

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
DEMEC - DEPARTAMENTO DE ENGENHARIA MECÂNICA

**AVALIAÇÃO DO IMPACTO DOS MODOS DE TRABALHO DE
MANIPULADORES ROBÓTICOS EM SÉRIE PARA RESOLUÇÃO
DE REDUNDÂNCIA**

Orientador:

Prof. Dr. João Vitor de Carvalho Fontes

Aluno:

João Vinicius Martin Cordeiro



SÃO CARLOS, SP
2023

JOÃO VINICIUS MARTIN CORDEIRO

**AVALIAÇÃO DO IMPACTO DOS MODOS DE TRABALHO DE
MANIPULADORES ROBÓTICOS EM SÉRIE PARA RESOLUÇÃO
DE REDUNDÂNCIA**

Projeto de Monografia apresentado ao DEMec -
Departamento de Engenharia Mecânica da Universidade
Federal de São Carlos, para obtenção do título de Bacharel
em Engenharia Mecânica.

Orientador: Prof. Dr. João Vitor de Carvalho Fontes

SÃO CARLOS, SP
2023



FUNDAÇÃO UNIVERSIDADE FEDERAL DE SÃO CARLOS

COORDENAÇÃO DO CURSO DE ENGENHARIA MECÂNICA - CCEMec/CCET

Rod. Washington Luís km 235 - SP-310, s/n - Bairro Monjolinho, São Carlos/SP, CEP 13565-905
Telefone: (16) 33519703 - <http://www.ufscar.br>

DP-TCC-FA nº 9/2023/CCEMec/CCET

Graduação: Defesa Pública de Trabalho de Conclusão de Curso

Folha Aprovação (GDP-TCC-FA)

FOLHA DE APROVAÇÃO

JOÃO VINICIUS MARTIN CORDEIRO

AVALIAÇÃO DO IMPACTO DOS MODOS DE TRABALHO DE MANIPULADORES ROBÓTICOS EM SÉRIE PARA RESOLUÇÃO DE REDUNDÂNCIA

Trabalho de Conclusão de Curso

Universidade Federal de São Carlos – Campus São Carlos

São Carlos, 23 de março de 2023

ASSINATURAS E CIÊNCIAS

Cargo/Função	Nome Completo
Orientador	João Vitor de Carvalho Fontes
Membro da Banca 1	Luís Antonio Oliveira Araujo
Membro da Banca 2	Sidney Bruce Shiki



Documento assinado eletronicamente por **Joao Vitor de Carvalho Fontes, Docente**, em 23/03/2023, às 18:17, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Luis Antonio Oliveira Araujo, Docente**, em 23/03/2023, às 18:17, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Sidney Bruce Shiki, Docente**, em 23/03/2023, às 18:17, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site <https://sei.ufscar.br/autenticacao>, informando o código verificador **0979800** e o código CRC **72D2B4C0**.

AGRADECIMENTOS

Primeiramente gostaria de dedicar este trabalho em memória do meu falecido pai Antônio Marcelo, que tanto me apoiou e me amou por toda sua vida. Dedico também em memória do meu falecido padrasto Sidnei, que sempre fez o possível para que eu pudesse estudar e estar presente em São Carlos.

Agradeço também à minha mãe Viviane Cristina por todo apoio durante toda minha vida e por fazer o possível para que eu tivesse sempre o melhor. Além disso, agradeço também ao meu irmão Luiz Felipe por me ajudar com caronas até São Carlos e também à minha namorada Rafaela Goulart, pelo apoio e ajuda nos anos finais da minha graduação.

Aos meus amigos Daniel Pio, Gabriel Chaib e Antonio Xavier por todos os bons momentos compartilhados durante toda a graduação, desde momentos para estudo até momentos de descontração e diversão.

Por fim, agradeço ao professor João Vitor de Carvalho Fontes por sempre me ajudar com qualquer dúvida da graduação, sendo muito paciente e prestativo, e pela grande orientação ao longo do desenvolvimento deste trabalho.

"A persistência é o caminho do êxito."

(Charles Chaplin)

RESUMO

Manipuladores robóticos estão cada vez mais presentes nas atividades industriais da atualidade, sendo utilizados para diversas tarefas que vão desde operações de soldagem até transporte e movimentação de cargas como um todo. Apesar de serem amplamente utilizados, muitas vezes sua aplicação não é feita da maneira mais otimizada e não levam em conta a melhor configuração para o manipulador, fazendo com que características como uso do espaço de trabalho, torque e consumo de energia não sejam utilizadas da melhor maneira possível. Para que um manipulador seja aplicado em uma determinada tarefa, é necessário que seja feita a sua modelagem matemática a fim de se obter as posições das juntas e dos elos do manipulador, bem como as suas velocidades e respectivas acelerações, levando em conta também as posições do efetuador final. A definição das posições das juntas é feita utilizando-se a cinemática inversa. O presente trabalho realiza essa modelagem para o caso de um manipulador serial do tipo RRR, no qual através dos cálculos feitos na cinemática inversa, encontram-se duas soluções para o problema, ou seja, duas configurações diferentes para as juntas do manipulador robótico, sendo elas a configuração com cotovelo para cima e a configuração com o cotovelo para baixo. Essas duas configurações, por sua vez, são chamadas de modos de trabalho de um manipulador robótico. Ao se atribuir o modo de trabalho do manipulador, é possível definir também as posições do efetuador final. Quando o efetuador final possui mais graus de liberdade do que a tarefa necessita, ocorre o fenômeno chamado de redundância no espaço de tarefas, assim é possível otimizar o desempenho utilizando os graus de liberdade excedentes. Esse excesso de graus de liberdade define um problema com infinitas soluções para as posições das juntas do robô. A definição dessas posições através de um processo de otimização é chamada de resolução da redundância, e pode ser feita através do método local. A resolução de redundância local é aquela que observa o gradiente de uma função custo a cada instante de tempo e decide as posições das juntas a partir disso. Com isso, esse trabalho tem como objetivo estudar os impactos das duas configurações de um manipulador robótico em série na resolução da redundância para otimizar seu desempenho cinemático e dinâmico. Além disso, através das comparações dos resultados entre os dois modos de trabalho do manipulador robótico, será possível visualizar qual o modo de trabalho mais eficaz para a localização cartesiana em que o manipulador se encontra.

Palavras-chave: Manipuladores robóticos. Resolução de redundância. Modos de trabalho. Soluções da cinemática inversa.

ABSTRACT

Robotic manipulators are increasingly present in the industrial activities of today, being used for various tasks ranging from welding operations to transportation and cargo handling as a whole. Despite being widely used, many times their application is not done in the most optimized way and they do not take into account the best configuration for the handler, causing characteristics such as workspace usage, torque and power consumption are not optimally utilized possible. Starting from the beginning, for a manipulator to be applied in a certain task, it is necessary to do its mathematical modeling in order to obtain the positions of the joints and links of the manipulator, their speeds and respective accelerations, also taking into account the end-effector positions. The definition of joint positions is done using inverse kinematics. The present work carries out this modeling for the case of a serial manipulator of the RRR type, in which, through the calculations made in the inverse kinematics, two solutions to the problem are found, that is, two different configurations for the joints of the robotic manipulator, being they are the elbow-up configuration and the elbow-down configuration. These two configurations, in turn, are called the working modes of a robotic manipulator. When assigning the manipulator's work mode, it is also possible to define the end-effector positions. When the end-effector has more degrees of freedom than the task needs, the phenomenon called redundancy occurs in the task space, so it is possible to optimize the performance using the excess degrees of freedom. This excess of degrees freedom defines a problem with infinite solutions for the joint positions of the robot. The definition of these positions through an optimization process is called redundancy resolution, and can be done through the local method. The local redundancy resolution is the one that observes the gradient of a cost function at each instant of time and decides joint positions from this. With this, this work aims to study the impacts of the two configurations of a robotic manipulator in series on the resolution redundancy to optimize its kinematic and dynamic performance. In addition, through comparisons of the results between the two working modes of the robotic manipulator, it will be possible to visualize which is the most effective working mode for the Cartesian location in which the manipulator is located..

Keywords: Robotic manipulators. Redundancy resolution. Working modes. Inverse kinematics solutions.

LISTA DE FIGURAS

Figura 1 – Manipulador serial de cadeia aberta KUKA, modelo KR 240 R3330. . .	16
Figura 2 – Representação das relações entre a cinemática direta e a cinemática inversa	18
Figura 3 – Manipulador do tipo RRR, com suas duas possíveis soluções.	19
Figura 4 – Estrutura de um manipulador robótico em série.	22
Figura 5 – Braço robótico industrial e sua estrutura antropomórfica.	22
Figura 6 – Modelo do manipulador planar serial com a configuração "cotovelo para cima".	24
Figura 7 – Espaço nulo dos manipuladores cinematicamente redundantes.	34
Figura 8 – Mínimos local e global de uma função não convexa em um determinado domínio.	35
Figura 9 – Trajetória realizada pelo efetuador final do manipulador.	41
Figura 10 – Simulação da trajetória do efetuador para o modo de trabalho do tipo "cotovelo para baixo".	42
Figura 11 – Simulação da trajetória do efetuador para o modo de trabalho do tipo "cotovelo para cima".	42
Figura 12 – Gráficos dos torques de cada junta para modo de trabalho do tipo "cotovelo para baixo" do Caso 1.	43
Figura 13 – Gráficos dos torques de cada junta para modo de trabalho do tipo "cotovelo para cima" do Caso 1.	43
Figura 14 – Média dos torques das juntas do manipulador para todos os casos e situações.	45
Figura 15 – Média dos torques das juntas do manipulador para o modo de trabalho do tipo "cotovelo para baixo".	46
Figura 16 – Média dos torques das juntas do manipulador para o modo de trabalho do tipo "cotovelo para cima".	46
Figura 17 – Torque das juntas do manipulador do caso 1a.	53
Figura 18 – Torque das juntas do manipulador do caso 1b.	54
Figura 19 – Torque das juntas do manipulador do caso 1c.	55
Figura 20 – Torque das juntas do manipulador do caso 1d.	56
Figura 21 – Torque das juntas do manipulador do caso 2a.	57
Figura 22 – Torque das juntas do manipulador do caso 2b.	58
Figura 23 – Torque das juntas do manipulador do caso 2c.	59
Figura 24 – Torque das juntas do manipulador do caso 2d.	60

LISTA DE TABELAS

Tabela 1 – Parâmetros DH para o manipulador serial.	24
Tabela 2 – Parâmetros utilizados para modelagem do manipulador RRR.	40
Tabela 3 – Dados da trajetória do efetuador final nas simulações.	41
Tabela 4 – Valores da função E e da norma infinita de cada junta $\ \tau\ _\infty$ para os dois casos simulados e suas derivações.	44

LISTA DE ABREVIATURAS E SIGLAS

GDL	Grau de liberdade
DH	Denavit-Hartenberg

LISTA DE SÍMBOLOS

m	Mobilidade do manipulador
l	Elo do manipulador
l_1, l_2, l_3	Comprimento dos elos
m_1, m_2, m_3	Massa dos elos
I_1, I_2, I_3	Momento de Inércia dos elos
b	Espessura dos elos
R	Grau de redundância
d	Dimensão do espaço das tarefas
j	Juntas do manipulador
$\theta_1, \theta_2, \theta_3$	Ângulos das juntas
\mathbf{q}	Vetor dos ângulos das juntas
ϕ	Orientação do efetuador final
x_1, x_2, x_3	Posições das juntas no eixo x
y_1, y_2, y_3	Posições das juntas no eixo y
x_e, y_e, z_e	Posição do efetuador final
\mathbf{p}_e	Vetor da posição do efetuador final
A	Matriz de Transformação Homogênea
T	Matriz de Transformação
J	Matriz Jacobiana
\dot{x}_e, \dot{y}_e	Velocidade linear do efetuador final
$\dot{\phi}$	Velocidade angular do efetuador final
$\dot{\mathbf{x}}$	Vetor das velocidades do efetuador final
$\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3$	Velocidade angular das juntas
$\dot{\mathbf{q}}$	Vetor das velocidades angulares das juntas

J^\dagger	Matriz Jacobiana pseudo-inversa
\dot{J}	Derivada da matriz Jacobiana
\ddot{x}_e, \ddot{y}_e	Aceleração linear do efetuador final
$\ddot{\phi}$	Aceleração angular do efetuador final
$\ddot{\mathbf{x}}$	Vetor das acelerações do efetuador final
$\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3$	Aceleração angular das juntas
$\ddot{\mathbf{q}}$	Vetor das acelerações angulares das juntas
τ_1, τ_2, τ_3	Torque das juntas
$\boldsymbol{\tau}$	Vetor dos torques das juntas
\mathcal{L}	Termo Lagrangiano
\mathcal{K}	Termo da energia cinética
\mathcal{U}	Termo da energia potencial
M	Matriz de Inércia
\mathbf{c}	Vetor de Coriolis
\mathbf{g}	Vetor das forças gravitacionais
p	Posição do efetuador final na trajetória
\dot{p}	Velocidade do efetuador final na trajetória
\ddot{p}	Aceleração do efetuador final na trajetória
t_i	Tempo inicial da trajetória
t_f	Tempo final da trajetória
$a_0 \dots a_5$	Coefficientes do polinômio de quinto grau
λ	Multiplicador de Lagrange
J_M^\dagger	Matriz Jacobiana ponderada pela matriz de Inércia
$J_{M^2}^\dagger$	Matriz Jacobiana ponderada pela matriz de Inércia ao quadrado
$\ \boldsymbol{\tau}\ _\infty$	Torque máximo da junta
A, B, C, D	Nomenclatura das trajetórias no quadrado

1, 2, 3, 4	Nomenclatura dos vértices do quadrado da trajetória
ϕ_{1A}	Orientação inicial do efetuador final no vértice 1 da trajetória A
E	Média dos torques das juntas do manipulador

SUMÁRIO

1 – INTRODUÇÃO	16
1.1 Objetivos	19
1.2 Estrutura do texto	20
2 – FUNDAMENTAÇÃO TEÓRICA	21
2.1 Características gerais dos manipuladores robóticos	21
2.1.1 Mobilidade dos manipuladores robóticos seriais	22
2.2 Modelagem de um manipulador planar serial	23
2.2.1 Parâmetros de Denavit-Hartenberg	23
2.2.2 Cinemática direta	24
2.2.3 Cinemática inversa	25
2.2.4 Matriz Jacobiana	27
2.2.5 Cinemática direta diferencial	28
2.2.6 Cinemática inversa diferencial	29
2.2.7 Dinâmica	29
2.3 Planejamento da trajetória do efetuador final	30
2.4 Redundância de manipuladores robóticos	32
2.4.1 Redundância cinemática da tarefa	32
2.5 Resolução da redundância	34
2.5.1 Otimização	35
2.6 Considerações finais do capítulo	36
3 – RESOLUÇÃO DA REDUNDÂNCIA	37
3.1 Resolução da redundância pelo método local	37
3.1.1 Caso 1 - Resolução local ponderada	37
3.1.2 Caso 2 - Resolução local não ponderada	38
3.2 Método de comparação dos resultados	39
3.3 Considerações finais do capítulo	39
4 – RESULTADOS	40
4.1 Características do manipulador robótico	40
4.2 Trajetória do efetuador final	41
4.3 Resultados obtidos através das simulações	41
4.4 Comparações dos resultados obtidos	44
4.5 Considerações finais do capítulo	47
5 – CONCLUSÃO	48

5.1	Trabalhos futuros	49
REFERÊNCIAS	50
Apêndices		52
APÊNDICE A	– Gráficos dos torques das juntas do manipulador para o caso 1 .	53
APÊNDICE B	– Gráficos dos torques das juntas do manipulador para o caso 2 .	57
APÊNDICE C	– Exemplo de código utilizado para o cálculo da cinemática inversa	61
APÊNDICE D	– Exemplo de código utilizado na resolução da redundância do caso 1	62
APÊNDICE E	– Exemplo de código utilizado na resolução da redundância do caso 2	71

1 INTRODUÇÃO

Um manipulador robótico pode ser definido como um dispositivo mecânico controlado por *software*, cuja finalidade é específica para diversos processos automatizados. Além disso, manipuladores robóticos podem utilizar sensores para auxiliar na orientação e movimentação de suas partes em diversas ocasiões preestabelecidas (LOPES, 2001).

Segundo Simplício, Lima *et al.* (2016), o uso de manipuladores robóticos industriais aumenta de forma significativa a cada ano devido à necessidade de produzir cada vez mais produtos em larga escala. Esses manipuladores são capazes de realizar trabalhos que muitas vezes a força humana não seria capaz, ou que levaria um tempo elevado para ser realizado manualmente. A implantação dessas máquinas tem sido cada vez mais frequentes em indústrias que visam automatizar seus processos de fabricação. Além disso, o uso de manipuladores robóticos na linha de produção traz inúmeras vantagens durante a fabricação da peça, como a padronização do produto, agilidade na produção, maior precisão dimensional e melhoria na qualidade final do produto.

Os robôs manipuladores podem ser classificados como aqueles que possuem uma base fixa e sua estrutura é composta por elos rígidos interligados por meio de articulações motorizadas, também chamadas de juntas ativas. Um manipulador é dito serial quando é composto por uma cadeia cinemática aberta, ou seja, os elos se conectam por apenas um caminho entre a base e o efetuador final, sendo este o elemento responsável por executar a tarefa (SICILIANO *et al.*, 2009). Na Figura 1 tem-se um exemplo de um manipulador serial de cadeia aberta.

Figura 1 – Manipulador serial de cadeia aberta KUKA, modelo KR 240 R3330.



Fonte: KUKA Roboter do Brasil (2020)

Na Figura 1 é mostrado o modelo KR 240 R3330, fabricado pela empresa KUKA

Industrial Robots. O modelo possui uma capacidade de carga nominal de até 240 kg e é utilizado pela indústria automobilística em atividades de montagem e manuseio de peças e equipamentos (KUKA Roboter do Brasil, 2020). Além disso, para entender melhor a aplicação dos manipuladores robóticos, é necessário se ter conhecimento sobre os seus graus de liberdade.

Pode-se definir os graus de liberdade (GDL) de um manipulador como sendo uma característica que determina a sua mobilidade e um fator importante na execução de uma tarefa. Esses graus são definidos pela quantidade de juntas presentes em um manipulador serial. Um manipulador que apresenta mais graus de liberdade do que o necessário para uma determinada tarefa é chamado de manipulador redundante (SICILIANO *et al.*, 2009).

Para Duarte (2001), é possível definir redundância como sendo um sistema que pode apresentar várias configurações para uma determinada posição do órgão terminal no espaço cartesiano. Já do ponto de vista mecânico, a redundância pode ser do tipo cinemática ou de atuação. De acordo com Weihmann (2011), a redundância cinemática pode ser utilizada para evitar que o manipulador atinja posições singulares em que ocorre a perda de mobilidade e aumento excessivo do torque nas juntas.

Segundo Fontes e Silva (2017), a redundância no espaço de tarefas é um caso de redundância cinemática, e nela o manipulador possui mais graus de liberdade do que o necessário para realizar uma tarefa. Esse excesso de graus de liberdade permite que o robô mantenha a mesma posição do efetuador final para variadas posições das juntas do manipulador. O processo de definir o posicionamento das juntas entre as infinitas possibilidades, enquanto o efetuador final executa uma tarefa, é chamado de resolução da redundância.

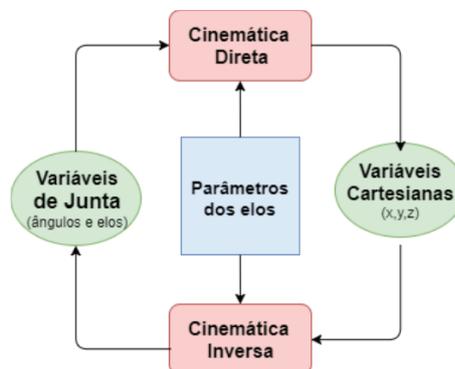
Além disso, a resolução da redundância é capaz de otimizar um manipulador através de critérios de performance tais como: evitar os torques limites dos atuadores, minimização da velocidade, aceleração, torque das juntas e energia (NENCHEV, 1989). De acordo com Ahuactzin e Gupta (1999), a resolução da redundância pode ser feita através da maneira local, que é aquela que executa a otimização observando o gradiente de uma função custo a cada instante de tempo.

A otimização do manipulador robótico é feita utilizando-se como base um modelo obtido através de cálculos cinemáticos e dinâmicos, para que sejam obtidas as posições e parâmetros necessários para determinar a melhor configuração do robô. A análise cinemática compreende tanto o estudo da posição (cinemática de posição) como o da velocidade (cinemática diferencial) de mecanismos.

De acordo com Weihmann (2011), a cinemática de posição pode ser dividida em cinemática direta e cinemática inversa. Na cinemática direta de posição, a posição linear ou angular das juntas ativas é conhecida e o objetivo é obter a posição de um elo ou de um ponto específico do mecanismo. Já na cinemática inversa, a posição (ou velocidade) de um ponto específico do mecanismo é conhecida e o objetivo é encontrar a posição

(ou velocidade) das juntas atuadas (ASADA; SLOTINE, 1991). A relação entre as duas cinemáticas de posição é demonstrada na Figura 2.

Figura 2 – Representação das relações entre a cinemática direta e a cinemática inversa



Fonte: Oliveira (2018)

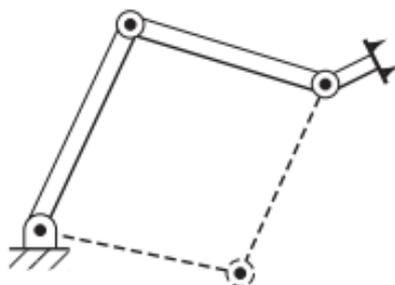
Quando se analisa seus movimentos por meio da cinemática, a complexidade geométrica aumenta bastante se o manipulador robótico apresenta muitos graus de liberdade, principalmente se a análise é baseada na cinemática inversa de robôs manipuladores (NUNES, 2016).

A cinemática inversa apresenta grandes desafios devido as equações serem não lineares e poderem ou não apresentar solução ou múltiplas soluções, pois é comum a presença de redundâncias nesta análise para manipuladores com muitos graus de liberdade (CRAIG, 2012). Por esse motivo, segundo Oliveira (2018), a estrutura matemática utilizada na modelagem de um manipulador é frequentemente requerida para programação de braços robóticos, uma vez que existe singularidades e infinitas possibilidades do efetuador final atingir determinada posição e orientação desejada.

