



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. Glauber Rodrigues Leite

Universidade Federal de Alagoas
Instituto de Computação
Maceió, Alagoas
26 de Julho 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. Glauber Rodrigues Leite

Banca Avaliadora:

Ícaro Bezerra Queiroz de Araújo Prof. Dr., UFAL
Arthur da Costa Vangasse Msc., UFMG

Maceió, Alagoas
26 de Julho de 2024



UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL

Instituto de Computação - IC

Campus A. C. Simões - Av. Lourival de Melo Mota, BL 12

Tabuleiro do Martins, Maceió/AL - CEP: 57.072-970

Telefone: (082) 3214-1401



Trabalho de Conclusão de Curso - TCC

Formulário de Avaliação

Nome do Aluno

H	U	G	O	T	A	L	L	Y	S	M	A	R	T	I	N	S
O	L	I	V	E	I	R	A									

Nº de Matrícula

1	7	1	1	1	6	6	3									
---	---	---	---	---	---	---	---	--	--	--	--	--	--	--	--	--

Título do TCC (Tema)

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

Banca Examinadora

Prof. Dr. Glauber Rodrigues Leite
Orientador

Glauber Rodrigues Leite
Assinatura

Prof. Dr. Ícaro Bezerra Queiroz de Araújo
Membro da banca

Ícaro B. Q. de Araújo
Assinatura

Msc. Arthur da Costa Vangasse
Membro da banca

Arthur da Costa Vangasse
Assinatura

Data da Defesa

26/07/2024

Nota Obtida

10,0 (Dez)

Observações

Coordenador do Curso De Acordo	<p><u>José Anísio Nascimento</u> Assinatura</p>
-----------------------------------	---



Dedicatória

Dedico este trabalho em primeiro lugar à minha família, cujo incentivo aos estudos foi fundamental para que eu chegasse até aqui. Em segundo lugar, à toda a comunidade acadêmica da Universidade Federal de Alagoas, sem a qual este trabalho jamais poderia ser concretizado. Em terceiro e último lugar, mas não menos importante, dedico a você, leitor, que se interessou de alguma maneira pelo mundo fascinante da robótica.

Agradecimentos

Em primeiro lugar, agradeço à minha família. Aos meus pais, Josenildo e Gilvanice, pelo amor e apoio incondicional ao longo de toda minha formação como ser humano e também durante minha jornada acadêmica. À minha irmã, Andressa, pelo exemplo de perseverança, inspiração e por sempre estar ao meu lado, tanto nos momentos de dificuldade como nos de alegria.

Ao corpo docente do IC em especial aos professores do curso de Engenharia da Computação, Erick de Andrade e Thiago Cordeiro, sou imensamente grato pelo conhecimento compartilhado durante minha formação. Um agradecimento especial ao meu orientador Glauber Leite que me introduziu ao mundo fascinante da robótica, durante o curso e também em alguns projetos de desenvolvimento que tive de oportunidade de participar. Sua orientação, conselhos e paciência foram fundamentais para que esse trabalho chegasse a sua forma final. Do mesmo modo não posso deixar de agradecer ao professor Ícaro Araújo pelo acolhimento durante a graduação não só na participação de projetos relevantes para a minha formação, mas também no esforço contínuo em trazer para o ambiente acadêmico do IC um grupo que une estudantes e professores em prol da pesquisa e divulgação científica.

Ao corpo administrativo do instituto, especialmente à secretaria, em nome do técnico Marcelo de Gusmão, agradeço pela eficiência e suporte contínuo, garantindo que todos os processos administrativos fossem realizados de forma ágil e sem complicações.

Aos meus amigos de graduação, Augusto, Valério, João, Kevin, Sofia, Álvaro, Roberto, Rafael e Thiago: vocês foram responsáveis por tornaram essa jornada mais divertida, meu sincero agradecimento. A amizade e o companheirismo de vocês nos momentos mais tensos e também nas horas de descontração foram essenciais para enfrentar os desafios e chegar até aqui.

A realização desta monografia só foi possível graças ao apoio e contribuição de muitas outras pessoas, às quais com certeza não puderam ser explicitamente mencionadas aqui. Gostaria de expressar minha mais profunda gratidão, a todos, o meu mais sincero obrigado!

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

Resumo

Manipuladores cinematicamente redundantes possuem mais graus de liberdade (DoFs) do que o necessário para realizar uma tarefa específica, oferecendo maior flexibilidade e destreza na execução de trajetórias. No entanto, esses DoFs adicionais também introduzem uma maior complexidade na cinemática inversa do manipulador, sendo necessário recorrer a esquemas de resolução de redundância, para encontrar uma solução particular otimizando critérios de desempenho do manipulador, como velocidade das juntas, manipulabilidade ou distância para obstáculos. Este trabalho, apresenta uma fundamentação teórica sobre os principais pontos relativos a cinemática inversa diferencial voltada para manipuladores redundantes. Para validar a metodologia proposta, foi implementado um ambiente virtual utilizando o simulador Webots e o Sistema Operacional de Robôs (ROS), permitindo a aplicação prática do esquema de controle *Resolved Rate Motion Control* (RRMC). As simulações realizadas mostram que o controlador é capaz de otimizar métricas de desempenho, sem violar as restrições cinemáticas primárias em diferentes cenários de execução de trajetórias cartesianas.

Palavras-chave: *manipuladores robóticos; cinemática inversa diferencial; resolução de redundância; simulação de robótica; resolved rate motion control.*

Abstract

Kinematically redundant manipulators have more degrees of freedom (DoFs) than necessary to perform a specific task, offering greater flexibility and dexterity in executing trajectories. However, these additional DoFs also introduce greater complexity in the manipulator's inverse kinematics, requiring the use of redundancy resolution schemes to find a particular solution that optimizes the manipulator's performance criteria, such as joint speed, manipulability, or distance to obstacles. This work presents a theoretical foundation on the main points related to differential inverse kinematics for redundant manipulators. To validate the proposed methodology, a virtual environment was implemented using the Webots simulator and the Robot Operating System (ROS), allowing the practical application of the *Resolved Rate Motion Control* (RRMC) control scheme. The simulations carried out show that the controller is capable of optimizing performance metrics without violating the primary kinematic constraints in different Cartesian trajectory execution scenarios.

Keywords: *robotic manipulators; differential inverse kinematics; redundancy resolution; robotics simulation; resolved rate motion control.*

Listas de Figuras

2.1	Representação de um ponto p em diferentes frames.	5
2.2	Atribuição de <i>frames</i> de acordo com a convenção de Denavit-Hartenberg. Adaptado de [Spong et al., 2020].	6
2.3	Braço planar do tipo 3R.	8
2.4	Elipsoide de manipulabilidade para diferentes configurações do manipulador planar 3R.	17
3.1	Modelo virtual do manipulador no Webots.	20
3.2	Cadeia cinemática visualizada no RViz.	22
3.3	Geometria de colisão do modelo virtual do manipulador.	23
3.4	Componentes que constituem uma ação dentro do ROS. Adaptado de [Bassa, 2023, p. 36].	24
3.5	Arquitetura de comunicação proposta para controle do manipulador.	25
3.6	Diagrama de blocos do controlador RRMC	25
3.7	Etapas seguidas na simulação das trajetórias executadas.	28
4.1	Diminuição da distância para o limite mecânico das juntas ao longo tempo.	30
4.2	Diferentes valores do ganho, influenciam diretamente na velocidade de convergência para a solução particular.	30
4.3	Posição do efetuador final mantém-se constante ao longo do tempo, devido à restrição primária.	31
4.4	Configuração final do manipulador nas simulações do segundo cenário (métrica distância das juntas).	32
4.5	Manipulabilidade em função do tempo para cada trajetória executada no segundo cenário.	33
4.6	Otimizando distância das juntas no segundo cenário.	33
4.7	Em determinadas trajetórias, o impacto do ganho na velocidade das juntas é significativo.	34
4.8	Erro da posição ao longo do tempo para o segundo cenário.	34

Lista de Tabelas

2.1	Descrição dos parâmetros Denavit-Hartenberg	7
2.2	Parâmetros DH para o braço planar 3R da figura 2.3.	8
3.1	Parâmetros DH para o manipulador 5R.	21
4.1	Valores absolutos de w_d obtidos na execução das simulações do primeiro cenário.	29
4.2	Valores de desempenho obtidos na execução das simulações do segundo cenário.	32
4.3	Valores de desempenho (métrica distância das juntas) obtidos na simulações no segundo cenário.	33

Listas de Símbolos

\mathbb{R}^3 Espaço tridimensional real.

$SO(3)$ Grupo especial ortogonal de ordem 3.

$SE(3)$ Grupo especial euclidiano de ordem 3.

q Vetor de configuração das juntas.

\dot{q} Vetor de velocidades das juntas.

\dot{q}_0 Vetor arbitrário de velocidades das juntas.

q_{iM} Limite superior de operação da junta.

q_{im} Limite inferior de operação da junta.

\bar{q}_i Valor médio de operação da junta.

$A_i(q_i)$ Matriz de transformação homogênea associada ao elo i .

T_n^0 Matriz de transformação homogênea da cinemática direta.

J Matriz jacobiana.

J^\dagger Pseudo-inversa da matriz jacobiana.

ξ Velocidade do efetuador final.

μ Medida de manipulabilidade.

w Métrica de performance.

$I(w)$ Índice de desempenho da métrica w .

E_i Posição inicial do efetuador.

E_t Posição desejada do efetuador.

Lista de Abreviaturas

DH Denavit-Hartenberg.

DoF *Degrees of Freedom*

IC Instituto de Computação.

PLA Ácido Polilático.

ROS *Robot Operating System.*

RRMC *Resolved Rate Motion Control.*

RTB *Robotics Toolbox for Python.*

RViz *ROS Visualization* (ferramenta de visualização do ROS).

SVD *Singular Value Decomposition.*

UFAL Universidade Federal de Alagoas.

UFMG Universidade Federal de Minas Gerais.

Sumário

1	Introdução	1
1.1	Motivação	2
1.2	Objetivos	3
1.2.1	Objetivos Gerais	3
1.2.2	Objetivos Específicos	3
1.3	Estrutura do texto	3
2	Fundamentação teórica	4
2.1	Representação de Poses no espaço	4
2.2	Cinemática Direta	6
2.2.1	Cinemática direta de um braço planar	7
2.3	Cinemática Diferencial	9
2.3.1	A Jacobiana do Manipulador	10
2.3.2	Singularidades	11
2.3.3	Cinemática Inversa Diferencial	13
2.4	Manipuladores redundantes	14
2.4.1	Pseudo-Inversa da Jacobiana	14
2.4.2	Medida de Manipulabilidade	15
2.4.3	Resolução de Redundância	16
3	Implementação	19
3.1	Simulação de manipuladores robóticos	19
3.2	Arquitetura de comunicação	22
3.3	O Algoritmo <i>Resolved Rate Motion Control</i>	25
3.4	Simulações	26
4	Resultados	29
4.1	Cenário 1	29
4.2	Cenário 2	32
5	Conclusão	35

Capítulo 1

Introdução

A redundância cinemática na robótica refere-se ao uso de graus de liberdade, em inglês *degrees of freedom* (DoF) adicionais além do mínimo necessário para executar uma tarefa específica. Em robôs industriais e manipuladores, singularidades cinemáticas ocorrem quando a configuração do robô resulta na perda de um ou mais DoFs, reduzindo sua capacidade de movimento. De maneira mais precisa, isso significa que há alguma direção (ou subespaço) no espaço cartesiano ao longo da qual é impossível mover o efetuador final, independentemente das velocidades empregadas nas juntas [Craig, 2004].

