

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**LUIZ GUSTAVO REZENDE**

**SIMULAÇÃO DA CINEMÁTICA INVERSA DE UM ROBÔ SCARA ATRAVÉS DA  
IMPLEMENTAÇÃO DE ALGORITMOS POR MÉTODOS NUMÉRICOS**

**CAMPO MOURÃO**

**2023**

**LUIZ GUSTAVO REZENDE**

**SIMULAÇÃO DA CINEMÁTICA INVERSA DE UM ROBÔ SCARA ATRAVÉS DA  
IMPLEMENTAÇÃO DE ALGORITMOS POR MÉTODOS NUMÉRICOS**

**Inverse kinematics simulation of a scara robot through the implementation of  
algorithms using numerical methods**

Trabalho de conclusão de curso de graduação,  
apresentado como requisito para obtenção do título de  
Bacharel em Engenharia Eletrônica da Universidade  
Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Flávio Luiz Rossini.

**CAMPO MOURÃO**

**2023**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**LUIZ GUSTAVO REZENDE**

**SIMULAÇÃO DA CINEMÁTICA INVERSA DE UM ROBÔ SCARA ATRAVÉS DA  
IMPLEMENTAÇÃO DE ALGORITMOS POR MÉTODOS NUMÉRICOS**

Trabalho de conclusão de curso de graduação,  
apresentado como requisito para obtenção do título de  
Bacharel da Federal do Paraná (UTFPR).

Data de aprovação: 24/Novembro/2023

---

Leandro Castilho Brolin  
Prof. Doutor  
Universidade Tecnológica Federal do Paraná (UTFPR)

---

Lucas Ricken Garcia  
Prof. Doutor  
Universidade Tecnológica Federal do Paraná (UTFPR)

---

Flávio Luiz Rossini  
Prof. Doutor  
Universidade Tecnológica Federal do Paraná (UTFPR)

**CAMPO MOURÃO**

**2023**

Dedico este trabalho a minha família, na esperança  
de retribuir todos incontáveis gestos de amor para  
que este sonho se tornasse possível.

## AGRADECIMENTOS

Para o Grande Arquiteto do Universo, pelo seu imensurável e perfeito amor sempre presente em minha vida, trazendo luz para abrilhantar meus caminhos e sabedoria para lidar com as dificuldades.

À minha família, minha mãe Lucia, meu pai Paulo, meu irmão Paulo Augusto, minha irmã Luciana (*in memorian*), minha esposa Claudia e minhas filhas Ângela e Clarice por todo afeto, apoio e inspiração durante todos esses anos.

Agradeço ainda, todo corpo docente de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná - Campo Mourão, em especial meu amigo, professor e orientador Dr. Flávio Luiz Rossini, por não medir esforços em me auxiliar durante esta jornada de forma justa e sábia.

A todos os demais que de alguma forma contribuíram para a realização deste trabalho.

Em algum lugar, alguma coisa incrível está  
esperando para ser descoberta.  
(Carl Sagan, 1934-1996)

## RESUMO

Este trabalho abordou a modelagem da cinemática direta e a simulação computacional da cinemática inversa do robô tipo SCARA (do inglês, *Selective Compliance Articulated Robot*) através da implementação de algoritmos por métodos numéricos, pelo *software* de simulação *Matrix Laboratory* (Matlab®). O estudo da cinemática, relaciona o comportamento da posição e orientação do manipulador no espaço cartesiano, sem levar em consideração as forças físicas envolvidas. Devido ao seu elevado grau de complexidade e equações não lineares, a cinemática inversa recorre a métodos numéricos para uma melhor ou possível solução do problema. Para tal manipulação, é necessário o levantamento de equações matemáticas relacionadas a cinemática do manipulador. Neste estudo, implementou-se a cinemática do robô SCARA, utilizou-se a convenção de Denavit-Hartenberg para as juntas e corpos móveis. Posteriormente, os métodos numéricos de otimização de Newton-Raphson, Levenberg-Marquardt e Halley foram implementados para a comparação dos algoritmos na plataforma de simulação. Os métodos obtiveram êxitos na execução, com erros inferiores a  $10^{-6}$  o que torna possível o estudo de trabalhos futuros.

Palavras-chave: algoritmos computacionais; método de Levenberg-Marquardt; método de Newton-Raphson; método de Halley; modelagem cinemática.

## ABSTRACT

This work addressed the modeling of direct kinematics and the computational simulation of the inverse kinematics of the SCARA type robot (Selective Compliance Articulated Robot) through the implementation of algorithms using numerical methods, using the Matrix Laboratory (Matlab®) simulation software. The study of kinematics relates the behavior of the manipulator's position and orientation in Cartesian space, without taking into account the physical forces involved. Due to its high degree of complexity and non-linear equations, inverse kinematics uses numerical methods for a better or possible solution to the problem. For such manipulation, it is necessary to survey mathematical equations related to the kinematics of the manipulator. In this study, the kinematics of the SCARA robot were implemented, using the Denavit-Hartenberg convention for joints and moving bodies. Subsequently, the numerical optimization methods of Newton-Raphson, Levenberg-Marquardt and Halley were implemented to compare the algorithms on the simulation platform. The methods were successful in execution, with errors lower than  $10^{-6}$  which makes it possible to study future work.

Keywords: computational algorithms; Levenberg-Marquardt's method; Newton-Raphson's method; Halley's method; kinematic modeling.



## LISTA DE ILUSTRAÇÕES

Figura 1–Instalações Industriais de Robôs ao Redor do Mundo .....	16
Figura 2–Números de Mestrados e Doutorados no Brasil.....	17
Figura 3–Maiores Mercados de Aplicações de Robôs de Serviços .....	20
Figura 4–Robôs Industriais .....	20
Figura 5–Manipulador Articulado .....	21
Figura 6–Manipulador Cartesiano.....	22
Figura 7–Manipulador Cilíndrico.....	22
Figura 8–Manipulador Paralelo .....	23
Figura 9–Manipulador Esférico .....	23
Figura 10–Manipulador SCARA .....	24
Figura 11–Variações de Tipos de Juntas .....	25
Figura 12–Braço Humano, DOF e Juntas .....	26
Figura 13–Vetor P em Relação ao Sistema de Coordenas {A} .....	27
Figura 14–Posição e Orientação de um Objeto no Espaço .....	28
Figura 15–Mapeamento Genérico .....	29
Figura 16–Parâmetros de Denavit-Hartenberg .....	32
Figura 17– Regra da Mão Direita.....	33
Figura 18– Manipulador SCARA e Sistemas de Referências .....	39
Figura 19 –Pseudocódigo.....	47
Figura 20– Manipulador SCARA eVariáveis.....	48

## LISTA DE TABELAS

Tabela 1 – Parâmetros de Denavit e Hatenberg.....	39
Tabela 2– Valores para Posição e Orientação Inicial do Robô.....	49
Tabela 3 – Resposta de NR.....	50
Tabela 4 – Resposta de LM e $\lambda = 0,1$ .....	51
Tabela 5 – Resposta de LM e $\lambda = 10$ .....	51
Tabela 6 – Resposta de Halley .....	52
Tabela 7 – Média de Comparação Entre os Algoritmos NR, LM e Halley .....	53

## LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
DH	DenavitHartenberg
DOF	<i>Degrees of Freedom</i>
GD	GradienteDescendente
GN	Gauss Newton
HD	<i>Hard Drive</i>
IFR	<i>International Federation of Robotics</i>
ISO	<i>International Organization for Standardization</i>
LM	Levenberg Marquardt
MATLAB	<i>Matrix Laboratory</i>
NR	Newton Raphson
P	Prismática
R	Rotativa
SCARA	<i>Selective Compliance</i>
TCC	Trabalho de Conclusão de Curso
UNICAMP	UniversidadeEstadual de Campinas
UTFPR	UniversidadeTecnológica Federal do Paraná

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>13</b>
<b>1.1</b>	<b>Objetivos</b> .....	<b>15</b>
1.1.1	Objetivo geral .....	15
1.1.2	Objetivos específicos.....	15
<b>1.2</b>	<b>Justificativa</b> .....	<b>16</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>19</b>
<b>2.1</b>	<b>Manipuladores Robóticos</b> .....	<b>19</b>
2.1.1	Configurações dos Manipuladores Robóticos .....	21
<u>2.1.1.1</u>	<u>Robô Articulado</u> .....	<u>21</u>
<u>2.1.1.2</u>	<u>Robô Cartesiano</u> .....	<u>22</u>
<u>2.1.1.3</u>	<u>Robô Cilíndrico</u> .....	<u>22</u>
<u>2.1.1.4</u>	<u>Robô Paralelo</u> .....	<u>23</u>
<u>2.1.1.5</u>	<u>Robô Esférico</u> .....	<u>23</u>
<u>2.1.1.6</u>	<u>Robô SCARA</u> .....	<u>24</u>
2.1.2	Movimentação dos Manipuladores Robóticos .....	24
<u>2.1.2.1</u>	<u>Juntas e Elos</u> .....	<u>25</u>
<b>2.2</b>	<b>Cinemática de Manipuladores Robóticos</b> .....	<b>26</b>
2.2.1	Movimentação de Corpos Rígidos.....	26
<b>2.3</b>	<b>Cinemática Direta</b> .....	<b>30</b>
<b>2.4</b>	<b>Parametrização de Manipuladores Robóticos</b> .....	<b>31</b>
<b>2.5</b>	<b>Cinemática Inversa</b> .....	<b>33</b>
2.5.1	Métodos numéricos .....	34
<u>2.5.1.1</u>	<u>Método de Newton-Raphson</u> .....	<u>35</u>
<u>2.5.1.2</u>	<u>Método de Levenberg-Marquardt</u> .....	<u>36</u>
<u>2.5.1.3</u>	<u>Método de Halley</u> .....	<u>36</u>
<b>3</b>	<b>METODOLOGIA</b> .....	<b>38</b>
<b>3.1</b>	<b>Modelagem Mecânica do Manipulador Robótico</b> .....	<b>38</b>
3.1.1	Parâmetros de Denavit-Hartenberg.....	39
3.1.2	Cinemática Direta.....	42
3.1.3	Cinemática Inversa.....	43
3.1.4	Critério de Parada e Informações.....	45
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b> .....	<b>48</b>
<b>4.1</b>	<b>Descrição das Variáveis</b> .....	<b>48</b>

4.2	Desempenho dos Métodos .....	50
5	CONCLUSÃO .....	54
	REFERÊNCIAS .....	56

## 1 INTRODUÇÃO

Eventos relevantes na evolução humana, mostram que nos primórdios da história já se utilizavam ferramentas e máquinas para auxílio de determinadas tarefas. Essa ideia, onde um robô possa ajudar pessoas, data-se do antigo Egito por volta de 3000 e 4000 A.C (Norton,2010). A palavra robô, tem origem no ano de 1921, com a peça do dramaturgo tcheco Karel Capek, derivada das palavras tchecas *robota* e *robotnik*, que significa respectivamente, trabalho obrigatório e servo (Mataric,2014). Um robô, é um manipulador multifuncional reprogramável projetado para mover peças, ferramentas, materiais ou algum dispositivo especial, através de variáveis de movimento programadas para a realização de uma variedade de tarefas (Spong;Vidyasargar, 2005). Um manipulador robótico é um conjunto de corpos, também chamados de elos, conectados em cadeia, através de juntas (Craig,2012). Em constante evolução e na busca por condições melhores, a revolução industrial no século XVIII, trouxe uma grande mudança na vida cotidiana das pessoas, aumento da produtividade e facilitação na mão de obra humana (Harari,2018). A revolução segue contínua e consolidada a uma nova etapa, a indústria 4.0. Esta, possui várias vertentes de estudos, aqui, busca-se a integração de técnicas e algoritmos para aprimorar o desempenho dos manipuladores robóticos (Rossini; Abreu; Oliveira,2023).

A robótica, é um campo de estudo em expansão, cuja a definição foi evoluindo ao longo do tempo, juntamente com seu próprio campo de aplicações, a partir da teoria de controle, da cibernética e da inteligência artificial (Mataric,2014). Dentro da robótica, uma das áreas fundamentais de análise é a cinemática, a qual abrange o conhecimento dos mecanismos. Segundo Craig (2012), a cinemática dos manipuladores é o estudo que compreende o mecanismo de movimento sem considerar as parcelas de forças físicas que causam imperfeições no movimento, buscando a localização de objetos no espaço tridimensional.

A utilização de ferramentas modernas como computadores e microcontroladores, possibilitam uma grande versatilidade em diferentes aplicações de manipuladores robóticos na indústria (Rossini; Abreu; Oliveira,2023). Porém, os robôs apresentam grandes dificuldades em suas modelagens matemáticas. A geometria de um manipulador pode variar conforme suas conexões, tornando-os cada

vez mais complexos conforme aumenta o número de juntas e elos. Com o propósito de lidar com a matemática aplicada ao robô, fixa-se, sistemas de referência às várias partes do mecanismo para que seja possível descrever as relações entre elas (Craig,2012). O método utilizado no presente trabalho para a parametrização da cinemática de estruturas de manipuladores é conhecido como Denevit-Hartenberg (Denavir; Hartenberg,1955) onde, define-se os valores para cada par de junta e elo que o robô possui.

A cinemática dos manipuladores pode ser dividida em duas, a cinemática direta e a cinemática inversa. A cinemática direta, consiste em obter a posição e a orientação final do efetuador através dos valores de juntas onde se tem os ângulos (Rossini; Abreu; Oliveira,2023). Ao contrário, tem-se a cinemática inversa, onde se conhece a posição desejada final do efetuador e deseja-se obter os possíveis ângulos de juntas que foram necessários para atingir tal valor (Craig,2012).

Por se tratar de equações não lineares, a cinemática inversa é considerada muito complexa (Kenwright,2012) e o modo tradicional para resolver os sistemas são através de métodos numéricos. Os sistemas de equações não lineares surgem em muitos problemas dentro da matemática, engenharia e computação devido a maioria dos sistemas físicos serem não lineares e quase sempre impossíveis de serem resolvidos de forma analítica (Souza,2015). Portanto, no presente trabalho serão utilizados os métodos clássicos de Newton-Rapshon, Levenberg-Marquardt e Halley para a análise da cinemática inversa do manipulador SCARA.

Existe uma vasta variedade de robôs industriais que são classificados conforme a sua configuração de juntas e elos. Aqui, aborda-se os robôs tipo SCARA, amplamente utilizados na automação industrial, possuem três juntas rotacionais paralelas e uma prismática (RRPR), sendo a última junta relacionada a ferramenta do manipulador, são vantajosos por serem robustos e rápidos, cerca de dez vezes mais rápidos que os outros robôs articulados (Rossini; Lima; Corrêa; Lopes; Barbosa; Barrozo, 2023).

