

# Rapport

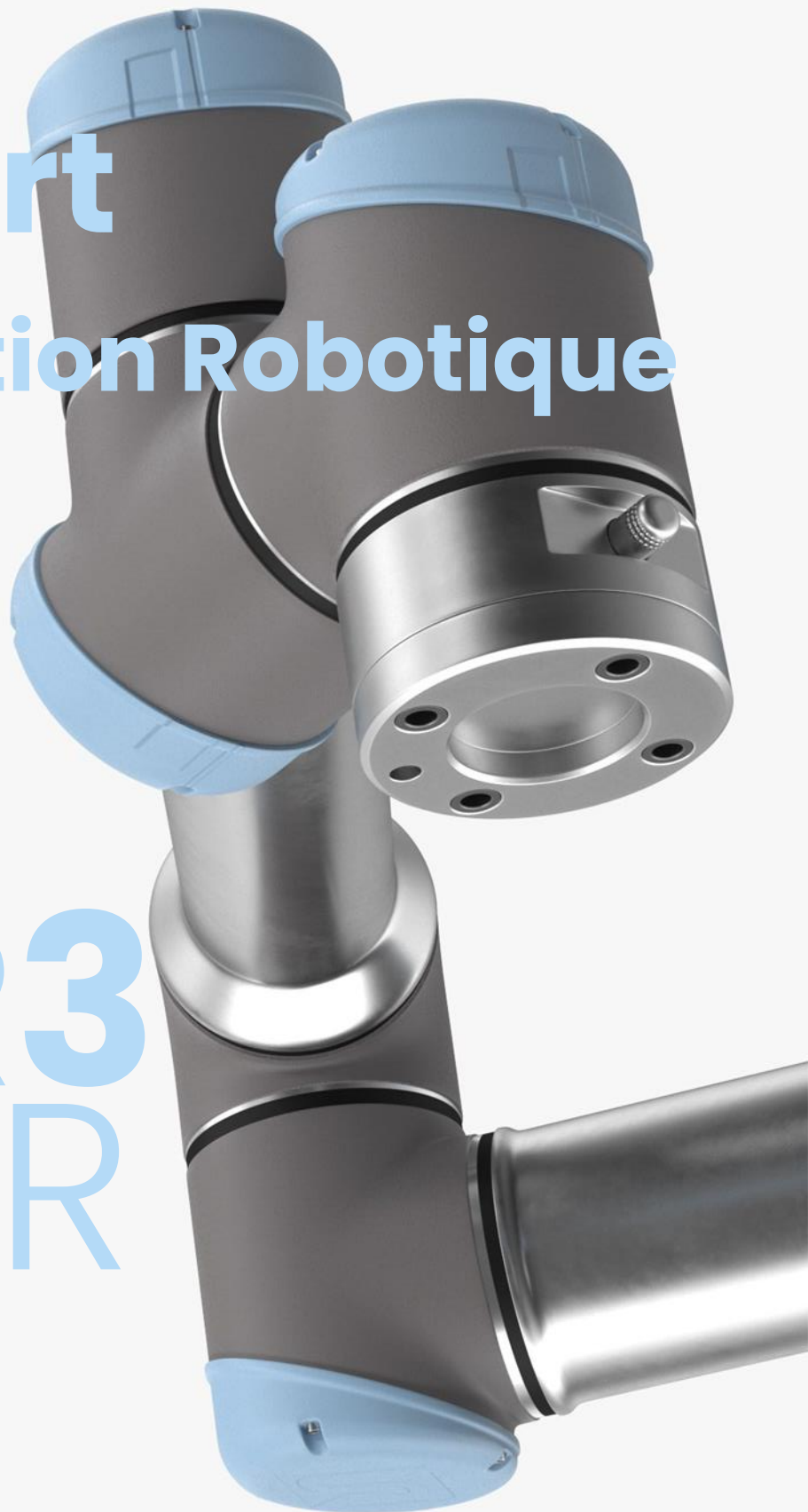
## Modélisation Robotique

### Membres du groupe :

- GHILOUFI Nour,
- FERHANI Amal,
- GHAZOUANI Souha,
- GIBERT Alexis

# UR3

## 6R/3R



Sommaire

Introduction..... 3

1 Modélisation UR3 (6R)..... 4

1.1 Dimensions du robot et placement des repères..... 4

1.2 Placement des repères et paramètres de DHM (modifiés)..... 4

2 Génération de mouvement pour le robot UR3 réduit (3R) ..... 4

2.1 Modélisation UR3 réduit (3R)..... 4

2.1.1 Placement des repères et paramètres de DHM (modifiés)..... 4

2.1.2 Equation du MGD..... 5

2.1.3 Résolution du MGI..... 5

2.1.4 Equation du MDD..... 6

2.1.5 Résolution du MDI..... 6

2.2 Génération de mouvement ..... 7

2.2.1 Calcul des temps de commutation  $T$ , de  $K1$  et  $K2$ ..... 7

2.2.2 Equations de  $X(s)$ ..... 7

2.2.3 Equations de loi de mouvement  $st, st, st$  ..... 7

2.2.4 Equations de  $Xt, Xt$  ..... 8

2.3 Primitive de mouvement..... 9

2.3.1 Explication sur la réalisation de la primitive  $\text{traj}(A, B, V1, V2)$  ..... 9

2.3.2 Particularités de notre projet..... 10

## **Introduction**

Le but de ce projet est de développer une primitive de mouvement afin de faire réaliser à l'organe terminal d'un robot manipulateur un mouvement imposé (géométrie et vitesse). La trajectoire dans l'espace opérationnel se fait à vitesse imposée et les modèles d'un robot manipulateur permettent de calculer les commandes en position, vitesse et accélération à envoyer au robot pour générer le mouvement désiré. La tâche est donc définie dans l'espace opérationnel et il est demandé de mettre en évidence le lien entre l'espace opérationnel et l'espace généralisé.

