

# Método Clássico de Denavit-Hartenberg Aplicado ao KUKA LBR iiwa

A. C. Pereira<sup>1</sup>, B. Bresolini<sup>1</sup>, F. H. N. Resende<sup>1</sup>, I. A. Bastos<sup>1</sup>,  
K. Geamonoud<sup>1</sup>, M. O. Coelho<sup>1</sup>, O. G. B. de O. e Oliveira<sup>1</sup>,  
T. O. Campagnani<sup>1</sup>

<sup>1</sup>Graduandos em Eng. Mecatrônica  
Departamento de Eng. Mecatrônica, CEFET-MG

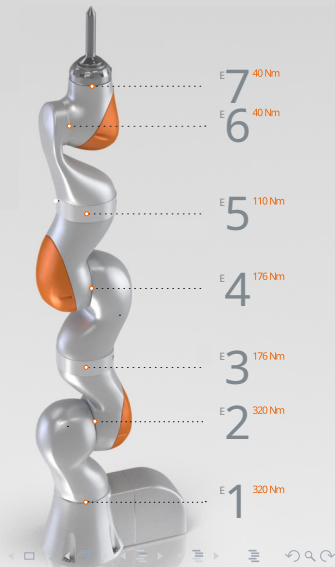
Divinópolis, Setembro 2020

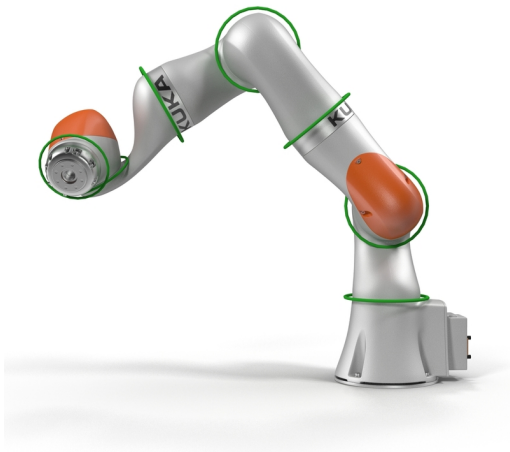
- 1 Apresentação
- 2 Modelagem
  - Matriz de Transformação
- 3 Validação

- 1 Apresentação
- 2 Modelagem
  - Matriz de Transformação
- 3 Validação

# KUKA LBR iiwa

O robô KUKA LBR iiwa é conhecido como novo antropomorfo e contém 7 graus de liberdade. Existem dois padrões vendidos: LBR iiwa 7 800 e **LBR iiwa 14 820**. O primeiro com carga efetiva de até 7 kg e o segundo com 14 kg.



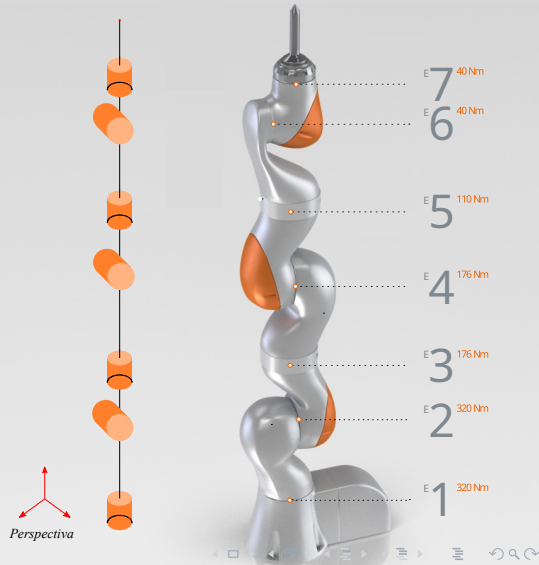


Sua estrutura cinemática é do tipo esférica-rotacional-esférica (SRS) e seu volume de trabalho é uma esfera de 1,7 a 1,8 m<sup>3</sup>.