Pode-se notar um crescimento expressivo no cenário da robótica no Brasil e no mundo. Assim, faz-se necessário o desenvolvimento de teorias e simulações que utilizam *softwares* em paralelo a construção propriamente dita dos manipuladores. O artigo de Souza *et al.* (2023), demonstra a construção de um aplicativo para planejamento de trajetória com base em um robô PUMA com seis graus de liberdade. Seguindo a linha de raciocínio de modelagem matemática em manipuladores

robóticos, tem-se outros artigos correlacionados para um melhor entendimento sobre (Sanches; Rossini,2016; Martines et al., 2015; Fudoli, et al., 2016; Oliveira et al., 2020).

Dada a posição e orientação do manipulador, a cinemática inversa consiste em encontrar todas as possíveis combinações de ângulos e de juntas que poderiam existir para se obter a posição e orientação final desejada. Devido ao seu alto grau de complexidade de equações matemáticas, é necessário o auxílio de ferramentas como: *softwares* e métodos numéricos para implementação dos algoritmos em busca de uma melhor solução.

## 1.1 Objetivos

Como base para o desenvolvimento do presente trabalho de conclusão de curso (TCC), tem-se, os objetivos específicos e geral. Dos quais deseja-se obter um resultado satisfatório ao fim do trabalho. O mesmo, será organizado em capítulos para facilitar a compreensão.

### 1.1.1 Objetivo geral

O objetivo geral tem como propósito desenvolver de maneira teórica a modelagem da cinemática de um manipulador genérico do tipo SCARA, bem como implementar os algoritmos de métodos numéricos iterativos para análise da cinemática inversa através do *software* (Matlab®). Os resultados serão analisados mediante a simulação dos métodos propostos, afim de compará-los. Aborda-se, tempo de execução, número de iterações e o erro.

### 1.1.2 Objetivos específicos

- Obter a modelagem da cinemática;
- Aplicar a convenção de Denavit-Hartenberg;
- Obter as matrizes de transformações homogêneas;
- Propor as ferramentas de análises não lineares;
- Desenvolver os algoritmos no *software* Matlab®;
- Obter a cinemática inversa através dos métodos de Newton-Raphson, Levenberg-Marquardt e Halley;

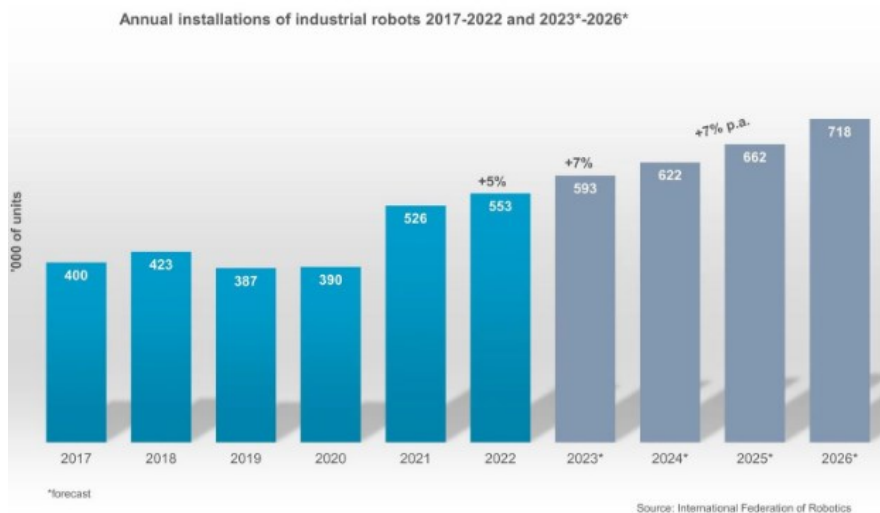


- Executar os algoritmos para extrair os dados de tempo de execução, número de iterações e erro;
- Comparar e analisar os resultados obtidos;

## 1.2 Justificativa

Pode-se ressaltar duas vertentes na área da robótica: a acadêmica e a industrial. A relevância dos manipuladores robóticos no meio industrial, destaca-se pela expansão jamais vista na história. Atualmente, o crescimento da população em escala global faz com que o aumento da demanda por sistemas robóticos cresça expressivamente, como pode-se observar na Figura 1. Busca-se soluções para resolver problemas relacionados a necessidade de substituir a mão de obra humana, em trabalhos repetitivos, de precisão e em condições sub humanas que seja limitada ou impossível de realizar.

**Figura 1 – Instalações Industriais de Robôs ao Redor do Mundo**



**Fonte: International Federation of Robotics (2023, p.2)**

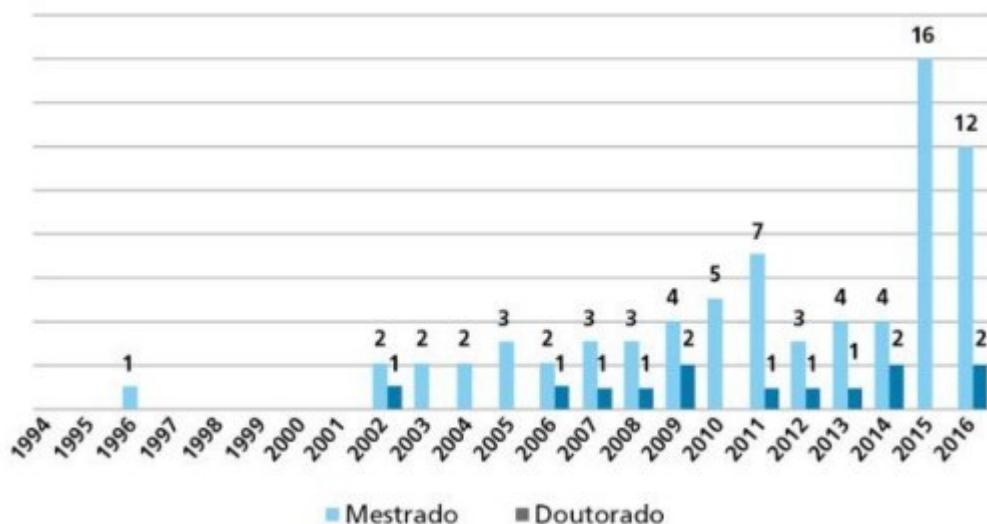
De acordo com a IFR (2023), foram registradas 553.052 instalações de robôs em fabricas de todo o mundo, com uma taxa de crescimento de 5% em 2022, superando o ano que passou. A expectativa é que 2023 supere 2022 com um aumento significativo de 7%, com mais de 590 mil unidades de robôs produzidos no mundo, diz a presidente da IFR Marina Bill (IFR,2023).

No Brasil, fomenta-se iniciativas para o desenvolvimento da indústria 4.0, que apesar de lenta, se comparado ao âmbito global, está crescendo novamente. Os

dados do IFR (2023) mostram que foram instalados 1.858 robôs no ano de 2022, em torno de 4% a mais em relação a 2021 e com grande potencial de aumento em 2023. Apesar de não ser um grande consumidor da indústria 4.0 no momento, o Brasil, precisa de conhecimento e desenvolvimento das áreas da robótica em geral. Pois, são de extrema importância para a competitividade global e avanços tecnológicos.

Do ponto de vista acadêmico, nota-se tendências de pesquisas e correlações de problemas ligados a robótica em expansão no Brasil e no mundo. O histórico da robótica na educação brasileira é relativamente nova e remete na década de 1980 através de algumas universidades, sendo a principal delas a Universidade Estadual de Campinas (Unicamp) (D'Abreu,2014). Com o intuito de corroborar o interesse acadêmico, Silva e Blikstein (2020) realizaram buscas na base de dados da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Google Acadêmico e outras citações encontradas nos próprios trabalhos, levando em consideração as palavras chaves relacionadas a robótica. A análise identificou somente 86 produções relacionadas ao tema entre 1994 a 2016, conforme a Figura 2 mostra.

**Figura 2 – Números de Mestrados e Doutorados no Brasil**



Fonte: Silva e Blikstein (2020, p.2)

De maneira geral, a utilização da robótica como recurso tecnológico na educação obteve repercussão nos meios escolares, acadêmicos e industriais nos últimos anos, com a popularidade do tema e também, com a necessidade de implementar essa tecnologia para o avanço do país. Segundo a Confederação

Nacional da indústria, as empresas que vem se destacando e obtendo os melhores proveitos são empresas que implementaram os conceitos da indústria 4.0 (Barbosa,2023).

O estudo da robótica agrega diversas áreas de conhecimento, tais como: matemática, física, programação, dentre outras. Portanto, afim de disseminar a educação na área da robótica e a constante necessidade de evoluir o país de maneira tecnológica, o trabalho proposto traz a modelagem da cinemática de um manipulador robótico tipo SCARA por sua grande atuação na indústria atual.

## 2 FUNDAMENTAÇÃO TEÓRICA

O presente capítulo tem como objetivo trazer ao leitor o raciocínio para a compressão do trabalho. Serão descritos os conceitos fundamentais da robótica, sobre as características físicas e teóricas que são necessárias para o desenvolvimento dos elementos cinemáticos dos manipuladores robóticos.

Posteriormente, devido à álgebra não linear decorrente da modelagem matemática do robô SCARA, torna-se crucial a revisão dos conceitos dos métodos numéricos que foram implementados no *software* Matlab® a fim de se obter uma resposta otimizada da cinemática inversa.

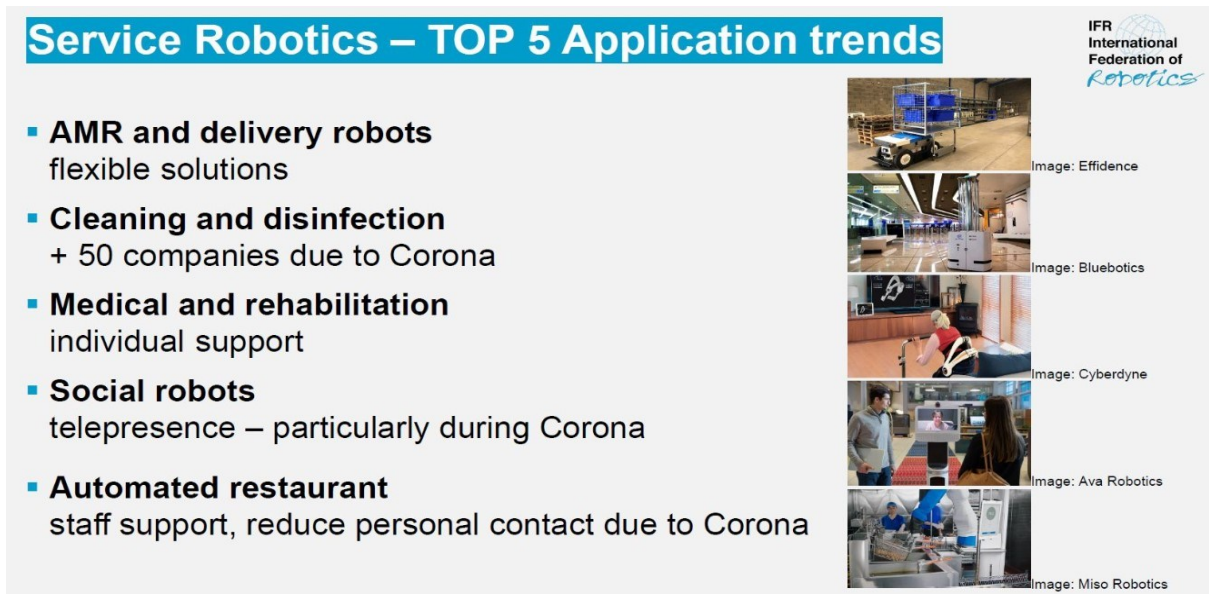
### 2.1 Manipuladores Robóticos

A definição do que é considerado um robô é algo impreciso até o momento, usualmente são classificados conforme a sua complexidade e funcionalidade. Segundo Mataric (2014), um robô é um sistema autônomo que existe no mundo físico, pode sentir e agir sobre o meio que está disposto e assim alcançar seus objetivos. Seguindo a linha de raciocínio, um robô precisa ter um corpo físico com juntas e elos, sensores, atuadores, efetadores e um sistema de controle. A *International Federation of Robotics* (IFR) aponta duas vertentes de robôs: os robôs de serviços (do inglês, *servisse robots*) e os robôs industriais (do inglês, *industrial robot*).

Os robôs de serviços são definidos como robôs de uso profissional ou pessoal que executam tarefas primordiais para equipamentos ou seres humanos. A Figura 3 apresenta as maiores áreas de aplicações para robôs de serviços no mundo. Já se fala também, em uma nova categoria de robôs, os robôs médicos, que são destinados a serem usados na área médica em geral (IFR, 2023b). Por possuir muitas formas e estruturas, bem como áreas de aplicação, os sistemas dos robôs de serviços são baseados sempre em algum grau de interação humano-robô, que podem ser sistemas totalmente autônomos.

Ainda, segundo a IFR (2023b) se um mecanismo com tecnologia robótica não atende todas as características descritas de um robô, ele é chamado de dispositivo robótico que pode ser um exoesqueleto sem energia, um manipulador remoto teleoperado, dispositivos háptico, dentre outros.

Figura 3 – Maiores Mercados de Aplicações de Robôs de Serviços



Fonte: International Federation of Robots (2021, p.5)

O uso do termo robô industrial é baseado na definição da Organização Internacional de Padronização ou ISO (*The International Organization for Standardization*). De acordo com a ISO 8373 um robô industrial é um manipulador multifuncional reprogramável e controlado automaticamente, programável em três eixos ou mais, que pode ser fixo no local ou fixado em um dispositivo móvel para o uso de aplicações industriais (ISO,2021).

Os robôs industriais são projetados para realizarem tarefas específicas com um alto desempenho. A habilidade do robô de mover seu corpo, braço e punho através de uma série de movimentos e posições depende diretamente de como ele é configurado mecanicamente (IFR,2023a). A Figura 4, exemplifica os robôs industriais.