## 1 Modélisation UR3 (6R)

### 1.1 Dimensions du robot et placement des repères

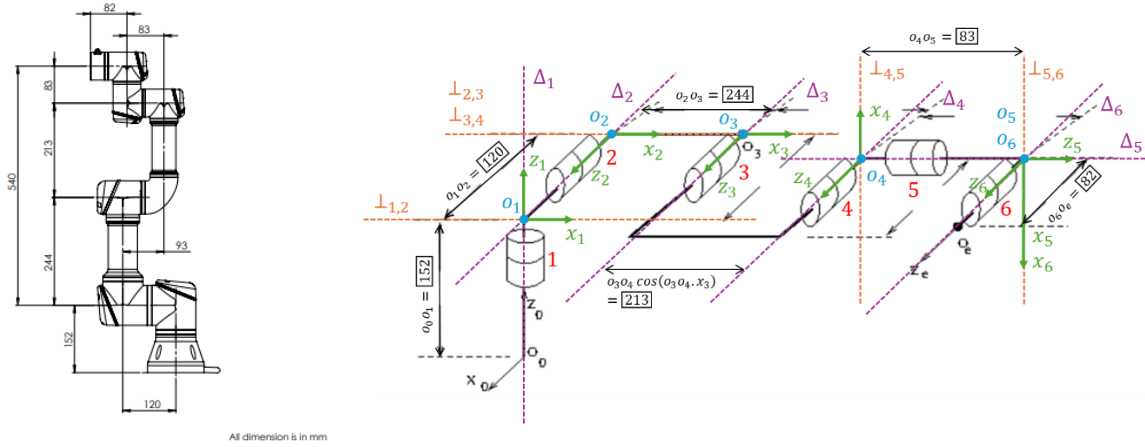


Figure 1: Dimensions géométriques du robot UR3 (à gauche) et placement des repères du robot UR3 (à droite)

### 1.2 Placement des repères et paramètres de DHM (modifiés)

	1	2	3	4	5	6
$a_{i-1}$	0	0	$o_2 o_3$	$o_3 o_4 \cos(o_3 o_4 \cdot x_3)$	0	0
$\alpha_{i-1}$	0	$\frac{\pi}{2}$	0	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$
$r_i$	$o_0 o_1$	$-o_1 o_2$	0	$o_3 o_4 \cos(o_3 o_4 \cdot z_4)$	$o_4 o_5$	0
$\theta_i$	$q_1$	$q_2$	$q_3$	$q_4$	$q_5$	$q_6$
$q_{i \text{ fig}}$	$\frac{\pi}{2}$	0	0	$\frac{\pi}{2}$	$-\pi$	0

Tableau 1: Paramètres de DHM (modifiés) pour le 6R

Grâce à Figure 1, on notera :

$$o_0 o_1 = 152 \quad o_1 o_2 = 120 \quad o_2 o_3 = 244 \quad o_3 o_4 \cos(o_3 o_4 \cdot x_3) = 213 \quad o_3 o_4 \cos(o_3 o_4 \cdot z_4) = 83 \quad o_4 o_5 = 83$$

## 2 Génération de mouvement pour le robot UR3 réduit (3R)

### 2.1 Modélisation UR3 réduit (3R)

#### 2.1.1 Placement des repères et paramètres de DHM (modifiés)

Figure 2 shows the coordinate frames  $O_0$  to  $O_3$  and the DH parameters for the 3R reduced robot. The DH parameters are:  $a_0 = 152$ ,  $a_1 = 120$ , and  $a_2 = 244$ . The DH parameters are also labeled as  $\Delta_1$  to  $\Delta_3$ .

