



Trabalho de Conclusão de Curso

Resolução de Redundância aplicada a um manipulador robótico com cinco graus de liberdade

de **Hugo Tallys Martins Oliveira**

orientado por

Prof. Dr., UFAL Ícaro Bezerra Queiroz de Araújo

Universidade Federal de Alagoas
Instituto de Computação
Maceió, Alagoas
01 de Fevereiro de 2024

UNIVERSIDADE FEDERAL DE ALAGOAS
Instituto de Computação

**RESOLUÇÃO DE REDUNDÂNCIA APLICADA A UM
MANIPULADOR ROBÓTICO COM CINCO GRAUS DE
LIBERDADE**

Trabalho de Conclusão de Curso submetido
ao Instituto de Computação da Universidade
Federal de Alagoas como requisito parcial
para a obtenção do grau de Engenheiro de
Computação.

Hugo Tallys Martins Oliveira

Orientador: Prof. Dr., UFAL Ícaro Bezerra Queiroz de Araújo

Banca Avaliadora:

Glauber Rodrigues Leite Dr., UFAL
Arthur Costa Vangasse Eng., UFMG

Maceió, Alagoas
01 de Fevereiro de 2024

UNIVERSIDADE FEDERAL DE ALAGOAS
Instituto de Computação

**RESOLUÇÃO DE REDUNDÂNCIA APLICADA A UM
MANIPULADOR ROBÓTICO COM CINCO GRAUS DE
LIBERDADE**

Trabalho de Conclusão de Curso submetido ao Instituto de Computação
da Universidade Federal de Alagoas como requisito parcial para a ob-
tenção do grau de Engenheiro de Computação.

Aprovado em 01 de Fevereiro de 2024:

Ícaro Bezerra Queiroz de Araújo,
Prof. Dr., UFAL, Orientador

Glauber Rodrigues Leite,
Dr., UFAL

Arthur Costa Vangasse,
Eng., UFMG

Dedicatória

Text
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Nome, abreviatura, ou nada, etc

Agradecimentos

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Data, Local

Nome, abreviatura, nada, etc

Study hard what interests you the most in the most undisciplined, irreverent and original manner possible. — Richard Feynmann

Resumo

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Palavras-chave: KW_1 ; KW_2 ; KW_3 ; KW_4 ; KW_5 .

Abstract

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Keywords: KW_1 ; KW_2 ; KW_3 ; KW_4 ; KW_5 .

Listas de Figuras

2.1	Dois sistemas de coordenadas, um ponto P e vetores que o representam.	3
2.2	Atribuição de <i>frames</i> de acordo com a convenção de Denavit-Hartenberg. Adaptado de [Spong et al., 2020].	7
2.3	Braço planar do tipo 3R.	9
2.4	Elipsoide de manipulabilidade para diferentes configurações do manipulador planar 3R.	19
3.1	Modelo virtual do manipulador no Webots.	22
3.2	Cadeia cinemática visualizada no RViz.	24
3.3	Geometria de colisão do modelo virtual do manipulador.	25
3.4	Arquitetura de comunicação proposta para controle do manipulador.	27
3.5	Diagrama de blocos do controlador RRC	27
3.6	Etapas seguidas em cada execução de uma trajectory_action	29
4.1	Resultados	31
4.2	Resultados	31
4.3	Resultados	32
4.4	Resultados	32
4.5	Resultados	33
4.6	Resultados	33
4.7	Resultados	34
4.8	Resultados	34

Lista de Tabelas

2.1	Descrição dos parâmetros Denavit-Hartenberg	8
2.2	Parâmetros DH para o braço planar 3R da figura 2.3.	9
3.1	Parâmetros DH para o manipulador 5R.	23

Lista de Símbolos

\mathbb{R} Conjunto dos números reais.

Lista de Abreviaturas

IC Instituto de Computação.

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Objetivos	1
1.3	Estrutura do texto	1
2	Fundamentação teórica	2
2.1	Poses no espaço	2
2.1.1	Representando posições e orientações no espaço	2
2.1.2	Transformações homogêneas	4
2.2	Cinemática Direta	5
2.2.1	Cadeias cinemáticas	5
2.2.2	Convenção de Denavit-Hartenberg	7
2.2.3	Cinemática direta de um braço planar	8
2.3	Cinemática Diferencial	10
2.3.1	A Jacobiana do Manipulador	10
2.3.2	Singularidades	14
2.3.3	Cinemática Inversa Diferencial	15
2.4	Manipuladores redundantes	16
2.4.1	Pseudo-Inversa da Jacobiana	16
2.4.2	Medida de Manipulabilidade	17
2.4.3	Resolução de Redundância	18
3	Implementação	21
3.1	Simulação de manipuladores robóticos	21
3.2	Arquitetura de comunicação	24
3.3	O Algoritmo <i>Resolved Rate Control</i>	26
3.4	Experimentos	26
4	Resultados	30
5	Conclusão	35

Capítulo 1

Introdução

1.1 Motivação

1.2 Objetivos

- Devise a kinematic model for a redundant manipulator
- Stabilish a kinematic control scheme to generate cartesian trajectories and null space motion

Specific objectives:

- Simulate the task on the denso vp6242 robot - test on the real robot (if possible)
(use ROS on both simulated environemnt and real robot to a seamless integration)

1.3 Estrutura do texto

Capítulo 2

Fundamentação teórica

2.1 Poses no espaço

O estudo da cinemática na robótica envolve principalmente o estabelecimento de sistemas de coordenadas (em inglês, *frames*) para representar posições e orientações de corpos rígidos, além da caracterização das transformações entre esses frames. Nesta seção, definiremos uma forma precisa para representar translações e rotações no espaço por meio de matrizes e vetores. Em seguida, apresentaremos as transformações homogêneas, que combinam as operações de rotação e translação em uma única multiplicação matricial, proporcionando uma maneira concisa de estabelecer a relação entre dois frames distintos.

2.1.1 Representando posições e orientações no espaço

Um sistema de coordenadas ou *frame* é definido por um ponto e um conjunto de vetores ortonormais que formam uma base no espaço considerado (dois vetores no plano e três vetores no espaço tridimensional). No contexto da robótica é muito comum a especificação de uma tarefa com base em coordenadas cartesianas relativas a algum frame de referência como ilustrado na figura 2.1. Podemos tomar o *frame* $o_0x_0y_0z_0$ como referência e expressar qualquer ponto P do espaço através de um vetor com origem em o_0 e extremidade em P de modo que escreveríamos:

$$p^0 = v_0^0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (2.1)$$

onde o expoente 0 indica que o vetor é expresso com coordenadas referentes ao *frame* $o_0x_0y_0z_0$. Contudo, ainda observando a figura 2.1, podemos expressar o mesmo ponto P com coordenadas referentes ao *frame* $o_1x_1y_1z_1$ através do vetor v_1^1 de modo que, teríamos por exemplo:

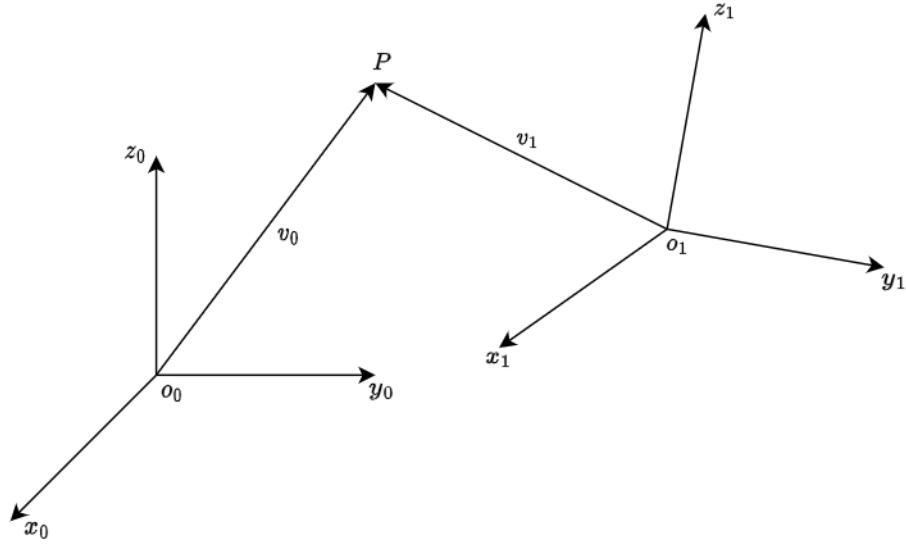


Figura 2.1: Dois sistemas de coordenadas, um ponto P e vetores que o representam.

$$\mathbf{p}^1 = \mathbf{v}_1^1 = \begin{bmatrix} 2 \\ -1.5 \\ 3 \end{bmatrix} \quad (2.2)$$

Além disso, poderíamos também calcular \mathbf{v}_1^0 e \mathbf{v}_0^1 de modo que as coordenadas do vetor dependem do sistema de referência utilizado.

Para estabelecer a relação completa entre os dois *frames* precisamos também considerar a orientação de um com relação ao outro. Podemos expressar relação de orientação entre os *frames* {0} e {1} utilizando a matriz de mudança de base R_1^0 dada por:

$$R_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix} \quad (2.3)$$

onde (\cdot) denota o produto escalar entre dois vetores.

A matriz R_1^0 faz parte de um conjunto de matrizes denominado grupo especial ortogonal de ordem 3 ($SO(3)$) que possui algumas propriedades interessantes dentre elas a facilidade de se calcular sua inversa:

$$(R_1^0)^{-1} = (R_1^0)^\top = R_0^1 \quad (2.4)$$

Se a relação entre os *frames* é composta apenas por uma rotação de θ radianos em torno de cada eixo ordenado, podemos calcular as seguintes matrizes de transformação elementares:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (2.5)$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (2.6)$$

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

2.1.2 Transformações homogêneas

Um deslocamento rígido no espaço pode ser representado por uma translação pura seguida de uma rotação pura. De maneira mais precisa, podemos defini-lo como um par ordenado (R, d) onde $R \in SO(3)$ e $d \in \mathbb{R}^3$. O grupo de todas as transformações rígidas no espaço tridimensional é denominado grupo especial euclidiano de ordem 3 ($SE(3)$) onde vemos claramente que $SE(3) = \mathbb{R}^3 \times SO(3)$.

Seja R_1^0 a matriz de rotação que especifica a orientação do sistema de coordenadas $\{1\}$ com relação ao sistema de coordenadas $\{0\}$ e o_1^0 o vetor de deslocamento une suas origens. Como visto na seção anterior, se P é um ponto fixado ao sistema de coordenadas $\{1\}$ com coordenadas locais dadas por p^1 então, podemos expressar as coordenadas de P com relação ao sistema de coordenadas $\{0\}$ usando o seguinte deslocamento rígido:

$$p^0 = R_1^0 p^1 + o_1^0 \quad (2.8)$$

A equação (2.8) pode ser reescrita de maneira mais compacta utilizando-se a seguinte formulação matricial:

$$\begin{bmatrix} p^0 \\ 1 \end{bmatrix} = \begin{bmatrix} R_1^0 & o_1^0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p^1 \\ 1 \end{bmatrix}, R_1^0 \in SO(3), o_1^0 \in \mathbb{R}^3 \quad (2.9)$$

onde vemos que os elementos de $SE(3)$ são matrizes de dimensão 4×4 e os vetores p^i são representados em *coordenadas homogêneas* com uma dimensão extra. A matriz T_1^0 é denominada matriz de *transformação homogênea* e utilizando o fato de que R_1^0 é uma matriz ortogonal, podemos facilmente calcular sua inversa:

$$(T_1^0)^{-1} = \begin{bmatrix} R_1^0 & o_1^0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} (R_1^0)^\top & -(R_1^0)^\top o_1^0 \\ 0 & 1 \end{bmatrix} \quad (2.10)$$

2.2 Cinemática Direta

O problema da cinemática em manipuladores consiste em descrever o movimento do manipulador sem considerar as forças e torques atuantes sob o mesmo tratando-se, portanto, de uma descrição puramente geométrica. Neste contexto, uma pergunta natural que surge é como podemos determinar a posição e orientação do efetuador final na cadeia cinemática dado um conjunto arbitrário de ângulos articulados. Este problema é conhecido na robótica como a cinemática direta de um manipulador e pode ser facilmente resolvido se associarmos a cada corpo rígido da cadeia um sistema de coordenadas (*frame*), expressando também as relações entre esses *frames* como transformações homogêneas. A pose do efetuador final fica determinada através de uma série de multiplicações matriciais. A colocação de *frames* pode ser feita de maneira sistemática através da utilização da convenção de Denavit-Hartenberg a qual fornece uma abordagem concisa para representar a estrutura geométrica do manipulador.

