



INSTITUTO FEDERAL  
ESPIRITO SANTO



# Robótica Industrial

## Engenharia de Controle e Automação – 9º Período

---

PROF. LUCAS VAGO SANTANA  
[lucas@ifes.edu.br](mailto:lucas@ifes.edu.br)



# Aula 07 – Cinemática Inversa e de Velocidade em 3D

---

- Movimento em 3D
- Cinemática Inversa de Posição 3D – Métodos Analíticos
  - Exemplo: Manipulador 2R em 3D
  - Exemplo: Manipulador 3R em 3D (Antropomórfico)
- Cinemática Inversa de Orientação em 3D
  - O punho esférico
  - Desacoplamento Cinemático
- Cinemática Inversa de Posição e Orientação 3D – Métodos Numéricos
  - Arranjo 3R com punho esférico
  - Erros de orientação: como tratar?



# Referências Bibliográficas

---

- CORKE, Peter. **QUT Robot Academy: The open online robotics education resource**. Disponível em: <<https://robotacademy.net.au/>>. Acesso em 27 fev. 2020.
- SCIavicco, Lorenzo; Siciliano, Bruno. **Modelling and Control of Robot Manipulators**. 2. ed. Springer-Verlag Berlin Heidelberg, 2000.
- Sppong, Mark W.; Hutchinson, Seth; Vidyasagar, M. **Robots Modeling and Control**. 1. ed. John Wiley & Sons, 2005.
- CORKE, Peter. **Robotics, Vision and Control: Fundamentals Algorithms in MATLAB**. 2. ed. Springer-Verlag Berlin Heidelberg, 2017.
- NIKU, Saeed B. **Introduction to Robotics: Analysis, Control, Applications**. 2. ed. Springer-Verlag Berlin Heidelberg, 2011.
- LYNCH, Kevin M.; PARK, Frank C. **Modern Robotics: Mechanics, Planning and Control**. 1. ed. Cambridge University Press, 2017.
- SICILIANO, Bruno; KHATIB, Oussama. **Springer Handbook of Robotics**. 2. ed. Springer-Verlag Berlin Heidelberg, 2016.



INSTITUTO FEDERAL  
ESPIRITO SANTO



# Movimento em 3D

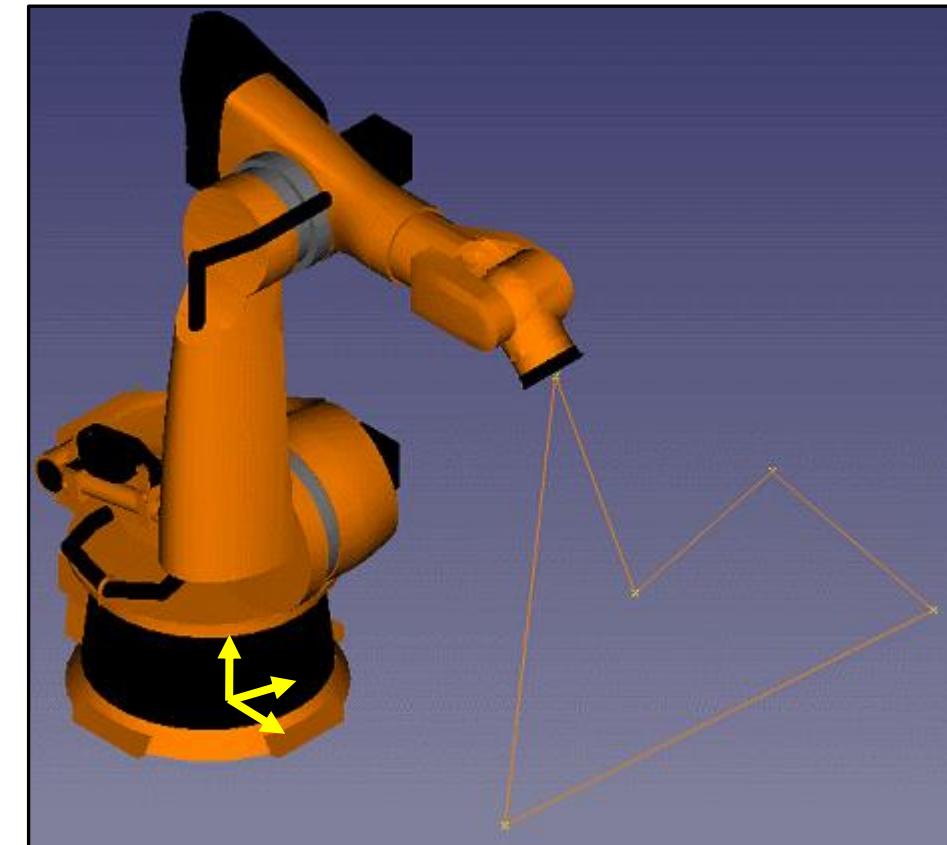


# Movimento em 3D

- Movendo-se em 3D, o efetuador final sofre variações simultâneas nas velocidades linear a angular;
- Em 3D, tal variação se mapeia como:

$$\nu = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad \begin{array}{l} \rightarrow \text{Parte Linear} \\ \rightarrow \text{Parte Rotacional} \end{array}$$

(Twist)



[https://wiki.freecadweb.org/Robot\\_tutorial](https://wiki.freecadweb.org/Robot_tutorial)



---

# Parte 1: Métodos Analíticos

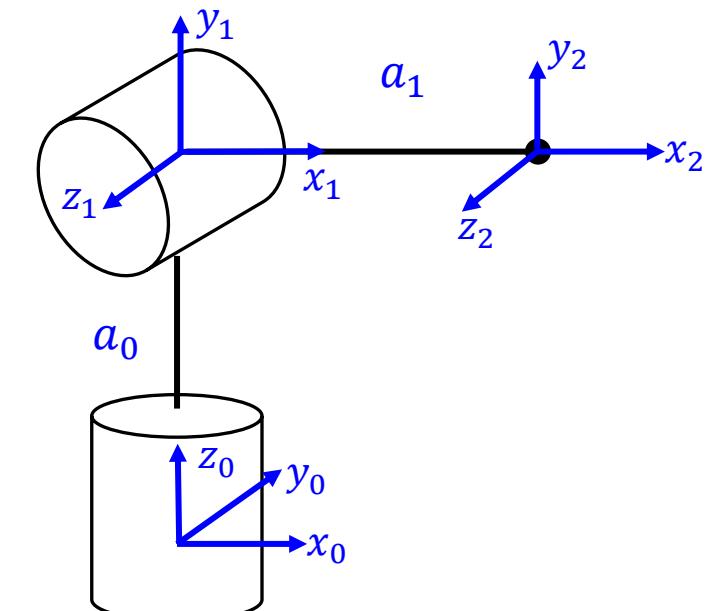
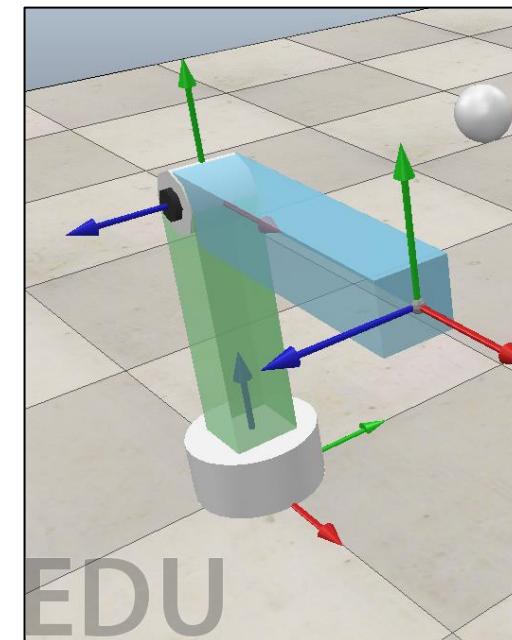


# Cinemática Inversa de Posição 3D



# Exemplo: Manipulador 2R em 3D

- Dado o arranjo do Manipulador 2R em 3D, determine sua cinemática inversa.

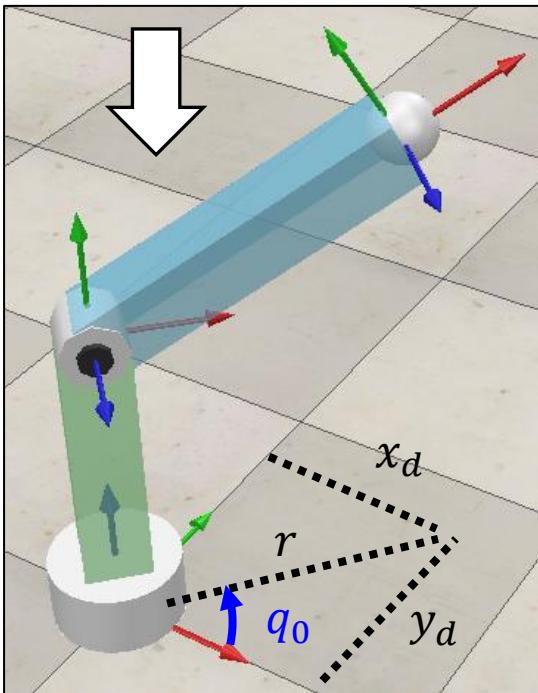




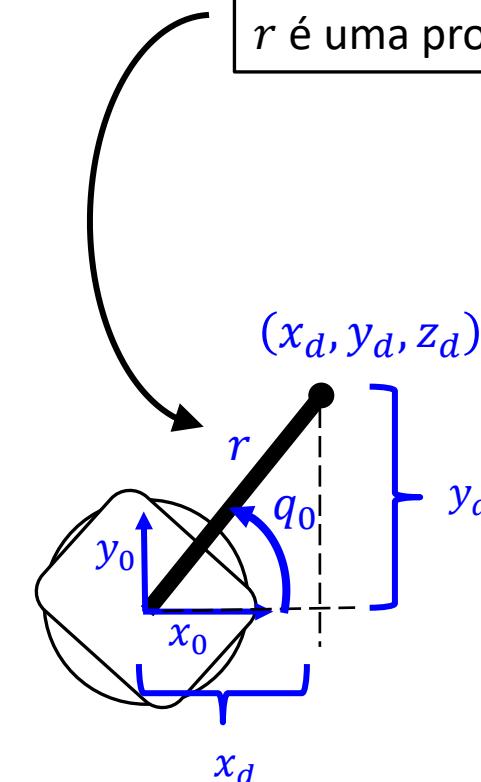
# Exemplo: Manipulador 2R em 3D

- Vista superior ( $q_n \neq 0^\circ$ ):

Olhando o robô dessa perspectiva



Observar que  $r \neq a_1$   
 $r$  é uma projeção de  $a_1$  no plano  $x_0, y_0$

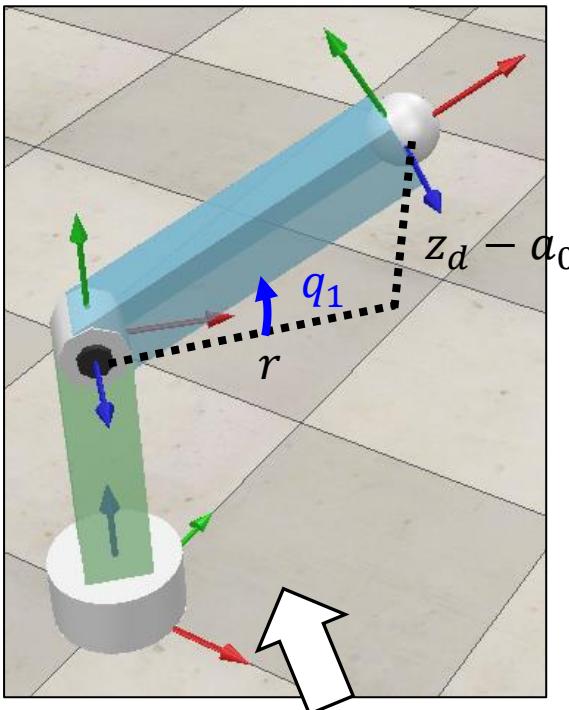


$$q_0 = \tan^{-1} \left( \frac{y_d}{x_d} \right)$$

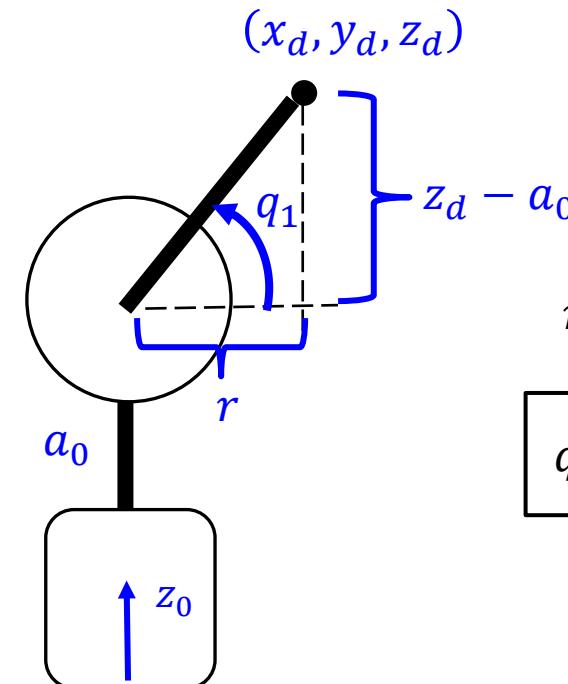


# Exemplo: Manipulador 2R em 3D

- Vista lateral ( $q_n \neq 0^\circ$ ):



Olhando o robô dessa perspectiva.