	1	2	3
$a_{i-1}$	0	0	$o_2 o_3$
$\alpha_{i-1}$	0	$\frac{\pi}{2}$	0
$r_i$	$o_0 o_1$	$-o_1 o_2$	0
$\theta_i$	$q_1$	$q_2$	$q_3$
$q_{i \text{ fig}}$	$\frac{\pi}{2}$	0	0

Tableau 2: Repères et paramètres de DHM (modifiés) pour le 3R

### 2.1.2 Equation du MGD

Avec les paramètres de la DHM, nous obtenons les matrices génériques suivantes :

$$T_{01} = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & 0 \\ \sin q_1 & \cos q_1 & 0 & 0 \\ 0 & 0 & 1 & o_1 o_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{12} = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & 0 \\ 0 & 0 & -1 & o_1 o_2 \\ \sin q_2 & \cos q_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{23} = \begin{bmatrix} \cos q_3 & -\sin q_3 & 0 & o_2 o_3 \\ \sin q_3 & \cos q_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{3n} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & b \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{0n} = T_{01} \times T_{12} \times T_{23} \times T_{3n}$$

Grâce à  $T_{0n}$ , on obtient le vecteur de position  $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$

$$\begin{aligned} x &= \cos(q_1) \times (a \times \cos(q_2 + q_3) + \cos(q_2) \times o_2 o_3) + \sin(q_1) \times (b - o_1 o_2) \\ y &= \sin(q_1) \times (a \times \cos(q_2 + q_3) + \cos(q_2) \times o_2 o_3) - \cos(q_1) \times (b - o_1 o_2) \\ z &= o_1 o_2 + \sin(q_2) \times o_2 o_3 + a \times \sin(q_2 + q_3) \end{aligned}$$

### 2.1.3 Résolution du MGI

#### 2.1.3.1 Détermination de $q_3$ et $q_2$

Le vecteur de position  $X$ , permet d'écrire :

$$x^2 + y^2 = a \times c_{2+3} + o_2 o_3 \times c_2 + b - o_1 o_2$$

Soit les formules du cours  $X \cos q_i + Y \cos(q_i + q_j)$  et  $X \sin q_i + Y \sin(q_i + q_j)$  suivantes :

$$\begin{aligned} (1): X \cos q_i + Y \cos(q_i + q_j) &= Z_1 \\ \cos q_j &= c_j = \frac{Z_1^2 + Z_2^2 - X^2 + Y^2}{2XY} \\ (2): X \sin q_i + Y \sin(q_i + q_j) &= Z_2 \\ s_i &= \frac{B_1 Z_2 - B_2 Z_1}{B_1^2 + B_2^2} \xrightarrow[B_2=Ys_j]{B_1=X+Yc_j} \frac{(X + Yc_j)Z_2 - (Ys_j)Z_1}{(X + Yc_j)^2 + (Ys_j)^2} \\ c_i &= \frac{B_1 Z_1 + B_2 Z_2}{B_1^2 + B_2^2} \xrightarrow[B_2=Ys_j]{B_1=X+Yc_j} \frac{(X + Yc_j)Z_1 + (Ys_j)Z_2}{(X + Yc_j)^2 + (Ys_j)^2} \end{aligned}$$

On peut ainsi écrire à l'aide de vecteur de position  $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ :

$$\begin{aligned} \Leftrightarrow \boxed{Z_1 = o_2 o_3 \cos q_2 + a \cos(q_2 + q_3)} & \quad \Leftrightarrow \boxed{Z_2 = o_2 o_3 \sin q_2 + a \sin(q_2 + q_3)} \\ \text{avec } \boxed{Z_1 = \pm \sqrt{x^2 + y^2 - (b - o_2 o_3)^2}} & \quad \text{avec } \boxed{Z_2 = z - o_1 o_2} \\ \xRightarrow{c_3=c_j} \boxed{q_3 = \text{Atan2}\left(\pm \sqrt{1 - c_3^2}, c_3\right)} & \quad \xRightarrow{s_2=s_i, c_2=c_i} \boxed{q_2 = \text{Atan2}(s_2, c_2)} \end{aligned}$$

