

UNIVERSIDADE ESTADUAL PAULISTA

"JÚLIO DE MESQUITA FILHO"

Programa de Pós-Graduação em Engenharia Elétrica

Sorocaba / São João da Boa Vista

LAÉRCIO RAIMUNDO FILHO

**Deep learning: uma abordagem baseada em redes neurais profundas para estimativa da
cinemática inversa aplicada na robótica**

São João da Boa Vista

2025

Laércio Raimundo Filho

Deep learning: uma abordagem baseada em redes neurais profundas para estimativa da cinemática inversa aplicada na robótica

Dissertação apresentada ao Conselho do Programa de Pós-Graduação em Engenharia Elétrica, interunidades entre o Instituto de Ciência e Tecnologia de Sorocaba e a Faculdade de Engenharia de São João da Boa Vista, Universidade Estadual Paulista "Júlio de Mesquita Filho", como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Área de Concentração: Automação

Linha de Pesquisa: Sistemas Mecatrônicos

Orientador: Prof. Dr. Wilian Miranda dos Santos

São João da Boa Vista

2025

R153d

Raimundo Filho, Laércio

Deep learning: uma abordagem baseada em redes neurais profundas para estimativa da cinemática inversa aplicada na robótica / Laércio

Raimundo Filho. -- São João da Boa Vista, 2025

114 p. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP),
Faculdade de Engenharia, São João da Boa Vista

Orientador: Wilian Miranda dos Santos

1. Manipuladores (Mecanismo). 2. Cinemática. 3. Redes neurais
(Computação). 4. Inteligência artificial. 5. Robótica. I. Título.

IMPACTO POTENCIAL DESTA PESQUISA

Esta dissertação investigou, por meio de uma Revisão Sistemática da Literatura e de experimentos computacionais, o uso de redes neurais profundas para estimar a cinemática inversa aproximada do manipulador industrial ABB IRB120 a partir da pose cartesiana do efetuador final. Os resultados obtidos demonstram que, após o tratamento adequado da redundância associada ao punho esférico, o modelo é capaz de aprender um mapeamento consistente entre posição, orientação e variáveis articulares. A abordagem proposta apresenta potencial aplicação em planejamento de movimento, controle robótico e geração de estimativas iniciais para métodos numéricos de refinamento da cinemática inversa. Além disso, a metodologia baseia-se na geração de *datasets* sintéticos por meio da cinemática direta e no treinamento de modelos reproduzíveis em hardware computacional acessível, o que amplia sua viabilidade para uso em ambientes acadêmicos, laboratoriais e industriais. Nesse contexto, os resultados obtidos, aliados à análise sistemática da literatura, indicam que a garantia de unicidade do mapeamento constitui um fator determinante para o desempenho de modelos baseados em aprendizado profundo aplicados à cinemática inversa, contribuindo para reduzir a dependência de deduções analíticas complexas, acelerar rotinas de estimação articular e favorecer o desenvolvimento de soluções mais flexíveis para automação, robótica industrial e sistemas mecatrônicos inteligentes.

POTENTIAL IMPACT OF THIS RESEARCH

This dissertation investigated, through a Systematic Literature Review and computational experiments, the use of deep neural networks to estimate the approximate inverse kinematics of the ABB IRB120 industrial manipulator from the Cartesian pose of the end-effector. The obtained results demonstrate that, after properly addressing the redundancy associated with the spherical wrist, the model is able to learn a consistent mapping between position, orientation, and joint variables. The proposed approach shows potential applications in motion planning, robotic control, and in providing initial estimates for numerical inverse kinematics refinement methods. Furthermore, the methodology is based on the generation of synthetic *datasets* through forward kinematics and on the training of reproducible models using accessible computational hardware, which enhances its feasibility for academic, laboratory, and industrial environments. In this context, the obtained results, combined with the systematic analysis of the literature, indicate that ensuring mapping uniqueness is a key factor for the performance of deep learning-based models applied to inverse kinematics, contributing to reduce the dependence on complex analytical derivations, accelerate joint estimation routines, and support the development of more flexible solutions for automation, industrial robotics, and intelligent mechatronic systems.



CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: Deep Learning: Uma abordagem baseada em redes neurais profundas para estimativa da cinemática inversa aplicada na robótica

AUTOR: LAÉRCIO RAIMUNDO FILHO

ORIENTADOR: WILIAN MIRANDA DOS SANTOS

Aprovado como parte das exigências para obtenção do Título de Mestre em Engenharia Elétrica, área: Automação pela Comissão Examinadora:

Prof. Dr. WILIAN MIRANDA DOS SANTOS (Participação Virtual)
Departamento de Engenharia Aeronáutica / UNESP / Câmpus de São João da Boa Vista - FESJBV

Prof. Dr. ROBERTO SANTOS INOUE (Participação Virtual)
Departamento de Computação / Universidade Federal de São Carlos

Prof. Dr. SAMUEL LOURENÇO NOGUEIRA (Participação Virtual)
Departamento de Engenharia Elétrica / Universidade Federal de São Carlos - UFSCar

São João da Boa Vista, 26 de novembro de 2025.

AGRADECIMENTOS

Agradeço à Universidade Estadual Paulista – UNESP, em particular aos campi de São João da Boa Vista – SP e de Sorocaba, pela qualidade de ensino, apoio financeiro e pelo excelente trabalho prestado por todos os funcionários que fazem parte desta instituição. Minhas necessidades acadêmicas sempre foram atendidas com muito profissionalismo e dedicação. Gratidão!

Agradeço ao Programa de Desenvolvimento Institucional (PDI), por meio da Pró-Reitoria de Pós-Graduação da UNESP (PROPG), pelo apoio concedido à pesquisa e à formação acadêmica, por meio da bolsa de mestrado.

Ao meu orientador, deixo minha imensa gratidão pela transferência de conhecimento, apoio, paciência e dedicação em todo esse processo de aprendizado. Muito obrigado por me acompanhar nessa trajetória de conhecimento e descobertas tecnológicas!

A Deus e aos meus familiares, obrigado por tudo!

RESUMO

A modelagem cinemática de manipuladores robóticos tem recebido grande atenção devido à necessidade crescente de soluções eficientes para posicionamento e orientação precisos. Nesse contexto, este trabalho realiza uma Revisão Sistemática da Literatura com o objetivo de identificar e analisar arquiteturas de redes neurais artificiais aplicadas à solução da cinemática inversa de manipuladores robóticos, bem como aplica aprendizado de máquina supervisionado para estimar a cinemática inversa aproximada do manipulador ABB IRB120. Enquanto a cinemática direta pode ser resolvida de forma relativamente simples por métodos clássicos, como a parametrização de Denavit–Hartenberg, a cinemática inversa permanece um problema significativamente mais complexo, especialmente em manipuladores com múltiplos graus de liberdade ou redundância estrutural. Para a etapa experimental, foi utilizado um conjunto de dados sintético gerado via cinemática direta a partir da amostragem aleatória das variáveis articulares. Uma Rede Neural Profunda (DNN) foi treinada para regressão multivariada, recebendo como entrada a pose cartesiana completa do efetuador (posição e orientação em quatérnions) e produzindo como saída os ângulos articulares correspondentes. As métricas Erro Absoluto Médio (MAE) e Raiz do Erro Quadrático Médio (RMSE) foram empregadas para avaliar o desempenho após a desnormalização das saídas. Os resultados mostraram que o uso irrestrito de todo o espaço articular, especialmente das juntas do punho esférico, gera um mapeamento não unívoco que inviabiliza o aprendizado. A restrição da junta q_6 eliminou a principal redundância associada ao *wrist twist*, permitindo que a rede aprenda um mapeamento consistente, reduzindo significativamente os erros angulares. Assim, demonstra-se que abordagens baseadas em aprendizado profundo podem atuar como estimadores aproximados de cinemática inversa, desde que o espaço articular seja tratado de forma a garantir unicidade do mapeamento, mostrando potencial para aplicações em planejamento de movimento, controle e inicialização de métodos numéricos de refinamento.

PALAVRAS-CHAVE: manipuladores (mecanismo); cinemática; redes neurais (computação); inteligência artificial; robótica.

ABSTRACT

The kinematic modeling of robotic manipulators has received significant attention due to the increasing demand for efficient solutions for precise positioning and orientation. In this context, this work conducts a Systematic Literature Review to identify and analyze artificial neural network architectures applied to solving the inverse kinematics problem of robotic manipulators, as well as applies supervised machine learning to estimate the approximate inverse kinematics of the ABB IRB120 manipulator. While forward kinematics can be solved relatively easily using classical methods such as Denavit–Hartenberg parameterization, inverse kinematics remains a significantly more complex problem, especially for manipulators with multiple degrees of freedom or structural redundancy. For the experimental stage, a synthetic dataset was generated via forward kinematics based on random sampling of joint variables. A Deep Neural Network (DNN) was trained for multivariate regression, taking as input the complete Cartesian pose of the end-effector (position and quaternion-based orientation) and producing as output the corresponding joint angles. The Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) metrics were used to evaluate performance after output denormalization. The results showed that using the full joint space, particularly the spherical wrist joints, leads to a non-unique mapping that hinders the learning process. Constraining joint q_6 eliminated the main redundancy associated with the *wrist twist*, enabling the network to learn a consistent mapping and significantly reducing angular errors. These findings demonstrate that deep learning-based approaches can act as approximate inverse kinematics estimators, provided that the joint space is properly constrained to ensure mapping uniqueness, showing potential for applications in motion planning, control, and initialization of numerical refinement methods.

KEYWORDS: manipulators (mechanism); kinematics; neural networks (computer science); artificial intelligence; robotics.

LISTA DE ILUSTRAÇÕES

Figura 1 – Processo de revisão sistemática da literatura	20
Figura 2 – Processo de seleção de artigos	25
Figura 3 – Distribuição percentual das publicações por base de dados	28
Figura 4 – Distribuição dos artigos identificados e selecionados por base de dados	29
Figura 5 – Distribuição temporal das publicações incluídas na revisão sistemática (2018–2025)	30
Figura 6 – Distribuição das arquiteturas de redes neurais identificadas na literatura	42
Figura 7 – Frequência dos ambientes de simulação utilizados nos estudos	48
Figura 8 – Distribuição dos paradigmas de aprendizado de máquina	50
Figura 9 – Principais paradigmas de aprendizado de máquina	66
Figura 10 – Modelagem robótica — Denavit-Hartenberg	69
Figura 11 – Estrutura de uma rede neural artificial simples.	74
Figura 12 – Representação de uma conexão sináptica em um <i>Perceptron</i>	74
Figura 13 – Processo genérico de pré-processamento, treino e validação de redes neurais artificiais.	75
Figura 14 – Comparação entre rede neural superficial e rede neural profunda.	76
Figura 15 – Robô ABB IRB 120	77
Figura 16 – Pipeline do método proposto para geração do dataset, treinamento da DNN e inferência da cinemática inversa aproximada.	80
Figura 17 – Arquitetura da rede neural profunda proposta.	86
Figura 18 – Predição no conjunto de teste com todas as juntas percorrendo seu intervalo completo de operação.	89
Figura 19 – Histórico do MSE ao longo do treinamento.	90
Figura 20 – Histórico do MAE ao longo do treinamento.	90
Figura 21 – Evolução do MSE durante o treinamento.	93
Figura 22 – Evolução do MAE durante o treinamento.	93
Figura 23 – Valores reais versus preditos das juntas no conjunto de teste.	94

LISTA DE TABELAS

1 – Questões de pesquisa (Q)	21
2 – Estratégia PICOC	22
3 – Critérios de inclusão e exclusão	24
4 – Processo de seleção dos estudos	24
5 – Formulário de extração dos dados	26
6 – Questões para Avaliação da Qualidade (QA)	27
7 – Artigos selecionados para a pesquisa	31
8 – Estatísticas descritivas da avaliação de qualidade	38
9 – Síntese das questões de pesquisa e principais resultados	39
10 – Principais arquiteturas de redes identificadas na literatura	40
11 – Tipos de sistemas robóticos utilizados nos estudos analisados	44
12 – Robôs industriais identificados nos estudos analisados	45
13 – Ambientes experimentais utilizados nos estudos analisados	46
14 – Principais simuladores e ambientes utilizados nos estudos analisados	47
15 – Paradigmas de aprendizado de máquina analisados na literatura	49
16 – Taxonomia dos métodos de geração de <i>datasets</i> para cinemática inversa	52
17 – Distribuição dos tipos de <i>datasets</i> utilizados nos estudos analisados	53
18 – Métricas de avaliação e desempenho dos estudos analisados	54
19 – Principais lacunas da literatura identificadas	58
20 – Codificação dos estudos selecionados na revisão sistemática	60
21 – Parâmetros de Denavit-Hartenberg (D-H) do manipulador ABB IRB 120	70
22 – Parâmetros cinemáticos — ABB IRB 120	78
23 – Métricas de erro angular por junta no espaço articular completo (graus).	91
24 – Métricas de erro angular por junta com restrição de ramo (graus).	94

LISTA DE ABREVIATURAS E SIGLAS

AI	Inteligência Artificial – <i>Artificial Intelligence</i>
ANN	Rede Neural Artificial – <i>Artificial Neural Network</i>
DNN	Rede Neural Profunda – <i>Deep Neural Network</i>
DL	Aprendizado Profundo – <i>Deep Learning</i>
ML	Aprendizado de Máquina – <i>Machine Learning</i>
MLP	Perceptron Multicamadas – <i>Multilayer Perceptron</i>
RNN	Rede Neural Recorrente – <i>Recurrent Neural Network</i>
LSTM	Memória de Longo Prazo Curta – <i>Long Short-Term Memory</i>
GRU	Unidade Recorrente Conectada – <i>Gated Recurrent Unit</i>
RBFNN	Rede Neural de Função de Base Radial – <i>Radial Basis Function Neural Network</i>
FK	Cinemática Direta – <i>Forward Kinematics</i>
IK	Cinemática Inversa – <i>Inverse Kinematics</i>
DH	Parâmetros de Denavit-Hartenberg – <i>Denavit-Hartenberg</i>
DoF	Graus de Liberdade – <i>Degrees of Freedom</i>
HTM	Matriz de Transformação Homogênea – <i>Homogeneous Transformation Matrix</i>
LQR	Regulador Quadrático Linear – <i>Linear Quadratic Regulator</i>
LQG	Gaussiano Quadrático Linear – <i>Linear Quadratic Gaussian</i>
MPC	Controle Preditivo Baseado em Modelo – <i>Model Predictive Control</i>
SMC	Controle por Modos Deslizantes – <i>Sliding Mode Control</i>
GA	Algoritmo Genético – <i>Genetic Algorithm</i>
PSO	Otimização por Enxame de Partículas – <i>Particle Swarm Optimization</i>
MAE	Erro Absoluto Médio – <i>Mean Absolute Error</i>

MSE	Erro Quadrático Médio – <i>Mean Squared Error</i>
RMSE	Raiz do Erro Quadrático Médio – <i>Root Mean Squared Error</i>
MIMO	Múltiplas Entradas e Múltiplas Saídas – <i>Multi-Input Multi-Output</i>
SISO	Entrada Única e Saída Única – <i>Single-Input Single-Output</i>
ROS	Sistema Operacional de Robôs – <i>Robot Operating System</i>
UNESP	Universidade Estadual Paulista

SUMÁRIO

1	INTRODUÇÃO	14
1.1	– Objetivos	15
1.2	– Justificativa	16
1.3	– Publicação decorrente deste trabalho	18
1.4	– Estrutura da dissertação	18
2	REVISÃO SISTEMÁTICA DA LITERATURA	19
2.1	– Metodologia da Revisão Sistemática da Literatura	19
2.1.1	– <i>Planejamento</i>	20
2.1.1.1	– <i>Questões de pesquisa</i>	20
2.1.1.2	– <i>Protocolo de revisão sistemática</i>	22
2.1.1.3	– <i>Avaliação do protocolo de revisão</i>	22
2.1.2	– <i>Condução</i>	22
2.1.2.1	– <i>Estratégia de busca</i>	23
2.1.2.2	– <i>Critérios de seleção dos estudos: inclusão e exclusão</i>	24
2.1.2.3	– <i>Processo de seleção dos estudos</i>	24
2.1.2.4	– <i>Processo de extração e síntese dos dados</i>	26
2.1.2.5	– <i>Avaliação de qualidade</i>	26
2.2	– Resultados e Discussões	28
2.2.1	– <i>Respostas e discussões referentes às questões de pesquisa (QR)</i>	38
3	FUNDAMENTAÇÃO TEÓRICA	65
3.1	– Cinemática direta e inversa de robôs manipuladores	67
3.1.1	– <i>Propriedades cinemáticas diretas</i>	68
3.1.2	– <i>Propriedades cinemáticas inversas</i>	72
3.2	– Introdução às redes neurais artificiais	73
3.2.1	– <i>Processo geral de treinamento de redes neurais</i>	75
3.3	– Conceitos de aprendizado profundo	75
4	MATERIAIS E MÉTODOS	77
4.1	– Materiais	77
4.1.1	– <i>O robô em estudo: ABB IRB 120</i>	77

4.1.2 – Ferramentas computacionais	78
4.2 – Métodos	79
4.2.1 – Geração do dataset sintético para treinamento da rede neural	81
4.2.1.1 – Definição dos limites articulares e restrição de ramo	81
4.2.1.2 – Amostragem massiva no espaço de juntas	81
4.2.1.3 – Normalização do espaço de pose	82
4.2.1.4 – Balanceamento geométrico do espaço operacional	82
4.2.1.5 – Construção final do dataset	83
4.2.2 – Desenvolvimento do modelo de rede neural	84
4.2.2.1 – Entradas, saídas e motivação da arquitetura	84
4.2.2.2 – Arquitetura da rede neural profunda	84
4.2.3 – Treinamento, validação e métricas de desempenho	86
5 RESULTADOS E DISCUSSÃO	89
5.1 – Resultados com o espaço articular completo	89
5.1.1 – Predição no conjunto de teste	89
5.1.2 – Evolução do treinamento	90
5.1.3 – Métricas quantitativas	91
5.1.4 – Análise dos resultados	91
5.2 – Considerações sobre a junta q_6 e a redundância do punho esférico	91
5.2.1 – Evolução do treinamento com restrição de ramo	92
5.2.2 – Desempenho no espaço angular	93
5.2.3 – Métricas de erro angular	94
5.3 – Correlações entre achados da literatura e os experimentos realizados	94
6 CONCLUSÃO	97
REFERÊNCIAS	99
APÊNDICE A – GERAÇÃO DO DATASET SINTÉTICO PARA O ABB IRB120	105
APÊNDICE B – TREINAMENTO DA REDE NEURAL PROFUNDA	109
APÊNDICE C – CODIGO DE INFERENCIA PARA A CINEMATICA IN- VERSA APROXIMADA	114

1 INTRODUÇÃO

A determinação da cinemática inversa continua sendo um dos problemas centrais e mais desafiadores da robótica manipuladora, especialmente em sistemas com alta não linearidade, múltiplas soluções possíveis ou redundância geométrica (DAYA; KHAWANDI; AKOUM, 2010). Embora a cinemática direta possa ser formulada de maneira sistemática por métodos clássicos, como a convenção de Denavit–Hartenberg, a cinemática inversa exige a resolução de relações acopladas entre posição e orientação, podendo resultar em soluções múltiplas, inexistentes ou numericamente instáveis. Esses fatores tornam seu cálculo um ponto crítico em aplicações industriais, controle em tempo real e planejamento de trajetórias (POLYDOROS; NALPANTIDIS, 2016).

No manipulador ABB IRB120, composto por seis graus de liberdade, o problema é ainda mais complexo devido à presença de um punho esférico formado pelas juntas q_4 , q_5 e q_6 . Esse arranjo introduz redundância de orientação, possibilitando que diferentes configurações articulares produzam a mesma pose cartesiana do efetuador. Tal característica afeta diretamente métodos baseados em aprendizado supervisionado, que pressupõem que cada entrada do conjunto de dados possua uma única saída consistente.

Paralelamente, técnicas de Aprendizado de Máquina e Redes Neurais Artificiais (ANNs) têm sido exploradas como alternativas para aproximar soluções de cinemática inversa, reduzindo o custo computacional e evitando a necessidade de derivações analíticas complexas. Redes neurais são modelos computacionais capazes de representar funções altamente não lineares, aprendendo mapeamentos entrada-saída a partir de exemplos (YANG; MOGHAVVEMI; TOLMAN, 2000). Sua aplicação em robótica tem demonstrado potencial em tarefas de regressão cinemática, desde manipuladores simples até arquiteturas mais complexas.

Trabalhos recentes mostram resultados promissores. Em Šegota et al. (2021), um MLP foi treinado com dados sintéticos do IRB120, obtendo estimativas compatíveis para parte do espaço articular. Em Car et al. (2021), métodos de *genetic programming* foram utilizados para gerar equações simplificadas da cinemática inversa, alcançando erros inferiores a $0,5^\circ$. Outros estudos, como Moradinezhad, Ghasemi e Hosseini (2017), demonstram que ANNs são eficazes para manipuladores com baixa complexidade geométrica. Entretanto, Dash, Choudhury e Senapati (2019) evidenciam limitações importantes: erros elevados, sobretudo em juntas internas, quando o conjunto de dados não cobre adequadamente o espaço articular ou quando o treinamento não incorpora mecanismos para lidar com redundâncias.

Diante desse cenário, observa-se na literatura uma diversidade significativa de abordagens baseadas

em redes neurais para a resolução da cinemática inversa, frequentemente apresentadas de forma isolada e com diferentes configurações experimentais. Essa heterogeneidade dificulta a comparação entre métodos e a identificação de tendências consolidadas, evidenciando a necessidade de uma análise sistemática da literatura que permita mapear, organizar e compreender as principais arquiteturas, estratégias e limitações reportadas na área.

Nesse contexto, este trabalho combina uma análise sistemática da literatura com o desenvolvimento de uma abordagem baseada em aprendizado supervisionado para aproximar a cinemática inversa do manipulador ABB IRB120. Diferentemente de bases externas, todo o conjunto de dados foi gerado de forma sintética a partir do modelo de cinemática direta do robô, utilizando amostragem aleatória das variáveis articulares e representação da orientação por quatérnions. O estudo examina dois cenários distintos: (i) aprendizado no espaço articular completo e (ii) aprendizado em um espaço reduzido obtido pela fixação da junta q_6 , removendo a redundância de orientação do punho esférico. A comparação entre esses cenários permite avaliar de forma sistemática a influência da redundância geométrica no desempenho da rede neural profunda.

1.1 Objetivos

O objetivo deste trabalho é investigar e analisar, por meio de uma Revisão Sistemática da Literatura, as arquiteturas de redes neurais artificiais aplicadas à solução da cinemática inversa de manipuladores robóticos, bem como desenvolver e avaliar um modelo baseado em Aprendizado de Máquina Supervisionado para aproximar a solução da cinemática inversa do manipulador ABB IRB120 a partir de sua pose cartesiana, analisando o impacto da redundância do punho esférico sobre o desempenho do processo de aprendizado.

Os objetivos específicos são:

- conduzir uma Revisão Sistemática da Literatura para identificar arquiteturas de redes neurais artificiais aplicadas à cinemática inversa de manipuladores robóticos;
- analisar os tipos de arquiteturas, datasets, métricas, ambientes experimentais, desafios e lacunas reportados na literatura;
- gerar conjuntos de dados sintéticos por meio da cinemática direta do ABB IRB120, considerando tanto o espaço articular completo quanto versões restritas que eliminam a redundância do punho;
- projetar uma rede neural profunda para realizar o mapeamento

$$f : \mathbb{R}^7 \rightarrow \mathbb{R}^n,$$

em que o vetor de entrada é composto pela posição e orientação do efetuador final,

$$(x, y, z, q_w, q_x, q_y, q_z),$$

sendo (x, y, z) a posição cartesiana e (q_w, q_x, q_y, q_z) o quatérnion de orientação. A rede estima como saída os ângulos articulares relevantes do manipulador, com $n = 6$ no espaço articular completo e $n = 5$ quando a junta q_6 é fixada;

- treinar e validar o modelo empregando normalização, regularização e estratégias de mitigação de sobreajuste, como *dropout*, L2 e *early stopping*;
- avaliar o desempenho por meio das métricas MSE, RMSE e MAE, analisando separadamente os cenários com e sem redundância;
- implementar uma função de inferência que permita estimar os ângulos articulares a partir de uma pose arbitrária, constituindo uma solução aproximada para a cinemática inversa do ABB IRB120.

1.2 Justificativa

A cinemática inversa (Inverse Kinematics – IK) constitui um dos problemas fundamentais no controle de manipuladores robóticos, sendo responsável por determinar as configurações das juntas necessárias para que o efetuador final alcance uma posição e orientação desejadas no espaço cartesiano (MALIK et al., 2022; RICE; AZAB; SAAD, 2024). Essa capacidade é essencial para a execução de tarefas que exigem elevada precisão e coordenação de movimento, como operações de manufatura industrial, montagem automatizada, procedimentos cirúrgicos assistidos por robôs e aplicações em ambientes remotos ou hostis, como a exploração espacial (JOSHI et al., 2025; SHAREEF, 2024). Nesse contexto, a resolução eficiente do problema de cinemática inversa permite a tradução de comandos definidos no espaço operacional em ações executáveis no espaço das juntas, possibilitando que manipuladores robóticos interajam de forma eficaz e segura com o ambiente (NUGROHO; YUNIARNO; PURNOMO, 2023; WAGAA; KALLEL; MELLOULI, 2023).

Apesar de sua relevância, a resolução da cinemática inversa apresenta desafios significativos quando abordada por métodos tradicionais. As soluções analíticas, embora precisas e computacionalmente eficientes, dependem fortemente da estrutura geométrica do manipulador e frequentemente não admitem soluções fechadas para robôs com arquiteturas complexas ou com múltiplos graus de liberdade (DoF) (GUMUS; KALYONCU, 2025; WAGAA; KALLEL; MELLOULI, 2023). Por outro lado, métodos

numéricos e iterativos, como aqueles baseados na matriz Jacobiana, apresentam maior flexibilidade, porém podem sofrer com problemas de convergência, elevada complexidade computacional e sensibilidade às condições iniciais, além de dificuldades em lidar com singularidades e mínimos locais (AYDOGMUS; BOZTAS, 2024; RICE; AZAB; SAAD, 2024; ZHAO et al., 2024). Tais limitações tornam essas abordagens menos adequadas para aplicações que exigem respostas rápidas e robustas em ambientes dinâmicos ou em sistemas com elevada redundância cinemática (ARAVINDDHAKSHAN; APTE; AKASH, 2021; SEMWAL et al., 2025).

Diante dessas limitações, abordagens baseadas em Inteligência Artificial, especialmente Redes Neurais Artificiais (Artificial Neural Networks – ANN) e técnicas de Deep Learning, têm emergido como alternativas promissoras para a estimativa da cinemática inversa. Esses métodos são capazes de aprender relações altamente não lineares entre posições cartesianas e configurações articulares a partir de dados, dispensando a modelagem matemática explícita do sistema robótico (CALZADA-GARCIA et al., 2025a; GUMUS; KALYONCU, 2025). Além disso, uma vez treinados, esses modelos podem fornecer estimativas rápidas durante a fase de inferência, o que os torna atrativos para aplicações em tempo real. Contudo, a literatura também aponta desafios relevantes associados a essas abordagens, como a necessidade de grandes volumes de dados rotulados, limitações na generalização para configurações não observadas e baixa interpretabilidade dos modelos, frequentemente tratados como sistemas de “caixa-preta” (DENG et al., 2024; DIPRASETYA; PöPPELBAUM; SCHWUNG, 2025; JOSHI et al., 2025).

Outro aspecto recorrente refere-se à dificuldade de garantir que modelos baseados em Deep Learning respeitem as restrições físicas e cinemáticas dos manipuladores, o que pode resultar em soluções inconsistentes ou inviáveis do ponto de vista mecânico. Além disso, manipuladores redundantes, que possuem múltiplas soluções para uma mesma pose do efetuador final, impõem desafios adicionais aos modelos de aprendizado, exigindo estratégias capazes de lidar com essa ambiguidade (CAGIGAS-MUÑIZ, 2023; DENG et al., 2024; HO et al., 2023). Soma-se a isso o fato de que grande parte dos estudos se concentra em manipuladores de menor complexidade ou em cenários simulados, permanecendo em aberto questões relacionadas à escalabilidade dessas abordagens para manipuladores com maior número de graus de liberdade e para condições mais próximas de aplicações reais (DENG et al., 2024; LU; ZOU; JIANG, 2022).

Adicionalmente, observa-se na literatura uma ampla diversidade de arquiteturas de redes neurais propostas para a resolução do problema da cinemática inversa, incluindo perceptrons multicamadas, redes convolucionais, redes recorrentes e outras variantes associadas ao paradigma de Deep Learning. Entretanto, esses estudos são frequentemente apresentados de forma isolada e voltados a aplicações específicas, o que dificulta a compreensão sistemática das arquiteturas mais empregadas, de seu

desempenho relativo e dos contextos em que são mais adequadas. Essa fragmentação do conhecimento evidencia a necessidade de uma análise estruturada da literatura que permita identificar, organizar e comparar as diferentes abordagens baseadas em redes neurais aplicadas à cinemática inversa de manipuladores robóticos.

Diante desse cenário, torna-se relevante investigar de forma sistemática como redes neurais profundas têm sido aplicadas à resolução da cinemática inversa em sistemas robóticos. Em particular, compreender quais arquiteturas de redes neurais têm sido empregadas, bem como identificar suas principais características, limitações e desafios, constitui um passo fundamental para a consolidação do conhecimento na área. Assim, esta pesquisa justifica-se pela necessidade de mapear e analisar criticamente essas abordagens, contribuindo para a identificação de tendências, lacunas e oportunidades de avanço no desenvolvimento de métodos baseados em Deep Learning aplicados à robótica. Nesse contexto, o presente estudo busca responder à seguinte questão de pesquisa principal.

Assim, o presente estudo busca responder à seguinte questão de pesquisa principal:

Quais tipos de arquiteturas de redes neurais artificiais têm sido aplicadas na resolução do problema da cinemática inversa de manipuladores robóticos?

1.3 Publicação decorrente deste trabalho

Parte dos resultados desenvolvidos nesta dissertação foi publicada no artigo intitulado *Deep Neural Network-Based Approach for Inverse Kinematics Estimation in Robotics*, apresentado e publicado nos anais do *28th International Congress of Mechanical Engineering (COBEM 2025)*.

1.4 Estrutura da dissertação

No Capítulo 2 é apresentada a revisão sistemática da literatura, com foco na identificação e análise de métodos baseados em redes neurais artificiais aplicados à solução da cinemática inversa de manipuladores robóticos, incluindo a caracterização das abordagens, métricas de avaliação e lacunas existentes. No Capítulo 3 é apresentada a fundamentação teórica, abordando os conceitos de cinemática de manipuladores robóticos, redes neurais artificiais e aprendizado profundo necessários ao desenvolvimento do modelo proposto. No Capítulo 4 são descritos os materiais e métodos, incluindo a geração do conjunto de dados sintético, a definição da arquitetura da rede neural e os procedimentos de treinamento e validação. No Capítulo 5 são apresentados os resultados e a análise dos cenários avaliados. No Capítulo 6 são apresentadas as conclusões e as direções para trabalhos futuros.

2 REVISÃO SISTEMÁTICA DA LITERATURA

Esta revisão sistemática tem como objetivo fornecer uma visão abrangente e estruturada sobre a aplicação de redes neurais artificiais (RNAs) na solução da cinemática inversa de robôs manipuladores, abordando os principais desafios, as técnicas utilizadas e o desempenho obtido. A complexidade da cinemática inversa levou pesquisadores a explorarem abordagens baseadas em *Machine Learning* (ML) e *Deep Learning* (DL) para minimizar erros e otimizar a precisão de modelos computacionais aplicados a sistemas robóticos.

2.1 Metodologia da Revisão Sistemática da Literatura

De acordo com Chalmers, Hedges e Cooper (2002), os primeiros relatos de Revisões Sistemáticas da Literatura (RSL) remontam aos séculos XVIII e XIX. Entretanto, conforme destacado por Lame (2019), foi entre as décadas de 1960 e 1980 que diversos estudos passaram a apresentar grande variabilidade de resultados, em decorrência da síntese e da apresentação inadequada dos dados coletados, evidenciada, por exemplo, pelas contradições relatadas em Chassin (1987).

Esse contexto impulsionou o surgimento da abordagem baseada em evidências, cujo objetivo é subsidiar a tomada de decisão com base nos melhores resultados científicos disponíveis, reduzindo a dependência de decisões fundamentadas exclusivamente em instituições ou em procedimentos não científicos (SACKETT et al., 1996).

