

Aproximação da Cinemática Inversa de um Robô Manipulador de 5 Juntas Rotativas através de Redes Neurais Globais e Locais

José Matheus Soares Ferreira

Pós-Graduação Eng. Elétrica e Computação
Universidade Federal do Ceará
Sobral, CE, Brasil
matheus.soares8890@gmail.com

David Nascimento Coelho

Engenharia da Computação
Universidade Federal do Ceará
Sobral, CE, Brasil
david.coelho@sobral.ufc.br

Icaro Bezerra Viana

Pós-Graduação Eng. Elétrica e Computação
Universidade Federal do Ceará
Sobral, CE, Brasil
icaro.viana@ufc.br

Abstract—The inverse kinematics calculation of robotic manipulators with multiple degrees of freedom is a challenge, where analytical methods are often not feasible. In this work, the approximation of the inverse kinematics of a robotic manipulator with five revolute joints is performed using two machine learning approaches: global models and local models, both based on Multi Layer Perceptron (MLP) neural networks. Initially, a dataset relating the joint and Cartesian spaces is generated through forward kinematics. Then, this dataset is filtered to remove redundant and unreachable samples for the manipulator. Finally, the models hyperparameters are optimized, and their performance is evaluated using the Mean Squared Error (MSE) in both the joint and Cartesian spaces. The results show that both approaches are viable for the given task, with the local approach outperforming the global one.

Index Terms—Robotics Manipulators; Multilayer Perceptron; Inverse Kinematics; Local Neural Network; Global Neural Network.

I. INTRODUÇÃO

A robótica tem desempenhado um papel essencial em diversas áreas da sociedade, desde a indústria, passando pela exploração espacial, e até a medicina, sendo impulsionada devido aos robôs possuírem uma maior eficiência, agilidade, precisão, flexibilidade e capacidade de executar tarefas perigosas ou de risco à saúde humana. Dentro desse campo, a modelagem da cinemática de manipuladores robóticos é fundamental para determinar sua movimentação no espaço [1].

Na robótica, há uma relação entre o espaço das juntas e o espaço operacional de um manipulador, e por isso a cinemática é dividida entre direta e inversa. A cinemática inversa é responsável por calcular as configurações das juntas a partir de uma posição desejada do efetuador. Diferente da cinemática direta, a inversa possui uma complexidade maior, já que é entendida como um problema mal posto, e apresenta desafios significativos devido à sua natureza não linear das equações, bem como à solvabilidade, o que dificulta a obtenção de uma solução fechada [2], [3].

Embora existam condições específicas que permitem a obtenção de soluções analíticas — como a presença de eixos de juntas que se cruzam ou ângulos articulares simplificados

—, a maioria dos manipuladores com muitos graus de liberdade (GDL) exigem abordagens numéricas para resolução da cinemática inversa. Segundo [3], mesmo quando o problema é solucionável, a resposta geralmente depende de métodos numéricos, sendo a solução analítica uma exceção, e não a regra. Além disso, o cálculo numérico tende a ser mais custoso computacionalmente do que a avaliação de expressões analíticas, o que torna desejável o desenvolvimento de soluções alternativas mais eficientes.

Para lidar com essas dificuldades, métodos baseados em aprendizado de máquina (AM) vêm sendo explorados nas últimas décadas, permitindo aproximar a cinemática inversa sem a necessidade de soluções analíticas complexas de manipuladores com muitos GDL. Algoritmos como Redes Neurais Artificiais (RNAs), *Support Vector Machines*, Regressão por Processos Gaussianos (PG) e Modelos Neuro-Fuzzy têm sido aplicados para modelar a relação entre coordenadas cartesianas e ângulos articulares, proporcionando maior flexibilidade e adaptabilidade ao problema [4].

Em [5], por exemplo, os autores exploraram a resolução da cinemática inversa de robôs seriais utilizando RNAs e Sistemas Artificiais de Inferência Neuro-Fuzzy, treinados para prever as configurações articulares de robôs com até sete GDL para uma determinada região limitada do espaço cartesiano. Os modelos demonstraram precisão significativa, apesar de desafios como erros de aproximação e tempo de treinamento. O trabalho apresentado em [6] propôs um algoritmo de redes neurais que aproxima a modelagem inversa em uma sub-região limitada do volume de trabalho de um robô de seis GDL, utilizando retropropagação com função de excitação adaptativa para melhorar a precisão e velocidade da solução. O estudo demonstrou que essa abordagem tornou-se uma alternativa promissora para a modelagem da cinemática inversa.

Em [4], sete algoritmos de AM foram comparados na solução da cinemática inversa limitando o sub-espaço operacional dos manipuladores Motoman HP6, PUMA 560 e Robô Planar, destacando o *Least Squares Support Vector Regression* e PG como os mais precisos para um conjunto específico de ângulos. Já o autor em [7] avaliou seis modelos locais para os mesmos

manipuladores e limitações operacionais, utilizando dados da modelagem direta em notação de Denavit-Hartenberg (DH). Os erros quadráticos médios (Mean Squared Error, MSE) e a distância entre trajetórias no espaço cartesiano foram analisados, e os modelos *Radial Basis Function* e *Local Model Network* apresentaram os melhores desempenhos globais.

Nesse contexto, este trabalho compara abordagens globais e locais de AM na aproximação da cinemática inversa de um manipulador robótico com 5 graus de liberdade. Diferentemente dos estudos anteriores, este busca mapear todo o volume de trabalho alcançável do espaço operacional do manipulador, superando desafios como não linearidade e singularidades. Para isso, propõe-se uma abordagem para tratar das singularidades de dados redundantes com técnicas de similaridade, e a remoção de pontos inalcançáveis. Adicionalmente, avalia-se os erros em dois espaços distintos, tanto das juntas como operacional, utilizando atributos de posição e orientação do efetuador. A metodologia adotada considera a estrutura física de um manipulador real, cuja modelagem direta é obtida por meio da notação de DH, permitindo a geração dos padrões de treinamento referentes às configurações do efetuador nesse modelo.