Figura 4 – Robôs Industriais



Fonte: Adaptado de IFR (2023, p.3)

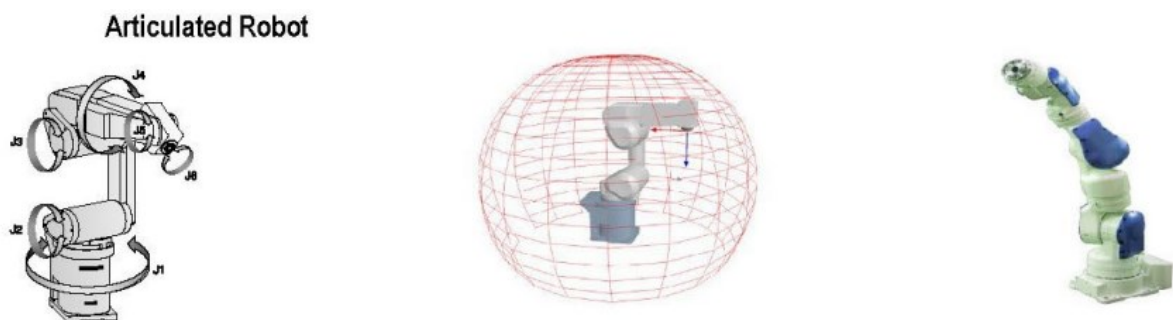
### 2.1.1 Configurações dos Manipuladores Robóticos

Os manipuladores robóticos quanto a sua estrutura, podem ser separados por dois grandes grupos: de cadeia aberta e de cadeia fechada (ALMEIDA,2018), que serão abordadas mais a fundo no decorrer do trabalho. Os robôs são dispostos em uma vasta variedade de tamanhos, formas e configurações mecânicas. Segundo Lynch e Park (2017), o robô é constituído de conexões de corpos rígidos de comprimento conhecido ligado a seu antecessor e seu sucessor por uma junção de um grau de liberdade, chamados de elos e juntas, sendo o primeiro elo da cadeia a base do robô e o último elo o efetuator (local da ferramenta). Devido suas diferentes configurações e graus de liberdade os manipuladores são divididos em: robôs articulados, cartesianos, cilíndricos, paralelos, SCARAS e esféricos (CRAIG,2012; IFR,2023). A seguir é apresentado um descritivo sobre cada um deles.

#### 2.1.1.1 Robô Articulado

O robô ou manipulador articulado possui uma configuração similar ao braço humano, comumente chamado de manipulador de juntas, com cotovelo ou antropomórfico (Craig,2012). Consiste em componentes análogos ao braço e antebraço que são fixados em uma base vertical. Estes, são conectados por duas juntas rotacionais no “ombro” e outra junta no “cotovelo”. No final do manipulador, no punho, pode-se ter duas ou mais juntas que irão depender da sua forma de trabalho. A Figura 5 traz um exemplo do robô articulado e sua respectiva área de trabalho. Segundo a IFR (2023), para ser considerado articulado o manipulador deve possuir um braço com no mínimo três juntas rotativas (RRR).

**Figura 5 – Manipulador Articulado**



**Fonte: IFR (2023, p.2)**

### 2.1.1.2 Robô Cartesiano

O manipulador cartesiano, linear ou pórtico, utiliza três elementos lineares para compor seus eixos  $x$ ,  $y$  e  $z$  no plano cartesiano como mostra a Figura 6. O braço possui três juntas prismáticas (PPP), sendo considerado uma das configurações mais simples dentre os manipuladores robóticos (Abreu *et al.*, 2022).

**Figura 6 – Manipulador Cartesiano**



Fonte: IFR (2023, p.4)

### 2.1.1.3 Robô Cilíndrico

A configuração do manipulador cilíndrico consiste em uma junta rotacional e duas juntas prismáticas de modo que seus eixos formem um sistema de coordenadas cilíndricas (RPP), podendo rotacionar até  $360^\circ$  em torno do seu próprio eixo (Cocota, 2013). Na Figura 7 abaixo, pode-se observar um manipulador cilíndrico.

**Figura 7 – Manipulador Cilíndrico**



Fonte: IFR (2023,p.4)

#### 2.1.1.4 Robô Paralelo

Os manipuladores paralelos são robôs cujos os braços possuem juntas prismáticas ou rotativas de forma simultâneas. São manipuladores de cadeias cinemáticas fechadas (Silva,2020). Na Figura 8 ilustra-se um exemplo de manipulador paralelo.

**Figura 8 – Manipulador Paralelo**

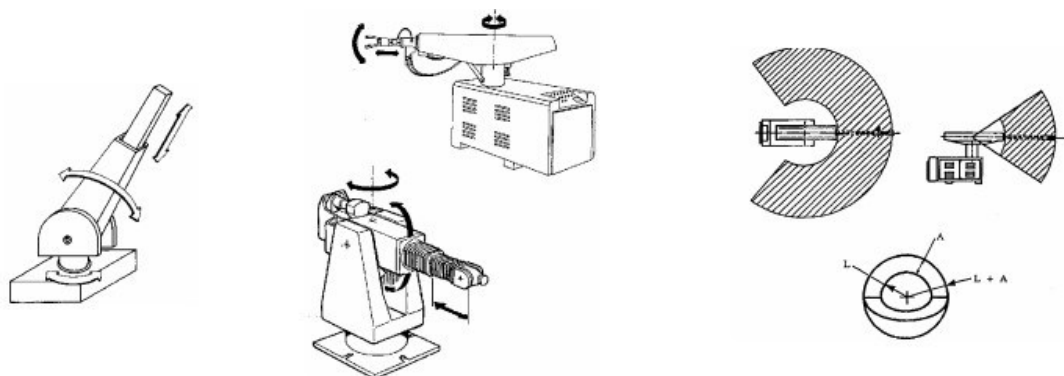


Fonte: IFR (2023, p.6)

#### 2.1.1.5 Robô Esférico

A configuração do robô esférico, também conhecido por manipulador polar, de acordo com Craig (2012) é formado pela substituição da terceira junta rotativa do manipulador articulado por uma junta prismática (RRP). Estas juntas provêm a capacidade do robô mover seu braço dentro de um espaço esférico, como mostra a Figura 9 a seguir.

**Figura 9 – Manipulador Esférico**



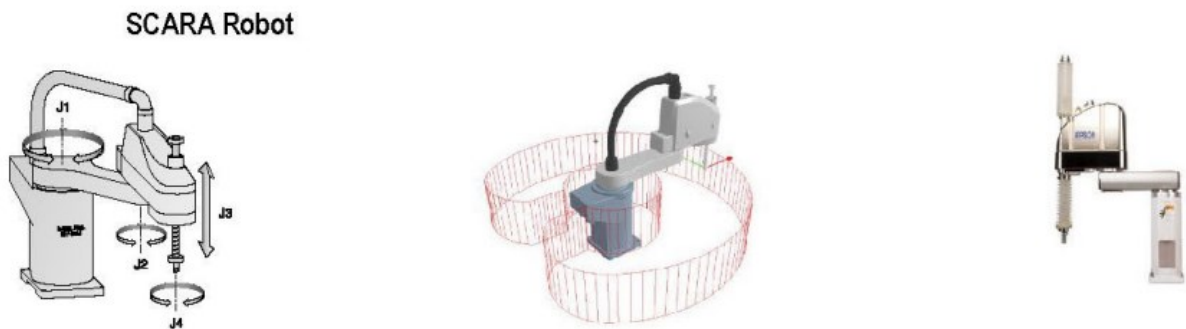
Fonte: Santos (2004, p.23)



### 2.1.1.6 Robô SCARA

O manipulador SCARA (*Selective Compliance Assembly Robot*) foi introduzido no Japão em 1979 com um conceito novo em relação a robôs de montagem (Almeida,2018). Possui uma configuração de duas juntas rotacionais em paralelo para se ter movimento no plano e uma junta prismática no final do efetuador (RRP) (Craig,2012). Embora o volume de trabalho do robô SCARA seja semelhante a esférica, ele é bastante diferente em sua gama de aplicações em relação ao robô polar/esférico. Esse manipulador é amplamente utilizado na indústria eletrônica para montagem de placas de componentes de circuitos impresso (Bayer, et. al., 2011). Um exemplo de manipulador SCARA e sua respectiva área de trabalho pode observado na Figura 10.

**Figura 10 – Manipulador SCARA**



Fonte: IFR (2023, p.3)

### 2.1.2 Movimentação dos Manipuladores Robóticos

O movimento do robô manipulador é definido pelo número total de movimentos que ele exerce, popularmente conhecido como grau de liberdade (*degrees of freedom – DOF*). De acordo com Mataric (2014), grau de liberdade é qualquer um dos números mínimos de coordenadas necessárias para especificar completamente o movimento de um sistema mecânico, ou seja, a maneira pela qual o sistema pode se mover. Considere uma porta, que consiste em um único corpo rígido conectado a uma parede por uma junta articulada, neste exemplo, pode-se dizer que esta porta possui somente um grau de liberdade, que convenientemente é representado pelo ângulo da junta da dobradiça (Lynch; Park, 2017).

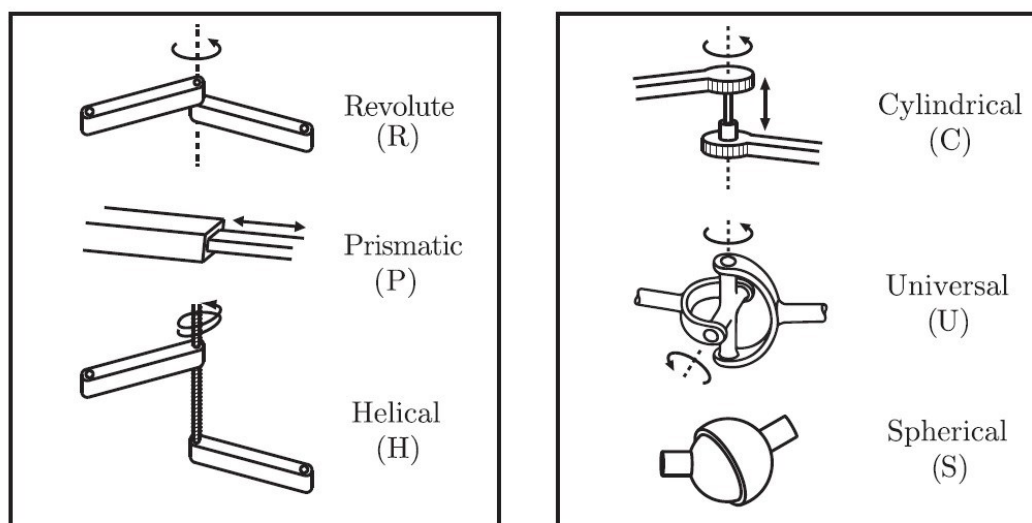
Ainda, segundo Lynch e Park (2017) conforme a complexidade do robô aumenta, é necessário ter mais graus de liberdade para ter eficiência em realizar os movimentos ou tarefas desejadas, portanto, as juntas restringem o movimento do corpo rígido.

### 2.1.2.1 Juntas e Elos

Mecanicamente, um manipulador robótico é composto de elos quase rígidos e juntas que permitem o movimento relativo dos elos vizinhos (Craig,2012). Os elos são corpos de natureza estática ou dinâmica responsáveis por ligar uma junta na outra (Romano; Dutra,2002). Eles, mantêm fixas as relações de juntas sucessivas e podem ser descritos com um determinado número de parâmetros geométricos. Os elos intermediários são separados por duas juntas, a anterior e a posterior. Nos elos extremos do manipulador só há uma junta delimitadora.

Dentro da robótica existem diversos tipos de juntas conforme mostra a Figura 11. Em maior número de aplicações industriais encontram-se as juntas prismáticas (P) e rotacionais (R), que combinadas visam compor um par cinemático e simplificar o processo de produção mecânica, além de obter um maior controle do movimento entre os elos (Abreu; Rossini; Oliveira, 2022).

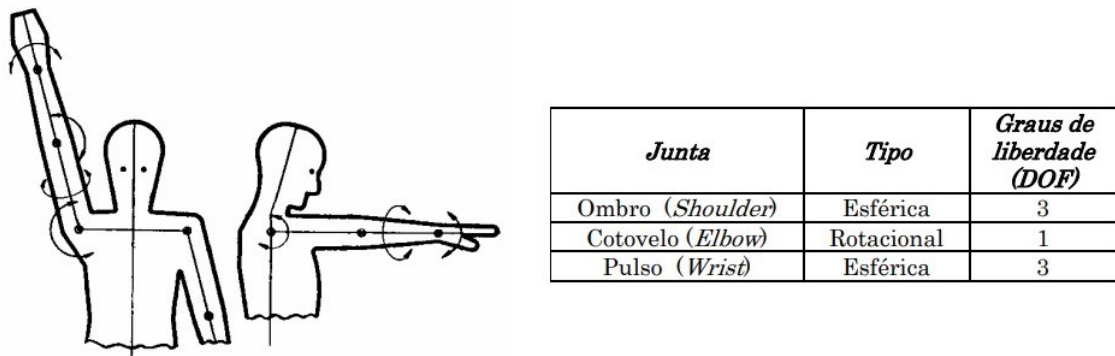
**Figura 11 – Variações de Tipos de Juntas**



Fonte: Lynch e Park (2017, p.34)

Para o movimento rotacional temos a junta rotativa (R) que permite movimentos em torno do eixo da junta, também chamada de revoluta ou dobradiça. A junta prismática (P) possui movimento translacional ou retilíneo ao longo da direção do eixo da junta e pode ser chamada de junta linear ou deslizante (Lynch; Park, 2017). A Figura 12 exemplifica os graus de liberdade de um braço, com exceção da mão e dos dedos, bem como os tipos de juntas atreladas ao ser humano.

Figura 12 – Braço Humano, DOF e Juntas



Fonte: Santos (2004, p.20)

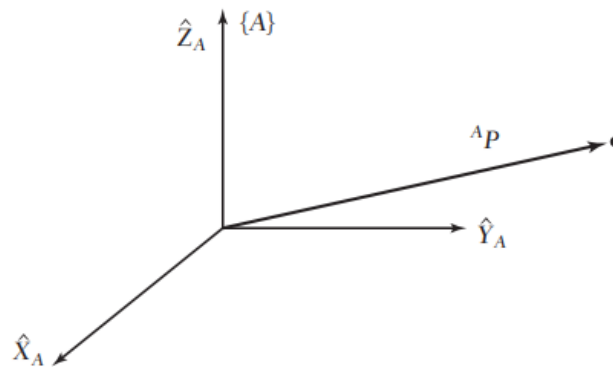
## 2.2 Cinemática de Manipuladores Robóticos

O estudo da cinemática, conforme citado no capítulo anterior, estuda o movimento dos corpos sem se preocupar com as forças que o causam (Craig, 2012). Ela, se divide em dois grandes nichos: a cinemática direta e a cinemática inversa. Para uma melhor compreensão do tema é fundamental o entendimento de alguns conceitos que são descritos no decorrer desta seção.