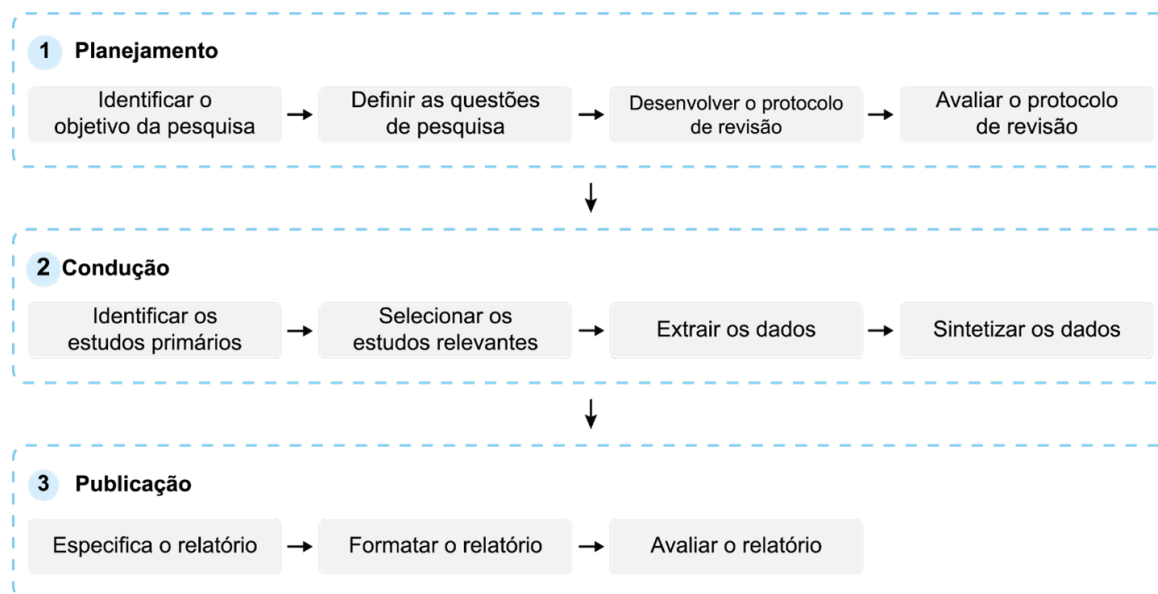
A metodologia adotada nesta Revisão Sistemática da Literatura para a construção do estado da arte acerca do tema investigado fundamenta-se no modelo proposto por Kitchenham e Charters (2007), os quais adaptaram o formato de RSL baseada em evidências, originalmente desenvolvido para a medicina, ao contexto da Engenharia de Software.

Em conformidade com o modelo proposto pelos autores, o desenvolvimento da RSL é realizado em um processo de três etapas, que se definem em:

1. Planejamento;
2. Condução;
3. Publicação.

De acordo com a ilustração apresentada na Figura 1.

Figura 1 – Processo de revisão sistemática da literatura



Fonte: Elaborada pelo Autor.

2.1.1 Planejamento

Segundo Kitchenham e Charters (2007), após a definição dos objetivos indicados na Seção 1.1, a fase de planejamento é definida em três estágios, a saber:

1. Definir as questões de pesquisa;
2. Desenvolver o protocolo de revisão;
3. Avaliar o protocolo de revisão.

2.1.1.1 Questões de pesquisa

De acordo com a justificativa da pesquisa apresentada na Seção 1.2, o principal objetivo desta revisão sistemática da literatura é responder às questões de pesquisa (Q) apresentadas na 1.

Tabela 1 – Questões de pesquisa (Q)

Q1. Quais tipos de arquiteturas de redes neurais artificiais têm sido aplicados para resolver o problema da cinemática inversa de robôs manipuladores?
Q2. Quais são os tipos de sistemas robóticos utilizados no experimento relacionado às redes neurais artificiais aplicadas na cinemática inversa?
Q3. Em quais plataformas ou ambientes de teste o experimento foi conduzido?
Q4. Quais são os principais <i>datasets</i> utilizados para treinar e validar os modelos de <i>Machine Learning</i> baseados em redes neurais para cinemática inversa?
Q5. Quais são as principais métricas utilizadas para avaliar o desempenho do modelo de treinamento e teste das redes neurais artificiais?
Q6. Quais são os desafios e limitações da aplicação de redes neurais artificiais na cinemática inversa?
Q7. Quais lacunas ainda existem na literatura sobre o uso de redes neurais para resolver a cinemática inversa de robôs manipuladores?

Fonte: Elaborada pelo Autor.

Para assegurar o rigor metodológico na formulação das questões, adotou-se o arcabouço conceitual da Prática Baseada em Evidências (PBE). Kitchenham e Charters (2007) destacam que, sob a perspectiva das diretrizes médicas, uma pergunta de pesquisa pode ser considerada eficaz quando contempla três dimensões fundamentais: População, Intervenção e Resultados.

Segundo os autores, a população refere-se ao grupo afetado pela intervenção; as intervenções, na ótica médica, envolvem a comparação entre duas ou mais alternativas de tratamento; e os resultados dizem respeito aos fatores clínicos e econômicos utilizados para comparar as intervenções.

Essas dimensões deram origem à estratégia PICO, acrônimo para *Population, Intervention, Comparison* e *Outcome* (População, Intervenção, Comparação e Resultados). Trata-se de uma ferramenta amplamente utilizada na Prática Baseada em Evidências (PBE) para estruturar questões de pesquisa de forma clara e sistemática.

Em uma abordagem mais recente, Petticrew e Roberts (2005) propuseram a extensão do modelo para PICOC, acrescentando o critério *Context* (Contexto), que especifica o ambiente ou as condições em que a intervenção é aplicada. O 2 apresenta a estratégia PICOC adotada nesta pesquisa.

Tabela 2 – Estratégia PICOC

Critério	Descrição
População	Estudos primários revisados por pares que apresentem aplicações de Redes Neurais Artificiais para alcançar a cinemática inversa de robôs.
Intervenção	Redes neurais artificiais, incluindo arquiteturas clássicas e profundas (<i>deep learning</i>), aplicadas ao problema da cinemática inversa.
Comparação	Comparação entre diferentes arquiteturas de redes neurais, bem como entre redes neurais e métodos convencionais de cinemática inversa, utilizando métricas de validação estabelecidas.
Resultado	Panorama das técnicas de <i>machine learning</i> baseadas em redes neurais aplicadas à cinemática inversa, incluindo arquiteturas, métricas de desempenho, desafios e lacunas da literatura.
Contexto	Estudos situados no domínio da Inteligência Artificial, <i>Machine Learning</i> , Redes Neurais Artificiais, Cinemática Inversa e Sistemas Robóticos.

Fonte: Elaborada pelo Autor.

2.1.1.2 Protocolo de revisão sistemática

Em conformidade com Kitchenham e Charters (2007), as Revisões Sistemáticas da Literatura (RSL) iniciam-se com a definição de um protocolo de revisão sistemática. Esse protocolo consiste em um plano que descreve como será conduzida a revisão sistemática proposta.

2.1.1.3 Avaliação do protocolo de revisão

Os autores destacam que a avaliação do protocolo de revisão pode ser conduzida por diferentes abordagens. Entre elas, destaca-se a contratação de especialistas externos para a análise do protocolo, especialmente em situações em que há disponibilidade de recursos financeiros destinados a esse propósito. Alternativamente, a avaliação pode ser realizada por orientadores de estudantes de pós-graduação, prática comum em pesquisas acadêmicas. No contexto desta revisão, adotou-se esta última abordagem, de modo que o protocolo de revisão foi avaliado pelo orientador responsável pelo acompanhamento desta pesquisa.

2.1.2 Condução

O processo de condução da revisão sistemática implica identificar fontes de pesquisa relevantes, selecionar os estudos primários que contribuirão para a investigação, avaliar a qualidade dos estudos selecionados, extrair de cada estudo primário os dados necessários para responder às questões de pesquisa e sintetizar os dados extraídos Kitchenham e Charters (2007).

2.1.2.1 *Estratégia de busca*

Inicialmente, foi realizada uma estratégia de busca preliminar nas bases de dados relevantes para esta pesquisa, com a finalidade de verificar se já existia alguma revisão sistemática da literatura que abordasse especificamente o tema desta pesquisa e, adicionalmente, identificar um estudo primário piloto (ŠEGOTA et al., 2021).

Em seguida, foram selecionadas as bases de dados IEEE Digital Library, ISI Web of Science e Scopus para a busca e coleta dos estudos primários relevantes necessários para o desenvolvimento desta revisão sistemática.

Após a escolha das bases de dados e em conformidade com a questão de pesquisa em destaque e com o critério PICOC, foram definidos os descritores e seus sinônimos para a construção da string de busca, conforme descrito a seguir.

Questão de pesquisa

Q1. Quais tipos de arquiteturas de redes neurais artificiais têm sido aplicados para resolver o problema da cinemática inversa de robôs manipuladores?

Critério PICOC

População: “Artificial Intelligence” OR “Machine Learning” OR “Artificial Neural Networks” OR “Deep Learning”

Intervenção: “Inverse Kinematics” AND (Robotic OR “Robotic Manipulators” OR “Robotic Systems”)

Comparação: “Dataset”

A busca nas bases de dados utilizando apenas os elementos de população e intervenção retornou os materiais necessários para a realização desta pesquisa. Os critérios de resultados e contexto encontram-se implicitamente inseridos na contextualização dos artigos selecionados, o que resultou em uma string de busca simples e eficaz para a produção deste material.

A string de busca foi construída em três etapas:

1. (“Artificial Intelligence” OR “Machine Learning” OR “Artificial Neural Networks” OR “Deep Learning”)
2. (“Inverse Kinematics”) AND (Robotic OR “Robotic Manipulators” OR “Robotic Systems”)
3. (“Dataset”)

Resultando na seguinte expressão final:

((“Artificial Intelligence” OR “Machine Learning” OR “Artificial Neural Networks” OR “Deep Learning”) AND (“Inverse Kinematics”) AND (Robotic OR “Robotic Manipulators” OR “Robotic Systems”) AND (“Dataset”))

2.1.2.2 Critérios de seleção dos estudos: inclusão e exclusão

Como apresentado na 3, foram definidos critérios de inclusão e exclusão para a seleção dos estudos, estabelecidos com base na questão principal da pesquisa e na necessidade de selecionar estudos primários relevantes revisados por pares.

Tabela 3 – Critérios de inclusão e exclusão

Critérios de inclusão (I)	Critérios de exclusão (E)
I1. Estudos primários e secundários.	E1. Artigos incompletos ou sem metodologia clara.
I2. Estudos que abordam assuntos sobre redes neurais artificiais aplicadas à cinemática inversa de robôs.	E2. Estudos terciários.
I3. Estudos que apresentam métricas de avaliação como MAE, MSE, etc.	E3. Estudos anteriores a 2015.
I4. Estudos disponíveis em periódicos e conferências de alto impacto (IEEE, Elsevier, Springer, etc.).	E4. Estudos que enfoquem apenas métodos tradicionais sem aplicação de redes neurais artificiais.
I5. Estudos publicados nos últimos 10 anos.	E5. Estudos que tratam de assuntos não relacionados à pesquisa.
I6. Estudos que podem complementar o desenvolvimento do trabalho.	E6. Literatura cinza (anais, relatórios, teses, dissertações, anais de congresso, etc.).
	E7. Revisões sem experimentação empírica.

Fonte: Elaborada pelo Autor.

2.1.2.3 Processo de seleção dos estudos

No processo de seleção dos estudos, foram aplicados filtros específicos nas bases de dados selecionadas, conforme apresentado na 4.

Tabela 4 – Processo de seleção dos estudos

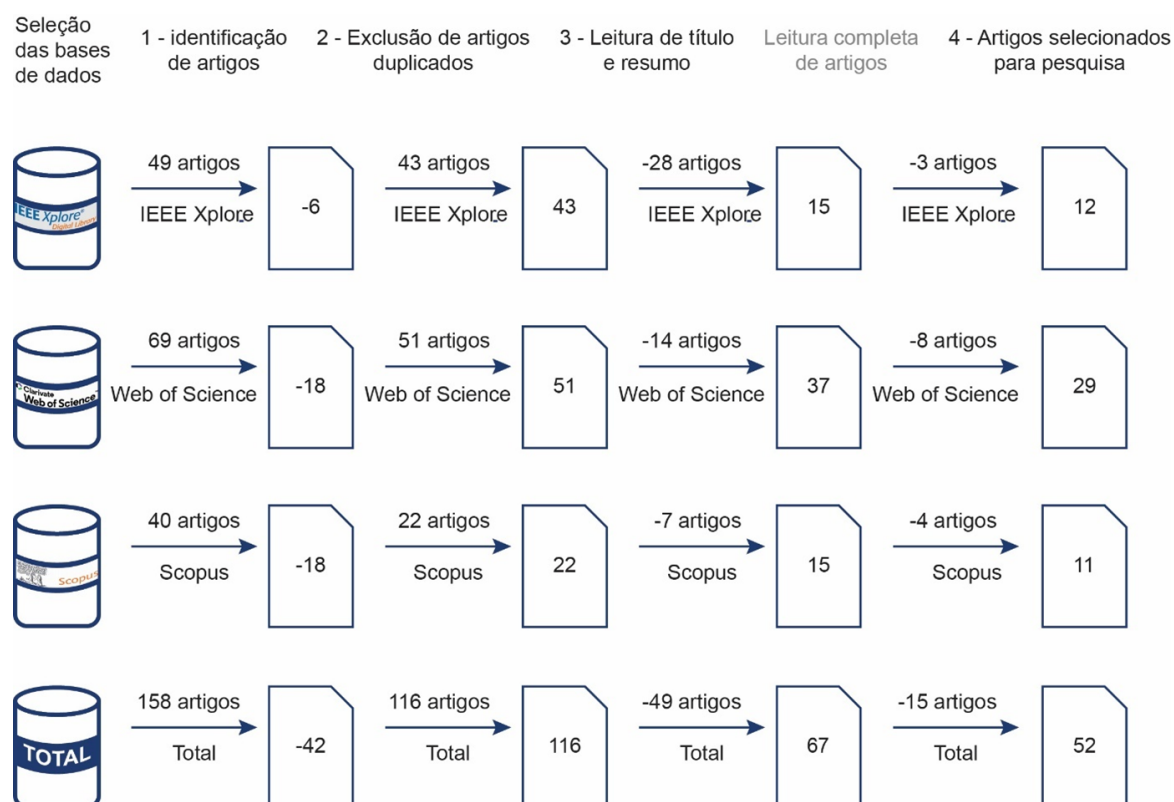
Base de dados	Aplicação de filtros
IEEE Digital Library	Período: 2015–2025.
ISI Web of Science	<i>Smart Search</i> , período de 2016–2025, somente artigos, artigos de revisão e acesso antecipado.
Scopus	Pesquisa básica, limitada a artigos de conferência e artigos.

Fonte: Elaborada pelo Autor.

O processo de seleção dos artigos foi conduzido com o apoio da ferramenta Parsif.al e do gerenciador de referências Mendeley, que contribuíram para a extração, organização, rastreabilidade e sistematização analítica dos materiais. A seleção dos estudos foi estruturada em quatro etapas sequenciais, conforme ilustrado na Figura 2, seguindo um protocolo metodológico previamente definido.

A Figura 2 apresenta o fluxo de seleção dos estudos incluídos nesta revisão, organizado em quatro etapas principais: identificação dos artigos nas bases de dados, remoção de duplicados, triagem por leitura de título e resumo e, por fim, avaliação por leitura completa dos artigos.

Figura 2 – Processo de seleção de artigos



Fonte: Elaborada pelo Autor.

Na etapa inicial de identificação, foram recuperados 158 estudos a partir das bases de dados selecionadas, sendo 49 provenientes da IEEE Xplore, 69 da Web of Science e 40 da Scopus. Em seguida, realizou-se a etapa de remoção de artigos duplicados, na qual foram excluídos 42 registros, resultando em 116 estudos únicos para a próxima fase de análise.

Posteriormente, foi conduzida a triagem por leitura de título e resumo, etapa em que foram excluídos 49 artigos por não atenderem aos critérios de inclusão definidos para a pesquisa. Após essa filtragem, 67 estudos permaneceram para avaliação mais detalhada.

Na etapa seguinte, realizou-se a leitura completa dos artigos, permitindo uma análise mais aprofundada da aderência dos estudos ao escopo da revisão. Durante essa fase, 15 artigos adicionais foram

excluídos, restando 52 estudos considerados adequados para compor o conjunto final de trabalhos analisados nesta pesquisa.

A análise por base de dados mostra que, ao final do processo de seleção, 12 estudos foram provenientes da IEEE Xplore, 29 da Web of Science e 11 da Scopus. Esses 52 artigos finais constituem o corpus utilizado para a extração de dados e para a análise das questões de pesquisa desta revisão sistemática da literatura.

2.1.2.4 Processo de extração e síntese dos dados

Para a extração e síntese dos dados, foi elaborado um formulário (apoiado pela ferramenta Parsif.al) para coletar informações que posteriormente auxiliariam na resposta às Questões de Pesquisa (QR). Como apresentado na 5, esse formulário também permitiu organizar os artigos de acordo com o tipo de análise específica, facilitando a extração e a organização dos dados dos estudos selecionados.

Tabela 5 – Formulário de extração dos dados

Propriedade da informação	Descrição das respostas	QR
Tipo de sistema	Quais tipos de arquiteturas de redes neurais artificiais têm sido aplicados para resolver o problema da cinemática inversa de robôs manipuladores?	QR1
Plataforma de teste	Quais são os tipos de sistemas robóticos utilizados no experimento relacionado às redes neurais artificiais aplicadas na cinemática inversa?	QR2
Arquitetura de rede	Em quais plataformas ou ambientes de teste o experimento foi conduzido?	QR3
Dataset	Quais são os principais <i>datasets</i> utilizados para treinar e validar os modelos de <i>Machine Learning</i> baseados em redes neurais para cinemática inversa?	QR4
Métricas	Quais são as principais métricas utilizadas para avaliar o desempenho do modelo de treinamento e teste das redes neurais artificiais?	QR5
Desafios e Limitações	Quais são os desafios e limitações da aplicação de redes neurais artificiais na cinemática inversa?	QR6
Lacunas	Quais lacunas ainda existem na literatura sobre o uso de redes neurais para resolver a cinemática inversa de robôs manipuladores?	QR7

Fonte: Elaborada pelo Autor.

2.1.2.5 Avaliação de qualidade

Segundo Kitchenham e Charters (2007), a avaliação da qualidade é essencial em uma revisão sistemática da literatura, pois permite explicar as diferenças entre os resultados encontrados nos

materiais pesquisados. Seguindo essa diretriz, foram definidas as questões para avaliação da qualidade (QA) dos estudos selecionados, conforme apresentado na 6.

Tabela 6 – Questões para Avaliação da Qualidade (QA)

QA	Questões de avaliação da qualidade	Critérios de Pontuação
QA1	O estudo apresenta de forma clara a arquitetura de rede neural utilizada, justificando sua escolha em relação às características do problema de cinemática inversa?	N / P / S
QA2	O problema de pesquisa é claramente formulado, com justificativa técnica para o uso de redes neurais artificiais frente a métodos convencionais de cinemática inversa?	N / P / S
QA3	O estudo apresenta o estado da arte em cinemática inversa e discute o papel das redes neurais artificiais nesse contexto?	N / P / S
QA4	Os objetivos da pesquisa são claramente definidos e estão alinhados ao problema e à aplicação das redes neurais na cinemática inversa?	N / P / S
QA5	O estudo discute limitações do método proposto e desafios atuais da área, apresentando sugestões concretas para pesquisas futuras?	N / P / S
QA6	O estudo identifica lacunas na literatura e limitações do método proposto, propondo abordagens ou estratégias para superá-las?	N / P / S

Fonte: Elaborada pelo Autor.

Cada estudo selecionado foi submetido às questões de qualidade, com pontuações atribuídas conforme os seguintes critérios:

N (Não) = 0 ponto

P (Parcialmente) = 0,5 ponto

S (Sim) = 1 ponto

A pontuação máxima possível por artigo é de 6 pontos, correspondendo ao total de questões avaliadas, o que equivale a 100% de qualidade.

O percentual de qualidade de cada artigo foi calculado pela seguinte expressão:

$$P_q = \left(\frac{V_{st}}{N_t} \right) \times 100$$

onde:

P_q = percentual de qualidade do artigo (%)

V_{st} = soma total das pontuações obtidas nas seis questões (0 a 6)

N_t = número total de questões avaliadas (fixo em 6)

Por exemplo, um artigo que obteve soma $V_{st} = 4,5$ terá:

$$P_q = \left(\frac{4,5}{6} \right) \times 100 = 75\%$$

Para fins de transparência metodológica, também foi calculado o percentual de artigos que efetivamente passaram pelo processo de avaliação em relação ao total da amostra inicial:

$$P_{sa} = \left(\frac{N_{ta}}{N_{tp}} \right) \times 100$$

onde:

P_{sa} = percentual de artigos avaliados

N_{ta} = número total de artigos efetivamente avaliados

N_{tp} = número total de artigos pesquisados (amostra inicial)

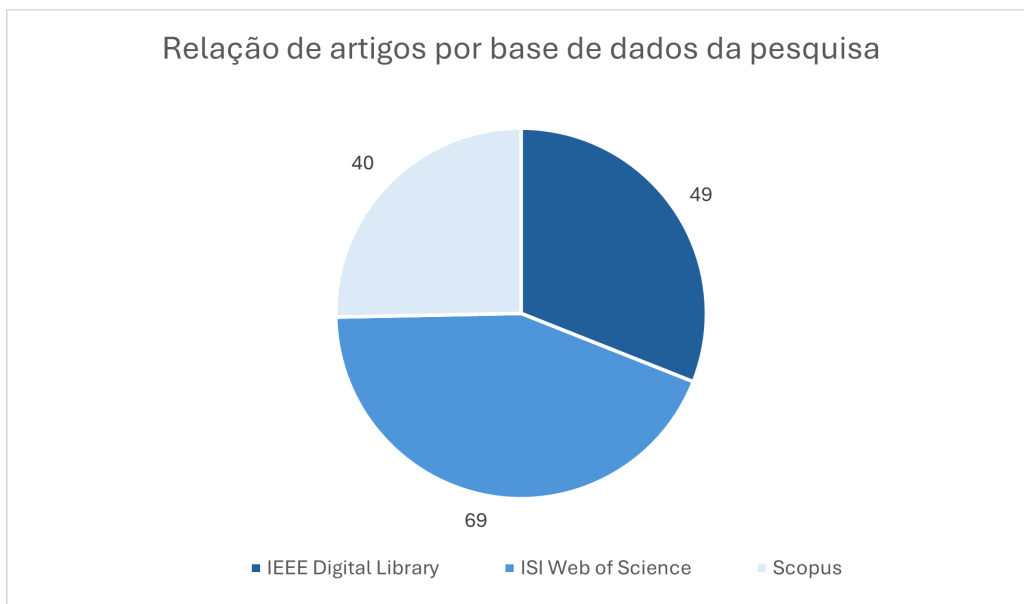
Considerando uma amostra inicial de $N_{tp} = 52$, caso todos os artigos tenham sido avaliados, tem-se:

$$P_{sa} = \left(\frac{52}{52} \right) \times 100 = 100\%$$

2.2 Resultados e Discussões

A primeira etapa da revisão sistemática consistiu na recuperação de publicações nas principais bases de dados bibliográficas da área. A Figura 3 apresenta a distribuição percentual dos documentos identificados em cada base de dados, considerando o total de registros recuperados antes da remoção de duplicatas e da aplicação dos critérios de elegibilidade.

Figura 3 – Distribuição percentual das publicações por base de dados



Fonte: Elaborada pelo Autor.

A busca nas principais bases de dados resultou em uma distribuição relativamente equilibrada entre

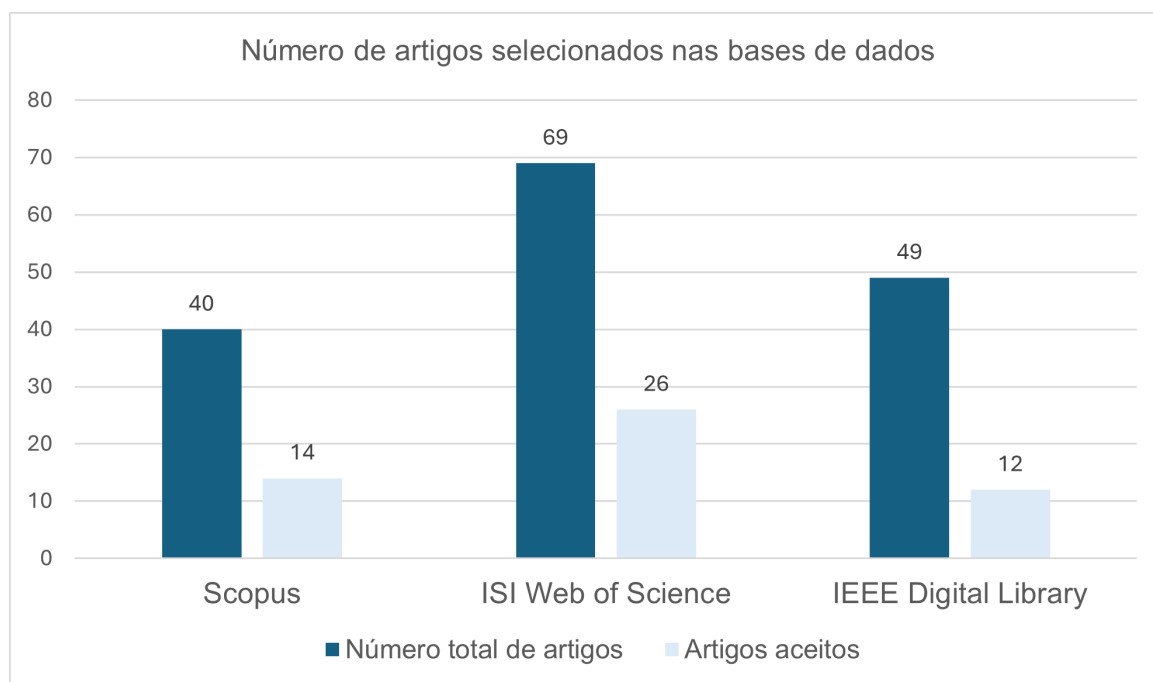
as fontes consultadas. A ISI Web of Science apresentou o maior número de registros, correspondendo a 44% do total de publicações identificadas (69 artigos), seguida pela IEEE Xplore Digital Library, com 31% (49 artigos), e pela Scopus, com 25% (40 artigos). Esses percentuais referem-se ao total de documentos recuperados antes da etapa de remoção de duplicatas.

A sobreposição entre essas bases é esperada, particularmente entre Scopus e Web of Science, que indexam um grande número de periódicos em comum. Essa característica justifica a necessidade da etapa subsequente de identificação e eliminação de registros duplicados no processo de revisão sistemática.

A predominância de registros na Web of Science sugere que o tema investigado apresenta consolidação em periódicos científicos de alto impacto internacional. Por sua vez, a participação expressiva da IEEE Xplore Digital Library evidencia a forte presença de resultados publicados em conferências técnicas, característica típica das áreas de engenharia, robótica e inteligência artificial, nas quais avanços metodológicos e experimentais são frequentemente divulgados inicialmente em eventos científicos especializados. A presença complementar de registros na base Scopus reforça o caráter multidisciplinar da temática investigada, abrangendo diferentes áreas da engenharia e da computação.

A Figura 4 apresenta uma síntese dos resultados obtidos no processo de triagem dos estudos, evidenciando a quantidade de artigos inicialmente identificados e aqueles selecionados após a aplicação dos critérios de inclusão e exclusão nas principais bases de dados consultadas na revisão: Scopus, ISI Web of Science e IEEE Xplore Digital Library.

Figura 4 – Distribuição dos artigos identificados e selecionados por base de dados



Fonte: Elaborada pelo Autor.

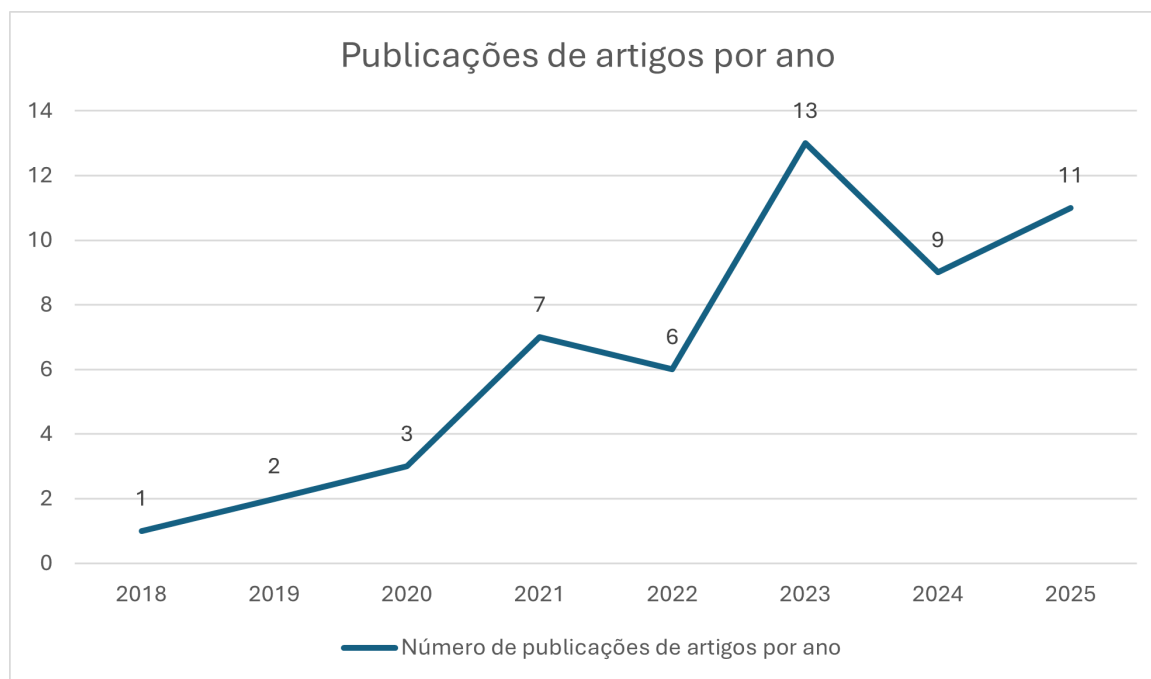
Observa-se que a ISI Web of Science apresentou o maior número de registros inicialmente identificados (69 artigos), seguida pela IEEE Xplore Digital Library (49 artigos) e pela Scopus (40 artigos).

Após a aplicação dos critérios de inclusão e exclusão, a Web of Science também se destacou como a principal fonte de estudos selecionados, contribuindo com 26 artigos para a análise final. A Scopus apresentou 14 artigos selecionados, enquanto a IEEE Xplore Digital Library contribuiu com 12 estudos.

Esses resultados indicam que a Web of Science não apenas apresentou maior volume inicial de registros, mas também maior contribuição para a composição da amostra final da revisão. Além disso, observa-se que, embora a IEEE Xplore Digital Library tenha apresentado um número considerável de registros iniciais, uma parcela menor desses estudos atendeu plenamente aos critérios de inclusão estabelecidos, possivelmente devido à natureza mais ampla e técnica das publicações presentes nessa base. Por outro lado, a Scopus apresentou uma proporção relativamente consistente de estudos selecionados em relação ao total identificado, reforçando sua relevância como fonte complementar de literatura científica na área investigada.

A Figura 5 apresenta a distribuição temporal dos artigos finais incluídos na revisão sistemática da literatura, considerando o período de 2018 a 2025.

Figura 5 – Distribuição temporal das publicações incluídas na revisão sistemática (2018–2025)



Fonte: Elaborada pelo Autor.

A análise temporal dos estudos selecionados evidencia um crescimento gradual da produção científica relacionada ao tema investigado ao longo dos últimos anos. Nos anos iniciais do período analisado, entre 2018 e 2020, observa-se um número reduzido de publicações, com 1, 2 e 3 artigos,

respectivamente, indicando que a temática ainda se encontrava em estágio inicial de desenvolvimento na literatura científica.

A partir de 2021, identifica-se um aumento mais consistente na quantidade de estudos publicados, com 6 artigos em 2021 e 6 artigos em 2022, sugerindo uma fase de consolidação do interesse acadêmico pelo tema. Esse crescimento torna-se ainda mais evidente em 2023, ano que apresenta o maior número de publicações incluídas na revisão, com 14 artigos, representando o pico de produção científica no período analisado.

Nos anos mais recentes, observa-se a manutenção de um volume relevante de publicações, com 9 artigos em 2024 e 11 artigos em 2025, indicando a continuidade do interesse da comunidade científica na temática investigada. É importante destacar que os dados referentes a 2025 podem ser considerados parciais, uma vez que o processo de indexação de publicações nas bases de dados pode continuar ao longo do ano.

De modo geral, a tendência observada sugere que o tema investigado vem ganhando relevância crescente na literatura científica recente, especialmente a partir da década de 2020, refletindo avanços tecnológicos e o aumento das aplicações de redes neurais artificiais e métodos de aprendizado de máquina em robótica.

A 7 apresenta os artigos selecionados para compor o corpus final desta revisão sistemática, após a aplicação dos critérios de inclusão, exclusão e avaliação de qualidade. Para cada publicação, são apresentados os seguintes elementos: identificador único (id), primeiro autor, título completo, ano de publicação e pontuação total de qualidade (TQA – *Total Quality Assessment*). A pontuação TQA foi calculada com base nas respostas aos critérios de avaliação de qualidade estabelecidos no 6, permitindo mensurar o rigor metodológico e a relevância científica de cada estudo em relação à aplicação de redes neurais profundas (*Deep Learning*) na resolução de problemas de cinemática inversa em robótica.