II. CINEMÁTICO DE MANIPULADORES

A compreensão e aplicação da cinemática são fundamentais para resolver o problema abordado neste trabalho, pois permitem descrever matematicamente o comportamento do manipulador robótico. Ao estabelecer a relação entre o espaço das juntas e o espaço operacional, essa modelagem fornece a base necessária para a geração dos padrões de treinamento, além de subsidiar a validação das soluções propostas por meio de técnicas de aprendizado de máquina.

A cinemática de manipuladores aborda o movimento dos robôs de base fixa sem considerar as forças envolvidas no mecanismo, focando na modelagem matemática da posição e orientação do efetuador em função dos deslocamentos das juntas. Esta tem como objetivo estabelecer a relação entre o espaço das juntas, definido pelas variáveis angulares das articulações, e o espaço operacional, caracterizado pela posição e orientação do efetuador no referencial cartesiano [1], [3].

A cinemática, na robótica, é tradicionalmente dividida em direta e inversa. A cinemática direta busca determinar a posição e a orientação do efetuador a partir dos ângulos das juntas e das dimensões geométricas dos elos, considerando o referencial da base do robô [3].

Para manipuladores com maior complexidade e elevado número de GDLs, utiliza-se a notação Denavit-Hartenberg. Essa abordagem fornece um padrão para modelar as translações e rotações entre os elos, facilitando a definição de sistemas de coordenadas para o robô. Na notação, quatro parâmetros são definidos para cada junta do manipulador: o ângulo de rotação da junta (θ_i), o deslocamento do elo (d_i), o comprimento do elo (a_i) e o ângulo de torção (α_i). Esses parâmetros descrevem, respectivamente, uma rotação em torno do eixo z , uma translação ao longo do eixo z , uma translação ao longo do eixo x , e uma rotação em torno do eixo x , conforme (1).

Com base nesses parâmetros, é possível construir uma matriz de transformação homogênea genérica, como pode ser visto em (2).

$$T_i^{i-1} = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \quad (1)$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

onde $s_{\theta_i} = \sin(\theta_i)$, $c_{\theta_i} = \cos(\theta_i)$, $s_{\alpha_i} = \sin(\alpha_i)$ e $c_{\alpha_i} = \cos(\alpha_i)$.

Para cada elo na cadeia cinemática do manipulador, é calculada uma matriz de transformação correspondente ao seu sistema de coordenadas locais, que, quando combinadas (3), descrevem a posição e a orientação do efetuador em relação à base [3], [8].

$$T_n^0 = T_1^0 \dots T_n^{n-1} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Em contrapartida, a cinemática inversa é mais complexa que a cinemática direta, pois envolve equações não lineares, tornando-se um problema mal posto. Isso significa que podem existir múltiplas soluções, nenhuma solução ou soluções não admissíveis para determinar a posição do efetuador final, impossibilitando uma solução de forma analítica. Tipicamente, as soluções da cinemática inversa são classificadas em analíticas e numéricas. As analíticas utilizam identidades trigonométricas e álgebra linear para resolver as equações de forma exata, sendo mais aplicáveis a manipuladores simples. Já as numéricas empregam métodos iterativos para lidar com a não linearidade das equações, oferecendo soluções aproximadas, sendo mais adequadas a sistemas com muitos graus de liberdade e maior complexidade [2], [3].

III. APRENDIZADO DE MÁQUINA

O uso de técnicas AM neste trabalho surge como uma alternativa para aproximar soluções da cinemática inversa de manipuladores robóticos, especialmente em cenários onde a obtenção analítica é complexa. Ao explorar padrões presentes nos dados gerados a partir da modelagem cinemática, os algoritmos de AM são empregados como ferramentas para mapear de forma preditiva a relação entre a posição desejada do efetuador e os ângulos articulares correspondentes, contribuindo diretamente para a solução do problema proposto.

O aprendizado de máquina é um campo da inteligência artificial que permite aos computadores aprenderem padrões, a partir de dados, sem programação explícita. Dentro do AM, a regressão é uma técnica que permite prever saídas contínuas com base em um conjunto de dados rotulados. Esta técnica busca encontrar uma função que minimize o erro entre os valores reais e previstos, sendo amplamente aplicada em modelos preditivos para ajuste de funções [9], [10].

Existem diferentes tipos de AM, classificados em Aprendizado Supervisionado, Aprendizado Não Supervisionado e

o Aprendizado por Reforço. No supervisionado, o algoritmo aprende a partir de dados rotulados. No não supervisionado, identifica padrões em dados sem rótulos. Já no aprendizado por reforço, aprende por meio de recompensas e punições, buscando maximizar os ganhos [9], [10].

Nesse contexto, modelos globais e locais são abordagens distintas no AM para capturar padrões nos dados. Modelos globais ajustam uma única função para todo o conjunto de dados, buscando generalização ampla e sendo eficazes em problemas com relações simples ou quando há muitos dados disponíveis. Já os modelos locais dividem o espaço de entrada em regiões menores, ajustando submodelos específicos para capturar variações locais. São úteis quando a relação entre entrada e saída muda significativamente entre diferentes regiões no espaço dos dados [11].

As redes neurais artificiais são modelos de AM inspirados no funcionamento do sistema nervoso biológico, compostos por unidades de processamento denominadas neurônios, responsáveis por processar as informações recebidas. Matematicamente, a rede Perceptron Simples (PS), proposta por Frank Rosenblatt em 1958, recebe um conjunto de entradas $\{x_1, x_2, \dots, x_m\}$ por meio de estruturas denominadas dendritos, responsáveis pela condução dos sinais de entrada. Cada entrada é associada a um peso sináptico correspondente $\{w_{k1}, w_{k2}, \dots, w_{km}\}$, o qual modula a intensidade da informação transmitida.

