

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

GUSTAVO CORRÊA DE OLIVEIRA

**OTIMIZAÇÃO DO DESEMPENHO DINÂMICO DE UM
MANIPULADOR ROBÓTICO PLANAR EM SÉRIE COM
REDUNDÂNCIA NO ESPAÇO DAS TAREFAS**

SÃO CARLOS
2022

GUSTAVO CORRÊA DE OLIVEIRA

**OTIMIZAÇÃO DO DESEMPENHO DINÂMICO DE UM
MANIPULADOR ROBÓTICO PLANAR EM SÉRIE COM
REDUNDÂNCIA NO ESPAÇO DAS TAREFAS**

Trabalho de Conclusão de Curso apresentado ao 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
2022



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 9/2022/CCEMec/CCET

Graduação: Defesa Pública de Trabalho de Conclusão de Curso

Folha Aprovação (GDP-TCC-FA)

FOLHA DE APROVAÇÃO

GUSTAVO CORREA DE OLIVEIRA

OTIMIZAÇÃO DO DESEMPENHO DINÂMICO DE UM MANIPULADOR ROBÓTICO PLANAR EM SÉRIE COM REDUNDÂNCIA NO ESPAÇO DAS TAREFAS

Trabalho de Conclusão de Curso

Universidade Federal de São Carlos – Campus São Carlos

São Carlos, 05 de abril de 2022

ASSINATURAS E CIÊNCIAS

| Cargo/Função | Nome Completo |
|-------------------|---|
| Orientador | João Vitor de Carvalho Fontes |
| Membro da Banca 1 | Luis Antonio Oliveira Araújo |
| Membro da Banca 2 | Tatiana de Figueiredo Pereira Alves Taveira Pazelli |



Documento assinado eletronicamente por **João Vitor de Carvalho Fontes, Docente**, em 05/04/2022, às 17:53, 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 Araújo, Docente**, em 05/04/2022, às 17:54, 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 **Tatiana de Figueiredo Pereira Alves Taveira Pazelli, Professor(a)**, em 05/04/2022, às 17:54, 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 **0644351** e o código CRC **11EC036D**.

Referência: Caso responda a este documento, indicar expressamente o Processo nº 23112.009211/2022-60

SEI nº 0644351

AGRADECIMENTOS

Primeiramente agradeço aos meus pais, José Benedito e Elisabete, por todo o esforço incansável e apoio familiar que permitiu que eu realizasse essa graduação que tanto almejei e à minha irmã Juliana pelo companheirismo em todos os momentos dessa jornada.

Aos meus caros amigos de graduação, Alexandre Duarte, Bruno Corrêa, Cícero Zanette, Gabriel Blanco, Gabriel Saretti, Guilherme Juvenário, Katylin Meira e Nathan Furtado pelas intermináveis horas de estudos, trabalhos em grupo e aos melhores momentos de descontração e convívio.

Agradeço aos grandes amigos Caio de Sordi, Luis Miguel Mansano, Letícia Zanucoli e Camila Vieira por todo o apoio, conversas, desabafos e diversos momentos felizes compartilhados.

Aos professores do departamento de Engenharia Mecânica, em especial ao Alexandre Tácito Malavolta pela confiança e introdução aos estudos de mecânica computacional e Luis Antônio Oliveira Araújo pela oportunidade de realizarmos uma iniciação científica que me permitiu entrar na área da robótica.

Por último, agradeço ao professor João Vitor de Carvalho Fontes pela primorosa orientação durante o desenvolvimento desse trabalho, com paciência e determinação para que eu entendesse os pontos em que apresentava dúvidas e por sempre indicar o caminho para que pudesse enfrentar os desafios desse projeto.

”Estamos no início dos tempos para a raça humana. Não é descabido que enfrentemos problemas. Mas há dezenas de milhares de anos no futuro. Nossa responsabilidade é fazer o que podemos, aprender o que podemos, aprimorar as soluções e passá-las adiante.”

(Richard Feynman)

RESUMO

Manipuladores robóticos estão cada vez mais presentes nas atividades industriais da atualidade. Eles são utilizados em tarefas como: pintura, operação de furação e soldagem, movimentação de cargas etc. Apesar desses robôs serem amplamente empregados, muitas vezes suas tarefas não são executadas de maneira otimizada. Existem diferentes características que podem ser utilizadas para definir a otimização de um manipulador, como manipulabilidade, torque, energia, condicionamento, entre outras. A redundância no espaço das tarefas ocorre quando o efetuador final possui mais graus de liberdade do que a tarefa necessita, 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. Existem dois tipos de resolução: local e global. A local observa o gradiente de uma função custo a cada instante de tempo e decide as posições das juntas a partir disso. Já a global observa uma função custo ao longo de todo o período de execução da tarefa para definir essas posições. Sendo assim, esse trabalho tem como objetivo estudar um manipulador robótico planar em série utilizando redundância no espaço das tarefas para otimizar seu desempenho dinâmico a fim de comparar os diversos métodos de resolução da redundância. Os métodos local e global foram comparados entre si e com o desempenho de um manipulador não redundante. Além dessa análise, foi estudado o impacto de outros fatores na otimização do desempenho dinâmico, como a orientação inicial do efetuador final e a utilização de um fator de ponderação das forças gravitacionais nas equações dinâmicas. A implementação do manipulador, simulação e otimização foram feitas utilizando o *software* MATLAB. Os resultados obtidos apontam que a resolução da redundância pelo método local apresentou um melhor desempenho dinâmico e que a otimização conjunta da orientação inicial e do fator de ponderação das forças gravitacionais colabora para isso.

Palavras-chave: Manipulador robótico. Resolução da redundância. Otimização.

ABSTRACT

Robotic manipulators are increasingly present in current industrial activities. They are used in tasks such as: painting, drilling and welding operations, material handling, etc. Although these robots are widely used, their tasks are often not performed optimally. There are different characteristics that can be used to define the optimization of a manipulator, such as manipulability, torque, energy, conditioning, among others. Task space redundancy occurs when the end-effector has more degrees of freedom than the task needs, so it is possible to optimize performance using the excess degrees of freedom. This excess of degrees of freedom defines a problem with infinite solutions for the robot joint positions. Defining these positions through an optimization process is called redundancy resolution. There are two types of resolution: local and global. The local observes the gradient of a cost function at each instant of time and decides the joint positions from that. The global one observes a cost function throughout the task execution period to define these positions. Therefore, this work aims to study a planar serial robotic manipulator using task space redundancy to optimize its dynamic performance in order to compare the different redundancy resolution methods. The local and global methods were compared with each other and with the performance of a non-redundant robotic manipulator. In addition to this analysis, the impact of other factors on the optimization of dynamic performance was studied, such as the initial orientation of the end-effector and the use of a weighting factor for gravitational forces in the dynamic equations. The manipulator implementation, simulation and optimization were done using the *software* MATLAB. The results obtained indicate that the resolution of the redundancy by the local method presented a better dynamic performance and that the joint optimization of the initial orientation and of the weighting factor of the gravitational forces collaborates for this.

Keywords: Robotic manipulator. Redundancy resolution. Optimization.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Manipuladores seriais em cadeia cinemática aberta: (a) modelo KR 6 R1820 HP; (b) modelo KR 240 R3330. | 17 |
| Figura 2 – Representação do efetuador final e Robô ABB IRB 140 com a ferramenta de furação utilizado no trabalho de Zanchettin <i>et al.</i> (2011). | 19 |
| Figura 3 – Estrutura dos manipuladores robóticos em série (a) e paralelo (b). | 22 |
| Figura 4 – Espaço nulo dos manipuladores cinematicamente redundantes. | 24 |
| Figura 5 – Mínimos local e global de uma função não convexa em um determinado domínio. | 26 |
| Figura 6 – Modelo do manipulador planar serial. | 29 |
| Figura 7 – Orientações inicial e final ótimas do efetuador em um segmento da trajetória. | 41 |
| Figura 8 – Caminho percorrido pelo efetuador final do manipulador. | 45 |
| Figura 9 – Média dos torques das juntas do manipulador para os casos mais otimizados da resolução local (1h e 2h), global (3 e 4) e do manipulador não redundante (5a, 5b e 5c). | 48 |
| Figura 10 – Média dos torques das juntas do manipulador para os casos da resolução local (1a, 1c, 1d, 1e, 1f, 2a, 2c, 2d, 2e, 2f) e do manipulador não redundante (5a, 5b e 5c). | 49 |
| Figura 11 – Média dos torques das juntas do manipulador para os casos mais otimizados da resolução local (1g, 1h, 2g e 2h), para comparação da otimização de k e α | 51 |
| Figura 12 – Torque das juntas do manipulador redundante do caso 1a. | 57 |
| Figura 13 – Torque das juntas do manipulador redundante do caso 1b. | 58 |
| Figura 14 – Torque das juntas do manipulador redundante do caso 1c. | 59 |
| Figura 15 – Torque das juntas do manipulador redundante do caso 1d. | 60 |
| Figura 16 – Torque das juntas do manipulador redundante do caso 1e. | 61 |
| Figura 17 – Torque das juntas do manipulador redundante do caso 1f. | 62 |
| Figura 18 – Torque das juntas do manipulador redundante do caso 1g. | 63 |
| Figura 19 – Torque das juntas do manipulador redundante do caso 1h. | 64 |
| Figura 20 – Torque das juntas do manipulador redundante do caso 2a. | 65 |
| Figura 21 – Torque das juntas do manipulador redundante do caso 2b. | 66 |
| Figura 22 – Torque das juntas do manipulador redundante do caso 2c. | 67 |
| Figura 23 – Torque das juntas do manipulador redundante do caso 2d. | 68 |
| Figura 24 – Torque das juntas do manipulador redundante do caso 2e. | 69 |
| Figura 25 – Torque das juntas do manipulador redundante do caso 2f. | 70 |
| Figura 26 – Torque das juntas do manipulador redundante do caso 2g. | 71 |

| | |
|---|----|
| Figura 27 – Torque das juntas do manipulador redundante do caso 2h. | 72 |
| Figura 28 – Torque das juntas do manipulador redundante do caso 3. | 73 |
| Figura 29 – Torque das juntas do manipulador redundante do caso 4. | 74 |
| Figura 30 – Torque das juntas do manipulador não redundante do caso 5a. | 75 |
| Figura 31 – Torque das juntas do manipulador não redundante do caso 5b. | 76 |
| Figura 32 – Torque das juntas do manipulador não redundante do caso 5c. | 77 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 – Parâmetros DH para o manipulador serial. | 30 |
| Tabela 2 – Parâmetros utilizados no modelo do manipulador serial. | 44 |
| Tabela 3 – Dados da trajetória do efetuador final nas simulações. | 45 |
| Tabela 4 – Valores da função E e dos torques máximos de cada junta $\ \tau\ _\infty$ para os cinco casos simulados e suas derivações. | 46 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|-----|----------------------------------|
| GDL | Grau de liberdade |
| GA | Genetic algorithm |
| SQP | Sequential quadratic programming |
| 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 |
| α | 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 |
| $\dot{\alpha}$ | 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 |
| $\ddot{\alpha}$ | 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 |
| α_{1A} | Orientação inicial do efetuador final no vértice 1 da trajetória A |
| k | Fator de ponderação das forças gravitacionais |
| E | Média dos torques das juntas do manipulador |

SUMÁRIO

| | |
|---|-----------|
| 1 – INTRODUÇÃO | 17 |
| 1.1 Objetivos | 20 |
| 1.2 Estrutura do texto | 20 |
| 2 – FUNDAMENTAÇÃO TEÓRICA | 21 |
| 2.1 Características dos manipuladores robóticos | 21 |
| 2.1.1 Mobilidade | 21 |
| 2.2 Redundância de manipuladores robóticos | 22 |
| 2.2.1 Redundância cinemática da tarefa | 23 |
| 2.3 Resolução da redundância e otimização | 25 |
| 2.3.1 Otimização | 25 |
| 2.4 Modelagem de um manipulador planar em série | 28 |
| 2.4.1 Parâmetros de Denavit-Hartenberg | 28 |
| 2.4.2 Cinemática direta | 29 |
| 2.4.3 Cinemática inversa | 31 |
| 2.4.4 Matriz Jacobiana | 32 |
| 2.4.5 Cinemática direta diferencial | 33 |
| 2.4.6 Cinemática inversa diferencial | 33 |
| 2.4.7 Dinâmica | 34 |
| 2.5 Planejamento da trajetória do efetuador final | 35 |
| 2.6 Considerações finais do capítulo | 37 |
| 3 – RESOLUÇÃO DA REDUNDÂNCIA | 38 |
| 3.1 Resolução da redundância pelo método local | 38 |
| 3.1.1 Caso 1 - Resolução local ponderada | 38 |
| 3.1.2 Caso 2 - Resolução local não ponderada | 40 |
| 3.2 Resolução da redundância pelo método global | 40 |
| 3.2.1 Caso 3 - Resolução global ponderada | 41 |
| 3.2.2 Caso 4 - Resolução global não ponderada | 42 |
| 3.3 Caso 5 - Manipulador não redundante | 42 |
| 3.4 Método de comparação dos resultados | 43 |
| 3.5 Considerações finais do capítulo | 43 |
| 4 – RESULTADOS | 44 |
| 4.1 Características do manipulador | 44 |
| 4.2 Trajetória do efetuador final | 44 |

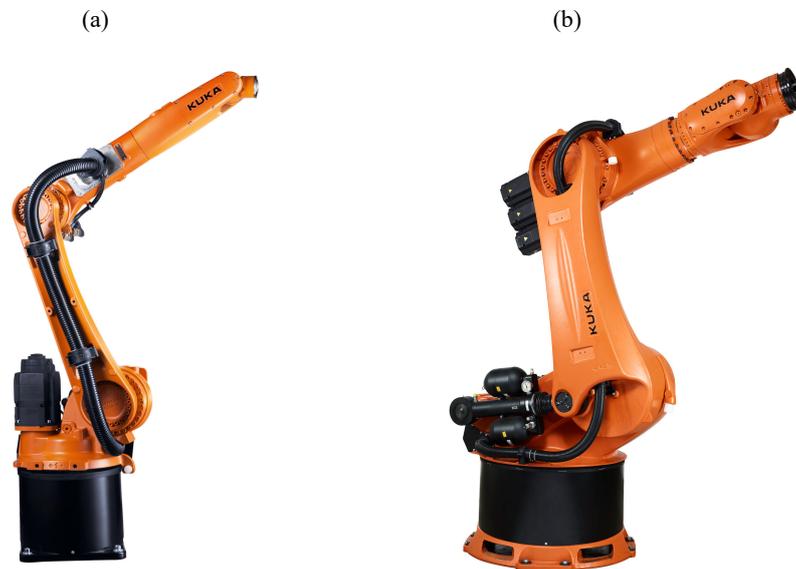
| | | |
|----------|--|-----------|
| 4.3 | Resultados simulados | 45 |
| 4.4 | Comparações dos resultados | 47 |
| 4.4.1 | Comparação entre a resolução da redundância utilizando o método local e global | 47 |
| 4.4.2 | Análise do impacto de k e α_{1A} na resolução da redundância | 49 |
| 4.5 | Considerações finais do capítulo | 51 |
| 5 | CONCLUSÃO | 52 |
| 5.1 | Trabalhos futuros | 52 |
| | REFERÊNCIAS | 54 |
| | | |
| | Apêndices | 56 |
| | APÊNDICE A –Gráficos dos torques das juntas do manipulador redundante do caso 1 | 57 |
| | APÊNDICE B –Gráficos dos torques das juntas do manipulador redundante do caso 2 | 65 |
| | APÊNDICE C –Gráfico dos torques das juntas do manipulador redundante do caso 3 | 73 |
| | APÊNDICE D –Gráfico dos torques das juntas do manipulador redundante do caso 4 | 74 |
| | APÊNDICE E –Gráficos dos torques das juntas do manipulador não redundante do caso 5 | 75 |
| | APÊNDICE F –Exemplo do código utilizado na resolução da redundância do caso 1h | 78 |
| | APÊNDICE G –Exemplo do código para cálculo da cinemática inversa | 85 |
| | APÊNDICE H –Exemplo do código da função <i>ga</i> para o caso 1h | 86 |
| | APÊNDICE I –Exemplo do código da função custo para o caso 1h | 87 |

1 INTRODUÇÃO

Atualmente os manipuladores robóticos são utilizados em diversas áreas da indústria realizando tarefas como: pintura, soldagem, inspeção, movimentação de cargas, montagem etc. Dada a sua ampla utilização, tais manipuladores são objetos de estudos em diversas instituições acadêmicas ao redor do mundo.

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). Exemplos de manipuladores seriais podem ser vistos na Figura 1.

Figura 1 – Manipuladores seriais em cadeia cinemática aberta: (a) modelo KR 6 R1820 HP; (b) modelo KR 240 R3330.



Fonte: KUKA Roboter do Brasil (2020).

Na Figura 1a é mostrado o modelo KR 6 R1820 HP fabricado pela empresa KUKA Industrial Robots. Sua capacidade de carga nominal é de 6 kg. Esse modelo é utilizado para soldagem, aplicação de colas e pintura. Na Figura 1b é apresentado o modelo KR 240 R3330 da mesma empresa. Este apresenta maior capacidade de carga nominal podendo elevar até 240 kg. É um modelo utilizado pela indústria automobilística em atividades de montagem e manuseio de peças e equipamentos (KUKA Roboter do Brasil, 2020).

Os graus de liberdade (GDL) de um manipulador são 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 é chamado de manipulador redundante (SICILIANO *et al.*, 2009).

Segundo o trabalho de Conkur e Buckingham (1997), existem muitas definições para redundância, sendo um termo utilizado para situações diferentes, mas que apresentam uma certa relação. Do ponto de vista da mecânica, a redundância pode ser de dois tipos: cinemática e 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.

A redundância no espaço das tarefas é um caso de redundância cinemática. Nessa redundância, conforme o trabalho de Fontes e Silva (2017), 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.

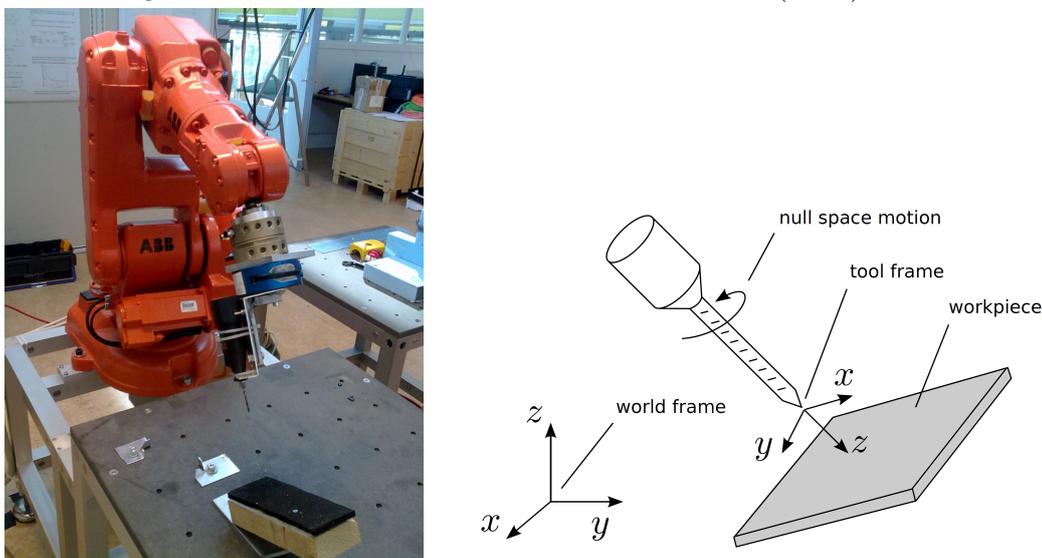
A resolução redundância de manipuladores robóticos está presente em diversos trabalhos de pesquisa. Zanchettin *et al.* (2011) apresentam uma otimização dos torques das juntas de um manipulador industrial redundante que realiza operações de furação, conforme apresentado na Figura 2. Nesse trabalho, é utilizado um manipulador de 6 GDL para realizar uma tarefa que necessita de apenas 5 GDL para posicionar a ferramenta de furação. A resolução otimizada da redundância foi simulada e verificada com o manipulador real.

Conforme Nenchev (1989), 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. De acordo com Ahuactzin e Gupta (1999), a resolução pode ser feita de duas formas: local ou global. A local executa a otimização observando o gradiente de uma função custo a cada instante de tempo. A global é realizada observando a função custo durante o tempo completo de execução de uma tarefa.

Na literatura é possível encontrar diversos trabalhos sobre os métodos de otimização local e global para a resolução de redundância em manipuladores robóticos.

Kazerounian e Wang (1988) apresentam um trabalho com o objetivo de minimizar a norma da velocidade das juntas e a energia cinética de um manipulador durante a execução de uma tarefa utilizando a resolução global. Já no trabalho de Nedungadi e Kazerounian (1989) é apresentada uma resolução local, com características de global, para a otimização dos torques das juntas. A solução proposta é comparada em termos de

Figura 2 – Representação do efetuador final e Robô ABB IRB 140 com a ferramenta de furação utilizado no trabalho de Zanchettin *et al.* (2011).



Fonte: Zanchettin *et al.* (2011).

estabilidade e torque com outros cinco diferentes métodos de resolução da redundância de manipuladores ao nível de aceleração.

Tringali e Cocuzza (2020) propõem um novo método global de otimização para um manipulador redundante em que primeiramente são definidas posições iniciais mais promissoras do robô para que sejam utilizadas como entrada no algoritmo de otimização. Foi proposto no trabalho a minimização da energia cinética e a norma do torque das juntas do robô.

Diferente dos trabalhos anteriores que consideram apenas a dinâmica interna dos manipuladores durante uma trajetória, Woolfrey, Lu e Liu (2019) propõem em seu trabalho a otimização dos torques das juntas em um manipulador redundante com a ação de forças externas sobre o efetuador final.

Contudo, nos trabalhos citados, as propostas para a otimização do desempenho dinâmico do manipulador optam por desconsiderar os efeitos da gravidade no cálculo das equações dinâmicas. Além disso, há poucos trabalhos em que a posição inicial do manipulador é levado em consideração na resolução da redundância.

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

1.1 Objetivos

O objetivo geral desse trabalho é melhorar o desempenho dinâmico de um manipulador planar serial de três GDL, com redundância no espaço das tarefas, através da resolução pelos métodos local e global e levando em consideração os efeitos da gravidade na otimização e da orientação inicial do efetuador final do manipulador.

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

- Modelar um manipulador robótico planar em série com três graus de liberdade com suas equações cinemáticas e dinâmicas;
- Implementar a modelagem no *software* MATLAB para que seja possível realizar as simulações necessárias neste trabalho;
- Comparar os resultados do manipulador redundante otimizado através da resolução da redundância utilizando os métodos local e global com um manipulador não redundante com a finalidade de analisar o desempenho de cada caso;
- Realizar comparações dos resultados da otimização da orientação inicial do efetuador final e com um fator de ponderação nas forças gravitacionais para definir o impacto desses elementos na resolução da redundância;

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, redundância cinemática, resolução da redundância através dos métodos local e global, cinemática, dinâmica e planejamento da trajetória do efetuador final;
- O capítulo 3 descreve os métodos de resolução da redundância utilizados e apresenta as funções custo e condições de contorno;
- O capítulo 4 apresenta os resultados obtidos com a resolução da redundância 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 robô manipulador, é preciso ter conhecimento das suas características e dos conceitos básicos de modelagem para a elaboração de um modelo matemático. Este modelo se faz necessário, pois ele será implementado em um ambiente de simulação no qual serão obtidos os resultados das ações propostas 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, redundância, resolução da redundância e otimização, modelagem cinemática e dinâmica e planejamento da trajetória.