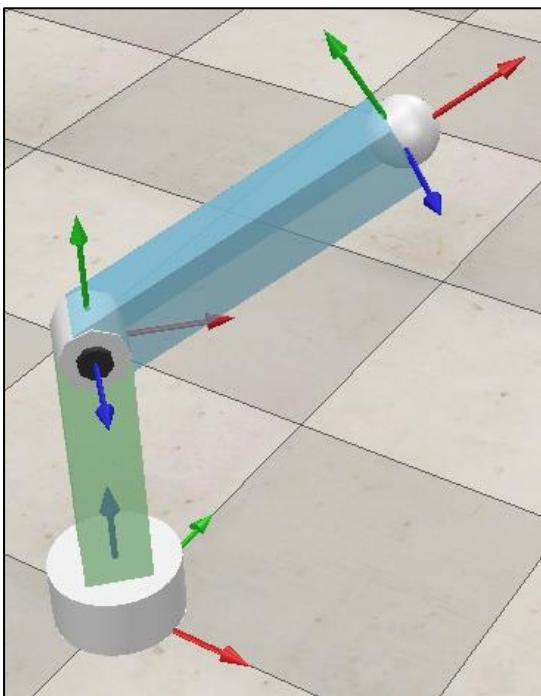
$$r = \sqrt{x_d^2 + y_d^2}$$

$$q_1 = \tan^{-1} \left( \frac{z_d - a_0}{r} \right)$$

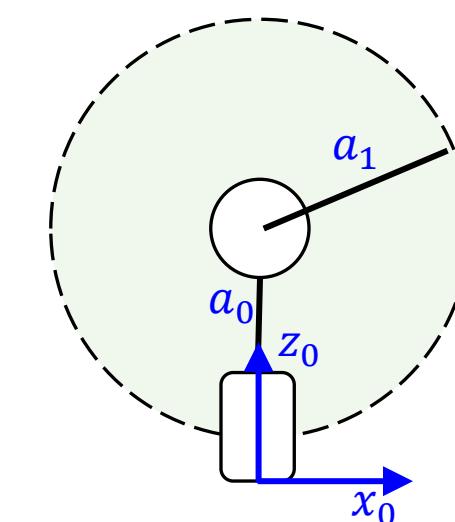


# Exemplo: Manipulador 2R em 3D

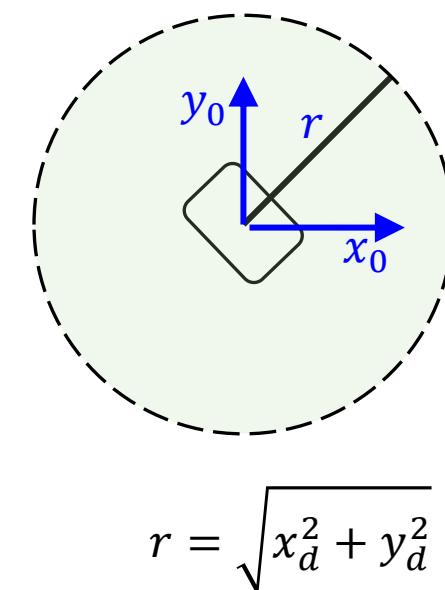
- Análise do espaço de trabalho (desconsidera colisão):



Vista lateral



Vista superior

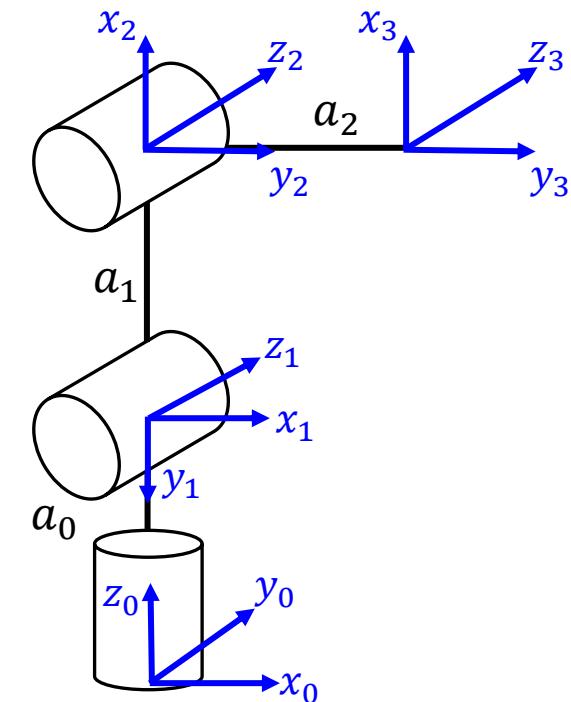
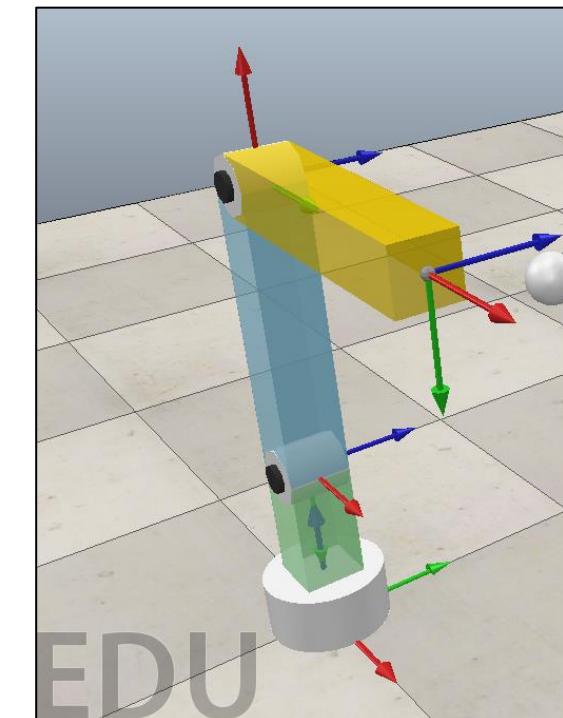


$$r \leq a_1$$



# Exemplo: Manipulador Antropomórfico

- Dado o arranjo do Manipulador 3R em 3D, determine sua cinemática inversa.

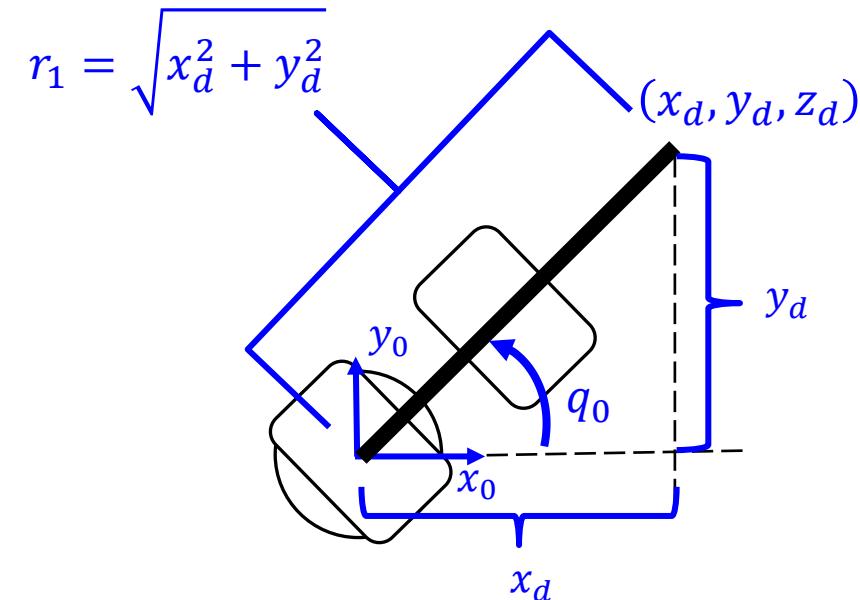
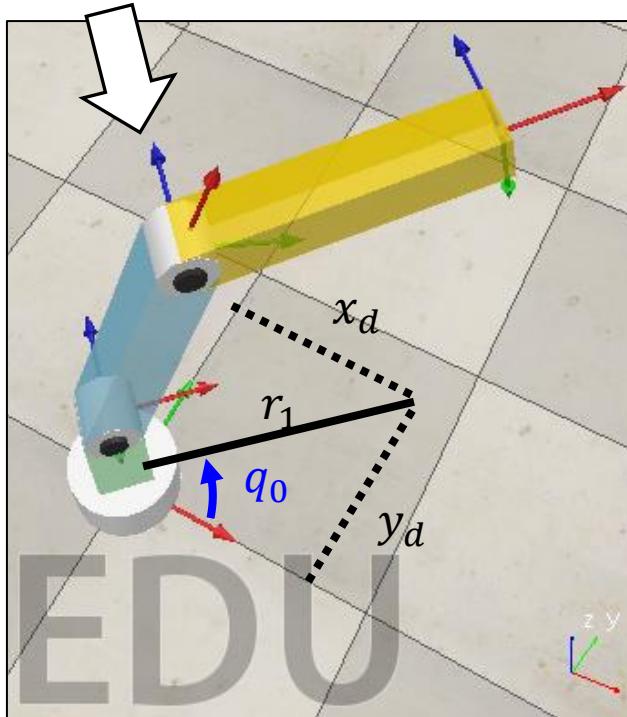




# Exemplo: Manipulador Antropomórfico

- Vista superior ( $q_n \neq 0^\circ$ ):

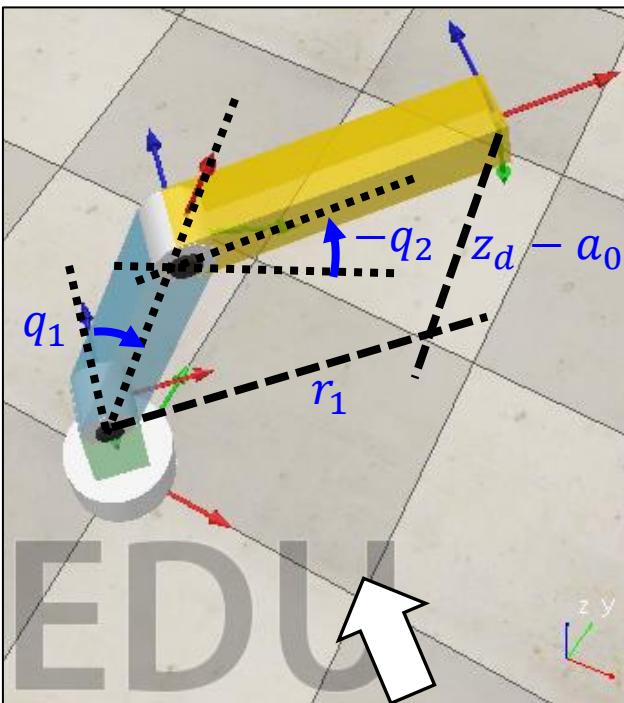
Olhando robô dessa perspectiva



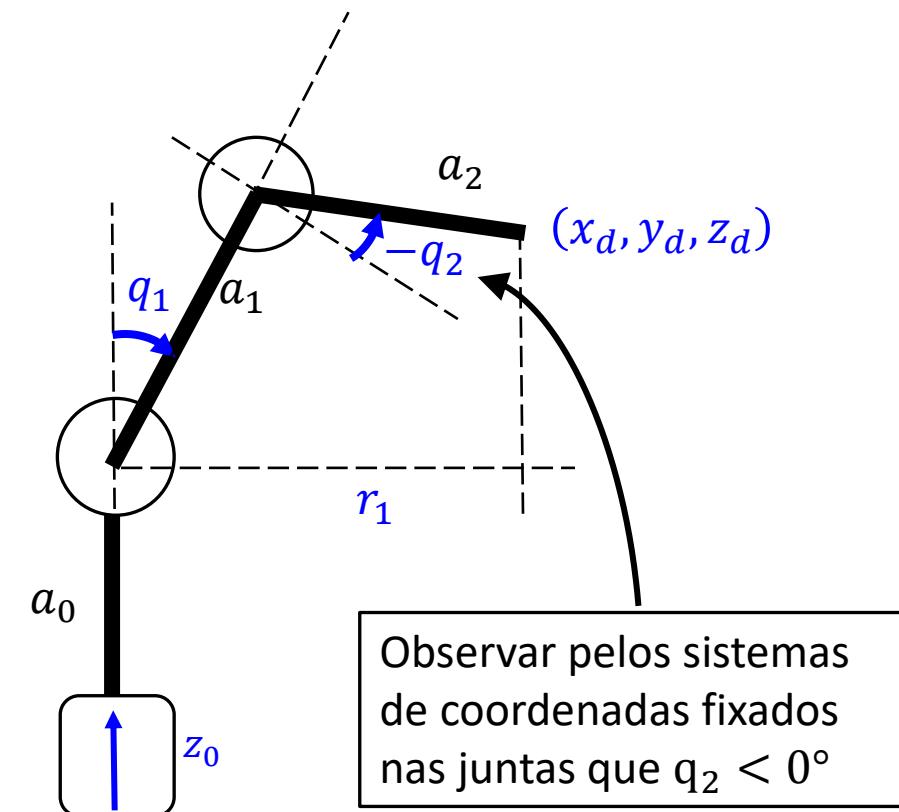


# Exemplo: Manipulador Antropomórfico

- Vista lateral ( $q_n \neq 0^\circ$ ):