2.2.1 Cadeias cinemáticas

Na robótica, uma cadeia cinemática pode ser definida como uma série de corpos rígidos, também denominados *elos*, conectados por *juntas* que permitem um movimento relativo das diferentes partes móveis de um manipulador. Cadeias cinemáticas formam a base do estudo de manipuladores robóticos e geralmente são representadas através de um grafo onde seus nós constituem os elos e as arestas as juntas.

Dependendo da topologia desse grafo, podemos classificar uma cadeia cinemática de diferentes formas. Numa cadeia serial aberta seu grafo consiste numa árvore onde cada nó possui apenas um único filho e o nó terminal da árvore usualmente representa o efetuador final (onde uma pinça ou garra robótica ficaria acoplada, por exemplo). Outros casos incluem grafos com ramificações (existe mais de um nó terminal) onde denominamos de cadeia paralela ou quando há a presença de ciclos onde temos uma cadeia fechada. Neste trabalho, iremos nos limitar à análise de cadeias abertas, as mais comuns no âmbito de manipuladores robóticos utilizados em aplicações industriais.

Os tipos de juntas presentes no cadeia também são importantes na definição do alcance e natureza do movimento do manipulador. As mais simples são as juntas prismáticas e de revolução, cada uma introduzindo um único grau de liberdade ao sistema. Juntas prismáticas permitem o movimento translacional ao longo de uma única direção enquanto que juntas de revolução possibilitam um movimento rotacional ao redor de um eixo específico.

Ademais das juntas básicas, tipos mais complexos incluem juntas esféricas, que introduzem dois graus de liberdade (rotação ao redor de dois eixos perpendiculares) e punhos esféricos, compostos por três juntas revolutas dispostas ortogonalmente, introduzindo três graus de liberdade ao sistema. Vale ressaltar que independente da complexidade da junta,

a maior parte pode ser reduzida a uma combinação dos dois tipos mais simples tornando suficiente a descrição de cadeias cinemáticas por meio de uma combinação de juntas prismáticas ou de revolução.

Um manipulador robótico com n juntas terá $n+1$ elos, uma vez que cada junta conecta exatamente dois elos. Iremos enumerar as juntas de 1 até n e elos de 0 a n sendo que o elo de número 0 representará a base do manipulador e elo n seu efetuador final. De acordo com essa convenção a junta i conecta os elos $i-1$ ao i e tem sua posição fixa com respeito ao elo anterior. Quando a junta i é atuada, o elo i se movimenta de modo que a base permanece fixa independente de qual junta é movimentada.

Iremos associar à i -ésima junta uma variável q_i representando no caso de uma junta de revolução o ângulo de rotação e no caso de uma junta prismática o deslocamento linear:

$$q_i = \begin{cases} \theta_i & \text{se a junta } i \text{ é de revolução} \\ d_i & \text{se a junta } i \text{ é prismática} \end{cases} \quad (2.11)$$

A análise cinemática é feita anexando ao elo i da cadeia o sistema de coordenadas $o_i x_i y_i z_i$. O *frame* $o_0 x_0 y_0 z_0$ associado à base do manipulador é denominado *frame* da base, inercial ou do mundo. Se $A_i(q_i)$ é a matriz de transformação homogênea que fornece a posição e orientação de $o_i x_i y_i z_i$ com relação a $o_{i-1} x_{i-1} y_{i-1} z_{i-1}$ então podemos dizer que a mesma é função unicamente da variável q_i de modo que:

$$A(q_i) = A_i = \begin{bmatrix} R_i^{i-1} & o_i^{i-1} \\ 0 & 1 \end{bmatrix} \quad (2.12)$$

Dessa forma, para $i < j$, a matriz T_j^i dada por:

$$T_j^i = A_{i+1} \cdots A_j = \begin{bmatrix} R_j^i & o_j^i \\ 0 & 1 \end{bmatrix} \quad (2.13)$$

expressa a orientação R_j^i e posição o_j^i de $o_j x_j y_j z_j$ com relação a $o_i x_i y_i z_i$. Vale ressaltar que a matriz R_j^i é calculada através da multiplicação matricial:

$$R_j^i = R_{i+1}^i \cdots R_j^{j-1} \quad (2.14)$$

e o vetor de posição utilizando-se a seguinte equação recursiva:

$$o_j^i = o_{j-1}^i + R_{j-1}^i o_j^{j-1} \quad (2.15)$$

O problema da cinemática direta pode ser então formulado como o simples cálculo da matriz T_n^0 que expressa a pose do efetuador final com relação ao *frame* da base:

$$T_n^0 = A_1 A_2 \cdots A_n = \begin{bmatrix} R_n^0 & o_n^0 \\ 0 & 1 \end{bmatrix} \quad (2.16)$$

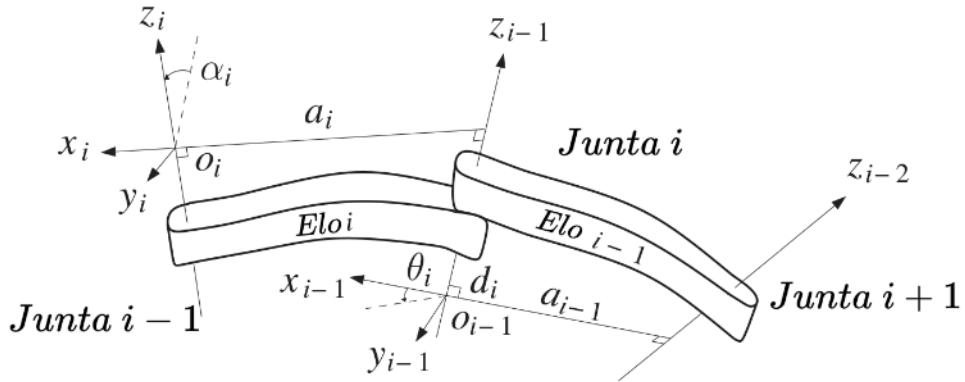


Figura 2.2: Atribuição de *frames* de acordo com a convenção de Denavit-Hartenberg. Adaptado de [Spong et al., 2020].

Tendo em vista a infinidade de possibilidades de se anexar os *frames* em cada elo, para poder calcular as matrizes A_i de forma mais precisa, vamos estabelecer uma convenção utilizando para isso os parâmetros introduzidos por Denavit e Hartenberg.

2.2.2 Convenção de Denavit-Hartenberg

A fixação de frames em cada elo pode ser feita de maneira arbitrária para se obter as matrizes de transformação A_i , permitindo assim o cálculo da cinemática direta. Contudo, o processo de determinação das mesmas matrizes para um manipulador com n elos começa a ser tornar cada vez mais complexo a medida que n cresce. A convenção de Denavit-Hartenberg consiste numa abordagem sistemática para a obtenção das matrizes A_i de modo a representar a relação entre frames consecutivos da forma mais concisa e possível além de propiciar uma padronização de como pesquisadores descrevem a estrutura cinemática de um manipulador robótico.

De maneira geral, para especificar a matriz de transformação homogênea seriam necessários 6 parâmetros: três deslocamentos para a componente de translação e três ângulos para a rotação. Na convenção de Denavit-Hartenberg, a matriz de transformação A_i associada ao i -ésimo elo é descrita através de apenas 4 parâmetros. Isso é obtido através da introdução de duas restrições na colocação dos frames em cada elo:

- (DH1) x_i intersecta o eixo z_{i-1}
- (DH2) x_i é perpendicular ao eixo z_{i-1}

O cálculo da matriz A_i fica condicionado à determinação dos parâmetros: θ_i (joint angle), d_i (link offset), a_i (link length) e α_i (link twist). Como A_i é função da única variável da junta, três parâmetros são sempre fixos, dependendo apenas da geometria existente entre os frames, enquanto o quarto parâmetro é livre: θ_i , no caso de uma junta

de revolução e d_i , no caso de uma junta prismática. Em posse dos parâmetros obtidos para cada par de elos consecutivos, o cálculo da matriz A_i é obtido através da relação:

$$A_i = Rot_z(\theta_i) \cdot Trans_z(d_i) \cdot Trans_x(a_i) \cdot Rot_x(\alpha_i) \quad (2.17)$$

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

$$\times \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

onde $c.$ e $s.$ denotam $\cos(\cdot)$ e $\sin(\cdot)$, respectivamente. A tabela 2.1 descreve de maneira detalhada a definição de cada parâmetro de Denavit-Hartenberg.

Parâmetro	Definição
θ_i	O ângulo entre os eixos \mathbf{x}_{i-1} e \mathbf{x}_i em torno do eixo \mathbf{z}_{i-1}
d_i	A distância da origem do sistema de coordenadas $\{i-1\}$ até o eixo \mathbf{x}_i ao longo do eixo \mathbf{z}_{i-1}
a_i	A distância entre os eixos \mathbf{z}_{i-1} e \mathbf{z}_i ao longo do eixo \mathbf{x}_i ; para eixos que se intersectam, é paralela a $\mathbf{z}_{i-1} \times \mathbf{z}_i$
α_i	O ângulo entre o eixo \mathbf{z}_{i-1} e o eixo \mathbf{z}_i em torno do eixo \mathbf{x}_i

Tabela 2.1: Descrição dos parâmetros Denavit-Hartenberg

2.2.3 Cinemática direta de um braço planar

Um braço planar é um tipo de manipulador serial cujo espaço de trabalho se limita a um plano. A figura 2.3 mostra um braço planar do tipo 3R, o qual possui três elos e três juntas de revolução acoplados em série. A escolha da colocação do frame da base $o_0x_0y_0z_0$ é totalmente arbitrária e ao tomar como indicado na figura (com o eixo z apontando para fora do papel) a fixação dos frames subsequentes na cadeia cinemática fica restrita a convenção de Denavit-Hartenberg adotada. A tabela DH para esse manipulador é dada

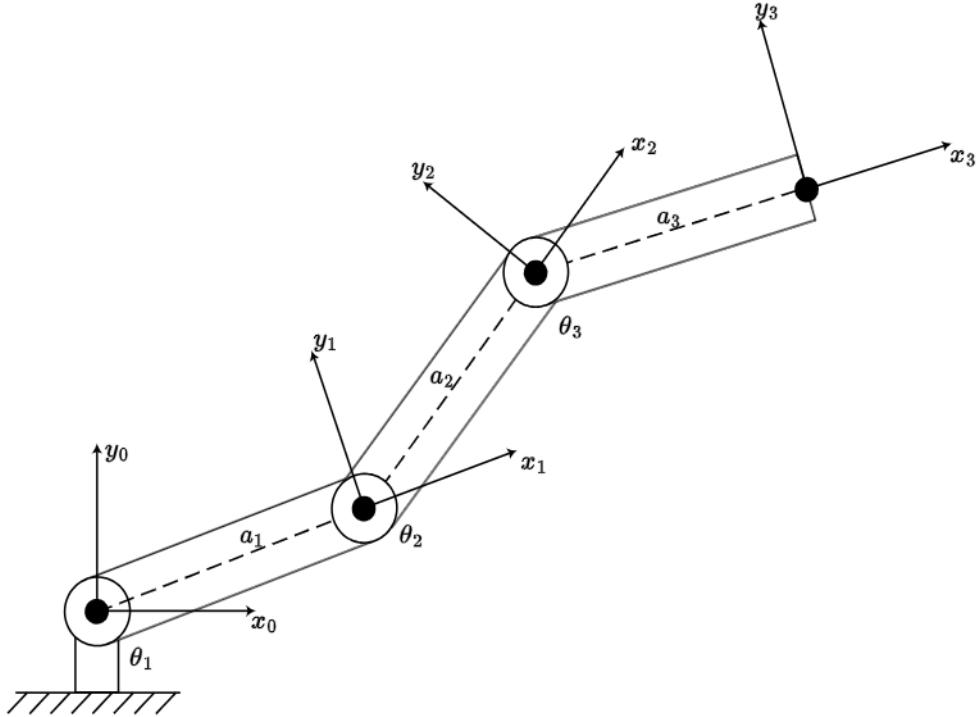


Figura 2.3: Braço planar do tipo 3R.

por:

Elo	θ	d	a	α
1	θ_1	0	a_1	0
2	θ_2	0	a_2	0
3	θ_3	0	a_3	0

Tabela 2.2: Parâmetros DH para o braço planar 3R da figura 2.3.

