

Modelagem Robô KUKA LBR iiwa

Adeilson C. Pereira *, B. Bresolini *, Felipe H. N. Resende *,
Ivna A. Bastos *, Kassianey Geamonoud *,
Marcela O. Coelho *, Otto G. B. de O. e Oliveira *,
Thalles O. Campagnani *

* Centro Federal de Educação Tecnológica de Minas Gerais,
Divinópolis - MG.

1. INTRODUÇÃO

Em tempos de transformação digital e indústria 4.0, a robótica tem ganhado cada vez mais espaço no setor industrial. Nesse âmbito tem-se como protagonistas os robôs manipuladores que atuam sob diversas frentes nos processos de produção, tais como, por exemplo, movimentação de cargas, soldagem, pintura em *spray*, esmerilhamento e afins (Spong et al., 2008, Cap. 1).

Para se realizar o controle dos robôs manipuladores deve-se obter modelos dinâmicos completos e precisos para então, com base nessas representações matemáticas, desenvolver o controle tanto do movimento livre quanto da interação com o ambiente (Gaz and De Luca, 2017). Para se modelar um robô o primeiro passo é obter a relação entre suas partes, isto é, estabelecer o quanto e como a alteração de uma variável de entrada afeta a posição e orientação do último link (“*end effector*”). A esse processo de descrever a pose da ferramenta do robô como uma função das variáveis das juntas do mesmo, damos o nome de Cinemática Direta dos Robôs (Siciliano et al., 2010), ou como nomeia Mihelj et al. (2018): Modelo Geométrico do Robô.

Uma vez constatada a ampla aplicação desses robôs manipuladores, torna-se imediatamente necessário o estudo e compreensão do funcionamento dos mesmos por profissionais atuantes na área de tecnologia. Nesse contexto, foi proposto aos alunos da disciplina de Dinâmica de Robôs do curso de Engenharia Mecatrônica do CEFET-MG um estudo sobre modelagem de robôs manipuladores, especificamente, a respeito do robô KUKA LBR iiwa.

Desse modo, o presente trabalho busca descrever os procedimentos realizados para obtenção de um modelo geométrico para o robô KUKA LBR iiwa. Além dessa introdução, este texto é composto por mais quatro seções: Descrição do Robô, Modelagem do Robô pelo Método DH, Validação do Modelo, e Simulação.

2. DESCRIÇÃO DO ROBÔ

O manipulador KUKA LBR iiwa é um braço robótico com 7 graus de liberdade projetado para trabalhar em conjunto com humanos de forma segura no mesmo espaço de trabalho. De acordo com o fabricante, humanos e robôs podem trabalhar juntos em tarefas altamente sensíveis em estreita cooperação, o que abre a possibilidade de novas aplicações para a robótica (KUKA, 2017a). Isso é possível devido ao fato de que o robô apresenta sensores de torque

em todos do eixos de rotação, e assim consegue se mover com precisão e detectar o contato com humanos e objetos.

O propósito a que se destina o manipulador KUKA é declarado na sua nomenclatura. Enquanto que LBR se refere a palavra alemã “*Leichtbauroboter*” cuja tradução literal é robô leve; o acrônimo iiwa vem do inglês “*intelligent industrial work assistant*” equivalente a assistente de trabalho industrial inteligente, no português.

O robô em questão é classificado como um antropomorfo não compensado, visto que sua estrutura cinemática é da forma esférica-rotacional-esférica (SRS), similar ao braço humano (base, ombro, cotovelo e punho), como observa-se na FIG. 1.



Figura 1. Robô KUKA LBR iiwa
Fonte: KUKA, p. 22, 2017b.

O KUKA LBR iiwa está disponível no mercado em duas versões, com capacidades de carga útil de 7 e 14 kg e apresentam alcance máximo de 800 mm e 820 mm, respectivamente. Na FIG. 2 é possível verificar o espaço de trabalho e os parâmetros das dimensões do robô,

enquanto na TAB. 1 é informado seus valores para ambos os modelos. Mais informações no Apêndice A.

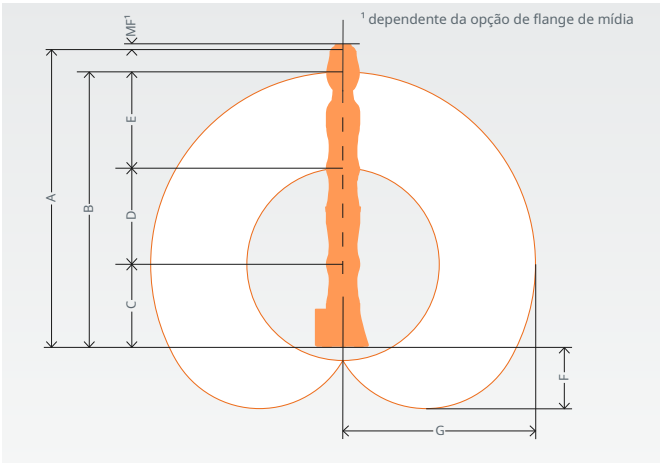


Figura 2. Dimensões do KUKA LBR iiwa
Fonte: KUKA, p. 30, 2017a. (Adaptação de cores e tradução).

Tabela 1. Dimensões, em milímetros, dos parâmetros da FIG. 2

Robô	A	B	C	D	E	F	G
7 R800	1,266	1,140	340	400	400	260	800
14 R820	1,306	1,180	360	420	400	255	820

2.1 Aplicações e Estudos de Casos

Existem diversas áreas de aplicação em que o manipulador KUKA LBR é utilizado. Devido às suas características funcionais e sua adaptabilidade em trabalhar ao lado do fator humano, o robô é utilizado, principalmente, em indústrias de tecnologia e de produção em massa. As principais tarefas delegadas ao manipulador são aquelas de repetição monótona, como em funções de pintar, colar, aplicar, paletizar, embalar, medir e testar. Há também a utilização do robô na área de usinagem mecânica e montagem de peças.