Tabela 7 – Artigos selecionados para a pesquisa

id	Autor	Título do artigo	Ano	TQA
S1	JOSHI et al.	Optimized inverse kinematics modeling and joint angle prediction for six-degree-of-freedom anthropomorphic robots with Explainable AI	2025	5.0
S2	GUMUS; KALYONCU	A novel architecture for artificial neural networks to solve the inverse kinematics problem in robotics	2025	5.0

Continua na próxima página

id	Autor	Título do artigo	Ano	TQA
S3	SEMWAL et al.	Generation of Inverse Kinematics (IK) Trajectory of 3 Link Manipulator using Deep Learning Model	2025	4.0
S4	ZHANG et al.	Invertible liquid neural network-based learning of inverse kinematics and dynamics for robotic manipulators	2025	5.0
S5	SIŁKA; WIECZOREK	Custom Sequential Neural Network for Inverse Kinematic Regression	2025	4.5
S6	MANJEGOWDA; RAO	Path-based evaluation of deep learning models for solving inverse kinematics in a revolute-prismatic robot	2025	4.5
S7	CALZADA-GARCIA et al.	Inverse Kinematics for Robotic Manipulators via Deep Neural Networks: Experiments and Results	2025	4.5
S8	CALZADA-GARCIA et al.	A Review on Inverse Kinematics, Control and Planning for Robotic Manipulators With and Without Obstacles via Deep Neural Networks	2025	5.5
S9	MURCIA; CASTRO; RIVILLAS	Comparison of Neural Networks with a Numerical Method for Obtaining the Inverse Kinematic of a Bipedal Robot	2025	5.0
S10	BOUZID; GRITLI; NARAYAN	Solving Inverse Kinematics of a 2-DoF Upper-Limb Exoskeleton Robot Through Metaheuristic Optimization of ANN Hyperparameters	2025	4.5
S11	DIPRASETYA; PöPPELBAUM; SCHWUNG	KineNN: Kinematic Neural Network for inverse model policy based on homogeneous transformation matrix and dual quaternion	2025	5.0
S12	RICE; AZAB; SAAD	Fusion IK: Solving Inverse Kinematics using a Hybridized Deep Learning and Evolutionary Approach	2024	5.0
S13	DENG et al.	Physics informed machine learning model for inverse dynamics in robotic manipulators	2024	5.0

Continua na próxima página

id	Autor	Título do artigo	Ano	TQA
S14	M. et al.	Estimation of Inverse Kinematics Solutions of a 2D Planar Robotic Manipulator using Feed-Forward Neural Network	2024	4.5
S15	BOUZID; GRITLI; NARAYAN	Optimized Inverse Kinematics of a 2-DoF Robotic Manipulator Using a Hybrid Approach Combining an ANN with a Metaheuristic Algorithm	2024	4.5
S16	BOUZID; NARAYAN; GRITLI	Investigating Neural Network Hyperparameter Variations in Robotic Arm Inverse Kinematics for Different Arm Lengths	2024	4.5
S17	BOUZID; NARAYAN; GRITLI	Hybrid Metaheuristic and Artificial Neural Network Approach for Solving Inverse Kinematics of a SCARA Manipulator Robot	2024	5.0
S18	AYDOGMUS; BOZTAS	Implementation of singularity-free inverse kinematics for humanoid robotic arm using Bayesian optimized deep neural network	2024	5.0
S19	ZHAO et al.	Inverse kinematics solution and control method of 6-degree-of-freedom manipulator based on deep reinforcement learning	2024	4.5
S20	SHAREEF	Artificial Neural Network-based Approach for Inverse Kinematics of PUMA 260 Robot	2024	5.0
S21	VU et al.	Machine learning-based framework for optimally solving the analytical inverse kinematics for redundant manipulators	2023	4.5
S22	CAGIGAS-MUÑOZ	Artificial Neural Networks for inverse kinematics problem in articulated robots	2023	5.0
S23	NUGROHO; YUNI-ARNO; PURNOMO	ARKOMA dataset: An open-source dataset to develop neural networks-based inverse kinematics model for NAO robot arms	2023	4.0

Continua na próxima página

id	Autor	Título do artigo	Ano	TQA
S24	WAGAA; KALLEL; MELLOULI	Analytical and deep learning approaches for solving the inverse kinematic problem of a high degrees of freedom robotic arm	2023	5.0
S25	BAI; HSIEH	A hybrid method using FABRIK and custom ANN in solving inverse kinematic for generic serial robot manipulator	2023	5.5
S26	HAMARSHEH et al.	An Artificial Neural Network Approach in Solving Inverse Kinematics of a 6 DOF KUKA Industrial Robot	2023	5.0
S27	HO et al.	A Deep Learning Approach to Navigating the Joint Solution Space of Redundant Inverse Kinematics and Its Applications to Numerical IK Computations	2023	6.0
S28	LIU et al.	Modeling and Control of Robotic Manipulators Based on Artificial Neural Networks: A Review	2023	5.0
S29	LAUREL et al.	Solving the Inverse Kinematics of a Five Axis CNC Machine Using Shallow and Deep Neural Networks	2023	5.0
S30	ZAFAR; MOOSAVI; SANFILIPPO	Inverse Kinematic Modelling of a 3-DOF Robotic Manipulator using Hybrid Deep Learning Models	2023	4.0
S31	SHAAR; GHAEB	Intelligent Solution for Inverse Kinematic of Industrial Robotic Manipulator Based on RNN	2023	4.5
S32	BAHANI et al.	The Inverse Kinematics Evaluation of 6-DOF Robots in Cooperative Tasks Using Virtual Modeling Design and Artificial Intelligence Tools	2023	4.5
S33	ESMAT; ABDEL- NASSER; SAIF	Solving Inverse Kinematics for 3R Manipulator Using Artificial Neural Networks	2023	4.5

Continua na próxima página

id	Autor	Título do artigo	Ano	TQA
S34	IBARRA-PÉREZ et al.	A Novel Inverse Kinematic Solution of a Six-DOF Robot Using Neural Networks Based on the Taguchi Optimization Technique	2022	5.0
S35	NGUYEN; CHEAH; TOH	An analytic layer-wise deep learning framework with applications to robotics	2022	5.0
S36	LU; ZOU; JIANG	A Neural Network Based Approach to Inverse Kinematics Problem for General Six-Axis Robots	2022	5.5
S37	ZHONG et al.	AcousNet: A Deep Learning Based Approach to Dynamic 3D Holographic Acoustic Field Generation from Phased Transducer Array	2022	5.0
S38	TAGLIANI; PELLEGRINI; AGGOGERI	Machine Learning Sequential Methodology for Robot Inverse Kinematic Modelling	2022	5.0
S39	MALIK et al.	A Deep Reinforcement-Learning Approach for Inverse Kinematics Solution of a High Degree of Freedom Robotic Manipulator	2022	5.0
S40	TANG; NOTASH	Neural network-based transfer learning of manipulator inverse displacement analysis	2021	4.5
S41	JIMÉNEZ-LÓPEZ et al.	Modeling of Inverse Kinematic of 3-DoF Robot, Using Unit Quaternions and Artificial Neural Network	2021	4.5
S42	WANG et al.	Deep-learning damped least squares method for inverse kinematics of redundant robots	2021	5.0
S43	TOQUICA et al.	An analytical and a Deep Learning model for solving the inverse kinematic problem of an industrial parallel robot	2021	4.5
S44	ARUNADEVI et al.	Machine Learning approach for Inverse Kinematics in Trajectory Planning of Pioneer 2 Manipulator with Cubic Spline Interpolation	2021	4.5

Continua na próxima página

id	Autor	Título do artigo	Ano	TQA
S45	ŠEGOTA et al.	Utilization of multilayer perceptron for determining the inverse kinematics of an industrial robotic manipulator	2021	5.0
S46	ARAVINDDHAKSHAN; APTE; AKASH	Neural Network Based Inverse Kinematic Solution of a 5 DOF Manipulator for Industrial Application	2021	5.0
S47	REN; BEN-TZVI	Learning inverse kinematics and dynamics of a robotic manipulator using generative adversarial networks	2020	5.0
S48	GAO	Inverse kinematics solution of Robotics based on neural network algorithms	2020	4.5
S49	Kamlesh Shah; MISHRA; Samprit Ray	Solution and validation of inverse kinematics using Deep artificial neural network	2020	5.0
S50	RAJA; DUTTA; DAS-GUPTA	Learning framework for inverse kinematics of a highly redundant mobile manipulator	2019	4.5
S51	SHAH; MISHRA	Experimental validation for inverse kinematics of a five axis robotic manipulator using deep artificial neural network	2019	5.0
S52	ZHOU et al.	Inverse kinematics solution for robotic manipulator based on extreme learning machine and sequential mutation genetic algorithm	2018	4.5

Fonte: Elaborado pelo autor.

Com o objetivo de avaliar o nível geral de qualidade metodológica dos estudos incluídos nesta revisão sistemática, foram calculadas métricas agregadas a partir das pontuações de avaliação de qualidade (TQA) atribuídas individualmente a cada artigo. Essas métricas permitem analisar quantitativamente o rigor científico do corpus selecionado e identificar o nível médio de qualidade metodológica dos estudos analisados. Para isso, foram considerados a soma total das pontuações de qualidade, a pontuação máxima possível, a média de qualidade dos estudos e o percentual médio de qualidade em relação ao valor máximo de pontuação.

1. Soma total das pontuações de qualidade

$$\sum TQA = 250$$

2. Pontuação máxima possível

Considerando que cada estudo poderia alcançar uma pontuação máxima de 6 pontos, correspondente às seis questões de avaliação de qualidade, a pontuação máxima possível para o conjunto de 52 artigos é calculada pela seguinte equação:

$$TQA_{\max} = 52 \times 6 = 312$$

3. Qualidade média dos artigos

A média de qualidade dos estudos foi obtida pela razão entre a soma das pontuações e o número total de artigos analisados:

$$\overline{TQA} = \frac{250}{52} \approx 4.81$$

Assim, a pontuação média de qualidade dos estudos analisados é de 4,81 pontos, considerando um valor máximo possível de 6 pontos por artigo.

4. Percentual médio de qualidade

Para facilitar a interpretação dos resultados, a média de qualidade também foi convertida em percentual em relação ao valor máximo possível de pontuação:

$$Qualidade_{mdia} = \frac{4.81}{6} \times 100 \approx 80.1\%$$

A 8 apresenta as estatísticas descritivas obtidas a partir das pontuações de qualidade atribuídas aos estudos selecionados.

Tabela 8 – Estatísticas descritivas da avaliação de qualidade

Métrica	Valor
Número de estudos	52
Pontuação mínima	4.0
Pontuação máxima	6.0
Pontuação média	4.81
Qualidade média	80.1%

Fonte: Elaborado pelo autor.

Dessa forma, os estudos selecionados apresentam, em média, aproximadamente 80% da pontuação máxima possível de qualidade, indicando que o corpus analisado possui bom nível de rigor metodológico e consistência científica.

2.2.1 *Respostas e discussões referentes às questões de pesquisa (QR)*

A cinemática inversa de robôs manipuladores é considerada um dos problemas centrais da robótica, consistindo em determinar os valores das articulações que resultam em uma posição e orientação desejadas do efector final. Tradicionalmente, métodos analíticos e numéricos têm sido empregados para resolver esse problema; entretanto, tais abordagens apresentam limitações quando aplicadas a robôs com múltiplos graus de liberdade, estruturas complexas ou em situações que envolvem singularidades.

Nos últimos anos, abordagens baseadas em *Deep Learning* têm despertado crescente interesse como alternativa para a estimativa da cinemática inversa. Redes neurais artificiais profundas possuem a capacidade de aproximar funções não lineares complexas, permitindo o controle mais eficiente de manipuladores robóticos, inclusive em configurações não triviais ou na presença de incertezas nos parâmetros do sistema. Nesse contexto, torna-se relevante analisar sistematicamente aspectos como arquiteturas de redes neurais utilizadas, tipos de robôs estudados, plataformas experimentais, conjuntos de dados empregados, métricas de avaliação e limitações identificadas na literatura.

As questões de pesquisa desta dissertação foram formuladas com o objetivo de fornecer uma compreensão abrangente sobre o uso de redes neurais artificiais na resolução do problema de cinemática inversa em robôs manipuladores. A 9 apresenta uma síntese das questões investigadas, bem como um resumo dos principais resultados identificados na literatura analisada. Essa visão geral permite observar de forma integrada como diferentes aspectos — incluindo arquiteturas de redes neurais, tipos de sistemas robóticos, ambientes experimentais, *datasets*, métricas de avaliação, desafios e lacunas de pesquisa — contribuem para a construção do estado da arte sobre o uso de redes neurais artificiais na resolução do problema de cinemática inversa.

Tabela 9 – Síntese das questões de pesquisa e principais resultados

Questão de Pesquisa	Síntese dos Resultados	Evidências (IDs)
QR1 – Arquiteturas de redes neurais aplicadas à cinemática inversa	Predominância de redes MLP/ANN <i>feedforward</i> para modelar o mapeamento não linear entre o espaço cartesiano e o espaço articular. Outras abordagens incluem DNN, RNN/LSTM, CNN e modelos híbridos, porém com menor frequência.	S2, S9, S14, S16, S21, S22, S23, S24, S26, S32, S33, S34, S40, S41, S44, S45, S46, S48, S50
QR2 – Tipos de sistemas robóticos utilizados	Predominância de manipuladores seriais industriais, especialmente robôs com seis ou sete graus de liberdade, utilizados em tarefas de manipulação e automação industrial. Outros sistemas incluem robôs humanoides, manipuladores cooperativos e robôs paralelos.	S1–S5, S10–S31, S33–S36, S38–S42, S44–S50, S52
QR3 – Plataformas e ambientes experimentais	A maior parte dos estudos utiliza ambientes de simulação para treinamento e validação dos modelos, frequentemente utilizando ferramentas como MATLAB, Gazebo e CoppeliaSim, com alguns estudos incluindo validação em robôs reais.	S5, S6, S10, S11, S14, S15, S22, S24, S29, S30, S31
QR4 – Tipos de datasets utilizados	Predominância de <i>datasets</i> sintéticos gerados por simulação cinemática, produzidos a partir de modelos matemáticos ou simuladores. A utilização de dados reais ainda é limitada.	S1, S3, S4, S10, S11, S15, S23, S27, S32, S35
QR5 – Métricas de avaliação dos modelos	As métricas mais utilizadas são MSE, RMSE e MAE, que avaliam o erro entre os valores articulares previstos pela rede neural e os valores reais obtidos pelo modelo cinemático.	S3, S6, S10, S22, S35, S42
QR6 – Desafios e limitações identificados	Os principais desafios incluem múltiplas soluções para a cinemática inversa, singularidades, dependência de dados simulados e limitações na capacidade de generalização dos modelos treinados.	S3, S10, S19, S38, S42, S47
QR7 – Lacunas de pesquisa na literatura	A literatura apresenta lacunas relacionadas à ausência de <i>benchmarks</i> padronizados, à limitada validação em robôs reais e à baixa exploração de arquiteturas profundas mais recentes.	S3, S10, S19, S38, S42, S47, S49

Fonte: Elaborado pelo autor.

A partir dessa visão geral da revisão sistemática, as subseções seguintes apresentam de forma detalhada as respostas e discussões referentes a cada uma das questões de pesquisa definidas neste estudo.

QR1 – Quais tipos de arquiteturas de redes neurais artificiais têm sido aplicados para resolver o problema da cinemática inversa de robôs manipuladores?

Em relação aos tipos de arquiteturas de redes neurais artificiais aplicadas à resolução do problema de cinemática inversa em robôs manipuladores, a análise dos estudos selecionados revelou uma ampla diversidade de abordagens. Ao todo, 52 estudos primários foram analisados, abrangendo diferentes paradigmas de redes neurais, desde arquiteturas clássicas baseadas em perceptrons multicamadas até métodos mais recentes baseados em aprendizado profundo e aprendizado por reforço. A 10 apresenta os principais tipos de arquiteturas identificadas na literatura e a frequência aproximada com que aparecem nos estudos analisados.

Tabela 10 – Principais arquiteturas de redes identificadas na literatura

Categoria de arquitetura	Exemplos nos estudos	Frequência aproximada
MLP / <i>Feedforward</i> / ANN clássica	S2, S9, S14, S16, S21, S22, S23, S24, S26, S32, S33, S34, S40, S41, S44, S45, S46, S48, S50	Mais frequente (50–60%)
<i>Deep Neural Networks</i> (DNN)	S29, S30, S35, S36, S42	Moderada
Redes recorrentes (RNN / LSTM / GRU)	S3, S6, S31, S43	Moderada
CNN (<i>Convolutional Neural Networks</i>)	S1, S7, S13, S37	Menor
Arquiteturas híbridas (ANN + metaheurísticas)	S10, S11, S15, S17, S38, S47, S52	Frequente
Modelos físicos guiados (<i>Physics-informed</i>)	S25	Raro
Modelos probabilísticos / <i>encoder-decoder</i>	S27	Raro
GAN (<i>Generative Adversarial Networks</i>)	S47	Muito raro
<i>Self-Organizing Maps</i> (SOM)	S49	Muito raro
<i>Reinforcement Learning</i>	S20, S39	Emergente
Arquiteturas customizadas (ex.: KineNN, ILNN)	S4, S5, S8, S11, S13, S19, S28	Emergente

Fonte: Elaborado pelo autor.

Entre as arquiteturas identificadas, as redes neurais *feedforward*, particularmente Multilayer Perceptrons (MLP) e Artificial Neural Networks (ANN), representam os modelos mais frequentemente utilizados na literatura (S2, S9, S14, S16, S21, S22, S23, S24, S26, S32, S33, S34, S40, S41, S44, S45, S46, S48, S50). Essas redes são comumente aplicadas para aproximar o mapeamento não linear entre a

pose cartesiana do efetuador final e as variáveis do espaço articular do robô, problema que caracteriza a formulação clássica da cinemática inversa.

Além das redes *feedforward*, diversas outras arquiteturas foram identificadas nos estudos analisados. Entre elas destacam-se redes neurais profundas (DNN) (S29, S30, S35, S36, S42), redes neurais recorrentes (RNN, LSTM e GRU) (S3, S6, S31, S43) e redes neurais convolucionais (CNN) (S1, S7, S13, S37). Também foram identificadas abordagens híbridas, nas quais redes neurais são combinadas com algoritmos de otimização metaheurísticos, como *Genetic Algorithms* (GA) ou *Particle Swarm Optimization* (PSO), conforme observado em diversos estudos (S10, S11, S15, S17, S38, S47, S52).

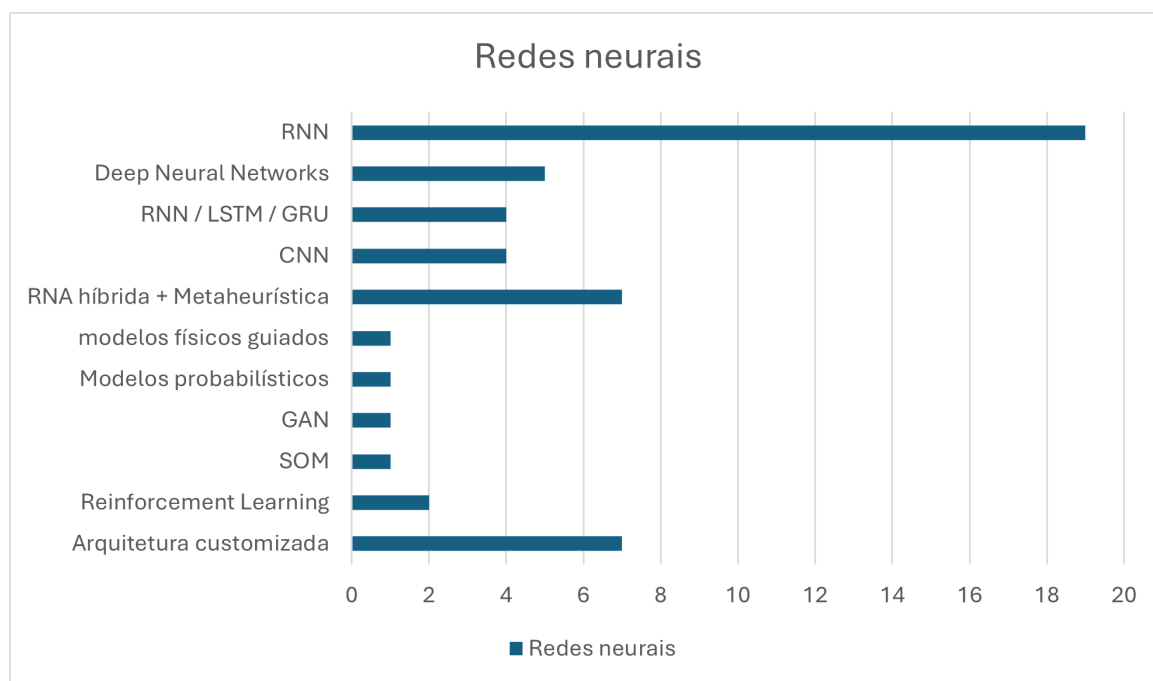
Estudos mais recentes exploram paradigmas alternativos de aprendizagem, incluindo aprendizado por reforço (S20, S39), redes generativas adversariais (GAN) (S47) e arquiteturas neurais especializadas desenvolvidas especificamente para modelar a cinemática inversa, como KineNN ou ILNN (S4, S5, S8, S11, S13, S19, S28). Outras abordagens menos frequentes incluem modelos físicos guiados (S25), modelos probabilísticos baseados em *encoder-decoder* (S27) e redes *Self-Organizing Maps* (SOM) (S49).

De modo geral, a análise dos estudos indica uma predominância clara de modelos *feedforward*, especialmente MLPs. Essa predominância pode ser explicada pela natureza do problema de cinemática inversa, que pode ser formulado como uma tarefa de regressão não linear que mapeia o espaço cartesiano do efetuador final para o espaço articular do robô. Nesse contexto, redes MLP são particularmente adequadas devido à sua capacidade universal de aproximação de funções.

Outras arquiteturas aparecem com menor frequência na literatura. Redes profundas e redes convolucionais foram exploradas principalmente em cenários que envolvem representações de entrada de maior dimensionalidade ou espaços de características mais complexos (S1, S7, S13, S29, S30, S35, S36, S37, S42). De forma semelhante, redes recorrentes têm sido utilizadas em cenários que envolvem dados sequenciais, como predição de trajetórias ou planejamento de movimento (S3, S6, S31, S43).

A Figura 6 apresenta a distribuição das principais arquiteturas de redes neurais identificadas nos estudos incluídos nesta revisão sistemática.

Figura 6 – Distribuição das arquiteturas de redes neurais identificadas na literatura



Fonte: Elaborado pelo autor.

Outro padrão relevante observado diz respeito à relação entre a complexidade do manipulador robótico e a arquitetura de rede utilizada. Para manipuladores relativamente simples, com dois ou três graus de liberdade, a maioria dos estudos utiliza arquiteturas MLP rasas, com uma ou duas camadas ocultas (S2, S9, S14, S16, S21, S22, S23, S24, S26). Essas redes geralmente são suficientes para aproximar o mapeamento da cinemática inversa devido à baixa dimensionalidade do problema.

Em contraste, manipuladores com cinco ou seis graus de liberdade, comuns na robótica industrial, frequentemente requerem arquiteturas mais profundas ou abordagens híbridas para alcançar níveis satisfatórios de precisão (S10, S11, S15, S17, S29, S30, S35, S36, S38, S42). Esses sistemas apresentam maior grau de não linearidade e podem possuir múltiplas soluções válidas de cinemática inversa.

Para manipuladores altamente redundantes, arquiteturas mais avançadas têm sido exploradas, incluindo redes profundas, modelos recorrentes e aprendizado por reforço (S3, S6, S20, S31, S39, S43).

A análise também revelou uma relação entre a arquitetura de rede utilizada e o tipo de representação de entrada adotada. A maioria das redes *feedforward* utiliza representações relativamente simples, como coordenadas cartesianas do efetuador final ou descrições completas de pose (S2, S9, S14, S16, S21, S22, S23, S24, S26). Em contraste, redes recorrentes são tipicamente empregadas quando o problema envolve dados sequenciais, como trajetórias robóticas ou sinais de controle dependentes do tempo (S3, S6, S31, S43). De maneira semelhante, abordagens baseadas em aprendizado por reforço

frequentemente utilizam estados do sistema como entrada, permitindo que o modelo aprenda políticas de controle por meio da interação com o ambiente (S20, S39).

Por fim, observa-se uma evolução gradual das arquiteturas utilizadas na literatura. Estudos iniciais concentraram-se principalmente em redes neurais *feedforward* clássicas treinadas com *backpropagation* (S2, S9, S14, S16, S21, S22). Posteriormente, abordagens híbridas combinando redes neurais com algoritmos evolutivos de otimização tornaram-se mais comuns (S10, S11, S15, S17, S38, S52). Mais recentemente, o avanço das técnicas de aprendizado profundo levou à exploração de redes profundas, redes convolucionais e modelos recorrentes (S1, S3, S6, S7, S29, S30, S35, S36, S37, S42, S43), bem como de abordagens emergentes baseadas em aprendizado por reforço e arquiteturas neurais especializadas para cinemática inversa (S4, S5, S8, S20, S28, S39).

Em síntese, os resultados desta revisão sistemática indicam que diversas arquiteturas de redes neurais têm sido aplicadas ao problema de cinemática inversa, incluindo modelos clássicos, arquiteturas profundas, redes recorrentes e abordagens híbridas. Contudo, redes neurais *feedforward*, particularmente perceptrons multicamadas, permanecem como a abordagem dominante na literatura. Embora arquiteturas mais avançadas tenham começado a aparecer em estudos recentes, sua adoção ainda é relativamente limitada, indicando oportunidades para futuras pesquisas explorarem arquiteturas modernas de aprendizado profundo e estruturas híbridas aplicadas à cinemática inversa.

QR2 – Quais são os tipos de sistemas robóticos utilizados nos experimentos relacionados às redes neurais artificiais aplicadas à cinemática inversa?

Esta questão de pesquisa teve como objetivo identificar quais tipos de sistemas robóticos foram utilizados nos experimentos envolvendo redes neurais artificiais aplicadas ao problema de cinemática inversa. Para responder a essa questão, foram analisados os sistemas robóticos descritos nos 52 estudos primários incluídos nesta revisão sistemática da literatura.

A análise revelou que diferentes tipos de manipuladores e plataformas robóticas foram empregados nos experimentos. Entretanto, observa-se uma predominância significativa de manipuladores robóticos seriais, particularmente robôs industriais utilizados em tarefas de manipulação e automação, conforme evidenciado na maioria dos estudos analisados (S1–S5, S10–S31, S33–S36, S38–S42, S44–S50, S52). Esses manipuladores incluem tanto robôs industriais reais quanto modelos genéricos ou simplificados utilizados em ambientes de simulação, frequentemente empregados em etapas iniciais de validação de algoritmos de aprendizado.

Com base nas características apresentadas nos estudos analisados, os sistemas robóticos foram agrupados em categorias de acordo com sua arquitetura cinemática. A 11 apresenta a distribuição dos

tipos de sistemas robóticos observados na literatura.

Tabela 11 – Tipos de sistemas robóticos utilizados nos estudos analisados

Tipo de sistema robótico	Número de estudos	Percentual	Estudos (IDs) e exemplos
Manipuladores seriais	43	83%	S1–S5, S10–S31, S33–S36, S38–S42, S44–S50, S52 (ex.: UR5, KUKA LBR iiwa, PUMA, Franka Emika Panda)
Sistemas biomiméticos / humanoides	4	8%	S6 – TIAGo / TIAGo++, S8 – NAO V6, S19 – NAO (braço esquerdo), S24 – NAO H25
Sistemas híbridos ou cooperativos	2	4%	S32 – ABB IRB 120 (<i>dual-arm</i>), S51 – manipulador móvel com braço robótico
Robôs paralelos	1	2%	S43 – ABB IRB 360 (Delta)
Sistemas especiais (exoesqueleto, matriz acústica, CNC)	2	4%	S9 – exoesqueleto de membro superior, S37 – <i>Phased Transducer Array</i> (PTA)

Fonte: Elaborado pelo autor.

Os resultados indicam que mais de quatro quintos dos estudos analisados utilizam manipuladores seriais, evidenciando que essa arquitetura continua sendo o principal cenário experimental para o estudo da cinemática inversa baseada em aprendizado de máquina. Essa predominância pode ser explicada pela ampla utilização desses manipuladores em aplicações industriais, bem como pela disponibilidade de modelos cinemáticos amplamente estabelecidos na literatura, o que facilita o desenvolvimento e a validação de algoritmos baseados em aprendizado de máquina.

Entre os robôs industriais identificados nos estudos analisados destacam-se modelos amplamente utilizados na pesquisa em robótica, como UR5/UR5e, KUKA LBR iiwa, ABB IRB 2400, ABB IRB 120, PUMA 260, PUMA 560, Franka Emika Panda e Sawyer (S1, S4, S11, S13, S22, S23, S27, S32, S35, S39, S45). Esses sistemas são frequentemente utilizados em pesquisas devido à sua versatilidade, precisão e à ampla disponibilidade de modelos cinemáticos documentados, o que facilita tanto a simulação quanto a implementação experimental.

Além dos manipuladores industriais, diversos estudos também empregaram manipuladores genéricos ou simplificados, como manipuladores planares de duas ou três juntas, manipuladores SCARA ou configurações articulares do tipo RRR e 6R (S2, S3, S5, S10, S14, S16, S18, S20, S21, S24, S29, S30, S31, S33, S34, S36, S38, S40, S41, S42, S44, S46, S47, S48, S49, S50, S52). Esses modelos são frequentemente utilizados em etapas iniciais de validação de algoritmos, pois permitem avaliar o desempenho das redes neurais em cenários com menor complexidade cinemática.

Outra categoria identificada na literatura inclui robôs humanoides e sistemas biomiméticos, como o robô humanoide NAO e o robô de serviço TIAGo, utilizados principalmente em estudos relacionados à

interação humano–robô e aplicações assistivas (S6, S8, S19, S24).

Também foram identificados sistemas robóticos especializados, incluindo exoesqueletos para membros superiores, manipuladores móveis, máquinas CNC de múltiplos eixos e matrizes acústicas de transdutores (S9, S37, S51). Embora menos frequentes, esses sistemas demonstram a aplicabilidade das abordagens baseadas em redes neurais em diferentes domínios da robótica, além da manipulação industrial tradicional.

A presença de robôs paralelos na literatura analisada foi bastante limitada, sendo identificado apenas um estudo utilizando um manipulador do tipo Delta (ABB IRB 360) (S43). Da mesma forma, alguns trabalhos investigaram arquiteturas híbridas, como manipuladores cooperativos de braço duplo ou sistemas que combinam base móvel e manipulador robótico (S32, S51).

Em relação à complexidade cinemática, observou-se que a maioria dos experimentos foi conduzida com manipuladores contendo seis graus de liberdade, configuração típica de robôs industriais capazes de posicionar e orientar o efetuador final no espaço tridimensional. Também foram identificados estudos envolvendo manipuladores redundantes com sete graus de liberdade, como os robôs KUKA LBR iiwa, Franka Emika Panda e Sawyer, que apresentam múltiplas soluções possíveis para o problema de cinemática inversa (S13, S23, S27, S39).

A 12 apresenta os principais robôs industriais identificados nos estudos analisados, bem como suas características cinemáticas.

Tabela 12 – Robôs industriais identificados nos estudos analisados

Robô industrial	ID do estudo	Observação
UR5 / UR5e	S4, S11, S35	Serial, 6 DoF
KUKA LBR iiwa	S13, S23	Serial, 7 DoF
ABB IRB 2400	S1, S13	Serial, 6 DoF
PUMA 260	S22	Serial, 6 DoF
PUMA 560	S13	Serial, 6 DoF
Franka Emika Panda	S27	Serial redundante, 7 DoF
Sawyer	S39	Serial redundante, 7 DoF

Fonte: Elaborado pelo autor.

De modo geral, os resultados indicam que a literatura sobre redes neurais aplicadas à cinemática inversa concentra-se predominantemente em manipuladores robóticos seriais industriais, especialmente aqueles com seis ou sete graus de liberdade. Outros tipos de sistemas robóticos aparecem com menor frequência e estão geralmente associados a aplicações específicas ou contextos experimentais particulares.

QR3 – Quais plataformas ou ambientes de teste foram utilizados nos experimentos?

Esta questão de pesquisa teve como objetivo identificar quais plataformas ou ambientes experimentais foram utilizados nos estudos que aplicam redes neurais artificiais à resolução do problema de cinemática inversa. Para responder a essa questão, foram analisados os ambientes de teste descritos nos 52 estudos primários incluídos nesta revisão sistemática da literatura.

A análise revelou que os experimentos foram conduzidos em diferentes tipos de ambientes experimentais. De modo geral, foram identificadas cinco categorias principais de ambientes de teste, conforme apresentado na 13.

Tabela 13 – Ambientes experimentais utilizados nos estudos analisados

Categoria de ambiente	Estudos	Observação
Simulação	S5, S6, S10, S11, S14, S15, S16, S17, S22, S24, S25, S29, S30, S31, S33, S40, S41, S42, S44, S45, S46, S47, S48	Estudos conduzidos exclusivamente em ambiente virtual.
Simulação + Robô Real (<i>Sim-to-Real</i>)	S1, S3, S4, S7, S9, S11, S13, S19, S20, S23, S26, S27, S32, S34, S35, S36, S37, S38, S39, S43, S47	Treinamento realizado em simulação com posterior validação em robôs físicos.
Robô Real	S2, S24	Experimentos conduzidos diretamente em hardware físico.
Revisão / Conceitual	S8, S28	Artigos de revisão da literatura sem experimentação direta.
Ambiente específico	S49	Experimentos conduzidos em cenários específicos, como terrenos irregulares com manipulador móvel.