- 1 Apresentação
- 2 Modelagem
  - Matriz de Transformação
- 3 Validação

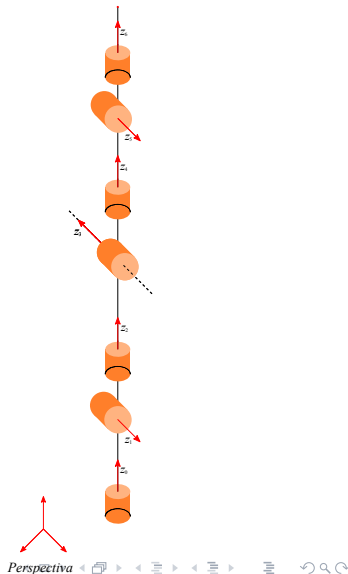
# Diagrama de arames

O primeiro passo para a aplicação do método clássico de Denavit-Hartenberg é desenhar o diagrama de arames para o manipulador robótico.



# Atribuição de frames

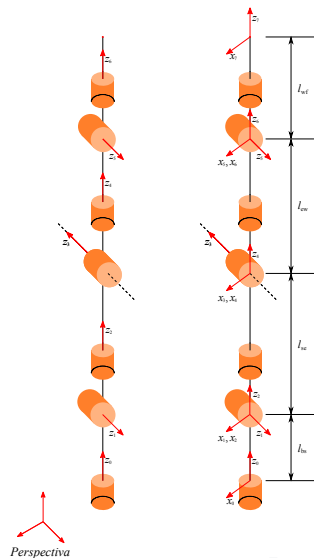
Inicialmente, deve-se atribuir os eixos  $z_i$  nos eixos de atuação, para cada junta  $i + 1$ , sendo  $i = 0, 1, 2 \dots, n - 1$ . A última junta é o **end-effector** ou o **tool frame** e, pode ser atribuído ao final.





# Atribuição de frames

De acordo com SPONG, HUTCHINSON e VIDYA-SAGAR (2005, p. 74), se  $z_{i-1}$  **intercepta**  $z_i$ ,  $x_i$  é escolhido no plano normal formado por  $z_i$  e  $z_{i-1}$ . O sentido positivo de  $x_i$  é arbitrária e a origem  $o_i$  pode ser atribuído no ponto de interseção de  $z_i$  e  $z_{i-1}$ . Se isto for feito, o parâmetro  $a_i$  será igual a 0.



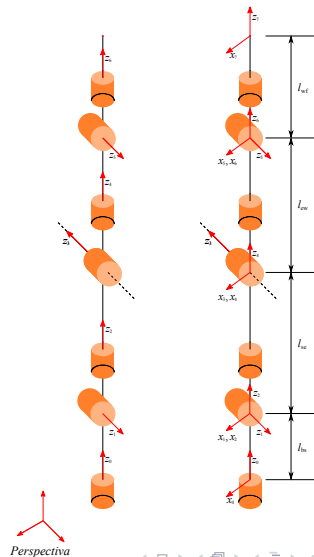
# Tabela de Denavit-Hartenberg

Os parâmetros dos *links* do KUKA LBR iiwa podem ser retirados da sua interpretação física da atribuição de frames.

Tabela: Parâmetros DH do KUKA LBR iiwa

$i$	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	0	$-\pi/2$	$l_{bs}$	$\theta_1^*$
2	0	$\pi/2$	0	$\theta_2^*$
3	0	$\pi/2$	$l_{se}$	$\theta_3^*$
4	0	$-\pi/2$	0	$\theta_4^*$
5	0	$-\pi/2$	$l_{ew}$	$\theta_5^*$
6	0	$\pi/2$	0	$\theta_6^*$
7	0	0	$l_{wf}$	$\theta_7^*$

O \* indica que é variável.