#### 2.1.3.2 Détermination de $q_1$

A l'aide des formules du cours  $X_1 \sin q_i + Y_1 \cos(q_i)$  et  $X_2 \sin q_i + Y_2 \cos(q_i)$  on peut écrire :

$$\begin{aligned} (1): X_1 \sin q_i + Y_1 \cos(q_i) &= Z_1 & (2): X_2 \sin q_i + Y_2 \cos(q_i) &= Z_2 \\ \Rightarrow s_i &= \frac{Z_1 Y_2 - Z_2 Y_1}{X_1 Y_2 - X_2 Y_1}, c_i = \frac{Z_2 X_1 - Z_1 X_2}{X_1 Y_2 - X_2 Y_1} \\ \Leftrightarrow \begin{cases} X_1 = b - o_1 o_2 \\ Y_1 = a c_{2+3} + o_2 o_3 c_2 \\ Z_1 = X \end{cases} & \quad \Leftrightarrow \begin{cases} X_2 = a c_{2+3} + o_2 o_3 c_2 \\ Y_2 = -b + o_1 o_2 \\ Z_2 = Y \end{cases} \\ \xRightarrow{s_1=s_i, c_1=c_i} \boxed{q_1 = \text{Atan2}(s_1, c_1)} & \end{aligned}$$

On peut représenter l'ensemble des  $q$  trouvé comme illustré Figure 2.

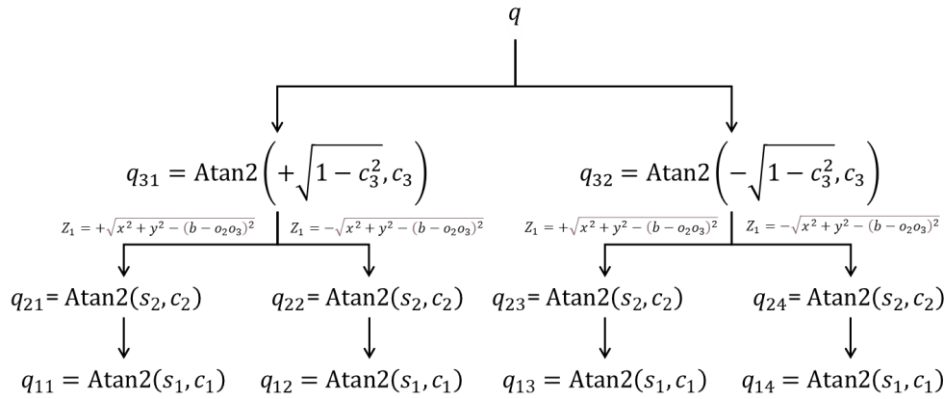


Figure 2 : Différents cas de  $q$

#### 2.1.4 Equation du MDD

La différenciation entre la jacobienne analytique et la jacobienne géométrique se manifeste par la manière dont les vitesses angulaires sont appréhendées. En pratique, la jacobienne analytique reflète les changements de vitesse angulaire associés à l'orientation spécifiée de l'organe terminal, tandis que la jacobienne géométrique quantifie les variations de vitesse angulaire autour des axes  $x$ ,  $y$  et  $z$ . Dans notre contexte, l'approche privilégiée consiste à calculer la jacobienne géométrique.

On a commencé par appliquer la formule pour le calcul de la matrice jacobienne pour obtenir cette matrice appelée  $J(q)_0$  qui représente le  $o_0\dot{o}_3$

$$J(q)_0 = \begin{bmatrix} o_1 o_2 c_1 & -o_2 o_3 c_1 s_2 & 0 \\ -o_1 o_2 s_1 & -o_2 o_3 s_2^2 & 0 \\ 0 & o_2 o_3 (c_2 s_1^2 + c_2 c_1^2) & 0 \end{bmatrix}$$

On cherche maintenant la matrice jacobienne jusqu'à l'organe terminal, pour cela on doit trouver  $o_0\dot{o}_E$

Comme  $o_0\dot{o}_E = o_0\dot{o}_3 + o_3\dot{o}_E$  et qu'on connaît  $o_0\dot{o}_3$  alors :

$$o_3\dot{o}_E = \begin{pmatrix} -as_1c_{2+3} + bc_1 \\ ac_1c_{2-3} + bs_1 \\ 0 \end{pmatrix} \dot{q}_1 + \begin{pmatrix} -ac_1s_{2+3} \\ -as_1s_{2-3} \\ ac_{2+3} \end{pmatrix} \dot{q}_2 + \begin{pmatrix} -ac_1s_{2+3} \\ as_1s_{2-3} \\ ac_{2+3} \end{pmatrix} \dot{q}_3$$