A importância da redundância está na capacidade de fornecer caminhos alternativos para o planejamento de movimentos, permitindo que o robô manobre ao redor de configurações singulares, mantendo a eficiência operacional e a segurança. Essa flexibilidade assegura que o robô continue a executar suas tarefas mesmo próximo a pontos singulares, comuns em tarefas complexas. Além disso, a resolução de redundância possibilita a otimização de outros critérios de desempenho, como minimizar o consumo de energia, reduzir o desgaste, melhorar a precisão e aumentar a capacidade de evasão de obstáculos.

Recentemente, [Li et al., 2023] propuseram um novo esquema de *Obstacle avoidance* (OA) para manipuladores redundantes baseado na pseudo-inversa da matriz jacobiana no nível das acelerações das juntas, demonstrando a viabilidade do método através de simulações com o manipulador robótico PA10 o qual possui 7 DoFs. Por outro lado, [Kuri et al., 2023] abordaram uma técnica de controle inteligente para um manipulador robótico espacial com 5 DoFs para executar trajetórias tridimensionais desejadas (linha, circular e elíptica) enquanto resolve a redundância (evitação dos limites das juntas) mostrando que a depender da tarefa, mesmo para manipuladores com um número menor de DoFs, a redundância pode ser explorada para melhorar o desempenho do robô.

O trabalho de [Ancona, 2017] validou técnicas de resolução de redundância em diversos cenários, destacando a importância da escolha do critério de desempenho para direcionar o comportamento do manipulador. A metodologia proposta, baseada em um procedimento de cinemática inversa e um controlador orientado a objetos, mostrou-se eficaz e versátil tanto em simulações quanto na prática real. Além disso, o trabalho explorou

a manipulação móvel redundante para operações hábeis e interação segura entre humanos e robôs em ambientes industriais, com uma contribuição importante na arquitetura de controle de software proposta, que é escalável, portátil, integrável com tecnologias heterogêneas e pronta para uso em ambientes de produção.

Por fim, [Hammond, 2011] propôs um método heurístico para configurar manipuladores robóticos com redundância cinemática, visando aumentar a precisão dos movimentos em tarefas específicas, utilizando o índice de resolução de movimento efetivo (EMR). Os resultados mostram que o índice EMR pode ser usado tanto como uma métrica de aptidão em otimização de design quanto como um objetivo na resolução de redundância, destacando o potencial dessas abordagens em aplicações de micromanipulação e montagem de precisão, melhorando significativamente a resolução de movimento em trajetórias específicas sem a necessidade de componentes de alta precisão e custo elevado.

1.1 Motivação

A abordagem tradicional no projeto de manipuladores teve como foco principal a minimização de custos e manutenção, utilizando o número mínimo de juntas necessárias para uma executar uma determinada tarefa levando, por exemplo, ao desenvolvimento dos robôs *Selective Compliance Assembly Robot Arm* (SCARA) para operações de *pick-and-place* [Siciliano and Khatib, 2008]. No entanto, essa abordagem minimalista apresenta uma série de limitações em aplicações do mundo real, onde fatores como limites de juntas, singularidades e obstáculos no espaço de trabalho estão presentes. Ao ter mais DoFs do que o estritamente necessário, os manipuladores redundantes podem alcançar maior destreza e versatilidade, tornando-os mais adequados para ambientes complexos e dinamicamente mutáveis.

Para além das restrições cinemáticas, a redundância permite a otimização de métricas de desempenho, como a minimização do torque ou do consumo de energia, melhorando a eficiência geral do sistema. Vale ressaltar que projetar manipuladores com juntas adicionais e garantir sua confiabilidade operacional é um processo complexo e custoso. Esquemas eficazes de resolução de redundância são críticos para o sucesso do planejamento e controle de movimentos, especialmente em ambientes dinâmicos. Apesar desses desafios, os benefícios da redundância cinemática em aumentar a destreza, versatilidade e eficiência fazem dela uma abordagem interessante em sistemas robóticos avançados.

1.2 Objetivos

1.2.1 Objetivos Gerais

Estudar e implementar a modelagem de cinemática inversa diferencial em manipuladores seriais redundantes, possibilitando a execução de trajetórias no espaço de trabalho que levam em conta não só as restrições cinemáticas do movimento mas também critérios de desempenho do manipulador.

1.2.2 Objetivos Específicos

- Estudar a modelagem de cadeias cinemáticas em manipuladores robóticos seriais
- Investigar o uso da cinemática direta e inversa diferenciais para execução de trajetórias cartesianas
- Implementar a lei de controle utilizando a cinemática direta diferencial e esquemas de resolução de redundância utilizando a pseudo inversa da matriz jacobiana
- Propor um ambiente virtual para execução de simulações utilizando o modelo de um manipulador robótico com 5 graus de liberdade
- Analisar o desempenho do esquema de controle em diferentes cenários de execução de trajetórias e critérios de desempenho.

1.3 Estrutura do texto

Este trabalho está estruturado de modo a introduzir os conceitos já difundidos da cinemática diferencial em manipuladores redundantes, bem como apresentar uma abordagem prática para resolução de redundância de um manipulador planar simples, com cinco graus de liberdade. No capítulo 2, é apresentada a fundamentação teórica, abordando conceitos essenciais como a representação de poses no espaço, transformações homogêneas, cinemática direta, cinemática diferencial, e a resolução de redundância. O capítulo 3 descreve a implementação prática, detalhando a simulação de manipuladores robóticos, a arquitetura de comunicação proposta e o algoritmo *Resolved Rate Motion Control*. No capítulo 4, são apresentados os resultados experimentais divididos em dois cenários distintos, seguidos pela conclusão no capítulo 5, onde são discutidos os principais pontos abordados e sugestões para trabalhos futuros.

Capítulo 2

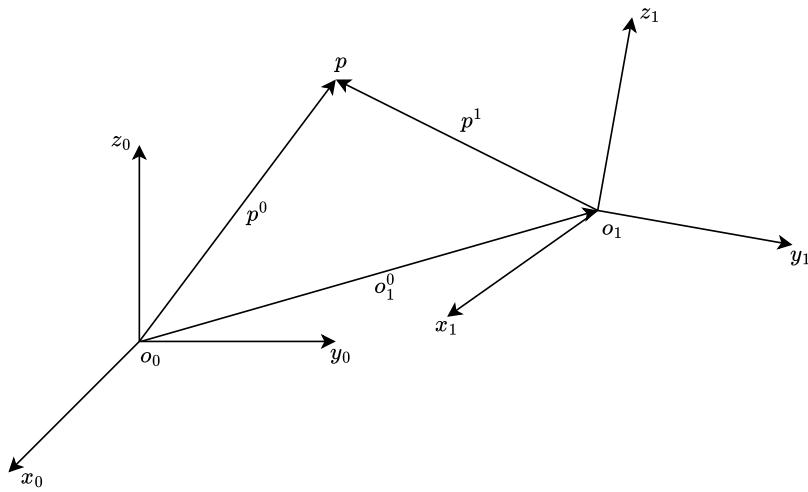
Fundamentação teórica

Este capítulo aborda diversos tópicos fundamentais para a compreensão da técnica de resolução de redundância utilizada ao longo do trabalho. Inicialmente, é discutido o uso de transformações homogêneas para estabelecer a relação entre dois *frames* da cadeia cinemática, dada por um deslocamento rígido no espaço de trabalho de um manipulador. Dada uma configuração do manipulador no espaço das juntas, calculamos a pose do efetuador final relativa à ao *frame* da base, resolvendo assim o problema de cinemática direta. Em seguida, a cinemática diferencial é introduzida para relacionar velocidades no espaço de configuração do manipulador com sua velocidade no espaço de trabalho, utilizando a matriz jacobiana. Em posse do mapeamento estabelecido para jacobiana torna-se possível resolver o problema da cinemática inversa diferencial onde são discutidos tópicos referentes às singularidades cinemáticas que surgem quando a matriz jacobiana não admite inversa e como a sua pseudo-inversa pode ser utilizada, em manipuladores redundantes, para a escolha soluções ótimas no espaço de configuração do manipulador.

2.1 Representação de Poses no espaço

A representação de poses no espaço na robótica envolve o uso de sistemas de coordenadas, ou *frames*, para descrever as posições e orientações de corpos rígidos, além das transformações entre esses frames. Um frame é composto por um ponto e um conjunto de vetores ortonormais que formam uma base do \mathbb{R}^3 . No contexto da robótica fixa, é comum especificar tarefas utilizando coordenadas cartesianas relativas a um frame de referência. Por exemplo, um ponto o_1^0 , representando uma translação, pode ser expresso como um vetor com origem em um frame $o_0x_0y_0z_0$ e extremidade em outro frame $o_1x_1y_1z_1$, onde o expoente indica em relação a qual frame as coordenadas são escritas. A orientação entre esses mesmos frames pode ser representada por uma matriz de rotação R_1^0 , pertencente ao grupo especial ortogonal de ordem 3 ($SO(3)$), que possui a útil propriedade no cálculo da sua inversa: $(R_1^0)^{-1} = (R_1^0)^\top = R_0^1$.

