}, z_2 = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}, X_T = \begin{pmatrix} 1710 \\ 0 \\ 550 \end{pmatrix}, X_1 = \begin{pmatrix} 0 \\ 0 \\ 550 \end{pmatrix}, X_2 = \begin{pmatrix} 150 \\ 0 \\ 550 \end{pmatrix}, X_3 = \begin{pmatrix} 975 \\ 0 \\ 550 \end{pmatrix}$$

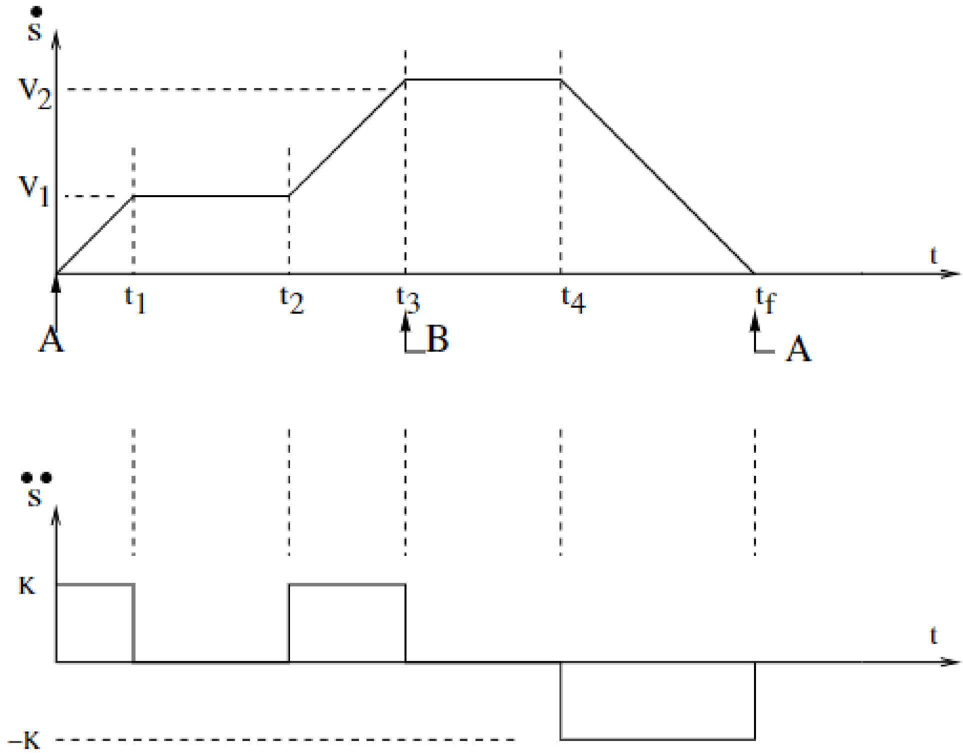
- **I.6 Résolution du MDI**

➔ I.6.a Numérique

L'objectif de du Modèle Dynamique Inverse (MDI) étant d'obtenir vitesses articulaires des différentes liaisons pour atteindre des vitesses linéaires et angulaires désirées de l'organe terminale. Possible que lorsque J(q) est inversible. Nous avons dans notre programme pris en compte le fait que J peut ne pas être carré. Dans ce cas on prend la pseudo inverse comme vu en cour.

$dq = J(q)^{-1} * dX$

II. Génération de mouvements



- **II.1 Choix des points A et B et plan choisi**

Nous avons fait le choix de permettre des points dans le plan (y,z) (donc la composante x doit être égale). Les points sont vérifiés afin d'être sûr qu' il soit bien dans l'espace de travail du robot.

- **II.2 Calcul des temps de commutation tc**

$$t_1 = \frac{V_1}{K}$$

$$t_2 = \frac{\pi \cdot \mathrm{ray} + \frac{V_1 \cdot t_1}{2} - V_1 \cdot \frac{V_2 - V_1}{K} - \frac{(V_2 - V_1)}{2} \cdot \frac{(V_2 - V_1)}{K}}{V_1}$$

$$t_3 = t_2 + \frac{V_2 - V_1}{K}$$

$$t_4 = t_3 + \frac{\pi \cdot \mathrm{ray} - \frac{V_2^2}{2K}}{V_2}$$

$$t_f = t_4 + \frac{V_2}{K}$$

ray = |B-A|/2

Détails des calculs en annexe, à la fin du rapport.