2.1 Características dos manipuladores robóticos

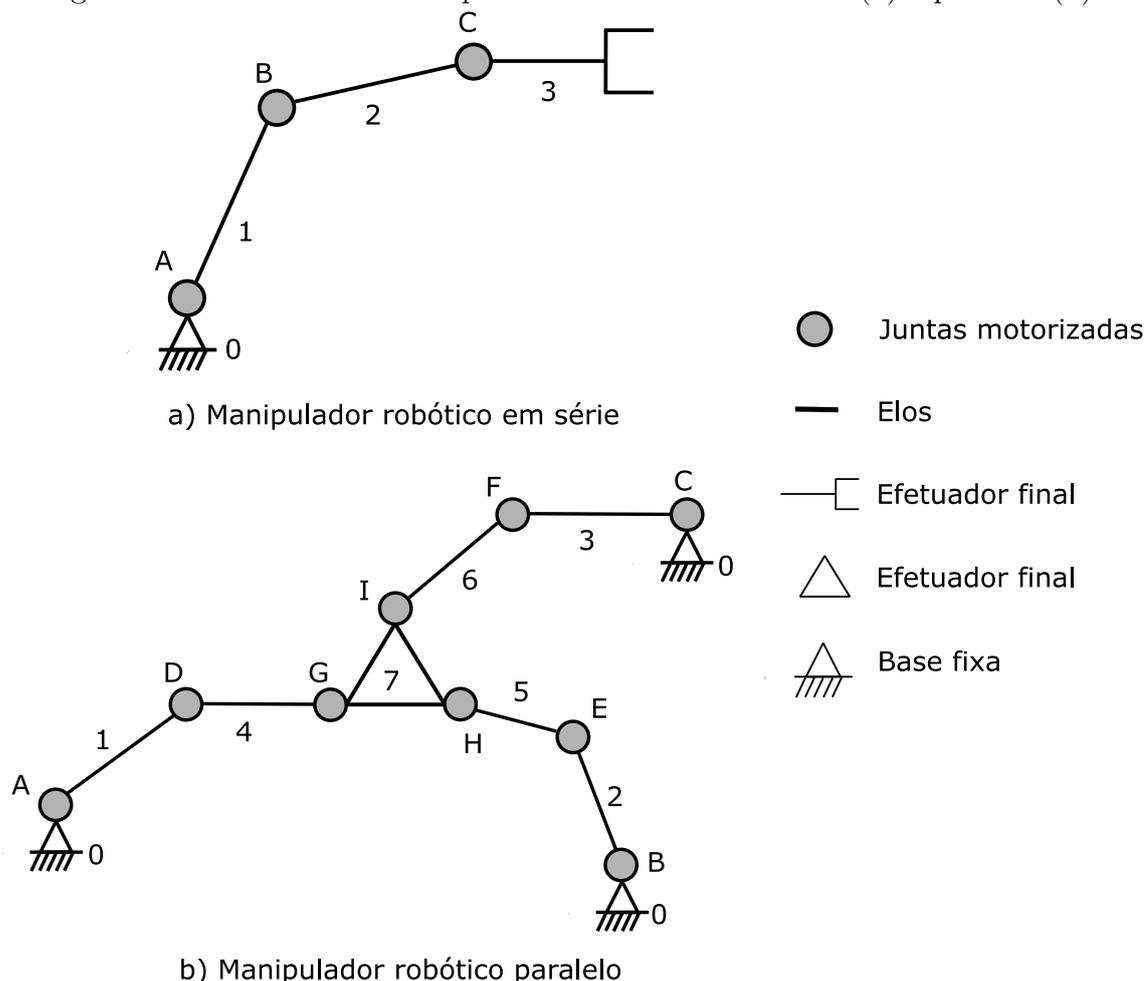
De modo geral, a classificação dos robôs é definida através da sua estrutura mecânica. Segundo Siciliano *et al.* (2009), eles podem ser divididos em duas categorias principais: aqueles que possuem uma base fixa, sendo chamados de robôs manipuladores e aqueles que possuem uma base móvel, chamados de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

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 cada um dos manipuladores pode ser visto na Figura 3. Uma vez que o trabalho tem como foco os manipuladores em série, os conceitos apresentados são sobre esse tipo de manipulador.

2.1.1 Mobilidade

A mobilidade dos manipuladores seriais está diretamente associado 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).

Figura 3 – Estrutura dos manipuladores robóticos em série (a) e paralelo (b).



Fonte: Adaptado de Weihmann (2011).

Desse modo, 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.

2.2 Redundância de manipuladores robóticos

Segundo Conkur e Buckingham (1997), existem muitas definições para redundância na robótica, sendo um termo utilizado para situações diferentes, mas que apresentam uma certa relação. De modo geral, a redundância em manipuladores podem ser classificadas em redundância de sensor e mecânica. Para o caso dos sensores, ela é definida como a situação

em que há mais sensores presentes no manipulador do que o necessário. Geralmente é utilizado em sistemas que demandam alta confiabilidade, precisão de posicionamento e pode ser considerado como um fator de segurança. A redundância mecânica ainda pode ser dividida em redundância de atuação e cinemática.

Apesar de um manipulador mecanicamente não redundante ser 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, como apresentado no trabalho de Kotlarski *et al.* (2010).

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 serie é igual ao número de juntas atuadas, não existe redundância de atuação em manipuladores de cadeia aberta.

Desse modo, esse trabalho tem o foco na redundância cinemática e uma breve revisão dos seus conceitos estão a apresentados a seguir.

2.2.1 Redundância cinemática da tarefa

Conforme Weihmann (2011), a redundância cinemática da tarefa leva em consideração o que o manipulador deve executar com seu efetuator final. Essas 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 2.

$$R = m - d \quad (2)$$

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 3.

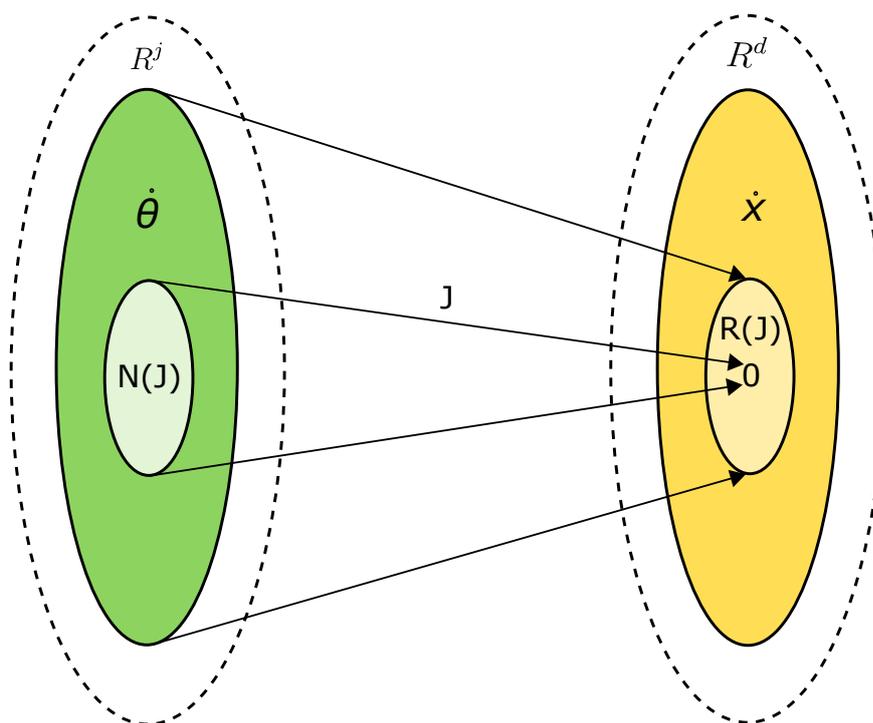
$$R = j - d \quad (3)$$

De modo geral, 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 ($R = 1$).

Além dessa definição, 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 4 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 uma posição específica na trajetória. Na Figura 4, 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 .

Figura 4 – Espaço nulo dos manipuladores cinematicamente redundantes.



Fonte: Adaptado de Weihmann (2011).

Com isso, 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.

2.3 Resolução da redundância e otimização

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 problema da redundância pode ser resolvido de duas maneiras: através do método local ou global.

O método local observa o gradiente de uma função custo a cada instante de tempo e determina, em tempo real, a próxima posição das juntas com base na configuração atual. Esse método é utilizado quando a função custo possui um cálculo rápido e menos complexo, segundo o trabalho de Nedungadi e Kazerouinian (1989).

O método global resolve a redundância levando em consideração uma trajetória completa do manipulador para determinar o valor da função custo. Desse modo, o método é computacionalmente mais trabalhoso, já que analisa várias vezes a mesma trajetória para obter o valor ótimo da função custo.

A vantagem do método local sobre o global 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 (AHUACTZIN; GUPTA, 1999). Entretanto, como mostrado em 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. A otimização global pode resolver esses dois pontos, mas suas desvantagens são: requerer um custo computacional elevado de execução e não ser possível utilizar esse método em tempo real de movimentação do robô.

2.3.1 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.

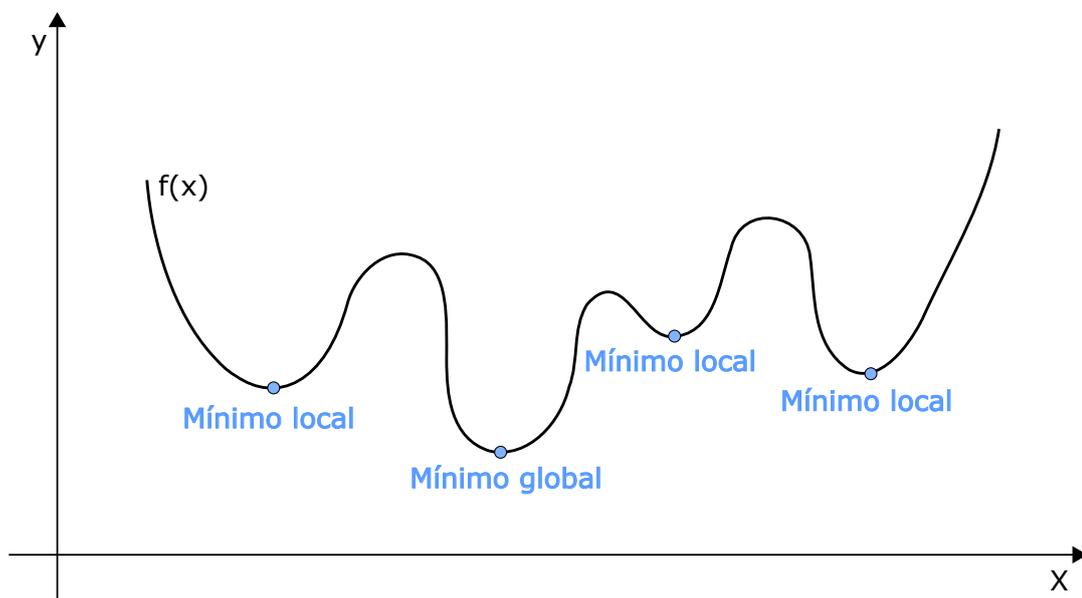
Esse tipo de problema pode ser definido como:

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

Sendo a função f 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. No caso dos problemas convexos, a solução encontrada pode ser considerada global, pois não há outra solução ótima. 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. Um exemplo de mínimos local e global em um dado domínio de uma função não convexa é mostrado na Figura 5.

Figura 5 – Mínimos local e global de uma função não convexa em um determinado domínio.



Fonte: Elaborado pelo autor (2022).

Apesar de uma função não convexa possuir um ótimo global, não é possível garantir que ele é o resultado encontrado pela otimização, como é o caso do presente trabalho. Esse problema de otimização possui várias formas de ser abordado, alguns são mais determinísticos e restritos e outros mais abrangentes na busca do melhor resultado possível. Nesse trabalho serão utilizados três estratégias: uma mais restrita, sendo o método de multiplicação de Lagrange, como apresentado em Nenchev (1989), Nedungadi e Kazerouinian (1989) e Woolfrey, Lu e Liu (2019), e as abrangentes, sendo elas o algoritmo genético (*genetic algorithm*, GA) e a programação quadrática sequencial (*sequential quadratic programming*, SQP). A implementação da otimização usará as seguintes funções implementadas no MATLAB: *fmincon* que executa o algoritmo SQP e *ga* para o algoritmo genético.

O algoritmo genético utiliza dos conceitos biológicos de população, indivíduo, hereditariedade, mutação, recombinação e seleção para determinar um valor ótimo da função custo. Um conjunto (população) de variáveis de otimização (indivíduos) é criado

no início do algoritmo e são testados pela função custo. Se um indivíduo está fora das restrições ou resulta em valores considerados inadequados, ele é descartado da população e um novo conjunto é gerado utilizando mutações, recombinações e a seleção natural. Esse processo é repetido por uma determinada quantidade de gerações ou até que os indivíduos convirjam, dentro de uma tolerância entre eles, para um valor que leve a uma solução ótima.

O GA apresenta a desvantagem de ser um algoritmo computacionalmente custoso em situações com grande quantidade de indivíduos a cada geração, levando muito tempo para ser executado e sem a garantia de atingir uma solução ótima global para problemas convexos. Por outro lado, a vantagem é que esse algoritmo não precisa de uma entrada de valor inicial para as variáveis de otimização, evitando que seja encontrado uma solução dependente desse valor (FONTES, 2019).

Nesse trabalho, a função *ga* foi utilizada com as configurações padrão do MATLAB tendo uma população de 50 indivíduos e um máximo de 100 gerações. A otimização é encerrada quando a variação do valor da função custo for menor que 10^{-6} ou no limite de 100 gerações. Um exemplo da implementação da função *ga* e demais variáveis de entrada estão apresentadas a seguir:

```
nvars = [];           %quantidade de variáveis de entrada
lb = [];             %limitante inferior dos valores das variáveis
ub = [];             %limitante superior dos valores das variáveis
A = [];              %matriz dos coeficientes da função de restrição
b = [];              %vetor dos valores para a função de restrição
Aeq = [];            %matriz dos coeficientes da restrição de desigualdade
beq = [];            %vetor dos valores para a restrição de desigualdade
nonlcon = [];        %vetor dos valores para a restrição não linear

options = optimoptions('ga','Display','iter','PlotFcn',@gaplotbestf)
[X,fval] = ga(@FCusto,nvars,A,b,Aeq,beq,lb,ub,nonlcon,options)
```

O algoritmo SQP é um método com convergência mais rápida e moderno em relação ao GA. Esse método é poderoso para solução de problemas reais uma vez que pode lidar com problemas não lineares utilizando uma combinação dos métodos de Newton, Kuhn-Tucker e dos multiplicadores de Lagrange. Esse método depende de um valor inicial para começar o processo de otimização, o que é uma desvantagem já que pode levar a solução para um ótimo local próximo do valor inicial escolhido.

Na função *fmincon*, foi utilizado as configurações padrão do MATLAB em que são permitidas no máximo 400 iterações. A otimização é encerrada quando a variação do valor da função custo for menor que 10^{-6} ou no limite de iterações. Um exemplo da

implementação da função *fmincon* e demais variáveis de entrada estão apresentadas a seguir:

```
nvars = [];           %quantidade de variáveis de entrada
lb = [];             %limitante inferior dos valores das variáveis
ub = [];             %limitante superior dos valores das variáveis
A = [];              %matriz dos coeficientes da função de restrição
b = [];              %vetor dos valores para a função de restrição
Aeq = [];            %matriz dos coeficientes da restrição de desigualdade
beq = [];            %vetor dos valores para a restrição de desigualdade
nonlcon = [];        %vetor dos valores para a restrição não linear
x0 = []              %vetor dos valores iniciais para fmincon obtido com o ga

options = optimoptions('fmincon','Display','iter','Algorithm','sqp');
D = fmincon(@FCusto,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
```

Desse modo, é interessante utilizar os métodos GA e SQP em conjunto, iniciando a otimização através do algoritmo genético, para obter uma solução quase-ótima, que servirá como entrada para o algoritmo SQP que com esse valor inicial poderá convergir para uma solução ótima.

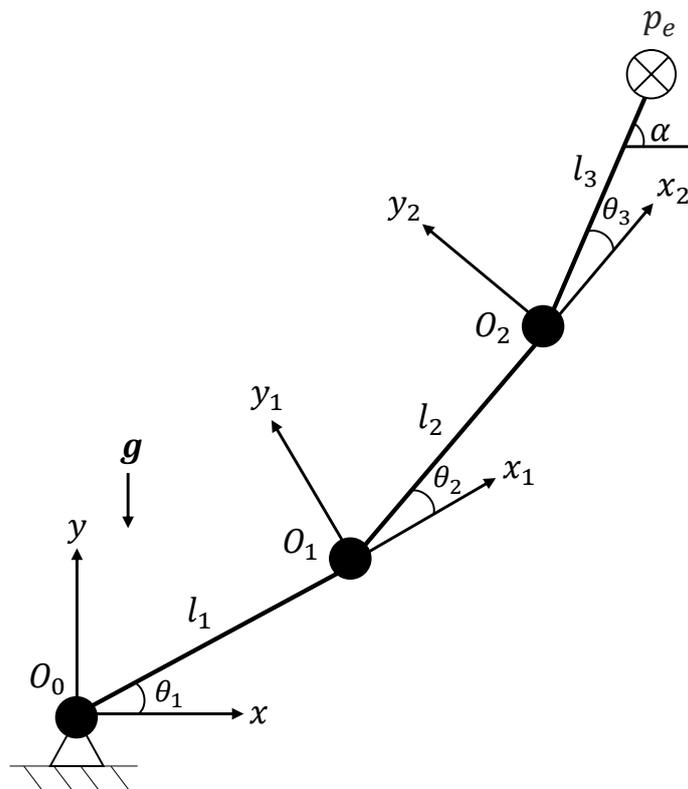
2.4 Modelagem de um manipulador planar em série

Para a modelagem do manipulador planar serial com redundância cinemática utilizado no projeto foi tomado como base o manipulador mostrado na Figura 6. 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 α é a orientação do efetuador final em relação ao sistema de referência fixo (x, y) . A posição do efetuador final é representado pelo vetor $\mathbf{p}_e = [x_e, y_e]$. 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.

2.4.1 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:

Figura 6 – Modelo do manipulador planar serial.



Fonte: Elaborado pelo autor (2022).

- 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} .
- α_i - ângulo de rotação ao redor do eixo x_i para deixar o eixo z_{i-1} paralelo com o eixo z_i .

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

2.4.2 Cinemática direta

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 ($i - 1$).

Tabela 1 – Parâmetros DH para o manipulador serial.

| i | θ_i | d_i | a_i | α_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 (2022).

A Equação 4 é a expressão matricial de A , na qual c e s representam as funções cosseno e seno, respectivamente.

$$A_i^{i-1} = \begin{bmatrix} c(\theta_i) & -s(\theta_i)c(\alpha_i) & s(\theta_i)s(\alpha_i) & a_i c(\theta_i) \\ s(\theta_i) & c(\theta_i)c(\alpha_i) & c(\theta_i)s(\alpha_i) & a_i s(\theta_i) \\ 0 & s(\alpha_i) & c(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

O próximo passo é 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 5. A parte da matriz composta pelos elementos R_{ij} representa a rotação e x_e, y_e, z_e a posição do efetuador final.

$$T_3^0 = A_1^0 A_2^1 A_3^2 = \begin{bmatrix} R_{11} & R_{12} & R_{13} & x_e \\ R_{21} & R_{22} & R_{23} & y_e \\ R_{31} & R_{32} & R_{33} & z_e \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

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

$$T_3^0 = \begin{bmatrix} R_{11} & R_{12} & x_e \\ R_{21} & R_{22} & y_e \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

A posição do efetuador final \mathbf{p}_e está representada pela Equação 7.

$$\mathbf{p}_e = \begin{bmatrix} x_e \\ y_e \end{bmatrix} = \begin{bmatrix} l_1 c(\theta_1) + l_2 c(\theta_1 + \theta_2) + l_3 c(\theta_1 + \theta_2 + \theta_3) \\ l_1 s(\theta_1) + l_2 s(\theta_1 + \theta_2) + l_3 s(\theta_1 + \theta_2 + \theta_3) \end{bmatrix} \quad (7)$$

O ângulo α pode ser calculado da seguinte forma:

$$\alpha = \theta_1 + \theta_2 + \theta_3 \quad (8)$$

2.4.3 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 α do efetuador final. É 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;

A cinemática inversa é essencial para modelar um manipulador que movimente-se em função de uma trajetória determinada para o efetuador final. Algebricamente, os ângulos das juntas podem ser determinados a partir da Equação 7. Considerando a posição da junta O_2 , temos:

$$\mathbf{p}_{O_2} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} l_1 c(\theta_1) + l_2 c(\theta_1 + \theta_2) \\ l_1 s(\theta_1) + l_2 s(\theta_1 + \theta_2) \end{bmatrix} \quad (9)$$

Somando as equações de x_2 e y_2 elevadas ao quadrado e isolando $\cos(\theta_2)$ obtêm-se:

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

As variáveis x_2 e y_2 podem ser obtidas em função de x_e , y_e e α :

$$x_2 = x_e - l_3 \cos(\alpha) \quad (11)$$

$$y_2 = y_e - l_3 \sin(\alpha) \quad (12)$$

Pela Equação 10 é possível obter o valor de $\cos(\theta_2)$ e determinar θ_2 :

$$\sin(\theta_2) = \sqrt{1 - \cos^2(\theta_2)} \quad (13)$$

$$\theta_2 = \tan^{-1} \left(\frac{\sin(\theta_2)}{\cos(\theta_2)} \right) \quad (14)$$

Pela Equação 9, utilizando a relação trigonométrica $\cos(\theta_1 + \theta_2) = \cos(\theta_1)\cos(\theta_2) - \sin(\theta_1)\sin(\theta_2)$, obtêm-se:

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

$$\text{sen}(\theta_1) = \frac{(l_1 + l_2 \cos(\theta_2))y_2 - (l_2 \text{sen}(\theta_2))x_2}{x_2^2 + y_2^2} \quad (16)$$

$$\theta_1 = \tan^{-1} \left(\frac{\text{sen}(\theta_1)}{\cos(\theta_1)} \right) \quad (17)$$

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

$$\theta_3 = \alpha - \theta_2 - \theta_1 \quad (18)$$

2.4.4 Matriz Jacobiana

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 (\mathbf{Jp}_i) e angular (\mathbf{Jo}_i), conforme a Equação 19, sendo n o número de juntas do manipulador.

$$J_{(6 \times n)} = \begin{bmatrix} \mathbf{Jp}_1 & \cdots & \mathbf{Jp}_n \\ \mathbf{Jo}_1 & \cdots & \mathbf{Jo}_n \end{bmatrix} \quad (19)$$

Para o caso do manipulador com juntas rotativas a matriz Jacobiana pode ser construída através da expressão da Equação 20, como apresentado em Siciliano *et al.* (2009).

$$\begin{bmatrix} \mathbf{Jp}_i \\ \mathbf{Jo}_i \end{bmatrix} = \begin{bmatrix} z_{i-1} \times (\mathbf{p}_e - \mathbf{p}_{i-1}) \\ z_{i-1} \end{bmatrix} \quad (20)$$

Os vetores \mathbf{z}_{i-1} , \mathbf{p}_e e \mathbf{p}_{i-1} podem ser definidos da seguinte forma:

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

$$R_{z_i}^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\text{sen}(\theta_i) & 0 \\ \text{sen}(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (21)$$

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

- $\mathbf{p}_e = [x_e, y_e, z_e]$, com $z_e = 0$, é o vetor posição do efetuador final obtido da matriz T_n^0 , conforme apresentado na Equação 7;
- $\mathbf{p}_{i-1} = [x_{i-1}, y_{i-1}, z_{i-1}]$, 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:

$$J_{(6 \times 3)} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (22)$$

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:

$$J_{(3 \times 3)} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \\ 1 & 1 & 1 \end{bmatrix} \quad (23)$$

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:

$$J_{(2 \times 3)} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{bmatrix} \quad (24)$$

2.4.5 Cinemática direta diferencial

Com a matriz Jacobiana é possível definir a cinemática direta diferencial para manipuladores não redundantes pela a relação da Equação 25, onde $\dot{\mathbf{x}} = [\dot{x}_e, \dot{y}_e, \dot{\alpha}]$ 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 $\dot{\mathbf{x}} = [\dot{x}_e, \dot{y}_e]$.

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

2.4.6 Cinemática inversa diferencial

A cinemática inversa diferencial de uma manipulador não redundante é definida pela Equação 26.

$$\dot{\mathbf{q}} = J^{-1}\dot{\mathbf{x}} \quad (26)$$

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 posições 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 27.

$$\mathbf{q}(t) = \int_0^t \dot{\mathbf{q}}(\varsigma) d(\varsigma) + \mathbf{q}(0) \quad (27)$$

A Equação 27 é resolvida utilizando integração simples nos *scripts* das simulações implementados no MATLAB.

2.4.7 Dinâmica

Com base nas equações cinemáticas é possível determinar as equações da dinâmica direta e inversa de uma manipulador não redundante, respectivamente, pelas Equações 28 e 29, 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.

$$\ddot{\mathbf{x}} = J\ddot{\mathbf{q}} + \dot{J}\dot{\mathbf{q}} \quad (28)$$

$$\ddot{\mathbf{q}} = J^{-1}(\ddot{\mathbf{x}} - \dot{J}\dot{\mathbf{q}}) \quad (29)$$

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

As equações dinâmicas podem ser obtidas pela formulação de Lagrange para um sistema conservativo, conforme a Equação 30 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.

$$\frac{\partial}{\partial t} \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \boldsymbol{\tau} \quad (30)$$

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

$$\mathcal{L} = \mathcal{K} - \mathcal{U} \quad (31)$$

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

$$\mathcal{K} = \sum_{i=1}^n \mathcal{K}_i = \sum_{i=1}^n \frac{1}{2} m_i x_{c_i}^2 + \frac{1}{2} \theta_i^2 I_i \quad (32)$$

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:

$$I_i = \frac{1}{12}m_i(l_i^2 + b_i^2) \quad (33)$$

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:

$$\mathcal{U} = \sum_{i=1}^n P_i = \sum_{i=1}^n m_i g y_{c_i} \quad (34)$$

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 30, é obtida a expressão para cálculo do torque da Equação 35, 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.

$$\boldsymbol{\tau} = M\ddot{\mathbf{q}} + \mathbf{c} + \mathbf{g} \quad (35)$$

2.5 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, é determinado as orientações de cada junta do manipulador durante a trajetória, sendo conhecido as posições do efetuador final apenas em pontos determinados do caminho.