Estudos de caso são apresentados na página web da fabricante do robô na qual algumas aplicações se destacam. Um desses estudos que merecem ser ressaltados é a solução desenvolvida pela fabricante KUKA para a montadora de veículos BMW em uma instalação localizada em Dingolfing, Alemanha.

Em uma determinada etapa do processo de fabricação de veículos, os funcionários do Grupo BMW precisavam montar caixas de engrenagens de eixos dianteiros. Estes carregavam e levantavam peças de até 5 kg, de difícil acesso, com tolerâncias milimétricas de ajuste. A solução desenvolvida foi utilizar o robô KUKA LBR para montagem das engrenagens. O manipulador trabalha de forma suspensa e realiza o trabalho “pesado”, poupando o esforço físico dos funcionários.

A garra foi projetada com revestimentos arredondados, protegendo assim os operadores de ferimentos. Além disso, sensores externos não são necessários devido à presença dos sensores de torque nas juntas que fornecem segurança aos

operadores. Na FIG. 3 pode-se observar uma foto do robô operando na aplicação descrita.



Figura 3. Instalação de peças em superfícies complexas de um monobloco da BMW
Fonte: KUKA, p. 28, 2017a.

Outro estudo de caso a ser analisado é a utilização do manipulador pela fabricante de equipamentos eletrônicos Siemens. No âmbito da fabricação de estatores, a Siemens, na unidade de Bad Neustadt, também na Alemanha, procurava por uma solução flexível para automatizar a atividade simples de entrega e posicionamento de peças, que até então era feita manualmente.

A solução criada foi o desenvolvimento de uma célula flexível com o robô de construção leve KUKA LBR iiwa. Assim, o robô, assume a função de retirar a peça a ser processada — o estator composto de um corpo básico de chapa elétrica e placa de alumínio — do porta-peças e encaminhá-la a um torno que fará a sua usinagem. Após este processo, o robô apanha as peças, faz uma limpeza por sopro e as destina à estação de medição.

Outros estudos de caso, onde a implementação do robô simplificou os processos tecnológicos, são de empresas como a Ford, Volkswagen e Perschmann Calibration GmbH.

Nas indústrias da Ford, vários robôs foram instalados para aplicar um cordão de vedação em diversas peças. Eles



Figura 4. Carregamento automático de um torno CNC na Siemens

Fonte: KUKA, p. 21, 2017a.

sempre aplicam o material exatamente na mesma posição e na quantidade definida. Assim é possível obter uma boa vedação na carroceria e evitar um retrabalho na pintura, eliminando custos.

Já na montadora alemã, Volkswagen, o robô KUKA LBR iiwa participa de processos de montagem, colagem e soldagem. Trabalhando ao lado do fator humano, o robô realiza suas funções com precisão e eficiência, como é mostrado na FIG 5.

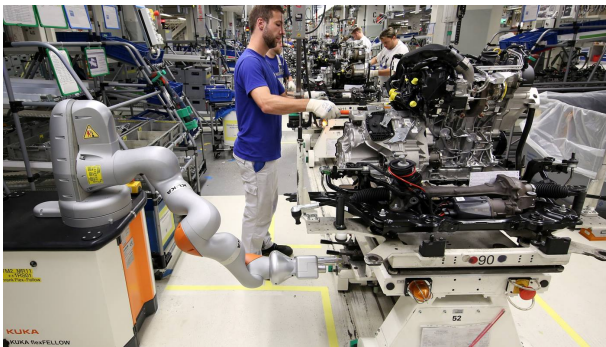


Figura 5. Montagem de peças na linha de produção

Fonte: KUKA, p. 21, 2017a.

A empresa Perschmann Calibration GmbH, especialista em serviços de calibração, utiliza o manipulador como ferramenta de medição. De acordo com o Dr. Detlef Rübesame, diretor Técnico da Perschmann Calibration, um cabelo humano tem aproximadamente 50 μm de espessura, os fios de uma aranha, cerca de 5 μm , e a precisão com a qual o robô faz sua calibração é de cerca de 0,5 μm , um valor extremamente importante para manter a qualidade de seus serviços.

Estes, e outros estudos de caso, apenas corroboram o quão vantajoso é a utilização do robô KUKA LBR iiwa.

3. MODELAGEM DO ROBÔ PELO MÉTODO DH

Existem diversas formas de se obter o modelo de um robô, tais como Denavit-Hartenberg clássico, Denavit-Hartenberg modificado e através das matrizes de transformação homogêneas de cada um dos *frames* no robô. Nesse trabalho o método escolhido para se obter a expressão matemática foi Denavit-Hartenberg clássico.

3.1 Diagrama de arames

O primeiro passo para a aplicação da técnica escolhida é fazer uma simplificação do manipulador robótico através da representação em diagrama de arames. Este deve ser feito com o robô em posição de *home* conforme pode-se verificar na FIG. 6.

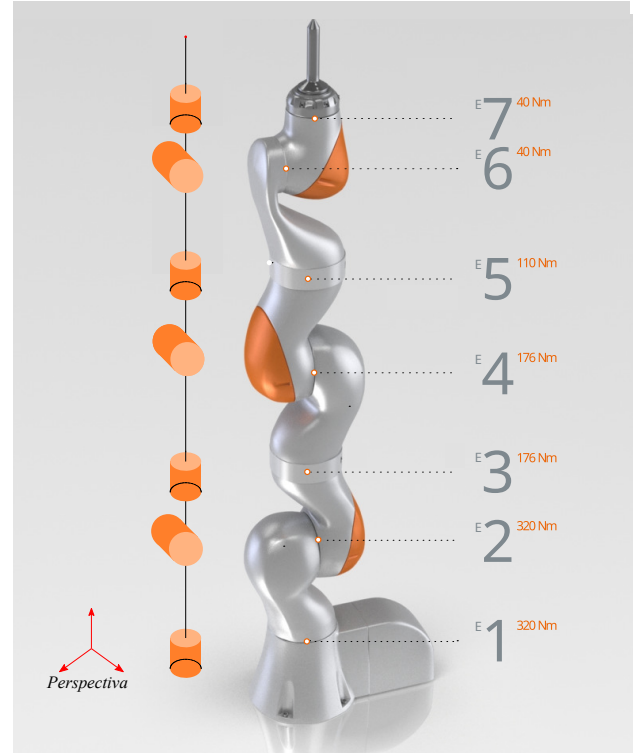


Figura 6. Posição *home* do manipulador.