Olhando robô dessa perspectiva



# Exemplo: Manipulador Antropomórfico

- Observe as relações:

$$r_1 = \sqrt{x_d^2 + y_d^2}$$

$$r_2 = z_d - a_0$$

$$r_3 = \sqrt{r_1^2 + r_2^2}$$

$$q_1 + \beta + \gamma = 90^\circ$$

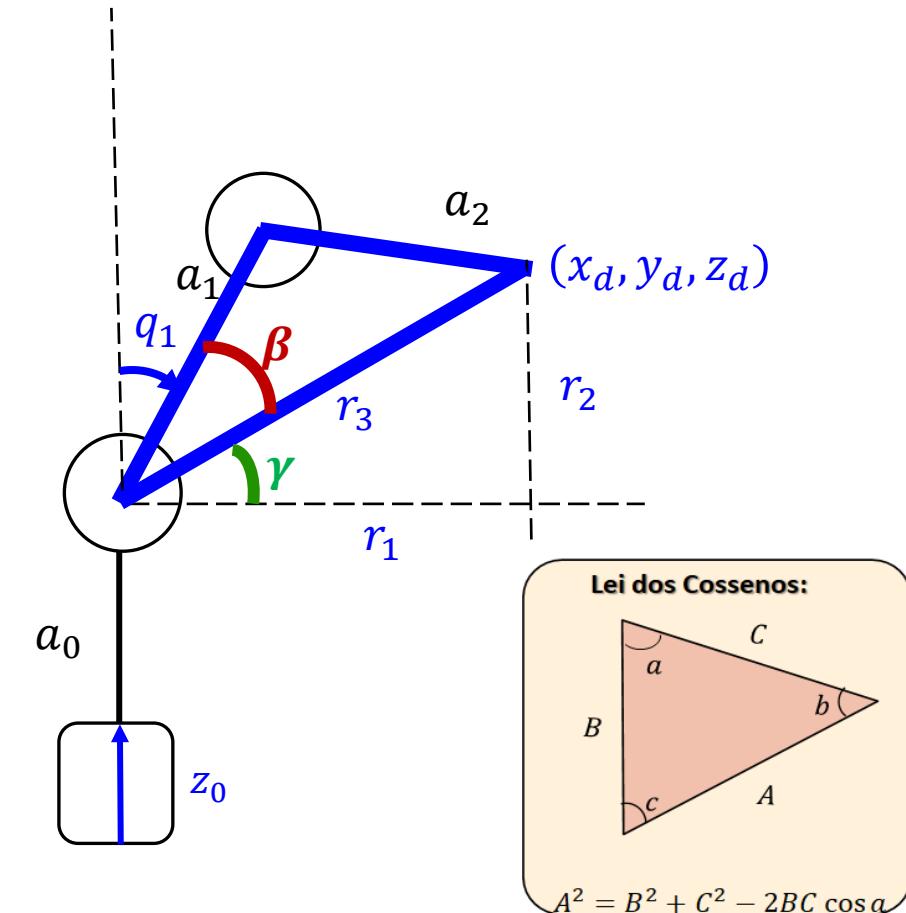
$$\gamma = \tan^{-1} \left( \frac{r_2}{r_1} \right)$$

- Aplicando a lei dos cossenos:

$$a_2^2 = r_3^2 + a_1^2 - 2 r_3 a_1 \cos(\beta)$$

$$\beta = \cos^{-1} \left( \frac{r_3^2 + a_1^2 - a_2^2}{2 r_3 a_1} \right)$$

$$q_1 = 90^\circ - \beta - \gamma$$





# Exemplo: Manipulador Antropomórfico

- Observe que:

$$\begin{cases} \phi = 180^\circ - \sigma \\ \phi = 90^\circ - (-q_2) \end{cases}$$

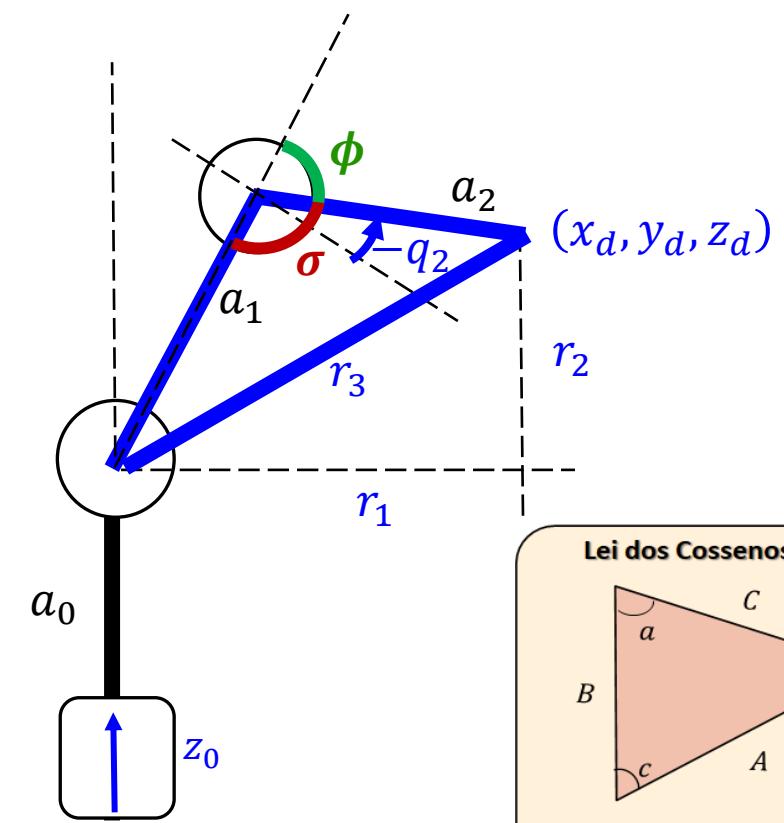

---

$$q_2 = 90^\circ - \sigma$$

$$r_3^2 = a_1^2 + a_2^2 - 2 a_1 a_2 \cos(\sigma)$$

$$\cos(\sigma) = \frac{a_1^2 + a_2^2 - r_3^2}{2 a_1 a_2}$$

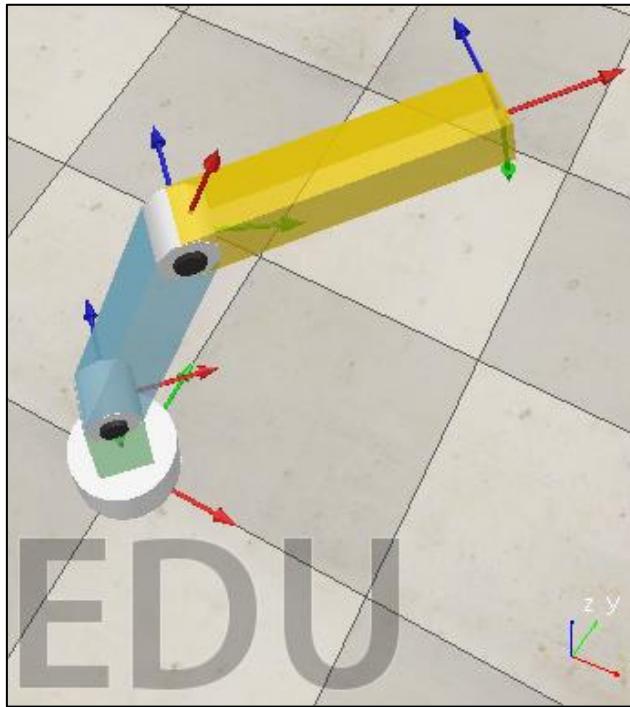
$$q_2 = 90^\circ - \cos^{-1} \left( \frac{a_1^2 + a_2^2 - r_3^2}{2 a_1 a_2} \right)$$



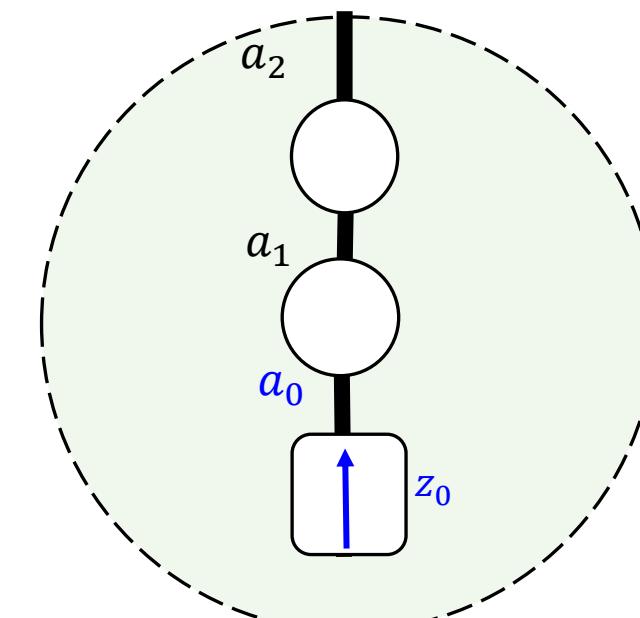


# Exemplo: Manipulador 3R em 3D

- Análise do espaço de trabalho (desconsidera colisão):

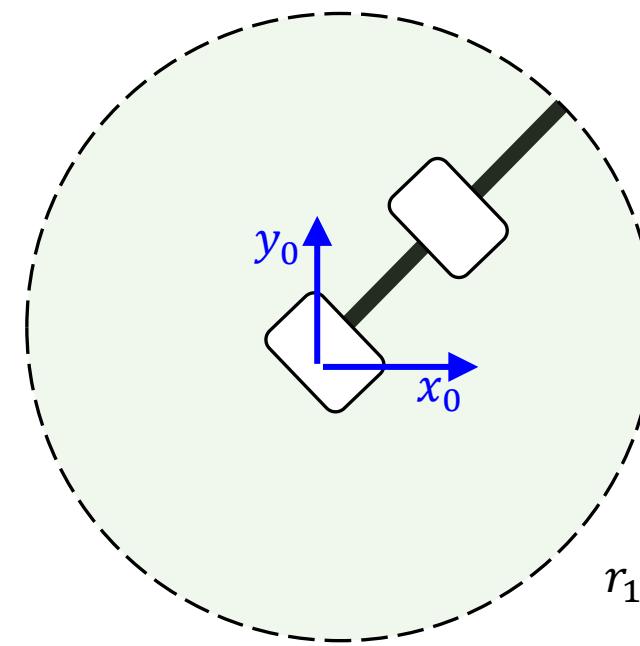


Vista lateral



$$a_0 - (a_1 + a_2) \leq z_d \leq a_0 + a_1 + a_2$$

Vista superior



$$r_1 = \sqrt{x_d^2 + y_d^2}$$

$$r_1 \leq a_1 + a_2$$



# Tarefa de Simulação

---

- Implementar e simular no CoppeliaSim o manipulador 2R em 3D e sua cinemática inversa;
- Implementar e simular no CoppeliaSim o manipulador 3R em 3D e sua cinemática inversa;
- Usar como solução da cinemática inversa o equacionamento analítico.



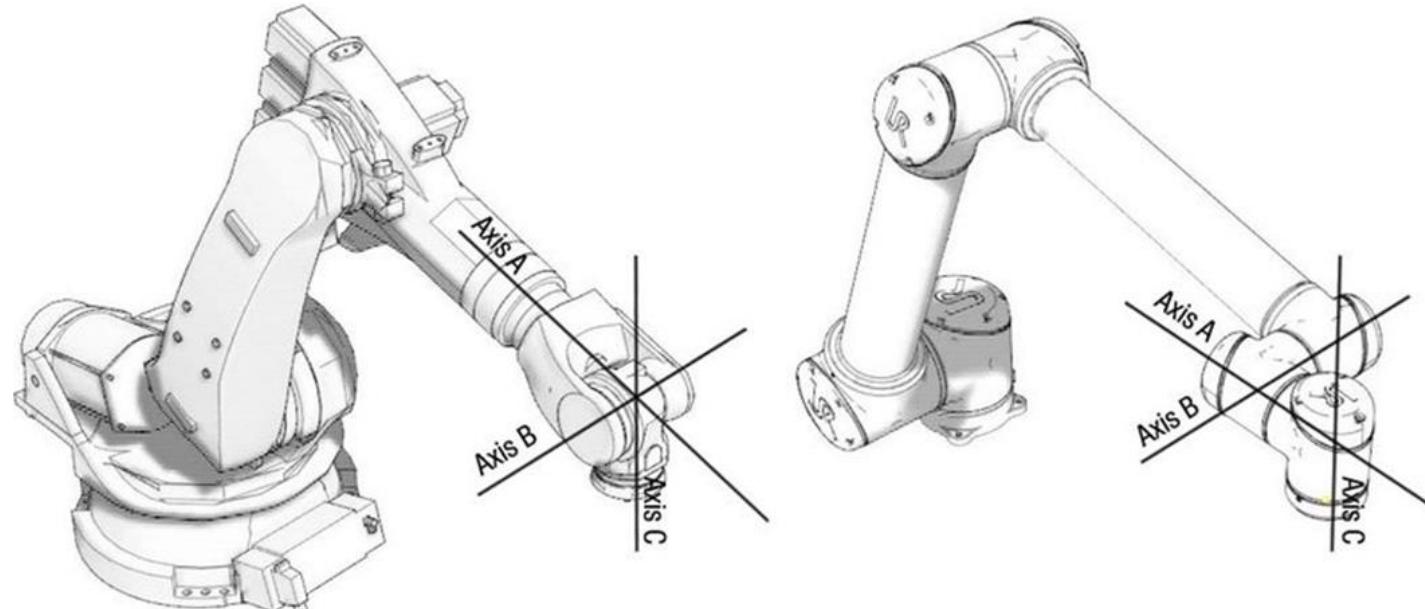
---

# Cinemática Inversa de Orientação 3D



# O Punho Esférico

- Arranjo formado por três juntas rotacionais, disposta com o mínimo deslocamento entre o ponto de rotação;
- Com ele, é possível atingir qualquer orientação 3D;