O grupo especial euclidiano de ordem 3 ($SE(3)$) combina rotações e translações em

Figura 2.1: Representação de um ponto p em diferentes frames.

uma única representação, facilitando a descrição de deslocamentos rígidos no espaço tridimensional. Um deslocamento rígido é definido por um par ordenado (R, d) , onde R é uma matriz de rotação em $SO(3)$ e d é um vetor de translação em \mathbb{R}^3 . Observando a Figura 2.1, podemos ver que a posição de um ponto p com relação a um frame de referência $o_0x_0y_0z_0$ pode ser expressa como:

$$p^0 = R_1^0 p^1 + o_1^0 \quad (2.1)$$

Assim, a equação 2.1 descreve um deslocamento rígido (rotação e translação) de um vetor fixado entre os dois frames. Representando-a no formato matricial, utilizando coordenadas homogêneas (com uma dimensão extra igual à unidade), teremos:

$$\begin{bmatrix} p^0 \\ 1 \end{bmatrix} = \begin{bmatrix} R_1^0 & o_1^0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p^1 \\ 1 \end{bmatrix} \quad (2.2)$$

ou numa notação mais compacta:

$$\tilde{p}^0 = T_1^0 \tilde{p}^1 \quad (2.3)$$

A matriz T_1^0 de dimensão 4×4 é denominada *matriz de transformação homogênea* e tem a característica de que a sua inversa pode ser calculada através das componentes de rotação e translação que compõe o deslocamento rígido:

$$(T_1^0)^{-1} = \begin{bmatrix} R_1^0 & o_1^0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} (R_1^0)^\top & -(R_1^0)^\top o_1^0 \\ 0 & 1 \end{bmatrix} \quad (2.4)$$

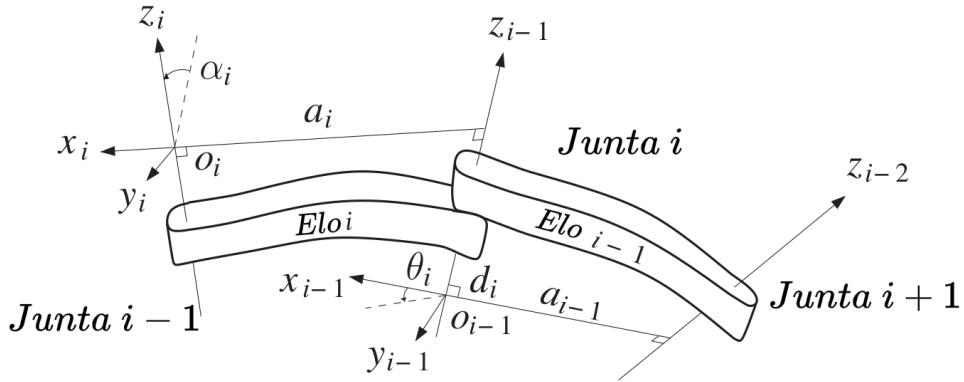


Figura 2.2: Atribuição de *frames* de acordo com a convenção de Denavit-Hartenberg. Adaptado de [Spong et al., 2020].

2.2 Cinemática Direta

Na robótica, uma cadeia cinemática é uma série de corpos rígidos conectados por juntas que permitem movimento relativo das partes móveis de um manipulador. Cadeias cinemáticas formam a base do estudo de manipuladores robóticos e geralmente são representadas por um grafo onde nós representam os elos e as arestas representam as juntas. Dependendo da topologia desse grafo, classifica-se uma cadeia cinemática de diferentes formas. Cadeias seriais do tipo aberta são as mais comuns no âmbito de manipuladores robóticos utilizados em aplicações industriais. Nesses casos, seu grafo é representado por uma 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).

As juntas prismáticas e de revolução são os tipos mais simples e introduzem um único grau de liberdade ao sistema. Juntas prismáticas permitem movimento translacional ao longo de uma direção, enquanto juntas de revolução possibilitam movimento rotacional ao redor de um eixo específico. Independentemente da complexidade das juntas, a maioria pode ser reduzida a uma combinação de juntas prismáticas e de revolução, facilitando a descrição das cadeias cinemáticas.

Para calcular a cinemática direta, a cada elo i da cadeia é associado um frame $o_i x_i y_i z_i$. Através da matriz de transformação homogênea $A_i(q_i)$, expressa-se a pose de $o_i x_i y_i z_i$ em relação a $o_{i-1} x_{i-1} y_{i-1} z_{i-1}$. Para um manipulador com n juntas a sua configuração é dada pelo vetor $q = [q_1 \ q_2 \ \dots \ q_n]^\top$ e a matriz de transformação homogênea T_n^0 que fornece a pose do efetuador final em relação ao frame da base é dada pela multiplicação matricial:

$$T_n^0(q) = A_1(q_1)A_2(q_2)\cdots A_n(q_n) = \begin{bmatrix} R_n^0 & o_n^0 \\ 0 & 1 \end{bmatrix} \quad (2.5)$$

Apesar da colocação de frames em cada elo ser arbitrária, Jacques Denavit e Richard Hartenberg introduziram uma convenção padronizada para fixação dos fra-

mes fornecendo uma representação mínima das transformações utilizando apenas 4 parâmetros [Denavit and Hartenberg, 1955]: ângulo da junta (θ_i), deslocamento do elo (d_i), comprimento do elo (a_i) e torção do elo (α_i). A figura 2.2 ilustra a relação entre dois frames de uma cadeia cinemática que segue tal convenção, fornecendo uma interpretação geométrica de cada parâmetro.

$$A_i = \text{Rot}_z(\theta_i) \cdot \text{Trans}_z(d_i) \cdot \text{Trans}_x(a_i) \cdot \text{Rot}_x(\alpha_i) \quad (2.6)$$

Tal representação mínima só é possível devido a introdução de duas restrições na colocação dos frames em cada elo [Spong et al., 2020]:

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

A tabela 2.1 descreve de maneira detalhada a definição de cada parâmetro na *convenção Denavit-Hartenberg* (DH).

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

2.2.1 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 por:

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 ($q_i = \theta_i$). Para $i = 1, 2, 3$ as matrizes A_i são calculadas com o auxílio da equação 2.6:

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.

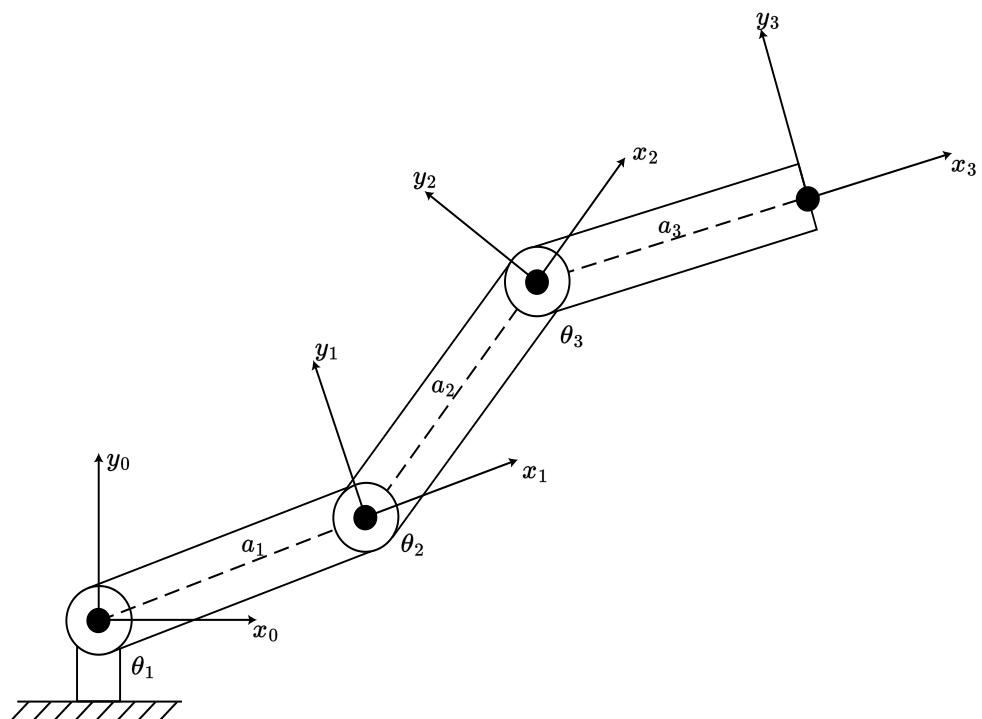


Figura 2.3: Braço planar do tipo 3R.

$$\mathbf{A}_i = \begin{bmatrix} c_i & -s_i & 0 & a_i c_i \\ s_i & c_i & 0 & a_i s_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

onde $\theta_1 + \theta_2 = \theta_{12}$, $\cos(\theta_1 + \theta_2) = c_{12}$ e assim por diante.

Já para as matrizes T_i^0 , utiliza-se a equação 2.5:

$$\mathbf{T}_1^0 = \mathbf{A}_1 \quad (2.8)$$

$$\mathbf{T}_2^0 = \mathbf{A}_1 \cdot \mathbf{A}_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1 c_1 + a_2 c_{12} \\ s_{12} & c_{12} & 0 & a_1 s_1 + a_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

$$\mathbf{T}_3^0 = \mathbf{A}_1 \cdot \mathbf{A}_2 \cdot \mathbf{A}_3 = \begin{bmatrix} c_{123} & -s_{123} & 0 & a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ s_{123} & c_{123} & 0 & a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

As três primeiras entradas da última coluna da matriz 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 foi apresentado como podemos estabelecer uma relação entre a configuração de um manipulador com n juntas e a pose do efetuador final no espaço SE3. Esta seção introduz o conceito da matriz jacobiana do manipulador, responsável por estabelecer um mapeamento linear entre um vetor de velocidades no espaço das juntas com a velocidade do efetuador no espaço de trabalho. A seção termina analisando o problema da cinemática inversa diferencial o qual proporciona a execução de trajetórias cartesianas, no espaço de trabalho do manipulador.

2.3.1 A Jacobiana do Manipulador

Dado um manipulador com n juntas vamos considerar

$$T_n^0(q) = \begin{bmatrix} R_n^0(q) & o_n^0(q) \\ 0 & 1 \end{bmatrix} \quad (2.11)$$

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 $q = [q_1 \ \dots \ q_n]^\top$.

As relações entre as componentes da velocidade do efetuador final são estabelecidas da seguinte forma:

$$v_n^0 = J_v \dot{q} \quad (2.12)$$

$$\omega_n^0 = J_\omega \dot{q} \quad (2.13)$$

onde v_n^0 e ω_n^0 expressam respectivamente as velocidades linear e angular do efetuador final relativas ao *frame* da base e J_v , J_ω são matrizes de dimensão $3 \times n$. De maneira mais compacta, pode-se escrever

$$\xi = J \dot{q} \quad (2.14)$$

onde:

$$\xi = \begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix} \text{ e } J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \quad (2.15)$$

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 $6 \times n$.