Fonte: Zarvandi, 2020. (Desenho dos frames e indicação dos eixos e seus torques).

3.2 Atribuição de z

Uma vez que o diagrama de arames foi definido então pode-se aplicar a convenção de Denavit-Hartenberg. Para determinar os frames do robô, deve-se atribuir os eixos z_i nos eixos de atuação para cada junta $i + 1$, sendo $i = 0, 1, 2, \dots, n - 1$, em que n é o número de juntas do manipulador. O eixo z_n , referente ao frame do *end-effector* ou *tool frame*, deve ser atribuído ao final. Após atribuir os eixos z_i , então pode-se identificar a origem dos frames i , em que $i = 1, 2, \dots, n - 1$ adotando a seguinte regra:

- Se z_{i-1} interceptar z_i , então a origem do frame i deve ser posicionada na interseção.
- Se não, mas se existe uma perpendicular comum a z_{i-1} e z_i , então a origem fica nessa reta perpendicular comum.
- Se nenhuma das opções anteriores são verdadeiras, então z_{i-1} e z_i são paralelos e assim a origem do sistema de coordenadas deve ser posicionada de forma que o modelo fique mais simples.

Para o robô aqui estudado, pôde-se adotar a primeira regra para todos os *frames*. Isso porque z_{i-1} e z_i se interceptam para todo $i = 1, 2, \dots, n - 1$.

3.3 Atribuição de x

Com os eixos z_i e origens determinados, então estabelece-se os eixos x_i , com $i = 1, 2, \dots, n-1$, considerando a seguinte regra:

- Se z_{i-1} interceptar z_i , então x_i deve ser posicionado na direção normal ao plano de z_{i-1} e z_i
- Se não, então determina-se x_i na perpendicular comum de z_{i-1} e z_i a partir da origem do *frame* i .

Para todo $i = 1, 2, \dots, n-1$, o caso do robô KUKA LBR iiwa se encaixou na primeira hipótese da atribuição dos eixos x_i .

3.4 Atribuição de y

Os eixos y_i foram escolhidos implicitamente de forma que todos os sistemas de coordenadas fossem dextrogiros, como requerido pelo método clássico de Denavit-Hartenberg. Portanto, optou-se por excluí-lo do diagrama da FIG. 7 para deixar mais fácil a sua visualização e obtenção dos parâmetros dos *links*.

3.5 Referência e End-effector

Por fim, foram atribuídos os *frames* 0 e n . Para o **end-effector**, já que o robô não apresenta garra, repetiu-se o *frame* 6, como instruído no método DH clássico. O *frame* $i = 0$ foi atribuído na base do robô. Embora a atribuição não seja a mais simples, visto que se sua origem fosse atribuída junto aos *frames* 1 e 2 reduzir-se-ia uma translação, ela facilitará na comparação dos resultados com a simulação no RoboDK. A FIG. 7 expõe a atribuição de *frames* descrita.

3.6 Tabela de Denavit-Hartenberg

Após dispor todos os *frames*, são determinados os parâmetros da tabela de Denavit-Hartenberg. Em cada uma de suas linhas é possível definir a matriz de transformação homogênea A_i do *frame* $i-1$ para o *frame* i , fazendo

$$A_i = \text{Rot}_{z, \theta_i} \text{Trans}_{z, d_i} \text{Trans}_{x, a_i} \text{Rot}_{x, \alpha_i}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

em que a_i é a distância entre z_{i-1} e z_i medido ao longo de x_i ; α_i o ângulo entre z_{i-1} e z_i mensurado no plano normal a x_i ; d_i a distância perpendicular entre a origem o_{i-1} e a interseção de x_i com z_{i-1} , medido ao longo de z_0 ; θ_i o ângulo entre x_0 e x_1 mensurado no plano normal a z_0 .

Uma vez obtidas as matrizes A_i para $i = 1, 2, \dots, n$, então pode-se determinar a matriz de transformação T do *frame* 0 para o *frame* n como

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

Para o braço robótico KUKA LBR iiwa, os parâmetros da tabela DH podem ser retirados da sua interpretação física da atribuição de *frames* vista na FIG. 7. A TAB. 2 expõe os valores obtidos.