De acordo com Craig (2012), um braço planar com três juntas rotacionais tem um grande espaço de trabalho dentro no plano, porque qualquer posição dentro do seu espaço de trabalho pode ser alcançada em qualquer orientação. Abaixo, a Figura 4 mostra um braço planar de três elos (RRR) com o efetuador em certa posição e orientação, onde as linhas pontilhadas indicam uma segunda configuração possível na qual as mesmas posição e orientação do efetuador são alcançadas.

O modo de trabalho (*working mode*) está relacionado com a cinemática inversa de posição do manipulador (WENGER; CHABLAT, 1998). De acordo com Weihmann (2011), quando conhecida a posição e orientação do efetuador, cada uma das soluções possíveis da cinemática inversa define um dos modos de trabalho do manipulador, sendo que os manipuladores seriais podem apresentar diferentes modos de trabalho. As expressões “cotovelo para cima” e “cotovelo para baixo” ou “ombro para cima” e “ombro para baixo” são também utilizadas para diferenciar os modos de trabalho de um manipulador (SCIAVICCO; SICILIANO, 2001). Vale destacar que neste trabalho o termo “modo de funcionamento”

Figura 3 – Manipulador do tipo RRR, com suas duas possíveis soluções.



Fonte: Craig (2012)

também será utilizado e terá o mesmo significado que o termo “modo de trabalho” definido anteriormente.

Segundo Weihmann (2011), na presença de redundância cinemática, o termo modo de trabalho não é estritamente correto pois, para a mesma posição e orientação do efetuador, o manipulador pode assumir infinitas configurações. Portanto, neste estudo assume-se que, a partir de uma posição inicial, todas as configurações que o manipulador consegue atingir sem passar por uma singularidade, mantendo a posição e a orientação do efetuador constantes, pertencem ao mesmo modo de trabalho.

Com isso, foi implementado no presente trabalho o modelo de um manipulador redundante planar serial de três GDL no *software* MATLAB considerando as duas configurações possíveis e procurou realizar nesse ambiente a resolução da redundância através do método local, considerando os efeitos da posição inicial do manipulador. Os resultados obtidos foram comparados entre variações dos casos propostos com a finalidade de analisar os impactos causados no desempenho dinâmico do robô.

1.1 Objetivos

O objetivo geral da presente monografia é estudar os modos de trabalho de um manipulador robótico serial, avaliando os possíveis impactos para a resolução de redundâncias e otimização das características gerais do robô. Dessa forma, será possível definir qual o modo de trabalho mais eficaz, com menor gasto de energia e que promova uma maior sustentabilidade em sua aplicação. Além disso, para o estudo dos modos de trabalho do manipulador robótico serão utilizadas diversas simulações feitas no *software* MATLAB para obtenção de resultados de trajetórias e comparação entre elas por meio de gráficos e ilustrações.

Para que esse objetivo seja atingido, os seguintes objetivos específicos foram propostos:

- Modelar os dois modos de trabalho possíveis para o manipulador robótico serial do

tipo RRR;

- Simular os dois modos de trabalho para o primeiro e segundo quadrante do espaço cartesiano, a fim de verificar a metodologia para um plano de trabalho espelhado;
- Calcular as variáveis cinemáticas e dinâmicas dos modos de trabalho por meio de simulações feitas no *software* MATLAB;
- Resolver a redundância dos dois modos de trabalho do manipulador robótico serial utilizando o método de resolução local;
- Comparar os valores dos torques obtidos para os dois modos de trabalho, para que seja possível definir o melhor modo de trabalho através dos resultados de eficácia do manipulador estudado.

1.2 Estrutura do texto

Esta monografia está organizada da seguinte forma:

- O capítulo 2 apresenta a fundamentação teórica que define o manipulador estudado nesse trabalho, com os conceitos de mobilidade, modelagem do manipulador estudado, cinemática direta, cinemática inversa e as duas possíveis soluções para um manipulador do tipo RRR, cinemática diferencial, dinâmica e planejamento da trajetória do efetuador final, redundância cinemática e resolução da redundância através do método local;
- O capítulo 3 descreve o método de resolução da redundância utilizado para os dois modos de trabalho do manipulador RRR e apresenta as funções custo e condições de contorno;
- O capítulo 4 apresenta os resultados obtidos com a resolução da redundância para as duas configurações do manipulador e que também foram simulados para o plano de trabalho espelhado, faz a comparação entre os resultados obtidos e as discussões relacionadas;
- O capítulo 5 contém as conclusões do trabalho e a indicação de trabalhos futuros;

2 FUNDAMENTAÇÃO TEÓRICA

Para realizar o projeto de um manipulador robótico serial, é necessário ter conhecimento sobre as suas características e sobre os conceitos envolvidos na modelagem do robô, para que assim seja possível a elaboração de um modelo matemático que possa ser implementado em um ambiente de simulação no qual serão obtidos os resultados propostos para o projeto. Desse modo, serão apresentados nesse capítulo os principais conceitos e considerações que foram utilizados no desenvolvimento do trabalho, sendo eles as definições de manipuladores robóticos, sua modelagem matemática para o caso de um manipulador do tipo RRR, o cálculo da cinemática direta, cálculo da cinemática inversa e suas duas soluções possíveis para as configurações do robô, cálculo das cinemáticas diferenciais, cálculo das equações dinâmicas e planejamento da trajetória, abordagem da redundância cinemática do manipulador, resolução da redundância e otimização.

2.1 Características gerais dos manipuladores robóticos

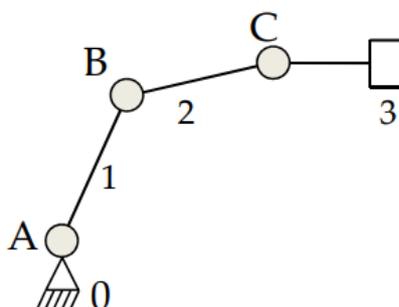
De modo geral, a classificação dos robôs é definida através da sua estrutura mecânica. Segundo Siciliano *et al.* (2009), eles podem ser divididos em duas categorias principais: aqueles que possuem uma base fixa, sendo chamados de robôs manipuladores e aqueles que possuem uma base móvel, chamados de robôs móveis.

Segundo Simplício, Lima *et al.* (2016), um manipulador robótico pode ser definido como um dispositivo mecânico controlado por *software*, cuja finalidade é específica para diversos processos automatizados. Atualmente, os manipuladores industriais são máquinas bastante avançadas e sofisticadas, voltadas para processos de automação específicos.

De acordo com Oliveira (2022), os manipuladores robóticos são formados por uma sequência de elos rígidos que são conectados através de articulações motorizadas, ou juntas. Essa sequência é chamada de cadeia. Ao final de uma cadeia de elos está localizado o efetuador final que é responsável por realizar a tarefa requerida ao robô. Os efetuadores finais podem ser: soldadores, furadeiras, sopradores de tinta, parafusadoras etc. Quando o manipulador possui um único caminho de elos entre a base e o efetuador final, ele é chamado de manipulador serial e a sua cadeia é aberta. No caso de existir mais de um caminho, esse robô é classificado como um manipulador paralelo e tem cadeia fechada (SICILIANO *et al.*, 2009). Um exemplo de um manipulador do tipo serial pode ser visto na Figura 4.

Além disso, esses manipuladores são geralmente antropomórficos, ou seja, semelhantes a um braço humano. Quando analisada a estrutura de um manipulador robótico, é possível identificar o tronco, braço, antebraço, mão, e suas juntas flexíveis que podem ser definidas como ombro, cotovelo e pulso (SIMPLÍCIO; LIMA *et al.*, 2016). Um braço

Figura 4 – Estrutura de um manipulador robótico em série.



Fonte: Weihmann (2011)

Figura 5 – Braço robótico industrial e sua estrutura antropomórfica.



Fonte: (SIMPLÍCIO; LIMA *et al.*, 2016)

robótico industrial e suas denominações antropomórficas estão representados na Figura 5.

Uma vez que o trabalho tem como foco os manipuladores em série, os conceitos apresentados são sobre esse tipo de manipulador, principalmente nos manipuladores seriais do tipo RRR.

2.1.1 Mobilidade dos manipuladores robóticos seriais

A mobilidade dos manipuladores seriais está diretamente associada ao número de juntas da sua estrutura. Essas juntas são divididas em dois tipos: prismática (P) e de revolução (R). A primeira permite a movimentação translacional entre dois elos, e a segunda o movimento rotacional. No caso dos robôs seriais, todas as juntas devem ser atuadas para que eles não percam sua sustentação. Sendo assim, é possível considerar no caso dos manipuladores seriais que a mobilidade é igual o número de juntas (WEIHMANN, 2011).

Assim, o cálculo da mobilidade (m) de um robô serial pode ser definido como a

soma do grau de liberdade (gdl_i) permitido por cada junta i do manipulador, sendo j o número total de juntas, conforme apresentado na Equação 1.

$$m = \sum_{i=1}^j gdl_i \quad (1)$$

O conceito de mobilidade é fundamental para definir a redundância de um manipulador robótico, a qual será abordada após a modelagem do manipulador RRR estudado neste trabalho.

2.2 Modelagem de um manipulador planar serial

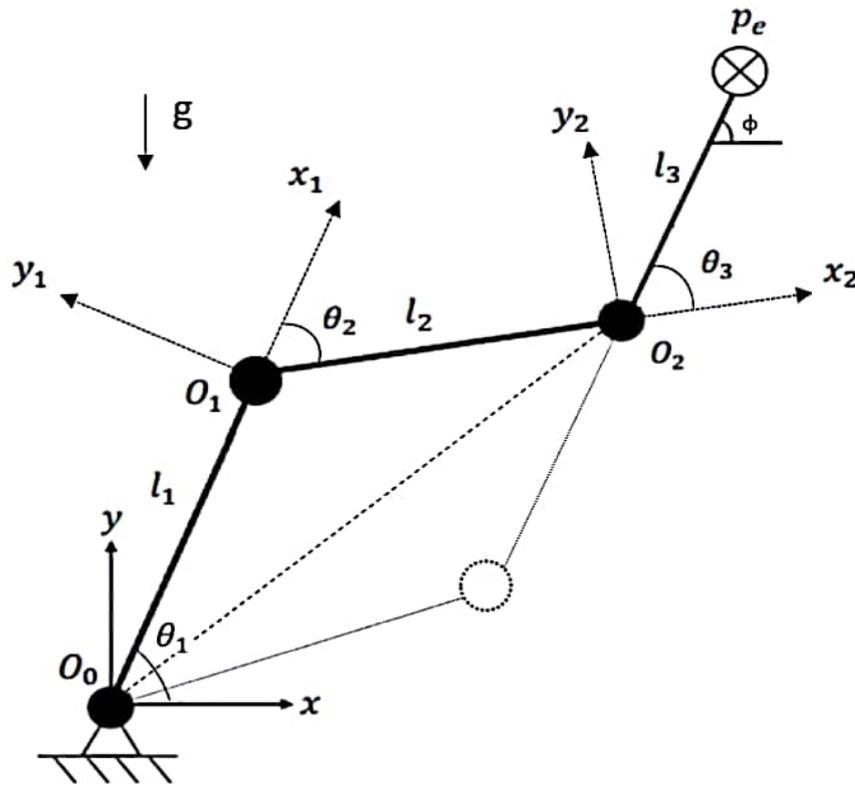
Para a modelagem do manipulador planar serial utilizado no projeto foi tomado como base o manipulador mostrado na Figura 6, que representa a configuração (modo de trabalho) conhecida como "cotovelo para cima". Esse manipulador possui três juntas de revolução (RRR) e θ_1 , θ_2 e θ_3 são os ângulos dessas juntas em relação ao elo anterior que são representados pelo vetor $\mathbf{q} = [\theta_1, \theta_2, \theta_3]$. Os comprimentos dos elos são representados por l_1 , l_2 e l_3 . O ângulo ϕ é a orientação do efetuador final em relação ao sistema de referência fixo (x, y) . A posição do efetuador final é representado pelo vetor $\mathbf{p}_e = [x_e, y_e]$. O manipulador modelado na Figura 6 apresenta também, em linhas tracejadas, o outro modo de trabalho possível. Além disso, na Figura 6 a aceleração da gravidade é representada pelo vetor \mathbf{g} com o seu sentido (sendo positiva contrária ao referencial y). O manipulador pode se movimentar sem restrições no plano (x, y) , sendo limitado apenas pelo plano de trabalho determinado por sua constituição física.

2.2.1 Parâmetros de Denavit-Hartenberg

Os parâmetros de Denavit-Hartenberg (parâmetros DH) são uma convenção proposta por Denavit e Hartenberg (1955) para padronizar coordenadas de sistemas de referências, sendo amplamente utilizados na modelagem de manipuladores robóticos. Os parâmetros são:

- θ_i - ângulo de rotação ao redor do eixo z_{i-1} para deixar o eixo x_{i-1} paralelo com o eixo x_i .
- d_i - deslocamento da origem O_{i-1} ao longo de z_{i-1} até o ponto de intersecção dos eixos z_{i-1} e x_i .
- a_i - deslocamento da origem O_i ao longo de x_i até o ponto de intersecção dos eixos x_i e z_{i-1} .
- ϕ_i - ângulo de rotação ao redor do eixo x_i para deixar o eixo z_{i-1} paralelo com o eixo z_i .

Figura 6 – Modelo do manipulador planar serial com a configuração "cotovelo para cima".



Fonte: Elaborado pelo autor (2023).

Para o caso do manipulador apresentado na Figura 6, os parâmetros DH são os apresentados na Tabela 1.

Tabela 1 – Parâmetros DH para o manipulador serial.

i	θ_i	d_i	a_i	ϕ_i
1	θ_1	0	l_1	0
2	θ_2	0	l_2	0
3	θ_3	0	l_3	0

Fonte: Elaborado pelo autor (2023).

2.2.2 Cinemática direta

De acordo com Oliveira (2022), a cinemática direta é utilizada para determinar a posição do efetuador final tendo como entrada as posições angulares das juntas do manipulador. Para isso, é necessário o cálculo da Matriz de Transformação Homogênea (A). Esta matriz expressa a posição do sistema de referência do elo analisado (i) em relação ao anterior ($i - 1$).

A Equação 2 é a expressão matricial de A , na qual c e s representam as funções cosseno e seno, respectivamente.

$$A_i^{i-1} = \begin{bmatrix} c(\theta_i) & -s(\theta_i)c(\phi_i) & s(\theta_i)s(\phi_i) & a_i c(\theta_i) \\ s(\theta_i) & c(\theta_i)c(\phi_i) & c(\theta_i)s(\phi_i) & a_i s(\theta_i) \\ 0 & s(\phi_i) & c(\phi_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Na sequência, é necessário realizar a obtenção da Matriz de Transformação (T). Essa matriz é utilizada para expressar a posição do efetuador final em relação ao sistema de referência inercial (x, y), podendo ser definida de forma simplificada pela Equação 3. A parte da matriz composta pelos elementos R_{ij} representa a rotação e x_e, y_e, z_e a posição do efetuador final.

$$T_3^0 = A_1^0 A_2^1 A_3^2 = \begin{bmatrix} R_{11} & R_{12} & R_{13} & x_e \\ R_{21} & R_{22} & R_{23} & y_e \\ R_{31} & R_{32} & R_{33} & z_e \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Simplificando para o caso do manipulador planar RRR da Figura 6, a matriz (T) pode ser representada conforme a Equação 4.

$$T_3^0 = \begin{bmatrix} R_{11} & R_{12} & x_e \\ R_{21} & R_{22} & y_e \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

A posição do efetuador final \mathbf{p}_e está representada pela Equação 5.

$$\mathbf{p}_e = \begin{bmatrix} x_e \\ y_e \end{bmatrix} = \begin{bmatrix} l_1 c(\theta_1) + l_2 c(\theta_1 + \theta_2) + l_3 c(\theta_1 + \theta_2 + \theta_3) \\ l_1 s(\theta_1) + l_2 s(\theta_1 + \theta_2) + l_3 s(\theta_1 + \theta_2 + \theta_3) \end{bmatrix} \quad (5)$$

O ângulo ϕ pode ser calculado da seguinte forma:

$$\phi = \theta_1 + \theta_2 + \theta_3 \quad (6)$$

2.2.3 Cinemática inversa

A cinemática inversa é o cálculo feito para determinar as posições das juntas do manipulador robótico em função da posição \mathbf{p}_e e orientação ϕ do efetuador final. Além disso, a cinemática inversa é essencial para modelar um manipulador que movimenta-se em função de uma trajetória determinada para o efetuador final. De acordo com Siciliano *et al.* (2009), a cinemática inversa possui um cálculo mais complexo em relação à cinemática direta, devido às seguintes situações:

- As equações são, em geral, não lineares e nem sempre podem ser resolvidas com uma forma fechada;
- Não existir uma solução, dependendo da posição do efetuador final desejada e da estrutura do manipulador;
- Existirem múltiplas soluções;
- Existirem infinitas soluções;

Para o presente trabalho, será utilizado a primeira solução geométrica para resolução da cinemática inversa em manipuladores seriais RRR apresentada por Siciliano *et al.* (2009). Com isso, a partir da Equação 7, é possível determinar o ângulo das juntas do manipulador. Considerando a posição da junta O_2 , tem-se a seguinte relação:

$$\mathbf{p}_{O_2} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} l_1 c(\theta_1) + l_2 c(\theta_1 + \theta_2) \\ l_1 s(\theta_1) + l_2 s(\theta_1 + \theta_2) \end{bmatrix} \quad (7)$$

A partir dela, é possível obter as variáveis x_2 e y_2 em função de x_e , y_e e ϕ :

$$x_2 = x_e - l_3 \cos(\phi) \quad (8)$$

$$y_2 = y_e - l_3 \sin(\phi) \quad (9)$$

Aplicando a Lei dos Cossenos no triângulo formado pelos segmentos l_1 , l_2 e pelo segmento entre O_0 e O_2 , tem-se a seguinte relação:

$$x_2^2 + y_2^2 = l_1^2 + l_2^2 - 2l_1 l_2 \cos(\theta_2) \quad (10)$$

Isolando $\cos(\theta_2)$ da equação acima, tem-se que:

$$\cos(\theta_2) = \frac{x_2^2 + y_2^2 - l_1^2 - l_2^2}{2l_1 l_2} \quad (11)$$

Para garantir a existência de uma solução, é necessário que $-1 \leq \cos(\theta_2) \leq 1$, caso contrário o resultado estará fora do espaço de trabalho do manipulador. Sendo assim, é possível determinar o valor de $\sin(\theta_2)$:

$$\pm \sin(\theta_2) = \sqrt{1 - \cos^2(\theta_2)} \quad (12)$$

Vale destacar que na Equação 12 o sinal positivo é utilizado para o modo de trabalho do tipo "cotovelo para baixo", enquanto o sinal negativo é utilizado para modo de trabalho do tipo "cotovelo para cima". A partir disso, é possível calcular o ângulo θ_2 :

$$\theta_2 = \tan^{-1} \left(\frac{\sin(\theta_2)}{\cos(\theta_2)} \right) \quad (13)$$

Pela Equação 7, utilizando a relação trigonométrica $\cos(\theta_1 + \theta_2) = \cos(\theta_1)\cos(\theta_2) - \text{sen}(\theta_1)\text{sen}(\theta_2)$, obtêm-se:

$$\cos(\theta_1) = \frac{(l_1 + l_2\cos(\theta_2))x_2 + (l_2\text{sen}(\theta_2))y_2}{x_2^2 + y_2^2} \quad (14)$$

$$\text{sen}(\theta_1) = \frac{(l_1 + l_2\cos(\theta_2))y_2 - (l_2\text{sen}(\theta_2))x_2}{x_2^2 + y_2^2} \quad (15)$$

Com isso, é possível determinar também o valor de θ_1 :

$$\theta_1 = \tan^{-1} \left(\frac{\text{sen}(\theta_1)}{\cos(\theta_1)} \right) \quad (16)$$

Por fim, o ângulo θ_3 pode ser determinado pela relação:

$$\theta_3 = \phi - \theta_2 - \theta_1 \quad (17)$$

2.2.4 Matriz Jacobiana

De acordo com Oliveira (2022), a Matriz Jacobiana (J) relaciona a velocidade do efetuador final com a velocidade das juntas. A matriz pode ser particionada em vetores (3x1) de velocidade linear (Jp_i) e angular (Jo_i), conforme a Equação 18, sendo n o número de juntas do manipulador.

$$J_{(6 \times n)} = \begin{bmatrix} Jp_1 & \cdots & Jp_n \\ Jo_1 & \cdots & Jo_n \end{bmatrix} \quad (18)$$

Como apresentado em Siciliano *et al.* (2009), para o caso do manipulador com juntas rotativas a matriz Jacobiana pode ser construída através da expressão da Equação 19.

$$\begin{bmatrix} Jp_i \\ Jo_i \end{bmatrix} = \begin{bmatrix} z_{i-1} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix} \quad (19)$$

Os vetores z_{i-1} , p_e e p_{i-1} podem ser definidos da seguinte forma:

- z_{i-1} é a terceira coluna da matriz de rotação $R_{z_{i-1}}^0$:

$$R_{z_i}^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\text{sen}(\theta_i) & 0 \\ \text{sen}(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (20)$$

Considerando que $R_{z_0}^0 = I_{3 \times 3}$.

- $p_e = [x_e, y_e, z_e]$, com $z_e = 0$, é o vetor posição do efetuador final obtido da matriz T_n^0 , conforme apresentado na Equação 5;

- $\mathbf{p}_{i-1} = [x_{i-1}, y_{i-1}, z_{i-1}]$, com $z_{i-1} = 0$, é o vetor posição das juntas obtido da matriz T_{i-1}^0 ;

Considerando que $T_{i-1}^0 = I_{4 \times 4}$.

Para o caso de estudo desse trabalho, a matriz Jacobiana para o manipulador planar RRR da Figura 6 é:

$$J_{(6 \times 3)} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (21)$$

Sendo $s_1 = \text{sen}(\theta_1)$, $s_{12} = \text{sen}(\theta_1 + \theta_2)$ e $s_{123} = \text{sen}(\theta_1 + \theta_2 + \theta_3)$, o que ocorre também para a função cosseno.