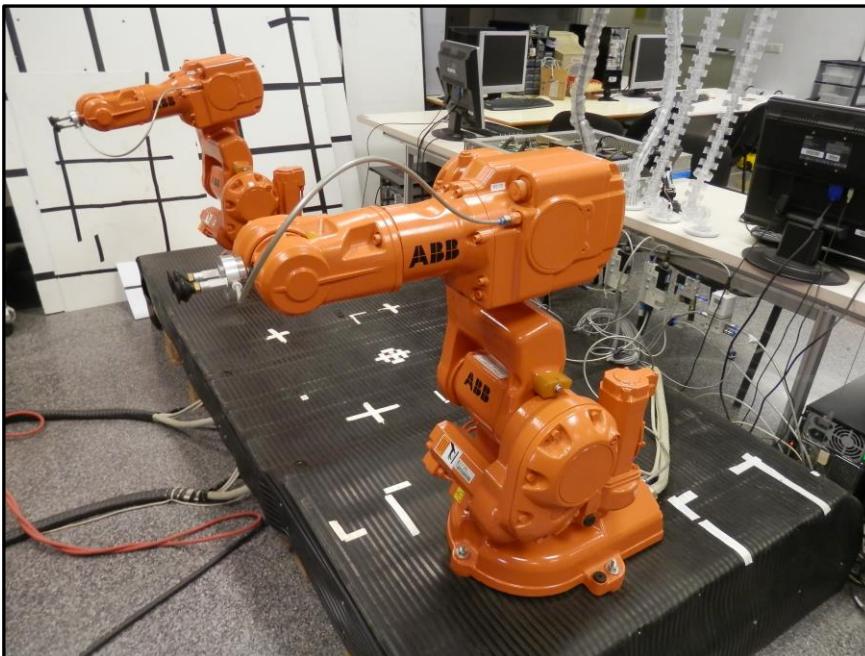
Fonte:

[https://www.researchgate.net/publication/300721200\\_An\\_Approach\\_to\\_Automated\\_Construction\\_Using\\_Adaptive\\_Programming](https://www.researchgate.net/publication/300721200_An_Approach_to_Automated_Construction_Using_Adaptive_Programming)

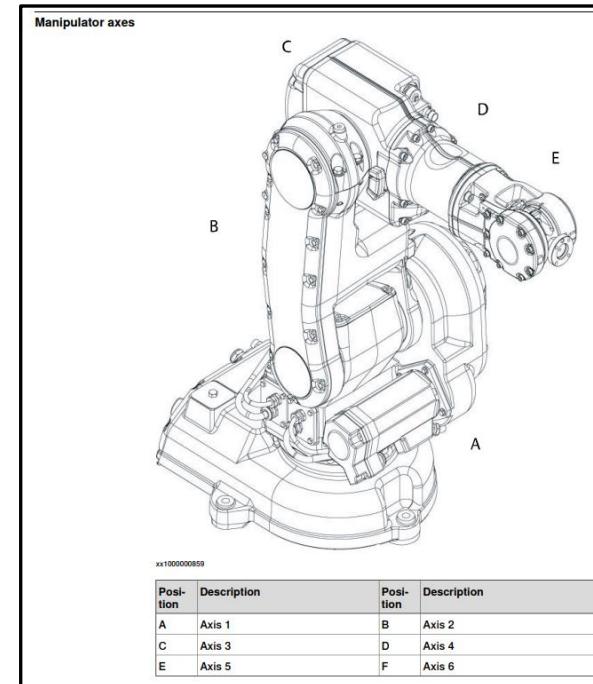


# O Punho Esférico

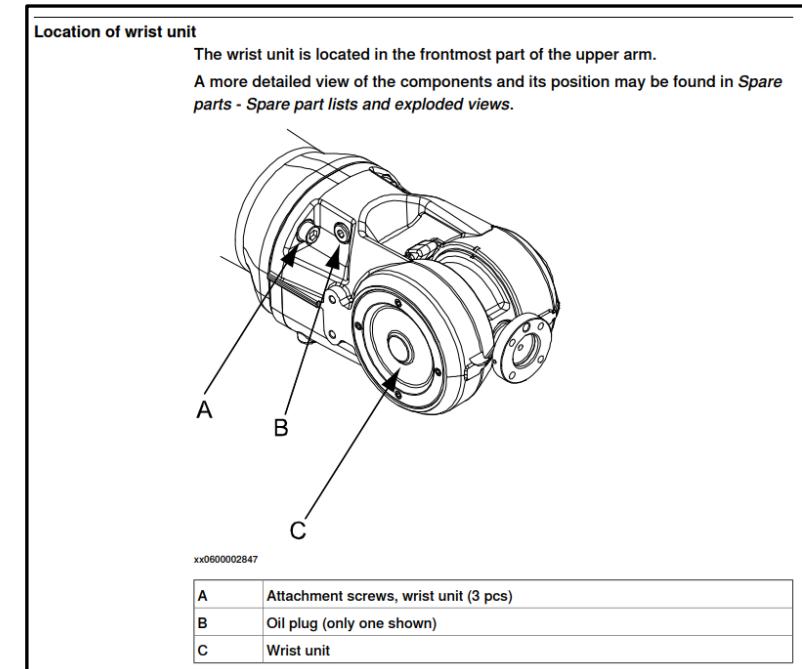
- Exemplo: ABB IRB 140



Fonte:  
Wikimedia [[Clique Aqui](#)]



Fonte:  
Manual 1 [[Clique Aqui](#)]

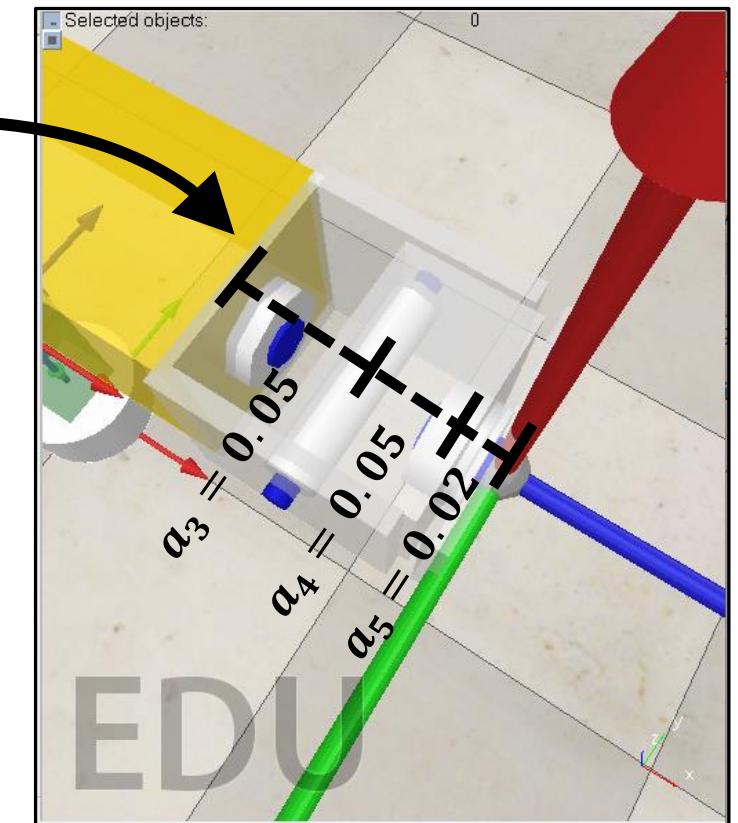
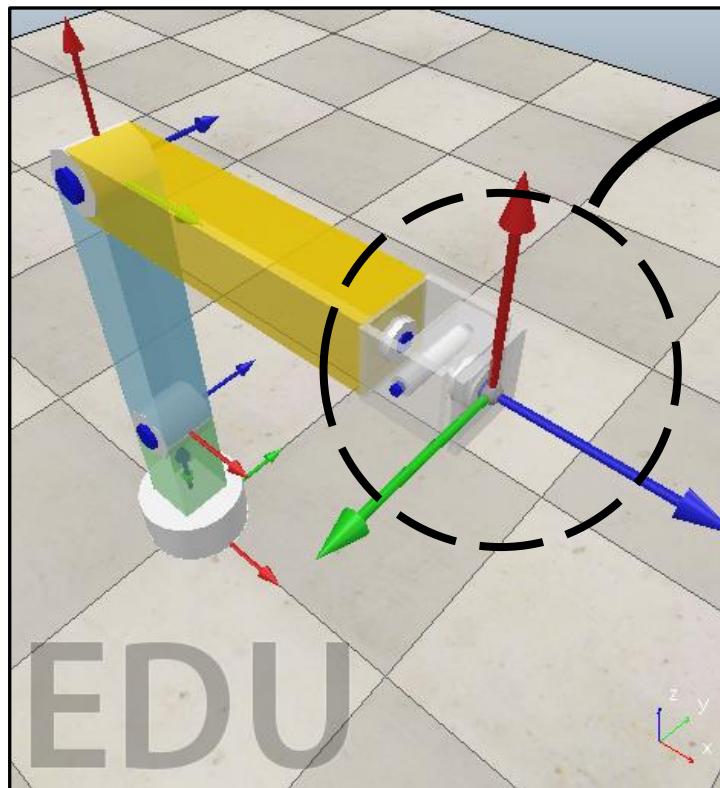


Fonte:  
Manual 2 [[Clique Aqui](#)]



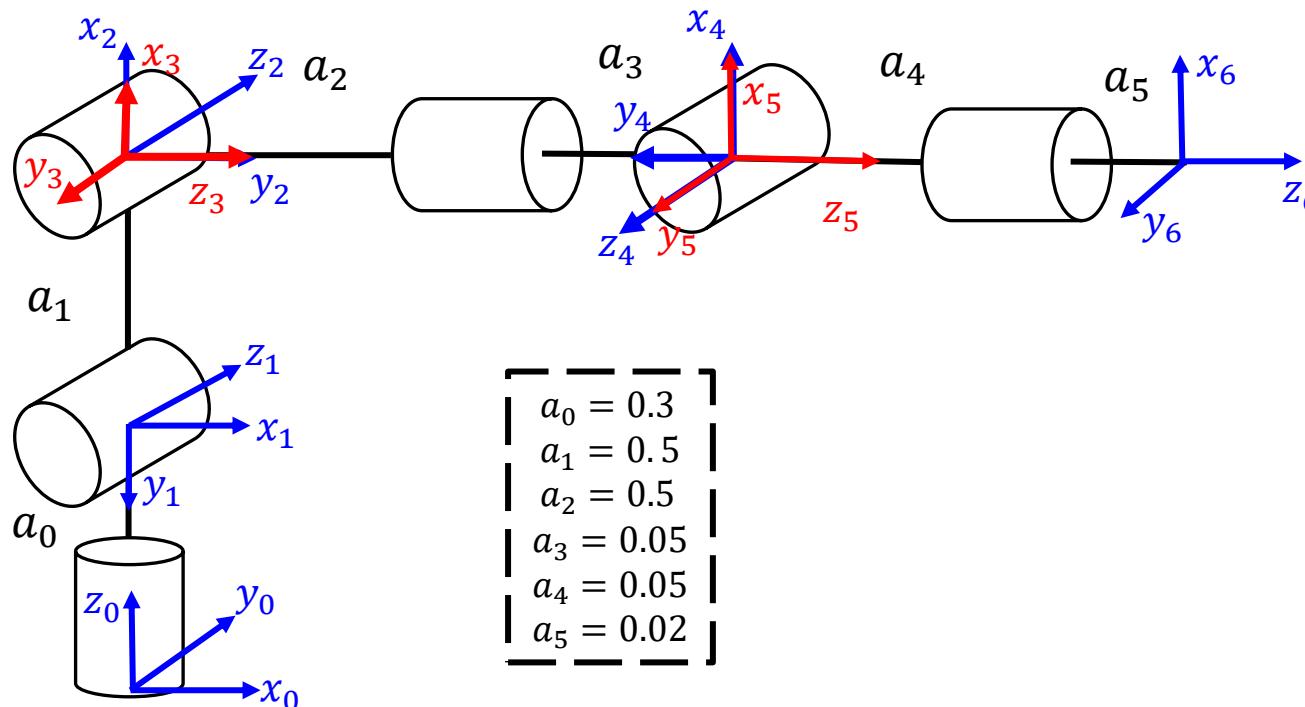
# O Punho Esférico

- No simulador:



# Arranjo 6R com Punho Esférico

- Arranjo do manipulador simulado:



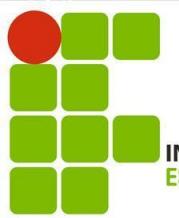
Índice da Junta ( $i$ )	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
0	$q_0$	$a_0$	0	-90
1	$q_1-90$	0	$a_1$	0
2	$q_2$	0	0	-90
3	$q_3$	$a_2+a_3$	0	-90
4	$q_4$	0	0	90
5	$q_5$	$a_4+a_5$	0	0

$\theta$ : Rotação em torno de  $z$ ;

$d$ : Translação sobre  $z$ ;

$a$ : Translação sobre  $x$ ;

$\alpha$ : Rotação em torno de  $x$ ;



INSTITUTO FEDERAL  
ESPIRITO SANTO



---

# Desacoplamento Cinemático



# Desacoplamento Cinemático

- Consiste em **separar posição de orientação** através da capacidade de movimento do punho esférico;
- No exemplo, as 3 primeiras juntas posicionam o centro do punho no espaço, enquanto as 3 últimas juntas (punho) orientam a ferramenta;
- Funciona **ajustando** o meio do punho esférico com sua junta central;

