

## **TRABALHO 001 - ROBÓTICA COMPUTACIONAL**



Pontifícia Universidade  
Católica de Campinas

**Aluno:** João Paulo Nunes Andrade

**RA:** 25002703

**Email:** joao.pna@puccampinas.edu.br

**Disciplina:** Robótica de Computacional

**Professor:** Douglas de Assis Ferreira

**Data de Entrega:** 05/09/2025

---



# Respostas das Questões

## Questão 1: Definição de Termos

**Acurácia:** Refere-se à capacidade de um robô alcançar um ponto desejado no espaço. Em outras palavras, é o quão próximo o robô consegue chegar de uma posição de destino. Uma alta acurácia significa que a diferença entre a posição teórica (calculada pelo modelo cinemático) e a posição real é mínima.

**Precisão:** É a capacidade de um robô retornar a um ponto previamente ensinado com repetibilidade. Um robô pode ter alta precisão, mas baixa acurácia. Isso significa que ele pode sempre retornar ao mesmo ponto, mas esse ponto pode não ser exatamente o desejado. A precisão é crucial para tarefas de repetibilidade, como linhas de montagem.

**Resolução:** É a menor alteração que um robô pode fazer em sua posição ou orientação. A resolução depende da tecnologia usada nos atuadores e sensores de posição do robô. Uma alta resolução permite movimentos mais finos e detalhados.

**Junta:** A junta é a parte do robô que permite o movimento relativo entre dois elos. É como as articulações no corpo humano. As juntas podem ser rotacionais (como um ombro) ou prismáticas (como um pistão), e são elas que dão ao robô seus graus de liberdade.

**Elo (Link):** É um corpo rígido que conecta duas juntas. Os elos formam a "estrutura óssea" do robô. A cinemática estuda a relação entre o movimento das juntas e a posição dos elos.

**Espaço de trabalho (workspace):** É o volume de espaço que a ponta do robô (o *end-effector*) pode alcançar. O espaço de trabalho é determinado pelo comprimento dos elos e pela amplitude de movimento de cada junta.

**Cinemática direta:** É o cálculo da posição e orientação da ponta do robô a partir dos valores dos ângulos ou posições de suas juntas. Com as informações de cada junta, o modelo cinemático direto te dá a localização final do robô.

**Cinemática inversa:** É o processo oposto à cinemática direta. A cinemática inversa calcula os valores das juntas (ângulos) necessários para que a ponta do robô alcance uma posição e orientação desejadas. Este cálculo é muito mais complexo, pois pode haver múltiplas soluções ou até nenhuma, dependendo da posição.



## **Questão 2: Classificação de Robôs por Geometria**

Os robôs manipuladores são classificados com base em sua configuração geométrica, que define o formato do seu espaço de trabalho e as suas capacidades de movimento. As principais geometrias são:

**Robô Cartesiano:** Utiliza apenas juntas prismáticas (deslizantes), que se movem em três eixos perpendiculares (X, Y e Z). Possui um espaço de trabalho em formato de cubo ou paralelepípedo, com alta precisão e rigidez.

**Robô Cilíndrico:** Combina juntas rotacionais e prismáticas. Tem uma junta rotacional na base e duas juntas prismáticas para os movimentos vertical e radial. Seu espaço de trabalho é em formato de cilindro.

**Robô Esférico (ou Polar):** Usa uma junta rotacional na base e duas juntas rotacionais para movimentos de inclinação e rotação. Opera com coordenadas polares, resultando em um espaço de trabalho esférico.

**Robô Articulado (ou de Junta Rotacional):** É o tipo mais comum e flexível, utilizando apenas juntas rotacionais. Geralmente possui 3 a 7 juntas, como um braço humano, e um espaço de trabalho complexo e irregular.

**Robô SCARA:** Especializado em montagem vertical. Possui duas juntas rotacionais horizontais e uma junta prismática vertical, sendo rígido no eixo vertical e flexível nos eixos horizontais.

**Robô Delta:** É um tipo de robô paralelo, com vários braços conectados a uma única plataforma móvel. Conhecido por sua alta velocidade e precisão em tarefas de "pick-and-place".