A matriz J pode ser simplificada eliminando as linhas nulas, resultando na uma matriz (3 x 3) no caso de uma manipulador não redundante:

$$J_{(3 \times 3)} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \\ 1 & 1 & 1 \end{bmatrix} \quad (22)$$

Para o estudo desse trabalho de uma manipulador redundante, a matriz J utilizada que relaciona apenas os GDL lineares com as velocidades das juntas é:

$$J_{(2 \times 3)} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{bmatrix} \quad (23)$$

2.2.5 Cinemática direta diferencial

Utilizando a matriz Jacobiana, é possível definir a cinemática direta diferencial para manipuladores não redundantes pela a relação da Equação 24, onde $\dot{\mathbf{x}} = [\dot{x}_e, \dot{y}_e, \dot{\phi}]$ é o vetor de velocidade do efetuador final e $\dot{\mathbf{q}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]$ é o vetor de velocidade das juntas de rotação. Para o caso de manipulador redundante desse trabalho $\dot{\mathbf{x}} = [\dot{x}_e, \dot{y}_e]$.

$$\dot{\mathbf{x}} = J\dot{\mathbf{q}} \quad (24)$$

2.2.6 Cinemática inversa diferencial

A cinemática inversa diferencial de um manipulador não redundante é definida pela Equação 25.

$$\dot{\mathbf{q}} = J^{-1}\dot{\mathbf{x}} \quad (25)$$

Conforme o trabalho de Oliveira (2022), essa equação só é possível de ser resolvida quando a matriz Jacobiana é quadrada de dimensão $n \times n$ e possui $\det(J) \neq 0$. Isso ocorre quando o manipulador é não redundante. No caso do manipulador planar RRR com redundância, a Jacobiana tem dimensão 2×3 , sendo assim impossível calcular a sua inversa. Desse modo, é preciso utilizar uma matriz pseudo-inversa de J . Esse método é utilizado quando sistemas de equações não possuem uma única resposta, como é o caso para os manipuladores redundantes. Um dos métodos para calcular a matriz pseudo-inversa será apresentada no Capítulo 3.

A posição das juntas \mathbf{q} pode ser obtido através da integração das velocidades $\dot{\mathbf{q}}$, sendo $\mathbf{q}(0)$ a posição inicial das juntas no intervalo de tempo, conforme a Equação 26.

$$\mathbf{q}(t) = \int_0^t \dot{\mathbf{q}}(\varsigma)d(\varsigma) + \mathbf{q}(0) \quad (26)$$

A Equação 26 será resolvida utilizando o método de integração simples nos *scripts* das simulações implementados no *software* MATLAB.

2.2.7 Dinâmica

Utilizando as equações cinemáticas obtidas anteriormente neste trabalho, é possível determinar as equações da dinâmica direta e inversa de um manipulador não redundante, respectivamente, pelas Equações 27 e 28, sendo $\ddot{\mathbf{x}} = [\ddot{x}_e, \ddot{y}_e, \ddot{\alpha}]$ a aceleração o efetuador final e $\ddot{\mathbf{q}} = [\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3]$ a aceleração das juntas do manipulador.

$$\ddot{\mathbf{x}} = J\ddot{\mathbf{q}} + \dot{J}\dot{\mathbf{q}} \quad (27)$$

$$\ddot{\mathbf{q}} = J^{-1}(\ddot{\mathbf{x}} - \dot{J}\dot{\mathbf{q}}) \quad (28)$$

Os métodos utilizados para obter $\ddot{\mathbf{q}}$ para os manipuladores redundantes serão apresentados no Capítulo 3, através dos métodos de resolução da redundância.

De acordo com o trabalho de Siciliano *et al.* (2009), as equações dinâmicas podem ser obtidas pela formulação de Lagrange para um sistema conservativo, conforme a Equação 29 na qual $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3]$ é o vetor dos torques, \mathbf{q} e $\dot{\mathbf{q}}$ são, respectivamente, os vetores de posição e velocidade angular dos elos.

$$\frac{\partial}{\partial t} \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \boldsymbol{\tau} \quad (29)$$

\mathcal{L} é o termo Lagrangiano dado pela diferença:

$$\mathcal{L} = \mathcal{K} - \mathcal{U} \quad (30)$$

\mathcal{K} é o termo de energia cinética para um corpo rígido para os elos do manipulador, definido como:

$$\mathcal{K} = \sum_{i=1}^n \mathcal{K}_i = \sum_{i=1}^n \frac{1}{2} m_i x_{c_i}^2 + \frac{1}{2} \theta_i^2 I_i \quad (31)$$

Sendo m_i a massa de cada elo, x_{c_i} a velocidade do centro de massa, θ_i a velocidade angular e I_i o momento de inércia dos elos. Nesse trabalho os elos são definidos como um prisma retangular, portanto:

$$I_i = \frac{1}{12} m_i (l_i^2 + b_i^2) \quad (32)$$

Em que l_i é o comprimento e b_i a espessura de cada elo.

\mathcal{U} é o termo da energia potencial para um corpo rígido para os elos do manipulador, sendo definido por:

$$\mathcal{U} = \sum_{i=1}^n P_i = \sum_{i=1}^n m_i g y_{c_i} \quad (33)$$

O termo g é a aceleração da gravidade e y_{c_i} é a altura do centro de massa de cada elo.

Através da resolução da Equação 29, é obtida a expressão para cálculo do torque da Equação 34, onde M é a matriz de inércia, \mathbf{c} o vetor das forças de Coriolis e centrípeta e \mathbf{g} o vetor dos termos gravitacionais.

$$\boldsymbol{\tau} = M\ddot{\mathbf{q}} + \mathbf{c} + \mathbf{g} \quad (34)$$

2.3 Planejamento da trajetória do efetuador final

Para que um manipulador seja capaz de executar uma tarefa é preciso determinar o caminho que o efetuador final deve seguir e, além disso, definir quais são as velocidades e acelerações ao percorrer esse caminho. Esse processo é chamado de planejamento da trajetória e pode ser realizado de duas formas: no espaço das tarefas ou no espaço das juntas. O primeiro é a situação em que a posição cartesiana do efetuador final é determinada a cada instante de tempo, ou seja, existe um controle total da posição do efetuador. Já no segundo caso, são determinadas as orientações de cada junta do manipulador durante a trajetória, sendo conhecido as posições do efetuador final apenas em pontos determinados do caminho (OLIVEIRA, 2022).

De acordo com Siciliano *et al.* (2009), o caminho pode ser obtido através de dois métodos: movimentação ponto-a-ponto ou movimentação através de uma sequência de

pontos. No primeiro caso, o manipulador deve sair de uma configuração inicial e chegar em uma posição final em um determinado tempo t_f . Já o segundo caso ocorre quando são adicionados mais pontos entre a posição inicial e final para obter melhor monitoramento da trajetória e com isso é aumentado a complexidade matemática e custo computacional.

Além disso, como o presente trabalho tem por objetivo seguir as características apresentadas por Oliveira (2022), há intenção de que nos casos analisados o efetuador final percorra o mesmo caminho para efeitos de comparação. Com isso, foi escolhido a trajetória gerada no espaço das tarefas com movimentação ponto-a-ponto.

O posicionamento e orientação do efetuador podem ser obtidos utilizando um polinômio, o que permite a utilização de condições de contorno para a trajetória. Para o presente trabalho, escolheu-se o polinômio de quinto grau da Equação 35, uma vez que de acordo com Craig (2012) polinômios de ordem superior costumam ser usados para determinados segmentos da trajetória estudada. Para determinar os coeficientes da equação, são utilizadas como condições de contorno as posições ($p(t)$), velocidades ($\dot{p}(t)$) e acelerações ($\ddot{p}(t)$) iniciais ($t = t_i$) e finais ($t = t_f$) da trajetória do efetuador. Para que essa trajetória seja suave, é considerado nulas as velocidades e acelerações nos pontos iniciais e finais.

$$p(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (35)$$

De acordo com o trabalho de Macfarlane e Croft (2003), esse polinômio fornece um perfil quadrático para o pulso, evitando descontinuidades nos pontos finais da trajetória, fazendo com que não haja grandes variações de velocidade e aceleração.

Os coeficientes do polinômio podem ser calculados a partir de um sistema de equações. As 6 equações necessárias para aplicar-se as condições de contorno podem ser obtidas derivando a Equação 35 e, obtendo assim, as expressões para velocidade e aceleração. O sistema pode ser representado matricialmente da seguinte maneira:

$$\mathbf{p} = D\mathbf{a} \quad (36)$$

$$\begin{bmatrix} p(t_i) \\ p(t_f) \\ \dot{p}(t_i) \\ \dot{p}(t_f) \\ \ddot{p}(t_i) \\ \ddot{p}(t_f) \end{bmatrix} = \begin{bmatrix} 1 & t_i & t_i^2 & t_i^3 & t_i^4 & t_i^5 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_i & 3t_i^2 & 4t_i^3 & 5t_i^4 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_i & 12t_i^2 & 20t_i^3 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \quad (37)$$

Substituindo as condições de contorno no vetor $\mathbf{p} = [p(t_i), p(t_f), 0, 0, 0, 0]$, os coeficientes das equações são obtidos por:

$$\mathbf{a} = D^{-1}\mathbf{p} \quad (38)$$

Desse modo, os coeficientes do polinômio da equação 35 podem ser calculados da seguinte forma:

$$\begin{aligned} a_0 &= p(t_i) \\ a_1 &= a_2 = 0 \\ a_3 &= \frac{10(p(t_f) - p(t_i))}{t_f^3}, \quad a_4 = \frac{-15(p(t_f) - p(t_i))}{t_f^4}, \quad a_5 = \frac{6(p(t_f) - p(t_i))}{t_f^5} \end{aligned} \quad (39)$$

2.4 Redundância de manipuladores robóticos

De acordo com Duarte (2001), é possível definir redundância como sendo um sistema que pode apresentar várias configurações para uma determinada posição do órgão terminal no espaço cartesiano. De maneira geral, a redundância em manipuladores podem ser classificadas em redundância de sensor e mecânica. Para o caso dos sensores, pode ser definida como a situação em que há mais sensores presentes no manipulador do que o necessário, sendo comumente utilizados em sistemas que demandam alta confiabilidade e precisão de posicionamento. A redundância mecânica ainda pode ser dividida em redundância de atuação e cinemática.

Segundo Kotlarski *et al.* (2010), mesmo que um manipulador mecanicamente não redundante seja capaz de realizar uma tarefa definida, o robô com redundância cinemática pode aproveitar dessa condição para executar a tarefa e ainda otimizar um aspecto de interesse, como melhorar o desempenho dinâmico reduzindo os torques das juntas ou evitar que o robô atinja posições singulares.

A redundância de atuação é a situação em que o número de juntas atuadas do manipulador é maior que a sua mobilidade, sendo esse um caso mais recorrente em manipuladores com cadeia fechada, dado que em manipuladores seriais todas as juntas devem ser atuadas para que seja possível a sua movimentação e sustentação. Portanto, como a mobilidade de um robô em série é igual ao número de juntas atuadas, não existe redundância de atuação em manipuladores de cadeia aberta (OLIVEIRA, 2022).

Com isso, o presente trabalho tem o foco na redundância cinemática e uma breve revisão dos seus conceitos estão a apresentados a seguir.

2.4.1 Redundância cinemática da tarefa

Os manipuladores com redundância cinemática possuem infinitas soluções para a cinemática inversa, isto quer dizer que para uma mesma posição fixa do efetuador final, as juntas podem se movimentar alterando a configuração do manipulador. Além disso, a redundância cinemática permite que a configuração do manipulador seja alterada, podendo assim evitar singularidades (FONTES, 2019).

Segundo Weihmann (2011), a redundância cinemática de tarefa leva em consideração o que o manipulador deve executar com seu efetuador final. As tarefas ocorrem no chamado espaço das tarefas (d) e é definido pelo número de graus de liberdade necessários para executá-la. A dimensão do espaço d deve ser menor ou igual a mobilidade m do manipulador para que seja possível realizar a tarefa. O grau de redundância cinemática da tarefa R é definido pela Equação 40:

$$R = m - d \quad (40)$$

Uma vez que o interesse está nas tarefas que um manipulador deve executar, neste trabalho, por simplificação, o termo redundância cinemática é utilizado para referir-se à redundância cinemática da tarefa. Sendo assim, um manipulador possui redundância cinemática quando $R > 0$.

Considerando o caso dos manipuladores seriais em que a mobilidade m é igual ao número de juntas j , o grau de redundância desses robôs pode ser definido conforme a Equação 41:

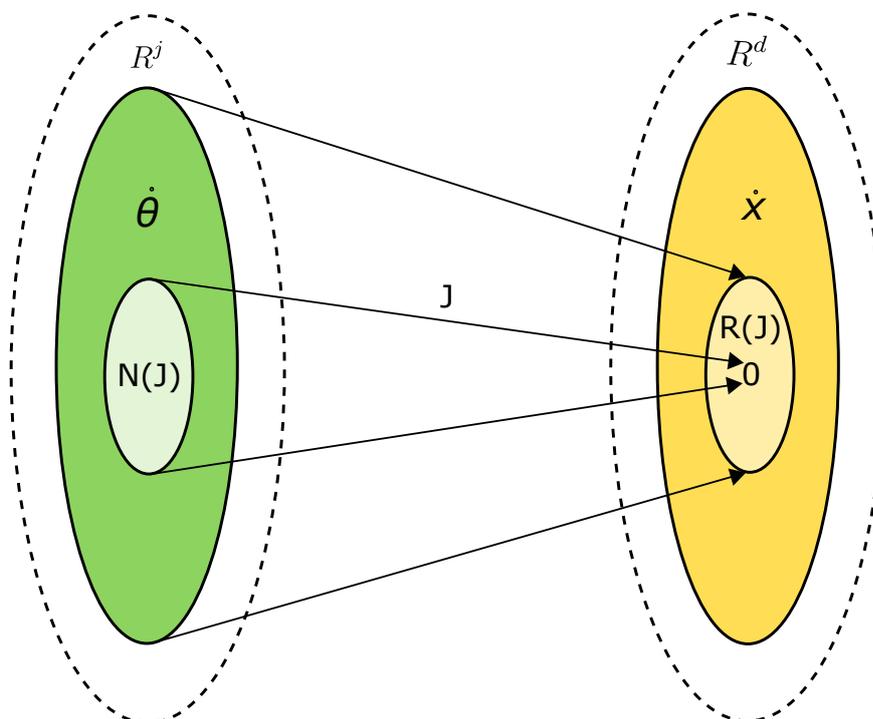
$$R = j - d \quad (41)$$

De acordo com Oliveira (2022), a redundância cinemática de um manipulador ocorre quando ele possui mais graus de liberdade do que a tarefa executada necessita. Se um manipulador serial com três juntas rotativas ($j = 3$) que trabalha no plano executa uma tarefa cuja a orientação do efetuador final não é relevante, implica em um espaço das tarefas com dimensão dois ($d = 2$). Sendo assim, esse manipulador possui redundância cinemática de grau um ($R = 1$).

A redundância cinemática também pode ser caracterizada como o caso em que existem movimentações das juntas que não alteram a posição do efetuador final. Como apresentado no trabalho de Weihmann (2011), a Figura 7 representa o mapeamento, utilizando um operador J , de um conjunto $N(J)$ de velocidades das juntas $\dot{\theta}$ que leva a um espaço nulo de velocidades \dot{x} do efetuador em uma determinada tarefa. No espaço $R(J)$ está representado o conjunto das velocidades que o efetuador pode assumir conforme a movimentação das juntas em um posição específica na trajetória. Na Figura 7, o conjunto R^j representa todas as movimentações possíveis das juntas do manipulador e o espaço R^d engloba as movimentações que o efetuador pode realizar considerando um espaço da tarefa de dimensão d .

Desse modo, é possível realizar a otimização de características do manipulador encontrando a configuração ótima das juntas para cada posição do efetuador durante a execução de uma tarefa.

Figura 7 – Espaço nulo dos manipuladores cinematicamente redundantes.



Fonte: Oliveira (2022).

2.5 Resolução da redundância

A resolução da redundância dos manipuladores robóticos com redundância cinemática consiste em determinar as posições das juntas do manipulador que permita o efetuador final executar uma tarefa e, ao mesmo tempo, consiga otimizar algum aspecto de interesse no robô, como reduzir os torques, aceleração e velocidade das juntas, melhorar o desempenho dinâmico ou reduzir o custo energético (NEDUNGADI; KAZEROUINIAN, 1989). O processo de resolução da redundância pode ser feito através do método local.

O método de resolução de redundância local consiste em definir uma função custo, que, considerando o ponto atual da configuração, direciona o movimento da junta redundante contra o gradiente da função custo, determinando a próxima posição das juntas com base na configuração atual (AHUACTZIN; GUPTA, 1999). De acordo com Nedungadi e Kazerouinian (1989), esse método é utilizado quando a função custo possui um cálculo rápido e menos complexo.

Segundo Ahuactzin e Gupta (1999), uma vantagem do método local é sua execução em tempo real o que permite lidar com obstáculos em movimento utilizando sensores de posicionamento que auxiliam a movimentação do robô durante a execução da tarefa. Porém, de acordo com Nenchev (1989), a resolução local pode apresentar dois problemas: não garantir uma boa solução otimizada e levar a instabilidades durante a movimentação do robô em longas trajetórias, ocasionando o aumento dos torques das juntas.

2.5.1 Otimização

A otimização é uma forma de determinar a melhor maneira de fazer algum processo. Matematicamente, é definida como o método de resolver problemas de maximização ou minimização de uma função em um dado domínio, sendo comum o caso em que as variáveis dessa função possuem um conjunto de restrições às quais a otimização deve respeitar (OLIVEIRA, 2022).

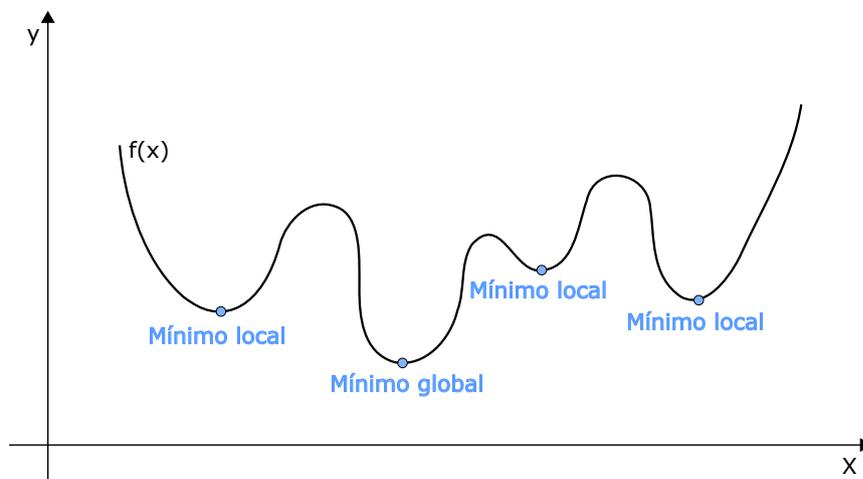
Esse tipo de problema pode ser definido como:

$$\begin{aligned} \text{minimizar: } & f(x) \\ \text{sujeito a: } & g_1(x), g_2(x), \dots, g_n(x) \end{aligned}$$

A função f é chamada de função custo ou função objetivo, x são as variáveis de otimização e g_n são as funções de restrição, as quais o ponto x deve satisfazer, sendo g_n também funções de x .

As soluções ótimas de um problema de otimização podem ser classificadas em solução ótima local ou global. Essa nomenclatura não tem relação direta com os métodos local e global da resolução da redundância. Para o caso de problemas convexos, segundo Fontes (2019), a solução ótima encontrada pode ser considerada uma solução global, uma vez que não há outra solução ótima para o problema. Os problemas de otimização não convexos possuem mais de uma solução ótima e pode ser considerada um ótimo local, entretanto, podem haver outras soluções locais que otimizam ainda mais a função custo. A Figura 8 mostra um exemplo de mínimos local e global em um dado domínio de uma função não convexa.

Figura 8 – Mínimos local e global de uma função não convexa em um determinado domínio.



Fonte: (OLIVEIRA, 2022).

Como o presente trabalho tem por finalidade encontrar um ótimo global como resultado da otimização, ainda assim não é possível garantir que ele seja resultado de uma

função não convexa, mesmo que essa possua um ótimo global. Com isso, neste trabalho será utilizado o método de multiplicação de Lagrange, sendo considerado uma estratégia mais restrita, como apresentado em Nenchev (1989), Nedungadi e Kazerouinian (1989) e Woolfrey, Lu e Liu (2019).

Vale destacar que, para as simulações feitas neste trabalho, as funções utilizadas em MATLAB foram baseadas naquelas já trabalhadas por Oliveira (2022) para o modo de trabalho da configuração "cotovelo para baixo" do manipulador.

2.6 Considerações finais do capítulo

Neste capítulo foram apresentadas toda a fundamentação teórica envolvida no desenvolvimento do trabalho e que foram empregados para realizar as simulações no *software* MATLAB. Foi abordado a modelagem do manipulador robótico RRR estudado, os equacionamentos da cinemática direta, da cinemática inversa para os dois modos de trabalho possíveis, dos cálculos dinâmicos e a definição da trajetória do manipulador. Além disso, o capítulo também abordou os conceitos de redundância cinemática e a sua resolução através do método local.

3 RESOLUÇÃO DA REDUNDÂNCIA

3.1 Resolução da redundância pelo método local

Para que um manipulador redundante execute a tarefa desejada, é necessário determinar as posições de suas juntas durante a trajetória do efetuador final, uma vez que o planejamento da trajetória do manipulador já foi considerado. De acordo com Oliveira (2022), esse problema é mais complexo do que resolver a cinemática inversa, quando comparado a manipuladores não redundantes, e pode ser resolvido como uma otimização com condições de contorno. Para isso, é necessário um método para determinar as posições das juntas que permita a execução da tarefa e que minimize uma função custo que representa uma característica de interesse no manipulador.

Pelo fato de que o presente trabalho utilizará uma função custo menos complexa e que pode ser calculada de forma mais rápida, utilizou-se o método local para resolução da redundância. Com isso, para efeitos de comparação, foram utilizadas duas funções custo, divididas em caso 1 e caso 2.