Ainsi, en additionnant  $J(q)_0$  avec  $o_3\dot{o}_E$ , on obtient finalement la matrice  $J$  suivante :

$$J = \begin{bmatrix} -as_1c_{2+3} + bc_1 - o_1o_2c_1 - o_2o_3c_2s_1 & -ac_1s_{2+3} - o_2o_3c_1s_2 & -ac_1s_{2+3} \\ ac_1c_{2+3} + bs_1 + (o_0o_1 + o_1o_2 - o_2o_3)s_1 + o_2o_3c_2c_1 & -as_1s_{2-3} - o_2o_3s_2s_1 & as_1s_{2-3} \\ 0 & ac_{2+3} + o_2o_3c_2 & ac_{2+3} \end{bmatrix}$$

#### 2.1.5 Résolution du MDI

Le modèle différentiel indirect, étant l'inverse du modèle différentiel direct, est calculé numériquement en inversant la jacobienne analytique  $J(q)$ . Il faut simplement vérifier que le déterminant de  $J(q)$  n'est pas nul pour que cette dernière soit inversible.

## 2.2 Génération de mouvement

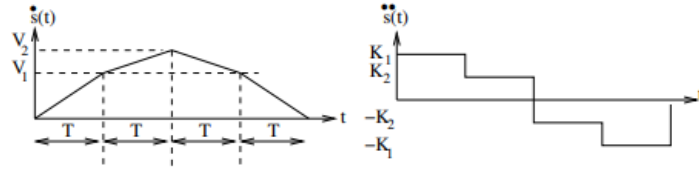


Figure 3 : Profil de vitesse et d'accélération de l'abscisse curviligne

### 2.2.1 Calcul des temps de commutation $T$ , de $K_1$ et $K_2$

Soit la distance entre  $A$  et  $B$  notée  $d_{AB}$  :

$$d_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2}$$

#### 2.2.1.1 Calcul de $T$

$$\begin{aligned} \int_0^T \dot{s}(t) \cdot dt &= \frac{1}{2} \cdot V_1 \cdot T \Leftrightarrow \int_T^{2T} \dot{s}(t) \cdot dt = V_2 \cdot T - \frac{V_1 - V_2}{2} \cdot T = T \cdot \frac{V_1 + V_2}{2} \\ \Leftrightarrow \int_0^{4T} \dot{s}(t) \cdot dt &= d_{AB} = 2 \cdot \left( \frac{1}{2} \cdot V_1 \cdot T \right) + 2 \cdot T \cdot \frac{V_1 + V_2}{2} \Leftrightarrow T \cdot (2 \cdot V_1 + V_2) = d_{AB} \\ \Leftrightarrow T &= \frac{d_{AB}}{2 \cdot V_1 + V_2} \end{aligned}$$

#### 2.2.1.2 Calcul de $K_1$ et $K_2$

$$\begin{aligned} \int_0^T \ddot{s}(t) \cdot dt &= K_1 \cdot T = V_1 \\ \Leftrightarrow K_1 &= \frac{V_1}{T} = V_1 \cdot \frac{2 \cdot V_1 + V_2}{d_{AB}} \\ \int_T^{2T} \ddot{s}(t) \cdot dt &= K_2 \cdot T + V_1 = V_2 \\ \Leftrightarrow K_2 &= \frac{V_2 - V_1}{T} = (V_2 - V_1) \cdot \frac{2 \cdot V_1 + V_2}{d_{AB}} \end{aligned}$$

### 2.2.2 Equations de $X(s)$

Tout d'abord, on a :  $x(t) = a \cdot s(t) + b$

Or, à  $t = 0 \Rightarrow s(0) = 0$  et à  $t = 4T \Rightarrow s(4T) = d_{AB}$

Donc :

- $x(0) = a \cdot 0 + b = x_A \Rightarrow b = x_A$
- $x(4T) = a \cdot d_{AB} + b = x_B \Rightarrow a = \frac{x_B - x_A}{d_{AB}}$