- **II.3 Equations de X(s)**

La loi de commande en position correspond à une abscisse curviligne s. Nous souhaitons que la trajectoire géométrique aille du point A au point B en suivant le vecteur U de la droite (AB) et en évoluant de 0% (0 / d) à 100% (s_{final} /d)

$$X(s) = \begin{pmatrix} x_A + U_x \cdot \frac{s}{d} \\ y_A + U_y \cdot \frac{s}{d} \\ z_A + U_z \cdot \frac{s}{d} \end{pmatrix}$$

- **II.4 Equations de loi de mouvement s(t) , $\dot{s}$(t) , $\ddot{s}$(t)**

➔ **II.4.a Équation de s(t)**

$$s(t) = \begin{cases} \frac{1}{2}Kt^2 & \text{si } t < t_1, \\ s(t_1) + V_1 \cdot (t - t_1) & \text{si } t_1 \leq t < t_2, \\ s(t_2) + V_1 \cdot (t - t_2) + \frac{1}{2}K(t - t_2)^2 & \text{si } t_2 \leq t < t_3, \\ s(t_3) + V_2 \cdot (t - t_3) & \text{si } t_3 \leq t < t_4, \\ s(t_4) + V_2 \cdot (t - t_4) - \frac{1}{2}K(t - t_4)^2 & \text{si } t \geq t_4. \end{cases}$$

➔ **II.4.b Équation de s'(t)**

$$v(t) = \begin{cases} K \cdot t & \text{si } t < t_1, \\ V_1 & \text{si } t_1 \leq t < t_2, \\ V_1 + K \cdot (t - t_2) & \text{si } t_2 \leq t < t_3, \\ V_2 & \text{si } t_3 \leq t < t_4, \\ V_2 - K \cdot (t - t_4) & \text{si } t \geq t_4. \end{cases}$$

➔ **II.4.c Équation de s''(t)**

$$a(t) = \begin{cases} K & \text{si } t < t_1, \\ 0 & \text{si } t_1 \leq t < t_2, \\ K & \text{si } t_2 \leq t < t_3, \\ 0 & \text{si } t_3 \leq t < t_4, \\ -K & \text{si } t \geq t_4. \end{cases}$$

Positions clés:

$$s(t_1) = \frac{1}{2}Kt_1^2,$$
$$s(t_2) = s(t_1) + V_1 \cdot (t_2 - t_1),$$
$$s(t_3) = s(t_2) + V_1 \cdot (t_3 - t_2) + \frac{1}{2}K(t_3 - t_2)^2,$$
$$s(t_4) = s(t_3) + V_2 \cdot (t_4 - t_3).$$

- **II.5 Equations de X(t), X˙ (t)**

  ➔ **II.4.c Équation de X(t)**

On sait que l'organe terminal doit parcourir un cercle qui commence au point A, passe par le point B et retourne au point A en respectant les lois de vitesse et d'accélération.
On commence par trouver les paramètres du cercle parcouru :

Rayon = $\sqrt{\left(y_b - \frac{y_a - y_b}{2}\right)^2 + \left(z_b - \frac{z_a - z_b}{2}\right)^2}$

x_centre = $x_a = x_b$ y_centre = $\frac{y_a - y_b}{2}$ z_centre = $\frac{z_a - z_b}{2}$

On sait que la distance parcourue par l'organe terminal est s(tf) = $2 \cdot \pi \cdot rayon$

or : $s(t) = \int_0^{tf} s'(t) = \sum_0^N$s'[i] (en dans notre cas discrétisé)

calcul des angles de l'organe terminal autour du cercle :

$\theta(t) = \frac{s(t)}{rayon} + \theta_0$ Avec $\theta_0 = atan2(z_a - \frac{z_a - z_b}{2}, y_a - \frac{y_a - y_b}{2})$

θ_0 est le déphasage qui représente l'angle au point A.

D'où les positions de l'organe terminal :

$$X(t) = \begin{cases} x(t) = x_a = x_b \\ y(t) = \frac{y_a - y_b}{2} + rayon \cdot \cos(\theta(t)) \\ z(t) = \frac{z_a - z_b}{2} + rayon \cdot \sin(\theta(t)) \end{cases}$$

➔ II.4.c Équation de X'(t)

Pour trouver les vitesses opérationnelles, nous avons dérivé les positions opérationnelles par rapport au pas d'échantillonnage. Le temps a été généré comme un vecteur de 0 à t_f avec un pas de $dt = 5\, ms$.