A trajetória do efetuador final gerada no espaço das tarefas pode parecer com uma movimentação mais natural do que a gerada no espaço das juntas. Além disso, com esse método é mais fácil de lidar com o desvio de obstáculos, entretanto, as desvantagens são: o custo computacional elevado, já que é preciso resolver a cinemática inversa a cada instante da trajetória e a movimentação das juntas não são consideradas suaves. As trajetórias geradas no espaço das juntas são executadas mais rápido, pois não necessitam que seja realizado a cinemática inversa a cada instante e possuem movimentação mais suave das juntas. A principal desvantagem é não garantir uma trajetória livre de colisões do efetuador (CASTRO, 2019).

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 (SICILIANO *et al.*, 2009).

Dado as características desse trabalho, há intenção de que nos casos analisados o efetuador final percorra o mesmo caminho para efeitos de comparação. Sendo assim, foi escolhido a trajetória gerada no espaço das tarefas com movimentação ponto-a-ponto.

O posicionamento e orientação do efetuador pode ser obtido utilizando um polinômio, o que permite a utilização de condições de contorno para a trajetória. Desse modo, escolheu-se o polinômio de quinto grau da Equação 36 e 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.

$$p(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (36)$$

Conforme apresentado no trabalho de Macfarlane e Croft (2003), esse polinômio fornece um perfil quadrático para o impulso, 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. Para aplicar as condições de contorno é preciso obter 6 equações, o que pode ser feito derivando a Equação 36 obtendo as expressões para velocidade e aceleração. Matricialmente o sistema pode ser representado como:

$$\mathbf{p} = D\mathbf{a} \quad (37)$$

$$\begin{bmatrix} p(t_i) \\ p(t_f) \\ \dot{p}(t_i) \\ \dot{p}(t_f) \\ \ddot{p}(t_i) \\ \ddot{p}(t_f) \end{bmatrix} = \begin{bmatrix} 1 & t_i & t_i^2 & t_i^3 & t_i^4 & t_i^5 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_i & 3t_i^2 & 4t_i^3 & 5t_i^4 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_i & 12t_i^2 & 20t_i^3 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \quad (38)$$

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:

$$\mathbf{a} = D^{-1}\mathbf{p} \quad (39)$$

Sendo assim, os coeficiente do polinômio da equação 36 podem ser calculados da seguinte forma:

$$\begin{aligned} a_0 &= p(t_i) \\ a_1 &= a_2 = 0 \\ a_3 &= \frac{10(p(t_f) - p(t_i))}{t_f^3}, \quad a_4 = \frac{-15(p(t_f) - p(t_i))}{t_f^4}, \quad a_5 = \frac{6(p(t_f) - p(t_i))}{t_f^5} \end{aligned} \quad (40)$$

2.6 Considerações finais do capítulo

Nesse capítulo foram apresentados as bases teóricas necessárias para a modelagem de um manipulador tais como: cinemática, dinâmica e definição da trajetória do efetuador final. Além disso, foi apresentado as definições de redundância cinemática e os métodos para sua resolução. Um deles é método local, que segundo Nenchev (1989) pode levar a instabilidades durante a movimentação do robô e o método global, que necessita de um grande esforço computacional para ser executado. No próximo capítulo são apresentados as estratégias definidas para a resolução da redundância pelos métodos local e global.

3 RESOLUÇÃO DA REDUNDÂNCIA

3.1 Resolução da redundância pelo método local

Uma vez considerado o planejamento da trajetória, para que um manipulador redundante execute a tarefa desejada, é preciso determinar as posições de suas juntas durante a trajetória do efetuador final. 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.

Como a proposta desse trabalho é a otimização do desempenho dinâmico de um manipulador planar redundante, a resolução tem que ser no nível da aceleração. Para efeitos de comparação, duas funções custo foram utilizadas, sendo separadas em caso 1 e caso 2.

3.1.1 Caso 1 - Resolução local ponderada

A função custo z_1 a ser minimizada para o caso 1 é definida sendo a multiplicação dos vetores dos torques das juntas $\boldsymbol{\tau}$ ponderada pela matriz inversa de inércia M do manipulador. Essa otimização é restringida pela condição apresentada na Equação 41, como proposto por Nedungadi e Kazerouinian (1989).

$$\begin{aligned} \text{minimizar: } z_1 &= \boldsymbol{\tau}^T M^{-1} \boldsymbol{\tau} \\ \text{sujeito a: } & J\ddot{\mathbf{q}} + \dot{J}\dot{\mathbf{q}} - \ddot{\mathbf{x}} = 0 \end{aligned} \quad (41)$$

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

$$Z = \boldsymbol{\tau}^T M^{-1} \boldsymbol{\tau} + \lambda^T (J\ddot{\mathbf{q}} + \dot{J}\dot{\mathbf{q}} - \ddot{\mathbf{x}}) \quad (42)$$

A solução deve satisfazer as seguintes condições:

$$\frac{\partial Z}{\partial \ddot{\mathbf{q}}} = 0 \quad \text{e} \quad \frac{\partial Z}{\partial \lambda} = 0 \quad (43)$$

Resolvendo as derivadas parciais apresentadas na Equação 43 tem-se:

$$\frac{\partial Z}{\partial \ddot{\mathbf{q}}} = 2(M\ddot{\mathbf{q}} + \mathbf{c} + \mathbf{g}) + J^T \lambda = 0 \quad (44)$$

$$\frac{\partial Z}{\partial \lambda} = J\ddot{\mathbf{q}} + \dot{J}\dot{\mathbf{q}} - \ddot{\mathbf{x}} = 0 \quad (45)$$

Isolando $\ddot{\mathbf{q}}$ na Equação 44 chega-se ao seguinte resultado:

$$\ddot{\mathbf{q}} = -0,5M^{-1}(J^T\lambda + 2\mathbf{c} + 2\mathbf{g}) \quad (46)$$

Substituindo a Equação 46 na Equação 45 tem-se:

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

Isolando λ da Equação 47:

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

Com esse termo isolado na Equação 48 é possível substituí-lo na Equação 46:

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

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. Simplificando a Equação 49:

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

Desse modo, a Equação 50 apresentada leva em consideração as forças gravitacionais em sua totalidade. Entretanto, nesse trabalho o termo \mathbf{g} teve sua influência avaliada. Para isso, o termo foi ponderado por um fator k , como apresentado na Equação 51.

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

As velocidades das juntas $\dot{\mathbf{q}}$ são calculadas utilizando integração simples, no MATLAB, das acelerações $\ddot{\mathbf{q}}$ e conseqüentemente as posições \mathbf{q} são obtidas com a integração de $\dot{\mathbf{q}}$.

Para avaliar a influência das forças gravitacionais na resolução local, as seguintes situações foram consideradas: $k = 0$, sendo assim retirado qualquer influencia dessas forças na otimização; k assume um valor otimizado para a minimizar a função custo. Esse valor otimizado de k foi obtido através das funções do MATLAB *ga* e *fmincon*.

Outro fator que foi levando em consideração nas otimizações é a orientação inicial α_i do efetuador final. Com as funções *ga* e *fmincon* foi obtido os ângulos α_i que melhor otimizavam a função custo.

Resumidamente, 4 situações foram observadas na resolução local do Caso 1:

- $k = 0$ e α_i com valor fixado;
- k otimizado e α_i com valor fixado;
- $k = 0$ e α_i otimizado;

- k e α_i otimizados simultaneamente.

A otimização de k foi feita utilizando a seguinte função custo e condição de contorno:

$$\begin{aligned} \text{minimizar: } z_k &= \frac{1}{n} \sum_{i=1}^n \sqrt{\boldsymbol{\tau}^T \boldsymbol{\tau}} \\ \text{sujeito a: } & 0 \leq k \leq 1 \end{aligned} \quad (52)$$

A otimização de α_i foi feita através de:

$$\begin{aligned} \text{minimizar: } z_{\alpha_i} &= \frac{1}{n} \sum_{i=1}^n \sqrt{\boldsymbol{\tau}^T \boldsymbol{\tau}} \\ \text{sujeito a: } & -\pi \leq \alpha_i \leq \pi \end{aligned} \quad (53)$$

3.1.2 Caso 2 - Resolução local não ponderada

Para o caso 2, foi utilizado a mesma metodologia do caso anterior. A diferença é a mudança da função custo z_2 que será minimizada, que nesse caso não é ponderada pela matriz inversa de inércia M . Ela está apresentada na Equação 54 e é definida pela multiplicação dos vetores dos torques $\boldsymbol{\tau}$.

$$\begin{aligned} \text{minimizar: } z_2 &= \boldsymbol{\tau}^T \boldsymbol{\tau} \\ \text{sujeito a: } & J\ddot{\mathbf{q}} + \dot{J}\dot{\mathbf{q}} - \ddot{\mathbf{x}} = 0 \end{aligned} \quad (54)$$

A resolução da redundância através do método local para o caso 2 está apresentado na Equação 55 com a matriz Jacobiana pseudo-inversa $J_{M^2}^\dagger$ sendo calculada através da seguinte expressão: $(M^2)^{-1} J^T (J(M^2)^{-1} J^T)^{-1}$.

$$\ddot{\mathbf{q}} = J_{M^2}^\dagger (\ddot{\mathbf{x}} - \dot{J}\dot{\mathbf{q}}) - (I - J_{M^2}^\dagger J) M^{-1} (\mathbf{c} + \mathbf{g}) \quad (55)$$

Igual ao caso 1, a resolução da redundância levou em consideração as forças gravitacionais ponderadas por um fator k e orientação inicial conforme as quatro situações apresentadas na seção anterior. As variáveis k e α_i também foram otimizadas utilizando as condições das Equações 52 e 53, respectivamente. A equação dinâmica com a ponderação do fator gravitacional está descrita na Equação 56:

$$\ddot{\mathbf{q}} = J_{M^2}^\dagger (\ddot{\mathbf{x}} - \dot{J}\dot{\mathbf{q}}) - (I - J_{M^2}^\dagger J) M^{-1} (\mathbf{c} + k\mathbf{g}) \quad (56)$$

3.2 Resolução da redundância pelo método global

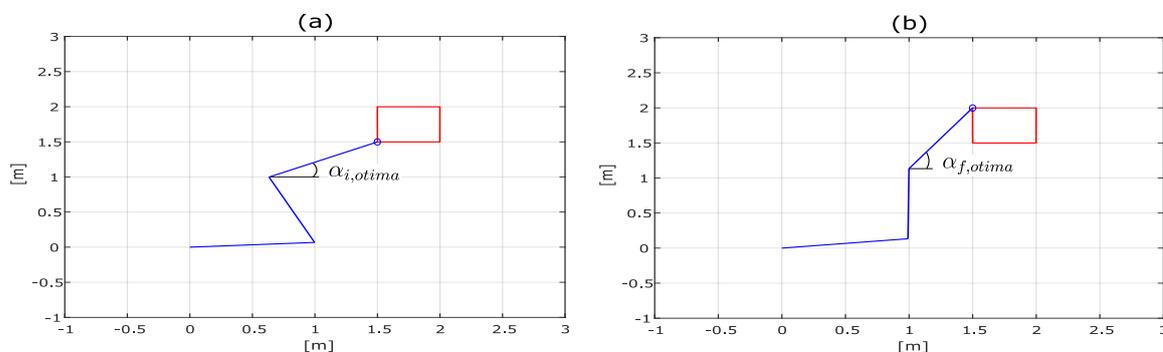
Para a resolução da redundância pelo método global, foi utilizado uma abordagem na qual as variáveis de otimização são as orientações ótimas do efetuador final nos pontos

iniciais ($\alpha_{i,otimo}$) e finais ($\alpha_{f,otimo}$) da trajetória. Conforme apresentado na Equação 36, a trajetória de α pode ser definido por:

$$\alpha(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3 + \alpha_4 t^4 + \alpha_5 t^5 \quad (57)$$

A orientação inicial ótima para um segmento de trajetória do efetuador está representada na Figura 7a), enquanto que na Figura 7b) é mostrado a orientação final ótima para esse segmento. É importante ressaltar que a orientação $\alpha_{f,otimo}$ de um segmento é necessariamente a orientação $\alpha_{i,otimo}$ do próximo segmento para manter a continuidade do movimento.

Figura 7 – Orientações inicial e final ótimas do efetuador em um segmento da trajetória.



Fonte: Elaborado pelo autor (2022).

Com essas orientações ótimas iniciais e finais da trajetória de α definidas, é possível obter os coeficientes α_0 a α_5 da Equação 57 utilizando como condição de contorno as velocidades e acelerações iniciais e finais como nulas, conforme apresentado na Seção 2.5. Dessa forma, é possível utilizar o vetor de velocidade do efetuador final para manipuladores não redundantes, $\dot{\mathbf{x}} = [\dot{x}_e, \dot{y}_e, \dot{\alpha}]$, já que com a definição da trajetória de α a redundância está resolvida. Assim, utilizando a Jacobiana apresentada na Equação 23, é possível calcular as velocidades das juntas com a Equação 26 e finalmente definir as posições das juntas do manipulador através da Equação 27 como é feito para os manipuladores não redundantes. As acelerações das juntas são obtidas com a Equação 29.

Como feito no método local, no global duas funções custos foram utilizadas, sendo separadas em caso 3 - resolução global ponderada e caso 4 - resolução global não ponderada.

3.2.1 Caso 3 - Resolução global ponderada

Nesse caso uma nova função custo z_3 foi utilizada, sendo definida como a média dos torques das juntas na execução completa da tarefa ponderada pela matriz inversa de inércia M . Foi definido também um limite de movimentação para a orientação α . Portanto,

a otimização pode ser descrita como:

$$\begin{aligned} \text{minimizar: } z_3 &= \frac{1}{n} \sum_{i=1}^n \sqrt{\boldsymbol{\tau}^T M^{-1} \boldsymbol{\tau}} \\ \text{sujeito a: } & -\pi \leq \alpha(t) \leq \pi \end{aligned} \quad (58)$$

Para obter as orientações ótimas do efetuador durante trajetória, inicialmente, foi determinado as orientações quase-ótimas iniciais e finais, $\alpha_{i,quase-otima}$ e $\alpha_{f,quase-otima}$, utilizando o algoritmo genético da função *ga* do MATLAB. Esses valores são, posteriormente, utilizados como entrada na função *fmincon* para determinar as orientações $\alpha_{i,otima}$ e $\alpha_{f,otima}$. Desse modo, é realizado a resolução da redundância pelo método global do manipulador, através da minimização da função z_3 . Com as orientações definidas, é possível determinar as velocidades e acelerações e dar continuidade para obter as posições das juntas do manipulador como apresentado anteriormente.

3.2.2 Caso 4 - Resolução global não ponderada

Seguindo a mesma metodologia do caso 3, foi utilizado a função custo z_4 sem a ponderação da matriz inversa de inércia M e que será minimizada através do método global. Essa função está apresentada na Equação 59 e é definida como a média dos torques das juntas durante a execução da tarefa.

$$\begin{aligned} \text{minimizar: } z_4 &= \frac{1}{n} \sum_{i=1}^n \sqrt{\boldsymbol{\tau}^T \boldsymbol{\tau}} \\ \text{sujeito a: } & -\pi \leq \alpha(t) \leq \pi \end{aligned} \quad (59)$$

3.3 Caso 5 - Manipulador não redundante

Para a simulação do manipulador não redundante, do qual os resultados serão utilizados como parâmetro de comparação com os resultados das resoluções local e global, é necessário definir a trajetória de α . Diferente dos casos 3 e 4 da resolução global, em que são otimizadas as orientações inicial ($\alpha_{i,otima}$) e final ($\alpha_{f,otima}$) da trajetória do efetuador, no caso 5 as orientações α_i e α_f são escolhidas arbitrariamente sem nenhuma otimização.

Com as orientações de α definidas, deve-se obter os coeficientes do polinômio de quinto grau da Equação 57 considerando as velocidades e acelerações iniciais e finais como nulas. Com isso, é possível determinar as acelerações das juntas através da Equação 29, considerando a Jacobiana completa da Equação 23 e a sua derivada. Para o cálculo das velocidades das juntas utiliza-se a Equação 26 e por fim as posições das juntas são obtidas através da integração simples da Equação 27.

Em todos os cinco casos os torques das juntas durante a execução da tarefa são calculados através da Equação 35.

3.4 Método de comparação dos resultados

Para 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 60, e a segunda, através da norma infinita dos torques $\|\tau\|_\infty$ de cada uma das juntas.

$$E = \frac{1}{n} \sum_{i=1}^n \sqrt{\tau^T \tau} \quad (60)$$

3.5 Considerações finais do capítulo

Ao longo desse capítulo apresentou-se como foi feito a resolução da redundância cinemática com os métodos local e global. Foi definido as funções custo e condições de contorno que devem ser utilizadas nas otimizações dos casos 1 a 4 e também na otimização de k e α . Além disso, apresentou-se as definições da simulação do manipulador não redundante utilizado no caso 5. No próximo capítulo são apresentados os resultados obtidos através das considerações introduzidas nesse capítulo.

4 RESULTADOS

Os resultados obtidos no trabalho estão apresentados nesse capítulo. Para obtê-los, o primeiro passo foi definir as características físicas do manipulador e a trajetória que o efetuador final deveria seguir, sendo esse caminho o mesmo para todos os casos simulados. Na sequência, foram realizados as resoluções da redundância utilizando o método local nos casos 1 e 2 e a resolução utilizando o método global nos casos 3 e 4. Também foram obtidos os resultados da simulação do manipulador serial não redundante do caso 5. Por fim, foi realizado as comparações entre os torques obtidos nas simulações utilizando a média de todos os torques das juntas (E), como apresentado na Equação 60, e também o valor máximo do torque de cada junta, obtido através norma infinita $\|\tau\|_\infty$ desses valores.

As seções desse capítulo estão organizadas da seguinte forma: características do manipulador, trajetória do efetuador final, resultados simulados, comparações dos resultados e considerações finais do capítulo.

4.1 Características do manipulador

O modelo do manipulador planar serial com três juntas de revolução utilizado nas simulações no MATLAB está apresentado na Figura 6 e possui as características apresentadas na Tabela 2. Os parâmetros l , b , m e I são, respectivamente, comprimento, espessura, massa e momento de inércia dos elos do robô.

Tabela 2 – Parâmetros utilizados no modelo do manipulador serial.

| 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 |

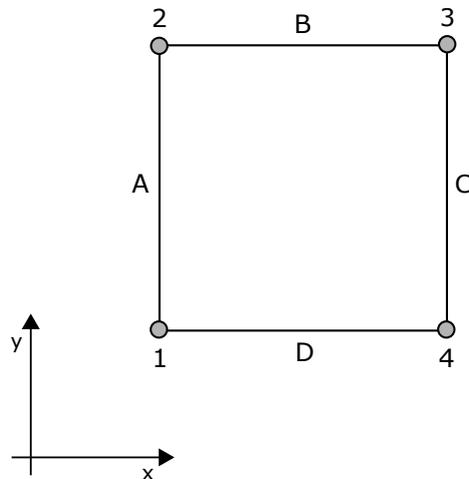
Fonte: Elaborado pelo autor (2022).

Como manipulador possui 3 juntas ($j = 3$) e a dimensão do espaço da tarefa é 2 ($d = 2$), ele tem grau de redundância cinemática $R = 1$, calculado através da Equação 3.

4.2 Trajetória do efetuador final

Nos casos simulados nesse trabalho, foi utilizado um caminho percorrido pelo efetuador final com o formato quadrado de lado $0,5m$, conforme apresentado na Figura 8. A trajetória foi fragmentada em quatro partes, A, B, C e D e os vértices são denominados de 1, 2, 3 e 4. As características da trajetória estão resumidas na Tabela 3.

Figura 8 – Caminho percorrido pelo efetuado final do manipulador.



Fonte: Elaborado pelo autor (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 | 1,50; 1,50 | A | 0,50 | 0,50 |
| 2 | 1,50; 2,00 | B | 0,50 | 0,50 |
| 3 | 2,00; 2,00 | C | 0,50 | 0,50 |
| 4 | 2,00; 1,50 | D | 0,50 | 0,50 |

Fonte: Elaborada pelo autor (2022).

Para exemplificar, na trajetória *A* o manipulador deve sair da posição do vértice 1 e chegar no vértice 2. As condições de contorno para esse segmento são as posições de ambos os vértices, p_{1A} e p_{2A} , as velocidades nesses pontos, \dot{p}_{1A} e \dot{p}_{2A} , que são nulas, assim como as acelerações \ddot{p}_{1A} e \ddot{p}_{2A} , como definido na Seção 2.5. Nos casos 3, 4 e 5, em que a trajetória do ângulo $\alpha(t)$ deve ser definida explicitamente, é realizado o mesmo procedimento. Entretanto, foi definido nesses casos que a orientação inicial e final da trajetória completa são diferentes, isso é $\alpha_{1A} \neq \alpha_{1D}$. No caso do manipulador não redundante simulado no trabalho, foi escolhido arbitrariamente, sem nenhuma otimização, as seguintes posições nos vértices 1, 2, 3, 4 e 1 respectivamente: α_{1A} , 60° , 45° , $22,5^\circ$ e 18° . Os valores de α_{1A} estão definidos na seção 4.3.

4.3 Resultados simulados

Com o modelo implementado e a trajetória definida, as simulações puderam ser realizadas utilizando *scripts* no MATLAB. Para as situações do caso 5, foram utilizadas três orientações iniciais diferentes para o efetuador e escolhidas arbitrariamente, sendo

elas: $\alpha_{1A} = 0^\circ$, $\alpha_{1A} = 15^\circ$ e $\alpha_{1A} = 30^\circ$. Além disso, como posto na Seção 3, foi realizado as simulações do caso 1 com resolução local ponderada e caso 2 com resolução local não ponderada, com as respectivas derivações otimizando k e α . Com a resolução global no caso 3, ponderada, e no caso 4, não ponderada, obteve-se cinco orientações de α otimizadas para o efetuador final nos vértices 1, 2, 3 e 4 em cada caso, sendo elas: $[-18,9^\circ, 9,2^\circ, 18,9^\circ, -10,9^\circ, -18,9^\circ]$ e $[-18,3^\circ, 18,3^\circ, 18,9^\circ, -10,3^\circ, -8,6^\circ]$ respectivamente. Por fim, foi realizado a simulação do caso 5 do manipulador não redundante em que a trajetória da orientação do efetuador foi pré-estabelecida e seus resultados serão utilizados para comparação com a resolução dos manipuladores redundantes.

Na Tabela 4, estão apresentados os resultados dos casos 1 a 5 e suas derivações. Nela estão 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 robô durante a execução da tarefa. Nas situações em que o fator k não é otimizado utiliza-se $k = 0$.

Tabela 4 – Valores da função E e dos torques máximos de cada junta $\|\tau\|_\infty$ para os cinco casos simulados e suas derivações.

| Caso | Situação | E [Nm] | $\ \tau_1\ _\infty$ [Nm] | $\ \tau_2\ _\infty$ [Nm] | $\ \tau_3\ _\infty$ [Nm] |
|--------|--|----------|--------------------------|--------------------------|--------------------------|
| 1a | $k = 0; \alpha_{1A} = 0^\circ$ | 51,03 | 82,63 | 32,31 | 8,77 |
| 1b | $k_{ot} = 0; \alpha_{1A} = 0^\circ$ | 51,03 | 82,63 | 32,31 | 8,77 |
| 1c | $k = 0; \alpha_{1A} = 15^\circ$ | 55,77 | 79,09 | 33,26 | 8,41 |
| 1d | $k_{ot} = 0,57; \alpha_{1A} = 15^\circ$ | 51,70 | 627,76 | 111,32 | 12,21 |
| 1e | $k = 0; \alpha_{1A} = 30^\circ$ | 62,27 | 93,33 | 42,30 | 7,21 |
| 1f | $k_{ot} = 0,16; \alpha_{1A} = 30^\circ$ | 54,32 | 215,59 | 50,28 | 9,52 |
| 1g | $k = 0; \alpha_{1Aot} = -23,5^\circ$ | 38,63 | 101,39 | 45,40 | 8,78 |
| 1h | $k_{ot} = 0,13; \alpha_{1Aot} = -24,1^\circ$ | 37,79 | 109,44 | 48,33 | 8,79 |
| 2a | $k = 0; \alpha_{1A} = 0^\circ$ | 48,63 | 50,33 | 30,00 | 8,71 |
| 2b | $k_{ot} = 0; \alpha_{1A} = 0^\circ$ | 48,63 | 50,33 | 30,00 | 8,71 |
| 2c | $k = 0; \alpha_{1A} = 15^\circ$ | 54,09 | 57,32 | 29,14 | 8,61 |
| 2d | $k_{ot} = 0,75; \alpha_{1A} = 15^\circ$ | 54,07 | 5301,60 | 2135,90 | 263,50 |
| 2e | $k = 0; \alpha_{1A} = 30^\circ$ | 61,04 | 387,90 | 153,58 | 22,35 |
| 2f | $k_{ot} = 0,18; \alpha_{1A} = 30^\circ$ | 54,83 | 1076,50 | 401,30 | 49,40 |
| 2g | $k = 0; \alpha_{1Aot} = -20,1^\circ$ | 39,41 | 59,77 | 51,65 | 8,95 |
| 2h | $k_{ot} = 0,06; \alpha_{1Aot} = -23,5^\circ$ | 37,41 | 90,23 | 60,75 | 10,30 |
| Caso 3 | | 53,58 | 139,75 | 35,49 | 8,56 |
| Caso 4 | | 48,39 | 116,40 | 30,83 | 8,58 |
| 5a | $\alpha_{1A} = 0^\circ$ | 66,65 | 178,82 | 28,20 | 8,26 |
| 5b | $\alpha_{1A} = 15^\circ$ | 60,81 | 121,11 | 28,22 | 8,26 |
| 5c | $\alpha_{1A} = 30^\circ$ | 60,05 | 107,88 | 28,23 | 8,26 |

Fonte: Elaborada pelo autor (2022).