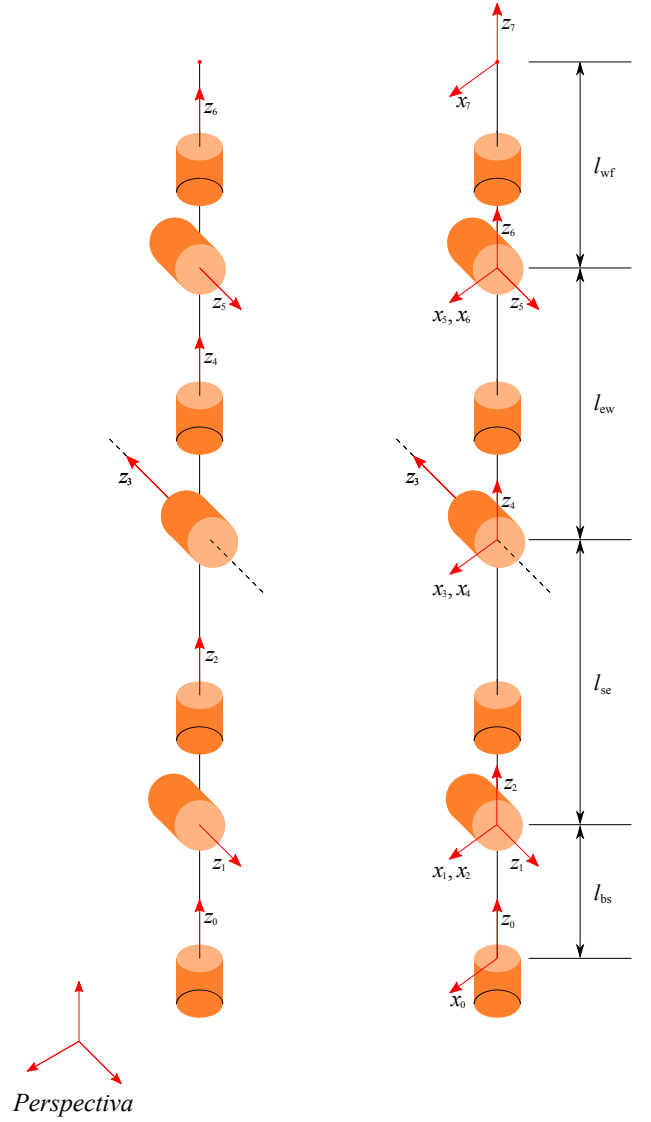


Figura 7. Atribuição dos *frames*

Tabela 2. Parâmetros DH do KUKA LBR iiwa

i	a_i	α_i	d_i	θ_i
1	0	$-\pi/2$	ℓ_{bs}	θ_1^*
2	0	$\pi/2$	0	θ_2^*
3	0	$\pi/2$	ℓ_{se}	θ_3^*
4	0	$-\pi/2$	0	θ_4^*
5	0	$-\pi/2$	ℓ_{ew}	θ_5^*
6	0	$\pi/2$	0	θ_6^*
7	0	0	ℓ_{wf}	θ_7^*

O * indica que é variável.

Para o *link* 1, segue $a_1 = 0$, $\alpha_1 = -\pi/2$, $d_1 = \ell_{bs}$ e θ_1 sendo a variável, então

$$A_1 = \begin{bmatrix} c_{\theta_1} & 0 & -s_{\theta_1} & 0 \\ s_{\theta_1} & 0 & c_{\theta_1} & 0 \\ 0 & -1 & 0 & \ell_{bs} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para o *link* 2, segue $a_2 = 0$, $\alpha_2 = \pi/2$, $d_2 = 0$ e θ_2 sendo a variável, então

$$A_2 = \begin{bmatrix} c_{\theta_2} & 0 & s_{\theta_2} & 0 \\ s_{\theta_2} & 0 & -c_{\theta_2} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para o *link* 3, segue $a_3 = 0$, $\alpha_3 = \pi/2$, $d_3 = \ell_{se}$ e θ_3 sendo a variável, então

$$A_3 = \begin{bmatrix} c_{\theta_3} & 0 & s_{\theta_3} & 0 \\ s_{\theta_3} & 0 & -c_{\theta_3} & 0 \\ 0 & 1 & 0 & \ell_{se} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para o *link* 4, segue $a_4 = 0$, $\alpha_4 = -\pi/2$, $d_4 = 0$ e θ_4 sendo a variável, então

$$A_4 = \begin{bmatrix} c_{\theta_4} & 0 & -s_{\theta_4} & 0 \\ s_{\theta_4} & 0 & c_{\theta_4} & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para o *link* 5, segue $a_5 = 0$, $\alpha_5 = -\pi/2$, $d_5 = \ell_{ew}$ e θ_5 sendo a variável, então

$$A_5 = \begin{bmatrix} c_{\theta_5} & 0 & -s_{\theta_5} & 0 \\ s_{\theta_5} & 0 & c_{\theta_5} & 0 \\ 0 & -1 & 0 & \ell_{ew} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para o *link* 6, segue $a_6 = 0$, $\alpha_6 = \pi/2$, $d_6 = 0$ e θ_6 sendo a variável, então

$$A_6 = \begin{bmatrix} c_{\theta_6} & 0 & s_{\theta_6} & 0 \\ s_{\theta_6} & 0 & -c_{\theta_6} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para o *link* 7, $a_7 = 0$, $\alpha_7 = 0$, $d_7 = \ell_{wf}$ e sendo θ_7 a variável, segue

$$A_7 = \begin{bmatrix} c_{\theta_7} & -s_{\theta_7} & 0 & 0 \\ s_{\theta_7} & c_{\theta_7} & 0 & 0 \\ 0 & 0 & 1 & \ell_{wf} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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 4×4 de variáveis reais. A solução dessa multiplicação, matriz de transformação T_7^0 , pode ser verificada no Apêndice B. Ela foi obtida via simulação computacional, realizada por meio de um código em Python 3, exposto no Apêndice C.

4. VALIDAÇÃO DO MODELO

Tendo encontrado o modelo cinemático, representado por T_7^0 , é importante realizar sua validação. Isto é, verificar a coerência entre a representação matemática e o robô real. Isso pode ser feito através de análise dimensional ou por simulação computacional. No primeiro caso, é aplicado ao modelo um conjunto de valores de entrada, ou seja, são aplicados na matriz T_7^0 um conjunto de valores para os ângulos das juntas. Assim, obtém-se uma matriz que representa a pose do *end-effector* em relação ao *frame* 0 que deve corresponder ao diagrama de arames desenhado para a entrada de dados aplicada ao modelo. É recomendado que esse procedimento seja feito para duas entradas diferentes.

Nesse trabalho aplicou-se um conjunto de ângulos que representam a posição *home*, $\theta_i = 0$ para $i = 1, \dots, 7$ e outro conjunto para uma posição escolhida dada por $\theta_1 = \theta_2 = \theta_3 = \theta_6 = \theta_7 = 0$, $\theta_4 = -\pi/2$ e $\theta_5 = \pi/3$. Assim, aplicando os ângulos das juntas nas equações apresentadas no Apêndice B, foram obtidas as matrizes T_{home} e $T_{escolhido}$.

$$T_{home} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1270 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$T_{escolhido} = \begin{bmatrix} 0 & 0 & 1 & 490 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 & 780 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

A matriz T_{home} confirma as duas características da pose do robô em *home*. A primeira é a translação que se concentra apenas no eixo z cujo valor é a soma das extensões de todos os *links*. A segunda característica se refere à rotação, a qual percebe-se ser de valor zero nos três eixos devido à matriz de rotação estar expressa como uma matriz identidade. A representação em diagrama de arames desta condição confirma as características supracitadas e pode ser visto na FIG. 8.