3.1.1 Caso 1 - Resolução local ponderada

Conforme proposto pelo trabalho de Nedungadi e Kazerouinian (1989), a função custo z_1 a ser minimizada para o caso 1 é definida como sendo a multiplicação dos vetores dos torques das juntas $\boldsymbol{\tau}$ ponderada pela matriz inversa de inércia M do manipulador. Essa otimização é restringida por uma condição que representa a cinemática do manipulador, na qual a aceleração do efetuador final é igual a aceleração das juntas. A otimização e a condição que a restringe são apresentadas na Equação 42.

$$\begin{aligned} \text{minimizar: } z_1 &= \boldsymbol{\tau}^T M^{-1} \boldsymbol{\tau} \\ \text{sujeito a: } J\ddot{\mathbf{q}} + \dot{J}\dot{\mathbf{q}} - \ddot{\mathbf{x}} &= 0 \end{aligned} \quad (42)$$

A otimização é realizada utilizando o método dos multiplicadores de Lagrange (λ), resultando na função Z :

$$Z = \boldsymbol{\tau}^T M^{-1} \boldsymbol{\tau} + \lambda^T (J\ddot{\mathbf{q}} + \dot{J}\dot{\mathbf{q}} - \ddot{\mathbf{x}}) \quad (43)$$

A solução deve satisfazer as condições abaixo:

$$\frac{\partial Z}{\partial \ddot{\mathbf{q}}} = 0 \quad \text{e} \quad \frac{\partial Z}{\partial \lambda} = 0 \quad (44)$$

Resolvendo as derivadas parciais apresentadas na Equação 44 tem-se:

$$\frac{\partial Z}{\partial \ddot{\mathbf{q}}} = 2(M\ddot{\mathbf{q}} + \mathbf{c} + \mathbf{g}) + J^T \lambda = 0 \quad (45)$$

$$\frac{\partial Z}{\partial \lambda} = J\ddot{\mathbf{q}} + \dot{J}\dot{\mathbf{q}} - \ddot{\mathbf{x}} = 0 \quad (46)$$

Isolando $\ddot{\mathbf{q}}$ na Equação 45 chega-se ao seguinte resultado:

$$\ddot{\mathbf{q}} = -0,5M^{-1}(J^T\lambda + 2\mathbf{c} + 2\mathbf{g}) \quad (47)$$

Substituindo a Equação 47 na Equação 46, tem-se:

$$-0,5JM^{-1}(J^T\lambda + 2\mathbf{c} + 2\mathbf{g}) + \dot{J}\dot{\mathbf{q}} - \ddot{\mathbf{x}} = 0 \quad (48)$$

Isolando λ da Equação 48:

$$\lambda = -2(JM^{-1}J^T)^{-1}(\ddot{\mathbf{x}} - \dot{J}\dot{\mathbf{q}} + JM^{-1}(\mathbf{c} + \mathbf{g})) \quad (49)$$

Assim, com o termo isolado na Equação 49 é possível substituí-lo na Equação 47:

$$\ddot{\mathbf{q}} = M^{-1}J^T(JM^{-1}J^T)^{-1}(\ddot{\mathbf{x}} - \dot{J}\dot{\mathbf{q}}) - (I - M^{-1}J^T(JM^{-1}J^T)^{-1}J)M^{-1}(\mathbf{c} + \mathbf{g}) \quad (50)$$

Desse modo, o termo $M^{-1}J^T(JM^{-1}J^T)^{-1}$ é a matriz pseudo-inversa J_M^\dagger da matriz Jacobiana ponderada pela matriz de Inércia. Por fim, Simplificando a Equação 50:

$$\ddot{\mathbf{q}} = J_M^\dagger(\ddot{\mathbf{x}} - \dot{J}\dot{\mathbf{q}}) - (I - J_M^\dagger J)M^{-1}(\mathbf{c} + \mathbf{g}) \quad (51)$$

Vale ressaltar que a Equação 51 apresentada leva em consideração as forças gravitacionais em sua totalidade. Além disso, as velocidades das juntas $\dot{\mathbf{q}}$ foram calculadas utilizando integração simples, no MATLAB, das acelerações $\ddot{\mathbf{q}}$ e conseqüentemente as posições \mathbf{q} são obtidas com a integração de $\dot{\mathbf{q}}$.

3.1.2 Caso 2 - Resolução local não ponderada

Para o caso da resolução local não ponderada utilizou-se a mesma metodologia do caso anterior. Porém, no caso 2 foi feita a mudança da função custo z_2 que será minimizada, que nesse caso não é ponderada pela matriz inversa de inércia M . A função custo utilizada para este caso está apresentada na Equação 52 e é definida pela multiplicação dos vetores dos torques $\boldsymbol{\tau}$.

$$\begin{aligned} \text{minimizar: } \quad & z_2 = \boldsymbol{\tau}^T \boldsymbol{\tau} \\ \text{sujeito a: } \quad & J\ddot{\mathbf{q}} + \dot{J}\dot{\mathbf{q}} - \ddot{\mathbf{x}} = 0 \end{aligned} \quad (52)$$

A resolução da redundância através do método local para o caso 2 está apresentado na Equação 53 com a matriz Jacobiana pseudo-inversa $J_{M^2}^\dagger$ sendo calculada através da seguinte expressão: $(M^2)^{-1}J^T(J(M^2)^{-1}J^T)^{-1}$.

$$\ddot{\mathbf{q}} = J_{M^2}^\dagger(\ddot{\mathbf{x}} - \dot{J}\dot{\mathbf{q}}) - (I - J_{M^2}^\dagger J)M^{-1}(\mathbf{c} + \mathbf{g}) \quad (53)$$

3.2 Método de comparação dos resultados

Assim como proposto pelo trabalho de Oliveira (2022), a fim de comparar os resultados obtidos com a resolução da redundância, será utilizado os valores dos torques das simulações de duas formas. A primeira, será através do valor médio dos torques de todas as juntas do manipulador, como apresentado na Equação 54, e a segunda, através da norma infinita dos torques $\|\tau\|_\infty$ de cada uma das juntas.

$$E = \frac{1}{n} \sum_{i=1}^n \sqrt{\tau^T \tau} \quad (54)$$

3.3 Considerações finais do capítulo

Neste capítulo foram apresentados os dois casos que foram utilizados para resolução da redundância através do método local. No caso 1, a função custo utilizada é ponderada pela matriz inversa de inércia M do manipulador, enquanto no caso 2 essa ponderação não é aplicada na função custo. Além disso, foi apresentado também os dois métodos que serão utilizados para comparação dos resultados obtidos, sendo eles o valor médio dos torques das juntas (E) e o valor da norma infinita dos torques ($\|\tau\|_\infty$) de cada uma das juntas do manipulador.

4 RESULTADOS

No presente capítulo são apresentados os resultados obtidos para as simulações feitas em MATLAB dos dois modos de trabalho de um manipulador RRR, levando em consideração também os resultados para o plano de trabalho espelhado, ou seja, para um manipulador situado no primeiro quadrante e, em seguida, no segundo quadrante do plano cartesiano. Para realizar as simulações, foi necessário definir as características físicas do manipulador robótico utilizado e a trajetória que seria realizada pelo efetuador final, para que assim fosse feita a resolução da redundância através do método local. Com isso, foram realizadas comparações entre os torques obtidos em cada junta do manipulador para cada caso e para os dois modos de trabalho utilizados. Além disso, a comparação dos torques de cada junta do manipulador também foi feita utilizando a média dos torques das juntas (E), conforme demonstrado na Equação 54, e também o valor máximo do torque de cada junta, obtido através da norma infinita $\|\tau\|_\infty$ desses valores.

A resolução do tema proposto está elencada neste capítulo da seguinte maneira: características do manipulador robótico, trajetória do efetuador final, resultados obtidos através das simulações, comparações dos resultados obtidos e considerações finais do capítulo.

4.1 Características do manipulador robótico

Os parâmetros utilizados para modelagem do manipulador robótico RRR em MATLAB são apresentados abaixo na Tabela 2, na qual os símbolos l , b , m e I são, respectivamente, comprimento, espessura, massa e momento de inércia dos elos do robô.

Tabela 2 – Parâmetros utilizados para modelagem do manipulador RRR.

Elo	$l[m]$	$b[m]$	$m[kg]$	$I[kg.m^2]$
1	1	0,10	5	0,42
2	1	0,10	2	0,16
3	1	0,10	1	0,08

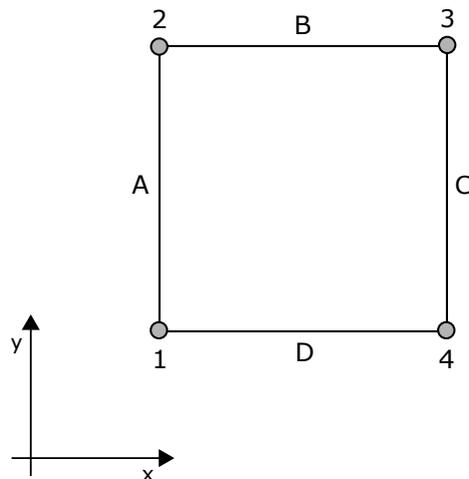
Fonte: Elaborado pelo autor (2023).

Vale destacar que, pelo fato de que o manipulador possui 3 juntas ($j = 3$) e a dimensão do espaço da tarefa é 2 ($d = 2$), o seu grau de redundância cinemática é de $R = 1$, calculado através da Equação 41.

4.2 Trajetória do efetuador final

Para este trabalho utilizou-se um caminho percorrido pelo efetuador final com o formato de um quadrado de lado $0,25\text{ m}$ e a trajetória foi dividida em quatro partes, sendo elas A, B, C e D e os vértices são denominados de 1, 2, 3 e 4, conforme apresentado na Figura 9. Além disso, os parâmetros assumidos para a trajetória, como posição, trajeto, comprimento do trajeto e tempo do trajeto estão apresentados na Tabela 3.

Figura 9 – Trajetória realizada pelo efetuado final do manipulador.



Fonte: Oliveira (2022) .

Tabela 3 – Dados da trajetória do efetuador final nas simulações.

Vértice	Posição (x;y) [m]	Trajeto	Comprimento do Trajeto [m]	Tempo do Trajeto [s]
1	1,25;1,25	A	0,25	0,50
2	1,25;1,50	B	0,25	0,50
3	1,50;1,50	C	0,25	0,50
4	1,50;1,25	D	0,25	0,50

Fonte: Elaborada pelo autor (2023).

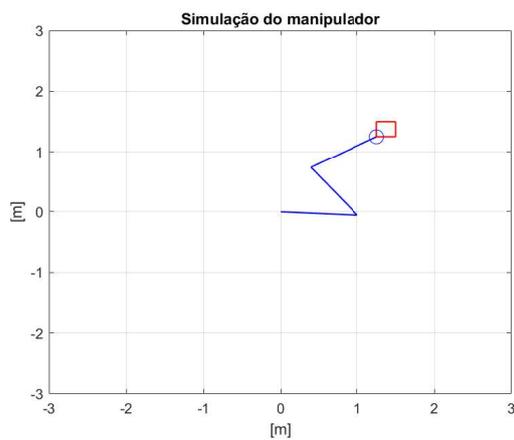
Conforme definido na Seção 2.3, para cada posição do vértice em que o manipulador realiza a sua trajetória, foram atribuídas as condições de contorno para os respectivos segmentos. As condições de contorno são as posições cartesianas dos vértices da trajetória, as velocidades nesses pontos, que são nulas, assim como as acelerações nesses pontos.

4.3 Resultados obtidos através das simulações

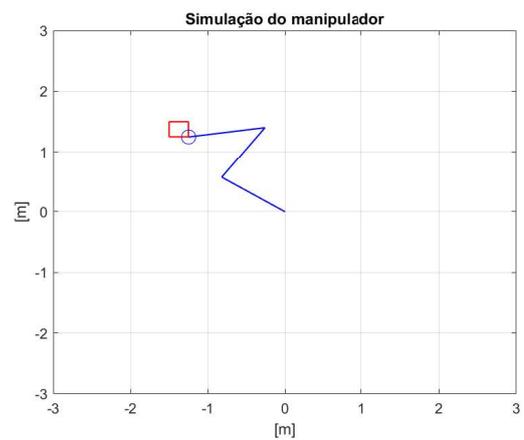
Uma vez implementado o modelo com os parâmetros e características gerais do manipulador RRR e definido os vértices e segmentos da trajetória a ser realizada pelo

efetuador, foram realizadas as simulações em MATLAB a fim de se obter os torques das juntas. Para isso, utilizou-se uma orientação de $\phi = 30^\circ$ para o efetuador, a qual foi escolhida arbitrariamente e aplicada a todas as simulações para os dois modos de trabalho do robô. Assim, foram simuladas as trajetórias do manipulador para cada modo de trabalho, localizados no primeiro e no segundo quadrante do plano cartesiano, conforme mostram as Figura 13 e 11.

Figura 10 – Simulação da trajetória do efetuador para o modo de trabalho do tipo “cotovelo para baixo”.



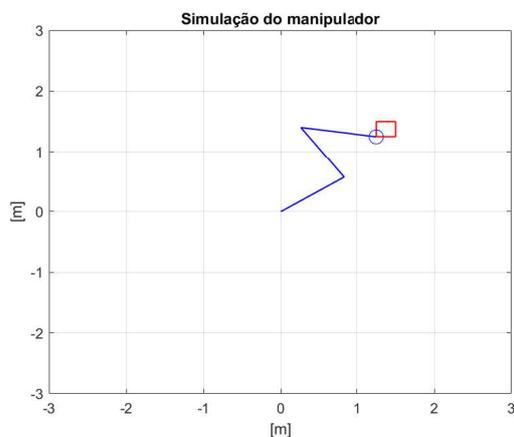
(a) Caso 1a.



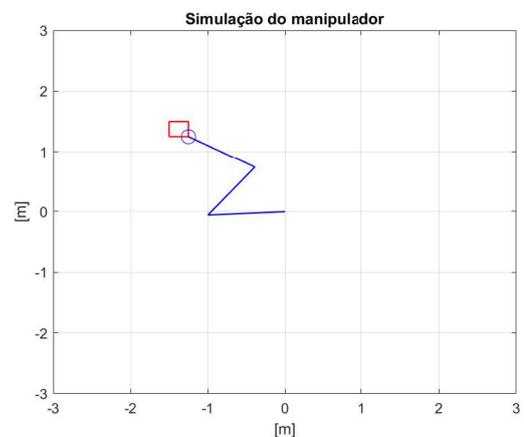
(b) Caso 1c.

Fonte: Elaborado pelo autor (2023).

Figura 11 – Simulação da trajetória do efetuador para o modo de trabalho do tipo “cotovelo para cima”.



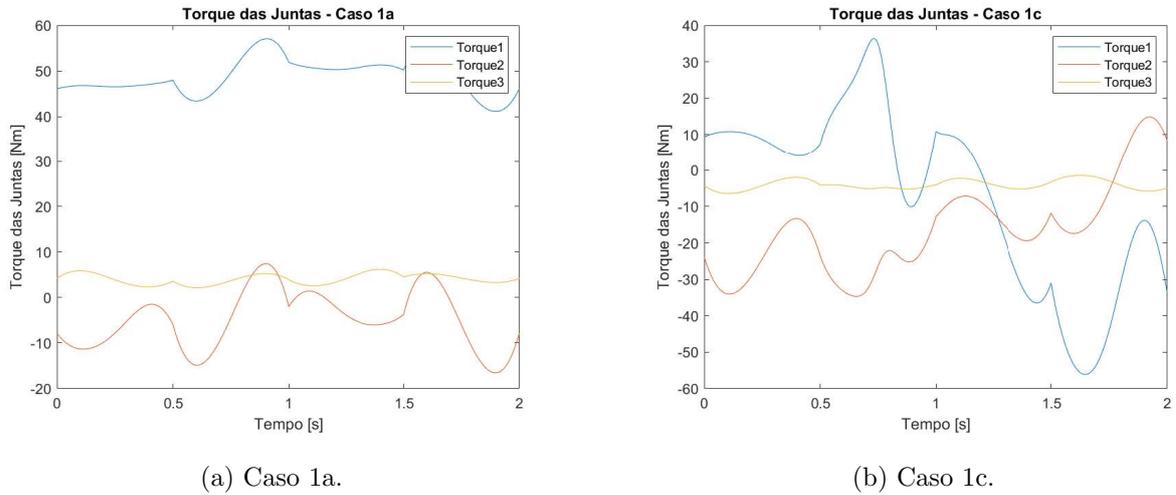
(a) Caso 1b.



(b) Caso 1d.

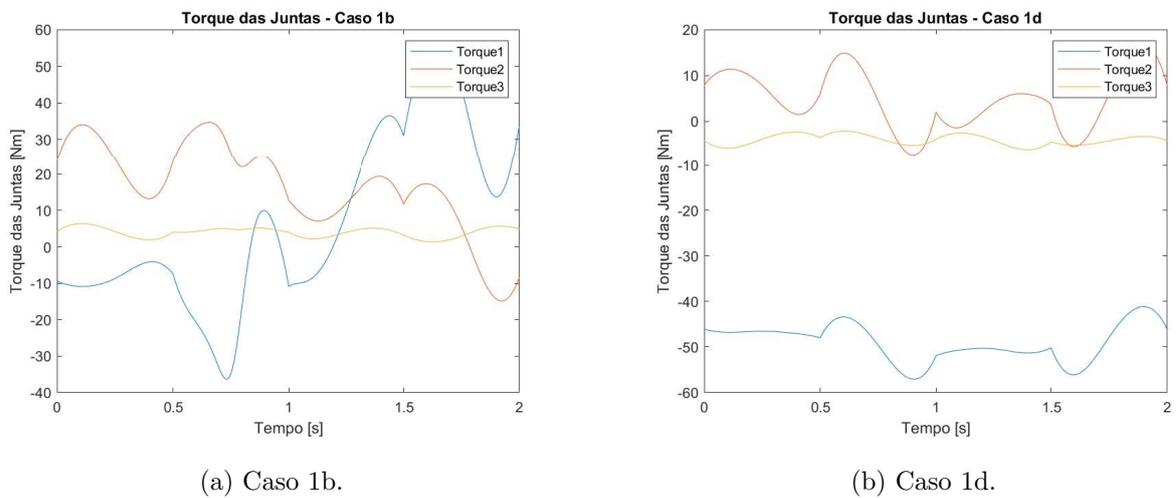
Fonte: Elaborado pelo autor (2023).

Figura 12 – Gráficos dos torques de cada junta para modo de trabalho do tipo “cotovelo para baixo” do Caso 1.



Fonte: Elaborado pelo autor (2023).

Figura 13 – Gráficos dos torques de cada junta para modo de trabalho do tipo “cotovelo para cima” do Caso 1.



Fonte: Elaborado pelo autor (2023).

Na Tabela 4, estão apresentados os resultados dos casos 1 e 2 e suas derivações, sendo que as siglas *a* e *b* referem-se ao modo de trabalho “cotovelo para baixo” e “cotovelo para cima”, respectivamente, situados no primeiro quadrante ($x \geq 0$ e $y \geq 0$) e as siglas *c* e *d* referem-se também ao modo de trabalho “cotovelo para baixo” e “cotovelo para cima”, respectivamente, porém situados no segundo quadrante ($x \leq 0$ e $y \geq 0$). Na Tabela 4 estão apresentados também o valor médio dos torques das juntas (E) e os torques máximos de cada junta ($\|\tau_1\|_\infty$, $\|\tau_2\|_\infty$ e $\|\tau_3\|_\infty$) do manipulador durante a execução da tarefa.

Tabela 4 – Valores da função E e da norma infinita de cada junta $\|\tau\|_\infty$ para os dois casos simulados e suas derivações.

Caso	Modo de trabalho	Quadrante	E [Nm]	$\ \tau_1\ _\infty$ [Nm]	$\ \tau_2\ _\infty$ [Nm]	$\ \tau_3\ _\infty$ [Nm]
1a	Para baixo	1°	49,86	57,07	16,56	6,21
1b	Para cima	1°	30,20	56,13	34,69	6,41
1c	Para baixo	2°	30,20	56,13	34,69	6,41
1d	Para cima	2°	49,86	57,07	16,56	6,21
2a	Para baixo	1°	49,69	51,63	16,89	6,32
2b	Para cima	1°	38,45	246,50	102,32	14,33
2c	Para baixo	2°	38,45	246,50	102,32	14,33
2d	Para cima	2°	49,69	51,63	16,89	6,32

Fonte: Elaborada pelo autor (2023).

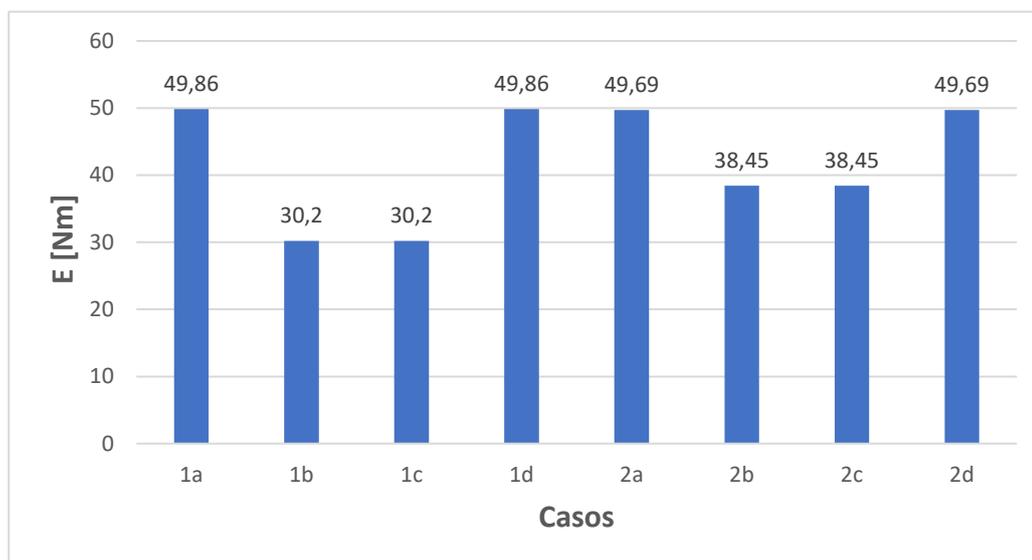
Os resultados presentes na Tabela 4 demonstram que, para os casos 1b e 1c, os valores do torque médio nas juntas (E) e dos torques máximos individuais ($\|\tau_1\|_\infty$, $\|\tau_2\|_\infty$ e $\|\tau_3\|_\infty$) são os mesmos e, apesar de haver uma redução em E , o torque na junta 2 é maior do que nos casos 1a e 1d. O mesmo ocorre nos casos 2b e 2c, porém nesses os valores dos torques máximos de cada junta são maiores do que no caso 1. Uma explicação plausível para esses valores reside no fato de que a metodologia utilizada para a resolução de redundância pelo método local faz com que o manipulador simulado procure o melhor modo de trabalho, a fim de se obter uma minimização do torque e fugindo de possíveis locais de singularidade. Por isso, nesses casos tem-se um torque máximo da junta 2 maior do que os demais, já que o manipulador muda bruscamente o seu modo de trabalho para evitar as singularidades. Na Tabela 4 é possível observar também que os valores de E e de $\|\tau_1\|_\infty$, $\|\tau_2\|_\infty$ e $\|\tau_3\|_\infty$ são os mesmos para as situações a e d e também para as situações b e c dos casos 1 e 2, resultado o qual será melhor discutido na Seção 4.4.