$X'(t) = \frac{dX(t)}{dt}$ Et en numérique : $X[i] = X[i + 1] - X[i]$

Et pour récupérer le dernier point, nous avons utilisé la fonction d'extrapolation en 1 dimension de scipy. Nous avons procédé de la même façon avec les accélérations opérationnelles.

III. Primitive de mouvement

- **III.1 Explication sur la réalisation de la primitive traj(A, B, V1, V2)**

La réalisation de la fonction trajectoire reprend la génération des coordonnées opérationnelles avec les équations décrites précédemment. Elle calcule les temps de commutation en fonction des vitesses et de l'accélération donnée. Elle définit les vecteurs de vitesses et d'accélérations opérationnelles en fonction des temps de commutation calculés.
Itère sur toutes les positions à atteindre et utilise le MGI pour trouver des configurations articulaires qui correspondent à la position de l'OT désirée. Nous avons introduit un mécanisme qui compare chaque solution à la précédente pour choisir la configuration la moins éloignée de la précédente pour éviter les discontinuités en cas de passage par une singularité par exemple. Elle calcule aussi les vitesses articulaires à l'aide des fonctions MDI et de calcule de la jacobienne géométrique. Enfin, elle affiche les fonctions demandées.

- **III.2 Particularités de notre projet**

Une des particularités de notre travail réside dans les simulations simplifiées du bras robotique. Nous avons commencé par mettre en œuvre des simulations statiques du bras afin de vérifier que le MGD était correctement calculé et que le MGI déterminait convenablement la configuration du bras pour des coordonnées données. Cela a grandement facilité et accéléré la vérification des réponses correctes. Une autre particularité utile pour confirmer l'exactitude de nos calculs a été l'implémentation de la fonction **"Jacob_analytique()"** dans **"modele_differentiel.py"**. Grâce à **sympy**, cette fonction nous permettait de vérifier que nos calculs utilisant les symboles **cos** et **sin** étaient corrects. De plus, lorsqu'on lui introduisait une liste de configuration des liaisons en entrée, elle calculait la jacobienne, ce qui nous permettait de confirmer si la fonction **"Jacob_geo"** la calculait également de manière appropriée. Toutes ces vérifications peuvent être retrouvées sur le fichier **"test_modele_differentiel.py"**.

IV. Simulation fonctionnelle

Notre programme offre la possibilité de simuler (avec plotly) la trajectoire obtenue après avoir entré les points A, B, les vitesses V1 et V2 ainsi que l'accélération souhaitée. Pour cela, il faut exécuter le fichier **"main.py"** et choisir l'option 2 :

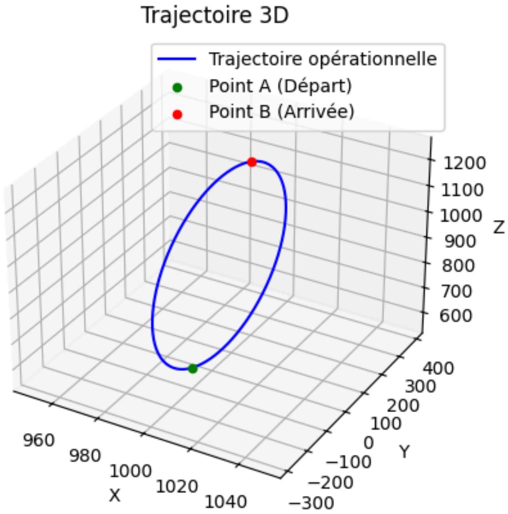
```
Menu:
1. Matrices,Mgd,Mgi
2. Visualiser des lois de mouvement temporelles et les graph associés
0. Quitter
Veuillez choisir une option:
```