Dados os valores fixos a_i que indicam o comprimento do elo i , as únicas variáveis livres no cálculo da cinemática direta são os ângulos das juntas ($q_i = \theta_i$), desse modo, vamos denotar $\theta_1 + \theta_2 = \theta_{12}$, $\cos(\theta_1 + \theta_2) = c_{12}$ e assim por diante. Para $i = 1, 2, 3$ as matrizes A_i são calculadas com o auxílio da equação 2.18:

$$\mathbf{A}_i = \begin{bmatrix} c_i & -s_i & 0 & a_i c_i \\ s_i & c_i & 0 & a_i s_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.19)$$

Já para as matrizes T_i^0 , utilizamos a equação 2.16:

$$\mathbf{T}_1^0 = \mathbf{A}_1 \quad (2.20)$$

$$\mathbf{T}_2^0 = \mathbf{A}_1 \cdot \mathbf{A}_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1 c_1 + a_2 c_{12} \\ s_{12} & c_{12} & 0 & a_1 s_1 + a_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.21)$$

$$\mathbf{T}_3^0 = \mathbf{A}_1 \cdot \mathbf{A}_2 \cdot \mathbf{A}_3 = \begin{bmatrix} c_{123} & -s_{123} & 0 & a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ s_{123} & c_{123} & 0 & a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.22)$$

As três primeiras entradas da última coluna da matriz \mathbf{T}_3^0 dão a posição $\mathbf{P} = [x \ y \ z]^\top$ do efetuador final em função da configuração do manipulador. Note que $z = 0$ quaisquer que sejam os ângulos das juntas pois, como esperado, o manipulador é planar. Além disso, analisando a componente de rotação, fica evidente que a orientação do efetuador final com relação ao frame da base é dada pela soma dos ângulos das juntas: $\psi = \theta_{123}$.

2.3 Cinemática Diferencial

Na seção anterior vimos como podemos estabelecer uma relação entre a configuração de um manipulador com n juntas com a pose do efetuador final no espaço SE3. Nesta seção, iremos investigar de que forma se dá a relação de um vetor de velocidades no espaço das juntas com a velocidade do efetuador final no espaço de trabalho. Iremos ver que a matriz Jacobiana, atuando como uma generalização da derivada para o caso multidimensional, é responsável por estabelecer um mapeamento linear entre as velocidades e tem papel crucial na caracterização da qualidade do movimento de um manipulador através da análise das singularidades cinemáticas. Por fim, iremos analisar o problema da cinemática inversa diferencial, uma aplicação direta do mapeamento estabelecido pela matriz Jacobiana, proporcionando a geração eficiente de trajetórias cartesianas (retilíneas) no espaço de trabalho do manipulador.

2.3.1 A Jacobiana do Manipulador

Dado um manipulador com n juntas vamos considerar

$$\mathbf{T}_n^0(q) = \begin{bmatrix} R_n^0(q) & o_n^0(q) \\ 0 & 1 \end{bmatrix} \quad (2.23)$$

a transformação homogênea que expressa a pose do efetuador final com relação ao *frame* da base que, como já vimos, é função apenas da configuração $q = [q_1 \ \dots \ q_n]^\top$.

Buscamos estabelecer relações da seguinte forma:

$$v_n^0 = J_v \dot{q} \quad (2.24)$$

$$\omega_n^0 = J_\omega \dot{q} \quad (2.25)$$

onde v_n^0 e ω_n^0 expressam respectivamente as velocidades linear e angular do efetuador final e J_v , J_ω são matrizes de dimensão $3 \times n$. De maneira mais compacta, podemos escrever

$$\xi = J \dot{q} \quad (2.26)$$

onde teremos:

$$\xi = \begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix} \text{ e } J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \quad (2.27)$$

O vetor ξ de dimensão 6×1 é denominado de velocidade do corpo rígido (em inglês *body velocity* ou *twist*). Note também que a matriz J , chamada *jacobiana do manipulador* ou simplesmente jacobiana, é usualmente uma matriz de dimensão $6 \times n$.

O cálculo da jacobiana, pode ser feito de maneira simples e sistemática ao analisarmos as componentes angular e linear separadamente. No primeiro caso, sabemos que a velocidade angular do efetuador final pode ser obtida através da soma sucessiva das velocidades angulares de cada elo:

$$\omega_n^0 = \omega_1^0 + \omega_2^0 + \cdots + \omega_n^0 \quad (2.28)$$

Se a junta i é prismática não há rotação em torno do eixo z_{i-1} de modo que $\omega_i^{i-1} = 0$. Caso contrário, se a junta i é de revolução a rotação dá se em torno do eixo z_{i-1} com magnitude \dot{q}_i de modo que:

$$\omega_i^0 = \dot{q}_i z_{i-1}^0 \quad (2.29)$$

onde obviamente teremos $z_0^0 = k = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$

A orientação do eixo z_i com relação ao *frame* da base é dada por $z_i^0 = R_i^0 k$, então substituindo na equação (2.28) podemos escrever:

$$\omega_n^0 = \rho_1 \dot{q}_1 k + \rho_2 \dot{q}_2 R_1^0 k + \cdots + \rho_n \dot{q}_n R_{n-1}^0 k \quad (2.30)$$

onde $\rho_i = 0$ se a junta i é prismática e $\rho_i = 1$ caso contrário. Fica claro então que a metade inferior da jacobiana é dada por:

$$J_\omega = \begin{bmatrix} \rho_1 k & \rho_2 R_1^0 k & \cdots & \rho_n R_{n-1}^0 k \end{bmatrix} \quad (2.31)$$

A metade superior da jacobiana é obtida calculando-se o vetor \dot{o}_n^0 . Aplicando a regra da cadeia, temos:

$$\dot{o}_n^0 = \frac{\partial o_n^0}{\partial q_1} \dot{q}_1 + \frac{\partial o_n^0}{\partial q_2} \dot{q}_2 + \cdots + \frac{\partial o_n^0}{\partial q_n} \dot{q}_n \quad (2.32)$$

onde fica claro que a i-ésima coluna de J_v é dada por:

$$J_{v_i} = \frac{\partial o_n^0}{\partial q_i} \quad (2.33)$$

Para obter a expressão de J_{v_i} vamos analisar novamente o caso de juntas prismáticas e de revolução separadamente. No caso de uma única junta prismática, então o efetuador final apresenta apenas translação ao longo do eixo z_{i-1} de modo que:

$$\dot{o}_n^0 = \dot{q}_i R_{i-1}^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \dot{q}_i z_{i-1}^0 \quad (2.34)$$

Comparando com a (2.28) vemos que:

$$J_{v_i} = z_{i-1}^0 \quad (2.35)$$

Já para o caso de uma junta de revolução, a velocidade linear do efetuador final devido ao movimento do elo i é da forma $\omega \times r$ dada pela sua componente tangencial ao círculo de centro no ponto o_{i-1} e extremidade no ponto o_n onde:

$$\begin{aligned} \omega &= \dot{q}_i z_{i-1}^0 \\ r &= o_n^0 - o_{i-1}^0 \end{aligned}$$

onde finalmente chegamos à expressão:

$$J_{v_i} = z_{i-1}^0 \times (o_n^0 - o_{i-1}^0) \quad (2.36)$$

Resumindo as equações obtidas acima, podemos calcular a jacobiana de qualquer manipulador serial utilizando o seguinte procedimento:

A parte superior da matriz jacobiana J_v será:

$$J_v = \begin{bmatrix} J_{v_1} & J_{v_2} & \cdots & J_{v_n} \end{bmatrix} \quad (2.37)$$

onde a i-ésima coluna J_{v_i} é dada por:

$$J_{v_i} = \begin{cases} z_{i-1}^0 \times (o_n^0 - o_{i-1}^0) & \text{se a junta } i \text{ é de revolução} \\ z_{i-1} & \text{se a junta } i \text{ é prismática} \end{cases} \quad (2.38)$$

Já a parte inferior da matriz jacobiana J_ω será:

$$J_\omega = \begin{bmatrix} J_{\omega_1} & J_{\omega_2} & \cdots & J_{\omega_n} \end{bmatrix} \quad (2.39)$$

onde a i -ésima coluna J_{ω_i} é dada por:

$$J_{\omega_i} = \begin{cases} z_{i-1}^0 & \text{se a junta } i \text{ é de revolução} \\ 0 & \text{se a junta } i \text{ é prismática} \end{cases} \quad (2.40)$$

Note que o cálculo da jacobiana torna-se possível apenas com o conhecimento da função de cinemática direta T_n^0 mostrando-se uma maneira simples e sistemática para calcular não só a velocidade do efetuador final mas também a velocidade de qualquer ponto da estrutura cinemática do manipulador.

Como exemplo, considere o manipulador planar 3R introduzido na seção anterior. Com base no procedimento descrito acima, a jacobiana do manipulador é dada por:

$$J(q) = \begin{bmatrix} z_0^0 \times (o_3^0 - o_0^0) & z_1^0 \times (o_3^0 - o_1^0) & z_2^0 \times (o_3^0 - o_2^0) \\ z_0^0 & z_1^0 & z_2^0 \end{bmatrix} \quad (2.41)$$

onde os termos envolvidos são:

$$\begin{aligned} o_0^0 &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & o_1^0 &= \begin{bmatrix} a_1 c_1 \\ a_1 s_1 \\ 0 \end{bmatrix} & o_2^0 &= \begin{bmatrix} a_1 c_1 + a_2 c_{12} \\ a_1 s_1 + a_2 s_{12} \\ 0 \end{bmatrix} & o_3^0 &= \begin{bmatrix} a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ 0 \end{bmatrix} \\ z_0^0 &= z_1^0 = z_2^0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned}$$

Desenvolvendo as expressões acima, obtemos:

$$J(q) = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - a_3 s_{123} & -a_2 s_{12} - a_3 s_{123} & -a_3 s_{123} \\ a_1 c_1 + a_2 c_{12} + a_3 c_{123} & a_2 c_{12} + a_3 c_{123} & a_3 c_{123} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (2.42)$$

Note que o manipulador planar não provoca qualquer translação ao longo do eixo z

uma vez que qualquer contribuição de J_{v_i} é nula na terceira componente. Além disso, a única componente de rotação influenciada pelo movimento das juntas é a rotação em torno também do eixo do eixo z evidenciado pela terceira componente de J_{ω_i} igual a 1.

2.3.2 Singularidades

A matriz jacobiana, de dimensão $6 \times n$, estabelece o mapeamento linear entre as velocidades das juntas e do efetuador final através da relação:

$$\xi = J(q)\dot{q} \quad (2.43)$$

que coloca de forma explícita a dependência da configuração atual do manipulador no cálculo de J . Tal mapeamento implica que qualquer vetor de velocidades do efetuador final é uma combinação linear das colunas da matriz jacobiana:

$$\xi = J_1\dot{q}_1 + J_2\dot{q}_2 + \cdots + J_n\dot{q}_n \quad (2.44)$$

Uma vez que $\xi \in \mathbb{R}^6$ o manipulador só conseguirá desempenhar uma velocidade arbitrária se todas as colunas de J forem linearmente independentes, ou seja, quando o posto da matriz jacobiana for igual a 6. Para uma matriz $J \in \mathbb{R}^{6 \times n}$ é sempre verdade que $\text{posto}(J) \leq \min(6, n)$. Com efeito, no caso do manipulador planar 3R tínhamos $n = 3$ e desse modo $\text{posto}(J) \leq 3$ evidenciando o fato de que o manipulador não consegue desenvolver qualquer velocidade no espaço de trabalho.

Configurações para as quais o posto da matriz jacobiana é menor que o máximo possível (em inglês *rank deficiency*) são denominadas de *singularidades* ou configurações singulares. Identificar configurações singulares é de grande importância para o controle de manipuladores por diversos motivos, entre eles:

- Singularidades representam configurações nas quais a mobilidade do manipulador é reduzida, ou seja, não é possível impor um movimento arbitrário ao efetuador final.
- Quando o manipulador está em uma singularidade, pode haver infinitas soluções para o problema de cinemática inversa.
- Nas proximidades de uma singularidade, pequenas variações nas velocidades no espaço operacional podem causar velocidades ilimitadas no espaço das juntas.