A função do corpo celular consiste em realizar a soma ponderada dessas entradas, adicionando-se um valor de viés ou limiar de ativação b_k , conforme expresso por:

$$u_k = w_{k1}x_1 + w_{k2}x_2 + \dots + w_{km}x_m - b_k. \quad (4)$$

Em seguida, a saída do neurônio é obtida aplicando-se uma função de ativação $\varphi(u_k)$, responsável por determinar o nível de ativação do neurônio, resultando na saída $y_k = \varphi(u_k)$.

O processo de aprendizagem da rede neural consiste em ajustar os pesos do conhecimento adquirido $\mathbf{W}(t+1)$ e o limiar de ativação por meio da equação de aprendizado:

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \eta e(t) \mathbf{X}(t), \quad (5)$$

em que η é o passo de aprendizado, $\mathbf{W}(t)$ o conhecimento atual, t é o passo de iteração e $e(t)$ o erro entre a saída desejada y e a saída gerada $\hat{y}$ pela rede, dado por $e(t) = y - \hat{y}$ [10].

As funções de ativação são componentes que introduzem não linearidade ao modelo neural, permitindo que ele capture padrões complexos. Diversas funções de ativação têm sido propostas na literatura, cada uma com características e aplicações específicas. Entre as mais utilizadas, destacam-se a Tangente Hiperbólica ($\tanh$), definida no intervalo $[-1, 1]$ e possui a vantagem de centralizar os valores de saída em torno de zero.

De forma semelhante, a função Sigmoid Logística mapeia as entradas para o intervalo $[0, 1]$, sendo uma função contínua que também suaviza as transições entre os estados do neurônio. Entretanto, ambas as funções de ativação Sigmoid e $\tanh$ apresentam limitações relacionadas ao problema do gradiente, especialmente em redes profundas, onde os valores de gradiente podem se tornar muito pequenos, prejudicando o processo

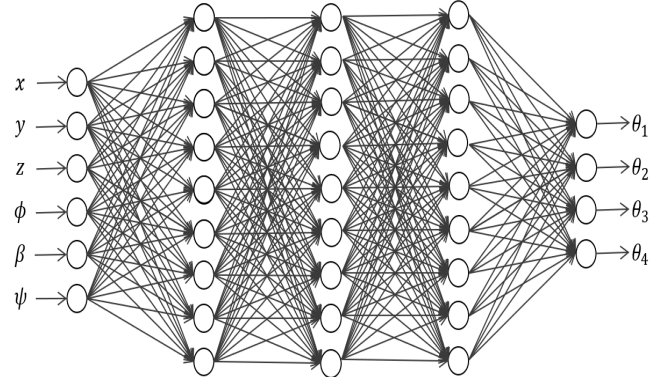


Figura 1. Modelo global de uma rede neural de múltiplas camadas

de retropropagação e, consequentemente, a atualização dos pesos sinápticos, processo conhecido como paralisia da rede. Por outro lado, a função de ativação *Rectified Linear Unit* (ReLU) lida melhor com o problema do gradiente próximos de zero, já que retorna zero para valores negativos e a própria entrada para valores positivos [12].

Redes neurais de múltiplas camadas (*Multi Layer Perceptron*, MLP), consistem em neurônios organizados em camadas densas, conforme ilustrado na Fig. 1, incluindo uma camada de entrada, responsável por captar as amostras $\mathbf{X}(t)$ e enviá-las para as camadas seguintes. Já as camadas ocultas recebem as saídas de outros neurônios, processa e propaga para as camadas seguintes. E por fim a camada de saída é responsável por fornecer a saída do modelo [13].

Para definir os pesos de uma MLP, pode-se utilizar o algoritmo de retropropagação do erro, no qual o erro de predição é minimizado por meio da atualização iterativa dos pesos sinápticos, propagando-se o erro desde a camada de saída até as camadas anteriores. Esse processo envolve inicialmente a propagação para frente, onde os dados de entrada percorrem as camadas ocultas até a camada de saída, produzindo a resposta da rede. Na etapa de propagação para trás, os erros dos neurônios de saída são calculados e propagados para as camadas anteriores, permitindo a atualização dos pesos sinápticos com base nos gradientes locais dos erros [10], [13]. A equação central do método baseia-se na atualização dos pesos $\mathbf{m}_k(t+1)$ pela regra:

$$\mathbf{m}_k(t+1) = \mathbf{m}_k(t) - \eta \frac{\partial J(t)}{\partial \mathbf{m}_k(t)}, \quad (6)$$

onde η é o passo de aprendizado, $\mathbf{m}_k(t)$ representa os pesos das conexões dos c neurônios para $k = 1, \dots, c$ e o gradiente do erro $J(t)$.

Dentre os tipos de aprendizado, o não supervisionado busca identificar padrões nos dados sem a necessidade de rótulos pré-definidos. No agrupamento (*clustering*), técnicas como o *K-means* particionam os dados em k clusters, onde elementos semelhantes são agrupados, e diferentes, separados. Essa abordagem é útil para segmentar dados em agrupamentos naturais em função da proximidade espacial no subespaço de características dos dados.

Este algoritmo destaca-se pela simplicidade conceitual e de implementação, complexidade computacional relativamente baixa e grande facilidade de adaptação e combinação com outros métodos de AM. O *K-means* inicia com k centróides escolhidos aleatoriamente e, iterativamente, realoca os pontos ao centróide mais próximo, recalculando as médias dos *clusters* até a convergência [10].