Fonte: Elaborado pelo autor.

Observa-se que a maior parte dos estudos foi conduzida em ambientes de simulação. Especificamente, 25 estudos realizaram experimentos exclusivamente em simulação, enquanto 22 trabalhos adotaram uma abordagem *Sim-to-Real*, combinando treinamento em simulação com validação em robôs físicos (S1, S3, S4, S7, S9, S11, S13, S19, S20, S23, S26, S27, S32, S34, S35, S36, S37, S38, S39, S43, S47). Por outro lado, apenas dois estudos realizaram experimentos exclusivamente em hardware físico, conforme observado em (S2, S24).

De forma geral, aproximadamente 88% dos estudos utilizaram ambientes de simulação em alguma etapa do processo experimental. Esse resultado evidencia a forte dependência da comunidade científica de plataformas simuladas para o treinamento, validação inicial e desenvolvimento de algoritmos baseados em aprendizado de máquina aplicados à robótica.

Essa predominância pode ser explicada principalmente pela necessidade de grandes volumes de dados de interação, especialmente em métodos baseados em aprendizado por reforço profundo, como observado em diversos estudos da revisão (S1, S3, S4, S5, S6, S7, S9, S10, S11, S13, S14, S15, S16, S17, S19, S20, S22, S23, S24, S25, S26, S27, S29, S30, S31, S32, S33, S34, S35, S36, S40). Nesse contexto, ambientes simulados permitem gerar dados em larga escala de forma segura e com menor custo computacional. Além disso, fatores práticos como segurança experimental, redução de custos e maior controle sobre as condições do experimento também contribuem para essa preferência.

Os principais simuladores identificados nos experimentos e que aparecem com maior frequência na literatura são apresentados na 14.

Tabela 14 – Principais simuladores e ambientes utilizados nos estudos analisados

Simulador / Ambiente	Exemplos de Estudos
MATLAB / Simulink / Simscape	S3, S6, S13, S22, S31, S33, S40, S41, S46, S47, S48, S49
Gazebo (ROS)	S7, S11, S23, S35, S42, S47
CoppeliaSim	S1, S3, S34
Unity	S11, S20
PyBullet / OpenAI Gym	S19
MuJoCo	S11

Fonte: Elaborado pelo autor.

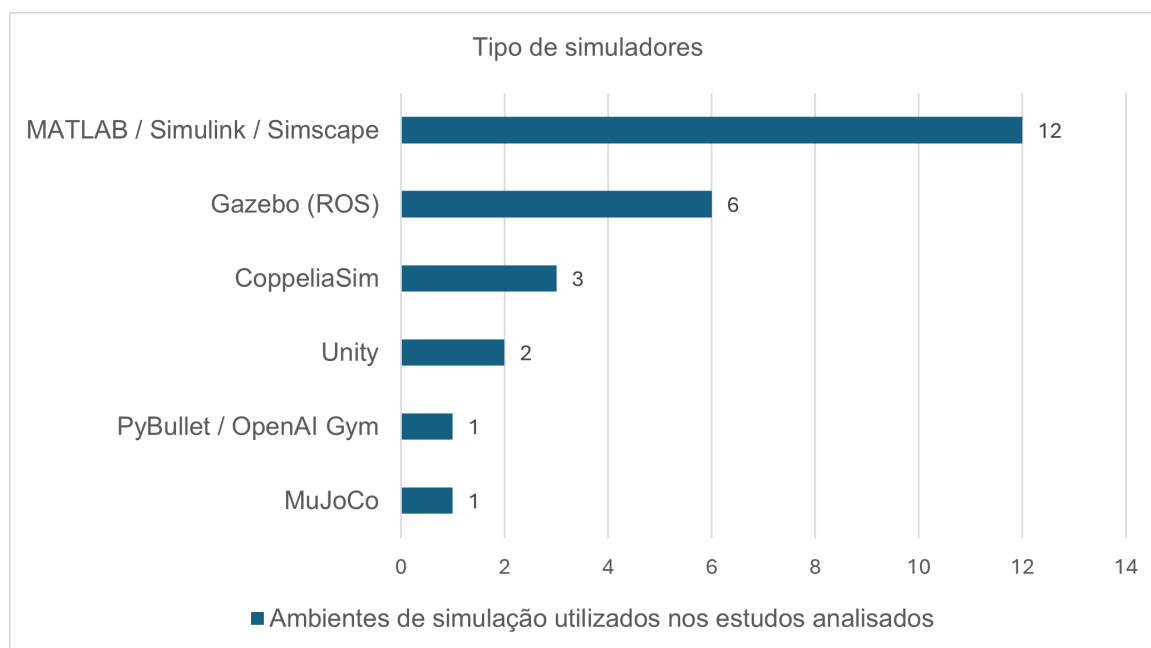
Entre os simuladores identificados nos estudos analisados, destacam-se principalmente MATLAB e suas extensões (Simulink e Simscape), amplamente utilizados em pesquisas focadas em modelagem dinâmica e validação de algoritmos de controle. Esses ambientes aparecem em diversos trabalhos da literatura analisada (S3, S6, S13, S22, S31, S33, S40, S41, S46, S47, S48, S49).

Outro ambiente frequentemente utilizado é o simulador Gazebo integrado ao framework ROS, particularmente em estudos que buscam maior proximidade com aplicações robóticas reais (S7, S11, S23, S35, S42, S47).

Outros ambientes de simulação identificados incluem CoppeliaSim (S1, S3, S34), frequentemente utilizado em experimentos envolvendo manipuladores robóticos, e o motor gráfico Unity, empregado em cenários de simulação visual e ambientes virtuais mais complexos (S11, S20). Simuladores como PyBullet/OpenAI Gym (S19) e MuJoCo (S11) aparecem de forma mais pontual, geralmente associados a pesquisas que exploram aprendizado por reforço profundo e simulação física de alta fidelidade.

A distribuição dos ambientes de simulação identificados na literatura também pode ser observada na Figura 7, que apresenta a frequência das plataformas utilizadas nos estudos analisados.

Figura 7 – Frequência dos ambientes de simulação utilizados nos estudos



Fonte: Elaborado pelo autor.

Conforme ilustrado na Figura 7, observa-se uma predominância clara do uso do ambiente MATLAB e de suas extensões (Simulink e Simscape) nos estudos analisados, sendo identificado em 12 trabalhos da literatura selecionada (S3, S6, S13, S22, S31, S33, S40, S41, S46, S47, S48, S49).

Em seguida, destaca-se o simulador Gazebo integrado ao ROS, presente em seis estudos (S7, S11, S23, S35, S42, S47).

Outros ambientes também foram identificados, embora com menor frequência, como CoppeliaSim (S1, S3, S34) e Unity (S11, S20). Simuladores como PyBullet/OpenAI Gym (S19) e MuJoCo (S11) aparecem em apenas um estudo cada.

De forma geral, os resultados indicam uma forte predominância de ambientes tradicionais de modelagem e simulação, particularmente aqueles baseados em MATLAB, embora simuladores mais modernos, voltados ao aprendizado de máquina e à robótica baseada em aprendizado por reforço, também estejam presentes na literatura analisada.

A 15 apresenta a classificação dos paradigmas de aprendizado de máquina identificados nos estudos analisados, bem como exemplos de métodos utilizados em cada abordagem.

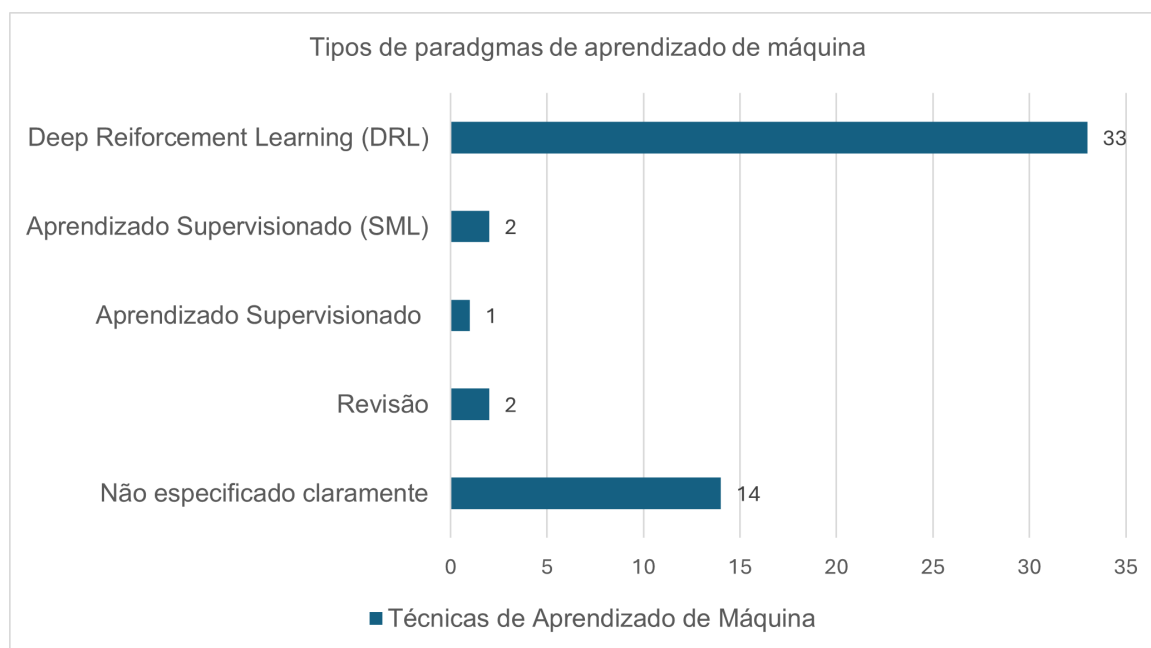
Tabela 15 – Paradigmas de aprendizado de máquina analisados na literatura

Paradigma de Aprendizado	Nº de Estudos	Estudos (IDs)	Exemplos de Métodos
Deep Reinforcement Learning (DRL)	33	S1, S3, S4, S5, S6, S7, S9, S10, S11, S13, S14, S15, S16, S17, S19, S20, S22, S23, S24, S25, S26, S27, S29, S30, S31, S32, S33, S34, S35, S36, S40	DDPG, PPO, SAC, TD3, DQN
Aprendizado Supervisionado (SML)	2	S51, S52	Deep ANN, Extreme Learning Machine
Aprendizado não supervisionado / SOM	1	S50	Kohonen Self-Organizing Map
Revisão / Survey	2	S8, S28	Revisão de RL em robótica
Não especificado claramente	14	S37, S38, S39, S41, S42, S43, S44, S45, S46, S47, S48	DRL mencionado genericamente ou método não detalhado

Fonte: Elaborado pelo autor.

A distribuição dos paradigmas de aprendizado de máquina indica uma forte predominância de métodos baseados em Aprendizado por Reforço Profundo (*Deep Reinforcement Learning – DRL*) na literatura analisada (S1, S3, S4, S5, S6, S7, S9, S10, S11, S13, S14, S15, S16, S17, S19, S20, S22, S23, S24, S25, S26, S27, S29, S30, S31, S32, S33, S34, S35, S36, S40), conforme também ilustrado na Figura 8.

Figura 8 – Distribuição dos paradigmas de aprendizado de máquina



Fonte: Elaborado pelo autor.

A distribuição dos paradigmas de aprendizado de máquina indica uma forte predominância de métodos baseados em Aprendizado por Reforço Profundo (DRL) na literatura analisada (S1, S3, S4, S5, S6, S7, S9, S10, S11, S13, S14, S15, S16, S17, S19, S20, S22, S23, S24, S25, S26, S27, S29, S30, S31, S32, S33, S34, S35, S36, S40), conforme também ilustrado na Figura 8.

No conjunto dos estudos primários identificados, 33 trabalhos, correspondendo a aproximadamente 62% da amostra analisada, utilizam explicitamente algoritmos de Deep Reinforcement Learning. Em contrapartida, apenas dois estudos utilizam aprendizado supervisionado tradicional, conforme observado em (S51, S52), enquanto um estudo emprega aprendizado não supervisionado baseado em Self-Organizing Maps (S50).

Além disso, foram identificados dois artigos de revisão da literatura (S8, S28). Por outro lado, constatou-se que em diversos estudos o paradigma de aprendizado não foi claramente especificado, sendo descrito apenas de forma genérica ou sem detalhamento metodológico suficiente (S37, S38, S39, S41, S42, S43, S44, S45, S46, S47, S48).

Esses resultados indicam que o aprendizado por reforço profundo se consolidou como o paradigma dominante nas pesquisas recentes envolvendo controle robótico e cinemática inversa baseada em aprendizado de máquina. A predominância do DRL pode ser atribuída à sua capacidade de lidar com problemas de controle contínuo e tomada de decisão sequencial, características intrínsecas de sistemas robóticos com múltiplos graus de liberdade.

Diferentemente das abordagens supervisionadas tradicionais — que aprendem um mapeamento

direto entre entrada e saída, como posição cartesiana e configuração articular — os algoritmos de aprendizado por reforço permitem que o agente aprenda políticas de controle diretamente a partir da interação com o ambiente, conforme demonstrado em diversos trabalhos da revisão (S1, S3, S4, S7, S9, S10, S11, S13, S19, S20, S23, S26, S27, S32, S34, S35, S36).

Essa característica torna o DRL particularmente adequado para tarefas robóticas complexas, como manipulação de objetos, planejamento de trajetória, controle de força e torque e tarefas de montagem, amplamente exploradas na literatura recente.

Por outro lado, os poucos trabalhos baseados em aprendizado supervisionado identificados na revisão sistemática tendem a focar em aproximações diretas da solução da cinemática inversa, utilizando redes neurais profundas ou métodos de aprendizado rápido, como Extreme Learning Machines, conforme observado em (S51, S52).

Em síntese, os resultados desta análise sugerem uma transição metodológica na literatura de robótica baseada em aprendizado, caracterizada pela migração de abordagens supervisionadas clássicas para métodos baseados em aprendizado por reforço profundo. Enquanto métodos supervisionados continuam sendo utilizados para aproximações diretas da cinemática inversa, os algoritmos de DRL vêm sendo cada vez mais adotados para aprender políticas de controle adaptativas em ambientes dinâmicos, ampliando o escopo de aplicações da inteligência artificial em robótica.

QR4 – Quais são os principais datasets utilizados para treinar e validar modelos de aprendizado de máquina baseados em redes neurais para cinemática inversa?

A análise dos estudos primários revela que os *datasets* utilizados para o treinamento de redes neurais aplicadas à cinemática inversa podem ser classificados de acordo com o método de geração dos dados.

De modo geral, observa-se que a maioria dos estudos utiliza conjuntos de dados sintéticos gerados a partir de modelos cinemáticos ou simuladores robóticos, enquanto uma parcela menor utiliza dados coletados diretamente de robôs físicos.

A 16 apresenta uma taxonomia consolidada dos principais métodos de geração de *datasets* identificados na literatura.

Tabela 16 – Taxonomia dos métodos de geração de *datasets* para cinemática inversa

Categoria de geração de dados	Descrição	Estudos representativos
Cinemática direta – <i>Forward Kinematics</i> (FK)	Geração de pares entrada-saída utilizando equações de cinemática direta para mapear variáveis de junta para posições cartesianas.	S2, S3, S9, S15, S17, S22, S31, S33, S41, S45, S49
Amostragem aleatória do espaço de juntas	Geração pseudoaleatória de ângulos de juntas seguida da aplicação da FK para gerar posições do efetuador.	S7, S15, S17, S36
Amostragem estruturada do <i>workspace</i>	Divisão sistemática do espaço de trabalho (quadrantes, grades ou camadas) para garantir melhor cobertura espacial.	S6, S27, S41
Trajetórias simuladas	Geração de dados a partir de trajetórias contínuas, como sinais sinusoidais, espirais ou movimentos naturais.	S3, S10, S16, S40
Simulação física em ambientes robóticos	Geração de dados utilizando simuladores robóticos como MuJoCo, Webots ou V-REP.	S4, S11, S19, S42
Simulação + robô real (<i>Sim-to-Real</i>)	Combinação de dados simulados com medições obtidas em robôs físicos para validação ou adaptação do modelo.	S11, S25, S38, S47
Coleta experimental real	Dados coletados diretamente em robôs físicos durante a execução de movimentos ou tarefas.	S24
Frameworks especializados ou algoritmos de otimização	Dados gerados por algoritmos evolutivos ou frameworks específicos (ex.: BioIK).	S10

Fonte: Elaborado pelo autor.

De acordo com a taxonomia apresentada, foi possível evidenciar que a geração sintética de dados baseada na aplicação da cinemática direta constitui o método dominante, sendo adotada na maioria dos estudos analisados neste trabalho.

A 16 apresenta a classificação detalhada dos estudos selecionados de acordo com o tipo de *dataset* utilizado para o treinamento dos modelos de aprendizado de máquina aplicados à cinemática inversa. Cada estudo foi identificado por um ID único (S1–S52), permitindo rastrear sua categorização nas diferentes classes de *datasets* consideradas nesta revisão. Essa organização facilita a identificação das abordagens predominantes na literatura e contribui para garantir maior transparência no processo de classificação dos estudos.

A partir dessa classificação, é possível observar a distribuição quantitativa dos tipos de *datasets* utilizados nos estudos analisados. A 17 apresenta o número de trabalhos e o percentual aproximado correspondente a cada categoria de *dataset* identificada.

Tabela 17 – Distribuição dos tipos de *datasets* utilizados nos estudos analisados

Tipo de dataset	Nº de estudos	Percentual aproximado
Sintético	40	75%
Híbrido	10	19%
Real	2	4%
Revisão	1	2%

Fonte: Elaborado pelo autor.

A 16 apresenta a classificação dos estudos selecionados de acordo com o tipo de *dataset* utilizado no treinamento de modelos de aprendizado de máquina aplicados à cinemática inversa. Observa-se uma predominância significativa de *datasets* sintéticos, empregados na maioria dos estudos analisados. Esses conjuntos de dados são geralmente gerados por meio de modelos de cinemática direta combinados com estratégias de amostragem do espaço de juntas.

Um número menor de estudos utiliza *datasets* híbridos, que combinam dados provenientes de simulação com dados coletados em robôs reais, enquanto apenas uma pequena parcela dos trabalhos emprega exclusivamente dados experimentais. Essa distribuição evidencia a forte dependência da literatura em relação a ambientes de simulação para a geração de dados de treinamento.

QR5 – Quais são as principais métricas utilizadas para avaliar o desempenho dos modelos no treinamento e teste das redes neurais artificiais?

Em relação às principais métricas utilizadas para avaliar o desempenho dos modelos durante as etapas de treinamento e teste das redes neurais artificiais, a análise dos estudos primários revela uma ampla variedade de indicadores empregados para avaliar o desempenho de modelos aplicados à cinemática inversa de manipuladores robóticos.

De modo geral, essas métricas podem ser categorizadas em quatro grupos principais: métricas de erro numérico, métricas angulares e cartesianas, métricas relacionadas ao tempo e à eficiência computacional, e métricas de classificação ou métricas híbridas.

A 18 apresenta as métricas identificadas nos estudos analisados, indicando quais indicadores foram utilizados em cada trabalho e suas respectivas abordagens de avaliação.

Tabela 18 – Métricas de avaliação e desempenho dos estudos analisados

Categoria	Métrica	Ocorrência na Tabela	Observações
Erro Numérico	MSE (Mean Squared Error)	S3, S6, S7, S10, S25, S14, S15, S16, S17, S1, S19, S24, S13, S26, S30, S31, S32, S33, S34, S35, S36, S38, S40, S42, S46, S49, S52	Usada como função de perda e métrica de validação; muito comum em regressão de ângulos/juntas
Erro Numérico	MAE (Mean Absolute Error)	S3, S5, S1, S13, S29, S38, S45, S48	Complementa MSE, avalia magnitude média do erro
Erro Numérico	RMSE (Root Mean Square Error)	S4, S1, S13, S38, S47, S48	Captura dispersão dos erros em unidades físicas consistentes
Erro Numérico	R^2 (Coeficiente de Determinação)	S1, S38, S45, S48	Mede a proporção da variabilidade explicada pelo modelo
Erro Numérico	Erro percentual / Porcentagem de acerto	S22, S13, S32	Avalia desempenho relativo das juntas ou posições
Angular	Erro de ângulo de junta, Euler, quaternions	S2, S4, S7, S10, S11, S1, S22, S13, S32, S38, S48	Avalia precisão das articulações; usado em conjunto com métricas cartesianas
Cartesiana	Distância Euclidiana, erro de posição, distância	S2, S4, S7, S9, S11, S1, S24, S13, S32, S38, S48	Avalia precisão do posicionamento do efetuador no espaço
Cartesiana	Distâncias geodésicas, Riemannian distance, Euclid distance	S7, S25, S47	Métricas avançadas para erros de orientação ou dinâmica inversa
Tempo / Eficiência	Tempo de inferência, tempo total de treinamento, número de épocas	S4, S9, S10, S25, S16, S1, S13, S27, S36, S42, S50, S52	Avalia aplicabilidade em tempo real e eficiência computacional

Continua na próxima página

Categoria	Métrica	Ocorrência na Ta- bela	Observações
Tempo / Eficiência	Move Time, Response Time, Memory Usage, Manipulabilidade	S9, S10, S25, S50	Complementa avaliação de desempenho prático
Classificação / Híbrido	Accuracy, Precision, Recall, F1-score	S5, S23, S13, S43	Avalia modelos de classificação ou seleção de soluções em manipuladores redundantes
Classificação / Híbrido	Fitness Score, Reward, Success Rate	S10, S39	Usado em aprendizado por reforço ou métodos híbridos

Fonte: Elaborado pelo autor.

As métricas de erro numérico são predominantes e incluem, principalmente, o Mean Squared Error (MSE), utilizado em grande parte dos estudos tanto como função de perda quanto como métrica de validação (S3, S6, S7, S14, S24, S26, S38). O Mean Absolute Error (MAE) e o Root Mean Square Error (RMSE) são frequentemente empregados em conjunto, permitindo avaliar tanto a magnitude absoluta do erro quanto a dispersão em torno da média (S3, S4, S5, S29, S38). Adicionalmente, métricas de ajuste, como o coeficiente de determinação R^2 , são aplicadas para quantificar a proporção da variabilidade dos dados explicada pelo modelo (S1, S45), enquanto erros percentuais ou porcentagens de acerto são utilizados para avaliar o desempenho relativo em diferentes juntas do manipulador (S20, S24, S32). Em alguns estudos, funções de perda compostas incorporam múltiplos termos de erro e regularização, integrando informações angulares e cartesianas para melhorar a generalização do modelo (S1, S4, S22).

As métricas angulares e cartesianas refletem a natureza física do problema da cinemática inversa. Métricas angulares, como erro de ângulo de junta, ângulos de Euler e quaternions, são utilizadas para avaliar a precisão da rede na predição das configurações articulares do manipulador (S2, S4, S7, S48). Métricas cartesianas, como distância Euclidiana do efetuador e erro de trajetória, permitem avaliar a capacidade do modelo de posicionar o efetuador final corretamente no espaço de trabalho (S2, S4, S7, S11, S38). A combinação dessas métricas em muitos estudos fornece uma avaliação mais completa, capturando tanto a exatidão das articulações quanto a precisão do posicionamento no espaço (S2, S4, S7, S13, S38). Métricas adicionais, como distâncias geodésicas, Riemannian distance, Fréchet distance e erro de atitude, também são reportadas em trabalhos que abordam problemas de orientação complexa ou dinâmica inversa (S7, S25, S47).

Além disso, várias pesquisas consideram métricas de tempo e eficiência computacional, refletindo

a necessidade de redes neurais capazes de operar em tempo real. Entre essas, destacam-se o tempo médio de inferência por amostra ou ponto, o tempo total de treinamento e o número de épocas até convergência (S4, S9, S25, S16, S1, S26, S36, S50, S51, S52). Estudos mais recentes incluem também métricas adicionais de desempenho, como consumo de memória, Move Time, Response Time e índices de manipulabilidade, evidenciando uma preocupação crescente com a aplicabilidade prática dos modelos (S9, S10, S25, S50).

Por fim, algumas abordagens híbridas ou baseadas em classificação empregam métricas de acurácia, precisão, recall e F1-score, bem como Fitness Score, Reward e Success Rate, principalmente em cenários envolvendo manipuladores redundantes ou aprendizado por reforço (S5, S10, S23, S13, S39, S43). Essas métricas são essenciais para avaliar a capacidade do modelo em selecionar soluções viáveis entre múltiplas alternativas ou otimizar trajetórias complexas, indo além da simples previsão de valores numéricos.

Em síntese, os estudos mostram que a escolha da métrica de avaliação está fortemente associada à arquitetura da rede, ao tipo de manipulador e à natureza da tarefa. A combinação de métricas angulares e cartesianas com métricas de erro numérico e eficiência computacional tende a fornecer a avaliação mais robusta do desempenho de redes neurais para cinemática inversa, permitindo não apenas comparar modelos, mas também garantir aplicabilidade em ambientes de controle em tempo real.

QR6 – Quais são os desafios e limitações da aplicação de redes neurais artificiais na cinemática inversa?

A partir da análise dos estudos primários investigados, observa-se que a aplicação de redes neurais artificiais (ANNs) na solução do problema de cinemática inversa (*Inverse Kinematics* – IK) apresenta diversos desafios e limitações. De modo geral, esses desafios podem ser organizados em cinco categorias principais: necessidade de grandes volumes de dados, problemas relacionados a singularidades e múltiplas soluções, desafios computacionais e de otimização, limitações de generalização entre simulação e sistemas reais, e complexidade de modelagem associada ao ajuste de hiperparâmetros.

Diversos estudos destacam que a eficácia das ANNs depende fortemente da disponibilidade de grandes conjuntos de dados para treinamento e validação (S2, S4, S15, S47, S50). Entretanto, a coleta de dados em ambientes reais é frequentemente cara, demorada e, em alguns casos, inviável, o que restringe a aplicabilidade prática dessas abordagens (S4). Além disso, dados ruidosos ou incompletos podem comprometer a precisão dos modelos e aumentar o risco de *overfitting* (S22, S13, S50).

Outro desafio relevante está relacionado à presença de singularidades nas configurações do manipulador robótico. Nessas situações, pequenas variações na posição desejada do efetuador podem

gerar grandes variações nos ângulos das juntas, resultando em soluções instáveis ou não únicas (S2, S25, S1, S19). Modelos supervisionados baseados em ANNs também apresentam dificuldades em lidar com problemas de mapeamento um-para-muitos, nos quais múltiplas configurações articulares podem corresponder à mesma posição do efetuador final (S24, S27). Esse fenômeno limita a robustez e a confiabilidade das soluções obtidas.

Além disso, o treinamento de redes neurais aplicadas à IK pode envolver custos computacionais elevados, especialmente em manipuladores com múltiplos graus de liberdade ou em problemas de alta dimensionalidade (S4, S15, S42, S43). Algumas abordagens baseadas em técnicas metaheurísticas ou híbridas também enfrentam dificuldades de convergência e podem ficar presas em mínimos locais (S10, S15, S27). Em muitos casos, métodos que buscam maximizar a precisão acabam sacrificando a velocidade de execução, o que dificulta sua aplicação em sistemas robóticos que demandam operação em tempo real (S52).

Outro aspecto frequentemente relatado na literatura refere-se às limitações de generalização dos modelos. Redes treinadas exclusivamente em ambientes simulados nem sempre apresentam bom desempenho quando aplicadas em sistemas robóticos reais (S8, S28, S22). Problemas adicionais podem surgir em regiões do espaço de trabalho pouco representadas no conjunto de treinamento (S6, S24), em manipuladores com parâmetros mecânicos incertos ou sujeitos a variações ambientais (S50, S51), bem como em cenários que envolvem interação humana ou ambientes dinâmicos (S10). Essas limitações reforçam a importância de validações experimentais em robôs físicos.

Por fim, a performance das ANNs aplicadas à cinemática inversa depende fortemente da escolha da arquitetura da rede e do ajuste adequado de hiperparâmetros (S13, S22). Redes com poucas camadas podem ser insuficientes para capturar a complexidade da relação entre o espaço cartesiano e o espaço de juntas (S40), enquanto manipuladores mais complexos frequentemente exigem arquiteturas mais profundas e modelos com maior capacidade representacional (S44, S36, S38, S39). No entanto, o aumento da complexidade da rede também pode dificultar o processo de treinamento e comprometer a capacidade de generalização. Além disso, a utilização isolada de determinadas métricas de avaliação, como o MSE, pode mascarar deficiências do modelo quando aplicado a trajetórias reais ou a tarefas robóticas mais complexas (S6).

QR7 – Quais lacunas ainda existem na literatura sobre o uso de redes neurais para resolver a cinemática inversa de robôs manipuladores?

Os desafios reportados pelos estudos primários revelam problemas recorrentes que ainda não foram completamente resolvidos pela literatura. Em particular, questões como a dependência de grandes

volumes de dados (S1, S3, S15, S47, S51), dificuldades associadas à presença de singularidades e à existência de múltiplas soluções (S1, S13, S18, S19, S23, S27), elevados custos computacionais e desafios relacionados à otimização dos modelos (S3, S15, S42, S43, S53), limitações de generalização entre ambientes simulados e sistemas reais (S5, S7, S9, S21, S23, S28, S51, S52) e a complexidade envolvida no ajuste de hiperparâmetros das redes neurais (S21, S25, S36, S38, S39, S40, S44) indicam áreas em que o conhecimento atual ainda permanece limitado.

A partir desses desafios, torna-se possível identificar lacunas estruturais na literatura, especialmente no que se refere à necessidade de padronização de protocolos experimentais, ao desenvolvimento de modelos mais robustos e generalizáveis e à realização de validações em cenários robóticos mais realistas. Conforme sintetizado na 19, diversas lacunas ainda permanecem abertas na pesquisa sobre o uso de redes neurais para a resolução da cinemática inversa de manipuladores robóticos.

Tabela 19 – Principais lacunas da literatura identificadas

Lacuna identificada	Descrição	Estudos relacionados
Falta de <i>datasets</i> padronizados	Muitos estudos utilizam dados sintéticos ou conjuntos de dados próprios, dificultando a reprodutibilidade e a comparação entre métodos.	S1, S3, S15, S21, S25, S47, S51
Escassez de validação em robôs reais	Grande parte dos experimentos é conduzida apenas em ambientes simulados, limitando a evidência de aplicabilidade prática.	S5, S7, S9, S21, S28
Limitações de generalização dos modelos	Modelos frequentemente apresentam degradação de desempenho em regiões do espaço de trabalho pouco representadas durante o treinamento.	S5, S21, S23, S51, S52
Tratamento insuficiente de singularidades e múltiplas soluções	A natureza multivalorada da cinemática inversa ainda representa um desafio significativo para redes neurais supervisionadas.	S1, S13, S18, S19, S23, S27
Falta de padronização nas métricas de avaliação	Diferentes estudos utilizam métricas distintas, dificultando comparações consistentes entre abordagens.	S5, S25, S36, S38
Custos computacionais e escalabilidade	O treinamento de redes profundas e modelos híbridos pode se tornar computacionalmente custoso em manipuladores com muitos graus de liberdade.	S3, S15, S42, S43, S53
Complexidade na definição da arquitetura e hiperparâmetros	Muitos estudos dependem de processos empíricos para definir a arquitetura da rede e ajustar hiperparâmetros.	S21, S25, S36, S38, S39, S40, S44

Fonte: Elaborado pelo autor

Entre as lacunas mais recorrentes identificadas destaca-se a ausência de *datasets* padronizados e publicamente disponíveis para o treinamento e a avaliação de modelos de redes neurais aplicados à

cinemática inversa. Muitos estudos utilizam conjuntos de dados sintéticos gerados especificamente para seus experimentos, como relatado em S1, S3, S15 e S51, o que limita a reprodutibilidade dos resultados e dificulta comparações diretas entre diferentes arquiteturas.

Outra lacuna relevante refere-se à validação experimental limitada em robôs físicos. Diversos trabalhos conduzem experimentos exclusivamente em ambientes de simulação, conforme observado em S5, S7, S9, S21 e S28. Embora a simulação permita gerar grandes volumes de dados de forma controlada, diferenças entre o ambiente simulado e o sistema físico podem comprometer o desempenho real do modelo.

Além disso, diversos estudos apontam limitações de generalização quando os modelos são aplicados a regiões do espaço de trabalho pouco representadas no conjunto de treinamento ou quando são utilizados em manipuladores com características distintas daquelas presentes durante o treinamento (S5, S21, S23, S51, S52). Esse problema evidencia a necessidade de abordagens capazes de aprender representações mais robustas e transferíveis.

A literatura também aponta dificuldades no tratamento de singularidades e múltiplas soluções válidas, características intrínsecas do problema de cinemática inversa. Estudos como S1, S13, S18 e S19 relatam instabilidades nas previsões da rede próximas a regiões singulares, enquanto trabalhos como S23 e S27 destacam limitações de redes supervisionadas para lidar com mapeamentos um-para-muitos.

Adicionalmente, observa-se falta de padronização nas métricas de avaliação, uma vez que diferentes estudos utilizam combinações variadas de métricas de erro, métricas angulares, métricas cartesianas e métricas de eficiência computacional (S5, S25, S36, S38). Essa diversidade dificulta a comparação direta entre diferentes modelos.