Quando a matriz jacobiana é quadrada, podemos usar o fato de que seu determinante se anula em configurações singulares, contudo ainda assim o problema de determinar o conjunto de configurações é difícil, pois precisamos resolver a equação

$$\det(J(q)) = 0 \quad (2.45)$$

que geralmente envolve termos com alto grau de não linearidade. Nas próximas seções, examinaremos técnicas que viabilizam um esquema de controle capaz de se afastar de configurações singulares ao explorar a redundância presente em manipuladores planares. Isso se refere a casos em que a matriz Jacobiana é retangular, apresentando mais velocidades no espaço das juntas (colunas) do que velocidades possíveis no espaço de trabalho (linhas).

2.3.3 Cinemática Inversa Diferencial

Se a matriz jacobiana definida na equação (2.43) é quadrada e não singular, podemos resolver o problema de cinemática inversa através da simples inversão da mesma:

$$\dot{q} = J^{-1}(q)\xi \quad (2.46)$$

Se a configuração inicial do manipulador $q(0)$ é conhecida as posições das juntas podem ser calculadas integrando as velocidades no tempo:

$$q(t) = q(0) + \int_0^t \dot{q}(\tau) d\tau \quad (2.47)$$

A integração em tempo discreto pode ser feita utilizando técnicas de métodos numéricos. A abordagem mais simples consiste na integração pelo método de Euler, onde as posições das juntas no instante atual t_k são utilizadas para calcular a configuração do manipulador no instante posterior $t_{k+1} = t_k + \Delta t$:

$$q(t_{k+1}) = q(t_k) + \dot{q}(t_k)\delta_t \quad (2.48)$$

onde δ_t é um intervalo de integração apropriado (tempo de amostragem).

O esquema de controle descrito acima é conhecido como *resolved rate control*, o qual consegue de maneira simples e elegante o solucionar o problema de gerar movimentos no efetuador final de velocidade constante sem recorrer à soluções numéricas ou analíticas para o cálculo da cinemática inversa. Tal esquema é útil na geração de trajetórias retilíneas no espaço de trabalho, conhecidas como trajetórias cartesianas, uma vez que a componente translacional da velocidade do efetuador final tem direção constante ao longo de todo o trajeto pode ser tratada de maneira independente da componente rotacional.

Como exemplo, ainda considerando o manipulador planar 3R, poderíamos apenas especificar um vetor de velocidades ξ que leva em conta as componentes do plano xy da velocidade linear e a componente de rotação angular em torno do eixo z de modo que:

$$\xi = \begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} \quad (2.49)$$

Assim a matriz jacobiana se torna livre das linhas que possuem apenas zeros:

$$J(q) = \begin{bmatrix} -a_1s_1 - a_2s_{12} - a_3s_{123} & -a_2s_{12} - a_3s_{123} & -a_3s_{123} \\ a_1c_1 + a_2c_{12} + a_3c_{123} & a_2c_{12} + a_3c_{123} & a_3c_{123} \\ 1 & 1 & 1 \end{bmatrix} \quad (2.50)$$

e contanto que não seja singular pode ser facilmente invertida. Desse modo, se quisermos por exemplo, gerar um movimento retilíneo no efetuador final paralelo ao eixo x do plano xy com velocidade constante, basta tomar $\xi = [v_x \ 0 \ 0]^\top$ com $0 < v_x \ll 1$.

2.4 Manipuladores redundantes

Manipuladores cinematicamente redundantes, são aqueles que possuem mais juntas do que o número estritamente necessário para a execução de uma determinada tarefa. Este excedente de juntas confere a esses manipuladores um nível aumentado de destreza, permitindo-lhes navegar em ambientes complexos com maior flexibilidade. Nesta seção vamos introduzir uma solução geral para o problema da cinemática inversa diferencial quando a matriz jacobiana é retangular, envolvendo o conceito da sua *pseudo-inversa*. Em seguida, utilizando a decomposição em valores singulares de J iremos fornecer uma descrição geométrica e qualitativa da destreza associada à uma dada configuração através dos conceitos do elipsoide e da medida de manipulabilidade. Por fim, vamos ver como podemos utilizar a solução geral fornecida pela pseudo-inversa para otimizar diferentes índices de performance com o objetivo de evitar singularidades e limites mecânicos das juntas.

2.4.1 Pseudo-Inversa da Jacobiana

Num manipulador cinematicamente redundante, a matriz jacobiana de dimensão $m \times n$ será retangular ($m < n$). Isso significa que J possui mais colunas do que linhas e nesse caso existem infinitas soluções para o problema de cinemática inversa diferencial. Uma solução possível nesse caso é tratar o problema como um de otimização, onde buscamos uma solução que minimize por exemplo $\|\dot{q}\|^2$. Pode ser mostrado que tal solução é dada por:

$$\dot{q} = J^\dagger \xi + (I_n - J^\dagger J)\dot{q}_0 \quad (2.51)$$

onde \dot{q}_0 é um vetor de velocidades arbitrário e a matriz J^\dagger é conhecida como *matriz inversa de Moore-Penrose* ou apenas *pseudo-inversa* de J e é dada por:

$$J^\dagger = J^\top (JJ^\top)^{-1} \quad (2.52)$$

Vale notar que o termo $I_n - J^\dagger J$ atua projetando o vetor \dot{q}_0 no espaço nulo de J . Com efeito, aplicando a jacobiana à esquerda na equação (2.51) temos:

$$\begin{aligned} J\dot{q} &= JJ^\dagger\xi + J(I_n - J^\dagger J)\dot{q}_0 \\ &= JJ^\top(JJ^\top)^{-1}\xi + (J - JJ^\top(JJ^\top)^{-1}J)\dot{q}_0 \\ &= \xi + (J - J)\dot{q}_0 \\ J\dot{q} &= \xi \end{aligned}$$

permitindo que o manipulador realize movimentos internos no espaço das juntas que não afetam a velocidade ξ do efetuador final.

2.4.2 Medida de Manipulabilidade

Uma maneira de investigar mais a fundo o mapeamento linear estabelecido pela jacobiana é entender como a mesma “deforma” o vetor \dot{q} de entradas para produzir o vetor ξ de saídas. Para isso, podemos considerar o disco formado pelo conjunto de velocidades com norma unitária:

$$\|\dot{q}\|^2 = q_1^2 + q_2^2 + \dots + q_n^2 \leq 1 \quad (2.53)$$

Substituindo a solução de menor norma $\dot{q} = J^\dagger\xi$:

$$\begin{aligned} \|\dot{q}\|^2 &= \dot{q}^\top \dot{q} \\ &= (J^\dagger\xi)^\top J^\dagger\xi \\ &= \xi^\top (J^\top (JJ^\top)^{-1})^\top J^\top (JJ^\top)^{-1}\xi \\ &= \xi^\top (JJ^\top)^{-1} (JJ^\top) (JJ^\top)^{-1}\xi \\ \|\dot{q}\|^2 &= \xi^\top (JJ^\top)^{-1}\xi \leq 1 \end{aligned} \quad (2.54)$$

A equação (2.54) define uma região no espaço de trabalho conhecido como *elipsoide de manipulabilidade* que representa todas as velocidades possíveis do efetuador final para uma dada configuração do manipulador. Esse fato pode ser facilmente verificado ao considerarmos a decomposição em valores singulares (SVD) da jacobiana $J = U\Sigma V^\top$:

$$\begin{aligned}
\|\dot{q}\|^2 &= \xi^\top (U\Sigma V^\top V\Sigma^\top U^\top)^{-1} \xi \\
&= \xi^\top (U\Sigma^2 U^\top)^{-1} \xi \\
&= \xi^\top U\Sigma^{-2} U^\top \xi \\
\|\dot{q}\|^2 &= (U^\top \xi)^\top \Sigma^{-2} (U^\top \xi)
\end{aligned} \tag{2.55}$$

onde sabemos que U e V são matrizes ortogonais, isto é $U^{-1} = U^\top$ e $V^{-1} = V^\top$. Além disso a matriz

$$\Sigma^{-2} = \begin{bmatrix} \sigma_1^{-2} & & & \\ & \sigma_2^{-2} & & \\ & & \ddots & \\ & & & \sigma_m^{-2} \end{bmatrix} \tag{2.56}$$

é diagonal e os termos que $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m$ são os valores singulares de J . Por fim, ao fazermos a substituição $w = U^\top \xi$ podemos reescrever a equação (2.55) como:

$$w^\top \Sigma^{-2} w = \sum_{i=1}^m \frac{w_i^2}{\sigma_i^2} \leq 1 \tag{2.57}$$

evidenciado que o disco é mapeado num elipsoide com eixos alinhados a um sistema de coordenadas rotacionado por U^\top . No sistema de coordenadas original, os semieixos do elipsoide são dados pelos vetores $\sigma_i u_i$.

A medida de manipulabilidade μ é definida como o produto dos valores singulares de J :

$$\mu = \sigma_1 \sigma_2 \cdots \sigma_m \tag{2.58}$$

que é proporcional ao volume do elipsoide de manipulabilidade. Ao passo que nos aproximamos de uma singularidade, um ou mais dos valores singulares de J se aproximam de zero, reduzindo o volume do elipsoide e consequentemente a destreza do manipulador. Isso pode ser visualizado na figura 2.4 onde para o braço planar 3R, o elipsoide de manipulabilidade é mostrado para diferentes configurações do manipulador e vai se tornando cada vez mais achatado à medida que nos aproximamos do limite do espaço de trabalho.

2.4.3 Resolução de Redundância

Ao estabelecermos a solução geral dada pela equação (2.51), dissemos que o vetor \dot{q}_0 pode ser escolhido arbitrariamente. Uma possibilidade é tomá-lo de forma a maximizar algum índice de performance w , para isso escolhendo o vetor na direção do gradiente:

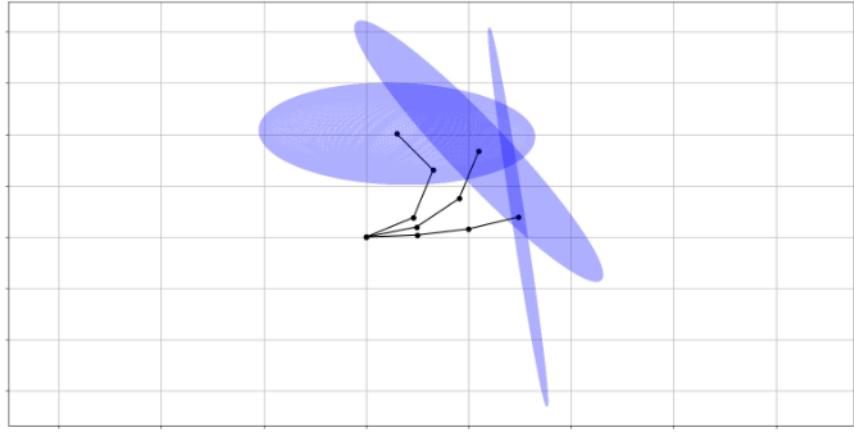


Figura 2.4: Elipsoide de manipulabilidade para diferentes configurações do manipulador planar 3R.

$$\dot{q}_0 = k_0 \left(\frac{\partial w(q)}{\partial q} \right)^\top \quad (2.59)$$

onde $k_0 > 0$ é uma constante positiva que determina o tamanho do passo.

Uma escolha natural para o índice de performance w é a *medida de manipulabilidade de Yoshikawa*:

$$w(q) = \sqrt{\det(J(q)J(q)^\top)} \quad (2.60)$$

onde vale ressaltar que é equivalente àquela definida na equação (2.58) uma vez que se λ_i são os autovalores de JJ^\top então $\sigma_i = \sqrt{\lambda_i}$.