4.4 Comparações dos resultados obtidos

Para facilitar a visualização dos resultados obtidos com as simulações e a fim de possibilitar uma análise mais ampla dos casos estudados, foram simulados gráficos comparativos dos valores das médias dos torques das juntas para todas as situações, para as situações em que o modo de trabalho é do tipo “cotovelo para baixo” e para as situações em que o modo é do tipo “cotovelo para cima”. A Figura 14 mostra o gráfico de todos os casos simulados.

Analisando o gráfico da Figura 14, pode-se observar que, para os casos 1b e 1c, os valores de E são menores do que aqueles obtidos para os casos 2b e 2c, uma vez que no Caso 2 a função custo não é ponderada pela matriz inversa de inércia M . Além disso, nos casos 1a e 1d os valores são muito semelhantes aos obtidos nos casos 2a e 2d, indicando que a ponderação não teve efeito significativo na resolução da redundância pelo método

Figura 14 – Média dos torques das juntas do manipulador para todos os casos e situações.



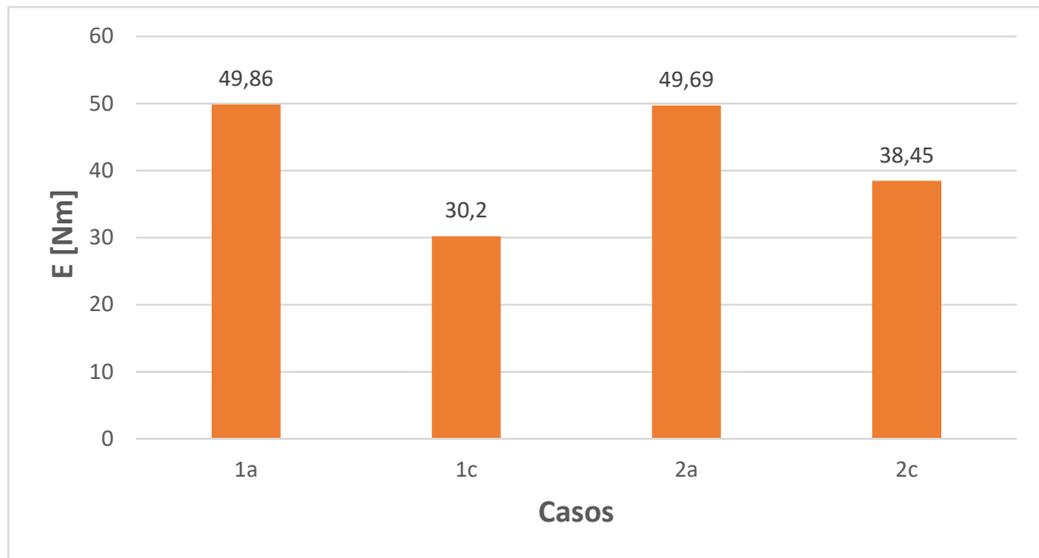
Fonte: Elaborado pelo autor (2023).

local.

Observando-se o gráfico da Figura 14 e os gráficos dos torques de cada junta dos Apêndices A e B, é possível notar que os valores das médias dos torques (E) e dos torques cada junta são os mesmos para as situações a e d e nas situações b e c dos casos 1 e 2, porém graficamente invertidos. Uma explicação para isso é o fato de que a configuração do robô é praticamente a mesma, porém de maneira espelhada, já que ao se espelhar o plano cartesiano de trabalho, invertendo o sinal do eixo x , também pode-se espelhar o modo de trabalho do manipulador ao inverter o sinal da Equação 12. Na sequência, tem-se o gráfico plotado para as situações em que o modo de trabalho adotado foi do tipo “cotovelo para baixo”, conforme mostra a Figura 15.

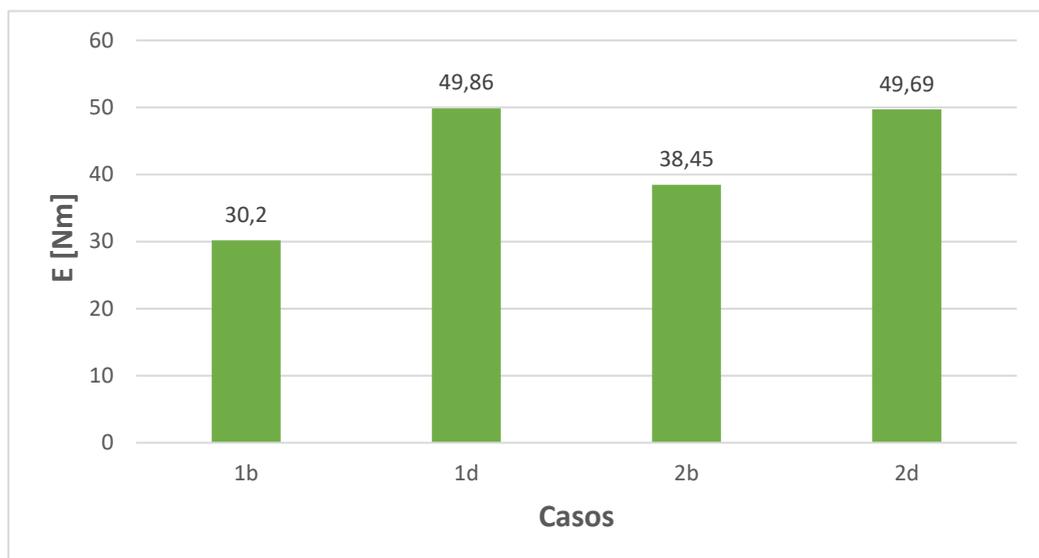
Observando o gráfico da Figura 15, é possível notar que para os casos 1c e 2c os valores de E são menores do que os valores obtidos nos casos 1a e 2a, uma vez em que nesses casos o manipulador está situado no segundo quadrante do plano cartesiano, ou seja, no plano de trabalho espelhado onde ocorrem mudanças no modo de trabalho para se evitar singularidades, visando minimizar o torque das juntas. A mudança brusca de modo de trabalho pode ser melhor visualizada através dos gráficos das Figuras 19 e 23, onde os casos 1c e 2c apresentam grandes variações nas curvas dos torques das juntas 1 e 2, enquanto nos casos 1a e 2a a curva dos respectivos torques é praticamente constante, conforme os gráficos das Figuras 17 e 21. Por fim, foi plotado também o gráfico para o caso em que utilizou-se o modo de trabalho do tipo “cotovelo para cima”, como mostra a Figura 16.

Figura 15 – Média dos torques das juntas do manipulador para o modo de trabalho do tipo “cotovelo para baixo”.



Fonte: Elaborado pelo autor (2023).

Figura 16 – Média dos torques das juntas do manipulador para o modo de trabalho do tipo “cotovelo para cima”.



Fonte: Elaborado pelo autor (2023).

O gráfico da Figura 16 permite se ter uma análise contrária àquela feita para o outro modo de trabalho, ou seja, em relação aos casos *1d* e *2d*, os valores da média dos torques

das juntas dos casos $1b$ e $2b$ são menores, pelo fato de que nesses o manipulador se encontra no primeiro quadrante do plano cartesiano e neste plano de trabalho o manipulador realiza trocas no modo de funcionamento a fim de se evitar singularidades, visando minimizar o torque das juntas. Além disso, os gráficos das Figuras 18 e 22 demonstram a mudança brusca de modo de trabalho, onde os casos $1b$ e $2b$ apresentam grandes variações nas curvas dos torques das juntas 1 e 2, enquanto nos casos $1d$ e $2d$ a curva dos respectivos torques é praticamente constante, como pode ser visto nas Figuras 20 e 24.

Vale salientar que, conforme apontado por Nenchev (1989), uma característica do método local para resolução de redundância é que este pode levar a variações bruscas dos torques das juntas do robô. Apesar da redução considerável de E em relação aos outros casos, essa característica pode ser observada nos gráficos das Figuras 18, 19, 22 e 23 dos casos $1b$, $1c$, $2b$ e $2c$, respectivamente.

4.5 Considerações finais do capítulo

Nesse capítulo foram apresentados os resultados obtidos através das simulações em MATLAB para os dois casos de resolução pelo método global e as discussões decorrentes para os dois modos de trabalho, levando em consideração o primeiro e segundo quadrante do plano cartesiano. Foi visto que o modo de trabalho do tipo “cotovelo para baixo” possui uma trajetória mais suave e com poucas variações de torque nas juntas quando o manipulador se encontra no primeiro quadrante, enquanto para o modo de trabalho do tipo “cotovelo para cima” tem-se um resultado semelhante quando o manipulador se encontra no segundo quadrante. Além disso, foi visto que quando o modo de trabalho do tipo “cotovelo para baixo” e “cotovelo para cima” estão situados respectivamente no segundo e no primeiro quadrante, tem-se uma variação brusca dos torques das juntas do robô em ambos os casos estudados.

5 CONCLUSÃO

No presente trabalho foi apresentada uma avaliação dos dois modos de trabalho possíveis de um manipulador robótico do tipo RRR para a resolução de redundância através do método local. A modelagem cinemática e dinâmica do manipulador RRR foram desenvolvidas e implementadas no *software* MATLAB e através desse modelo foi possível realizar a resolução da redundância através do método local. A fim de se otimizar o espaço de trabalho e a trajetória realizada pelo manipulador, a resolução da redundância pelo método local foi feita para duas funções custos diferentes, divididas em caso 1 e caso 2, para que se pudesse avaliar a consistência dos resultados em condições distintas.

Em relação à avaliação do impacto dos modos de trabalho do manipulador robótico, os dois casos de resolução de redundância pelo método local foram simulados para os dois modos de trabalho do robô, sendo eles o modo do tipo “cotovelo para baixo” e o modo do tipo “cotovelo para cima”. Além disso, para os dois modos de funcionamento e para os dois casos de resolução, foram feitas simulações considerando o plano de trabalho espelhado em que o robô estava situado, ou seja, as simulações foram feitas tanto para o primeiro quadrante quanto para o segundo quadrante do plano cartesiano.

Os resultados obtidos com as simulações demonstraram que, para ambos os casos, o melhor modo de trabalho do robô depende do ponto de vista geométrico, ou seja, se o robô está localizado no primeiro ou no segundo quadrante. Para o um manipulador situado no primeiro quadrante, o modo de trabalho do tipo “cotovelo para baixo” apresenta um valor médio dos torques das juntas (E) maior em relação ao outro modo, porém sem variações bruscas dos torques das juntas individuais. Já para um manipulador situado no segundo quadrante do plano cartesiano, o melhor modo de trabalho seria o do tipo “cotovelo para cima”, pela mesma justificativa apresentada, uma vez que essa seria a configuração do robô espelhada relação ao primeiro quadrante.

Com isso, as variações bruscas de torque nas juntas do manipulador para alguns dos casos simulados deve-se ao fato de que a metodologia da resolução de redundância local implementada em MATLAB busca sempre o melhor modo de trabalho para o manipulador, ou seja, o modo que possui um menor torque e que evita possíveis singularidades. Este fenômeno foi apresentado por Nenchev (1989) como sendo uma característica do método de resolução local e também uma desvantagem do mesmo quando comparado à outros métodos de resolução de redundância. Vale destacar que, conforme feito no trabalho de Oliveira (2022), essa desvantagem do método local, visualizada através das variações bruscas do torque das juntas, pode ser solucionada realizando a resolução da redundância através do método local combinado com a otimização conjunta da orientação inicial do efetuador e do fator de ponderação das forças gravitacionais.

5.1 Trabalhos futuros

Para sequência da linha de estudo abordada neste trabalho e também a fim de se ampliar os conteúdos relacionados, alguns pontos podem ser levantados para trabalhos futuros:

- Realizar a resolução da redundância do modelo proposto através do método global, considerando também os dois modos de trabalho possíveis;
- Otimizar a orientação inicial do efetuador e o fator de ponderação das forças gravitacionais, a fim de se comparar os resultados para um manipulador redundante otimizado;
- Utilizar um modelo mais próximo de um manipulador real, considerando os motores das juntas com suas limitações físicas de movimentação e torque suportado;
- Explorar novos métodos de resolução da redundância, tais como: inteligência artificial e redes neurais.

REFERÊNCIAS

- AHUACTZIN, J.-M.; GUPTA, K. The kinematic roadmap: A motion planning based global approach for inverse kinematics of redundant robots. **Robotics and Automation, IEEE Transactions on**, v. 15, p. 653 – 669, 09 1999.
- ASADA, H.; SLOTINE, J.-J. **Robot analysis and control**. [S.l.]: John Wiley & Sons, 1991.
- CRAIG, J. J. Robótica. 3^a edição. **Rev. Atual**, 2012.
- DENAVID, J.; HARTENBERG, R. S. A kinematic notation for lower-pair mechanisms based on matrices. **Trans. ASME E, Journal of Applied Mechanics**, v. 22, p. 215–221, June 1955.
- DUARTE, F. Controlo de robots redundantes. **Millenium**, Instituto Politécnico de Viseu, p. 52–83, 2001.
- FONTES, J. V. de C. **Controle de um manipulador planar paralelo com redundância cinemática**. Tese (Doutorado), 2019. Disponível em: <<https://doi.org/10.11606/t.18.2019.tde-22052019-152125>>.
- FONTES, J. V. de C.; SILVA, M. M. da. Simulation and experimental verification of a global redundancy resolution for a 3prrr prototype. In: **Multibody Mechatronic Systems**. Springer International Publishing, 2017. p. 315–324. Disponível em: <https://doi.org/10.1007/978-3-319-67567-1_30>.
- KOTLARSKI, J. *et al.* Optimization strategies for additional actuators of kinematically redundant parallel kinematic machines. **IEEE**, maio 2010. Disponível em: <<https://doi.org/10.1109/robot.2010.5509982>>.
- KUKA Roboter do Brasil. **Robôs industriais da KUKA**. 2020. Disponível em: <<https://www.kuka.com/pt-br/produtos-serviços/sistemas-de-robôs/robôs-industriais>>.
- LOPES, A. M. Modelação cinemática e dinâmica de manipuladores de estrutura em série. **Porto: FEUP**, 2001.
- MACFARLANE, S.; CROFT, E. Jerk-bounded manipulator trajectory planning: design for real-time applications. **IEEE Transactions on Robotics and Automation**, Institute of Electrical and Electronics Engineers (IEEE), v. 19, n. 1, p. 42–52, fev. 2003. Disponível em: <<https://doi.org/10.1109/tra.2002.807548>>.
- NEDUNGADI, A.; KAZEROUINIAN, K. A local solution with global characteristics for the joint torque optimization of a redundant manipulator. **Journal of Robotic Systems**, v. 6, n. 5, p. 631–654, 1989. ISSN 10974563.
- NENCHEV, D. N. Redundancy resolution through local optimization: A review. **Journal of Robotic Systems**, v. 6, n. 6, p. 769–798, 1989. ISSN 10974563.
- NUNES, R. F. Mapeamento da cinemática inversa de um manipulador robótico utilizando redes neurais artificiais configuradas em paralelo. Universidade Estadual Paulista (UNESP), 2016.

OLIVEIRA, G. C. d. Otimização do desempenho dinâmico de um manipulador robótico planar em série com redundância no espaço das tarefas. Universidade Federal de São Carlos, 2022.

OLIVEIRA, L. R. B. d. Cinemática inversa para simulação de figuras articuladas. Universidade do Vale do Rio dos Sinos, 2018.

SCIAVICCO, L.; SICILIANO, B. **Modelling and control of robot manipulators**. [S.l.]: Springer Science & Business Media, 2001.

SICILIANO, B. *et al.* **Robotics**. London: Springer London, 2009. 632 p. (Advanced Textbooks in Control and Signal Processing). ISSN 1098-6596. ISBN 978-1-84628-641-4. Disponível em: <<http://link.springer.com/10.1007/978-1-84628-642-1>>.

SIMPLÍCIO, P. V. G.; LIMA, B. R. *et al.* Manipuladores robóticos industriais. **Caderno de Graduação-Ciências Exatas e Tecnológicas-UNIT-SERGIPE**, v. 3, n. 3, p. 85–85, 2016.

WEIHMANN, L. **Modelagem e Otimização de Forças e Torques Aplicados por Robôs com Redundância Cinemática e de Atuação em Contato com o Meio**. 332 p. Tese (Tese) — Universidade Federal de Santa Catarina, 2011.

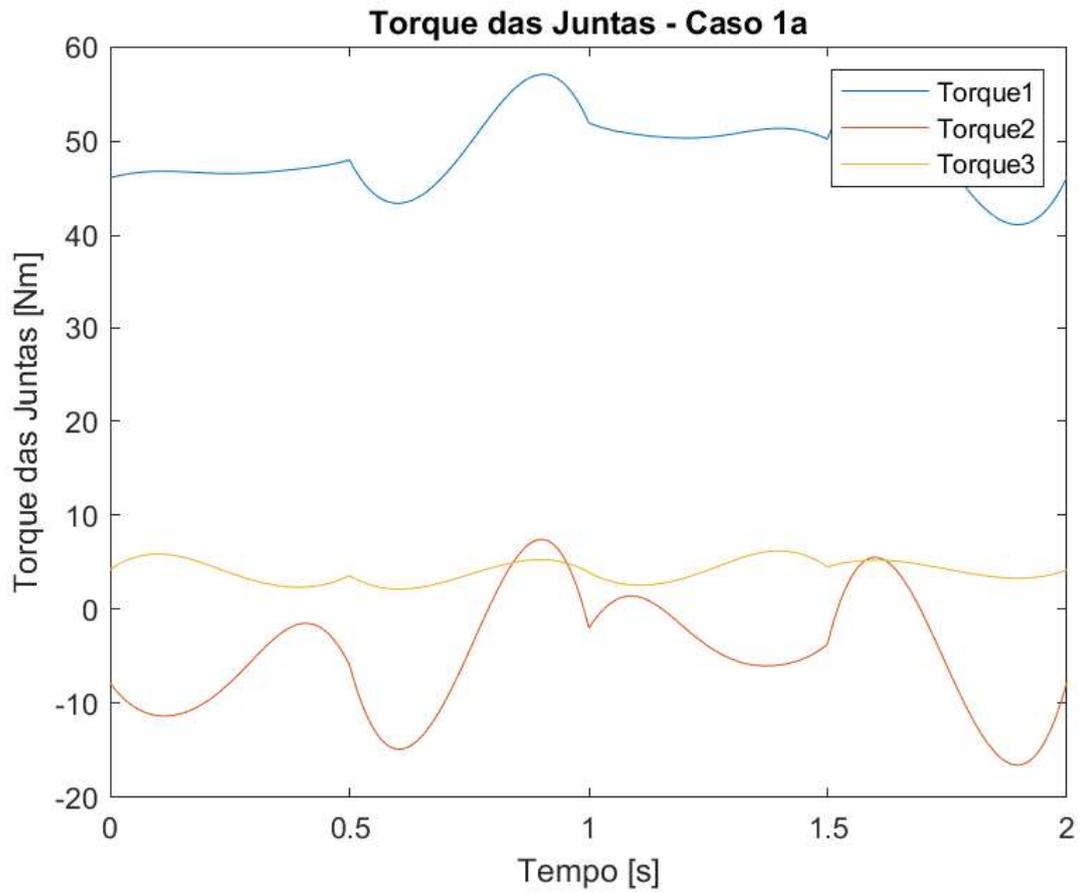
WENGER, P.; CHABLAT, D. Workspace and assembly modes in fully-parallel manipulators: A descriptive study. In: **Advances in Robot kinematics: Analysis and control**. [S.l.]: Springer, 1998. p. 117–126.

WOOLFREY, J.; LU, W.; LIU, D. A Control Method for Joint Torque Minimization of Redundant Manipulators Handling Large External Forces. **Journal of Intelligent and Robotic Systems: Theory and Applications**, Journal of Intelligent & Robotic Systems, v. 96, n. 1, p. 3–16, 2019. ISSN 15730409.