A parte superior da matriz jacobiana J_v será:

$$J_v = \begin{bmatrix} J_{v_1} & J_{v_2} & \dots & J_{v_n} \end{bmatrix} \quad (2.16)$$

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

$$J_{v_i} = \begin{cases} z_{i-1}^0 \times (o_n^0 - o_{i-1}^0) & \text{se a junta } i \text{ é de revolução} \\ z_{i-1} & \text{se a junta } i \text{ é prismática} \end{cases} \quad (2.17)$$

Já a parte inferior da matriz jacobiana J_ω será:

$$J_\omega = \begin{bmatrix} J_{\omega_1} & J_{\omega_2} & \dots & J_{\omega_n} \end{bmatrix} \quad (2.18)$$

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

$$J_{\omega_i} = \begin{cases} z_{i-1}^0 & \text{se a junta } i \text{ é de revolução} \\ 0 & \text{se a junta } i \text{ é prismática} \end{cases} \quad (2.19)$$

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 eficiente 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:

$$J(q) = \begin{bmatrix} z_0^0 \times (o_3^0 - o_0^0) & z_1^0 \times (o_3^0 - o_1^0) & z_2^0 \times (o_3^0 - o_2^0) \\ z_0^0 & z_1^0 & z_2^0 \end{bmatrix} \quad (2.20)$$

onde as 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, obtém-se:

$$J(q) = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - a_3 s_{123} & -a_2 s_{12} - a_3 s_{123} & -a_3 s_{123} \\ a_1 c_1 + a_2 c_{12} + a_3 c_{123} & a_2 c_{12} + a_3 c_{123} & a_3 c_{123} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (2.21)$$

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 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:

$$\xi = J(q)\dot{q} \quad (2.22)$$

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:

$$\xi = J_1\dot{q}_1 + J_2\dot{q}_2 + \cdots + J_n\dot{q}_n \quad (2.23)$$

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 têm-se $n = 3$ e desse modo $\text{posto}(J) \leq 3$ evidenciando o fato de que o manipulador não consegue desenvolver uma velocidade arbitrária com todas as componentes translacionais e rotacionais.

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 [Spong et al., 2020]:

- 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, uma vez que é necessário resolver a equação

$$\det(J(q)) = 0 \quad (2.24)$$

que geralmente envolve termos com alto grau de não linearidade. Nas próximas seções, serão examinadas 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 (2.22) é quadrada e não singular, o problema de cinemática inversa pode ser resolvido através da simples inversão da mesma:

$$\dot{q} = J^{-1}(q)\xi \quad (2.25)$$

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

$$q(t) = q(0) + \int_0^t \dot{q}(\tau)d\tau \quad (2.26)$$

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 $t_{k+1} = t_k + \Delta t$:

$$q(t_{k+1}) = q(t_k) + \dot{q}(t_k)\delta_t \quad (2.27)$$

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

O esquema de controle descrito acima é conhecido como *Resolved Rate Motion Control* (RRMC), 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 [Corke, 2023]. Tal esquema é útil na geração de trajetórias contínuas e suaves no espaço de trabalho, conhecidas como trajetórias cartesianas.

Como exemplo, ainda considerando o manipulador planar 3R, ao especificar apenas 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 , têm-se que:

$$\xi = \begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} \quad (2.28)$$

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

$$J(q) = \begin{bmatrix} -a_1s_1 - a_2s_{12} - a_3s_{123} & -a_2s_{12} - a_3s_{123} & -a_3s_{123} \\ a_1c_1 + a_2c_{12} + a_3c_{123} & a_2c_{12} + a_3c_{123} & a_3c_{123} \\ 1 & 1 & 1 \end{bmatrix} \quad (2.29)$$

e contanto que não seja singular pode ser facilmente invertida. Como exemplo, para gerar um movimento retilíneo no efetuador final paralelo ao eixo x do plano xy com

velocidade constante, basta tomar $\xi = \begin{bmatrix} v_x & 0 & 0 \end{bmatrix}^\top$.

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. Esta seção introduz 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 obtém-se uma descrição geométrica e qualitativa da destreza associada à uma dada configuração através dos conceitos do elipsoide e da medida de manipulabilidade. Ao final da seção, utiliza-se 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

Em um 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 viável é formular o problema como um de otimização linear com restrições, onde a solução ótima é obtida minimizando o custo quadrático das velocidades das juntas:

$$\arg \min_{\dot{q}} \|\dot{q} - \dot{q}_0\|^2 \text{ sujeito a } J\dot{q} = \xi \quad (2.30)$$

onde \dot{q}_0 é um vetor de velocidades arbitrário. Pode ser mostrado que nesse caso, a solução ótima é dada por [Siciliano, 2009]:

$$\dot{q} = J^\dagger \xi + (I_n - J^\dagger J)\dot{q}_0 \quad (2.31)$$

onde a matriz J^\dagger é conhecida como *matriz inversa de Moore-Penrose* ou apenas *pseudo-inversa* de J e é dada por:

$$J^\dagger = J^\top (JJ^\top)^{-1} \quad (2.32)$$

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 (2.31):

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

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, considerando o disco formado pelo conjunto de velocidades com norma unitária:

$$\|\dot{q}\|^2 = q_1^2 + q_2^2 + \cdots + q_n^2 \leq 1 \quad (2.33)$$

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

$$\begin{aligned}
\|\dot{q}\|^2 &= \dot{q}^\top \dot{q} \\
&= (J^\dagger\xi)^\top J^\dagger\xi \\
&= \xi^\top (J^\top (JJ^\top)^{-1})^\top J^\top (JJ^\top)^{-1}\xi \\
&= \xi^\top (JJ^\top)^{-1} (JJ^\top) (JJ^\top)^{-1}\xi \\
\|\dot{q}\|^2 &= \xi^\top (JJ^\top)^{-1}\xi \leq 1
\end{aligned} \quad (2.34)$$

A equação (2.34) 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 verificado considerando-se a decomposição em valores singulares (SVD) da jacobiana $J = U\Sigma V^\top$:

$$\begin{aligned}
\|\dot{q}\|^2 &= \xi^\top (U\Sigma V^\top V\Sigma^\top U^\top)^{-1}\xi \\
&= \xi^\top (U\Sigma^2 U^\top)^{-1}\xi \\
&= \xi^\top U\Sigma^{-2} U^\top \xi \\
\|\dot{q}\|^2 &= (U^\top \xi)^\top \Sigma^{-2} (U^\top \xi)
\end{aligned} \quad (2.35)$$

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

$$\Sigma^{-2} = \begin{bmatrix} \sigma_1^{-2} & & & \\ & \sigma_2^{-2} & & \\ & & \ddots & \\ & & & \sigma_m^{-2} \end{bmatrix} \quad (2.36)$$

é diagonal e os termos que $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m$ são os valores singulares de J . Por fim, substituindo $w = U^\top \xi$ a equação (2.35) é reescrita como:

$$w^\top \Sigma^{-2} w = \sum_{i=1}^m \frac{w_i^2}{\sigma_i^2} \leq 1 \quad (2.37)$$

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 semi-eixos 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 :

$$\mu = \sigma_1 \sigma_2 \cdots \sigma_m \quad (2.38)$$

que é proporcional ao volume do elipsoide de manipulabilidade. Próximo a 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 achulado à medida que nos aproximamos do limite do espaço de trabalho.

2.4.3 Resolução de Redundância

A solução geral dada pela equação (2.31), introduz o termo dado pelo vetor 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 [Siciliano, 2009]:

$$\dot{q}_0 = k_0 \left(\frac{\partial w(q)}{\partial q} \right)^\top \quad (2.39)$$

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* [Yoshikawa, 1983]:

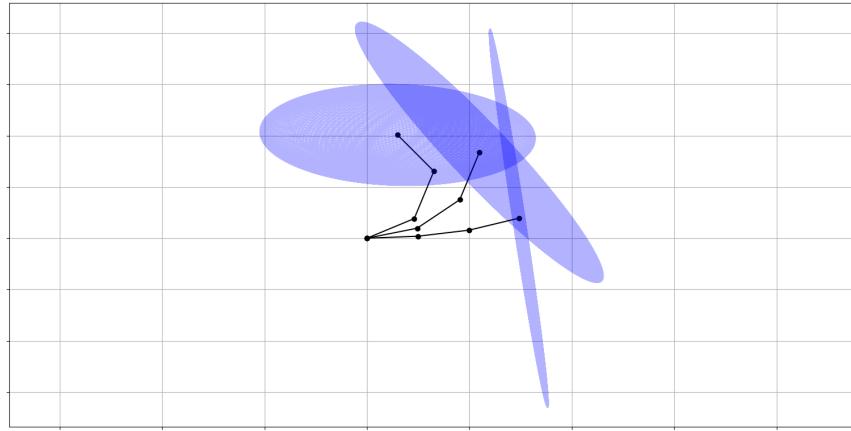


Figura 2.4: Elipsoide de manipulabilidade para diferentes configurações do manipulador planar 3R.

$$w_m(q) = \sqrt{\det(J(q)J(q)^\top)} \quad (2.40)$$

onde vale ressaltar que é equivalente àquela definida na equação (2.38) uma vez que se λ_i são os autovalores de JJ^\top então $\sigma_i = \sqrt{\lambda_i}$.

Uma alternativa ao cálculo da 2.40 é utilizar uma métrica mais simples como a *distância para os limites mecânicos das juntas* dada por:

$$w_d(q) = -\frac{1}{2n} \sum_{i=1}^n \left(\frac{q_i - \bar{q}_i}{q_{iM} - q_{im}} \right)^2 \quad (2.41)$$

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:

$$\frac{\partial w_d(q)}{\partial q_i} = -\frac{1}{n} \frac{q_i - \bar{q}_i}{(q_{iM} - q_{im})^2} \quad (2.42)$$

Outro ponto a salientar é que a escolha do tamanho do passo k_0 é crucial para a performance do algoritmo [Siciliano and Khatib, 2008]. 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 a uma diminuição ou até mesmo não convergência do valor de w devido a oscilações em torno do ponto de máximo local.