Por fim, desafios relacionados a custos computacionais elevados e à complexidade do ajuste de hiperparâmetros ainda representam barreiras para a aplicação prática dessas abordagens, especialmente em manipuladores com muitos graus de liberdade (S3, S15, S42, S43, S53) e em arquiteturas profundas que exigem extensa experimentação para definição adequada de sua configuração (S21, S25, S36, S38, S39, S40, S44).

Diante das lacunas identificadas, esta pesquisa contribui ao consolidar de forma sistemática o conhecimento existente sobre o uso de redes neurais artificiais para a resolução do problema de cinemática inversa em manipuladores robóticos. A revisão sistemática conduzida permitiu mapear arquiteturas de redes neurais empregadas, tipos de manipuladores analisados, conjuntos de dados utilizados, ambientes experimentais adotados, métricas de avaliação e principais desafios reportados nos estudos primários.

Ao integrar essas dimensões analíticas, este trabalho fornece uma visão estruturada do estado da arte, identificando tendências de pesquisa, limitações metodológicas e oportunidades para investigações

futuras. Além disso, ao evidenciar lacunas relacionadas à generalização dos modelos, à padronização de *datasets*, à validação em sistemas robóticos reais e à definição de métricas de avaliação consistentes, esta pesquisa oferece direcionamentos para o desenvolvimento de abordagens mais robustas e comparáveis, contribuindo para o avanço da aplicação de técnicas de aprendizado de máquina em problemas de cinemática inversa.

A análise dos estudos selecionados nesta revisão sistemática permitiu identificar diferentes abordagens para a aplicação de redes neurais artificiais na resolução do problema de cinemática inversa em manipuladores robóticos. As questões de pesquisa investigaram, respectivamente, as arquiteturas de redes neurais empregadas (RQ1), os tipos de robôs analisados (RQ2), os ambientes experimentais utilizados (RQ3), os *datasets* empregados para treinamento dos modelos (RQ4), as métricas de avaliação adotadas (RQ5), os principais desafios e limitações identificados (RQ6) e, por fim, as lacunas de pesquisa existentes (RQ7).

Como forma de sintetizar as principais características metodológicas dos estudos analisados, a 20 apresenta a codificação dos estudos incluídos nesta revisão sistemática.

Tabela 20 – Codificação dos estudos selecionados na revisão sistemática

ID	Ano	Robô / Sistema	Tipo	Arquitetura de Rede	Dataset	Plataforma	Métrica
S1	2025	Robô antropomórfico 6-DoF	Serial	XAI-ANN	Sintético	Simulação	MSE
S2	2025	Manipulador robótico	Serial	ANN	Sintético	Simulação	MSE
S3	2025	Manipulador 3-link	Serial	Deep Learning	Sintético	Simulação	MSE
S4	2025	Manipulador robótico	Serial	Liquid Neural Network	Sintético	Simulação	MSE
S5	2025	Manipulador robótico	Serial	Sequential NN	Híbrido	Simulação + Real	RMSE
S6	2025	Robô revoluto-prismatic	Híbrido	Deep Learning	Sintético	Simulação	MSE

Continua na próxima página

ID	Ano	Robô / Sistema	Tipo	Arquitetura de Rede	Dataset	Plataforma	Métrica
S7	2025	Manipulador robótico	Serial	Deep Neural Network	Sintético	Simulação	MSE
S8	2025	Manipuladores robóticos	Revisão	Deep Learning	Revisão	—	—
S9	2025	Robô bípedo humanoide	Humanoide	Neural Network	Sintético	Simulação	MSE
S10	2025	Exoesqueleto 2-DoF	Exoesqueleto	ANN	Sintético	Simulação	MSE
S11	2025	Manipulador robótico	Serial	KineNN	Sintético	Simulação	MSE
S12	2024	Manipulador robótico	Serial	Deep Learning + EA	Híbrido	Simulação	RMSE
S13	2024	Manipulador robótico	Serial	Physics-informed ML	Sintético	Simulação	MSE
S14	2024	Manipulador planar	Serial	Feedforward NN	Sintético	MATLAB	MSE
S15	2024	Manipulador robótico	Serial	ANN + Metaheurística	Híbrido	Simulação	RMSE
S16	2024	Braço robótico	Serial	ANN	Sintético	Simulação	MSE
S17	2024	Robô SCARA	SCARA	ANN + Metaheurística	Híbrido	Simulação	RMSE
S18	2024	Braço robótico humanoide	Humanoide	Deep NN	Híbrido	Simulação	RMSE
S19	2024	Manipulador 6-DoF	Serial	Deep Reinforcement Learning	Sintético	Simulação	Reward
S20	2024	PUMA 260	Serial	ANN	Sintético	Simulação	MSE

Continua na próxima página

ID	Ano	Robô / Sistema	Tipo	Arquitetura de Rede	Dataset	Plataforma	Métrica
S21	2023	Manipulador redundante	Serial	Machine Learning	Sintético	Simulação	RMSE
S22	2023	Robô articulado	Serial	ANN	Sintético	Simulação	MSE
S23	2023	NAO Robot	Humanoide	Neural Network	Real	Robô real	MAE
S24	2023	Robô de alto DoF	Serial	Deep Learning	Sintético	Simulação	RMSE
S25	2023	Manipulador serial	Serial	ANN + FA-BRIK	Sintético	Simulação	MSE
S26	2023	KUKA industrial	Serial	ANN	Híbrido	Simulação	RMSE
S27	2023	Manipulador redundante	Serial	Deep Learning	Sintético	Simulação	RMSE
S28	2023	Manipuladores robóticos	Revisão	Neural Networks	Revisão	—	—
S29	2023	CNC 5-axis	Serial	Shallow + Deep NN	Sintético	Simulação	MSE
S30	2023	Manipulador 3-DoF	Serial	Deep Learning	Híbrido	Simulação	RMSE
S31	2023	Manipulador industrial	Serial	RNN	Sintético	Simulação	RMSE
S32	2023	Robôs cooperativos 6-DoF	Serial	ANN	Híbrido	Simulação	RMSE
S33	2023	Manipulador 3R	Serial	ANN	Sintético	MATLAB	MSE
S34	2022	Robô 6-DoF	Serial	Neural Network	Sintético	Simulação	MSE
S35	2022	Manipulador robótico	Serial	Deep Learning	Sintético	Simulação	RMSE

Continua na próxima página

ID	Ano	Robô / Sistema	Tipo	Arquitetura de Rede	Dataset	Plataforma	Métrica
S36	2022	Robô 6-axis	Serial	Neural Network	Sintético	Simulação	MSE
S37	2022	Sistema acústico robótico	Especial	Deep Learning	Sintético	Simulação	MSE
S38	2022	Manipulador robótico	Serial	Machine Learning	Sintético	Simulação	RMSE
S39	2022	Manipulador alto DoF	Serial	Deep RL	Sintético	Simulação	Reward
S40	2021	Manipulador robótico	Serial	Transfer Learning NN	Sintético	Simulação	RMSE
S41	2021	Robô 3-DoF	Serial	ANN	Híbrido	Simulação	RMSE
S42	2021	Robô redundante	Serial	Deep Learning	Sintético	Simulação	RMSE
S43	2021	Robô paralelo	Paralelo	Deep Learning	Sintético	Simulação	MSE
S44	2021	Pioneer 2	Serial	Machine Learning	Híbrido	Simulação	RMSE
S45	2021	Manipulador industrial	Serial	MLP	Real	Robô real	MAE
S46	2021	Manipulador 5-DoF	Serial	Neural Network	Sintético	Simulação	MSE
S47	2020	Manipulador robótico	Serial	GAN	Sintético	Simulação	RMSE
S48	2020	Manipulador robótico	Serial	Neural Network	Sintético	Simulação	MSE
S49	2020	Manipulador robótico	Serial	Deep ANN	Sintético	Simulação	MSE
S50	2019	Manipulador móvel redundante	Serial	Machine Learning	Sintético	Simulação	RMSE

Continua na próxima página

ID	Ano	Robô / Sistema	Tipo	Arquitetura de Rede	Dataset	Plataforma	Métrica
S51	2019	Manipulador 5-axis	Serial	Deep ANN	Sintético	Robô real	RMSE
S52	2018	Manipulador robótico	Serial	Extreme Learning Machine	Sintético	Simulação	MSE

Fonte: Elaborado pelo autor

A codificação dos estudos apresentada na 20 permite observar, de forma estruturada, as principais características metodológicas dos trabalhos analisados, incluindo o tipo de robô investigado, as arquiteturas de redes neurais empregadas, os tipos de *datasets* utilizados, as plataformas experimentais adotadas e as métricas de avaliação aplicadas. Essa síntese facilita a identificação de tendências na literatura e reforça as lacunas de pesquisa identificadas, servindo como base para o desenvolvimento da proposta metodológica apresentada nos capítulos seguintes desta tese.

3 FUNDAMENTAÇÃO TEÓRICA

Desde o ano 2000, a tecnologia de braços robóticos com automontagem tem atraído a atenção de muitos fabricantes de robôs. À medida que a indústria robótica avança, ela gradualmente traz grandes benefícios à produção fabril e ao ambiente social em que está inserida (HSIEH; XU; LIN, 2023).

Atualmente, robôs industriais são empregados em uma variedade de aplicações e estão disponíveis em uma ampla gama de configurações, sistemas de acionamento, conexões físicas e cargas úteis (DAHARI; TAN, 2011).

De acordo com Eliseev, Geguev e Ivanov (2021), um manipulador robótico é um dispositivo automático composto por um manipulador e um dispositivo de controle. O manipulador é um mecanismo que serve para controlar a posição espacial de ferramentas, objetos de trabalho, unidades e elementos estruturais. Não obstante, o dispositivo de controle gera ações de comando que determinam os movimentos necessários para as conexões cinemáticas do manipulador.

De acordo com Parikh e Lam (2005), um manipulador robótico é um conjunto de juntas e elos conectados entre si, configurando-se em uma cadeia cinemática que forma um mecanismo capaz de realizar uma tarefa específica.

Robôs manipuladores são utilizados principalmente para mover objetos e realizar diversas operações tecnológicas. Além disso, esses dispositivos podem ser programados para executar determinadas atividades de forma mais eficiente do que os humanos (NURBA et al., 2022).

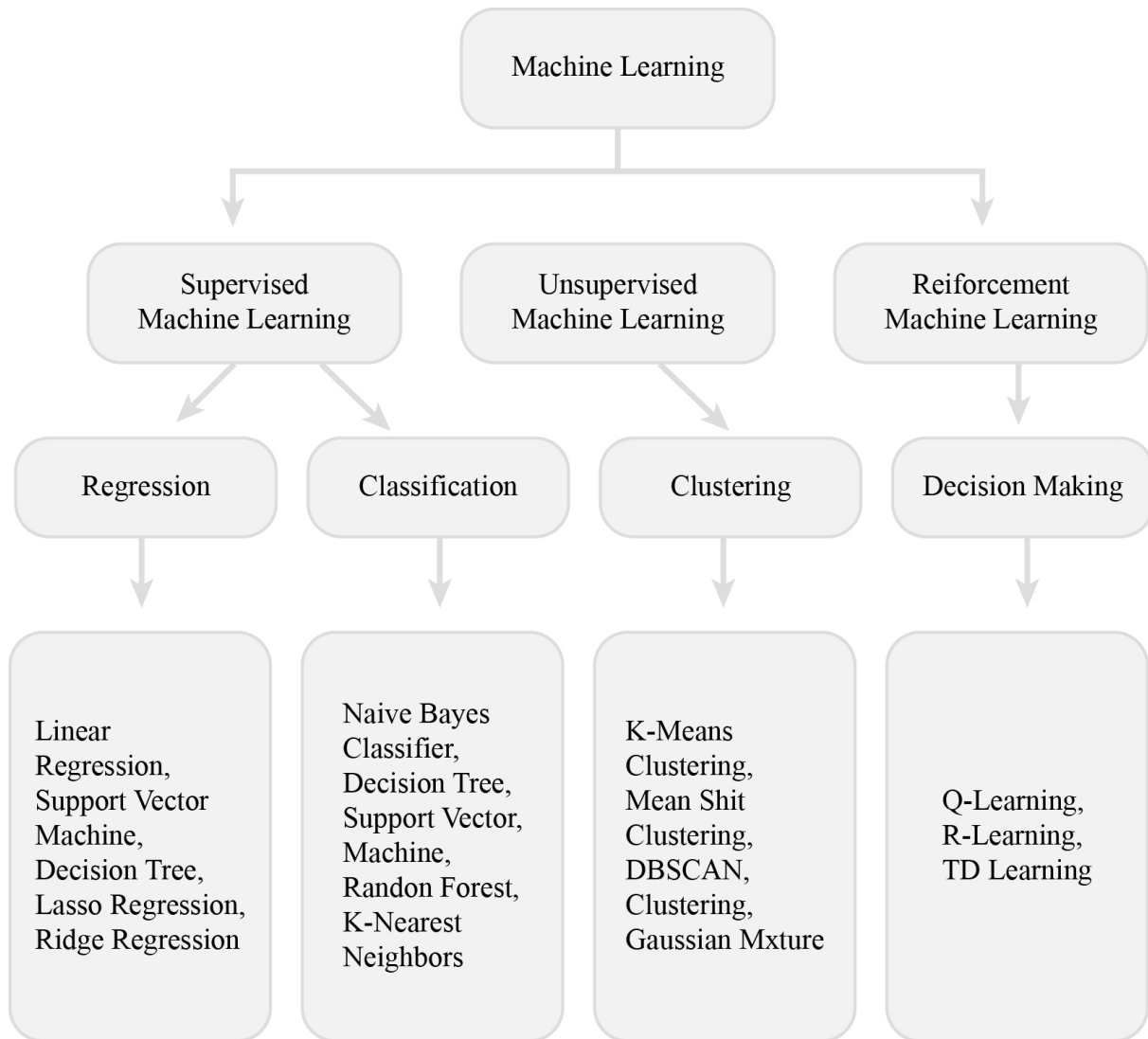
O Aprendizado de Máquina é uma técnica de Inteligência Artificial que permite que os sistemas adquiram conhecimento automaticamente, sem a necessidade de programação explícita. Essa técnica baseia-se principalmente na concepção de que programas de computador podem acessar dados e aprender a partir deles. No entanto, esse processo parte da inspeção dos dados e da busca por padrões, com o objetivo de tomar melhores decisões. O objetivo principal é permitir que os sistemas aprendam automaticamente, sem intervenção humana, não sendo necessário obter conhecimento por consciência, mas sim identificar padrões presentes nos dados. Dessa forma, os algoritmos de *Machine Learning* (ML) eliminam a necessidade de intervenção humana direta no processo de aprendizagem (SARAVANAN; SUJATHA, 2018).

De acordo com Gupta et al. (2022), uma boa maneira de entender o algoritmo de aprendizado de máquina é por meio de um exemplo do processo de aprendizagem, ou seja, a máquina aprende de forma semelhante aos humanos, como se aprendesse com experiências passadas. Pettit et al. (2021) explicam que o Aprendizado Profundo — *Deep Learning* (DL) — é uma subcategoria inserida dentro do paradigma de ML e é definido pelo uso de arquiteturas de redes neurais. A arquitetura de rede

neural mais básica é uma rede totalmente conectada ou densa, chamada *Feedforward*, ou ainda uma rede multicamadas, denominada *perceptron* multicamadas.

O *Machine Learning* (ML) é dividido em três abordagens principais, a saber: Aprendizado de Máquina Supervisionado — *Supervised Machine Learning* (SML), Aprendizado de Máquina Não Supervisionado — *Unsupervised Machine Learning* (UML), e Aprendizado por Reforço — *Reinforcement Learning* (RL), conforme ilustrado na Figura 9.

Figura 9 – Principais paradigmas de aprendizado de máquina



Fonte: Elaborada pelo Autor.

Vercio et al. (2020) conceitua o Aprendizado de Máquina Supervisionado (SML) como a busca por algoritmos que raciocinam a partir de instâncias fornecidas externamente para produzir hipóteses gerais, que então fazem previsões sobre instâncias futuras. Em outras palavras, o objetivo do aprendizado supervisionado é construir um modelo conciso da distribuição dos rótulos de classe com base em

atributos preditivos. O classificador resultante é, então, utilizado para atribuir rótulos de classe a instâncias de teste, nas quais os valores dos atributos preditivos são conhecidos, mas os rótulos de classe são desconhecidos.

O *Unsupervised Machine Learning* (UML) é uma abordagem de aprendizado de máquina voltada à detecção de padrões em conjuntos de dados com pontos não rotulados ou não controlados. Nesse tipo de aprendizado, um sistema de inteligência artificial recebe apenas os dados de entrada, sem os dados de saída correspondentes. Assim, o aprendizado de máquina não supervisionado, diferentemente do aprendizado supervisionado, não requer a presença de uma pessoa para supervisionar o modelo (LIMA et al., 2018). O cientista de dados permite que o sistema aprenda por conta própria, observando os dados e identificando padrões. Em outras palavras, essa subcategoria do aprendizado de máquina possibilita que um sistema opere sobre dados específicos sem a necessidade de instruções externas (NAEEM et al., 2023).

Neste trabalho, adota-se o paradigma de aprendizado supervisionado, uma vez que o objetivo é aprender um mapeamento entre a pose cartesiana do manipulador robótico e os ângulos de suas juntas a partir de um conjunto de dados rotulado.

3.1 Cinemática direta e inversa de robôs manipuladores

A cinemática é dividida em duas grandes áreas elementares de estudo: a cinemática direta e a cinemática inversa. O principal desafio presente nessas áreas consiste em encontrar o equacionamento ideal para que o efetuador do manipulador esteja corretamente posicionado no espaço de trabalho.

Nas palavras de Dikmenli (2022), qualquer elemento desse movimento cinemático é considerado rígido, e as conexões entre esses componentes são chamadas de juntas. De acordo com o autor, existem vários tipos de juntas, porém as que mais se destacam no âmbito industrial são as juntas prismáticas (deslizantes) e rotativas. Cada junta que forma o corpo cinemático do robô define o grau de liberdade desse mecanismo. Em outras palavras, se o robô possui 6 graus de liberdade, isso significa que ele é formado por 6 juntas, que lhe permitem executar movimentos específicos em um espaço de trabalho determinado.

Em concordância com Huang et al. (2012), a cinemática direta consiste em calcular o movimento e a posição dos efetuadores finais a partir dos ângulos conhecidos de cada junta. No entanto, a cinemática inversa se concentra em determinar o ângulo de cada articulação com base na trajetória e na posição final do efetuador.

Contudo, conforme explana Pozna, Horváth e Hollósi (2016), a modelagem da cinemática direta, ao herdar as transformações homogêneas do modelo geométrico, apresenta um problema com solução

algébrica que corresponde à multiplicação de matrizes e é relativamente simples. Em contrapartida, o problema da cinemática inversa depende da estrutura do robô e requer um cálculo representado por uma solução analítica que, na maioria das vezes, não é trivial, e cuja solução pode não ser encontrada.

De acordo com Das, Slotine e Sheridan (1988), robôs redundantes normalmente são constituídos com mais de 6 graus de liberdade. A solução para esse tipo de problema cinemático pode ser infinita, por se tratar de um sistema de equações não lineares.

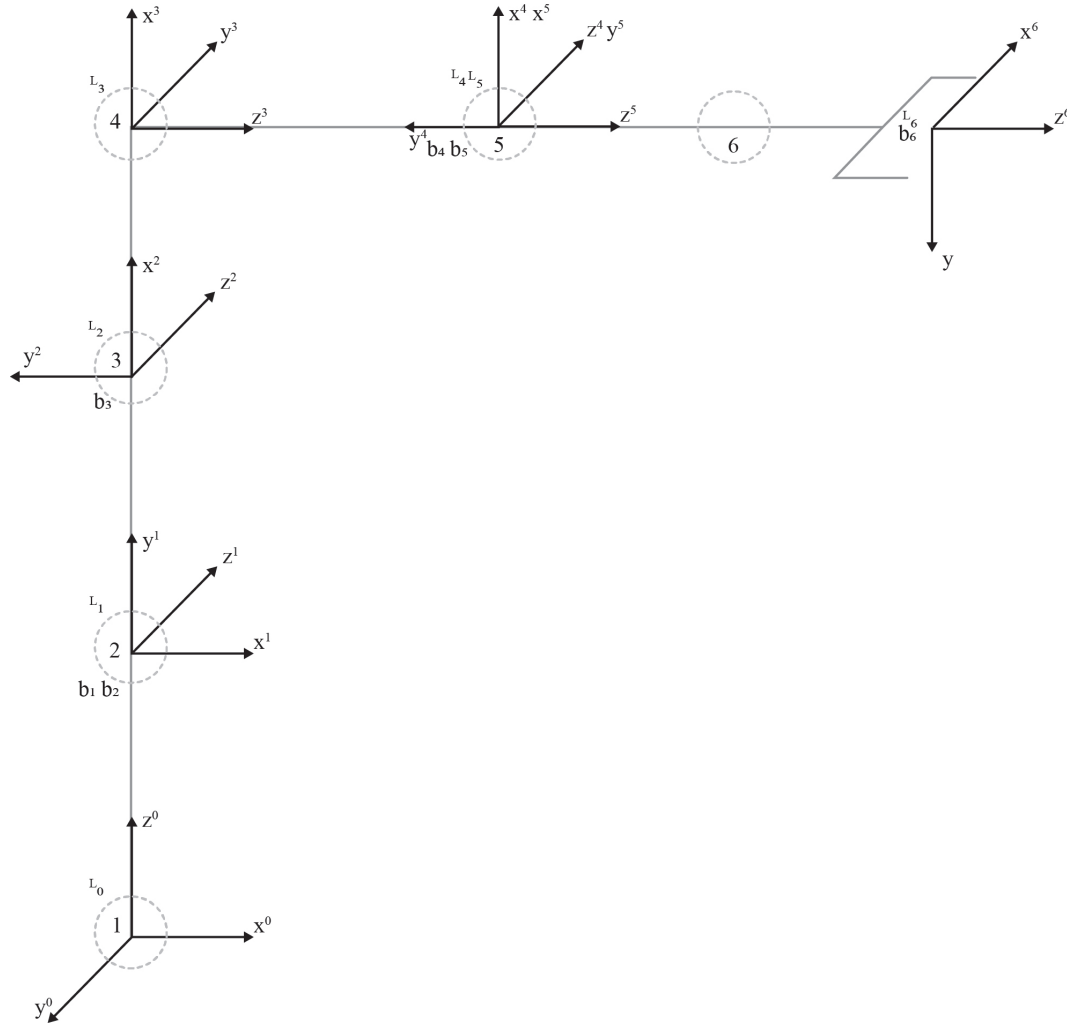
As soluções tradicionais de cinemática inversa analítica incluem métodos baseados em eliminação algébrica e métodos numéricos, como o de Newton–Raphson, que apresenta convergência lenta e é sensível às singularidades do robô. Em contraste, há também abordagens baseadas no método Jacobiano, amplamente utilizadas para o cálculo da cinemática inversa. Além disso, existem métodos baseados em Computação Suave (*Soft Computing*), que podem ser aplicados por meio de sistemas baseados em redes adaptativas, sistemas *neuro-fuzzy* e outras técnicas de inteligência computacional. Esses métodos exploram a relação causal entre as variáveis de entrada e saída do sistema, aprendendo o mapeamento entre a pose do manipulador e suas variáveis articulares a partir de dados (JANG; SUN; MIZUTANI, 1997).

Embora seja considerada uma metodologia relativamente simples, o método de Denavit-Hartenberg é amplamente utilizado no cálculo da cinemática direta e serve para ilustrar a complexidade envolvida no equacionamento necessário para determinar a posição do robô.

3.1.1 Propriedades cinemáticas diretas

Em concordância com Šegota et al. (2021), para a modelagem e execução do método de Denavit-Hartenberg (D-H) no manipulador robótico ABB IRB 120, foi elaborado o esquema apresentado na Figura 10. Na estrutura, cada junta do manipulador robótico é representada por um círculo pontilhado, identificado por L^0 a L^5 , que indicam as origens dos sistemas de coordenadas de cada junta. Os eixos de cada sistema de coordenadas são representados por x^0 até x^5 , y^0 até y^5 e z^0 até z^5 . O sistema de coordenadas do efetuador final é representado por L^6 , sendo x^6 , y^6 e z^6 os vetores normal, deslizante e de aproximação, respectivamente. Os símbolos b_1 a b_6 representam as interseções entre o eixo x do sistema de coordenadas atual e o eixo z do sistema de coordenadas seguinte. Os parâmetros cinemáticos do manipulador robótico em estudo estão apresentados na 21. Com base nesses valores, são construídas as matrizes de transformação homogênea de cada junta, conforme o modelo definido na Equação (3.1).

Figura 10 – Modelagem robótica — Denavit-Hartenberg



Fonte: Elaborada pelo autor.

$$T_k^{k-1} = \begin{bmatrix} \cos \theta_k & -\sin \theta_k \cos \alpha_k & \sin \theta_k \sin \alpha_k & a_k \cos \theta_k \\ \sin \theta_k & \cos \theta_k \cos \alpha_k & -\cos \theta_k \sin \alpha_k & a_k \sin \theta_k \\ 0 & \sin \alpha_k & \cos \alpha_k & d_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

Através do modelo da matriz na Equação (3.1), formam-se as matrizes geradas a partir das Equações (3.2) a (3.7).

$$T_0^1 = \begin{bmatrix} \cos(q_1) & 0 & -\sin(q_1) & 0 \\ \sin(q_1) & 0 & \cos(q_1) & 0 \\ 0 & -1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

Tabela 21 – Parâmetros de Denavit-Hartenberg (D-H) do manipulador ABB IRB 120

k	1	2	3	4	5	6
θ_k	q_1	q_2	q_3	q_4	q_5	q_6
a_k	0	l_2	l_3	0	0	0
d_k	l_1	0	0	0	0	l_6
α_k	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	$-\frac{\pi}{2}$	$\frac{\pi}{2}$	0

Fonte: Elaborada pelo autor.

$$T_1^2 = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & l_2 \cos(q_2) \\ \sin(q_2) & \cos(q_2) & 0 & l_2 \sin(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$$T_2^3 = \begin{bmatrix} \cos(q_3) & 0 & -\sin(q_3) & l_3 \cos(q_3) \\ \sin(q_3) & 0 & \cos(q_3) & l_3 \sin(q_3) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$$T_3^4 = \begin{bmatrix} \cos(q_4) & 0 & -\sin(q_4) & 0 \\ \sin(q_4) & 0 & \cos(q_4) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$$T_4^5 = \begin{bmatrix} \cos(q_5) & 0 & -\sin(q_5) & 0 \\ \sin(q_5) & 0 & \cos(q_5) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

$$T_5^6 = \begin{bmatrix} \cos(q_6) & -\sin(q_6) & 0 & 0 \\ \sin(q_6) & \cos(q_6) & 0 & 0 \\ 0 & 0 & 1 & l_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

Multiplicando as matrizes homogêneas apresentadas nas Equações (3.2) a (3.7), obtém-se a matriz de transformação homogênea final do manipulador, apresentada na Equação (3.9).

$$F_{\text{base}}^t(q) = \begin{bmatrix} R(q) & P(q) \\ V_1^T & \sigma \end{bmatrix} \quad (3.8)$$

$$T_0^6 = \begin{bmatrix} T_{00} & T_{01} & T_{02} & T_{03} \\ T_{10} & T_{11} & T_{12} & T_{13} \\ T_{20} & T_{21} & T_{22} & T_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

Os elementos da matriz de transformação homogênea T_0^6 , definida na Equação (3.9), são dados por:

$$\begin{aligned} T_{00} &= [(S_1 S_4 + C_1 C_4 C_{2,3}) C_5 - S_5 S_{2,3} C_1] C_6 + (S_1 C_4 - S_4 C_1 C_{2,3}) S_6 \\ T_{10} &= [(S_1 C_4 C_{2,3} - S_4 C_1) C_5 - S_1 S_5 S_{2,3}] C_6 - (S_1 S_4 C_{2,3} + C_1 C_4) S_6 \\ T_{20} &= -(S_5 C_{2,3} + S_{2,3} C_4 C_5) C_6 + S_4 S_6 S_{2,3} \\ T_{01} &= -[(S_1 S_4 + C_1 C_4 C_{2,3}) C_5 - S_5 S_{2,3} C_1] S_6 + (S_1 C_4 - S_4 C_1 C_{2,3}) C_6 \\ T_{11} &= [-(S_1 C_4 C_{2,3} + S_4 C_1) C_5 + S_1 S_5 S_{2,3}] S_6 - (S_1 S_4 C_{2,3} + C_1 C_4) C_6 \\ T_{21} &= (S_5 C_{2,3} + S_{2,3} C_4 C_5) S_6 + S_4 S_{2,3} C_6 \\ T_{02} &= -(S_1 S_4 + C_1 C_4 C_{2,3}) S_5 - S_{2,3} C_1 C_5 \\ T_{12} &= (-S_1 C_4 C_{2,3} + S_4 C_1) S_5 - S_1 S_{2,3} C_5 \\ T_{22} &= S_5 S_{2,3} C_4 - C_5 C_{2,3} \\ T_{03} &= l_4 S_1 S_4 S_5 - l_4 S_5 C_1 C_4 C_{2,3} - l_4 S_{2,3} C_1 C_5 + l_1 S_{2,3} C_1 + l_3 C_1 C_2 + l_5 C_1 C_{2,3} \\ T_{13} &= l_4 S_1 S_5 C_4 C_{2,3} - l_4 S_1 S_{2,3} C_5 - l_1 S_1 S_{2,3} + l_3 S_1 C_2 + l_5 S_1 C_{2,3} + l_4 S_4 S_5 C_1 \\ T_{23} &= l_3 S_2 + l_4 S_5 S_{2,3} C_4 - l_5 S_{2,3} - l_4 C_5 C_{2,3} - l_1 C_{2,3} + l_2 \end{aligned}$$

em que l_1 a l_6 representam os comprimentos dos elos do manipulador robótico, medidos a partir da base até o efetuador final. Os símbolos S e C representam, respectivamente, as funções trigonométricas seno e cosseno dos ângulos articulares. Os índices associados indicam o ângulo da junta utilizado. Por exemplo, $\sin(q_1)$ é representado por S_1 , enquanto $\cos(q_2)$ é representado por C_2 . Quando mais de um índice é utilizado, indica-se a soma dos ângulos correspondentes; por exemplo, $\cos(q_2 + q_3)$ é representado por $C_{2,3}$.

3.1.2 Propriedades cinemáticas inversas

A cinemática inversa tem como objetivo determinar os ângulos articulares necessários para que o efetuador final do robô atinja uma posição e orientação desejadas no espaço. Para manipuladores de 6 graus de liberdade com pulso esférico, como o ABB IRB 120, é possível aplicar uma solução analítica dividida em duas etapas: cinemática inversa de posição (para calcular q_1 , q_2 e q_3) e cinemática inversa de orientação (para q_4 , q_5 e q_6).

Cinemática inversa de posição

A posição do efetuador é dada pelo vetor $P = [x \ y \ z]^T$. Considerando que o eixo 4 inicia no centro do pulso, a posição do centro do pulso (*wrist center*) é calculada por:

$$P_{wc} = P - l_6 \cdot R \cdot \hat{z}_6 \quad (3.10)$$

Onde R é a matriz de rotação do efetuador e $\hat{z}_6$ é o eixo z local do efetuador (tipicamente $[0 \ 0 \ 1]^T$ na base do efetuador). O vetor $P_{wc} = [x_{wc}, y_{wc}, z_{wc}]^T$ é usado para calcular os ângulos q_1 , q_2 e q_3 :

$$q_1 = \arctan 2(y_{wc}, x_{wc}) \quad (3.11)$$

Os ângulos q_2 e q_3 podem ser obtidos por relações trigonométricas baseadas na geometria do manipulador:

$$r = \sqrt{x_{wc}^2 + y_{wc}^2} \quad (3.12)$$

$$s = z_{wc} - l_1 \quad (3.13)$$

$$D = \frac{r^2 + s^2 - l_2^2 - l_3^2}{2l_2l_3} \quad (3.14)$$

$$q_3 = \arctan 2(\sqrt{1 - D^2}, D) \quad (3.15)$$

$$q_2 = \arctan 2(s, r) - \arctan 2(l_3 \sin(q_3), l_2 + l_3 \cos(q_3)) \quad (3.16)$$

Cinemática inversa de orientação

Com os valores de q_1 , q_2 e q_3 conhecidos, é possível calcular a matriz de rotação do punho R_0^3 (produto das matrizes de rotação das juntas 1 a 3). A orientação desejada do efetuador é representada pela matriz R_0^6 . A matriz de rotação equivalente ao pulso (juntas 4 a 6) é então dada por:

$$R_3^6 = (R_0^3)^T \cdot R_0^6 \quad (3.17)$$

Supondo que R_3^6 seja dada por:

$$R_3^6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Os ângulos finais q_4 , q_5 e q_6 podem ser extraídos como:

$$q_5 = \arccos(r_{33}) \quad (3.18)$$

$$q_4 = \arctan 2(r_{23}, r_{13}) \quad (3.19)$$

$$q_6 = \arctan 2(r_{32}, -r_{31}) \quad (3.20)$$

Essa abordagem assume que não há singularidade (isto é, $q_5 \neq 0$ ou $q_5 \neq \pi$). Em casos singulares, soluções alternativas devem ser consideradas, como fixar $q_4 = 0$ e resolver apenas q_6 .