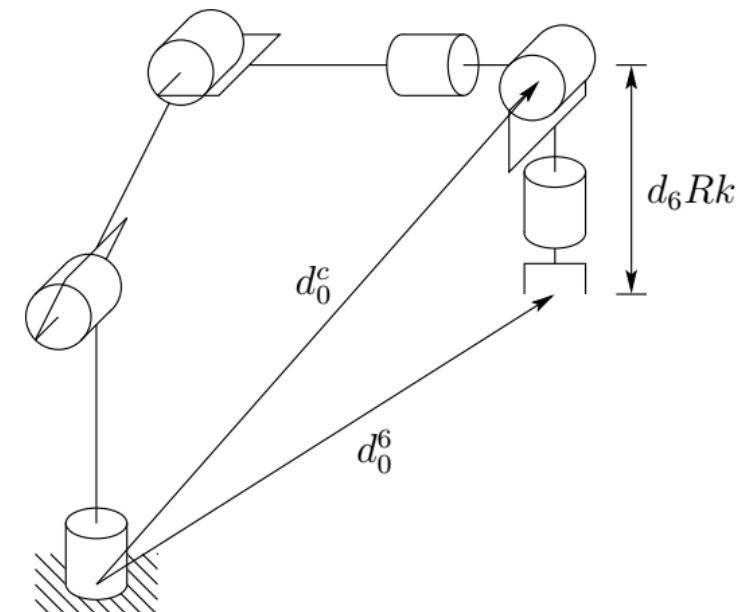


Fig. 3.12 Kinematic decoupling

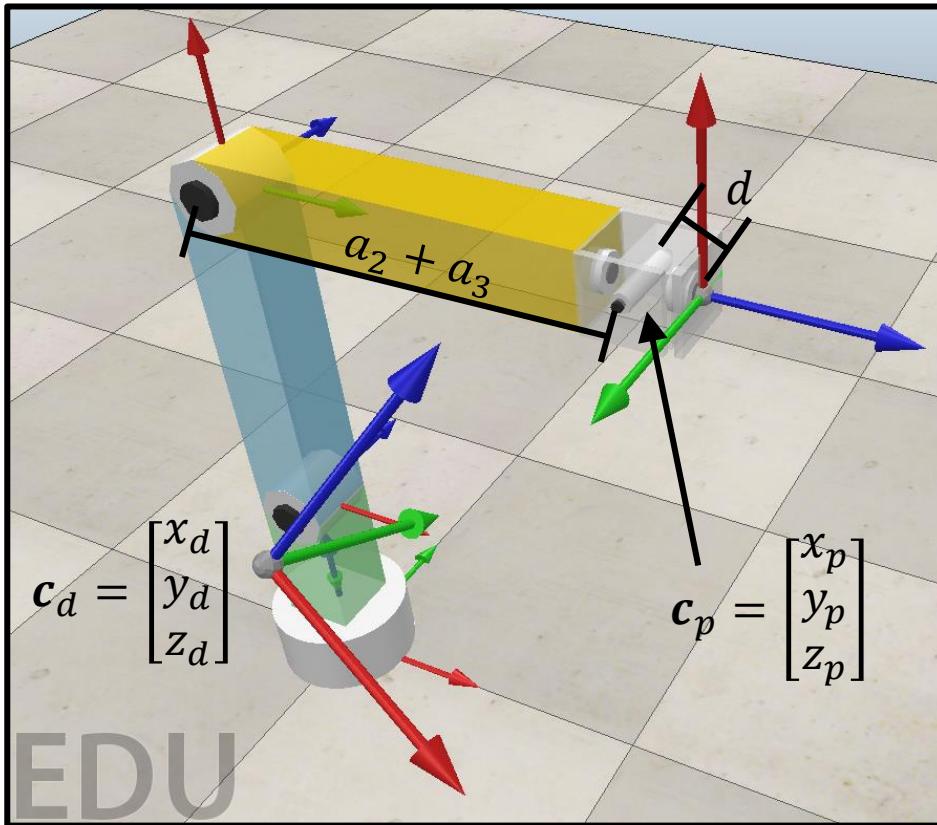
Fonte: Spong, Hutchinson e Vidyasagar (2005).



---

# Desacoplamento Cinemático da Posição

# Desacoplamento Cinemático – Posição



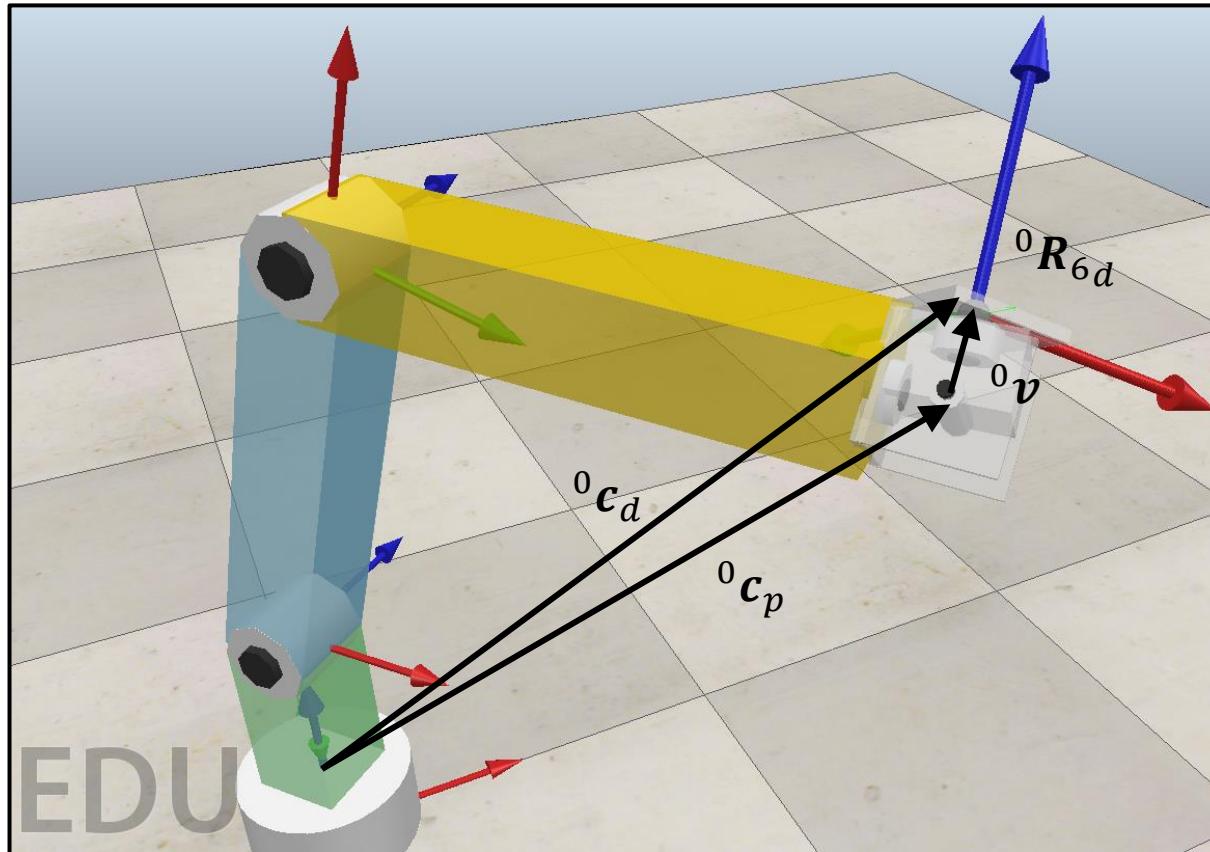
## Legenda:

- $c_p$  – coordenadas que representam a posição que o centro do punho deve assumir.
  - Deve-se considerar o elo final da parte 3R como a soma ( $a_2 + a_3$ ) . Assim, o controle de posicionamento irá posicionar o centro do punho adequadamente;
- $c_d$  – coordenadas que representam a posição desejada para a ferramenta no espaço 3D;
- ${}^0 R_{6d}$  – matriz de rotação contendo a orientação desejada para a ferramenta no espaço 3D;
- $d$  – distância entre o centro do punho e o acoplamento da ferramenta. Observe que  $d = a_4 + a_5$ .

$${}^0 R_{6d} = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix}$$



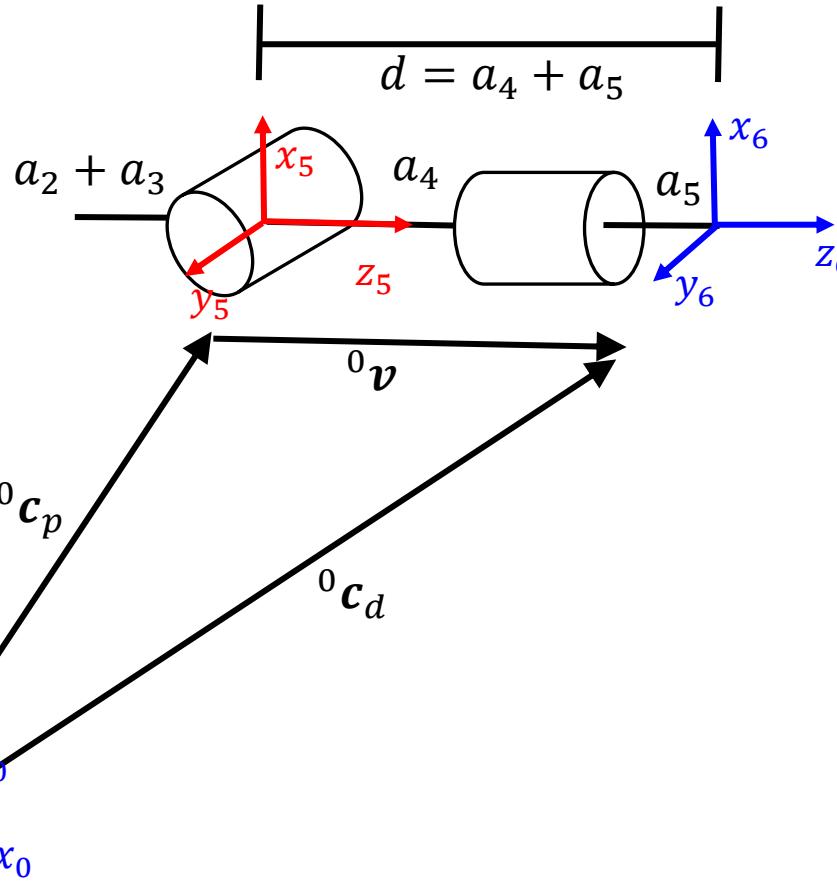
# Desacoplamento Cinemático – Posição



- Spong, Hutchinson e Vidyasagar (2005) observam que:
$${}^0\boldsymbol{c}_d = {}^0\boldsymbol{c}_p + {}^0\boldsymbol{v}$$
$${}^0\boldsymbol{c}_p = {}^0\boldsymbol{c}_d - {}^0\boldsymbol{v}$$
- O vetor  ${}^0\boldsymbol{v}$  é uma distância  $d$ , deslocada sobre o eixo z da penúltima para última junta do punho esférico, compensada da rotação desejada. Isto é:

$${}^0\boldsymbol{v} = {}^0\boldsymbol{R}_{6d} \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix}$$

# Desacoplamento Cinemático – Posição



- Dados:

$${}^0\boldsymbol{c}_p = {}^0\boldsymbol{c}_d - {}^0\boldsymbol{v}$$

$${}^0\boldsymbol{c}_p = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} \quad {}^0\boldsymbol{c}_d = \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} \quad {}^0\boldsymbol{v} = {}^0\boldsymbol{R}_{6d} \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix}$$

$${}^0\boldsymbol{R}_{6d} = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix}$$

- Conclui-se:

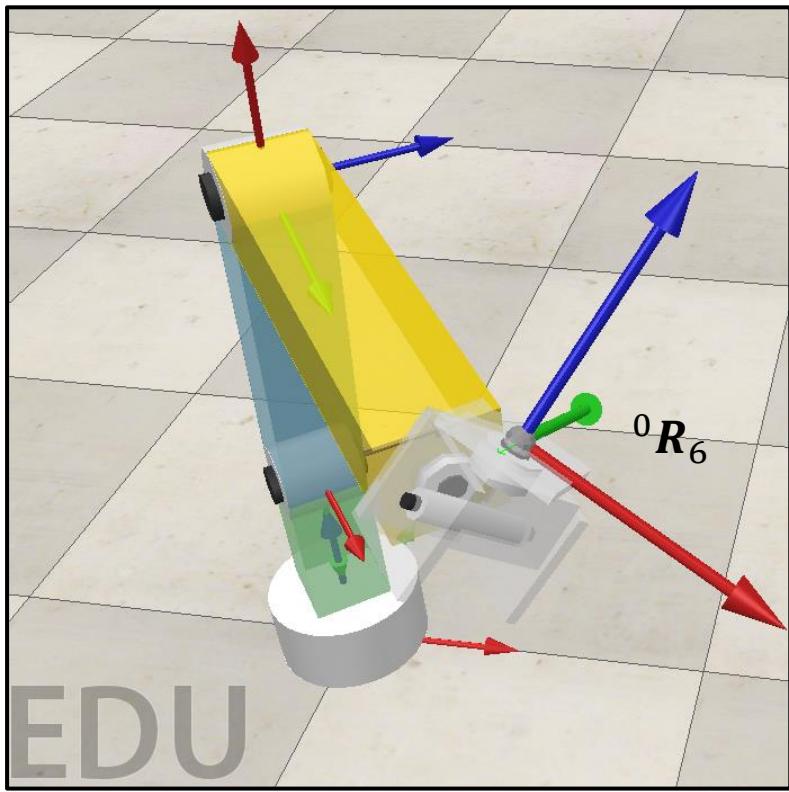
$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} x_d - d r_{02} \\ y_d - d r_{12} \\ z_d - d r_{22} \end{bmatrix}$$