O próximo capítulo, aplica 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 cartesianas no espaço de trabalho $\mathcal{T} \in$

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

Este capítulo explora 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 das simulações. Por fim, iremos detalhar o esquema de controle e bibliotecas utilizadas na implementação do algoritmo RRMC para execução de trajetórias retilíneas bem como a metodologia utilizada para 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 estatística), 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, am-

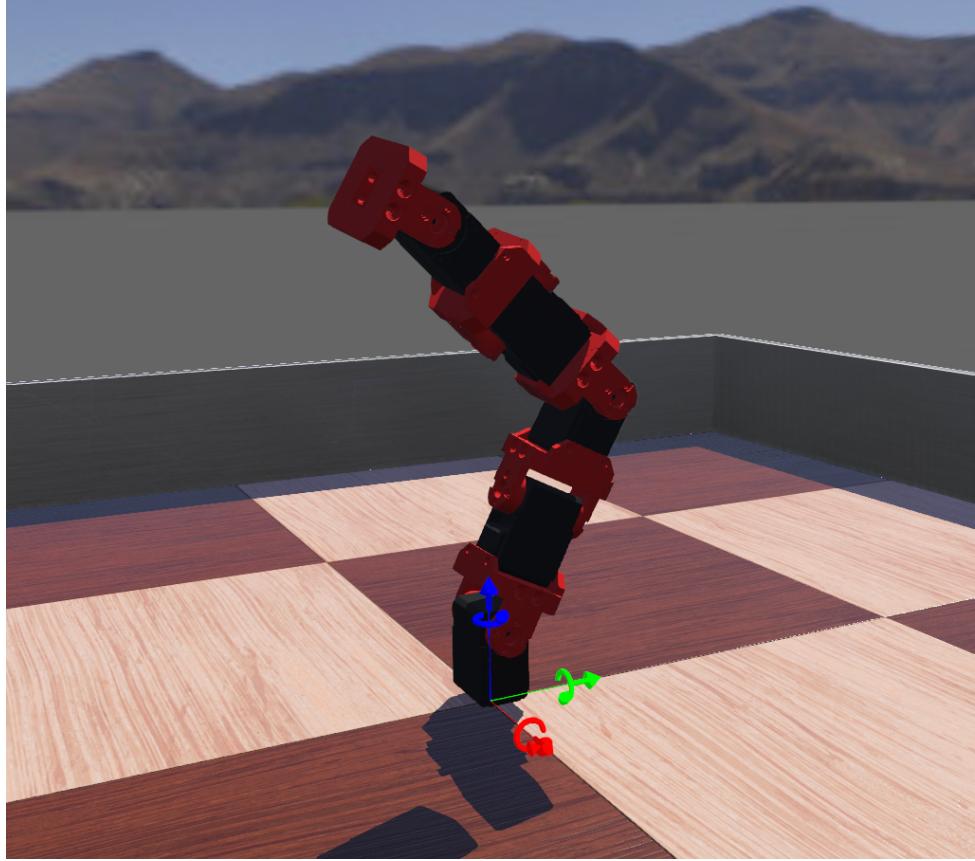


Figura 3.1: Modelo virtual do manipulador no Webots.

bientes aquáticos), linguagens de programação disponíveis para controle do robô, formatos suportados, extensibilidade, documentação entre outros.

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 ROS, 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, como indicado na figura 3.2. 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 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*

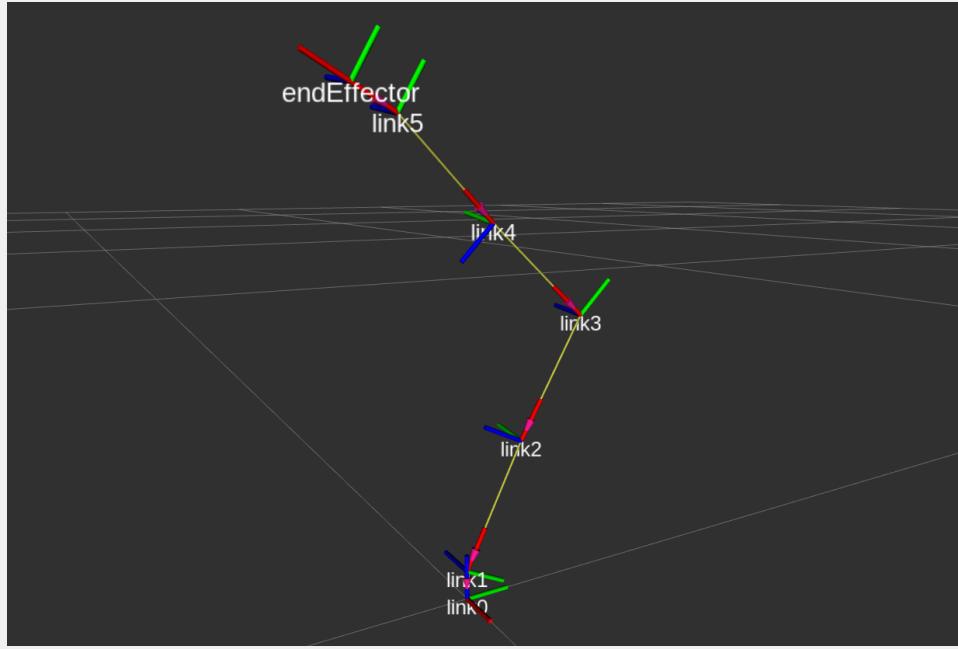


Figura 3.2: Cadeia cinemática visualizada no RViz.

(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 é estimada pelo próprio simulador 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 intrínseco 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 ROS, com uma documentação detalhada de como configurar um projeto e o uso básico de troca de mensagens entre diferentes processos.

A arquitetura de comunicação do ROS é baseada em uma estrutura de grafo, onde nós representam processos individuais que interagem com outros nós recebendo e enviando mensagens, através de tópicos. Idealmente, um nó deve ser responsável por uma única

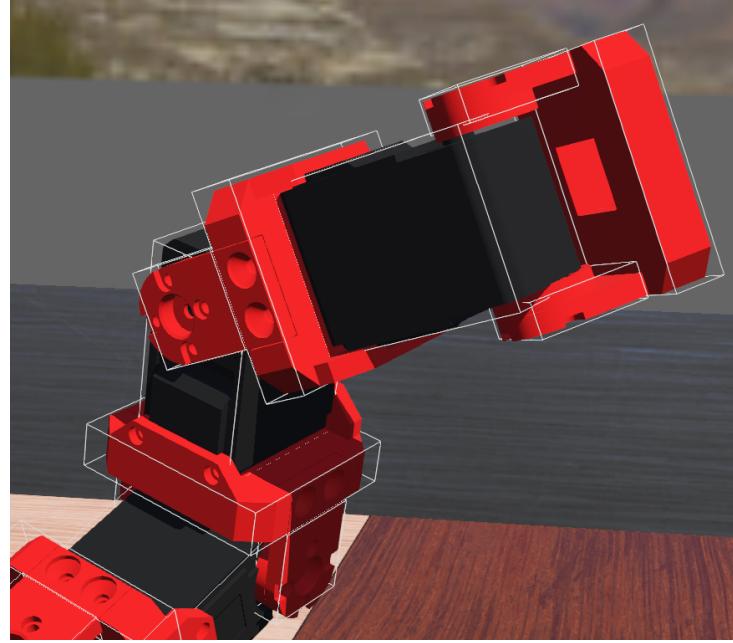


Figura 3.3: Geometria de colisão do modelo virtual do manipulador.

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.

Serviços e Ações no ROS

Serviços e ações constituem outras formas de comunicação entre nós no grafo do ROS e implementam uma abordagem de troca de mensagens do tipo cliente/servidor. Serviços representam ações que um nó pode executar com um início e fim definidos, retornando um único resultado e são normalmente usados para tarefas que possuem requisição e retorno, como por exemplo capturar uma imagem de um único quadro, dispensando o processamento contínuo. Por outro lado, ações são destinadas a tarefas de longa duração e são construídas com base em diferentes tópicos e serviços. A figura 3.4 ilustra as componentes fundamentais de um nó de ação: um objetivo, um *feedback* e um resultado. Um nó do tipo cliente de ação envia um objetivo para um nó servidor de ação, que reconhece a requisição, retorna um fluxo contínuo de dados através de um *feedback* até que a ação seja concluída ou cancelada, quando por fim retorna um resultado.

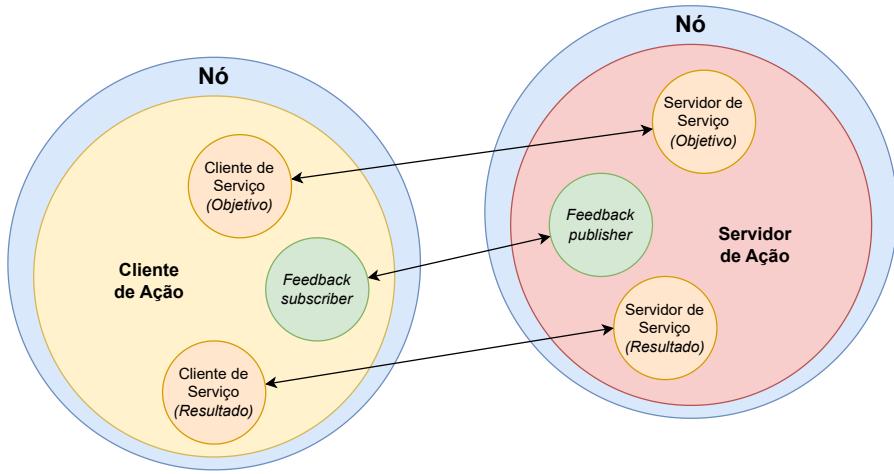


Figura 3.4: Componentes que constituem uma ação dentro do ROS. Adaptado de [Bassa, 2023, p. 36].

Grafo de comunicação no ROS

Com o objetivo de se executar trajetórias no espaço de trabalho do manipulador, a arquitetura de comunicação foi projetada de modo especificar um objetivo através de uma ação do tipo **trajectory_action** (nós *trajectory_action_server* e *trajectory_action_client*) e controlar o manipulador (nós *snake_driver* e *snake_controller*). A arquitetura de comunicação proposta para o controle do manipulador é ilustrada no grafo da figura 3.5, 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 instanciado 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 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.

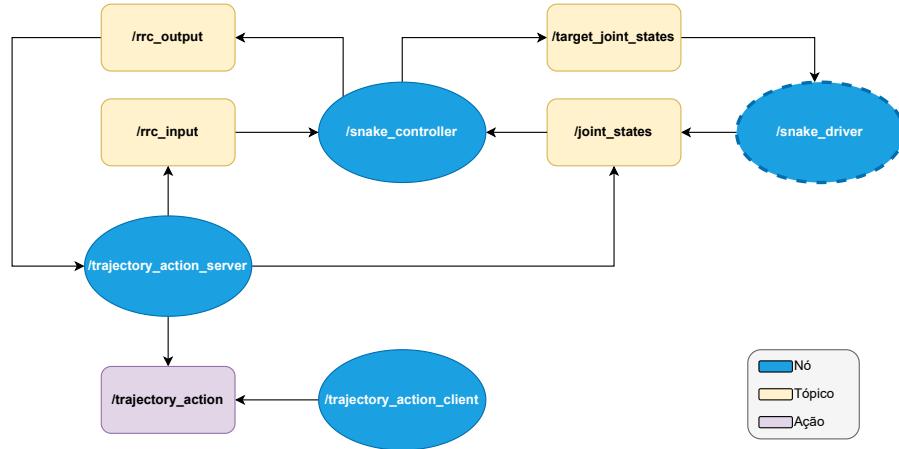


Figura 3.5: Arquitetura de comunicação proposta para controle do manipulador.

