

**UNIVERSIDADE DE CAXIAS DO SUL – UCS**  
**CAMPUS UNIVERSITÁRIO DA REGIÃO DOS VINHEDOS – CARVI**  
**ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIA**  
**CURSO DE ENGENHARIA ELETRÔNICA**

**JONATHAN CERBARO**

**COMPARATIVO ENTRE MÉTODOS JACOBIANOS PARA SOLUÇÃO DA  
CINEMÁTICA INVERSA DE UM MANIPULADOR BASEADO EM QUATÉRNIOS  
DUAIS**

**BENTO GONÇALVES**

**2019**

**JONATHAN CERBARO**

**COMPARATIVO ENTRE MÉTODOS JACOBIANOS PARA SOLUÇÃO DA  
CINEMÁTICA INVERSA DE UM MANIPULADOR BASEADO EM QUATÉRNIOS  
DUAIS**

Trabalho de Conclusão de Curso II  
apresentado ao Centro de Ciências  
Exatas, da Natureza e de Tecnologia da  
Universidade de Caxias do Sul como  
requisito parcial para obtenção do título de  
Engenheiro Eletrônico.

Orientador: Prof. Me. Patric Janner  
Marques

**BENTO GONÇALVES**

**2019**

**JONATHAN CERBARO**

**COMPARATIVO ENTRE MÉTODOS JACOBIANOS PARA SOLUÇÃO DA  
CINEMÁTICA INVERSA DE UM MANIPULADOR BASEADO EM QUATÉRNIOS  
DUAIS**

Trabalho de Conclusão de Curso II  
apresentado ao Centro de Ciências  
Exatas, da Natureza e de Tecnologia da  
Universidade de Caxias do Sul como  
requisito parcial para obtenção do título de  
Engenheiro Eletrônico.

Orientador: Prof. Me. Patric Janner  
Marques

**Aprovado em: 25/11/2019**

**Banca Examinadora:**

---

Prof. Me. Patric Janner Marques  
Universidade de Caxias do Sul – UCS

---

Profa. Dra. Marilda Machado Spindola  
Universidade de Caxias do Sul – UCS

---

Prof. Me. Maurício Cagliari Tosin  
Universidade de Caxias do Sul – UCS

Dedico este trabalho aos meus pais,  
Laurindo Cerbaro e Marliza Juliati Cerbaro,  
por terem transmitido a mim valores de  
honra e dever.

## **AGRADECIMENTOS**

Agradeço aos meus pais, Laurindo Cerbaro e Marliza Juliati Cerbaro, que compreenderam minha ausência e cansaço em tempos de estudo, não medindo esforços para me apoiar.

Agradeço a minha namorada e melhor amiga, Caroline Laís Casagrande, por ter estado ao meu lado durante todos esses anos, fornecendo apoio emocional e mostrando que nunca estive sozinho nessa jornada.

Agradeço a todos meus amigos, que forneceram muitos momentos de alegria, abrandado as dificuldades que enfrentei, e que souberam me aconselhar em momentos de necessidade.

Agradeço também aos professores da Universidade de Caxias do Sul que colaboraram no meu amadurecimento acadêmico. Agradecimento especial ao meu orientador, Patric Janner Marques, que soube guiar-me entre meus devaneios sendo justo em suas críticas e sábio em seus elogios.

E aos demais que, de alguma forma, mesmo sem saber, colaboraram com meu trabalho.

*“Não existe triunfo sem perda, não há vitória sem sofrimento, não há liberdade sem sacrifício.”*

**J.R.R. Tolkien**

## RESUMO

A cinemática inversa é um problema em aberto na robótica, devido à sua alta complexidade e não-linearidade. Comparativo entre os diversos métodos de solução desta problemática é tópico de tendência no meio acadêmico. Também há evidente interesse acadêmico na aplicação de ferramentas modernas para solução da cinemática direta, como o uso dos quatérnios duais unitários, em detrimento das clássicas matrizes de transformação homogêneas. Neste trabalho foram comparados quatro métodos Jacobianos para solução da cinemática inversa, descrevendo a estrutura de um manipulador de sete graus de liberdade através da álgebra dos quatérnios duais. Os métodos foram implementados e analisados em ambiente de simulação, através do *software* Matlab e foram coletadas métricas de desempenho como tempos de execução, taxa de sucesso, precisão posicional e comportamento do erro ao longo do tempo em cada método, avaliando 10 mil situações. Os resultados foram analisados no formato de tabelas e gráficos comparativos, com o objetivo de observar o desempenho de cada algoritmo nos diversos aspectos.

**Palavras-chave:** Cinemática inversa, métodos Jacobianos, quatérnios duais unitários, manipuladores robóticos, cálculo numérico.

## ABSTRACT

Inverse kinematics is an open problem in robotics, due to its high complexity and non-linearity. Comparative among the several methods of solution of this problematic is topic of tendency in the academic environment. There is also an evident academic interest in the application of modern tools for the solution of forward kinematics, such as the use of unit dual quaternions, to the detriment of classic homogeneous transformation matrices. In this work four Jacobian methods for the solution of inverse kinematics were compared, describing the structure of a manipulator with seven degrees of freedom through the dual quaternion algebra. The methods were implemented and analyzed in a simulation environment through Matlab software and performance metrics such as execution times, success rate, positional accuracy and error behavior over time for each method, evaluating 10,000 situations. The results were analyzed in the means of tables and comparative graphs, aiming to observe the performance of each algorithm in the various aspects.

**Keywords:** Inverse kinematics, Jacobian methods, unit dual quaternions, robotic manipulators, numerical calculation.

## LISTA DE FIGURAS

Figura 1 – Encomendas anuais de robôs industriais no mundo, por continente. ....	16
Figura 2 – Publicações anuais para “ <i>inverse kinematics 7 DoF manipulators</i> ”. ....	17
Figura 3 – Publicações anuais para “ <i>Jacobian inverse kinematics</i> ”. ....	17
Figura 4 – Publicações anuais para “ <i>dual quaternion kinematics</i> ”. ....	18
Figura 5 – Publicações anuais para “ <i>comparison inverse kinematics methods</i> ”. ....	19
Figura 6 – Estrutura de eixos do Kuka LWR. ....	25
Figura 7 – Eixo torcido para exemplificar os parâmetros DH. ....	26
Figura 8 – Representação dos parâmetros $d_i$ em um robô Kuka LWR. ....	28
Figura 9 – Comprimentos reais de um robô Kuka LWR. ....	29
Figura 10 – Cinemática direta de robô planar com 3 DoF. ....	30
Figura 11 – Múltiplas soluções de IK para um manipulador PUMA 560. ....	41
Figura 12 – Exemplo do espaço de trabalho de um manipulador planar de 2 DoF. ....	41
Figura 13 – Classificação dos métodos de solução para a IK. ....	43
Figura 14 – Aproximação linear da matriz Jacobiana. ....	44
Figura 15 – Comparativo da solução pela pseudoinversa e pelo DLS. ....	50
Figura 16 – Manipulador planar próximo de uma singularidade. ....	52
Figura 17 – Representação do erro de posição em relação ao tempo. ....	58
Figura 18 – Representação do erro de orientação em relação ao tempo. ....	58
Figura 19 – Comportamento do erro ao longo do tempo. ....	70
Figura 20 – Quantidade média de iterações exigidas em cada método. ....	72
Figura 21 – Tempo médio para computação da IK exigido em cada método. ....	73
Figura 22 – Tempo médio para solução de um problema exigido em cada método. ....	73
Figura 23 – Distribuição do tempo de solução dos problemas em cada método. ....	74
Figura 24 – Distribuição do tempo de solução dos problemas, em escala. ....	75
Figura 25 – Tempo de execução versus norma do erro inicial. ....	77

## LISTA DE TABELAS

Tabela 1 – Parâmetros DH de um manipulador antropomórfico de 7 DoF.....	27
Tabela 2 – Parâmetros DH do manipulador Kuka LWR. ....	29
Tabela 3 – Média de iterações em que o algoritmo atingiu a precisão.....	56
Tabela 4 – Média de iterações em que o algoritmo abortou. ....	56
Tabela 5 – Tempo computacional para solução de IK pelo Matlab, em $\mu s$ . ....	59
Tabela 6 – Métricas de desempenho da IK em robô cirúrgico de 8 DoF.....	60
Tabela 7 – Taxa de sucesso dos algoritmos. ....	70
Tabela 8 – Desempenho do método Jacobiana Transposta. ....	71
Tabela 9 – Desempenho do método Jacobiana Pseudoinversa. ....	71
Tabela 10 – Desempenho do método DLS. ....	71
Tabela 11 – Desempenho do método SDLS. ....	72
Tabela 12 – Coeficientes de variação para o tempo de execução em cada método.	75
Tabela 13 – Comparativo de menores tempos de solução entre os métodos.....	76
Tabela 14 – Correlação entre o tempo de execução do problema e o erro inicial.....	77
Tabela 15 – Contabilização das causas de parada por falha. ....	79
Tabela 16 – Iterações adicionais médias para amostras marcadas como falha. ....	79

## LISTA DE ABREVIATURAS E SIGLAS

DH	Denavit-Hartenberg
DLS	<i>Damped Least Squares</i>
DoF	<i>Degrees of Freedom</i>
FK	<i>Forward Kinematics</i>
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IK	<i>Inverse Kinematics</i>
JIRA	<i>Japanese Industrial Robot Association</i>
MTH	Matrizes de Transformação Homogêneas
QDU	Quatérnios Duais Unitários
SDLS	<i>Selectively Damped Least Squares</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>14</b>
1.1	JUSTIFICATIVA.....	15
1.2	OBJETIVO GERAL.....	21
1.3	OBJETIVOS ESPECÍFICOS .....	21
1.4	RESTRIÇÕES DO TRABALHO.....	21
<b>2</b>	<b>REFERENCIAL TEÓRICO .....</b>	<b>23</b>
2.1	MANIPULADORES ROBÓTICOS .....	23
2.2	PARAMETRIZAÇÃO DE MANIPULADORES ROBÓTICOS .....	25
2.3	CINEMÁTICA DIRETA .....	30
<b>2.3.1</b>	<b>Cinemática Direta Via Matrizes Homogêneas .....</b>	<b>31</b>
<b>2.3.2</b>	<b>Quatérnios.....</b>	<b>33</b>
<b>2.3.3</b>	<b>Quatérnios Duais .....</b>	<b>35</b>
<b>2.3.4</b>	<b>Cinemática Direta Via Quatérnios Duais Unitários .....</b>	<b>37</b>
2.4	CINEMÁTICA INVERSA.....	39
<b>2.4.1</b>	<b>Formulação do Problema de Cinemática Inversa .....</b>	<b>39</b>
<b>2.4.2</b>	<b>Matriz Jacobiana.....</b>	<b>43</b>
<b>2.4.3</b>	<b>Cálculo de Erro e Condições de Parada.....</b>	<b>46</b>
<b>2.4.4</b>	<b>Métodos Numéricos Jacobianos.....</b>	<b>47</b>
2.4.4.1	Jacobiana Transposta .....	48
2.4.4.2	Jacobiana Pseudoinversa.....	49
2.4.4.3	Mínimos Quadrados Amortecidos.....	50
2.4.4.4	Mínimos Quadrados Amortecidos Seletivamente .....	51
2.5	ANÁLISE DOS TRABALHOS RELEVANTES .....	55
<b>3</b>	<b>METODOLOGIA APLICADA .....</b>	<b>62</b>

3.1	BIBLIOTECA DE FUNÇÕES .....	62
3.2	GERAÇÃO DAS POSES INICIAIS E FINAIS .....	63
3.3	ALGORITMO DE CINEMÁTICA DIRETA .....	64
3.4	ALGORITMOS DE CINEMÁTICA INVERSA .....	65
3.5	MÉTRICAS DE DESEMPENHO .....	66
<b>4</b>	<b>RESULTADOS .....</b>	<b>68</b>
4.1	CONSTANTES E LIMITES APLICADOS .....	68
4.2	DESEMPENHO DOS ALGORITMOS .....	69
4.3	ANÁLISE DAS AMOSTRAS COM FALHA NA RESOLUÇÃO .....	78
<b>5</b>	<b>DISCUSSÕES E CONCLUSÕES .....</b>	<b>81</b>
	<b>REFERÊNCIAS .....</b>	<b>85</b>

## 1 INTRODUÇÃO

Em um apanhado histórico de eventos relevantes na história da robótica, Shyamanta e Hazakira (2018) relatam os anseios de matemáticos e filósofos em relação a autômatos. Destaque é dado ao polímata Leonardo da Vinci (1452-1519), que em 1495 projetou um robô humanoide no formato de um cavaleiro medieval capaz de mover seus braços, cabeça e mandíbula. Os autores também descrevem o interesse de Aristóteles (384 a.C – 322 a.C), que imaginava a abolição da escravatura com o surgimento de máquinas automatizadas.

Como destacado por Shyamanta e Hazakira (2018), o estudo da robótica é motivado pelo desejo de criar máquinas com potencial de sintetizar o comportamento humano, a fim de que elas sejam capazes de executar tarefas cotidianas com maior eficiência. Os autores explicam que há uma gama de conceitos fundamentais necessários para a implementação de um robô funcional, abrangendo diversas áreas da engenharia. Sobre este assunto, Craig (2012) enfatiza que, em geral, não é o caso de um único indivíduo ter o domínio de todo o campo da robótica.

Uma das áreas fundamentais da robótica é a cinemática, a qual compreende o estudo dos mecanismos robóticos sem considerar parcelas como forças, torques e imperfeições mecânicas que causam o movimento (CRAIG, 2012). Denavit e Hartenberg (1953) criaram um método para a parametrização cinemática de estruturas robóticas formadas pela conexão de juntas e elos. Os autores definem quatro valores a serem atribuídos para cada par de junta e elo:  $\alpha$ ,  $a$ ,  $\theta$  e  $d$ , nomeando-os de Parâmetros DH, em alusão às suas iniciais.

Ainda sobre a cinemática, a aplicação de ferramentas matemáticas modernas em problemas cinemáticos é realizada desde a década de 90, como os estudos publicados por Funda e Paul (1990), Aspragathos e Dimitros (1998) e Kenwright (2012). Nestes trabalhos, analisa-se a álgebra dos quatérnios duais na solução do problema de cinemática direta, sendo que o objetivo comum entre eles é otimizar as ferramentas computacionais aplicadas a nível matemático, buscando maneiras de reduzir ao mínimo possível a quantidade de operações algébricas necessárias.

O problema antagônico à cinemática direta, é a problemática da cinemática inversa, a qual é um tema em aberto no meio acadêmico, por ser consideravelmente

complexa e de soluções altamente não-lineares (KENWRIGHT, 2013). Aristidou et al. (2017) realizaram a análise do estado da arte das soluções existentes, agrupando-as em quatro categorias, com base no referencial matemático e computacional empregado: métodos analíticos, numéricos, *data-driven* e híbridos.

Entre os métodos numéricos citados por Aristidou et al. (2017), há os implementados com base na matriz Jacobiana. Esta matriz é amplamente aplicada em cinemática inversa, porém, ela possui restrições quanto a posições específicas de difícil computação, podendo levar a cálculos impossíveis: as chamadas singularidades (NAKAMURA e HANAFUSA, 1986).

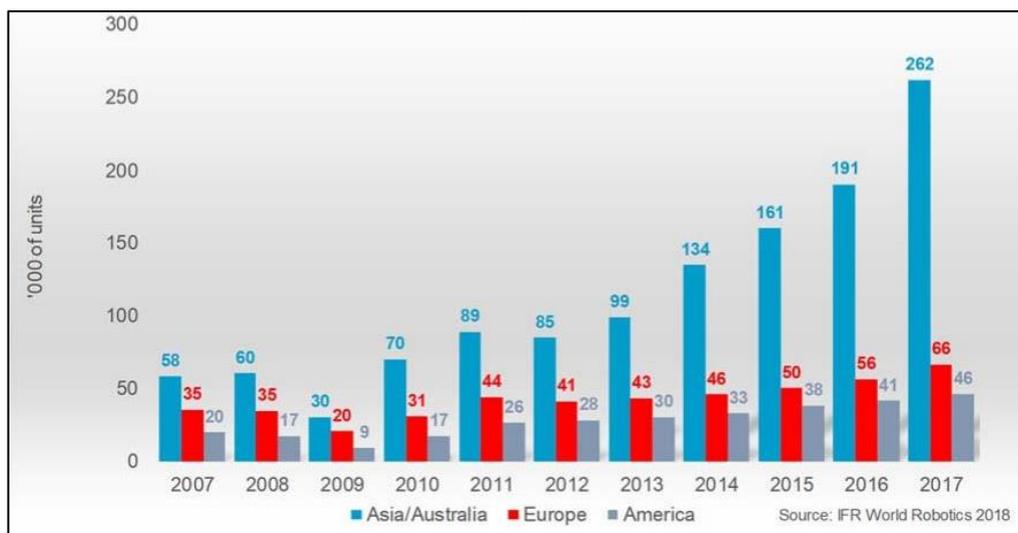
Os principais métodos numéricos Jacobianos para solução de cinemática inversa tratam de contornar o problema da singularidade. Tais técnicas são comparadas por autores como Buss e Kim (2004), Pham et al. (2018), e Omisore et al. (2018), através de experimentos baseados na coleta de métricas de desempenho como precisão posicional, esforço computacional, tempo de execução e taxa de sucesso do algoritmo.

Diante do contexto apresentado, propõe-se a obtenção de um comparativo entre os métodos Jacobianos, em cenário simulado, aplicando as técnicas de cinemática inversa em um manipulador robótico Kuka LWR, o qual possui sete juntas rotacionais em sua estrutura, ou sete graus de liberdade (7 DoF). Informações como a taxa de sucesso do algoritmo, a quantidade de iterações necessárias, o tempo decorrido a cada iteração, o tempo de processamento total e o comportamento do erro de pose ao longo do tempo foram coletadas. Estas informações foram as métricas de desempenho dos algoritmos avaliados.

## 1.1 JUSTIFICATIVA

A proposta deste trabalho justifica-se em dois cenários: industrial e acadêmico. Do ponto de vista industrial, o crescimento contínuo da indústria robótica é significativo, fato corroborado pelo gráfico da Figura 1, o qual mostra uma estimativa da quantidade encomendada de robôs industriais anualmente no mundo. Conforme visto na Figura 1, é notável o crescimento expressivo em todos os continentes, em especial na Ásia/Austrália.

Figura 1 – Encomendas anuais de robôs industriais no mundo, por continente.



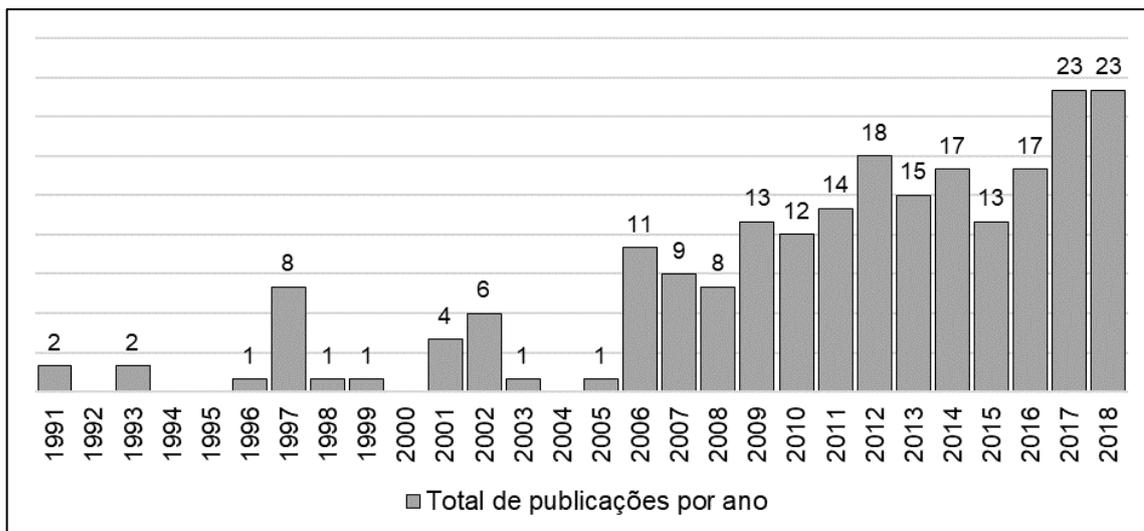
Fonte: IFR World Robotics (2018).

No Brasil, o Ministério da Indústria, Comércio Exterior e Serviços (MDIC) e a Agência Brasileira de Desenvolvimento Industrial (ABDI), lançaram em março de 2018 uma iniciativa de incentivo ao desenvolvimento da Indústria 4.0. Uma das medidas visa a aceleração da robotização em busca de maior colaboração dos robôs na força de trabalho, explica Vermulm (2018). Segundo o autor, a tecnologia de robotização de processos produtivos já é utilizada, e a tendência é de que robôs autônomos assumam maior relevância em cenários de automação digital.

Além da relevância da robótica evidenciada na área industrial, observa-se tendência do meio acadêmico na investigação de problemas envolvendo o controle de robôs, através de questões relacionadas às suas características cinemáticas e no desenvolvimento de ferramentas computacionais.

Com intuito de ratificar o interesse acadêmico na área da robótica, realizaram-se buscas no repositório de artigos acadêmicos do Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE), utilizando combinações de palavras-chave relevantes para este trabalho. Primeiramente, foi realizada uma busca com o termo “*inverse kinematics 7 DoF manipulators*”, onde a pesquisa retornou 221 resultados, sendo que a distribuição temporal destas publicações é demonstrada na Figura 2.

Figura 2 – Publicações anuais para “*inverse kinematics 7 DoF manipulators*”.

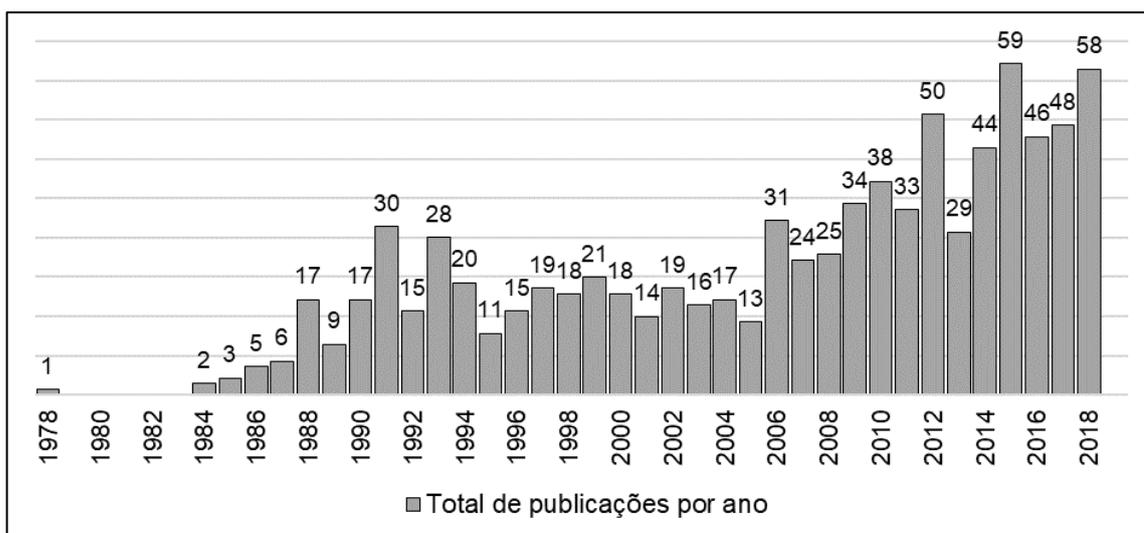


Fonte: O autor.

A tendência observada no gráfico da Figura 2, evidencia o aumento do interesse acadêmico no controle cinemático de manipuladores robóticos de 7 DoF nas publicações recentes, em especial após o ano de 2006.