### 2.2.1 Movimentação de Corpos Rígidos

Uma vez definido o sistema de coordenadas no espaço euclidiano tridimensional  $\hat{X}$ ,  $\hat{Y}$  e  $\hat{Z}$ , pode-se localizar um ponto qualquer através de um vetor posição de dimensões  $3 \times 1$  (Craig, 2012). Para referenciar tais movimentos, são estipulados sistemas de coordenadas afim de definir a posição do corpo no espaço, utilizando uma referência inercial (Farnese, 2020). A Figura 13 demonstra um sistema de coordenadas  $\{A\}$  com três vetores unitários ortogonais representados por  ${}^A P$ . Que por sua vez, representa uma posição no espaço ou um conjunto de três números.

**Figura 13 – Vetor P em Relação ao Sistema de Coordenadas {A}**



Fonte: Craig (2012, p.54)

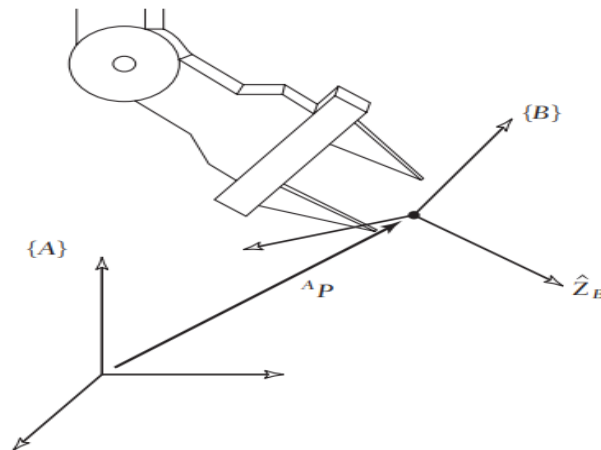
De forma individual, os elementos do vetor posição  ${}^A P$  são definidos com dados de  $x$ ,  $y$  e  $z$  do espaço tridimensional.

$${}^A P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (1)$$

sendo  $p_x$  o valor ao longo do eixo  $\hat{X}$ ,  $p_y$  o valor ao longo do eixo  $\hat{Y}$  e  $p_z$  o valor ao longo do eixo  $\hat{Z}$ .

Além da posição, dentro da robótica faz-se necessário descrever a orientação de um corpo no espaço euclidiano tridimensional. De acordo com Craig (2012) para descrever a orientação de um corpo, fixa-se um sistema de coordenadas ao corpo e posteriormente o descreve com relação ao sistema de referência. Como exemplo, a Figura 14 apresenta uma forma de descrever o sistema de coordenadas  $\{B\}$  através da relação de vetores unitários de seus eixos principais em relação ao sistema de coordenadas  $\{A\}$ .

Figura 14 – Posição e Orientação de um Objeto no Espaço



Fonte: Craig (2012, p.54)

Por conveniência, segundo Craig (2012) a matriz de rotação  $R$  de dimensões  $3 \times 3$ , descreve em particular  $\{B\}$  em relação a  $\{A\}$  e são compostas pelos vetores unitários de um sistema de coordenadas em relação ao outro. Assim, um conjunto de três vetores pode ser usado para especificar uma orientação.

$${}^A R_B = [{}^A \hat{X}_B \quad {}^A \hat{Y}_B \quad {}^A \hat{Z}_B] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2)$$

sendo,  $r_{ij}$  os valores de expressão escalar que compõem a matriz  ${}^A R_B$ . Os componentes de qualquer vetor são projeções desse mesmo vetor nas direções unitárias de seu sistema de referência (Craig, 2012). Portanto, cada componente da matriz de rotação pode ser escrito como o produto escalar de um par de vetores unitários.

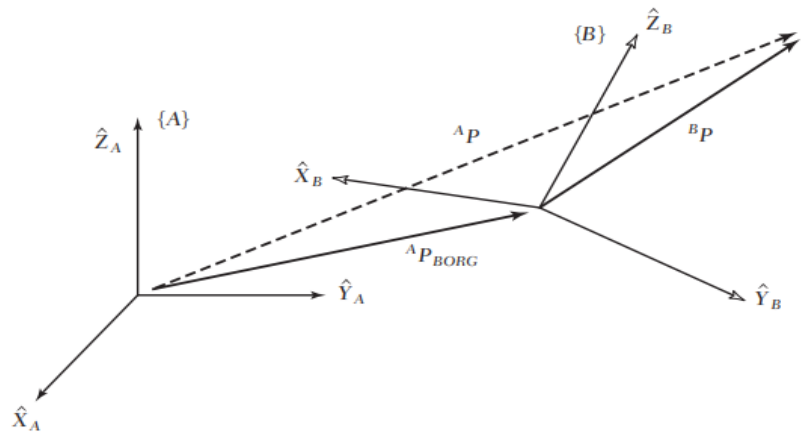
Foram descritas até o momento, a posição e a orientação do manipulador dadas as Equações (1) e (2). Mas, para descrever totalmente onde encontra-se o manipulador robótico faz-se necessário um mapeamento que aponta a posição e a orientação de forma simultânea. Por conveniência, utiliza-se como base a origem do sistema de referência fixado ao corpo  $\{B\}$ . Dado  ${}^B P$ , tem-se:

$${}^A P = {}^A R_B {}^B P + {}^A P_{BORG} \quad (3)$$

sendo,  ${}^A P_{BORG}$  o vetor que localiza a origem de  $\{B\}$ .

A Equação (3) descreve o mapeamento de um sistema de referência para um outro segundo sistema de referência, exemplificada na Figura 15.

**Figura 15 – Mapeamento Genérico**



Fonte: Craig (2012, p.57)

O processo para encontrar  ${}^A P$  ocorre multiplicando  ${}^B P$  pela matriz rotacional de  $\{B\}$  em relação a  $\{A\}$ , mais a variável  ${}^A P_{BORG}$  (Craig, 2012). Afim de escrever a Equação (3) na forma matricial, Craig (2012) sugere definir um operador matricial  $4 \times 4$  e usar um vetor posição  $4 \times 1$ , assim define-se:

$$\begin{bmatrix} {}^A P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^A R_B & | & {}^A P_{BORG} \\ 0_{1 \times 3} & | & 1 \end{bmatrix} + \begin{bmatrix} {}^B P \\ 1 \end{bmatrix} \quad (4)$$

sendo,

$${}^A T_B = \begin{bmatrix} {}^A R_B & | & {}^A P_B \\ 0_{1 \times 3} & | & 1 \end{bmatrix} \quad (5)$$

a Equação (5) é chamada de matriz de transformação homogênea que permite a rotação e a translação da transformação geral na forma de uma única matriz. À matriz de transformação também serve para descrever sistemas de referência. O sistema de referência  $\{B\}$  em relação a  $\{A\}$  é descrito como  ${}^A T_B$ .

Um corpo rígido no espaço pode ser descrito pelo seu movimento de translação com base no frame de referência, e sua orientação expressa pelas componentes do vetor unitário anexado ao corpo de referência do mesmo frame (Siciliano *et al.*, 2010). De acordo com Farnese (2020) a matriz de transformada homogênea pode ser simplificada pela equação abaixo,

$${}^{n-1}T_n = \begin{bmatrix} {}^{n-1}R_n & {}^{n-1}P_n \\ 0 & 1 \end{bmatrix} \quad (6)$$

a Equação (6) representa a matriz de transformação homogênea  ${}^{n-1}T_n$ , que relaciona a matriz de rotação  ${}^{n-1}R_n$  e o vetor posição  ${}^{n-1}P_n$  de um manipulador robótico com  $n$  graus de liberdade.

### 2.3 Cinemática Direta

O conceito da cinemática direta (do inglês, *forward kinematics*) diz respeito ao cálculo da posição e orientação final do efetuador dada a configuração das juntas. Para obter tais ângulos e deslocamento entre duas conexões deve-se considerar o grau de mobilidade associado as variáveis das juntas dos robôs que podem ser rotacionais (movimento angular) ou prismática (deslocamento linear), assim é possível conhecer a posição do manipulador em relação ao frame de referência, que normalmente é a base do robô (Lynch; Park, 2017)..

Para descrever a cinemática direta através de equações, utilize-se a concatenação das transformadas homogêneas da cadeia cinemática do manipulador (Farnese, 2020). As matrizes de transformações homogêneas individuais podem se relacionar para obter a posição final do manipulador (Lynch; Park, 2017).. Exemplo, o último frame  $\{n\}$  de um manipulador robótico pode ser descrito como:

$${}^0T_n = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 \dots {}^{n-1}T_n \quad (7)$$

Através da Equação (7) a multiplicação individual de cada matriz ocorre sempre do último frame  $\{n\}$  até a base referencial  $\{0\}$  na origem do sistema. Substitui-se a Equação (6) e aplica-se a Equação (7), assim a forma matricial pode ser expressa por:

$${}^0T_n = \begin{bmatrix} {}^0R_1 & {}^0P_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^1R_2 & {}^1P_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^2R_3 & {}^2P_3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^3R_4 & {}^3P_4 \\ 0 & 1 \end{bmatrix} \dots \begin{bmatrix} {}^{n-1}R_n & {}^{n-1}P_n \\ 0 & 1 \end{bmatrix} \quad (8)$$

, o objetivo da concatenação das transformadas homogêneas é determinar o efeito acumulado das variáveis de junta na orientação e posição do efetuador final. Nota-se que a multiplicação das matrizes homogêneas individuais deve sempre ocorrer da direita para esquerda (Craig, 2012). De acordo com Spong *et al.* (2005) ainda é possível simplificar a solução aplicando-se os parâmetros de Denavit-Hartenberg.

## 2.4 Parametrização de Manipuladores Robóticos

Para que se seja possível conhecer a localização do robô manipulador dentro do espaço tridimensional é necessário estabelecer alguns parâmetros. De acordo com Craig (2012) a posição e a orientação de um objeto são descritas matematicamente fixando-se uma referência sobre o sistema de coordenadas cartesianas (frame) do robô.

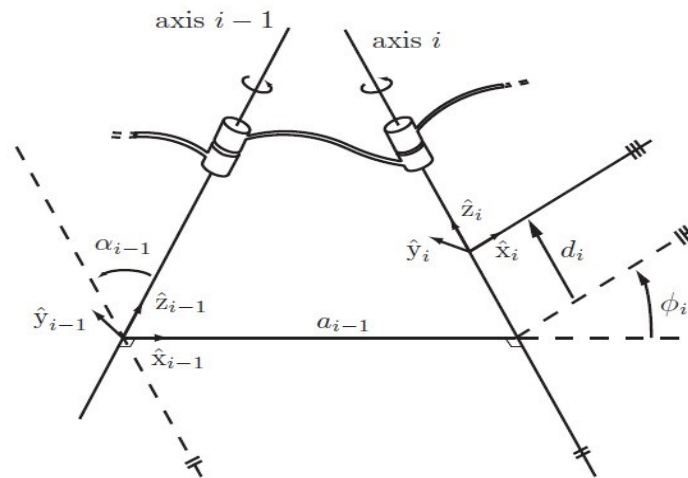
Uma estrutura física rígida no espaço tridimensional pode transladar e rotacionar em qualquer dos eixos  $\hat{X}$ ,  $\hat{Y}$  e  $\hat{Z}$  de seu frame de referência (Lynch; Park, 2017)..

A parametrização de Denavit-Hartenberg (DH) propõe uma metodologia baseada em matrizes muito bem estruturada para obter os sistemas de coordenadas e as transformações associadas a cada elo do manipulador robótico para sistematizar a descrição da cinemática de sistemas articulados com  $n$  graus de liberdade (DENAVID, 1955). Uma vez que estes sistemas de coordenadas fixados ao elo são dispostos, as transformações entre sistemas adjacentes podem ser descritas por uma simples matriz de transformação de coordenadas homogênea (Schilling, 1990). Existem outros métodos disponíveis para a descrição de sistemas robóticos (Rossini; Abreu; Oliveira, 2023). No trabalho presente, aborda-se somente a convecção de DH.

A notação de DH pode ser dividida em quatro etapas fundamentais que se subdividem em vários passos repetidos parcialmente em ciclos, que envolve parâmetros para cada par de junta-elo (Santos, 2004). Na Figura 16, observa-se que são estabelecidos os sistemas de referências.



**Figura 16 – Parâmetros de Denavit-Hartenberg**



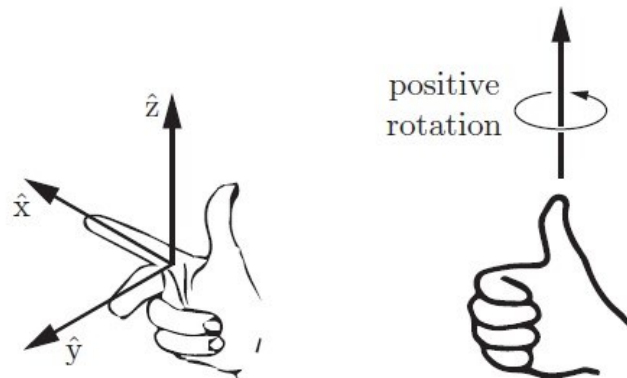
**Fonte: Lynch e Park (2017, p.63)**

A interseção dos eixos de um sistema de coordenadas define a origem do mesmo e, portanto, pode-se descrever a posição relativa entre os próprios sistemas. Logo, as variáveis são definidas como  $\alpha_i$ ,  $a_i$ ,  $d_i$ , e  $\Phi_i = \theta_i$  que representam:

- a)  $\alpha_{i-1}$  representa o ângulo formado entre  $Z_{i-1}$  e  $Z_i$ , medido ao longo de  $X_{i-1}$ ;
- b)  $a_{i-1}$  representa a distância de  $Z_{i-1}$  até  $Z_i$ , medido ao longo de  $X_{i-1}$ ;
- c)  $\theta_i$  representa o ângulo entre  $X_{i-1}$  e  $X_i$ , medido ao longo de  $Z_i$ ;
- d)  $d_i$  representa a distância de  $X_{i-1}$  até  $X_i$ , medido ao longo de  $Z_i$ .

O método propõe transferir a origem de um sistema cartesiano para o frame subsequente, rotacionando e transladando o frame de referência. Essa denotação deve seguir uma certa ordem para aplicar os quatro passos fundamentais. Segundo (Craig, 2012), o primeiro passo a considerar é o comprimento do elo  $a_i$ . Logo após, para definir o elo de torção  $\alpha_i$ , aplica-se a regra da mão direita, descrita na Figura 17, para estipular os ângulos entre os eixos  $i - 1$  e  $i$ , frame a frame. O deslocamento de elo  $d_i$  é medido ao longo do eixo da junta  $i$ . Esta variável é o comprimento do elo  $a_{i-1}$  até o elo  $a_i$  que cruza o eixo  $i$ , que vão variar apenas em eixos prismáticos. Por fim, o ângulo de junta  $\theta_i$  que é adotada para juntas rotacionais.