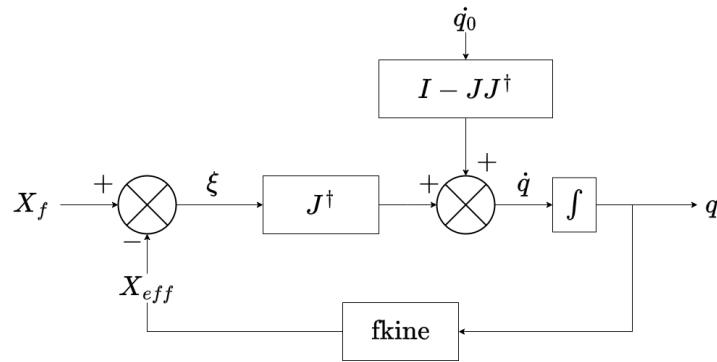


Figura 3.6: Diagrama de blocos do controlador RRMC

- **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 Motion Control*

Para realizar o controle da trajetória do manipulador e também a resolução da redundância, foi utilizado o esquema de controle *Resolved Rate Motion Control* (RRMC) definido no capítulo anterior. A figura 3.6 ilustra o diagrama de blocos do controlador RRMC implementado pelo nó *snake_controller*. Dada uma taxa de variação da posição do efetuador final ξ e um vetor de velocidades das juntas \dot{q}_0 , o controlador atua atualizando a configuração do manipulador de acordo com a equação 2.31.

Para a realização das simulações, foi implementado¹ o algoritmo 1 que escolhe o vetor

¹código disponível em <https://github.com/hugotallys/snakesim>

\dot{q}_0 de acordo com o gradiente dado pela equação 2.39. As duas métricas apresentadas anteriormente foram calculadas: *distância para os limites mecânicos das juntas* e *medida de manipulabilidade de Yoshikawa*. No primeiro caso, o gradiente é calculado analiticamente de acordo com a equação 2.42. Já para a manipulabilidade, o gradiente é estimado numericamente através de diferenças finitas, considerando um passo h suficientemente pequeno.