### **Questão 3: Prós e Contras da Automação**

A automação de fábricas por robôs tem um profundo impacto social, com diversas vantagens e desvantagens:

#### **Prós da Automatização:**

**Aumento da Produtividade e Eficiência:** Robôs podem trabalhar 24/7, aumentando a produção.

**Melhora da Qualidade e Consistência:** Realizam tarefas repetitivas com alta precisão, eliminando erros humanos.

**Segurança no Trabalho:** Tarefas perigosas ou insalubres podem ser transferidas para robôs, protegendo os trabalhadores.

**Criação de Novas Oportunidades:** A automação cria empregos em programação, manutenção e engenharia de robótica.

#### **Contras da Automatização:**

**Perda de Empregos:** Pode resultar em desemprego, especialmente para trabalhadores que realizam tarefas manuais e repetitivas.

**Aumento da Desigualdade Social:** A demanda por trabalhadores qualificados aumenta, enquanto a por mão de obra menos qualificada diminui.

**Custo Inicial Elevado:** O investimento em robótica é alto, o que pode dar uma vantagem competitiva a grandes empresas.

**Dependência Tecnológica:** As fábricas se tornam dependentes da tecnologia, e falhas no sistema podem interromper a produção.



## Questão 4: Modelo Cinemático do Robô LWA3 (Código MATLAB)

O objetivo é construir o modelo cinemático direto para o robô Schunk LWA3 usando o método de Denavit-Hartenberg (DH).

Tabela de Parâmetros de DH:

Junta	ai	(mm)	αi	Graus	di	(mm)	0i	Graus
1	0	90	290	01	//	//	//	//
2	0	-90	0	0202	90	//	//	//
3	0	90	328	03	//	//	//	//
4	0	-90	0	04	90	//	//	//
5	0	90	276.5	05	//	//	//	//
6	0	-90	0	06	90	//	//	//
7	0	0	171.7	07	//	//	//	//