# Desacoplamento Cinemático da Orientação



# Desacoplamento Cinemático – Orientação



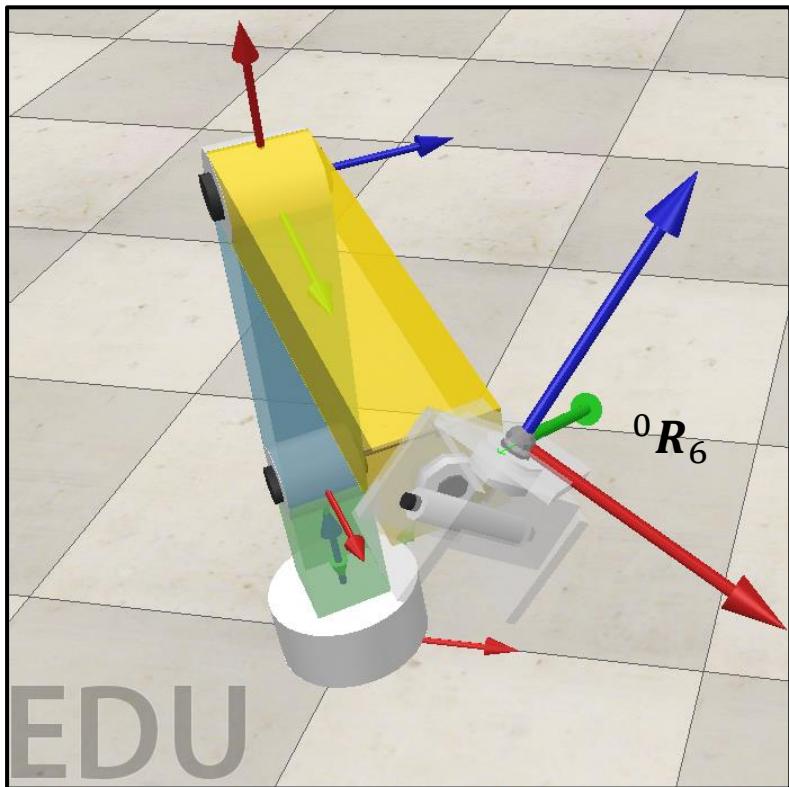
EDU

- Supondo a situação ilustrada, são dados:
  - ${}^0 R_{6d}$  - orientação desejada (alvo)
  - ${}^0 R_3$  - orientação obtida da cinemática inversa das 3 primeiras juntas do 3R
- Sabe-se das propriedades das Matrizes de Rotação que:
- Resolvendo para  ${}^3 R_6$ :

$${}^3 R_6 = ({}^0 R_3)^{-1} {}^0 R_{6d} \rightarrow {}^3 R_6 = {}^0 R_3^T {}^0 R_{6d}$$



# Desacoplamento Cinemático – Orientação



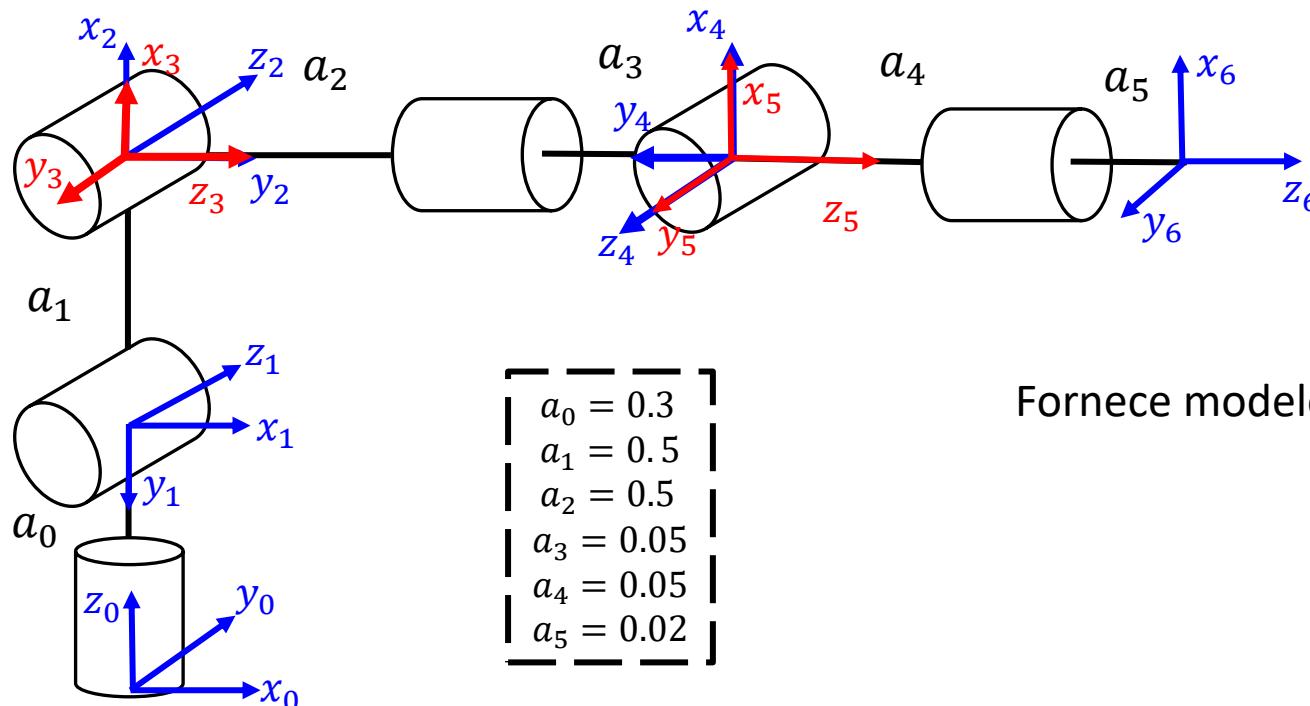
- A expressão:

$${}^3\mathbf{R}_6 = {}^0\mathbf{R}_3^T {}^0\mathbf{R}_{6_d}$$

- Contém em seus elementos a orientação requerida para as 3 últimas juntas do robô.

# Desacoplamento Cinemático – Orientação

- Considerando apenas o modelo das rotações 3D:



Fornece modelo para  ${}^3R_6$

Índice da Junta ( $i$ )	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
0	$q_0$	$a_0$	0	-90
1	$q_1-90$	0	$a_1$	0
2	$q_2$	0	0	-90
3	$q_3$	$a_2+a_3$	0	-90
4	$q_4$	0	0	90
5	$q_5$	$a_4+a_5$	0	0

$\theta$ : Rotação em torno de z;

$d$ : Translação sobre z;

$a$ : Translação sobre x;

$\alpha$ : Rotação em torno de x;

[https://colab.research.google.com/drive/14MT-m5uyTp736\\_TyuCrkMuKcb1tZLn3O?usp=sharing](https://colab.research.google.com/drive/14MT-m5uyTp736_TyuCrkMuKcb1tZLn3O?usp=sharing)



# Desacoplamento Cinemático – Orientação

- Modelo da Matriz de Rotação:

$${}^3\mathbf{R}_6 = \begin{bmatrix} -\sin(q_3)\sin(q_5) + \cos(q_3)\cos(q_4)\cos(q_5) & -\sin(q_3)\cos(q_5) - \sin(q_5)\cos(q_3)\cos(q_4) & \sin(q_4)\cos(q_3) \\ \sin(q_3)\cos(q_4)\cos(q_5) + \sin(q_5)\cos(q_3) & -\sin(q_3)\sin(q_5)\cos(q_4) + \cos(q_3)\cos(q_5) & \sin(q_3)\sin(q_4) \\ -\sin(q_4)\cos(q_5) & \sin(q_4)\sin(q_5) & \cos(q_4) \end{bmatrix}$$

- De onde se pode extrair as orientações como:

$$q_4 = \cos^{-1}(r_{22})$$

$$q_3 = \tan^{-1}\left(\frac{r_{12}}{r_{02}}\right)$$

$$q_5 = \tan^{-1}\left(\frac{r_{21}}{r_{20}}\right)$$





# Desacoplamento Cinemático – Orientação

## ■ Problema:

- Ao extrair os ângulos, usam-se apenas 3 elementos da uma matriz de rotação que contém ao todo 9 elementos;
- Isto pode causar soluções inválidas que requerem validação e condições de contorno que permitam testar e validar a solução encontrada;
- Na preparação desta aula, por tentativa e erro, eu validei a solução abaixo no problema simulado;

## ■ Método adotado:

- Partindo de  $q_5 = \tan^{-1} \left( \frac{r_{21}}{r_{20}} \right)$



**1º Teste (inválido):**  $q_5 = \tan^{-1} \left( \frac{r_{21}}{r_{20}} \right)$

**2º Teste (inválido):**  $q_5 = -\tan^{-1} \left( \frac{r_{21}}{r_{20}} \right)$

**3º Teste (válido):**  $q_5 = 180^\circ - \tan^{-1} \left( \frac{r_{21}}{r_{20}} \right)$





# Simulação Guiada

- Mostrar no simulador CoppeliaSim o método de desacoplamento cinemático em funcionamento;
- **Mostrar o procedimento adotado para ajuste da orientação espacial.**



---

# Parte 2: Métodos Numéricos



# Cinemática Inversa de Posição 3D



# Jacobiano em 3D

- A relação a seguir permanece válida em 3D:

$$\nu = J(q)\dot{q}$$

- Porém,  $J(q)$  em 3D é mais difícil de se obter;
- Interessados em detalhes:

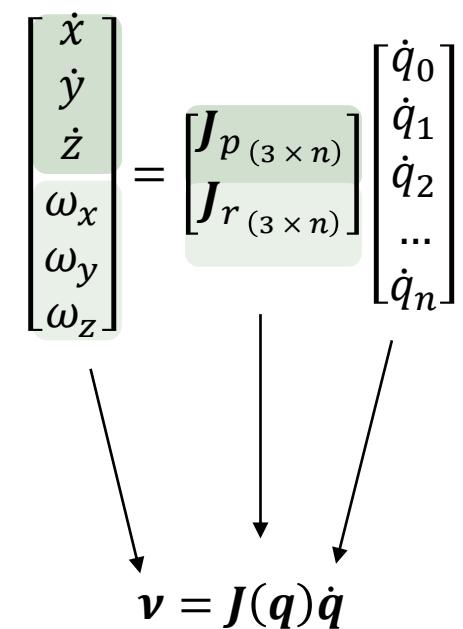
- Capítulo 3 (Seções 3.1 e 3.2) do livro ***Modelling and Control of Robot Manipulators*** (Sciavicco e Siciliano; 2000);
- Capítulo 4 (Seções 4.6 e 4.7) do livro ***Robots Modelling and Control*** (Spong, Hutchinson e Vidyasagar; 2005)

# Obtendo o Jacobiano em 3D

- O Jacobiano é uma matriz de dimensões  $6 \times n$  onde:
  - $n$ : número total de juntas do robô
- Pode-se subdividi-lo em duas partes:
  - **Posicional** – relativa a  $[\dot{x} \quad \dot{y} \quad \dot{z}]^T$ :
    - $J_p$  - Matriz  $3 \times n$  que mapeia contribuições das juntas na parte linear do Jacobiano
  - **Rotacional** – relativa a  $[\omega_x \quad \omega_y \quad \omega_z]^T$ :
    - $J_r$  - Matriz  $3 \times n$  que mapeia contribuições das juntas na parte rotacional do Jacobiano

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} J_{p(3 \times n)} \\ J_{r(3 \times n)} \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dots \\ \dot{q}_n \end{bmatrix}$$

$\nu = J(q)\dot{q}$





# Obtendo o Jacobiano em 3D

- Método de obtenção do Jacobiano:
  - Consiste em escrever a Matriz Jacobiana a partir de uma tabela de apoio na modelagem;
  - Esta tabela contém modelos matemáticos que representam as contribuições que as velocidades das juntas do robô ( $\dot{q}$ ) tem nas velocidades do *twist* ( $v$ );
- A tabela de apoio contém:
  - 2 linhas: representam as partes do Jacobiano (Posicional e Rotacional);
  - 2 colunas: representam os tipos de junta existentes (Prismática e Rotacional);

# Obtendo o Jacobiano em 3D

- Tabela de apoio na modelagem do Jacobiano:

Parte do Jacobiano	Contribuição da junta, se ela for <u>PRISMÁTICA</u>	Contribuição da junta, se ela for <u>ROTACIONAL</u>
$J_p$	${}^0R_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	${}^0R_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times ({}^0o_n - {}^0o_i)$
$J_r$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	${}^0R_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} J_p_{(3 \times n)} \\ J_r_{(3 \times n)} \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dots \\ \dot{q}_n \end{bmatrix}$$