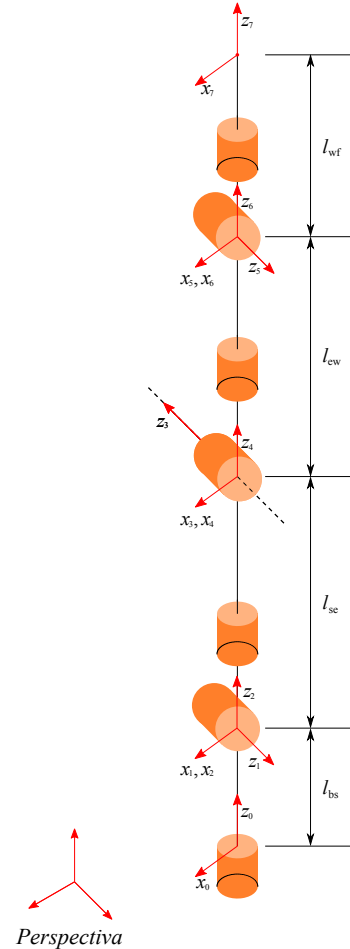


Figura 8. Diagrama esquemático da posição *home*

Já para a matriz $T_{\text{escolhida}}$, tem-se rotação de -90° em θ_4 , o que distribui as translações nos eixos z_0 , com valor de 780 mm representando a soma dos *links* bs e se ou do *frame* zero ao quatro, e o restante da extensão do robô, 490 mm, fica direcionado paralelamente ao eixo x_0 .

Após a curvatura de -90° em θ_4 , tem-se a rotação do *link* 5 que faz com que o restante do braço do robô rotacione. O *frame* final se dá portanto pelo resultado de uma composição de duas rotações e das translações relacionadas acima relação ao *frame* zero.

O diagrama de arames mostrado na FIG. 9 mostra os passos tomados para a avaliação desta nova pose. Em (a) o *frame* 4 é rotacionado em -90° ; em (b) é feita a composição total do robô para tal rotação. Para (c) é mostrada a rotação do eixo 5 devido a rotação em θ_5 e (d) é mostrado a configuração final do robô.

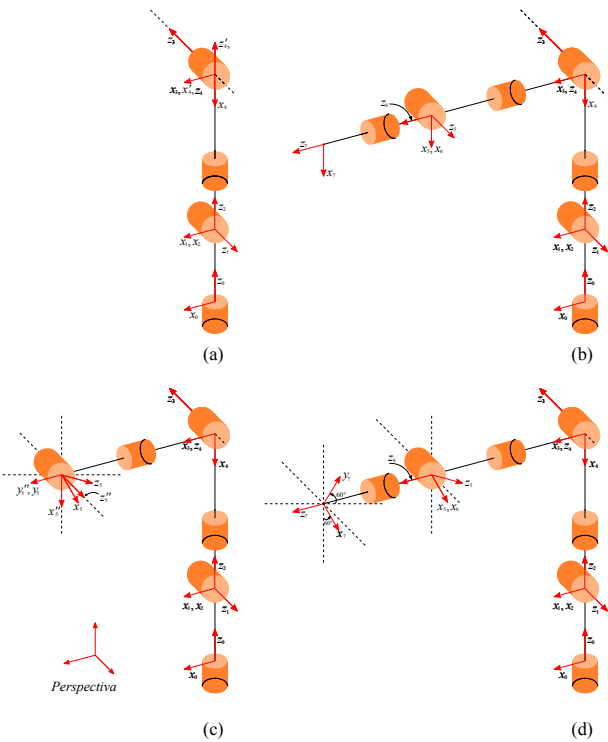


Figura 9. Passo-a-passo da representação esquemática da posição escolhida

Note que a rotação do *end-effector* pode ser analisado resolvendo

$$R_7^0 = \begin{bmatrix} x_7^0 & y_7^0 & z_7^0 \end{bmatrix} = \begin{bmatrix} x_7 \cdot x_0 & y_7 \cdot x_0 & z_7 \cdot x_0 \\ x_7 \cdot y_0 & y_7 \cdot y_0 & z_7 \cdot y_0 \\ x_7 \cdot z_0 & y_7 \cdot z_0 & z_7 \cdot z_0 \end{bmatrix} \quad (4)$$

Nota-se inicialmente que $z_7 \cdot x_0 = 1$ então $x_7 \cdot x_0 = 0$, $y_7 \cdot x_0 = 0$, $z_7 \cdot y_0 = 0$ e $z_7 \cdot z_0 = 0$, visto que a matriz de rotação é ortonormal.

Fazendo a projeção de x_7 em y_0 segue $\sin 60^\circ = \sqrt{3}/2$. Para x_7 em z_0 , temos $-\cos 60^\circ$. Ainda, repetindo o processo para y_7 , chega-se a $y_7 \cdot y_0 = \cos 60^\circ$ e $y_7 \cdot z_0 = \sin 60^\circ$. Portanto, a matriz que representa a orientação do *frame* 7 é

$$R_7^0 = \begin{bmatrix} 0 & 0 & 1 \\ \sqrt{3}/2 & 1/2 & 0 \\ -1/2 & \sqrt{3}/2 & 0 \end{bmatrix} \quad (5)$$

Comparando (3) com (5), verifica-se que a orientação da matriz de transformação é como se esperava. Destarte, o modelo cinemático do robô KUKA iiwa obtido foi validado.

5. SIMULAÇÃO

A simulação foi realizada no software RoboDK, software offline utilizado para simulação de robôs industriais. Para realização da simulação foram utilizados as posições escolhidas na análise dimensional. Ao se comparar os resultados de T_{home} e $T_{\text{escolhida}}$, é possível comprovar que a matriz T representa o modelo cinemático do robô.