Também foi realizada uma pesquisa com o termo “*Jacobian inverse kinematics*”, com o intuito de contextualizar no tempo os estudos realizados em cinemática inversa, nos quais foi empregada a matriz Jacobiana. A busca totalizou 858 publicações e sua distribuição ao longo dos anos é apresentada na Figura 3.

Figura 3 – Publicações anuais para “*Jacobian inverse kinematics*”.



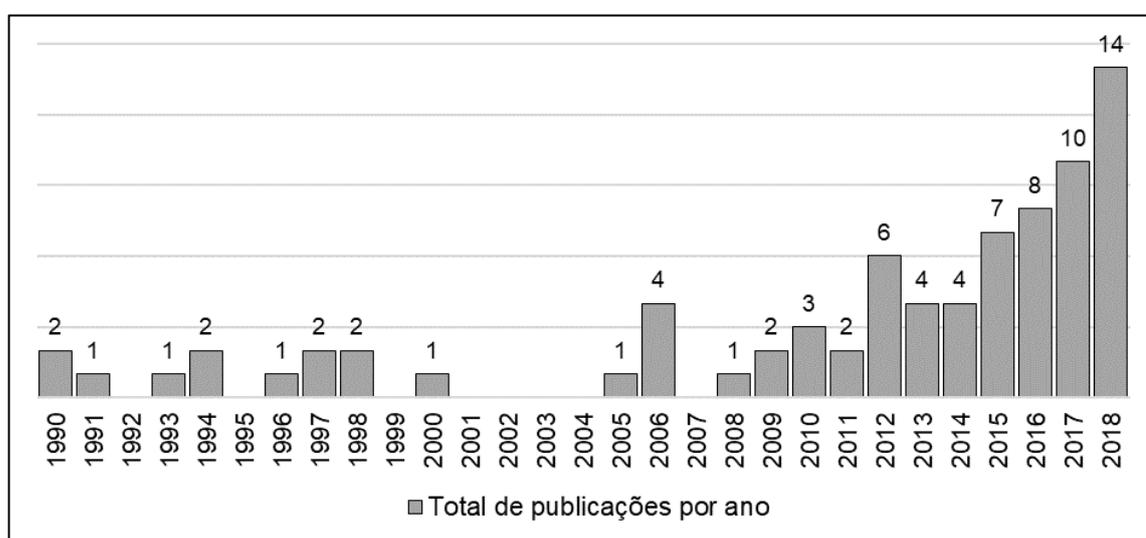
Fonte: O autor.

Os dados apresentados na Figura 3, mostram que a cinemática inversa empregando a matriz Jacobiana é um tema abordado desde o fim da década de 70,

porém, tornou-se foco de pesquisa apenas a partir de meados dos anos 80, com uma quantidade considerável de publicações realizadas. O interesse teve crescimento expressivo a partir de 2006, e mantém-se como um assunto relevante até os dias atuais.

Outra pesquisa relevante realizada foi com o termo “*dual quaternion kinematics*”, a qual objetiva investigar historicamente os trabalhos realizados empregando a álgebra dos quatérnios duais em análises cinemáticas. Foram listadas 80 publicações e sua distribuição temporal é mostrada na Figura 4.

Figura 4 – Publicações anuais para “*dual quaternion kinematics*”.

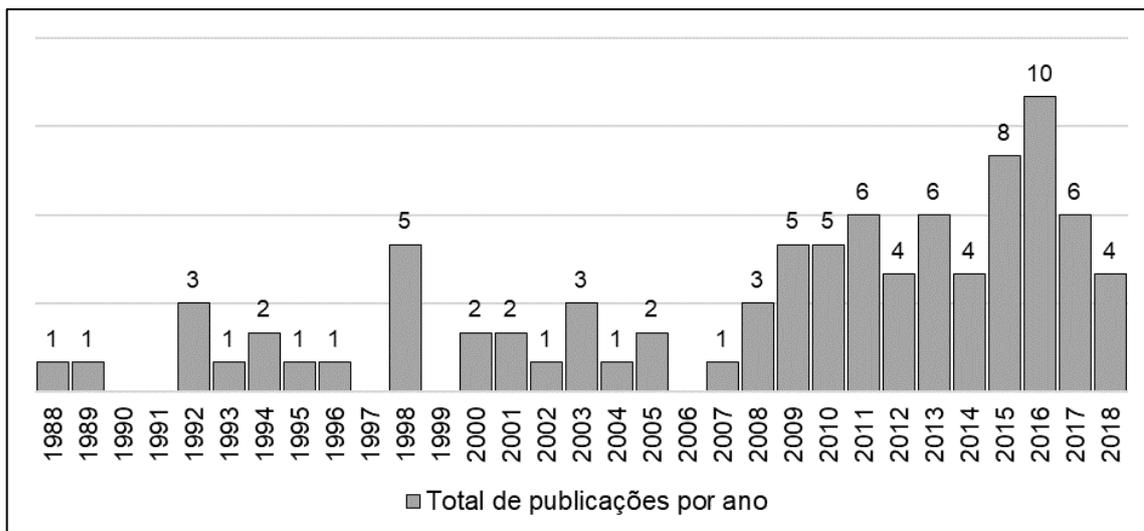


Fonte: O autor.

Observa-se na Figura 4, que os quatérnios duais foram empregados em estudos cinemáticos desde a década de 90, porém, sua importância despertou maior interesse acadêmico a partir do ano de 2012 e está atualmente em seu auge de pesquisas.

Como o foco do presente trabalho é, em síntese, a realização de um estudo comparativo entre métodos para solução de cinemática inversa, uma quarta pesquisa foi realizada no repositório do IEEE, desta vez utilizando o termo “*comparison inverse kinematics methods*”. O intuito foi verificar o interesse acadêmico em realizar comparações entre métodos de cinemática inversa. A pesquisa contabilizou 88 publicações realizadas ao longo das últimas quatro décadas, conforme pode ser visto na Figura 5.

Figura 5 – Publicações anuais para “*comparison inverse kinematics methods*”.



Fonte: O autor.

A distribuição apresentada na Figura 5, mostra que os métodos de cinemática inversa são comparados desde o final dos anos 80, sendo objeto de estudos de maneira crescente, principalmente a partir de 2009, e tendo seu pico de relevância no ano de 2016. Conclui-se, através das pesquisas realizadas na base do IEEE, que o tema deste trabalho é de forte tendência e deve ser considerado relevante em projetos acadêmicos.

Diversos artigos e documentos acadêmicos foram analisados em busca de referenciais teóricos e de metodologias para coleta de métricas de desempenho da cinemática inversa. No Quadro 1, destacam-se os principais documentos que formam a base teórica de pesquisa, sendo que o indicador de relevância (“Rel.”) denota o quão relevante o referido documento é para a elaboração deste trabalho, em uma escala de 0 a 10, sendo 10 para relevância máxima.

Quadro 1 – Documentos acadêmicos de maior relevância para o presente projeto.

Título	Autor(es)	Ano	Tema	Abordagem	Conclusão	Rel.
<i>Selectively damped least squares for inverse kinematics</i>	Buss, Samuel R.; Kim, Jin Su	2004	Apresentação o método SDLS para cinemática inversa.	Demonstração dos conceitos matemáticos; Comparativo de desempenho entre os métodos Jacobianos existentes e o proposto.	O SDLS é o método Jacobiano mais eficiente quando seu maior tempo de execução é aceitável. O DLS é recomendado havendo conhecimento prévio da melhor constante de amortecimento.	10
<i>Set-point control of robot end-effector pose using dual quaternion feedback</i>	Pham et al.	2018	Desenvolvimento de um sistema de controle de pose para de um manipulador com 6 DoF utilizando representação por quatérnios duais.	Elaboração de sistema de controle de <i>setpoint</i> ; Coleta das métricas de desempenho de métodos Jacobianos através de problemas gerados aleatoriamente.	O formalismo dos quatérnios duais melhora a performance em relação aos métodos alternativos. O sistema de controle proposto é de simples implementação, porém ao custo de menor performance.	9
<i>Deeply-learnt damped least-squares (DL-DLS) method for inverse kinematics of snake-like robots</i>	Omisore et al.	2018	Treinamento de rede neural profunda para previsão da melhor constante de amortecimento do método DLS.	Treinamento de rede neural para otimizar o método DLS; Comparativo das métricas de desempenho do algoritmo proposto com métodos Jacobianos e outras soluções alternativas.	Métodos baseados em mínimos quadrados amortecidos têm métricas de desempenho superiores a todas as alternativas implementadas. A rede neural apresentou as melhores métricas, porém com alto custo computacional.	9
<i>A beginners guide to dual-quaternions: what they are, how they work, and how to use them for 3D character hierarchies</i>	Kenwright, Ben	2012	Análise da álgebra dos quatérnios duais.	Demonstração da eficiência dos quatérnios duais na descrição de estruturas 3D.	Quatérnios duais são a ferramenta moderna de maior interesse para descrição cinemática de estruturas hierárquicas.	8
<i>Position-based kinematics for 7-DoF serial manipulators with global configuration control, joint limit and singularity avoidance</i>	Faria et al.	2018	Metodologia para desvio de singularidades em manipulador de 7 DoF, considerando limitações de juntas.	Dedução analítica da cinemática inversa com detecção de configurações singulares; Medição de desempenho em poses factíveis geradas aleatoriamente.	O método proposto é relevante em cenários onde a trajetória do efetuador deve ser contínua e previsível.	7
<i>Inverse Kinematics Techniques in Computer Graphics: A Survey</i>	Aristidou et al.	2017	Revisão do estado da arte das soluções de cinemática inversa.	Revisão bibliográfica dos métodos de cinemática inversa descritos em literatura.	O artigo trouxe um apanhado de metodologias mostrando o desenvolvimento histórico e tendências futuras.	7

Fonte: O autor.

## 1.2 OBJETIVO GERAL

O objetivo geral do presente trabalho é a realização de um comparativo entre quatro métodos Jacobianos para a solução de cinemática inversa de um manipulador de 7 DoF baseado em quatérnios duais.

## 1.3 OBJETIVOS ESPECÍFICOS

Ressaltam-se os objetivos específicos, os quais são necessários para alcançar o objetivo geral:

- I. Criar uma biblioteca de funções no *software* Matlab para as operações envolvendo quatérnios e quatérnios duais;
- II. Desenvolver um algoritmo para resolver a cinemática direta do atuador, baseado em quatérnios duais;
- III. Implementar os algoritmos de cinemática inversa a serem avaliados;
- IV. Coletar métricas de desempenho na solução dos problemas gerados;
- V. Realizar o comparativo de desempenho dos métodos implementados.

## 1.4 RESTRIÇÕES DO TRABALHO

A cinemática inversa tem como objetivo encontrar a combinação de ângulos de junta de uma estrutura robótica que leva o efetuador, ou ferramenta, do ponto A ao ponto B, posicionando-o na orientação escolhida. O formato que esta trajetória pode assumir é um bloco funcional da robótica que deve ser estudado posteriormente. A geração de trajetória em si é complexa o suficiente para não fazer parte do escopo do trabalho, limitando-se o estudo somente à avaliação de métodos para a solução de cinemática inversa.

Os testes foram realizados no âmbito da simulação, utilizando-se do *software* Matlab. Desta forma, o erro concatenado que vai desde imprecisões de cálculo até incertezas mecânicas em um modelo real foram abstraídos, uma vez que isso poderia inviabilizar a análise de maneira isolada do desempenho de cada método.

Por fim, não foram implementadas condicionais que impedem a colisão entre partes da estrutura do manipulador ou limitações no ângulo de rotação das juntas. Isso significa que podem ocorrer cenários onde o manipulador irá curvar-se e

atravessar a si mesmo ou assumir posições impraticáveis, uma vez que se trata apenas de um objeto matemático sem limitações físicas. Tal cenário permitiu a análise da cinemática inversa de maneira isolada, sem interferência de limitações forçadas.

## 2 REFERENCIAL TEÓRICO

Com a finalidade de traçar o raciocínio levando à compreensão do escopo deste trabalho, este capítulo transcorre uma revisão de conceitos fundamentais de robótica, enfatizando os elementos cinemáticos de manipuladores. Primeiramente, é apresentado o que é um manipulador robótico e define-se nomenclaturas para sua constituição elementar. Após, descreve-se a metodologia de Denavit-Hartenberg para parametrização de estruturas robóticas. No decorrer do tema, elabora-se o problema da cinemática direta e sua solução através do método clássico, discorrendo sobre suas limitações.

Em seguida, revisa-se a álgebra dos quatérnios duais, a ferramenta moderna que melhor descreve a estrutura para solucionar a cinemática direta, de acordo com os trabalhos de Kenwright (2012; 2013), Vidakovic et al. (2014) e Sharkawy e Aspragathos (2017). Posteriormente, demonstra-se a aplicação dos quatérnios duais como solução da cinemática direta, sendo a base fundamental dos métodos de cinemática inversa implementados.

No decorrer do capítulo, é formulado o problema da cinemática inversa, sendo este o foco de atenção no presente trabalho. Neste contexto, são descritas as características da cinemática inversa e conceitos matemáticos envolvidos em sua solução, como a matriz Jacobiana e o *loop* de controle de erro de pose, descrito no domínio dos quatérnios duais. O capítulo encerra com a apresentação de quatro métodos de solução para a cinemática inversa.

### 2.1 MANIPULADORES ROBÓTICOS

A distinção literal entre o que é e o que não é considerado um robô é algo indefinido, sendo normalmente uma classificação atrelada à complexidade e à capacidade de reprogramação do dispositivo: quando uma máquina pode ser reprogramada para se adaptar à execução de diversas tarefas, então ela é comumente considerada um robô. Em contrapartida, quando uma máquina foi projetada de forma limitada a executar um determinado conjunto de tarefas, então esta é considerada uma máquina de automação fixa (CRAIG, 2012).

Dentro do que se considera um robô, existem diversas estruturas que podem ser separadas em categorias. Shyamanta e Hazakira (2018) descrevem cada uma das

seis categorias em que se subdividem os diferentes tipos de robôs, de acordo com a nomenclatura da *Japanese Industrial Robot Association* (JIRA), como segue:

- a) manipuladores;
- b) numericamente controlados;
- c) sensatos;
- d) adaptativos;
- e) inteligentes;
- f) sistemas mecatrônicos inteligentes.

A classe dos manipuladores recebe destaque no trabalho de Craig (2012), onde o autor afirma que esta é a estrutura robótica mais importante na indústria. Chen, Li e Kwok (2011), afirmam que o uso de manipuladores robóticos impulsionou a produtividade do setor de manufatura devido à versatilidade em sua reprogramação.

Shyamanta e Hazakira (2018) reiteram que esta é a configuração de robô significativamente mais comum na indústria, definindo-os como sendo uma estrutura fisicamente ancorada a uma estação de trabalho constituída por uma cadeia de juntas controláveis que permite posicionar o seu efetuador, ou ferramenta, em qualquer posição e orientação dentro do espaço de trabalho.

Dentre os manipuladores robóticos descritos em literatura, um dos que mais recebe destaque é o Kuka LWR, um manipulador antropomórfico produzido pela *Kuka Industrial Robots*<sup>1</sup>, o qual tem sido objeto de estudos cinemáticos e dinâmicos recentes como os realizados por Sharkawy e Aspragathos (2017), Khatib, Khudir e Luca (2017) e Rigano et al. (2018).

Na Figura 6, apresenta-se o Kuka LWR e sua estrutura de eixos. Observa-se que este manipulador é constituído de: punho (1), módulo das juntas (2) e *frame* da base (3). O punho é a parte onde as ferramentas da tarefa a ser executada pela máquina podem ser acopladas. O módulo das juntas é a parte móvel do robô que guia o punho. O *frame* da base é a parte que permanece ancorada à estação de trabalho. Os pontos de A1 a A7 são as sete juntas rotacionais que proporcionam liberdade de movimento à estrutura.

---

<sup>1</sup> Site oficial da *Kuka Industrial Robots*: <https://www.kuka.com/pt-br>

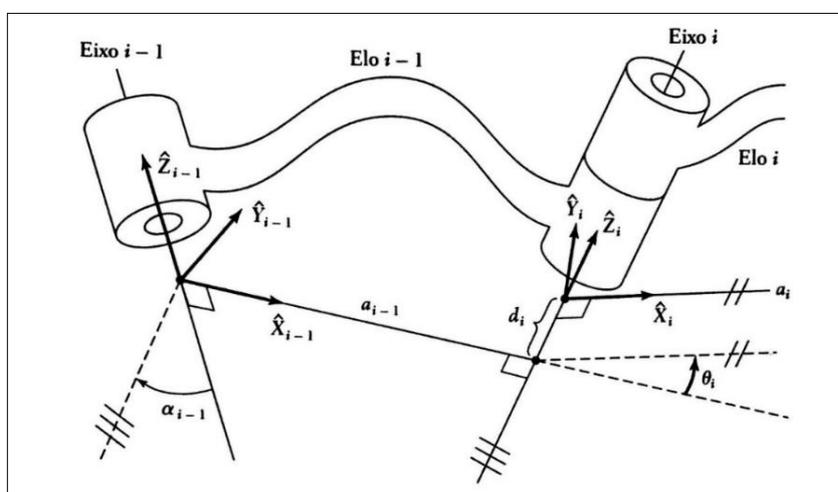


eixos  $\hat{X}$ ,  $\hat{Y}$  e  $\hat{Z}$  de seu *frame* de referência, o que implica na necessidade de definir 6 variáveis para descrevê-lo por completo. Diz-se, então, que este ponto possui 6 graus de liberdade, ou *Degrees of Freedom* (DoF) (LYNCH e PARK, 2017).

Segundo Craig (2012), caso um manipulador seja formado por menos de 6 DoF, ele não será capaz de alcançar todas as posições dentro de seu espaço de trabalho. Robôs com menos de 6 DoF são usados para executar tarefas específicas, porém é necessário estudar previamente suas limitações. Faria et al. (2018) explicam que o estudo de manipuladores com 7 DoF, como o Kuka LWR, tem como objetivo aumentar a versatilidade do robô, para que os mesmos possam lidar de forma mais eficiente com configurações próximas a singularidades. Isso acontece porque o grau de liberdade extra fornece menos restrições ao problema de cinemática inversa, aumentando a quantidade de configurações capazes de alcançar uma mesma pose. Tais definições serão abordadas com mais detalhes na seção 2.4.

Denavit e Hartenberg (1953) desenvolveram um método para descrição de estruturas robóticas, baseado em matrizes. Em seu trabalho, é apresentada uma notação que envolve a atribuição de quatro parâmetros para cada par junta-elo, a qual pode ser aplicada em qualquer estrutura de corpos rígidos. Segundo os próprios autores, a notação através dos então denominados parâmetros DH, torna possível descrever completamente manipuladores robóticos, sendo uma ferramenta confiável para procedimentos analíticos. Craig (2012) apresenta, através da Figura 7, um eixo torcido com sua forma exagerada para exemplificar os quatro parâmetros DH de forma visual, também fornecendo uma explicação sobre cada variável apresentada.

Figura 7 – Eixo torcido para exemplificar os parâmetros DH.



Fonte: Craig (2012, p. 65).

Na Figura 7, observa-se que são estabelecidos sistemas cartesianos de referência  $(\hat{X}\hat{Y}\hat{Z})_i$  e  $(\hat{X}\hat{Y}\hat{Z})_{i-1}$  de maneira a estabelecer orientações relativas entre os eixos do elo  $i$ . Logo, definem-se os parâmetros  $\alpha_i$ ,  $a_i$ ,  $\theta_i$  e  $d_i$ , como sendo:

- a)  $\alpha_i$  representa o ângulo formado entre  $\hat{Z}_i$  e  $\hat{Z}_{i+1}$  em torno do eixo  $\hat{X}_i$ ;
- b)  $a_i$  representa a distância entre  $\hat{Z}_i$  e  $\hat{Z}_{i+1}$  ao longo do eixo  $\hat{X}_i$ ;
- c)  $\theta_i$  representa o ângulo formado entre  $\hat{X}_i$  e  $\hat{X}_{i-1}$  em torno do eixo  $\hat{Z}_i$ ;
- d)  $d_i$  representa a distância formado entre  $\hat{X}_i$  e  $\hat{X}_{i-1}$  ao longo do eixo  $\hat{Z}_i$ .

Os parâmetros DH são definidos seguindo uma ordem de transformação convencional. Tal ordem visa transferir a origem de um sistema cartesiano para o seguinte, rotacionando e trasladando seu *frame* de referência, de modo que o mesmo coincida com o *frame* da junta seguinte. Conforme destacado por Radavelli et al. (2014), a ordem de aplicação dos parâmetros DH deve ser: primeiramente uma rotação em  $\alpha$ , seguida de uma translação em  $a$ , uma rotação em  $\theta$  e uma translação ao longo de  $d$ .

Como estes parâmetros devem ser definidos para cada par junta-elo, constrói-se uma tabela com as informações para cada grau de liberdade da estrutura robótica. A notação funciona também para juntas mais complexas como, por exemplo, uma junta esférica, com 3 DoF. Para a descrição cinemática de uma estrutura assim, pode-se considerar que uma junta com  $n$  graus de liberdade é o equivalente a  $n$  juntas de 1 grau de liberdade conectadas por  $n - 1$  elos de comprimento zero (KENWRIGHT, 2013).

A Tabela 1, apresenta os Parâmetros DH de um manipulador antropomórfico de 7 DoF, como o caso do Kuka LWR, como descrito por Faria et al. (2018).

Tabela 1 – Parâmetros DH de um manipulador antropomórfico de 7 DoF.

$i$	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	0	$-\pi/2$	$d_b$	$\theta_1$
2	0	$\pi/2$	0	$\theta_2$
3	0	$\pi/2$	$d_{bc}$	$\theta_3$
4	0	$-\pi/2$	0	$\theta_4$
5	0	$-\pi/2$	$d_{cp}$	$\theta_5$
6	0	$\pi/2$	0	$\theta_6$
7	0	0	$d_{pe}$	$\theta_7$

Fonte: Adaptado de Faria et al. (2018).

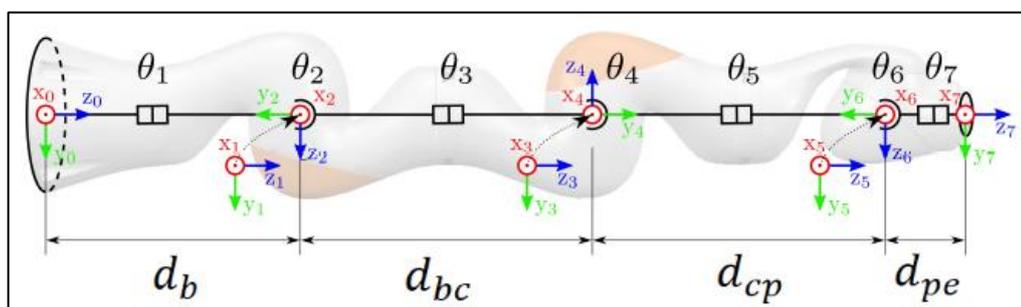
Na Tabela 1, os parâmetros  $d_i$  significam:

- $d_b$  é o comprimento do elo da base;
- $d_{bc}$  é o comprimento do elo ligando a base ao cotovelo;
- $d_{cp}$  é o comprimento do elo ligando o cotovelo ao punho;
- $d_{pe}$  é o comprimento do elo ligando o punho ao efetuador.

Em referência à Tabela 1, todos os parâmetros são constantes, exceto os valores de  $\theta_i$ . Isso significa que para esse manipulador a variável de junta é  $\theta_i$  em todas as juntas. Neste documento, o termo “variável de junta” é tratado como sinônimo do parâmetro  $\theta$ .

Faria et al. (2018) explicam, através da Figura 8, a relação entre os parâmetros  $d_i$  constantes na Tabela 1 e os comprimentos físicos do manipulador. Também é evidenciada a posição e orientação dos eixos de referência ao longo do robô, de modo a atrelar as rotações e translações dos parâmetros DH com suas contrapartes reais.