**Figura 17– Regra da Mão Direita**



Fonte: Lynch e Park (2017, p.62)

Com as informações obtidas para cada grau de liberdade da estrutura robótica, pode-se construir uma tabela para descrever o movimento cinemático do manipulador robótico (Kenwright, 2013).

## 2.5 Cinemática Inversa

A cinemática inversa (do inglês, *inverse kinematics*) tem como objetivo obter os valores de juntas representadas por  $\theta_n$ , dada a posição e a orientação do manipulador no espaço tridimensional (Santos, 2004). O grande problema dos manipuladores robóticos para encontrar uma solução dentro da cinemática inversa é que ela possui equações não lineares, bem como soluções múltiplas ou ainda, soluções nulas (Aristidou *et al.*, 2017).

Conforme Craig (2012) diz, há duas possíveis abordagens para encontrar a solução da cinemática inversa: através de métodos numéricos ou de soluções analíticas (algébrica e geométrica).

Um manipulador constituído por  $\{n\}$  juntas, sendo a variável de junta representado por  $\theta_n$ , pode ser descrita como:

$$\hat{\theta}_n = [\theta_1, \theta_2, \theta_3, \theta_4, \dots, \theta_n]^T \quad (9)$$

, que consiste no vetor resultante de dimensões  $1 \times n$  que irá proporcionar condições necessárias em termos de ângulos de juntas para o destino final do efetuador (Aristidou *et al.*, 2017).

Os métodos analíticos são baseados na geometria da cadeia cinemática ou na equação algébrica da cinemática do manipulador, na qual, tem sua aplicação dificultada conforme aumenta o número de graus de liberdade do robô (Farnese, 2020). A utilização de métodos numéricos para a solução da cinemática inversa, consiste em métodos iterativos que buscam encontrar a raiz da equação, a solução do problema (Lynch; Park, 2017)..

De certa forma, a estrutura física do robô limita as operações analíticas, tornando-as quase impossíveis para manipuladores com altos graus de liberdade. A cinemática inversa consegue abranger sistemas mais genéricos, com altos graus de liberdade (Farnese, 2020). O aumento de graus de liberdade pela análise de métodos numéricos resulta em uma solução mais custosa, necessita de maiores recursos computacionais e conseqüentemente tempos maiores para encontrar a raiz da questão. Porém, é factível o uso da cinemática inversa para robôs com altos graus de liberdade.

### 2.5.1 Métodos numéricos

Sistemas de equações não lineares são compostos por combinações de funções algébricas e transcendentais, tais como funções trigonométricas, exponenciais, logarítmicas, etc. São problemas que se encontram no cotidiano, dentro das áreas de matemática, robótica e engenharias de forma geral. Em sua grande maioria, é impossível ter uma solução de forma analítica, ao contrário dos problemas lineares (Taylor, 2005).

Nas últimas décadas, a evolução dos computadores modificou os estudos sobre a utilização de métodos numéricos para a resolução de problemas não lineares (Reis, 2019). Os métodos numéricos iterativos, aqui exemplificados, são baseados em uma sequência de vetores que partem de uma estimativa inicial, podendo convergir ou não para uma raiz do sistema (Conte, 1980).

Os métodos numéricos buscam soluções para cinemática inversa através de algoritmos que reduzem o erro entre a posição real e a desejada a cada nova iteração (Aristidou *et al.*, 2017). Serão analisados três métodos numéricos iterativos no decorrer deste trabalho que são: método de Newton-Raphson, método de Levenberg-Marquardt e o método de Halley.

### 2.5.1.1 Método de Newton-Raphson

A aplicação do método de Newton-Raphson busca determinar as raízes ou zeros de uma equação não linear por meio de iterações (Souza, 2015). Através de uma estimativa inicial para a raiz, diversas correções de forma sequencial são estimadas até que solução atinja uma resposta otimizada desejada (Ruggiero; Lopes, 2014). Dada uma função  $F(\theta)$ , o método é obtido aplicando-se a expansão de Taylor em torno de  $(\theta_n)$ , assim:

$$F(\theta) = F(\theta_n) + \frac{F'(\theta_n)}{1!}(\theta - \theta_n) + \frac{F''(\theta_n)}{2!}(\theta - \theta_n)^2 + \dots \quad (10)$$

, sendo,  $F'(\theta_n)$  o termo da primeira derivada e  $F''(\theta_n)$  o termo da segunda derivada da função em relação a  $\theta$ . Pode-se ainda, substituir o  $\theta$  por  $\theta_{n+1}$  e considerar somente até o termo de primeira ordem da série, se  $\theta_{n+1}$  é uma raiz da equação e  $\theta_n$  é um valor próximo a  $\theta_{n+1}$ . Assim, pode-se igualar a função a zero. Sendo  $F(\theta) = 0$ , pode ser expressa por:

$$F(\theta_{n+1}) = F(\theta_n) + F'(\theta_n)(\theta_{n+1} - \theta_n) = 0 \quad (11)$$

, deixa-se em termos de  $\theta_{n+1}$ , obtém-se:

$$\theta_{n+1} = \theta_n - \frac{F(\theta_n)}{F'(\theta_n)} \quad (12)$$

, sendo, que o valor de  $\theta_{n+1}$  representa a aproximação para a raiz da função  $F(\theta)$ . De forma matricial, a Equação (12) pode ser reescrita como:

$$\theta_{n+1} = \theta_n - J^{-1}(\theta_n)F(\theta_n) \quad (13)$$

, em que  $J$  é uma matriz Jacobiana que contém as derivadas parciais das funções não lineares e que relaciona as taxas de variações da posição e orientação do efetuador com as variáveis de junta do manipulador, para ter uma aproximação linear do movimento (BUSS, 2009). Através do método, espera-se obter uma convergência quadrática, desde que o valor inicial seja conhecido e que  $J^{-1}(\theta_n)$  exista.

### 2.5.1.2 Método de Levenberg-Marquardt

O método de Levenberg-Marquardt é também chamado de métodos dos mínimos quadrados, e tem como objetivo ajustar um conjunto de dados de forma a minimizar a soma dos quadrados das distâncias entre os pontos ajustados e os pontos amostrados por meio de uma sequência de atualizações de valores (Lima, 2022).

Este método é uma combinação de dois algoritmos de otimização, Gauss-Newton e o Gradiente Descendente (do inglês, *Gradient Descent*) e sua equação é descrita como:

$$\theta_{n+1} = \theta_n - (J^T(\theta_n)J(\theta_n) + \lambda I)^{-1}J^T(\theta_n)F(\theta_n) \quad (14)$$

sendo  $J^T(\theta_n)$  a matriz Jacobiana transposta, e  $I$  a matriz identidade que ajusta a dimensão das variáveis para que o cálculo seja possível.

Segundo o próprio Levenberg (1944) ao introduzir o parâmetro chamado de amortecimento ajustável  $\lambda I$ , conforme descrito Equação (14), o algoritmo apresenta maior estabilidade. O parâmetro ajustável influencia na direção e no tamanho do passo do sistema, portanto a medida em que o parâmetro cresce, a direção tende a se aproximar do Gradiente Descendente (GD), caso contrário, se o parâmetro aproximar de zero ele tende a direção de Gauss-Newton (GN) (Alves, 2021).

De acordo com Buss (2009) o parâmetro de amortecimento  $\lambda$  deve ser grande o suficiente para manter o sistema controlado e pequeno o bastante para não comprometer a taxa de convergência.

### 2.5.1.3 Método de Halley

Este método foi proposto no ano de 1694 por Edmund Halley para o cálculo das raízes de um polinômio. O brilhante cientista da época, ficou famoso por calcular a órbita do cometa que recebeu seu nome (Ferreira, 2013). Este método exige avaliação dos termos derivativos de primeira e segunda ordem para encontrar a raiz, sendo expresso por:

$$\theta_{n+1} = \theta_n - \left[ I - \frac{1}{2}L(\theta_n) \right]^{-1} J^{-1}(\theta_n)F(\theta_n) \quad (15)$$

sendo,

$$L(\theta_n) = J^{-1}(\theta_n)[H(\theta_n)]^T \quad (16)$$

em que  $H$  é responsável por organizar as derivadas parciais de segunda ordem da função, conhecida como matriz Hessiana.

Em geral, por apresentar uma ordem de convergência maior, possui um número menor de iterações para obter a raiz (Ferreira, 2013).

### 3 METODOLOGIA

Neste capítulo será apresentada a metodologia aplicada ao trabalho, no qual visa a implementação e comparação de algoritmos de métodos numéricos como solução da cinemática inversa do manipulador robótico tipo SCARA. Como o desenvolvimento do trabalho necessita de cálculos complexos os ensaios foram baseados em um ambiente de simulação. A plataforma utilizada foi a *Matrix Laboratory* (Matlab®).

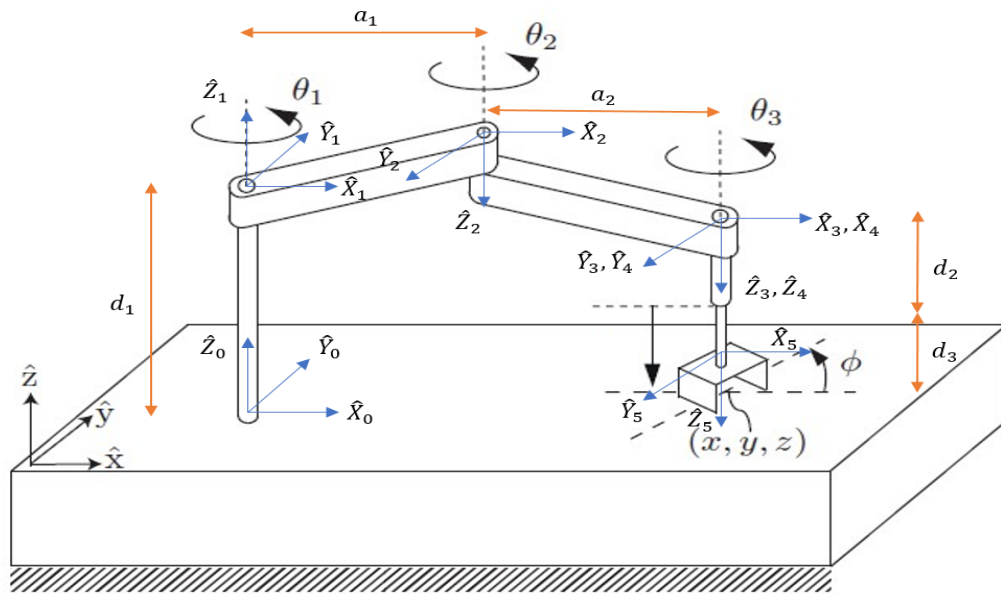
O capítulo descreve a estrutura do robô no qual fixa-se um frame de referência mapeando o manipulador no espaço tridimensional  $X$ ,  $Y$  e  $Z$ . A seguir, aplica-se o método da cinemática direta e a conversão dos parâmetros de Denavit-Hartenberg para extrair dados e obter as matrizes de transformação homogênea. Ainda, implementa-se as matrizes de transformação homogênea para cada par de elo-junta do manipulador, a fim de descrever, através da concatenação das matrizes individuais, a trajetória do robô. Por fim, utiliza-se os métodos numéricos de funções não lineares para obter a solução da cinemática inversa em função dos ângulos de juntas  $\theta_n$ .

#### 3.1 Modelagem Mecânica do Manipulador Robótico

O robô proposto neste trabalho é do tipo SCARA, ou seja, possui duas juntas rotacionais (RR) e uma prismática (P) o que proporciona quatro graus de liberdade, sendo que, os três primeiros são para o posicionamento do manipulador e o último para a orientação do efetuador (local da ferramenta). Neste trabalho com fins didáticos, a junta final não é considerada.

Conforme descrito no capítulo anterior, na Seção 2.2, é preciso saber a posição inicial do robô e fixá-lo em torno de um eixo no espaço tridimensional  $X$ ,  $Y$  ou  $Z$ . A Figura 18 apresenta os sistemas de coordenadas atribuídos ao manipulador e seus respectivos frames fixados a partir de uma referência inicial, no caso, o frame zero. A escolha do eixo  $Z$  como base, possibilita a manipulação dos movimentos de rotação e translação dos elos e juntas do manipulador.