### Código e Modelo MATLAB ([modelo\\_LWA3.m](#))

```
% Configuração Inicial e Definição de Variáveis
% Limpa o workspace, fecha figuras e limpa a janela de comando.
clear all
close all
clc

% Define as variáveis simbólicas para as dimensões e ângulos das juntas.
syms th dx dy dz L1 L2 L3 L4 L5 L6 L7 real
% Variáveis de junta para cada grau de liberdade.
syms th1 th2 th3 th4 th5 th6 th7 real
% Matrizes genéricas de translação e rotação para a cinemática.
T = [1 0 0 dx
     0 1 0 dy
     0 0 1 dz
     0 0 0 1];
Rx = [1 0 0 0
     0 cos(th) -sin(th) 0
     0 sin(th) cos(th) 0
     0 0 0 1];
Rz = [cos(th) -sin(th) 0 0
     sin(th) cos(th) 0 0
     0 0 1 0
     0 0 0 1];

%% Encadeamento de Frames e Cinemática Direta
% Frame {0} (base do robô), a matriz identidade.
F0 = eye(4);
% Encadeamento de transformações para cada junta.
Tr_0_1 = double(subs(T, [dx dy dz], [0 0 0]));
TH_0_1 = Tr_0_1 * subs(Rz, th, th1);
```



```

F1 = (F0 * TH_0_1);
Tr_1_2 = subs(T , [dx dy dz] , [0 0 L1]) * subs(Rx , th, deg2rad(+90));
TH_1_2 = Tr_1_2 * subs(Rz, th, th2);
F2 = (F1 * TH_1_2);
Tr_2_3 = subs(T , [dx dy dz] , [0 L2 0]) * subs(Rx , th, deg2rad(-90));
TH_2_3 = Tr_2_3 * subs(Rz, th, th3);
F3 = (F2 * TH_2_3);
Tr_3_4 = subs(T , [dx dy dz] , [0 0 L3]) * subs(Rx , th, deg2rad(+90));
TH_3_4 = Tr_3_4 * subs(Rz, th, th4);
F4 = (F3 * TH_3_4);
Tr_4_5 = subs(T , [dx dy dz] , [0 L4 0]) * subs(Rx , th, deg2rad(-90));
TH_4_5 = Tr_4_5 * subs(Rz, th, th5);
F5 = (F4 * TH_4_5);
Tr_5_6 = subs(T , [dx dy dz] , [0 0 L5]) * subs(Rx , th, deg2rad(+90));
TH_5_6 = Tr_5_6 * subs(Rz, th, th6);
F6 = (F5 * TH_5_6);
Tr_6_7 = subs(T , [dx dy dz] , [0 L6 0]) * subs(Rx , th, deg2rad(-90));
TH_6_7 = Tr_6_7 * subs(Rz, th, th7);
F7 = (F6 * TH_6_7);
% Frame {8} (punho do robô).
Tr_7_8 = subs(T , [dx dy dz] , [0 0 L7]);
F8 = (F7 * Tr_7_8);
%% Atribuição de valores e cálculo numérico
% Valores de comprimento dos elos em metros.
l1 = 0.18;
l2 = 0.169;
l3 = 0.159;
l4=0.14825;l5=0.12825;l6=0.12585;
l7=0.04585;
% Posição das juntas em graus, convertida para radianos.
t1 = deg2rad(0);
t2 = deg2rad(0);
t3 = deg2rad(0);
t4 = deg2rad(0);
t5 = deg2rad(0);
t6 = deg2rad(0);
t7 = deg2rad(45);
% Substitui as variáveis simbólicas por valores numéricos.
F1 = double(subs(F1 , th1 , t1));
F2 = double(subs(F2 , [L1 th1 th2], [l1 t1 t2]));
F3 = double(subs(F3 , [L2 L1 th1 th2 th3], [l2 l1 t1 t2 t3]));
F4 = double(subs(F4 , [L3 L2 L1 th1 th2 th3 th4], [l3 l2 l1 t1 t2 t3 t4]));
F5 = double(subs(F5 , [L4 L3 L2 L1 th1 th2 th3 th4 th5], [l4 l3 l2 l1 t1 t2 t3 t4 t5]));
F6 = double(subs(F6 , [L5 L4 L3 L2 L1 th1 th2 th3 th4 th5 th6], [l5 l4 l3 l2 l1 t1 t2 t3 t4 t5 t6]));
F7 = double(subs(F7 , [L6 L5 L4 L3 L2 L1 th1 th2 th3 th4 th5 th6 th7], [l6 l5 l4 l3 l2 l1 t1 t2 t3 t4 t5 t6 t7]));
F8 = double(subs(F8 , [L7 L6 L5 L4 L3 L2 L1 th1 th2 th3 th4 th5 th6 th7], [l7 l6 l5 l4 l3 l2 l1 t1 t2 t3 t4 t5 t6 t7]));
% Fatores de escala e marcação para o plot.
esc = 0.05;
mark=10;
%% Visualização Gráfica do Robô
% Configura a figura para plotagem 3D.
plot3(F0(1,4) , F0(2,4) , F0(3,4) , 'om', 'linewidth', 2 , 'markersize', mark);
axis equal;
grid on;
xlabel('x'); ylabel('y'); zlabel('z');
hold on;
% Plota o frame {0}.
plot3([F0(1,4) F0(1,4)+esc*F0(1,1)] , [F0(2,4) F0(2,4)+esc*F0(2,1)] , ...
[F0(3,4) F0(3,4)+esc*F0(3,1)] , 'b', 'linewidth', 2)
text(F0(1,4)+esc*F0(1,1) , F0(2,4)+esc*F0(2,1) , F0(3,4)+esc*F0(3,1) , 'x_{\{0\}}')
plot3([F0(1,4) F0(1,4)+esc*F0(1,2)] , [F0(2,4) F0(2,4)+esc*F0(2,2)] , [F0(3,4)
F0(3,4)+esc*F0(3,2)] , 'g', 'linewidth', 2)