Para comprovar o exposto acima, inicialmente, analisa-se a posição do modelo em *home*, como indicado na FIG. 10.

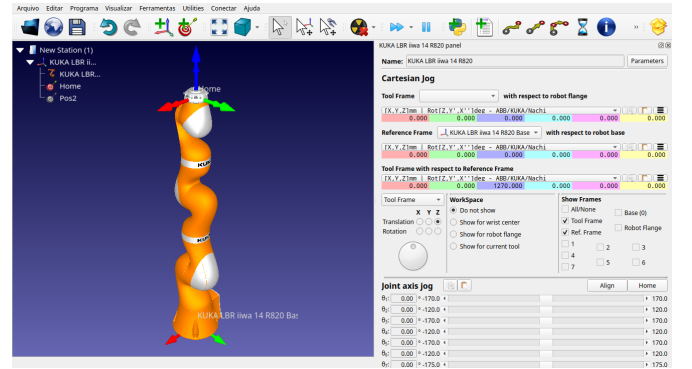


Figura 10. Posição home do robô KUKA LBR iiwa

A análise da FIG. 10 indica que, inicialmente, não há rotações do *frame* zero para o *frame* final, porém existe um único deslocamento no eixo z_0 correspondente ao valor total da soma dos links, sendo este de 127 cm.

na condição seguinte, o robô foi simulado para a posição escolhida, como indicado na FIG. 11.

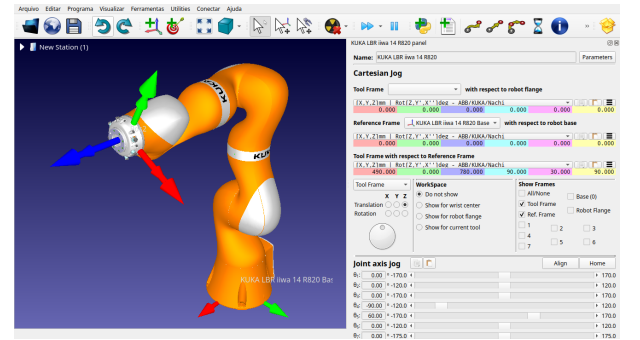


Figura 11. Posição escolhida para o robô KUKA LBR iiwa

É possível observar que, do *frame* zero ao quatro tem-se uma translação de da pose *home* para a escolhida, ocorreu uma rotação de 60° no *link* 5 (θ_5) e de -90° no *link* 4 (θ_4). Além disso, a translação do *frame* zero para o sete passa a ser de 78 cm no eixo z e de 49 cm em x .

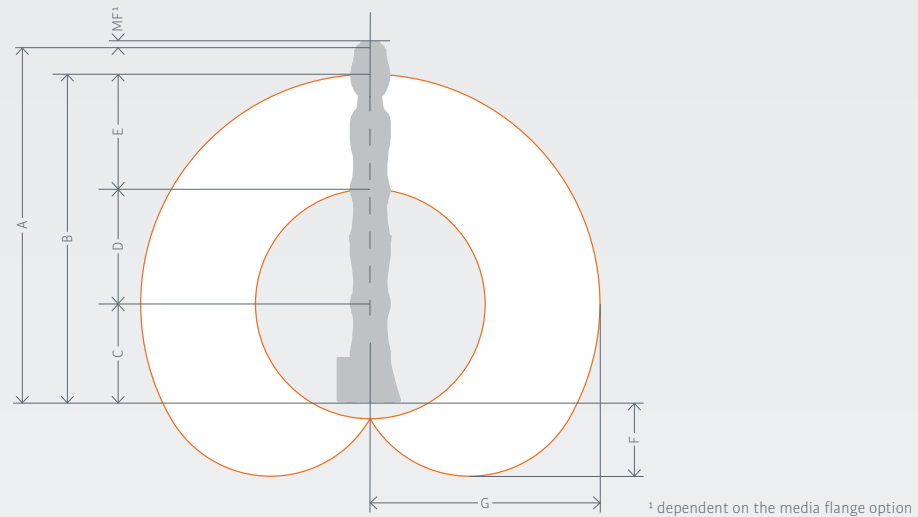
REFERÊNCIAS

Gaz, C. and De Luca, A. (2017). Payload estimation based on identified coefficients of robot dynamics—with an application to collision detection. In *2017 IEEE/RSJ*

- International Conference on Intelligent Robots and Systems (IROS)*, 3033–3040. IEEE.
- KUKA (2017a). *Sensitive robotics: LBR iiwa*. URL <http://www.kuka.com/>.
- KUKA (2017b). *ii feel you*. URL <https://www.kuka.com/>.
- Mihelj, M., Bajd, T., Ude, A., Lenarčič, J., Stanovnik, A., Munih, M., Rejc, J., and Šlajpah, S. (2018). *Robotics*. Springer International Publishing.
- Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2010). *Robotics: modelling, planning and control*. Springer Science & Business Media.
- Spong, M.W., Hutchinson, S., and Vidyasagar, M. (2008). *Robot dynamics and control*. John Wiley & Sons.
- Zarvandi, M. (2020). Modelo 3D KUKA LBR iiwa. URL <https://free3d.com/ko/3d-model/kuka-lbr-iiwa-6625.html>.