$\nu = J(q)\dot{q}$

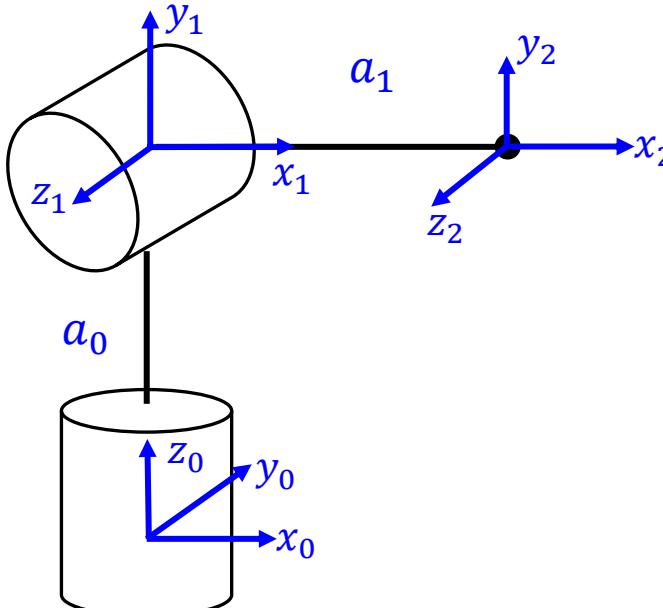
Legenda:

- $i$ : índice de preenchimento. Representa uma coluna da matriz  $J(\mathbf{q})$  (começa em  $i = 0$ )
- ${}^0R_i$ : matriz de rotação do sistema  $\{i\}$  para  $\{0\}$
- ${}^0o_n$ : vetor de coordenadas da origem do sistema  $\{n\}$  dado no referencial  $\{0\}$
- ${}^0o_i$ : vetor de coordenadas da origem do sistema  $\{i\}$  dado no referencial  $\{0\}$



# Exemplo: Manipulador 2R em 3D

- Dado o arranjo cinemático em 3D, determinar o Jacobiano:
  - Inserir os sistemas de coordenadas pelas regras DH;
  - Preencher a tabela de DH;



Índice da Junta ( $i$ )	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
0	q0	a0	0	90
1	q1	0	a1	0

$\theta$ : Rotação em torno de  $z$ ;  
 $d$ : Translação sobre  $z$ ;  
 $a$ : Translação sobre  $x$ ;  
 $\alpha$ : Rotação em torno de  $x$ ;



# Exemplo: Manipulador 2R em 3D

- Em seguida, calcule as transformações homogêneas da tabela DH linha a linha:

$${}^0T_1 = \begin{bmatrix} \cos(q_0) & 0 & \sin(q_0) & 0 \\ \sin(q_0) & 0 & -\cos(q_0) & 0 \\ 0 & 1 & 0 & a_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1T_2 = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & a_1 \cos(q_1) \\ \sin(q_1) & \cos(q_1) & 0 & a_1 \sin(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0T_2 = {}^0T_1 {}^1T_2 = \begin{bmatrix} \cos(q_0) \cos(q_1) & -\sin(q_1) \cos(q_0) & \sin(q_0) & a_1 \cos(q_0) \cos(q_1) \\ \sin(q_0) \cos(q_1) & -\sin(q_0) \sin(q_1) & -\cos(q_0) & a_1 \sin(q_0) \cos(q_1) \\ \sin(q_1) & \cos(q_1) & 0 & a_0 + a_1 \sin(q_1) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

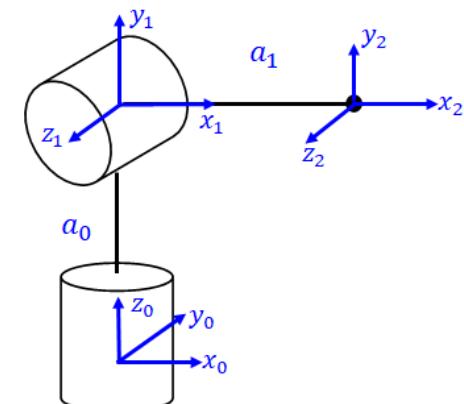
[https://colab.research.google.com/drive/1xIUpXViZPUGjAbTTqzwOnJR\\_kBCsHaRC](https://colab.research.google.com/drive/1xIUpXViZPUGjAbTTqzwOnJR_kBCsHaRC)

# Exemplo: Manipulador 2R em 3D

- Depois, determine o formato da Matriz Jacobiana baseado no número de juntas:
- Neste caso:
  - O Jacobiano do Manipulador 2R será de dimensões  $(6 \times 2)$ ;
  - Pois o manipulador possui duas juntas ( $n = 2$ );
  - Separando a parte linear ( $J_p$ ) em dois vetores coluna ( $J_{p_1}$ ) e ( $J_{p_2}$ );
  - Separando a parte rotacional ( $J_r$ ) em dois vetores coluna ( $J_{r_1}$ ) e ( $J_{r_2}$ );
  - É possível escrever o Jacobiano como:

$$J(\dot{q})_{(6 \times 2)} = \begin{bmatrix} J_{p1(3 \times 1)} & J_{p2(3 \times 1)} \\ J_{r1(3 \times 1)} & J_{r2(3 \times 1)} \end{bmatrix}$$

$$\dot{q} = \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \end{bmatrix} \quad \boldsymbol{\nu} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

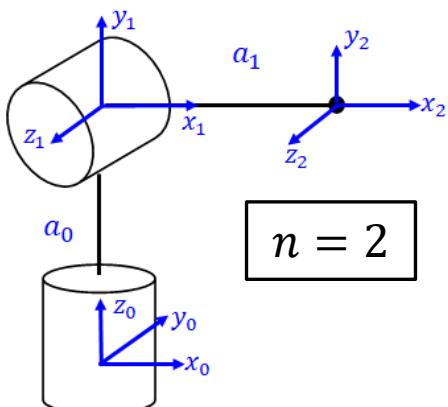




# Exemplo: Manipulador 2R em 3D

- Selecionando na tabela de apoio o tipo de junta adequado, é possível preencher a Matriz Jacobiana como:

Parte do Jacobiano	Contribuição da junta, se ela for ROTACIONAL
$J_p$	${}^0 R_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times ({}^0 o_n - {}^0 o_i)$
$J_r$	${}^0 R_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$



# Exemplo: Manipulador 2R em 3D

- O modelo obtido para a Matriz Jacobiana:

$$J(\mathbf{q})_{(6 \times 2)} = \begin{bmatrix} {}^0R_0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times ({}^0o_2 - {}^0o_0) & {}^0R_1 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times ({}^0o_2 - {}^0o_1) \\ {}^0R_0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & {}^0R_1 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix}$$

- Utiliza informações óbvias e informações da cinemática direta para preenchimento.
- **Informações óbvias:**
  - Rotação de  $\{0\}$  para  $\{0\}$  é identidade;
  - Vetor de coordenadas da origem de  $\{0\}$  dada em  $\{0\}$  é um vetor nulo;

$${}^0R_0 = I_{3 \times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$${}^0o_0 = [0 \quad 0 \quad 0]^T$$



# Exemplo: Manipulador 2R em 3D

- Informações da Cinemática Direta:
  - Dada a transformação homogênea de  $\{1\}$  para  $\{0\}$  ( ${}^0T_1$ ) , observa-se nela:

$${}^0T_1 = \begin{bmatrix} \cos(q_0) & 0 & \sin(q_0) & 0 \\ \sin(q_0) & 0 & -\cos(q_0) & 0 \\ 0 & 1 & 0 & a_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0R_1 = \begin{bmatrix} \cos q_0 & 0 & \sin q_0 \\ \sin q_0 & 0 & -\cos q_0 \\ 0 & 1 & 0 \end{bmatrix}$$

Rotação de  $\{1\}$  para  $\{0\}$

$${}^0o_1 = [0 \ 0 \ a_0]^T$$

Vetor de coordenadas da origem de  $\{1\}$  dada em  $\{0\}$



# Exemplo: Manipulador 2R em 3D

- Informações da Cinemática Direta.
  - Dada a transformação homogênea de {2} para {0} ( ${}^0T_2$ ), observa-se nela:

$${}^0T_2 = \begin{bmatrix} \cos(q_0)\cos(q_1) & -\sin(q_1)\cos(q_0) & \sin(q_0) & a_1 \cos(q_0)\cos(q_1) \\ \sin(q_0)\cos(q_1) & -\sin(q_0)\sin(q_1) & -\cos(q_0) & a_1 \sin(q_0)\cos(q_1) \\ \sin(q_1) & \cos(q_1) & 0 & a_0 + a_1 \sin(q_1) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$${}^0o_2 = \begin{bmatrix} a_1 \cos q_0 \cos q_1 \\ a_1 \sin q_0 \cos q_1 \\ a_0 + a_1 \sin q_1 \end{bmatrix}$$

Vetor de coordenadas da origem de {2} dada em {0}

# Exemplo: Manipulador 2R em 3D

- Substituindo valores e efetuando operações:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left( \begin{bmatrix} a_1 \cos q_0 \cos q_1 \\ a_1 \sin q_0 \cos q_1 \\ a_0 + a_1 \sin q_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \begin{bmatrix} \cos q_0 & 0 & \sin q_0 \\ \sin q_0 & 0 & -\cos q_0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left( \begin{bmatrix} a_1 \cos q_0 \cos q_1 \\ a_1 \sin q_0 \cos q_1 \\ a_0 + a_1 \sin q_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ a_0 \end{bmatrix} \right) \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} a_1 \cos q_0 \cos q_1 \\ a_1 \sin q_0 \cos q_1 \\ a_0 + a_1 \sin q_1 \end{bmatrix} \begin{bmatrix} \sin q_0 & a_1 \cos q_0 \cos q_1 \\ -\cos q_0 & a_1 \sin q_0 \cos q_1 \\ 0 & a_1 \sin q_1 \end{bmatrix} \begin{bmatrix} \sin q_0 \\ -\cos q_0 \\ 0 \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \end{bmatrix}$$

Produto Cruzado entre  
Vetores



# Exemplo: Manipulador 2R em 3D

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} [0] & \begin{bmatrix} a_1 \cos q_0 \cos q_1 \\ a_1 \sin q_0 \cos q_1 \\ a_0 + a_1 \sin q_1 \end{bmatrix} \\ [0] & \begin{bmatrix} \sin q_0 \\ -\cos q_0 \\ 0 \end{bmatrix} \times \begin{bmatrix} a_1 \cos q_0 \cos q_1 \\ a_1 \sin q_0 \cos q_1 \\ a_1 \sin q_1 \end{bmatrix} \\ [1] & \begin{bmatrix} \sin q_0 \\ -\cos q_0 \\ 0 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \end{bmatrix}$$



Produto Cruzado:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \times \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} -a_1 \sin q_0 \cos q_1 & -a_1 \sin q_1 \cos q_0 \\ a_1 \cos q_0 \cos q_1 & -a_1 \sin q_0 \sin q_1 \\ 0 & a_1 \sin^2 q_0 \cos q_1 + a_1 \cos^2 q_0 \cos q_1 \\ 0 & \sin q_0 \\ 0 & -\cos q_0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \end{bmatrix}$$



$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \underbrace{\begin{bmatrix} -a_1 \sin q_0 \cos q_1 & -a_1 \cos q_0 \sin q_1 \\ a_1 \cos q_0 \cos q_1 & -a_1 \sin q_0 \sin q_1 \\ 0 & a_1 \cos q_1 \\ 0 & \sin q_0 \\ 0 & -\cos q_0 \\ 1 & 0 \end{bmatrix}}_{J(\mathbf{q})} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \end{bmatrix}$$



# Tarefa de Simulação

---

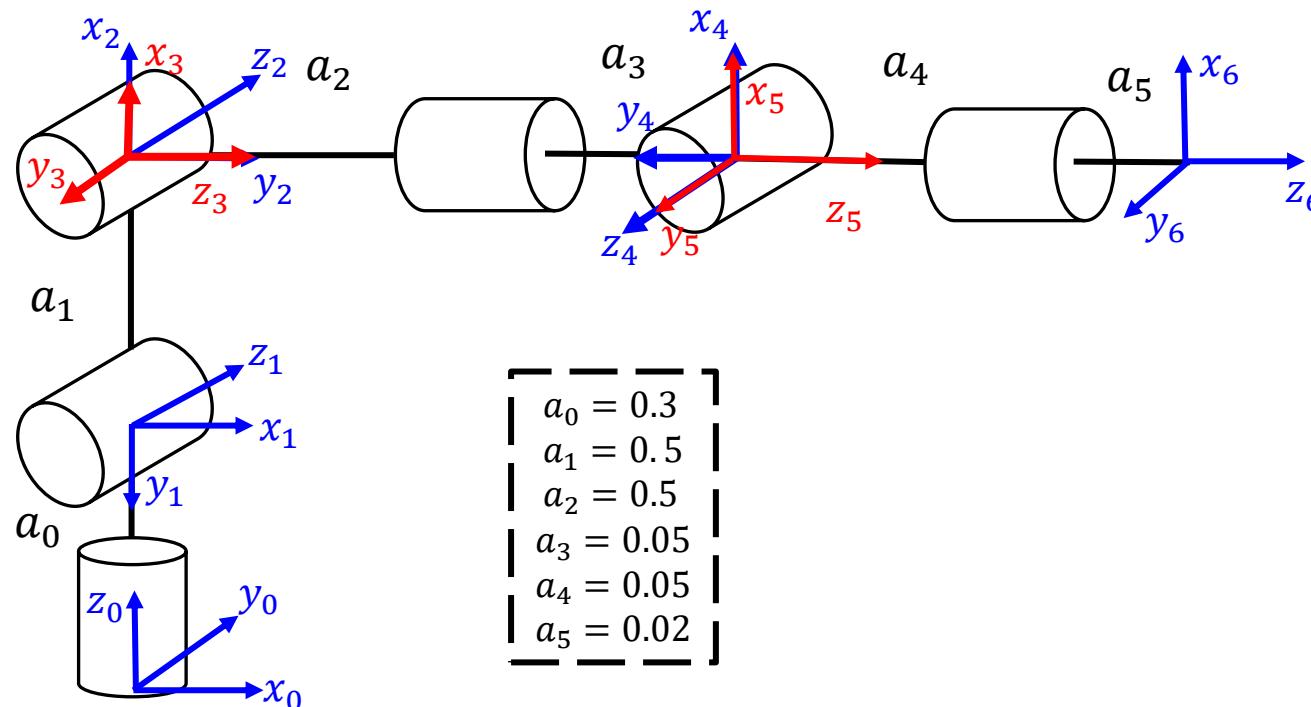
- Simular no CoppeliaSim o **manipulador 2R em 3D** e sua cinemática inversa pelo **método numérico**;
- Adaptar o código para implementar e simular no CoppeliaSim o **manipulador 3R em 3D** e sua cinemática inversa.