**Figura 18 – Manipulador SCARA e Sistemas de Referências**



Fonte: Adaptado de Lynch e Park (2017, p. 81)

### 3.1.1 Parâmetros de Denavit-Hartenberg

Dado o robô manipulador e suas informações atribuídas, conforme mostra a Figura 18, pode-se então, extrair os dados necessários para aplicar os parâmetros de Denavit-Hartenberg exemplificado na Seção 2.4. Assim, constrói-se uma tabela de informações conforme descrita abaixo:

**Tabela 1 – Parâmetros de Denavit e Hatenberg**

<i>Juntas (i)</i>	$a_{i-1}$	$\alpha_{i-1}$	$d_{i-1}$	$\theta_{i-1}$
1	0	0	$d_1$	$\theta_1$
2	$a_1$	$\pi$	0	$\theta_2$
3	$a_2$	0	$d_2 *$	0
4	0	0	0	$\theta_3$
5	0	0	$d_3$	0

Fonte: Autoria Própria (2023)

Para o cálculo da cinemática direta e a concatenação das matrizes de transformação homogênea é preciso aplicar os parâmetros de Denavit-Hartenberg extraídos da Tabela 1 na matriz de transformada genérica  $4 \times 4$ , expressa por:



$${}^{n-1}T_n = Rot(Z, \theta_n) Trans(Z, d_n) Trans(X, a_n) Rot(X, \alpha_n) \quad (17)$$

sendo,  $Rot(Z, \theta_n)$  o movimento de rotação em torno do eixo  $Z$  em relação a  $\theta_n$ ,  $Trans(Z, d_n)$  o movimento de translação em torno do eixo  $Z$  em relação a  $d_n$ ,  $Trans(X, a_n)$  o movimento de translação em torno do eixo  $X$  em relação a  $a_n$  e por fim,  $Rot(X, \alpha_n)$  que é o movimento de rotação em torno do eixo  $X$  em relação a  $\alpha_n$  (Craig, 2012). Pode-se reescrever a Equação (17) como:

$${}^{n-1}T_n(\theta_n, d_n, a_n, \alpha_n) = T_Z(\theta_n)T_Z(d_n)T_X(a_n)T_X(\alpha_n) \quad (18)$$

, para descrever na forma matricial, fixa-se a matriz de rotação sobre o eixo  $Z$ , expressa por:

$$Rot(z, \theta) = R_Z(\theta) = \begin{bmatrix} \cos(\theta) & -\text{sen}(\theta) & 0 & 0 \\ \text{sen}(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

pode-se substituir a Equação (18) por:

$${}^{n-1}T_n = \dots \begin{bmatrix} c_{\theta_n} & -s_{\theta_n} & 0 & 0 \\ s_{\theta_n} & c_{\theta_n} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_n & -s\alpha_n & 0 \\ 0 & s\alpha_n & c\alpha_n & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (20)$$

sendo  $c = \text{cosseno}$  e  $s = \text{seno}$ . Através do *software* Matlab®, foram calculadas as multiplicações das matrizes, tendo como resposta à matriz genérica, exemplificada por:

$${}^{n-1}T_n = \begin{bmatrix} C(\theta_n) & -S(\theta_n)C(\alpha_n) & S(\theta_n)S(\alpha_n) & a_n C(\theta_n) \\ S(\theta_n) & C(\theta_n)C(\alpha_n) & -C(\theta_n)S(\alpha_n) & a_n S(\theta_n) \\ 0 & S(\alpha_n) & C(\alpha_n) & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$

aplica-se então a Equação (7) com o intuito deconcatenar as matrizes individuais, o que resulta em:

$${}^0T_5 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 \quad (22)$$

, utiliza-se os parâmetros de Denavit-Hartenberg na Equação (21) e aplica-se na forma da Equação (22), assim as matrizes de transformação homogênea individuais para cada par de elo-junta do manipulador que foi descrito na Figura 18 são expressas por:

$${}^0T_1 = \begin{bmatrix} c_1 - s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (23)$$

sendo  ${}^0T_1$  a matriz de transformação homogênea do frame {1} em relação ao {0}. O seguinte frame é expresso da forma:

$${}^1T_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_1 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (24)$$

em que  ${}^1T_2$  é a matriz de transformação homogênea do frame {2} em relação ao {1}. A matriz seguinte em relação ao frame {3}, é dada por:

$${}^2T_3 = \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_2 * \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (25)$$

sendo  ${}^2T_3$  a matriz de transformação homogênea do frame {3} em relação ao {2}. O frame posterior é representado por:

$${}^3T_4 = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (26)$$

sendo,  ${}^3T_4$  a matriz de transformação homogênea do frame {4} em relação ao {3}. E por fim, tem-se a matriz:

$${}^4T_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

sendo  ${}^4T_5$  a matriz de transformação homogênea do frame {5} em relação ao {4}. Assim, obtém-se:

$${}^0T_5 = \begin{bmatrix} c(\theta_{123}) & s(\theta_{123}) & 0 & a_1 c(\theta_1) + a_2 c(\theta_{12}) \\ s(\theta_{123}) & -c(\theta_{123}) & 0 & a_1 s(\theta_1) + a_2 s(\theta_{12}) \\ 0 & 0 & -1 & d_1 - d_2 * -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (28)$$

em que,  $\theta_{123}$  descreve  $(\theta_1 + \theta_2 + \theta_3)$  e  $\theta_{12}$  significa  $(\theta_1 + \theta_2)$  e  $d_2 *$  representa que  $d_2$  é variável por depender da junta prismática (P). A matriz resultante é  ${}^0T_5$ . Os cálculos foram desenvolvidos através do *software* Matlab®.

### 3.1.2 Cinemática Direta

Como sabe-se, na Seção 2.2.1 foram amostrados os dados necessários que constituem a matriz de transformação homogênea (Vidakovic,2014). Onde se tem:

$${}^{n-1}T_n = \begin{bmatrix} t_{11} & t_{12} & t_{13} & p_x \\ t_{21} & t_{22} & t_{23} & p_y \\ t_{31} & t_{32} & t_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} Rot(t_{3 \times 3}) & \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (29)$$

De acordo com Santos (2004), em uma matriz de transformação geométrica homogênea, como descrita na Equação (29), existe uma sub matriz de dimensão  $3 \times 3$  que contém os versores das três direções dos três eixos, capaz de expressar a orientação do novo sistema de coordenadas em relação ao sistema anterior. Essa tal orientação descrita por  $Rot(t_{3 \times 3})$  é o resultado de uma sequência de três rotações em torno dos três eixos distintos (Santos, 2004). Pode-se igualar as Equações (28) e (29) para extrair a cinemática direta do manipulador SCARA, conforme:

$${}^0T_5 = \begin{bmatrix} c(\theta_{123}) & s(\theta_{123}) & 0 & a_1 c(\theta_1) + a_2 c(\theta_{12}) \\ s(\theta_{123}) & -c(\theta_{123}) & 0 & a_1 s(\theta_1) + a_2 s(\theta_{12}) \\ 0 & 0 & -1 & d_1 - d_2 * -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & p_x \\ t_{21} & t_{22} & t_{23} & p_y \\ t_{31} & t_{32} & t_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (30)$$

sendo  $p_x$ ,  $p_y$  e  $p_z$  as componentes da posição do manipulador.

As componentes  $Rot(t_{3 \times 3})$  e  $p(x, y, z)_{1 \times 3}$  especificam a posição e a orientação do sistema de referência  $\{n\}$  em relação ao sistema de referência  $\{0\}$  do robô (Craig, 2012). Assim, pode-se obter:

$$\begin{aligned} p_x &= a_1 \cos(\theta_1) + a_2 (\cos(\theta_1) + \cos(\theta_2)) \\ p_y &= a_1 \sin(\theta_1) + a_2 (\sin(\theta_1) + \sin(\theta_2)) \\ p_z &= d_1 - d_2 * -d_3 \end{aligned} \quad (31)$$

e descrita através da função  $F(\theta_n)$  na forma matricial tem-se:

$$\begin{bmatrix} F(\theta_1) \\ F(\theta_2) \\ F(\theta_3) \end{bmatrix} = \begin{bmatrix} a_1 \cos(\theta_1) + a_2 (\cos(\theta_1) + \cos(\theta_2)) \\ a_1 \sin(\theta_1) + a_2 (\sin(\theta_1) + \sin(\theta_2)) \\ d_1 - d_2 * -d_3 \end{bmatrix} \quad (32)$$

sendo  $F(\theta_n)$  as posições do manipulador no espaço tridimensional  $X$ ,  $Y$  e  $Z$ , e a orientação se dá por meio dos ângulos de  $\theta_1, \theta_2$  e  $\theta_3 = d_2 *$ .

### 3.1.3 Cinemática Inversa

Implementar a cinemática inversa de um manipulador significa determinar os ângulos de juntas  $\theta_n$  a partir da posição final  $p_{x,y,z}$  previamente conhecida. A cinemática inversa que contempla o desenvolvimento deste trabalho é estimada através dos algoritmos de iterações de métodos numéricos, dos quais, a cada iteração realizada obtém-se uma resposta que aproxima ou afasta da raiz da equação até atingir um resultado satisfatório para o frame de destino. Sendo:

$$\begin{bmatrix} F(\theta_1) \\ F(\theta_2) \\ F(\theta_3) \end{bmatrix} = \begin{bmatrix} a_1 \cos(\theta_1) + a_2 (\cos(\theta_1) + \cos(\theta_2)) - x \\ a_1 \sin(\theta_1) + a_2 (\sin(\theta_1) + \sin(\theta_2)) - y \\ d_1 - d_2 * -d_3 - z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (33)$$

portanto, para aplicar o primeiro algoritmo de iteração, o método de Newton-Raphson, conforme descrito em 2.5.1.1, substitui-se os termos na Equação (13) para calcular a resposta em relação aos ângulos, assim:

$$\theta_{123} = \theta_{123} - J^{-1}(\theta_{123})F(\theta_{123}) \quad (34)$$

em que  $\theta_{123}$  representa o vetor orientação  $[\theta_1, \theta_2, \theta_3]^T$ . O cálculo da matriz Jacobiana apresenta as derivadas de primeira ordem da função, o que resulta em:

$$\begin{bmatrix} J(\theta_1) \\ J(\theta_2) \\ J(\theta_3) \end{bmatrix} = \begin{bmatrix} -a_1 \sin(\theta_1) - a_2 \sin(\theta_1 + \theta_2) & -a_2 \sin(\theta_1 + \theta_2) & 0 \\ a_1 \cos(\theta_1) + a_2 \cos(\theta_1 + \theta_2) & a_2 \cos(\theta_1 + \theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (35)$$

Para o manipulador SCARA, que apresenta duas juntas rotativas (R) representadas por  $\theta_1$  e  $\theta_2$  e uma junta prismática (P) descrita por  $\theta_3$ . O fator de relevância no descolamento da junta prismática é somente atribuído a variável  $d_2$  \* e por ser uma constante com deslocamento linear sua derivada é um, como pode ser observada na Equação (35).

O método de Levenberg-Marquardt, além da obtenção da matriz Jacobiana apresentada na Equação (35), é necessário a inserção da matriz identidade  $I$  de dimensões  $3 \times 3$  que é expressa por:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (36)$$

e do fator de amortecimento ajustável  $\lambda$ , que resulta em:

$$\theta_{123} = \theta_{123} - (J^T(\theta_{123})J(\theta_{123}) + \lambda I)^{-1}J^T(\theta_{123})F(\theta_{123}) \quad (37)$$

de maneira mais complexa, o cálculo do método de LM também envolve o computo da matriz Jacobina transposta, representada por  $J^T$  como pode ser vista na Equação (37).

O método de Halley, por apresentar uma ordem de convergência maior que os dois outros métodos propostos, possui maior dificuldade nos cálculos, pois,

necessita do computo da matriz Hessiana bem como do adicional da matriz identidade de dimensões  $3 \times 3$  conforme descrito na Equação (15), ao substituir a equação tem-se:

$$\theta_{123} = \theta_{123} - \left[ I - \frac{1}{2}L(\theta_{123}) \right]^{-1} J^{-1}(\theta_{123})F(\theta_{123}) \quad (38)$$

sendo  $L(\theta_{123})$  expressa por:

$$L(\theta_{123}) = J^{-1}(\theta_{123})[H(\theta_{123})]^T \quad (39)$$

A matriz Hessiana é calculada através da derivada de segunda ordem da função  $F(\theta_n)$ , o que resulta em:

$$\begin{bmatrix} H(\theta_1) \\ H(\theta_2) \\ H(\theta_3) \end{bmatrix} = \begin{bmatrix} -a_1 \cos(\theta_1) & -a_2 \cos(\theta_1 + \theta_2) & -a_2 \cos(\theta_1 + \theta_2) & 0 \\ -a_1 \sin(\theta_1) & -a_2 \sin(\theta_1 + \theta_2) & -a_2 \sin(\theta_1 + \theta_2) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (40)$$

#### 3.1.4 Critério de Parada e Informações

Em muitos problemas do mundo físico, não é possível obter soluções exatas. Portanto, não se pode calcular exatamente os erros associados aos métodos numéricos. Nestes casos, é preciso implementar uma função que faça em algum momento o algoritmo parar de executar.

A rotina dos métodos numéricos conforme descrita anteriormente, faz com que os métodos iterativos vão se afinando em buscado zero da função. Conforme a função se aproxima de um número cada vez menor é interessante que se crie um critério capaz de reconhecer que a função já retornou um valor relativamente pequeno que satisfaça a condição do robô. Um valor próximo de zero o bastante para que o resultado seja satisfatório quando comparado com a situação real em que o robô se encontra e assim, o algoritmo pode sair da rotina e evitar um excesso de iterações.

Como os frames são mensurados a partir de coordenadas cartesianas no espaço tridimensional que descreve a posição e orientação dos manipuladores, deve-se na versão física do robô, ser levado em consideração as limitações dos ângulos de juntas para que não ocorra a interseção dos elos, haja visto, ser um grande problema

na hora de produzir um robô de fato. Para fins didáticos, este trabalho não leva em consideração os limites dos ângulos de juntas bem como a manipulação das ferramentas do efetuador.


Para os cálculos do manipulador robótico são estipulados valores iniciais dos ângulos de juntas e das distâncias entre os elos. Essas condições iniciais, chamadas também de chutes iniciais representam a posição e orientação da qual o robô se encontra no espaço tridimensional. São de extrema importância, pois, dependendo do chute inicial a função matemática não linear pode apresentar pontos de singularidades capazes de implicar em uma convergência lenta ou rápida, ocasionar um ponto de inflexão, que salta de um ponto próximo da raiz para um ponto distante e jamais irá convergir, ou ainda, apresentar múltiplas soluções. No decorrente trabalho, foram fixados ângulos e distâncias para o manipulador afim de evitar pontos de singularidades.

A execução dos algoritmos de métodos numéricos que retorna os ângulos de juntas rotativas, podem apresentar soluções que não são cabíveis para a análise de um manipulador robótico em um plano tridimensional. Para tanto, faz-se necessário a implementação de uma função capaz de enquadrar os ângulos em apenas  $2\pi$  ou  $360^\circ$  graus, que exclui graus superiores não relevantes para se fazer as análises propostas no decorrente trabalho.

O computo das medidas temporais dos métodos números desenvolvidos, foram obtidos através da função *tic* e *toc*. Esta ferramenta é amplamente utilizada para medir o tempo de execução em algoritmos no *software* Matlab®. Todos os algoritmos propostos foram desenvolvidos e executados no *software* Matlab® na versão R2023a, instalado em um computador com o sistema operacional *Windows 10 Pro* com um poder de processamento de 64 bits. Possuí ainda, um processador Intel Core i5-2450M a 2,5GHz, capacidade de 4Gb de memória RAM e uma unidade de disco rígido HD (do inglês, *Hard Disk Drive*) SSD de 240Gb.

A fim de simplificar a execução do algoritmo no Matlab® a Figura 19 apresenta um pseudocódigo para descrever o funcionamento do código:

**Figura 19 – Pseudocódigo**

- Início
1. Declarações das variáveis  $\leftarrow (a_1, a_2, d_1, d_2, d_3, \theta_1 \text{ e } \theta_2)$
  2. Posição Desejada  $\leftarrow [p_x = 50, p_y = -80, p_z = 10]^T$
  3. Enquanto ( erro  $> 10^{-6}$  )
  4. Atualiza função do Método (NR, LM e Halley)
  5. Sai do *Loop*
  6. Enquadra em ângulos  $\rightarrow (2\pi)$
  7. Verifica Cinemática Direta  $\rightarrow$  ( Posição Desejada )
- Fim
- 

Fonte: Própria (2023).