Considerando os resultados apresentados, é possível notar que, na maioria das situações, a resolução da redundância nos casos 1, 2, 3 e 4 consegue reduzir os valores médios

dos torques (E) durante a execução da tarefa, indicando uma melhora do desempenho dinâmico do manipulador quando comparados com o caso 5 (manipulador não redundante). Entretanto, o valor máximo dos torques de cada junta não apresenta uma melhora constante em todas as situações. É possível notar, principalmente nos casos da resolução global 3 (ponderada) e 4 (não ponderada), que apesar da redução de E , os valores dos torques $\|\tau_1\|_\infty$, $\|\tau_2\|_\infty$ e $\|\tau_3\|_\infty$ não apresentam uma vantagem significativa em comparação ao caso 5. Além disso, há um aumento abrupto nos torques máximos em situações dos casos da resolução local 1 (ponderada) e 2 (não ponderada).

Para auxiliar nas análises, foram feitos os gráficos apresentados nos Apêndices A, B, C, D e E com os valores dos torques de cada junta do robô durante a execução da tarefa. Além dos gráficos, os códigos utilizados na simulação e otimização do caso 1h estão apresentados nos Apêndices F, G, H e I e servem de exemplo para os outros casos do trabalho.

4.4 Comparações dos resultados

Como são muitos casos diferentes simulados, para melhor análise dos resultados essa seção foi dividida conforme os objetivos propostos nesse trabalho. Por isso, é feita a comparação do manipulador com resolução da redundância através dos métodos local (casos 1 - ponderada e 2 - não ponderada) e global (casos 3 - ponderada e 4 - não ponderada) com o manipulador não redundante (caso 5). Outra comparação foi feita entre os casos 1 e 2 para determinar o impacto da utilização do fator de ponderação k da gravidade e da escolha da orientação inicial do efetuador final.

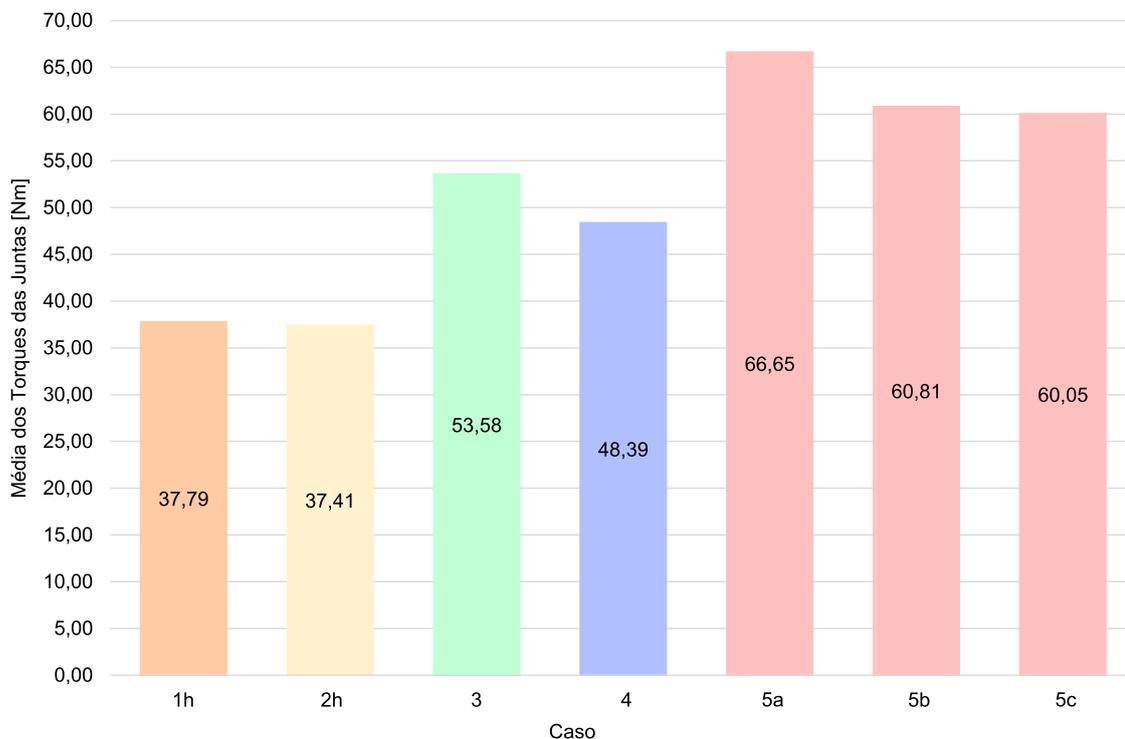
4.4.1 Comparação entre a resolução da redundância utilizando o método local e global

Para melhor visualização dos resultados foi feito o gráfico da Figura 9. Nele está apresentado a média dos torques das juntas do manipulador (E) obtidos com a resolução da redundância local (1h e 2h, ambos com k e α_{1A} otimizados), global (3 e 4) e os resultados do manipulador não redundante (5a, 5b e 5c). Foi escolhido apenas os casos 1h e 2h do método local por serem aqueles que k e α_{1A} estão otimizados.

Conforme a Figura 9, é nítido que a resolução da redundância local apresenta uma maior redução do valor de E . Quando comparado com os casos 5a, 5b e 5c, o 1h apresentou uma redução de 43,30 %, 37,86 % e 37,07 % respectivamente. Para o 2h as reduções são de 43,87 %, 38,48 % e 37,70 %. Entre o caso 1h e 2h não há uma diferença considerável na redução de E , tendo uma variação de apenas 1,01 %.

Para os casos da resolução da redundância com o método global, as reduções de E pelo caso 3 são de 19,61 %, 11,89 % e 10,77 %. Com o 4, há a diminuição em 27,40 %, 20,42 % e 19,41 % quando comparado com 5a, 5b e 5c respectivamente. Entre os casos 3 e 4, o último resultou em uma média dos torques das juntas 9,68 % menor.

Figura 9 – Média dos torques das juntas do manipulador para os casos mais otimizados da resolução local (1h e 2h), global (3 e 4) e do manipulador não redundante (5a, 5b e 5c).



Fonte: Elaborado pelo autor (2022).

O caso 2h (método local com k e α_{1A} otimizados) apresenta uma redução de E de 30,18 % e 22,69 %, quando comparado com os casos 3 e 4 (método global), respectivamente. Com isso, nas condições simuladas nesse trabalho, é possível afirmar que a resolução da redundância pelo método local é mais vantajoso em relação ao método global quando analisado o desempenho dinâmico através da média dos torques das juntas durante a execução da tarefa.

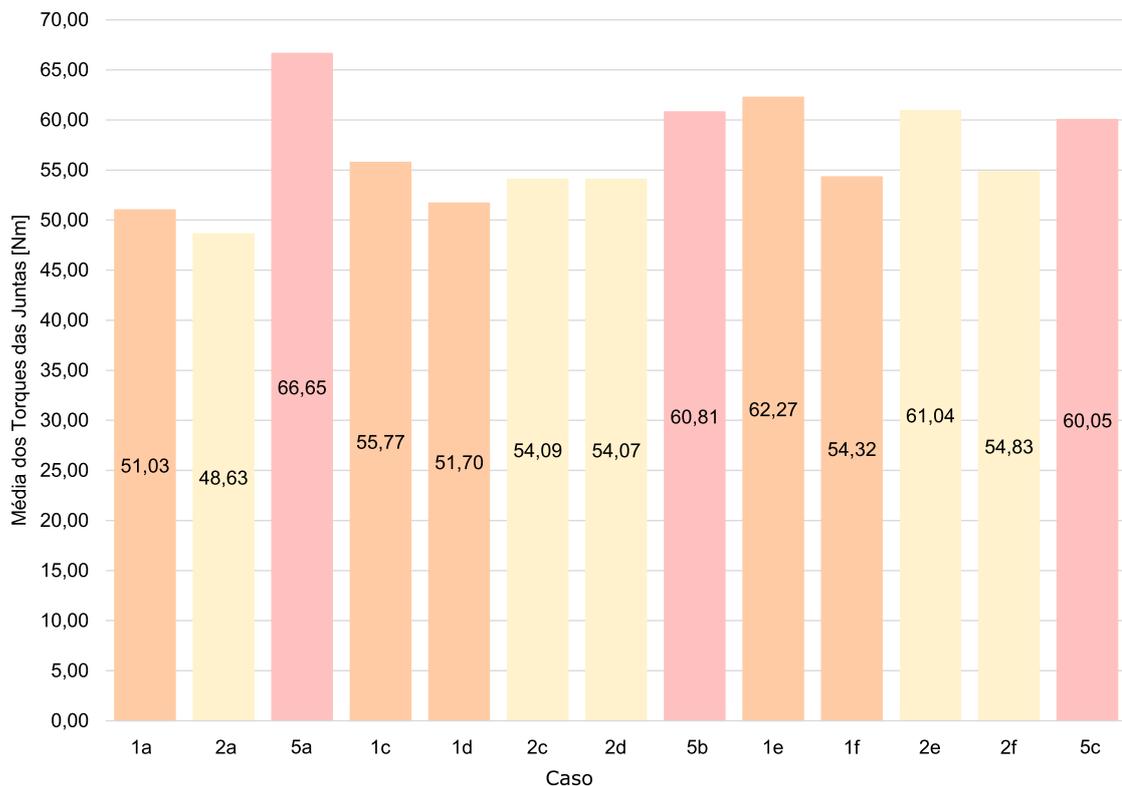
Apesar da redução considerável de E , é preciso salientar que o método local pode levar a variações bruscas dos torques das juntas do robô, como foi apontado por Nenchev (1989). Analisando apenas os casos já expostos nessa seção, principalmente no caso 2h, é visto no gráfico da Figura 27 que há uma grande variação do torque das juntas 1 e 2 no tempo de 1 s e 1,5 s. Já para o método global, as variações dos torques são mais suaves e controladas, como visto nos gráficos das Figuras 28 e 29.

Entretanto, a instabilidade dos torques no caso local pode ser evitada dependendo da função custo utilizada. Quando se compara as curvas dos torques das juntas entre os casos 1h e 2h, pelas Figuras 19 e 27, é possível notar que as variações bruscas de torque são resolvidas utilizando a função custo ponderada pela matriz de inércia na otimização no caso 1h. As curvas de torque obtidas para esse caso são suaves como as do método global nos casos 3 e 4 das Figuras 28 e 29.

4.4.2 Análise do impacto de k e α_{1A} na resolução da redundância

A análise do impacto do fator k e da orientação inicial α_{1A} na resolução da redundância levou em consideração todas as 16 situações dos casos 1 (ponderada) e 2 (não ponderada) da resolução local. Como é possível notar na Tabela 4, o caso 1a ($k = 0$; $\alpha_{1A} = 0^\circ$) tem os mesmos resultados do 1b ($k_{ot} = 0$; $\alpha_{1A} = 0^\circ$), o que também ocorre com os casos 2a ($k = 0$; $\alpha_{1A} = 0^\circ$) e 2b ($k_{ot} = 0$; $\alpha_{1A} = 0^\circ$). Isso ocorre porque o valor otimizado de k é zero em 1b e 2b, sendo esse o mesmo valor escolhido para 1a e 2a. Portanto, são utilizados apenas os dados dos casos 1a e 2a nos gráficos e nas análises. Para melhor representar os resultados foi feito o gráfico da Figura 10. Por ele, nota-se que em todos os casos que foi feita alguma otimização (1b, 2b, 1d, 2d, 1f e 2f) obteve-se uma performance melhor do que o manipulador não redundante (casos 5a - $\alpha_{1A} = 0^\circ$, 5b - $\alpha_{1A} = 15^\circ$ e 5c - $\alpha_{1A} = 30^\circ$).

Figura 10 – Média dos torques das juntas do manipulador para os casos da resolução local (1a, 1c, 1d, 1e, 1f, 2a, 2c, 2d, 2e, 2f) e do manipulador não redundante (5a, 5b e 5c).



Fonte: Elaborado pelo autor (2022).

Os casos 1a, 2a e 5a possuem a mesma orientação $\alpha_{1A} = 0^\circ$. Com a resolução local, 1a teve uma redução de 23,43 % de E em comparação com o 5a. Já para o 2a esse redução foi de 27,04 %.

Os casos 1c ($k = 0$; $\alpha_{1A} = 15^\circ$), 1d ($k_{ot}=0,57$; $\alpha_{1A} = 15^\circ$), 2c ($k = 0$; $\alpha_{1A} = 15^\circ$),

2d ($k_{ot} = 0,75$; $\alpha_{1A} = 15^\circ$) em comparação com 5b ($\alpha_{1A} = 15^\circ$), reduziram E em 8,29 %, 14,98 %, 11,05 % e 11,08 % respectivamente. 1c e 2c não têm o fator k otimizado, já para 1d e 2d esse valor foi otimizado.

Já os casos 1e ($k = 0$; $\alpha_{1A} = 30^\circ$), 1f ($k_{ot} = 0,16$; $\alpha_{1A} = 30^\circ$), 2e ($k = 0$; $\alpha_{1A} = 30^\circ$), 2f ($k_{ot} = 0,18$; $\alpha_{1A} = 30^\circ$) e 5c possuem a orientação inicial $\alpha_{1A} = 30^\circ$. Para 1e e 2e, k não foi otimizado e para 1f e 2f, houve otimização. Os casos 1e e 2e apresentaram um aumento de E em comparação ao 5c de 3,69 % e 1,64 % respectivamente. Com 1f e 2f houve uma redução de 9,54 % e 8,69 % respectivamente.

Analisando o caso 1, é possível notar que a otimização de k sozinho resultou em redução de E . Comparando 1c ($k = 0$; $\alpha_{1A} = 15^\circ$) e 1d ($k_{ot} = 0,57$; $\alpha_{1A} = 15^\circ$) houve uma redução de 7,30 % na média dos torques. A mesma situação ocorre em 1e ($k = 0$; $\alpha_{1A} = 30^\circ$) e 1f ($k_{ot} = 0,16$; $\alpha_{1A} = 30^\circ$), com uma redução de E em 12,77 % com a otimização de k . Quando se analisa o caso 2, há uma pequena redução comparando 2c ($k = 0$; $\alpha_{1A} = 15^\circ$) e 2d ($k_{ot} = 0,75$; $\alpha_{1A} = 15^\circ$) de 0,04 %. Já entre 2e ($k = 0$; $\alpha_{1A} = 30^\circ$) e 2f ($k_{ot} = 0,18$; $\alpha_{1A} = 30^\circ$) ocorre uma redução de 10,17 %.

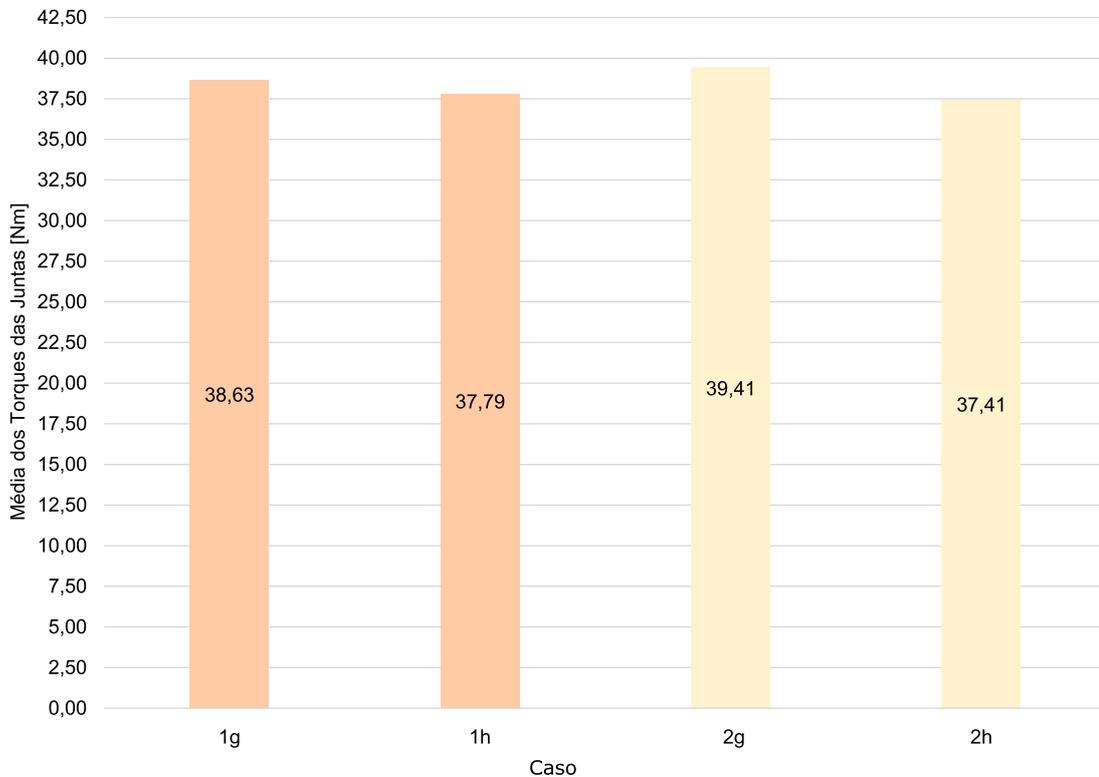
Através dos gráficos das Figuras 15, 17, 23, 24 e 25 é visto uma variação abrupta do torque nos casos 1d, 1f, 2d, 2e e 2f, chegando a atingir um valor de até 5301,60 [Nm] no 2e, conforme apresentado na Tabela 4. Isso pode indicar que na tentativa de melhorar o desempenho dinâmico, a estratégia de otimizar o fator k pode levar o manipulador a posições que aumentam excessivamente os torques, ou seja, em posições de singularidades.

Considerando a otimização conjunta da orientação inicial e do fator k , há os casos 1g e 2g em que apenas α_{1A} é otimizado e 1h e 2h nos quais ambos são otimizados. Com base nos gráficos das Figuras 10 e 11 é possível afirmar que a otimização de apenas α_{1A} tem uma grande contribuição para a redução de E .

Quando a otimização de k também é realizada, há uma pequena redução entre 1g ($k = 0$; $\alpha_{1Aot} = -23,5^\circ$) e 1h ($k_{ot} = 0,13$; $\alpha_{1Aot} = -24,1^\circ$) e entre 2g ($k = 0$; $\alpha_{1Aot} = -20,1^\circ$) e 2h ($k_{ot} = 0,06$; $\alpha_{1Aot} = -23,5^\circ$), sendo elas de 2,17 % e 5,07 % respectivamente. A otimização de α_{1A} e k no caso 1 (resolução local ponderada) apresentou melhores resultados em relação às curvas dos torques quando comparadas com as do caso 2 (resolução local não ponderada). Isso pode ser visto comparando as Figuras 18 e 19 com as Figuras 26 e 27.

Com base nesses resultados, é possível afirmar que a otimização da orientação inicial do efetuador final tem um impacto maior na melhora do desempenho dinâmico em relação à redução da média dos torques das juntas do manipulador. Quando combinado com a otimização de k é visto uma leve melhora nesse quesito. Entretanto, a otimização levando em consideração apenas o fator k não apresentou grandes vantagens em comparação.

Figura 11 – Média dos torques das juntas do manipulador para os casos mais otimizados da resolução local (1g, 1h, 2g e 2h), para comparação da otimização de k e α .



Fonte: Elaborado pelo autor (2022).

4.5 Considerações finais do capítulo

Nesse capítulo, foram apresentados os resultados e as discussões decorrentes. Foi visto que a resolução da redundância pelo método local levou a uma maior redução da média dos torques das juntas quando comparado com a resolução pelo método global. Entretanto, o método local pode levar a variações bruscas dos torques durante a tarefa, o que pode ser contornado dependendo da função custo utilizada na otimização. Outro ponto importante, foi a observação de que a otimização do fator k sozinho, apesar da redução de E , pode gerar instabilidade nos torques levando o manipulador a posições singulares. Já a otimização da orientação inicial α_{1A} se mostrou mais eficiente na melhoria do desempenho dinâmico e quando combinado com a otimização de k obteve-se os melhores resultados na redução da média dos torques das juntas.

5 CONCLUSÃO

O projeto proposto de simular e otimizar um manipulador planar em série através da redundância cinemática foi realizado durante esse trabalho com a intenção de utilizar a redundância para melhorar o desempenho dinâmico do robô. A modelagem dinâmica e cinemática do manipulador planar serial RRR foi desenvolvida e implementada no MATLAB. Com esse modelo, foi possível realizar a resolução da redundância cinemática através dos métodos local e global e simular um manipulador não redundante utilizado para comparação de resultados. Para as otimizações utilizou-se duas funções custo diferentes com a intenção de determinar se os resultados são consistentes em condições distintas.

Além disso, avaliou-se os seguintes aspectos nesse processo: o impacto da utilização do fator de ponderação k que multiplica o termo da gravidade nas equações dinâmicas, a influência da orientação inicial do efetuador final α_{1A} na resolução da redundância e, por fim, a comparação de desempenho entre o método local e global com o intuito de definir qual deles traz mais benefícios ao manipulador.

Os resultados obtidos apontam que a resolução da redundância pelo método local apresenta um desempenho dinâmico melhor quando comparado com o método global e o manipulador não redundante, obtendo essa vantagem apenas quando pelo menos a otimização de α_{1A} é realizada.

Em relação a otimização de k e α_{1A} , os resultados indicam que a otimização de k sozinho não é tão vantajoso, podendo até piorar o desempenho dinâmico em alguns casos, o que pode ser o indício de que essa otimização pode levar o manipulador a posições próximas de singularidades. Já a otimização de α_{1A} levou a melhora significativa ao desempenho dinâmico. Quando combinadas as otimizações do fator k e α_{1A} são obtidos os melhores resultados da resolução pelo método local.

Portanto, a instabilidade durante a movimentação do robô, apontada por Nenchev (1989) como uma das desvantagens do método local, pode ser solucionada realizando esse método de resolução da redundância combinado com a otimização conjunta da orientação inicial do efetuador e do fator de ponderação das forças gravitacionais. Essa solução depende também da função custo utilizada na otimização, sendo a função ponderada pela matriz de inércia do caso 1 a que resultou em torques suaves durante a execução da tarefa.

5.1 Trabalhos futuros

Para ampliar o estudo desse tema, trabalhos futuros podem ter foco em outros aspectos além dos apresentados nesse trabalho, tais como:

- Incluir na resolução da redundância cinemática estratégias para evitar que o manipulador passe por posições de singularidade, evitando assim o aumento excessivo dos

torques das juntas, principalmente se for utilizado a resolução pelo método local;

- Investigar as causas da elevação abrupta dos torques na resolução da redundância pelo método local;
- Utilizar um modelo mais próximo de um manipulador real, considerando os motores das juntas com suas limitações físicas de movimentação e torque suportado. Como alternativa esses modelos podem ser implementados e simulados no ambiente *Simulink* do MATLAB;
- 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.

CASTRO, S. **Trajectory planning for robot manipulators: Student Lounge**. 2019. Disponível em: <<https://blogs.mathworks.com/student-lounge/2019/11/06/robot-manipulator-trajectory/#>>.

CONKUR, E. S.; BUCKINGHAM, R. Clarifying the definition of redundancy as used in robotics. **Robotica**, v. 15, n. 5, p. 583–586, sep 1997. ISSN 0263-5747. Disponível em: <https://www.cambridge.org/core/product/identifler/S0263574797000672/type/journal{_}arti>.

DENAVIT, 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.

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>.