Apesar do cálculo da manipulabilidade ser relativamente simples, o mesmo não se pode dizer para o cálculo do seu gradiente devido à complexidade das expressões envolvidas. Uma alternativa é utilizar uma métrica mais simples como a *distância para os limites mecânicos das juntas* dada por:

$$w(q) = -\frac{1}{2n} \sum_{i=1}^n \left(\frac{q_i - \bar{q}_i}{q_{iM} - q_{im}} \right)^2 \quad (2.61)$$

Ao maximizar tal índice, espera-se que o manipulador mantenha-se próximo ao ponto central de atuação de cada junta, evitando assim configurações singulares no limite do espaço de trabalho. Além disso, o vetor gradiente pode ser calculado de maneira analítica onde cada coordenada é dada por:

$$\frac{\partial w(q)}{\partial q_i} = -\frac{1}{n} \frac{q_i - \bar{q}_i}{(q_{iM} - q_{im})^2} \quad (2.62)$$

Outro ponto a salientar é que a escolha do tamanho do passo k_0 é crucial para a performance do algoritmo. Se k_0 for muito pequeno o processo de otimização pode se tornar muito lento, enquanto que se k_0 for muito grande isso pode levar até mesmo uma diminuição ou até mesmo não convergência do valor de w devido a oscilações em torno do ponto de máximo local.

No próximo capítulo, iremos aplicar os conceitos apresentados até agora na concepção de um ambiente simulado para um manipulador redundante com cinco graus de liberdade. O objetivo principal será a execução de trajetórias retilíneas no espaço de trabalho $\mathcal{T} \in \text{SE}(3)$ que levam apenas em consideração a posição do efetuador e com isso explorar a resolução de redundância para otimizar diferentes índices relacionados a configuração do manipulador durante a execução da trajetória.

Capítulo 3

Implementação

Neste capítulo, iremos explorar as ferramentas e métodos utilizados na concepção de um ambiente robótico simulado para um manipulador com cinco graus de liberdade do tipo 5R. O manipulador tem uma cadeia cinemática simples que pode ser entendida como a composição de dois outros braços planares, mas que permite que a posição do efetuador final não esteja limitada, por exemplo, a um plano de altura constante. Começaremos explorando a modelagem da cadeia cinemática e também a representação do modelo virtual do robô dentro do simulador *Webots*. Em seguida, iremos definir a arquitetura de comunicação proposta para se controlar o manipulador utilizando o conceito de *Actions* presente no framework *Robot Operating System* (ROS), o qual permitiu uma implementação modularizada para execução dos experimentos. Por fim, iremos detalhar o esquema de controle e bibliotecas utilizadas na implementação do algoritmo *Resolved Rate Control* para execução de trajetórias retilíneas bem como os experimentos realizados para se avaliar a resolução de redundância na execução de tais trajetórias.

3.1 Simulação de manipuladores robóticos

Simuladores de física tornam possível a pesquisa e desenvolvimento na robótica, pois permitem que os pesquisadores testem e validem métodos teóricos inicialmente ou exclusivamente em um simulador, uma vez que os robôs em si são frequentemente caros, frágeis e escassos [Collins et al., 2021]. Os simuladores oferecem um ambiente acessível e barato de prototipação com uma variedade de robôs disponíveis e prontos para uso, sem o risco de danificar o equipamento físico economizando assim tempo e recursos. A simulação pode ser executada mais rápido do que em tempo real (o que é especialmente importante para abordagens baseadas em aprendizado ou análises de natureza estatístico), é paralelizável e não requer intervenção física para reiniciar um ambiente.

Para se implementar um ambiente simulado diversas características do simulador devem ser levadas em conta, tais como: o modelo do robô a ser simulado, os sensores e atuadores disponíveis, o ambiente físico a ser simulado (por exemplo, correntes de ar,

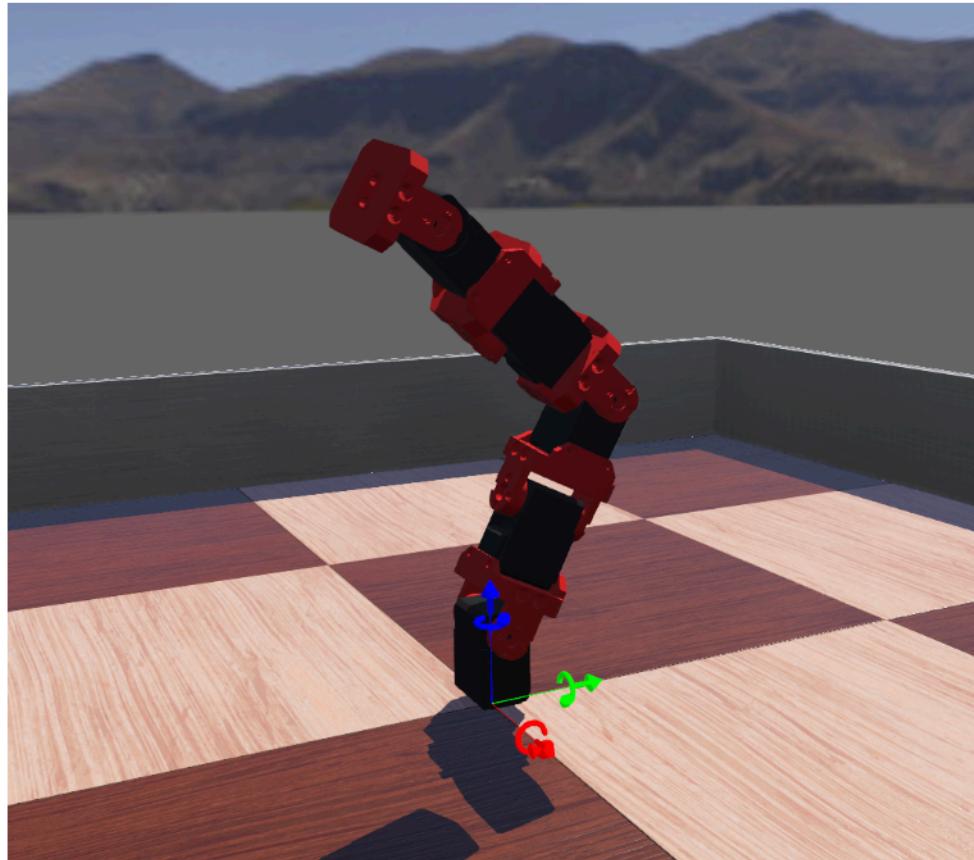


Figura 3.1: Modelo virtual do manipulador no Webots.

ambientes aquáticos, etc), linguagens de programação disponíveis para controle do robô, formatos suportados, extensibilidade, documentação etc.

Neste trabalho, optamos por utilizar a linguagem de programação *Python* devido à disponibilidade de pacotes para computação numérica e visualização (*Numpy*, *Pandas* e *Matplotlib*) bem como voltadas exclusivamente para a aplicações relacionadas à robótica como *Robotics Toolbox for Python* [Corke and Haviland, 2021]. O simulador escolhido foi o Webots [Michel, 2004] devido a possibilidade de prototipar o modelo virtual do manipulador do zero aliado a testes rápidos de diversos designs. Além disso, o Webots oferece suporte oficial ao ROS2, o que permitiu a implementação de um sistema modularizado, onde cada componente executa uma tarefa específica, facilitando a manutenção e a extensão do mesmo, inclusive para o interfaceamento com um robô real.

Modelagem da Cadeia Cinemática

A estrutura cinemática do manipulador 5R foi pensada de modo a ser uma simples cadeia de juntas rotacionais, permitindo uma fácil construção do robô real, como por exemplo, sendo composto por uma sequência de servo motores conectados por soquetes como ilustrado na figura 3.1. A cadeia cinemática é similar ao que já vimos no exemplo dos braço planar 3R, contudo juntas consecutivas possuem eixos de rotação ortogonais

entre si. Com isso, a posição do efetuador final não fica restrita a um plano perpendicular ao eixo de rotação das juntas, o que permite especificarmos no espaço de trabalho, vetores com três coordenadas para compor a trajetória a ser seguida. Como $n = 5$ e $m = 3$ o manipulador tem um grau de redundância de duas juntas excedentes. A tabela 3.1 resume os parâmetros DH utilizados para modelar a cadeia cinemática.

Elo	θ	d	a	α
1	θ_1	0	0.06	$\pi/2$
2	θ_2	0	0.06	$-\pi/2$
3	θ_3	0	0.06	$\pi/2$
4	θ_4	0	0.06	$-\pi/2$
5	θ_5	0	0.02	$\pi/2$

Tabela 3.1: Parâmetros DH para o manipulador 5R.

A fixação de frames imposta pela convenção DH nem sempre permite que o frame da base do manipulador coincida com o frame do mundo, introduzindo nesse caso a utilização de *offsets* nos parâmetros variáveis das juntas, ou transformações fixas entre os frames que não mudam conforme as juntas são atuadas. Optamos por adotar uma abordagem mais direta, onde introduzimos a transformação \mathbf{T}_0^w que relaciona o frame do mundo $\{w\}$ com o frame da base do manipulador $\{0\}$:

$$\mathbf{T}_0^w = \text{Trans}_z(0.04) \cdot \text{Rot}_z(\pi) \cdot \text{Rot}_y(\pi/2) \quad (3.1)$$

Assim, para se calcular a cinemática direta do manipulador, prosseguimos de maneira usual na cadeia cinemática adicionando a transformação \mathbf{T}_0^w ao início da multiplicação matricial:

$$\mathbf{T}_5^w = \mathbf{T}_0^w \cdot \mathbf{T}_1^0 \cdot \mathbf{T}_2^1 \cdot \mathbf{T}_3^2 \cdot \mathbf{T}_4^3 \cdot \mathbf{T}_5^4 \quad (3.2)$$

Por outro lado, durante a etapa da cinemática inversa diferencial, precisamos especificar vetores livres ξ^w que indicam a velocidade cartesiana do efetuador no mundo em termos do frame da base do manipulador. Para isso, utilizamos a matriz de rotação da transformação inversa (do frame $\{0\} \rightarrow \{w\}$), isto é:

$$\xi^0 = \text{Rot}(\mathbf{T}_0^w)^\top \xi^w \quad (3.3)$$

Modelo dinâmico do manipulador

Com o intuito de conferir um caráter mais realista para a simulação, um modelo dinâmico para o robô foi construído especificando as propriedades físicas e geometrias de colisão de cada elo no simulador. A figura 3.3 mostra as formas primitivas do tipo *Box*

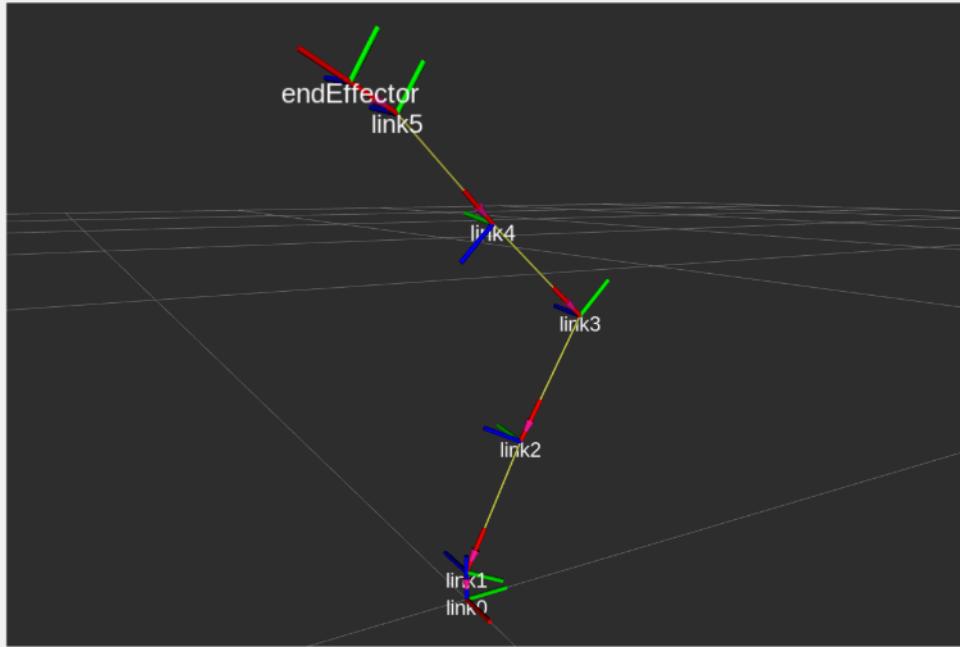


Figura 3.2: Cadeia cinemática visualizada no RViz.