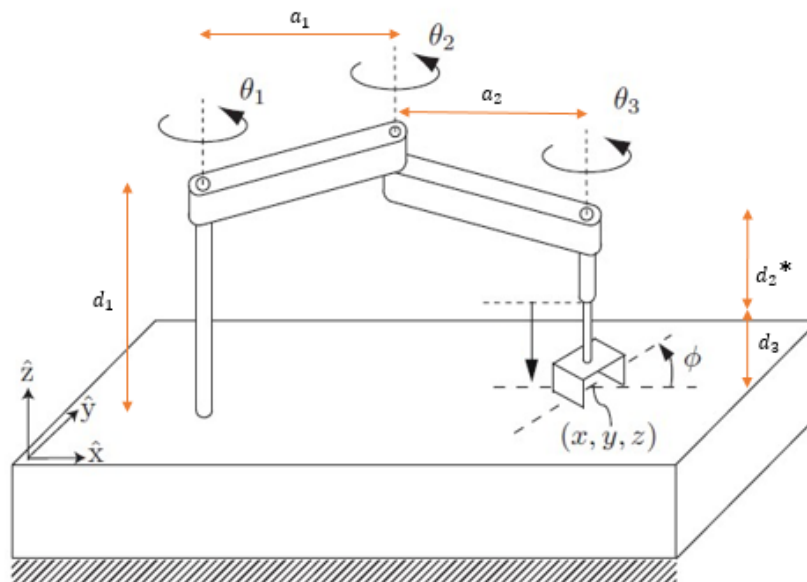
## 4 RESULTADOS E DISCUSSÕES

No decorrente capítulo serão apresentados os resultados obtidos de forma experimental simulada com a metodologia descrita no capítulo anterior. Primeiro, ocorre a definição das variáveis, cálculos e apresentação dos resultados obtidos de forma individual para cada método. De forma posterior, os métodos são comparados para obter uma análise de desempenho computacional que envolve variáveis de: tempo de execução, número de iteração e erro. Ao comparar os métodos na execução do critério de tempo, fez-se necessário a implementação de uma rotina de dez repetições, a fim de obter um resultado mais preciso em relação ao tempo.

### 4.1 Descrição das Variáveis

A fundamentação teórica das variáveis do manipulador SCARA descritas no presente trabalho, podem ser encontradas na Seção 3.1. Em uma avaliação inicial a Figura 20 abaixo, tem-se:

Figura 20 – Manipulador SCARA e Variáveis



Fonte: Adaptado de Lynch e Park (2017, p. 91)

onde, foram aplicados valores constantes iniciais para  $d_1, d_2^*, d_3$  sendo que o valor máximo do eixo  $Z$  é a distância da base do manipulador até a primeira junta rotativa,

ou seja, é igual ao comprimento de  $d_1$ . A variável  $d_2^*$  na Figura 18 representa a junta prismática (P). Para o eixo  $X$ , tem-se a distância de  $a_1$  e  $a_2$  e por fim, os valores dos ângulos de juntas de  $\theta_1, \theta_2$  conforme descrito abaixo:

**Tabela 2 – Valores para Posição e Orientação Inicial do Robô.**

$a$ (mm)	$d$ (mm)	$\theta$ (rad)
$a_1 = 200$	$d_1 = 100$	$\theta_1 = \frac{\pi}{2}$
$a_2 = 150$	$d_2^* = 50$	$\theta_2 = \frac{\pi}{4}$
-	$d_3 = 40$	-

Fonte: Própria (2023)

, os valores iniciais são dispostos conforme a Tabela 2.

Sendo  $F(x) = F(\theta)$ . Os métodos propostos englobam um fator crucial na análise, que são os erros de posição e orientação do manipulador. As condições para o critério de parada para fins experimentais foram de utilizar um erro máximo de  $1e^{-6}$ , ou seja,  $|erro|_{max} = 10^{-6}$ . Esse parâmetro é suficiente para que o manipulador tenha uma aproximação ótima em relação a real desejada, haja visto que em outras literaturas foram utilizados erros de 0,1 até 0,0001 (Wang; Cao, 2020; Omisore, 2018; Buss; Kin, 2004).

Dada a posição inicial do robô, agora é preciso estipular a posição final do manipulador para aplicar a cinemática inversa que irá trabalhar com a manipulação dos ângulos de juntas até obter o ponto final desejado. Para tal, o valor que foi adotado é de:

$$F(p_x, p_y, p_z) = \begin{bmatrix} 50 \\ -80 \\ 10 \end{bmatrix} \quad (41)$$

sendo que  $p_x, p_y$  e  $p_z$  são as respectivas coordenadas de posição do efetuador no espaço tridimensional.

## 4.2 Desempenho dos Métodos

O desempenho do método de Newton-Rapshon construído com base na cinemática inversa apresenta os seguintes valores:

**Tabela 3 – Resposta de NR**

<b>Iterações</b>	<b>Cinemática Direta (mm)</b> <b><math>(p_x, p_y, p_z)</math></b>	<b>Ângulos de Juntas (rad)</b> <b><math>(\theta_1, \theta_2)</math></b>
<b>1</b>	$[45,51929 \quad -336,11052 \quad 10,00000]^T$	$(-1,6514 \quad , \quad 6,7871)$
<b>2</b>	$[-231,73447 \quad 261,02091 \quad 10,00000]^T$	$(-4,0503 \quad , \quad 12,7156)$
<b>3</b>	$[53,243325 \quad -24,48239 \quad 10,00000]^T$	$(-18,8139 \quad , \quad 47,3006)$
<b>4</b>	$[35,19584 \quad -77,70044 \quad 10,00000]^T$	$(-19,2369 \quad , \quad 47,5256)$
<b>5</b>	$[51,22486 \quad -78,41534 \quad 10,00000]^T$	$(-19,0481 \quad , \quad 47,5852)$
<b>6</b>	$[49,98142 \quad -79,99009 \quad 10,00000]^T$	$(-19,0660 \quad , \quad 47,5898)$
<b>7</b>	$[50,00001 \quad -79,99998 \quad 10,00000]^T$	$(-19,0658 \quad , \quad 47,5900)$
<b>8</b>	$[49,99999 \quad -79,99999 \quad 10,00000]^T$	$(-19,0658 \quad , \quad 47,5900)$

Fonte: Própria (2023)

nota-se, através da Tabela 3 que o algoritmo de NR encontrou a raiz com oito iterações, um erro calculado em  $10^{-12}$  e o tempo de execução de 0,0422138 segundos.

No método de Levenberg-Marquardt leva-se em consideração a constante de amortecimento ajustável, lambda. No primeiro momento lambda é ajustada para um número relativamente pequeno de 0,1, sendo:

Tabela 4 – Resposta de LM e  $\lambda = 0, 1$ 

Iterações	Cinemática Direta ( $p_x, p_y, p_z$ )	Ângulos de Juntas (rad) ( $\theta_1, \theta_2$ )
1	[45,52070 -336,11749 13,63636] <sup>T</sup>	(-1,6414 , 6,7769)
2	[-231,71838 261,00893 10,33057] <sup>T</sup>	(-4,0379 , 12,7182)
3	[46,16355 39,17841 10,03005] <sup>T</sup>	(-18,6533 , 46,9264)
4	[80,63309 -73,12574 10,00273] <sup>T</sup>	(-18,7550 , 47,6896)
5	[52,65261 -74,36689 10,00024] <sup>T</sup>	(-19,0200 , 47,5673)
6	[49,82673 -79,94523 10,00002] <sup>T</sup>	(-19,0675 , 47,5890)
7	[50,00023 -79,99980 10,00000] <sup>T</sup>	(-19,0657 , 47,5899)
8	[50,0000 -80,00000 10,00000] <sup>T</sup>	(-19,0658 , 47,5899)

Fonte: Própria (2023)

, com um erro calculado em  $10^{-6}$  e o tempo de execução de 0,0439002 segundos, também apresentou oito iterações para encontrar a raiz. No segundo teste do algoritmo de LM o valor de lambda é elevado em 100 vezes e assim:

Tabela 5 – Resposta de LM e  $\lambda = 10$ 

Iterações	Cinemática Direta ( $p_x, p_y, p_z$ )	Ângulos de Juntas (rad) ( $\theta_1, \theta_2$ )
1	[45,65733 -336,79444 46,36363] <sup>T</sup>	(-1,6442 , 6,7705)
2	[-229,41017 259,05195 43,05785] <sup>T</sup>	(-4,1180 , 12,8710)
3	[160,55386 120,09280 40,05259] <sup>T</sup>	(-11,1563 , 29,4643)
...	...	...
184	[50,00000 -80,00000 10,00000] <sup>T</sup>	(-1,8081 , -9,8908)

Fonte: Própria (2023)

, nota-se que o valor do amortecimento ajustável possui influência de maneira drástica no desempenho e comportamento do método. Foram calculadas 184 iterações para que a função encontrasse a solução esperada, respeitando o critério de parada. Com um erro calculado em  $10^{-6}$  e o tempo de execução de 0,0479545 segundos. Nota-se

que lambda pode deixar o sistema instável, a ponto de distanciar a função da convergência.

Por fim, o método de Halley que tem como principal destaque manipulações das derivadas de segunda ordem, que proporciona maior quantidade de manipulações matemáticas e conseqüentemente maiores operações computacionais.

**Tabela 6 – Resposta de Halley**

<b>Iterações</b>	<b>Cinemática Direta (<math>p_x, p_y, p_z</math>)</b>	<b>Ângulos de Juntas (rad) (<math>\theta_1, \theta_2</math>)</b>
<b>1</b>	[376,32722 168,29472 10,00000] <sup>T</sup>	(0,5343 , - 10,4179)
<b>2</b>	[228,56857 74,05977 10,00000] <sup>T</sup>	(0,7352 , - 8,0283)
<b>3</b>	[64,42906 -61,23766 10,00000] <sup>T</sup>	(0,4828 , - 9,1869)
...	...	...
<b>27</b>	[49,99999 -80,00000 10,00000] <sup>T</sup>	(-0,2163 , - 8,9586)

Fonte: Própria (2023)

O método de Halley apresentou 27 iterações, com um tempo de execução de 0,0459902 e um erro de  $10^{-6}$ .

A terceira junta do manipulador robótico, a junta prismática (P), possui um valor de 50mm inicial. O mesmo, permanece com este valor até o final das iterações, variando somente os ângulos das juntas rotativas (R)  $\theta_1$  e  $\theta_2$ , como pode ser observado nas tabelas acima.

Para comparar os algoritmos descritos acima de forma mais eficiente foi implementada uma função na qual executou cada método dez vezes a fim de extrair uma média do tempo de execução, o que resultou em:

Tabela 7 – Média de Comparação Entre os Algoritmos NR, LM e Halley

Métodos	Tempo de Execução (s)	Nº de Iterações	Erro
<b>NR</b>	0,0424127	8	$10^{-12}$
<b>LM (<math>\lambda = 0, 1</math>)</b>	0,0434642	8	$10^{-6}$
<b>LM (<math>\lambda = 10</math>)</b>	0,0469822	184	$10^{-6}$
<b>Halley</b>	0,0454014	27	$10^{-6}$

Fonte: Própria (2023)

Os resultados obtidos através da implementação de cada algoritmo, mostra suas particularidades conforme a Tabela 7.

O NR tem o menor tempo de execução e erro amostrado. O fator de amortecimento  $\lambda$  do método de LM apresentou soluções distintas conforme o mesmo é manipulado, o que corrobora as literaturas (Alves, 2021; Benatti, 2016; Lima, 2022) em que se afirmou, que quando o fator de amortecimento cresce ele tende a direção do algoritmo de GD e ao se aproximar de zero ele tende ao algoritmo de GN. Pode-se obter uma convergência mais rápida e com menos iterações com o fator menor que um, caso contrário a equação vai para o lado oposto da raiz, dificultando ou não existindo uma solução.

O método de Halley calcula as derivadas de segunda ordem, consequentemente, o custo por iteração e requerimento de memória são maiores, portanto, apresentou um tempo maior para a execução do algoritmo. Porém, o método de Halley em geral necessita de menos iterações do que os outros métodos propostos aqui para atingir a mesma precisão (Ferreira, 2013). Devido ao alto custo computacional para problema de grande porte, o método de Halley é pouco explorado na atualidade.

## 5 CONCLUSÃO

O estudo da cinemática inversa é fundamental para a realização de tarefas de movimentação dos manipuladores robóticos. Ela, apresenta uma melhor resposta de soluções, ou seja, realiza uma economia de movimentos de parte das juntas. O presente trabalho abordou uma análise da cinemática direta e inversa de um robô tipo SCARA, onde, algoritmos de métodos numéricos buscam uma solução ótima para o sistema que foi implementado através do *software* Matlab®.

O manipulador SCARA proposto para o trabalho, foi escolhido devido a sua grande participação em obras literárias voltadas para área da robótica e implementações industriais, principalmente na indústria de eletro/eletrônico para montagem de placas.

Primeiro, foram levantadas as equações para a modelagem da cinemática direta e feita a aplicação dos parâmetros de Denavit-Hartenberg. Posteriormente, foram propostos os métodos para a solução da cinemática inversa, das quais utilizaram os métodos numéricos propostos para obter as soluções em cadeia aberta.

Na simulação, foram desenvolvidos os algoritmos de Newton-Raphson, Levenberg-Marquardt e Halley através do *software* Matlab® que apresentou os dados individuais de maneira satisfatória. Por fim, foram feitas as análises de número de iterações, tempo de execução e erro.

Existem diversos métodos numéricos iterativos como por exemplo: Broyden, máxima descida, Chebyshev, Potra-Ptak dentre outros. Para o presente trabalho os métodos abordados foram: Newton-Raphson, Levenberg-Marquardt e Halley. Quando se trabalha com problemas de otimização não linear há sempre uma preocupação com relação a escolha do melhor método a ser empregado que visa uma boa aproximação e ao mesmo tempo não tenha um custo computacional elevado e possa ser resolvido na prática.

O algoritmo de Newton-Raphson é provavelmente o mais utilizado para se obter a solução de sistemas não lineares. Nota-se que o mesmo reorganiza as equações de modo que resolva equações lineares a cada iteração. Não possui um critério de convergência geral, no qual depende totalmente da natureza da função e da precisão da aproximação inicial. O método de Levenberg-Marquardt do mesmo modo que NR, o algoritmo calcula um incremento de atualização diferenciado por um

fator amortecimento que age estabilizando o sistema e pode ser a grande questão para a convergência. No algoritmo de Halley, tem-se o cálculo dos termos derivativos de primeira e segunda ordem para alcançar a convergência. Por este motivo, ele é um método que foi pouco explorado até o presente momento. Na atualidade, o método de Halley é considerado, juntamente com o método da secante uma redescoberta na literatura. Com a evolução dos computadores este método passa a ser muito plausível para futuras deduções de cálculos de precisão.