Esse algoritmo utiliza a distância euclidiana $d(x_i, \mu_j)$ para medir a proximidade entre os pontos x_i , definidos com p dimensões, e os centróides μ_j , sendo definida como:

$$d(x_i, \mu_j) = \sqrt{\sum_{n=1}^p (x_{in} - \mu_{jn})^2} \quad (7)$$

O objetivo do *K-means* é minimizar a função das variâncias intra-cluster, que mede a compactação dos grupos:

$$J = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2, \quad (8)$$

onde C_i é o conjunto de pontos do cluster i e μ_i seu centróide. O objetivo do algoritmo é minimizar J , o que implica em clusters mais compactos e bem definidos.

IV. MATERIAIS E MÉTODOS

O manipulador robótico utilizado neste trabalho foi desenvolvido como um módulo didático pelo grupo de pesquisa em robótica do IFCE [14], conforme ilustrado na Fig. 2, e fabricado por impressão 3D com filamento de *Polylactic Acid* (PLA). Sua estrutura mecânica é composta por cinco elos interligados: base, braço, antebraço, punho e garra. Este robô possui cinco juntas rotacionais, configurando uma cadeia cinemática de cinco GDL, caracterizando-o como um modelo 5R.

A implementação da cinemática direta foi baseada na notação DH, com o objetivo de modelar o comportamento cinemático direto do robô. Deste modo, realizou-se a enumeração sequencial das juntas e elos do robô, enquanto os sistemas de referência locais foram definidos e fixados em cada componente da estrutura robótica, conforme ilustrado na Fig. 2.

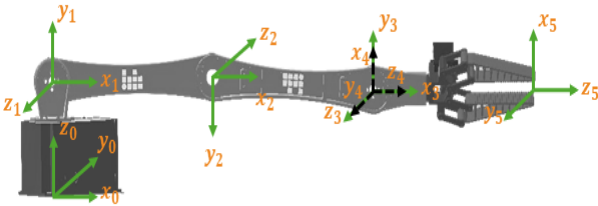


Figura 2. Sistema de coordenadas locais do manipulador

O mecanismo é definido por cinco graus de liberdade, associados as juntas rotativas que determinam o comportamento

do sistema por meio de cinco variáveis de juntas, denotadas por θ_i , onde $i = 1, \dots, 5$. A modelagem da cinemática direta do robô é fundamentada nos parâmetros DH, os quais descrevem as características geométricas de cada ligamento e articulação, conforme apresentado na Tabela I.

Tabela I
PARÂMETROS DE DH DO MANIPULADOR DIDÁTICO

i	θ (°)	a (cm)	d (cm)	α (°)
1	θ_1	0	10	90
2	θ_2	18	0	180
3	θ_3	18	0	-180
4	$\theta_4 + 90$	0	0	90
5	θ_5	0	18	0

Utilizando os parâmetros de DH, aplicou-se (2) para construir as cinco matrizes de transformação homogênea específicas, as quais relacionam os sistemas de coordenadas locais de cada elo com o elo anterior correspondente. Em seguida, essas matrizes de transformação homogênea são multiplicadas para obter uma matriz total, conforme a equação:

$$T_5^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4, \quad (9)$$

no qual estabelece a relação entre o sistema de referência da base e o sistema de referência do efetuador. Portanto, definindo os elementos da matriz (3), como:

$$r_{11} = s_{\theta_1} s_{\theta_5} - s_{\theta_2 - \theta_3 + \theta_4} c_{\theta_1} c_{\theta_5} \quad (10)$$

$$r_{12} = s_{\theta_1} c_{\theta_5} + s_{\theta_5} s_{\theta_2 - \theta_3 + \theta_4} c_{\theta_1} \quad (11)$$

$$r_{13} = c_{\theta_1} c_{\theta_2 - \theta_3 + \theta_4} \quad (12)$$

$$p_x = 18(c_{\theta_2} + c_{\theta_2 - \theta_3} + c_{\theta_2 - \theta_3 + \theta_4}) c_{\theta_1} \quad (13)$$

$$r_{21} = -s_{\theta_1} s_{\theta_2 - \theta_3 + \theta_4} c_{\theta_5} - s_{\theta_5} c_{\theta_1} \quad (14)$$

$$r_{22} = s_{\theta_1} s_{\theta_5} s_{\theta_2 - \theta_3 + \theta_4} - c_{\theta_1} c_{\theta_5} \quad (15)$$

$$r_{23} = s_{\theta_1} c_{\theta_2 - \theta_3 + \theta_4} \quad (16)$$

$$p_y = 18(c_{\theta_2} + c_{\theta_2 - \theta_3} + c_{\theta_2 - \theta_3 + \theta_4}) s_{\theta_1} \quad (17)$$

$$r_{31} = c_{\theta_5} c_{\theta_2 - \theta_3 + \theta_4} \quad (18)$$

$$r_{32} = -s_{\theta_5} c_{\theta_2 - \theta_3 + \theta_4} \quad (19)$$

$$r_{33} = s_{\theta_2 - \theta_3 + \theta_4} \quad (20)$$

$$p_z = 18s_{\theta_2} + 18s_{\theta_2 - \theta_3} + 18s_{\theta_2 - \theta_3 + \theta_4} + 10 \quad (21)$$

Para a aproximação da cinemática inversa é fundamental a geração dos padrões de treinamento dos algoritmos de AM. A modelagem da cinemática direta, representada em (9), permite mapear a relação entre o espaço operacional e o espaço das juntas do manipulador, criando uma base de dados que reflete o volume de trabalho e o comportamento das articulações do robô.

Para a criação da base de dados, a resolução do espaço das juntas foi definida de modo a refletir as limitações físicas do robô e, ao mesmo tempo, minimizar a geração de amostras redundantes, interpretadas como múltiplas soluções. Como resultado, foram estabelecidas faixas de ângulos para cada junta, levando em conta os limites de atuação de cada servomecanismo acoplado às juntas do robô, bem como o

Tabela II
RESOLUÇÃO DO ESPAÇO DAS JUNTAS

Junta	Faixa de ângulo (°)	Amostras por GDL	Resolução (graus/pontos)
1	$0 \leq \theta_1 \leq 120$	25	5
2	$0 \leq \theta_2 \leq 120$	11	12
3	$0 \leq \theta_3 \leq 120$	11	12
4	$0 \leq \theta_4 \leq 120$	11	12
5	$\theta_5 = 0$	1	1

número de amostras por grau de liberdade, considerando a quantidade mínima necessária de amostras no conjunto de ângulos das juntas para viabilizar a cinemática inversa com o uso de RNAs [15]. Portanto, definiu-se a resolução do espaço das juntas conforme apresentado na Tabela II.