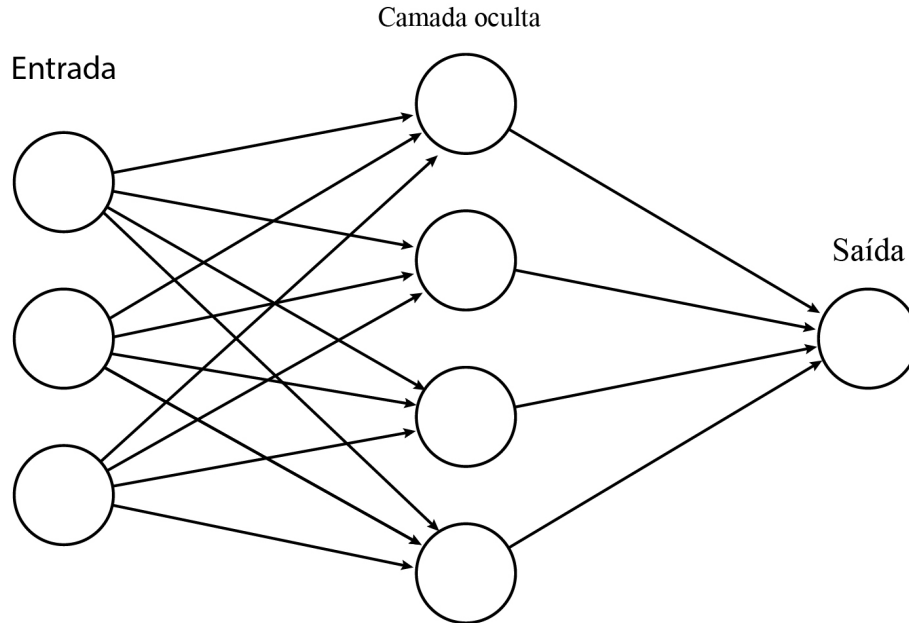
Essa solução analítica, embora exata, é suscetível a erros de modelagem e singularidades. Por isso, métodos alternativos baseados em aprendizado de máquina, como os apresentados neste trabalho, mostram-se promissores, especialmente para aplicações em tempo real ou com robôs redundantes.

3.2 Introdução às redes neurais artificiais

As Redes Neurais Artificiais (RNAs) constituem modelos computacionais desenvolvidos para reproduzir, de forma simplificada, mecanismos de processamento de informação observados no sistema nervoso biológico. O conceito fundamental foi introduzido por McCulloch e Pitts, em 1943, ao demonstrarem que neurônios artificiais conectados poderiam realizar operações lógicas e cálculos complexos (MCCULLOCH; PITTS, 1943). Esse modelo, conhecido como MCP, utilizava combinações lineares de entradas seguidas de uma função de ativação binária, estabelecendo as bases para o desenvolvimento posterior das redes neurais (VOULODIMOS et al., 2018).

Em 1958, Rosenblatt apresentou o *Perceptron*, o primeiro modelo treinável de neurônio artificial (LIU et al., 2023b). A estrutura simplificada de uma rede neural artificial é mostrada na Figura 11.

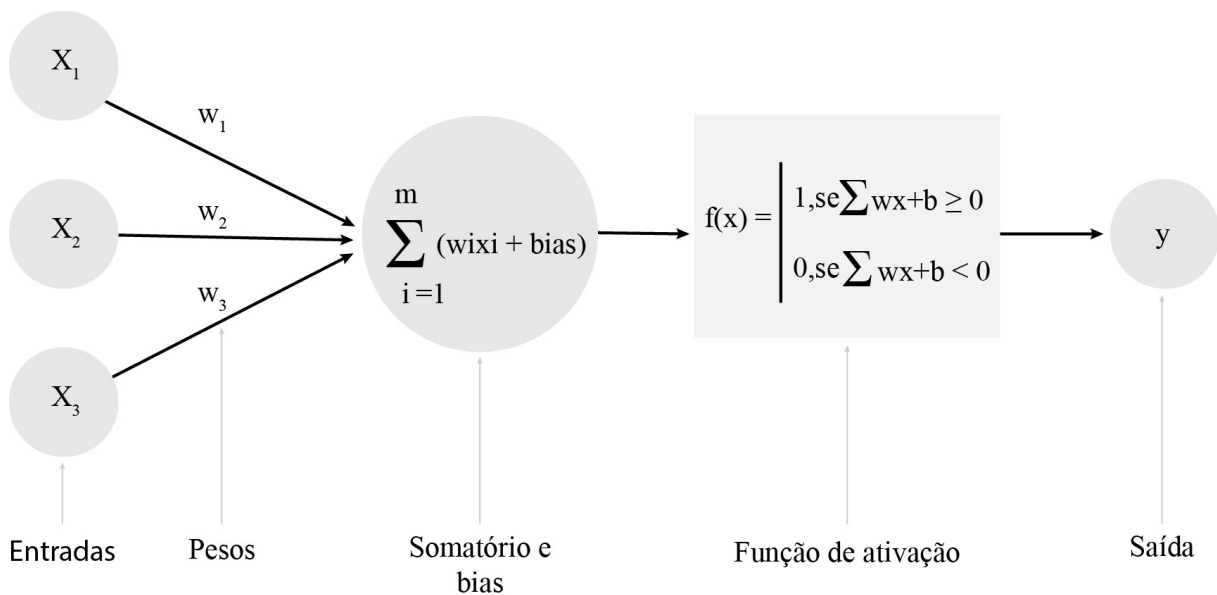
Figura 11 – Estrutura de uma rede neural artificial simples.



Fonte: Elaborada pelo Autor.

Conforme discutido por Kriegeskorte e Golan (2019), redes neurais artificiais aprendem ajustando os pesos sinápticos das conexões internas. Esses pesos determinam o grau de influência de cada entrada no neurônio. A Figura 12 ilustra o funcionamento de um *Perceptron*, destacando o papel dos pesos, do termo de *bias* e da função de ativação (WHITTINGTON; BOGACZ, 2019; GEUBBELMANS et al., 2024).

Figura 12 – Representação de uma conexão sináptica em um *Perceptron*.

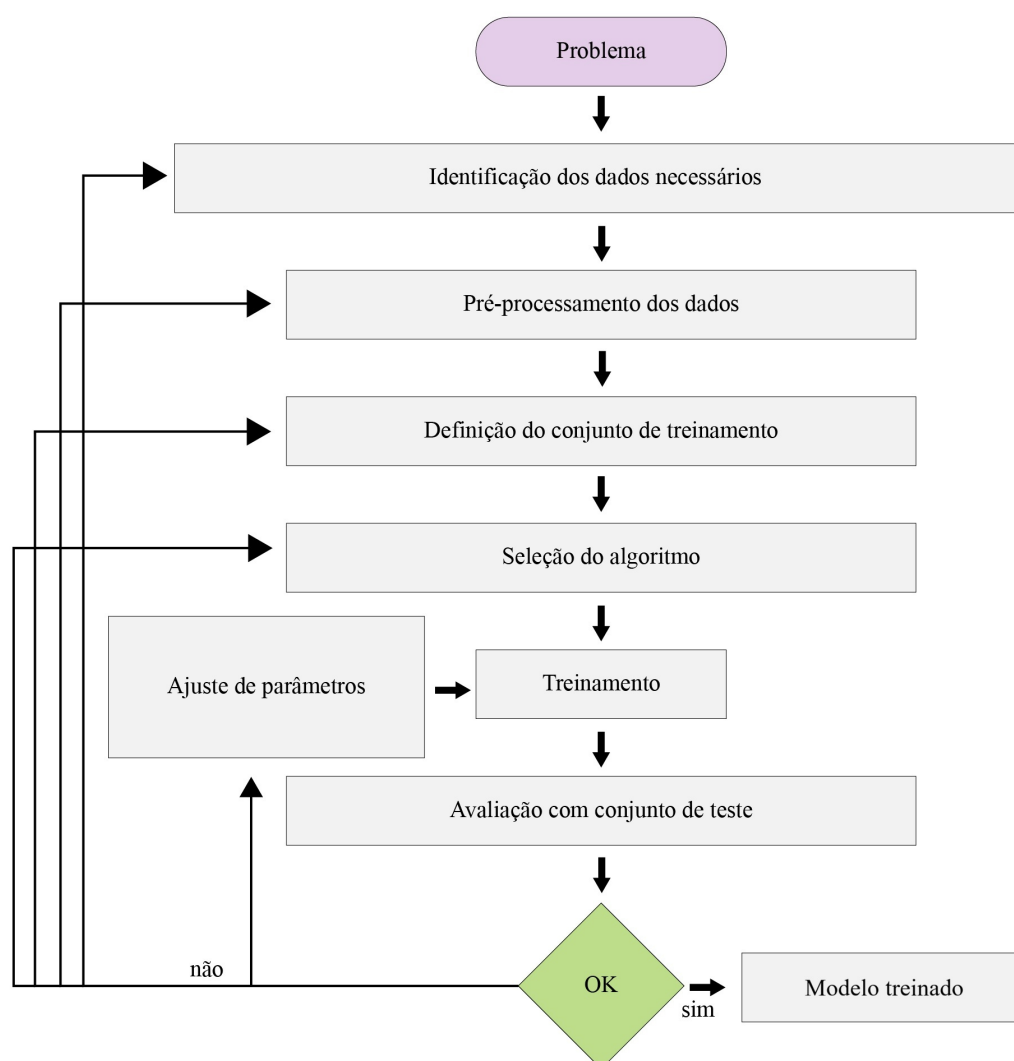


Fonte: Elaborada pelo Autor.

3.2.1 Processo geral de treinamento de redes neurais

O treinamento de uma rede neural envolve um conjunto padronizado de etapas, amplamente aceito na literatura de aprendizado de máquina. Essas etapas incluem a definição do problema, preparação e pré-processamento dos dados, divisão entre conjuntos de treinamento e teste, seleção da arquitetura, ajuste de hiperparâmetros, treinamento supervisionado e avaliação final. A Figura 13 ilustra esse fluxo de trabalho genérico.

Figura 13 – Processo genérico de pré-processamento, treino e validação de redes neurais artificiais.



Fonte: Elaborada pelo Autor.

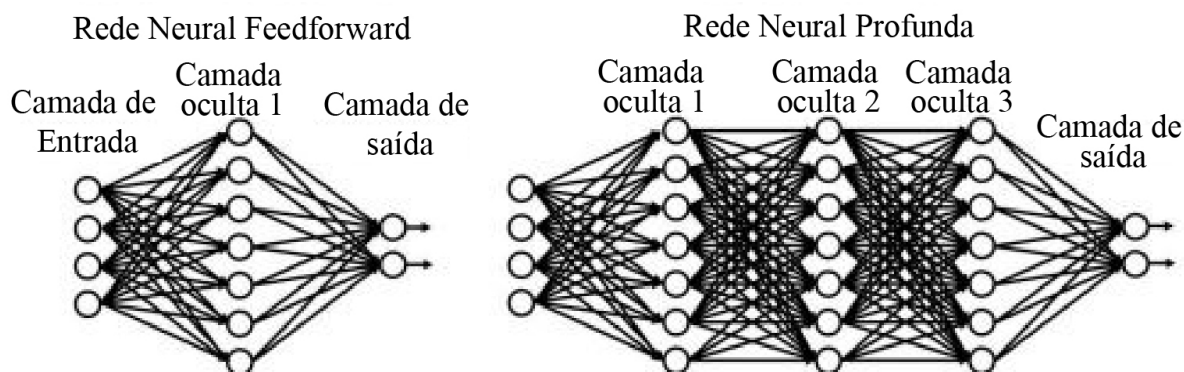
3.3 Conceitos de aprendizado profundo

O Aprendizado Profundo (*Deep Learning* – DL) é uma subárea do aprendizado de máquina que emprega arquiteturas compostas por múltiplas camadas de neurônios artificiais. Redes totalmente conectadas (*feedforward*) tornam-se redes profundas quando possuem duas ou mais camadas ocultas,

aumentando sua capacidade de modelar relações altamente não lineares (HUANG; PANG; ZHANG, 2023).

O avanço no poder computacional e na disponibilidade de grandes volumes de dados tem permitido o uso de arquiteturas cada vez mais profundas, capazes de resolver problemas complexos em diversas áreas (SERRE, 2019). A Figura 14 apresenta uma comparação entre uma rede neural superficial e uma rede profunda.

Figura 14 – Comparação entre rede neural superficial e rede neural profunda.



Fonte: Elaborada pelo Autor.

4 MATERIAIS E MÉTODOS

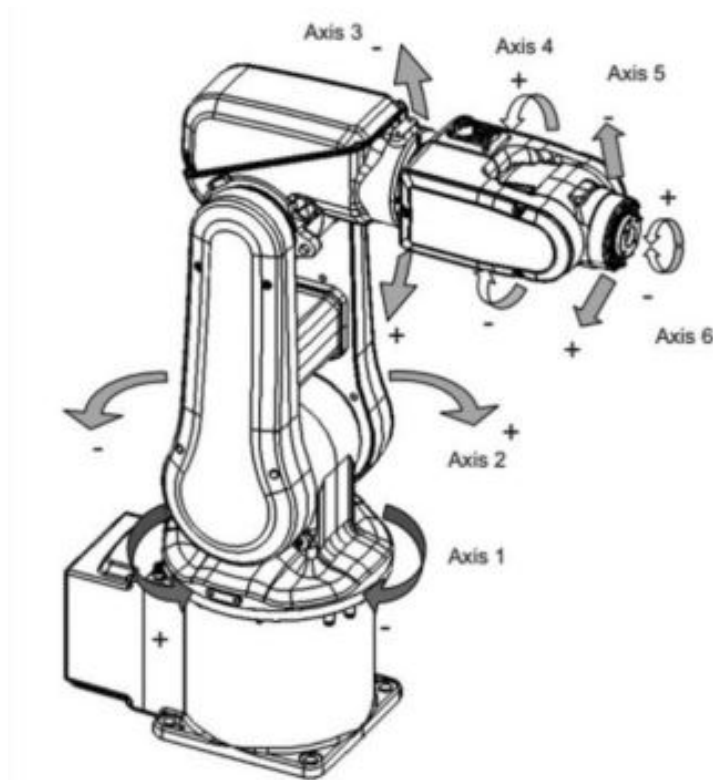
4.1 Materiais

Esta seção apresenta os principais recursos físicos e computacionais utilizados no desenvolvimento deste trabalho, incluindo o manipulador robótico adotado como caso de estudo e as ferramentas de *software* empregadas na geração do *dataset* sintético, no treinamento da rede neural profunda e na execução das simulações.

4.1.1 O robô em estudo: ABB IRB 120

O manipulador industrial ABB IRB 120, ilustrado na Figura 15, foi selecionado como plataforma de referência para este estudo. Esse robô possui seis graus de liberdade, arquitetura do tipo série, e é amplamente empregado em aplicações industriais que exigem precisão, agilidade e baixo consumo de energia.

Figura 15 – Robô ABB IRB 120



Fonte: (ŠEGOTA et al., 2021).

As características mecânicas e os limites articulares de cada junta do IRB120, conforme fornecidos pelo fabricante e resumidos na 22, estabelecem a região admissível de operação no espaço de juntas.

Esses limites foram utilizados neste trabalho como base para a geração sintetizada das configurações articulares empregadas na construção do dataset.

Tabela 22 – Parâmetros cinemáticos — ABB IRB 120

Junta	Limite inferior (°)	Limite superior (°)	Faixa (°)
1	−165	165	330
2	−110	110	220
3	−70	110	180
4	−160	160	320
5	−120	120	240
6	−180	180	360

Fonte: Elaborada pelo Autor.

O IRB 120 possui massa total de 25 kg, carga útil de 3 kg e alcance horizontal de 580 mm, características que o tornam adequado para experimentação e validação de métodos de cinemática, planejamento de movimento e controle. Neste trabalho, o modelo analítico do manipulador foi utilizado exclusivamente para: (i) computar a cinemática direta durante a geração do dataset; e (ii) definir restrições geométricas e articulares empregadas no balanceamento do espaço operacional sintetizado.

4.1.2 Ferramentas computacionais

Para a implementação da metodologia proposta, utilizou-se a linguagem de programação Python (versão 3.x) em conjunto com bibliotecas amplamente difundidas na área de aprendizado de máquina e robótica:

- **NumPy** e **Pandas** — manipulação de vetores, matrizes e estruturas de dados;
- **scikit-learn** — normalização, clusterização (*MiniBatch K-Means*) e métricas de avaliação;
- **TensorFlow / Keras** — implementação, treinamento e validação da rede neural profunda;
- **Matplotlib** — geração de gráficos, curvas de treinamento e dispersões Real vs. Predito.

Todos os experimentos realizados neste trabalho — incluindo a geração do dataset sintético, o treinamento da rede neural profunda e a execução da função de inferência — foram conduzidos em

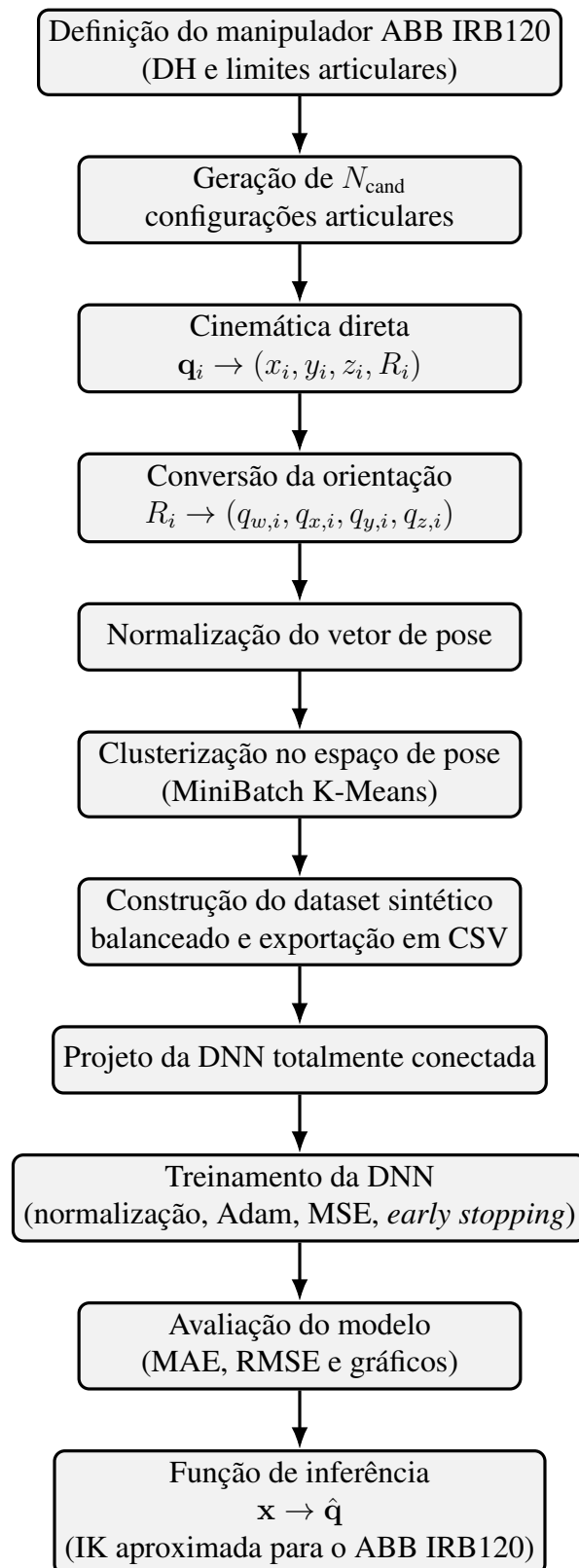
um *laptop* convencional, sem recursos dedicados de aceleração por GPU. O equipamento utilizado possui um processador Intel Core i5-1135G7 (11^a geração, 4 núcleos e 8 *threads*, frequência base de 2.40 GHz), 8 GB de memória RAM e GPU integrada Intel Iris Xe Graphics. Mesmo operando em uma plataforma de hardware simples, o emprego de técnicas de otimização, como inicialização He-normal, regularização L2, *dropout* e *early stopping*, permitiu realizar o treinamento da rede de forma eficiente e com tempos de execução compatíveis com a complexidade do modelo adotado. Esses resultados demonstram que a metodologia proposta pode ser reproduzida em equipamentos acessíveis, sem necessidade de infraestrutura computacional de alto desempenho.

4.2 Métodos

A metodologia adotada neste trabalho foi estruturada de forma a permitir o desenvolvimento sistemático de um modelo de aprendizado profundo capaz de aproximar a solução da cinemática inversa do manipulador ABB IRB120 a partir de sua pose cartesiana. Para isso, dividiu-se o processo em três grandes etapas: (i) geração de um conjunto de dados sintético e balanceado cobrindo o espaço operacional do robô; (ii) projeto, implementação e treinamento de uma Rede Neural Profunda (DNN) para mapear posição e orientação em quaternions para ângulos articulares; e (iii) definição de um algoritmo de inferência que permite estimar as juntas do manipulador de forma contínua e aproximada.

A Figura 16 apresenta o fluxograma geral dessas etapas, mostrando o encadeamento lógico entre a modelagem cinemática, a construção do dataset, o treinamento do modelo e a implementação final da função de cinemática inversa baseada em aprendizado profundo.

Figura 16 – Pipeline do método proposto para geração do dataset, treinamento da DNN e inferência da cinemática inversa aproximada.



Fonte: Elaborada pelo Autor.

4.2.1 Geração do dataset sintético para treinamento da rede neural

Esta subseção descreve o processo adotado para a construção do dataset sintético utilizado no treinamento da rede neural destinada a aproximar a solução da cinemática inversa do manipulador ABB IRB120. Todos os dados foram gerados artificialmente a partir da cinemática direta do robô, permitindo a eliminação de ruídos experimentais e garantindo um conjunto de amostras bem distribuído no espaço operacional. O código completo utilizado para a geração do dataset encontra-se disponível no Apêndice A.

4.2.1.1 Definição dos limites articulares e restrição de ramo

A amostragem inicial foi realizada diretamente no espaço de juntas, respeitando os limites articulares do ABB IRB120. Além desses limites, impôs-se uma restrição adicional na junta do punho, forçando

$$q_5 > 10^\circ.$$

Essa restrição foi necessária para evitar:

- soluções espelhadas do punho (“*wrist flip*”), que surgem quando $q_5 \approx 0^\circ$ e tornam a orientação não unívoca no espaço de juntas;
- singularidades estruturais associadas ao alinhamento dos eixos do punho.

Dessa forma, o dataset resultante pertence a um único ramo cinemático, assegurando que o mapeamento entre pose e juntas seja funcionalmente unívoco, condição essencial para o aprendizado supervisionado.

4.2.1.2 Amostragem massiva no espaço de juntas

Foram geradas inicialmente

$$N_{\text{cand}} = 60\,000$$

configurações articulares aleatórias, uniformemente distribuídas dentro dos limites especificados.

Cada configuração articular é representada por

$$\mathbf{q}_i = [q_{1,i}, q_{2,i}, \dots, q_{6,i}],$$

em que $i = 1, \dots, N_{\text{cand}}$ denota o índice da amostra.

Cada vetor de juntas foi convertido para o espaço operacional por meio da cinemática direta do manipulador, utilizando o modelo Denavit–Hartenberg previamente validado. A posição cartesiana correspondente é dada por

$$\mathbf{p}_i = [x_i, y_i, z_i],$$

enquanto a orientação foi representada por um quaternion unitário

$$\mathbf{q}_i^{(\text{ori})} = [q_{w,i}, q_{x,i}, q_{y,i}, q_{z,i}],$$

evitando singularidades inerentes às representações angulares.

Essa superamostragem é fundamental, pois uma distribuição uniforme no espaço de juntas não resulta, em geral, em uma distribuição uniforme no espaço cartesiano.

4.2.1.3 Normalização do espaço de pose

Como posição e orientação possuem escalas de magnitude distintas, o vetor de pose

$$\mathbf{s}_i = [x_i, y_i, z_i, q_{w,i}, q_{x,i}, q_{y,i}, q_{z,i}]$$

foi normalizado utilizando padronização do tipo *z-score*. Essa transformação impede que as componentes translacionais dominem a métrica de distância durante a etapa de agrupamento.

4.2.1.4 Balanceamento geométrico do espaço operacional

Apesar da amostragem uniforme no espaço articular, o espaço operacional do robô apresenta regiões de densidade muito distintas. Para garantir uma distribuição homogênea de poses, aplicou-se um processo de clusterização utilizando o algoritmo *MiniBatch K-Means*.

Definiu-se o número final desejado de amostras como:

$$N_{\text{final}} = 10\,000.$$

A clusterização foi realizada no espaço de pose normalizado. Para cada centroide obtido, selecionou-se o candidato mais próximo, garantindo assim que o conjunto final represente uniformemente o envelope operacional do manipulador.

Esse procedimento é superior à simples amostragem aleatória, pois:

- reduz redundâncias no conjunto de dados;

- aumenta a diversidade geométrica das amostras;
- melhora generalização do modelo de aprendizado.

4.2.1.5 Construção final do dataset

O dataset final contém, para cada amostra:

- juntas reais: $[q_1, q_2, q_3, q_4, q_5, q_6]$;
- posição cartesiana: $[x, y, z]$;
- quaternion de orientação: $[q_w, q_x, q_y, q_z]$.

O conjunto final, após balanceamento e eliminação de redundâncias, possui dimensão aproximada de:

$$N \approx 10\,000 \text{ amostras únicas.}$$

Esse dataset foi então exportado em formato CSV para uso direto no treinamento da rede neural de mapeamento pose–juntas.

Com o objetivo de sistematizar as etapas descritas anteriormente, o Algoritmo 1 apresenta o procedimento adotado para a geração e balanceamento do dataset sintético utilizado neste estudo.

Algorithm 1 Geração do Dataset Sintético do IRB120

Input: Limites articulares $\mathbf{q}_{\min}, \mathbf{q}_{\max}$ e modelo de cinemática direta

Output: Dataset balanceado contendo $(\mathbf{q}, \mathbf{p}, \mathbf{q}^{(ori)})$

1. Amostragem no espaço de juntas:

Gerar N_{cand} vetores articulares $\mathbf{q}_i \sim U(\mathbf{q}_{\min}, \mathbf{q}_{\max})$ respeitando $q_5 > 10^\circ$.

2. Cinemática direta:

Para cada $\mathbf{q}_i$, computar posição e orientação:

$$\mathbf{p}_i = FK_p(\mathbf{q}_i), \quad \mathbf{R}_i = FK_R(\mathbf{q}_i), \quad \mathbf{q}_i^{(ori)} = \text{quat}(\mathbf{R}_i).$$

3. Normalização do vetor de pose:

Aplicar padronização:

$$\mathbf{s}_i = \text{zscore}([\mathbf{p}_i, \mathbf{q}_i^{(ori)}]).$$

4. Balanceamento via clusterização:

Executar *MiniBatch K-Means* com N_{final} clusters sobre $\mathbf{s}_i$;

Selecionar o candidato mais próximo de cada centroide.

5. Construção do dataset:

Montar linhas contendo

$$(\mathbf{q}_i, \mathbf{p}_i, \mathbf{q}_i^{(ori)})$$

e salvar em arquivo CSV.

4.2.2 Desenvolvimento do modelo de rede neural

Após a geração e filtragem do conjunto de dados descrita na Seção 4.2.1, foi desenvolvido um modelo de *Deep Neural Network* (DNN) capaz de estimar simultaneamente os seis ângulos articulares do manipulador ABB IRB120 a partir de sua pose cartesiana completa. No presente trabalho, adota-se um modelo unificado e de maior profundidade, com um conjunto de entrada ampliado e uma arquitetura projetada especificamente para capturar dependências não lineares entre posição, orientação e estados articulares. O código completo utilizado para a definição, treinamento e validação da DNN encontra-se apresentado no Apêndice B.

4.2.2.1 Entradas, saídas e motivação da arquitetura

O vetor de entrada do modelo é composto por sete variáveis:

$$\mathbf{x} = [x, y, z, q_w, q_x, q_y, q_z],$$

onde (x, y, z) representam a posição do efetuador e (q_w, q_x, q_y, q_z) constituem o quaternion que descreve sua orientação. A adoção de quaternions evita ambiguidades e singularidades presentes em ângulos de Euler, permitindo que a rede aprenda de forma estável o mapeamento pose-juntas.

A saída da rede é o vetor articular estimado:

$$\hat{\mathbf{q}} = [\hat{q}_1, \hat{q}_2, \hat{q}_3, \hat{q}_4, \hat{q}_5, \hat{q}_6],$$

correspondente ao conjunto completo de juntas do IRB120.

Essa escolha possibilita o aprendizado conjunto das correlações entre os elos do manipulador, evitando inconsistências que surgem quando cada junta é tratada como um problema independente.

4.2.2.2 Arquitetura da rede neural profunda

A rede neural empregada neste trabalho é do tipo totalmente conectada (*feedforward*) e foi projetada para aproximar o mapeamento não linear entre a pose cartesiana do efetuador final e as variáveis articulares do manipulador ABB IRB120. Como entrada, a rede recebe um vetor com sete variáveis, composto pela posição cartesiana (x, y, z) e pela orientação representada por quatérnions (q_w, q_x, q_y, q_z) . Como saída, o modelo produz as seis variáveis articulares estimadas $(q_1, q_2, q_3, q_4, q_5, q_6)$.

A arquitetura adotada possui três camadas ocultas densas, com 256, 512 e 256 neurônios, respectivamente. Em todas as camadas ocultas foi empregada a função de ativação ReLU (*Rectified Linear Unit*), por sua capacidade de representar relações não lineares complexas e por favorecer a

estabilidade do processo de treinamento em redes profundas. Além disso, foi utilizada a inicialização He-normal, adequada para redes com ativação ReLU, contribuindo para uma propagação mais estável dos gradientes ao longo das camadas.

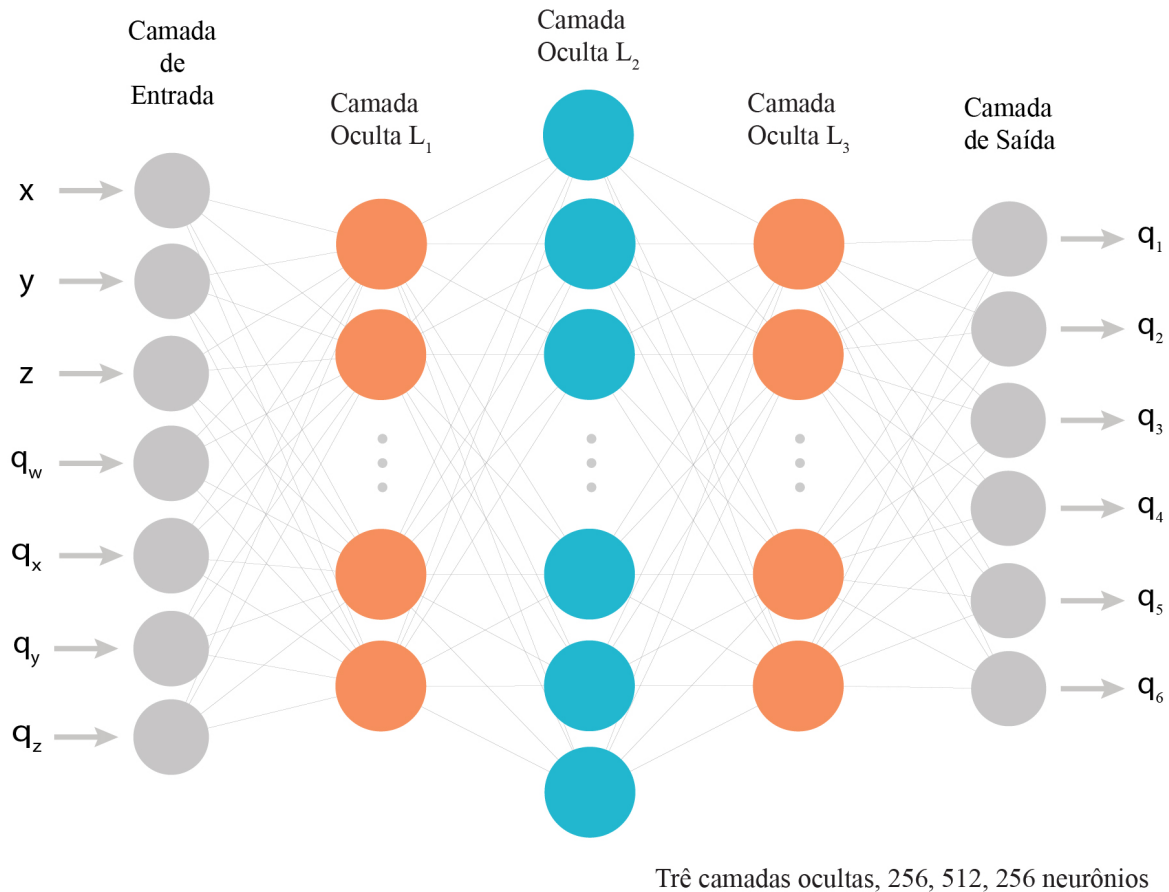
Com o objetivo de mitigar problemas de sobreajuste e melhorar a capacidade de generalização do modelo, foram incorporados mecanismos de regularização. Especificamente, adotou-se regularização L2 nos pesos das camadas densas e camadas de *dropout* entre as camadas ocultas. As taxas de *dropout* utilizadas foram de 10% após a primeira camada oculta, 20% após a segunda e 10% após a terceira. A camada de saída é composta por seis neurônios com ativação linear, uma vez que o problema tratado é de regressão multivariada contínua.