```



```

text(F0(1,4)+esc*F0(1,2) , F0(2,4)+esc*F0(2,2) , F0(3,4)+esc*F0(3,2) , 'y_{\{0\}}')
plot3([F0(1,4) F0(1,4)+esc*F0(1,3)] , [F0(2,4) F0(2,4)+esc*F0(2,3)] , [F0(3,4)
F0(3,4)+esc*F0(3,3)] , 'r', 'linewidth', 2)
text(F0(1,4)+esc*F0(1,3) , F0(2,4)+esc*F0(2,3) , F0(3,4)+esc*F0(3,3) , 'z_{\{0\}}')
plot3([F0(1,4) F1(1,4)] , [F0(2,4) F1(2,4)] , [F0(3,4) F1(3,4)] , 'k')
% Plota o frame {1}.
plot3(F1(1,4) , F1(2,4) , F1(3,4) , 'om', 'linewidth', 2 , 'markersize', mark);
text(F1(1,4) , F1(2,4) , F1(3,4)-0.2 , '\{1\}')
plot3([F1(1,4) F1(1,4)+esc*F1(1,1)] , [F1(2,4) F1(2,4)+esc*F1(2,1)] , [F1(3,4)
F1(3,4)+esc*F1(3,1)] , 'b', 'linewidth', 2)
text(F1(1,4)+esc*F1(1,1) , F1(2,4)+esc*F1(2,1) , F1(3,4)+esc*F1(3,1) , 'x_{\{1\}}')
plot3([F1(1,4) F1(1,4)+esc*F1(1,2)] , [F1(2,4) F1(2,4)+esc*F1(2,2)] , [F1(3,4)
F1(3,4)+esc*F1(3,2)] , 'g', 'linewidth', 2)
text(F1(1,4)+esc*F1(1,2) , F1(2,4)+esc*F1(2,2) , F1(3,4)+esc*F1(3,2) , 'y_{\{1\}}')
plot3([F1(1,4) F1(1,4)+esc*F1(1,3)] , [F1(2,4) F1(2,4)+esc*F1(2,3)] , [F1(3,4)
F1(3,4)+esc*F1(3,3)] , 'r', 'linewidth', 2)
text(F1(1,4)+esc*F1(1,3) , F1(2,4)+esc*F1(2,3) , F1(3,4)+esc*F1(3,3) , 'z_{\{1\}}')
plot3([F1(1,4) F2(1,4)] , [F1(2,4) F2(2,4)] , [F1(3,4) F2(3,4)] , 'k')
% Plota o frame {2}.
plot3(F2(1,4) , F2(2,4) , F2(3,4) , 'om', 'linewidth', 2 , 'markersize', mark);
text(F2(1,4) , F2(2,4) , F2(3,4)-0.2 , '\{2\}')
plot3([F2(1,4) F2(1,4)+esc*F2(1,1)] , [F2(2,4) F2(2,4)+esc*F2(2,1)] , [F2(3,4)
F2(3,4)+esc*F2(3,1)] , 'b', 'linewidth', 2)
text(F2(1,4)+esc*F2(1,1) , F2(2,4)+esc*F2(2,1) , F2(3,4)+esc*F2(3,1) , 'x_{\{2\}}')
plot3([F2(1,4) F2(1,4)+esc*F2(1,2)] , [F2(2,4) F2(2,4)+esc*F2(2,2)] , [F2(3,4)
F2(3,4)+esc*F2(3,2)] , 'g', 'linewidth', 2)
text(F2(1,4)+esc*F2(1,2) , F2(2,4)+esc*F2(2,2) , F2(3,4)+esc*F2(3,2) , 'y_{\{2\}}')
plot3([F2(1,4) F2(1,4)+esc*F2(1,3)] , [F2(2,4) F2(2,4)+esc*F2(2,3)] , [F2(3,4)
F2(3,4)+esc*F2(3,3)] , 'r', 'linewidth', 2)
text(F2(1,4)+esc*F2(1,3) , F2(2,4)+esc*F2(2,3) , F2(3,4)+esc*F2(3,3) , 'z_{\{2\}}')
plot3([F2(1,4) F3(1,4)] , [F2(2,4) F3(2,4)] , [F2(3,4) F3(3,4)] , 'k')
% Plota o frame {3}.
plot3(F3(1,4) , F3(2,4) , F3(3,4) , 'om', 'linewidth', 2 , 'markersize', mark);