Algoritmo 1 *Resolved Rate Motion 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:        $q_{\text{mid}} \leftarrow 0.5 \times (q_{\text{max}}[i] + q_{\text{min}}[i])$ 
10:       $\dot{q}_0[i] \leftarrow (-k_0/n) \times (q[i] - q_{\text{mid}}) \div (q_{\text{max}}[i] - q_{\text{min}}[i])^2$ 
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  $\text{clipLimits}(q + \dot{q}\delta_t, q_{\text{max}}, q_{\text{min}})$            $\triangleright$  Restringe aos limites das juntas
20: end procedure

```

3.4 Simulações

As simulações focaram na avaliação da resolução de redundância sob valores variados do ganho (k_0) e envolveram dois cenários. No primeiro cenário, o manipulador foi controlado de modo a otimizar a distância para o limite mecânico das juntas, executando movimentos internos no seu espaço nulo, mantendo a posição do efetuador final estacionária. No segundo cenário, além de movimentos internos para otimizar seu índice de manipulabilidade, o manipulador seguiu uma trajetória cartesiana entre a posição inicial E_i e um outro ponto E_t correspondente à posição desejada no seu espaço de trabalho. Os índices de desempenho, configuração das juntas e a posição do efetuador final foram registrados na simulação de modo a se avaliar a eficácia do esquema da resolução de redundância na execução de tais simulações.

Em cada iteração do *loop de trajetória*, o vetor de velocidade é calculado e fornecido

como entrada para o controlador RRMC. A execução da trajetória para sempre que o número máximo de iterações é atingido (em ambos os cenários, definido como 500) ou se a norma do vetor de erro de posição se torna menor que 0,01 (apenas no segundo cenário).

Para avaliar o desempenho em cada trajetória, calculamos a pontuação:

$$I(w) = \int_0^t |w(q(\tau))| d\tau \quad (3.4)$$

a fim de capturar informações não apenas sobre o estado final da trajetória, mas também ao longo de toda a sua execução.

O fluxograma da figura 3.7 detalha o processo geral para a execução das trajetórias do manipulador robótico. Inicialmente, são escolhidas duas configurações (q_0 e q_f). Em seguida, calcula-se as posições inicial e desejada (E_i e E_t) do efetuador final em ambas as configurações. Verifica-se se há colisão com o chão (plano $z = 0$): caso positivo, retorna-se à escolha das configurações. Se não há colisão, escolhe-se um ganho k_0 e calcula-se o erro de posição ξ . Avalia-se se há condições de parada da trajetória; se sim, os resultados são salvos e a trajetória é reiniciada com um novo ganho k_0 . Se não, atualiza-se E_i conforme o RRMC e o processo continua até não haver mais ganhos k_0 para escolher, concluindo o a execução das trajetórias.

Vale ressaltar que devido a natureza complexa da determinação do espaço de trabalho do manipulador, procurou-se simplesmente escolher os pontos inicial e desejado de modo que sejam posições atingíveis no espaço de trabalho do manipulador, executando-se uma trajetória cartesiana na direção do vetor unitário entre os dois pontos. Além disso, a escolha de tais pontos foi feita de maneira aleatória ao longo de diversas simulações, de modo a não privilegiar nenhuma configuração específica do manipulador.

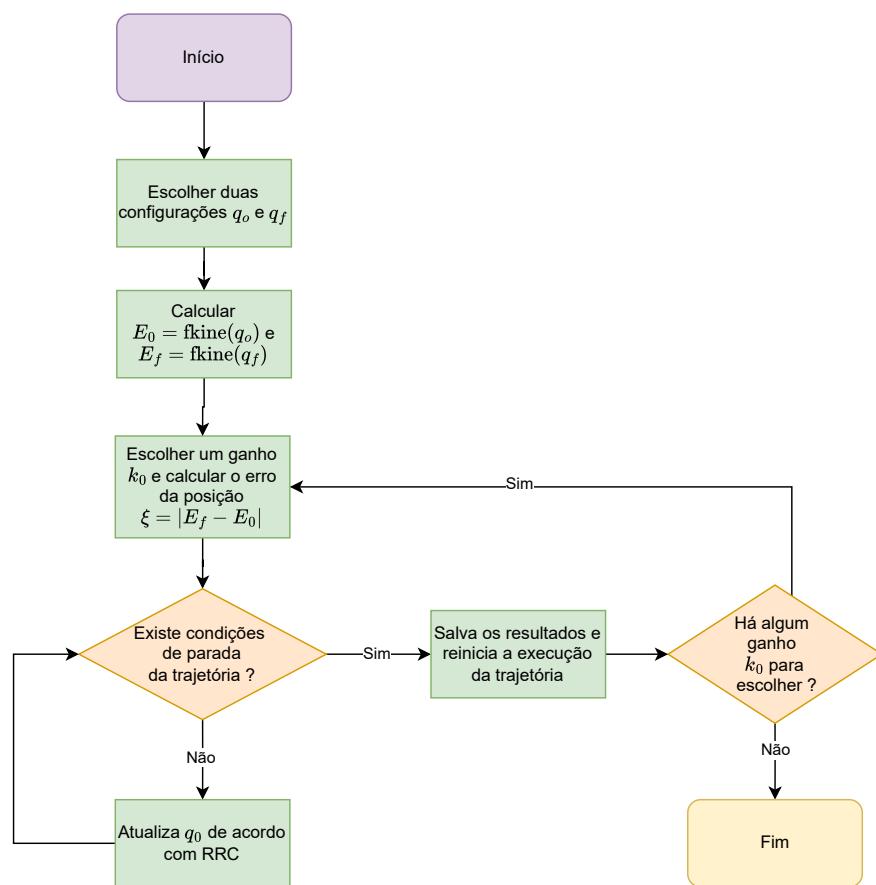


Figura 3.7: Etapas seguidas na simulação das trajetórias executadas.

Capítulo 4

Resultados

Este capítulo apresenta os resultados obtidos a partir da simulação da execução de trajetórias cartesianas para o manipulador 5R utilizando o esquema de controle RRMC. Em cada cenário, foi realizado um conjunto de 4 execuções de uma mesma trajetória variando-se o ganho k_0 , com o objetivo de avaliar a eficácia do controlador em otimizar os índices de desempenho fornecidos, submetido a diferentes restrições cinemáticas.

4.1 Cenário 1

No primeiro cenário, o controlador atualiza a configuração do manipulador de modo a trazer algumas posições das juntas o mais próximo possível de zero, enquanto a restrição primária imposta pela velocidade do efetuador final $\xi = 0$ mantém a configuração final do manipulador numa solução particular, como pode ser visto na Figura 4.2. Conforme aumenta-se o ganho k_0 , a configuração do manipulador converge mais rapidamente para a solução ótima e na figura 4.1 observa-se que valores menores do módulo da distância para os limites mecânicos das juntas w_d são obtidos, ao custo de um valor de erro ligeiramente maior, como indicado na tabela 4.1. Por fim, a Figura 4.3 mostra as três componentes de posição do efetuador final constantes ao longo de toda a simulação, mostrando que a restrição primária não é violada.

k_0	$I(w_d)$	Erro posição (m)
0	0.8729	1.8535e-06
200	0.4829	0.0013
400	0.4488	0.0023
800	0.4435	0.0048

Tabela 4.1: Valores absolutos de w_d obtidos na execução das simulações do primeiro cenário.

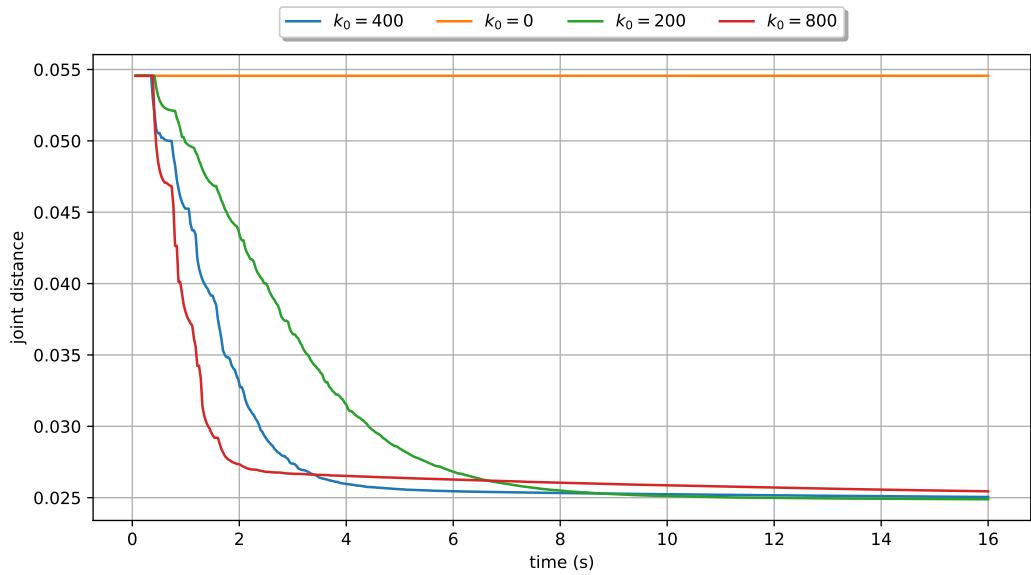


Figura 4.1: Diminuição da distância para o limite mecânico das juntas ao longo tempo.

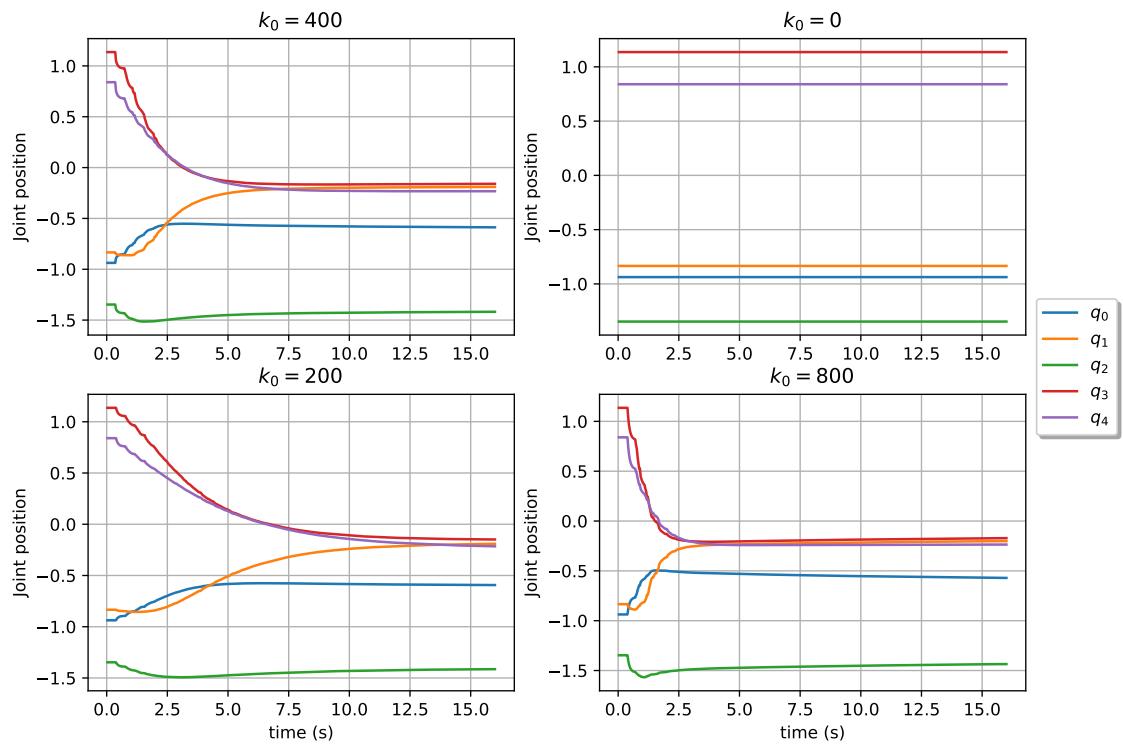


Figura 4.2: Diferentes valores do ganho, influenciam diretamente na velocidade de convergência para a solução particular.

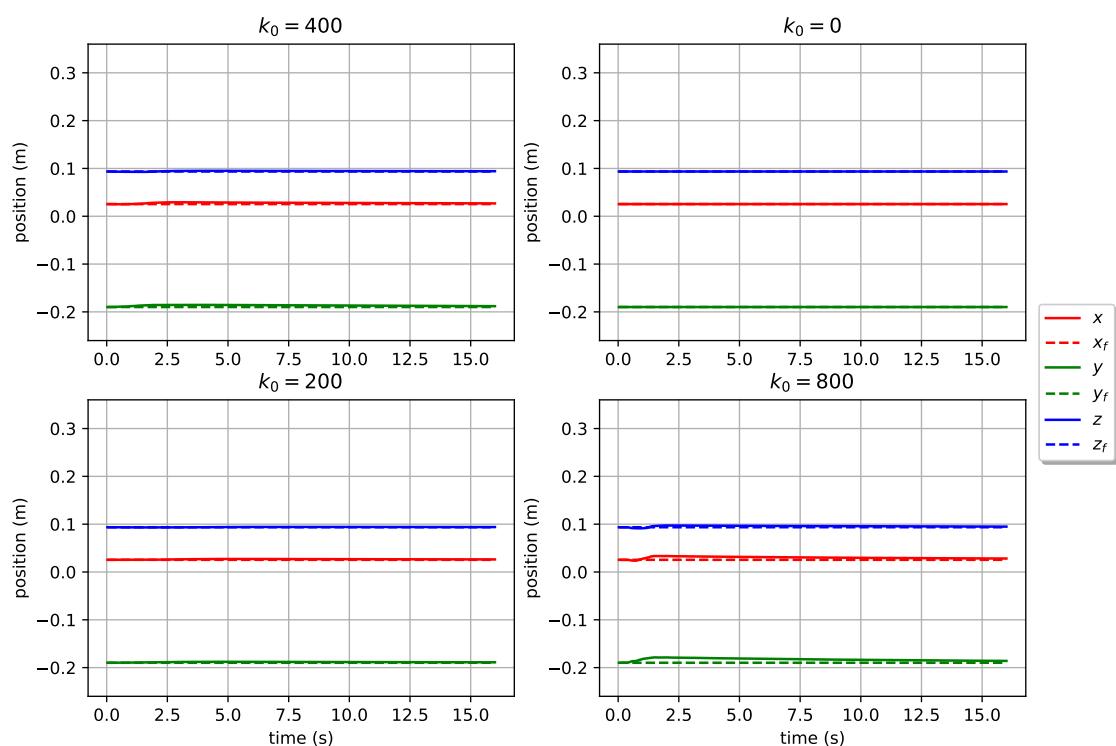
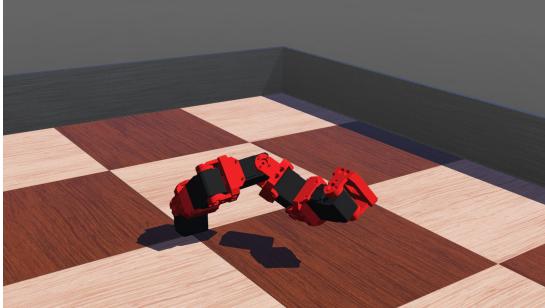


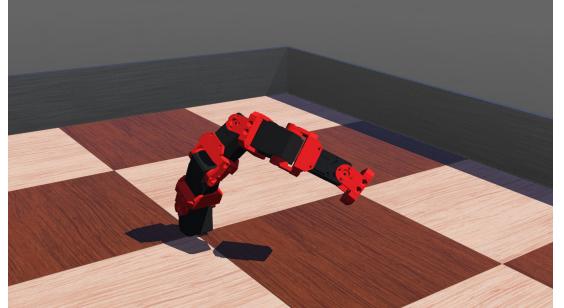
Figura 4.3: Posição do efetuador final mantém-se constante ao longo do tempo, devido à restrição primária.

4.2 Cenário 2

No segundo cenário, com a introdução de uma velocidade inicial não nula do efetuador final, isto é $\|\xi\| > 0$, a manipulabilidade w_m muda ao longo da execução da trajetória, aproximando-se de um valor mínimo no seu final (figura 4.5). A resolução de redundância atualiza a configuração do manipulador de modo maximizar a manipulabilidade sem violar a restrição primária.



(a) Configuração final para $k_0 = 0$.



(b) Configuração final para $k_0 = 800$.

Figura 4.4: Configuração final do manipulador nas simulações do segundo cenário (métrica distância das juntas).