En faisant la même chose pour  $y(t)$  et  $z(t)$ , on obtient :

$$x(t) = \frac{x_B - x_A}{d_{AB}} \cdot s(t) + x_A$$

$$y(t) = \frac{y_B - y_A}{d_{AB}} \cdot s(t) + y_A$$

$$z(t) = \frac{z_B - z_A}{d_{AB}} \cdot s(t) + z_A$$

### 2.2.3 Equations de loi de mouvement $s(t)$ , $\dot{s}(t)$ , $\ddot{s}(t)$

- $0 \leq t \leq T$ 
  - $\ddot{s}(t) = K_1$
  - $\dot{s}(t) = K_1 \cdot t$
  - $s(t) = K_1 \cdot t^2$
- $T \leq t \leq 2T$ 
  - $\ddot{s}(t) = K_2$
  - $\int_T^t \ddot{s}(u) \times du = \dot{s}(t) - \dot{s}(T) = \dot{s}(t) - V_1$   
 $= \int_T^t K_2 \times du = K_2(t - T)$   
 $\Leftrightarrow \dot{s}(t) = K_2(t - T) + V_1$
  - $\int_T^t \dot{s}(u) \times du = s(t) - s(T) = s(t) - \frac{1}{2} \times K_1 \times T^2$

$$\begin{aligned}
 &= \int_T^t (K_2(u - T) + V_1) \times du = \left[ K_2 \times \frac{u^2}{2} + (-K_2 \times T + V_1)u \right]_T^t = \left[ K_2 \times \frac{(u-T)^2}{2} \right]_T^t + V_1(t - T) \\
 &= \frac{K_2}{2} (t - T)^2 + V_1(t - T) \\
 &\Leftrightarrow \boxed{s(t) = \frac{K_2}{2} (t - T)^2 + V_1(t - T) + \frac{1}{2} K_1 T^2}
 \end{aligned}$$

- $2T \leq t \leq 3T$ 
  - $\boxed{\ddot{s}(t) = -K_2}$
  - $\int_{2T}^t \ddot{s}(u) \times du = \dot{s}(t) - \dot{s}(2T) = \dot{s}(t) - V_2$   
 $= \int_{2T}^t -K_2 \times du = -K_2(t - 2T)$   
 $\Leftrightarrow \boxed{\dot{s}(t) = -K_2(t - 2T) + V_2}$
  - $\int_{2T}^t \dot{s}(u) \times du = s(t) - s(2T) = s(t) - \left( \frac{T^2}{2} (K_1 + K_2) + V_1 T \right)$   
 $= \int_{2T}^t (-K_2(u - 2T) + V_2) \times du = \left[ -K_2 \times \frac{(u-2T)^2}{2} \right]_{2T}^t + V_2(t - 2T)$   
 $= -\frac{K_2}{2} (t - 2T)^2 + V_2(t - 2T)$   
 $\Leftrightarrow \boxed{s(t) = -\frac{K_2}{2} (t - 2T)^2 + V_2(t - 2T) + \frac{T^2}{2} (K_1 + K_2) + V_1 T}$
- $3T \leq t \leq 4T$ 
  - $\boxed{\ddot{s}(t) = -K_1}$
  - $\int_{3T}^t \ddot{s}(u) \times du = \dot{s}(t) - \dot{s}(3T) = \dot{s}(t) - V_1$   
 $= \int_{3T}^t -K_1 \times du = -K_1(t - 3T)$   
 $\Leftrightarrow \boxed{\dot{s}(t) = -K_1(t - 3T) + V_1}$
  - $\int_{3T}^t \dot{s}(u) \times du = s(t) - s(3T) = s(t) - \left( \frac{T^2}{2} K_1 + (V_1 + V_2) T \right)$   
 $= \int_{3T}^t (-K_1(u - 3T) + V_1) \times du = \left[ -K_1 \times \frac{(u-3T)^2}{2} \right]_{3T}^t + V_1(t - 3T)$   
 $= -\frac{K_1}{2} (t - 3T)^2 + V_1(t - 3T)$   
 $\Leftrightarrow \boxed{s(t) = -\frac{K_1}{2} (t - 3T)^2 + V_1(t - 3T) + \frac{T^2}{2} K_1 + (V_1 + V_2) T}$

#### 2.2.4 Equations de $X(t)$ , $\dot{X}(t)$

- $X(t)$ :