Apêndices

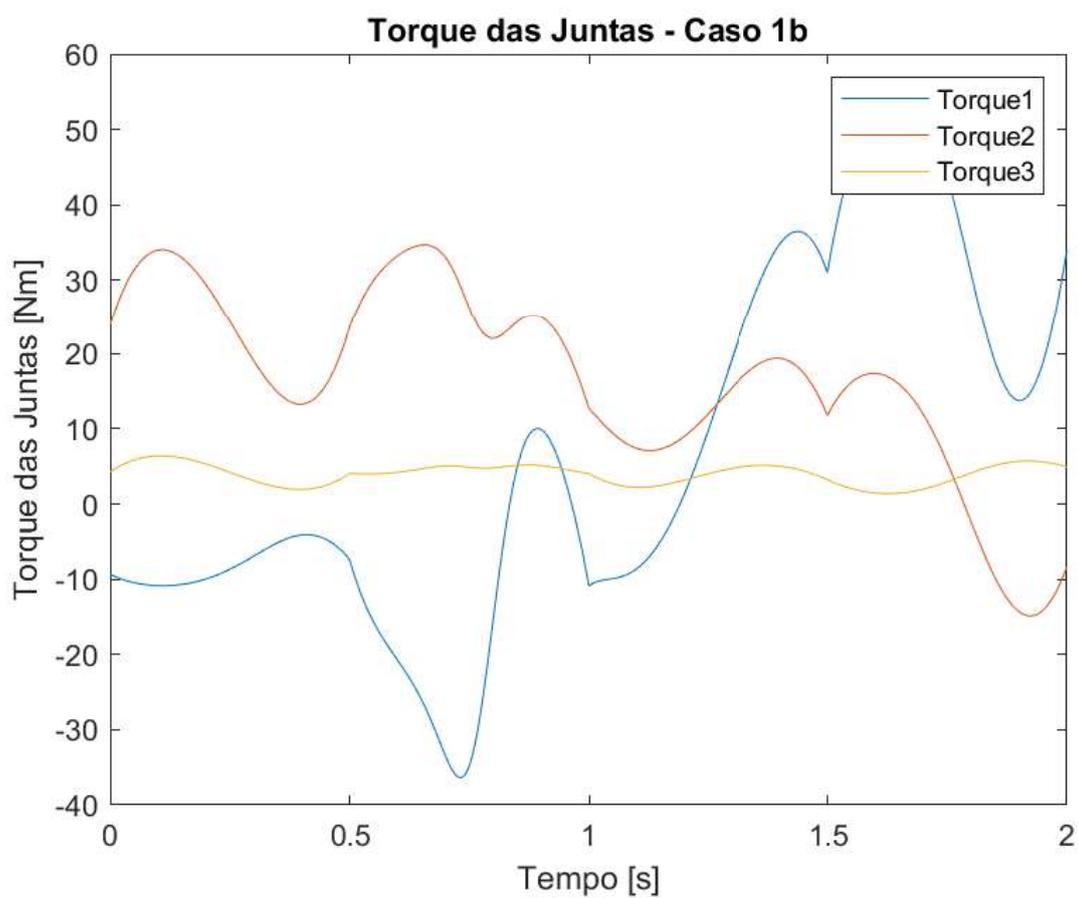
APÊNDICE A – Gráficos dos torques das juntas do manipulador para o caso 1

Figura 17 – Torque das juntas do manipulador do caso 1a.



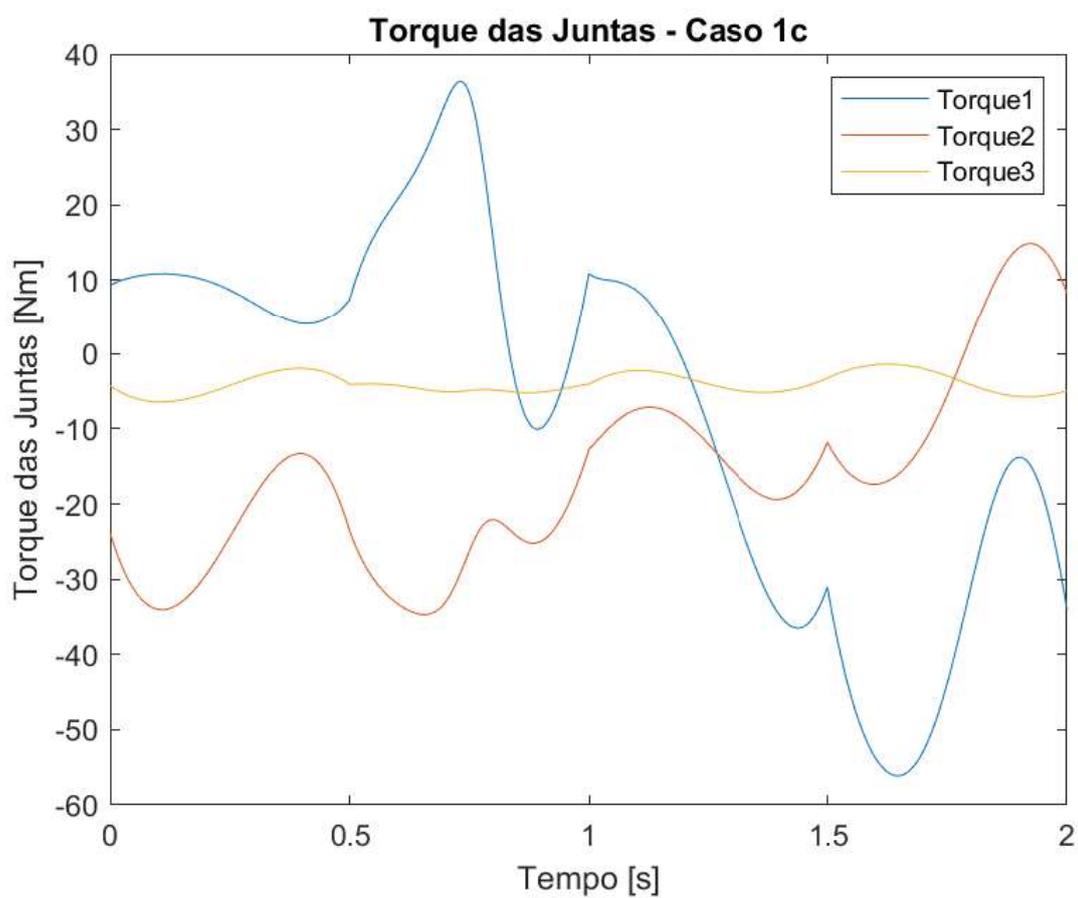
Fonte: Elaborado pelo autor (2023).

Figura 18 – Torque das juntas do manipulador do caso 1b.



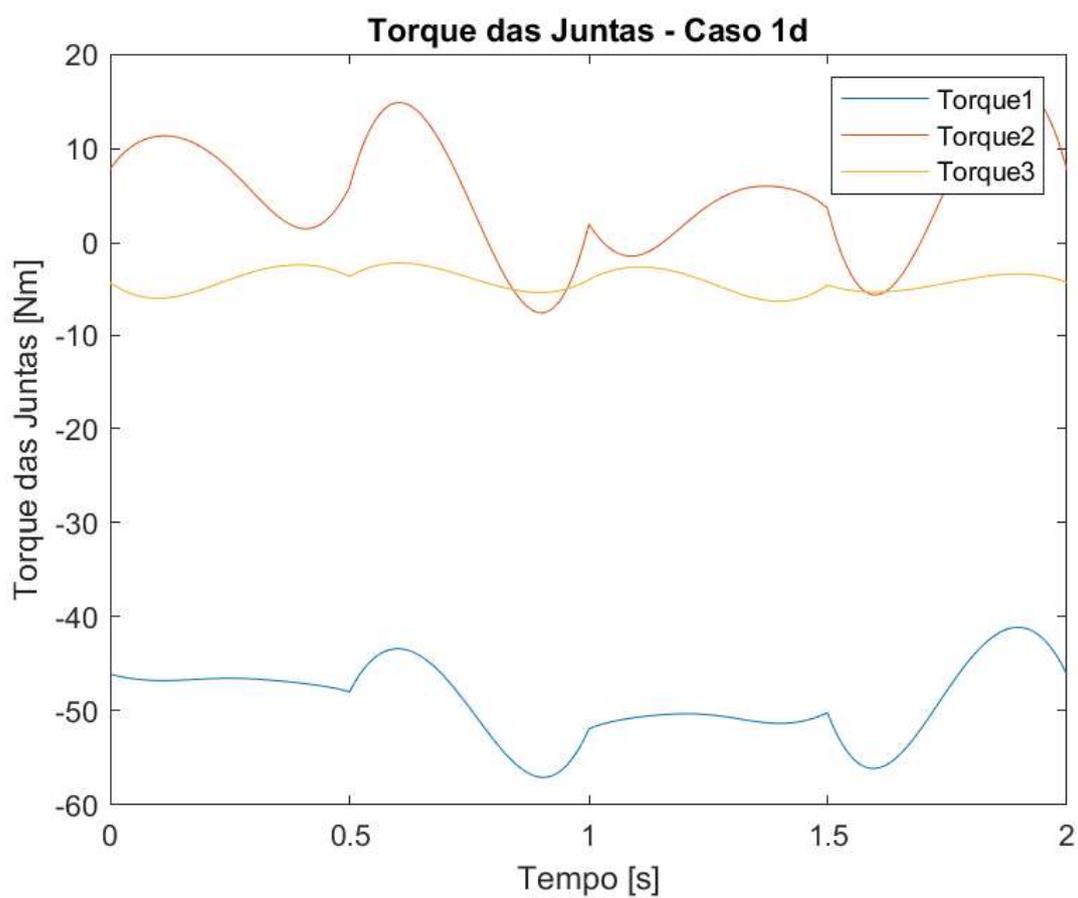
Fonte: Elaborado pelo autor (2023).

Figura 19 – Torque das juntas do manipulador do caso 1c.



Fonte: Elaborado pelo autor (2023).

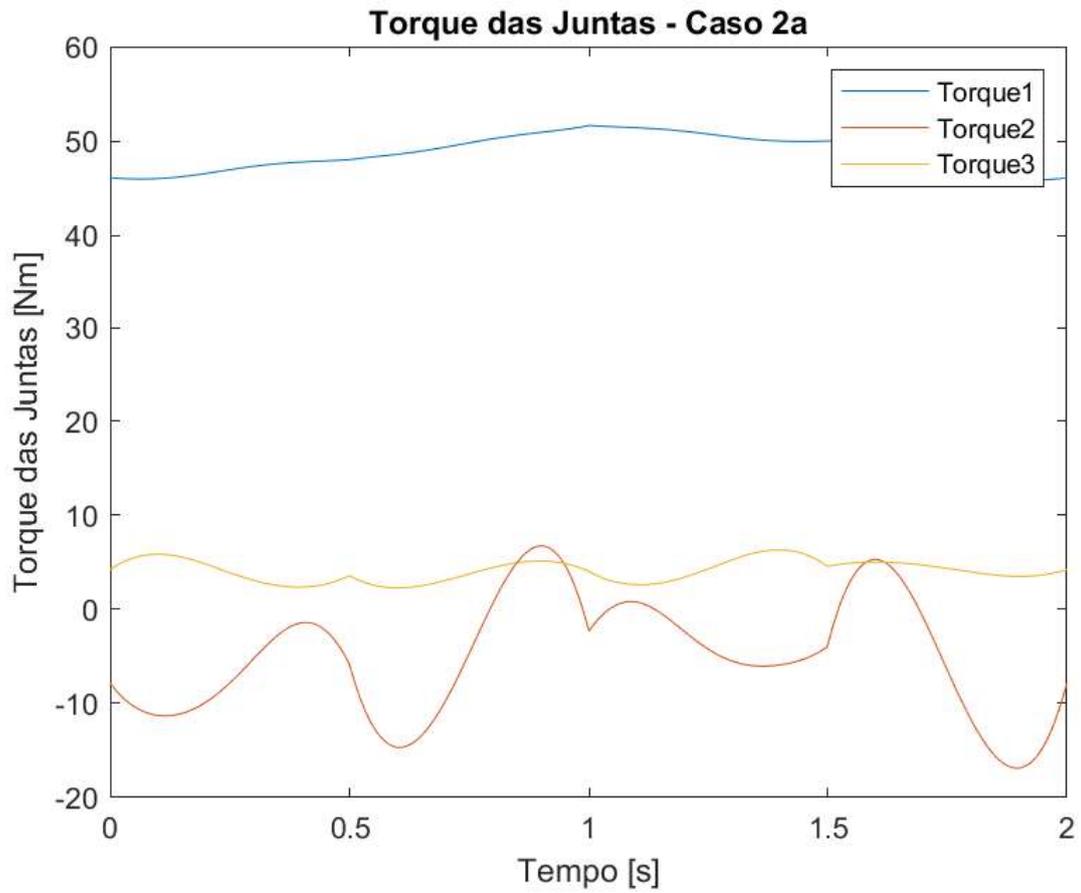
Figura 20 – Torque das juntas do manipulador do caso 1d.



Fonte: Elaborado pelo autor (2023).

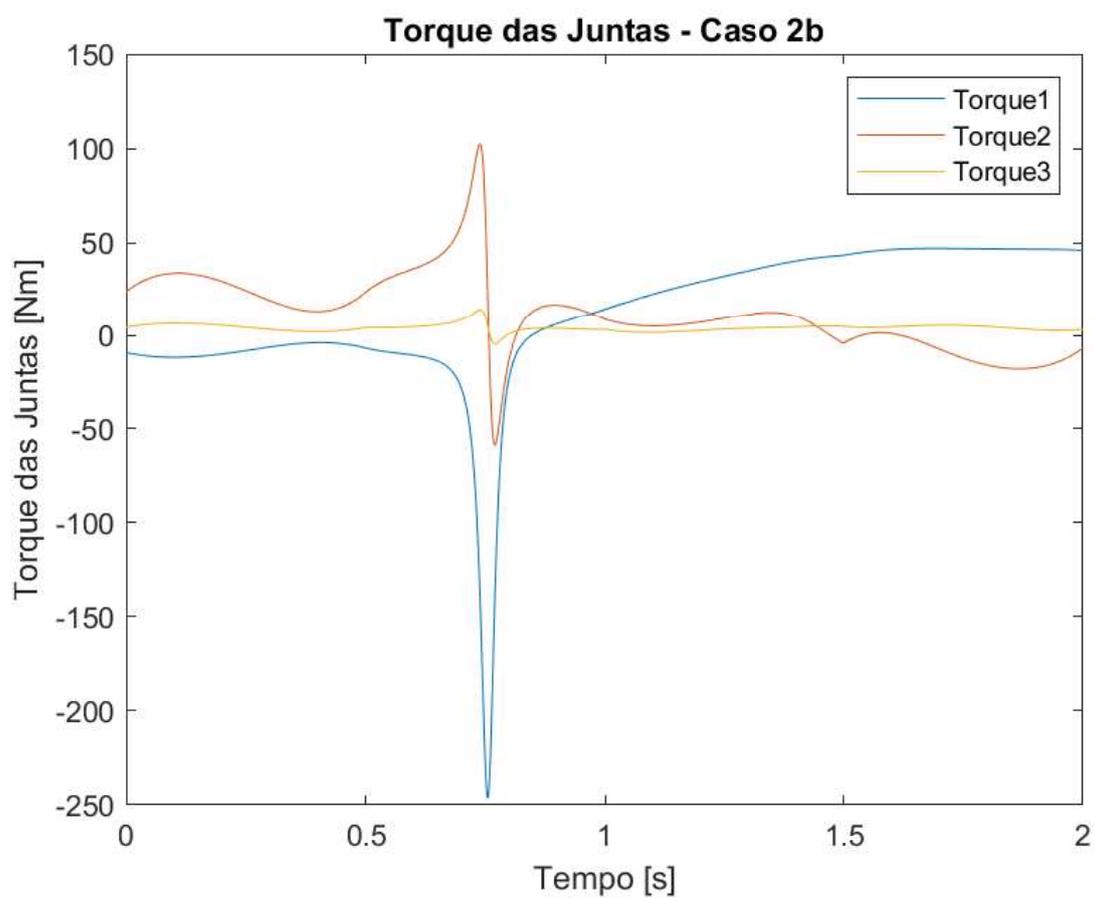
APÊNDICE B – Gráficos dos torques das juntas do manipulador para o caso 2

Figura 21 – Torque das juntas do manipulador do caso 2a.



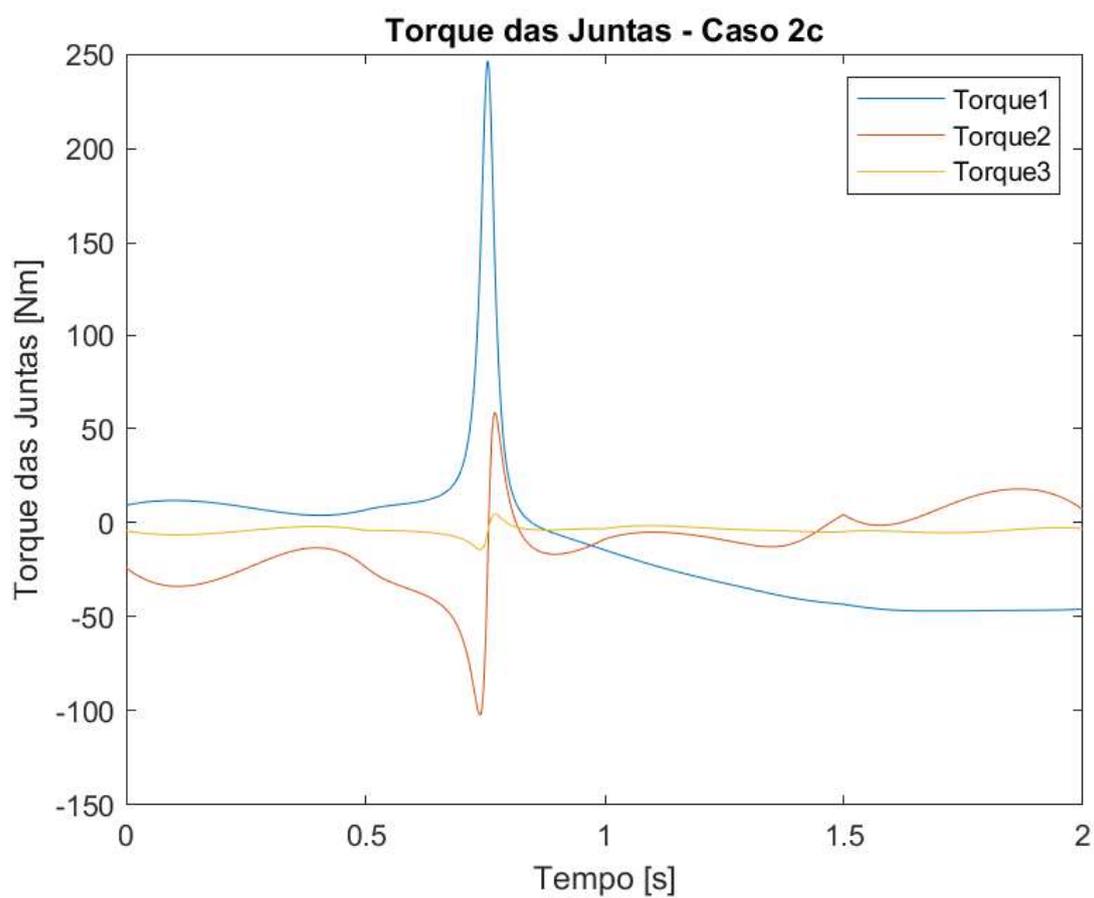
Fonte: Elaborado pelo autor (2023).

Figura 22 – Torque das juntas do manipulador do caso 2b.



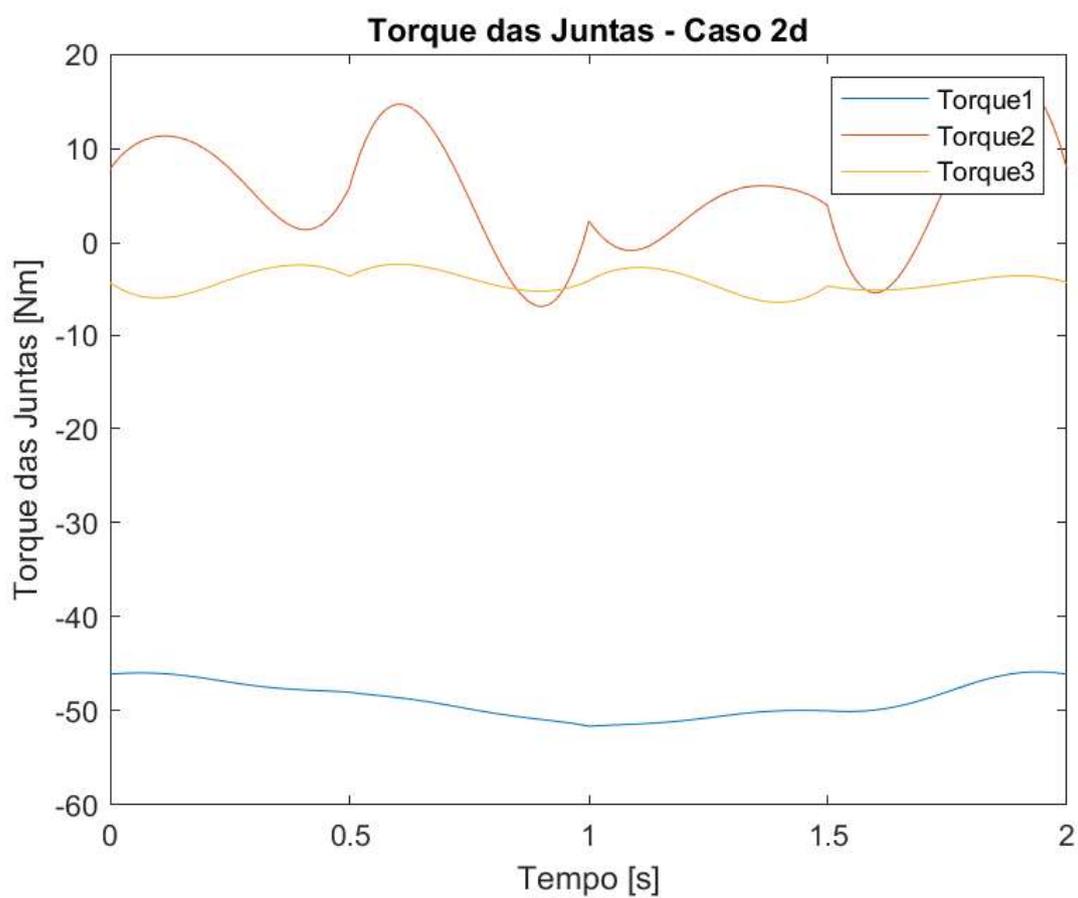
Fonte: Elaborado pelo autor (2023).

Figura 23 – Torque das juntas do manipulador do caso 2c.



Fonte: Elaborado pelo autor (2023).

Figura 24 – Torque das juntas do manipulador do caso 2d.



Fonte: Elaborado pelo autor (2023).