KAZEROUNIAN, K.; WANG, Z. Global versus Local Optimization in Redundancy Resolution of Robotic Manipulators. **The International Journal of Robotics Research**, v. 7, n. 5, p. 3–12, oct 1988. ISSN 0278-3649. Disponível em: <<http://journals.sagepub.com/doi/10.1177/027836498800700501>>.

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-robô/robôs-industriais.>>

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.; KAZEROUNIAN, 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.

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>>.

TRINGALI, A.; COCUZZA, S. Globally optimal inverse kinematics method for a redundant robot manipulator with linear and nonlinear constraints. **Robotics**, v. 9, n. 3, p. 1–24, 2020. ISSN 22186581.

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.

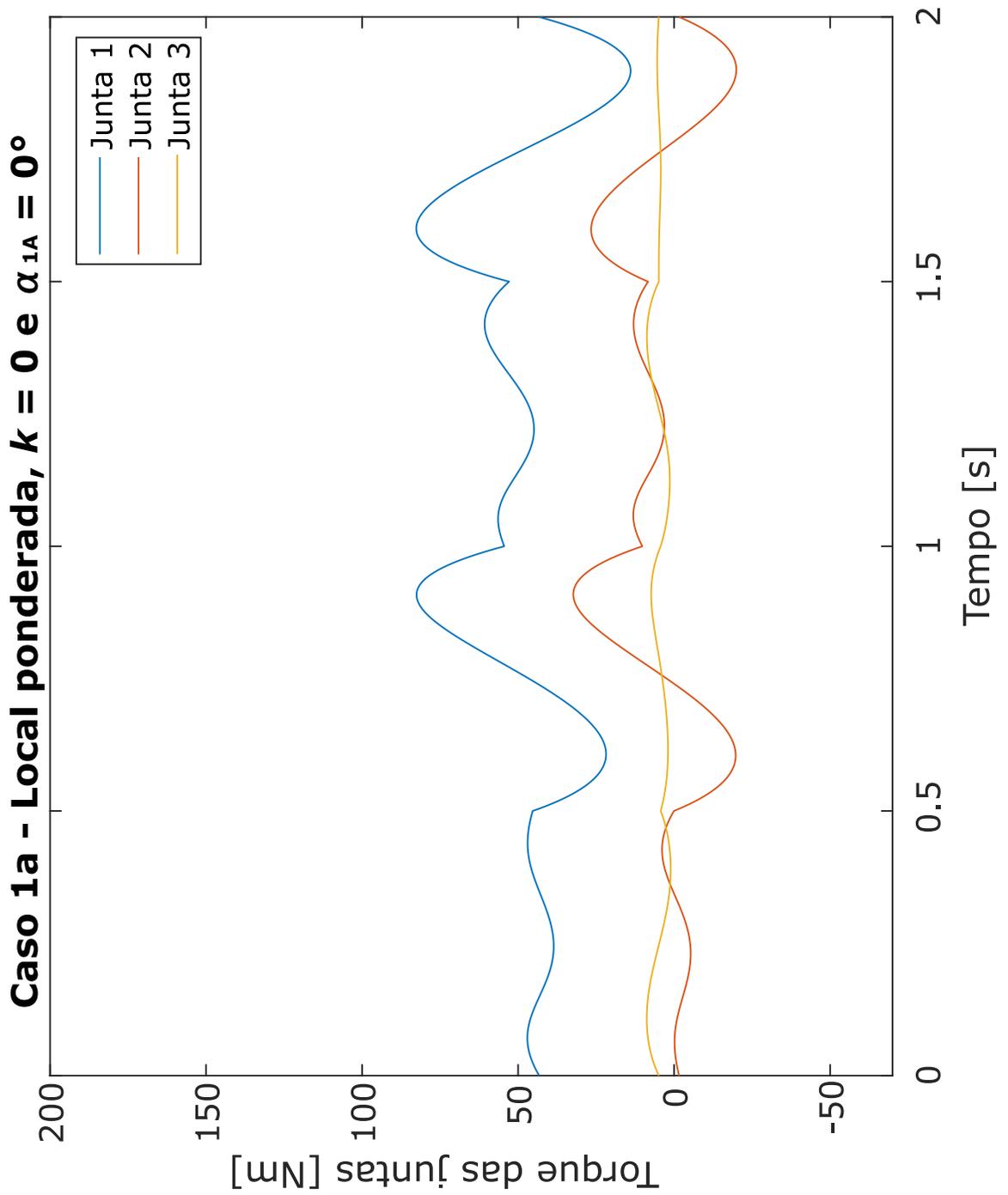
WOOLFREY, J.; LU, W.; LIU, D. A Control Method for Joint Torque Minimization of Redundant Manipulators Handling Large External Forces. **Journal of Intelligent and Robotic Systems: Theory and Applications**, Journal of Intelligent & Robotic Systems, v. 96, n. 1, p. 3–16, 2019. ISSN 15730409.

ZANCHETTIN, A. M. *et al.* Exploiting task redundancy in industrial manipulators during drilling operations. In: **2011 IEEE International Conference on Robotics and Automation**. IEEE, 2011. p. 128–133. ISBN 978-1-61284-386-5. ISSN 10504729. Disponível em: <<http://ieeexplore.ieee.org/document/5979624/>>.

Apêndices

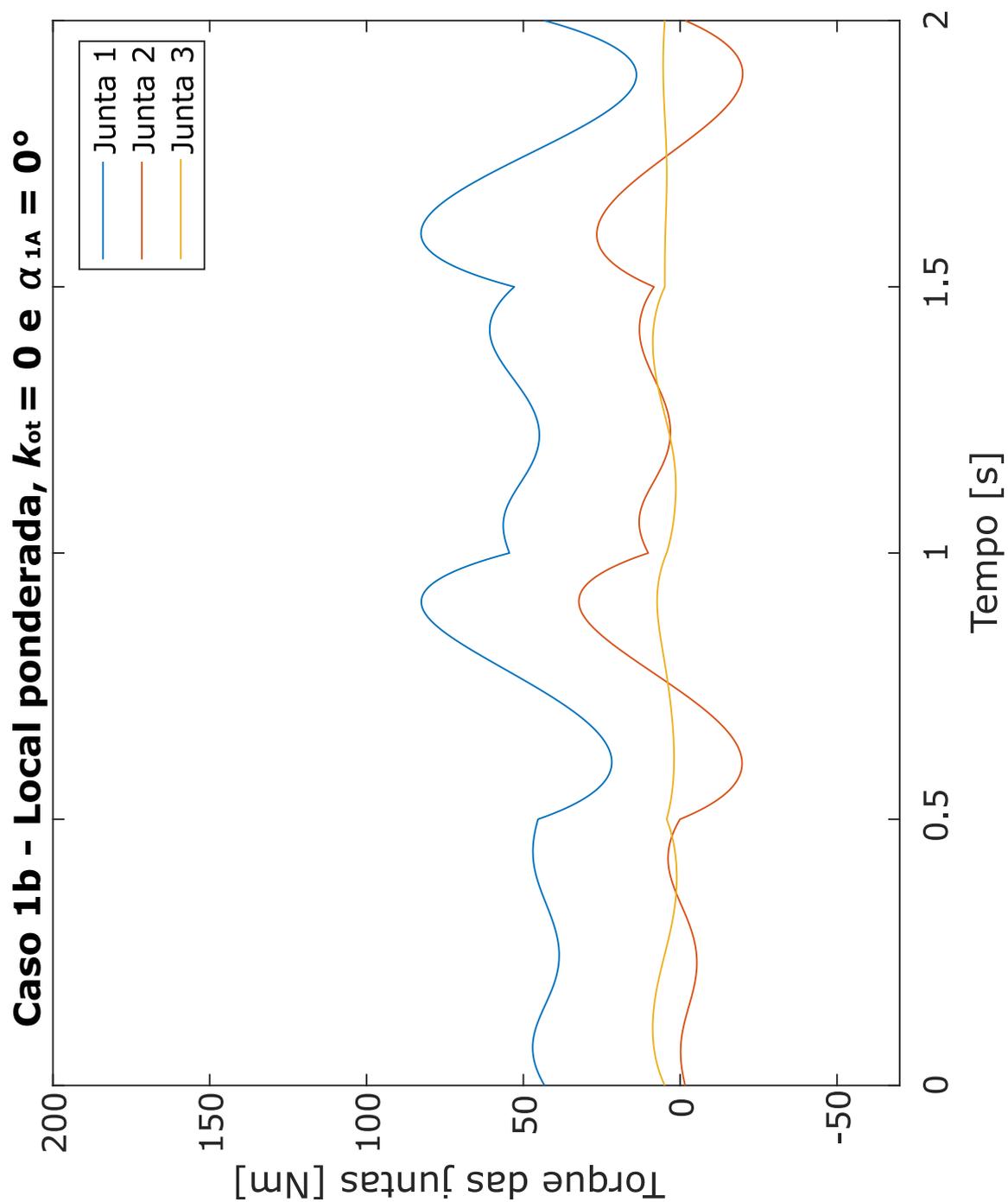
APÊNDICE A – Gráficos dos torques das juntas do manipulador redundante do caso 1

Figura 12 – Torque das juntas do manipulador redundante do caso 1a.



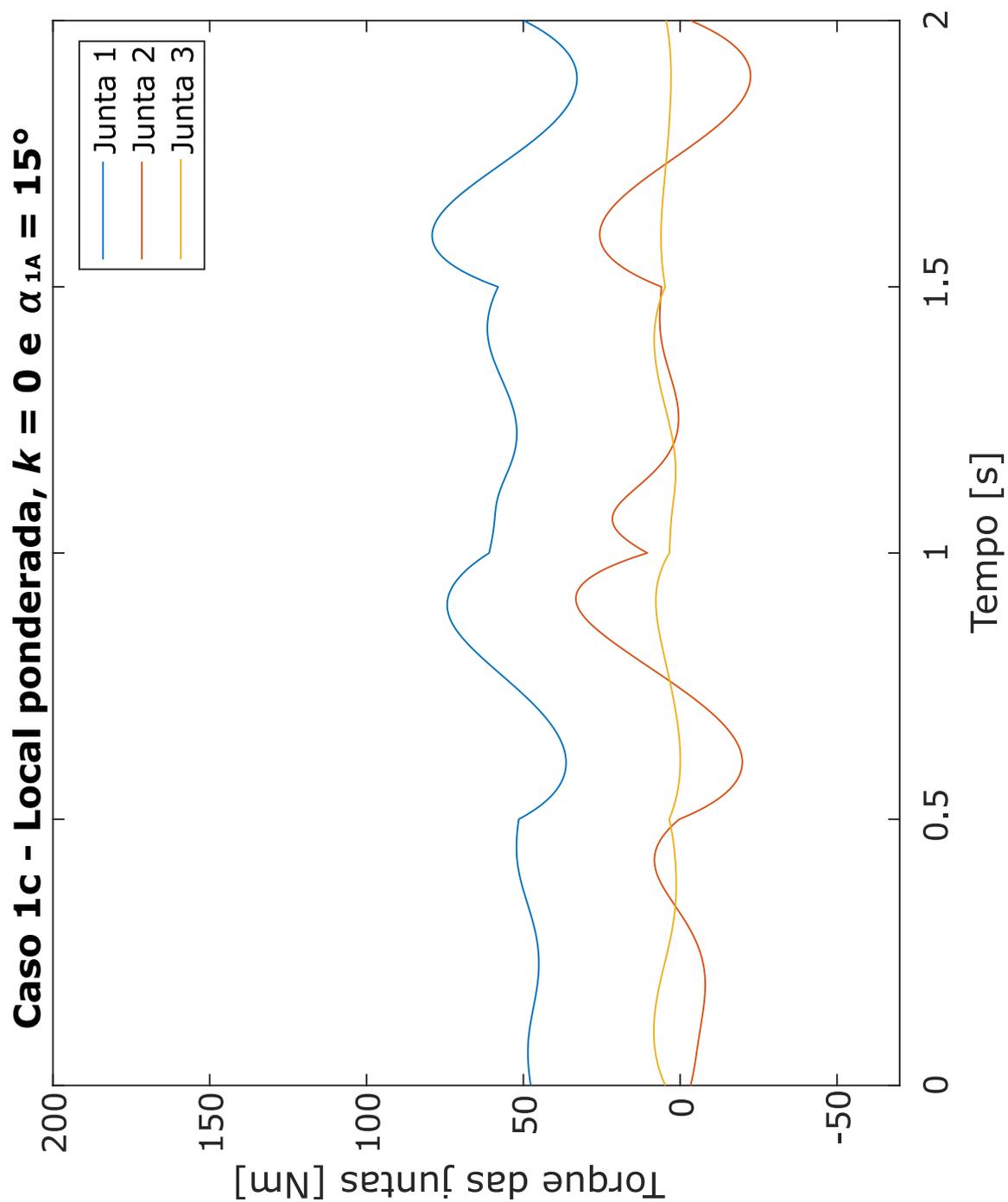
Fonte: Elaborado pelo autor (2022).

Figura 13 – Torque das juntas do manipulador redundante do caso 1b.



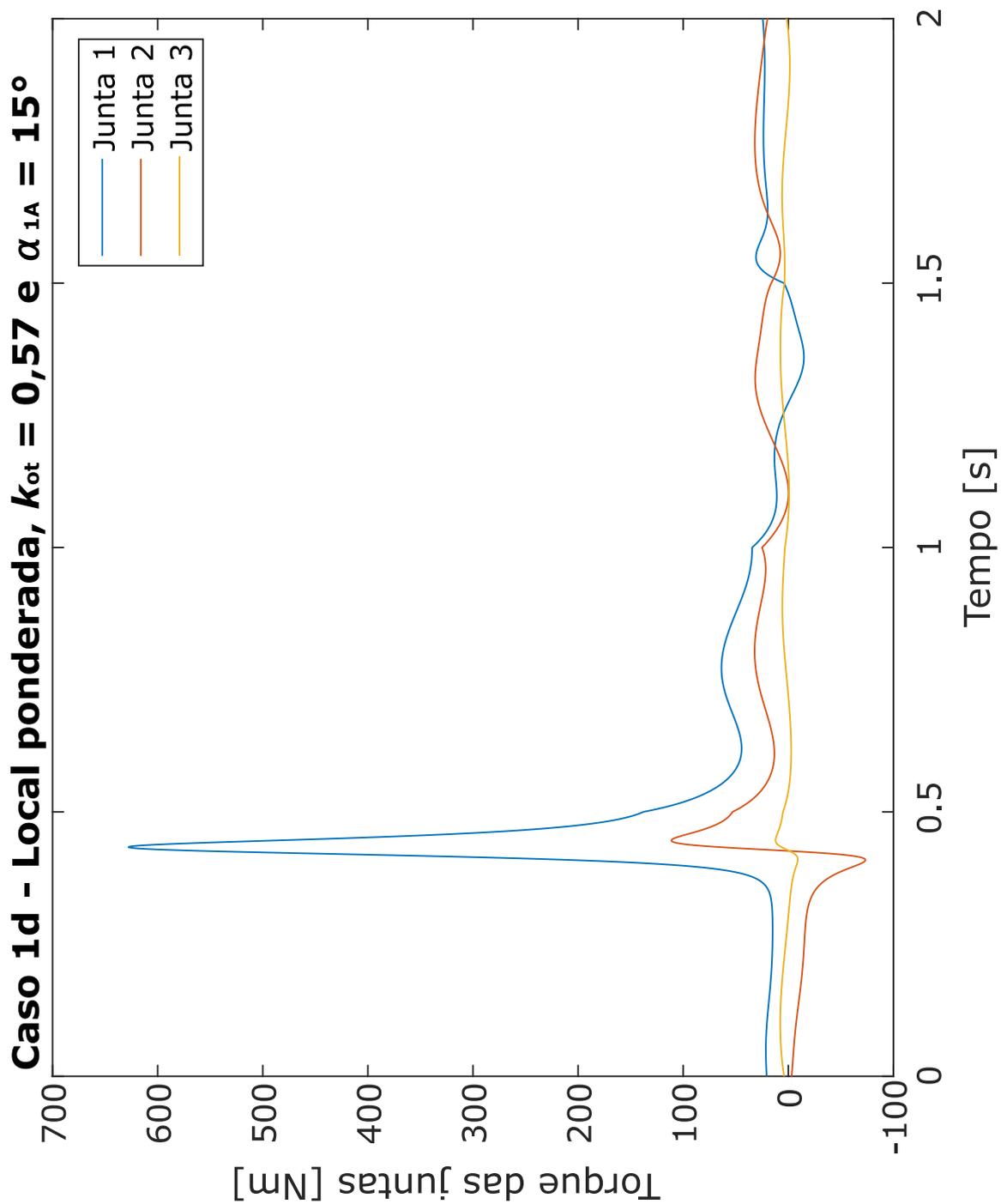
Fonte: Elaborado pelo autor (2022).

Figura 14 – Torque das juntas do manipulador redundante do caso 1c.



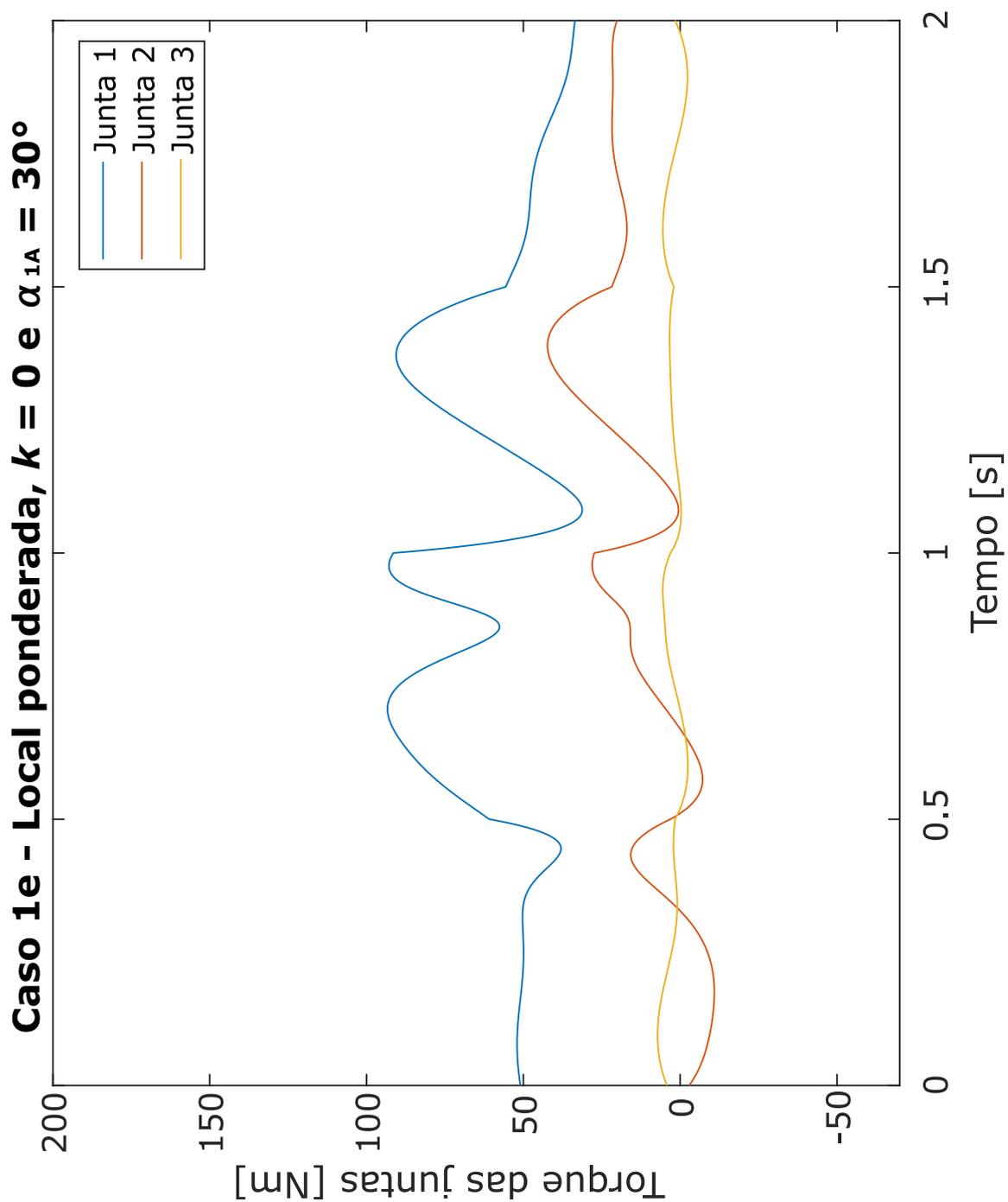
Fonte: Elaborado pelo autor (2022).

Figura 15 – Torque das juntas do manipulador redundante do caso 1d.



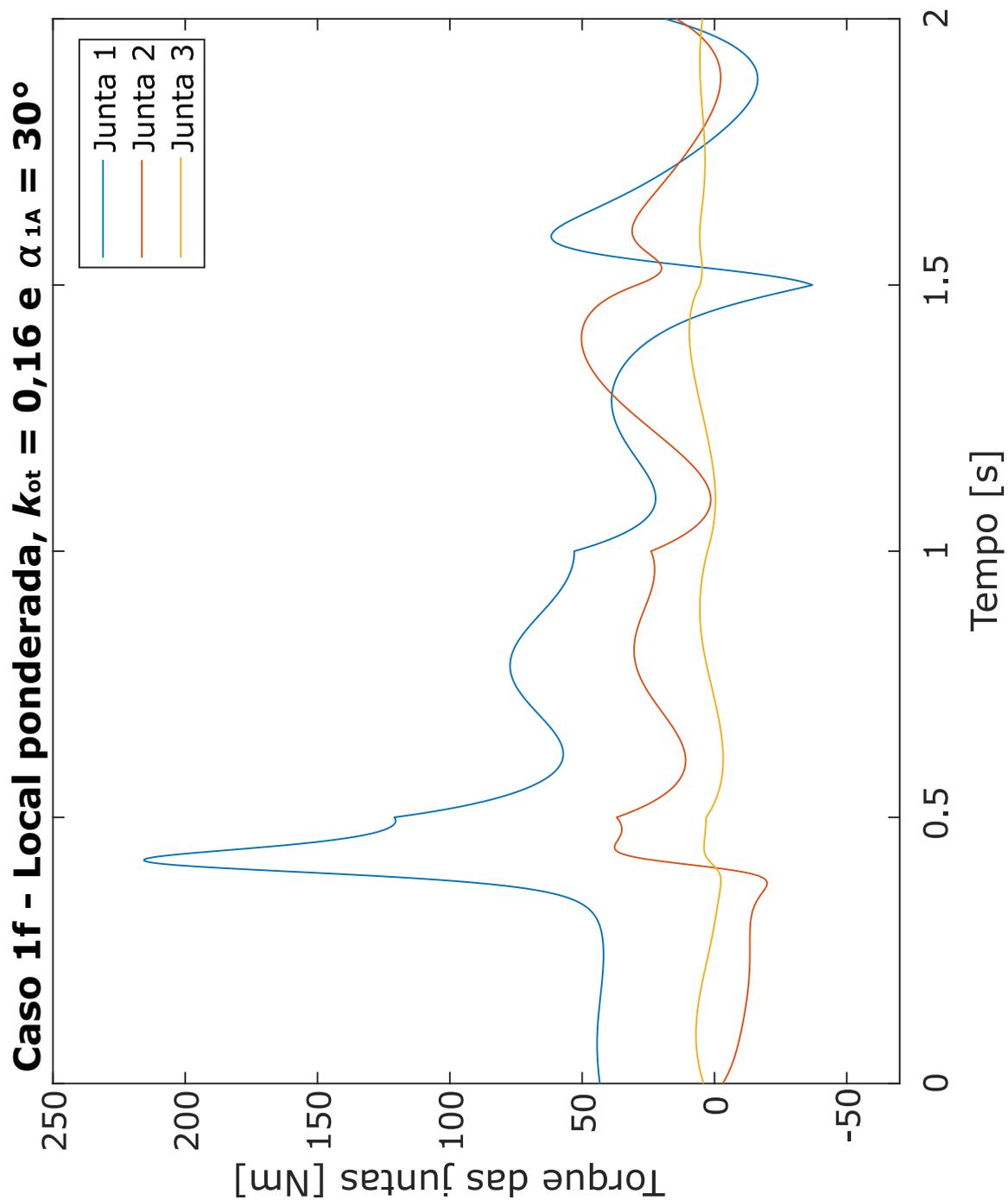
Fonte: Elaborado pelo autor (2022).

Figura 16 – Torque das juntas do manipulador redundante do caso 1e.