A definição dessa configuração arquitetural foi realizada empiricamente a partir de experimentos preliminares, buscando um equilíbrio entre capacidade de modelagem, estabilidade do treinamento e desempenho de generalização no conjunto de validação. De forma resumida, a arquitetura final é composta por:

- camada de entrada com 7 variáveis;
- primeira camada oculta com 256 neurônios, ativação ReLU e regularização L2;
- camada de *dropout* com taxa de 10%;
- segunda camada oculta com 512 neurônios, ativação ReLU e regularização L2;
- camada de *dropout* com taxa de 20%;
- terceira camada oculta com 256 neurônios, ativação ReLU e regularização L2;
- camada de *dropout* com taxa de 10%;
- camada de saída com 6 neurônios e ativação linear.

A estrutura da rede neural proposta é ilustrada na Figura 17, na qual se observa o fluxo de informações desde a camada de entrada, formada pelas variáveis de pose do efetuador, até a camada de saída, correspondente às juntas estimadas do manipulador.

Figura 17 – Arquitetura da rede neural profunda proposta.



Fonte: Elaborada pelo Autor.

Esse arranjo fornece à rede capacidade expressiva suficiente para modelar as relações altamente não lineares associadas ao problema de cinemática inversa do manipulador, sem comprometer a estabilidade do treinamento. A combinação entre profundidade moderada, regularização L2 e *dropout* permite obter um modelo robusto, capaz de aprender padrões relevantes do conjunto de dados sintético e produzir estimativas consistentes para as variáveis articulares correspondentes às poses desejadas.

4.2.3 Treinamento, validação e métricas de desempenho

O treinamento da rede neural profunda proposta seguiu um conjunto de etapas cuidadosamente estruturadas para garantir estabilidade, boa capacidade de generalização e reprodutibilidade dos resultados. Inicialmente, todas as variáveis de entrada e saída foram normalizadas para o intervalo $[-1, 1]$ utilizando o escalonamento MinMax, procedimento que reduz discrepâncias de magnitude entre posição, orientação e ângulos articulares, acelerando a convergência do modelo. Em seguida, o conjunto de dados foi dividido aleatoriamente em 80% para treinamento e 20% para validação, utilizando semente fixa para garantir consistência entre diferentes execuções.

O treinamento utilizou o otimizador Adam com taxa de aprendizado $\eta = 10^{-3}$ e função de custo baseada no Erro Quadrático Médio (MSE). Para evitar sobreajuste, foi aplicada a técnica de *early stopping*, monitorando a perda de validação e interrompendo o processo após 25 épocas consecutivas sem melhoria significativa, restaurando automaticamente os melhores pesos encontrados durante o processo. O modelo foi treinado por até 400 épocas, utilizando tamanho de lote de 64 amostras.

A avaliação do desempenho foi realizada utilizando duas métricas amplamente empregadas em regressão: o Erro Quadrático Médio (MSE) e o Erro Absoluto Médio (MAE). O MSE, definido pela Equação 4.1, quantifica a diferença quadrática média entre os valores previstos e os valores reais:

$$MSE = \frac{1}{N} \sum_{n=1}^N (P_n - \hat{P}_n)^2, \quad (4.1)$$

em que N é o número de amostras, P_n representa o valor real e $\hat{P}_n$ o valor estimado pelo modelo. Por elevar a diferença ao quadrado, o MSE penaliza erros maiores de forma mais severa.

Já o Erro Absoluto Médio (MAE), definido pela Equação 4.2, calcula a média das diferenças absolutas:

$$MAE = \frac{1}{N} \sum_{n=1}^N |P_n - \hat{P}_n|, \quad (4.2)$$

proporcionando interpretação direta do erro médio em unidades físicas (radianos ou graus).

Ambas as métricas foram utilizadas tanto durante o processo de validação quanto na análise final apresentada no Capítulo 5. Por fim, após o treinamento, foi implementada uma função de inferência capaz de estimar diretamente os seis ângulos articulares a partir de uma pose arbitrária especificada em termos de posição e orientação em quaternion, constituindo uma solução contínua e aproximada de cinemática inversa para o manipulador ABB IRB120. A função de inferência obtida a partir do modelo treinado está descrita no Algoritmo 2, e o código correspondente encontra-se disponível no Apêndice C.

Algorithm 2 Inferência de Cinemática Inversa Aproximada utilizando a DNN treinada

Input: Posição cartesiana (x, y, z) e orientação em quaternion unitário (q_w, q_x, q_y, q_z)

Output: Estimativa das juntas articuladas $\hat{\mathbf{q}} = [\hat{q}_1, \dots, \hat{q}_6]$

Entrada: Construir o vetor de pose

$$\mathbf{x} = [x, y, z, q_w, q_x, q_y, q_z].$$

Etapa 1 — Normalização:

Aplicar o escalonador de entrada treinado:

$$\mathbf{x}_s = S_{\text{in}}(\mathbf{x}).$$

Etapa 2 — Inferência por Rede Neural:

Propagar $\mathbf{x}_s$ pela DNN treinada:

$$\mathbf{q}_s = f_{\theta}(\mathbf{x}_s),$$

onde f_{θ} representa o modelo aprendido.

Etapa 3 — Desscala Final:

Aplicar transformação inversa para retornar ao espaço articular real:

$$\hat{\mathbf{q}} = S_{\text{out}}^{-1}(\mathbf{q}_s).$$

Retornar: $\hat{\mathbf{q}}$

5 RESULTADOS E DISCUSSÃO

5.1 Resultados com o espaço articular completo

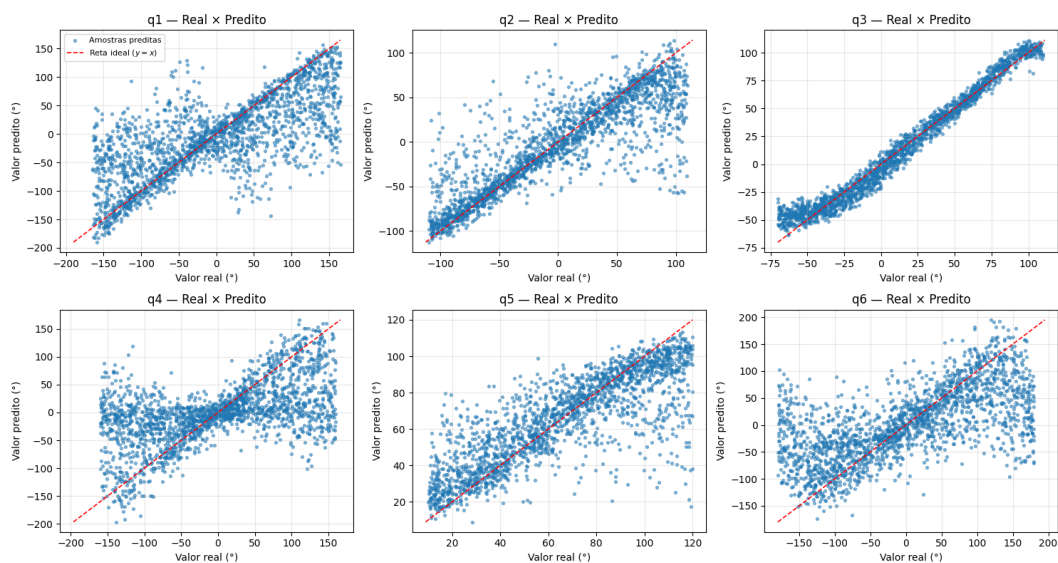
Nesta etapa inicial, avaliou-se o desempenho da rede neural utilizando um conjunto de dados no qual cinco das seis juntas do manipulador ABB IRB120 variaram ao longo de toda a amplitude especificada pelo fabricante, conforme os limites apresentados na 22. A única exceção foi a junta q_5 , cuja variação foi limitada ao intervalo de $[10^\circ, 120^\circ]$. Essa restrição foi necessária para evitar a região próxima de $q_5 = 0^\circ$ e $q_5 = 180^\circ$, onde ocorre a conhecida singularidade do punho esférico: nessa condição, os eixos das juntas q_4 e q_6 tornam-se colineares, resultando na perda de um grau de liberdade efetivo em orientação e na consequente multiplicidade de soluções para (q_4, q_6) .

Mesmo com essa limitação em q_5 , o restante do espaço articular foi explorado sem restrições adicionais de ramo ou medidas para mitigar a redundância intrínseca do punho esférico. Dessa forma, o experimento representa o comportamento da rede neural quando confrontada com a forma mais geral e completa do problema de cinemática inversa do IRB120.

5.1.1 Predição no conjunto de teste

A Figura 18 apresenta os gráficos *Real × Predito* para todas as juntas no conjunto de teste. Observa-se elevada dispersão nos dados, com considerável afastamento da reta ideal $y = x$, especialmente nas juntas q_1 , q_4 e q_6 .

Figura 18 – Predição no conjunto de teste com todas as juntas percorrendo seu intervalo completo de operação.

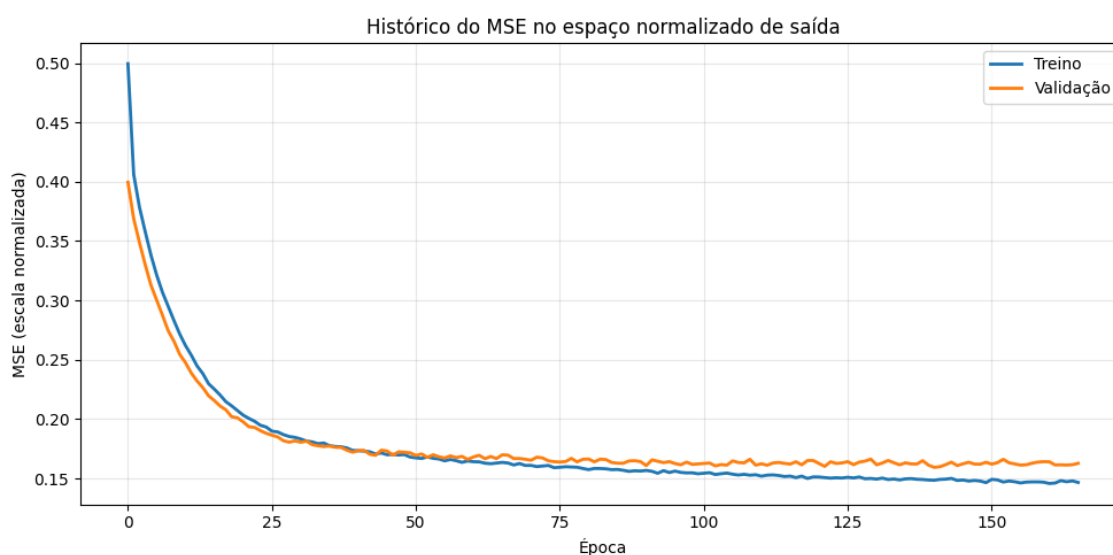


Fonte: Elaborada pelo Autor.

5.1.2 Evolução do treinamento

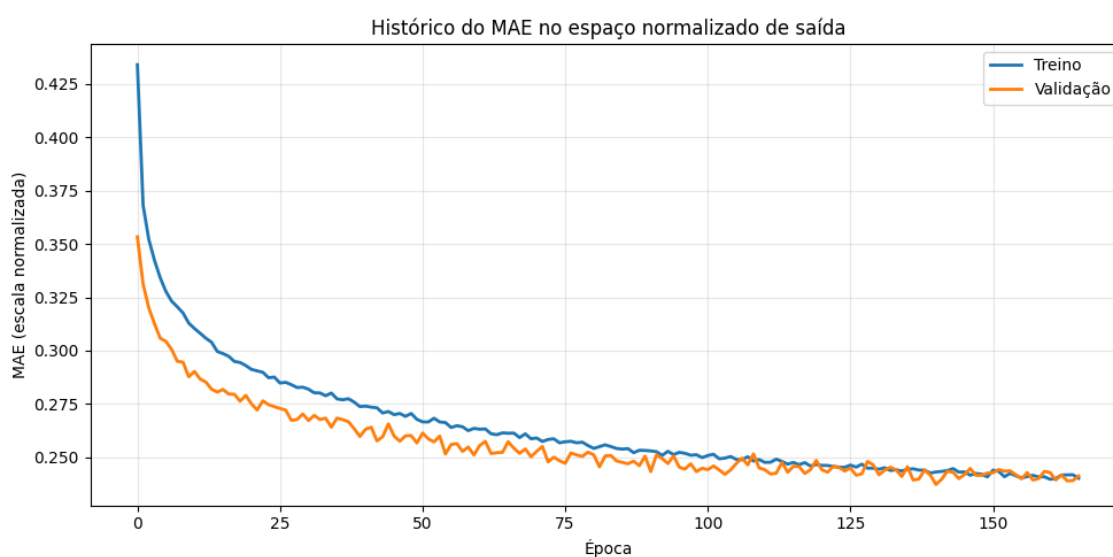
A Figura 19 e a Figura 20 mostram o histórico da função de custo (MSE) e do erro absoluto médio (MAE). A convergência é lenta, e os valores permanecem elevados mesmo após muitas épocas, evidenciando dificuldade do modelo em aprender um mapeamento consistente no espaço completo de 6 DoF.

Figura 19 – Histórico do MSE ao longo do treinamento.



Fonte: Elaborada pelo Autor.

Figura 20 – Histórico do MAE ao longo do treinamento.



Fonte: Elaborada pelo Autor.

5.1.3 Métricas quantitativas

A 23 apresenta os valores de RMSE e MAE após a desnormalização, representando o erro físico no espaço articular do manipulador.

Tabela 23 – Métricas de erro angular por junta no espaço articular completo (graus).

Junta	RMSE (°)	MAE (°)
q_1	63.913	46.344
q_2	31.973	21.318
q_3	8.111	5.961
q_4	73.977	55.402
q_5	19.152	14.049
q_6	76.613	57.957
Média	45.623	33.505

Fonte: Elaborado pelo autor.

5.1.4 Análise dos resultados

Os gráficos da Figura 18 revelam grande dispersão e baixa correlação entre valores reais e preditos. As juntas q_1 , q_4 e q_6 exibem comportamento particularmente crítico, com erros elevados e ausência de estrutura linear no espaço de predição.

Esse comportamento é coerente com os valores da 23, nos quais o erro médio excede 45° (RMSE) e 33° (MAE), tornando o modelo inadequado para aplicações de cinemática inversa.

O padrão observado é típico de problemas em que a solução da cinemática inversa não é unívoca. No caso de manipuladores com punho esférico, diferentes combinações de (q_4, q_5, q_6) podem produzir a mesma orientação cartesiana, resultando em múltiplas soluções articulares para uma mesma pose. Como consequência, exemplos quase idênticos na entrada do modelo podem corresponder a saídas incompatíveis, o que torna o aprendizado supervisionado inviável.

Essa constatação motivou a análise aprofundada da redundância do punho esférico, apresentada na Seção 5.2.

5.2 Considerações sobre a junta q_6 e a redundância do punho esférico

Durante a investigação dos resultados iniciais, verificou-se que a junta q_6 exerce papel central na inconsistência observada no mapeamento entre pose cartesiana e variáveis articulares. Essa junta

corresponde à rotação em torno do eixo z do *flange* do manipulador e integra o conjunto de três eixos concorrentes que caracterizam o punho esférico do ABB IRB120.

Quando o efetuador final apresenta simetria axial — como considerado no conjunto de dados utilizado neste trabalho — a rotação em torno desse eixo não altera a orientação efetiva da ferramenta. Consequentemente, diferentes valores de q_6 podem produzir a mesma pose cartesiana.

Assim, o mapeamento entre pose cartesiana e variáveis articulares

$$(x, y, z, q_w, q_x, q_y, q_z) \rightarrow (q_1, \dots, q_6)$$

não é estritamente unívoco, pois múltiplas configurações articulares podem corresponder à mesma pose do efetuador. Essa característica introduz uma redundância associada à rotação da junta q_6 , o que pode dificultar o aprendizado direto da cinemática inversa por modelos de regressão.

Para reduzir essa não unicidade, adotou-se a fixação da junta q_6 em valor constante durante a geração de um novo *dataset*. Além disso, manteve-se a restrição de ramo previamente estabelecida para a junta q_5 , descrita na Seção 4.2.1.1, na qual se impõe

$$q_5 > 10^\circ.$$

Essa restrição evita configurações próximas à singularidade do punho esférico e impede a ocorrência de soluções espelhadas (*wrist flip*), garantindo que o conjunto de dados pertença a um único ramo da cinemática inversa.

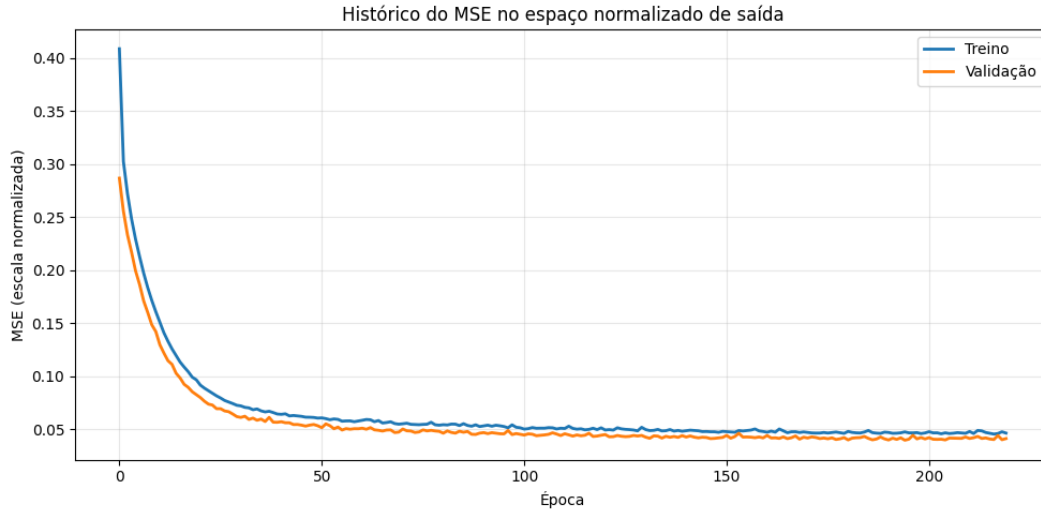
Com a fixação de q_6 e a manutenção dessa restrição em q_5 , o espaço de soluções articulares foi parcialmente regularizado, reduzindo a ambiguidade estrutural presente no conjunto de dados original.

Entretanto, mesmo após essas modificações, o problema permanece intrinsecamente multivalorado, pois diferentes configurações articulares ainda podem produzir a mesma pose cartesiana. Consequentemente, o mapeamento cinemático inverso não é estritamente unívoco, o que impõe uma limitação fundamental à precisão que pode ser alcançada por um modelo de regressão determinístico.

5.2.1 Evolução do treinamento com restrição de ramo

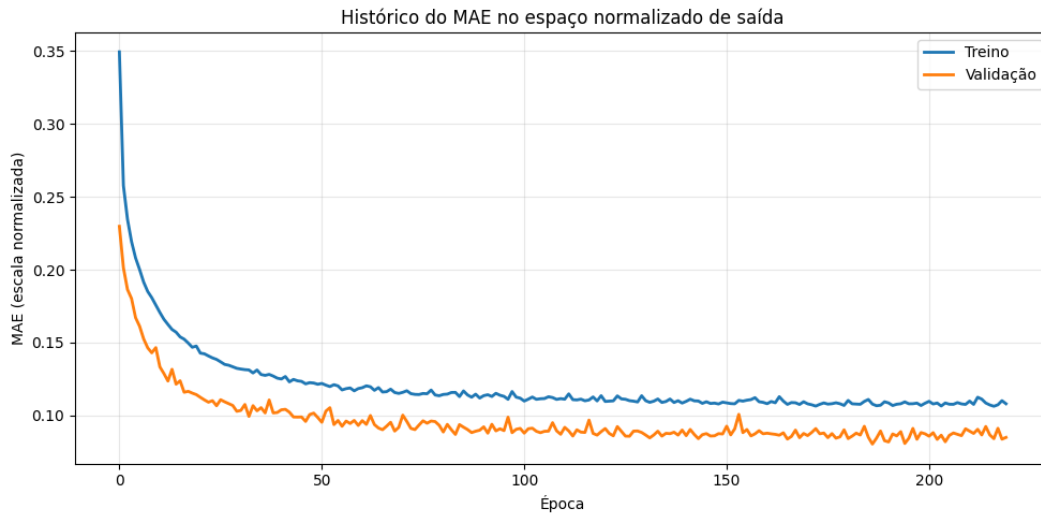
A Figura 21 e a Figura 22 mostram, respectivamente, a evolução do MSE e do MAE após a fixação de q_6 . Observa-se convergência mais rápida, valores substancialmente menores e comportamento estável entre treino e validação, sem indícios de sobreajuste.

Figura 21 – Evolução do MSE durante o treinamento.



Fonte: Elaborada pelo Autor.

Figura 22 – Evolução do MAE durante o treinamento.

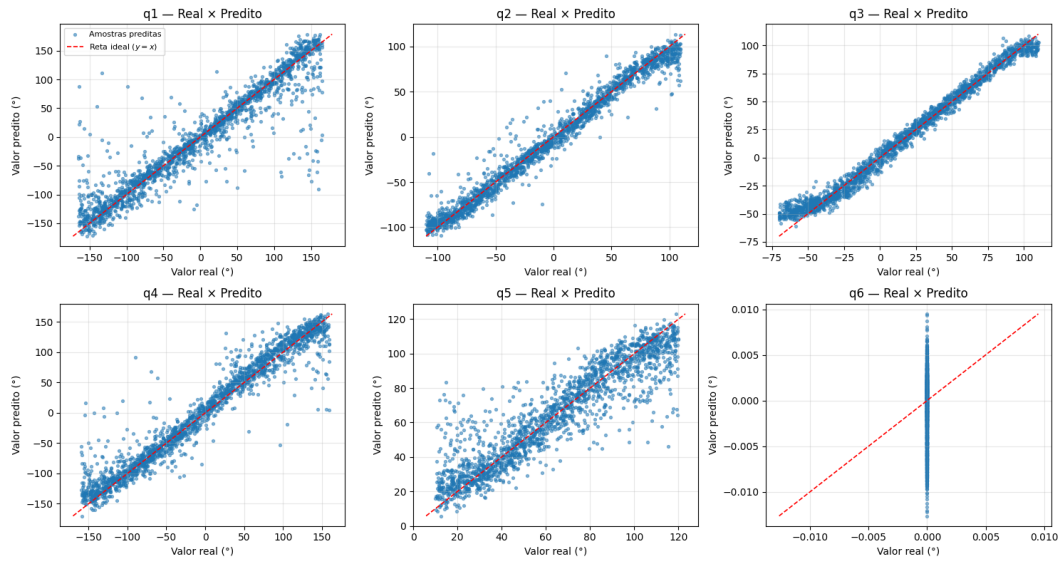


Fonte: Elaborada pelo Autor.

5.2.2 Desempenho no espaço angular

A Figura 23 apresenta os gráficos *Real* $\times$ *Predito* em graus para o conjunto de teste. Observa-se boa aproximação da reta $y = x$, particularmente nas juntas q_2 , q_3 e q_5 , indicando concordância entre os valores reais e estimados pela rede. As juntas q_1 e q_4 apresentam maior dispersão nas regiões extremas do intervalo articular, embora ainda com desempenho substancialmente superior ao observado no cenário sem restrições. A junta q_6 , mantida constante durante a geração do *dataset*, apresenta erro praticamente nulo, conforme esperado.

Figura 23 – Valores reais versus preditos das juntas no conjunto de teste.



Fonte: Elaborada pelo Autor.

5.2.3 Métricas de erro angular

A 24 apresenta os valores de RMSE e MAE após a desnormalização.

Tabela 24 – Métricas de erro angular por junta com restrição de ramo (graus).

Junta	RMSE (°)	MAE (°)
q_1	29.163	15.916
q_2	10.780	7.429
q_3	6.886	5.161
q_4	25.134	14.647
q_5	12.089	8.851
q_6	0.001	0.001
Média	14.009	8.667

Fonte: Elaborado pelo autor.

Os resultados mostram redução significativa dos erros quando comparados ao caso com o espaço articular completo, confirmando que a eliminação da redundância do punho esférico é essencial para que a rede aprenda um mapeamento consistente.

5.3 Correlações entre achados da literatura e os experimentos realizados

Com base nos achados da Revisão Sistemática da Literatura apresentada no Capítulo 2, diversos trabalhos investigam o uso de técnicas de aprendizado de máquina para aproximar a cinemática

inversa de manipuladores robóticos. No estudo conduzido por Šegota et al. (2021), foi utilizado o manipulador ABB IRB120 com um conjunto de dados sintético gerado por meio da cinemática direta e uma arquitetura do tipo *Multilayer Perceptron*. Nesse caso, cada pose cartesiana do conjunto de dados está associada a uma única configuração articular, o que estabelece implicitamente um mapeamento funcional bem definido entre entradas e saídas. Nessas condições, os autores reportaram boa precisão na regressão das variáveis articulares.

Entretanto, manipuladores com punho esférico podem apresentar múltiplas configurações articulares válidas para uma mesma pose do efetuador final. Em particular, a rotação associada à junta q_6 introduz uma redundância de orientação (*wrist twist*), tornando o mapeamento entre pose cartesiana e variáveis articulares intrinsecamente multivalorado. Essa característica dificulta o aprendizado direto da cinemática inversa por modelos de regressão determinísticos.

Complementarmente, Car et al. (2021) empregaram *genetic programming* para gerar expressões simbólicas capazes de aproximar a solução da cinemática inversa do ABB IRB120, reportando erros médios inferiores a $0,5^\circ$ em todas as juntas. No presente trabalho, os erros observados foram superiores aos relatados nesse estudo, o que pode ser atribuído à presença de múltiplas soluções articulares associadas a uma mesma pose cartesiana. Ainda assim, após a fixação da junta q_6 e a manutenção da restrição de ramo em q_5 , observou-se redução consistente dos erros e maior estabilidade no treinamento.

No trabalho de Moradinezhad, Ghasemi e Hosseini (2017), redes neurais artificiais foram aplicadas à cinemática inversa de um manipulador com três graus de liberdade. Nesse estudo, a rede foi treinada com 4818 amostras para mapear variáveis associadas à pose do efetuador diretamente para as coordenadas articulares, obtendo erros reduzidos após normalização dos dados. Comparativamente, o problema tratado nesta dissertação envolve um manipulador com seis graus de liberdade e forte acoplamento geométrico entre as juntas, especialmente na região do punho esférico, o que aumenta a dificuldade do problema de regressão.

Em contraste, Dash, Choudhury e Senapati (2019) relataram desempenho insatisfatório ao utilizar redes neurais para regressão da cinemática inversa em um manipulador com seis graus de liberdade, observando erros elevados principalmente nas juntas q_2 e q_6 . Os autores atribuem esse resultado à baixa qualidade do conjunto de dados, à ausência de otimização de hiperparâmetros e à complexidade geométrica do sistema. Esse comportamento é consistente com os resultados iniciais obtidos neste trabalho ao se considerar o espaço articular completo, quando a rede apresentou dificuldades na estimação de q_6 , com erros médios superiores a 45° e ausência de relação linear clara entre valores reais e preditos.

De modo geral, os resultados obtidos confirmam os achados da literatura, indicando que o desem-

penho de modelos baseados em aprendizado de máquina para cinemática inversa depende da unicidade do mapeamento entre pose cartesiana e variáveis articulares, bem como da qualidade do conjunto de dados utilizado no treinamento. A dificuldade observada no aprendizado do espaço articular completo e a necessidade de fixação de q_6 estão em concordância com as limitações discutidas por Dash, Choudhury e Senapati (2019). Por outro lado, a redução dos erros e a estabilização do treinamento após a eliminação da redundância associada ao punho esférico são consistentes com os resultados reportados por Šegota et al. (2021), Car et al. (2021) e Moradinezhad, Ghasemi e Hosseini (2017).

6 CONCLUSÃO

Este trabalho investigou, por meio de uma Revisão Sistemática da Literatura e de experimentos computacionais, o uso de redes neurais profundas para a aproximação da cinemática inversa do manipulador industrial ABB IRB120, avaliando como a redundância cinemática do punho esférico afeta a capacidade de aprendizado do modelo. A análise sistemática permitiu identificar arquiteturas, estratégias de treinamento, limitações e lacunas associadas ao uso de redes neurais na resolução da cinemática inversa, fornecendo subsídios para o desenvolvimento da abordagem experimental adotada neste trabalho. Foram conduzidos dois experimentos principais: o primeiro utilizando o espaço articular quase completo, com todas as juntas variando em seus limites nominais, exceto q_5 , cuja amplitude foi restringida a $[10^\circ, 120^\circ]$ para evitar a singularidade do punho; e o segundo considerando adicionalmente a fixação de q_6 , de modo a eliminar a redundância de orientação associada ao *wrist twist*.

Os resultados demonstraram que a rede neural é incapaz de aprender um mapeamento consistente quando o punho esférico opera com liberdade total. Nesse cenário, verificaram-se elevada dispersão nos gráficos *Real* $\times$ *Predito*, ausência de correlação linear entre entrada e saída e erros médios superiores a 45° (RMSE) e 33° (MAE). A análise evidenciou que a redundância da junta q_6 introduz múltiplas soluções articulares para uma mesma pose cartesiana, impossibilitando o aprendizado supervisionado de uma função determinística, mesmo quando a singularidade associada a q_5 é evitada.

Com a fixação de q_6 , o problema de regressão torna-se bem condicionado. O modelo passa a operar em um espaço articular sem redundâncias estruturais, permitindo a aprendizagem de um mapeamento unívoco entre posição, orientação e variáveis articulares. Nessa configuração, a rede neural apresentou melhoria substancial no desempenho, atingindo erros médios da ordem de 14° (RMSE) e 8.67° (MAE), com comportamento estável ao longo do treinamento e forte aderência à reta ideal nos gráficos de regressão das juntas q_2 , q_3 e q_5 .

Os experimentos confirmam que o uso de redes neurais profundas é viável para a aproximação da cinemática inversa do ABB IRB120 quando o espaço articular é adequadamente restrito para remover redundâncias estruturais do punho. Nessas condições, o modelo pode fornecer estimativas iniciais para métodos numéricos de refinamento ou ser empregado em tarefas de planejamento de movimento em que tolerâncias de erro moderadas são aceitáveis.

Como perspectivas futuras, recomenda-se investigar arquiteturas especializadas para representação de orientação, explorar parametrizações alternativas (ângulos de Euler, *rotation vectors* ou representações contínuas de $SO(3)$), empregar estratégias de amostragem direcionada para ampliar a cobertura

do espaço operacional e considerar métodos híbridos que combinem redes neurais com solucionadores analíticos ou numéricos. Tais direções podem ampliar a aplicabilidade prática de modelos baseados em aprendizado de máquina no controle cinemático de manipuladores industriais.

REFERÊNCIAS

- ARAVINDDHAKSHAN, S.; APTE, S.; AKASH, S. M. Neural network based inverse kinematic solution of a 5 dof manipulator for industrial application. **Journal of Physics: Conference Series**, IOP Publishing, v. 1969, n. 1, p. 012010, jul 2021.
- ARUNADEVI, M. et al. Machine learning approach for inverse kinematics in trajectory planning of pioneer 2 manipulator with cubic spline interpolation. In: **2021 5th International Conference on Computing Methodologies and Communication (ICCMC)**. [S.l.: s.n.], 2021. p. 807–813.
- AYDOGMUS, O.; BOZTAS, G. Implementation of singularity-free inverse kinematics for humanoid robotic arm using bayesian optimized deep neural network. **Measurement**, v. 229, p. 114471, 2024. ISSN 0263-2241. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0263224124003567>>.
- BAHANI, A. et al. The inverse kinematics evaluation of 6-dof robots in cooperative tasks using virtual modeling design and artificial intelligence tools. **International Journal of Mechanical Engineering and Robotics Research**, v. 12, n. 2, p. 121–130, 2023.
- BAI, Y.; HSIEH, S.-J. A hybrid method using fabrik and custom ann in solving inverse kinematic for generic serial robot manipulator. **The International Journal of Advanced Manufacturing Technology**, v. 130, n. 9, p. 4883–4904, 2024.
- BOUZID, R.; GRITLI, H.; NARAYAN, J. Optimized inverse kinematics of a 2-dof robotic manipulator using a hybrid approach combining an ann with a metaheuristic algorithm. In: **2024 IEEE International Conference on Artificial Intelligence Green Energy (ICAIGE)**. [S.l.: s.n.], 2024. p. 1–6.
- BOUZID, R.; GRITLI, H.; NARAYAN, J. Solving inverse kinematics of a 2-dof upper-limb exoskeleton robot through metaheuristic optimization of ann hyperparameters. In: **2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)**. [S.l.: s.n.], 2025. v. 3, p. 1–6.
- BOUZID, R.; NARAYAN, J.; GRITLI, H. Hybrid metaheuristic and artificial neural network approach for solving inverse kinematics of a scara manipulator robot. In: **2024 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)**. [S.l.: s.n.], 2024. p. 385–392.
- BOUZID, R.; NARAYAN, J.; GRITLI, H. Investigating neural network hyperparameter variations in robotic arm inverse kinematics for different arm lengths. In: **2024 Third International Conference on Power, Control and Computing Technologies (ICPC2T)**. [S.l.: s.n.], 2024. p. 351–356.
- CAGIGAS-MUÑIZ, D. Artificial neural networks for inverse kinematics problem in articulated robots. **Engineering Applications of Artificial Intelligence**, v. 126, p. 107175, 2023. ISSN 0952-1976.
- CALZADA-GARCIA, A. et al. Inverse kinematics for robotic manipulators via deep neural networks: Experiments and results. **Applied Sciences**, v. 15, n. 13, 2025. ISSN 2076-3417.
- CALZADA-GARCIA, A. et al. A review on inverse kinematics, control and planning for robotic manipulators with and without obstacles via deep neural networks. **Algorithms**, v. 18, n. 1, 2025. ISSN 1999-4893.