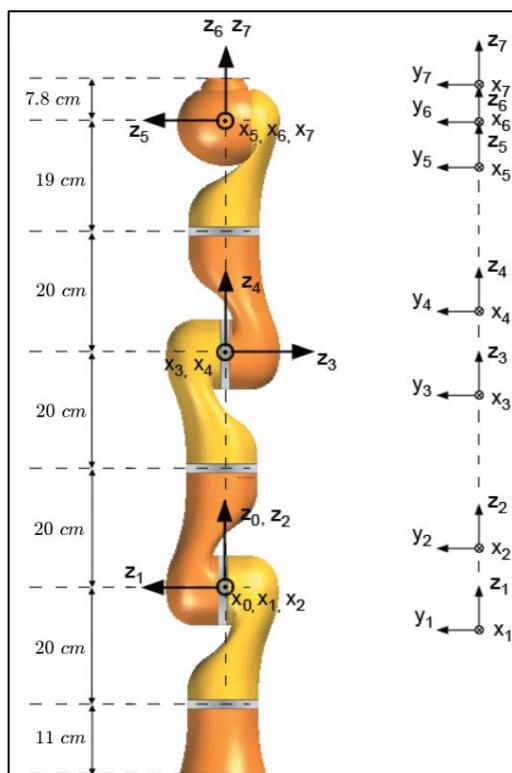
Figura 8 – Representação dos parâmetros  $d_i$  em um robô Kuka LWR.



Fonte: Adaptado de Faria et al. (2018).

Complementando os dados da Figura 8, os comprimentos físicos do manipulador Kuka LWR são destacados na Figura 9. Tais valores são constantes, uma vez que a variável de junta é  $\theta$  em toda a estrutura. Também são descritos os eixos de referência de cada junta em compatibilidade com os parâmetros DH.

Figura 9 – Comprimentos reais de um robô Kuka LWR.



Fonte: Adaptado de Coppelia Robotics (2019).

Combinando as informações fornecidas pela Tabela 1, pela Figura 8 e pela Figura 9, obtêm-se os valores físicos dos parâmetros  $d_i$ , apresentados na Tabela 2, a qual contempla todos os parâmetros DH do Kuka LWR:

Tabela 2 – Parâmetros DH do manipulador Kuka LWR.

$i$	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	0	$-\pi/2$	31cm	$\theta_1$
2	0	$\pi/2$	0cm	$\theta_2$
3	0	$\pi/2$	40cm	$\theta_3$
4	0	$-\pi/2$	0cm	$\theta_4$
5	0	$-\pi/2$	39cm	$\theta_5$
6	0	$\pi/2$	0cm	$\theta_6$
7	0	0	7,8cm	$\theta_7$

Fonte: O Autor.

Os parâmetros DH constantes na Tabela 2 foram utilizados neste trabalho para construir o mecanismo de forma algébrica no simulador para, então, poder manipulá-lo com a cinemática direta e inversa. Tais procedimentos de cinemática são abordados nas seções 2.3 e 2.4.

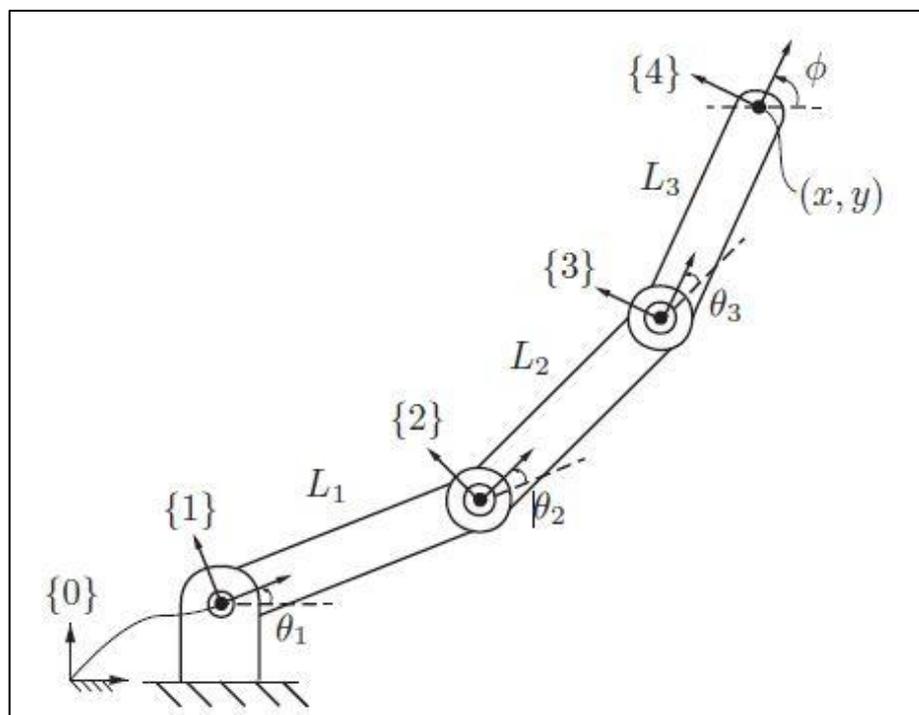
## 2.3 CINEMÁTICA DIRETA

A cinemática de um robô é o estudo dos mecanismos robóticos sem considerar parcelas como forças, torques e imperfeições mecânicas que causam o movimento (CRAIG, 2012). O cálculo da posição e orientação cartesiana de um efetuador, tendo conhecimento de todas suas variáveis de junta, chama-se cinemática direta, do inglês *Forward Kinematics* (FK) (LYNCH e PARK, 2017). Para uma estrutura formada por juntas rotacionais, onde a variável de junta é  $\theta$ , a FK é o processo que relaciona a pose do efetuador ( $s$ ) a uma função  $f(\theta)$ , conforme a equação (1) (ARISTIDOU et al. 2017).

$$s = f(\theta) \quad (1)$$

Para exemplificar a problemática, Lynch e Park (2017) fornecem um exemplo simplificado para solução da FK, onde o objetivo é conhecer a posição cartesiana  $(x, y)$  e o ângulo de inclinação  $\phi$  do efetuador de um robô planar com 3 DoF, o mesmo representado na Figura 10. A solução só é possível tendo o conhecimento dos comprimentos de elo  $L_i$  e dos ângulos  $\theta_i$  de cada junta.

Figura 10 – Cinemática direta de robô planar com 3 DoF.



Fonte: Lynch e Park (2017, p. 138).

No caso do robô planar com 3 DoF da Figura 10, aplicam-se as equações (2), (3) e (4) para encontrar as variáveis de pose do efetuador.

$$x = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \quad (2)$$

$$y = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \quad (3)$$

$$\phi = \theta_1 + \theta_2 + \theta_3 \quad (4)$$

Esta é a solução para um problema simplificado que exige apenas fundamentos básicos de trigonometria. No caso de problemas mais complexos, como robôs que atuam no espaço tridimensional e com maior número de DoF, a solução torna-se consideravelmente mais difícil e, portanto, é necessário o uso de outros métodos (LYNCH e PARK, 2017).

### 2.3.1 Cinemática Direta Via Matrizes Homogêneas

Craig (2012) descreve em seu trabalho como aplicar os parâmetros DH em uma estrutura de matrizes de transformação homogêneas (MTH) para obter a transformação de um sistema cartesiano em outro de forma hierárquica, fornecendo informações de posição e orientação do efetuador.

Primeiramente, Craig (2012) utiliza a notação da equação (5) para descrever um ponto  $p$  no espaço tridimensional, com as coordenadas  $x$ ,  $y$  e  $z$  relativas ao seu sistema de referência. As equações (6), (7) e (8) realizam a mudança de orientação do ponto  $p$ , rotacionando seu sistema de referência em torno dos eixos  $\hat{X}$ ,  $\hat{Y}$  e  $\hat{Z}$ , respectivamente, em um ângulo  $\beta$ . Neste caso, adota-se a notação  $\beta$  para ângulos genéricos com a finalidade de não confundir com os ângulos  $\theta$  ou  $\alpha$  relativos aos parâmetros DH.

$$p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (5)$$

$$R_x(\beta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta) & -\sin(\beta) \\ 0 & \sin(\beta) & \cos(\beta) \end{bmatrix} \quad (6)$$

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \text{sen}(\beta) \\ 0 & 1 & 0 \\ -\text{sen}(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (7)$$

$$R_z(\beta) = \begin{bmatrix} \cos(\beta) & -\text{sen}(\beta) & 0 \\ \text{sen}(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

A partir desta notação, Vidakovic et al. (2014) demonstram a construção das chamadas matrizes homogêneas por meio da equação (9), sendo  $T_A$  a MTH que mapeia a pose de um corpo rígido em relação a um sistema cartesiano de referência.

$$T_A = \begin{bmatrix} R(\beta) & \cdots & p \\ \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (9)$$

Assim, Craig (2012) define a matriz resultante da transformação do sistema  $T_{i-1}$  no sistema  $T_i$ , unindo aos parâmetros DH e sendo  $T_i^{i-1}$  a MTH do *frame*  $i - 1$  no *frame*  $i$ , conforme equação (10).

$$T_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\text{sen}(\theta_i) & 0 & a_{i-1} \\ \text{sen}(\theta_i)\cos(\alpha_{i-1}) & \cos(\theta_i)\cos(\alpha_{i-1}) & -\text{sen}(\alpha_{i-1}) & -\text{sen}(\alpha_{i-1})d_i \\ \text{sen}(\theta_i)\text{sen}(\alpha_{i-1}) & \cos(\theta_i)\text{sen}(\alpha_{i-1}) & \cos(\alpha_{i-1}) & \cos(\alpha_{i-1})d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

Craig (2012) também demonstra como é possível concatenar as transformações dos frames partindo da base e passando por todas as juntas até o final da hierarquia, conforme equação (11), onde  $T_n^0$  é a MTH do efetuador em relação à base do manipulador de  $n$  DoF.

$$T_n^0 = T_1^0 T_2^1 T_3^2 \dots T_n^{n-1} \quad (11)$$

Kenwright (2012) explica que, apesar das MTH serem uma ferramenta amplamente utilizada na solução da FK, isso se deve apenas ao fato de que matrizes são ensinadas em álgebra linear no início dos cursos de graduação e são uma escolha instintiva por sua familiaridade. O autor descreve as diversas restrições das referidas matrizes:

- a) exigem um mínimo de 12 parâmetros para representar estruturas de apenas 6 DoF, o que demanda maior uso de recursos computacionais;

- b) a parcela rotacional da MTH,  $R(\beta)$ , é composta por colunas ortogonais. Sucessivos cálculos farão essa ortogonalidade perder-se gradualmente a cada iteração e será necessário empregar condicionais no algoritmo para normalizar a matriz novamente;
- c) geração de trajetória é prejudicada, pois a interpolação entre matrizes é difícil;
- d) visualizar o ângulo e o eixo em que uma matriz está rotacionando não é feito por simples observação.

Apesar das MTH serem a solução clássica da FK, suas limitações trazem a necessidade de investigar ferramentas modernas para otimizar a solução do problema. Apresenta-se, então, a álgebra dos quatérnios e quatérnios duais em busca de compensação de tais restrições.

### 2.3.2 Quatérnios

A história dos quatérnios remonta aos meados do século XIX como uma curiosidade algébrica e geométrica (FAMILTON, 2015). Os quatérnios foram introduzidos por William Rowan Hamilton em 1844, o qual desenvolveu seu trabalho publicando o livro “*Elements of Quaternions*” em 1866.

Familton (2015) caracteriza o problema que Hamilton tentou resolver durante 10 anos, ou seja, obter uma forma de descrever pontos em 3 dimensões da mesma forma que é possível descrever pontos em 2 dimensões através de números complexos na forma retangular  $(a + bi)$ . Hamilton (1844) tentou introduzir o termo  $cj$ , onde  $j$  também é uma parcela imaginária como  $i$ , porém, não foi possível resolver a multiplicação de dois números no formato  $(a + bi + cj)(x + yi + zj)$ . Sua formulação resultava em um termo em excesso:  $(bz - cy)^2$ . Sobre este termo, do qual culminou uma solução, Hamilton declara:

...Mas este é apenas o quadrado do termo  $k$  no desenvolvimento do produto  $(a + bi + cj)(x + yi + zj)$ , se garantirmos que  $ij = k$  e  $ji = -k$ , como antes. E aqui caiu em mim a noção de que devemos admitir, em algum sentido, uma *quarta dimensão* de espaço para cálculo com tríades; ou transferindo o paradoxo para a álgebra, devemos admitir um terceiro imaginário distinto de símbolo  $k$ , a não ser confundido nem com  $i$  e nem com  $j$ , mas igual ao produto do primeiro como multiplicador e o segundo como multiplicando; portanto, fui conduzido a introduzir os *quatérnios*, de tal forma que  $a + ib + jc + kd$ , ou  $(a, b, c, d)$ . (HAMILTON, 1844, p. 492-493)

Desta conclusão, surge a notação padrão para os termos de um quatérnio, conforme equação (12), onde  $(w, x, y, z) \in \mathbb{R}$  e os termos  $i, j$  e  $k$  são mutuamente ortogonais. Como  $w$  é um escalar e a condição de ortogonalidade atrela os termos  $x, y$  e  $z$ , então uma notação alternativa para  $q$  é dada na equação (13), onde  $w \in \mathbb{R}$  é o elemento do quatérnio desassociado da parcela complexa e  $v \in \mathbb{R}^3$  é o versor  $xyz$ . Segundo Kenwright (2012), esta notação é a mais comum para um quatérnio, logo, será a notação adotada neste documento.

$$q = w + xi + yj + zk \quad (12)$$

$$q = [w, v] \quad (13)$$

Hamilton (1866, p. 158-160) formaliza sua descoberta descrevendo as relações entre os termos imaginários que diferem um quatérnio de um simples vetor de 4 escalares, de acordo com a equação (14).

$$i^2 = j^2 = k^2 = ijk = -1 \quad (14)$$

Kenwright (2012), descreve as operações aritméticas fundamentais dos quatérnios utilizadas em seu trabalho, sendo que a notação  $(a \circ b)$  representa o produto escalar entre  $a$  e  $b$  e a notação  $(a \times b)$  representa o produto vetorial entre  $a$  e  $b$ :

a) adição de 2 quatérnios, conforme equação (15);

$$q_1 + q_2 = [w_1 + w_2, v_1 + v_2] \quad (15)$$

b) complexo conjugado, conforme equação (16);

$$q^* = [w, -v] \quad (16)$$

c) norma ou magnitude, conforme equação (17);

$$|q| = qq^* \quad (17)$$

d) multiplicação de dois quatérnios, conforme equação (18).

$$q_1 q_2 = [w_1 w_2 - v_1 \circ v_2, w_1 v_2 + w_2 v_1 + (v_1 \times v_2)] \quad (18)$$

Baseado no trabalho de Hamilton, McDonald (2010) faz um apanhado de relações geométricas e algébricas dos quatérnios e sua aplicação na orientação de objetos tridimensionais. Destas relações, pode-se rotacionar um vetor  $p \in \mathbb{R}^3$  em torno de um eixo  $u \in \mathbb{R}^3$  em um ângulo  $\theta$ , conforme equação (19), complementada pela equação (20), onde  $p'$  é o vetor em  $\mathbb{R}^3$  resultante da rotação de  $p$  em  $\theta$  ao longo do eixo  $u$ .

$$p' = qpq^{-1} \quad (19)$$

$$q = \cos\left(\frac{\theta}{2}\right) + \text{sen}\left(\frac{\theta}{2}\right)u \quad (20)$$

Para computar  $q^{-1}$ , o quatérnio  $q$  deve ter magnitude unitária. Neste caso, o inverso de  $q$  é igual à sua matriz transposta  $q^T$ , que é igual ao seu complexo conjugado  $q^*$ , relação destacada na equação (20) (MCDONALD, 2010).

$$q^{-1} = q^T = q^* = w - xi - yj - zk \quad (21)$$

### 2.3.3 Quatérnios Duais

Os números duais foram introduzidos por Clifford (1873). Em seu trabalho, Clifford faz menção aos “vetores de Hamilton”, referindo-se à teoria de quatérnios. Os quatérnios são quantidades que possuem magnitude e direção, porém não tem indicação de posição. Exemplificando, uma linha ligando os pontos  $A$  e  $B$  é descrita pelo mesmo quatérnio que descreve uma linha ligando os pontos  $C$  e  $D$ , desde que  $\overline{AB}$  seja do mesmo tamanho e paralela a  $\overline{CD}$  (CLIFFORD, 1873).

A noção de que  $\overline{AB}$  não coincide obrigatoriamente com  $\overline{CD}$  em sua posição espacial, apenas coincidindo em magnitude e direção, motiva a introdução dos números duais em busca de uma descrição completa de posição, orientação e magnitude espacial (CLIFFORD, 1873). Clifford estabelece assim um novo campo da álgebra em seu livro “*Mathematical Papers*” em 1882. Um número dual genérico é definido conforme equação (22), sendo  $r$  a parcela real,  $d$  a parcela complexa e  $\varepsilon$  o chamado operador dual, o qual respeita mutuamente as condições descritas nas equações (23) e (24).

$$z = r + d\varepsilon \quad (22)$$

$$\varepsilon^2 = 0 \quad (23)$$

$$\varepsilon \neq 0 \quad (24)$$

Kenwright (2012) explica que quando os termos  $r$  e  $d$  de um número dual são quatérnios, tem-se uma notação unificada de posição espacial, orientação e magnitude. Cada quatérnio dual é formado por 2 quatérnios (ou seja: 8 elementos) e podem, então, representar rotações e translações. O autor afirma que a representação de um quatérnio dual pode ser feita como na equação (25) ou conforme equação (26). Para fins de notação neste documento, considera-se  $q$  como um quatérnio de Hamilton e  $Q$  um quatérnio dual de Clifford.

$$Q = q_r + q_d \varepsilon \quad (25)$$

$$Q = [q_r \quad q_d] \quad (26)$$

Kenwright (2012) faz um apanhado das relações de interesse da álgebra envolvendo quatérnios duais com a cinemática robótica, obtendo assim:

a) adição de dois quatérnios duais, conforme equação (27);

$$Q_1 + Q_2 = q_{r_1} + q_{r_2} + (q_{d_1} + q_{d_2})\varepsilon \quad (27)$$

b) multiplicação de quatérnios duais, conforme equação (28);

$$Q_1 \otimes Q_2 = q_{r_1} q_{r_2} + (q_{r_1} q_{d_2} + q_{d_1} q_{r_2})\varepsilon \quad (28)$$

c) conjugado de um quatérnio dual, conforme equação (29);

$$Q^* = q_r^* + q_d^* \varepsilon \quad (29)$$

d) magnitude de um quatérnio dual, conforme equação (30).

$$|Q| = QQ^* \quad (30)$$

Kenwright (2012) também explica que um quatérnio dual possui duas condições a serem cumpridas para que sua magnitude tenha valor unitário, apresentadas nas equações (31) e (32).

$$|Q| = 1 \quad (31)$$

$$q_r^* q_d + q_d^* q_r = 0 \quad (32)$$

A garantia de que o quatérnio dual tem magnitude unitária é um fator chave para que se possa representar qualquer transformação rotacional ou translacional (KENWRIGHT, 2012). Quatérnios Duais Unitários serão referenciados por sua sigla, QDU.

Observa-se na equação (28) que, para a multiplicação de quatérnios duais, é necessário utilizar as relações de multiplicação de um quatérnio de Hamilton conforme Equação (18). O símbolo “ $\otimes$ ” é usado para representar a multiplicação de quatérnios duais, como utilizado por Vidakovic et al. (2014).

Para representar a informação de rotação e translação de um corpo rígido em uma única variável de estado de forma unívoca, usa-se o QDU apresentado na equação (34), onde  $\theta$  é o ângulo de rotação e  $t$  é a translação em relação ao sistema de coordenadas  $\hat{u}_x, \hat{u}_y, \hat{u}_z$  (KENWRIGHT, 2012).

$$Q = [q_r \quad q_d] \quad (33)$$

$$Q = \left\{ \left[ \cos\left(\frac{\theta}{2}\right) \quad \hat{u}_x \sin\left(\frac{\theta}{2}\right) \quad \hat{u}_y \sin\left(\frac{\theta}{2}\right) \quad \hat{u}_z \sin\left(\frac{\theta}{2}\right) \right] \left[ 0 \quad \frac{t_x}{2} \quad \frac{t_y}{2} \quad \frac{t_z}{2} \right] \right\} \quad (34)$$

De acordo com a notação apresentada na equação (34), o primeiro quatérnio que forma o número dual está diretamente associado à rotação do *frame* de referência, enquanto o segundo está relacionado à translação do mesmo.

### 2.3.4 Cinemática Direta Via Quatérnios Duais Unitários

Para usar a metodologia através dos QDU na cinemática direta, é necessário obter um operador na forma de quatérnio dual capaz de transformar um sistema de coordenadas em outro usando as informações fornecidas pelos parâmetros  $\alpha$ ,  $a$ ,  $\theta$  e  $d$ , da mesma maneira que a equação (11) opera no método de MTH. Radavelli et al. (2014) descrevem tal operador por meio da equação (35), onde  $QD_{T_i}^{i-1}$  representa o quatérnio dual que transforma o sistema de coordenadas posicionado na origem da junta  $(i - 1)$  no sistema posicionado na origem da junta  $i$ .

$$QD_{T_i}^{i-1} = \begin{bmatrix} \cos\left(\frac{\alpha_i}{2}\right) \cos\left(\frac{\theta_i}{2}\right) \\ \sin\left(\frac{\alpha_i}{2}\right) \cos\left(\frac{\theta_i}{2}\right) \\ \sin\left(\frac{\alpha_i}{2}\right) \sin\left(\frac{\theta_i}{2}\right) \\ \cos\left(\frac{\alpha_i}{2}\right) \sin\left(\frac{\theta_i}{2}\right) \\ -\frac{a_i}{2} \sin\left(\frac{\alpha_i}{2}\right) \cos\left(\frac{\theta_i}{2}\right) - \frac{d_i}{2} \cos\left(\frac{\alpha_i}{2}\right) \sin\left(\frac{\theta_i}{2}\right) \\ \frac{a_i}{2} \cos\left(\frac{\alpha_i}{2}\right) \cos\left(\frac{\theta_i}{2}\right) - \frac{d_i}{2} \sin\left(\frac{\alpha_i}{2}\right) \sin\left(\frac{\theta_i}{2}\right) \\ \frac{a_i}{2} \cos\left(\frac{\alpha_i}{2}\right) \sin\left(\frac{\theta_i}{2}\right) + \frac{d_i}{2} \sin\left(\frac{\alpha_i}{2}\right) \cos\left(\frac{\theta_i}{2}\right) \\ -\frac{a_i}{2} \sin\left(\frac{\alpha_i}{2}\right) \sin\left(\frac{\theta_i}{2}\right) + \frac{d_i}{2} \cos\left(\frac{\alpha_i}{2}\right) \cos\left(\frac{\theta_i}{2}\right) \end{bmatrix} \quad (35)$$

A partir do operador apresentado na equação (35), a criação da estrutura robótica se dá da mesma maneira que através das MTH, concatenando o quatérnio dual de cada par junta-elo com próximo até o efetuador (KENWRIGHT, 2012). Tal operação ocorre como destacado na equação (36), onde  $n$  é o número de graus de liberdade do manipulador.

$$Q_{0-n} = Q_0 \otimes Q_1 \otimes Q_2 \otimes \dots \otimes Q_n \quad (36)$$

Kenwright (2012) também apresenta o método para extrair a informação de posição e orientação do efetuador através dos quatérnios  $q_r$  e  $q_d$  que formam  $Q_{0-n}$ . A informação de orientação é obtida conforme equação (37) e a posição conforme equação (38), sendo que  $q_r^*$  é o quatérnio complexo conjugado de  $q_r$ , computado conforme equação (16).

$$r = q_r \quad (37)$$

$$t = 2q_d q_r^* \quad (38)$$

Aspragathos e Dimitros (1998) fazem um comparativo entre as MTH e outros métodos de cinemática robótica, incluído os QDU. Em todos os quesitos estudados, os QDU exigem menor esforço computacional e espaço de memória. Os autores também expõem o fato de que as MTH necessitam de ao menos 12 parâmetros para representar as 6 componentes de posição e orientação de corpos rígidos, enquanto

QDU exigem apenas 8. Seus resultados são corroborados em um estudo semelhante feito por Vidakovic et al. (2014).