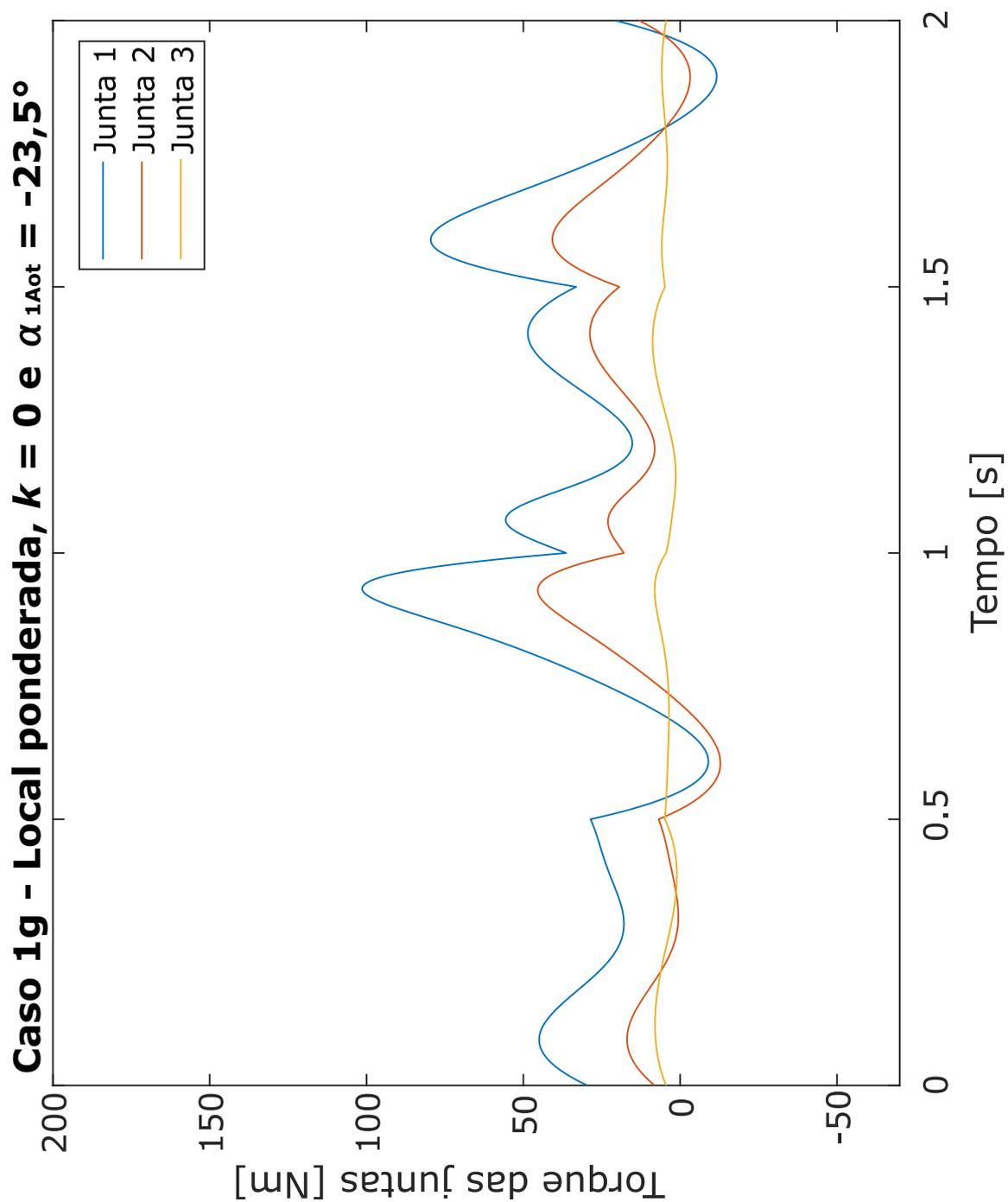
Fonte: Elaborado pelo autor (2022).

Figura 17 – Torque das juntas do manipulador redundante do caso 1f.



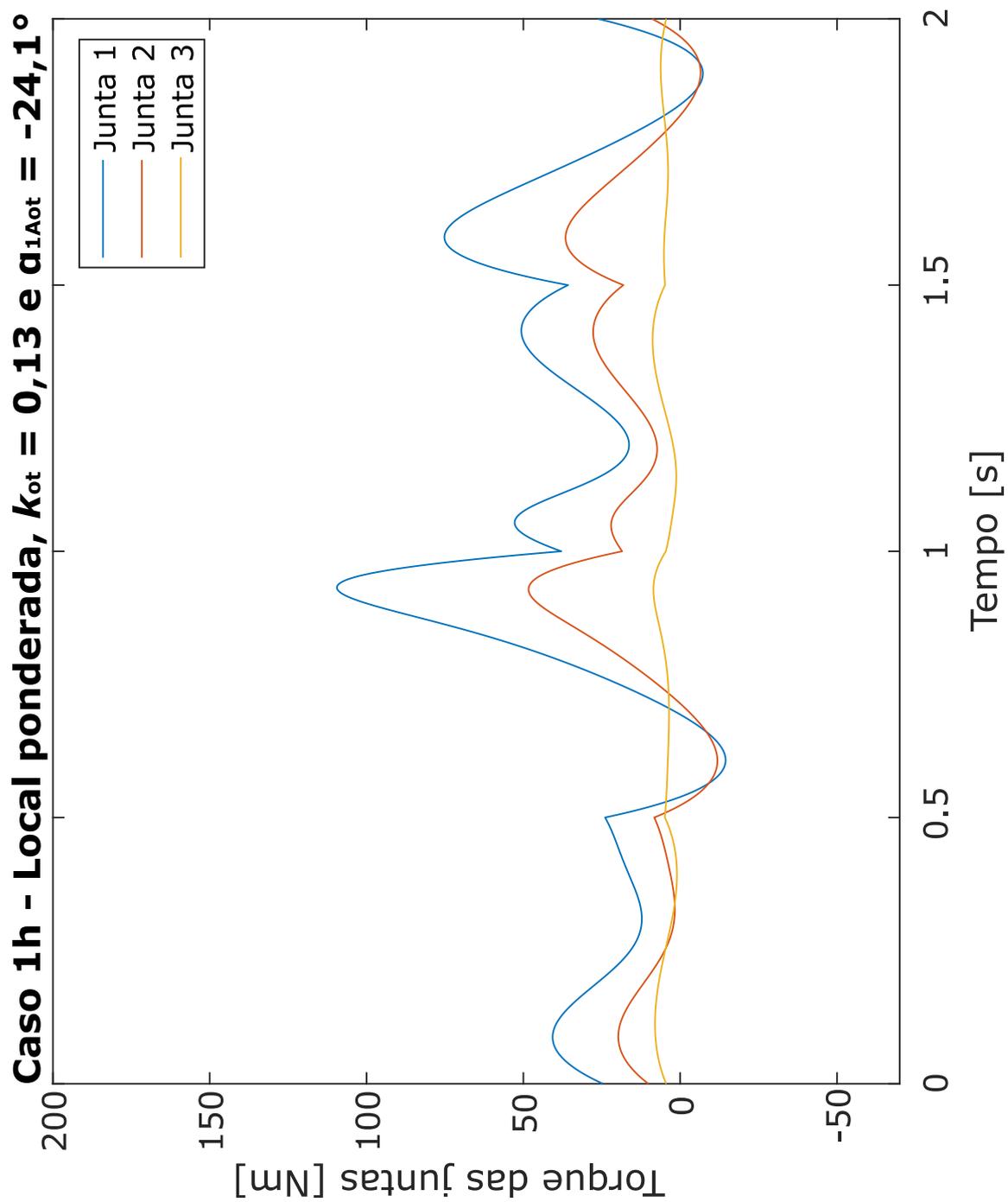
Fonte: Elaborado pelo autor (2022).

Figura 18 – Torque das juntas do manipulador redundante do caso 1g.



Fonte: Elaborado pelo autor (2022).

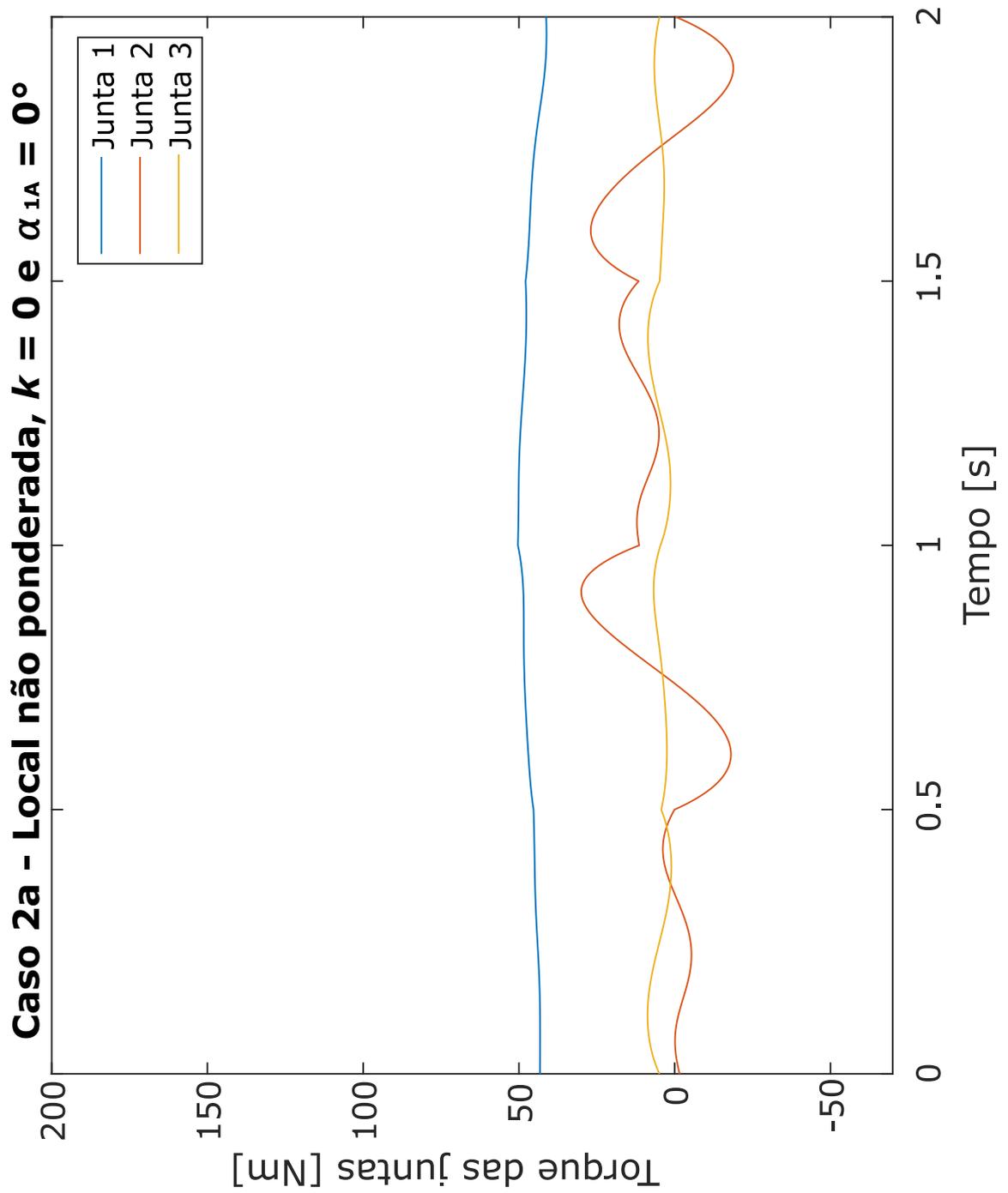
Figura 19 – Torque das juntas do manipulador redundante do caso 1h.



Fonte: Elaborado pelo autor (2022).

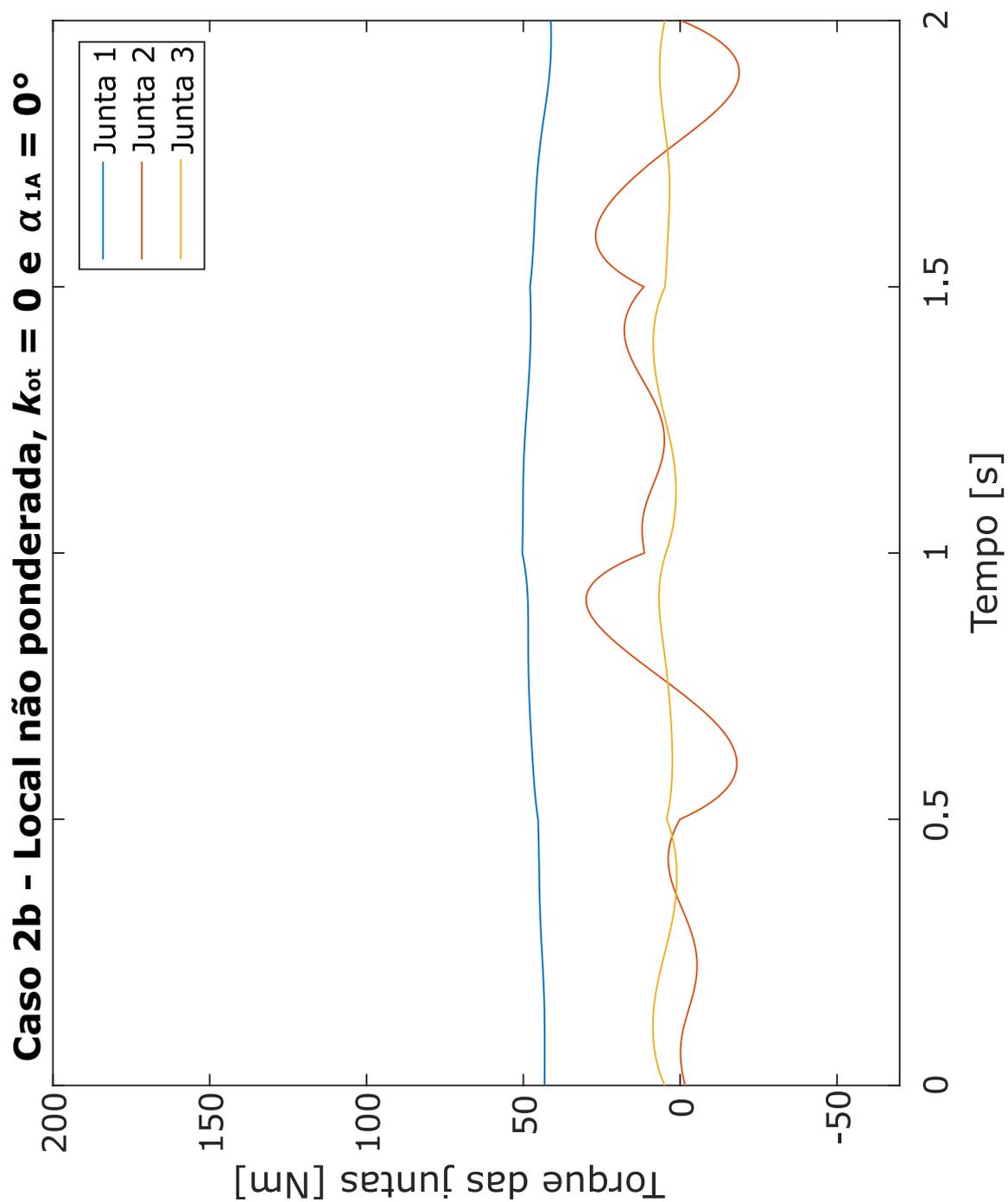
APÊNDICE B – Gráficos dos torques das juntas do manipulador redundante do caso 2

Figura 20 – Torque das juntas do manipulador redundante do caso 2a.



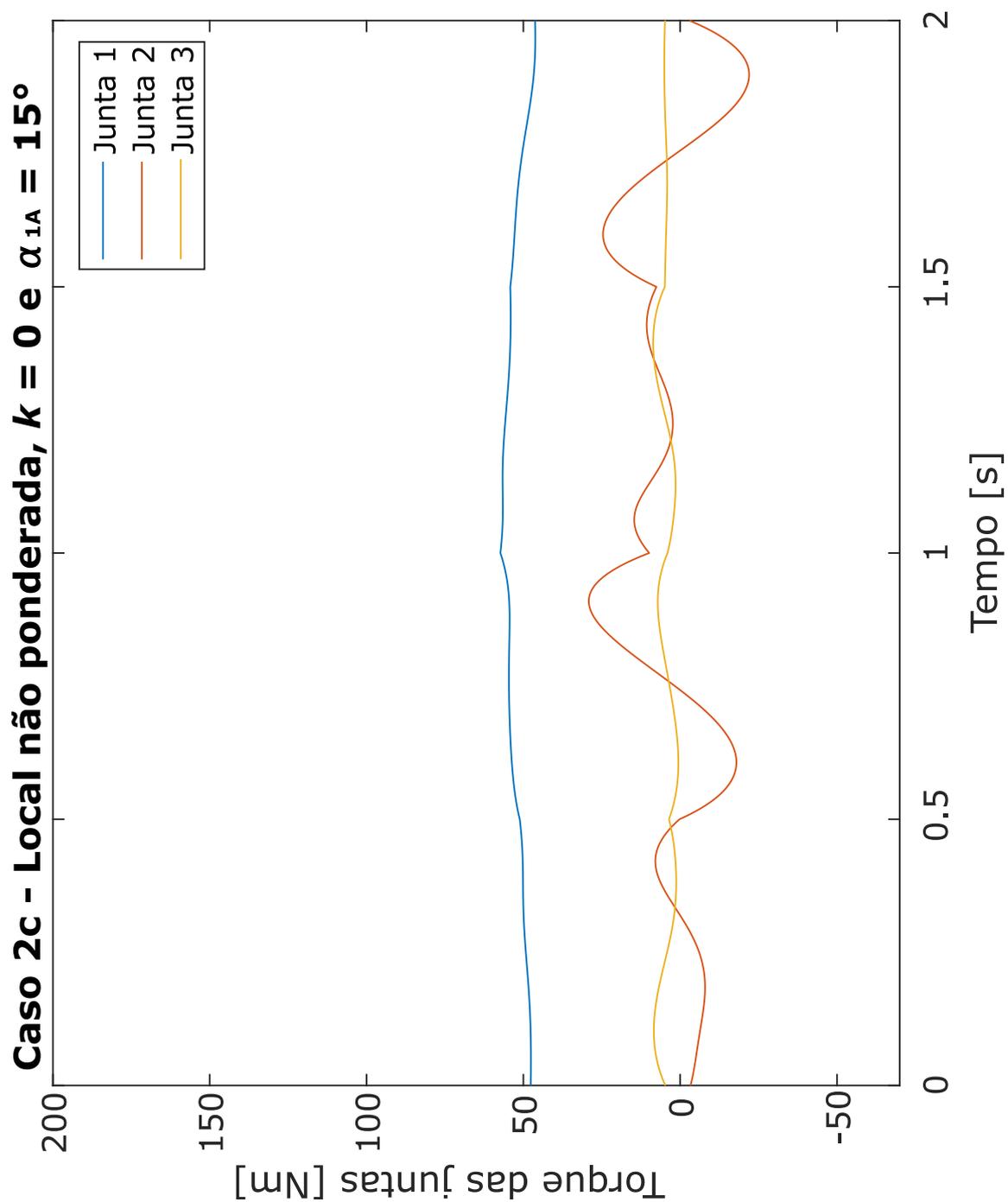
Fonte: Elaborado pelo autor (2022).

Figura 21 – Torque das juntas do manipulador redundante do caso 2b.



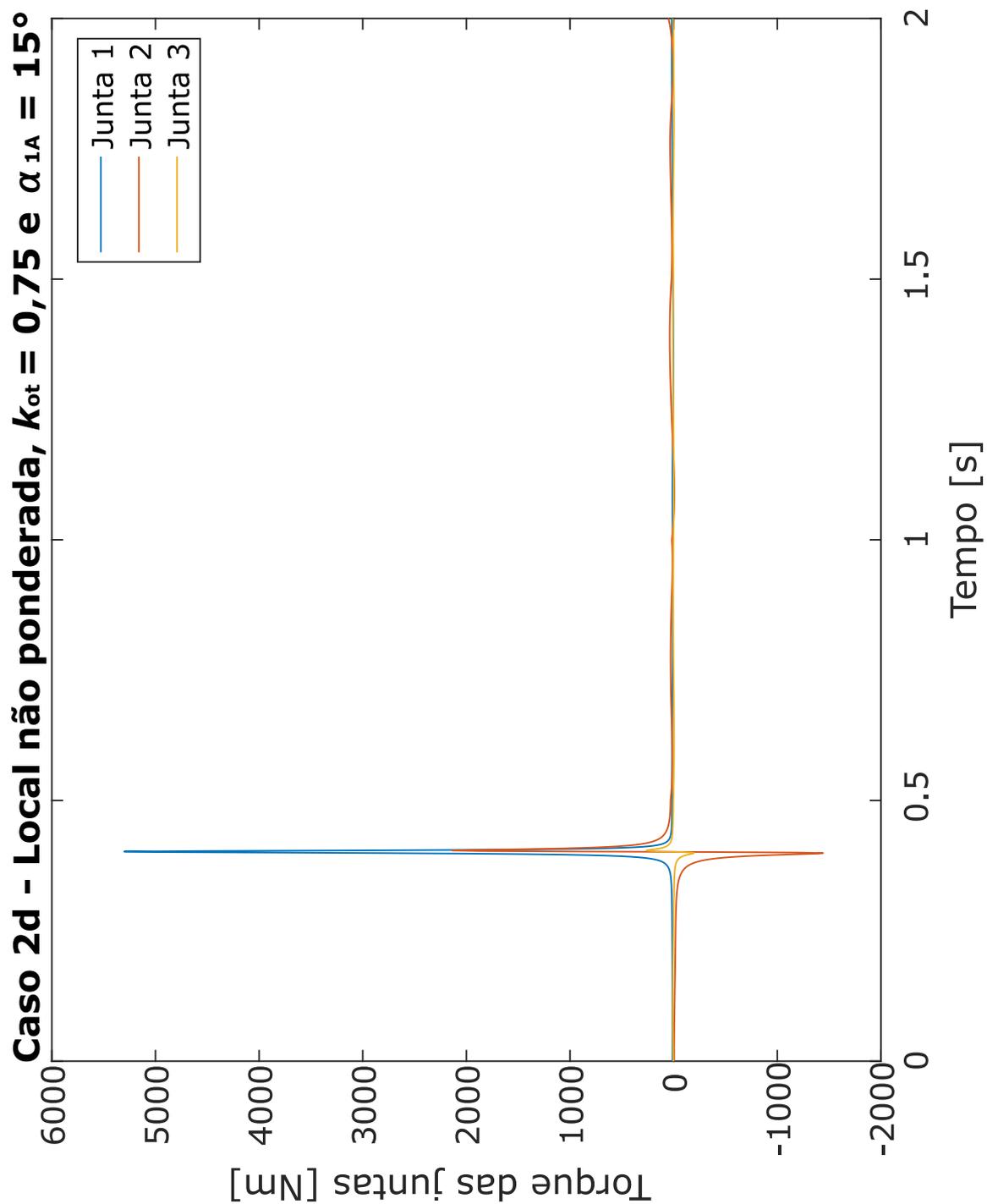
Fonte: Elaborado pelo autor (2022).

Figura 22 – Torque das juntas do manipulador redundante do caso 2c.



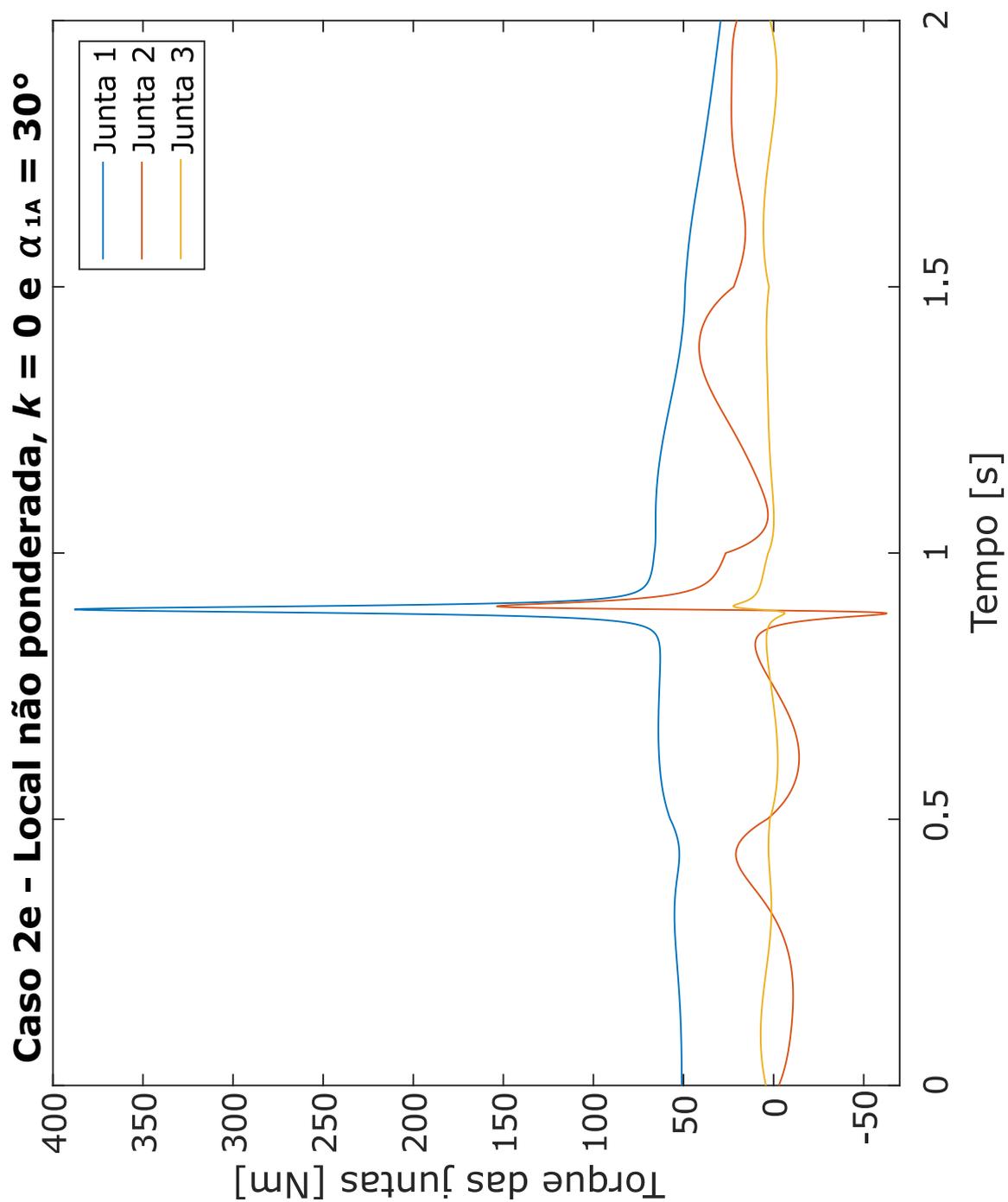
Fonte: Elaborado pelo autor (2022).