- CAR, Z. et al. Determining inverse kinematics of a serial robotic manipulator through the use of genetic programming. In: **Proceedings of the 8th International Congress of Serbian Society of Mechanics**. Kragujevac, Serbia: [s.n.], 2021.
- CHALMERS, I.; HEDGES, L. V.; COOPER, H. A brief history of research synthesis. **Evaluation & the Health Professions**, v. 25, p. 12–37, 3 2002. ISSN 0163-2787.
- CHASSIN, M. R. Does inappropriate use explain geographic variations in the use of health care services? **JAMA**, v. 258, p. 2533, 11 1987. ISSN 0098-7484.
- DAHARI, M.; TAN, J.-D. Forward and inverse kinematics model for robotic welding process using kr-16ks kuka robot. In: **2011 Fourth International Conference on Modeling, Simulation and Applied Optimization**. [S.l.: s.n.], 2011. p. 1–6.
- DAS, H.; SLOTINE, J.-E.; SHERIDAN, T. Inverse kinematic algorithms for redundant systems. In: **Proceedings. 1988 IEEE International Conference on Robotics and Automation**. [S.l.: s.n.], 1988. p. 43–48 vol.1.
- DASH, K. K.; CHOUDHURY, B. B.; SENAPATI, S. K. Inverse kinematics solution of a 6-dof industrial robot. In: NAYAK, J. et al. (Ed.). **Soft Computing in Data Analytics**. Singapore: Springer Singapore, 2019. (Advances in Intelligent Systems and Computing, v. 758), p. 183–192. ISBN 978-981-13-0514-6.
- DAYA, B.; KHAWANDI, S.; AKOUM, M. Applying neural network architecture for inverse kinematics problem in robotics. **Journal of Software Engineering and Applications**, v. 3, p. 230–239, 2010.
- DENG, W. et al. Physics informed machine learning model for inverse dynamics in robotic manipulators. **Applied Soft Computing**, v. 163, p. 111877, 2024. ISSN 1568-4946.
- DIKMENLI, S. Forward and inverse kinematics solution of 6-DOF robots with offset and spherical wrists. **Eurasian Journal of Science Engineering and Technology**, v. 3, n. 1, p. 14–28, 2022.
- DIPRASETYA, M. R.; PÖPPELBAUM, J.; SCHWUNG, A. Kinenn: Kinematic neural network for inverse model policy based on homogeneous transformation matrix and dual quaternion. **Robotics and Computer-Integrated Manufacturing**, v. 94, p. 102945, 2025. ISSN 0736-5845.
- ELISEEV, P.; GEGUEV, A.; IVANOV, A. Hardware design of the serial link manipulator system. In: **2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)**. [S.l.: s.n.], 2021. p. 314–317.
- ESMAT, M.; ABDEL-NASSER, M.; SAIF, A.-W. A. Solving inverse kinematics for 3r manipulator using artificial neural networks. In: **2023 20th International Multi-Conference on Systems, Signals Devices (SSD)**. [S.l.: s.n.], 2023. p. 116–120.
- GAO, R. Inverse kinematics solution of robotics based on neural network algorithms. **Journal of Ambient Intelligence and Humanized Computing**, v. 11, n. 12, p. 6199–6209, 2020.
- GEUBBELMANS, M. et al. Artificial neural networks and deep learning. **American Journal of Orthodontics and Dentofacial Orthopedics**, v. 165, n. 2, p. 248–251, fev. 2024.
- GUMUS, M. S.; KALYONCU, M. A novel architecture for artificial neural networks to solve the inverse kinematics problem in robotics. **Journal of the Brazilian Society of Mechanical Sciences and Engineering**, v. 47, n. 10, p. 469, 2025.

- GUPTA, V. et al. An overview of supervised machine learning algorithm. In: **2022 11th International Conference on System Modeling Advancement in Research Trends (SMART)**. [S.l.: s.n.], 2022. p. 87–92.
- HAMARSHEH, Q. et al. An artificial neural network approach in solving inverse kinematics of a 6 dof kuka industrial robot. In: **2023 20th International Multi-Conference on Systems, Signals Devices (SSD)**. [S.l.: s.n.], 2023. p. 157–163.
- HO, C.-K. et al. A deep learning approach to navigating the joint solution space of redundant inverse kinematics and its applications to numerical ik computations. **IEEE Access**, v. 11, p. 2274–2290, 2023.
- HSIEH, Y.-Z.; XU, F.-X.; LIN, S.-S. Deep convolutional generative adversarial network for inverse kinematics of self-assembly robotic arm based on the depth sensor. **IEEE Sensors Journal**, v. 23, n. 1, p. 758–765, 2023.
- HUANG, J. et al. A new method for solving inverse kinematics of an industrial robot. In: **2012 International Conference on Computer Science and Electronics Engineering**. [S.l.: s.n.], 2012. v. 3, p. 53–56.
- HUANG, L.; PANG, H.; ZHANG, H. Research on the forward kinematics of stewart platform based on broyden algorithm and newton-raphson algorithm. In: **2023 9th Annual International Conference on Network and Information Systems for Computers (ICNISC)**. [S.l.: s.n.], 2023. p. 402–406.
- IBARRA-PÉREZ, T. et al. A novel inverse kinematic solution of a six-dof robot using neural networks based on the taguchi optimization technique. **Applied Sciences**, v. 12, n. 19, 2022. ISSN 2076-3417.
- JANG, J.; SUN, C.; MIZUTANI, E. Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [book review]. **IEEE Transactions on Automatic Control**, v. 42, n. 10, p. 1482–1484, 1997.
- JIMÉNEZ-LÓPEZ, E. et al. Modeling of inverse kinematic of 3-dof robot, using unit quaternions and artificial neural network. **Robotica**, v. 39, n. 7, p. 1230–1250, 2021.
- JOSHI, R. C. et al. Optimized inverse kinematics modeling and joint angle prediction for six-degree-of-freedom anthropomorphic robots with explainable ai. **ISA Transactions**, v. 157, p. 340–356, 2025. ISSN 0019-0578.
- Kamlesh Shah, S.; MISHRA, R.; Samprit Ray, L. Solution and validation of inverse kinematics using deep artificial neural network. **Materials Today: Proceedings**, v. 26, p. 1250–1254, 2020. ISSN 2214-7853. 10th International Conference of Materials Processing and Characterization.
- KITCHENHAM, B. A.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering**. [S.l.], 2007.
- KRIEGESKORTE, N.; GOLAN, T. Neural network models and deep learning. **Current Biology**, v. 29, n. 7, p. R231–R236, 2019. ISSN 0960-9822.
- LAME, G. Systematic literature reviews: An introduction. **Proceedings of the Design Society: International Conference on Engineering Design**, Cambridge University Press, v. 1, p. 1633–1642, 7 2019. ISSN 2220-4342.
- LAUREL, K. E. Y. et al. Solving the inverse kinematics of a five axis cnc machine using shallow and deep neural networks. In: **2023 IEEE 15th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)**. [S.l.: s.n.], 2023. p. 1–6.

- LIMA, A. T. et al. Teleoperation of an abb irb 120 robotic manipulator and barretthand bh8-282 using a geomagic touch x haptic device and ros. In: **2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)**. [S.l.: s.n.], 2018. p. 188–193.
- LIU, Z. et al. Modeling and control of robotic manipulators based on artificial neural networks: A review. **Iranian Journal of Science and Technology, Transactions of Mechanical Engineering**, v. 47, n. 4, p. 1307–1347, 2023.
- LIU, Z. et al. Modeling and control of robotic manipulators based on Artificial Neural Networks: A review. **Iranian Journal of Science and Technology, Transactions of Mechanical Engineering**, v. 47, n. 4, p. 1307–1347, 2023. ISSN 2228-6187.
- LU, J.; ZOU, T.; JIANG, X. A neural network based approach to inverse kinematics problem for general six-axis robots. **Sensors**, v. 22, n. 22, 2022. ISSN 1424-8220.
- M., N. et al. Estimation of inverse kinematics solutions of a 2d planar robotic manipulator using feed-forward neural network. In: **2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS)**. [S.l.: s.n.], 2024. p. 1040–1044.
- MALIK, A. et al. A deep reinforcement-learning approach for inverse kinematics solution of a high degree of freedom robotic manipulator. **Robotics**, v. 11, n. 2, 2022. ISSN 2218-6581.
- MANJEGOWDA, N.; RAO, M. Path-based evaluation of deep learning models for solving inverse kinematics in a revolute-prismatic robot. **Scientific Reports**, v. 15, n. 1, p. 33953, 2025. Disponível em: <<https://doi.org/10.1038/s41598-025-10940-z>>.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, n. 4, p. 115–133, dez. 1943. ISSN 0007-4985.
- MORADINEZHAD, R.; GHASEMI, J.; HOSSEINI, M. Kinematic synthesis of parallel manipulator via neural network approach. **International Journal of Engineering**, v. 30, n. 9, p. 1363–1370, set. 2017.
- MURCIA, H. D. C.; CASTRO, C. L. G.; RIVILLAS, J. A. C. Comparison of neural networks with a numerical method for obtaining the inverse kinematic of a bipedal robot. **2025 IEEE Colombian Conference on Applications of Computational Intelligence (ColCACI)**, p. 1–6, 2025.
- NAEEM, S. et al. An unsupervised machine learning algorithms: Comprehensive review. **International Journal of Computing and Digital Systems**, v. 13, p. 911 – 921, 2023.
- NGUYEN, H.-T.; CHEAH, C. C.; TOH, K.-A. An analytic layer-wise deep learning framework with applications to robotics. **Automatica**, v. 135, p. 110007, 2022. ISSN 0005-1098.
- NUGROHO, A.; YUNIARNO, E. M.; PURNOMO, M. H. Arkoma dataset: An open-source dataset to develop neural networks-based inverse kinematics model for nao robot arms. **Data in Brief**, v. 51, p. 109727, 2023. ISSN 2352-3409.
- NURBA, H. P. et al. Performance evaluation of 3 dof arm robot with forward kinematics denavit-hartenberg method for coffee maker machine. In: **2022 16th International Conference on Telecommunication Systems, Services, and Applications (TSSA)**. [S.l.: s.n.], 2022. p. 1–6.
- PARIKH, P.; LAM, S. A hybrid strategy to solve the forward kinematics problem in parallel manipulators. **IEEE Transactions on Robotics**, v. 21, n. 1, p. 18–25, 2005.

- PETTICREW, M.; ROBERTS, H. **Reviews in the Social Sciences: A Practical Guide**. [S.l.]: Blackwel, 2005. v. 6. 1-336 p. ISSN 1473-3145.
- PETTIT, R. W. et al. Artificial intelligence, machine learning, and deep learning for clinical outcome prediction. **Emerging Topics in Life Sciences**, v. 5, n. 6, p. 729–745, dez. 2021. ISSN 2397-8554.
- POLYDOROS, A. S.; NALPANTIDIS, L. A reservoir computing approach for learning forward dynamics of industrial manipulators. In: **2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2016. p. 612–618.
- POZNA, C. R.; HORVÁTH, E.; HOLLÓSI, J. The inverse kinematics problem, a heuristical approach. In: **2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMII)**. [S.l.: s.n.], 2016. p. 299–304.
- RAJA, R.; DUTTA, A.; DASGUPTA, B. Learning framework for inverse kinematics of a highly redundant mobile manipulator. **Robotics and Autonomous Systems**, v. 120, p. 103245, 2019. ISSN 0921-8890.
- REN, H.; BEN-TZVI, P. Learning inverse kinematics and dynamics of a robotic manipulator using generative adversarial networks. **Robotics and Autonomous Systems**, v. 124, p. 103386, 2020. ISSN 0921-8890.
- RICE, S.; AZAB, A.; SAAD, S. Fusion ik: Solving inverse kinematics using a hybridized deep learning and evolutionary approach. **Manufacturing Letters**, v. 41, p. 9–18, 2024. ISSN 2213-8463. 52nd SME North American Manufacturing Research Conference (NAMRC 52). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2213846324000622>>.
- SACKETT, D. L. et al. Evidence based medicine: what it is and what it isn't. **BMJ**, v. 312, p. 71–72, 1996. ISSN 0959-8138.
- SARAVANAN, R.; SUJATHA, P. A state of art techniques on machine learning algorithms: A perspective of supervised learning approaches in data classification. In: **2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)**. [S.l.: s.n.], 2018. p. 945–949.
- ŠEGOTA, S. B. et al. Utilization of multilayer perceptron for determining the inverse kinematics of an industrial robotic manipulator. **International Journal of Advanced Robotic Systems**, v. 18, n. 4, p. 1729881420925283, 2021.
- SEMWAL, V. B. et al. Generation of inverse kinematics(ik) trajectory of 3 link manipulator using deep learning model. **2025 IEEE International Conference on Computer, Electronics, Electrical Engineering & their Applications (IC2E3)**, p. 1–6, 2025.
- SERRE, T. Deep learning: The good, the bad, and the ugly. **Annual Review of Vision Science**, v. 5, p. 399–426, set. 2019.
- SHAAR, A.; GHAEB, J. A. Intelligent solution for inverse kinematic of industrial robotic manipulator based on rnn. In: **2023 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)**. [S.l.: s.n.], 2023. p. 74–79.
- SHAH, S. K.; MISHRA, R. Experimental validation for inverse kinematics of a five axis robotic manipulator using deep artificial neural network. **International Journal of Innovative Technology and Exploring Engineering (IJITEE)**, v. 9, n. 1, p. 3943–3946, 2019.
- SHAREEF, I. R. Artificial neural network-based approach for inverse kinematics of puma 260 robot. **International Journal of Integrated Engineering**, 2024.

- SIŁKA, J.; WIECZOREK, M. Custom sequential neural network for inverse kinematic regression. In: **2025 International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2025. p. 1–8.
- TAGLIANI, F. L.; PELLEGRINI, N.; AGGOGERI, F. Machine learning sequential methodology for robot inverse kinematic modelling. **Applied Sciences**, v. 12, n. 19, 2022. ISSN 2076-3417.
- TANG, H.; NOTASH, L. Neural network-based transfer learning of manipulator inverse displacement analysis. **Journal of Mechanisms and Robotics**, v. 13, n. 3, p. 035004, 04 2021. ISSN 1942-4302.
- TOQUICA, J. S. et al. An analytical and a deep learning model for solving the inverse kinematic problem of an industrial parallel robot. **Computers Industrial Engineering**, v. 151, p. 106682, 2021. ISSN 0360-8352.
- VERCIO, L. L. et al. Supervised machine learning tools: A tutorial for clinicians. **Journal of Neural Engineering**, v. 17, n. 6, p. 062001, nov. 2020. ISSN 1741-2552.
- VOULODIMOS, A. et al. Deep learning for computer vision: A brief review. **Computational Intelligence and Neuroscience**, v. 2018, p. 7068349, 2018.
- VU, M. et al. Machine learning-based framework for optimally solving the analytical inverse kinematics for redundant manipulators. **Mechatronics**, v. 91, p. 102970, 2023. ISSN 0957-4158.
- WAGAA, N.; KALLEL, H.; MELLOULI, N. Analytical and deep learning approaches for solving the inverse kinematic problem of a high degrees of freedom robotic arm. **Engineering Applications of Artificial Intelligence**, v. 123, p. 106301, 2023. ISSN 0952-1976.
- WANG, X. et al. Deep-learning damped least squares method for inverse kinematics of redundant robots. **Measurement**, v. 171, p. 108821, 2021. ISSN 0263-2241.
- WHITTINGTON, J. C.; BOGACZ, R. Theories of error back-propagation in the brain. **Trends in Cognitive Sciences**, v. 23, n. 3, p. 235–250, 2019. ISSN 1364-6613.
- YANG, S.; MOGHAVVEMI, M.; TOLMAN, J. Modeling of robot inverse kinematics using two ann paradigms. In: **2000 TENCON Proceedings. Intelligent Systems and Technologies for the New Millennium (Cat. No.00CH37119)**. [S.l.: s.n.], 2000. v. 3, p. 173–177 vol.3.
- ZAFAR, M. H.; MOOSAVI, S. K. R.; SANFILIPPO, F. Inverse kinematic modelling of a 3-dof robotic manipulator using hybrid deep learning models. **Procedia CIRP**, v. 120, p. 213–218, 2023. ISSN 2212-8271. 56th CIRP International Conference on Manufacturing Systems 2023.
- ZHANG, Y. et al. Invertible liquid neural network-based learning of inverse kinematics and dynamics for robotic manipulators. **Scientific Reports**, v. 15, n. 1, p. 42311, 2025.
- ZHAO, C. et al. Inverse kinematics solution and control method of 6-degree-of-freedom manipulator based on deep reinforcement learning. **Scientific Reports**, v. 14, n. 1, p. 12467, 2024.
- ZHONG, C. et al. Acousnet: A deep learning based approach to dynamic 3d holographic acoustic field generation from phased transducer array. **IEEE Robotics and Automation Letters**, v. 7, n. 2, p. 666–673, 2022.
- ZHOU, Z. et al. Inverse kinematics solution for robotic manipulator based on extreme learning machine and sequential mutation genetic algorithm. **International Journal of Advanced Robotic Systems**, v. 15, n. 4, p. 1729881418792992, 2018.

APÊNDICE A – GERAÇÃO DO DATASET SINTÉTICO PARA O ABB IRB120

```

1 import numpy as np
2 import pandas as pd
3
4 from sklearn.cluster import MiniBatchKMeans
5 from sklearn.preprocessing import StandardScaler
6 from sklearn.metrics import pairwise_distances_argmin_min
7
8 # DH parameters for ABB IRB120
9 pi = np.pi
10 a = np.array([0.0, 0.270, 0.070, 0.0, 0.0, 0.0])
11 d = np.array([0.290, 0.0, 0.0, 0.302, 0.0, 0.072])
12 alpha = np.array([-pi/2, 0.0, -pi/2, pi/2, -pi/2, 0.0])
13 theta_offset = np.zeros(6)
14
15 # Joint limits with branch restriction
16 q_min = np.deg2rad(np.array([-165, -110, -70, -160, 10, 0]))
17 q_max = np.deg2rad(np.array([165, 110, 110, 160, 120, 0]))
18
19 # DH transform
20 def dh_transform(a_i, d_i, alpha_i, theta_i):
21     ca, sa = np.cos(alpha_i), np.sin(alpha_i)
22     ct, st = np.cos(theta_i), np.sin(theta_i)
23     return np.array([
24         [ ct, -st*ca, st*sa, a_i*ct],
25         [ st, ct*ca, -ct*sa, a_i*st],
26         [ 0, sa, ca, d_i],
27         [ 0, 0, 0, 1]
28     ])
29
30 # Forward kinematics
31 def fkine_irb120(q):
32     T = np.eye(4)
33     for i in range(6):
34         T = T @ dh_transform(a[i], d[i], alpha[i], q[i] + theta_offset[i])
35     return T
36
37

```

```

38 # Rotation matrix to quaternion (w, x, y, z)
39 def rotm_to_quat(R):
40     tr = R[0,0] + R[1,1] + R[2,2]
41     if tr > 0:
42         S = np.sqrt(tr + 1.0) * 2.0
43         qw = 0.25 * S
44         qx = (R[2,1] - R[1,2]) / S
45         qy = (R[0,2] - R[2,0]) / S
46         qz = (R[1,0] - R[0,1]) / S
47     elif (R[0,0] > R[1,1]) and (R[0,0] > R[2,2]):
48         S = np.sqrt(1.0 + R[0,0] - R[1,1] - R[2,2]) * 2.0
49         qw = (R[2,1] - R[1,2]) / S
50         qx = 0.25 * S
51         qy = (R[0,1] + R[1,0]) / S
52         qz = (R[0,2] + R[2,0]) / S
53     elif R[1,1] > R[2,2]:
54         S = np.sqrt(1.0 + R[1,1] - R[0,0] - R[2,2]) * 2.0
55         qw = (R[0,2] - R[2,0]) / S
56         qx = (R[0,1] + R[1,0]) / S
57         qy = 0.25 * S
58         qz = (R[1,2] + R[2,1]) / S
59     else:
60         S = np.sqrt(1.0 + R[2,2] - R[0,0] - R[1,1]) * 2.0
61         qw = (R[1,0] - R[0,1]) / S
62         qx = (R[0,2] + R[2,0]) / S
63         qy = (R[1,2] + R[2,1]) / S
64         qz = 0.25 * S
65
66     q = np.array([qw, qx, qy, qz])
67     return q / np.linalg.norm(q)
68
69 # Candidate generation
70 N_CAND = 60000
71 N_FINAL = 10000
72
73 q_cand = []
74 pose_cand = []
75
76 for _ in range(N_CAND):
77     q_rand = q_min + (q_max - q_min) * np.random.rand(6)

```

```

78     T = fkine_irbl20(q_rand)
79     p = T[:3, 3]
80     qw, qx, qy, qz = rotm_to_quat(T[:3, :3])
81     q_cand.append(q_rand)
82     pose_cand.append([p[0], p[1], p[2], qw, qx, qy, qz])
83
84 q_cand = np.array(q_cand)
85 pose_cand = np.array(pose_cand)
86
87 # Pose space clustering
88 scaler = StandardScaler()
89 pose_norm = scaler.fit_transform(pose_cand)
90
91 kmeans = MiniBatchKMeans(
92     n_clusters=N_FINAL,
93     batch_size=2048,
94     max_iter=200,
95     random_state=42
96 )
97 kmeans.fit(pose_norm)
98
99 closest_idx, _ = pairwise_distances_argmin_min(
100     kmeans.cluster_centers_,
101     pose_norm
102 )
103
104 closest_idx = np.unique(closest_idx)
105 q_final = q_cand[closest_idx]
106 pose_fin = pose_cand[closest_idx]
107
108 # Shuffle
109 perm = np.random.permutation(q_final.shape[0])
110 q_final = q_final[perm]
111 pose_fin = pose_fin[perm]
112
113 # DataFrame and CSV output
114 df = pd.DataFrame({
115     "q1": q_final[:,0],
116     "q2": q_final[:,1],
117     "q3": q_final[:,2],

```

```
118     "q4": q_final[:,3],
119     "q5": q_final[:,4],
120     "q6": q_final[:,5],
121     "x": pose_fin[:,0],
122     "y": pose_fin[:,1],
123     "z": pose_fin[:,2],
124     "qw": pose_fin[:,3],
125     "qx": pose_fin[:,4],
126     "qy": pose_fin[:,5],
127     "qz": pose_fin[:,6]
128 })
129
130 df.to_csv("irb120_ik_dataset_quat.csv", index=False)
131 print(df.shape)
```

Listing A.1 – Código de geração do dataset sintético do ABB IRB120

APÊNDICE B – TREINAMENTO DA REDE NEURAL PROFUNDA

```

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 from sklearn.model_selection import train_test_split
6 from sklearn.preprocessing import MinMaxScaler
7 from sklearn.metrics import mean_squared_error, mean_absolute_error
8
9 from keras.models import Sequential
10 from keras.layers import Dense, Dropout
11 from keras.optimizers import Adam
12 from keras.regularizers import l2
13 from keras.callbacks import EarlyStopping
14
15 # Auxiliary functions for angular error
16 def wrap_to_pi(angle_rad):
17     """Wrap angle in radians to [-pi, pi]."""
18     return (angle_rad + np.pi) % (2 * np.pi) - np.pi
19
20 def wrap_to_180(angle_deg):
21     """Wrap angle in degrees to [-180, 180]."""
22     return (angle_deg + 180.0) % 360.0 - 180.0
23
24 # 1. Load dataset (single-branch, quaternion)
25 csv_name = "irb120_ik_dataset_quat.csv"
26 DATA = pd.read_csv(csv_name)
27
28 print("Dataset shape:", DATA.shape)
29 print("Columns:", list(DATA.columns))
30 print(DATA.head())
31
32 # Inputs: position + orientation (quaternion)
33 X = DATA[["x", "y", "z", "qw", "qx", "qy", "qz"]].values # (N, 7)
34
35 # Outputs: joint angles
36 joint_names = ["q1", "q2", "q3", "q4", "q5", "q6"]
37 Y = DATA[joint_names].values # (N, 6)

```

```

38
39 print("X shape:", X.shape)
40 print("Y shape:", Y.shape)
41
42 # 2. Normalization and train/test split
43 sc_in = MinMaxScaler(feature_range=(-1, 1))
44 sc_out = MinMaxScaler(feature_range=(-1, 1))
45
46 X_scaled = sc_in.fit_transform(X)
47 Y_scaled = sc_out.fit_transform(Y)
48
49 X_train, X_test, Y_train, Y_test = train_test_split(
50     X_scaled, Y_scaled,
51     test_size=0.2,
52     shuffle=True,
53     random_state=42
54 )
55
56 print("X_train:", X_train.shape, "X_test:", X_test.shape)
57 print("Y_train:", Y_train.shape, "Y_test:", Y_test.shape)
58
59 # 3. Deep neural network (7 inputs, 6 outputs)
60 model = Sequential()
61 model.add(
62     Dense(
63         256,
64         activation="relu",
65         input_dim=7,
66         kernel_initializer="he_normal",
67         kernel_regularizer=l2(1e-4)
68     )
69 )
70 model.add(Dropout(0.1))
71 model.add(
72     Dense(
73         512,
74         activation="relu",
75         kernel_initializer="he_normal",
76         kernel_regularizer=l2(1e-4)
77     )

```

```

78 )
79 model.add(Dropout(0.2))
80 model.add(
81     Dense(
82         256,
83         activation="relu",
84         kernel_initializer="he_normal",
85         kernel_regularizer=l2(1e-4)
86     )
87 )
88 model.add(Dropout(0.1))
89 model.add(Dense(6, activation="linear")) # q1..q6
90
91 optimizer = Adam(learning_rate=1e-3)
92 model.compile(loss="mse", optimizer=optimizer, metrics=["mae"])
93
94 print(model.summary())
95
96 # 4. Early stopping
97 early = EarlyStopping(
98     monitor="val_loss",
99     patience=25,
100     restore_best_weights=True,
101     min_delta=1e-6,
102     verbose=1
103 )
104
105 # 5. Training
106 EPOCHS = 400
107 BATCH = 64
108
109 history = model.fit(
110     X_train, Y_train,
111     epochs=EPOCHS,
112     batch_size=BATCH,
113     validation_data=(X_test, Y_test),
114     callbacks=[early],
115     verbose=1
116 )
117

```

```

118 print("\nTraining finished.")
119
120 # 6. Evaluation: RMSE and MAE in rad and deg (with angle wrap)
121 # Prediction on test set (normalized space)
122 Y_pred_scaled = model.predict(X_test)
123
124 # Back to original joint space (radians)
125 Y_test_orig = sc_out.inverse_transform(Y_test)
126 Y_pred      = sc_out.inverse_transform(Y_pred_scaled)
127
128 rmse_rad = []
129 mae_rad  = []
130
131 print("\nPer-joint metrics (rad), angular error in [-pi, pi]:")
132 for i, name in enumerate(joint_names):
133     e_rad = wrap_to_pi(Y_pred[:, i] - Y_test_orig[:, i])
134     rmse = np.sqrt(mean_squared_error(np.zeros_like(e_rad), e_rad))
135     mae  = mean_absolute_error(np.zeros_like(e_rad), e_rad)
136     rmse_rad.append(rmse)
137     mae_rad.append(mae)
138     print(f"{name}: RMSE = {rmse:.6f} rad | MAE = {mae:.6f} rad")
139
140 print(f"\nMean RMSE (rad): {np.mean(rmse_rad):.6f}")
141 print(f"Mean MAE (rad): {np.mean(mae_rad):.6f}")
142
143 # Same in degrees
144 Y_test_deg = np.rad2deg(Y_test_orig)
145 Y_pred_deg = np.rad2deg(Y_pred)
146
147 rmse_deg = []
148 mae_deg  = []
149
150 print("\nPer-joint metrics (deg), angular error in [-180, 180]:")
151 for i, name in enumerate(joint_names):
152     e_deg = wrap_to_180(Y_pred_deg[:, i] - Y_test_deg[:, i])
153     rmse = np.sqrt(mean_squared_error(np.zeros_like(e_deg), e_deg))
154     mae  = mean_absolute_error(np.zeros_like(e_deg), e_deg)
155     rmse_deg.append(rmse)
156     mae_deg.append(mae)
157     print(f"{name}: RMSE = {rmse:.3f} deg | MAE = {mae:.3f} deg")

```

```

158
159 print(f"\nMean RMSE (deg): {np.mean(rmse_deg):.3f} deg")
160 print(f"Mean MAE (deg): {np.mean(mae_deg):.3f} deg")
161
162 # 7. Training history plots
163 plt.figure(figsize=(10, 5))
164 plt.plot(history.history["loss"], label="Train - loss")
165 plt.plot(history.history["val_loss"], label="Validation - loss")
166 plt.xlabel("Epoch")
167 plt.ylabel("MSE")
168 plt.title("Training history - Loss")
169 plt.legend()
170 plt.grid(True)
171 plt.show()
172
173 plt.figure(figsize=(10, 5))
174 plt.plot(history.history["mae"], label="Train - MAE")
175 plt.plot(history.history["val_mae"], label="Validation - MAE")
176 plt.xlabel("Epoch")
177 plt.ylabel("MAE")
178 plt.title("Training history - MAE")
179 plt.legend()
180 plt.grid(True)
181 plt.show()
182
183 # 8. Real vs predicted (deg) scatter plots for q1..q6
184 plt.figure(figsize=(15, 8))
185 for i, name in enumerate(joint_names):
186     plt.subplot(2, 3, i + 1)
187     plt.scatter(Y_test_deg[:, i], Y_pred_deg[:, i], s=8, alpha=0.5)
188     m = min(Y_test_deg[:, i].min(), Y_pred_deg[:, i].min())
189     M = max(Y_test_deg[:, i].max(), Y_pred_deg[:, i].max())
190     plt.plot([m, M], [m, M], "r--")
191     plt.xlabel("Real (deg)")
192     plt.ylabel("Predicted (deg)")
193     plt.title(f"{name} - Real x Predicted")
194     plt.grid(True)
195 plt.tight_layout()
196 plt.show()

```

Listing B.1 – Código de treinamento da DNN para aproximacao da cinematica inversa do ABB IRB120

APÊNDICE C – CODIGO DE INFERENCIA PARA A CINEMATICA INVERSA APROXIMADA

```

1  import numpy as np
2  import pandas as pd
3
4  from sklearn.preprocessing import MinMaxScaler
5  from keras.models import load_model
6
7  # 1. Load dataset used to fit the scalers
8  csv_name = "irb120_ik_dataset_quat.csv"
9  DATA = pd.read_csv(csv_name)
10
11 # Inputs: position + orientation (quaternion)
12 X = DATA[["x", "y", "z", "qw", "qx", "qy", "qz"]].values
13
14 # Outputs: joint angles
15 joint_names = ["q1", "q2", "q3", "q4", "q5", "q6"]
16 Y = DATA[joint_names].values
17
18 # 2. Rebuild input/output scalers (same configuration as in training)
19 sc_in = MinMaxScaler(feature_range=(-1, 1))
20 sc_out = MinMaxScaler(feature_range=(-1, 1))
21
22 sc_in.fit(X)
23 sc_out.fit(Y)
24
25 # 3. Load trained model from file
26 model = load_model("dnn_irb120_model.h5")
27
28 # 4. Inverse kinematics approximation using full pose (x, y, z, quaternion)
29 def ik_full_quat(x, y, z, qw, qx, qy, qz):
30     """
31     Approximate IK using the trained DNN.
32     Input:  x, y, z, qw, qx, qy, qz  (SI units and unit quaternion)
33     Output: joint vector [q1..q6] in radians
34     """
35     arr = np.array([[x, y, z, qw, qx, qy, qz]], dtype=float)
36     arr_s = sc_in.transform(arr)

```

```

37     q_s    = model.predict(arr_s)
38     q      = sc_out.inverse_transform(q_s)
39     return q[0] # shape (6,)
40
41 # 5. Example of usage
42 if __name__ == "__main__":
43     # Example pose: position (0.2, 0.1, 0.3) and identity quaternion
44     qw_ex, qx_ex, qy_ex, qz_ex = 1.0, 0.0, 0.0, 0.0
45     q_pred = ik_full_quat(0.2, 0.1, 0.3, qw_ex, qx_ex, qy_ex, qz_ex)
46
47     print("Predicted joints (rad):", q_pred)
48     print("Predicted joints (deg):", np.rad2deg(q_pred))

```

Listing C.1 – Código de inferencia utilizando a DNN treinada para o ABB IRB120