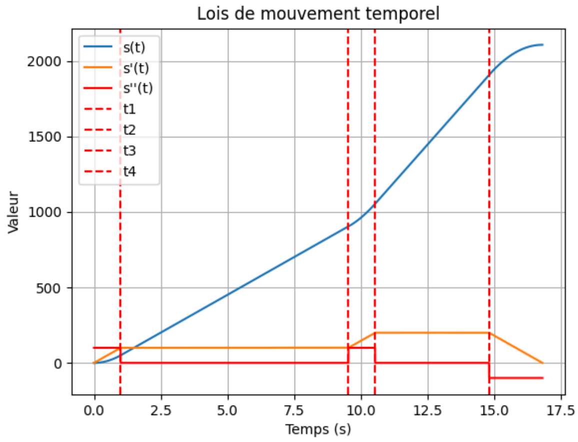
Puis entrer les paramètres souhaités :

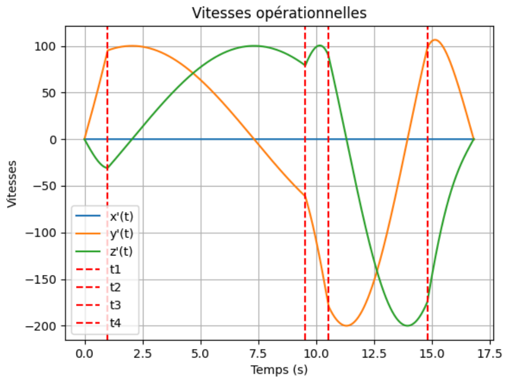
```
Veuillez choisir une option: 2
Vitesse 1 :
100
Vitesse 2 :
200
Coordonnée x partagé par A et B :
1000
Coordonnée y pour A :
-100
Coordonnée z pour A :
600
Coordonnée y pour B :
200
Coordonnée z pour B :
1200
Acceleration :
100
A = [1000.  -100.   600.] B = [1000.   200.  1200.] V1 = 100.0 V2 = 200.0
pos init [1000.  -100.   600.] pos final [1000.          -100.00024451   600.00012225]
Voulez vous lancer une simulation avec ces données?
1: Oui 2: Non
```

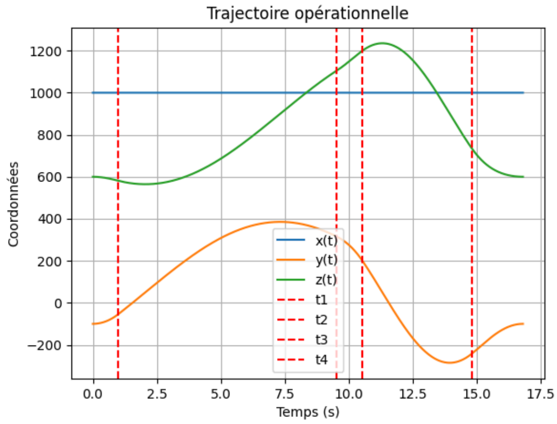
Un excellent choix de valeurs pour les vitesses et les accélérations afin de tester l'animation et observer clairement les variations de vitesse est le suivant :
V1 = 500, V2 = 3000, K = 3000.

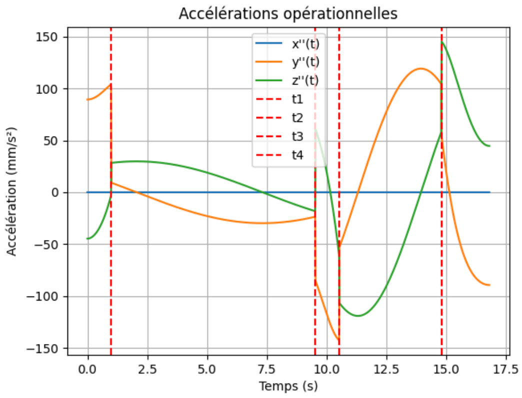
Vous aurez toutes les courbes demandées :

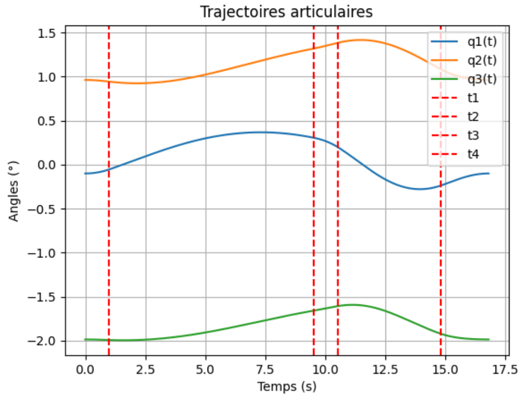

Trajectoire 3D
Trajectoire opérationnelle
Point A (Départ)
Point B (Arrivée)
X
Y
Z