A escolha das resoluções, ou seja, o passo utilizado para discretizar o movimento das juntas do robô, leva em consideração a geometria do manipulador articulado. Atribuiu-se um maior número de amostras ao θ_1 devido à sua maior influência nos eixos x e y , conforme pode ser observado em (13) e (17), enquanto as amostras restantes foram distribuídas de forma equilibrada entre os outros ângulos, que afetam mais o eixo z .

Optou-se ainda por mapear somente o conjunto de ângulos, excluindo a influência da junta θ_5 , visto que esta influencia apenas na orientação do efetuador final do manipulador. E dado que o objetivo do trabalho é desenvolver um algoritmo capaz de posicionar o efetuador no espaço de trabalho desejado sem considerar necessariamente sua orientação, a inclusão desta junta resultaria em redundâncias adicionais na base de dados [16].

Nesse sentido, a geração da base de dados deu-se pela aplicação das amostras do conjunto de ângulos definidos para o mapeamento das juntas na matriz (3) que modela a cinemática direta. Com isso, foram gerados 33275 dados brutos, valor resultante do produto das amostras por graus de liberdades de cada junta. Esses dados incluem amostras das saídas dos ângulos das juntas e os atributos das matrizes de transformação homogênea, representando as posições alcançadas pela garra do robô.

A qualidade dos dados é essencial para o desempenho dos algoritmos de AM, e a falta de pré-processamento adequado pode comprometer a fidelidade dos modelos e aumentar os custos computacionais [9]. Neste trabalho, o pré-processamento da base de dados envolveu a remoção de pontos no semi-plano negativo de z , devido as restrições impostas pelas limitações físicas do manipulador, e a eliminação de redundâncias, além da eliminação de redundâncias, entendidas como múltiplas soluções alcançáveis pelo sistema articulado. Ao todo, foram retiradas 4675 amostras cujo valor da coordenada z era negativo, restando 28600 amostras do conjunto inicial.

Posteriormente, as coordenadas cartesianas (x, y, z) foram selecionadas como atributos para o mapeamento da cinemática inversa do manipulador. No entanto, também foram incorporados os ângulos de orientação *Roll* (ϕ), *Pitch* (β)

e *Yaw* (ψ), obtidos a partir da decomposição dos elementos de rotação da matriz de DH [5]. A inclusão desses atributos justifica-se pela relevância dos parâmetros de orientação no contexto da cinemática completa do manipulador, uma vez que contribuem para a compreensão geral dos modelos de AM sobre o comportamento espacial do efetuador. Além disso, mesmo em aplicações nas quais o objetivo principal não é modelar explicitamente a orientação, observou-se a necessidade de definir esses parâmetros para assegurar o correto funcionamento do sistema robótico.

Em seguida, amostras com coordenadas (x, y, z) consideradas muito próximas a outras foram identificadas e removidas, utilizando a Distância Euclidiana (7) para calcular a proximidade entre os pontos em um raio de similaridade $r_s \leq 1$ cm. Para selecionar uma amostra do conjunto de pontos semelhantes, desenvolve-se uma função de custo que maximize as posições angulares θ_2 e θ_4 , e minimize as posições angulares θ_3 . Isso reduziu o número de amostras para 6890. Este conjunto reduzido de amostras foi utilizado para o treinamento dos modelos.

Inicialmente, foram explorados métodos lineares de regressão para solucionar o problema da cinemática inversa. No entanto, esses modelos não apresentaram desempenho satisfatório devido à natureza altamente não linear da relação entre o espaço operacional e o espaço das juntas, bem como a existência de múltiplas soluções. Essa limitação motivou a adoção de abordagens baseadas em redes neurais, capazes de capturar de forma mais eficiente as complexidades envolvidas na cinemática inversa do manipulador.

Com o objetivo de solucionar o problema da cinemática inversa do manipulador didático, foram avaliadas e implementadas duas abordagens de modelos de regressão baseadas nas arquiteturas de modelo neural global e local. Cada modelo tem a função de aprender o comportamento da cinemática inversa para os ângulos $[\theta_1, \theta_2, \theta_3, \theta_4]$ a partir dos vetores de atributos $[x, y, z, \phi, \beta, \psi]$, que contêm informações de posição e orientação no espaço de trabalho do robô.

O paradigma global, conforme ilustrado na Fig. 1, aproxima a cinemática inversa mapeando todo o espaço operacional alcançável pelo manipulador por meio de uma rede neural do tipo MLP configurada em um sistema de Múltiplas Entradas e Múltiplas Saídas (*Multiple Input, Multiple Output*, MIMO). Enquanto o paradigma local, ilustrado na Fig. 3, particiona o espaço operacional em sub-regiões, utilizando a técnica de agrupamento *k-means*, e, para cada uma dessas regiões, é definido uma MLP específica. Neste modelo, a seleção do modelo local adequado é realizada com base na amostra de entrada, sendo o algoritmo *K-Means* responsável por indicar a rede correspondente.