APÊNDICE C – Exemplo de código utilizado para o cálculo da cinemática inversa

```

function Teta = Cinematica_Inversa(x)

    phi = x(3);
    %orientacao do tcp em relacao ao eixo x

    %comprimento dos elos em metros
    L1 = 1;
    L2 = 1;
    L3 = 1;
    b = 0.1;          %altura da barra do link

    %posição e orientacao do tcp
    xe = x(1);       %posicao do tcp no eixo x
    ye = x(2);       %posicao do tcp no eixo y

    %eq cinematica reversa
    x2 = xe - L3*cos(phi);
    y2 = ye - L3*sin(phi);
    d = sqrt(x2^2 + y2^2);

    c2 = (d^2 - L1^2 - L2^2)/(2*L1*L2);
    s2 = -(sqrt(1-(c2)^2));
    c1 = ((L1+L2*c2)*x2 + (L2*s2)*y2)/d;
    s1 = ((L1+L2*c2)*y2 - (L2*s2)*x2)/d;

    Teta1 = atan2(s1,c1);
    Teta2 = atan2(s2,c2);
    Teta3 = phi - Teta1 - Teta2;

    Teta = [Teta1 Teta2 Teta3];

end

```

APÊNDICE D – Exemplo de código utilizado na resolução da redundância do caso 1

```

clear all
close all
clc

%posicao inicial

phi = (pi-(pi/6));
%phi = (pi/6);

aaa=1.25;
bbb=1.5;
Ap = [-aaa,aaa];
Bp = [-aaa,bbb];
Cp = [-bbb,bbb];
Dp = [-bbb,aaa];
Teta_FC = Cinematica_Inversa([Ap phi]);

Teta1 = Teta_FC(1);    %converte p/ radianos
Teta2 = Teta_FC(2);    %converte p/ radianos
Teta3 = Teta_FC(3);    %converte p/ radianos

Teta1dot = 0;
Teta2dot = 0;
Teta3dot = 0;

Tetadot = [Teta1dot; Teta2dot; Teta3dot];

L1 = 1;    %comprimento do link 1 em metros
L2 = 1;    %comprimento do link 2 em metros
L3 = 1;    %comprimento do link 3 em metros
b = 0.1;    %altura da barra do link
lc1 = L1/2;    lc2 = L2/2;    lc3 = L3/2;    %posição dos cgs

m1 = 5;    m2 = 2;    m3 = 1;    %massa dos links
g = 9.81;    %aceleração da gravidade
Iz1 = (1/12)*m1*(L1^2 + b^2);
Iz2 = (1/12)*m2*(L2^2 + b^2);
Iz3 = (1/12)*m3*(L3^2 + b^2);    %momentos de inercia dos links

I = eye(3); %Matriz Identidade 3x3
counter = 1;    %contador de iteracoes
dt = 0.001; %step time
time = 0;    %tempo de simulacao
E_anterior = 0;

```

```
%calculo do polinomio
```

```
ao = Ap(1);  
a3 = (10*(Bp(1) - Ap(1)))/(0.5^3);  
a4 = (-15*(Bp(1) - Ap(1)))/(0.5^4);  
a5 = (6*(Bp(1) - Ap(1)))/(0.5^5);
```

```
bo = Bp(1);  
b3 = (10*(Cp(1) - Bp(1)))/(0.5^3);  
b4 = (-15*(Cp(1) - Bp(1)))/(0.5^4);  
b5 = (6*(Cp(1) - Bp(1)))/(0.5^5);
```

```
co = Cp(1);  
c3 = (10*(Dp(1) - Cp(1)))/(0.5^3);  
c4 = (-15*(Dp(1) - Cp(1)))/(0.5^4);  
c5 = (6*(Dp(1) - Cp(1)))/(0.5^5);
```

```
do = Dp(1);  
d3 = (10*(Ap(1) - Dp(1)))/(0.5^3);  
d4 = (-15*(Ap(1) - Dp(1)))/(0.5^4);  
d5 = (6*(Ap(1) - Dp(1)))/(0.5^5);
```

```
%AGORA PRA Y
```

```
aoy = Ap(2);  
a3y = (10*(Bp(2) - Ap(2)))/(0.5^3);  
a4y = (-15*(Bp(2) - Ap(2)))/(0.5^4);  
a5y = (6*(Bp(2) - Ap(2)))/(0.5^5);
```

```
boy = Bp(2);  
b3y = (10*(Cp(2) - Bp(2)))/(0.5^3);  
b4y = (-15*(Cp(2) - Bp(2)))/(0.5^4);  
b5y = (6*(Cp(2) - Bp(2)))/(0.5^5);
```

```
coy = Cp(2);  
c3y = (10*(Dp(2) - Cp(2)))/(0.5^3);  
c4y = (-15*(Dp(2) - Cp(2)))/(0.5^4);  
c5y = (6*(Dp(2) - Cp(2)))/(0.5^5);
```

```
doy = Dp(2);  
d3y = (10*(Ap(2) - Dp(2)))/(0.5^3);  
d4y = (-15*(Ap(2) - Dp(2)))/(0.5^4);  
d5y = (6*(Ap(2) - Dp(2)))/(0.5^5);
```

```

for t = 0:dt:2 %tempo

%Calcular a posicao das duas articulacoes do meio

pL1x(counter) = L1*cos(Teta1);
pL1y(counter) = L1*sin(Teta1);

pL2x(counter) = pL1x(counter) + L2*cos(Teta1+Teta2);
pL2y(counter) = pL1y(counter) + L2*sin(Teta1+Teta2);

pL3x(counter) = pL2x(counter) + L3*cos(Teta1+Teta2+Teta3);
pL3y(counter) = pL2y(counter) + L3*sin(Teta1+Teta2+Teta3);

%Matriz de Inércia no instante t
M(:, :, counter) = [Iz1 + Iz2 + Iz3 + m1*lc1^2 + m2*(L1^2 + lc2^2 + 2*L1*lc2*cos(Teta2) +
m3*(L1^2 + L2^2 + lc3^2 + 2*L1*L2*cos(Teta2) + 2*L2*lc3*cos(Teta3) +
2*L1*lc3*cos(Teta2 + Teta3)))(Iz2 + Iz3 + m2*(lc2^2 + L1*lc2*cos(Teta2)) +
m3*(L2^2 + lc3^2 + 2*L2*lc3*cos(Teta3) + L1*L2*cos(Teta2) + L1*lc3*cos(Teta2 + Teta3)))
(Iz3 + m3*(lc3^2 + L2*lc3*cos(Teta3) + L1*lc3*cos(Teta2 + Teta3)));
(Iz2 + Iz3 + m2*(lc2^2 + L1*lc2*cos(Teta2)) + m3*(L2^2 + lc3^2 + 2*L2*lc3*cos(Teta3) +
L1*L2*cos(Teta2) + L1*lc3*cos(Teta2 + Teta3)))(Iz2 + Iz3 + m2*lc2^2 + m3*(L2^2 + lc3^2 +
2*L2*lc3*cos(Teta3)))(Iz3 + m3*(lc3^2 + L2*lc3*cos(Teta3)));
(Iz3 + m3*(lc3^2 + L2*lc3*cos(Teta3) + L1*lc3*cos(Teta2 + Teta3)))
(Iz3 + m3*(lc3^2 + L2*lc3*cos(Teta3)) (Iz3 + m3*lc3^2)];

%Matriz das forças centrífugas e de coriolis no instante t
C(:, :, counter) = [(-L1*L2*m3*sin(Teta2)*Teta2dot^2 - L1*lc2*m2*sin(Teta2)*Teta2dot^2 -
L2*lc3*m3*sin(Teta3)*Teta3dot^2 - L1*lc3*m3*sin(Teta2 + Teta3)*Teta2dot^2 -
L1*lc3*m3*sin(Teta2 + Teta3)*Teta3dot^2 - 2*L1*L2*m3*sin(Teta2)*Teta1dot*Teta2dot -
2*L1*lc2*m2*sin(Teta2)*Teta1dot*Teta2dot - 2*L2*lc3*m3*sin(Teta3)*Teta1dot*Teta3dot -
2*L2*lc3*m3*sin(Teta3)*Teta2dot*Teta3dot -
2*L1*lc3*m3*sin(Teta2 + Teta3)*Teta1dot*Teta2dot -
2*L1*lc3*m3*sin(Teta2 + Teta3)*Teta1dot*Teta3dot -
2*L1*lc3*m3*sin(Teta2 + Teta3)*Teta2dot*Teta3dot);
(L1*L2*m3*sin(Teta2)*Teta1dot^2 + L1*lc2*m2*sin(Teta2)*Teta1dot^2 -
L2*lc3*m3*sin(Teta3)*Teta3dot^2 +
L1*lc3*m3*sin(Teta2 + Teta3)*Teta1dot^2 - 2*L2*lc3*m3*sin(Teta3)*Teta1dot*Teta3dot -
2*L2*lc3*m3*sin(Teta3)*Teta2dot*Teta3dot);
(L2*lc3*m3*sin(Teta3)*Teta1dot^2 + L2*lc3*m3*sin(Teta3)*Teta2dot^2 +
L1*lc3*m3*sin(Teta2 + Teta3)*Teta1dot^2 + 2*L2*lc3*m3*sin(Teta3)*Teta1dot*Teta2dot)];

%Matriz das forças gravitacionais no instante t
G(:, :, counter) = [(g*lc3*m3*cos(Teta1 + Teta2 + Teta3) + g*L1*m2*cos(Teta1) +
g*L1*m3*cos(Teta1) + g*lc1*m1*cos(Teta1) + g*L2*m3*cos(Teta1 + Teta2) +
g*lc2*m2*cos(Teta1 + Teta2));
(g*lc3*m3*cos(Teta1 + Teta2 + Teta3) + g*L2*m3*cos(Teta1 + Teta2) +
g*lc2*m2*cos(Teta1 + Teta2));

```

```

(g*lc3*m3*cos(Teta1 + Teta2 + Teta3));

%Jacobiana no instante t
J(:, :, counter) = [-L1*sin(Teta1)-L2*sin(Teta1+Teta2)-L3*sin(Teta1+Teta2+Teta3),
-L2*sin(Teta1+Teta2)-L3*sin(Teta1+Teta2+Teta3), -L3*sin(Teta1+Teta2+Teta3);
L1*cos(Teta1)+L2*cos(Teta1+Teta2)+L3*cos(Teta1+Teta2+Teta3),
L2*cos(Teta1+Teta2)+L3*cos(Teta1+Teta2+Teta3), L3*cos(Teta1+Teta2+Teta3)];

%Derivada da Jacobiana no instante t
Jdot(:, :, counter) = [-Teta2dot*(L2*cos(Teta1 + Teta2) + L3*cos(Teta1 + Teta2 + Teta3)) -
Teta1dot*(L2*cos(Teta1 + Teta2) + L1*cos(Teta1) + L3*cos(Teta1 + Teta2 + Teta3)) -
L3*Teta3dot*cos(Teta1 + Teta2 + Teta3), - Teta1dot*(L2*cos(Teta1 + Teta2) +
L3*cos(Teta1 + Teta2 + Teta3)) - Teta2dot*(L2*cos(Teta1 + Teta2) +
L3*cos(Teta1 + Teta2 + Teta3)) - L3*Teta3dot*cos(Teta1 + Teta2 + Teta3), -
L3*Teta1dot*cos(Teta1 + Teta2 + Teta3) - L3*Teta2dot*cos(Teta1 + Teta2 + Teta3) -
L3*Teta3dot*cos(Teta1 + Teta2 + Teta3);
-Teta1dot*(L2*sin(Teta1 + Teta2) + L1*sin(Teta1) + L3*sin(Teta1 + Teta2 + Teta3)) -
Teta2dot*(L2*sin(Teta1 + Teta2) + L3*sin(Teta1 + Teta2 + Teta3)) -
L3*Teta3dot*sin(Teta1 + Teta2 + Teta3), - Teta1dot*(L2*sin(Teta1 + Teta2) +
L3*sin(Teta1 + Teta2 + Teta3)) - Teta2dot*(L2*sin(Teta1 + Teta2) +
L3*sin(Teta1 + Teta2 + Teta3)) - L3*Teta3dot*sin(Teta1 + Teta2 + Teta3), -
L3*Teta1dot*sin(Teta1 + Teta2 + Teta3) - L3*Teta2dot*sin(Teta1 + Teta2 + Teta3) -
L3*Teta3dot*sin(Teta1 + Teta2 + Teta3)];

%Matriz Jacobiana Pseudo-Inversa
JM(:, :, counter) = inv(M)*J'(inv(J*inv(M)*J'));

%posicao desejada do end-effector no tempo tk (x,y) em metros (para calcular o erro)

if t <= 0.5

%polinomios
pAx=(ao + a3*t^3 + a4*t^4 + a5*t^5);
pAy=(aoy+a3y*t^3+a4y*t^4+a5y*t^5);

%vetor posicao desejada
xe(:, :, counter) = [(pAx); (pAy)];
xex(counter) = (pAx); %localizacao da posicao em relacao a x
xey(counter) = (pAy); %localizacao da posicao em relacao a y

%derivada da posicao desejada do end-effector
xexdot(counter) = 3*a3*t^2+4*a4*t^3+5*a5*t^4;
xeydot(counter) = 3*a3y*t^2+4*a4y*t^3+5*a5y*t^4;
xexdot(:, :, counter) = [xexdot(counter); xeydot(counter)];

```

```

%segunda derivada da posicao desejada do end-effector
xex2dot(counter) = 6*a3*t+12*a4*t^2+20*a5*t^3;
xey2dot(counter) = 6*a3y*t+12*a4y*t^2+20*a5y*t^3;
xe2dot(:, :, counter) = [xex2dot(counter); xey2dot(counter)];

end

if t > 0.5 && t <= 1

%polinomios
pBx=(bo + b3*(t-0.5)^3 + b4*(t-0.5)^4 + b5*(t-0.5)^5);
pBy=(boy+b3y*(t-0.5)^3+b4y*(t-0.5)^4+b5y*(t-0.5)^5);

%vetor posicao desejada
xe(:, :, counter) = [pBx; pBy];
xex(counter) = pBx;           %localizacao da posicao em relacao a x
xey(counter) = pBy;           %localizacao da posicao em relacao a y

%derivada da posicao desejada do end-effector
xexdot(counter) = 3*b3*(t-0.5)^2 + 4*b4*(t-0.5)^3 + 5*b5*(t-0.5)^4;
xeydot(counter) = 3*b3y*(t-0.5)^2 + 4*b4y*(t-0.5)^3 + 5*b5y*(t-0.5)^4;
xedot(:, :, counter) = [xexdot(counter); xeydot(counter)];

%segunda derivada da posicao desejada do end-effector
xex2dot(counter) = 6*b3*(t-0.5) + 12*b4*(t-0.5)^2 + 20*b5*(t-0.5)^3;
xey2dot(counter) = 6*b3y*(t-0.5) + 12*b4y*(t-0.5)^2 + 20*b5y*(t-0.5)^3;
xe2dot(:, :, counter) = [xex2dot(counter); xey2dot(counter)];

end

if t > 1 && t <= 1.5

%polinomios
pCx=(co + c3*(t-1)^3 + c4*(t-1)^4 + c5*(t-1)^5);
pCy=(coy + c3y*(t-1)^3 + c4y*(t-1)^4 + c5y*(t-1)^5);

%vetor posicao desejada
xe(:, :, counter) = [pCx; pCy];
xex(counter) = pCx;           %localizacao da posicao em relacao a x
xey(counter) = pCy;           %localizacao da posicao em relacao a y

%derivada da posicao desejada do end-effector
xexdot(counter) = 3*c3*(t-1)^2 + 4*c4*(t-1)^3 + 5*c5*(t-1)^4;
xeydot(counter) = 3*c3y*(t-1)^2 + 4*c4y*(t-1)^3 + 5*c5y*(t-1)^4;
xedot(:, :, counter) = [xexdot(counter); xeydot(counter)];

```

```

    %segunda derivada da posicao desejada do end-effector
    xex2dot(counter) = 6*c3*(t-1) + 12*c4*(t-1)^2 + 20*c5*(t-1)^3;
    xey2dot(counter) = 6*c3y*(t-1) + 12*c4y*(t-1)^2 + 20*c5y*(t-1)^3;
    xe2dot(:, :, counter) = [xex2dot(counter); xey2dot(counter)];

end

if t > 1.5 && t <= 2

    %polinomios
    pDx=(do + d3*(t-1.5)^3 + d4*(t-1.5)^4 + d5*(t-1.5)^5);
    pDy=(doy + d3y*(t-1.5)^3 + d4y*(t-1.5)^4 + d5y*(t-1.5)^5);

    %vetor posicao desejada
    xex(counter) = pDx;           %localizacao da posicao em relacao a x
    xey(counter) = pDy;           %localizacao da posicao em relacao a y
    xe(:, :, counter) = [pDx; pDy];

    %derivada da posicao desejada do end-effector
    xexdot(counter) = 3*d3*(t-1.5)^2 + 4*d4*(t-1.5)^3 + 5*d5*(t-1.5)^4;
    xeydot(counter) = 3*d3y*(t-1.5)^2 + 4*d4y*(t-1.5)^3 + 5*d5y*(t-1.5)^4;
    xedot(:, :, counter) = [xexdot(counter); xeydot(counter)];

    %segunda derivada da posicao desejada do end-effector
    xex2dot(counter) = 6*d3*(t-1.5) + 12*d4*(t-1.5)^2 + 20*d5*(t-1.5)^3;
    xey2dot(counter) = 6*d3y*(t-1.5) + 12*d4y*(t-1.5)^2 + 20*d5y*(t-1.5)^3;
    xe2dot(:, :, counter) = [xex2dot(counter); xey2dot(counter)];

end

%calculo da aceleração das juntas no tempo tk

q0dot2 = (inv(M(:, :, counter))*(C(:, :, counter))); %%Nesse sem o termo da Gravidade
Tetadot2 = JM*(xd2dot - Jdot*Tetadot) - (I - (JM*J))*q0dot2;

Teta1dot2 = Tetadot2(1,1);
Teta2dot2 = Tetadot2(2,1);
Teta3dot2 = Tetadot2(3,1);

%calculo da velocidade das juntas no tempo tk

Teta1dot = Teta1dot + dt*Teta1dot2;   %velocidade de Teta1 no tempo tk
Teta2dot = Teta2dot + dt*Teta2dot2;   %velocidade de Teta2 no tempo tk
Teta3dot = Teta3dot + dt*Teta3dot2;   %velocidade de Teta3 no tempo tk

Tetadot = [Teta1dot; Teta2dot; Teta3dot];

```

```
%calcula da posição das juntas no tempo tk
```

```
Teta1 = Teta1 + dt*Teta1dot;    %orientação de Teta1 no tempo tk
Teta2 = Teta2 + dt*Teta2dot;    %orientação de Teta2 no tempo tk
Teta3 = Teta3 + dt*Teta3dot;    %orientação de Teta3 no tempo tk
```

```
%Calculando o Torque
```

```
Torque(:, :, counter) = M*Tetadot2 + C + G;
```

```
Tau1Result(counter) = Torque(1,1,counter);
```

```
Tau2Result(counter) = Torque(2,1,counter);
```

```
Tau3Result(counter) = Torque(3,1,counter);
```

```
%Resultados
```

```
Teta1Result(counter) = Teta1;    %vetor com a orientacao de Teta1 para cada tk
```

```
Teta2Result(counter) = Teta2;    %vetor com a orientacao de Teta2 para cada tk
```

```
Teta3Result(counter) = Teta3;    %vetor com a orientacao de Teta3 para cada tk
```

```
Teta1dotResult(counter) = Teta1dot;    %vetor com a velocidade de Teta1 para cada tk
```

```
Teta2dotResult(counter) = Teta2dot;    %vetor com a velocidade de Teta2 para cada tk
```

```
Teta3dotResult(counter) = Teta3dot;    %vetor com a velocidade de Teta3 para cada tk
```

```
Teta1dot2Result(counter) = Teta1dot2;    %vetor com a velocidade de Teta1 para cada tk
```

```
Teta2dot2Result(counter) = Teta2dot2;    %vetor com a velocidade de Teta2 para cada tk
```

```
Teta3dot2Result(counter) = Teta3dot2;    %vetor com a velocidade de Teta3 para cada tk
```

```
%Função Custo de Energia
```

```
E_custo(counter) = sqrt(Torque(:, :, counter)'*Torque(:, :, counter));
```

```
E_custo_media = mean(E_custo);
```

```
time(counter) = t;    %vetor com cada instante de tempo de simulação
```

```
counter = counter + 1;    %contador de iterações
```

```
end
```

```
%Normas infinitas de cada junta
```

```
Norma_inf = [norm(Tau1Result,inf) norm(Tau2Result,inf) norm(Tau3Result,inf)]
```

```
%Energia Total
```

```
E_custo_media
```

```
%%
```

```
%Plot a trajetoria do manipulador
```

```
for k = 1:1:counter
```

```
    if (mod(k,10)==0)    %plotar a cada 10 iteracoes
```

```
        figure(1)
```

```

    plot(xex,xey,'r','linewidth',1);
    hold on
    plot([0,pL1x(k)],[0,pL1y(k)],'b','linewidth',1);
    hold on
    plot([pL1x(k),pL2x(k)],[pL1y(k),pL2y(k)],'b','linewidth',1);
    plot([pL2x(k),pL3x(k)],[pL2y(k),pL3y(k)],'b','linewidth',1);
    plot(pL3x(k),pL3y(k),'bo','markersize',10);
    title ('Simulação do manipulador')
    xlabel ('[m]')
    ylabel ('[m]')
    xlim([-3 3]);ylim([-3 3]);
    grid on
    drawnow;
    hold off
end
end
%%
%plotar as orientacoes de cada junta ao longo da trajetoria
figure(2)
plot(time,Teta1Result)
hold on
plot(time,Teta2Result)
hold on
plot(time,Teta3Result)
legend('Teta1', 'Teta2', 'Teta3')
title ('Orientação das Juntas')
xlabel ('Tempo [s]')
ylabel ('Orientação das Juntas [rad]')

%plotar as velocidades de cada junta ao longo da trajetoria
figure(3)
plot(time,Teta1dotResult)
hold on
plot(time,Teta2dotResult)
hold on
plot(time,Teta3dotResult)
legend('Teta1dot', 'Teta2dot', 'Teta3dot')
title ('Velocidade das Juntas')
xlabel ('Tempo [s]')
ylabel ('[rad/s]')

%plotar as acelerações de cada junta ao longo da trajetoria
figure(4)
plot(time,Teta1dot2Result)
hold on
plot(time,Teta2dot2Result)
hold on