(caixas) que foram usadas para compor a geometria de cada par servo-soquete que compõe um elo da cadeia. Para o simulador computar o modelo dinâmico ao longo do tempo, foram fornecidas as informações de massa do servo-motor (disponível na especificação do fabricante) e no caso dos soquetes, a densidade do material PLA foi utilizada para estimar sua massa com base no volume da geometria modelada (conjunto de caixas). Além disso, a definição das matrizes de inércia ficou por conta do próprio simulador, que estima seu valor com base na massa fornecida e na posição e orientação das primitivas utilizadas durante a modelagem. Vale ressaltar que a adição das propriedades dinâmicas tem caráter apenas de aproximação de um cenário mais real, tendo em vista que o esquema de controle proposto atua apenas na velocidade das juntas ao passo que a posição do motor é controlada de maneira automática pelo simulador através de um PID intrinseco à simulação de um dispositivo como o motor.

3.2 Arquitetura de comunicação

A comunicação do controlador com o ambiente simulado do robô foi feita através do uso do framework ROS, que consiste num conjunto de bibliotecas e pacotes de software que facilitam a troca de mensagens entre diferentes componentes de um sistema robótico. O próprio simulador Webots oferece suporte nativo ao ROS2, com uma documentação detalhada de como configurar um projeto e o uso básico de troca de mensagens entre diferentes processos.

A arquitetura do ROS2 é baseada em uma estrutura de grafo, onde nós representam

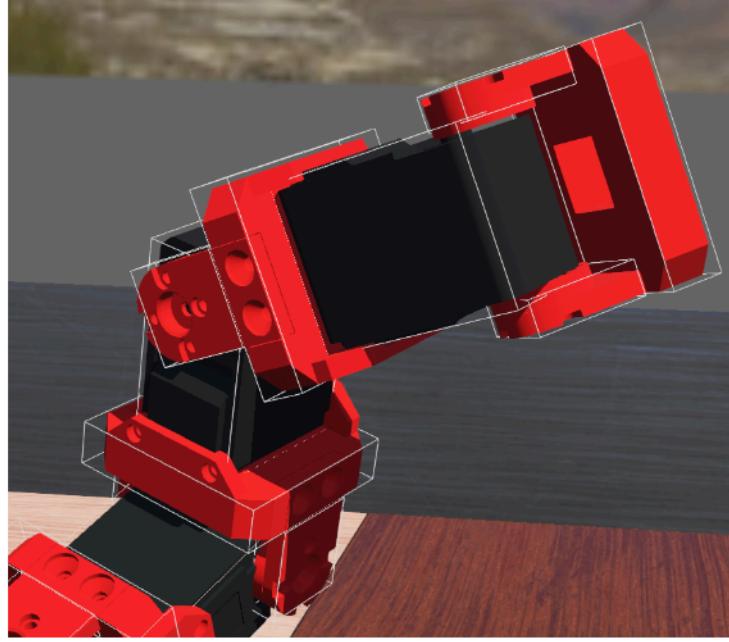


Figura 3.3: Geometria de colisão do modelo virtual do manipulador.

processos individuais que interagem com outros nós recebendo e enviando mensagens, através de tópicos. Idealmente, um nó deve ser responsável por uma única tarefa modular, como controlar os motores do robô ou enviar dados coletados por um sensor de distância.

Os tópicos são os canais pelos quais os nós trocam mensagens e seguem um modelo *publisher/subscriber*, onde nós publicam mensagens em um tópico e outros nós se inscrevem para receber essas mensagens, de maneira completamente anônima. As mensagens passadas nos tópicos podem variar amplamente e geralmente são definidas pelo usuário, cobrindo dados de sensores, comandos de controle de motores, informações de estado, comandos de atuadores entre outros.

ROS2 Actions

As ações do ROS2 expandem as capacidades do Sistema Operacional de Robôs, proporcionando uma maneira de executar tarefas longas e orientadas a objetivos. Ao contrário dos tópicos, que são adequados para passagem simples de mensagens, as ações são projetadas para interações mais complexas, como mover um robô para uma localização específica ou executar uma sequência de ações. As ações consistem em três componentes principais: o objetivo, o resultado e o feedback. O cliente envia um objetivo para o servidor, que então executa a ação solicitada. Durante a execução, o servidor pode fornecer feedback ao cliente para mantê-lo informado sobre o progresso. Uma vez concluída a ação, o servidor envia um resultado de volta ao cliente. Este padrão de comunicação assíncrona permite o manuseio robusto de tarefas que podem levar algum tempo para serem concluídas, proporcionando melhor controle e monitoramento do sistema como um todo. As ações do ROS2

são uma ferramenta poderosa para construir aplicativos robóticos avançados, permitindo que os robôs executem tarefas complexas com precisão e eficiência.

Grafo de comunicação no ROS2

A arquitetura de comunicação proposta para o controle do manipulador é ilustrada no grafo da figura 3.4, onde temos nós, tópicos e ações associados a execução de uma trajetória. A seguir temos uma breve descrição do funcionamento de cada nó:

- **snake_driver** - Nó instânciado pelo simulador responsável por controlar os motores do manipulador. Está inscrito no tópico `target_joint_states` que recebe a configuração das juntas desejada para o manipulador e publica no tópico `joint_states` os valores lidos pelos sensores de posição de cada motor. Este nó pode ser substituído por um nó que se comunique com um robô real, bastando que a interface de comunicação seja mantida, garantindo uma transferência natural do ambiente simulado para testes físicos.
- **snake_controller** - Nó responsável por implementar a lógica de controle do manipulador. Este nó está inscrito/publica nos dois tópicos `target_joint_states` e `joint_states` anteriores para interação com o *driver* do manipulador. Além disso, se inscreve num tópico `rrc_input`, recebendo parâmetros de controle do algoritmo 1 e publica no tópico `rrc_output` dados relativos à posição do efetuador final e métricas de desempenho.
- **trajectory_action_server** - Nó responsável por receber um objetivo de posição e informações do efetuador final para execução de uma trajetória no espaço de trabalho do manipulador, publicando no tópico `rrc_input` os parâmetros de controle.
- **trajectory_action_client** - Nó responsável por enviar um objetivo de posição para o servidor de ação `trajectory_action_server` e receber o resultado da execução da trajetória. Através da interface de ação, o cliente recebe feedback do progresso da execução da trajetória e o resultado final, salvando todos os dados em um arquivo de *log*, para posterior análise.

3.3 O Algoritmo *Resolved Rate Control*

3.4 Experimentos

cenário 1 -*i* q0 == qf (variação de métrica controla o tempo de execução)

cenário 2 -*i* q0 != qf (erro da posição e variação de métrica controla o tempo de execução)

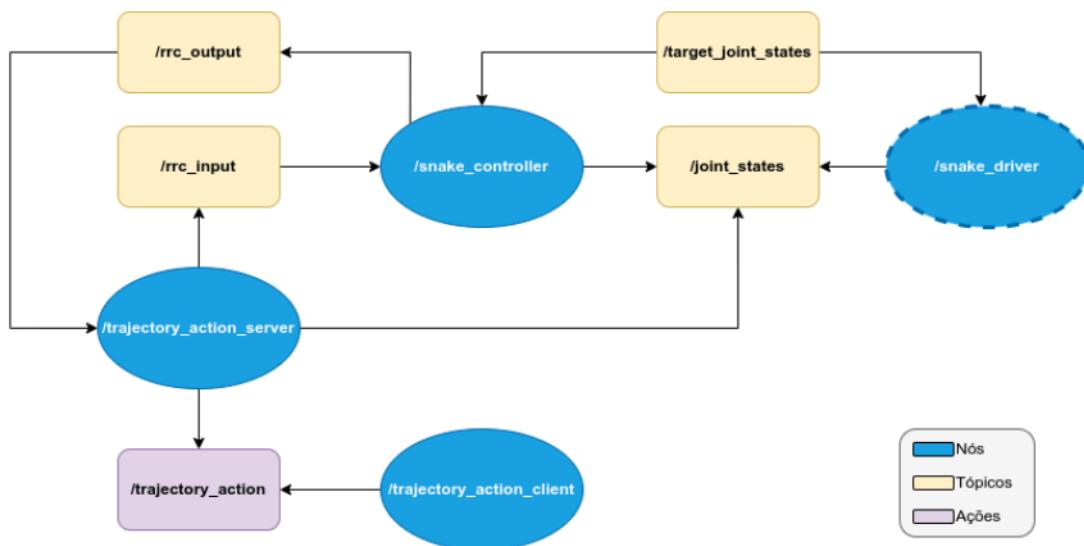


Figura 3.4: Arquitetura de comunicação proposta para controle do manipulador.

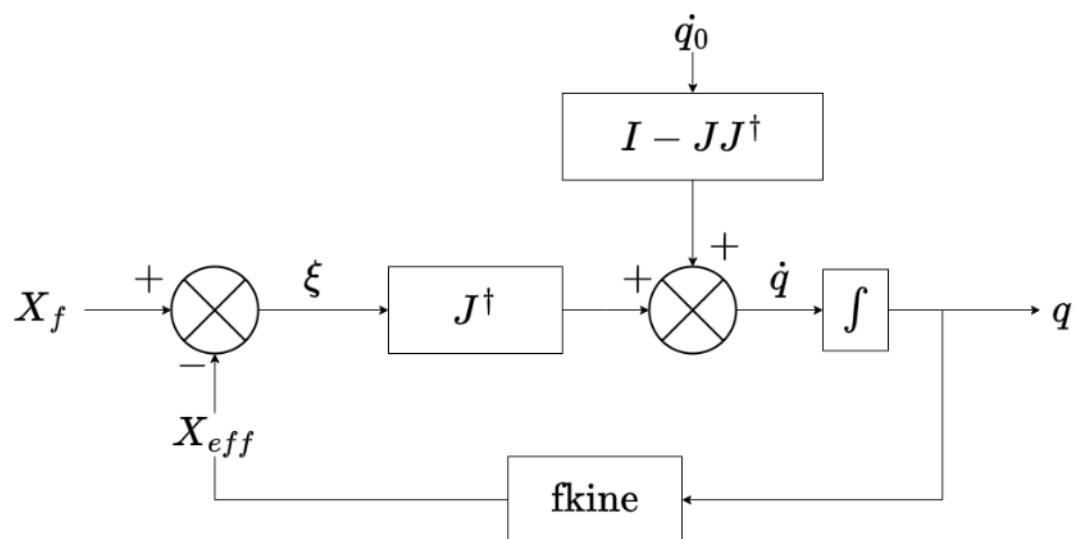


Figura 3.5: Diagrama de blocos do controlador RRC

Algoritmo 1 Resolved Rate Controller - Atualizando o estado das juntas

```

1: procedure UPDATEJOINTPOSITION( $q, \xi, k_0, \delta_t, \text{metricName}$ )
2:    $\xi \leftarrow \text{Rot}(\mathbf{T}_0^w)^\top \xi$ 
3:    $J \leftarrow \text{Jacobian}(q)$ 
4:    $J^\dagger \leftarrow J^\top (JJ^\top)^{-1}$ 
5:    $n \leftarrow \text{length}(q)$ 
6:    $\dot{q}_0 \leftarrow \text{array}(\text{size} : n)$ 
7:   for  $i \leftarrow 0$  to  $n - 1$  do                                 $\triangleright$  Calculando o gradiente da métrica
8:     if  $\text{metricName} = \text{joint\_distance}$  then
9:        $q_{mid} \leftarrow 0.5 \times (q_{\max}[i] + q_{\min}[i])$ 
10:       $\dot{q}_0[i] \leftarrow (-k_0/n) \times (q[i] - q_{mid}) \div (q_{\max}[i] - q_{\min}[i])^2$ 
11:    else if  $\text{metricName} = \text{manipulability}$  then
12:       $q_+, q_- \leftarrow \text{copy}(q), \text{copy}(q)$ 
13:       $q_+[i] \leftarrow q_+[i] + h$ 
14:       $q_-[i] \leftarrow q_-[i] - h$ 
15:       $\dot{q}_0[i] \leftarrow k_0 \times (\text{manipulability}(q_+) - \text{manipulability}(q_-)) \div (2 \times h)$ 
16:    end if
17:  end for
18:   $\dot{q} \leftarrow J^\dagger \xi + (I - J^\dagger J)\dot{q}_0$ 
19:  return  $\text{clipLimits}(q + \dot{q}\delta_t, q_{\max}, q_{\min})$            $\triangleright$  Restringe aos limites das juntas
20: end procedure

```

cenário 3 - $q_0 \neq qf$ com $q_0 \sim N(0, 0.1)$ (erro da posição e variação de métrica controla o tempo de execução) repetir 100x