	$0 \leq t \leq T$	$T \leq t \leq 2T$	$2T \leq t \leq 3T$	$3T \leq t \leq 4T$
$x(t)$	$\frac{x_B - x_A}{d_{AB}} \cdot K_1$ $\cdot t^2 + x_A$	$\frac{x_B - x_A}{d_{AB}} \cdot \left( \frac{K_2}{2} (t - T)^2 \right.$ $+ V_1(t - T)$ $\left. + \frac{1}{2} K_1 T^2 \right)$ $+ x_A$	$\frac{x_B - x_A}{d_{AB}} \cdot \left( -\frac{K_2}{2} (t - 2T)^2 \right.$ $+ V_2(t - 2T)$ $\left. + \frac{T^2}{2} (K_1 + K_2) + V_1 T \right)$ $+ x_A$	$\frac{x_B - x_A}{d_{AB}} \cdot \left( -\frac{K_1}{2} (t - 3T)^2 \right.$ $+ V_1(t - 3T)$ $\left. + \frac{T^2}{2} K_1 + (V_1 + V_2) T \right)$ $+ x_A$
$y(t)$	$\frac{y_B - y_A}{d_{AB}} \cdot K_1$ $\cdot t^2 + y_A$	$\frac{y_B - y_A}{d_{AB}} \cdot \left( \frac{K_2}{2} (t - T)^2 \right.$ $+ V_1(t - T)$ $\left. + \frac{1}{2} K_1 T^2 \right)$ $+ y_A$	$\frac{y_B - y_A}{d_{AB}} \cdot \left( -\frac{K_2}{2} (t - 2T)^2 \right.$ $+ V_2(t - 2T)$ $\left. + \frac{T^2}{2} (K_1 + K_2) + V_1 T \right)$ $+ y_A$	$\frac{y_B - y_A}{d_{AB}} \cdot \left( -\frac{K_1}{2} (t - 3T)^2 \right.$ $+ V_1(t - 3T)$ $\left. + \frac{T^2}{2} K_1 + (V_1 + V_2) T \right)$ $+ y_A$
$z(t)$	$\frac{z_B - z_A}{d_{AB}} \cdot K_1$ $\cdot t^2 + z_A$	$\frac{z_B - z_A}{d_{AB}} \cdot \left( \frac{K_2}{2} (t - T)^2 \right.$ $+ V_1(t - T)$ $\left. + \frac{1}{2} K_1 T^2 \right) + z_A$	$\frac{z_B - z_A}{d_{AB}} \cdot \left( -\frac{K_2}{2} (t - 2T)^2 \right.$ $+ V_2(t - 2T)$ $\left. + \frac{T^2}{2} (K_1 + K_2) + V_1 T \right)$ $+ z_A$	$\frac{z_B - z_A}{d_{AB}} \cdot \left( -\frac{K_1}{2} (t - 3T)^2 \right.$ $+ V_1(t - 3T)$ $\left. + \frac{T^2}{2} K_1 + (V_1 + V_2) T \right)$ $+ z_A$



- $\dot{X}(t)$ :

	$0 \leq t \leq T$	$T \leq t \leq 2T$	$2T \leq t \leq 3T$	$3T \leq t \leq 4T$
$\dot{x}(t)$	$\frac{x_B - x_A}{d_{AB}} \cdot K_1 \cdot t + x_A$	$\frac{x_B - x_A}{d_{AB}} \cdot (K_2(t - T) + V_1) + x_A$	$\frac{x_B - x_A}{d_{AB}} \cdot (-K_2(t - 2T) + V_2) + x_A$	$\frac{x_B - x_A}{d_{AB}} \cdot (-K_1(t - 3T) + V_1) + x_A$
$\dot{y}(t)$	$\frac{y_B - y_A}{d_{AB}} \cdot K_1 \cdot t + y_A$	$\frac{y_B - y_A}{d_{AB}} \cdot (K_2(t - T) + V_1) + y_A$	$\frac{y_B - y_A}{d_{AB}} \cdot (-K_2(t - 2T) + V_2) + y_A$	$\frac{y_B - y_A}{d_{AB}} \cdot (-K_1(t - 3T) + V_1) + y_A$
$\dot{z}(t)$	$\frac{z_B - z_A}{d_{AB}} \cdot K_1 \cdot t + z_A$	$\frac{z_B - z_A}{d_{AB}} \cdot (K_2(t - T) + V_1) + z_A$	$\frac{z_B - z_A}{d_{AB}} \cdot (-K_2(t - 2T) + V_2) + z_A$	$\frac{z_B - z_A}{d_{AB}} \cdot (-K_1(t - 3T) + V_1) + z_A$

## 2.3 Primitive de mouvement

### 2.3.1 Explication sur la réalisation de la primitive `traj(A, B, V1, V2)`

La fonction `traj(A, B, V1, V2)` réalise le calcul de la trajectoire du robot entre deux points **A** et **B** dans l'espace opérationnel. Cette trajectoire est générée en respectant les contraintes de vitesses maximales **V1** et **V2** pour les deux segments de mouvement, et la fonction retourne les données nécessaires pour visualiser et analyser la trajectoire dans différentes dimensions.