```

```
plot(time,Teta3dot2Result)
legend('Teta1dot2', 'Teta2dot2', 'Teta3dot2')
title ('Aceleração das Juntas')
xlabel ('Tempo [s]')
ylabel ('[rad/s]')

%plotar os torque de cada junta ao longo da trajetoria
figure(5)
plot(time,Tau1Result)
hold on
plot(time,Tau2Result)
hold on
plot(time,Tau3Result)
legend('Torque1', 'Torque2', 'Torque3')
title ('Torque das Juntas - Caso 1d')
xlabel ('Tempo [s]')
ylabel ('Torque das Juntas [Nm]')
```

APÊNDICE E – Exemplo de código utilizado na resolução da redundância do caso 2

```

clear all
close all
clc

%posicao inicial

phi = (pi-(pi/6));
%phi = (pi/6);

aaa=1.25;
bbb=1.5;
Ap = [-aaa,aaa];
Bp = [-aaa,bbb];
Cp = [-bbb,bbb];
Dp = [-bbb,aaa];
Teta_FC = Cinematica_Inversa([Ap phi]);

Teta1 = Teta_FC(1);    %converte p/ radianos
Teta2 = Teta_FC(2);    %converte p/ radianos
Teta3 = Teta_FC(3);    %converte p/ radianos

Teta1dot = 0;
Teta2dot = 0;
Teta3dot = 0;

Tetadot = [Teta1dot; Teta2dot; Teta3dot];

L1 = 1;    %comprimento do link 1 em metros
L2 = 1;    %comprimento do link 2 em metros
L3 = 1;    %comprimento do link 3 em metros
b = 0.1;    %altura da barra do link
lc1 = L1/2;    lc2 = L2/2;    lc3 = L3/2;    %posição dos cgs

m1 = 5;    m2 = 2;    m3 = 1;    %massa dos links
g = 9.81;    %aceleração da gravidade
Iz1 = (1/12)*m1*(L1^2 + b^2);
Iz2 = (1/12)*m2*(L2^2 + b^2);
Iz3 = (1/12)*m3*(L3^2 + b^2);    %momentos de inercia dos links

I = eye(3); %Matriz Identidade 3x3
counter = 1;    %contador de iteracoes
dt = 0.001; %step time
time = 0;    %tempo de simulacao
E_anterior = 0;

```

```
%calcula do polinomio
```

```
ao = Ap(1);  
a3 = (10*(Bp(1) - Ap(1)))/(0.5^3);  
a4 = (-15*(Bp(1) - Ap(1)))/(0.5^4);  
a5 = (6*(Bp(1) - Ap(1)))/(0.5^5);
```

```
bo = Bp(1);  
b3 = (10*(Cp(1) - Bp(1)))/(0.5^3);  
b4 = (-15*(Cp(1) - Bp(1)))/(0.5^4);  
b5 = (6*(Cp(1) - Bp(1)))/(0.5^5);
```

```
co = Cp(1);  
c3 = (10*(Dp(1) - Cp(1)))/(0.5^3);  
c4 = (-15*(Dp(1) - Cp(1)))/(0.5^4);  
c5 = (6*(Dp(1) - Cp(1)))/(0.5^5);
```

```
do = Dp(1);  
d3 = (10*(Ap(1) - Dp(1)))/(0.5^3);  
d4 = (-15*(Ap(1) - Dp(1)))/(0.5^4);  
d5 = (6*(Ap(1) - Dp(1)))/(0.5^5);
```

```
%AGORA PRA Y
```

```
aoy = Ap(2);  
a3y = (10*(Bp(2) - Ap(2)))/(0.5^3);  
a4y = (-15*(Bp(2) - Ap(2)))/(0.5^4);  
a5y = (6*(Bp(2) - Ap(2)))/(0.5^5);
```

```
boy = Bp(2);  
b3y = (10*(Cp(2) - Bp(2)))/(0.5^3);  
b4y = (-15*(Cp(2) - Bp(2)))/(0.5^4);  
b5y = (6*(Cp(2) - Bp(2)))/(0.5^5);
```

```
coy = Cp(2);  
c3y = (10*(Dp(2) - Cp(2)))/(0.5^3);  
c4y = (-15*(Dp(2) - Cp(2)))/(0.5^4);  
c5y = (6*(Dp(2) - Cp(2)))/(0.5^5);
```

```
doy = Dp(2);  
d3y = (10*(Ap(2) - Dp(2)))/(0.5^3);  
d4y = (-15*(Ap(2) - Dp(2)))/(0.5^4);  
d5y = (6*(Ap(2) - Dp(2)))/(0.5^5);
```

```

for t = 0:dt:2 %tempo

    %Calcular a posicao das duas articulacoes do meio

    pL1x(counter) = L1*cos(Teta1);
    pL1y(counter) = L1*sin(Teta1);

    pL2x(counter) = pL1x(counter) + L2*cos(Teta1+Teta2);
    pL2y(counter) = pL1y(counter) + L2*sin(Teta1+Teta2);

    pL3x(counter) = pL2x(counter) + L3*cos(Teta1+Teta2+Teta3);
    pL3y(counter) = pL2y(counter) + L3*sin(Teta1+Teta2+Teta3);

    %Matriz de Inércia no instante t
    M(:, :, counter) = [Iz1 + Iz2 + Iz3 + m1*lc1^2 + m2*(L1^2 + lc2^2 + 2*L1*lc2*cos(Teta2) +
    m3*(L1^2 + L2^2 + lc3^2 + 2*L1*L2*cos(Teta2) + 2*L2*lc3*cos(Teta3) +
    2*L1*lc3*cos(Teta2 + Teta3)))(Iz2 + Iz3 + m2*(lc2^2 + L1*lc2*cos(Teta2)) +
    m3*(L2^2 + lc3^2 + 2*L2*lc3*cos(Teta3) + L1*L2*cos(Teta2) + L1*lc3*cos(Teta2 + Teta3)))
    (Iz3 + m3*(lc3^2 + L2*lc3*cos(Teta3) + L1*lc3*cos(Teta2 + Teta3)));
    (Iz2 + Iz3 + m2*(lc2^2 + L1*lc2*cos(Teta2)) + m3*(L2^2 + lc3^2 + 2*L2*lc3*cos(Teta3) +
    L1*L2*cos(Teta2) + L1*lc3*cos(Teta2 + Teta3)))(Iz2 + Iz3 + m2*lc2^2 + m3*(L2^2 + lc3^2 +
    2*L2*lc3*cos(Teta3)))(Iz3 + m3*(lc3^2 + L2*lc3*cos(Teta3)));
    (Iz3 + m3*(lc3^2 + L2*lc3*cos(Teta3) + L1*lc3*cos(Teta2 + Teta3)))
    (Iz3 + m3*(lc3^2 + L2*lc3*cos(Teta3)) (Iz3 + m3*lc3^2)];

    %Matriz das forças centrífugas e de coriolis no instante t
    C(:, :, counter) = [(-L1*L2*m3*sin(Teta2)*Teta2dot^2 - L1*lc2*m2*sin(Teta2)*Teta2dot^2 -
    L2*lc3*m3*sin(Teta3)*Teta3dot^2 - L1*lc3*m3*sin(Teta2 + Teta3)*Teta2dot^2 -
    L1*lc3*m3*sin(Teta2 + Teta3)*Teta3dot^2 - 2*L1*L2*m3*sin(Teta2)*Teta1dot*Teta2dot -
    2*L1*lc2*m2*sin(Teta2)*Teta1dot*Teta2dot - 2*L2*lc3*m3*sin(Teta3)*Teta1dot*Teta3dot -
    2*L2*lc3*m3*sin(Teta3)*Teta2dot*Teta3dot -
    2*L1*lc3*m3*sin(Teta2 + Teta3)*Teta1dot*Teta2dot -
    2*L1*lc3*m3*sin(Teta2 + Teta3)*Teta1dot*Teta3dot -
    2*L1*lc3*m3*sin(Teta2 + Teta3)*Teta2dot*Teta3dot);
    (L1*L2*m3*sin(Teta2)*Teta1dot^2 + L1*lc2*m2*sin(Teta2)*Teta1dot^2 -
    L2*lc3*m3*sin(Teta3)*Teta3dot^2 +
    L1*lc3*m3*sin(Teta2 + Teta3)*Teta1dot^2 - 2*L2*lc3*m3*sin(Teta3)*Teta1dot*Teta3dot -
    2*L2*lc3*m3*sin(Teta3)*Teta2dot*Teta3dot);
    (L2*lc3*m3*sin(Teta3)*Teta1dot^2 + L2*lc3*m3*sin(Teta3)*Teta2dot^2 +
    L1*lc3*m3*sin(Teta2 + Teta3)*Teta1dot^2 + 2*L2*lc3*m3*sin(Teta3)*Teta1dot*Teta2dot)];

    %Matriz das forças gravitacionais no instante t
    G(:, :, counter) = [(g*lc3*m3*cos(Teta1 + Teta2 + Teta3) + g*L1*m2*cos(Teta1) +
    g*L1*m3*cos(Teta1) + g*lc1*m1*cos(Teta1) + g*L2*m3*cos(Teta1 + Teta2) +
    g*lc2*m2*cos(Teta1 + Teta2));
    (g*lc3*m3*cos(Teta1 + Teta2 + Teta3) + g*L2*m3*cos(Teta1 + Teta2) +
    g*lc2*m2*cos(Teta1 + Teta2));

```

```

(g*lc3*m3*cos(Teta1 + Teta2 + Teta3));

%Jacobiana no instante t
J(:, :, counter) = [-L1*sin(Teta1)-L2*sin(Teta1+Teta2)-L3*sin(Teta1+Teta2+Teta3),
-L2*sin(Teta1+Teta2)-L3*sin(Teta1+Teta2+Teta3), -L3*sin(Teta1+Teta2+Teta3);
L1*cos(Teta1)+L2*cos(Teta1+Teta2)+L3*cos(Teta1+Teta2+Teta3),
L2*cos(Teta1+Teta2)+L3*cos(Teta1+Teta2+Teta3), L3*cos(Teta1+Teta2+Teta3)];

%Derivada da Jacobiana no instante t
Jdot(:, :, counter) = [-Teta2dot*(L2*cos(Teta1 + Teta2) + L3*cos(Teta1 + Teta2 + Teta3)) -
Teta1dot*(L2*cos(Teta1 + Teta2) + L1*cos(Teta1) + L3*cos(Teta1 + Teta2 + Teta3)) -
L3*Teta3dot*cos(Teta1 + Teta2 + Teta3), - Teta1dot*(L2*cos(Teta1 + Teta2) +
L3*cos(Teta1 + Teta2 + Teta3)) - Teta2dot*(L2*cos(Teta1 + Teta2) +
L3*cos(Teta1 + Teta2 + Teta3)) - L3*Teta3dot*cos(Teta1 + Teta2 + Teta3), -
L3*Teta1dot*cos(Teta1 + Teta2 + Teta3) - L3*Teta2dot*cos(Teta1 + Teta2 + Teta3) -
L3*Teta3dot*cos(Teta1 + Teta2 + Teta3);
-Teta1dot*(L2*sin(Teta1 + Teta2) + L1*sin(Teta1) + L3*sin(Teta1 + Teta2 + Teta3)) -
Teta2dot*(L2*sin(Teta1 + Teta2) + L3*sin(Teta1 + Teta2 + Teta3)) -
L3*Teta3dot*sin(Teta1 + Teta2 + Teta3), - Teta1dot*(L2*sin(Teta1 + Teta2) +
L3*sin(Teta1 + Teta2 + Teta3)) - Teta2dot*(L2*sin(Teta1 + Teta2) +
L3*sin(Teta1 + Teta2 + Teta3)) - L3*Teta3dot*sin(Teta1 + Teta2 + Teta3), -
L3*Teta1dot*sin(Teta1 + Teta2 + Teta3) - L3*Teta2dot*sin(Teta1 + Teta2 + Teta3) -
L3*Teta3dot*sin(Teta1 + Teta2 + Teta3)];

%Matriz Jacobiana Pseudo-Inversa
M2(:, :, counter) = M(:, :, counter)*M(:, :, counter);
JM2(:, :, counter) = inv(M2)*J'(inv(J*inv(M2)*J'));

%posicao desejada do end-effector no tempo tk (x,y) em metros (para calcular o erro)

if t <= 0.5

%polinomios
pAx=(ao + a3*t^3 + a4*t^4 + a5*t^5);
pAy=(aoy+a3y*t^3+a4y*t^4+a5y*t^5);

%vetor posicao desejada
xe(:, :, counter) = [(pAx); (pAy)];
xex(counter) = (pAx); %localizacao da posicao em relacao a x
xey(counter) = (pAy); %localizacao da posicao em relacao a y

%derivada da posicao desejada do end-effector
xexdot(counter) = 3*a3*t^2+4*a4*t^3+5*a5*t^4;

```

```

xeydot(counter) = 3*a3y*t^2+4*a4y*t^3+5*a5y*t^4;
xedot(:, :, counter) = [xexdot(counter); xeydot(counter)];

%segunda derivada da posicao desejada do end-effector
xex2dot(counter) = 6*a3*t+12*a4*t^2+20*a5*t^3;
xey2dot(counter) = 6*a3y*t+12*a4y*t^2+20*a5y*t^3;
xe2dot(:, :, counter) = [xex2dot(counter); xey2dot(counter)];

end

if t > 0.5 && t <= 1

    %polinomios
    pBx=(bo + b3*(t-0.5)^3 + b4*(t-0.5)^4 + b5*(t-0.5)^5);
    pBy=(boy+b3y*(t-0.5)^3+b4y*(t-0.5)^4+b5y*(t-0.5)^5);

    %vetor posicao desejada
    xe(:, :, counter) = [pBx; pBy];
    xex(counter) = pBx;           %localizacao da posicao em relacao a x
    xey(counter) = pBy;           %localizacao da posicao em relacao a y

    %derivada da posicao desejada do end-effector
    xexdot(counter) = 3*b3*(t-0.5)^2 + 4*b4*(t-0.5)^3 + 5*b5*(t-0.5)^4;
    xeydot(counter) = 3*b3y*(t-0.5)^2 + 4*b4y*(t-0.5)^3 + 5*b5y*(t-0.5)^4;
    xedot(:, :, counter) = [xexdot(counter); xeydot(counter)];

    %segunda derivada da posicao desejada do end-effector
    xex2dot(counter) = 6*b3*(t-0.5) + 12*b4*(t-0.5)^2 + 20*b5*(t-0.5)^3;
    xey2dot(counter) = 6*b3y*(t-0.5) + 12*b4y*(t-0.5)^2 + 20*b5y*(t-0.5)^3;
    xe2dot(:, :, counter) = [xex2dot(counter); xey2dot(counter)];

end

if t > 1 && t <= 1.5

    %polinomios
    pCx=(co + c3*(t-1)^3 + c4*(t-1)^4 + c5*(t-1)^5);
    pCy=(coy + c3y*(t-1)^3 + c4y*(t-1)^4 + c5y*(t-1)^5);

    %vetor posicao desejada
    xe(:, :, counter) = [pCx; pCy];
    xex(counter) = pCx;           %localizacao da posicao em relacao a x
    xey(counter) = pCy;           %localizacao da posicao em relacao a y

    %derivada da posicao desejada do end-effector
    xexdot(counter) = 3*c3*(t-1)^2 + 4*c4*(t-1)^3 + 5*c5*(t-1)^4;
    xeydot(counter) = 3*c3y*(t-1)^2 + 4*c4y*(t-1)^3 + 5*c5y*(t-1)^4;

```

```

xedot(:, :, counter) = [xexdot(counter); xeydot(counter)];

%segunda derivada da posicao desejada do end-effector
xex2dot(counter) = 6*c3*(t-1) + 12*c4*(t-1)^2 + 20*c5*(t-1)^3;
xey2dot(counter) = 6*c3y*(t-1) + 12*c4y*(t-1)^2 + 20*c5y*(t-1)^3;
xe2dot(:, :, counter) = [xex2dot(counter); xey2dot(counter)];

end

if t > 1.5 && t <= 2

%polinomios
pDx=(do + d3*(t-1.5)^3 + d4*(t-1.5)^4 + d5*(t-1.5)^5);
pDy=(doy + d3y*(t-1.5)^3 + d4y*(t-1.5)^4 + d5y*(t-1.5)^5);

%vetor posicao desejada
xex(counter) = pDx; %localizacao da posicao em relacao a x
xey(counter) = pDy; %localizacao da posicao em relacao a y
xe(:, :, counter) = [pDx; pDy];

%derivada da posicao desejada do end-effector
xexdot(counter) = 3*d3*(t-1.5)^2 + 4*d4*(t-1.5)^3 + 5*d5*(t-1.5)^4;
xeydot(counter) = 3*d3y*(t-1.5)^2 + 4*d4y*(t-1.5)^3 + 5*d5y*(t-1.5)^4;
xedot(:, :, counter) = [xexdot(counter); xeydot(counter)];

%segunda derivada da posicao desejada do end-effector
xex2dot(counter) = 6*d3*(t-1.5) + 12*d4*(t-1.5)^2 + 20*d5*(t-1.5)^3;
xey2dot(counter) = 6*d3y*(t-1.5) + 12*d4y*(t-1.5)^2 + 20*d5y*(t-1.5)^3;
xe2dot(:, :, counter) = [xex2dot(counter); xey2dot(counter)];

end

%calculo da aceleração das juntas no tempo tk
q0dot2 = (inv(M(:, :, counter))*(C(:, :, counter))); %%Nesse sem o termo da Gravidade
Tetadot2 = JM2*(xd2dot - Jdot*Tetadot) - (I - (JM2*J))*q0dot2;

Teta1dot2 = Tetadot2(1,1);
Teta2dot2 = Tetadot2(2,1);
Teta3dot2 = Tetadot2(3,1);

%calculo da velocidade das juntas no tempo tk
Teta1dot = Teta1dot + dt*Teta1dot2; %velocidade de Teta1 no tempo tk
Teta2dot = Teta2dot + dt*Teta2dot2; %velocidade de Teta2 no tempo tk
Teta3dot = Teta3dot + dt*Teta3dot2; %velocidade de Teta3 no tempo tk

```

```

Tetadot = [Teta1dot; Teta2dot; Teta3dot];

%calculo da posição das juntas no tempo tk

Teta1 = Teta1 + dt*Teta1dot;    %orientação de Teta1 no tempo tk
Teta2 = Teta2 + dt*Teta2dot;    %orientação de Teta2 no tempo tk
Teta3 = Teta3 + dt*Teta3dot;    %orientação de Teta3 no tempo tk

%Calculando o Torque

Torque(:, :, counter) = M*Tetadot2 + C + G;

Tau1Result(counter) = Torque(1,1,counter);
Tau2Result(counter) = Torque(2,1,counter);
Tau3Result(counter) = Torque(3,1,counter);

%Resultados

Teta1Result(counter) = Teta1;    %vetor com a orientacao de Teta1 para cada tk
Teta2Result(counter) = Teta2;    %vetor com a orientacao de Teta2 para cada tk
Teta3Result(counter) = Teta3;    %vetor com a orientacao de Teta3 para cada tk
Teta1dotResult(counter) = Teta1dot;    %vetor com a velocidade de Teta1 para cada tk
Teta2dotResult(counter) = Teta2dot;    %vetor com a velocidade de Teta2 para cada tk
Teta3dotResult(counter) = Teta3dot;    %vetor com a velocidade de Teta3 para cada tk
Teta1dot2Result(counter) = Teta1dot2;    %vetor com a velocidade de Teta1 para cada tk
Teta2dot2Result(counter) = Teta2dot2;    %vetor com a velocidade de Teta2 para cada tk
Teta3dot2Result(counter) = Teta3dot2;    %vetor com a velocidade de Teta3 para cada tk

%Função Custo de Energia

E_custo(counter) = sqrt(Torque(:, :, counter)'*Torque(:, :, counter));
E_custo_media = mean(E_custo);

time(counter) = t;    %vetor com cada instante de tempo de simulação
counter = counter + 1;    %contador de iterações

end

%Normas infinitas de cada junta
Norma_inf = [norm(Tau1Result,inf) norm(Tau2Result,inf) norm(Tau3Result,inf)]

%Energia Total
E_custo_media

%%
%Plot a trajetoria do manipulador
for k = 1:1:counter

```

```

    if (mod(k,10)==0) %plotar a cada 10 iteracoes
        figure(1)
        plot(xex,xey,'r','linewidth',1);
        hold on
        plot([0,pL1x(k)],[0,pL1y(k)],'b','linewidth',1);
        hold on
        plot([pL1x(k),pL2x(k)],[pL1y(k),pL2y(k)],'b','linewidth',1);
        plot([pL2x(k),pL3x(k)],[pL2y(k),pL3y(k)],'b','linewidth',1);
        plot(pL3x(k),pL3y(k),'bo','markersize',10);
        title ('Simulação do manipulador')
        xlabel ('[m]')
        ylabel ('[m]')
        xlim([-3 3]);ylim([-3 3]);
        grid on
        drawnow;
        hold off
    end
end
%%
%plotar as orientacoes de cada junta ao longo da trajetoria
figure(2)
plot(time,Teta1Result)
hold on
plot(time,Teta2Result)
hold on
plot(time,Teta3Result)
legend('Teta1', 'Teta2', 'Teta3')
title ('Orientação das Juntas')
xlabel ('Tempo [s]')
ylabel ('Orientação das Juntas [rad]')

%plotar as velocidades de cada junta ao longo da trajetoria
figure(3)
plot(time,Teta1dotResult)
hold on
plot(time,Teta2dotResult)
hold on
plot(time,Teta3dotResult)
legend('Teta1dot', 'Teta2dot', 'Teta3dot')
title ('Velocidade das Juntas')
xlabel ('Tempo [s]')
ylabel ('[rad/s]')

%plotar as acelerações de cada junta ao longo da trajetoria
figure(4)
plot(time,Teta1dot2Result)
hold on

```

```
plot(time,Teta2dot2Result)
hold on
plot(time,Teta3dot2Result)
legend('Teta1dot2', 'Teta2dot2', 'Teta3dot2')
title ('Aceleração das Juntas')
xlabel ('Tempo [s]')
ylabel ('[rad/s]')

%plotar os torque de cada junta ao longo da trajetoria
figure(5)
plot(time,Tau1Result)
hold on
plot(time,Tau2Result)
hold on
plot(time,Tau3Result)
legend('Torque1', 'Torque2', 'Torque3')
title ('Torque das Juntas - Caso 2d')
xlabel ('Tempo [s]')
ylabel ('Torque das Juntas [Nm]')
```