Para ambos os modelos de AM definiu-se um fluxo de etapas para o treinamento, validação e teste. Essencialmente, os dados da base de dados são particionados em conjunto de 70% das amostras para treinamento, 20% para validação e os 10% dos dados restante para testes. Para tanto, adotou-se a validação cruzada *Random Subsampling*, onde repete-se o processo de divisão aleatória em $n_r = 3$ iterações,

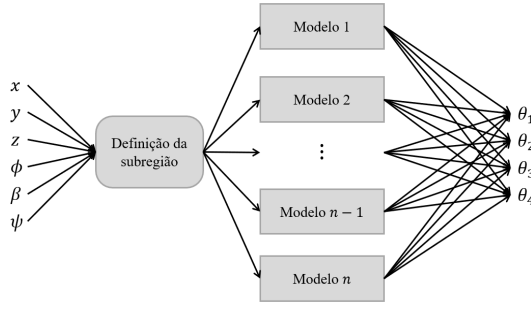


Figura 3. Modelos locais de redes MLP configurada em subsistemas MIMO

Tabela III
HIPERPARÂMETROS DO ESPAÇO DE BUSCA DOS MODELOS

Camadas	Faixa de Neurônios	Passo
1	[20, 6000]	5
2	[100, 500]	10
3	[100, 500]	35
4	[100, 400]	55
Funções de Ativação: ReLU, Tanh		

reservando as proporções de divisão dos dados, garantindo que os modelos fossem testados com diferentes subconjuntos mutuamente exclusivos [10]. Em seguida, é aplicada uma normalização em cada conjunto de dados gerados, com o objetivo de ajustar os dados para terem a mesma escala.

A seleção de hiperparâmetros dos modelos locais e globais foi realizada através da técnica *Random Search*, explorando 2000 combinações diferentes de hiperparâmetros. Essa técnica apresenta vantagens em termos de eficiência computacional, já que em vez de testar exaustivamente todas as combinações possíveis dos hiperparâmetros de um modelo, apenas seleciona aleatoriamente um subconjunto de combinações, o que permite encontrar boas soluções dado o tempo [17].

Os hiperparâmetros fixos definidos para os modelos incluem um número máximo de 1000 épocas, um erro tolerado de 1×10^{-4} , um critério de parada antecipada após 10 iterações sem melhora, um fator de *momentum* de 0,9 e uma regularização $\alpha = 1 \times 10^{-4}$. Para o modelo global, o espaço de busca variou em arquiteturas de MLP com até 4 camadas (cujas faixas de neurônios por camada é mostrada na Tabela III), e funções de ativação ReLU e tanh. Já os modelos locais tiveram uma configuração do espaço de busca semelhante a Tabela III, porém nas MLPs locais limitou-se o número de camadas ocultas, variando entre 1 a 3 camadas.

A técnica para avaliar os modelos de regressão adotada foi o MSE, dado pela equação:

$$\text{MSE}(\hat{f}) = \frac{1}{n} \sum_{i=1}^n (\xi_i)^2, \quad (22)$$

esse critério calcula a média dos quadrados das diferenças entre os valores previstos pelo modelo $\hat{f}(x_i)$ e os valores reais y_i observados nos dados [10], portanto $\xi_i = y_i - \hat{f}(x_i)$.

Essa métrica foi calculada em duas abordagens, a primeira considera o espaço das juntas $\text{MSE}(\theta_i)$, analisando a saída

esperada e a saída estimada, ambas desnormalizados, dos ângulos das articulações do manipulador robótico. Na segunda abordagem, no espaço operacional $\text{MSE}(\mathbb{R}^3)$, compara a posição cartesiana desejada com a posição reconstruída a partir dos ângulos desnormalizados preditos pelo modelo na modelagem da cinemática direta.

Para os modelos de agrupamento, onde não há rótulos definidos, métricas internas e externas avaliam a qualidade das divisões. O coeficiente de silhueta mede a coesão e separação dos grupos no método *k-means*. Já a inércia, utilizada no método do cotovelo, analisa a soma das distâncias dos pontos ao centróide, ajudando a determinar o número ideal de *clusters*. A escolha de k é feita no ponto onde a redução da inércia começa a estabilizar, garantindo uma segmentação equilibrada dos dados [10].

O coeficiente de silhueta $s(i)$ avalia a separação dos grupos gerados para cada ponto i , dado pela largura média de silhueta:

$$s = \frac{1}{N} \sum_{i=1}^N \frac{b(i) - a(i)}{\max(a(i), b(i))}, \quad (23)$$

onde $a(i)$ é a distância média entre o ponto i e todos os demais pontos do mesmo *cluster*, e $b(i)$ é a menor distância média entre o ponto i e todos os pontos do *cluster* mais próximo ao qual i não pertence. O coeficiente de silhueta varia entre $[-1, 1]$. Valores próximos de 1 indicam que o ponto está bem ajustado ao seu próprio grupo e longe dos grupos vizinhos, enquanto valores próximos de -1 indicam que o ponto foi mal classificado [10]. Por outro lado, o método do cotovelo, do inglês *elbow method*, é uma técnica usada para identificar o número ideal de grupos k , analisando a soma das variâncias internas dos *clusters*, também chamada de inércia. A inércia é a soma das distâncias quadradas das observações ao seu centróide, como indicado em (8).

V. RESULTADOS E DISCUSSÕES

A validação do modelo de cinemática direta foi conduzida por meio da parametrização de DH, plicando-se a um modelo de braço robótico desenvolvido com a biblioteca *Robotics Toolbox for Python*, em comparação com um manipulador físico real. Para isso, foram realizados experimentos utilizando amostras compostas por posições angulares previamente conhecidas e de fácil verificação. Essas configurações foram aplicadas tanto no modelo simulado quanto nos servomecanismos do sistema robótico físico.

Durante os testes, a posição cartesiana teórica do efetuador foi obtida a partir do conjunto de ângulos, utilizando os recursos da biblioteca para calcular e visualizar a configuração espacial do manipulador simulado. No caso do manipulador físico, sua composição espacial foi observada manualmente, e a posição do efetuador foi estimada com base em medições diretas.