Explications détaillées sur la réalisation de la primitive `traj(A, B, V1, V2)` :

Point clé	Description
<b>Initialisation des paramètres</b>	Si les paramètres <b>A</b> , <b>B</b> , <b>V1</b> , et <b>V2</b> ne sont pas fournis, la fonction utilise les valeurs par défaut définies dans l'objet UR3.  Calcul de la distance <b>dAB</b> entre les points <b>A</b> et <b>B</b> .
<b>Calcul des paramètres temporels</b>	Calcul de la durée totale du mouvement ( <b>T</b> ) en utilisant les vitesses maximales <b>V1</b> et <b>V2</b> .  Calcul des coefficients <b>K1</b> et <b>K2</b> utilisés pour générer les lois de mouvement.
<b>Calcul des lois de mouvement</b>	Utilisation de la fonction <code>inspace</code> pour générer une séquence de temps <b>t</b> allant de 0 à 4 fois la durée totale <b>T</b> .  Utilisation de conditions pour définir les lois de mouvement ( <b>s0</b> , <b>s1</b> , <b>s2</b> ) en fonction du temps.  Ces lois de mouvement décrivent la position, la vitesse, et l'accélération du robot le long de la trajectoire.
<b>Calcul de la trajectoire opérationnelle</b>	Utilisation des lois de mouvement pour calculer la position dans l'espace opérationnel ( <b>xt</b> , <b>yt</b> , <b>zt</b> ) à chaque instant de temps.  Ces positions décrivent la trajectoire du robot entre les points <b>A</b> et <b>B</b> dans l'espace opérationnel.
<b>Calcul des coordonnées dans l'espace de travail et des trajectoires des joints</b>	Utilisation des positions de l'outil pour calculer les coordonnées dans l'espace de travail ( <b>x0</b> , <b>y0</b> , <b>z0</b> ) au fil du temps.  Utilisation des lois de mouvement pour calculer les vitesses ( <b>x1</b> , <b>y1</b> , <b>z1</b> ) et les accélérations ( <b>x2</b> , <b>y2</b> , <b>z2</b> ) de l'outil dans l'espace de travail.  Calcul de la vitesse de l'outil ( <b>endEffector_speed</b> ) au fil du temps.

	Utilisation de la cinématique inverse (MGI et MDI) pour calculer les trajectoires des joints ( $q_0$ , $q_1$ ) correspondant aux positions de l'outil.
<b>Retour des données</b>	La méthode renvoie les trajectoires des joints calculées au fil du temps ( $q_0$ , $q_1$ ).

En résumé, la méthode `traj` génère une trajectoire planifiée pour le robot RRR en utilisant des équations de mouvement et en respectant les vitesses initiales et finales. Elle calcule également les positions et vitesses de l'outil dans l'espace de travail, ainsi que les trajectoires des joints correspondantes.

### 2.3.2 Particularités de notre projet

Point clé	Description
<b>Installation / Vérification des dépendances logicielles</b>	Nous avons inclus des commandes <code>os.system</code> pour installer les dépendances logicielles ( <code>numpy</code> et <code>matplotlib</code> ) afin de garantir le fonctionnement du programme peu importe le poste Linux utilisé
<b>Approche orientée objet</b>	Pour favoriser la modularité de notre solution, favoriser sa réutilisation et sa maintenance nous avons structuré notre code en utilisant l'approche orientée objet.
<b>Méthodes orientés utilisateur et paramètres optionnels</b>	Notre code permet la modification des paramètres du robot, tels que les constantes géométriques et les paramètres de trajectoire, à la fois via des méthodes d'entrée utilisateur ( <code>input_</code> ) et des méthodes de modification directe ( <code>set_</code> ). De plus nous avons également inclus des méthodes de lecture ( <code>read_</code> ) afin de pouvoir vérifier l'état des paramètres à n'importe quel instant.
<b>Code optimisé</b>	Nous avons tenté d'optimiser un maximum le programme en : <ul style="list-style-type: none"><li>• Limitant le nombre de ligne et le nombre de variables sans compromettre la lisibilité</li><li>• Evitant les bouts de programmes répétitifs</li><li>• Affectant des valeurs par défaut et des typages formatés</li><li>• Mettant en place des sécurités pour contrecarrer toutes erreurs éventuelles dans la saisie de l'utilisateur</li></ul>
<b>Documentation et code en anglais</b>	Nous codé notre programme et avons rédigé notre documentation entièrement en anglais afin de toucher l'ensemble de la communauté scientifique.

#### Pour conclure notre programme est :

- Fonctionnel
- Ergonomique (facile d'utilisation, modulaire, réutilisable)
- Bien optimisé
- Documenté en anglais