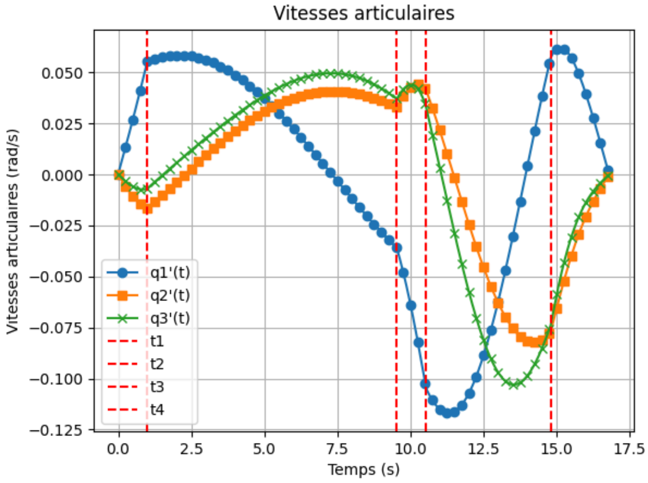

Lois de mouvement temporel
s(t)
s'(t)
s''(t)
t1
t2
t3
t4
Valeur
Temps (s)


Vitesses opérationnelles
x'(t)
y'(t)
z'(t)
t1
t2
t3
t4
Vitesses
Temps (s)


Trajectoire opérationnelle
x(t)
y(t)
z(t)
t1
t2
t3
t4
Coordonnées
Temps (s)


Accélérations opérationnelles
x''(t)
y''(t)
z''(t)
t1
t2
t3
t4
Accélération (mm/s²)
Temps (s)

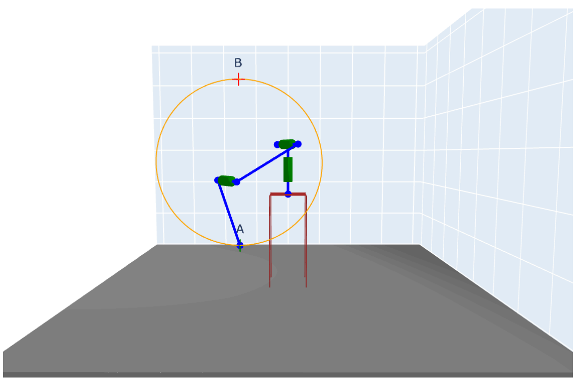

Trajectoires articulaires
q1(t)
q2(t)
q3(t)
t1
t2
t3
t4
Angles (°)
Temps (s)

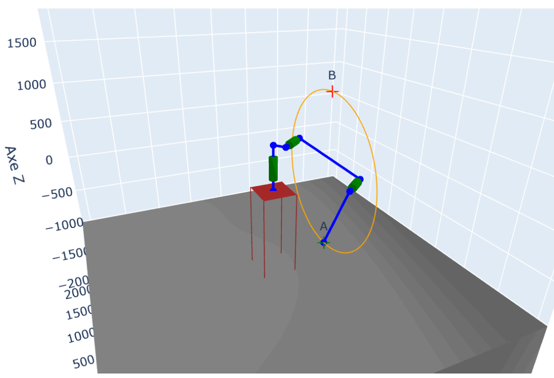


Puis une fois les paramètres entrés, il faudra accepter de lancer une simulation en choisissant l'option 1 et attendre qu'elle soit générée :

```
Voulez vous lancer une simulation avec ces données?
1: Oui 2: Non
1
Génération de la simulation...
```

Vous obtiendrez une simulation semblable à celle-ci. Pour ce projet, nous avons choisi d'imaginer le robot installé sur une petite base, à la fois stable et rigide, capable de supporter ses mouvements et d'offrir ainsi une plus grande liberté de mouvement. Il ne manque plus qu'à vous positionner dans l'angle de vue souhaité et cliquer sur "rejouer" (vous pouvez changer l'angle de vue quand la simulation est en pause) :





V. Conclusion

Ce projet a été une belle occasion de mettre en pratique nos compétences en robotique tout en relevant des défis stimulants. Travailler sur la modélisation et la génération de trajectoire nous a permis de mieux comprendre la logique qui sous-tend le mouvement des robots, et nous avons beaucoup appris en cherchant des solutions adaptées à chaque étape.