A simulação ilustrada na Fig. 4 evidenciou a precisão do modelo teórico, validando a posição final do efetuador (46; 9; 0; 28) a partir dos ângulos (0; 50; 45; 0; 0) fornecidos. Em [18], também são apresentados testes iniciais com a

cinemática direta do manipulador robótico embarcado em um gênio digital. Os resultados demonstraram que a modelagem via notação DH foi eficaz na representação do comportamento esperado do sistema, confirmando a adequação do modelo para aplicações em simulações de tarefas de manipulação robótica.

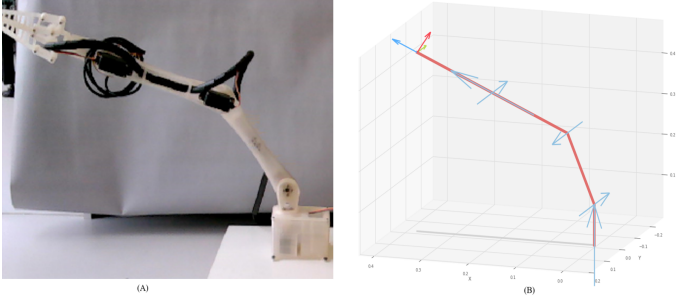


Figura 4. Manipulador robótico (A) estrutura física (B) simulação da modelagem

A geração da base de dados considerou as restrições da cinemática direta e os limites físicos das juntas, resultando em um conjunto de treinamento otimizado. A decisão de fixar θ_5 simplificou a geração dos padrões de treinamento sem afetar a precisão, e a distribuição diferenciada das amostras permitiu uma cobertura eficiente do espaço cartesiano. Foram aplicados filtros para remover amostras inatingíveis e redundantes, reduzindo a 20,7% o volume de dados sem comprometer a diversidade das amostras. Como resultado, a base final com 6890 amostras garantiu um conjunto mais esparsos, conforme ilustrado na Fig. 5, melhorando o aprendizado dos modelos e reduzindo o tempo de processamento.

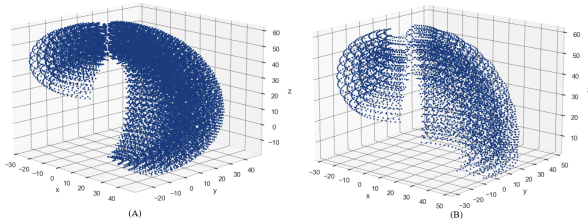


Figura 5. Volume de trabalho do manipulador didático (A) base de dados total (B) base de dados com as redundâncias reduzidas.

A otimização dos hiperparâmetros, realizada por *Random Search* com validação cruzada, permitiu identificar configurações que maximizassem a precisão na predição dos ângulos do manipulador robótico. O modelo global, configurado para prever simultaneamente os ângulos em todo o volume de trabalho, teve sua melhor arquitetura definida com quatro camadas ocultas, (320, 375, 265, 155) neurônios, e função de ativação ReLU. Esse modelo alcançou uma média do $MSE(\theta_i) = 2,6^\circ$ no espaço das juntas, e uma média do $MSE(\mathbb{R}^3) = 1,53 \times 10^{-1}$ cm no espaço operacional. As Tabelas IV e V mostram os erros, respectivamente, por junta e por coordenada no plano cartesiano.

A seleção do número ideal de *clusters* k para o modelo local foi determinada com base na análise das métricas de

Tabela IV
ERROS $MSE(\theta_i)$ DOS ÂNGULOS DAS JUNTAS PREDITOS PELO MODELO GLOBAL

Junta	Treinamento ($^\circ$)	Teste ($^\circ$)
θ_1	$9,38 \times 10^{-1}$	$9,77 \times 10^{-1}$
θ_2	1,69	1,75
θ_3	4,44	4,22
θ_4	3,08	3,39

Tabela V
ERROS DO ESPAÇO OPERACIONAL $MSE(\mathbb{R}^3)$ PREDITOS PELO MODELO GLOBAL

Coordenadas	Treinamento (cm)	Teste (cm)
x	$2,372 \times 10^{-1}$	$2,406 \times 10^{-1}$
y	$1,763 \times 10^{-1}$	$1,810 \times 10^{-1}$
z	$1,842 \times 10^{-1}$	$1,939 \times 10^{-1}$

inércia e coeficiente de silhueta. A análise da inércia, conforme ilustrado no gráfico da Fig. 6, indicou uma queda significativa entre $k = 2$ e $k = 4$, com ganhos marginais após esse ponto, sugerindo que $k = 4$ seria um valor adequado. O coeficiente de silhueta, indicado na Fig. 6, atingiu seu pico nesse mesmo valor, indicando a melhor separação e coesão dos grupos.

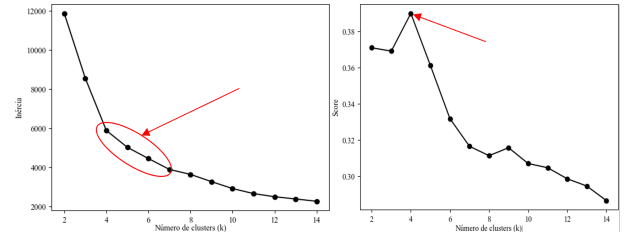


Figura 6. Método do cotovelo e método da silhueta aplicados no espaço operacional do manipulador

A partir da definição de k , o algoritmo K-means foi aplicado aos dados, particionando estes conforme a Fig. 7.

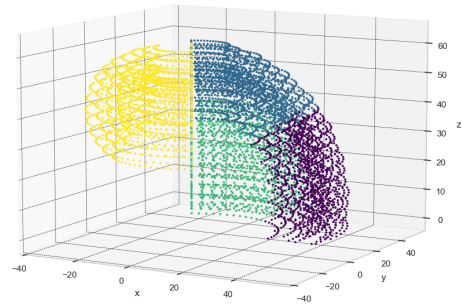


Figura 7. Espaço operacional do manipulador dividido em sub regiões

Após a definição das quatro regiões do espaço operacional do robô, foi treinado um modelo de regressão do tipo rede MLP para cada região. A configuração das arquiteturas dos modelos locais foram selecionadas na etapa de otimização de hiperparâmetros, e foram definidos conforme ilustrado na Tabela VI.