Sharkawy e Aspragathos (2017) realizaram um estudo comparativo entre dois métodos para solucionar a cinemática direta. Neste trabalho, os autores comparam um método via Produto de Exponenciais com um método via QDU utilizando o *software* Matlab na estrutura do robô Kuka LWR. O desenvolvimento deste trabalho mostra que os QDU exigem uma quantidade consideravelmente menor de operações matemáticas e de espaço de memória para serem computados. Os autores concluem afirmando que os QDU são preferíveis para aplicações em tempo real e que a comunidade robótica deve considerar o uso amplo desta ferramenta para tratar problemas que envolvem rotação e translação de efetuadores.

Kenwright (2013) descreve os quatérnios duais como uma técnica eficiente computacionalmente, direta, minimalista e livre de singularidades de representação para computar rotações e translações em uma única variável de estado e, portanto, permitem a descrição cinemática de estruturas articuladas de forma unívoca e com volume de álgebra reduzido. Marinho, Figueredo e Adorno (2015), afirmam que as vantagens da descrição cinemática baseada em quatérnios duais, também representa um cenário relevante para estudos envolvendo geração e controle de trajetória para o efetuador.

## 2.4 CINEMÁTICA INVERSA

A cinemática inversa, do inglês *Inverse Kinematics* (IK), é um componente vital na robótica que pode ser solucionado de diversas maneiras. Para estruturas mais simples de poucos elos é possível formular uma solução analítica, porém, em estruturas mais complexas, é necessário recorrer a métodos numéricos para encontrar a configuração desejada dada a complexidade do problema (KENWRIGHT, 2013).

### 2.4.1 Formulação do Problema de Cinemática Inversa

O problema da cinemática inversa é antagônico ao da cinemática direta: dada certa posição e orientação do efetuador em relação ao *frame* da base, procura-se o conjunto de variáveis de junta que satisfaçam tal configuração (CRAIG, 2012). Kenwright (2013) denomina a IK como um problema altamente não-linear e Aristidou

et al. (2017) complementam afirmando que, para cumprir determinada pose, é possível haver múltiplas soluções bem como é possível que não exista solução alguma.

O objetivo da IK para um manipulador constituído por  $n$  juntas rotacionais, sendo  $\theta$  a variável de junta, é encontrar o vetor  $\Theta = [\theta_1, \theta_2, \dots, \theta_n]$  que cumpra as condições de pose de destino do efetuador. Para uma estrutura formada por juntas rotacionais, a IK pode ser descrita como o inverso da função que relaciona a pose do efetuador ( $s$ ) a uma função  $f(\theta)$ , conforme equação (39) (ARISTIDOU et al., 2017).

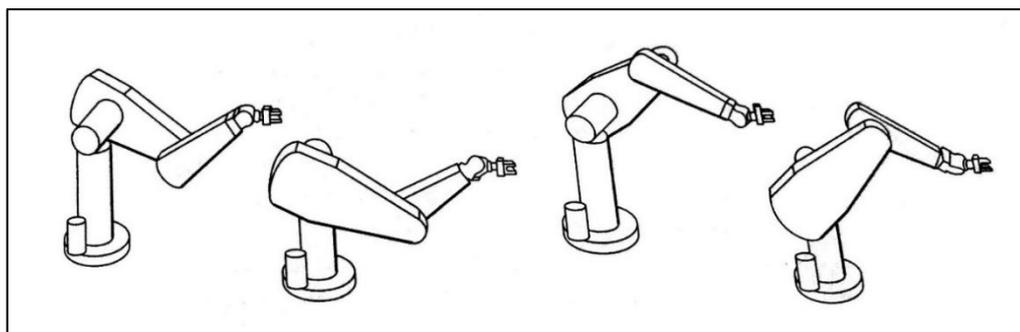
$$s = f^{-1}(\theta) \quad (39)$$

Aristidou et al. (2017) classificam os problemas de cinemática inversa em dois tipos: sub-restritos (ou redundantes) e sobre-restritos. Os autores baseiam tal classificação no fato de que a quantidade de soluções para a IK depende da posição cartesiana de destino, também chamada de alvo, e da quantidade de DoF do manipulador. Sendo assim, os problemas de IK se resumem em:

- a) redundante: quando o alvo está dentro do alcance do manipulador para uma estrutura com mais de 2 DoF. Para configurações desse tipo é possível obter de 2 a infinitas soluções que satisfazem a configuração desejada;
- b) sobre-restrito: ocorre caso o alvo esteja a uma distância maior que o atuador possa alcançar, ou onde nenhuma rotação de juntas pode levar o manipulador até o alvo.

Através da Figura 11, Craig (2012) usa o manipulador PUMA 560 para exemplificar quatro possíveis soluções para uma pose do efetuador. Observa-se que, em todas as configurações apresentadas, o efetuador está posicionado no mesmo ponto cartesiano e orientado na mesma direção em relação à base, configurando um problema redundante.

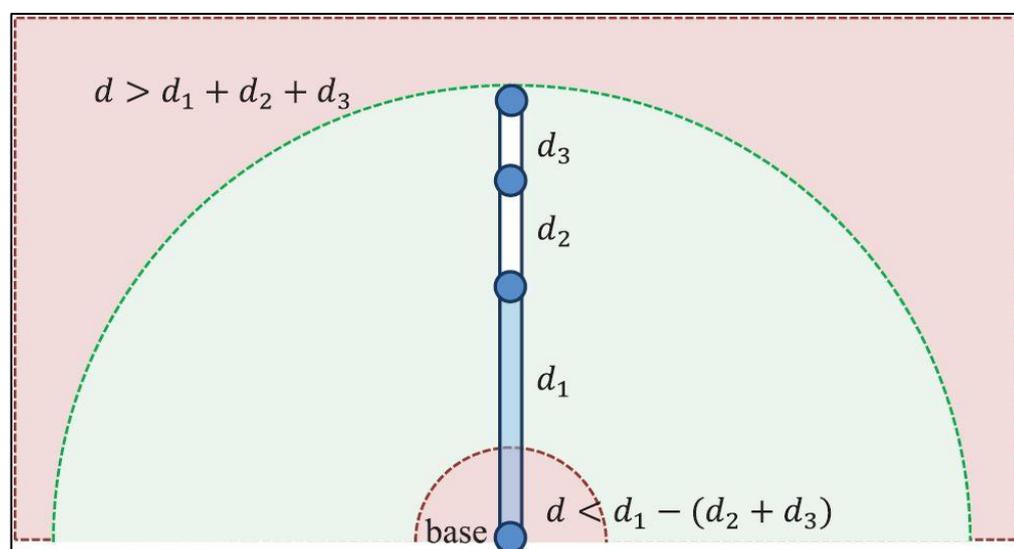
Figura 11 – Múltiplas soluções de IK para um manipulador PUMA 560.



Fonte: Craig (2012, p. 97)

Já na Figura 12, Aristidou et al. (2017) apresentam um manipulador planar como exemplo simplificado para um problema de IK sobre-restrito. A região destacada em vermelho perto da base não pode ser alcançada pelo efetuador pois é onde a distância até o alvo é menor que o comprimento  $d_1 - (d_2 + d_3)$ , ou seja, dado o comprimento de cada um dos 3 elos ( $d_n$ ) é impossível um arranjo capaz de posicionar o efetuador nesta região. Da mesma forma, a região destacada em vermelho no exterior da figura é inalcançável, porque excede o alcance máximo do manipulador, dada a soma dos comprimentos dos elos,  $d_1 + d_2 + d_3$ .

Figura 12 – Exemplo do espaço de trabalho de um manipulador planar de 2 DoF.



Fonte: Aristidou et al. (2017)

Definir as regiões não-alcançáveis para o efetuador é uma tarefa de interesse, no entanto é um trabalho complexo para manipuladores com mais de 3 DoF. Apesar da importância de definir *a priori* se o alvo é alcançável ou não, este é um assunto que requer estudos mais aprofundados e detalhados, como explicam Aristidou et al. (2017). Segundo Buss e Kim (2004), em casos onde a posição de destino está fora

de alcance, o manipulador se estende na direção do alvo sem poder alcançá-lo e, portanto, não permite que o erro convirja a zero, causando oscilações. Os autores afirmam que, em algoritmos numéricos, devem ser implementadas condicionais de parada caso o manipulador não seja mais capaz de reduzir o erro, ou caso tenha oscilações que aumentam o erro sem contribuir para a convergência.

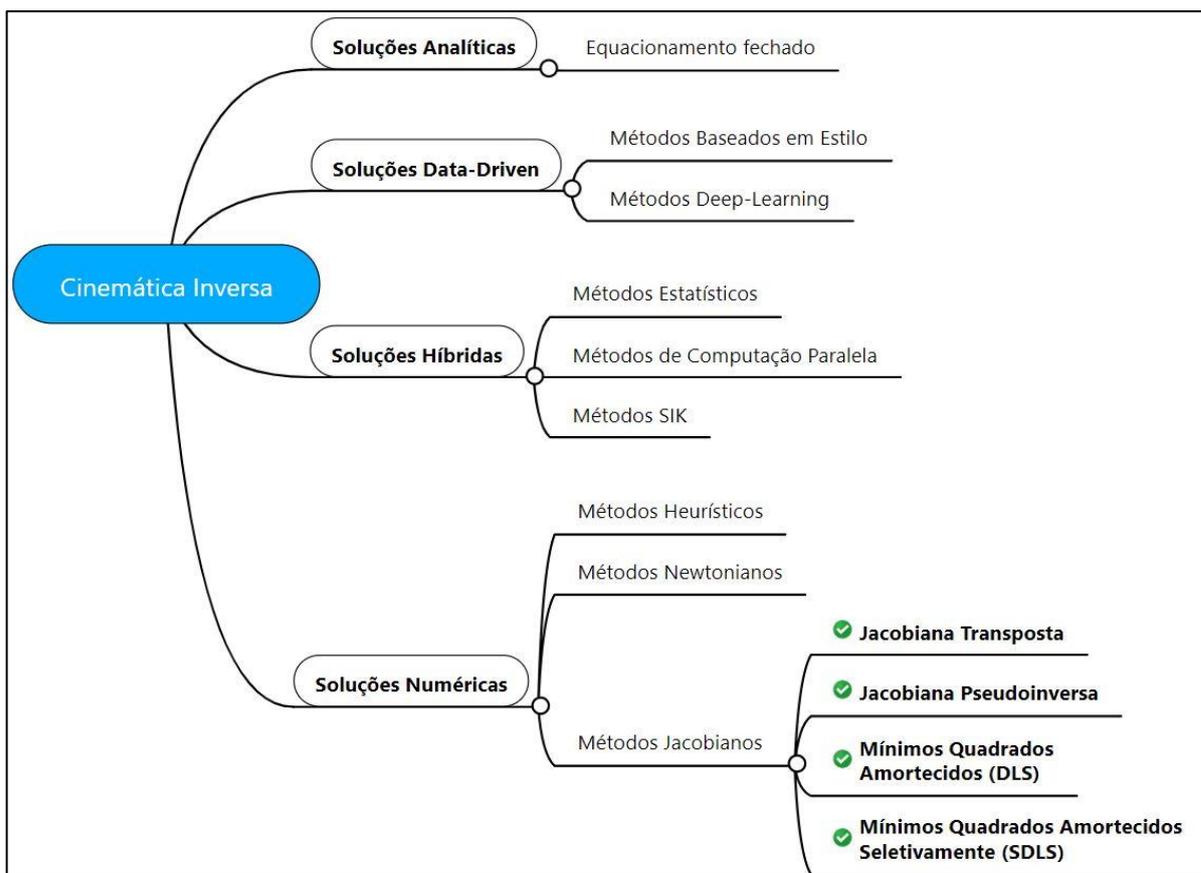
Ainda conforme o trabalho de Aristidou et al. (2017), a cinemática inversa pode ser solucionada através de métodos analíticos. Para tais métodos, dada a posição desejada do efetuador, busca-se um conjunto de equações capazes de fornecer uma ou todas as soluções para a IK, com base em suposições do projetista.

Craig (2012) aborda os aspectos de soluções analíticas dando ênfase ao método de Pieper, o qual fornece uma solução para a IK, porém exige que o manipulador possua até 6 DoF e que ao menos três de seus eixos consecutivos cruzem-se em um único ponto. Craig (2012) destaca que tal método pode ser aplicado à maior parte dos manipuladores industriais, porém, o mesmo não fornece uma solução genérica para a cinemática inversa, cobrindo apenas casos especiais.

Apesar da simplicidade de solução analítica em estruturas com poucos DoF, os cálculos de IK tornam-se significativamente complexos em cadeias com maior quantidade de graus de liberdade e no espaço tridimensional. Também não há escalabilidade, significando que há dificuldade em lidar com estruturas complexas como robôs antropomórficos que podem ter em torno de 70 DoF trabalhando em conjunto em diversos efetuadores. Tais métodos são rápidos computacionalmente e não sofrem com o problema de singularidade, o qual será abordado no decorrer da subseção 2.4.2, porém, são recomendados apenas para hierarquias simples e com configurações restritas (ARISTIDOU et al., 2017).

A dificuldade na obtenção de um equacionamento analítico prático motiva a criação de diversos métodos alternativos para solucionar a cinemática inversa. Ainda conforme Aristidou et al. (2017), os métodos desenvolvidos podem ser divididos em quatro grupos: analíticos, numéricos, *data-driven* e híbridos. Estes grupos são subdivididos em categorias, conforme estrutura apresentada na Figura 13.

Figura 13 – Classificação dos métodos de solução para a IK.



Fonte: O autor.

Na Figura 13, destacam-se os métodos Jacobianos de solução numérica, os quais são o foco deste trabalho. Uma análise detalhada dos fundamentos matemáticos destes métodos é feita na subseção 2.4.4.

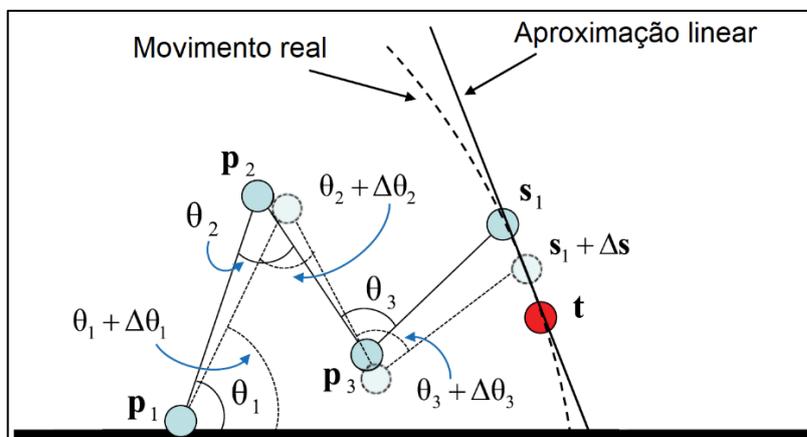
## 2.4.2 Matriz Jacobiana

A matriz Jacobiana é o elemento crítico em algoritmos usados para gerar poses e trajetórias de estruturas robóticas, sendo a matriz de coeficientes que relaciona as taxas de variação da pose do efetuador com as variáveis de junta do manipulador. Tal conjunto de coeficientes pode ser construído de várias maneiras e, devido ao seu amplo uso, há diversos métodos de computação para ela (WALDRON, WANG e BOLIN, 1985).

Através da Figura 14, Aristidou et al. (2017) representam a aproximação linear objetivada com o uso da matriz Jacobiana no movimento de um efetuador. Após incrementar os ângulos das juntas por um valor  $\Delta\theta$ , o efetuador é deslocado em uma trajetória curvilínea. Os autores explicam que, para deslocamentos suficientemente

pequenos, considera-se que a aproximação linear coincide com o movimento real, validando a aplicabilidade da Jacobiana.

Figura 14 – Aproximação linear da matriz Jacobiana.



Fonte: Adaptado de Aristidou et al. (2017).

Buss (2009) revisa em seu trabalho os conceitos algébricos da matriz Jacobiana. Uma notação para esta matriz é conforme a equação (40), onde a matriz é dada em função dos ângulos de junta  $\theta$ . Na equação (40),  $\partial s$  representa a variação pose do efetuador em dado instante, atrelada à variação de  $\theta_i$ .

$$J(\theta) = \left( \frac{\partial s}{\partial \theta_i} \right) \quad (40)$$

A partir dessa relação, Buss (2009) apresenta a equação (41), a qual descreve a variação da pose do efetuador,  $\dot{s}$ , em função da variação dos ângulos de junta,  $\dot{\theta}$ . A notação  $\dot{x}$  representa a derivada de primeira ordem da variável  $x$ .

$$\dot{s} = J(\theta)\dot{\theta} \quad (41)$$

A matriz Jacobiana da equação (41) denota a variação de posição e orientação do efetuador simultaneamente, uma vez que é uma matriz formada por estas duas componentes, de acordo com a notação da equação (42) (ENGARDT, HEIMBÜRGER e SYDHOFF, 2012).

$$J(\theta)\dot{\theta} = \begin{bmatrix} v_s \\ \omega_s \end{bmatrix} \quad (42)$$

Na equação (42),  $v_s$  é a variação da posição e  $\omega_s$  é a variação da orientação do efetuador. Tais relações, por representar derivadas de primeira ordem de

grandezas posicionais, também são referidas como velocidade translacional e velocidade rotacional, respectivamente (ÖZGÜRA e MEZOUARB, 2015).

Segundo Buss (2009), pode-se utilizar a equação (43) para computar a matriz Jacobiana de uma junta rotacional.

$$\frac{\partial s}{\partial \theta_i} = v_i \times (s - p_i) \quad (43)$$

Na equação (43),  $v_i$  é um vetor unitário direcionado ao longo do eixo de rotação da junta  $i$  e o ponto  $p_j$  é a posição do efetuador  $s$ . Conforme destacado por Buss (2009), se o efetuador não é afetado pela junta  $i$ , então equação (43) resultará em zero. Kenwright (2012) afirma que essa é a forma prática de calcular a Jacobiana em algoritmos computacionais.

A Jacobiana é aplicada em métodos de IK numéricos, os quais buscam incrementar ou decrementar os valores de  $\theta$  de toda a estrutura a cada iteração por um  $\Delta\theta$ . Como  $J$  é uma aproximação linear da variação da pose do efetuador ( $\Delta s$ ), tal variação, pode ser estimada por meio da equação (44) (BUSS, 2009).

$$\Delta s \approx J\Delta\theta \quad (44)$$

Ainda conforme Buss (2009),  $\Delta\theta$  deve ser escolhido de tal forma que a variação da pose do efetuador seja aproximadamente igual ao erro  $\vec{e}$  existente entre a pose atual e a desejada, conforme pode ser observado na equação (45). Dessa forma, o algoritmo poderá atualizar os ângulos das juntas a cada iteração ou atualizar os ângulos uma vez após obter um valor de  $\Delta s$  suficientemente próximo ao desejado. O cálculo do erro de pose  $\vec{e}$  exige uma metodologia própria que admita o uso de quatérnios duais unitários, tema o qual será revisado na subseção 2.4.3.

$$\vec{e} = \Delta s \quad (45)$$

Nakamura e Hanafusa (1986) explicam que, para resolver a IK em função do vetor  $\theta = [\theta_1, \theta_2, \dots, \theta_n]$  em métodos baseados em Jacobianas, é necessário computar a inversa da matriz  $J(\theta)$ , denotada como  $J^{-1}(\theta)$ . O requisito para que seja possível realizar essa operação é que o determinante de  $J(\theta)$  seja diferente de zero, ou seja,  $\det\{J(\theta)\} \neq 0$ . Quando essa condição não é atendida, o vetor  $\theta$  está em uma situação chamada de configuração singular, ou singularidade.

As configurações singulares podem existir até mesmo dentro da área de alcance do manipulador, e nesses pontos a cinemática inversa não pode ser resolvida pela matriz Jacobiana. Ademais, comumente é exigida uma grande variação do ângulo  $\theta$  de alguma junta em pontos próximos a uma singularidade, causando um súbito erro na posição do efetuador (NAKAMURA e HANAFUSA, 1986).

Segundo Waldron, Wang e Bolin (1985), as configurações singulares podem existir em qualquer ponto da área de trabalho do manipulador. Em estruturas pequenas, é possível equacionar analiticamente os pontos onde o determinante da Jacobiana é zero, porém, pode ser uma tarefa impossível em estruturas com maior número de DoF. Os autores afirmam que as singularidades, em muitos manipuladores industriais, podem ser descritas através de condições algébricas simples, porém, com um pequeno aumento na complexidade da estrutura, a localização destes pontos pode tornar-se completamente imprevisível.

Dessa forma, o problema de singularidades na inversão da Jacobiana é inerente de tais métodos e deve ser tratado dentro do algoritmo (NAKAMURA e HANAFUSA, 1986). Uma maneira de lidar com o problema das singularidades é evitar a inversão direta da matriz Jacobiana, através de artifícios matemáticos, e amortecer sua influência em regiões próximas às singularidades. Tais técnicas são um dos pontos que diferem os métodos Jacobianos que serão analisados na subseção 2.4.4.

### **2.4.3 Cálculo de Erro e Condições de Parada**

Em sistemas de controle, o intuito é controlar uma variável para que a diferença entre seu valor atual e o valor desejado convirja a zero, pois este é indicativo de que a planta atingiu o *setpoint*. Projetar um sistema de controle para um manipulador construído no espaço de quatérnios duais exige que as variáveis de controle cumpram suas condições de erro zero no domínio dos números duais (Pham et al., 2018).

Ainda conforme Pham et al. (2018), para definir o erro da pose de um manipulador, deve existir o controle tanto da posição quanto da orientação cartesiana. Para controlar o erro de posição, uma simples diferença de vetores é suficiente, porém, o erro de orientação é computado de maneira diferente dependendo de como ela foi parametrizada. Comumente usam-se dois *loops* de controle separados: um para posição e outro para orientação.

O erro de orientação é calculado a partir do quatérnio relativo entre a orientação do efetuador atual e a orientação desejada, o qual transforma  $q_{ef}$  em  $q_d$ . A equação (46) retorna o quatérnio relativo de orientação, denominado  $q_{rel}$ , conforme descrito por Mayo (1979). Na equação (46),  $q_{rel}$  é o quatérnio (não necessariamente unitário) que representa o erro de orientação,  $q_{ef}^*$  é o complexo conjugado do quatérnio de orientação do efetuador na pose atual e  $q_d$  é o quatérnio da orientação desejada.

$$q_{rel} = q_d q_{ef}^* \quad (46)$$

Em situações onde o problema de IK é sobre-restrito, como definido na subseção 2.4.1, o efetuador não alcançará a pose desejada, portanto,  $Q_{erro}$  não convergirá a zero, independentemente da quantidade de iterações. Para estes casos, é necessário definir uma situação de parada em que o programa abortará o cálculo se alguma das condições estabelecidas ocorrer. As condições de parada também devem atuar para o tratamento de situações onde o efetuador está próximo a uma singularidade, mesmo que o problema seja redundante (NAKAMURA e HANAFUSA, 1986).