text(F3(1,4) , F3(2,4) , F3(3,4)-0.2 , '\{3\}')
plot3([F3(1,4) F3(1,4)+esc*F3(1,1)] , [F3(2,4) F3(2,4)+esc*F3(2,1)] , [F3(3,4)
F3(3,4)+esc*F3(3,1)] , 'b', 'linewidth', 2)
text(F3(1,4)+esc*F3(1,1) , F3(2,4)+esc*F3(2,1) , F3(3,4)+esc*F3(3,1) , 'x_{\{3\}}')
plot3([F3(1,4) F3(1,4)+esc*F3(1,2)] , [F3(2,4) F3(2,4)+esc*F3(2,2)] , [F3(3,4)
F3(3,4)+esc*F3(3,2)] , 'g', 'linewidth', 2)
text(F3(1,4)+esc*F3(1,2) , F3(2,4)+esc*F3(2,2) , F3(3,4)+esc*F3(3,2) , 'y_{\{3\}}')
plot3([F3(1,4) F3(1,4)+esc*F3(1,3)] , [F3(2,4) F3(2,4)+esc*F3(2,3)] , [F3(3,4)
F3(3,4)+esc*F3(3,3)] , 'r', 'linewidth', 2)
text(F3(1,4)+esc*F3(1,3) , F3(2,4)+esc*F3(2,3) , F3(3,4)+esc*F3(3,3) , 'z_{\{3\}}')
plot3([F3(1,4) F4(1,4)] , [F3(2,4) F4(2,4)] , [F3(3,4) F4(3,4)] , 'k')
% Plota o frame {4}.
plot3(F4(1,4) , F4(2,4) , F4(3,4) , 'om', 'linewidth', 2 , 'markersize', mark);
text(F4(1,4) , F4(2,4) , F4(3,4)-0.2 , '\{4\}')
plot3([F4(1,4) F4(1,4)+esc*F4(1,1)] , [F4(2,4) F4(2,4)+esc*F4(2,1)] , [F4(3,4)
F4(3,4)+esc*F4(3,1)] , 'b', 'linewidth', 2)
text(F4(1,4)+esc*F4(1,1) , F4(2,4)+esc*F4(2,1) , F4(3,4)+esc*F4(3,1) , 'x_{\{4\}}')
plot3([F4(1,4) F4(1,4)+esc*F4(1,2)] , [F4(2,4) F4(2,4)+esc*F4(2,2)] , [F4(3,4)
F4(3,4)+esc*F4(3,2)] , 'g', 'linewidth', 2)
text(F4(1,4)+esc*F4(1,2) , F4(2,4)+esc*F4(2,2) , F4(3,4)+esc*F4(3,2) , 'y_{\{4\}}')
plot3([F4(1,4) F4(1,4)+esc*F4(1,3)] , [F4(2,4) F4(2,4)+esc*F4(2,3)] , [F4(3,4)
F4(3,4)+esc*F4(3,3)] , 'r', 'linewidth', 2)
text(F4(1,4)+esc*F4(1,3) , F4(2,4)+esc*F4(2,3) , F4(3,4)+esc*F4(3,3) , 'z_{\{4\}}')
plot3([F4(1,4) F5(1,4)] , [F4(2,4) F5(2,4)] , [F4(3,4) F5(3,4)] , 'k')
% Plota o frame {5}.
plot3(F5(1,4) , F5(2,4) , F5(3,4) , 'om', 'linewidth', 2 , 'markersize', mark);
text(F5(1,4) , F5(2,4) , F5(3,4)-0.2 , '\{5\}')
plot3([F5(1,4) F5(1,4)+esc*F5(1,1)] , [F5(2,4) F5(2,4)+esc*F5(2,1)] , [F5(3,4)
F5(3,4)+esc*F5(3,1)] , 'b', 'linewidth', 2)
text(F5(1,4)+esc*F5(1,1) , F5(2,4)+esc*F5(2,1) , F5(3,4)+esc*F5(3,1) , 'x_{\{5\}}')
plot3([F5(1,4) F5(1,4)+esc*F5(1,2)] , [F5(2,4) F5(2,4)+esc*F5(2,2)] , [F5(3,4)
F5(3,4)+esc*F5(3,2)] , 'g', 'linewidth', 2)

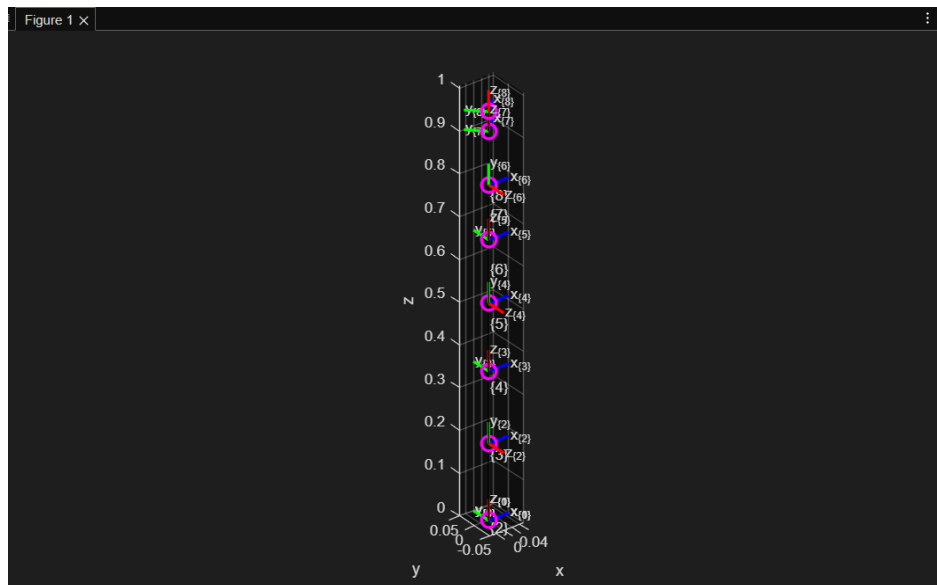
```