Apêndice A. DIMENSÕES DO ROBÔ

Technical data



Workspace	Dimensions A	Dimensions B	Dimensions C	Dimensions D	Dimensions E	Dimensions F	Dimensions G	Volume
LBR iiwa 7 R800	1,266 mm	1,140 mm	340 mm	400 mm	400 mm	260 mm	800 mm	1.7 m ³
LBR iiwa 14 R820	1,306 mm	1,180 mm	360 mm	420 mm	400 mm	255 mm	820 mm	1.8 m ³

LBR iiwa	LBR iiwa 7 R800	LBR iiwa 14 R820
Rated payload	7 kg	14 kg
Number of axes	7	7
Wrist variant	In-line wrist	In-line wrist
Mounting flange A7	DIN ISO 9409-1-A50	DIN ISO 9409-1-A50
Installation position	any	any
Positioning accuracy (ISO 9283)	± 0.1 mm	± 0.1 mm
Axis-specific torque accuracy	± 2 %	± 2 %
Weight	23.9 kg	29.9 kg
Protection rating	IP 54	IP 54

Axis data / Range of motion		LBR iiwa 7 kg		LBR iiwa 14 kg	
		Maximum torque	Maximum velocity	Maximum torque	Maximum velocity
Axis 1 (A1)	± 170°	176 Nm	98°/s	320 Nm	85°/s
Axis 2 (A2)	± 120°	176 Nm	98°/s	320 Nm	85°/s
Axis 3 (A3)	± 170°	110 Nm	100°/s	176 Nm	100°/s
Axis 4 (A4)	± 120°	110 Nm	130°/s	176 Nm	75°/s
Axis 5 (A5)	± 170°	110 Nm	140°/s	110 Nm	130°/s
Axis 6 (A6)	± 120°	40 Nm	180°/s	40 Nm	135°/s
Axis 7 (A7)	± 175°	40 Nm	180°/s	40 Nm	135°/s

Programmable Cartesian stiffness

Min. (X, Y, Z)	0.0 N/m	0.0 N/m
Max. (X, Y, Z)	5,000 N/m	5,000 N/m
Min. (A, B, C)	0.0 N/rad	0.0 N/rad
Max. (A, B, C)	300 Nm/rad	300 Nm/rad

KUKA Sunrise Cabinet

Processor	Quad-core processor
Hard drive	SSD
Interfaces	USB, EtherNet, DVI-I
Protection rating	IP20
Dimensions (D x W x H)	500 mm x 483 mm x 190 mm
Weight	23 kg

Power supply connection

Rated supply voltage	AC 110 V to 230 V
Permissible tolerance of rated voltage	± 10 %
Mains frequency	50 Hz ± 1 Hz or 60 Hz ± 1 Hz
Mains-side fusing	2 x 16 A slow-blowing

30,000 operating hours

Sensitive robotics...LBR iiwa

Apêndice B. MATRIZ DE TRANSFORMAÇÃO T

A multiplicação das matrizes n matrizes de transformação homogênea resulta em

$$T_n^0 = \prod_{i=1}^n A_i = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.1})$$

na qual os termos são

$$r_{11} = \left[\left[((-s_1 s_3 + c_1 c_2 c_3) c_4 + s_2 s_4 c_1) c_5 + (-s_1 c_3 - s_3 c_1 c_2) s_5 \right] c_6 + ((-s_1 s_3 + c_1 c_2 c_3) s_4 - s_2 c_1 c_4) s_6 \right] c_7 + \left[- \left[(-s_1 s_3 + c_1 c_2 c_3) c_4 + s_2 s_4 c_1 \right] s_5 + (-s_1 c_3 - s_3 c_1 c_2) c_5 \right] s_7 \quad (\text{B.2})$$

$$r_{12} = - \left[\left[((-s_1 s_3 + c_1 c_2 c_3) c_4 + s_2 s_4 c_1) c_5 + (-s_1 c_3 - s_3 c_1 c_2) s_5 \right] c_6 + ((-s_1 s_3 + c_1 c_2 c_3) s_4 - s_2 c_1 c_4) s_6 \right] s_7 + \left[- \left[(-s_1 s_3 + c_1 c_2 c_3) c_4 + s_2 s_4 c_1 \right] s_5 + (-s_1 c_3 - s_3 c_1 c_2) c_5 \right] c_7 \quad (\text{B.3})$$

$$r_{13} = \left[((-s_1 s_3 + c_1 c_2 c_3) c_4 + s_2 s_4 c_1) c_5 + (-s_1 c_3 - s_3 c_1 c_2) s_5 \right] s_6 - ((-s_1 s_3 + c_1 c_2 c_3) s_4 - s_2 c_1 c_4) c_6 \quad (\text{B.4})$$

$$r_{21} = \left[\left[((s_1 c_2 c_3 + s_3 c_1) c_4 + s_1 s_2 s_4) c_5 + (-s_1 s_3 c_2 + c_1 c_3) s_5 \right] c_6 + ((s_1 c_2 c_3 + s_3 c_1) s_4 - s_1 s_2 c_4) s_6 \right] c_7 + \left[- \left[(s_1 c_2 c_3 + s_3 c_1) c_4 + s_1 s_2 s_4 \right] s_5 + (-s_1 s_3 c_2 + c_1 c_3) c_5 \right] s_7 \quad (\text{B.5})$$

$$r_{22} = - \left[\left[((s_1 c_2 c_3 + s_3 c_1) c_4 + s_1 s_2 s_4) c_5 + (-s_1 s_3 c_2 + c_1 c_3) s_5 \right] c_6 + ((s_1 c_2 c_3 + s_3 c_1) s_4 - s_1 s_2 c_4) s_6 \right] s_7 + \left[- \left[(s_1 c_2 c_3 + s_3 c_1) c_4 + s_1 s_2 s_4 \right] s_5 + (-s_1 s_3 c_2 + c_1 c_3) c_5 \right] c_7 \quad (\text{B.6})$$

$$r_{23} = \left[\left[(s_1 c_2 c_3 + s_3 c_1) c_4 + s_1 s_2 s_4 \right] c_5 + (-s_1 s_3 c_2 + c_1 c_3) s_5 \right] s_6 - \left[(s_1 c_2 c_3 + s_3 c_1) s_4 - s_1 s_2 c_4 \right] c_6 \quad (\text{B.7})$$

$$r_{31} = \left[((-s_2 c_3 c_4 + s_4 c_2) c_5 + s_2 s_3 s_5) c_6 + (-s_2 s_4 c_3 - c_2 c_4) s_6 \right] c_7 + \left[- (-s_2 c_3 c_4 + s_4 c_2) s_5 + s_2 s_3 c_5 \right] s_7 \quad (\text{B.8})$$