Kenwright (2012) resolve a cinemática inversa utilizando um conjunto de condições de parada para o algoritmo baseado no erro de pose, no número de iterações realizadas e na variação de resultados entre uma iteração e outra. O autor descreve o método resolvendo uma equação em função de uma variável  $x$ , partindo de um valor inicial  $x_0$ , e tendo como resultado de cada *loop* um valor  $\Delta x$ . Na iteração seguinte, o valor de  $x_0$  é substituído por  $x_0 + \Delta x$  e o processo se repete até que uma das seguintes condicionais ocorra:

- a) se o número máximo de iterações pré-definido for atingido;
- b) se  $Q_{erro}$  tiver valor abaixo de um limite mínimo (precisão);
- c) se  $|\Delta x|_i$  for menor que determinada tolerância;
- d) se  $|\Delta x|_i = |\Delta x|_{i-1}$ , dentro de determinada tolerância;

#### 2.4.4 Métodos Numéricos Jacobianos

Como a IK é comumente um problema de múltiplas soluções, a escolha da melhor pose, ou melhor solução, depende de muitos fatores, conforme explicado por

Aristidou et al. (2017). Entre estes fatores, o método aplicado pode ser capaz de representar uma ou diversas soluções do problema. Caso a metodologia forneça mais de uma solução, é necessário implementar uma tomada de decisão. Os autores também afirmam que há situações em que o sistema é dependente de tempo de execução em detrimento de sua precisão. Outros sistemas podem exigir precisão posicional, sem preocupação quanto ao esforço computacional. Certas aplicações podem exigir otimização na função de custo do movimento da estrutura, baseado em fatores dinâmicos, e pode haver a exigência de fatores subjetivos, como a naturalidade do movimento e da pose para simular configurações antropomórficas. Este último elemento é mensurado apenas por experimentos neurofisiológicos e de pura observação, concluem os autores.

Dada a necessidade de combinar o melhor método à situação prática em questão, faz-se necessária a análise comparativa dos algoritmos disponíveis. Com base no desempenho de cada método, pode definir-se qual cumpre melhor as diretrizes desejadas.

Aristidou et al. (2017) definem os métodos numéricos como os que resolvem a IK através de algoritmos que reduzem o erro entre a posição real e a desejada a cada iteração, respeitando as condições de parada. Nesta subseção, são abordados os métodos numéricos Jacobianos, os quais fazem parte do escopo deste trabalho.

Conforme explicado na seção 2.4.2, a matriz Jacobiana é aplicada em métodos de IK numéricos onde busca-se incrementar ou decrementar os valores de  $\theta$  de toda a estrutura a cada iteração por um  $\Delta\theta$ . Também foi esclarecido que, para solucionar a equação  $s = f^{-1}(\theta)$ , a qual representa a solução da IK, é necessário realizar a inversão desta matriz,  $J^{-1}(\theta)$ . De acordo com Waldron, Wang e Bolin (1985) e Nakamura e Hanafusa (1986), a inversão da Jacobiana pode ser impossível devido à presença de configurações singulares e, portanto, há a necessidade de utilizar métodos que contornem este problema.

#### 2.4.4.1 Jacobiana Transposta

Uma maneira de evitar a inversão direta da Jacobiana é utilizando o método da Jacobiana Transposta,  $J^T$ , o qual consiste em encontrar  $\Delta\theta$  através da equação (47) (BUSS, 2009). O autor esclarece que a transposta de uma matriz não é igual à sua

inversa, porém, conforme o teorema e a prova apresentados em seu trabalho, o uso desta equivalência é justificado.

$$\Delta\theta = \alpha J^T \vec{e} \quad (47)$$

Na equação (47),  $\alpha$  é um escalar que não deve ser confundido com o parâmetro DH de mesma nomenclatura. Seu valor é escolhido de modo a minimizar o valor do erro de pose  $\vec{e}$  a cada iteração. Uma maneira de atualizar o valor de  $\alpha$  é fazer com que a variação da pose do efetuador seja aproximadamente a Jacobiana multiplicada pela variação dos ângulos de junta, conforme descrito na equação (41). Uma forma de fazer esta operação é aplicando a equação (48) (BUSS, 2009).

$$\alpha = \frac{[\vec{e} \circ JJ^T \vec{e}]}{[JJ^T \vec{e} \circ JJ^T \vec{e}]} \quad (48)$$

Buss (2009) afirma que essa metodologia é a de implementação mais simples e execução mais rápida, porém introduz oscilações na pose do efetuador, dificultando sua convergência ao ponto desejado.

#### 2.4.4.2 Jacobiana Pseudoinversa

A Jacobiana pseudoinversa,  $J^\dagger$ , também é conhecida como inversa de Moore-Penrose, segundo Aristidou et al. (2017), é uma alternativa para encontrar a solução de IK com o intuito de atingir o objetivo com os menores valores de  $\Delta\theta$  possíveis (ou seja, encontra os mínimos quadrados de  $\Delta\theta$ ), através da formulação apresentada na equação (49) (BUSS, 2009).

$$\Delta\theta = J^T (JJ^T)^{-1} \vec{e} \quad (49)$$

Buss (2009) e Nakamura e Hanafusa (1986) explicam que esta formulação do problema também tem a propriedade de convergir a uma solução, mesmo que o alvo esteja fora do alcance do manipulador, sempre minimizando a magnitude do erro. No entanto, para configurações próximas às singularidades a Jacobiana pseudoinversa irá gerar grandes variações de ângulo entre uma iteração e outra mesmo para pequenos deslocamentos de efetuador, o que pode gerar instabilidade no processo.

### 2.4.4.3 Mínimos Quadrados Amortecidos

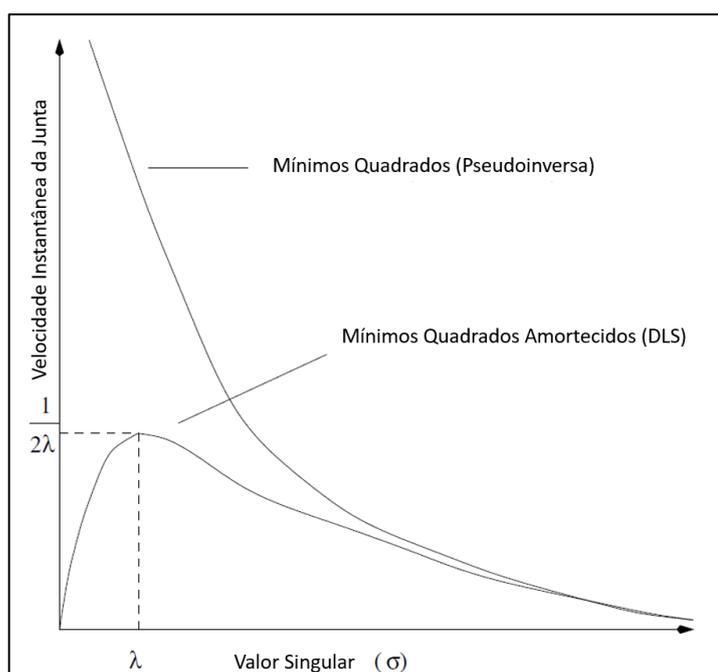
O algoritmo de Levenberg–Marquardt, ou Mínimos Quadrados Amortecidos, do inglês *Damped Least Squares* (DLS), foi desenvolvido com o objetivo de evitar os problemas de oscilação em pontos próximos às configurações singulares presentes na Jacobiana pseudoinversa. O método DLS utiliza uma constante de amortecimento  $\lambda$  para abrandar a ação da Jacobiana, o que aumenta a estabilidade do algoritmo nas proximidades de uma singularidade (BUSS, 2009).

Ao inserir a constante de amortecimento na equação (49), obtém-se a equação (50), conforme apresentado por Buss (2009). O autor explica que essa constante deve ter seu valor escolhido previamente, devendo ser grande o suficiente para manter o sistema controlado próximo às singularidades, porém, pequeno o suficiente para não comprometer a taxa de convergência do algoritmo. Na equação (50),  $I$  é a matriz identidade que ajusta a dimensão das variáveis para que o cálculo seja factível.

$$\Delta\theta = J^T (JJ^T + \lambda^2 I)^{-1} \vec{e} \quad (50)$$

No gráfico da Figura 15, Ge (2000) compara o comportamento da velocidade instantânea das juntas nos métodos de pseudoinversa e DLS. Quanto mais próximo o sistema está de uma configuração singular, menor é o valor singular  $\sigma$ .

Figura 15 – Comparativo da solução pela pseudoinversa e pelo DLS.



Fonte: Ge (2000)

De acordo com Ge (2000) e conforme demonstrado na Figura 15, quanto mais próximo o sistema está de uma configuração singular no método da pseudo inversa, maior será a velocidade instantânea da junta, divergindo quando  $\sigma = 0$ . Para o método DLS, a constante de amortecimento faz com que o sistema tenha velocidade instantânea reduzida nas proximidades da singularidade e reduz esta velocidade a zero caso o sistema entre exatamente na configuração singular.

#### 2.4.4.4 Mínimos Quadrados Amortecidos Seletivamente

O método dos Mínimos Quadrados Amortecidos Seletivamente, do inglês *Selectively Damped Least Squares* (SDLS), foi introduzido por Buss e Kim (2004). Para a compreensão das implicações do uso desse método, é necessário expor o conceito da Decomposição em Valores Singulares, do inglês *Singular Value Decomposition* (SVD). Segundo Buss (2009), a operação SVD fatora a matriz Jacobiana em três matrizes, conforme equação (51).

$$J = UDV^T \quad (51)$$

Buss (2009) demonstra que, na equação (51), as componentes  $U$  e  $V$  são matrizes ortogonais e  $D$  é uma matriz diagonal. Se  $J$  é uma matriz  $m \times n$ , então  $U$  é  $m \times m$ ,  $D$  é  $m \times n$  e  $V$  é  $n \times n$ . Usa-se a notação  $u_i$  para as colunas da matriz  $U$  e  $v_i$  para as colunas da matriz  $V$ . O autor também explica que os únicos termos diferentes de zero na matriz  $D$  são os valores  $\sigma_i$  ao longo de sua diagonal principal, sendo que  $\sigma_i \geq 0$  e estão dispostos em ordem crescente. Os valores de  $\sigma_i$  são os chamados valores singulares da matriz  $J$ .

Sendo o termo  $\sigma_r$  o menor valor diferente de zero em  $D$ , então  $r$  é o posto da matriz Jacobiana. Essas definições implicam que a Jacobiana pode ser reescrita conforme equação (52) (BUSS, 2009).

$$J = \sum_{i=1}^r \sigma_i u_i v_i^T \quad (52)$$

A equação (50), usada para calcular os valores de  $\Delta\theta$  no método DLS, pode ser reescrita utilizando os termos da SVD, conforme a equação (53) (BUSS, 2009).

$$\Delta\theta = VD^T(DD^T + \lambda^2 I)^{-1}U^T\vec{e} = VEU^T \quad (53)$$

Ainda de acordo com Buss (2009), a matriz  $E$  da equação (53) é diagonal e de dimensões  $n \times m$  com os termos  $\tau_i$  em sua diagonal principal, calculados conforme equação (54).

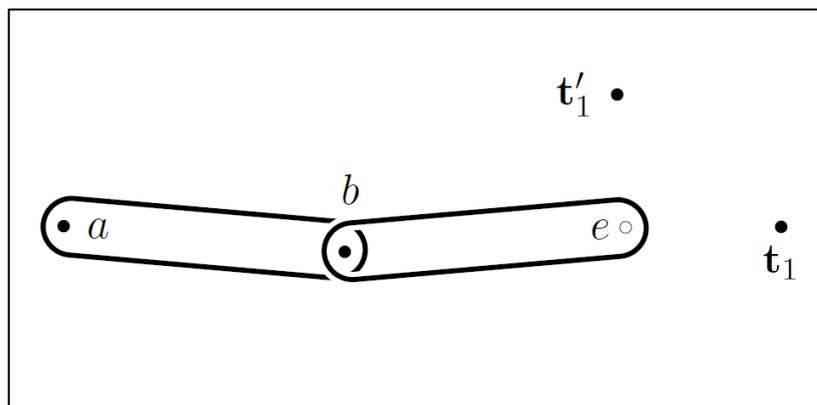
$$\tau_i = \frac{\sigma_i}{\sigma_i^2 + \lambda^2} \quad (54)$$

Portanto, substituindo-se a equação (54) na equação (53), obtêm-se a equação (55), a qual é a solução do método DLS através da decomposição em valores singulares. Buss (2009) afirma que, conforme  $\sigma$  aproxima-se de zero, a equação (55) efetivamente amortece a performance do algoritmo próximo a singularidades.

$$\Delta\theta = \sum_{i=1}^r \tau_i v_i u_i^T \vec{e} \quad (55)$$

Apesar da eficácia em contornar o problema de singularidades, o método DLS exige a definição prévia de uma constante de amortecimento  $\lambda$ , conforme demonstrado em 2.4.4.3. Buss e Kim (2004) expõem que essa necessidade traz uma limitação ao método utilizando um exemplo básico com o robô planar de 2 DoF. Na Figura 16, o ponto  $a$  é a base da estrutura,  $a$  e  $b$  são juntas rotacionais,  $e$  é o efetuador e os dois elos são praticamente colineares.

Figura 16 – Manipulador planar próximo de uma singularidade.



Fonte: Buss e Kim (2004).

Buss e Kim (2004) explicam as duas situações antagônicas representadas na Figura 16:

- a) no caso em que a IK está sendo resolvida para que o efetuador  $e$  aponte na direção de  $t_1$ , a pseudoinversa gera um grande ângulo no sentido anti-horário para a junta  $a$  e um grande ângulo no sentido horário para a junta  $b$ . Isso significa que o ponto  $t_1$  está próximo a uma singularidade e seria necessário um valor elevado de  $\lambda$  para amortecer o movimento pelo método DLS;
- b) no caso em que a IK está sendo resolvida para que o efetuador  $e$  aponte na direção de  $t_1'$ , a pseudoinversa gera ângulos no sentido anti-horário para as juntas  $a$  e  $b$ . Nesse caso, a aproximação linear da Jacobiana é precisa para os ângulos envolvidos e o ponto  $t_1'$  está distante de singularidades. Isso significa um movimento maior das juntas é apropriado e que um valor diminuto para  $\lambda$  é adequado.

Analisando as duas situações opostas que ocorrem em uma mesma estrutura, Buss e Kim (2004) concluem que a posição relativa entre o efetuador e o alvo deve ser levada em consideração na escolha da constante de amortecimento de forma seletiva e introduzem, assim, o método dos mínimos quadrados amortecidos seletivamente (SDLS). Os autores explicam que essa seleção é feita observando cada junta da estrutura individualmente e levando em conta o quanto ela está colaborando para o movimento do efetuador. Essa colaboração é comparada com a distância do efetuador até o ponto de interesse. Se a colaboração da junta é maior que a distância até o alvo, então o movimento da junta é amortecido mais severamente. Buss e Kim (2004) também definem o valor máximo para a variação do ângulo de uma junta a cada iteração como  $\gamma_{máx} = \frac{\pi}{4}$ .

Para realizar a análise dessas colaborações, primeiramente Buss e Kim (2004) consideram  $\alpha_i = u_i^T \vec{e}$  e rearranjam a equação (55) para encontrar a equação (56).

$$\Delta\theta = \sum_{i=1}^r \alpha_i \tau_i v_i \quad (56)$$

Em seguida, os autores realizam o equacionamento para o ângulo máximo que cada vetor singular pode contribuir para o movimento de uma junta,  $\gamma_i$ , a cada iteração conforme equações (57), (58), e (59).

$$N_i = \sum_{j=1}^k \|u_{j,i}\| \quad (57)$$

$$M_i = \sigma_i^{-1} \sum_{j=1}^n |v_{j,i}| \cdot \left| \left( \frac{\partial s}{\partial \theta} \right)_i \right| \quad (58)$$

$$\gamma_i = \min \left( 1, \frac{N_i}{M_i} \right) \cdot \gamma_{máx} \quad (59)$$

A equação (57) representa o somatório da magnitude de todos os vetores  $u_j$  da  $i$ -ésima coluna da matriz  $U$ . Na equação (58),  $|v_{j,i}|$  é o módulo dos vetores  $v_i$  da  $i$ -ésima coluna da matriz  $V$  e a expressão  $\left| \left( \frac{\partial s}{\partial \theta} \right)_i \right|$  representa o módulo da Jacobiana da  $i$ -ésima junta, portanto,  $M_i$  computa a distância em que o efetuador move-se como consequência de uma rotação na junta  $i$ . O máximo ângulo de rotação da junta  $i$ , definido como  $\gamma_i$ , é calculado conforme relação exposta na equação (59) (BUSS e KIM, 2004).

Completando a definição do método SDLS, Buss e Kim (2004) definem a variável auxiliar  $\varphi_i$  na equação (60), sendo que  $w_i = \sigma_i^{-1} \alpha_i v_i$ .

$$\varphi_i = \begin{cases} w_i, & \text{se } \|w_i\| \leq \gamma_i \\ \gamma_i \frac{w_i}{\|w_i\|}, & \text{caso contrário} \end{cases} \quad (60)$$

Finalizando a definição do algoritmo, a variação dos ângulos de cada junta  $\varphi$  são postos em escala para que não seja violado o limite máximo de variação imposto,  $\gamma_{máx}$ . Os autores definem a equação (61) para os valores de  $\Delta\theta$  de cada iteração, salientando que devem ser considerados apenas os valores de  $\sigma_i \neq 0$ . Os autores salientam que tanto na equação (61) quanto na equação (62),  $\|x\|$  é a norma-1 de  $x$ , e não a norma Euclidiana de  $x$ .

$$\Delta\theta = \begin{cases} \varphi_i, & \text{se } \|\varphi_i\| \leq \gamma_{m\acute{a}x} \\ \gamma_{m\acute{a}x} \frac{\varphi_i}{\|\varphi_i\|}, & \text{caso contr\acute{a}rio} \end{cases} \quad (61)$$

## 2.5 ANÁLISE DOS TRABALHOS RELEVANTES

Nesta seção, é feita a revisão dos trabalhos recentes envolvendo a obtenção de métricas de desempenho e metodologias de comparação dos algoritmos numéricos de IK, descritos na subseção 2.4.4. Os dados coletados e as formas de apresentação dos comparativos descritos pelos autores foram base para formulação da proposta deste trabalho.

Buss e Kim (2004), em seu trabalho que introduziu o método SDLS, realizaram um experimento comparando a Jacobiana transposta, DLS e SDLS em estruturas com múltiplos efetuadores em ambiente simulado. Os autores não consideraram limitações nas juntas, o que significa que todas as articulações puderam mover-se livremente dentro de seu alcance. O experimento foi projetado para testar a eficácia dos métodos em atingir a posição desejada, dentro de faixas de tolerância.

A rotina de testes do referido experimento de Buss e Kim (2004) considerou a alocação de pontos utilizados como alvo do efetuador de maneira descontínua, executando os métodos de IK até o efetuador chegar na posição desejada ou até falhar em continuar se aproximando. Os testes foram realizados 10.000 vezes para uma precisão de 0,1, 0,01, 0,001 e 0,0001 unidades de distância e foi realizada a contagem de iterações necessárias para a convergência do algoritmo.

Nos experimentos de Buss e Kim (2004), o método DLS foi testado com duas constantes de amortecimento:  $\lambda = 1,1$  e  $\lambda = 0,7$ , denominados DLS<sub>1</sub> e DLS<sub>2</sub>, respectivamente. Também foi mensurado o tempo de execução médio de uma iteração. O algoritmo dos autores considerou fim de rotina quando uma das condicionais foi atingida:

- a) se o efetuador atingiu uma distância menor que a precisão pré-definida;
- b) se a distância em que o efetuador se deslocou entre uma iteração e outra foi menor que  $10^{-5}$  unidades de distância;
- c) se distância do efetuador ao alvo aumentou em três iterações consecutivas, devido a oscilações;

d) se 4.000 iterações foram realizadas.

O SDLS foi o método que convergiu com o menor número de iterações em todos os cenários em que o algoritmo teve sucesso em chegar ao alvo. Para os pontos onde o algoritmo abortou sem atingir a precisão desejada, o DLS foi o que chegou mais próximo do alvo no menor número de iterações. Apesar das poucas iterações da Jacobiana transposta, os autores destacam que esta não convergiu em situações onde os outros algoritmos obtiveram sucesso. Isto ocorreu porque a oscilação de  $J^T$  abortou o teste em muitas das execuções, comprometendo sua performance.

A contagem média de iterações para cada método é apresentada na Tabela 3 (para os pontos onde o algoritmo cumpriu a métrica de precisão) e na Tabela 4 (para os pontos onde o código abortou). Os testes de Buss e Kim (2004) foram realizados em um computador *Pentium* 2.8GHz, sendo que os algoritmos foram implementados em linguagem C++. Os tempos de execução médios de uma iteração foram de  $6,5\mu s$  para a Jacobiana transposta,  $18,5\mu s$  para o DLS e  $120\mu s$  para o SDLS.

Tabela 3 – Média de iterações em que o algoritmo atingiu a precisão.

Método	Precisão de 0,1	Precisão de 0,01	Precisão de 0,001	Precisão de 0,0001
$J^T$	44,0	51,6	53,4	53,9
DLS <sub>1</sub>	61,5	231,6	467,1	635,2
DLS <sub>2</sub>	29,7	110,9	228,3	327,2
SDLS	39,3	48,7	52,2	53,5

Fonte: Adaptado de Buss e Kim (2004).

Tabela 4 – Média de iterações em que o algoritmo abortou.

Método	Precisão de 0,1	Precisão de 0,01	Precisão de 0,001	Precisão de 0,0001
$J^T$	44,0	142,27	142,3	142,3
DLS <sub>1</sub>	116,1	448,9	1.166,7	1.706,2
DLS <sub>2</sub>	75,2	318,7	863,6	1.289,5
SDLS	173,5	336,0	548,1	686,0

Fonte: Adaptado de Buss e Kim (2004).

Os autores concluem que o método SDLS é recomendado para casos em que é necessária maior precisão da posição do efetuador e quando seu maior tempo de execução é aceitável. Em casos em que houver conhecimento prévio da melhor

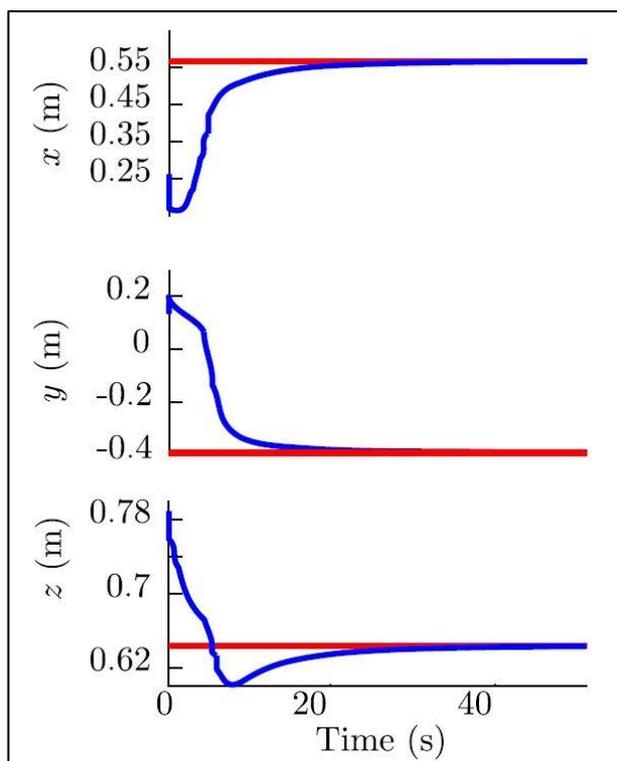
constante de amortecimento para o DLS, sua performance e facilidade de implementação o tornam o método indicado.

No trabalho de Pham et al. (2018), foi implementado um sistema de controle de *setpoint* para a pose do efetuador em um manipulador de 6 DoF, descrito através dos parâmetros DH e aplicando a álgebra dos QDU. Para a cinemática inversa, foram comparados os métodos da Jacobiana transposta e da pseudoinversa em um algoritmo que aplica teoria de controle, convergindo a IK utilizando um bloco integrador.

Pham et al. (2018) demonstraram a metodologia usada para estabelecer o cenário de testes e coletar as métricas de desempenho. O algoritmo dos autores foi executado 1.000 vezes e em cada uma delas foi gerada, de maneira aleatória, uma configuração inicial para as juntas do manipulador. Analogamente, foi gerada uma pose de destino. A geração das poses aleatórias foi feita utilizando uma distribuição uniforme de probabilidades para o ângulo das juntas.