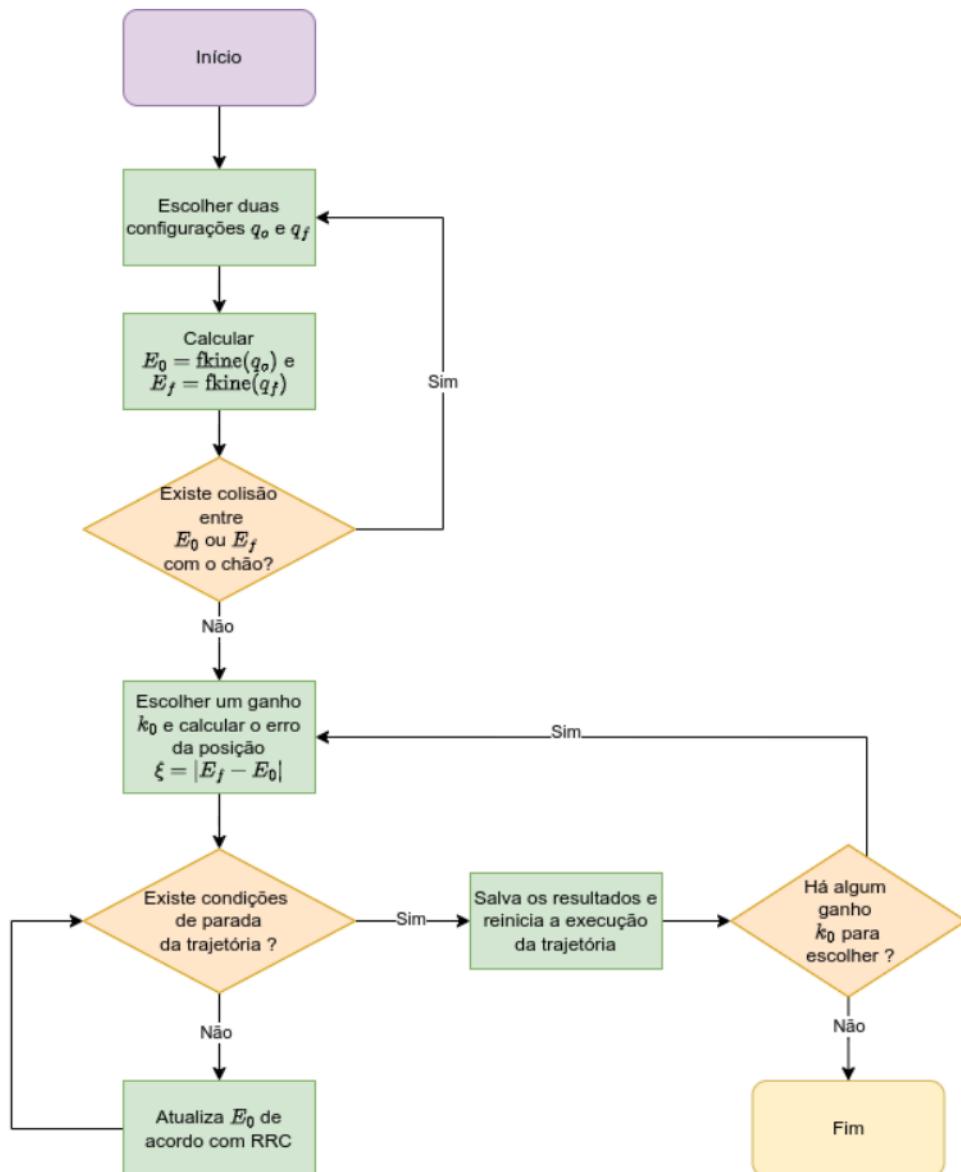


Figura 3.6: Etapas seguidas em cada execução de uma **trajectory_action**.