Tabela VI
ARQUITETURAS DAS REDES NEURAIS SELECIONADOS PARA CADA
MODELO LOCAL

Modelo	Neurônios	Ativação
1	(380, 240, 485)	ReLU
2	(485, 380, 380)	ReLU
3	(205, 415, 450)	ReLU
4	(415, 310, 485)	ReLU

Tabela VII
ERROS $MSE(\theta_i)$ DOS ÂNGULOS DAS JUNTAS PREDITOS PELOS MODELOS
LOCAIS

Junta	Treinamento ($^\circ$)	Teste ($^\circ$)
θ_1	$2,34 \times 10^{-1}$	$5,24 \times 10^{-1}$
θ_2	$4,87 \times 10^{-1}$	$8,44 \times 10^{-1}$
θ_3	1,44	2,55
θ_4	1,06	1,41

Tabela VIII
ERROS DO ESPAÇO OPERACIONAL $MSE(\mathbb{R}^3)$ PREDITOS PELOS MODELOS
LOCAIS

Coordenadas	Treinamento (cm)	Teste (cm)
x	$4,24 \times 10^{-2}$	$5,65 \times 10^{-2}$
y	$4,62 \times 10^{-2}$	$6,26 \times 10^{-2}$
z	$5,94 \times 10^{-2}$	$7,19 \times 10^{-2}$

Os modelos locais alcançaram uma média do $MSE(\theta_i) = 1,33^\circ$, conforme ilustrado na Tabela VII, no espaço das juntas e para o espaço operacional o erro médio $MSE(\mathbb{R}^3) = 6,33 \times 10^{-2}$ cm, conforme ilustrado na Tabela VIII.

Comparando as duas abordagens, mesmo com MLPs contendo menos camadas ocultas, o modelo de regressão local obteve melhores resultados em relação ao modelo de regressão global.

VI. CONCLUSÃO

Neste trabalho, o problema da cinemática inversa de um manipulador robótico foi abordado através de dois paradigmas de aprendizado de máquina: modelos globais e locais, ambos baseados em redes neurais MLP. Após a geração de amostras a partir da cinemática direta do manipulador, os modelos foram treinados para solucionar o problema em questão. Erros médios quadráticos, no espaço cartesiano, da ordem de 10^{-1} cm para os modelos globais e 10^{-2} cm para os modelos locais evidenciam a viabilidade de ambas as abordagens. Em breve, estes modelos serão embarcados em um processador para gerar as trajetórias, no espaço cartesiano, do manipulador estudado.

REFERÊNCIAS

- [1] P. Corke, *Robotics, Vision and Control*. Berlin Heidelberg: Springer, 2017.
- [2] T. Ogawa e H. Kanada, "Solution for Ill-posed inverse kinematics of robot arm by network inversion," *Journal of Robotics*, vol. 2010, 2010.
- [3] J. J. Craig, *Introduction to Robotics*. USA: Pearson Educacion, 2006.
- [4] D. B. Melo, "Algoritmos de aprendizagem para aproximação da cinemática inversa de robôs manipuladores: um estudo comparativo," Ph.D. dissertation, Universidade Federal do Ceará, 2015." unpublished.
- [5] J. Demby, Y. Gao, e G. N. DeSouza, "A study on solving the inverse kinematics of serial robots using artificial neural network and fuzzy neural network," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2019, pp. 1–6.
- [6] R. Gao, "Inverse kinematics solution of Robotics based on neural network algorithms," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 12, pp. 6199–6209, 2020.
- [7] H. Í. P. Fontinele, "Modelos locais para aproximação da cinemática inversa de robôs redundantes: um estudo comparativo," Ph.D. dissertation, Universidade Federal do Ceará, 2015.
- [8] J. Denavit e R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *Journal of Applied Mechanics*, vol. 22, pp. 215–221, 1955.
- [9] S. Russell e P. Norvig, *Artificial Intelligence: A Modern Approach*, 3ª ed. Boston, MA, USA: Pearson, 2016.
- [10] K. Faceli, A. C. Lorena, J. Gama, e A. C. P. L. F. de Carvalho, *Inteligência Artificial: uma abordagem de aprendizado de máquina*. Rio de Janeiro: Editora LTC, 2011.
- [11] L. Bottou e V. Vapnik, "Local learning algorithms," *Neural Computation*, vol. 4, no. 6, pp. 888–900, 1992.
- [12] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
- [13] D. E. Rumelhart, G. E. Hinton, e R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [14] Robótica IFCE, "Projeto de robótica em instituições de ensino fundamental e médio," 2022. Disponível em: <https://robotica.ifce.edu.br/>. Acessado em: 14-09-2024
- [15] D.-W. Lim e Y.-K. Lee, "On the number of training samples for inverse kinematics solutions by artificial neural networks," in *2019 16th International Conference on Ubiquitous Robots (UR)*, Jeju, South Korea, 2019, pp. 61–64.
- [16] R. F. Nunes, "Mapeamento da cinemática inversa de um manipulador robótico utilizando redes neurais artificiais configuradas em paralelo," Mestrado, Universidade Estadual Paulista (Unesp), Brasil, 2016.
- [17] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. 2, pp. 281–305, 2012.
- [18] L. G. M. Lima, J. M. S. Ferreira, D. N. Coelho, R. R. De Melo, and W. S. Da Silva, "Development and validation of a digital twin for a 5-revolute joint robotic manipulator," in *Proc. 2025 Brazilian Conference on Robotics (CROS)*, vol. 1, pp. 1–6, 2025.