# Matriz de Transformação Homogênea

Usando os parâmetros obtidos na tabela, em cada linha é obtida a matriz de transformação homogênea  $A_i$  do *link*  $i$  para o *link*  $i - 1$ , fazendo

$$A_i = \text{Rot}_{Z, \theta_i} \text{Trans}_{Z, d_i} \text{Trans}_{X, a_i} \text{Rot}_{X, \alpha_i}$$
$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ \hline 0 & 0 & 0 & 1 \end{bmatrix}$$

A matriz de transformação  $T$  de  $n$  para 0 é então determinada por

$$T_n^0 = \prod_{i=0}^n A_i$$

# Matriz de Transformação Homogênea

Para o *link* 1, segue  $a_1 = 0$ ,  $\alpha_1 = -\pi/2$ ,  $d_1 = \ell_{bs}$  e  $\theta_1$  sendo a variável, então

$$A_1 = \left[ \begin{array}{ccc|c} c_{\theta_1} & 0 & -s_{\theta_1} & 0 \\ s_{\theta_1} & 0 & c_{\theta_1} & 0 \\ 0 & -1 & 0 & \ell_{bs} \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

Para o *link* 2, segue  $a_2 = 0$ ,  $\alpha_2 = \pi/2$ ,  $d_2 = 0$  e  $\theta_2$  sendo a variável, então

$$A_2 = \left[ \begin{array}{ccc|c} c_{\theta_2} & 0 & s_{\theta_2} & 0 \\ s_{\theta_2} & 0 & -c_{\theta_2} & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

Para o *link* 3, segue  $a_3 = 0$ ,  $\alpha_3 = \pi/2$ ,  $d_3 = \ell_{se}$  e  $\theta_3$  sendo a variável, então

$$A_3 = \left[ \begin{array}{ccc|c} c_{\theta_3} & 0 & s_{\theta_3} & 0 \\ s_{\theta_3} & 0 & -c_{\theta_3} & 0 \\ 0 & 1 & 0 & \ell_{se} \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

Para o *link* 4, segue  $a_4 = 0$ ,  $\alpha_4 = -\pi/2$ ,  $d_4 = 0$  e  $\theta_4$  sendo a variável, então

$$A_4 = \left[ \begin{array}{ccc|c} c_{\theta_4} & 0 & -s_{\theta_4} & 0 \\ s_{\theta_4} & 0 & c_{\theta_4} & 0 \\ 0 & -1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

# Matriz de Transformação Homogênea

Para o *link* 5, segue  $a_5 = 0$ ,  $\alpha_5 = -\pi/2$ ,  $d_5 = \ell_{ew}$  e  $\theta_5$  sendo a variável, então

$$A_5 = \left[ \begin{array}{ccc|c} c_{\theta_5} & 0 & -s_{\theta_5} & 0 \\ s_{\theta_5} & 0 & c_{\theta_5} & 0 \\ 0 & -1 & 0 & \ell_{ew} \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

Para o *link* 6, segue  $a_6 = 0$ ,  $\alpha_6 = \pi/2$ ,  $d_6 = 0$  e  $\theta_6$  sendo a variável, então

$$A_6 = \left[ \begin{array}{ccc|c} c_{\theta_6} & 0 & s_{\theta_6} & 0 \\ s_{\theta_6} & 0 & -c_{\theta_6} & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

Para o *link* 7,  $a_7 = 0$ ,  $\alpha_7 = 0$ ,  $d_7 = \ell_{wf}$  e sendo  $\theta_7$  a variável, segue

$$A_7 = \left[ \begin{array}{ccc|c} c_{\theta_7} & -s_{\theta_7} & 0 & 0 \\ s_{\theta_7} & c_{\theta_7} & 0 & 0 \\ 0 & 0 & 1 & \ell_{wf} \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

# Matriz de Transformação Homogênea

Por fim, a matriz de transformação homogênea do frame 7 para o frame 0 é encontrada fazendo

$$T_7^0 = A_1 A_2 A_3 A_4 A_5 A_6 A_7 \quad (1)$$

uma multiplicação de 7 matrizes  $4 \times 4$  de variáveis reais.