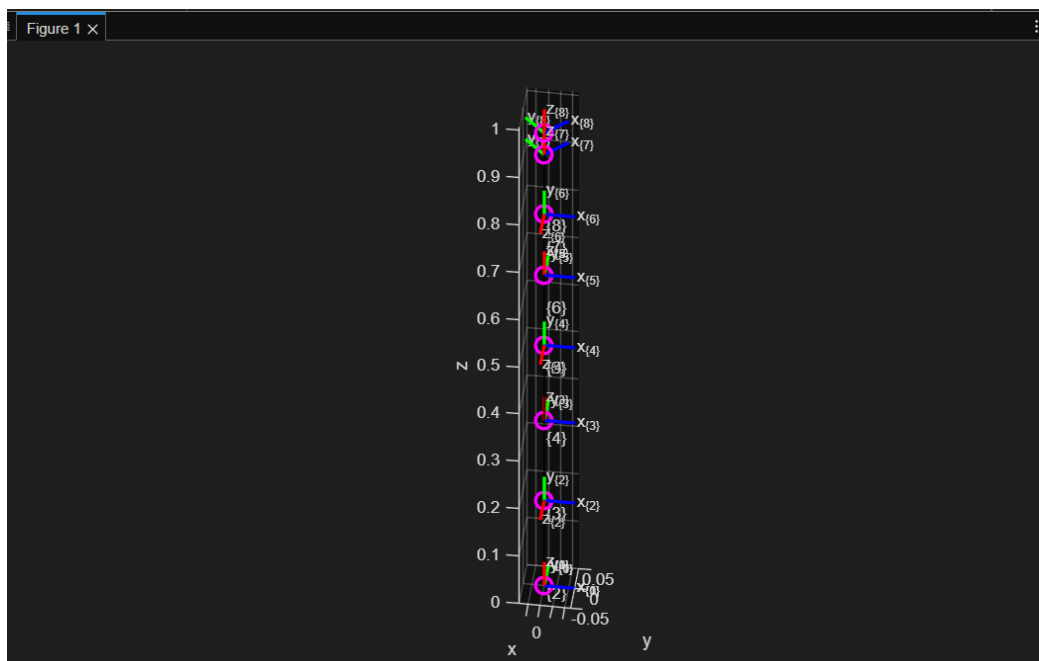
```

text(F5(1,4)+esc*F5(1,2), F5(2,4)+esc*F5(2,2), F5(3,4)+esc*F5(3,2), 'y_{\{5\}}')
plot3([F5(1,4) F5(1,4)+esc*F5(1,3)], [F5(2,4) F5(2,4)+esc*F5(2,3)], [F5(3,4)
F5(3,4)+esc*F5(3,3)], 'r','linewidth',2)
text(F5(1,4)+esc*F5(1,3), F5(2,4)+esc*F5(2,3), F5(3,4)+esc*F5(3,3), 'z_{\{5\}}')
plot3([F5(1,4) F6(1,4)], [F5(2,4) F6(2,4)], [F5(3,4) F6(3,4)], 'k')
% Plota o frame {6}.
plot3(F6(1,4), F6(2,4), F6(3,4), 'om','linewidth',2,'markersize',mark);
text(F6(1,4), F6(2,4), F6(3,4)-0.2, '\{6\}')
plot3([F6(1,4) F6(1,4)+esc*F6(1,1)], [F6(2,4) F6(2,4)+esc*F6(2,1)], [F6(3,4)
F6(3,4)+esc*F6(3,1)], 'b','linewidth',2)
text(F6(1,4)+esc*F6(1,1), F6(2,4)+esc*F6(2,1), F6(3,4)+esc*F6(3,1), 'x_{\{6\}}')
plot3([F6(1,4) F6(1,4)+esc*F6(1,2)], [F6(2,4) F6(2,4)+esc*F6(2,2)], [F6(3,4)
F6(3,4)+esc*F6(3,2)], 'g','linewidth',2)
text(F6(1,4)+esc*F6(1,2), F6(2,4)+esc*F6(2,2), F6(3,4)+esc*F6(3,2), 'y_{\{6\}}')
plot3([F6(1,4) F6(1,4)+esc*F6(1,3)], [F6(2,4) F6(2,4)+esc*F6(2,3)], [F6(3,4)
F6(3,4)+esc*F6(3,3)], 'r','linewidth',2)
text(F6(1,4)+esc*F6(1,3), F6(2,4)+esc*F6(2,3), F6(3,4)+esc*F6(3,3), 'z_{\{6\}}')
plot3([F6(1,4) F7(1,4)], [F6(2,4) F7(2,4)], [F6(3,4) F7(3,4)], 'k')
% Plota o frame {7}.
plot3(F7(1,4), F7(2,4), F7(3,4), 'om','linewidth',2,'markersize',mark);
text(F7(1,4), F7(2,4), F7(3,4)-0.2, '\{7\}')