---

# Método Numérico: Através do Jacobiano 6R (3R com Punho Esférico)

# Arranjo 3R com Punho Esférico



Índice da Junta ( $i$ )	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
0	$q_0$	$a_0$	0	-90
1	$q_1-90$	0	$a_1$	0
2	$q_2$	0	0	-90
3	$q_3$	$a_2+a_3$	0	-90
4	$q_4$	0	0	90
5	$q_5$	$a_4+a_5$	0	0

$\theta$ : Rotação em torno de  $z$ ;  
 $d$ : Translação sobre  $z$ ;  
 $a$ : Translação sobre  $x$ ;  
 $\alpha$ : Rotação em torno de  $x$ ;



# Jacobiano do 3R com Punho Esférico

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} {}^0R_0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times ({}^0o_6 - {}^0o_0) & {}^0R_1 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times ({}^0o_6 - {}^0o_1) & {}^0R_2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times ({}^0o_6 - {}^0o_2) & {}^0R_3 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times ({}^0o_6 - {}^0o_3) & {}^0R_4 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times ({}^0o_6 - {}^0o_4) & {}^0R_5 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times ({}^0o_6 - {}^0o_5) \\ {}^0R_0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & {}^0R_1 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & {}^0R_2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & {}^0R_3 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & {}^0R_4 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & {}^0R_5 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix}$$

Parte do Jacobiano	Contribuição da junta, se ela for ROTACIONAL
$J_p$	${}^0R_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times ({}^0o_n - {}^0o_i)$
$J_r$	${}^0R_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

Observar que Jacobiano é  $(6 \times 6)$ , pois  $n = 6$ .

[https://colab.research.google.com/drive/1Y3QRvgWh3Cc5M6ljB4sZgQm\\_7G2uX5f6](https://colab.research.google.com/drive/1Y3QRvgWh3Cc5M6ljB4sZgQm_7G2uX5f6)



# Jacobiano do 3R com Punho Esférico

$$\begin{bmatrix} (a_4 + a_5) \sin(q_3) \sin(q_4) \cos(q_0) - (a_1 \sin(q_1) + a_2 \cos(q_1 + q_2) + a_3 \cos(q_1 + q_2) + a_4 \sin(q_4) \sin(q_1 + q_2) \cos(q_3) + a_4 \cos(q_4) \cos(q_1 + q_2) + a_5 \sin(q_4) \sin(q_1 + q_2) \cos(q_3) + a_5 \cos(q_4) \cos(q_1 + q_2)) \sin(q_0) \\ (a_4 + a_5) \sin(q_0) \sin(q_3) \sin(q_4) + (a_1 \sin(q_1) + a_2 \cos(q_1 + q_2) + a_3 \cos(q_1 + q_2) + a_4 \sin(q_4) \sin(q_1 + q_2) \cos(q_3) + a_4 \cos(q_4) \cos(q_1 + q_2) + a_5 \sin(q_4) \sin(q_1 + q_2) \cos(q_3) + a_5 \cos(q_4) \cos(q_1 + q_2)) \cos(q_0) \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Coluna 1

$$\begin{bmatrix} (a_1 \cos(q_1) - a_2 \sin(q_1 + q_2) - a_3 \sin(q_1 + q_2) + a_4 \sin(q_4) \cos(q_1 + q_2) - a_4 \sin(q_1 + q_2) \cos(q_4) + a_5 \sin(q_4) \cos(q_3) \cos(q_1 + q_2) - a_5 \sin(q_1 + q_2) \cos(q_4)) \cos(q_0) \\ (a_1 \cos(q_1) - a_2 \sin(q_1 + q_2) - a_3 \sin(q_1 + q_2) + a_4 \sin(q_4) \cos(q_3) \cos(q_1 + q_2) - a_4 \sin(q_1 + q_2) \cos(q_4) + a_5 \sin(q_4) \cos(q_3) \cos(q_1 + q_2) - a_5 \sin(q_1 + q_2) \cos(q_4)) \sin(q_0) \\ -a_1 \sin(q_1) - a_2 \cos(q_1 + q_2) - a_3 \cos(q_1 + q_2) - a_4 \sin(q_4) \sin(q_1 + q_2) \cos(q_3) - a_4 \cos(q_4) \cos(q_1 + q_2) - a_5 \sin(q_4) \sin(q_1 + q_2) \cos(q_3) - a_5 \cos(q_4) \cos(q_1 + q_2) \\ -\sin(q_0) \\ \cos(q_0) \\ 0 \end{bmatrix}$$

Coluna 2

$$\begin{bmatrix} -(a_2 \sin(q_1 + q_2) + a_3 \sin(q_1 + q_2) - a_4 \sin(q_4) \cos(q_3) \cos(q_1 + q_2) + a_4 \sin(q_1 + q_2) \cos(q_4) - a_5 \sin(q_4) \cos(q_3) \cos(q_1 + q_2) + a_5 \sin(q_1 + q_2) \cos(q_4)) \cos(q_0) \\ -(a_2 \sin(q_1 + q_2) + a_3 \sin(q_1 + q_2) - a_4 \sin(q_4) \cos(q_3) \cos(q_1 + q_2) + a_4 \sin(q_1 + q_2) \cos(q_4) - a_5 \sin(q_4) \cos(q_3) \cos(q_1 + q_2) + a_5 \sin(q_1 + q_2) \cos(q_4)) \sin(q_0) \\ -a_2 \cos(q_1 + q_2) - a_3 \cos(q_1 + q_2) - a_4 \sin(q_4) \sin(q_1 + q_2) \cos(q_3) - a_4 \cos(q_4) \cos(q_1 + q_2) - a_5 \sin(q_4) \sin(q_1 + q_2) \cos(q_3) - a_5 \cos(q_4) \cos(q_1 + q_2) \\ -\sin(q_0) \\ \cos(q_0) \\ 0 \end{bmatrix}$$

Coluna 3



# Jacobiano do 3R com Punho Esférico

$$\begin{aligned} & (a_4 \sin(q_0) \cos(q_3) - a_4 \sin(q_3) \sin(q_1 + q_2) \cos(q_0) + a_5 \sin(q_0) \cos(q_3) - a_5 \sin(q_3) \sin(q_1 + q_2) \cos(q_0)) \sin(q_4) \\ & - (a_4 \sin(q_0) \sin(q_3) \sin(q_1 + q_2) + a_4 \cos(q_0) \cos(q_3) + a_5 \sin(q_0) \sin(q_3) \sin(q_1 + q_2) + a_5 \cos(q_0) \cos(q_3)) \sin(q_4) \\ & - (a_4 + a_5) \sin(q_3) \sin(q_4) \cos(q_1 + q_2) \\ & \quad \cos(q_0) \cos(q_1 + q_2) \\ & \quad \sin(q_0) \cos(q_1 + q_2) \\ & \quad - \sin(q_1 + q_2) \end{aligned}$$

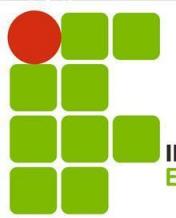
Coluna 4

$$\begin{aligned} & a_4 \sin(q_0) \sin(q_3) \cos(q_4) - a_4 \sin(q_0) \cos(q_1 + q_2) + a_4 \sin(q_1 + q_2) \cos(q_0) \cos(q_3) \cos(q_4) + a_5 \sin(q_0) \sin(q_3) \cos(q_4) - a_5 \sin(q_4) \cos(q_0) \cos(q_1 + q_2) + a_5 \sin(q_1 + q_2) \cos(q_0) \cos(q_3) \cos(q_4) \\ & - a_4 \sin(q_0) \sin(q_4) \cos(q_1 + q_2) + a_4 \sin(q_0) \sin(q_1 + q_2) \cos(q_3) \cos(q_4) - a_4 \sin(q_3) \cos(q_0) \cos(q_4) - a_5 \sin(q_0) \sin(q_4) \cos(q_1 + q_2) + a_5 \sin(q_0) \sin(q_1 + q_2) \cos(q_3) \cos(q_4) - a_5 \sin(q_3) \cos(q_0) \cos(q_4) \\ & a_4 \sin(q_4) \sin(q_1 + q_2) + a_4 \cos(q_3) \cos(q_4) \cos(q_1 + q_2) + a_5 \sin(q_4) \sin(q_1 + q_2) + a_5 \cos(q_3) \cos(q_4) \cos(q_1 + q_2) \\ & \quad \sin(q_0) \cos(q_3) - \sin(q_3) \sin(q_1 + q_2) \cos(q_0) \\ & \quad - \sin(q_0) \sin(q_3) \sin(q_1 + q_2) - \cos(q_0) \cos(q_3) \\ & \quad - \sin(q_3) \cos(q_1 + q_2) \end{aligned}$$

Coluna 5

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ (\sin(q_4) \sin(q_1 + q_2) \cos(q_3) + \cos(q_4) \cos(q_1 + q_2)) \cos(q_0) + \sin(q_0) \sin(q_3) \sin(q_4) \\ (\sin(q_4) \sin(q_1 + q_2) \cos(q_3) + \cos(q_4) \cos(q_1 + q_2)) \sin(q_0) - \sin(q_3) \sin(q_4) \cos(q_0) \\ \sin(q_4) \cos(q_3) \cos(q_1 + q_2) - \sin(q_1 + q_2) \cos(q_4) \end{bmatrix}$$

Coluna 6



---

# Sinais de Erro para Realimentar Controlador Controle Resolvido por Taxa de Movimento



# O Erro de Posição

- O erro de posição é cartesiano, isto é, pode ser retirado diretamente da subtração das variáveis do Alvo (desejadas) e do Efetuador Final (calculado a partir dos encoders do robô e Cinemática Direta, por exemplo):

$$\tilde{x}_{pos} = x_{d_{pos}} - x_{pos} = \begin{bmatrix} x_d - x \\ y_d - y \\ z_d - z \end{bmatrix}$$



# O Erro de Orientação

---

- Porém, o erro de **orientação não é trivial**;
- Sciavicco e Scililiano (2000), na seção 3.7.3 – *Orientation Error*, informam não é possível lidar de forma ingênuas com o erro angular;
- É necessário fazer uso de notações de orientação do tipo vetor ângulo ou **quatérnios** para mensurar erros angulares;
- Os **quatérnios unitários** são uma representação de orientação espacial 3D que, diferentemente dos ângulos de Euler e das Matrizes de Rotação, não possuem singularidades e são recomendados para este fim.



# O Erro de Orientação

- Quatérnios unitários são outra forma de representar uma orientação espacial;
- Trata-se de uma representação abstrata de 4 dimensões, composto de uma parte escalar ( $\eta$ ) e três partes vetoriais ( $\epsilon$ );
- Numericamente falando um quatérnio é um vetor de 4 números reais, isto é:

$$Q = \{\eta, \epsilon\} \in \mathbb{R}^4$$

**Observação:**

Um quatérnio é equivalente em 3 dimensões de um número complexo



# O Erro de Orientação

- Os quatérnios podem ser obtidos de matrizes de rotação e reconvertisdos nelas;
- Assim, dada uma matriz  $\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$
- O quatérnio equivalente é obtido como:

$$Q = \{\eta, \epsilon\} \quad \eta = \frac{1}{2}\sqrt{r_{11} + r_{22} + r_{33} + 1}$$

$$\epsilon = \frac{1}{2} \begin{bmatrix} \operatorname{sgn}(r_{32} - r_{23})\sqrt{r_{11} - r_{22} - r_{33} + 1} \\ \operatorname{sgn}(r_{13} - r_{31})\sqrt{r_{22} - r_{33} - r_{11} + 1} \\ \operatorname{sgn}(r_{21} - r_{12})\sqrt{r_{33} - r_{11} - r_{22} + 1} \end{bmatrix}$$

$\operatorname{sgn}(x) = 1$  for  $x \geq 0$

$\operatorname{sgn}(x) = -1$  for  $x < 0$



# O Erro de Orientação

- Por fim, supondo que são dados:
  - Orientação desejada da ferramenta através de uma Matriz de Rotação ( $R_d$ );
  - Orientação atual do Efetuador Final ( $R_q$ ) – suponha calculo obtido da Cinemática Direta e dos sensores das juntas do robô;
- O erro de orientação ( $e_O$ ) pode ser calculado como (Sciavicco; Scililiano, 2000):

$$e_O = \Delta\epsilon = \eta(\mathbf{q})\epsilon_d - \eta_d\epsilon(\mathbf{q}) - \mathbf{S}(\epsilon_d)\epsilon(\mathbf{q}), \quad (3.87)$$

- Sendo  $\mathbf{S}(\epsilon_d)$  o operador de Matriz Antissimétrica no vetor  $\epsilon_d$

$$\epsilon_d = [\epsilon_0 \quad \epsilon_1 \quad \epsilon_2] \quad \mathbf{S}(\epsilon_d) = \begin{bmatrix} 0 & -\epsilon_2 & \epsilon_1 \\ \epsilon_2 & 0 & -\epsilon_0 \\ -\epsilon_1 & \epsilon_0 & 0 \end{bmatrix}$$