Figura 23 – Torque das juntas do manipulador redundante do caso 2d.



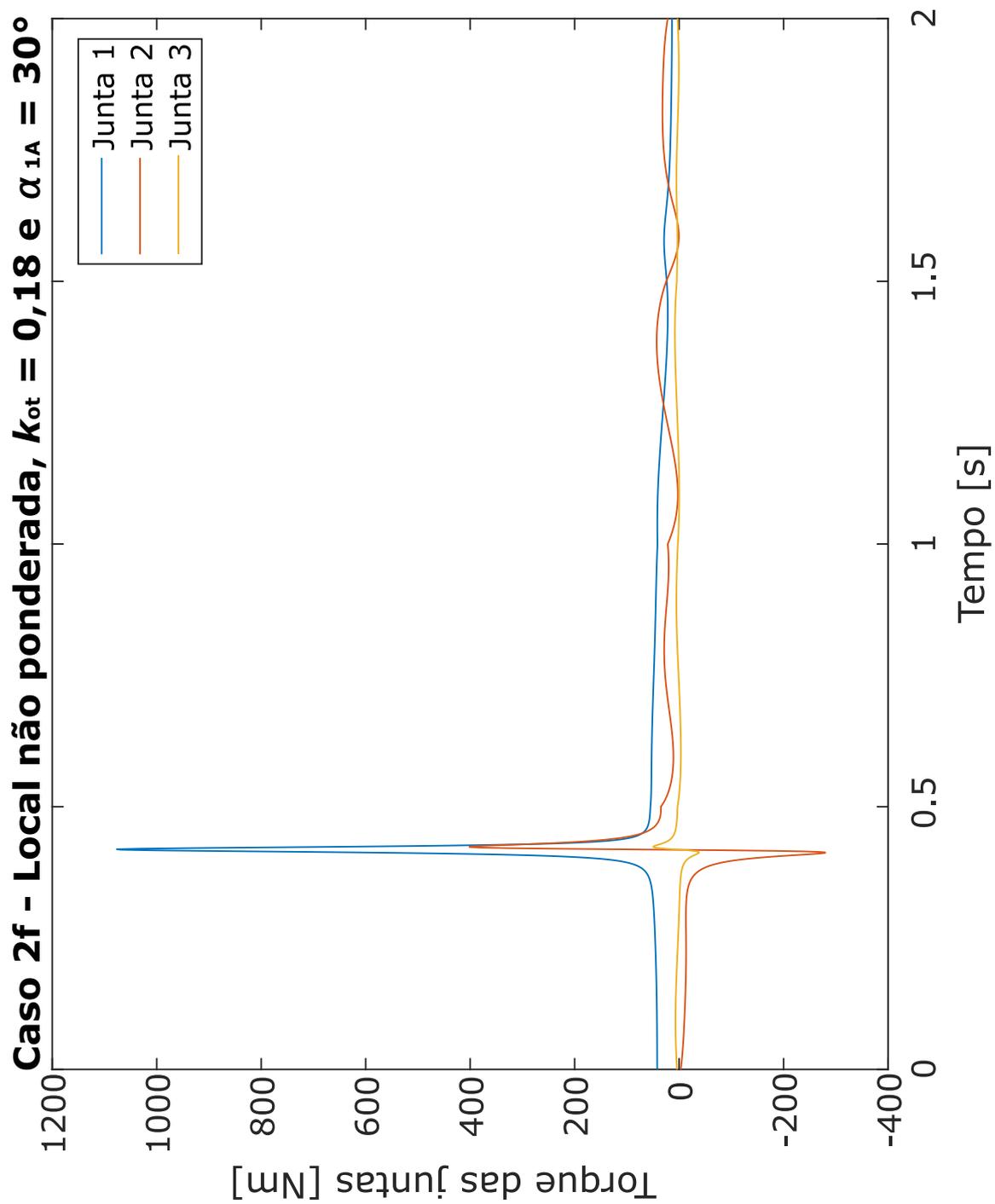
Fonte: Elaborado pelo autor (2022).

Figura 24 – Torque das juntas do manipulador redundante do caso 2e.



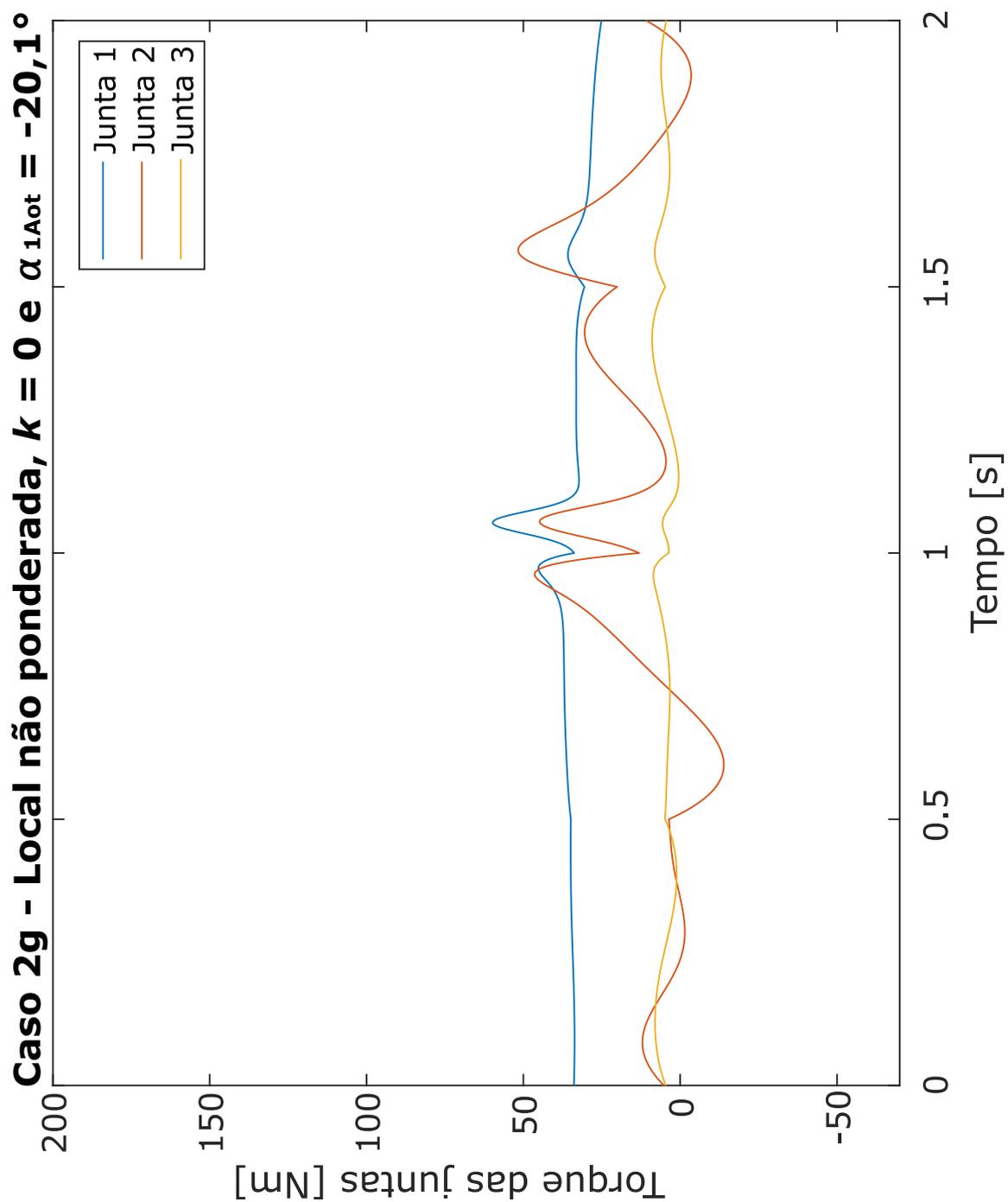
Fonte: Elaborado pelo autor (2022).

Figura 25 – Torque das juntas do manipulador redundante do caso 2f.



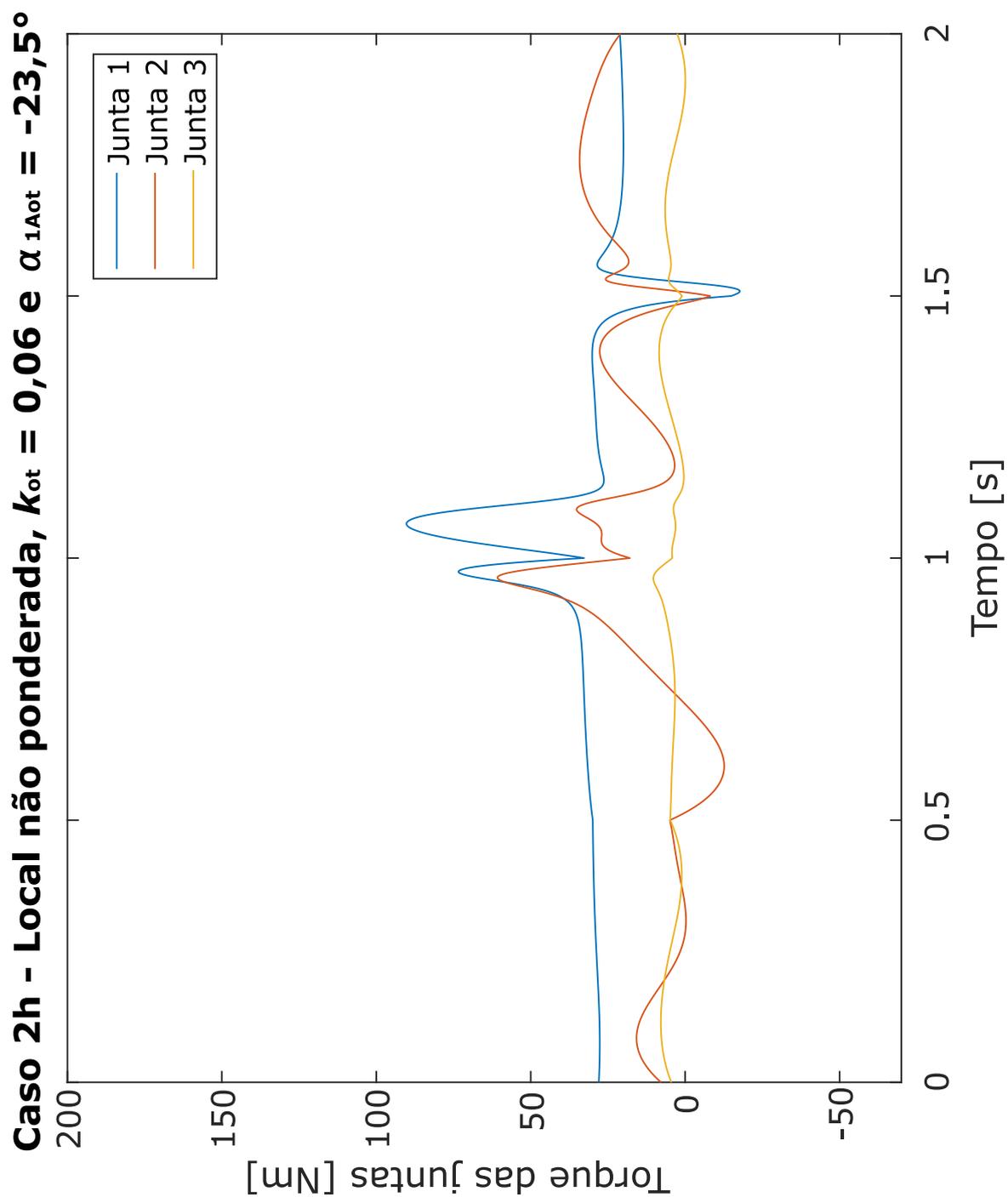
Fonte: Elaborado pelo autor (2022).

Figura 26 – Torque das juntas do manipulador redundante do caso 2g.



Fonte: Elaborado pelo autor (2022).

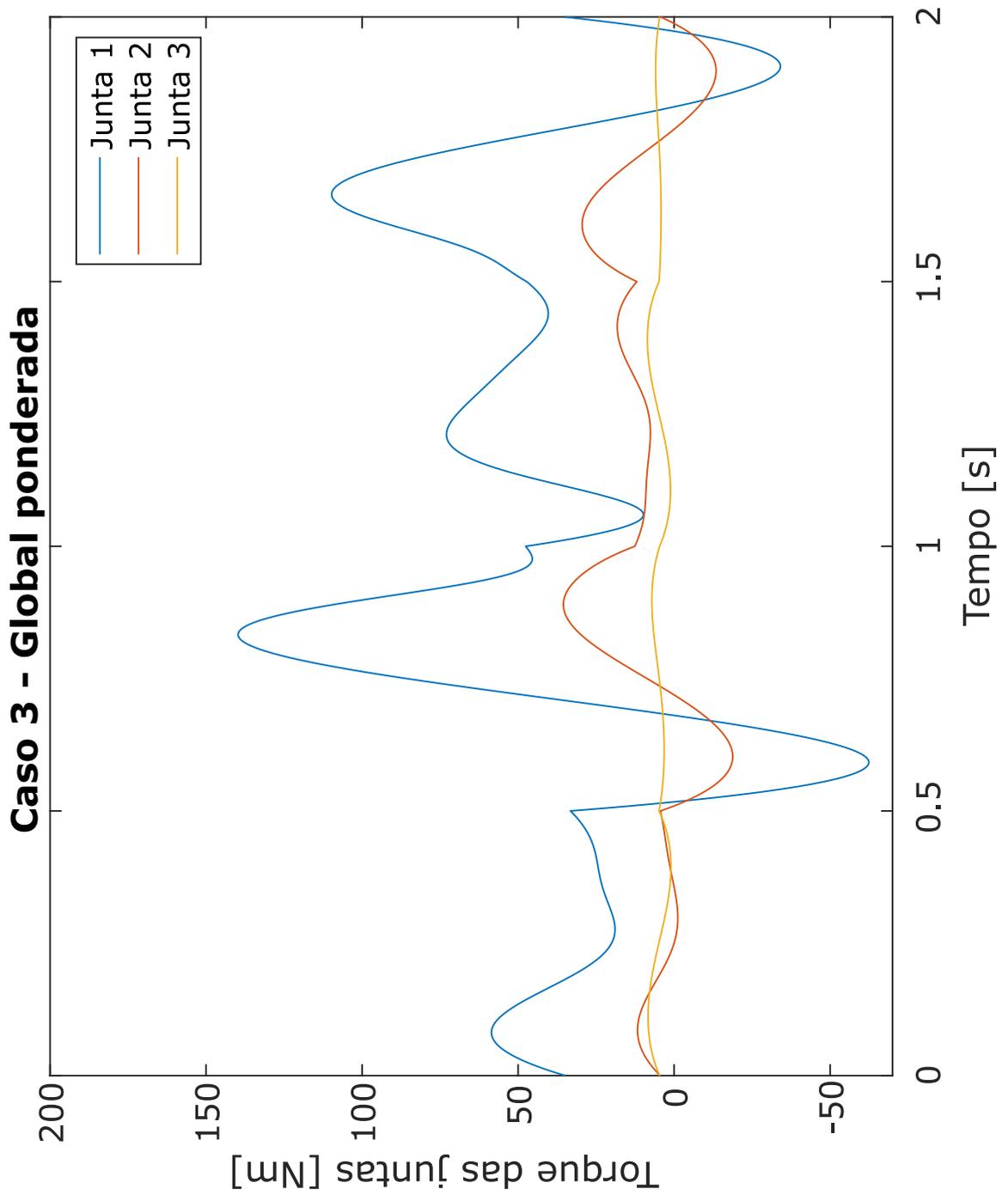
Figura 27 – Torque das juntas do manipulador redundante do caso 2h.



Fonte: Elaborado pelo autor (2022).

APÊNDICE C – Gráfico dos torques das juntas do manipulador redundante do caso 3

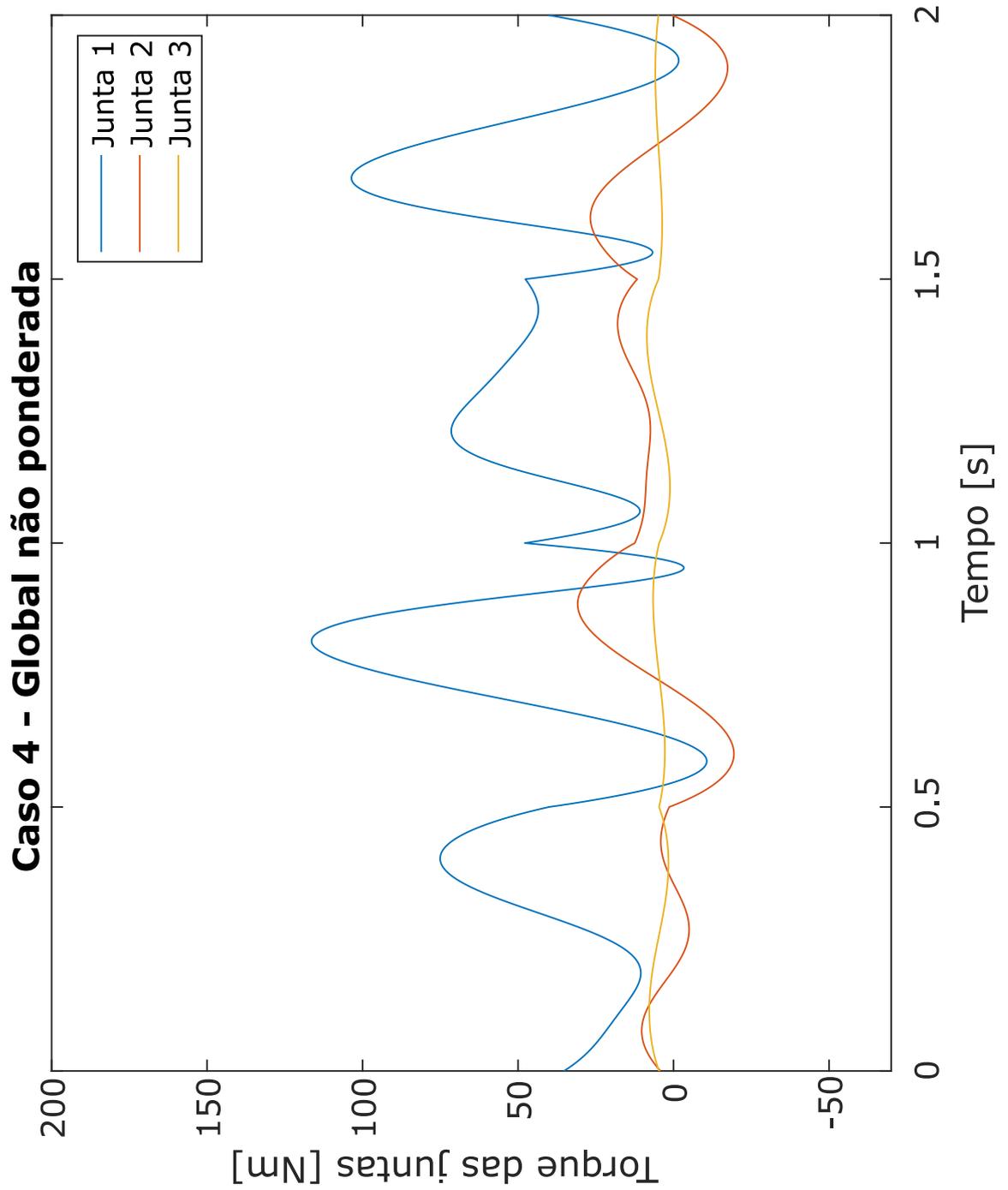
Figura 28 – Torque das juntas do manipulador redundante do caso 3.



Fonte: Elaborado pelo autor (2022).

APÊNDICE D – Gráfico dos torques das juntas do manipulador redundante do caso 4

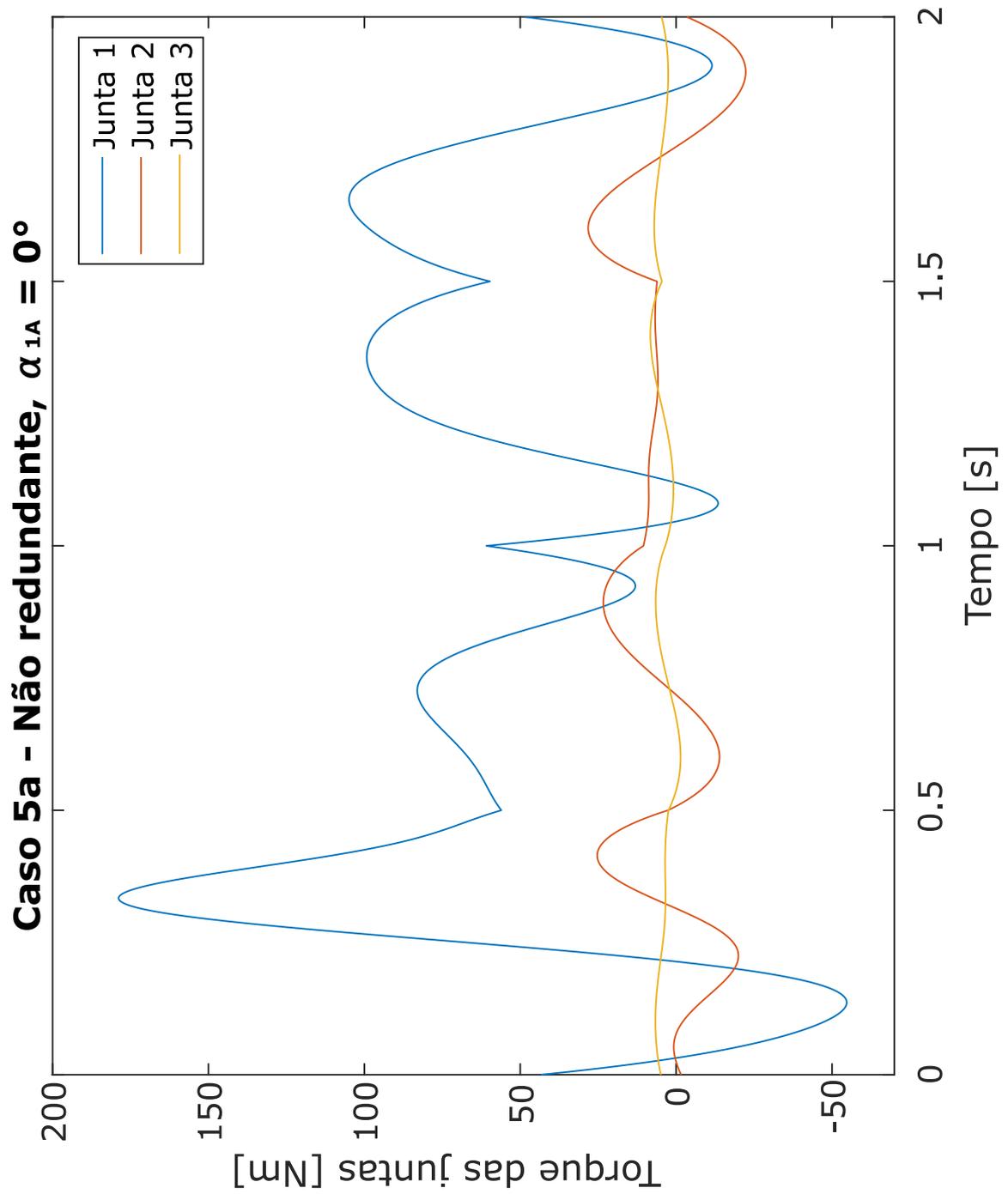
Figura 29 – Torque das juntas do manipulador redundante do caso 4.