Capítulo 4

Resultados

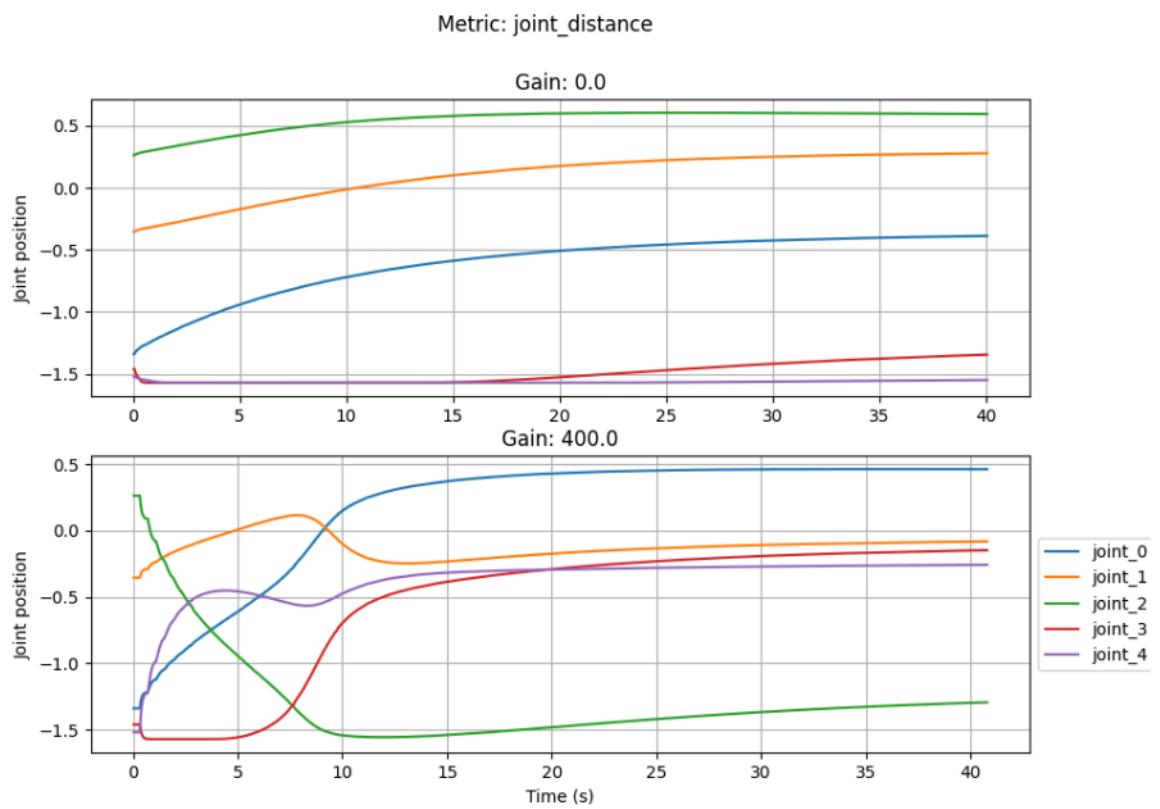


Figura 4.1: Resultados

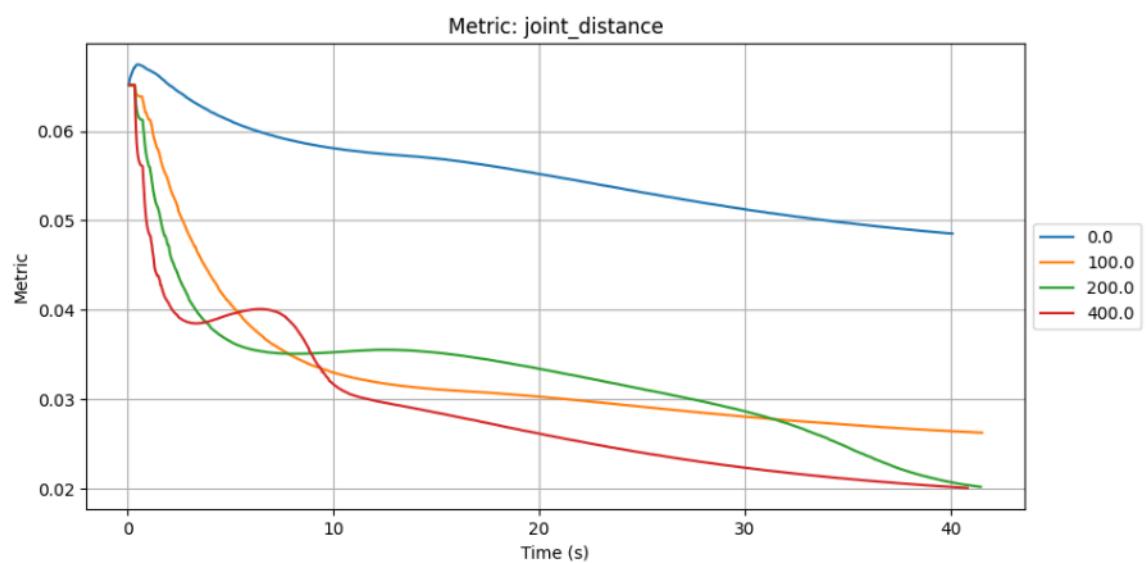


Figura 4.2: Resultados

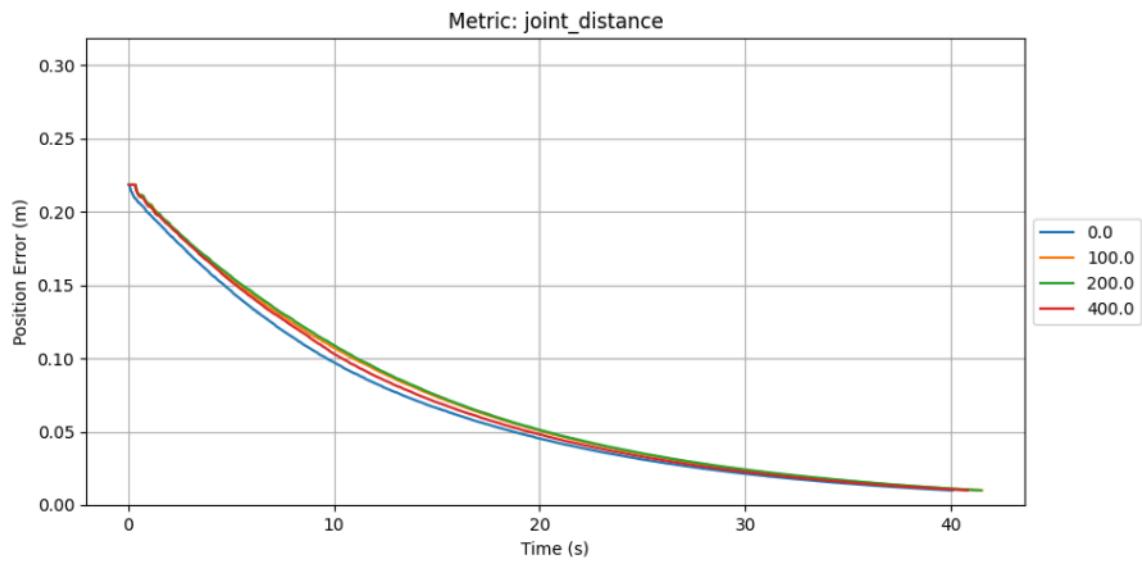


Figura 4.3: Resultados

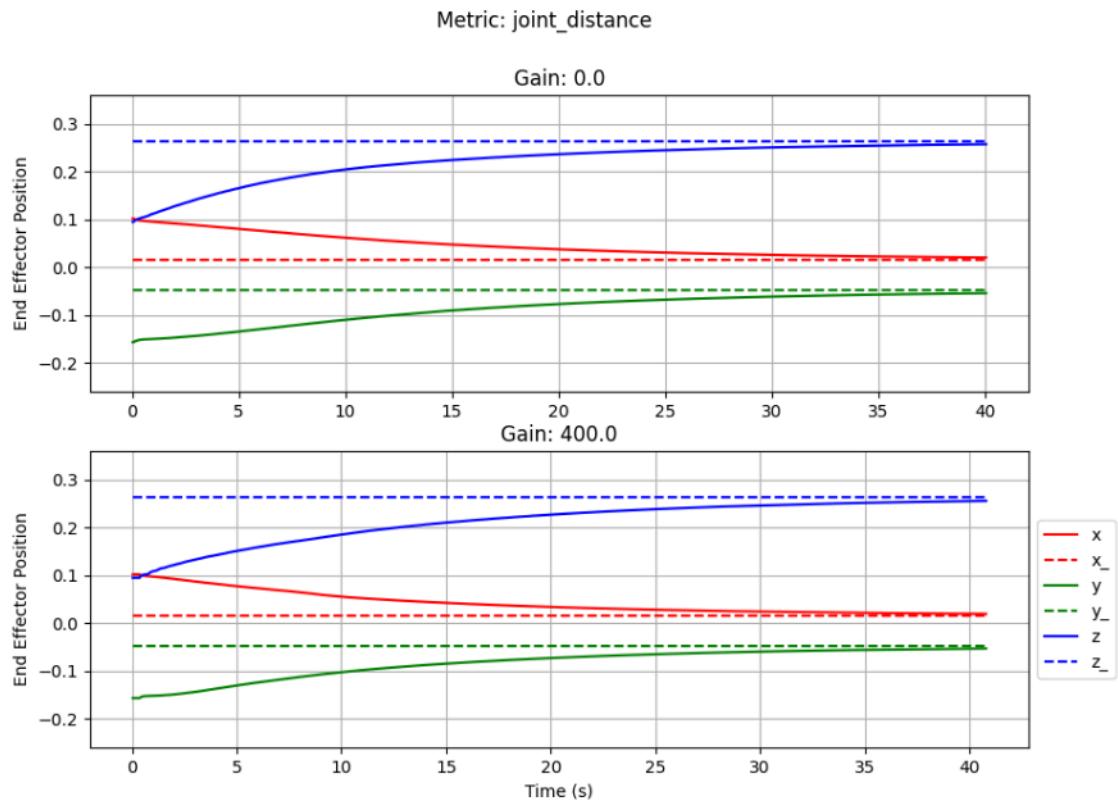


Figura 4.4: Resultados

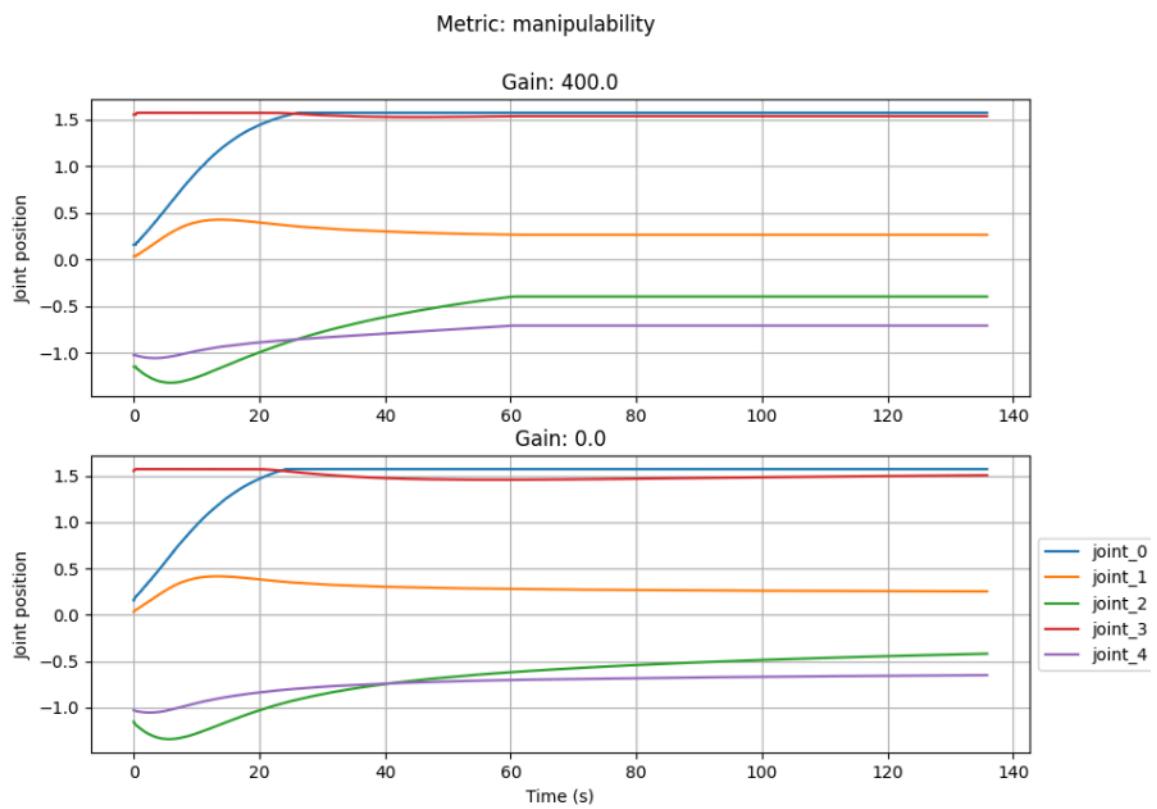


Figura 4.5: Resultados

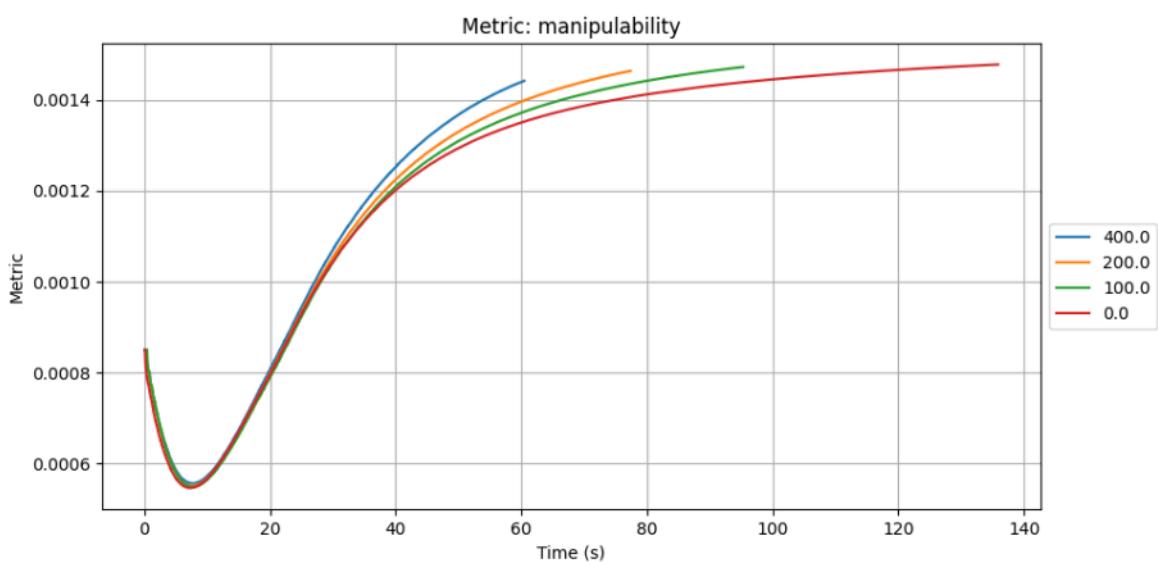


Figura 4.6: Resultados

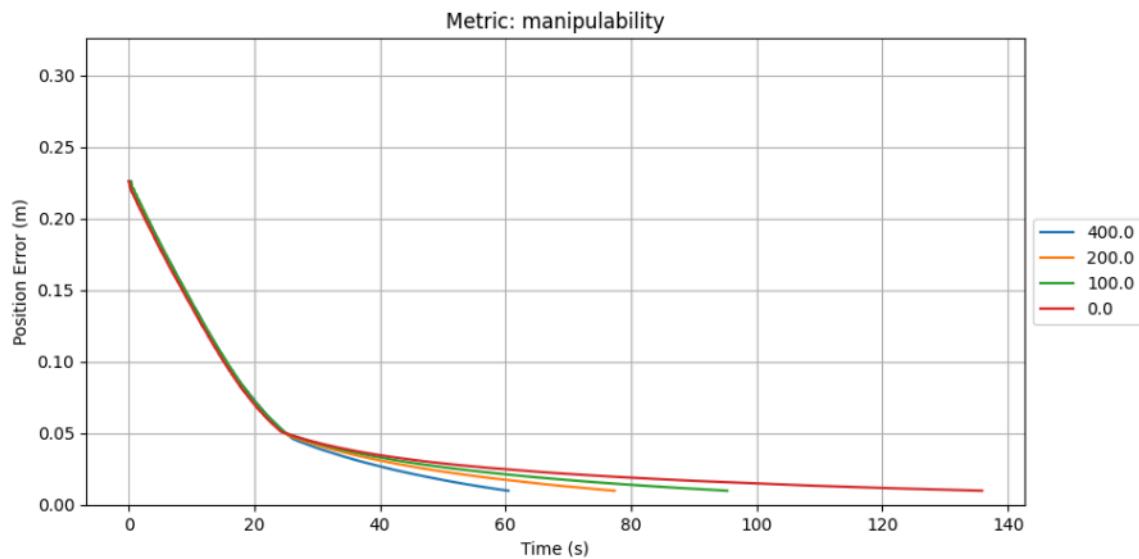


Figura 4.7: Resultados

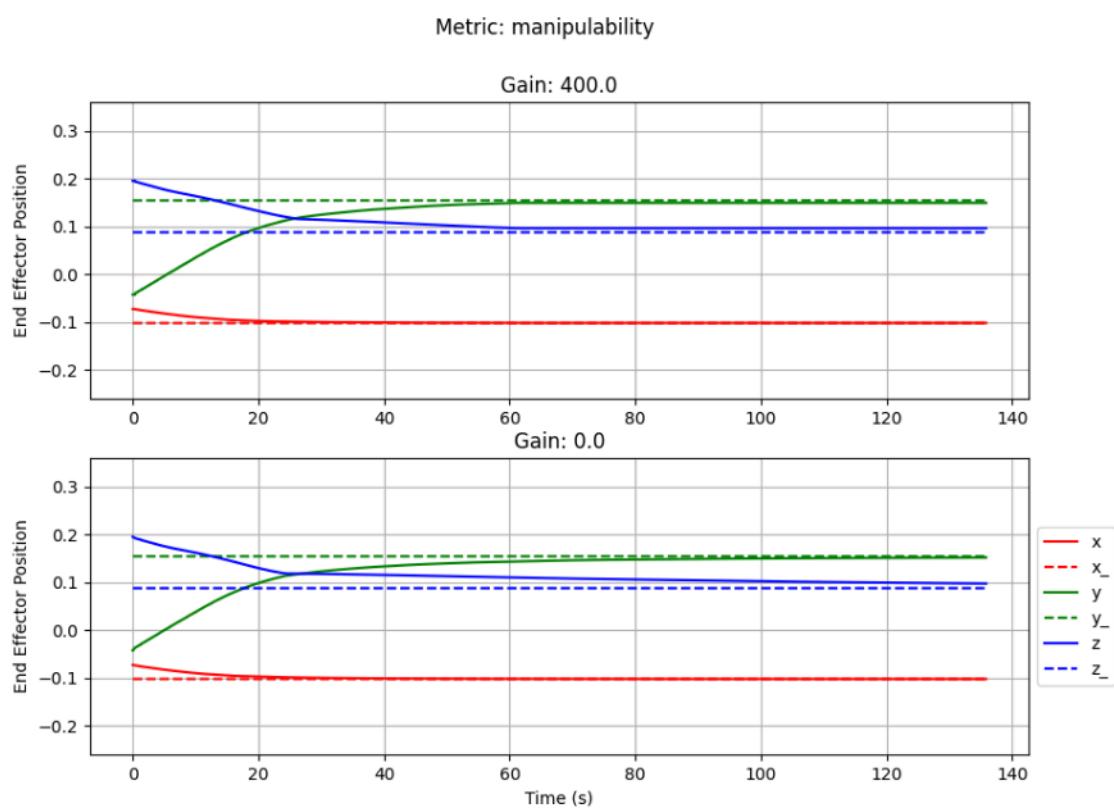


Figura 4.8: Resultados

Capítulo 5

Conclusão

Bibliografia

- [Collins et al., 2021] Collins, J., Chand, S., Vanderkop, A., and Howard, D. (2021). A review of physics simulators for robotic applications. *IEEE Access*, 9:51416–51431.
- [Corke and Haviland, 2021] Corke, P. and Haviland, J. (2021). Not your grandmother’s toolbox—the robotics toolbox reinvented for python. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11357–11363. IEEE.
- [Michel, 2004] Michel, O. (2004). Cyberbotics ltd. webotsTM: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):5.
- [Spong et al., 2020] Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2020). *Robot modeling and control*. Wiley, Hoboken, NJ, second edition edition.