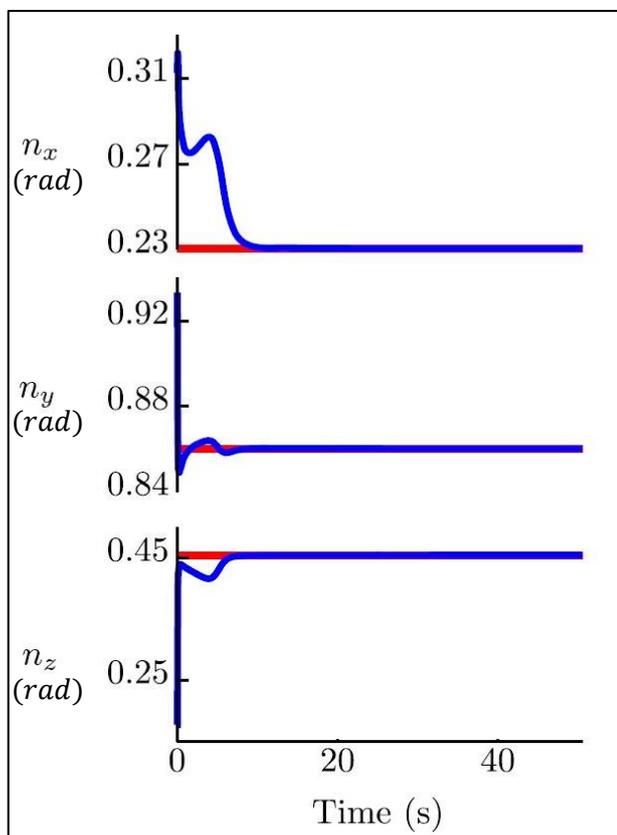
Depois de gerar a configuração inicial das juntas, Pham et al. (2018) aplicaram a cinemática direta através dos QDU para encontrar a pose inicial do efetuador e, então, executaram o algoritmo de cinemática inversa para coletar os dados de desempenho. A Figura 17 e a Figura 18 são exemplos de como os autores apresentaram graficamente o erro de posição e orientação cartesiana do efetuador nos eixos  $x$ ,  $y$  e  $z$ , com o decorrer do tempo de execução. As simulações foram realizadas no *software* Matlab.

Figura 17 – Representação do erro de posição em relação ao tempo.



Fonte: Pham et al. (2018)

Figura 18 – Representação do erro de orientação em relação ao tempo.



Fonte: Adaptado de Pham et al. (2018)

Na Figura 17 e na Figura 18, a linha vermelha representa o *setpoint* de posição e orientação do efetuador em cada eixo cartesiano. A linha azul denota as variáveis em relação ao tempo convergindo para o valor desejado.

Faria et al. (2018) realizaram o estudo cinemático em um robô Kuka de 7 DoF, mapeando suas configurações singulares e propondo um método que altera os ângulos de junta do manipulador para evitar posições que não podem ser resolvidas pela IK. Os autores destacam a relevância de um manipulador com um grau de liberdade além dos seis necessários para cobrir toda a área de trabalho, afirmando que isso oferece vantagem estrutural para desviar de obstáculos e evitar singularidades. Os manipuladores com 7 DoF são assunto de tendência, sendo aplicados em uma ampla gama de campos de pesquisa, explicam os autores.

No trabalho de Faria et al. (2018), não são realizados comparativos entre métodos de cinemática inversa, no entanto, é apresentada uma maneira de coletar dados de desempenho da mesma. Na Tabela 5, são apresentados os dados mensurados no código implementado através do *software* Matlab, destacando os tempos de execução da cinemática inversa, dos intervalos de computação e do algoritmo completo.

Tabela 5 – Tempo computacional para solução de IK pelo Matlab, em  $\mu s$ .

<b>Etapa</b>	<b>Média</b>	<b>Desvio Padrão</b>	<b>Máximo</b>
Cinemática inversa	230,3	21,93	642,0
Intervalos de computação	822,3	88,61	622,0
Algoritmo completo	1.132,0	77,65	3.345,0

Fonte: Adaptado de Faria et al. (2018).

As informações apresentadas na Tabela 5 foram coletadas em um computador com Sistema Operacional Ubuntu 14.04, processador Intel i7-4710MQ a 2.5GHz e 16GB de memória RAM. Os dados foram dispostos em seu valor médio, valor máximo e desvio padrão de 10.000 amostras coletadas, dados em  $\mu s$ .

Em um trabalho recente, Omisore et al. (2018) desenvolveram uma rede neural de aprendizado profundo (*Deep-Learning Neural Network*) para prever o valor mais eficiente da constante de amortecimento  $\lambda$  no método DLS. Em uma rotina desenvolvida com o auxílio do *software* Matlab, os autores compararam a rede neural

proposta com outros 8 métodos de cinemática inversa, em uma estrutura projetada para procedimentos cirúrgicos minimamente invasivos de 8 DoF.

As métricas de desempenho coletadas por Omisore et al. (2018) foram: taxa de sucesso, número de iterações, tempo de execução e iterações por *ms*. Os autores explicam que a taxa de sucesso é o percentual de execuções em que o algoritmo encontrou uma solução para a IK dentro do limite de erro pré-definido, sem exceder a quantidade máxima de iterações.

Omisore et al. (2018) relatam que utilizaram em seus experimentos uma constante de amortecimento  $\lambda = 1,1$  para o método DLS, um erro máximo de posição cartesiana de  $0,1mm$  e um máximo de 500 iterações do algoritmo em cada problema gerado. Os dados de destino do efetuador foram gerados combinando os possíveis valores de ângulos de junta espaçados em  $0,32rad$ , totalizando mais de 1 milhão de pontos que foram utilizados para aprendizado de sua rede neural.

Para a execução dos testes, Omisore et al. (2018) utilizaram um computador com processador Intel i3-3420 a 2.4GHz e os resultados obtidos encontram-se resumidos na Tabela 6, onde os métodos que não são baseados na matriz Jacobiana foram suprimidos. A notação utilizada pelos autores na coluna que expressa o número de iterações não deve ser confundida com uma incerteza de medição, uma vez que a intenção foi utilizar a notação “ $\pm$ ” para representar o valor médio acompanhado de sua variação entre o mínimo e máximo.

Tabela 6 – Métricas de desempenho da IK em robô cirúrgico de 8 DoF.

<b>Método</b>	<b>Sucesso</b>	<b>Nº de iterações</b>	<b>Tempo de execução (<i>ms</i>)</b>	<b>Iterações por <i>ms</i></b>
<i>Deep-learning</i> DLS	91,59%	33,02 $\pm$ 39,60	9,2 $\pm$ 16,92	3,59 $\pm$ 1,62
SDLS	82,53%	75,99 $\pm$ 92,01	50,26 $\pm$ 30,79	1,51 $\pm$ 1,17
DLS	41,89%	115,91 $\pm$ 95,58	43,93 $\pm$ 26,80	2,64 $\pm$ 0,92
$J^T$	0%	500	1.007,36 $\pm$ 13,58	0,49 $\pm$ 0,13

Fonte: Adaptado de Omisore et al. (2018).

Os resultados obtidos no experimento de Omisore et al. (2018) demonstram que as 500 iterações não foram suficientes para a Jacobiana transposta convergir para a solução. Os métodos baseados em mínimos quadrados amortecidos resultaram em métricas superiores às de todas as outras técnicas analisadas.

Quanto à métrica que relaciona uma tolerância ao erro do efetuador, observa-se que nos trabalhos de Buss e Kim (2004) e Omisore et al. (2018) apenas houve controle da posição cartesiana, sem menção relativa à orientação do efetuador. No trabalho de Pham et al. (2018), calculou-se o erro de pose utilizando a notação unificada dos quatérnios duais, controlando tanto a posição quanto a orientação do efetuador. Pham et al. (2018) não atribuíram uma tolerância ao erro para indicar o término da operação, uma vez que seu algoritmo foi executado de forma contínua para manter o erro próximo a zero.

### 3 METODOLOGIA APLICADA

Neste capítulo apresenta-se a metodologia de trabalho a qual visa a implementação e comparação de métodos numéricos Jacobianos para a solução de cinemática inversa em um manipulador robótico Kuka LWR de 7 DoF. Como parte do trabalho, buscou-se a descrição da estrutura do manipulador por meio dos parâmetros DH e a resolução da cinemática direta utilizando a álgebra dos quatérnios duais unitários. Os ensaios foram realizados em um ambiente simulado fazendo uso da plataforma Matlab.

Na sequência do capítulo, descreve-se sobre a composição e estrutura da biblioteca de funções desenvolvida neste trabalho. Em seguida, é apresentada a metodologia utilizada para a geração da base de problemas, os quais foram resolvidos pelos algoritmos de cinemática inversa. Ainda, transcorre-se sobre a explicação do método de cinemática direta e a conversão dos parâmetros DH para o domínio dos quatérnios duais. Após, é realizada a descrição dos métodos de cinemática inversa que foram implementados, bem como os detalhes de sua configuração. Por fim, apresenta-se as métricas de desempenho sobre os resultados obtidos, a fim de obter o comparativo entre os métodos.

#### 3.1 BIBLIOTECA DE FUNÇÕES

Os parâmetros DH do manipulador robótico Kuka LWR (ver Tabela 2) foram empregados em código no *software* Matlab, a fim de possibilitar o manuseio da estrutura do manipulador em ambiente de simulação. Para manter o rastreamento da pose do efetuador, e posterior cálculo de erro, implementou-se a cinemática direta utilizando a álgebra dos QDU.

As operações matemáticas dos quatérnios e quatérnios duais foram programadas com o intuito de criar uma biblioteca dedicada de funções, as quais foram utilizadas no decorrer dos algoritmos repetidas vezes. As funções seguiram a teoria apresentada por Kenwright (2012) e McDonald (2010), descritas nas seções 2.3.2 e 2.3.3, e têm as seguintes nomenclaturas e aplicações:

- a) *NormaQuat*, computa a norma de um quatérnio;
- b) *InvQuat*, computa a inversa de um quatérnio;
- c) *ConjQuat*, computa o conjugado de um quatérnio;

- d) *MultQuat*, computa o produto de dois quatérnios;
- e) *NormaDualQuat*, computa a norma de um quatérnio dual;
- f) *ConjDualQuat*, computa o conjugado de um quatérnio dual;
- g) *MultDualQuat*, computa a multiplicação de dois quatérnios duais;
- h) *DH2DQ*, realiza a conversão de notação dos parâmetros DH para o domínio dos quatérnios duais.

### 3.2 GERAÇÃO DAS POSES INICIAIS E FINAIS

Com intuito de formular uma base de problemas de cinemática inversa grande o suficiente para uma análise estatística adequada, foram geradas 10 mil poses iniciais e 10 mil poses finais de maneira uniformemente aleatória, de acordo com a metodologia utilizada por Pham et al. (2018). Para isso, foi utilizada a função do Matlab “*unifrnd()*”, a qual permite criar uma matriz de números aleatórios de distribuição uniforme e dentro de um intervalo especificado. Sumariamente, 10 mil problemas de cinemática inversa foram resolvidos por cada um dos quatro métodos.

A fim de criar esta base de problemas no Matlab, foram gerados vetores de 10 mil valores uniformemente aleatórios no intervalo aberto à direita  $[0, 2\pi[$ . Cada vetor foi relacionado a uma das sete juntas do manipulador, ou seja, foi formada uma base de 10 mil conjuntos de sete ângulos aleatórios. Cada conjunto foi atribuído aos parâmetros  $\theta_i$  do efetuador. Logo, estas foram as configurações iniciais do manipulador para cada problema de cinemática inversa.

De forma análoga ao método que gerou a configuração inicial do manipulador, foi realizado um procedimento para gerar as 10 mil poses de destino, também conforme metodologia de Pham et al. (2018). Para tal, foram gerados 10 mil conjuntos de sete ângulos aleatórios, e o algoritmo de cinemática direta calculou o QDU da pose do efetuador, formando assim as poses de destino.

Salienta-se que os ângulos utilizados para gerar as poses de destino não foram acessíveis pelo algoritmo de cinemática inversa, uma vez que o objetivo deste é calcular tais ângulos. Ademais, como o problema de IK pode ter múltiplas soluções conforme Aristidou et al. (2017), os ângulos encontrados pelos algoritmos não coincidem obrigatoriamente com os que deram origem à pose. Isso significa que os

ângulos aqui atribuídos apenas tiveram a finalidade de garantir que as poses são factíveis dentro do alcance do manipulador.

### 3.3 ALGORITMO DE CINEMÁTICA DIRETA

Ao atribuir uma configuração inicial, e após cada iteração do algoritmo de IK, a pose do efetuador foi verificada através do cálculo da cinemática direta. Isso permitiu obter a informação de erro em relação à pose de destino a cada iteração. Para isso, os parâmetros DH  $a$ ,  $\alpha$  e  $d$  do manipulador, foram alocados em código e os parâmetros  $\theta$  foram atribuídos conforme configuração inicial gerada.

Os parâmetros DH de cada problema foram convertidos para o domínio dos quatérnios duais unitários através do operador  $QD_{T_i}^{i-1}$ , apresentado por Radavelli et al. (2014) e descrito na subseção 2.3.4. Na equação (62), o operador  $QD_{T_i}^{i-1}$  descreve o QDU que transforma o *frame* da junta  $i - 1$  no *frame* da junta  $i$ .

$$QD_{T_i}^{i-1} = \begin{bmatrix} \cos\left(\frac{\alpha_i}{2}\right) \cos\left(\frac{\theta_i}{2}\right) \\ \sin\left(\frac{\alpha_i}{2}\right) \cos\left(\frac{\theta_i}{2}\right) \\ \sin\left(\frac{\alpha_i}{2}\right) \sin\left(\frac{\theta_i}{2}\right) \\ \cos\left(\frac{\alpha_i}{2}\right) \sin\left(\frac{\theta_i}{2}\right) \\ -\frac{a_i}{2} \sin\left(\frac{\alpha_i}{2}\right) \cos\left(\frac{\theta_i}{2}\right) - \frac{d_i}{2} \cos\left(\frac{\alpha_i}{2}\right) \sin\left(\frac{\theta_i}{2}\right) \\ \frac{a_i}{2} \cos\left(\frac{\alpha_i}{2}\right) \cos\left(\frac{\theta_i}{2}\right) - \frac{d_i}{2} \sin\left(\frac{\alpha_i}{2}\right) \sin\left(\frac{\theta_i}{2}\right) \\ \frac{a_i}{2} \cos\left(\frac{\alpha_i}{2}\right) \sin\left(\frac{\theta_i}{2}\right) + \frac{d_i}{2} \sin\left(\frac{\alpha_i}{2}\right) \cos\left(\frac{\theta_i}{2}\right) \\ -\frac{a_i}{2} \sin\left(\frac{\alpha_i}{2}\right) \sin\left(\frac{\theta_i}{2}\right) + \frac{d_i}{2} \cos\left(\frac{\alpha_i}{2}\right) \cos\left(\frac{\theta_i}{2}\right) \end{bmatrix} \quad (62)$$

A transformação aplicada por  $QD_{T_i}^{i-1}$  foi estendida para as sete juntas do manipulador, por meio da aplicação da equação (63), respeitando a abordagem de Kenwright (2012). Esta transformação resultará no QDU que representa a pose do efetuador,  $Q_{ef}$ .

$$Q_{ef} = Q_0 \otimes Q_1 \otimes Q_2 \otimes Q_3 \otimes Q_4 \otimes Q_5 \otimes Q_6 \otimes Q_7 \quad (63)$$

O erro de posição foi computado através da diferença dos vetores atuais e de destino. O erro de orientação foi computado conforme relação apresentada na subseção 2.4.3, por meio da aplicação da equação (64).

$$q_{rel} = q_d q_{ef}^* \quad (64)$$

Os cálculos de cinemática direta e de erro foram realizados após a execução do bloco de cinemática inversa, com a finalidade de verificar se a condicional de parada relativa à precisão foi cumprida, representando o sucesso do algoritmo em resolver o problema.

### 3.4 ALGORITMOS DE CINEMÁTICA INVERSA

Os algoritmos de cinemática inversa que contemplam o escopo deste trabalho, foram capazes de calcular a cada iteração, um vetor  $\Delta\theta$ , o qual incrementou ou decrementou o valor de cada ângulo de junta até atingir a pose de destino, ou até ser abortado por uma das condicionais de parada descritas na presente sessão. Os métodos propostos utilizaram a matriz Jacobiana, a qual relaciona a taxa de variação do ângulo de cada junta à taxa de variação da pose do efetuador.

A matriz Jacobiana foi computada conforme descrição apresentada por Buss (2009), respeitando as abordagens descritas na subseção 2.4.4 por meio da aplicação da equação (65). Vale observar que o termo  $v_i$  é o vetor unitário que aponta na direção do eixo de rotação da junta  $i$ , sendo que o ponto  $p_i$  está localizado na origem da junta  $i$  e  $s$  é a pose do efetuador no momento da computação. Esses parâmetros são extraídos da equação (63), entre as transformações de um *frame* no seguinte.

$$\frac{\partial s}{\partial \theta_i} = v_i \times (s - p_i) \quad (65)$$

Após a obtenção da matriz Jacobiana, foi implementado o bloco funcional dos métodos de IK. Para fins de comparação de desempenho, esta foi a única parte do código alterada em cada rotina de testes, de forma a isolar a influência de cada método de cinemática inversa. Entende-se por uma “rotina de testes” como a execução dos 10 mil problemas de IK gerados. Cada algoritmo resolveu os mesmos 10 mil problemas.

Como descrito anteriormente, foram implementados quatro métodos diferentes para encontrar os valores de  $\Delta\theta$ , sendo eles: Jacobiana Transposta, Jacobiana Pseudoinversa, DLS e SDLS. Independente do caso, o método de IK teve como objetivo levar o efetuador até a pose de destino nas 10 mil situações distintas geradas. Cada problema teve seu processo encerrado caso alguma das seguintes condicionais de parada tenha ocorrido, atribuídas com base no trabalho de Buss e Kim (2004):

- a) se o efetuador atingiu erro menor que uma tolerância pré-definida, caracterizando sucesso na solução do problema;
- b) se o efetuador se deslocou menos que uma tolerância pré-definida entre uma iteração e outra, caracterizando uma falha, uma vez que o algoritmo parou de progredir na solução do problema ou que a convergência está lenta;
- c) se a distância do efetuador ao alvo aumentou por uma quantidade consecutiva de iterações, caracterizando uma falha devido ao excesso de oscilações;
- d) se uma elevada e determinada quantidade de iterações foi realizada, o que caracteriza excesso de tentativas de convergência sem sucesso.

Como a posição é mensurada em termos de distância cartesiana e a orientação é descrita através de ângulos de eixo, optou-se por representar a grandeza que combina ambas as notações como “unidades de medida”.

Não foram implementadas as limitações nos ângulos das juntas existentes na versão física do manipulador, assim como nos experimentos de Buss e Kim (2004). Desta forma, é possível a coleta das métricas de desempenho dos algoritmos de IK isoladamente, sem a interferência de limitações forçadas. Os algoritmos foram desenvolvidos e executados no *software* Matlab versão R2017b, instalado em um computador com o Sistema Operacional Windows 10 Home, que possui o processador Intel Core i5-6600 a 3,3GHz e a capacidade de 16Gb de memória RAM.

### 3.5 MÉTRICAS DE DESEMPENHO

As métricas de desempenho foram definidas com base nos trabalhos de Buss e Kim (2004), Pham et al. (2018), Faria et al. (2018) e Omisore et al. (2018), sendo escolhidas as seguintes:

- a) taxa de sucesso: consiste no percentual de problemas em que o algoritmo atingiu erro menor que a tolerância pré-definida, sem interrupção de uma das demais condicionais de parada;
- b) quantidade de iterações: consiste na quantidade de iterações que o algoritmo precisou para completar o problema, até o limite máximo permitido;
- c) tempo por iteração: consiste na medição do tempo de execução do bloco funcional que resolve a IK. Os tempos de execução dos problemas em que houve sucesso foram analisados separadamente dos que abortaram;
- d) tempo de processamento: métrica que considera o tempo total que o algoritmo levou para encerrar um problema, considerando todas as etapas. Os tempos de execução dos problemas em que houve sucesso foram analisados separadamente dos que abortaram;
- e) erro ao longo do tempo: consiste em uma representação de forma gráfica do comportamento do erro de pose do efetuador ao longo do tempo.

As medidas temporais necessárias para as métricas de desempenho foram obtidas utilizando as funções “*tic*” e “*toc*”. Esta ferramenta é utilizada para medir o tempo de execução em códigos do Matlab, como em trabalhos recentes que buscam otimização de algoritmos, realizados por Joseph e Hameed (2014), Ghadle e Muley (2015) e Hwang, Mitz e Murray (2019).

As métricas foram armazenadas em variáveis do programa durante a execução dos algoritmos. Todos os períodos de computação foram implementados fora do bloco funcional da cinemática inversa, de modo a não afetar seu tempo de execução, o que comprometeria a análise dos resultados.

A realização de 10 mil testes uniformemente aleatórios visou reduzir variações espúrias nos dados coletados. Tais variações podem ser introduzidas por problemas específicos de IK, os quais podem ser solucionados mais rapidamente por um método que apresenta desempenho inferior em outros casos. Isso vale para a situação recíproca. As imprecisões na função responsável pela medição de tempo também são reduzidas devido à quantidade de testes realizados.

## 4 RESULTADOS

Neste capítulo, transcorrem os resultados obtidos experimentalmente de acordo com metodologia descrita no Capítulo 3. Primeiramente, demonstra-se a definição das constantes utilizadas no decorrer dos experimentos, em seguida apresenta-se resultados expressos graficamente e na forma de tabelas. Os resultados foram analisados em situações distintas, como no caso em que os problemas foram solucionados com êxito individualmente por cada um dos algoritmos, na situação onde um conjunto de problemas foram solucionados por todos os algoritmos, e no caso que houve falhas de execução no decorrer do processo.

### 4.1 CONSTANTES E LIMITES APLICADOS

As constantes apresentadas em literatura, descritos na seção 2.5, foram aplicadas pelos respectivos autores de maneira empírica e relativa à estrutura do manipulador associado a cada estudo. O fato observado foi que tais parâmetros não correspondem aos que fornecem o melhor resultado comparativo para o Kuka LWR. Estruturalmente, o manipulador em questão é diferente da estrutura com múltiplos efetuadores apresentada por Buss e Kim (2004), do manipulador de 6 DoF utilizado por Pham et al. (2018) e do robô cirúrgico de Omisore et al. (2018), portanto foi necessário determinar tais constantes empiricamente.

Em uma avaliação inicial, onde se permitiu que as rotinas executassem até 4.000 iterações, observou-se que tal limite era elevado para a estrutura de 7 DoF, uma vez que todos os métodos resolviam a cinemática inversa exigindo menos de 50 iterações na maioria dos casos, necessitando um máximo aproximado de 500 iterações. O limite foi ajustado para o valor máximo de 500 iterações para se resolver um único problema. O ajuste para um reduzido limite máximo de iterações se justifica uma vez que, devido ao sétimo grau de liberdade presente no Kuka LWR, esta estrutura tem maior mobilidade e alcança o objetivo com menos iterações.

A constante  $\lambda$ , utilizada no método DLS, também foi ajustada para adaptar-se ao manipulador experimentado. Os testes foram realizados variando o valor da constante de modo a reduzir a quantidade média de iterações executadas até a solução dos problemas. Sendo assim, o valor  $\lambda = 0,1$  permitiu ao método DLS a solução dos problemas na menor média de tempo. Observou-se que reduzir ou

aumentar este valor culminava no aumento na quantidade média de iterações e até mesmo em oscilações que levavam o sistema à instabilidade.