plot3([F7(1,4) F7(1,4)+esc*F7(1,1)], [F7(2,4) F7(2,4)+esc*F7(2,1)], [F7(3,4)
F7(3,4)+esc*F7(3,1)], 'b','linewidth',2)
text(F7(1,4)+esc*F7(1,1), F7(2,4)+esc*F7(2,1), F7(3,4)+esc*F7(3,1), 'x_{\{7\}}')
plot3([F7(1,4) F7(1,4)+esc*F7(1,2)], [F7(2,4) F7(2,4)+esc*F7(2,2)], [F7(3,4)
F7(3,4)+esc*F7(3,2)], 'g','linewidth',2)
text(F7(1,4)+esc*F7(1,2), F7(2,4)+esc*F7(2,2), F7(3,4)+esc*F7(3,2), 'y_{\{7\}}')
plot3([F7(1,4) F7(1,4)+esc*F7(1,3)], [F7(2,4) F7(2,4)+esc*F7(2,3)], [F7(3,4)
F7(3,4)+esc*F7(3,3)], 'r','linewidth',2)
text(F7(1,4)+esc*F7(1,3), F7(2,4)+esc*F7(2,3), F7(3,4)+esc*F7(3,3), 'z_{\{7\}}')
plot3([F7(1,4) F8(1,4)], [F7(2,4) F8(2,4)], [F7(3,4) F8(3,4)], 'k')
% Plota o frame {8}.
plot3(F8(1,4), F8(2,4), F8(3,4), 'om','linewidth',2,'markersize',mark);
text(F8(1,4), F8(2,4), F8(3,4)-0.2, '\{8\}')
plot3([F8(1,4) F8(1,4)+esc*F8(1,1)], [F8(2,4) F8(2,4)+esc*F8(2,1)], [F8(3,4)
F8(3,4)+esc*F8(3,1)], 'b','linewidth',2)
text(F8(1,4)+esc*F8(1,1), F8(2,4)+esc*F8(2,1), F8(3,4)+esc*F8(3,1), 'x_{\{8\}}')
plot3([F8(1,4) F8(1,4)+esc*F8(1,2)], [F8(2,4) F8(2,4)+esc*F8(2,2)], [F8(3,4)
F8(3,4)+esc*F8(3,2)], 'g','linewidth',2)
text(F8(1,4)+esc*F8(1,2), F8(2,4)+esc*F8(2,2), F8(3,4)+esc*F8(3,2), 'y_{\{8\}}')
plot3([F8(1,4) F8(1,4)+esc*F8(1,3)], [F8(2,4) F8(2,4)+esc*F8(2,3)], [F8(3,4)
F8(3,4)+esc*F8(3,3)], 'r','linewidth',2)
text(F8(1,4)+esc*F8(1,3), F8(2,4)+esc*F8(2,3), F8(3,4)+esc*F8(3,3), 'z_{\{8\}}')