# Matriz de Transformação

A multiplicação das matrizes  $n$  matrizes de transformação homogênea resulta em

$$T_n^0 = \left[ \begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{22} & r_{33} & d_z \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (2)$$

Para a qual os termos serão expostos a seguir.

$$r_{11} = \left[ \left[ ((-s_1 s_3 + c_1 c_2 c_3) c_4 + s_2 s_4 c_1) c_5 + (-s_1 c_3 - s_3 c_1 c_2) s_5 \right] c_6 + \right. \\ \left. ((-s_1 s_3 + c_1 c_2 c_3) s_4 - s_2 c_1 c_4) s_6 \right] c_7 + \left[ - \left[ (-s_1 s_3 + c_1 c_2 c_3) c_4 + \right. \right. \\ \left. \left. s_2 s_4 c_1 \right] s_5 + (-s_1 c_3 - s_3 c_1 c_2) c_5 \right] s_7$$

$$r_{12} = - \left[ \left[ ((-s_1 s_3 + c_1 c_2 c_3) c_4 + s_2 s_4 c_1) c_5 + (-s_1 c_3 - s_3 c_1 c_2) s_5 \right] c_6 \right. \\ \left. + ((-s_1 s_3 + c_1 c_2 c_3) s_4 - s_2 c_1 c_4) s_6 \right] s_7 + \left[ - \left[ (-s_1 s_3 + c_1 c_2 c_3) c_4 \right. \right. \\ \left. \left. + s_2 s_4 c_1 \right] s_5 + (-s_1 c_3 - s_3 c_1 c_2) c_5 \right] c_7$$



$$r_{13} = \left[ ((-s_1 s_3 + c_1 c_2 c_3) c_4 + s_2 s_4 c_1) c_5 + (-s_1 c_3 - s_3 c_1 c_2) s_5 \right] s_6 \\ - ((-s_1 s_3 + c_1 c_2 c_3) s_4 - s_2 c_1 c_4) c_6$$

$$r_{21} = \left[ \left[ ((s_1 c_2 c_3 + s_3 c_1) c_4 + s_1 s_2 s_4) c_5 + (-s_1 s_3 c_2 + c_1 c_3) s_5 \right] c_6 \right. \\ \left. + ((s_1 c_2 c_3 + s_3 c_1) s_4 - s_1 s_2 c_4) s_6 \right] c_7 + \left[ - \left[ (s_1 c_2 c_3 + s_3 c_1) c_4 \right. \right. \\ \left. \left. + s_1 s_2 s_4 \right] s_5 + (-s_1 s_3 c_2 + c_1 c_3) c_5 \right] s_7$$

$$r_{22} = - \left[ \left[ \left( (s_1 c_2 c_3 + s_3 c_1) c_4 + s_1 s_2 s_4 \right) c_5 + (-s_1 s_3 c_2 + c_1 c_3) s_5 \right] c_6 \right. \\ \left. + \left( (s_1 c_2 c_3 + s_3 c_1) s_4 - s_1 s_2 c_4 \right) s_6 \right] s_7 + \left[ - \left[ (s_1 c_2 c_3 + s_3 c_1) c_4 \right. \right. \\ \left. \left. + s_1 s_2 s_4 \right] s_5 + (-s_1 s_3 c_2 + c_1 c_3) c_5 \right] c_7$$

$$r_{23} = \left[ \left[ (s_1 c_2 c_3 + s_3 c_1) c_4 + s_1 s_2 s_4 \right] c_5 + (-s_1 s_3 c_2 + c_1 c_3) s_5 \right] s_6 \\ - \left[ (s_1 c_2 c_3 + s_3 c_1) s_4 - s_1 s_2 c_4 \right] c_6$$

$$r_{31} = \left[ ((-s_2 c_3 c_4 + s_4 c_2) c_5 + s_2 s_3 s_5) c_6 + (-s_2 s_4 c_3 - c_2 c_4) s_6 \right] c_7 \\ + \left[ -(-s_2 c_3 c_4 + s_4 c_2) s_5 + s_2 s_3 c_5 \right] s_7$$