Nous avons aussi apprécié la satisfaction de voir notre code produire des résultats concrets. Cette expérience nous a permis de consolider nos connaissances et de mieux appréhender le domaine.

VI. Annexes

Calcul des temps de commutation tc :

Temps de commutations

$$\int_0^{t_1} \ddot{s}(t)\,dt = [\dot{s}(t)]_0^{t_1} = \dot{s}(t_1) - \dot{s}(0) = V_1$$

$$\int_0^{t_1} \ddot{s}(t)\,dt = k \times t_1 = V_1 \Rightarrow \boxed{t_1 = \frac{V_1}{k}}$$

$$\int_{t_2}^{t_3} \ddot{s}(t)\,dt = [\dot{s}(t)]_{t_2}^{t_3} = \dot{s}(t_3) - \dot{s}(t_2) = V_2 - V_1$$

$$\int_{t_2}^{t_3} \ddot{s}(t)\,dt = k \times (t_3 - t_2) \Rightarrow \boxed{t_3 - t_2 = \frac{V_2 - V_1}{k}}$$

$$\int_{t_4}^{t_f} \ddot{s}(t)\,dt = -k(t_f - t_4) = \dot{s}(t_f) - \dot{s}(t_4) = 0 - V_2$$

$$\Rightarrow \boxed{t_f - t_4 = \frac{V_2}{k}}$$

Arc de cercle

$$\int_{t_0}^{t_3} \dot{s}(t)\,dt = [s(t)]_{t_0}^{t_3} = \pi \times \frac{|B-A|}{2}$$

$$\int_{t_0}^{t_3} \dot{s}(t)\,dt = \frac{V_1}{2} t_1 + (t_2 - t_1)V_1 + V_1(t_3 - t_2) + \frac{V_2 - V_1}{2}(t_3 - t_2)$$

$$= \frac{V_1}{2} t_1 + t_2 V_1 - V_1 t_1 + V_1 \frac{(V_2 - V_1)}{k} + \frac{(V_2 - V_1)^2}{2k}$$

$$= \frac{-V_1}{2} t_1 + t_2 V_1 + \frac{V_1(V_2 - V_1)}{k} + \frac{(V_2 - V_1)^2}{2k}$$

$$\Rightarrow \boxed{t_2 = \left[\left(\pi \times \frac{|B-A|}{2}\right) + \frac{V_1}{2} t_1 - \frac{V_1(V_2 - V_1)}{k} - \frac{(V_2 - V_1)^2}{2k}\right] \times \frac{1}{V_1}}$$

$$\boxed{t_3 = \frac{V_2 - V_1}{k} + t_2}$$

$$\int_{t_3}^{tf} \dot{s}(t) = \frac{\pi|B-A|}{2}$$

$$= V_2(t_4 - t_3) + \frac{V_2}{2}(tf - t_4)$$

$$\frac{\pi|B-A|}{2} = V_2 t_4 - V_2 t_3 + \frac{V_2}{2}\,\frac{\pi|B-A|}{V_2}$$

$$V_2 t_4 = \frac{\pi|B-A|}{2} + V_2 t_3 - \frac{V_2 \pi|B-A|}{2V_2}$$

$$\boxed{t_4 = \frac{\pi|B-A|}{2V_2} + t_3 - \frac{\pi|B-A|}{2V_2}}$$

$$\int_{t_3}^{tf} \dot{s}(t)\,dt = \frac{\pi|B-A|}{2}$$

$$\frac{\pi|B-A|}{2} = V_2(t_4 - t_3) + \frac{V_2}{2}(tf - t_4)$$

$$= V_2 t_4 - V_2 t_3 + \frac{V_2^2}{2k} \qquad \text{cf}$$

$$V_2 t_4 = \frac{\pi|B-A|}{2} + V_2 t_3 - \frac{V_2^2}{2k}$$

$$\boxed{t_4 = \frac{\pi|B-A|}{2V_2} + t_3 - \frac{V_2}{2k}}$$

$$tf - t_4 = \frac{V_2}{k} \Rightarrow \boxed{tf = \frac{V_2}{k} + t_4}$$