```







## **Referências**

### **Materiais Didáticos e Aulas**

- Material didático utilizado na disciplina, Campinas, SP, 2025.

### **Cursos Complementares**

- SENAI. Lógica de Programação. 2024.
- SENAI. Metrologia. 2024.
- SENAI. Competência Transversal Tecnologia da Informação e Comunicação. 2023.
- MANUAL DO MUNDO. Arduino e Robótica Básica. 2023.

### **Artigos Científicos e Técnicos**

- ALURA. Introdução à Robótica: Conceitos Fundamentais e Aplicações. Disponível em: <https://www.alura.com.br/artigos/introducao-robotica>. Acesso em: 04 de setembro de 2025.
- NVIDIA. IA para Robótica. Disponível em: <https://www.nvidia.com/pt-br/industries/robotics/>. Acesso em: 04 de setembro de



2025.

- NVIDIA. Robotics Fundamentals Learning Path. Disponível em: <https://www.nvidia.com/pt-br/learn/learning-path/robotics/>. Acesso em: 04 de setembro de 2025.

### **Experiências Profissionais**

- ARAÚJO ENGENHARIA. Assistente administrativo de TI e geral. Vinhedo, SP, 2024.
  - ARAÚJO ENGENHARIA. Assistente administrativo de Pesquisa, Desenvolvimento e Inovação (PDI). Vinhedo, SP, 2024.
- 

### **Instruções de Formatação**

Para a formatação deste relatório, foram aplicadas as seguintes normas da ABNT:

- **Fonte:** O texto e todos os elementos do trabalho estão em Arial, com tamanho 11, 12, 13 e 14.
- **Margens:** As margens foram configuradas com 3 cm na superior e esquerda, e 2 cm na inferior e direita.
- **Espaçamento:** O espaçamento entre linhas é de 1,5 cm para o corpo do texto, com espaçamento simples para citações diretas longas, referências e legendas de figuras e tabelas.
- **Alinhamento:** Todo o conteúdo do texto está alinhado de forma justificada.