$$r_{32} = - \left[ ((-s_2 c_3 c_4 + s_4 c_2) c_5 + s_2 s_3 s_5) c_6 + (-s_2 s_4 c_3 - c_2 c_4) s_6 \right] s_7 \\ + \left[ -(-s_2 c_3 c_4 + s_4 c_2) s_5 + s_2 s_3 c_5 \right] c_7$$

$$r_{33} = \left[ (-s_2 c_3 c_4 + s_4 c_2) c_5 + s_2 s_3 s_5 \right] s_6 - (-s_2 s_4 c_3 - c_2 c_4) c_6$$

$$\begin{aligned}d_x = 90 \bigg[ ((-s_1 s_3 + c_1 c_2 c_3) c_4 + s_2 s_4 c_1) c_5 + (-s_1 c_3 - s_3 c_1 c_2) s_5 \bigg] s_6 \\ - 90 \bigg[ (-s_1 s_3 + c_1 c_2 c_3) s_4 - s_2 c_1 c_4 \bigg] c_6 - 400 (-s_1 s_3 + c_1 c_2 c_3) s_4 \\ + 400 s_2 c_1 c_4 + 420 s_2 c_1\end{aligned}$$

$$\begin{aligned}d_y = 90 \bigg[ ((s_1 c_2 c_3 + s_3 c_1) c_4 + s_1 s_2 s_4) c_5 + (-s_1 s_3 c_2 + c_1 c_3) s_5 \bigg] s_6 \\ - 90 \bigg[ (s_1 c_2 c_3 + s_3 c_1) s_4 - s_1 s_2 c_4 \bigg] c_6 - 400 (s_1 c_2 c_3 + s_3 c_1) s_4 \\ + 400 s_1 s_2 c_4 + 420 s_1 s_2\end{aligned}$$

$$\begin{aligned}d_z = 90 \bigg[ (-s_2 c_3 c_4 + s_4 c_2) c_5 + s_2 s_3 s_5 \bigg] s_6 - 90 (-s_2 s_4 c_3 - c_2 c_4) c_6 \\ + 400 s_2 s_4 c_3 + 400 c_2 c_4 + 420 c_2 + 360\end{aligned}$$

- 1 Apresentação
- 2 Modelagem
  - Matriz de Transformação
- 3 Validação

# Implementação: Set Up

```
1 import sympy as sp      # Pacote para manipulações simbólicas
2 from sympy import pi, sin, cos
3 from sympy.physics.vector import init_vprinting      # Usada para
  ↪  imprimir em latex
4 from sympy.physics.mechanics import dynamicsymbols
5
6 init_vprinting(use_latex='mathjax', pretty_print=False)      #
  ↪  Configurando o vprinting
7
8 theta1, theta2, theta3, theta4 = dynamicsymbols('theta1 theta2
  ↪  theta3 theta4')
9 theta5, theta6, theta7 = dynamicsymbols('theta5 theta6 theta7')
10 l1, l2, theta, alpha, a, d = dynamicsymbols('l1 l2 theta alpha a d')
```

# Implementação: Tabela DH

```
1  # Valores das distancias (LBR iiwa 14 820)
2  dbs, dse, dew, dwf = 360, 420, 400, 90
3
4  # Matriz DH:          alfi, ai, di, thetai
5  TDH = sp.Matrix([ [-pi/2, 0, dbs, theta1],
6                    [ pi/2, 0,   0, theta2],
7                    [ pi/2, 0, dse, theta3],
8                    [-pi/2, 0,   0, theta4],
9                    [-pi/2, 0, dew, theta5],
10                   [ pi/2, 0,   0, theta6],
11                   [   0, 0, dwf, theta7]])
```