Um ponto interessante a se observar é como o efeito de introduzir valores crescentes do ganho afeta o erro de posição, com um pequeno *trade-off* entre a otimização da métrica e a precisão no controle da posição do efetuador final (tabelas 4.2 e 4.3), tendo em vista o aumento das velocidades das juntas. Esse fenômeno pode ser melhor observado nas figuras 4.7 e 4.8, onde a velocidade das juntas aumenta significativamente ao longo da execução da trajetória, e o erro se mantém maior conforme o ganho k_0 aumenta. Isso pode ser explicado pelo fato de que as métricas de desempenho (no caso da Figura 4.6 w_d) são otimizadas como um objetivo secundário em relação à execução da trajetória. Desse modo, a escolha de trajetórias que minimizam o critério de desempenho acabam tornando o processo de otimização mais difícil, necessitando assim um valor de k_0 maior para convergir para uma solução ótima. Tal solução é demonstrada na figura 4.4, onde vemos a configuração final do manipulador, para o caso onde não há otimização da métrica ($k_0 = 0$) e quando a métrica é otimizada na direção do seu gradiente ($k_0 = 800$).

k_0	$I(w_m)$	Erro posição (m)
0	0.0507	0.0138
2000	0.0597	0.0150
4000	0.0606	0.0151
8000	0.0609	0.0149

Tabela 4.2: Valores de desempenho obtidos na execução das simulações do segundo cenário.

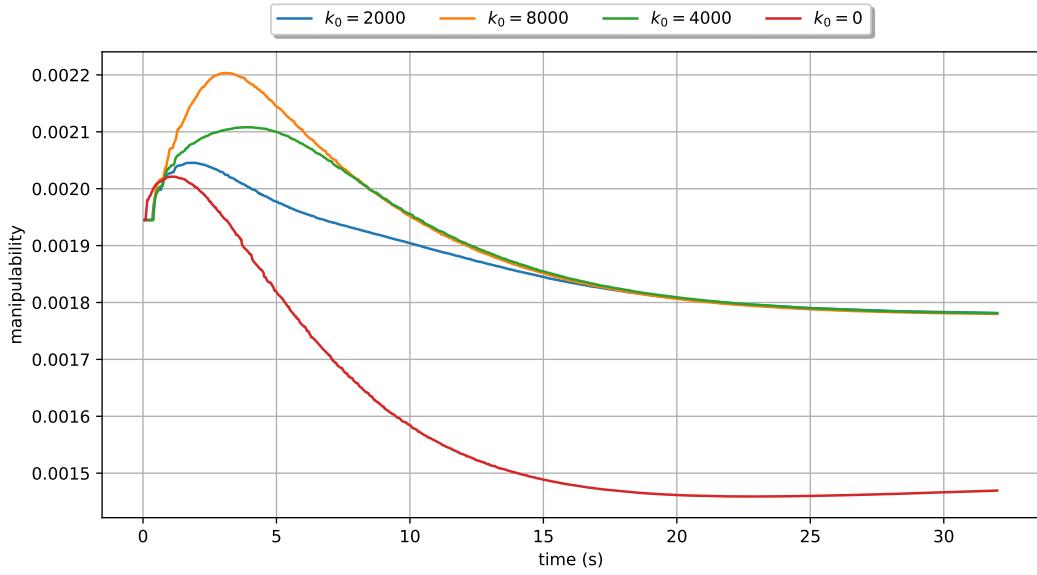


Figura 4.5: Manipulabilidade em função do tempo para cada trajetória executada no segundo cenário.

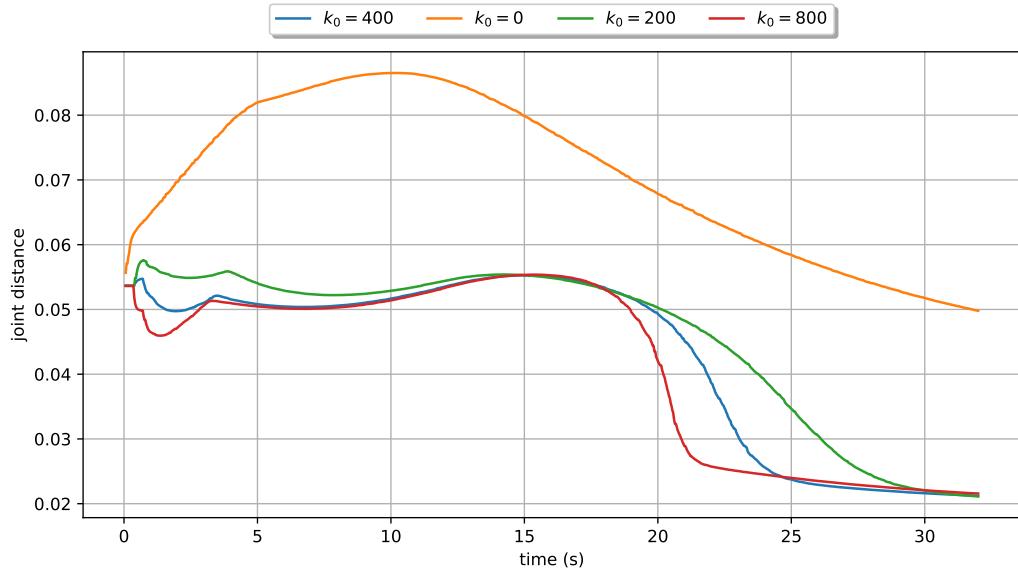


Figura 4.6: Otimizando distância das juntas no segundo cenário.

k_0	$I(w)$	Erro posição (m)
0	2.2622	0.0301
200	1.4727	0.0350
400	1.3777	0.0379
800	1.3251	0.0418

Tabela 4.3: Valores de desempenho (métrica distância das juntas) obtidos na simulações no segundo cenário.

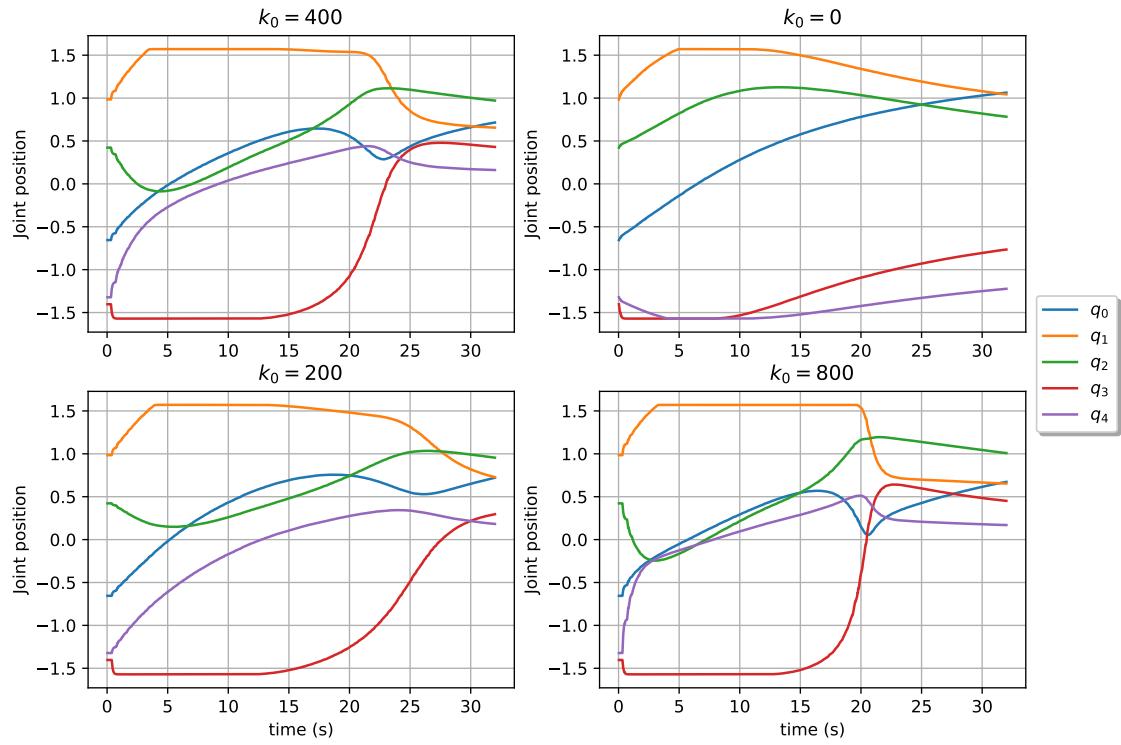


Figura 4.7: Em determinadas trajetórias, o impacto do ganho na velocidade das juntas é significativo.

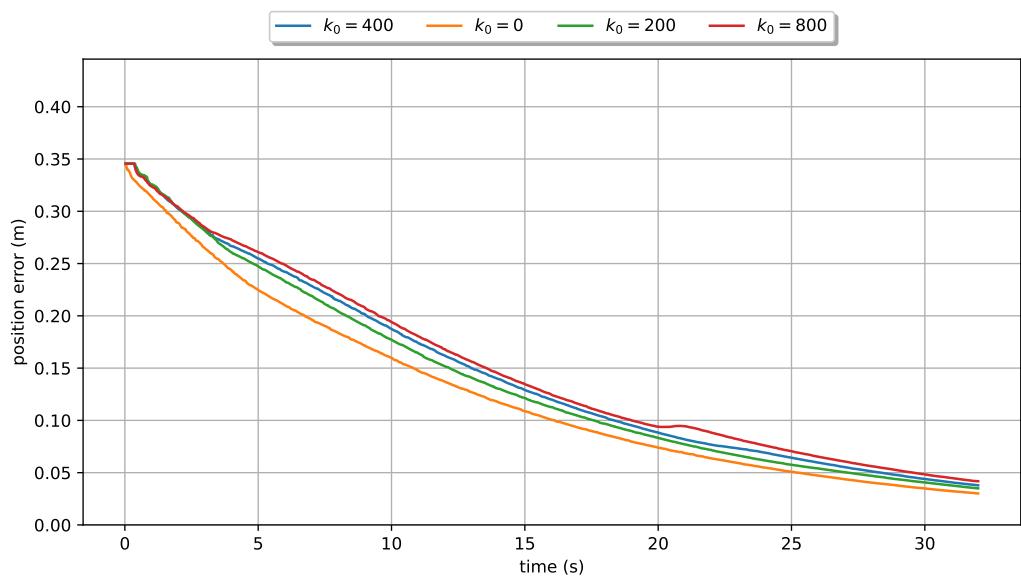


Figura 4.8: Erro da posição ao longo do tempo para o segundo cenário.

Capítulo 5

Conclusão

Este trabalho discutiu a aplicação de esquemas de controle com resolução de redundância para abordar os desafios impostos pelas singularidades na cinemática diferencial inversa de um manipulador serial com cinco graus de liberdade. Foi proposto ambiente completamente simulado para testar e validar a estratégia de controle, proporcionando também uma comunicação robusta entre controlador e o modelo virtual do manipulador usando o Sistema Operacional de Robôs (ROS). A abordagem foi avaliada em cenários distintos, propiciando uma análise qualitativa da execução de trajetórias em diferentes condições de restrições cinemáticas e de desempenho do manipulador.

Perspectivas futuras para o trabalho, incluem o estudo da escolha ótima de um ganho k_0 , visando otimizar o desempenho do manipulador em diferentes cenários e também uma análise teórica/estatística mais aprofundada dos resultados obtidos, com o objetivo de avaliar a robustez do controlador e a convergência para soluções ótimas. Além disso, é desejável explorar cadeias cinemáticas mais complexas, como as dos manipuladores do tipo *elbow* e *wrist*, que possuem um maior grau de redundância tipicamente maior que 7, permitindo a resolução não só a nível do controle da posição, mas também da orientação do efetuador final. Vale ressaltar também que uma continuidade natural para o trabalho é a transferência dos resultados para um robô real e a integração em um *framework* de planejamento de trajetórias, permitindo a validação do esquema de controle em cenários mais realistas e desafiadores.

Bibliografia

- [Ancona, 2017] Ancona, R. (2017). Redundancy modelling and resolution for robotic mobile manipulators: a general approach. *Advanced Robotics*, 31(13):706–715.
- [Bassa, 2023] Bassa, M. M. (2023). *A very informal journey through ROS 2*. Leanpub.
- [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.
- [Corke, 2023] Corke, P. I. (2023). *Robotics, vision and control: fundamental algorithms in Python*. Springer, Cham, 3rd ed., 2023 edition. OCLC: 1378937773.
- [Craig, 2004] Craig, J. J. (2004). *Introduction to Robotics: Mechanics and Control*. Pearson Prentice Hall, Upper Saddle River, NJ, 3rd edition.
- [Denavit and Hartenberg, 1955] Denavit, J. and Hartenberg, R. S. (1955). A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics*, 22(2):215–221.
- [Hammond, 2011] Hammond, F. L. (2011). Configuring kinematically redundant robotic manipulators to increase effective task-specific motion resolution. In *2011 IEEE International Conference on Mechatronics and Automation*, pages 34–39. IEEE.
- [Kuri et al., 2023] Kuri, S. K., Halder, K., and Orlando, M. F. (2023). Som network with weighted least norm matrix based redundancy resolution for a 5-dof spatial robotic manipulator. In *2023 7th International Conference on Computer Applications in Electrical Engineering-Recent Advances (CERA)*, pages 1–6. IEEE.
- [Li et al., 2023] Li, Q., Cang, N., Zhang, W., Guo, D., and Zhang, C. (2023). A pseudo-inverse redundancy-based resolution scheme at the acceleration level to control robotic arm motion. In *2023 6th International Conference on Robotics, Control and Automation Engineering (RCAE)*, pages 18–22. IEEE.

- [Michel, 2004] Michel, O. (2004). Cyberbotics Ltd. webotsTM: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):5.
- [Siciliano, 2009] Siciliano, B., editor (2009). *Robotics: modelling, planning and control*. Advanced textbooks in control and signal processing. Springer, London. OCLC: ocn144222188.
- [Siciliano and Khatib, 2008] Siciliano, B. and Khatib, O., editors (2008). *Springer Handbook of Robotics*. Springer, Berlin, Heidelberg.
- [Spong et al., 2020] Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2020). *Robot modeling and control*. Wiley, Hoboken, NJ, second edition edition.
- [Yoshikawa, 1983] Yoshikawa, T. (1983). Analysis and control of robot manipulators with redundancy.