Nota-se, que dentro das simulações, temos dois destaques. O método de LM com o fator de amortecimento menor que um, com uma média de tempo de execução de 0,0434642 segundos e o método de NR, que possui um erro relativamente menor de  $10^{-12}$  e com um tempo de execução de 0,0424127 segundos. Pode-se se dizer que o método com o melhor desempenho seria o Newton-Raphson, dentro das condições aplicadas neste trabalho.

O trabalho aqui apresentado é de grande valia para fins didáticos, pois foram abordados temas e conceitos relativos à cinemática dos manipuladores robóticos, além de, lidar com o conceito da cinemática inversa em uma cadeia de solução aberta. Espera-se que com esse trabalho, ocorra engajamento nos estudos da robótica, cálculo numérico, algoritmos computacionais e áreas relacionadas.

Como propostas de estudos futuros, pode-se adicionar a implementação do algoritmo que leva em consideração os limites de juntas do manipulador robótico, e também a possibilidade de utilizar outras soluções para a cinemática inversa como o caso de soluções analíticas, soluções *data-driven* e soluções híbridas. Ainda, a implementação do algoritmo de Halley para eventos com maiores números de iterações para análise da sua precisão.



## REFERÊNCIAS

- ABREU, Leonardo De Melo; ROSSINI, Flávio Luiz; OLIVEIRA, Luiz Fernando Pinto de. **Modelagem Cinemática de um Robô Modelo SCARA e Desenvolvimento de Aplicativo para Análise Gráfica**. In: Anais do XII Seminário de Extensão e Inovação & XXVII Seminário de Iniciação Científica e Tecnológica da UTFPR, 2022, UTFPR Santa Helena. Anais do XII Seminário de Extensão e Inovação & XXVII Seminário de Iniciação Científica e Tecnológica da UTFPR. Recife, Brasil: Even3, 2022. Disponível em: <https://doi.org/10.29327/1152426.1-2>. Acesso em: 13 set. 2023.
- ALMEIDA, M, V, G. **Estudo da Concepção de um Robô Paralelo de Três Graus de Liberdade**. Universidade Federal do Rio Grande do Sul. Porto Alegre, 2018.
- ALVES, T. W. **Novo Método Iterativo de Localização da Câmera Baseado no Conceito de Resection-Intersection**. Universidade Federal do Rio Grande do Sul. Porto Alegre, 2021.
- ARISTIDOU, A. *et al.* **Inverse Kinematics Techniques in Computer Graphics: a survey**. **Computer graphics fórum**. Helsinki: Jhon Wiley & Sons Ltd, 2017.
- BAYER, F. M. *et. al.* **Automação de Sistemas**. Santa Maria, 2011.
- BARBOSA, F. C. **Engenharia: A Máquina que Constrói o Futuro**. 13. ed. Piracanjuba: Conhecimento Livre, 2023.
- BENATTI, K. A. **O Método de Levenberg-Marquardt para o Problema de Quadrados Mínimos não Linear**. Dissertação de Mestrado. Universidade Federal do Paraná, Curitiba, 2016.
- 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.
- CONTE, S. D.; BOOR, C de. **Elementary Numerical Analysis, na Algorithmic Approach**. McGraw Hill Book Company, São Paulo, 1980.
- COCOTA, J. A. N. J. **Elementos de Robotica**. In: Apostila de engenharia de controle e automação. Universidade Federal de Ouro Preto. Ouro Preto, 2013.
- CRAIG, J. J. **Robótica**. 3. ed. São Paulo: Pearson education, 2012.
- D'ABREU, J, V, V. **Robótica Pedagógica: Percursos e Perspectivas**. In: Workshop de Robótica Educacional. São Paulo: USP, 2014.
- DELGADO, J. E. *et al.* **Fractional order passivity based adaptive controller for a robot manipulator type scara**. Mexico, 2020.
- DENAVIT, J. **Description and Displacement Analysis of Mechanics Based on the 2x2 dual Matrices**. 1 ed. University of Northwestern, Evanston, 1955.

DENAVID, J.; HARTENBERG, R. S. **A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices**. Journal of Applied Mechanics, p 215-221, New York, 1955.

FARNESE, Matheus A. **Algoritmos Evolutivos Para Solução Da Cinemática Inversa De Robôs Com Até Sete Graus De Liberdade: Um Estudo Da Aplicabilidade Em Processos De Mineração**. Minas Gerais, 2020.

FERREIRA, D. G. **Métodos de Otimização de Terceira Ordem**. Dissertação de Mestrado. Universidade Estadual de Campinas. Campinas, 2013.

FUDOLI, Fábio; BLESSA, Lucas; SOUZA, Júlio; ROSSINI, Flávio Luiz. **Modelagem da Cinemática Direta e Inversa do Robô SCARA EPSON G10-851S**. In: III Simpósio de Tecnologia e Engenharia Eletrônica - III SIMTEEL, 2016.

HALLEY, E. **Methodus Nova, Accurata And Facilis Inveniendi Radices AEquationum Quarumcumque Generalier, Sine Praevia Reductione**. Philos. p. 18:136-148, 1694.

INTERNATIONAL FEDERATION OF ROBOTICS. **World Robotics 2021 – Service Robots Report Released**. Disponível em: <https://ifr-press-releases/news/service-robots-hit-double-digit-growth-worldwide>. Acesso em: 05 out. 2023.

INTERNATIONAL FEDERATION OF ROBOTICS. **World Robotics 2023 PDF**. Disponível em: [https://ifr.org/img/worldrobotics/WR\\_Industrial\\_Robots\\_2023\\_Sources\\_and\\_Methods.pdf](https://ifr.org/img/worldrobotics/WR_Industrial_Robots_2023_Sources_and_Methods.pdf). Acesso em: 06 out. 2023.

INTERNATIONAL FEDERATION OF ROBOTICS. **World Robotics Report 2023**. Disponível em: <https://ifr.org/news/world-robotics-2023-report-asia-ahead-of-europe-and-the-americas>. Acesso em: 04 out. 2023.

INTERNATIONAL FEDERATION OF ROBOTICS. **World Robotics 2023 – Industrial Robots**. Disponível em: <https://ifr.org/industrial-robots>. Acesso em: 06 out. 2023a.

INTERNATIONAL FEDERATION OF ROBOTICS. **World Robotics 2023 – Service Robots**. Disponível em: <https://ifr.org/service-robots>. Acesso em: 05 out. 2023b.

KENWRIGTH, B. **Real-Time Character Inverse Kinematics Using The Gauss-Seidel Iterative Approximation Method**. The Fourth International Conference on Creative Content Technologies. Nice, 2012.

LEVENBERG, K. **A Method For The Solution Of Certain Problems In Least Squares**. Quarterly of Applied Mathematics, 2. p. 164-168, 1944.

LIMA, G. W. **Ajuste Ótimo por Levenberg-Marquardt de Métodos de Previsão para Iniciação de Trinca**. Dissertação de Mestrado. Pontifícia Universidade Católica. Rio de Janeiro, 2022.

LYNCH, K. M.; PARK, F.C. **Modern Robotics**. 1. ed. Cambridge University Press, 2017.

MATARIC, Maja J. **Introdução a Robótica**. 1. ed. São Paulo: Blucher, 2014.

MARTINES, Francine; MELO, Larissa; CHICHANOSKI, Gustavo; MUNARINI, Bruno; ROSSINI, Flávio Luiz. **Modelagem Matemática da Cinemática Direta do Robô Manipulador SCARA Modelo G-10/G-20**. In: V Ciclo de Palestras: Perspectivas Matemáticas - V CIPEM, 2015.

NORTON, Robert L. **Cinemática e Dinâmica dos Mecanismos**. 1. ed. São Paulo: Mc Graw Hill, 2010.

OLIVEIRA, Luiz Fernando Pinto; ROSSINI, Flavio Luiz. **Modeling, Simulation and Analysis of Locomotion Patterns for Hexapod Robots**. IEEE Latin America Transactions, v. 16, n. 2, p. 375-383, fev. 2018. Disponível em: <https://doi.org/10.1109/tla.2018.8327389>. Acesso em: 09 set. 2023.

OLIVEIRA, Luiz Fernando Pinto de; ROSSINI, Flávio Luiz; SILVA, Manuel Fernando; MOREIRA, Antonio Paulo. **Modeling, Simulation and Implementation of Locomotion Patterns for Hexapod Robots**. In: 2020 IEEE CONGRESO BIENAL DE ARGENTINA (ARGENCON), 2020, Resistencia, Argentina. 2020 IEEE Congreso Bienal de Argentina (ARGENCON). [S. l.]: IEEE, 2020. Disponível em: <https://doi.org/10.1109/argencon49523.2020.9505570>. Acesso em: 13 set. 2023.

OMISORE, O. M.; Han, S. *et al.* **Deeply-Learnt Damped Least-Squares (DL-DLS) Method for Inverse Kinematics of Snake-Like Robots**. Neural networks: the official journal of the International Neural Network Society, 107, 34–47. 2018. Disponível em: <https://doi.org/10.1016/j.neunet.2018.06.018>. Acesso em: 14 out. 2023.

PENHA, Bruno Schuavab; QUEIROZ, Mariana Emer; ROSSINI, Flávio Luiz. **Modelagem e Análise da Cinemática Direta e Inversa de Manipulador Robótico com Cinco Juntas Rotativas**. In: III Simpósio de Tecnologia e Engenharia Eletrônica - III SIMTEEL, 2016.

REIS, R. A. **Implementação de um Código Computacional Destinado a Solução de Sistemas de Equações Lineares e Não Lineares via Métodos Iterativos: Aplicações em Treliças Metálicas**. Ouro Preto, 2019.

ROMANO, V. F.; DUTRA, M. S. **Introdução a Robótica Industrial: Aplicação na Indústria de Manufatura e de Processo**. São Paulo: Edgard Blücher, 2002.

ROSSINI, Flávio Luiz; ABREU, Leonardo de Melo; OLIVEIRA, Luiz Fernando Pinto de. **Modelagem Cinemática, Planejamento e Geração de Trajetória para um Robô Modelo SCARA Através de Aplicativo para Análise Gráfica**. In: ROSSINI, Flávio Luiz; ABREU, Leonardo de Melo; OLIVEIRA, Luiz Fernando Pinto de. Tópicos Especiais em Engenharia: inovações e avanços tecnológicos 7. [S. l.]: Aya Editora, 2023. p. 43-58. ISBN 9786553792999. Disponível em: <https://doi.org/10.47573/aya.5379.2.211.4>. Acesso em: 13 set. 2023.

ROSSINI, Flávio Luiz; LIMA, Bruno Suracci de; CORRÊA, João Henrique Dias; LOPES, João Marcos Pericaro; BARBOSA, Reginaldo Ferreira de Sousa; BARROZO, Yuri Ruzzene. **Modeling, Simulation, Motion Trajectory Planning and Nonlinear Control in the Joint Space of the Manipulator Robot SCARA T3 401SS Manufacturer Epson**. In: ROSSINI, Flávio Luiz *et al.* Development and its applications in scientific knowledge. [S. l.]: Seven Editora, 2023a. Disponível em: <https://doi.org/10.56238/devopinterscie-248>. Acesso em: 15 set. 2023

RUGGIERO, M. A. G.; LOPES, V. L. R. **Cálculo Numérico: Aspectos Teóricos e Computacionais**. 2ed. São Paulo: Pearson Makron Books, 2014.

SANCHES, Hugo Eduardo; ROSSINI, Flávio Luiz. **Modelagem e Controlador um Manipulador Robótico**. Journal of Exact Sciences - JES, v. 9, n. 1, p. 5-13, 8 abr.2016.Disponível

em:[https://www.mastereditora.com.br/periodico/20160728\\_073843.pdf](https://www.mastereditora.com.br/periodico/20160728_073843.pdf). Acesso em: 10 set. 2023.

SANTOS, V. M. F. **Robótica Industrial**. Universidade de Aveiro, 2004.

SCHILLING, Robert J. **Fundamentals of Robotics: Analysis and Control**. EnglewoodCliffs Prentice Hall, 1990.

SILVA, G. G. **Solução de Cinemática de Plataforma de Stewart em Aprendizado de Máquina**. Universidade Estadual de Campinas, 2020.

SILVA, R. B; BLIKSTEIN, P. **Robótica Educacional: Experiências inovadoras na Educação Brasileira**. Porto Alegre: Penso, 2020.

SICILIANO, B; SCIAVICCO, L; VILLANI, L; *et al.* **Robotics: modelling, planning and control**. Advanced textbooks in control and signal processing. Napoli, 2010.

SOUZA, E. A. **Métodos Iterativos Para Problemas Não Lineares**. Volta Redonda, 2015. Dissertação.

SOUZA, Eber Delgado de; ROSSINI, Flávio Luiz; OLIVEIRA, Luiz Fernando Pinto de. **Desenvolvimento de um aplicativo no ambiente app designer do software matlab® para planejamento de trajetória do robô puma 560**. In: SOUZA, Eber Delgado de; ROSSINI, Flávio Luiz; OLIVEIRA, Luiz Fernando Pinto de. Engenharia elétrica e de computação: docência, pesquisa e inovação tecnológica. [S. l.]: Atena Editora, 2023. p. 87-109. ISBN 9786525809465. Disponível em: <https://doi.org/10.22533/at.ed.4652316018>. Acesso em: 28 set. 2023.

SPONG, M. W; VIDYASARGAR, M. **Robot Modeling and Control**. New York: John Wiley & Sons, 2005.

TAYLOR, J. R. **Classical Mechanics**. University Science Books, 2005.

THE INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/TC 299 Robotics 2021**. Disponível em: <https://committee.iso.org/home/tc299>. Acesso em: 05 out. 2023.

VIDAKOVIC, J. Z. *et al.* **Advanced quaternion forward kinematics algorithm including overview of different methods for robot kinematics**. Belgrado, 2014.

WANG, X; CAO, J.; LIU, X.; CHEN, L.; HU, H. **An Enhanced Step-Size Gaussian Damped Least Squares Method Based on Machine Learning for Inverse Kinematics of Redundant Robots**. in IEEE Access, vol. 8, pp. 68057-68067, 2020. Disponível em: doi: 10.1109/ACCESS.2020.2986421. Acesso em: 15 out. 2023.