# Implementação: $A_i$

```
1 Rot = sp.Matrix([
2     [cos(theta), -sin(theta)*cos(alpha), sin(theta)*sin(alpha)],
3     [sin(theta), cos(theta)*cos(alpha), -cos(theta)*sin(alpha)],
4     [ 0, sin(alpha), cos(alpha)]
5         ])
6
7 Tran = sp.Matrix([a*cos(theta), a*sin(theta), d])
8 S = sp.Matrix([[0, 0, 0, 1]])
9
10 # Matriz Ai geral
11 A = sp.Matrix.vstack(sp.Matrix.hstack(Rot, Tran), S)
```



# Implementação: MTH $T_7^0$

```
1  # Matriz A de cada linha
2  A1 = A.subs({alpha:TDH[0,0],a:TDH[0,1],d:TDH[0,2],theta:TDH[0,3]})
3  A2 = A.subs({alpha:TDH[1,0],a:TDH[1,1],d:TDH[1,2],theta:TDH[1,3] })
4  A3 = A.subs({alpha:TDH[2,0],a:TDH[2,1],d:TDH[2,2],theta:TDH[2,3] })
5  A4 = A.subs({alpha:TDH[3,0],a:TDH[3,1],d:TDH[3,2],theta:TDH[3,3] })
6  A5 = A.subs({alpha:TDH[4,0],a:TDH[4,1],d:TDH[4,2],theta:TDH[4,3] })
7  A6 = A.subs({alpha:TDH[5,0],a:TDH[5,1],d:TDH[5,2],theta:TDH[5,3] })
8  A7 = A.subs({alpha:TDH[6,0],a:TDH[6,1],d:TDH[6,2],theta:TDH[6,3]})
9
10 # Matriz de transf. homogenea de 7 para 0
11 T = A1*A2*A3*A4*A5*A6*A7
```

```
1 # Posição Home
2 Thome = T.subs({ theta1:0, theta2:0,
   ↪ theta3:0, theta4:0, theta5:0,
   ↪ theta6:0, theta7:0 })
3
4 # Posição qualquer
5 q1, q2, q3      = 0, 0, 0
6 q4, q5, q6, q7 = -pi/2, pi/3, 0, 0
7
8 T_ = T.subs({ theta1:q1, theta2:q2,
   ↪ theta3:q3, theta4:q4, theta5:q5,
   ↪ theta6:q6, theta7:q7 })
```

Ao qual retorna

$$T_{\text{home}} = \left[ \begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1270 \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

e

$$T_- = \left[ \begin{array}{ccc|c} 0 & 0 & 1 & 490 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 & 780 \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