No método SDLS, Buss e Kim (2004) atribuem para o manipulador alvo de estudo o valor máximo para  $\Delta\theta$  da junta  $i$  em uma dada iteração como  $\gamma_{m\acute{a}x} = \frac{\pi}{4}$ . Porém, este valor gera resultados de baixo desempenho quando aplicados à estrutura do Kuka LWR. Assim como a constante  $\lambda$ , o valor de  $\gamma_{m\acute{a}x}$  também foi atribuído de modo a minimizar o tempo médio exigido. O valor de  $\gamma_{m\acute{a}x} = \frac{3}{2}\pi$  possibilitou ao método SDLS uma menor média de tempo. Reduzir ou aumentar esta constante causou um desempenho inferior, levando em alguns casos à incapacidade do sistema de resolver a cinemática inversa dentro do limite de iterações.

Por fim, uma vez que a variável que engloba o erro de posição e orientação é um quatérnio dual, fez-se necessário atribuir novos parâmetros para associar tal dado com as condicionais de parada. Para fins experimentais, foi atribuído um erro máximo de posição e orientação como  $\|erro\|_{m\acute{a}x} = 1 + 10^{-6}$  unidades de medida, salientando que  $\|erro\| = 1$  significa que o quatérnio dual desejado é igual ao quatérnio dual relativo. Logo, para a condicional de parada que ocorre quando o algoritmo não está convergindo ao destino, estabeleceu-se o valor  $\|var\|_{m\acute{i}n} = 1 + 10^{-12}$  unidades de medida. E, para a última condicional de parada, a qual indica excesso de oscilações, estipulou-se o valor de 5 iterações consecutivas havendo incremento no erro.

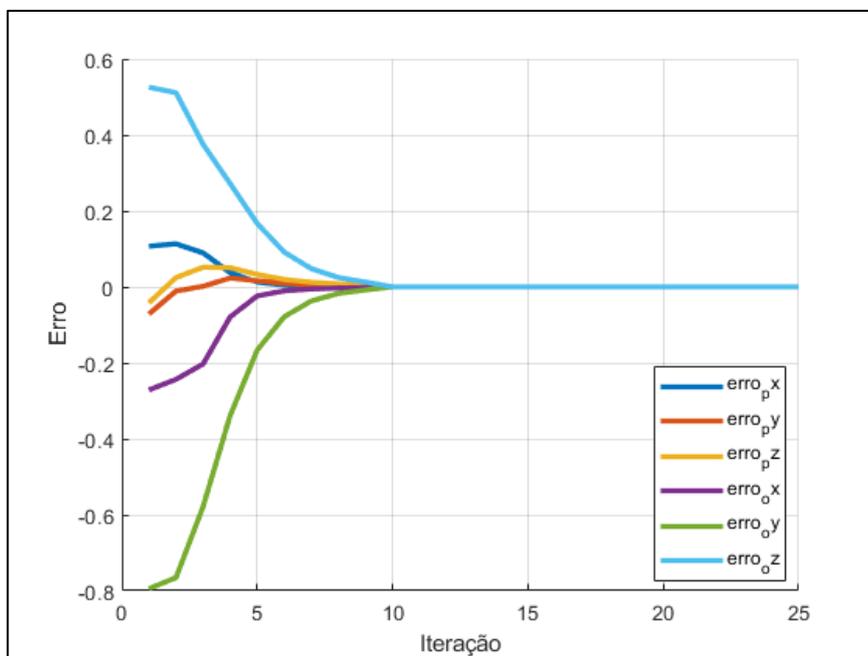
## 4.2 DESEMPENHO DOS ALGORITMOS

Após a execução dos 10 mil problemas de cinemática inversa pelos 4 métodos implementados, os dados de desempenho foram coletados e analisados. O comportamento dos erros de posição e orientação em  $x$ ,  $y$  e  $z$  ao longo da execução dos algoritmos foi observado como forma de validação da correta convergência da pose do efetuador à configuração desejada.

Na Figura 19, observa-se o comportamento do erro de um dos problemas, o qual foi resolvido pelo método SDLS. As seis variáveis de erro tendem a convergir para zero simultaneamente, podendo haver oscilações no decorrer da execução. Para fins de visualização gráfica, optou-se por apresentar o resultado de apenas um dos

problemas, porém, o comportamento de convergência simultânea foi observado de maneira similar nos demais casos dos quatro métodos.

Figura 19 – Comportamento do erro ao longo do tempo.



Fonte: O autor

Com a finalidade de comparar a eficiência de cada algoritmo na solução dos problemas, construiu-se a Tabela 7, na qual consta a taxa de sucesso de cada algoritmo, sendo que este dado representa a quantidade de problemas que cada método encerrou dentro da condição de  $||erro|| \leq ||erro||_{máx}$ . Para fins de esclarecimento da notação, a Jacobiana Transposta é representada pelo símbolo  $J^T$  e a Jacobiana Pseudoinversa é representada pelo símbolo  $J^\dagger$ .

Tabela 7 – Taxa de sucesso dos algoritmos.

Método	Casos de Sucesso	%
$J^T$	7640	76,4%
$J^\dagger$	9787	97,9%
DLS	9747	97,5%
SDLS	9812	98,1%

Fonte: O Autor.

Observa-se que o método SDLS tem uma maior taxa de sucesso em relação aos demais, sendo que  $J^T$  apresentou performance significativamente menor,

frequentemente abortando a execução do algoritmo devido a uma das condicionais de parada.

Nas Tabelas 8, 9, 10 e 11 são apresentados os valores mínimos, máximos, médios e de desvio padrão da seguinte lista de parâmetros para cada método avaliado: quantidade de iterações necessárias, tempo de resolução do problema, tempo transcorrido durante uma iteração e tempo transcorrido durante o bloco de cinemática inversa de cada iteração. Vale observar que os dados representam os problemas onde o algoritmo encerrou em condição de sucesso, e que os tempos de iteração e de cinemática inversa foram expressos em microssegundos ( $\mu s$ ) e o tempo de solução dos problemas foi expresso em milissegundos ( $ms$ ).

Tabela 8 – Desempenho do método Jacobiana Transposta.

<b>Dado</b>	<b>Iterações</b>	<b>Tempo IK (<math>\mu s</math>)</b>	<b>Tempo Iteração (<math>\mu s</math>)</b>	<b>Tempo Problema (<math>ms</math>)</b>
Mínimo	7	0,169	3,923	3,621
Máximo	485	0,852	9,720	438,517
Médio	71	0,188	4,252	34,016
Desvio padrão	51	0,212	0,659	24,645

Fonte: O Autor.

Tabela 9 – Desempenho do método Jacobiana Pseudoinversa.

<b>Dado</b>	<b>Iterações</b>	<b>Tempo IK (<math>\mu s</math>)</b>	<b>Tempo Iteração (<math>\mu s</math>)</b>	<b>Tempo Problema (<math>ms</math>)</b>
Mínimo	3	0,177	3,961	2,002
Máximo	145	1,276	10,417	69,654
Médio	16	0,201	4,276	8,204
Desvio padrão	8	1,068	1,174	3,937

Fonte: O Autor.

Tabela 10 – Desempenho do método DLS.

<b>Dado</b>	<b>Iterações</b>	<b>Tempo IK (<math>\mu s</math>)</b>	<b>Tempo Iteração (<math>\mu s</math>)</b>	<b>Tempo Problema (<math>ms</math>)</b>
Mínimo	4	0,190	3,972	2,233
Máximo	435	1,714	9,693	201,625
Médio	14	0,218	4,317	7,339
Desvio padrão	16	0,109	0,511	7,862

Fonte: O Autor.

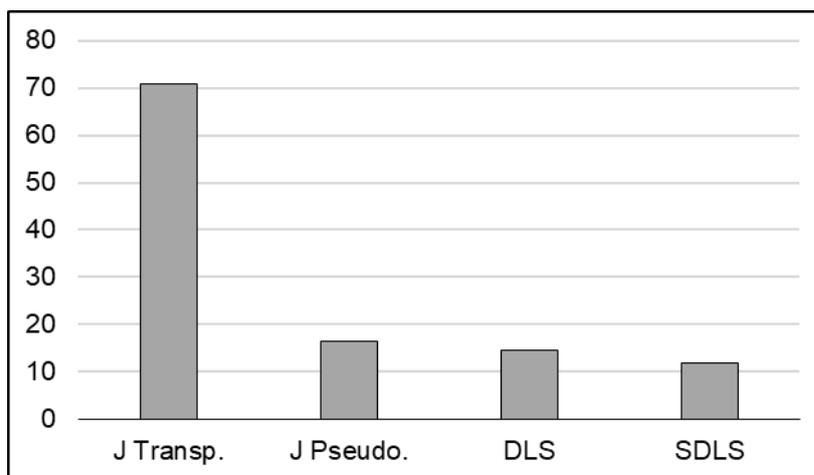
Tabela 11 – Desempenho do método SDLS.

Dado	Iterações	Tempo IK ( $\mu s$ )	Tempo Iteração ( $\mu s$ )	Tempo Problema ( $ms$ )
Mínimo	3	0,643	4,438	1,854
Máximo	349	2,476	12,535	182,903
Médio	12	0,708	4,790	6,603
Desvio padrão	7	0,595	0,775	3,718

Fonte: O Autor.

Nas figuras 20, 21 e 22, apresenta-se o comparativo entre as características de tempo de execução, tempo de cinemática inversa e iterações necessárias de cada método. Tal demonstração objetiva analisar o comportamento de cada algoritmo durante sua execução e como isto influenciou no tempo de resposta para solução de cada problema. A quantidade média de iterações exigida pelos algoritmos é apresentada na Figura 20.

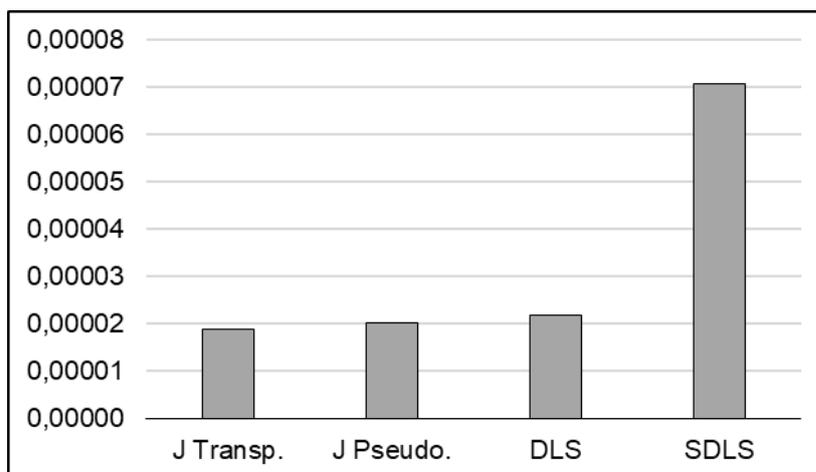
Figura 20 – Quantidade média de iterações exigidas em cada método.



Fonte: O autor

Na Figura 21 visualiza-se o tempo médio, em segundos, para a computação da cinemática inversa.

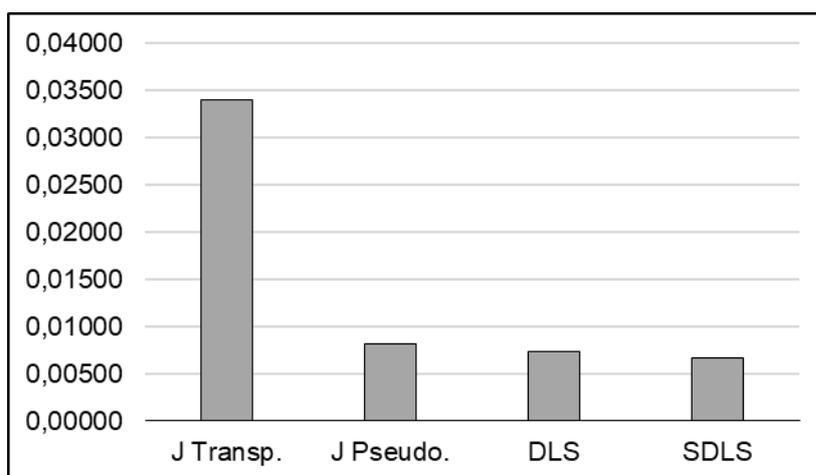
Figura 21 – Tempo médio para computação da IK exigido em cada método.



Fonte: O autor

Na Figura 22, descreve-se o tempo médio, em segundos, para a solução de um problema.

Figura 22 – Tempo médio para solução de um problema exigido em cada método.



Fonte: O autor

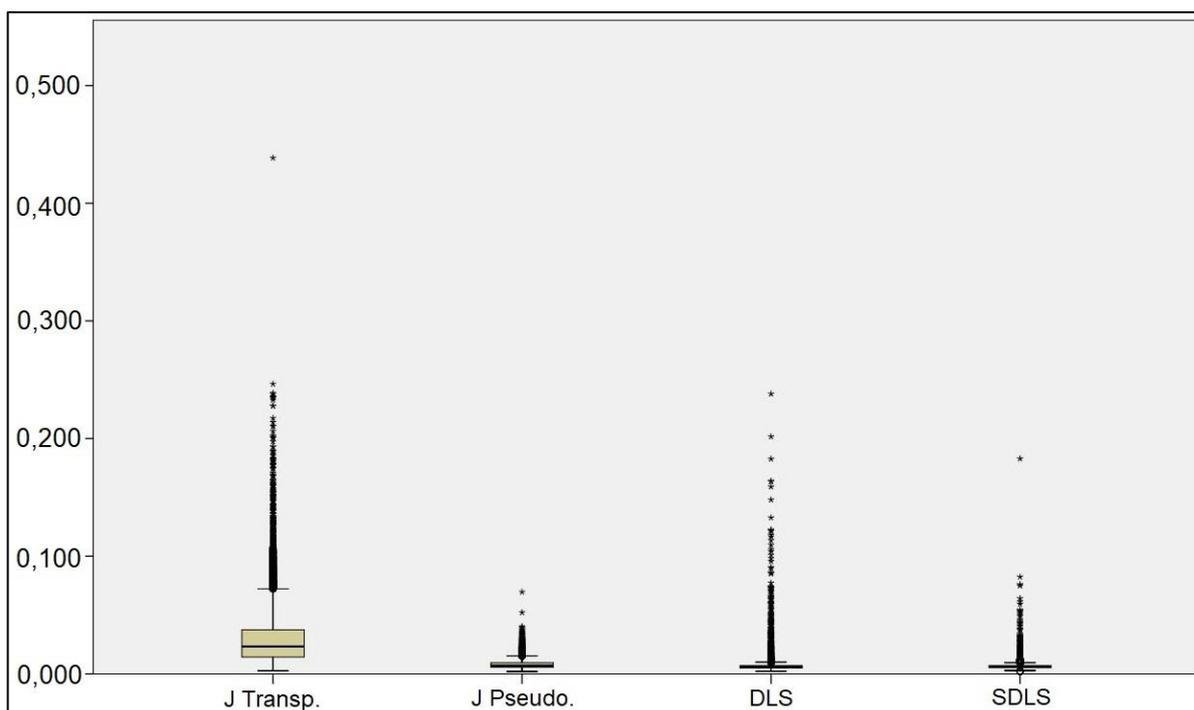
Observa-se na Figura 21 que o método SDLS exige maior tempo de computação para a cinemática inversa, uma vez que é necessário calcular a SVD da matriz Jacobiana e obter os ângulos relativos a cada vetor singular. Em contrapartida, de acordo com a Figura 20, o método exige uma quantidade média menor de iterações para solucionar dado problema, fazendo com que este algoritmo tenha apresentado a menor média de tempo para solução dos problemas, conforme dados apresentados na Figura 22.

O método da Jacobiana Transposta apresentou o pior desempenho médio. O algoritmo exigiu uma quantidade de iterações maior que os demais e, mesmo tendo

apresentado a computação da cinemática inversa mais rápida, foi o método que teve a maior média de tempo de solução de um problema.

Com o objetivo de visualizar os resultados de maneira detalhada, a Figura 23 apresenta a distribuição do tempo de solução de cada um dos quatro métodos analisados. As amostras representadas como asteriscos (\*) na Figura 23 representam os 5% dos problemas de desempenho discrepante em relação ao valor médio.

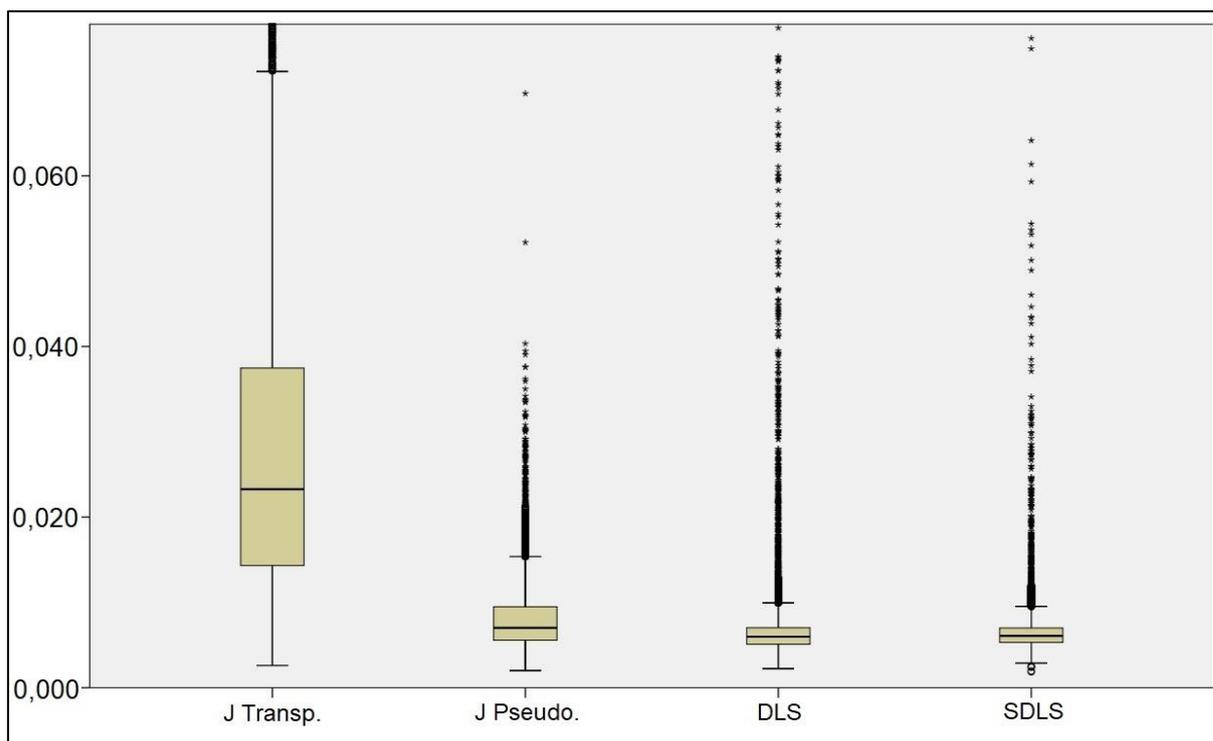
Figura 23 – Distribuição do tempo de solução dos problemas em cada método.



Fonte: O autor

Devido à dificuldade de analisar a informação na proximidade da média das amostras na Figura 23, optou-se por representar os dados em escala na Figura 24. As amostras compreendidas entre o primeiro e o terceiro quartil da Figura 24 englobam 50% dos valores mais próximos à linha mediana, sendo que os limites superior e inferior do diagrama de caixa contemplam 95% das amostras. Os gráficos foram gerados através do *software* IBM SPSS.

Figura 24 – Distribuição do tempo de solução dos problemas, em escala.



Fonte: O autor

De acordo com a média de tempo de solução dos problemas apresentada na Figura 22, o método SDLS possui o melhor desempenho, porém a Figura 24 mostra que a performance da Jacobiana Pseudoinversa e do DLS são semelhantes considerando-se a distribuição dos valores amostrados. De modo a visualizar numericamente a dispersão dos dados coletados, a Tabela 12 contém os coeficientes de variação para o tempo de execução de um problema em cada método, calculados pela razão entre o desvio padrão e a média dos dados.

Tabela 12 – Coeficientes de variação para o tempo de execução em cada método.

Método	Coeficiente de variação
$J^T$	72,45%
$J^\dagger$	47,98%
DLS	107,12%
SDLS	56,30%

Fonte: O Autor.

Os dados apresentados na Tabela 12 são parâmetros para a variabilidade do tempo de execução em cada método. Quanto maior o coeficiente de variação, maior é a possibilidade de obter dados variados entre uma solução rápida e uma solução lenta para um problema de cinemática inversa. O maior coeficiente apresentado é

para o método DLS, o que significa que este é o método que mais variou em relação ao seu valor médio.

Uma vez que os dados descritos no decorrer do presente capítulo possuem comportamento semelhante, excetuando-se o comportamento da Jacobiana Trasposta, fez-se necessária a análise dos problemas de cinemática inversa onde os quatro métodos obtiveram sucesso simultaneamente. Na Tabela 13 descreve-se a contagem de problemas em que cada método obteve o melhor desempenho de tempo em relação aos demais.

Tabela 13 – Comparativo de menores tempos de solução entre os métodos.

<b>Método</b>	<b>Casos de menor tempo</b>	<b>%</b>
$J^T$	2	0,03%
$J^\dagger$	1.708	23,33%
DLS	3.222	44,00%
SDLS	2.390	32,64%

Fonte: O Autor.

Dentre os casos em que os quatro algoritmos obtiveram sucesso, descritos na Tabela 13, observa-se que o método DLS solucionou o problema com o menor tempo em 44,00% das amostras, enquanto a Jacobiana Pseudoinversa obteve o melhor resultado em 23,33% dos casos. Destaca-se que o método SDLS teve o melhor resultado em apenas aproximadamente 1/3 dos problemas, apesar de seu tempo médio ser o menor. Tais resultados são consequência da grande variação dos tempos de execução entre os problemas de cada algoritmo, descritos na Figura 24 e na Tabela 12.

Os dados de desempenho coletados sugerem que o sistema pode ser otimizado caso seja possível ter o conhecimento *a priori* de qual é o método mais rápido na solução de dado problema. Tal fato é observado uma vez que o algoritmo cujo tempo de solução foi o menor em cada caso não é necessariamente o que tem o menor tempo na média das amostras.

O erro inicial de cada problema é conhecido previamente, portanto realizou-se um experimento com o objetivo de verificar a possibilidade de determinar uma relação entre o tempo de execução do problema e a norma do erro inicial, onde foi calculada a correlação entre estas variáveis. A correlação foi calculada através do *software* IBM SPSS, onde foram considerados os tempos dos problemas resolvidos com sucesso

por cada método em relação à norma do erro inicial. Os resultados estão presentes na Tabela 14.