Fonte: Elaborado pelo autor (2022).

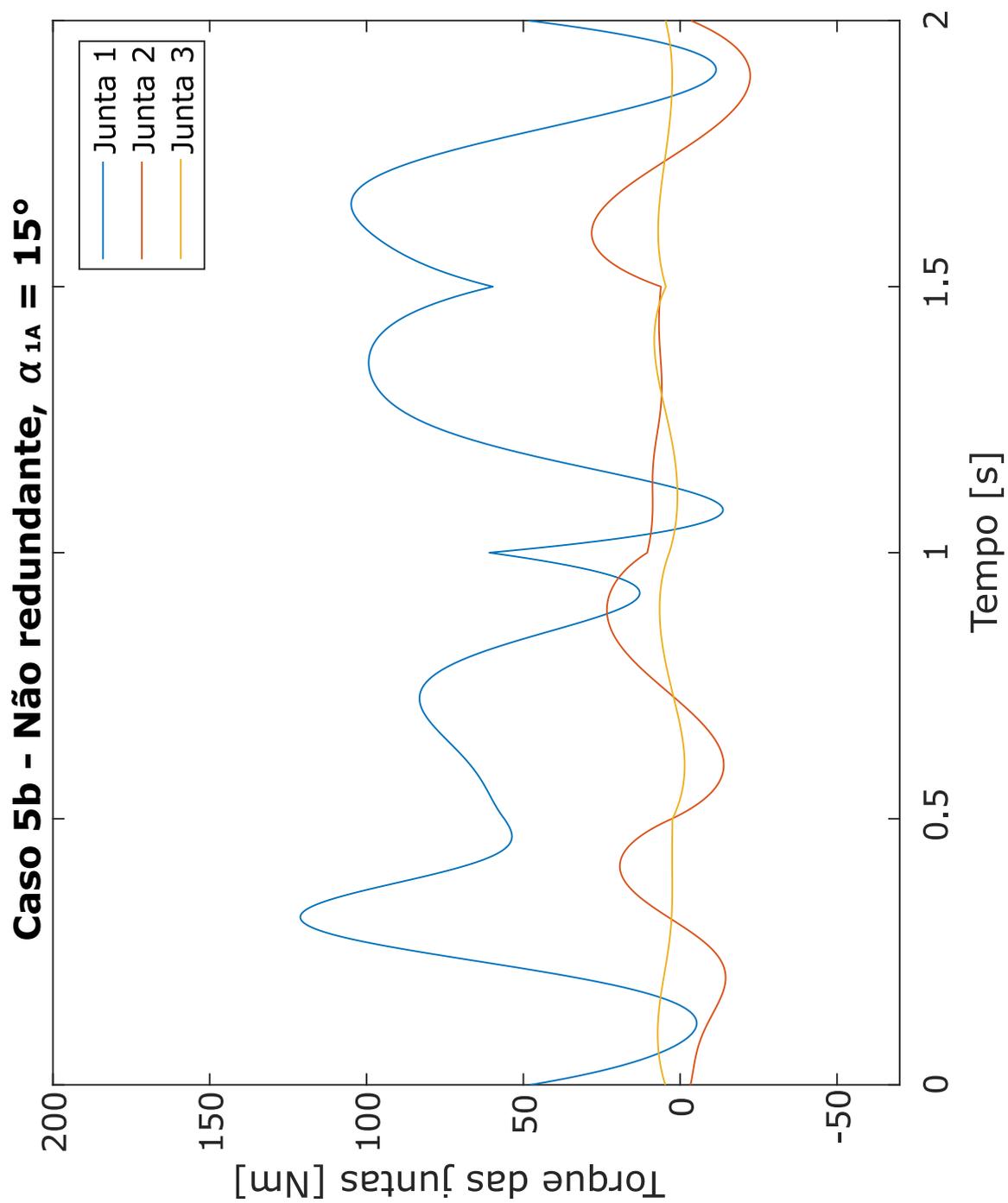
APÊNDICE E – Gráficos dos torques das juntas do manipulador não redundante do caso 5

Figura 30 – Torque das juntas do manipulador não redundante do caso 5a.



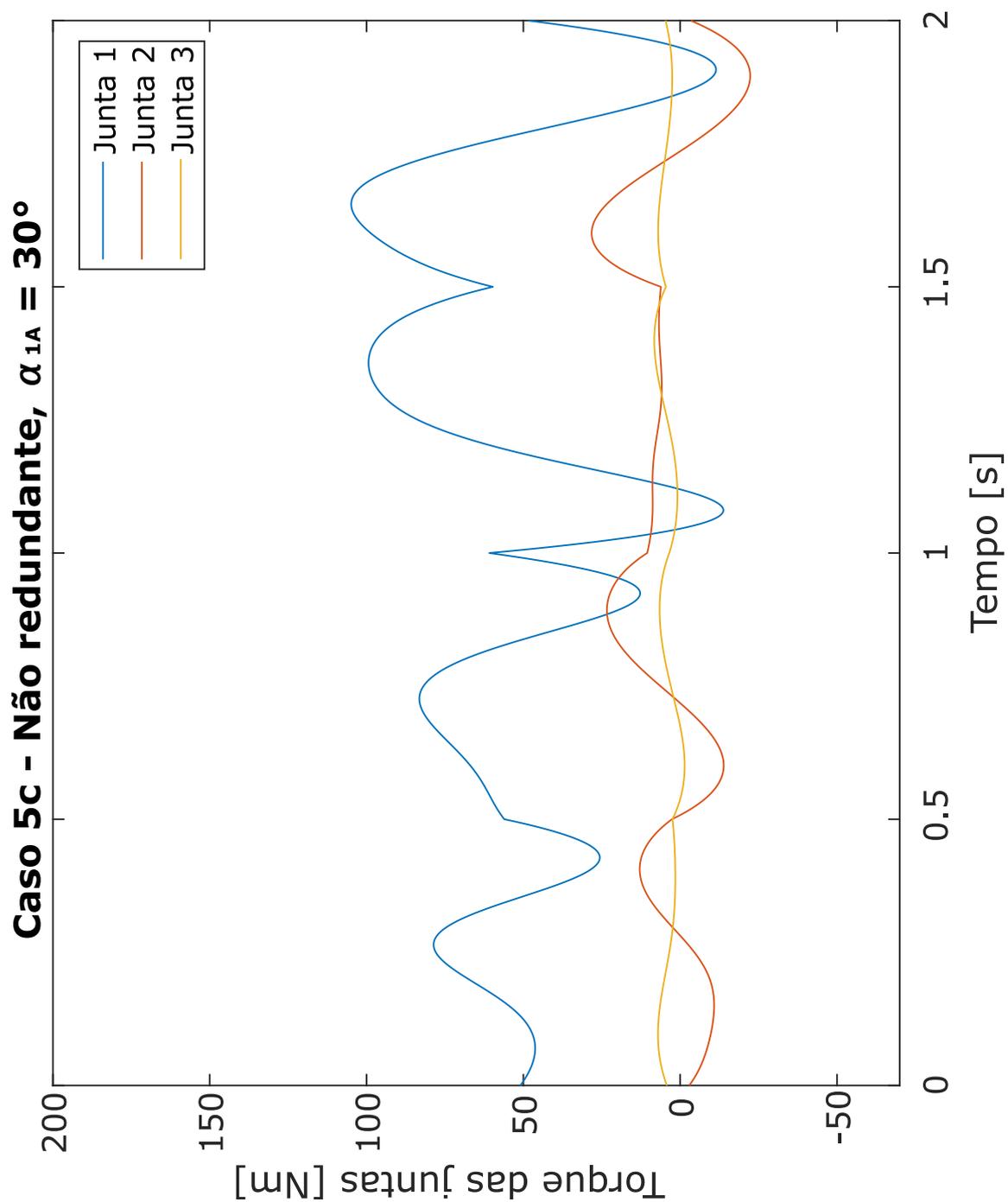
Fonte: Elaborado pelo autor (2022).

Figura 31 – Torque das juntas do manipulador não redundante do caso 5b.



Fonte: Elaborado pelo autor (2022).

Figura 32 – Torque das juntas do manipulador não redundante do caso 5c.



Fonte: Elaborado pelo autor (2022).

APÊNDICE F – Exemplo do código utilizado na resolução da redundância do caso 1h

```
close all; clear all; clc
```

```
%Otimização do manipulador
```

```
lb = [0, -pi]; %limitante inferior para o coeficiente K
ub = [1, pi]; %limitante superior para o coeficiente K
A = []; %matriz dos coeficientes da função de restrição
b = []; %vetor dos valores da função de restrição
Aeq = []; %matriz dos coeficientes da restrição de desigualdade
beq = []; %vetor dos valores da restrição de desigualdade
nonlcon = []; %restrição não linear
```

```
x0 = [0.112063169479370 -0.423897504806519]; %chute de K e alpha otimizado por GA
```

```
options = optimoptions('fmincon','Display','iter','Algorithm','sqp');
D = fmincon(@FCusto_Otim_K_AlphaInicial,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
```

```
%=====
```

```
%coeficiente K resultante da otimização
```

```
K = D(1); %valor de ponderação da gravidade
alpha = D(2); %orientacao do efetuador
```

```
%função para cálculo da cinemática inversa
```

```
Teta_FC = Cinematica_Inversa(alpha);
```

```
%orientações iniciais das juntas do manipulador
```

```
Teta1 = Teta_FC(1); Teta2 = Teta_FC(2); Teta3 = Teta_FC(3);
```

```
%velocidades iniciais das juntas do manipulador
```

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

```
%vetor das velocidades iniciais das juntas do manipulador
```

```
Tetadot = [Teta1dot; Teta2dot; Teta3dot];
```

```
%velocidade inicial
```

```
Tetadot = [0; 0; 0];
```

```
L1 = 1; %comprimento do elo 1 em metros
```

```
L2 = 1; %comprimento do elo 2 em metros
```

```
L3 = 1; %comprimento do elo 3 em metros
```

```
lc1 = L1/2; lc2 = L2/2; lc3 = L3/2; %posição dos cgs
```

```
b = 0.1; %espessura da barra do link
```

```

m1 = 5;      m2 = 2;      m3 = 1;  %massas dos elos [kg]
g = 9.81;    %aceleração da gravidade [m/s²]

%momentos de inercia dos elos
Iz1 = (1/12)*m1*(L1^2 + b^2);
Iz2 = (1/12)*m2*(L2^2 + b^2);
Iz3 = (1/12)*m3*(L3^2 + b^2);

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

for t = 0:dt:2 %tempo

    %Calcular a posicao das duas articulacoes do meio (para usar no plot)
    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);

    %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, dado uma matriz peso sendo a matriz de inercia
JM = inv(M)*J'(inv(J*inv(M)*J'));

%posicao do end-effector no tempo t
xex(counter) = L1*cos(Teta1)+L2*cos(Teta1+Teta2)+L3*cos(Teta1+Teta2+Teta3);
xey(counter) = L1*sin(Teta1)+L2*sin(Teta1+Teta2)+L3*sin(Teta1+Teta2+Teta3);
alpha(counter) = Teta1 + Teta2 + Teta3;

%posicao desejada do efetuator no tempo t

if t <= 0.5

%vetor posicao desejada
xd = [1.5; 1.5 + 40*t^3 - 120*t^4 + 96*t^5];

```

```

    xdx = 1.5;
    xdy = 1.5 + 40*t^3 - 120*t^4 + 96*t^5;

    %derivada da posicao desejada do efetuator
    xdxdot = 0;
    xdydot = 120*t^2 - 480*t^3 + 480*t^4;
    xddot = [xdxdot; xdydot];

    %segunda derivada da posicao desejada do efetuator
    xdx2dot = 0;
    xdy2dot = 240*t^1 - 1440*t^2 + 1920*t^3;
    xd2dot = [xdx2dot; xdy2dot];

end

if t > 0.5 && t<= 1

    %vetor posicao desejada
    xd = [1.5 + 40*(t-0.5)^3 - 120*(t-0.5)^4 + 96*(t-0.5)^5; 2];
    xdx = 1.5 + 40*(t-0.5)^3 - 120*(t-0.5)^4 + 96*(t-0.5)^5;
    xdy = 2;

    %derivada da posicao desejada do efetuator
    xdxdot = 120*(t-0.5)^2 - 480*(t-0.5)^3 + 480*(t-0.5)^4;
    xdydot = 0;
    xddot = [xdxdot; xdydot];

    %segunda derivada da posicao desejada do efetuator
    xdx2dot = 240*(t-0.5)^1 - 1440*(t-0.5)^2 + 1920*(t-0.5)^3;
    xdy2dot = 0;
    xd2dot = [xdx2dot; xdy2dot];

end

if t > 1 && t<= 1.5

    %vetor posicao desejada
    xd = [2; 2 - 40*(t-1)^3 + 120*(t-1)^4 - 96*(t-1)^5];
    xdx = 2;
    xdy = 2 - 40*(t-1)^3 + 120*(t-1)^4 - 96*(t-1)^5;

    %derivada da posicao desejada do efetuator
    xdxdot = 0;
    xdydot = -120*(t-1)^2 + 480*(t-1)^3 - 480*(t-1)^4;
    xddot = [xdxdot; xdydot];

    %segunda derivada da posicao desejada do efetuator

```

```

    xdx2dot = 0;
    xdy2dot = -240*(t-1)^1 + 1440*(t-1)^2 - 1920*(t-1)^3;
    xd2dot = [xdx2dot; xdy2dot];

end

if t > 1.5 && t <= 2

    %vetor posicao desejada
    xdx = 2 - 40*(t-1.5)^3 + 120*(t-1.5)^4 - 96*(t-1.5)^5;
    xdy = 1.5;
    xd = [2 - 40*(t-1.5)^3 + 120*(t-1.5)^4 - 96*(t-1.5)^5; 1.5];

    %derivada da posicao desejada do efetuador
    xdxdot = -120*(t-1.5)^2 + 480*(t-1.5)^3 - 480*(t-1.5)^4;
    xdydot = 0;
    xddot = [xdxdot; xdydot];

    %segunda derivada da posicao desejada do efetuador
    xdx2dot = -240*(t-1.5)^1 + 1440*(t-1.5)^2 - 1920*(t-1.5)^3;
    xdy2dot = 0;
    xd2dot = [xdx2dot; xdy2dot];

end

%calculo da aceleração das juntas no tempo t
Tetadot2 = JM*(xd2dot - Jdot*Tetadot) - (I - (JM*J))*(inv(M))*(C + K*G);

%calculo da velocidade das juntas no tempo t
Tetadot = Tetadot + dt*Tetadot2;      %[Teta1dot; Teta2dot; Teta3dot];

%calculo da posição das juntas no tempo t
Teta1 = Teta1 + dt*Tetadot(1,1);      %orientação de Teta1 no tempo t
Teta2 = Teta2 + dt*Tetadot(2,1);      %orientação de Teta2 no tempo t
Teta3 = Teta3 + dt*Tetadot(3,1);      %orientação de Teta3 no tempo t

%Calculando o Torque
Torque = M*Tetadot2 + C + G;          %Vetor dos Torques das juntas no tempo t

%Resultados
Teta1Result(counter) = Teta1;          %vetor com a orientacao de Teta1
Teta2Result(counter) = Teta2;          %vetor com a orientacao de Teta2
Teta3Result(counter) = Teta3;          %vetor com a orientacao de Teta3
Teta1dotResult(counter) = Tetadot(1,1); %vetor com a velocidade de Teta1
Teta2dotResult(counter) = Tetadot(2,1); %vetor com a velocidade de Teta2
Teta3dotResult(counter) = Tetadot(3,1); %vetor com a velocidade de Teta3
Teta1dot2Result(counter) = Tetadot2(1,1); %vetor com a aceleração de Teta1

```

```

Teta2dot2Result(counter) = Tetadot2(2,1);           %vetor com a aceleração de Teta2
Teta3dot2Result(counter) = Tetadot2(3,1);           %vetor com a aceleração de Teta3
Tau1Result(counter) = Torque(1,1);                  %vetor com o torque da junta 1
Tau2Result(counter) = Torque(2,1);                  %vetor com o torque da junta 2
Tau3Result(counter) = Torque(3,1);                  %vetor com o torque da junta 3

%Calculo da media dos torques
E_soma(counter) = sqrt(Torque'*Torque);
E_media = mean(E_soma);

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

end

%Norma infinita de cada uma das juntas
Norma_inf_individual = [norm(Tau1Result,inf) norm(Tau2Result,inf) norm(Tau3Result,inf)]

%Media dos torques
E_media

%plotar 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),xex(k)], [pL2y(k),xey(k)], 'b','linewidth',1);
        plot(xex(k),xey(k), 'bo', 'markersize',10);
        title ('Simulação do manipulador')
        xlabel ('[m]')
        ylabel ('[m]')
        xlim([-1 3]);ylim([-1 3]);
        grid on
        drawnow;
        hold off
    end
end

%plotar os torque de cada junta ao longo da trajetoria
figure(2)
plot(time,Tau1Result)
hold on
plot(time,Tau2Result)

```

```
hold on
plot(time,Tau3Result)
legend('Junta 1', 'Junta 2', 'Junta 3')
title ('Torque das juntas')
xlabel ('Tempo [s]')
ylabel ('Torque das juntas [Nm]')
ylim([-70 200]);
```

APÊNDICE G – Exemplo do código para cálculo da cinemática inversa

```

function Teta = Cinematica_Inversa(x)

    alpha = x(1);      %orientacao do efetuador em relacao ao eixo x

    %comprimento dos elos em metros
    L1 = 1;
    L2 = 1;
    L3 = 1;

    %posição e orientacao do efetuador
    xe = 1.5;         %posicao do efetuador no eixo x
    ye = 1.5;         %posicao do efetuador no eixo y

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

    c2 = (d - 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 = alpha - Teta1 - Teta2;

    Teta = [Teta1 Teta2 Teta3];

end

```

APÊNDICE H – Exemplo do código da função *ga* para o caso 1h

```
clc, clear all, close all

nvars = 2;      %quantidade de variáveis de entrada (2, k e alpha)
lb = [0, -pi]; %limitante inferior para K e alpha
ub = [1, pi];  %limitante superior para K e alpha
A = [];        %matriz dos coeficientes da função de restrição
b = [];        %vetor dos valores da função de restrição
Aeq = [];      %matriz dos coeficientes da restrição de desigualdade
beq = [];      %vetor dos valores da restrição de desigualdade
nonlcon = [];  %restrição não linear

options = optimoptions('ga','Display','iter','PlotFcn',@gaplotbestf)
[X,fval] = ga(@FCusto_Otim_K_AlphaInicial,nvars,A,b,Aeq,beq,lb,ub,nonlcon,options)
```

APÊNDICE I – Exemplo do código da função custo para o caso 1h

```

function E = FCusto_Otim_AlphaInicial(x)

    %coeficiente K resultante da otimização
    K = x(1);          %fator que multiplica a gravidade
    alpha = x(2);     %orientacao do efetuador em relacao ao eixo x

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

    %posição dos cgs dos elos
    lc1 = L1/2;
    lc2 = L2/2;
    lc3 = L3/2;

    %posição e orientacao do efetuador
    xe = 1.5;        %posicao do efetuador no eixo x
    ye = 1.5;        %posicao do efetuador no eixo y

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

    c2 = (d - L1^2 - L2^2)/(2*L1*L2);
    s2 = sqrt(1-(c2)^2);

    %Penalização para alphas que levam à singularidade
    if ~isreal(s2)
        num = rand;
        E = num*10e09;
        return
    end

    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 = alpha - Teta1 - Teta2;

    %velocidade inicial

```

```

Tetadot = [0; 0; 0];

m1 = 5;      m2 = 2;      m3 = 1;  %massas dos elos [kg]
g = 9.81;    %aceleração da gravidade [m/s^2]

%momentos de inercia dos links
Iz1 = (1/12)*m1*(L1^2 + b^2);
Iz2 = (1/12)*m2*(L2^2 + b^2);
Iz3 = (1/12)*m3*(L3^2 + b^2);

I = eye(3);      %Matriz Identidade 3x3
dt = 0.001;      %step time
counter = 1;

for t = 0:dt:2 %tempo

%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, dado uma matriz peso sendo a matriz de inercia
JM = inv(M)*J*(inv(J*inv(M)*J));

%posicao do efetuador no tempo t
xex = L1*cos(Teta1)+L2*cos(Teta1+Teta2)+L3*cos(Teta1+Teta2+Teta3);
yex = L1*sin(Teta1)+L2*sin(Teta1+Teta2)+L3*sin(Teta1+Teta2+Teta3);
alpha = Teta1 + Teta2 + Teta3;

%posicao desejada do efetuador no tempo t

if t <= 0.5

%vetor posicao desejada
xe = [1.5; 1.5 + 40*t^3 - 120*t^4 + 96*t^5];
xex = 1.5;
yex = 1.5 + 40*t^3 - 120*t^4 + 96*t^5;

%derivada da posicao desejada do efetuador
xexdot = 0;
yexdot = 120*t^2 - 480*t^3 + 480*t^4;

```

```
xedot = [xexdot; xeydot];

%segunda derivada da posicao desejada do efetuator
xex2dot = 0;
xey2dot = 240*t^1 - 1440*t^2 + 1920*t^3;
xe2dot = [xex2dot; xey2dot];

end

if t > 0.5 && t<= 1

    %vetor posicao desejada
    xe = [1.5 + 40*(t-0.5)^3 - 120*(t-0.5)^4 + 96*(t-0.5)^5; 2];
    xex = 1.5 + 40*(t-0.5)^3 - 120*(t-0.5)^4 + 96*(t-0.5)^5;
    xey = 2;

    %derivada da posicao desejada do efetuator
    xexdot = 120*(t-0.5)^2 - 480*(t-0.5)^3 + 480*(t-0.5)^4;
    xeydot = 0;
    xedot = [xexdot; xeydot];

    %segunda derivada da posicao desejada do efetuator
    xex2dot = 240*(t-0.5)^1 - 1440*(t-0.5)^2 + 1920*(t-0.5)^3;
    xey2dot = 0;
    xe2dot = [xex2dot; xey2dot];

end

if t > 1 && t<= 1.5

    %vetor posicao desejada
    xe = [2; 2 - 40*(t-1)^3 + 120*(t-1)^4 - 96*(t-1)^5];
    xex = 2;
    xey = 2 - 40*(t-1)^3 + 120*(t-1)^4 - 96*(t-1)^5;

    %derivada da posicao desejada do efetuator
    xexdot = 0;
    xeydot = -120*(t-1)^2 + 480*(t-1)^3 - 480*(t-1)^4;
    xedot = [xexdot; xeydot];

    %segunda derivada da posicao desejada do efetuator
    xex2dot = 0;
    xey2dot = -240*(t-1)^1 + 1440*(t-1)^2 - 1920*(t-1)^3;
    xe2dot = [xex2dot; xey2dot];

end
```

```

if t > 1.5 && t <= 2

    %vetor posicao desejada
    xex = 2 - 40*(t-1.5)^3 + 120*(t-1.5)^4 - 96*(t-1.5)^5;
    xey = 1.5;
    xe = [2 - 40*(t-1.5)^3 + 120*(t-1.5)^4 - 96*(t-1.5)^5; 1.5];

    %derivada da posicao desejada do efetuador
    xexdot = -120*(t-1.5)^2 + 480*(t-1.5)^3 - 480*(t-1.5)^4;
    xeydot = 0;
    xedot = [xexdot; xeydot];

    %segunda derivada da posicao desejada do efetuador
    xex2dot = -240*(t-1.5)^1 + 1440*(t-1.5)^2 - 1920*(t-1.5)^3;
    xey2dot = 0;
    xe2dot = [xex2dot; xey2dot];

end

    %calculo da aceleração das juntas no tempo t
    Tetadot2 = JM*(xe2dot - Jdot*Tetadot) - (I - (JM*J))*(inv(M))*(C + K*G);

    %calculo da velocidade das juntas no tempo t
    Tetadot = Tetadot + dt*Tetadot2;          %[Teta1dot; Teta2dot; Teta3dot];

    %calculo da posição das juntas no tempo t
    Teta1 = Teta1 + dt*Tetadot(1,1);          %orientação de Teta1 no tempo t
    Teta2 = Teta2 + dt*Tetadot(2,1);          %orientação de Teta2 no tempo t
    Teta3 = Teta3 + dt*Tetadot(3,1);          %orientação de Teta3 no tempo t

    %Calculando o Torque
    Torque = M*Tetadot2 + C + G;

    %Função Custo
    E_soma(counter) = sqrt(Torque'*Torque);
    E_media = mean(E_soma);

    counter = counter + 1; %contador de iterações

end

E = E_media;

end

```