$$r_{32} = - \left[((-s_2 c_3 c_4 + s_4 c_2) c_5 + s_2 s_3 s_5) c_6 + (-s_2 s_4 c_3 - c_2 c_4) s_6 \right] s_7 + \left[- (-s_2 c_3 c_4 + s_4 c_2) s_5 + s_2 s_3 c_5 \right] c_7 \quad (\text{B.9})$$

$$r_{33} = \left[(-s_2 c_3 c_4 + s_4 c_2) c_5 + s_2 s_3 s_5 \right] s_6 - (-s_2 s_4 c_3 - c_2 c_4) c_6 \quad (\text{B.10})$$

$$d_x = 90 \left[((-s_1 s_3 + c_1 c_2 c_3) c_4 + s_2 s_4 c_1) c_5 + (-s_1 c_3 - s_3 c_1 c_2) s_5 \right] s_6 - 90 \left[(-s_1 s_3 + c_1 c_2 c_3) s_4 - s_2 c_1 c_4 \right] c_6 - 400 (-s_1 s_3 + c_1 c_2 c_3) s_4 + 400 s_2 c_1 c_4 + 420 s_2 c_1 \quad (\text{B.11})$$

$$d_y = 90 \left[((s_1 c_2 c_3 + s_3 c_1) c_4 + s_1 s_2 s_4) c_5 + (-s_1 s_3 c_2 + c_1 c_3) s_5 \right] s_6 - 90 \left[(s_1 c_2 c_3 + s_3 c_1) s_4 - s_1 s_2 c_4 \right] c_6 - 400 (s_1 c_2 c_3 + s_3 c_1) s_4 + 400 s_1 s_2 c_4 + 420 s_1 s_2 \quad (\text{B.12})$$

$$d_z = 90 \left[(-s_2 c_3 c_4 + s_4 c_2) c_5 + s_2 s_3 s_5 \right] s_6 - 90 (-s_2 s_4 c_3 - c_2 c_4) c_6 + 400 s_2 s_4 c_3 + 400 c_2 c_4 + 420 c_2 + 360 \quad (\text{B.13})$$

Apêndice C. MEMÓRIA DE CÁLCULO

A memória de cálculo foi realizada via código em Python 3, usando

```
1 import sympy as sp      # Pacote para manipulações simbólicas
2 from sympy import pi, cos, sin
3 from sympy.physics.vector import init_vprinting      # Usada para printar em latex
4 from sympy.physics.mechanics import dynamicsymbols
5
6 init_vprinting(use_latex='mathjax', pretty_print=False)      # Configurando o vprinting
7
8 # Variáveis simbólicas
9 theta1, theta2, theta3, theta4 = dynamicsymbols('theta1 theta2 theta3 theta4')
10 theta5, theta6, theta7 = dynamicsymbols('theta5 theta6 theta7')
11 l1, l2, theta, alpha, a, d = dynamicsymbols('l1 l2 theta alpha a d')
12
13 # Valores das distancias
14 dbs, dse, dew, dwf = 360, 420, 400, 90
15
16 # Matriz DH:      alfi, ai, di, theta
17 TDH = sp.Matrix([ [-pi/2, 0, dbs, theta1],
18                   [ pi/2, 0, 0, theta2],
19                   [ pi/2, 0, dse, theta3],
20                   [-pi/2, 0, 0, theta4],
21                   [-pi/2, 0, dew, theta5],
22                   [ pi/2, 0, 0, theta6],
23                   [ 0, 0, dwf, theta7]])
24
25 # Rotação de theta em z com Rotação de alpha em x
26 Rot = sp.Matrix([[cos(theta), -sin(theta)*cos(alpha), sin(theta)*sin(alpha)],
27                  [sin(theta), cos(theta)*cos(alpha), -cos(theta)*sin(alpha)],
28                  [0, sin(alpha), cos(alpha)]]
29
30 # Translação dos links
31 Tran = sp.Matrix([a*cos(theta), a*sin(theta), d])
32 # Distorção e Escala
33 S = sp.Matrix([[0, 0, 0, 1]])
34
35 # Matriz A genérica
36 A = sp.Matrix.vstack(sp.Matrix.hstack(Rot, Tran), S)
37
38
39 # Matriz A de cada linha
40 A1 = A.subs({ alpha:TDH[0,0], a:TDH[0,1], d:TDH[0,2], theta:TDH[0,3] })
41 A2 = A.subs({ alpha:TDH[1,0], a:TDH[1,1], d:TDH[1,2], theta:TDH[1,3] })
42 A3 = A.subs({ alpha:TDH[2,0], a:TDH[2,1], d:TDH[2,2], theta:TDH[2,3] })
43 A4 = A.subs({ alpha:TDH[3,0], a:TDH[3,1], d:TDH[3,2], theta:TDH[3,3] })
44 A5 = A.subs({ alpha:TDH[4,0], a:TDH[4,1], d:TDH[4,2], theta:TDH[4,3] })
45 A6 = A.subs({ alpha:TDH[5,0], a:TDH[5,1], d:TDH[5,2], theta:TDH[5,3] })
46 A7 = A.subs({ alpha:TDH[6,0], a:TDH[6,1], d:TDH[6,2], theta:TDH[6,3] })
47
48 # Matriz de transformação de 7 (end-effector) para 0
49 T = A1*A2*A3*A4*A5*A6*A7
50
51 # Validação: home (todas as juntas em 0)
52 Thome = T.subs({ theta1:0, theta2:0, theta3:0, theta4:0, theta5:0, theta6:0, theta7:0 })
53
54 # Validação: posição qualquer (comparar o resultado com a interpretação física)
55 q1, q2, q3, q4, q5, q6, q7 = 0, 0, 0, -pi/2, pi/3, 0, 0
56 T_ = T.subs({ theta1:q1, theta2:q2, theta3:q3, theta4:q4, theta5:q5, theta6:q6, theta7:q7 })
```