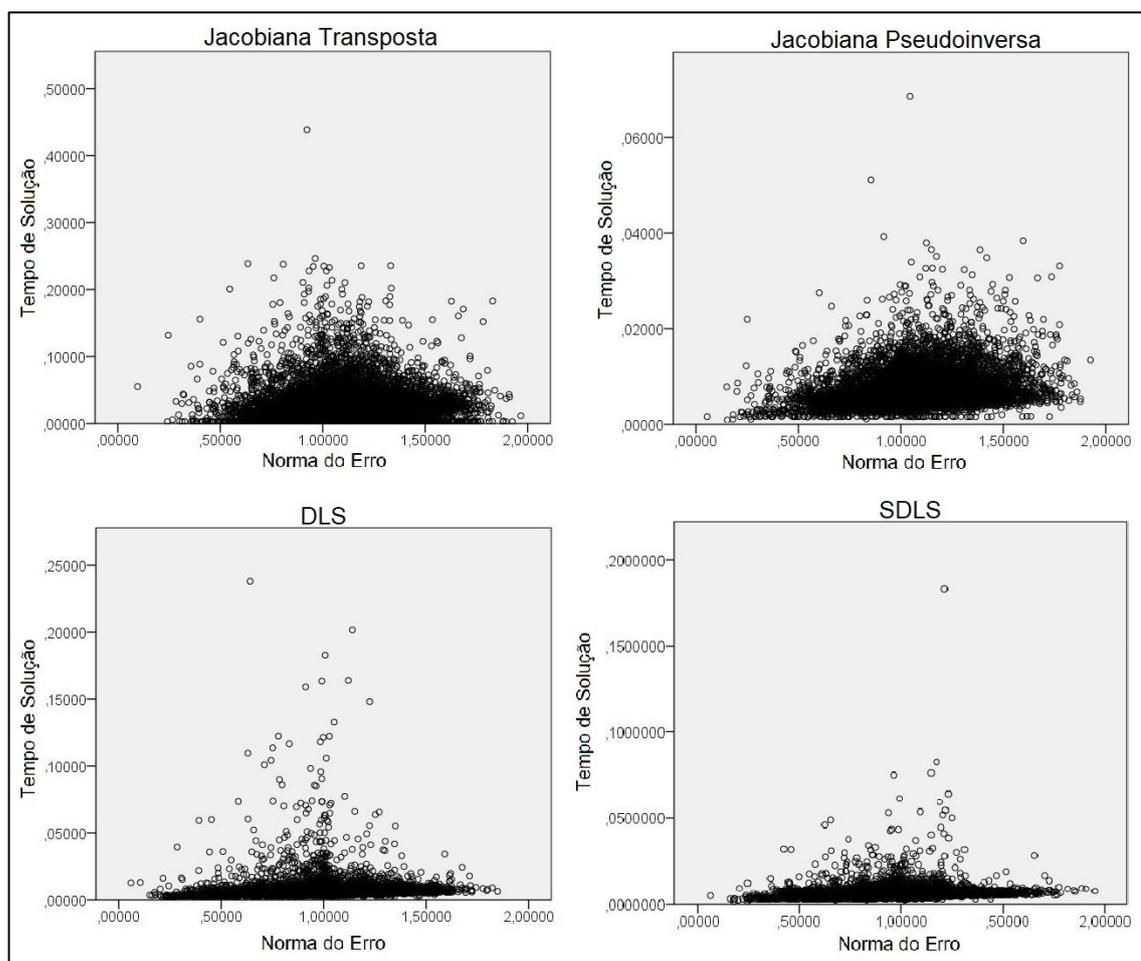
Tabela 14 – Correlação entre o tempo de execução do problema e o erro inicial.

Método	Correlação Tempo x Erro Inicial
$J^T$	-0,014
$J^\dagger$	0,255
DLS	0,023
SDLS	0,063

Fonte: O Autor.

Os dados descritos na Tabela 14 demonstram que não há uma correlação significativa entre o erro inicial e o tempo de execução em nenhum dos métodos analisados. A baixa correlação pode ser observada na Figura 25, onde são apresentados os gráficos de dispersão entre o tempo de solução de um problema em relação à norma do erro inicial.

Figura 25 – Tempo de execução versus norma do erro inicial.



Fonte: O autor

Com base na baixa correlação entre o erro inicial e o tempo de solução, não foi possível definir uma relação que indique previamente qual é o método mais eficiente na solução de dado problema. A norma do erro inicial foi utilizada para esta análise pois é a informação conhecida *a priori* que relaciona o estado atual do manipulador e a pose de destino. Relacionar informações conhecidas a indicativos de qual é o método mais eficiente em cada problema é tema de projetos de pesquisa futuros.

Os tempos de execução tendem a ser menores quando o erro também é menor, ou seja, quando a norma do quaternio dual de erro é próxima a 1, sendo este o motivo para a maior concentração de amostras próximas a *Norma do Erro* = 1. Nos gráficos da Figura 25 também é possível observar o comportamento espúrio de uma pequena quantidade de amostras, o qual pode ter sido causado por processos paralelos executados pelo Sistema Operacional no decorrer da rotina de testes. O conjunto de 10 mil amostras visou minimizar a influência de tal adversidade na base de dados.

### 4.3 ANÁLISE DAS AMOSTRAS COM FALHA NA RESOLUÇÃO

A análise das amostras que abortaram devido a uma das condicionais de parada consistiu em contabilizar a quantidade de problemas de cada método em tal condição. Padronizou-se a identificação com o algarismo “1” para a condição de sucesso, e as demais condições como:

- a) “2” caso a variação da pose do efetuador entre uma iteração e outra tenha sido menor que  $||var||_{min} = 1 + 10^{-12}$ ;
- b) “3” caso o erro da pose do efetuador tenha aumentado em 5 iterações consecutivas, caracterizando oscilação;
- c) “4” caso o algoritmo tenha atingido o máximo de 500 iterações.

Consta na Tabela 15 a contagem de amostras abortadas devido a cada condicional entre os métodos implementados. Devido à elevada contagem de paradas pela causa “3” na Jacobiana Transposta em relação à contagem dos demais métodos, evidencia-se os problemas relacionados na literatura quanto à dificuldade de convergência do algoritmo. Esta informação agrega-se à baixa taxa de sucesso em comparação aos demais métodos, apresentada na Tabela 7.

Tabela 15 – Contabilização das causas de parada por falha.

<b>Método</b>	<b>Causa “2”</b>	<b>Causa “3”</b>	<b>Causa “4”</b>
$J^T$	186	2167	7
$J^\dagger$	0	213	0
DLS	14	238	1
SDLS	1	187	0

Fonte: O Autor.

O objetivo de abortar o algoritmo quando uma das condicionais pré-estabelecidas ocorre é indicar que a solução do problema pode não ser viável em determinadas situações práticas. Conforme o exemplo de aplicação em animação por computação gráfica de Buss e Kim (2004), oscilações durante a solução da cinemática inversa podem produzir um efeito trêmulo, a depender da taxa de atualização dos *frames*, e a baixa taxa de convergência pode fazer com que a estrutura não seja capaz de seguir a trajetória na velocidade estabelecida.

Com a finalidade de verificar o desempenho dos métodos caso a aplicação em questão tolere oscilações ou baixas taxas de convergência, realizou-se um experimento nas amostras onde houve a interrupção da execução. Para tal, estes problemas foram executados novamente, porém ignorando as condições de parada “2” e “3” e permitindo até 4.000 iterações.

Os quatro algoritmos foram capazes de resolver a cinemática inversa em todos os problemas em que previamente haviam acusado falha. Para comparar os resultados obtidos neste experimento com os resultados dos testes que consideraram as condicionais de parada, descreve-se na Tabela 16 a quantidade média de iterações adicionais que levaram à solução dos referidos problemas.

Tabela 16 – Iterações adicionais médias para amostras marcadas como falha.

<b>Método</b>	<b>Iterações adicionais</b>
$J^T$	108
$J^\dagger$	14
DLS	62
SDLS	19

Fonte: O Autor.

As iterações adicionais contabilizam as execuções do bloco de cinemática inversa realizadas entre o momento em que haveria ocorrido uma condicional de

parada e a solução completa do problema. Os dados apresentados na Tabela 16 representam a média destas iterações para cada método implementado.

Os resultados descritos na Tabela 16 mostram que os quatro métodos foram capazes de resolver problemas de cinemática inversa em todas as configurações aleatórias apresentadas, sendo limitados apenas por condicionais relativas à aplicação, não apresentando comportamento divergente em nenhum dos 10 mil casos. A determinação *a priori* do potencial de oscilação e de problemas relativos à taxa convergência em um determinado problema de cinemática inversa é objeto de estudos futuros.

## 5 DISCUSSÕES E CONCLUSÕES

No desenvolvimento deste trabalho, observou-se em primeiro momento o desempenho da álgebra dos quatérnios duais unitários na descrição unívoca e minimalista de corpos rígidos no espaço cartesiano. Os conceitos destas operações algébricas foram implementados em bibliotecas dedicadas do Matlab e aplicados na descrição cinemática de um manipulador robótico de 7 DoF, o Kuka LWR. A escolha desta estrutura de manipulador teve a finalidade de demonstrar os conceitos cinemáticos abordados em uma estrutura de juntas praticável, respeitando as limitações de projeto definidas.

Os quatérnios duais unitários foram aplicados na etapa de cinemática direta do manipulador. O uso desta ferramenta permitiu definir a posição e orientação do efetuador a cada iteração dos algoritmos utilizando uma quantidade reduzida de operações algébricas, tornando toda a execução dos problemas mais rápida em relação a técnicas que empregam matrizes de transformação homogênea, conforme documentado por Funda e Paul (1990), Aspragathos e Dimitros (1998) e Kenwright (2012).

O objetivo geral deste trabalho foi realizar o comparativo de quatro métodos numéricos de solução para a cinemática inversa do manipulador Kuka LWR. Foram escolhidos quatro métodos de acordo com a literatura analisada: Jacobiana Transposta, Jacobiana Pseudoinversa, DLS e SDLS. Estes diferenciam-se pela maneira como realizam a inversão da matriz Jacobiana, a qual é responsável por descrever a relação entre a taxa de variação dos ângulos de cada junta e a taxa de variação da pose do efetuador.

Na implementação da matriz Jacobiana, observou-se que a notação de orientação dada pelos quatérnios foi fundamental para definir o vetor de erro, o qual contém seis termos, o que torna a dimensão das matrizes compatível com as equações teóricas. A relevância desta notação foi constatada ao observar que, em documentos relatando o uso de métodos Jacobianos sem a aplicação de quatérnios duais, como em Buss e Kim (2004), apenas há o controle da posição do efetuador. A álgebra dos quatérnios permite a construção de uma variável de erro que leva ao controle de posição e orientação do efetuador simultaneamente, como descrito em Pham et al. (2018).

Os métodos de solução para a cinemática inversa foram implementados e avaliados utilizando uma base de problemas uniformemente aleatória. A aleatoriedade na definição da pose inicial e final dos problemas visou criar situações que englobam diversos cenários sem preferência por configurações que possam ressaltar as características de um método em detrimento de outro.

A base de problemas incluiu 10 mil amostras compostas por ângulos de junta de 0 a  $2\pi$  e apenas uma configuração de destino para cada caso. Esta configuração permitiu coletar dados de cada método sem inserir condições forçadas nas articulações do manipulador e sem a necessidade de estabelecer uma trajetória específica, permitindo o algoritmo a levar o manipulador da pose de origem até o destino sem limitações que afetem seu desempenho.

No decorrer dos experimentos, houve a necessidade de estabelecer constantes para a execução dos algoritmos. A constante de amortecimento aplicada no método DLS é diferente para cada estrutura robótica e depende da situação de cinemática inversa em questão. Para a execução dos testes foi estabelecido previamente o valor de  $\lambda = 0,1$ , uma vez que este valor permitiu ao DLS a menor média de tempos de solução para os problemas apresentados. O mesmo método foi utilizado para definir a o valor  $\gamma_{m\acute{a}x} = \frac{3}{2}\pi$  no método SDLS.

As constantes relacionadas às condicionais de parada dos algoritmos foram definidas com a finalidade de adaptar seus valores à aplicação dos quatérnios duais unitários. As constantes de erro neste trabalho não representam distâncias espaciais absolutas, mas sim uma combinação das notações de diferença de distância e diferença de orientação. Foi definido  $\|erro\|_{m\acute{a}x} = 1 + 10^{-6}$  unidades de medida para a norma máxima do quatérnio dual de erro para configurar sucesso na solução de um problema. Também foi definido  $\|var\|_{m\acute{i}n} = 1 + 10^{-12}$  unidades de medida para a mínima variação de pose aceita entre uma iteração e outra. Tais valores são empíricos e foram obtidos com base nos dados descritos em literatura e adaptados para a realidade do presente trabalho.

Ademais, estabeleceu-se a quantidade máxima de 500 iterações por método. A limitação no número de iterações não trouxe informações de desempenho relevantes, uma vez que cada método tem tempos de execução diferentes em cada

iteração. Este limite tem a finalidade de evitar laços infinitos nos algoritmos e a coleta excessiva de dados.

Os resultados obtidos experimentalmente ressaltam a não-linearidade da cinemática inversa. Não foi possível afirmar qual dentre os métodos implementados é o mais eficiente em termos de velocidade na solução da IK. No entanto, é evidente o desempenho inferior da Jacobiana Transposta em relação aos demais, a qual pôde resolver 76,4% dos problemas dentro das condições estabelecidas, enquanto a Jacobiana Pseudoinversa resolveu 97,9%, o DLS 97,5% e o SDLS 98,1% dos problemas. A conclusão é que a Jacobiana Transposta é uma aproximação de baixo desempenho que introduz oscilação demasiada no sistema e sua rápida computação não compensa a baixa taxa de convergência.

O desempenho mensurado para a Jacobiana Pseudoinversa, DLS e SDLS é semelhante, exigindo uma análise da distribuição das amostras para avaliar a diferença entre suas métricas. O SDLS apresentou a maior taxa de sucesso na resolução de problemas dentro das condições de parada, com 98,1% de problemas resolvidos, e a menor média de tempo de solução, sendo essa de 6,603 *ms*. Apesar dos melhores resultados médios, o SDLS foi o método mais rápido em 32,64% dos casos, enquanto o DLS foi o mais rápido em 44,00% e a Jacobiana Pseudoinversa em 23,33% das amostras, o que mostra que o melhor algoritmo depende da situação em questão.

Os dados coletados mostraram que não há um método estabelecido como mais eficiente que os demais em termos de tempo de solução. Isto é observado analisando o alto coeficiente de variação das amostras de cada método, conforme descrito na Tabela 12. Em casos onde o efetuador encontra-se próximo à pose de destino, os algoritmos exigem menor tempo para encontrar a solução, porém este tempo não aumenta proporcionalmente ao erro, apresentando comportamento não-linear, de acordo com dados coletados e expostos na Figura 25.

Esta situação instigou a realização de um experimento objetivando uma relação entre dados conhecidos e o desempenho de cada algoritmo. Para isso foi calculada a correlação entre o erro inicial de cada problema de cinemática inversa com o tempo de solução apresentado. Não foram encontrados valores de correlação que permitam computar previamente o método que solucionará dado problema no menor tempo,

sendo este um possível tema de projetos futuros envolvendo técnicas de aprendizado de máquina, por exemplo.

Os casos onde os algoritmos abortaram devido a uma das condicionais de parada foram executados novamente em um experimento onde estas condições foram desabilitadas. Este cenário permitiu analisar o desempenho de cada método em situações onde a oscilação e a baixa taxa de convergência não são fatores críticos na escolha do método empregado. Todos os métodos foram capazes de solucionar a cinemática inversa nos casos previamente marcados como falha, o que indica que todos convergem a uma resposta para a estrutura do Kuka LWR. Uma análise recomendada para trabalhos futuros seria considerar as limitações de junta do modelo real e avaliar os diferentes resultados neste quesito.

Em conclusão, houve êxito na análise comparativa entre os métodos de solução para cinemática inversa. Com exceção da Jacobiana Transposta, devido ao seu baixo desempenho, os algoritmos tiveram desempenho semelhante solucionando praticamente todos os problemas propostos dentro dos limites de oscilação e taxa de convergência estabelecidos. O fator que diferenciou os métodos é inerente à não-linearidade dos problemas de cinemática inversa, sendo cabível a análise futura de sistemas inteligentes de otimização. Também não foram detectados problemas de divergência envolvendo configurações singulares, possivelmente devido à mobilidade fornecida pelos sete graus de liberdade da estrutura adotada.

Espera-se continuidade neste trabalho em análises que englobem as limitações de junta, velocidade e aceleração, trajetória, fatores dinâmicos como torques e imperfeições mecânicas, estruturas com múltiplos efetuadores trabalhando em conjunto e demais aplicações que envolvem fatores subjetivos, como o realismo na simulação dos movimentos de seres vivos. Os conceitos analisados neste trabalho podem ser aplicados a qualquer estrutura robótica serial, bastando alterar os parâmetros DH e realizando o ajuste das constantes para melhor desempenho, o que possibilita o desenvolvimento dos estudos em robótica de maneira geral.

## REFERÊNCIAS

- ARISTIDOU, A. et al. **Inverse kinematics techniques in computer graphics: a survey**. Computer graphics forum. Helsinki: John Wiley & Sons Ltd. 2017. p. 1-24.
- ASPRAGATHOS, N. A.; DIMITROS, J. K. A comparative study of three methods for robot kinematics. **IEEE transactions on systems, man, and cybernetics**, 2 abr. 1998.
- BUSS, S. R. **Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods**. University of California. San Diego. 2009.
- BUSS, S. R.; KIM, J. S. **Selectively damped least squares for inverse kinematics**. University of California. San Diego. 2004.
- CHEN, S.; LI, Y.; KWOK, N. M. Active vision in robotic systems: A survey of recent developments. **The international journal of robotics research**, n. 11, Agosto 2011.
- CLIFFORD, W. K. A preliminary sketch of biquaternions. **Proceedings of the London mathematical society**, Londres, 1873.
- CLIFFORD, W. K. **Mathematical papers**. Londres: Macmillan and Co., 1882.
- COPPELIA ROBOTICS. **Notes on the Kuka LWR4 dynamic model**, 2019. Disponível em: <<http://www.coppeliarobotics.com/>>. Acesso em: 18 maio 2019.
- CRAIG, J. J. **Robótica**. São Paulo: Pearson Education do Brasil, 2012.
- DENAIVIT, J.; HARTENBERG, R. S. **A kinematic notation for lower-pair mechanisms based on matrices**. ASME applied mechanics division. New York. 1953.
- ENGARDT, M.; HEIMBÜRGER, A.; SYDHOFF, P. **Manipulability index optimization for a planar robotic arm**. Royal Institute of Technology. Estocolmo. 2012.
- FAMILTON, J. C. **Quaternions: a history of complex noncommutative rotation groups in theoretical physics**. Columbia University. New York City. 2015.
- FARIA, C. et al. Position-based kinematics for 7-DoF serial manipulators with global configuration control, joint limit and singularity avoidance. **Mechanism and machine theory**, v. 121, p. 317-334, mar. 2018.
- FUNDA, J.; PAUL, R. P. A Computational Analysis of Screw Transformations In Robotics. **IEEE transactions on robotics and automation**, Filadélfia, v. 6, p. 348 - 356, jun. 1990.

GE, K. T. **Solving inverse kinematics constraint problems for highly articulated models**. University of Waterloo. Waterloo. 2000.

GHADLE, K. P.; MULEY, Y. M. Travelling salesman problem with MATLAB programming. **International journal of advances in applied mathematics and mechanics**, p. 258-266, 2015.

HAMILTON, W. R. On quaternions, or on a new system of imaginaries in algebra. **Philosophical magazine and journal of science**, Londres, Edinburgo e Dublin, v. 25, p. 489-495, 1844.

HAMILTON, W. R. **Elements of quaternions**. Londres: Longmans, Green, & Company, 1866.

HWANG, J.; MITZ, A. R.; MURRAY, E. A. NIMH MonkeyLogic: Behavioral control and data acquisition in MATLAB. **Journal of neuroscience methods**, v. 323, p. 13-21, 2019.

INTERNATIONAL FEDERATION OF ROBOTICS. IFR. **WR 2018 industrial robots executive summary**, 2018. Disponível em: <<https://ifr.org/>>. Acesso em: 04 jun. 2019.

JOSEPH, S. M.; HAMEED, A. **Online handwritten malayalam character recognition using LIBSVM in Matlab**. 2014 IEEE national conference on communication, signal processing and networking (NCCSN). Palakkad, India: IEEE. 2014.

KENWRIGHT, B. **A beginners guide to dual-quaternions: what they are, how they work, and how to use them for 3D character hierarchies**. School of Computing Science, Newcastle University. Newcastle. 2012.

KENWRIGHT, B. **Real-time character inverse kinematics using the Gauss-Seidel iterative approximation method**. The fourth international conference on creative content technologies. Nice, França: IARIA. 2012. p. 63-68.

KENWRIGHT, B. Inverse kinematics with dual-quaternions, exponential-maps, and joint limits. **International journal on advances in intelligent systems**, Newcastle, v. 6, 2013.

KHATIB, M.; KHUDIR, K. A.; LUCA, A. D. **Visual coordination task for human-robot collaboration**. 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). Vancouver: IEEE. 2017. p. 3762-3768.

KUKA INDUSTRIAL ROBOTS, 2019. Disponível em: <[https://www.kuka.com/-/media/kuka-downloads/imported/48ec812b1b2947898ac2598aff70abc0/spez\\_lbr\\_iiwa\\_en.pdf](https://www.kuka.com/-/media/kuka-downloads/imported/48ec812b1b2947898ac2598aff70abc0/spez_lbr_iiwa_en.pdf)>. Acesso em: 12 maio 2019.

LYNCH, K. M.; PARK, F. C. **Modern robotics mechanics, planning, and control**. Cambridge: Cambridge University Press, 2017.

MARINHO, M. M.; FIGUEREDO, L. F. C.; ADORNO, B. V. **A dual quaternion linear-quadratic optimal controller for trajectory tracking**. 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). Hamburgo, Alemanha: IEEE. 2015. p. 4047-4052.

MAYO, R. A. Relative quaternion state transition relation. **Journal of Guidance, Control, and Dynamics**, v. 2, n. 1, p. 44-48, 1979.

MCDONALD, J. Teaching quaternions is not complex. **Computer graphics forum**, Chicago, v. 29, p. 2447-2455, 2010.

NAKAMURA, Y.; HANAFUSA, H. Inverse kinematic solutions with singularity robustness for robot manipulator control. **Journal of dynamic systems, measurement, and control**, v. 108, p. 163-171, set. 1986.

OMISORE, O. M. et al. Deeply-learnt damped least-squares (DL-DLS) method for inverse kinematics of snake-like robots. **Neural networks**, 2018.

ÖZGÜRA, E.; MEZOUARB, Y. Kinematic modeling and control of a robot arm using unit dual quaternions. **Robotics and autonomous systems**, v. 77, p. 66-73, set. 2015.

PHAM, H. L. et al. Set-point control of robot end-effector pose using dual quaternion feedback. **Robotics and computer-integrated manufacturing**, v. 52, p. 100-110, 2018.

RADAVELLI, L. A. et al. **A screw dual quaternion operator for serial robot kinematics**. 14th pan-american congress of applied mechanics. Santiago: [s.n.]. 2014.

RIGANO, G. F. et al. **Towards a robot hardware abstraction layer (R-HAL) leveraging the XBot software framework**. 2018 second IEEE international conference on robotic computing. Laguna Hills: IEEE. 2018. p. 175-176.

SHARKAWY, A. N.; ASPRAGATHOS, N. **A comparative study of two methods for forward kinematics and Jacobian matrix determination**. International conference on mechanical, system and control engineering. St. Petersburg: IEEE. 2017. p. 179-183.

SHYAMANTA M. HAZARIKA, U. S. D. Robotics: history, trends, and future directions. In: SHYAMANTA M. HAZARIKA, U. S. D. **Materials forming, machining and tribology**. Suíça: Springer International Publishing, 2018. p. 213-239.

THE MATHWORKS INC. **Site oficial do Matlab**, 2019. Disponível em: <<https://www.mathworks.com/products/matlab.html>>. Acesso em: 09 Junho 2019.

THE MATHWORKS INC. Technical Articles and Newsletters. **Improvements to tic and toc functions for measuring absolute elapsed time performance in MATLAB**, 2019. Disponível em: <<https://www.mathworks.com/company/newsletters/articles/improvements-to-tic-and-toc-functions-for-measuring-absolute-elapsed-time-performance-in-matlab.html>>. Acesso em: 1 Maio 2019.

VERMULM, R. Políticas para o desenvolvimento da indústria 4.0 no Brasil, São Paulo, Agosto 2018.

VIDAKOVIC, J. Z. et al. Advanced quaternion forward kinematics algorithm including overview of different methods for robot kinematics. **FME transactions**, Belgrado, 1 jan. 2014. 189-199.

WALDRON, K. J.; WANG, S. L.; BOLIN, S. J. A study of the Jacobian matrix of serial manipulators. **Journal of mechanisms transmissions and automation in design**, v. 107, n. 2, p. 230-237, Junho 1985.