# Simulação $T_{home}$

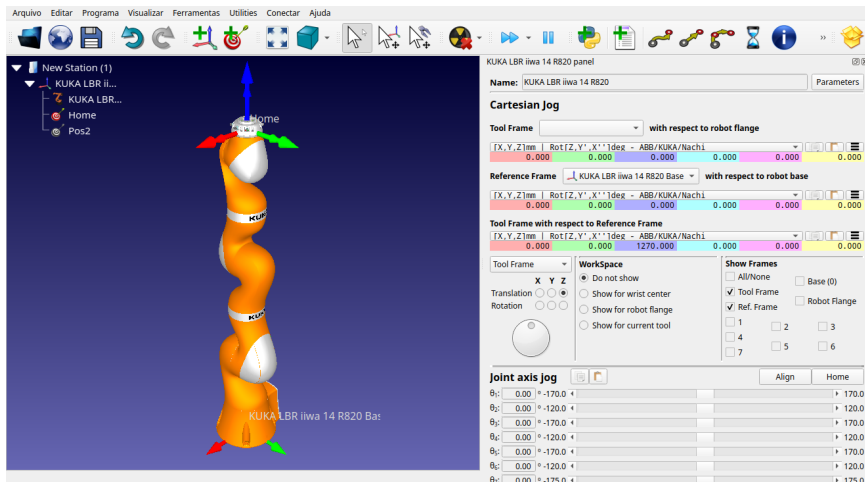


Figura:  $T_{home}$  resultante do RoboDK

# Simulação T<sub>1</sub>

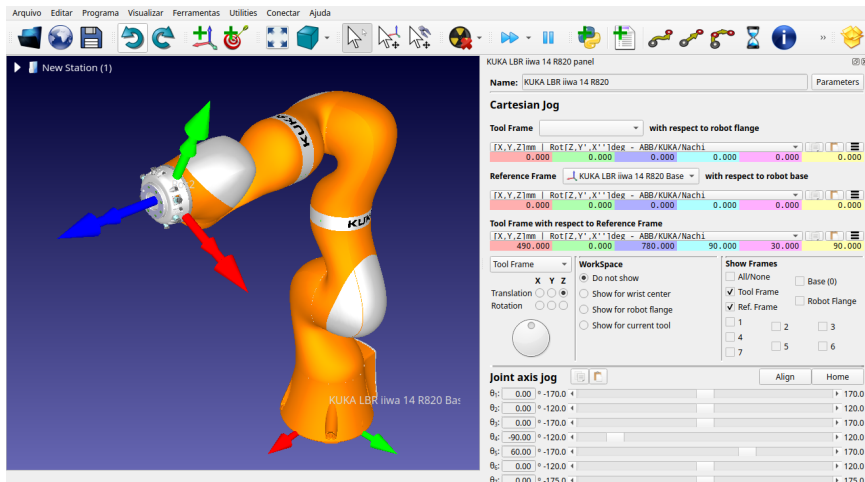


Figura: T<sub>1</sub> resultante do RoboDK

# Comparação do software com a matriz $T_{\text{home}}$

Para a posição de *home* tem-se

$$T_{\text{home}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1270 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Isto é, uma translação em  $z_0$  de 1270 mm e uma rotação de  $0^\circ$ . Como retorna a simulação no RoboDK.



## Comparação do software com a matriz T

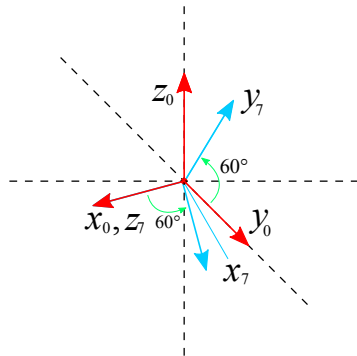
A matriz de rotação é então calculada fazendo

$$R_7^0 = \begin{bmatrix} x_7 \cdot x_0 & y_7 \cdot x_0 & z_7 \cdot x_0 \\ x_7 \cdot y_0 & y_7 \cdot y_0 & z_7 \cdot y_0 \\ x_7 \cdot z_0 & y_7 \cdot z_0 & z_7 \cdot z_0 \end{bmatrix}$$

que resulta em

$$R_7^0 = \begin{bmatrix} 0 & 0 & 1 \\ \cos 60^\circ & \sin 60^\circ & 0 \\ -\sin 60^\circ & \cos 60^\circ & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 1 \\ \sqrt{3}/2 & 1/2 & 0 \\ -1/2 & \sqrt{3}/2 & 0 \end{bmatrix}$$



(3) **Figura:** Diagrama de arames para a configuração escolhida

# Comparação do software com a matriz $T_-$

No RoboDK, a pose do *end-effector* é retornada com a translação nos eixos, mas a rotação em  $z$ ,  $y'$  e  $x''$ . Isto é, a composição de rotações no frame atual para os valores indicados. Para a configuração escolhida em  $T_-$ , são dados

$$R_7^0 = \text{Rot}_{z, 90^\circ} \text{Rot}_{y, 30^\circ} \text{Rot}_{x, 90^\circ}$$

Aplicando os valores

$$R = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{3}/2 & 0 & 1/2 \\ 0 & 1 & 0 \\ -1/2 & 0 & \sqrt{3}/2 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

que resultam em

$$R = \begin{bmatrix} 0 & 0 & 1 \\ \sqrt{3}/2 & 1/2 & 0 \\ -1/2 & \sqrt{3}/2 & 0 \end{bmatrix} \quad (4)$$